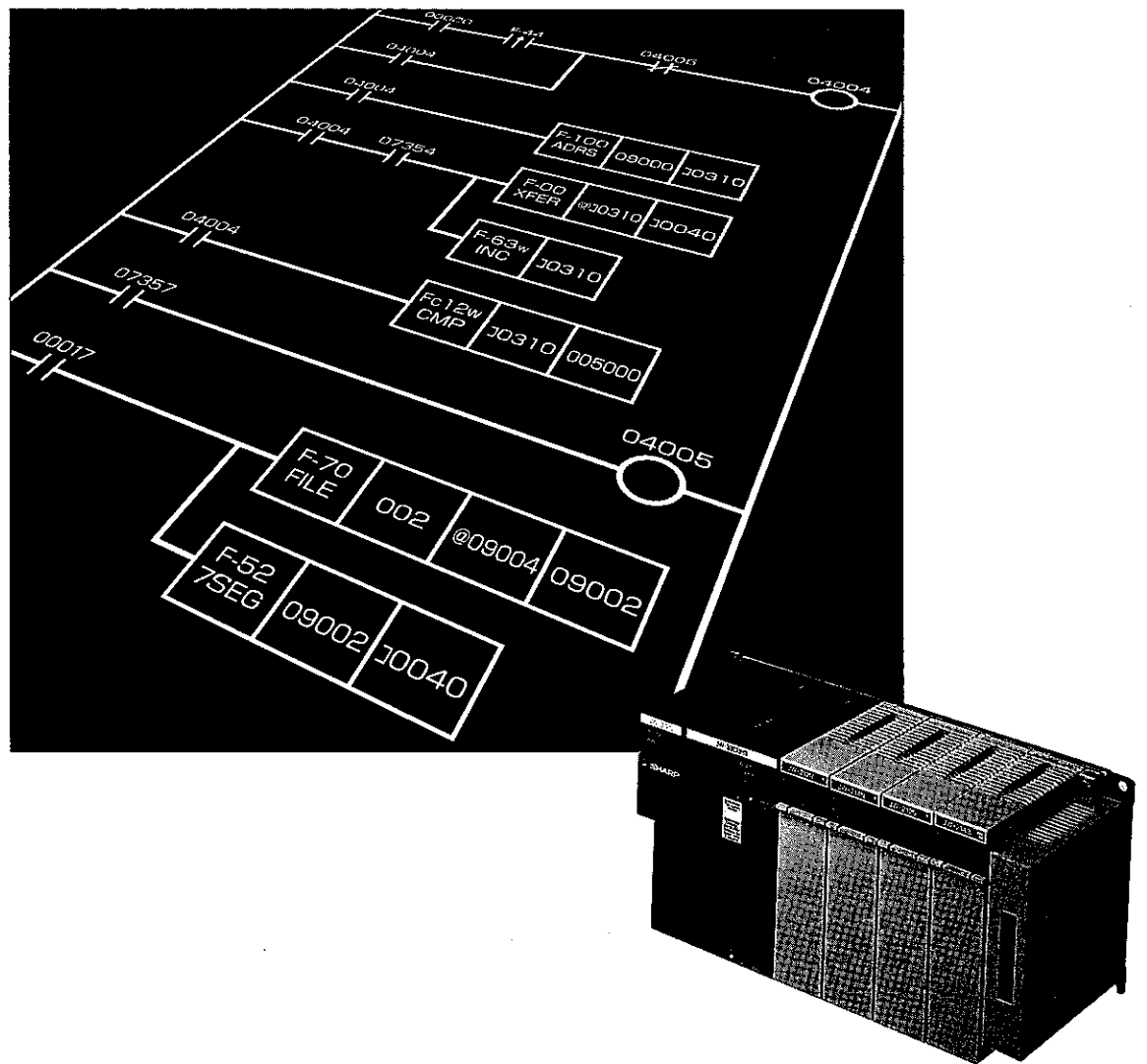


シャーププログラマブルコントローラ

ニューサテライト JW30H

プログラミングマニュアル・ラダー命令編



このたびは、シャープ プログラマブルコントローラJW30Hをお買いあげいただき、まことにありがとうございます。
います。

本書(プログラミングマニュアル・ラダー命令編)は、JW30Hの命令語について説明しています。

ご使用前に、この「プログラミングマニュアル・ラダー命令編」及び「ユーザーズマニュアル・ハード編」、
JW30Hのシステムを構成するユニットに付属の「取扱説明書」をよくお読みいただき、十分理解して、正しく使
用してください。

なお、この「プログラミングマニュアル・ラダー命令編」はJW30Hの「取扱説明書」、「ユーザーズマニュアル・
ハード編」とともに必ず保存してください。万一ご使用中にわからないことが生じたとき、きっとお役に立ち
ます。

おねがい

- ・ 本書の内容については十分注意して作成しておりますが、万一ご不審な点、お気付きの
 ことがありましたらお買いあげの販売店、あるいは当社サービス会社までご連絡ください。
- ・ 本書の内容の一部又は全部を無断で複製することは禁止しています。
- ・ 本書の内容は、改良のため予告なしに変更することがありますので、あらかじめご了承ください。

目 次

第1章 概 要	1・1
第2章 データメモリ	2・1
2 - 1 ファイル番号とファイルアドレス 2・1	
2 - 2 データメモリの種類 2・2	
〔1〕データメモリの種類 2・3	
〔2〕データメモリの機能 2・4	
〔3〕特殊リレー 2・5	
〔4〕TMR、CNT、MDのデータ格納領域 2・6	
〔5〕リレー領域のバイトアドレス 2・8	
〔6〕レジスタ領域の予約領域 2・8	
2 - 3 ファイル0のメモリマップ 2・11	
〔1〕バイトアドレス順 2・11	
〔2〕ファイルアドレス順 2・14	
2 - 4 入出力リレーの割付 2・15	
〔1〕I/O登録の種類 2・15	
〔2〕最大入出力点数と入出力リレーの割付 2・16	
〔3〕入出力リレー番号の割付例 2・17	
〔4〕サポートツールによるI/O登録 2・17	
第3章 パラメータメモリ	3・1
3 - 1 特殊I/Oユニットのパラメータメモリ 3・1	
3 - 2 オプションユニットのパラメータメモリ 3・2	
第4章 プログラムメモリ	4・1
第5章 システムメモリ	5・1
5 - 1 システムメモリ 5・1	
5 - 2 システムメモリの内容 5・3	
第6章 コントロールユニットの動作	6・1
6 - 1 運転サイクル 6・1	
〔1〕動作フローチャート 6・1	
〔2〕パワーON処理 6・2	
〔3〕スキャンサイクル 6・3	
6 - 2 自己診断 6・8	
〔1〕自己診断内容 6・9	
〔2〕停止出力 6・10	
〔3〕特殊リレー 6・10	
〔4〕異常コード 6・11	
〔5〕異常時の出力ユニットのON / OFF状態 6・11	
第7章 命令語一覧	7・1
〔1〕番号順 7・1	
〔2〕動作による分類 7・12	
〔3〕命令語処理時間 7・17	
第8章 基本命令の説明	8・1
8 - 1 ビット処理部の動作 8・1	
8 - 2 各基本命令について 8・2	
〔1〕STR / OUT 8・2	
〔2〕STR NOT 8・2	
〔3〕AND 8・3	
〔4〕AND NOT 8・3	
〔5〕OR 8・3	
〔6〕OR NOT 8・4	

〔 7 〕 AND STR	8・4
〔 8 〕 OR STR	8・5
〔 9 〕 TMR (タイマ命令)	8・6
〔 10 〕 CNT (カウンタ命令)	8・7
〔 11 〕 MD (メンテナンスディスプレイ)	8・9
8 - 3 ラダー設計に関する留意事項	8・12
〔 1 〕 リレー盤用ラダー図から書換えを必要とする回路	8・12
〔 2 〕 入出力一括処理方式	8・13
〔 3 〕 プログラム順序による影響	8・14
〔 4 〕 プログラムの簡略化	8・15
〔 5 〕 直並列回路のプログラム	8・15
第 9 章 応用命令の説明	9・1
9 - 1 応用命令に関する留意事項	9・1
〔 1 〕 数値の表現方法	9・1
〔 2 〕 ソースとデスティネーション	9・3
〔 3 〕 間接アドレス指定	9・3
〔 4 〕 応用命令とスタックレジスタ	9・4
〔 5 〕 演算実行条件	9・7
〔 6 〕 データ処理命令とフラグ	9・7
〔 7 〕 倍長演算	9・10
〔 8 〕 符号付演算	9・12
〔 9 〕 データメモリのブロックと基準アドレス	9・13
〔 10 〕 ファイル番号を指定する命令	9・14
〔 11 〕 数値信号の入出力方法	9・15
9 - 2 各応用命令について	9・19
各応用命令の参照ページは7・1ページから7・11ページを参照願います。	

第 1 章 概 要

プログラマブルコントローラ ニューサテライトJW30Hは、RISC型CPUを採用することにより、中規模クラスながら大規模機以上の基本命令処理速度を実現したプログラマブルコントローラです。プログラミングについて以下の特長があります。

特 長

1. 基本命令処理速度 (38ナノ秒 / 命令) を実現

RISC型CPUの採用で、10K語 (基本命令のみ) のプログラムでも約1ミリ秒で演算することができます。これにより、ラインスピードの向上はもとより、繰り返し精度向上による加工精度アップが図れます。

2. オプションユニットとのデータ交換も高速

処理速度の高速化に対応し、コントロールユニットとオプションユニットとのデータ交換も当社従来比約5倍に高速化しました。

3. 115.2Kビット / s対応の高速シリアルポート

パソコン、サポートツールとのデータ転送時間を大幅に短縮しました。

4. 情報処理量の増大対応可能な大容量メモリを搭載

プログラムメモリ63K語、リレー点数3万点、レジスタ25Kバイト (ファイル1含む) のメモリを準備、さらにファイルレジスタ1984Kバイトもご使用いただけます。(JW-33CUH3を使用時)

5. 最大8ラック、最大入出力3072点の制御が可能

最大8ラック、合計64台の入出力・特殊I/Oユニットが実装できるため、最大制御入出力点数3072点が可能です。(32点/64点ユニット各々32台実装時：JW-33CUH1/H2/H3の場合)

6. プログラムの設計効率を大幅に改善

従来1枚のロール紙だったプログラムを、機能毎に分割する構造化プログラム機能により、ブロック毎の管理 (ページ管理) が可能になります。また、作成したプログラムをライブラリとして保存して再使用できるライブラリ機能も搭載しています。(多機能プログラマ、ラダーソフトの取扱説明書をご参照下さい)

7. 3つの数値表現方式 (8進法 / 10進法 / 16進法) からの選択が可能、応用命令もさらに充実

データメモリ、応用命令の定数やプログラムアドレス等は、ユーザーの慣れた標記で表現できますので、他社モデルをお使いの方でも違和感なくご使用いただけます。(本書では初期値の進数で記載しています。) また、応用命令も従来機種JW20H、JW50H / 70H / 100Hの命令を包含、更にリレー出力付比較命令、符号付四則演算命令等を追加しました。

コントロールユニットについて

JW30HコントロールユニットJW-31CUH1、JW-32CUH1、JW-33CUH1/H2/H3の主な性能仕様を示します。

項目	JW-31CUH1	JW-32CUH1	JW-33CUH1	JW-33CUH2	JW-33CUH3
最大入出力点数	512点	1024点	3072点	3072点	3072点
プログラム容量	7.5K語	15.5K語/ 31.5k語(*1)	31.5K語	63K語	63K語
データメモリ	レジスタ(*2)	9Kバイト	25Kバイト	25Kバイト	25Kバイト
	ファイルメモリ	なし	64Kバイト/ 32Kバイト(*1)	128Kバイト	448Kバイト
時計機能	なし	あり			
コミュニケーションポート伝送速度	なし	115200/57600/38400/19200/9600/4800/2400 /1200ビット/s			
命令処理速度	基本命令 : 0.038 μs (従来機の24%アップ) 転送命令 (F-00) : 1.22 μs				
最大ラック数	8ラック (I/Oバス拡張アダプタ使用時)				
I/O増設ケーブル総延長距離	最大50m (ベース間最大50m)				
オプションユニットとのデータ交換時間	基本ベースJW-34KB/36KB/38KBを使用すると従来機の約5倍の速度				

(*1) JW-32CUH1は、ユニットのスイッチ (SW 2) の設定により、プログラム容量 / ファイル容量 (ファイル 2) を選択できます。(15.5K語 / 64Kバイト、または31.5K語 / 32Kバイト)

(*2) レジスタには、応用命令で直接指定できるファイル 1 のレジスタ (16Kバイト) を含みます。
(2・2ページ参照)

参 考

JW30HコントロールユニットJW-31CUH/32CUH/33CUH(従来機)の主な性能仕様は次のとおりです。
 なお、本書ではJW-31CUH/32CUH/33CUHの説明を併記しています。

項目	JW-31CUH	JW-32CUH	JW-33CUH	
最大入出力点数	512点	1024点	2048点	
プログラム容量	7.5K語	15.5K語	31.5K語	
データメモリ	レジスタ(*2)	9Kバイト	25Kバイト	25Kバイト
	ファイルメモリ	なし	64Kバイト	128Kバイト
時計機能	なし	あり		
コミュニケーションポート伝送速度	なし	19200/9600/4800/2400 /1200/600ビット/s		
命令処理速度	基本命令 : 0.05 μs 転送命令 (F-00) : 1.60 μs			
最大ラック数	4ラック (*3)			
I/O増設ケーブル総延長距離	最大14m (*3) (ベース間最大10m)			
オプションユニットとのデータ交換時間	基本ベースJW-34KB/36KB/38KBを使用しても従来と同じ速度			

(*2) レジスタには、応用命令で直接指定できるファイル 1 のレジスタ (16Kバイト) を含みます。
 (*3) 従来機は I / O 拡張アダプタ (JW-31EA/32EA) を使用できません。

第 2 章 データメモリ

2 - 1 ファイル番号とファイルアドレス

データメモリは、用途別にファイル番号を割り付けています。各ファイルはファイルアドレスで管理されており、ファイルアドレスは応用命令の間接アドレス設定時、キーブリレー領域の設定時等に使用します。

各コントロールユニットは下記ファイル番号のメモリを標準実装しています。

ファイル番号 (HEX)	ファイルアドレス (OCT)	容 量	JW30Hコントロールユニット [<input type="checkbox"/> 実装範囲]				
			JW-31CUH/H1	JW-32CUH/H1	JW-33CUH/H1	JW-33CUH2	JW-33CUH3
0	000000 ~ 035777	15Kバイト					
1	000000 ~ 037777	16Kバイト					
2	000000 ~ 177777	64Kバイト		1			
3	000000 ~ 177777	64Kバイト					
10	000000 ~ 177777	64Kバイト					
11	000000 ~ 177777	64Kバイト					
12	000000 ~ 177777	64Kバイト					
13	000000 ~ 177777	64Kバイト					
14	000000 ~ 177777	64Kバイト					
15	000000 ~ 177777	64Kバイト					
16	000000 ~ 177777	64Kバイト					
17	000000 ~ 177777	64Kバイト					
18	000000 ~ 177777	64Kバイト					
19	000000 ~ 177777	64Kバイト					
1A	000000 ~ 177777	64Kバイト					
1B	000000 ~ 177777	64Kバイト					
1C	000000 ~ 177777	64Kバイト					
1D	000000 ~ 177777	64Kバイト					
1E	000000 ~ 177777	64Kバイト					
1F	000000 ~ 177777	64Kバイト					
20	000000 ~ 177777	64Kバイト					
21	000000 ~ 177777	64Kバイト					
22	000000 ~ 177777	64Kバイト					
23	000000 ~ 177777	64Kバイト					
24	000000 ~ 177777	64Kバイト					
25	000000 ~ 177777	64Kバイト					
26	000000 ~ 177777	64Kバイト					
27	000000 ~ 177777	64Kバイト					
28	000000 ~ 177777	64Kバイト					
29	000000 ~ 177777	64Kバイト					
2A	000000 ~ 177777	64Kバイト					
2B	000000 ~ 177777	64Kバイト					
2C	000000 ~ 177777	64Kバイト					

1 JW-32CUH1は、ユニット内部スイッチにより、ファイル2の容量を64Kバイト（000000 ~ 177777）または、32Kバイト（000000 ~ 077777）に設定します。JW-32CUHのファイル2の容量は64Kバイト（000000 ~ 177777）固定です。

各ファイルには下記の用途があります。

ファイル番号	用途	使い方
0	リレー タイマ (TMR) カウンタ(CNT)	基本命令での直接アクセス (読出 / 書込)
	レジスタ	応用命令での直接 / 間接アクセス (読出 / 書込) 2
	構造化プログラム用 メモリ (59000 ~ 89777 E0000 ~ E5777)	JW-31CUH/H1使用時、ラダーソフトによる構造化 プログラム作成時に使用 4
1	レジスタ	応用命令での直接 / 間接アクセス (読出 / 書込) 2
	コメントメモリ	ラダーソフトによるコメント登録時に使用 3
2	レジスタ	応用命令での間接アクセス (読出 / 書込) 2
	コメントメモリ	ラダーソフトによるコメント登録時に使用 3
	構造化プログラム用 メモリ	JW-32CUH/H1使用時、ラダーソフトによる構造化 プログラム作成時に使用 4
3	レジスタ	応用命令での間接アクセス (読出 / 書込) 2
	コメントメモリ	ラダーソフトによるコメント登録時に使用 3
	構造化プログラム用 メモリ	JW-33CUH/H1使用時、ラダーソフトによる構造化 プログラム作成時に使用 4
10、11、12、13	レジスタ	応用命令での間接アクセス (読出 / 書込) 2
	コメントメモリ	ラダーソフトによるコメント登録時に使用 3
14	レジスタ	応用命令での間接アクセス (読出 / 書込) 2
	コメントメモリ	ラダーソフトによるコメント登録時に使用 3
	構造化プログラム用 メモリ	JW-33CUH2使用時、ラダーソフトによる構造化 プログラム作成時に使用 4
15、16、17、18、19、 1A、1B、1C、1D、1E、 1F、20、21、22、23、 24、25、26、27、28、 29、2A、2B	レジスタ	応用命令での間接アクセス (読出 / 書込) 2
	コメントメモリ	ラダーソフトによるコメント登録時に使用 3
2C	レジスタ	応用命令での間接アクセス (読出 / 書込) 2
	コメントメモリ	ラダーソフトによるコメント登録時に使用 3
	構造化プログラム用 メモリ	JW-33CUH3使用時、ラダーソフトによる構造化 プログラム作成時に使用 4

- 1 ファイル0の詳細は、2・3～2・14ページを参照してください。
- 2 応用命令での直接アクセスとは、応用命令のソース、デスティネーションにレジスタ番号（ファイル0の場合）またはファイルアドレス（ファイル1の場合）を設定してデータの読出 / 書込を行うことです。間接アクセスとは、ソース、デスティネーションで間接アドレス指定により指定されたレジスタのデータの読出 / 書込を行うことです。（9・3ページ参照）
- 3 ファイル1、2、3、10～2Cをコメントメモリとして使用する場合は、システムメモリ#220、#224、#225に使用領域を設定します。（5・10ページ参照）
- 4 構造化プログラムについては、「構造化プログラミングマニュアル」を参照してください。

注意 各ファイル番号のメモリは、レジスタ、コメントメモリ、構造化プログラム用メモリのいずれか1つの用途に使用できます。同時に2つ以上の用途には使用できません。

2 - 2 データメモリ(ファイル0)の種類

(1) データメモリの種類

ファイル0のメモリにはリレー、TMR/CNT、レジスタ等の種類があります。

種 類	容 量	リレー番号(OCT)	バイトアドレス(OCT)	ファイルアドレス(OCT)
リレー 1	30720点 (3840バイト)	00000 ~ 15777	コ0000 ~ コ1577	000000 ~ 001577
		20000 ~ 75777	コ2000 ~ コ7577	030000 ~ 035577
TMR / CNTの接点	1024点 (128バイト)	TまたはC 0000 ~ 0777	————	001600 ~ 001777
		TまたはC 1000 ~ 1777	————	035600 ~ 035777
TMR / CNT / MD の現在値	2048バイト	————	b0000 ~ b1777	002000 ~ 003777
			b2000 ~ b3777	026000 ~ 027777
ファイル0 レジスタ 9 Kバイト	512バイト	————	09000 ~ 09777	004000 ~ 004777
	512バイト	————	19000 ~ 19777	005000 ~ 005777
	512バイト	————	29000 ~ 29777	006000 ~ 006777
	512バイト	————	39000 ~ 39777	007000 ~ 007777
	512バイト	————	49000 ~ 49777	010000 ~ 010777
	512バイト	————	59000 ~ 59777	011000 ~ 011777
	512バイト	————	69000 ~ 69777	012000 ~ 012777
	512バイト	————	79000 ~ 79777	013000 ~ 013777
	512バイト	————	89000 ~ 89777	014000 ~ 014777
	512バイト	————	99000 ~ 99777	015000 ~ 015777
	512バイト	————	E0000 ~ E0777	016000 ~ 016777
	512バイト	————	E1000 ~ E1777	017000 ~ 017777
	512バイト	————	E2000 ~ E2777	020000 ~ 020777
	512バイト	————	E3000 ~ E3777	021000 ~ 021777
	512バイト	————	E4000 ~ E4777	022000 ~ 022777
	512バイト	————	E5000 ~ E5777	023000 ~ 023777
	512バイト	————	E6000 ~ E6777	024000 ~ 024777
	512バイト	————	E7000 ~ E7777	025000 ~ 025777

1 リレー内には次の固定エリアがあります。

種 類	容 量	リレー番号(OCT)	バイトアドレス(OCT)	ファイルアドレス(OCT)
特殊リレー	64点(8バイト)	07300 ~ 07377	コ0730 ~ コ0737	000730 ~ 000737
オプションユニット用リレー	2560点(320バイト)	10000 ~ 14777	コ1000 ~ コ1477	001000 ~ 001477
オプションユニット用フラグ	448点(56バイト)	15000 ~ 15677	コ1500 ~ コ1567	001500 ~ 001567
I/Oリンク用フラグ	64点(8バイト)	15700 ~ 15777	コ1570 ~ コ1577	001570 ~ 001577
I/Oリンク用リレー	2048点(256バイト)	20000 ~ 23777	コ2000 ~ コ2377	030000 ~ 030377
特殊I/Oユニット [基本システム]用リレー	4096点(512バイト)	30000 ~ 37777	コ3000 ~ コ3777	031000 ~ 031777
特殊I/Oユニット [リモートI/O子局]用リレー	1024点(128バイト)	40000 ~ 41777	コ4000 ~ コ4177	032000 ~ 032177

データメモリのアドレスはすべて8進数(OCT)で扱います。(レジスタ領域の4桁目の9は例外)
ただし、システムメモリ(#115)の設定により、サポートツール(JW-14PG、JW-50PG等)に表示
する進数を8進数/10進数/16進数から選択できます。

〔 2 〕 データメモリの機能

リ レ ー 領 域	入力ユニットを装着した領域	<ul style="list-style-type: none"> ・毎スキャンサイクルの入出力処理で入力ユニットのON / OFF状態を読み込み、1 スキャンサイクル中保持します。 1 ・プログラムで入力情報（接点、データ）として使用します。
	出力ユニットを装着した領域	<ul style="list-style-type: none"> ・ユーザープログラムで、コイル、デスティネーションとして演算結果を書き込みます。 ・入出力処理で出力ユニットにON / OFF状態を転送します。 ・演算結果はプログラム中で接点、ソースとして使用できます。
	オプションユニット、I/Oリンク親局ユニット、特殊I/Oユニットを装着した領域	<ul style="list-style-type: none"> ・オプションユニット(JW-21CM, JW-22CM, JW-21MN, JW-25CM)、I / Oリンク親局ユニット (JW-23LM, JW-23LMH)、特殊I / Oユニット (JW-21HC, JW-22HC, JW-24AD, JW-22DA, JW-21DU, JW-22DU, JW-21PS, JW-264N, JW-262S) を装着すると、入出力リレーとして16点 (2 バイト) 占有しますがこの16点は使用しないダミー領域です。
	ユニット未装着領域	<ul style="list-style-type: none"> ・外部に出力する必要のない演算結果の一時記憶に使用します。(補助リレー) ・ユーザープログラムでコイル、デスティネーションとして演算結果を書き込みます。 ・演算結果はプログラム中、接点、ソースとして使用できます。
	特殊リレー領域	<ul style="list-style-type: none"> ・自己診断結果の異常コードや、演算フラグ等に使用されます。 ・ユーザープログラムでコイル、デスティネーションとして使用できません。接点、ソースとして使用できます。
	オプションユニット用リレー、I/Oリンク用リレー、特殊I/Oユニット用リレー領域	<ul style="list-style-type: none"> ・オプションユニット、I/Oリンク親局ユニット、特殊I/Oユニットへの制御信号又は入出力データ格納用に使用できます。
T M R / C N T / M D 領 域	TMRとして使用している場合	<ul style="list-style-type: none"> ・減算タイマは現在値が0になるとTMR接点がONします。 ・加算タイマは現在値 = 設定値になるとTMR接点がONします。 ・TMR接点はプログラム中何度でも使用できます。 ・現在値はプログラム中、ソース (特殊用途としてデスティネーション) として使用できます。
	CNTとして使用している場合	<ul style="list-style-type: none"> ・減算カウンタは現在値が0になるとCNT接点がONします。 ・加算カウンタは現在値 = 設定値になるとCNT接点がONします。 ・CNT接点はプログラム中何度でも使用できます。 ・現在値はプログラム中、ソース (特殊用途としてデスティネーション) として使用できます。
	MDとして使用している場合	<ul style="list-style-type: none"> ・出力指示条件がONのとき、現在値領域にMD情報を書き込みます。 ・現在値領域のMD情報はプログラム中ソースとして使用できます。
	TMR、CNT、MDとして使用していない場合	<ul style="list-style-type: none"> ・現在値領域 (b x x x x) をレジスタとして使用できます。
	レジスタ	<ul style="list-style-type: none"> ・ユーザープログラムでデスティネーションとして演算結果を書き込みます。 ・演算結果はプログラム中、ソースとして使用します。 ・プログラマ等からデータの書き込みもできます。 ・時計機能の現在値読み出し及び、時計の設定を行います。(JW-31CUH/H1は除く) ・リンクユニットのフラグ及び通信条件を設定します。 ・自己診断結果を格納します。 ・デバイス機能での表示と入力のコントロール用に使用します。

- 1 入力ユニットを装着している領域は、入出力処理で読み込んだON / OFF状態を次のサイクルの入出力処理まで保持しますが、プログラム中でこれをコイル、デスティネーションとして使用すると、そのスキャンサイクル中は演算結果によりデータメモリが書き換ります。
- 2 ソース、デスティネーションとは応用命令で、演算結果を入れるレジスタをデスティネーション、演算前のデータを入れるレジスタをソースと呼びます。詳細は、9・1ページ「応用命令に関する留意事項」を参照してください。

〔3〕 特殊リレー

07300～07377の64点は、下記のように特殊リレー領域です。

07300		07340	自己診断結果の異常コードを収納する特殊レジスタでバイトアドレスコ0734として扱います。
07301		07341	
07302		07342	
07303		07343	
07304		07344	
07305		07345	
07306		07346	
07307		07347	
07310		07350	ノンキャリアフラグ エラーフラグ キャリアフラグ ゼロフラグ
07311		07351	
07312		07352	
07313		07353	
07314		07354	
07315		07355	
07316		07356	
07317		07357	
07320		07360	0.1秒クロック イニシャライズパルス ヒューズ切れ 1秒クロック 設定値変更スイッチ 常時OFFの接点
07321		07361	
07322		07362	
07323		07363	
07324		07364	
07325		07365	
07326		07366	
07327		07367	
07330		07370	メモリ異常 CPU異常 電池異常 入出力異常 オプション異常 特殊入出力ユニット異常フラグ 増設電源異常フラグ 電源異常
07331		07371	
07332		07372	
07333		07373	
07334		07374	
07335		07375	
07336		07376	
07337		07377	
		15766	キーデバイススイッチ
		15767	表示デバイススイッチ

これらのキーリレーは、07365を除き、CPUから書き込まれる領域で、ユーザープログラムでは接点、ソースとして使用してください。

コイル、デスティネーションとして使用できません。

2バイト以上のデータメモリを扱う命令や、分配・抽出命令、一括転送命令では特に注意してください。

空欄のリレーは予約領域のため、ユーザープログラムでは使用しないで下さい。

(例)

F-15	コ0000	コ0010	コ0732
MUL			

〔(コ0001、コ0000) × (コ0011、コ0010) の演算結果を(コ0735、コ0734、コ0733、コ0732)の4バイトに書き込む命令です。〕

コ0734、コ0735の特殊領域に演算結果が書き込まれてしまいます。

①07340~07347 (コ0734)

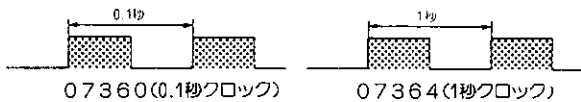
- 現在発生している異常内容のコードを格納する特殊レジスタです。
- 異常が回復すると異常コードはクリアします。
- 異常コードは6・8ページ「自己診断」を参照してください。

②07354~07357 (フラグ)

- フラグに影響を与える応用命令の実行時、演算内容に応じてセットします。
- 詳細は9・7ページ「データ処理命令とフラグ」を参照してください。

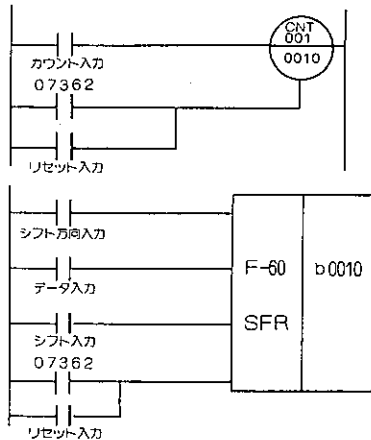
③07360 (0.1秒クロック)、07364 (1秒クロック)

- CNT命令のクロックや各種応用命令のクロックとして使用します。

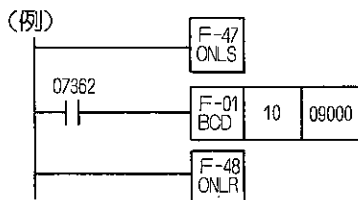


④07362(イニシャライズパルス)

コントロールユニットの運転開始直後の1演算サイクルの間ONになります。
カウンタやシフトレジスタの初期リセット(イニシャライズ)に使用します。



- イニシャライズパルスを立ち上がり演算命令の入力として使用する場合、レベル演算命令内で使用する必要があります。



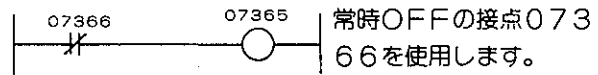
⑤07363 (ヒューズ切れ)

- JW-262Sのヒューズ切れ検出時ONします。

⑥07365 (設定値変更スイッチ)

- プログラマ等の周辺装置でキーリレー領域(初期状態は07000~15777, 20000~75777)以外のリレーをセット、リセットするとき、プログラム上07365をONとする必要があります。

- 常時ONとするプログラム



⑦07366 (常時OFFの接点)

- プログラムで常時OFF(a接点として使用)、常時ON(b接点として使用)となる接点として使用します。

⑧07370~07377 (自己診断結果)

- 自己診断結果、異常内容に応じた接点がONとなります。
- 詳細は6・8ページ「自己診断」を参照してください。

⑨15766~15767 (デバイススイッチ)

- プログラマ(JW-13PG等)の表示部とキーをPCプログラムで制御するときONにします。
- 使用方法はハンディプログラマJW-13PGの取扱説明書を参照してください。

{4} TMR、CNT、MDのデータ格納領域

b0000~b3777の2048バイトはTMR、CNTの現在値、MD命令のMD情報を格納する領域です。TMR、CNT、MDは合計1024点で、1点当り、2バイトを使用します。TMR、CNT、MD番号とb××××の領域の関係は次のようになります。

TMR、CNT、MD番号	データ格納領域
0000	b0000、b0001
0001	b0002、b0003
0002	b0004、b0005
0003	b0006、b0007
...	...
1776	b3774、b3775
1777	b3776、b3777

b0000~b3777をデータ処理命令(F-00等)で指定すれば、TMR、CNTの現在値を演算に使用できます。

TMR、CNT、MDには3種類のフォーマットがあります。

分類1	分類2	分類3
TMR	DTMR(BCD) UTMR(BCD)	DTMR(BIN) UTMR(BIN)
CNT	DCNT(BCD) UCNT(BCD)	DCNT(BIN) UCNT(BIN)
MD	—	—

MDおよび、分類2、分類3のTMR、CNT番号は000~777です。

(分類1のデータフォーマット)

	7	6	5	4	3	2	1	0	
TMR	$(\times 10^0)$				$(\times 10^{-1})$				n
	8	4	2	1	8	4	2	1	
CNT	$(\times 10^1)$				$(\times 10^0)$				n
	8	4	2	1	8	4	2	1	
MD	$(\times 10^1)$				$(\times 10^0)$				n
	8	4	2	1	8	4	2	1	
	MDのとき	入力情報			$(\times 10^2)$				n+1
	1	S ₁	S ₂	S ₃	8	4	2	1	n+1

※1 TMR、CNTの動作中は1(ON)、非計測又はリセット状態では0(OFF)となります。

- 数値はBCDで扱います。
- n、n+1はアドレス順を表わします。

(分類2のデータフォーマット)

設定値0~7999のTMR、CNTです。
Dは減算、Uは加算を表わします。

	7	6	5	4	3	2	1	0	
DTMR (BCD)	$(\times 10^0)$				$(\times 10^{-1})$				n
	8	4	2	1	8	4	2	1	
UTMR (BCD)	$(\times 10^0)$				$(\times 10^{-1})$				n
	8	4	2	1	8	4	2	1	
DCNT (BCD)	$(\times 10^1)$				$(\times 10^0)$				n
	8	4	2	1	8	4	2	1	
UCNT (BCD)	$(\times 10^1)$				$(\times 10^0)$				n
	8	4	2	1	8	4	2	1	
	リセット	$(\times 10^2)$			$(\times 10^1)$				n+1
	※1	4	2	1	8	4	2	1	n+1
	リセット	$(\times 10^3)$			$(\times 10^2)$				n+1
	※1	4	2	1	8	4	2	1	n+1

※1 TMR、CNTの動作中は1(ON)、非計測又はリセット状態では0(OFF)となります。

- 数値はBCDで扱います。
- n、n+1はアドレス順を表わします。
- TMR、CNT、U、Dの判別はPCのプログラムで管理します。

(分類3のデータフォーマット)

設定値0~32767のTMR、CNTです。
Dは減算、Uは加算を表わします。

	7	6	5	4	3	2	1	0	
DTMR (BIN)	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	n
	リセット	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	
UTMR (BIN)	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	n
	リセット	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	
DCNT (BIN)	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	n
	リセット	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	
UCNT (BIN)	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	n
	リセット	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	

- 数値はBIN(バイナリ)で扱います。
- TMRの数値単位は0.1秒です。
- TMRを10msタイマとして使用中のb1000~b1777の領域はレジスタとして使用できません。
- 10msタイマ領域の設定方法
システムメモリ#227の設定で100msタイマを10msタイマに変更できます。

#227の設定値	10msタイマの領域
000(8) (00(H))	全て100msタイマ
345(8) (E5(H))	0700~0777が10msタイマ
ビットのON/OFFでTMR0400~0777の各領域を10ms/100msタイマに設定可能(#227参照)	

DTMR、UTMRでは100msタイマとしてのみ動作し、10msタイマにはなりません。

〔5〕リレー領域のバイトアドレス

JW30Hは、AND、ORといったビット単位の演算のみではなく、四則演算や転送といったデータ処理の機能を豊富に備えたプログラマブルコントローラです。データ処理命令は、バイト単位またはワード単位で扱います。リレー領域をデータ処理の対象とするとき、これらの領域をバイトアドレスで指定します。

バイトアドレスはリレー番号と対応したバイト単位のアドレスで、5桁のリレー番号の最下位桁を捨てた上4桁にバイト単位であることを明確にするためコ（コードの意味）を付加したものです。

(例) 0201702016020150201402013020120201102010

のバイトアドレスはコ0201となります。

応用命令でソース、テストイネーションとして、リレー領域をバイト指定するとき、このバイトアドレスを使用します。

〔6〕レジスタ領域の予約領域

レジスタ09000～E7777は、応用命令を使用したデータ格納用に使用できます。ただしJW30Hの内部処理用及びオプションユニットのために割付けられたアドレスがあります。

- (1) リンクユニットJW-21CMで使用する領域
JW-21CMでリモートI/O、データリンク (DL1/DL9) 機能を使用する場合、ユニットNo.スイッチを「5」に設定すると89000～89777の512バイトは、JW-21CMのデータ領域となります。詳細は「JW-21CMユーザーズマニュアル」を参照してください。
- (2) JW10リンクユニットJW-25CMで使用する領域
JW-25CMでユニットNo.スイッチを下記に設定すると、59000～99757は、JW-25CMのデータ領域となります。

ユニットNo.スイッチ	レジスタ番号
SW2	59000～69757
SW3	69000～79757
SW4	79000～89757
SW5	89000～99757

詳細は「JW-25CMユーザーズマニュアル」を参照してください。

- (3) デバイス機能で使用する領域
プログラマ (JW-14PG等) の表示部とキーをPCプログラムで制御するとき使用するレジスタです。使用方法はJW-14PGの取扱説明書を参照してください。

デバイスモード キー入力用	レジスタ(1バイト) 99667
デバイスモード 表示用	レジスタ(64バイト) 99670～99767

- (4) 時計機能で使用する領域
JW-32CUH/H1、JW-33CUH/H1/H2/H3には時計機能があります。時計データの読み出しと時刻設定用にレジスタ99770～99777の8バイトを使用します。ただし時計機能でレジスタを使用しないときはシステムメモリ#223に001₍₈₎をセットしてください。

レジスタ番号	内 容
99770	秒 : 00～59 (BCD)
99771	分 : 00～59 (BCD)
99772	時 : 00～23 (BCD)
99773	日 : 01～31 (BCD)
99774	月 : 01～12 (BCD)
99775	年 : 00～99 (BCD)
99776	曜日 : 0～6 (BCD)
99777	コントロール

- 1) 月ごとの「日付」と、「うるう年」計算は自動的にを行います。
- 2) 「年」は、西暦の下2桁を表わします。うるう年の自動判別は4年ごとの年数のみの判別です。(92年、96年、00年はうるう年と判別します。)
- 3) 曜日は、時刻セット時の曜日に合わせてください。日付けが変わるごとに0~6に順次変化します。曜日については年月日設定による計算はしません。

曜	日	日	月	火	水	木	金	土
BCD値	00	01	02	03	04	05	06	

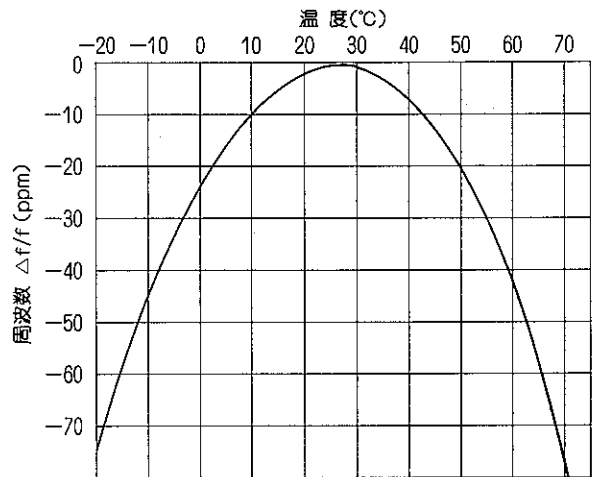
- 4) コントロールはレジスタ99777のビットをセットすると動きます。

内容	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
ON	時刻合せ	未使用			30秒補正	未使用		時計停止
OFF	時刻モータ	未使用			---	未使用		時計運転

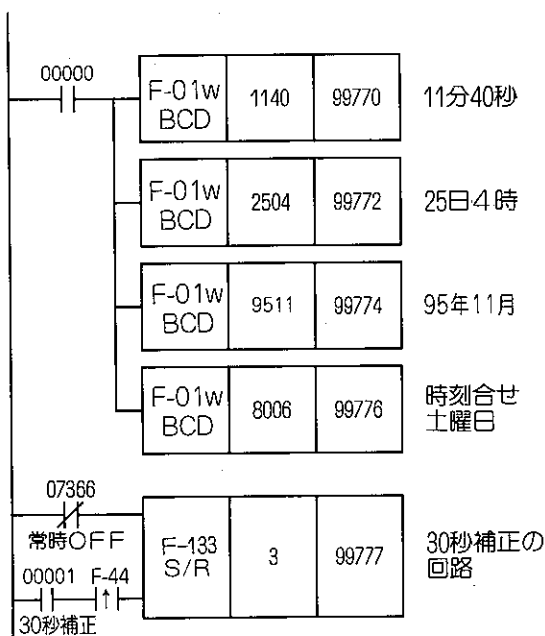
- D₀は時計の運転・停止を指示します。停止にするといつまでも止っています。
- D₃は30秒補正に使用します。現在時刻の秒値により変わります。
0~29秒……「00」秒となり1分の桁上げ無し。
30~59秒……「00」秒となり1分の桁上げ有り。
※ 30秒補正はF-133 (ビットセット/リセット) で「ON」してください。補正が完了すると本PCからリセットします。
- D₇は「ON」で時刻合せになります。「OFF」では、時刻モータとなります。時刻合せはI/O演算の終了後に行うため、下記の回路で行うと時計停止 (D₀ON) にしなくても使用できます。D₇は「ON」にすると時刻合せ完了で本PCからリセットします。(例95年11月25日4時11分40秒土曜日にセットします。)

- レジスタ99777のD₀とD₇のビットは応用命令等で連続「ON」にしないでください。時計が正常に作動しなくなります。
- 時計として有り得ないデータ(例2月30日等)はセットしないでください。時計が正常に動作しないときがあります。
- 時計の精度は±1秒/日 (at25°C) です。ただし、温度は時計素子の周囲温度によります。

《参考》時計素子の温度特性、なお、11.574ppmで1秒/1日の誤差となります。



- システムメモリ #010~#017もレジスタ99770~99777と同じはたらきをします。プログラマ(JW-13PG等)からの時刻合せに使用します。



(5) 異常履歴格納領域

システムメモリ#210=002octのとき、レジスタE6000~E7777は、コントロールユニットと各オプションユニット(JW-21CM/22CM/21MN/25CM)の異常履歴を格納します。システムメモリ#210=000のときは、E0000~E7777は汎用レジスタとして使えます。

1) レジスタの割付け

オプションユニットのユニットNo.スイッチの設定で割り付けます。

レジスタ	ユニットNo.の設定	内 容	
E6000 } E6177	6	オプションユニット (JW-21CM/22CM/21MN/25CM) の異常コード	
E6200 } E6377	5		
E6400 } E6577	4		
E6600 } E6777	3		
E7000 } E7177	2		
E7200 } E7377	1		
E7400 } E7577	0		
E7600 } E7777	コントロール ユニット		<ul style="list-style-type: none"> ●コントロールユニットの異常コード ●PGインターフェイスの異常 ●コミュニケーションポートの異常

2) 格納する異常データ

異常データは1つ当たり16/バイトで構成しています。

アドレス	内 容	
n+0	秒	
n+1	分	
n+2	時	
n+3	日	
n+4	月	
n+5	年	
n+6	曜日	
n+7	異常コード	異常コードを格納 ※1
n+10	0	※3
n+11	発生回数	000~377(8) ※4
n+12		
n+13		
n+14		
n+15		

※1 コントロールユニットがJW-31CUH/H1のときはアドレスn+0~n+6に格納しているデータは無視してください。JW-31CUH/H1には時計機能がないため、正しいデータは格納しません。

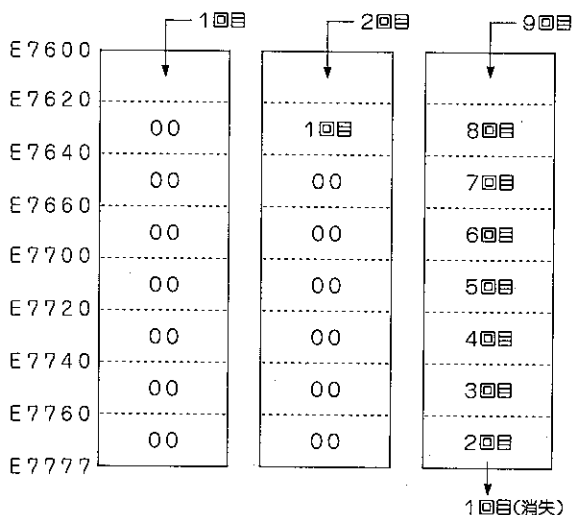
※2 コントロールユニットの時、6・8ページ「自己診断」の異常コードを格納します。オプションユニットの時はそれぞれのユニットの異常コードを格納します。

※3 入出力ユニット異常の時、上位4ビットでラック番号(0~3)下位4ビットでスロット番号(0~7)を格納します。
オプションユニットおよびI/Oリンク親局ユニットの異常のときは「00(H)」を格納します。

※4 同じ異常コードが連続して発生したときは(例13:電源異常)、発生回数を+1加算し、377(8)まで加算します。377(8)回以上発生したときは377(8)のままとなります。発生時刻・日付けは最初に発生した値です。

3) 異常データの格納

異常データは割付けられたレジスタ領域に異常発生順に8回まで格納します。異常が8回以上になると、最初に格納した異常データから順に消失します。



(6) 構造化プログラムで使用する領域

JW30HはラダーソフトJW-52SP/92SPを使用して構造化プログラムを作成できますが、その場合、下記のレジスタ領域を使用します。

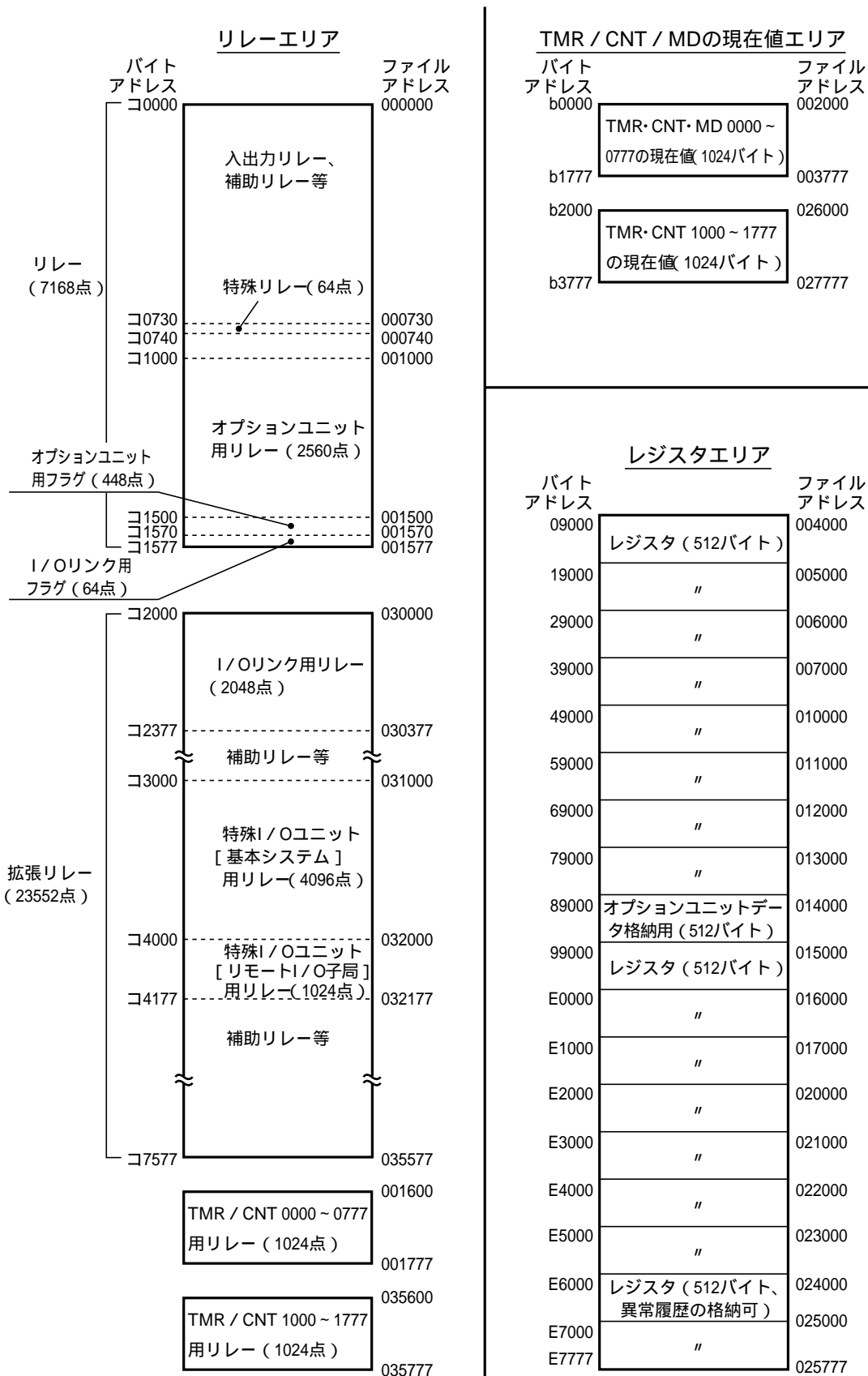
機種名	使用レジスタ領域
JW-31CUH/H1	59000~89777, E0000~E5777
JW-32CUH/H1	ファイル2のレジスタ
JW-33CUH/H1	ファイル3のレジスタ
JW-33CUH2	ファイル14のレジスタ
JW-33CUH3	ファイル2Cのレジスタ

構造化プログラムに関しては、「構造化プログラミングマニュアル」を参照願います。

2 - 3 ファイル0のメモリマップ

〔1〕バイトアドレス順

ファイル0のメモリマップをリレーエリア、TMR / CNT / MDの現在値エリア、レジスタエリアに分けてバイトアドレス順に示します。



~ の詳細：次ページ参照

・リレー、拡張リレーの ~ (前ページ参照)

種 類	ユニットNo.スイッチ の設定値	バイトアドレス	リレー番号	容 量		
				点数(バイト数)	合計点数	
特殊リレー	-	コ0730 ~ コ0737	07300 ~ 07377	64点(8バイト)	64点	
オプション ユニット用リレー	0	コ1000 ~ コ1077	10000 ~ 10777	512点(64バイト)	2560点	
	1	コ1100 ~ コ1177	11000 ~ 11777	"		
	2	コ1200 ~ コ1277	12000 ~ 12777	"		
	3	コ1300 ~ コ1377	13000 ~ 13777	"		
	4	コ1400 ~ コ1477	14000 ~ 14777	"		
オプション ユニット用フラグ	0	コ1500 ~ コ1507	15000 ~ 15077	64点(8バイト)	448点	
	1	コ1510 ~ コ1517	15100 ~ 15177	"		
	2	コ1520 ~ コ1527	15200 ~ 15277	"		
	3	コ1530 ~ コ1537	15300 ~ 15377	"		
	4	コ1540 ~ コ1547	15400 ~ 15477	"		
	5	コ1550 ~ コ1557	15500 ~ 15577	"		
	6	コ1560 ~ コ1567	15600 ~ 15677	"		
I/Oリンク 用フラグ	0	コ1570 ~ コ1571	15700 ~ 15717	16点(2バイト)	64点	
	1	コ1572 ~ コ1573	15720 ~ 15737	"		
	2	コ1574 ~ コ1575	15740 ~ 15757	"		
	3	コ1576 ~ コ1577	15760 ~ 15777	"		
I/Oリンク用 リレー	0	コ2000 ~ コ2077	20000 ~ 20777	512点(64バイト)	2048点	
	1	コ2100 ~ コ2177	21000 ~ 21777	"		
	2	コ2200 ~ コ2277	22000 ~ 22777	"		
	3	コ2300 ~ コ2377	23000 ~ 23777	"		
特殊I/Oユニット [基本システム] 用リレー	R0	0	コ3000 ~ コ3017	30000 ~ 30177	128点(16バイト)	4096点
		1	コ3020 ~ コ3037	30200 ~ 30377	"	
		2	コ3040 ~ コ3057	30400 ~ 30577	"	
		3	コ3060 ~ コ3077	30600 ~ 30777	"	
		4	コ3100 ~ コ3117	31000 ~ 31177	"	
		5	コ3120 ~ コ3137	31200 ~ 31377	"	
		6	コ3140 ~ コ3157	31400 ~ 31577	"	
		7	コ3160 ~ コ3177	31600 ~ 31777	"	
	R1	0	コ3200 ~ コ3217	32000 ~ 32177	128点(16バイト)	
		1	コ3220 ~ コ3237	32200 ~ 32377	"	
		2	コ3240 ~ コ3257	32400 ~ 32577	"	
		3	コ3260 ~ コ3277	32600 ~ 32777	"	
		4	コ3300 ~ コ3317	33000 ~ 33177	"	
		5	コ3320 ~ コ3337	33200 ~ 33377	"	
		6	コ3340 ~ コ3357	33400 ~ 33577	"	
7	コ3360 ~ コ3377	33600 ~ 33777	"			

R0、R1はベースユニットのラック番号を示します。
は次ページに続きます。

種 類	ユニットNo.スイッチ の設定値	バイトアドレス	リレー番号	容 量		
				点数(バイト数)	合計点数	
特殊I/Oユニット [基本システム] 用リレー	R2	0	コ3400 ~ コ3417	34000 ~ 34177	128点(16バイト)	4096点
		1	コ3420 ~ コ3437	34200 ~ 34377	"	
		2	コ3440 ~ コ3457	34400 ~ 34577	"	
		3	コ3460 ~ コ3477	34600 ~ 34777	"	
		4	コ3500 ~ コ3517	35000 ~ 35177	"	
		5	コ3520 ~ コ3537	35200 ~ 35377	"	
		6	コ3540 ~ コ3557	35400 ~ 35577	"	
	7	コ3560 ~ コ3577	35600 ~ 35777	"		
	R3	0	コ3600 ~ コ3617	36000 ~ 36177	128点(16バイト)	
		1	コ3620 ~ コ3637	36200 ~ 36377	"	
		2	コ3640 ~ コ3657	36400 ~ 36577	"	
		3	コ3660 ~ コ3677	36600 ~ 36777	"	
		4	コ3700 ~ コ3717	37000 ~ 37177	"	
		5	コ3720 ~ コ3737	37200 ~ 37377	"	
6		コ3740 ~ コ3757	37400 ~ 37577	"		
7	コ3760 ~ コ3777	37600 ~ 37777	"			
特殊I/Oユニット [リモート/子局] 用リレー	0	コ4000 ~ コ4017	40000 ~ 40177	128点(16バイト)	1024点	
	1	コ4020 ~ コ4037	40200 ~ 40377	"		
	2	コ4040 ~ コ4057	40400 ~ 40577	"		
	3	コ4060 ~ コ4077	40600 ~ 40777	"		
	4	コ4100 ~ コ4117	41000 ~ 41177	"		
	5	コ4120 ~ コ4137	41200 ~ 41377	"		
	6	コ4140 ~ コ4157	41400 ~ 41577	"		
	7	コ4160 ~ コ4177	41600 ~ 41777	"		

R2、R3はベースユニットのラック番号を示します。

留 意 点 (ユニット No. スイッチの設定値)

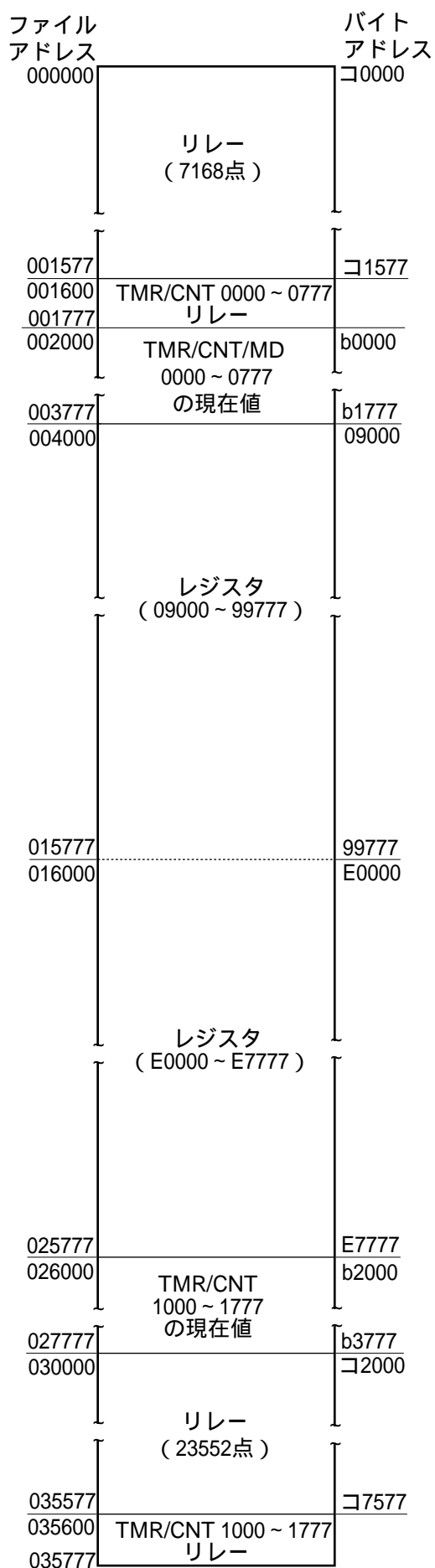
オプションユニット、I/Oリンクユニット、特殊I/OユニットのユニットNo. スイッチ設定について下記に注意してください。

1. オプションユニットとオプションユニットとの間で、ユニットNo. スイッチの設定値を重複させないでください。
2. I/OリンクユニットとI/Oリンクユニットとの間で、ユニットNo. スイッチの設定値を重複させないでください。
3. 同一ベースユニットに実装されている特殊I/Oユニットと特殊I/Oユニットとの間で、ユニットNo. スイッチの設定値を重複させないでください。
4. オプションユニット、I/Oリンクユニット、特殊I/Oユニットの間ではユニットNo. スイッチの設定値を重複して使用できます。

	形 名
オプションユニット	JW-21CM、JW-22CM、JW-21MN、JW-25CM、JW-255CM、JW-20FL5 JW-20FLT、JW-20DN
I/Oリンクユニット	JW-23LMH
特殊I/Oユニット	JW-264N、JW-262S、JW-21HC、JW-22HC、JW-24AD、JW-22DA JW-21DU、JW-22DU、JW-21SU、JW-21PS

〔 2 〕 ファイルアドレス順

ファイル 0 のメモリマップをファイルアドレス順に示します。



2 - 4 入出力リレーの割付

入力/出力/特殊/オプションユニットのリレー番号は、電源ON時(コントロールユニットのPROTECTスイッチ:OFF)での自動登録、またはサポートツール(JW30H対応)のI/O登録(自動登録/テーブル作成)により割り付けられます。

割付は基本/増設ベースユニットのラック/スロット番号ごとに、実装ユニットの種類により、リレー点数がJW30Hのコントロールユニットに登録されます。

コントロールユニットのPROTECTスイッチがOFFでシステムメモリ#247=0の場合、電源ONで自動登録されます。また、自動登録後、運転中はPROTECTスイッチをONにするか、#247=003octを書き込み自動登録を禁止にしてください。

登録されたリレー番号は、JW30Hのモード変更(停止から運転)時に自己診断により照合されます。照合の結果、実装ユニットと異なると、コントロールユニットのFAULTランプが点灯し、JW30Hは動作を停止します。また、異常コード60(テーブル照合エラー)がシステムメモリ#160に格納されます。

[1] I/O登録の種類

JW30HのI/O登録には、「自動登録」と「テーブル作成」の2種類があります。

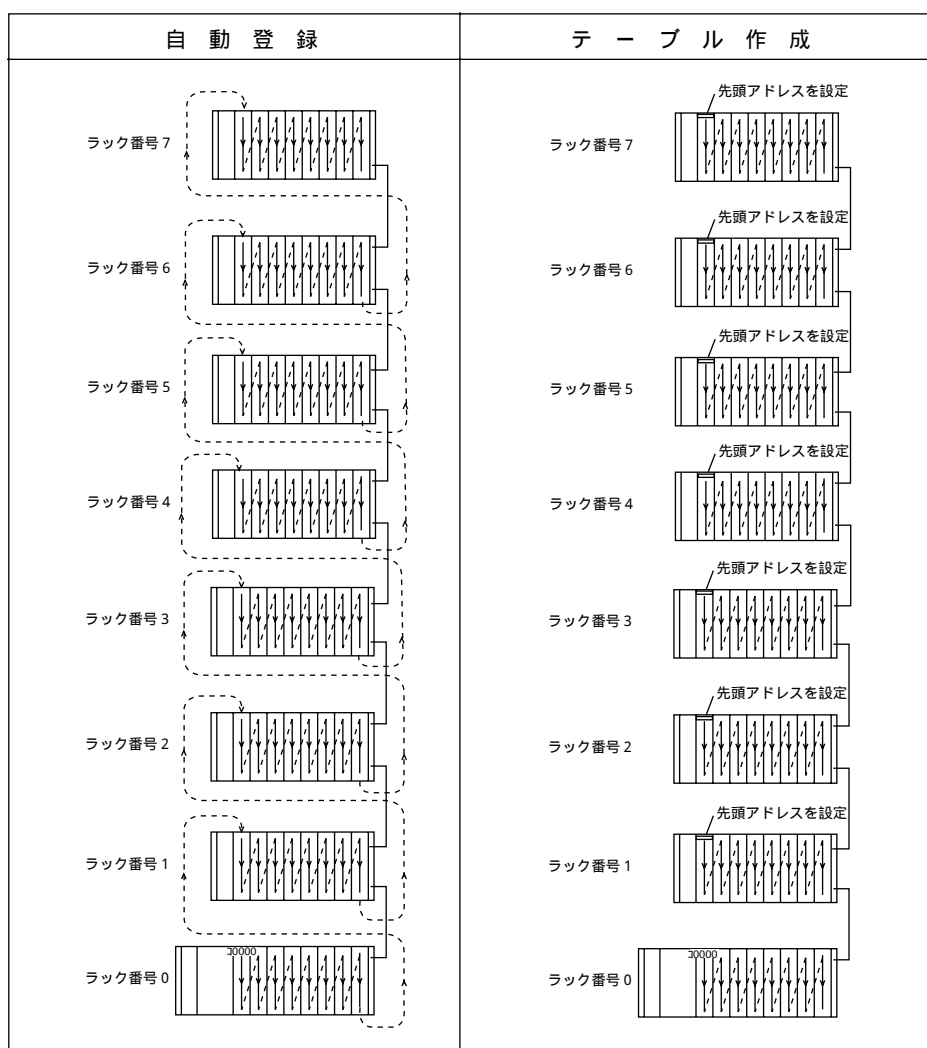
(1) 自動登録

ラック番号0~7の先頭アドレスを、コ0000からの連続アドレスで自動的に設定します。

(2) テーブル作成

増設ベースユニット(ラック番号1~7)に、リレー番号の先頭アドレスを偶数アドレス(コ0000~コ1577の範囲内)で設定します。

前ラックのリレー番号と次ラックのリレー番号が重複しないようにしてください。



〔 2 〕 最大入出力点数と入出力リレーの割付

各コントロールユニットには、最大制御入出力点数があり、また、実装ユニットの種類により、実装可能台数が異なります。

また、各ユニットの最大制御入出力点数に影響するリレー点数と入出力リレー割付点数は異なりますので注意してください。

各コントロールユニットの最大制御入出力点数

コントロールユニット機種名	最大制御入出力点数	最大入出力リレー割付点数	入出力リレー領域	
			自動登録	テーブル作成
JW-31CUH1	512点	1280点	10000 ~ 10237	10000 ~ 11577
JW-32CUH1	1024点	1536点	10000 ~ 10277	
JW-33CUH1/2/3	3072点	2048点	10000 ~ 10377	

各ユニットのリレー点数と実装可能台数

ユニットの種類	最大制御入出力点数に影響するリレー点数	入出力リレー割付点数	最大実装台数	実装可能ラック
8点入力/出力ユニット	16点	16点	64台	ラック0~7
16点入力/出力ユニット	16点	16点	64台	ラック0~7
32点入力/出力/入出力ユニット	32点	32点	64台	ラック0~7
特殊I/Oユニット	64点入力/出力	64点 (*1)	32台	ラック0~3
	64点入力/出力以外	0		
I/Oリンク親局ユニット	0	16点	4台	ラック0
オプションユニット	0	16点	7台	ラック0
アキスロット	0	16点	-	ラック0~7

(*1) 64点入力/出力ユニットの制御リレーは、特殊I/O用リレー領域 (13000 ~ 13777) を使用します。

ユニット実装例

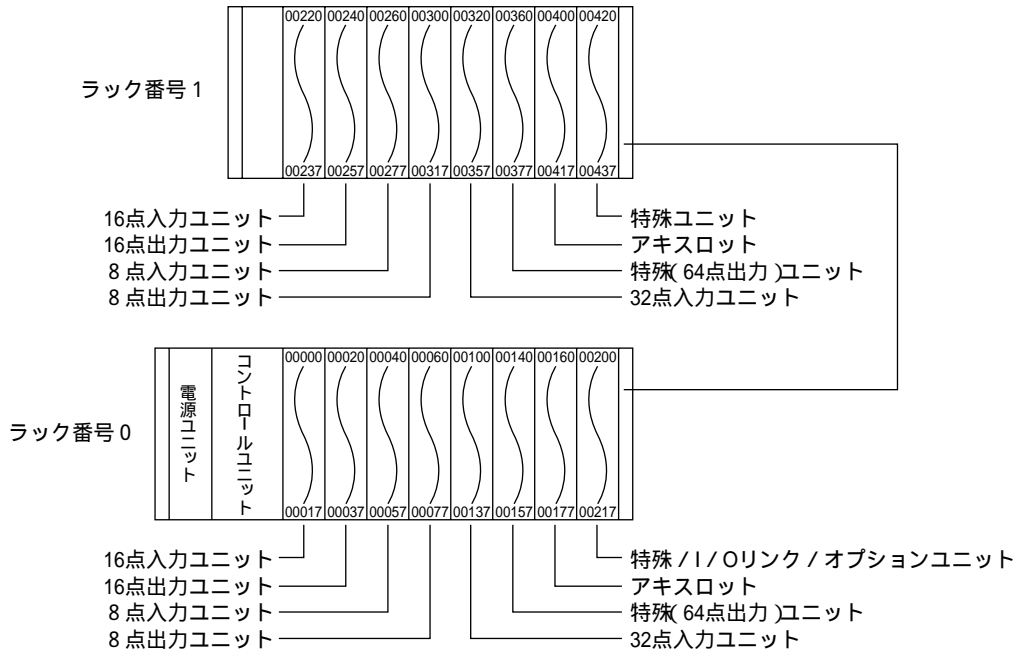
コントロールユニット機種名	最大実装ユニット数				制御入出力点数	入出力リレー割付点数
	8点・16点入力/出力ユニット	32点入力/出力/入出力ユニット	特殊I/Oユニット (64点入力/出力)	左記以外のユニット実装可能台数 (アキスロット含む) [] 内は特殊I/Oユニット		
JW-31CUH1	32台	0	0	32台 [32台]	512点 (16点×32)	1024点 (16点×32+16点×32)
	0	16台	0	48台 [32台]	512点 (32点×16)	1280点 (32点×16+16点×48)
	0	0	8台	56台 [24台]	512点 (64点×8)	1024点 (16点×8+16点×56)
JW-32CUH1	64台	0	0	0	1024点 (16点×64)	1024点 (16点×64)
	0	32台	0	32台 [0]	1024点 (32点×32)	1536点 (32点×32+16点×32)
	0	0	16台	48台 [16台]	1024点 (64点×16)	1024点 (16点×16+16点×48)
JW-33CUH1 JW-33CUH2 JW-33CUH3	64台	0	0	0	1024点 (16点×64)	1024点 (16点×64)
	0	64台	0	0	2048点 (32点×64)	2048点 (32点×64)
	0	0	32台	32台 [0]	2048点 (64点×32)	1024点 (16点×32+16点×32)
	0	32台	32台	0	3072点 (32点×32+64点×32)	1536点 (32点×32+16点×32)

上記は、基本ベースユニットJW-38KB (8スロット)、増設ベースユニットJW-38ZB (8スロット) を7台使用した場合です。(合計8スロット×8ラック=64台)

〔 3 〕 入出力リレー番号の割付例

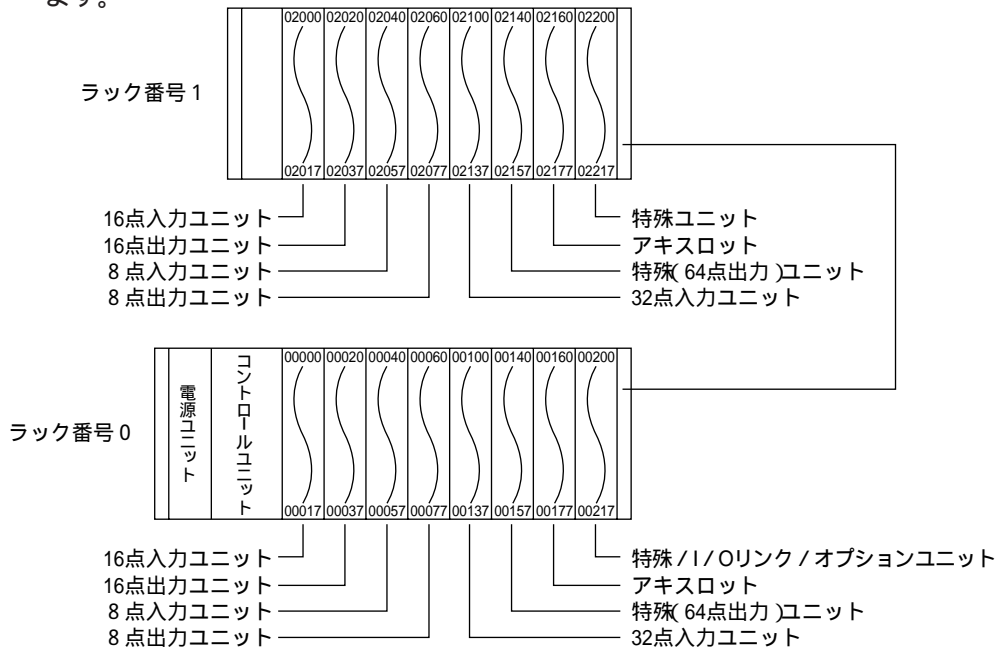
（ 1 ） 自動登録時の例

下記システム構成でのリレー番号を示します。



（ 2 ） テーブル作成時の例

下記システム構成で、ラック番号 1 の先頭アドレスを C0200 に設定した場合のリレー番号を示します。



〔 4 〕 サポートツールによる I/O 登録

次のサポートツール (JW30H 対応) を使用すると、メニュー操作により I/O 登録できます。操作方法は各機種の取扱説明書を参照願います。

[JW30H 対応のサポートツール]

- ・ JW-14PG
- ・ JW-100SP
- ・ JW-13PG (B マーク付)
- ・ JW-92SP (Ver 5.5 以上)
- ・ JW-50PG (Ver 5.5 以上)
- ・ JW-52SP (Ver 5.5 以上)

第 3 章 パラメータメモリ

3 - 1 特殊I/Oユニットのパラメータ

特殊I/Oユニット(JW-21HC、JW-22HC、JW-24AD、JW-22DA、JW-21PS、JW-21SU)の動作条件を設定します。使用領域は特殊I/OユニットのユニットNo. スイッチにより決定されます。

1 ユニット当たり128バイトを使用します。

ユニットNo. スイッチの 設定値	パラメータアドレス									
	ラック0		ラック1		ラック2		ラック3		リモートI/O子局	
0	T-00	000 ~ 177	T-10	000 ~ 177	T-20	000 ~ 177	T-30	000 ~ 177	T-40	000 ~ 177
1	T-01	000 ~ 177	T-11	000 ~ 177	T-21	000 ~ 177	T-31	000 ~ 177	T-41	000 ~ 177
2	T-02	000 ~ 177	T-12	000 ~ 177	T-22	000 ~ 177	T-32	000 ~ 177	T-42	000 ~ 177
3	T-03	000 ~ 177	T-13	000 ~ 177	T-23	000 ~ 177	T-33	000 ~ 177	T-43	000 ~ 177
4	T-04	000 ~ 177	T-14	000 ~ 177	T-24	000 ~ 177	T-34	000 ~ 177	T-44	000 ~ 177
5	T-05	000 ~ 177	T-15	000 ~ 177	T-25	000 ~ 177	T-35	000 ~ 177	T-45	000 ~ 177
6	T-06	000 ~ 177	T-16	000 ~ 177	T-26	000 ~ 177	T-36	000 ~ 177	T-46	000 ~ 177
7	T-07	000 ~ 177	T-17	000 ~ 177	T-27	000 ~ 177	T-37	000 ~ 177	T-47	000 ~ 177

JW30H未対応のサポートツール JW-12PG等、JW-2PGは除く)を使用してパラメータを設定する場合は、ファイルEのファイルアドレスに設定します。ファイルEのファイルアドレスに設定する場合は、システムメモリ #260 = 50HEXを設定する必要があります。(JW50H/70H/100Hと認識)

ユニットNo. スイッチの 設定値	ファイルEのファイルアドレス (8 進)				
	ラック0	ラック1	ラック2	ラック3	リモートI/O子局
0	000000 ~ 000177	002000 ~ 002177	004000 ~ 004177	006000 ~ 006177	010000 ~ 010177
1	000200 ~ 000377	002200 ~ 002377	004200 ~ 004377	006200 ~ 006377	010200 ~ 010377
2	000400 ~ 000577	002400 ~ 002577	004400 ~ 004577	006400 ~ 006577	010400 ~ 010577
3	000600 ~ 000777	002600 ~ 002777	004600 ~ 004777	006600 ~ 006777	010600 ~ 010777
4	001000 ~ 001177	003000 ~ 003177	005000 ~ 005177	007000 ~ 007177	011000 ~ 011177
5	001200 ~ 001377	003200 ~ 003377	005200 ~ 005377	007200 ~ 007377	011200 ~ 011377
6	001400 ~ 001577	003400 ~ 003577	005400 ~ 005577	007400 ~ 007577	011400 ~ 011577
7	001600 ~ 001777	003600 ~ 003777	005600 ~ 005777	007600 ~ 007777	011600 ~ 011777

パラメータの内容は各特殊I/Oユニットのユーザズマニュアルを参照願います。

3 - 2 オプションユニットのパラメータ

オプションユニット(JW-21CM等)の動作条件を設定します。

使用領域はオプションユニットのユニットNo. スイッチにより決定されます。

1ユニット当たり64バイト使用します。

JW30H未対応のサポートツール(JW-12PG等、JW-2PGは除く)を使用してパラメータを設定する場合は、ファイルEのファイルアドレスに設定します。ファイルEのファイルアドレスに設定する場合は、システムメモリ #260 = 50HEXを設定する必要があります。(JW50H/70H/100Hと認識)

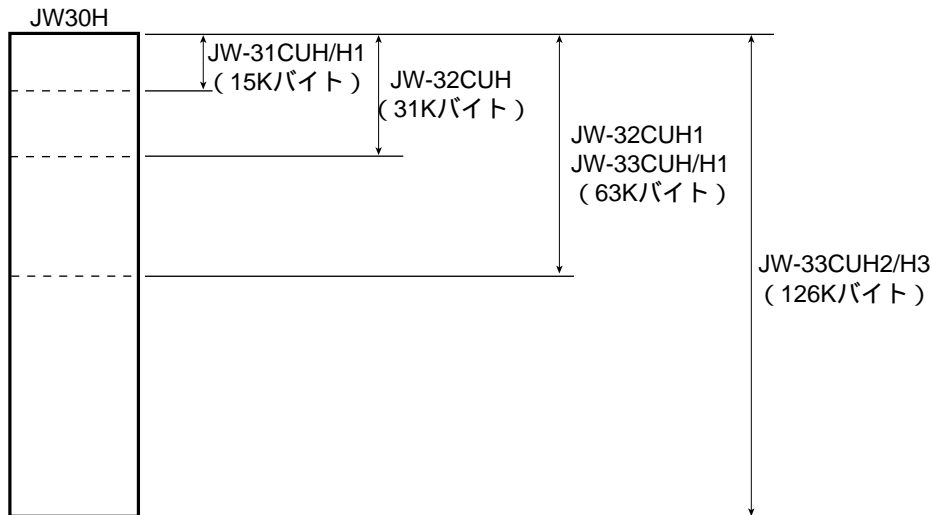
ユニットNo. スイッチ	パラメータ アドレス	ファイルEのファイル アドレス(8進)
0	00 ~ 77	012000 ~ 012077
1	00 ~ 77	012100 ~ 012177
2	00 ~ 77	012200 ~ 012277
3	00 ~ 77	012300 ~ 012377
4	00 ~ 77	012400 ~ 012477
5	00 ~ 77	012500 ~ 012577
6	00 ~ 77	012600 ~ 012677

パラメータの内容はオプションユニットのユーザズマニュアルを参照願います。

第 4 章 プログラムメモリ

プログラムメモリとは、ユーザープログラムを書き込む領域で、PCが運転中はプログラムの先頭から順次読み出し、プログラム内容に応じて演算します。

JW30Hではコントロールユニットによりプログラム容量が異なります。



プログラムメモリの容量を表現するとき、一般にバイト数ではなく、語数を用います。

(1 Kバイトは0.5K語と表現)

1 Kは1024を示します。従って、7.5K語は7680語となります。

プログラムメモリをクリアすると、最終アドレスにはEND命令(F-40)、その他のアドレスにはNOP命令(演算しない命令)が書き込まれます。

プログラムアドレスは8進数で表現しますが、システムメモリ(#115)の設定により、JW-13PG等のサポートツールの表示を8進/10進/16進から選択できます。

コントロールユニット	プログラムメモリ容量 (語数)	プログラムアドレス		
		8進数	10進数	16進数
JW-31CUH/H1	7.5K語	00000 ~ 16777	00000 ~ 07679	0000 ~ 1DFF
JW-32CUH	15.5K語	00000 ~ 36777	00000 ~ 15871	0000 ~ 3DFF
JW-32CUH1 (*1)	15.5K語	00000 ~ 36777	00000 ~ 15871	0000 ~ 3DFF
	31.5K語	00000 ~ 76777	00000 ~ 32255	0000 ~ 7DFF
JW-33CUH/H1	31.5K語	00000 ~ 76777	00000 ~ 32255	0000 ~ 7DFF
JW-33CUH2/H3	63.0K語	000000 ~ 076777	00000 ~ 32255	0000 ~ 7DFF
		100000 ~ 176777	32678 ~ 65023	8000 ~ FDFF

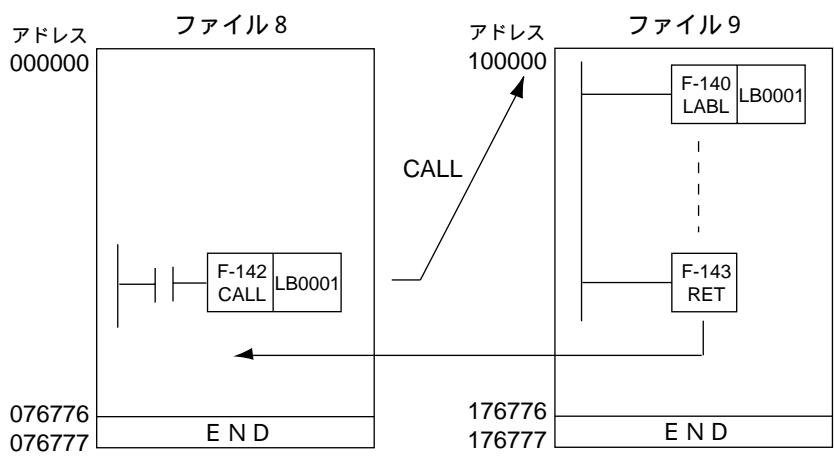
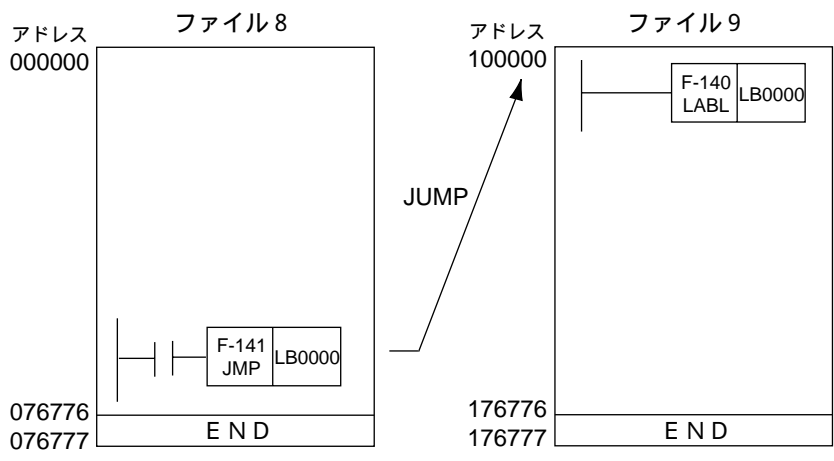
(*1) JW-32CUH1は、ユニット内部スイッチにより、プログラムメモリ容量を15.5K語または、31.5K語に設定します。

命令には1語命令、2語命令、3語命令、4語命令があり、1語は2バイトで構成しています。

	代表的な命令	使用バイト数
1語命令	STR、AND等(リレー番号00000 ~ 15777octを使用時)	2
2語命令	TMR、CNT等 STR、AND等(リレー番号20000 ~ 75777octを使用時)	4
3語命令	F-00、F-01等	6
4語命令	F-10、F-11等	8

【JW-33CUH2/H3 のプログラムメモリ使用時の注意事項】

- JW-33CUH2/H3 のプログラムメモリは、前半部分のファイル 8 (000000 ~ 076777) と後半部分のファイル 9 (100000 ~ 176777) があります。
- ファイル 9 は、JUMP 命令 (F-141、F-151) による分岐先、および CALL 命令 (F-142、F-148) によるサブルーチンプログラムの格納先として使用します。



- ファイル 8 とファイル 9 を跨ぐような命令語の書込、挿入、削除はできません。
- 検索、プログラムチェックの範囲は全エリアに対して行います。
- ファイル 9 で異常が発生した場合、システムメモリ #052、#053 には、異常アドレス (100000 ~ 176777) がそのまま格納されます。

第 5 章 システムメモリ

システムメモリはJW30Hの各種機能を設定するのに使用します。

5 - 1 システムメモリ

システムメモリは#0000～#2177の1152バイトのメモリで、電池でバックアップしています。
 下記はユーザーに開放されたメモリ番号ですが、これ以外のメモリ番号は予約領域のため、データを書き込まないでください。

システムメモリ番号 8進(OCT)	初期値	内 容	詳細 ページ
#010	時刻	秒	時計のモニタ (BCD) [JW-32CUH/H1 JW-33CUH/H1/H2/H3] JW-31CUH/H1には機能なし
#011		分	
#012		時	
#013		日	
#014		月	
#015		年	
#016		曜日	
#017	0	コントロール	
#030	0	下位 スキャンタイムの最小値のモニタ	5・4
#031		上位 (BCD)	
#032		下位 毎スキャンタイムの現在値のモニタ	5・4
#033		上位 (BCD)	
#034		下位 スキャンタイムの最大値のモニタ	5・4
#035		上位 (BCD)	
#046	0	異常を検知したI/Oのラック、スロットのモニタ (OCT)	5・5
#050	0	異常オプションユニットのスイッチ番号のモニタ	5・6
#051	0	異常I/Oリンク親局ユニットのスイッチ番号のモニタ	5・6
#052	0	下位 プログラムの異常アドレスのモニタ	5・7
#053		上位 (OCT)	
#114	0	サポートツールでの数値表現指示	5・7
#115	0		
#136	0	ツール機種の設定	5・8
#160	0	自己診断結果の異常コード	5・8
#161			
#162			
#163			
#164			
#165			
#166			
#167			
#170	0	オプションユニットの異常コード	5・8
#171			
#172			
#173			
#174			
#175			
#176			
#177			

システムメモリ番号 8進(OCT)	初期値	内 容	詳細 ページ
# 201	0	TMRのリセット条件設定	5・8
# 202		CNTのリセット条件設定	5・8
# 206	0	ヒューズ断検出時、運転継続 / 停止の設定	5・8
# 207	0	オプションユニット異常時、運転継続 / 停止の設定	5・9
# 210	002 _{OCT}	異常履歴格納領域の使用選択	5・9
# 211	0	I/Oリンク親局ユニット異常時、運転継続 / 停止の設定	5・9
# 220	0	コメントメモリ使用領域の先頭ファイル番号の設定	5・9
# 222	0	PG/COMM 2 ポートの通信方式設定	5・9
# 223	0	時計機能の選択	5・9
# 224	0	コメントメモリ使用領域の設定	5・10
# 225	0		
# 226	0	一定スキャン時間の設定	5・10
# 227	0	10msタイマ機能の設定	5・11
# 230	300 _{OCT}	下位 キープリレー領域の設定	5・11
# 231	1	上位 (OCT)	
# 232	0	下位 出力保持アドレスの設定	5・12
# 233		上位 (OCT)	
# 234	0	コミュニケーションポート 1 の設定	5・13
# 235	0		
# 236	0	コミュニケーションポート 2 の設定	5・13
# 237	0		
# 240	0	割込処理の設定	5・14
# 241	377 _{OCT}		
# 242	0		
# 243	0		
# 246	010 _{DCML}	瞬停検出時間延長の設定	5・16
# 247	0	I/Oアドレスの登録方法の設定	5・16
# 250	000 _{OCT}	下位 キープリレー領域の設定 (拡張エリア)	5・11
# 251	060 _{OCT}	上位	
# 252	000 _{OCT}	下位 出力保持アドレスの設定 (拡張エリア)	5・12
# 253	060 _{OCT}	上位	
# 255	0	ROM運転モードの設定	5・17
# 256	200 _{OCT}	ROM格納領域の設定	5・17
# 257	不定	BCCチェックコード	5・17
# 260	0	機種モード	5・17

システムメモリ番号はJW30H対応サポートツールにより 8 進 / 10 進 / 16 進に変換できます。

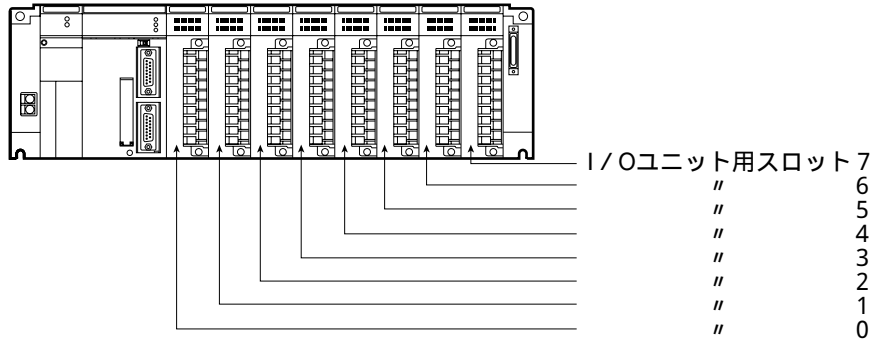
5 - 2 システムメモリの内容

システムメモリ 番号(OCT)	設定項目	内 容																											
# 010	時計機能 [JW-32CUH/H1 JW-33CUH/H1 /H2/H3]	秒：00～59 (BCD)																											
# 011		分：00～59 (BCD)																											
# 012		時：00～23 (BCD)																											
# 013		日：01～31 (BCD) コントロールユニット内の回路により月ごとの「日付」と、「うるう年」の計算を自動的に行います。																											
# 014		月：00～12 (BCD)																											
# 015		年：00～99 (BCD) 西暦の下2桁を表わします。うるう年の自動判別は4年ごとの判別のみです。 (96年、00年、04年はうるう年と判別します。)																											
# 016		曜日：00～06 (BCD) 時刻をセットするときの曜日に合わせてください。日付が変わるごとに00～06に順次変化します。曜日については、年月日設定による計算はしません。																											
		<table border="1"> <thead> <tr> <th>曜 日</th> <th>日</th> <th>月</th> <th>火</th> <th>水</th> <th>木</th> <th>金</th> <th>土</th> </tr> </thead> <tbody> <tr> <td>BCD値</td> <td>00</td> <td>01</td> <td>02</td> <td>03</td> <td>04</td> <td>05</td> <td>06</td> </tr> </tbody> </table>	曜 日	日	月	火	水	木	金	土	BCD値	00	01	02	03	04	05	06											
曜 日		日	月	火	水	木	金	土																					
BCD値		00	01	02	03	04	05	06																					
# 017		<p>コントロール：コントロールは、システムメモリ#017のビットセットで働きます。</p> <table border="1"> <thead> <tr> <th>内容</th> <th>D7</th> <th>D6</th> <th>D5</th> <th>D4</th> <th>D3</th> <th>D2</th> <th>D1</th> <th>D0</th> </tr> </thead> <tbody> <tr> <td>ON</td> <td>時刻合せ</td> <td colspan="3">未使用</td> <td>30秒補正</td> <td colspan="2">未使用</td> <td>時計停止</td> </tr> <tr> <td>OFF</td> <td>時刻モニタ</td> <td colspan="3">未使用</td> <td>—</td> <td colspan="2">未使用</td> <td>時計運転</td> </tr> </tbody> </table> <ul style="list-style-type: none"> ・30秒補正 ... 010octをセットすると0～29秒は0秒となります。30～59秒では0秒となり1分の桁上げとなります。 ・時刻合せ ... 001octで時計を停止し、#010～016に時刻を入力します。つづいて200octを書き込み、時刻をセットします。時刻をセットするとD0とD7はOFFになり時計が動作します。 <p>(注) 001octをセットするとシステムメモリ#010～#016、レジスタ99770～99776の表示は更新しませんが、(見かけ上時計停止)内部での時計は動作しています。</p>	内容	D7	D6	D5	D4	D3	D2	D1	D0	ON	時刻合せ	未使用			30秒補正	未使用		時計停止	OFF	時刻モニタ	未使用			—	未使用		時計運転
内容		D7	D6	D5	D4	D3	D2	D1	D0																				
ON		時刻合せ	未使用			30秒補正	未使用		時計停止																				
OFF		時刻モニタ	未使用			—	未使用		時計運転																				

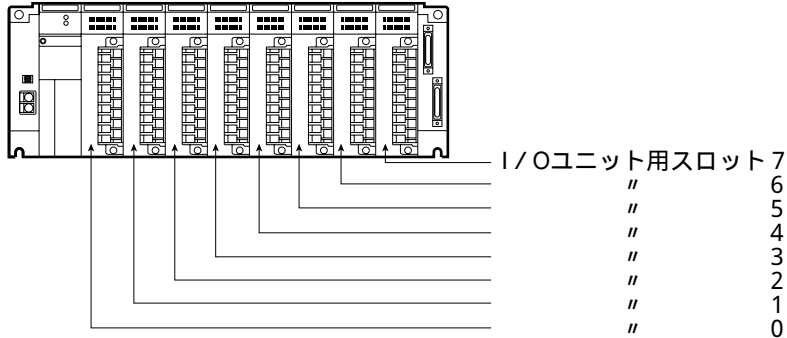
- ・システムメモリ#223が000octのとき、レジスタ99770～99777を使用して、PCのプログラムでも時刻をセットできます。2・8ページ「時計機能で使用する領域」を参照してください。
- ・プログラマ(JW-14PG等)を使用すると「イニシャルモード」で時刻設定できます。
- ・時計機能はバッテリーでバックアップしています。

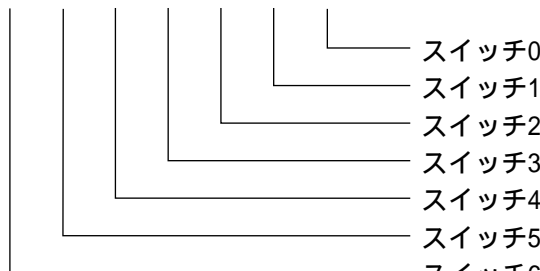
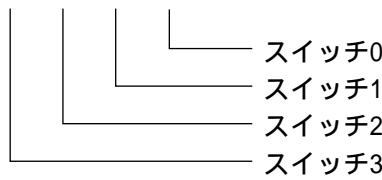
# 046	異常を検知した I/Oスロット番号のモニタ	<p>PCの入出力処理中に下記異常を検出すると、最初に異常検出したラック番号とスロット番号を格納します。異常ユニットを特定するための参考データになります。（異常内容により正しいラック番号とスロット番号が格納されない場合があります）</p> <p style="text-align: right;">異常コード(BCD)</p> <ul style="list-style-type: none"> I/Oデータバスエラー..... 44 出力データチェックエラー..... 42 実装ユニットエラー..... 40 I/Oベースエラー..... 48 テーブル照合エラー..... 60 ユニットNo. スイッチ照合エラー..... 61 テーブル登録エラー..... 70 ユニットなしエラー..... 71 I/O点数オーバーエラー..... 72 ユニットNo. スイッチ設定エラー..... 73 <p>[ラック番号とスロット番号のビット配置]</p> <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 0 5px;">6</td><td style="padding: 0 5px;">5</td><td style="padding: 0 5px;">4</td><td style="padding: 0 5px;">3</td><td style="padding: 0 5px;">2</td><td style="padding: 0 5px;">1</td><td style="padding: 0 5px;">0</td> </tr> <tr> <td style="border: 1px solid black; text-align: center; width: 20px;">4</td><td style="border: 1px solid black; text-align: center; width: 20px;">2</td><td style="border: 1px solid black; text-align: center; width: 20px;">1</td><td style="border: 1px solid black; text-align: center; width: 20px;">8</td><td style="border: 1px solid black; text-align: center; width: 20px;">4</td><td style="border: 1px solid black; text-align: center; width: 20px;">2</td><td style="border: 1px solid black; text-align: center; width: 20px;">1</td> </tr> <tr> <td colspan="3" style="text-align: center;">└──────────┘</td><td colspan="4" style="text-align: center;">└──────────┘</td> </tr> <tr> <td colspan="3" style="text-align: center;">ラック番号</td><td colspan="4" style="text-align: center;">スロット番号</td> </tr> </table> <p>・ラック番号は基本ベースユニットが⁰、増設ベースユニットはラック番号スイッチの設定値になります。なお、I/Oバス拡張アダプタ使用時は、増設ベースユニット(JW-34ZB/36ZB/38ZB)とI/Oバス拡張アダプタ(JW-32EA)のラック番号スイッチの組み合わせでラック番号が決まります。 スロット番号は、基本/増設ベースユニットのI/Oユニット用スロットの番号を格納します。</p>	6	5	4	3	2	1	0	4	2	1	8	4	2	1	└──────────┘			└──────────┘				ラック番号			スロット番号			
		6	5	4	3	2	1	0																						
4	2	1	8	4	2	1																								
└──────────┘			└──────────┘																											
ラック番号			スロット番号																											

[例] 基本ベースユニット : JW-28KB/38KB



増設ベースユニット : JW-38ZB



<p>#050</p>	<p>異常オプションユニットのスイッチ番号のモニタ</p>	<p>自己診断の結果、システムメモリ #160に異常コード53（オプション異常）を格納しているとき、本システムメモリをモニタすると、基本ベースユニットに実装している異常オプションユニットのスイッチ番号を確認できます。</p> <p>スイッチ番号は、ビットパターンで表示します。</p> <p>..... 正常 異常</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table>  <p>[例] 下記のように表示しているときは、スイッチ番号2のオプションユニットが異常であることを示します。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px; background-color: black;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table> <ul style="list-style-type: none"> ・オプションユニットは、基本ベースユニットに最大7台まで実装できます。 ・増設ベースユニットには、オプションユニットを実装できません。 	7	6	5	4	3	2	1	0																
7	6	5	4	3	2	1	0																			
<p>#051</p>	<p>異常I/Oリンク親局ユニットのスイッチ番号のモニタ</p>	<p>自己診断の結果、システムメモリ #160に異常コード53（オプション異常）を格納しているとき、本システムメモリをモニタすると、基本ベースユニットに実装している異常I/Oリンク親局ユニットのスイッチ番号を確認できます。</p> <p>スイッチ番号は、ビットパターンで表示します。</p> <p>..... 正常 異常</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table>  <p>[例] 下記のように表示しているときは、スイッチ番号2のI/Oリンク親局ユニットが異常であることを示します。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px; background-color: black;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table> <ul style="list-style-type: none"> ・I/Oリンク親局ユニットは、基本ベースユニットに最大4台まで実装できます。 ・増設ベースユニットには、I/Oリンク親局ユニットを実装できません。 	7	6	5	4	3	2	1	0																
7	6	5	4	3	2	1	0																			

自己診断の結果、システムメモリ #160に異常コード24（命令コードチェック異常）を格納しているとき、本システムメモリをモニタすると、ユーザープログラム中の異常発生アドレスを確認できます。異常発生アドレスは、ビットパターンで表示します。

..... 0
..... 1

#053 #052

6桁目 5桁目 4桁目 3桁目 2桁目 1桁目

[例] 下記のように表示しているときは、ユーザープログラム中のアドレス1300番地（8進数）が異常であることを示します。

#053 #052

0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0

0 0 1 3 0 0

命令語を3パターンに分類してそれぞれを何進数で表示するかを選択します。また、データメモリ（リレー/タイマ・カウンタ/レジスタ番号）・プログラムアドレス・ラベルの3種類も選択します。

#115 #114

7 6 5 4 3 2 1 0 7 6 5 4 3 2 1 0

データメモリ
プログラムアドレス
システムメモリアドレス
ラベル
パラメータアドレス
命令群A-1
命令群A-2
命令群A-3

各2ビットの設定値の内容は次のとおりです。

各2ビットの設定値	内 容
00	初期値
01	8進表示
10	10進表示
11	16進表示

初期値は #114 の場合、9・19～297ページの各応用命令の説明で記述している進数です。
#115 の場合、8進数です。

アドレス/リレー/レジスタ/ラベル/応用命令にて、8/10/16進の選択

#114
#115

・命令群の分類

A-1群	定数がある転送/比較命令 F-01、F-01w、F-07、F-07w、F-08、F-08w、Fc12、Fc12w、Fx12、Fx12w、F-71、F-71w、F-91、Fc180、Fc180w、Fc181、Fc181w、Fc182、Fc182w、Fc183、Fc183w、Fc184、Fc184w、Fc185、Fc185w
A-2群	ビットパターン指定に定数がある命令 Fc13、Fc13w、Fx13、Fx13w、Fc14、Fc14w、Fx14、Fx14w、Fc17、Fc17w、Fx17、Fx17w、Fc18、Fc18w、Fx18、Fx18w、
A-3群	バイト数指定に定数がある命令 F-67、F-68、F-70、F-70w、F-72、F-72w、F-73、F-73w、F-74、F-74w、F-79、F-79w、F-144、F-174、F-175、F-252、F-253

# 136	ツール機種の 設定	<p>プログラマJW-2PGをJW30Hに接続する場合、JW30Hにツール機種を設定します。</p> <p>002 OCT ... JW-2PGとの接続</p> <p>002 OCT 以外 ... JW-2PG以外との接続（初期値：000OCT）</p>
# 160 ~ # 167	自己診断結果 の異常コード	<p>自己診断の結果、異常と判断した場合、異常内容に応じ、異常コードを格納します。</p> <ul style="list-style-type: none"> ・ # 160 ~ # 167はシフトレジスタとして働き、8回の異常発生を記憶できます。異常コードの詳細は「自己診断」を参照してください。 ・ 異常発生時刻を含めた情報はE7600 ~ E7777に格納します。2・10ページ「異常履歴格納領域」を参照してください。 ・ 異常が解消してもクリアしませんので、クリアする必要があるときは、サポートツールで0を書き込んでください。
# 170 ~ # 177	オプション ユニットの 異常コード	<p>オプションユニットの自己診断の結果、異常と判断した場合、異常内容に応じ異常コードを格納します。</p> <ul style="list-style-type: none"> ・ # 170 ~ # 177はシフトレジスタとして働き、8回の異常発生を記憶できます。異常コードの詳細はリンクユニットJW-21CM等の「取扱説明書」を参照してください。 ・ 異常発生時刻を含めた情報は、E6000 ~ E7577にオプションユニットのユニットNo.スイッチの設定内容ごとに格納します。2・10ページ「異常履歴格納領域」を参照してください。 ・ すべてのオプションユニットが# 170 ~ # 177を共用します。 ・ 異常が解消してもクリアしませんので、クリアする必要があるときは、サポートツールで0を書き込んでください。
# 201	TMRのリセット 条件設定	<p>TMR命令の復電時の状態を設定します。</p> <p>000 OCT ... 復電時リセット（初期状態）</p> <p>001 OCT ... 復電時の状態記憶</p> <ul style="list-style-type: none"> ・ TMR命令はDTMR(BCD)、DTMR(BIN)、UTMR(BCD)、UTMR(BIN)も含まれます。
# 202	CNTのリセット 条件設定	<p>CNT命令、応用命令（下記）のリセット入力条件を設定します。</p> <p>000 OCT ... ONでリセット（初期状態）</p> <p>001 OCT ... OFFでリセット</p> <ul style="list-style-type: none"> ・ CNT命令はDCNT(BCD)、DCNT(BIN)、UCNT(BCD)、UCNT(BIN)も含まれます。 ・ 応用命令の種類 <ul style="list-style-type: none"> F - 6 0 (F / B SFR) F - 1 6 0 (NSFR) F - 6 0 w (F / B SFR) F c 1 6 0 (NSFR) F - 6 0 d (F / B SFR) F - 2 6 1 (RCNT) F - 6 2 (U / DC) F c 2 6 1 (RCNT) F - 6 2 w (U / DC) F - 6 2 d (U / DC)
# 206	ヒューズ断検出 時運転継続 / 停 止の設定	<p>出力ユニット（JW-262S）でヒューズ断（異常コード49）を検出したときPC運転の継続又は停止を設定します。</p> <p>000 OCT 運転継続</p> <p>010 OCT 停止</p> <p>初期状態は0 0 0に設定されています。</p>

# 207	オプションユニット異常時 運転継続 / 停止 の設定	<p>オプションユニットでユニット異常（異常コード53）を検出したときPC運転の継続又は停止を設定します。 設定はオプションユニットのユニットNo. スイッチ番号のビットをONすると運転継続になります。</p> <p style="text-align: center;">  </p> <p style="text-align: right;">..... 停止 運転継続</p> <p style="text-align: right;">オプションユニットのユニット No. スイッチ番号</p> <p>初期状態は0 0 0で全ビット停止状態（ ）です。</p>								
# 210	異常格納領域の 選択	<p>異常コードを格納するレジスタ領域を選択します。 000 OCT ... 非使用 002 OCT ... レジスタE6000 ~ E7777 OCT（初期値：002OCT）</p>								
# 211	I/Oリンク親局 ユニット異常時 運転継続 / 停止 の設定	<p>I/Oリンク親局ユニットでユニット異常（異常コード53）を検出したときPC運転の継続又は停止を設定します。 設定はI/Oリンク親局ユニットのユニットNo. スイッチ番号のビットをONすると運転継続になります。</p> <p style="text-align: center;">  </p> <p style="text-align: right;">..... 停止 運転継続</p> <p style="text-align: right;">I/Oリンク親局ユニットのユニット No. スイッチ番号</p> <p>初期状態は0 0 0で全ビット停止状態（ ）です。</p>								
# 220	コメントメモ リ使用領域の 先頭ファイル 番号の設定 [JW-33CUH2/H3]	<p>ファイル番号が2桁のファイルメモリをコメントメモリに使用する場 合、先頭ファイル番号01 ~ 03、10 ~ 2CHEXを設定します。 ただし、その場合は# 224の先頭ファイル番号は0にしておく必要があ ります。</p>								
# 222	PG / COMM 2 ポートの通信 方式設定 [JW-32CUH/H1 JW-33CUH/H1 /H2/H3]	<p>コミュニケーションポート 2（PG / COMM 2 ポート）の通信方式を 設定します。</p> <table border="1" data-bbox="635 1377 1453 1646"> <thead> <tr> <th>設定値(HEX)</th> <th>通信方式</th> </tr> </thead> <tbody> <tr> <td>0 0</td> <td>RS-232C（RTS信号は通電中ON） RS-422A（4線式の1：1接続のみ）</td> </tr> <tr> <td>0 2</td> <td>RS-232C接続のみ（RTS信号はデータ送信中OFF、 データ送信中以外ON） ・RTS信号でフロー制御時に設定してください。 （注）サポートツールを使用できません。</td> </tr> <tr> <td>0 4</td> <td>RS-422A接続のみ（4線式の1：N接続が可能）</td> </tr> </tbody> </table> <p>02と04の設定は、JW30HコントロールユニットのJW-32CUH1/33CUH1/ 33CUH2/33CUH3ではソフトウェアバージョンがVer3.5以上、JW-32CUH/ 33CUHではVer3.0以上で可能です。</p>	設定値(HEX)	通信方式	0 0	RS-232C（RTS信号は通電中ON） RS-422A（4線式の1：1接続のみ）	0 2	RS-232C接続のみ（RTS信号はデータ送信中OFF、 データ送信中以外ON） ・RTS信号でフロー制御時に設定してください。 （注）サポートツールを使用できません。	0 4	RS-422A接続のみ（4線式の1：N接続が可能）
設定値(HEX)	通信方式									
0 0	RS-232C（RTS信号は通電中ON） RS-422A（4線式の1：1接続のみ）									
0 2	RS-232C接続のみ（RTS信号はデータ送信中OFF、 データ送信中以外ON） ・RTS信号でフロー制御時に設定してください。 （注）サポートツールを使用できません。									
0 4	RS-422A接続のみ（4線式の1：N接続が可能）									
# 223	時計機能の選択 [JW-32CUH/H1 JW-33CUH/H1 /H2/H3]	<p>時計機能をレジスタ上でコントロールするための設定です。使用状態 にするとPCプログラムで時刻合せができます。 000 OCT ... レジスタ使用（初期状態） 001 OCT ... 非使用 レジスタは99770 ~ 99777を使用します。2・8ページ「時計機能で使用する 領域」を参照してください。 ・ JW-31CUH/H1を使用しているときのシステムメモリ# 223は予約領域です。</p>								

<p># 224 # 225</p>	<p>コメントメモリ 使用領域の設定</p>	<p>多機能プログラマおよびJWモデル用ラダーソフトでリレー・タイマ、カウンタ・レジスタ及びF-90 (REM) 命令のコメントをPCに書き込む (PCから読み出す) コメントメモリの先頭アドレスとバイト数を設定します。</p> <p style="text-align: center;">7 6 5 4 3 2 1 0</p> <p># 224 <table border="1" style="display: inline-table;"><tr><td style="width: 50px;">先頭ファイル番号</td><td style="width: 50px;">コメント種類</td></tr></table></p> <p># 225 <table border="1" style="display: inline-table;"><tr><td style="width: 100px;">コメントメモリバイト数</td></tr></table></p> <ul style="list-style-type: none"> 先頭ファイル番号 コメントに使用するファイル番号(1,2,3)です。ファイルアドレスは000000から使用します。 <p>(注1) JW-33CUH2/H3でファイル番号10~2CH_{HEX}を先頭ファイル番号にする場合、# 220に設定します。その場合# 224の先頭ファイル番号は0にしておく必要があります。</p> <ul style="list-style-type: none"> コメント種類 <ul style="list-style-type: none"> 1 HEX JW-92SP(V4.0A以前)のシンボル・コメントでシンボル (半角6文字)のみをPC本体に設定します。(8バイト単位) 3 HEX JW-92SP(V4.0A以前)のシンボル・コメントでシンボル (半角6文字)とコメント (半角24文字)をPC本体に設定します。(32バイト単位) 8 HEX JW-92SP(V5.0以降) JW-52SPのシンボル・コメントでシンボル (半角16文字)のみPC本体に設定します。(20バイト単位) A HEX JW-92SP(V5.0以降) JW-52SPのシンボル・コメントでシンボル (半角16文字)とコメント (半角28文字)をPC本体に設定します。(48バイト単位) コメントメモリバイト数 8Kバイト単位です。設定値×8Kバイトになります。 <p>初期値は# 224、# 225ともに000です。</p> <p>(注2) 機種により、コメントメモリとして使用できるファイルメモリが異なります。</p> <p>(注3) ファイル1の次はファイル2、ファイル2の次はファイル3、ファイル3の次はファイル10とコメントメモリバイト数に応じてファイル間を連続で使用されます。</p>	先頭ファイル番号	コメント種類	コメントメモリバイト数
先頭ファイル番号	コメント種類				
コメントメモリバイト数					

<p># 226</p>	<p>コンスタント スキャン</p>	<p>PCのスキャンタイムを任意に設定できます。設定はBCD値で01~99msです。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td># 0226設定</td> <td>PCのスキャン時間</td> </tr> <tr> <td>00(BCD)</td> <td>最小スキャン</td> </tr> <tr> <td>01~99(BCD)</td> <td>01~99msのスキャンタイム</td> </tr> </table> <p>初期状態は00に設定されています。</p> <p>スキャンタイムの設定よりプログラム演算が長くかかるときは、プログラム演算時間でPCスキャンタイムが決まります。</p> <p>PCスキャンタイムは入出力処理開始と同時に時間を測定しています。</p> <div style="text-align: center;"> </div> <p>設定時間n精度 = 設定値 ± 1 msです。</p>	# 0226設定	PCのスキャン時間	00(BCD)	最小スキャン	01~99(BCD)	01~99msのスキャンタイム
# 0226設定	PCのスキャン時間							
00(BCD)	最小スキャン							
01~99(BCD)	01~99msのスキャンタイム							

227 10msタイマ機能の選択

ビットのON / OFFでTMR0400 ~ 0777の各領域を、10ms / 100msタイマに設定できます。

7	6	5	4	3	2	1	0
□	□	□	□	0	0	0	0

TMR0400 ~ 0477 (1(ON)...10msタイマ)
 TMR0500 ~ 0577 (0(OFF)...100msタイマ)
 TMR0600 ~ 0677 初期状態 : 000oct
 TMR0700 ~ 0777

345octに設定するとTMR0700 ~ 0777のみ10msタイマになります。
 DTMR、UTMRでは100msタイマとしてのみ動作し、10msタイマになりません。

230 # 231 キープリレー領域の設定

キープリレー領域を初期状態から増減したい場合に設定します。設定は8点単位で行い、設定数値はファイルアドレスを8進数で設定します。

	設定範囲(ワード、8進)	バイトアドレス
#230、#231	000000 ~ 001577	コ0000 ~ コ1577
#250、#251	030000 ~ 035577	コ2000 ~ コ7577

キープリレー領域は設定したバイトアドレスのリレー以降最終バイトアドレス(コ1577、コ7577)のリレーとなります。

[例] 02000以降(02000 ~ 15777)をキープリレーに設定する場合

#231	#230	
0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0	#230 = 200oct
		#231 = 000oct
0	2	
↓		
ファイルアドレス000200 (バイトアドレスコ0200)		

[例] 07000以降(07000 ~ 15777)をキープリレーに設定する場合 (初期状態)

#231	#230	(初期値)
0 0 0 0 0 0 0 1	1 1 0 0 0 0 0 0	#230 = 300oct
		#231 = 001oct
0	7	
↓		
ファイルアドレス000700 (バイトアドレスコ0700)		

[例] 20000以降(20000 ~ 75777)をキープリレーに設定する場合 (初期状態)

#251	#250	(初期値)
0 0 1 1 0 0 0 0	0 0 0 0 0 0 0 0	#250 = 000oct
		#251 = 060oct
0	3	
↓		
ファイルアドレス030000 (バイトアドレスコ2000)		

250 # 251 キープリレー領域の設定 (拡張リレーエリア)

# 232 # 233	出力保持 アドレスの 設定	<p>本体停止時に出力ユニットの出力を保持する出力リレーの先頭アドレスを設定します。</p> <p>設定は8点単位で行い、設定数値はファイルアドレスを8進数で設定します。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;"></th> <th style="width: 40%;">設定範囲(ワード、8進)</th> <th style="width: 45%;">バイトアドレス</th> </tr> </thead> <tbody> <tr> <td>#232、#233</td> <td>000000 ~ 001577</td> <td>コ0000 ~ コ1577</td> </tr> <tr> <td>#252、#253</td> <td>030000 ~ 035577</td> <td>コ2000 ~ コ7577</td> </tr> </tbody> </table> <p>出力保持領域は設定したバイトアドレスのリレー以降最終バイトアドレス(コ1577、コ7577)のリレーとなります。</p> <p>[例] 00200以降(00200 ~ 15777)を出力保持に設定する場合</p> <div style="display: flex; align-items: center;"> <table border="1" style="margin-right: 10px;"> <tr><td style="text-align: center;">#233</td><td style="text-align: center;">#232</td></tr> <tr><td style="text-align: center;">0 0 0 0 0 0 0 0</td><td style="text-align: center;">0 0 0 1 0 0 0 0</td></tr> </table> <div style="margin-left: 20px;"> <p>0 0 0 0 0 2 0</p> <p>ファイルアドレス000020 (バイトアドレスコ0020)</p> </div> <div style="margin-left: 20px; border-left: 1px solid black; padding-left: 5px;"> <p>#232 = 020oct</p> <p>#233 = 000oct</p> </div> </div> <p>[例] 00000以降(00000 ~ 15777)を出力保持に設定する場合 (初期状態)</p> <div style="display: flex; align-items: center;"> <table border="1" style="margin-right: 10px;"> <tr><td style="text-align: center;">#231</td><td style="text-align: center;">#230</td></tr> <tr><td style="text-align: center;">0 0 0 0 0 0 0 0</td><td style="text-align: center;">0 0 0 0 0 0 0 0</td></tr> </table> <div style="margin-left: 20px;"> <p>0 0 0 0 0 0 0</p> <p>ファイルアドレス000000 (バイトアドレスコ0000)</p> </div> <div style="margin-left: 20px; border-left: 1px solid black; padding-left: 5px;"> <p>(初期値)</p> <p>#232 = 000oct</p> <p>#233 = 000oct</p> </div> </div> <p>[例] 20000以降(20000 ~ 75777)を出力保持に設定する場合 (初期状態)</p> <div style="display: flex; align-items: center;"> <table border="1" style="margin-right: 10px;"> <tr><td style="text-align: center;">#253</td><td style="text-align: center;">#252</td></tr> <tr><td style="text-align: center;">0 0 1 1 0 0 0 0</td><td style="text-align: center;">0 0 0 0 0 0 0 0</td></tr> </table> <div style="margin-left: 20px;"> <p>0 3 0 0 0 0</p> <p>ファイルアドレス030000 (バイトアドレスコ2000)</p> </div> <div style="margin-left: 20px; border-left: 1px solid black; padding-left: 5px;"> <p>(初期値)</p> <p>#252 = 000oct</p> <p>#253 = 060oct</p> </div> </div>		設定範囲(ワード、8進)	バイトアドレス	#232、#233	000000 ~ 001577	コ0000 ~ コ1577	#252、#253	030000 ~ 035577	コ2000 ~ コ7577	#233	#232	0 0 0 0 0 0 0 0	0 0 0 1 0 0 0 0	#231	#230	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	#253	#252	0 0 1 1 0 0 0 0	0 0 0 0 0 0 0 0	# 252 # 253	出力保持 アドレスの 設定 (拡張リレーエリア)	<p>出力保持を解除した領域は、PCをプログラムモードに切換えたり、自己診断の結果PCの運転が停止したときに出力ユニットのラッチをリセットし、出力がOFFとなります。</p> <p>(例)</p> <div style="display: flex; align-items: center;"> <table border="1" style="margin-right: 10px;"> <tr><td style="text-align: center;">00100</td><td style="text-align: center;">00011</td></tr> <tr><td style="text-align: center;"> </td><td style="text-align: center;">○ </td></tr> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">モータモード</th> <th style="width: 20%;">プログラムモード</th> <th style="width: 20%;">モータモード</th> <th style="width: 20%;">プログラムモード</th> <th style="width: 20%;">モータモード</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">PC 運転</td> <td style="text-align: center;">PC 停止</td> <td style="text-align: center;">PC 運転</td> <td style="text-align: center;">PC 停止</td> <td style="text-align: center;">PC 運転</td> </tr> </tbody> </table> <div style="margin-top: 10px;"> <p>入力 00100</p> <p>出力 00011</p> <p>(a)00011 が出力保持領域にあるとき</p> <p>(b)00011 が出力保持解除領域にあるとき</p> </div> </div> <p>自己診断の結果、異常を検知しPCの運転が停止する場合、異常内容によっては、出力保持を解除した領域の出力ユニットをOFFにできない場合があります。PC異常時にOFFにする必要がある出力は、電源ユニットの停止出力を直列に接続してください。</p>	00100	00011		○	モータモード	プログラムモード	モータモード	プログラムモード	モータモード	PC 運転	PC 停止	PC 運転	PC 停止	PC 運転
			設定範囲(ワード、8進)	バイトアドレス																																				
#232、#233	000000 ~ 001577	コ0000 ~ コ1577																																						
#252、#253	030000 ~ 035577	コ2000 ~ コ7577																																						
#233	#232																																							
0 0 0 0 0 0 0 0	0 0 0 1 0 0 0 0																																							
#231	#230																																							
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0																																							
#253	#252																																							
0 0 1 1 0 0 0 0	0 0 0 0 0 0 0 0																																							
00100	00011																																							
	○																																							
モータモード	プログラムモード	モータモード	プログラムモード	モータモード																																				
PC 運転	PC 停止	PC 運転	PC 停止	PC 運転																																				

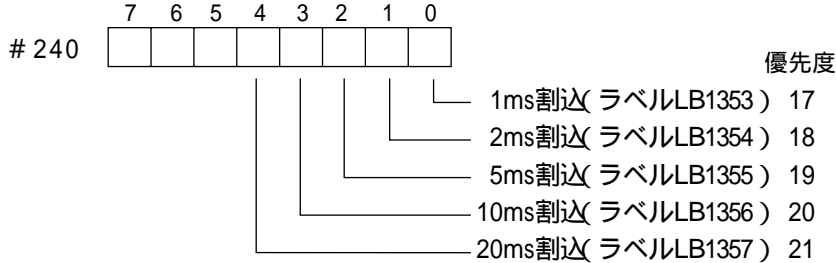
<p># 234 # 235</p>	<p>コミュニケーションポート1 の設定</p> <p>[JW-32CUH/H1 JW-33CUH/H1 /H2/H3]</p>	<p>コミュニケーションポート1 (PG / COMM1ポート) の使用条件を設定します。 #234はD₀ ~ D₅のビットを設定します。</p> <p>7 6 5 4 3 2 1 0 # 234 <input type="text"/> - <input type="text"/> - <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/></p> <p>送信速度(600 ~ 19200ビット/s) パリティ(なし、奇数、偶数) ストップビット(1ビット、2ビット)</p> <table border="1"> <thead> <tr> <th>データ</th> <th>D₅</th> <th>ストップ</th> <th>D₄ D₃</th> <th>パリティ</th> <th>D₂ D₁ D₀</th> <th>送信速度</th> </tr> </thead> <tbody> <tr> <td>7ビット 固定</td> <td>0</td> <td>1ビット</td> <td>0 0</td> <td>なし</td> <td>0 0 0</td> <td>19200ビット/s</td> </tr> <tr> <td></td> <td>1</td> <td>2ビット</td> <td>0 1</td> <td>奇数</td> <td>0 0 1</td> <td>9600</td> </tr> <tr> <td></td> <td></td> <td></td> <td>1 0</td> <td>偶数</td> <td>0 1 0</td> <td>4800</td> </tr> <tr> <td></td> <td></td> <td></td> <td>1 1</td> <td>不可</td> <td>0 1 1</td> <td>2400</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>1 0 0</td> <td>1200</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td>*1</td> <td>1 0 1</td> <td>115200</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td>*2</td> <td>1 1 0</td> <td>57600</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td>*2</td> <td>1 1 1</td> <td>38400</td> </tr> </tbody> </table> <p>(注) 送信速度の設定で、JW-32CUH/33CUH(従来機)には *2の設定はありません。また、*1に設定すると600ビット/sとなります。 コミュニケーションポートはリンクユニットJW-21CMのコマンドモードと同じ通信内容のため #235に局番を設定します。</p> <p># 235 <input type="text"/> 局 番</p> <p>設定範囲は001 ~ 037_{OCT}です。 初期値は # 234、# 235ともに000です。</p>	データ	D ₅	ストップ	D ₄ D ₃	パリティ	D ₂ D ₁ D ₀	送信速度	7ビット 固定	0	1ビット	0 0	なし	0 0 0	19200ビット/s		1	2ビット	0 1	奇数	0 0 1	9600				1 0	偶数	0 1 0	4800				1 1	不可	0 1 1	2400						1 0 0	1200					*1	1 0 1	115200					*2	1 1 0	57600					*2	1 1 1	38400
データ	D ₅	ストップ	D ₄ D ₃	パリティ	D ₂ D ₁ D ₀	送信速度																																																											
7ビット 固定	0	1ビット	0 0	なし	0 0 0	19200ビット/s																																																											
	1	2ビット	0 1	奇数	0 0 1	9600																																																											
			1 0	偶数	0 1 0	4800																																																											
			1 1	不可	0 1 1	2400																																																											
					1 0 0	1200																																																											
				*1	1 0 1	115200																																																											
				*2	1 1 0	57600																																																											
				*2	1 1 1	38400																																																											
<p># 236 # 237</p>	<p>コミュニケーションポート2 の設定</p> <p>[JW-32CUH/H1 JW-33CUH/H1 /H2/H3]</p>	<p>コミュニケーションポート2 (COMM2ポート) の使用条件を設定します。 #236はD₀ ~ D₅のビットを設定します。</p> <p>7 6 5 4 3 2 1 0 # 236 <input type="text"/> - <input type="text"/> - <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/></p> <p>送信速度(600 ~ 19200ビット/s) パリティ(なし、奇数、偶数) ストップビット(1ビット、2ビット)</p> <table border="1"> <thead> <tr> <th>データ</th> <th>D₅</th> <th>ストップ</th> <th>D₄ D₃</th> <th>パリティ</th> <th>D₂ D₁ D₀</th> <th>送信速度</th> </tr> </thead> <tbody> <tr> <td>7ビット 固定</td> <td>0</td> <td>1ビット</td> <td>0 0</td> <td>なし</td> <td>0 0 0</td> <td>19200ビット/s</td> </tr> <tr> <td></td> <td>1</td> <td>2ビット</td> <td>0 1</td> <td>奇数</td> <td>0 0 1</td> <td>9600</td> </tr> <tr> <td></td> <td></td> <td></td> <td>1 0</td> <td>偶数</td> <td>0 1 0</td> <td>4800</td> </tr> <tr> <td></td> <td></td> <td></td> <td>1 1</td> <td>不可</td> <td>0 1 1</td> <td>2400</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>1 0 0</td> <td>1200</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td>*1</td> <td>1 0 1</td> <td>115200</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td>*2</td> <td>1 1 0</td> <td>57600</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td>*2</td> <td>1 1 1</td> <td>38400</td> </tr> </tbody> </table> <p>(注) 送信速度の設定で、JW-32CUH/33CUH(従来機)には *2の設定はありません。また、*1に設定すると600ビット/sとなります。 コミュニケーションポートはリンクユニットJW-21CMのコマンドモードと同じ通信内容のため #237に局番を設定します。</p> <p># 237 <input type="text"/> 局 番</p> <p>設定範囲は001 ~ 037_{OCT}です。 初期値は # 236、# 237ともに000です。</p>	データ	D ₅	ストップ	D ₄ D ₃	パリティ	D ₂ D ₁ D ₀	送信速度	7ビット 固定	0	1ビット	0 0	なし	0 0 0	19200ビット/s		1	2ビット	0 1	奇数	0 0 1	9600				1 0	偶数	0 1 0	4800				1 1	不可	0 1 1	2400						1 0 0	1200					*1	1 0 1	115200					*2	1 1 0	57600					*2	1 1 1	38400
データ	D ₅	ストップ	D ₄ D ₃	パリティ	D ₂ D ₁ D ₀	送信速度																																																											
7ビット 固定	0	1ビット	0 0	なし	0 0 0	19200ビット/s																																																											
	1	2ビット	0 1	奇数	0 0 1	9600																																																											
			1 0	偶数	0 1 0	4800																																																											
			1 1	不可	0 1 1	2400																																																											
					1 0 0	1200																																																											
				*1	1 0 1	115200																																																											
				*2	1 1 0	57600																																																											
				*2	1 1 1	38400																																																											

240
241
242
243

割込処理の設定

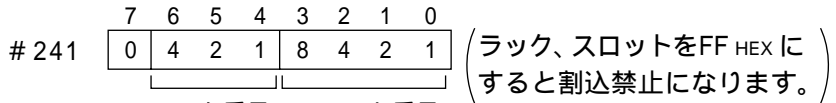
PCの割込演算条件を設定します。タイマ割込と入力割込が21種類あります。割込は、PC演算中にはもちろん、I/O処理中にも実行します。各割込がかかると指定ラベルのサブルーチンをコールします。

- ・ # 240は各ビットごとに割込時間が異なります。ラベルの番号は割込時のサブルーチンラベルです。5種類の時間を全て使用できます。

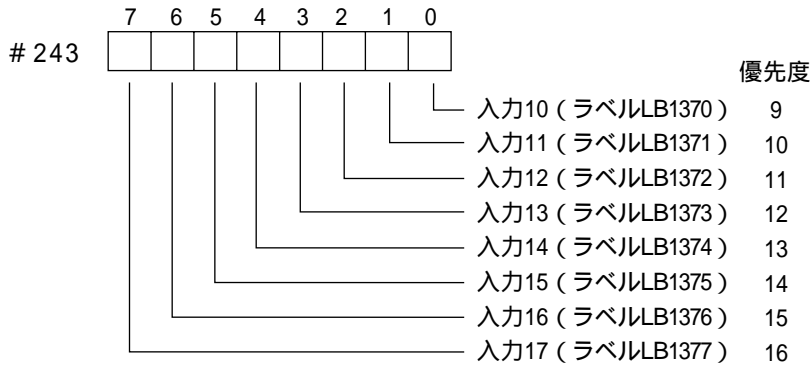
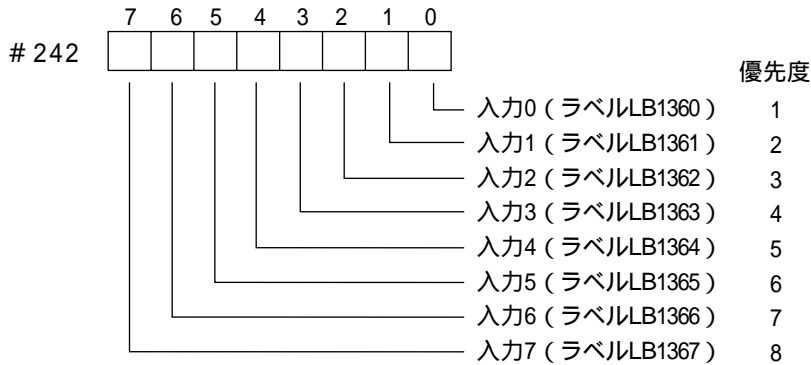


は使用 は非使用です。

- ・ # 241 ~ # 243は入力ユニットの入力割込条件を設定します。PCはラック番号とスロット番号で指定した入力ユニットを1msごとにモニタし、入力信号の立上り(OFF ON)または立下り(ON OFF)で割込を検出し指定ラベル(F-140)のサブルーチンをコールします。
- ・ # 241は割込に使用する入力ユニットのラック番号とスロット番号を設定します。入力ユニットが32点ユニットのときは、前半の16点を割込用入力として使用できます。
- ・ # 242、# 243は、# 241で指定した入力信号16点の立上りまたは立下りのどちらで割込用サブルーチンをコールするかを設定します。
- ・ # 242、243で指定する各ビットは、入力ユニットの16点に対応しているとともに各ビットには、使用するサブルーチンラベルが指定されています。



ラック番号 スロット番号
0~7 0~7



は0 ... 立下り(ON OFF)で割込みます。

は1 ... 立上り(OFF ON)で割込みます。

240、# 242、# 243の初期値は000、# 241の初期値はFFHEXです。

留意点

240
241
242
243

割込処理の設定

- ・割込はF-14〔CALL〕命令やF-14〔CAL+〕と同じ使い方をします。注意事項も同じです。
- ・サブルーチンのラベルはF-4〔END〕以降に設けてプログラムを作ってください。
- ・割込プログラムは、1ms以内の演算にしてください。1ms以上だと割込が無視されるときがあります。
- ・入力割込のラック、スロット番号に出力ユニットや特殊I/Oユニットを実装しても割込が働きますのでユニット実装に注意してください。
- ・リモートI/O子局ユニットJW-21RSに実装した入力ユニットの入力割込は行いません。
- ・タイマ割込、入力割込ではジャンプ先ラベル(F140で指定)が無いとき割込は無視されます。
- ・割込が複数個同時に発生した場合、優先度の高い(数値の小さい)ものから処理します。
- ・入出力ユニット等とのデータ交換中に割込が発生した場合は、次のように実行します。

	JW-31CUH1 JW-32CUH1 JW-33CUH1/H2/H3	JW-31CUH JW-32CUH JW-33CUH
入出力ユニット 特殊I/Oユニット	1	1
オプションユニット	2	

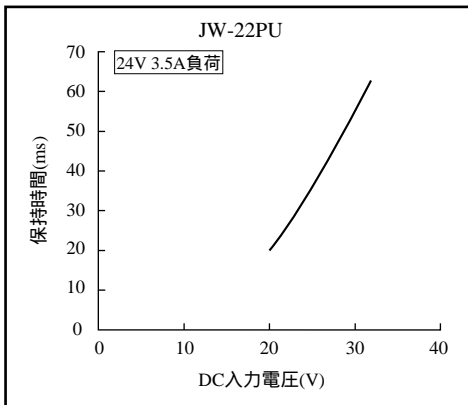
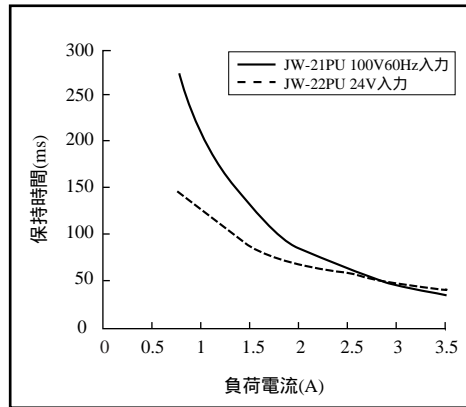
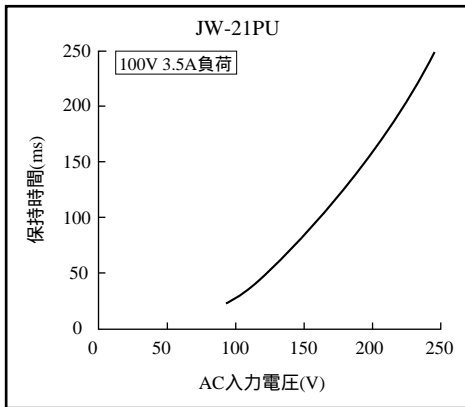
1 データ交換が終了後に、割込を実行します。

2 データ交換中に、割込を実行します。

- ・割込を設定中のときはブ레이크機能ははたらきません。
- ・入力ユニットが32点ユニットのとき、後半の16点も1ms毎にリフレッシュされますので、レーシング(6.5ページ参照)に注意してください。
- ・64点入力ユニットは割込入力に使用できません。
- ・割込プログラム内で、サブルーチンコール命令(F-142〔CALL〕、F-148〔CAL+〕)は使用できません。

# 246	瞬停検出時間 延長の設定	瞬停検出時間を0～255msまで可変するときには設定します。設定数値は10進で設定します。 初期設定は010（10ms）に設定されています。
-------	-----------------	---

- ・システムメモリ # 246（瞬停検出時間延長の設定）の設定を可変するときには、電源入力電圧及び負荷電流によって可変可能な範囲が決まります。下記の保持時間特性、出力保持時間特性のグラフを参照して瞬停検出時間の延長を行ってください。
- ・瞬停中、コントロールユニットのすべての動作は停止し、復旧後、連続した動作を行います。また、瞬停検出時間設定以上（停電）での復旧後は、電源投入時の処理を行います。
- ・基本ベースユニット及び増設ベースユニットでの5V容量とのかね合いのため、十分注意してご使用ください。



# 247	I/Oアドレスの 登録方法の 設定	<p>I/Oアドレスの登録方法を設定します。</p> <p>000 電源ON時自動登録 ・基本ベースから増設ベース（ラック番号順）に、リレー番号が00000_{oct}から連続して割り付けられます。</p> <p>003_{oct} 電源ON時の自動登録が禁止されます。 自動登録後003を書いて自動登録を禁止にして運転を行ってください。</p> <p>004_{oct} ラック先頭アドレスを設定可 ・ラック(1～7)ごとの先頭アドレスの設定が有効となります。 ・先頭アドレスの設定は、サポートツール（JW-14PG/13PG、JW-100SP、JW-52SP/92SP）で行います。</p>
-------	-------------------------	---

# 255	ROM運転 モードの設定	ROM運転モードを設定します。				
		設定値 OCT HEX	ROM RAM 転送 (電源ON時)	電源ON 転送後の データメモリ	電源ON 転送後の モード	ツールによる ROM RAM 転送
		000 00	しない	-	-	可
		021 11	する	保持	電源OFF時のモード	可
		042 22	する	クリア	停止	可
		104 44	する	クリア	運転	可
		ROM内に格納されているデータは保持されます。 初期値は000OCTです。 詳細はユーザズマニュアル・ハード編の「ROM運転」を参照してください。				
# 256	ROM化内容 の設定	ROM運転を行うときのROM化内容を設定します。				
		設定値 OCT HEX	ROM化される内容 [: ROM化される、x : ROM化されない]			
		システムメモリ (#200 ~ #2177)	プログラム (7.5K語 ~ 63.0K語)	レジスタ (8Kバイト) (*1)	ファイル1 (16Kバイト)	ファイル2 (64Kバイト) (*2)
		200 80		x	x	x
		201 81			x	x
		202 82		x		
		203 83	x		x	x
		204 84	x	x		
		205 85				x
		206 86	x			
		(*1) 09000 ~ 99777、E0000 ~ E5777の8 Kバイトです。E6000 ~ E7777はROM化されません。 (*2) JW-32CUH1でファイル容量32Kバイト(プログラム容量31.5K語)設定時は、ファイル2のROM化容量は32Kバイトとなります。 ファイル3、10 ~ 2CはROM化されません。 初期値は200OCTです。				

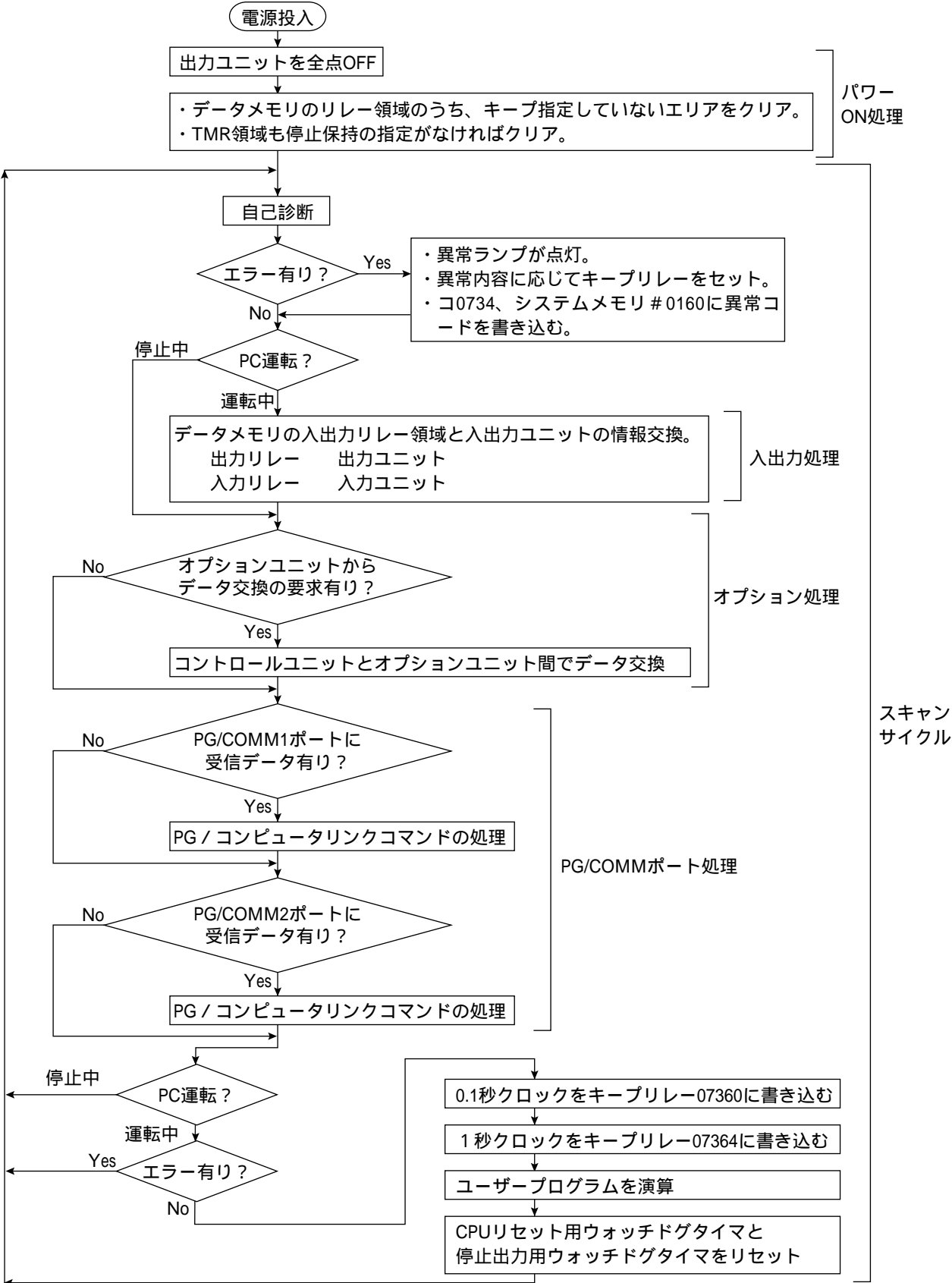
# 257	BCCチェック	システムメモリ # 200 ~ 256までのBCCチェックコードを、JW30Hが自動計算して登録します。
-------	---------	--

# 260	PC機種モード	JW30Hに未対応のサポートツールを使用する場合、サポートツールがJW30Hをどの機種として認識するかを設定します。 50HEX(120OCT)... JW50H / 70H / 100Hとして認識 サポートツールで操作できるのは、JW50H / 70H / 100Hの機能範囲内となる。 50HEX 以外..... JW20Hとして認識 サポートツールで操作できるのは、JW20Hの機能範囲内となる。
-------	---------	---

第 6 章 コントロールユニットの動作

6-1 運転サイクル

〔1〕動作フローチャート



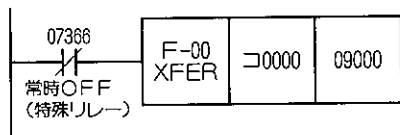
F-80命令を使用すると、演算処理中でも入出力を処理できます。
 入出力/オプション/PG/COMMポートの処理中でも、割込機能(#240~#243)でプログラムを演算できます。

〔 2 〕 パワー ON 処理

電源を投入すると、データメモリをイニシャライズします。このイニシャライズの結果、データメモリは次の様になります。

データメモリ	アドレス	内 容
リ レ ー	00000 ~ 15777 20000 ~ 75777	システムメモリ#230、#231(00000 ~ 15777)、#250、#251(20000 ~ 75777)にキープ機能の開始アドレスを指定できます。 キープ機能指定以前のアドレス 全てOFF キープ機能指定以後のアドレス 停電前のON/OFF状態を保持
TMR、CNT、 MD	0000 ~ 1777	〔 TMR 〕 システムメモリ#201に電源投入時の状態を指定できます。 000.....現在値は設定値になります。TMR接点はリセットします。 001.....現在値は停電前の値を保持。TMR接点は停電前のON/OFF状態を保持。
		〔 CNT 〕 現在値は停電前の状態を保持。 CNT接点も停電前の状態を保持。
		〔 MD 〕 MDデータ、入力情報とも停電前の状態を保持。
レ ジ ス タ	09000 ~ 99777 E0000 ~ E7777	停電前の値を保持。
特殊ユニット用 パラメータ	各000 ~ 177	設定内容を保持。
オプションユニット用 パラメータ	各000 ~ 077	設定内容を保持。
フ ァ イ ル 1	000000 ~ 037777	停電前の値を保持。
フ ァ イ ル 2	000000 ~ 177777	停電前の値を保持。
フ ァ イ ル 3	000000 ~ 177777	停電前の値を保持。
フ ァ イ ル 10 } フ ァ イ ル 2C	000000 ~ 177777	停電前の値を保持。

- 電源投入時、イニシャライズ処理の前に、各出力ユニット内の出力データ用ラッチをリセットし、全出力はOFFとなります。
- 電源投入時、データメモリはイニシャライズしますが、最初のスキャンサイクルの入出力処理によってデータメモリの入出力リレー領域は次の様に変化します。
 - (1)入力ユニットを装着している領域
入力ユニットに接続した入力機器（リミットスイッチ等）のON/OFF状態にしたがってON又はOFFとなります。
 - (2)出力ユニットを装着している領域および入出力ユニット未装着の領域
ユーザープログラムの演算に入るまでイニシャライズ処理の状態から変化しません。
 - (3)入力ユニットのダミー領域
ユーザープログラムの演算に入るまでイニシャライズ処理の状態から変化しません。
- ROM運転時のパワーON処理については、「ユーザーズマニュアル・ハード編」を参照して下さい。
- 立上り（OFF→ON）で動作する応用命令は最初のスキャンでは演算完了と同じ状態になっており下記回路では演算しません。



(3) スキャンサイクル

パワーON処理が終了と、スキャンサイクルに入ります。スキャンサイクルはハードウェアチェックからプログラム終了（F-40のEND命令が書かれているステップの実行）までで構成し、プログラム終了後は再びハードウェアチェックに戻り以下この動作を繰り返します。この1サイクルに要する時間をスキャンタイムと呼びます。

- (1)ハードウェアチェック
コントロールユニットのハードウェアが正常に機能することを自己診断します。
 - a.RAMチェック
データメモリ用RAMが書き込み、読み出し可能であるかチェックします。
 - データメモリ用RAMのチェック専用領域を使います。
 - b.ハードウェア動作チェック
ビット処理（AND、OR等の演算）用のアキュムレータ、スタックが正しく動作するかチェックします。
 - c.サムチェック機能の動作チェック
命令の演算実行開始時にプログラムメモリのサムチェックを行います。このチェックはハードウェアで行っています。このハードウェアが正しく機能しているかチェックします。
 - d.データバスのチェック
ハードウェアチェックの段階では入出力ユニットとデータの交換を行う入出力データバスはフローティング状態になっているのが正常です。もしフローティング状態でなければ入出力用データバス異常として処理します。
 - e.システムメモリ設定チェック
システムメモリ#200～#256までのサムチェックコードを#257に格納し毎サイクル、サムチェックが正しいかチェックします。
- 自己診断は、上記の5種類以外に次の各項目がありません。
 - ①サムチェック
 - ②出力データチェック
 - ③実装ユニットチェック
 - ④I/Oベースチェック
 - ⑤テーブル照合エラーチェック
 - ⑥ユニットNo.スイッチ照合チェック
 - ⑦テーブル登録チェック
 - ⑧ユニットなしエラーチェック
 - ⑨I/O点数チェック
 - ⑩ユニットNo.スイッチ設定チェック
 - ⑪特殊I/O異常
 - ⑫オプション異常
 - ⑬電源異常
 - ⑭増設電源異常

6・8ページ「自己診断」を参照してください。

(2) フラグのクリア

データ処理命令には、演算の結果、フラグ(Flag)に影響を与えるものがあります。毎スキャンサイクルのユーザープログラム処理の前にフラグをクリアします。フラグに関しては9・7ページ「データ処理命令とフラグ」を参照してください。

(3) 入出力処理

ベースユニットに装着した入出力ユニットとデータメモリの間でデータの交換を行います。入出力リレー番号の若い入出力ユニットから順に選択して処理します。

a. 入出力ユニット処理

選択したユニットが入力ユニットの場合、入力ユニットに接続した入力機器(リミットスイッチ等)のON/OFF状態を、この入力ユニットに相当するデータメモリのアドレス位置に書き込みます。

選択したユニットが出力ユニットの場合、この出力ユニットに相当するアドレス位置のデータメモリの内容を出力ユニットのラッチに書き込み、出力ユニットはON又はOFFと変化します。

b. 入出力ユニットの自己診断機能

入出力ユニットに関する自己診断機能は、つぎの各項目があります。

- ① I/Oデータバスチェック
- ② 出力データチェック
- ③ 実装ユニットチェック
- ④ I/Oバスチェック
- ⑤ テーブル照合エラーチェック
- ⑥ ユニットなしエラーチェック
- ⑦ I/O点数チェック

6・8ページ自己診断を参照してください。

- 電源投入後の1サイクル目は、「パワーON処理」でイニシャライズしたデータメモリの内容を、出力ユニットに書き込み、以後のサイクルは、1回前のサイクルの演算結果を出力ユニットに書き込みます。
- 入力ユニット装着領域で、入力機器を接続していない部分は、入出力処理でOFFとしてデータメモリに読み込みます。従って補助リレーには使えません。
- 出力ユニット装着領域で、出力機器を接続していない部分は、補助リレーとして使えます。(ただし、入出力処理で出力ユニットにはデータメモリの内容を書き込み、出力ユニットのLEDは点灯します。)
- リレー領域のユニット未装着領域および出力ユニット装着領域で出力機器の未接続部分を補助リレーとして使用した場合、将来入出力機器の追加でこの領域を使用すると、プログラムの大巾変更(他の領域に補助リレーを移す)が必要です。

- 8点の入力/出力ユニットを使用しても、リレー割り付けは16点となります。入力/出力機器を接続できるのは、前半の8点のみです。入力ユニットの後半8点は使用できません。出力ユニットの後半8点は補助リレーとして使用できます。

(4) ウォッチドグタイム

CPUが内部処理フローに従い、正常に動作しているかどうかをハードウェアのウォッチドグタイムでチェックしています。

スキャンサイクルを正常に処理している場合、CPUからウォッチドグタイムにリセットが掛るため、タイムアップすることはありません。

何らかの原因でスキャンが異常となると、CPUからのリセットが掛からずウォッチドグタイムがタイムアップします。これによってCPUはリセットされ、プログラムモードになります。このとき電源ユニットの「停止出力」はOFF、「RUN(運転)」LEDは点滅、コントロールユニットの「FAULT(異常)」LEDは消灯になります。

(5) プログラム、オプションユニットからのリクエストに対する処理

プログラマからのモニタ/設定値変更、オプションユニットとのデータ交信を行います。

コントロールユニットに対してプログラマやオプションユニットからメモリリクエスト(コントロールユニット内のデータメモリ、ユーザープログラムメモリに対して書き込み、読み出しを要求する信号)があれば、コントロールユニット内のCPUはDMA動作状態となります。

(6) 0.1秒クロック(07360)、1秒クロック(07364)の設定

0.1秒クロックのON/OFF状態を特殊リレー07360に、1秒クロックのON/OFF状態を特殊リレー07364に書き込みます。

(7) ユーザープログラム処理

ユーザープログラムメモリの先頭からプログラムを順次読み出し、プログラム内容に従い演算します。

STR、STR NOT、AND、AND NOT、OR、OR NOT、AND STR、OR STRの各命令は演算結果をアキュムレーター、スタックレジスタに格納します。

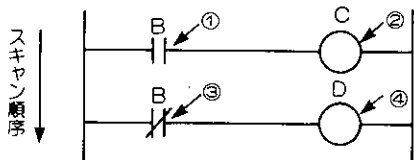
OUT、TMR、CNT、MD及び殆んど応用命令(F-××)は演算結果をデータメモリに書き込みます。

- 各命令の詳細は第8章、第9章を参照してください。
- ユーザープログラムの演算に先だち、「入出力処理」において入力ユニットのON/OFF状態を一括してデータメモリに読み込み、各命令の演算はデータメモリの内容を参照する方式を採用しているため「入力のレーシング」といった異常現象は発生しません。

参考 入力のレーシング現象

命令の演算時にその都度入力ユニットのON/OFF状態を読み込む場合、次のような現象が起ります。

(例)



(入力BがONのときコイルCをON、入力BがOFFのときコイルDをONとするプログラム)

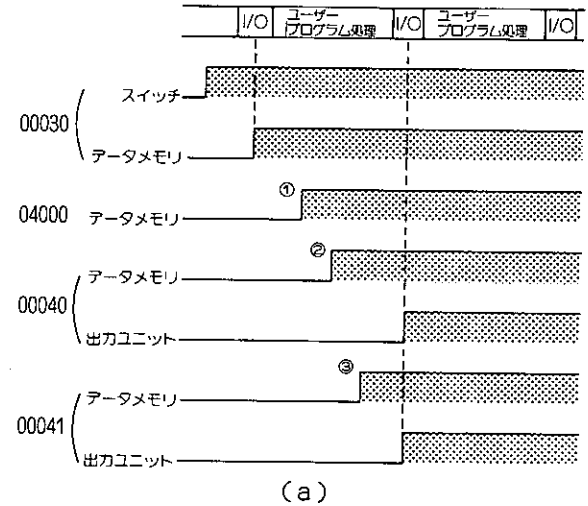
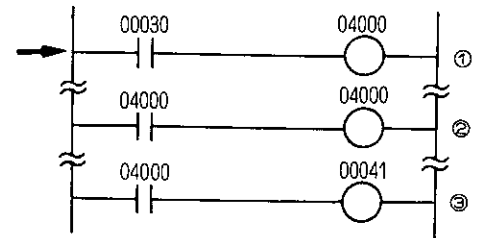
上図のプログラムでは $C = \bar{D}$ となるはずですが、①で入力Bの状態を入力ユニットからアキュムレーターに入れたときBはONであったとします(CはON)。ところが③の演算までに入力Bの状態がOFFに変化すると、③の演算ではBはOFFとして扱われ、コイルDがONし、C、DともにONという論理的に矛盾した結果になります。

このように入力の変化するタイミングにより誤動作したり、しなったりするため、原因の判らない故障につながる可能性があります。

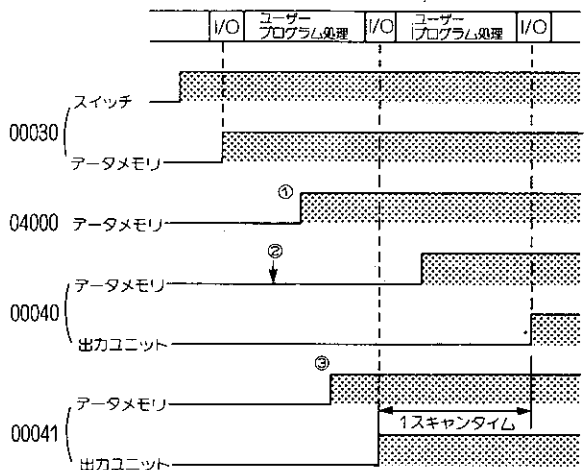
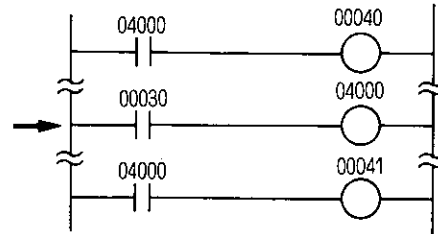
「入出力処理」を一括して行うPCではこのような現象は起りません。

- OUT命令のようにデータメモリに演算結果を書き込む命令は演算の都度、データメモリに演算結果を書き込みます。ただし出力ユニットの状態は次のスキャンサイクルの入出力処理まで変化しません。
- OUT命令のようにデータメモリに演算結果を書き込む命令の後に、当該データメモリを接点として使用する命令があると、OUT命令で書き換えられた内容に基づき演算します。

(例1)



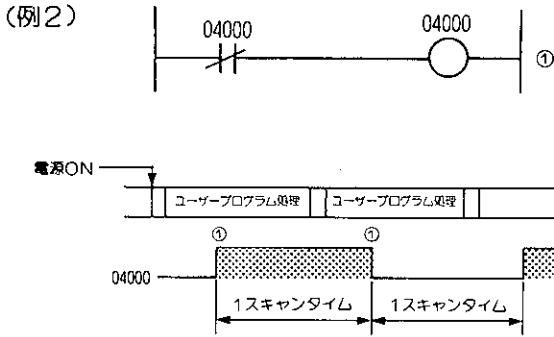
(a)



(b)

(a)と(b)のようにプログラムの書き込み順を入れ替えると、演算結果が異なったものとなります。並列に処理するリレー盤では、(a)も(b)も差はありませんが、直列処理形のプログラマブルコントローラ(現在市販されているプログラマブルコントローラは殆んど直列処理形です)では上記のような現象が起こります。したがってコイルの補助接点を使うとき(例1では04000)、次の事項に注意してプログラムを作成してください。

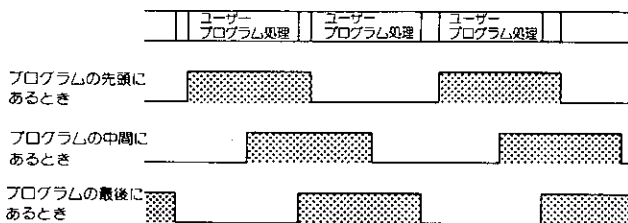
「コイルの前に書かれた補助接点の状態変化は、コイルの状態が変わった次のスキャンに生ずる。」



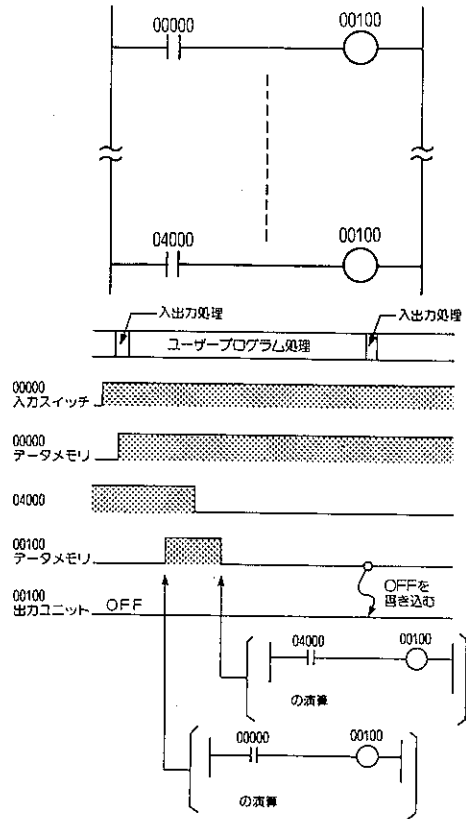
例2は「コイルの前に書かれた補助接点の状態変化は、コイルの状態が変化した次のスキャンに生ずる」ことを逆に応用したもので、1スキャンサイクルごとにON/OFFを繰り返します。(発振回路)

このパルスは、点滅回路の基本クロックや、1スキャンおきの演算起動信号として使用できます。

(注) タイムチャートに示すように、発振回路のプログラムがユーザープログラムメモリ上でどの位置に書かれているかによって、ON/OFFとなるタイミングが変わります。このパルスを演算の起動信号として使用するときは注意してください。



●プログラム上、同一リレー番号をコイルとして複数回使用すると、プログラムENDでは、最後にコイルとして使用したプログラムの演算結果がデータメモリに残り、これが次のスキャンサイクルの入出力処理で出力ユニットに書き込まれるため、目的と違った動作となることがあります。



コイルの複数使用があると、プログラマ(JW-14PG等)で、プログラムチェックを行うと、「DOUBLE OUT」と表示します。

(8) スキャンタイム

I/O処理から次のサイクルのI/O処理までの1スキャンに要する時間をスキャンタイムと呼び、次のようにして概略計算できます。(5・4ページ参照)

1スキャンタイム(T) = 0.2ms(固定値:自己診断等) + 入出力処理時間(t₁) + オプション処理時間(t₂) + ユーザープログラム処理時間(t₃)

① 入出力処理時間 (t₁)

CPUが入力ユニットの入力情報を読み込み、出力ユニットへ出力情報を書き込むのに必要とする時間です。本P Cの場合、16点当たり平均10μsです。

② オプション処理時間 (t₂)

CPUがオプションユニットとの情報交換に要する時間です。

オプションユニットの種類、状態により大きく異なります。

【例】

オプションユニット	オプション処理時間	使用条件
JW-21CM	約0.5ms (コントロールユニットとJW-21CM間)	※1
JW-22CM	約1.7ms (コントロールユニットとJW-22CM間)	※2

- ※1
 - ・データリンクDL9機能を使用
 - ・コントロールユニットにJW-31CUH1/32CUH1/33CUH1/33CUH2/33CUH3、基本ベースユニットにJW-34KB/36KB/38KBを使用
 - ・リンクバイト数が512バイトのとき
- ※2
 - ・コントロールユニットにJW-31CUH1/32CUH1/33CUH1/33CUH2/33CUH3、基本ベースユニットにJW-34KB/36KB/38KBを使用
 - ・転送バイト数が2304バイトのとき

③ユーザープログラム処理時間(t_3)

プログラムアドレス00000からEND命令までの全命令の処理時間の合計です。各命令の処理時間は第7章「命令語一覧表」を参照してください。

- 応用命令の処理時間は実行時と非実行時で異なります。

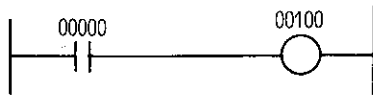
④END命令

プログラムメモリをクリアすると、プログラムメモリにすべてNOP命令を書き込み、最終アドレスにはF-40 (END命令) を書き込みます。この状態でプログラムメモリの途中まで命令を書き込んだ場合、NOP命令の処理時間をスキャンタイムに加算します。最後のプログラムを書き込んだアドレスの次にF-40を書き込むと、そのアドレスでユーザープログラムの処理を終り、スキャンタイムを短くできます。

コントロールユニット	NOP命令の処理時間
JW-31CUH JW-32CUH JW-33CUH	0.05 μ s
JW-31CUH1 JW-32CUH1 JW-33CUH1 JW-33CUH2 JW-33CUH3	0.038 μ s

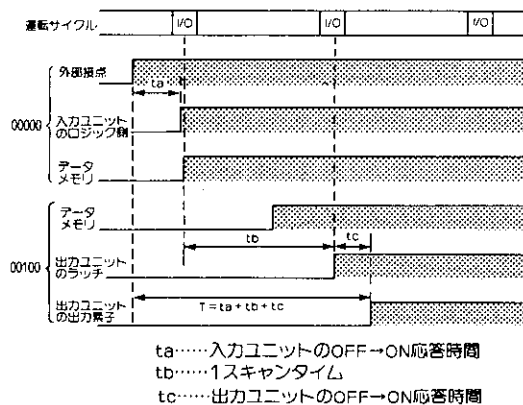
第9章「応用命令の説明」のF-40を参照してください。

入出力ユニットの応答時間を含めたPC全体の応答時間は次のようになります。

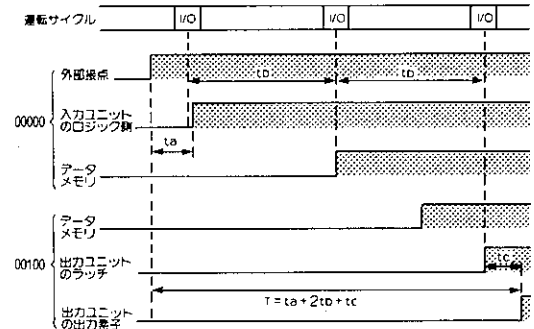


上記のプログラムで、外部接点00000が変化してから、出力ユニットの出力素子（トランジスタ、トライアック、リレー）が変化するまでの時間を示します。

(a)最も短時間の場合

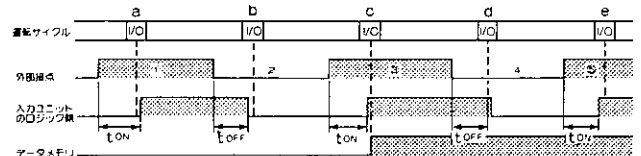


(b)最も長時間の場合



ON→OFFの場合も入力ユニット、出力ユニットの応答時間による遅れが影響します。

外部接点のON/OFF状態を確実にデータメモリに取り込むには、入力ユニットのロジック側のONまたはOFFの時間として、1スキャンタイム以上必要です。



①の外部接点のONは、入力ユニットのロジック側がONとなったとき、既に当該入力の入出力処理が終了し、bの入出力処理の直前に入力ユニットのロジック側はOFFとなるため、データメモリはOFFのままとなります。

③での外部接点のONは、cの入出力処理の直前に入力ユニットのロジック側もONになっているため、データメモリにはONを書き込みます。

④の外部接点のOFFは、dの入出力処理の時、入力ユニットのロジック側はまだONのため、データメモリはONのままとなります。eの入出力処理では、入力ユニットが再びONのため、データメモリはONを維持します。このように入力ユニットのロジック側のON/OFFの時間が1スキャンタイムより短いと、データメモリに取り込んだり、取り込まなかったりします。

入力ユニットのロジック側のON/OFF時間、入出力ユニットの応答時間に関してはJW30Hの「ユーザーズマニュアル・ハード編」の7・12ページ「入出力ユニットご使用時の留意事項」を参照してください。

6 - 2 自己診断

JW30Hはコントロールユニット等の各ユニットを自己診断し、異常が発生すると下表の状態になります。これにより異常の原因を究明し、その対策を行ってください。

項目	内容	P C の 運転状態	停止 出力	コントロール ユニット	電源ユニットの表示灯		特殊リレー	異常コード(BCD)							
				FAULT (異常)	POWER (電源)	R U N (運転中)		特殊レジスタ	システムメモリ						
								コ0734	# 160 ~ 167						
自 己 診 断	メモリ異常	命令コードチェック	停 止	開	点 灯	消 灯	07370	20	24						
		システムメモリ 設定チェック							23						
		プログラム ROMチェック							25						
		プログラム サムチェック							26						
		I/O登録テーブル チェック							28						
	CPU異常	ウォッチドグタイマ			消 灯	点 滅	—	00	31						
		RAMチェック (R/W)			点 灯	点 灯	消 灯	07371	32						
		ハードウェア チェック							35						
	入 出 力 異 常	リフレッ シュ時			I/Oデータバス	消 灯	点 灯	消 灯	07373	40	44				
					出力データチェック						42				
					実装ユニットチェック						40				
					I/Oベース異常						48				
					ユニットバイト数 チェック						45				
		テーブル 照合時			テーブル照合エラー					消 灯	点 灯	消 灯	07373	60	60
					スイッチ照合エラー										61
					テーブル登録エラー										70
		ユニットなしエラー			71										
		テーブル 登録時			I/O点数オーバー					消 灯	点 灯	消 灯	07373	70	
	スイッチ設定エラー				73										
	特殊I/O異常	ハードエラー			運 転	閉	消 灯	点 灯	点 灯	07375	46				
		パラメータエラー									47				
		JW-262Sのヒューズ断									運 転	閉	消 灯	点 灯	点 灯
	オプション異常	ハードエラー			停 止	開	点 灯	点 灯	消 灯	07374	50	53			
		オプションコマンドエラー			運 転	閉	消 灯	点 灯	点 灯			54			
システム保護エラー		運 転	閉	消 灯	点 灯	点 灯	55								
電 源 異 常	停電 / 電圧低下	停 止	開	消 灯	消 灯	消 灯	07377	10	13						
増設電源異常	停電 / 電圧低下						07376	40	43						
電 池 異 常	電池電圧低下 / 電池未挿入	運 転	閉	点 灯	点 灯	点 灯	07372	20	22						
停 止 出 力	リレー出力、AC100 / 200V DC30V、1A、PC運転中はON (閉)														

システムメモリ # 206、# 207のヒューズ断時またはオプション異常時の設定により、各項目の上欄または下欄の状態になります。

(設定)	→	(状態)
運転継続	→	上 欄
停止	→	下 欄

(注) 運転中に自己診断により異常を検出した場合、異常コードは格納されますが、PCの運転状態 / 停止出力 / 表示灯(FAULT等) / 特殊リレーの状態は次のとおりです。

- ・異常状態がウォッチドグタイマ(300ms)以内に復旧すれば、上表の状態にはなりません。
- ・異常状態がウォッチドグタイマ(300ms)を越えて継続していると、上表の状態になります。

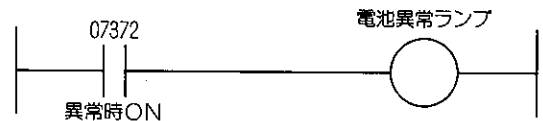
(1) 自己診断内容

- (1) 命令コードチェック
すべての命令の演算実行時プログラムメモリのコード異常をチェックします。命令コード異常のプログラムアドレスはシステムメモリ#052、#053に格納します。
- (2) システムメモリ設定チェック
システムメモリ#200~#256のサムチェックを行います。
- (3) プログラムROMチェック
ROM運転で、プログラムのROM→RAM転送時、ROMのサムチェックを行います。フラッシュROM不良のとき異常となります。
- (4) プログラムサムチェック
ユーザープログラムの書込や修正を行うと自動的にサムチェックコードを生成、電源投入時にサムチェックを実行しユーザープログラムの内容が変わっていないかどうかをチェックしています。サムチェックの場合、変化したプログラムの場所は特定できませんので万一プログラムサムチェック異常となった場合は、システムメモリ、プログラム、データ(必要に応じて)の再転送(書込)を行い復旧させる必要があります。
- (5) I/O登録テーブルチェック
I/O登録の際に登録データのサムチェックコードを生成、電源投入時にサムチェックを実行し登録データの内容が変わっていないかどうかをチェックしています。万一I/O登録テーブルチェック異常となった場合はI/O登録を行う必要があります。
- (6) ウォッチドグタイマ
CPUのウォッチドグタイマがタイムアップするところの異常になります。(プログラムモードと同じ状態)
- (7) RAMチェック
毎スキャンサイクルごとにデータメモリ用RAMが書き込み、読み出し可能であるかチェックします。
- (8) ハードウェア動作
毎スキャンサイクルごとにアキュムレータ、スタックが正しく動作することをチェックします。
- (9) I/Oデータバス
入出力処理の前に入出力データバスがフローティング状態であることを確認します。システムメモリ#046に異常ユニット位置を格納します。
- (10) 出力データチェック
入出力処理の中で、出力ユニットに出力したデータは再度読み出し照合しています。照合NGの場合エラーになります。
●特殊ユニットは本チェックを行いません。
- (11) 実装ユニットチェック
入出力処理の中で、CPUは各ユニットとデータ交換を行うときに、I/Oテーブルに登録したユニットの実装状態と照合します。照合NGの場合エラーになります。
- (12) I/Oベース異常
入出力処理の前にベースユニット内蔵の総てのI/Oポートのゲートが閉じているかチェックします。開いているポートがあった場合エラーになります。
- (13) ユニットバイト数チェック
16/32/64点の入出力ユニットを処理時に、I/Oテーブルに登録されたユニットの種類(バイト数)と内部のバイトカウンタ値が異なる場合、このエラーになります。
- (14) テーブル照合エラー
電源投入時またはモード変更(停止モード→運転モード)時に各ユニットの実装状態と既に登録しているI/Oテーブルの内容を照合します。照合NGの時、このエラーになります。なお、増設ベースのラック番号スイッチの設定、I/O増設ケーブルの接続状態が変化した時もこのエラーになります。
- (15) スイッチ照合エラー(ユニットNo.スイッチ照合エラー)
電源投入時またはモード変更(停止モード→運転モード)時に、特殊/オプションI/Oリンク親局ユニットの場合、ユニットNo.スイッチの設定も照合します。照合NGの場合、このエラーになります。
- (16) テーブル登録エラー
I/Oテーブル登録時に、増設ベースユニットのラック番号スイッチの誤設定、I/O増設ケーブルの誤接続等の初期エラーが発生している場合、このエラーになります。
- (17) ユニットなしエラー
I/Oテーブル登録時に、テーブルデータの内容が「ユニット実装空間が全く無い状態」として登録された時、このエラーになります。ユニットが全く実装されていない場合の状態ではありません。
- (18) I/O点数オーバーエラー
I/Oテーブル登録時に、ユニットの装着数が多すぎて、I/O点数がコントロールユニットの制御入出力点数を越えた場合、このエラーになります。

- (19) スイッチ設定エラー
(ユニットNo.スイッチ設定エラー)
I/Oテーブル登録時に、特殊/オプションユニット等のユニットNo.スイッチの設定が重複している場合、このエラーになります。
- (20) 特殊I/Oハードエラー
特殊ユニット自身の異常により、特殊ユニット内蔵のCPUのウォッチドグタイマが働いた時にこのエラーになります。
- (21) 特殊I/Oパラメータエラー
コントロールユニットが特殊ユニットにパラメータを転送したとき、パラメータ照合がNGの場合、このエラーになります。
- (22) 特殊I/O(JW-262S)のヒューズ断
JW-262Sのヒューズが切れたときこの異常になります。
JW-262Sに外部電源が供給されていない場合もヒューズ断異常となります。
システムメモリ #206を「運転停止」の設定にした場合は特に注意してください。
- (23) オプションハードエラー
オプションI/Oリンク親局ユニット自身の異常により、ユニット内蔵のCPUのウォッチドグタイマが働いた時、このエラーになります。
- (24) オプションコマンドエラー
オプションユニットとコントロールユニット間のデータ交換コマンドをチェックします。ノイズ等の外的要因によりコマンド内容が適正でない場合、このエラーになります。
- (25) システム保護エラー
ノイズ等の外的要因によりオプションユニットとコントロールユニット間のデータ交換コマンドからコントロールユニットのシステム領域に書込要求があった場合、このエラーになります。
- (26) 電源異常
JW30Hは10ms以下の瞬時停電の場合、これにตอบสนองせず運転を続行します。
これ以上の停電の場合、CPUが停止し停止出力が開放となります。
停電が復旧すると自動的に運転を再開します。
●電源電圧が徐々に低下(スローダウン)してきた場合、定格電圧の85%以下になるとCPUは停止し、停止出力が開放となります。
この場合も電源電圧が復旧すれば自動的に運転を再開します。

- (27) 増設電源異常
増設電源の電圧(DC5V)が4.5V以下になったとき、この異常となります。
この異常が発生時、異常となった増設電源を取り付けているベースユニットの出力ユニットはリセットします。
●増設電源異常が発生したとき、他の入出力異常が同時に発生し、異常履歴として増設電源異常より優先して格納することがあります。

- (28) 電池異常
メモリバックアップ用電池の電圧が正常であるかチェックします。
特殊リレー07372を使って、電池異常時ランプを点灯させたり、ブザーを鳴らせます。
PCに電源が投入されている限り、電池異常状態でもPCの運転には影響ありませんが、万一の停電にそなえ、できるだけ速やかに電池を交換してください。



(2) 停止出力

- 自己診断により異常と判断したとき、「開」となる出力で正常運転中は「閉」です。
- システムの非常停止回路にJW30Hの停止出力を接続すると、PC異常時、システムを非常停止できます。

(注) 異常状態がウォッチドグタイマ(300ms)以内に復旧すれば、停止出力は「開」になりません。
ウォッチドグタイマ(300ms)を越えて異常状態が継続していると、停止出力は「開」になります。

(3) 特殊リレー

データメモリの特殊リレー領域に自己診断結果を書き込みます。

自己診断結果異常を検知しPCが停止した場合、周辺装置により特殊リレー(07370~07377)を検索し異常内容を確認できます。

- 自己診断は毎スキャンサイクルごとに行い、異常が回復すればPCは運転を再開し停止出力も閉となります。また自己診断用特殊リレーもリセットします。
- 特殊リレーの内07372(電池異常)、07374(オプション異常)だけが、PC演算で出力ユニットから取り出せます。他のリレーはJW-21CMのコンピュータリンクや周辺装置で読み出してください。
なお、特殊リレー内容は、データリンクで読み出せません。
- 07377(電源異常)のリレーは、電源投入時の1スキャンだけONします。

(4) 異常コード

1. 特殊レジスタ

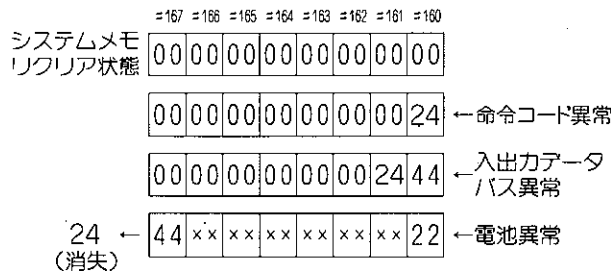
自己診断結果、異常と判断した場合、データメモリの特
殊レジスタ（バイトアドレス0734）に異常コードを書き込み
ます。

- 異常発生中に他の異常が発生した場合、優先順位の
高い方の異常コードに書き換わります。
- 異常が回復すると異常コードはクリアします。

2. システムメモリ

自己診断結果、異常と判断した場合、システムメモ
リ（#160～#167）にも異常コードを書き込み
ます。

#160～#167はシフトレジスタとして働き、
8回の異常発生を記憶できます。異常が8回以上にな
ると、最初に書き込んだ異常コードから順に消失し
ます。



- 特殊レジスタには代表コードを書き込みますが、シ
ステムメモリには異常内容をさらに分類した個別コー
ドを書き込みます。
- システムメモリの異常コードは異常回復後もクリア
しません。クリアするときは、プログラマ等の周辺装置
でシステムメモリ（#160～#167）に「00」
を書き込んでください。
- 同じ異常が連続して発生した場合、異常コードは書き
込みません。

3. レジスタ

- レジスタE7600～E7777に異常発生時刻を含んだ異
常コード内容を格納します。（システムメモリ#210に
002ocrを設定時）

(5) 異常時の出力ユニットの ON/OFF状態

自己診断結果、PCが停止する場合の出力ユニットの
ON/OFF状態は、システムメモリ#232、#233
（00000～15777）、#252、#253（20000～75777）
の設定内容により決まります。

- 出力保持アドレス以前の出力ユニット——OFF
- 出力保持アドレス以後の出力ユニット——停止直前の
ON/OFF状態を保持

ただし、異常内容によっては出力保持アドレス以前の出
力ユニットをOFFにできない場合があります。PC異常
時にOFFにする必要がある出力は、コントロールユニ
ットの停止出力を直列に接続してください。接続方法に
関してはJW30Hの「取扱説明書」及び「ユーザスマニ
ュアル」を参照してください。

第 7 章 命令語一覽

〔 1 〕 番号順

命令語	シンボル	語数	機 能	実行条件	フラグ				参照ページ
					ゼロ 07357	キャリー 07356	エラー 07355	カネリ 07354	
STR		1 2	a 接点で論理を開始。中間結果の記憶						8・2
STR NOT		1 2	b 接点で論理を開始。中間結果の記憶						2
AND		1 2	論理積						3
AND NOT		1 2	論理積否定						3
OR		1 2	論理和						3
OR NOT		1 2	論理和否定						4
AND STR		1	中間結果との論理積						4
OR STR		1	中間結果との論理和						5
OUT		1 2	演算結果の出力						2
TMR		2	タイマ(減算式) スタート入力(ONで計数) TMR番号(0000 ~ 1777) 設定値 (0.1 ~ 199.9秒) 内部クロック0.1秒又は0.01秒 (0.01 ~ 19.99秒)	スタート 入力 ON					6
DTMR (BCD)		3	タイマ(減算式) スタート入力(ONで計数) TMR番号(000 ~ 777) 設定値(0.1 ~ 799.9秒)	スタート 入力 ON					6
DTMR (BIN)		3	タイマ(減算式) スタート入力(ONで計数) TMR番号(000 ~ 777) 設定値(0.1 ~ 3276.7秒)	スタート 入力 ON					6
UTMR (BCD)		3	タイマ(加算式) スタート入力(ONで計数) TMR番号(000 ~ 777) 設定値(0.1 ~ 799.9秒)	スタート 入力 ON					6
UTMR (BIN)		3	タイマ(加算式) スタート入力(ONで計数) TMR番号(000 ~ 777) 設定値(0.1 ~ 3276.7秒)	スタート 入力 ON					6
CNT		2	カウンタ(減算式) 計数入力 CNT番号(0000 ~ 1777) リセット入力 設定値(1 ~ 1999)	計数入力 ↑					7
DCNT (BCD)		3	カウンタ(減算式) 計数入力 CNT番号(000 ~ 777) リセット入力 設定値(1 ~ 7999)	計数入力 ↑					7
DCNT (BIN)		3	カウンタ(減算式) 計数入力 CNT番号(000 ~ 777) リセット入力 設定値(1 ~ 32767)	計数入力 ↑					7
UCNT (BCD)		3	カウンタ(加算式) 計数入力 CNT番号(000 ~ 777) リセット入力 設定値(1 ~ 7999)	計数入力 ↑					7
UCNT (BIN)		3	カウンタ(加算式) 計数入力 CNT番号(000 ~ 777) リセット入力 設定値(1 ~ 32767)	計数入力 ↑					7
MD		2	メンテナンスディスプレイ 入力情報 MD番号(000 ~ 777) 出力指示端子 MDデータ(000 ~ 999) 拡張出力	出力表示 端子 ON					9

語数の上段はリレー番号000000 ~ 15777_{OCT}、TMR/CNT接点番号0000 ~ 0777_{OCT}を使用する場合があります。
下段はリレー番号20000 ~ 75777_{OCT}、TMR/CNT接点番号1000 ~ 1777_{OCT}を使用する場合があります。

命令語	シンボル	語数	機能	実行条件	フラグ				参照ページ								
					ゼロ 07357	キャリー 07356	エラー 07355	オーバー 07354									
F-00	<table border="1"><tr><td>F-00</td><td>S</td><td>D</td></tr><tr><td>XFER</td><td></td><td></td></tr></table>	F-00	S	D	XFER			3	データレジスタ間の1バイト転送	↑					9・19		
F-00	S	D															
XFER																	
F-00w	<table border="1"><tr><td>F-00w</td><td>S</td><td>D</td></tr><tr><td>XFER</td><td></td><td></td></tr></table>	F-00w	S	D	XFER			3	データレジスタ間の1ワード転送	↑					20		
F-00w	S	D															
XFER																	
F-00d	<table border="1"><tr><td>F-00d</td><td>S</td><td>D</td></tr><tr><td>XFER</td><td></td><td></td></tr></table>	F-00d	S	D	XFER			3	データレジスタ間の2ワード転送	↑					21		
F-00d	S	D															
XFER																	
F-01	<table border="1"><tr><td>F-01</td><td>n</td><td>D</td></tr><tr><td>BCD</td><td></td><td></td></tr></table>	F-01	n	D	BCD			3	BCD定数(2桁)の転送	↑					22		
F-01	n	D															
BCD																	
F-01w	<table border="1"><tr><td>F-01w</td><td>n</td><td>D</td></tr><tr><td>BCD</td><td></td><td></td></tr></table>	F-01w	n	D	BCD			3	BCD定数(4桁)の転送	↑					23		
F-01w	n	D															
BCD																	
F-02	<table border="1"><tr><td>F-02</td><td>D₁</td><td>D₂</td></tr><tr><td>XCHG</td><td></td><td></td></tr></table>	F-02	D ₁	D ₂	XCHG			3	レジスタ間(1バイト)のデータ交換	↑					24		
F-02	D ₁	D ₂															
XCHG																	
F-02w	<table border="1"><tr><td>F-02w</td><td>D₁</td><td>D₂</td></tr><tr><td>XCHG</td><td></td><td></td></tr></table>	F-02w	D ₁	D ₂	XCHG			3	レジスタ間(1ワード)のデータ交換	↑					25		
F-02w	D ₁	D ₂															
XCHG																	
F-02d	<table border="1"><tr><td>F-02d</td><td>D₁</td><td>D₂</td></tr><tr><td>XCHG</td><td></td><td></td></tr></table>	F-02d	D ₁	D ₂	XCHG			3	レジスタ間(2ワード)のデータ交換	↑					26		
F-02d	D ₁	D ₂															
XCHG																	
F-03	<table border="1"><tr><td>F-03</td><td>S</td><td>D</td></tr><tr><td>→BIN</td><td></td><td></td></tr></table>	F-03	S	D	→BIN			3	BCD(2桁) BIN(8ビット)変換	↑	0	0	↕	0	27		
F-03	S	D															
→BIN																	
F-03w	<table border="1"><tr><td>F-03w</td><td>S</td><td>D</td></tr><tr><td>→BIN</td><td></td><td></td></tr></table>	F-03w	S	D	→BIN			3	BCD(4桁) BIN(16ビット)変換	↑	0	0	↕	0	28		
F-03w	S	D															
→BIN																	
F-04	<table border="1"><tr><td>F-04</td><td>S</td><td>D</td></tr><tr><td>→BCD</td><td></td><td></td></tr></table>	F-04	S	D	→BCD			3	BIN(8ビット) BCD(2桁)変換	↑					29		
F-04	S	D															
→BCD																	
F-04w	<table border="1"><tr><td>F-04w</td><td>S</td><td>D</td></tr><tr><td>→BCD</td><td></td><td></td></tr></table>	F-04w	S	D	→BCD			3	BIN(16ビット) BCD(6桁)変換	↑					30		
F-04w	S	D															
→BCD																	
F-05	<table border="1"><tr><td>F-05</td><td>S</td><td>D</td></tr><tr><td>DMPX</td><td></td><td></td></tr></table>	F-05	S	D	DMPX			3	1バイトデータの分配	↑					31		
F-05	S	D															
DMPX																	
F-05w	<table border="1"><tr><td>F-05w</td><td>S</td><td>D</td></tr><tr><td>DMPX</td><td></td><td></td></tr></table>	F-05w	S	D	DMPX			3	1ワードデータの分配	↑					33		
F-05w	S	D															
DMPX																	
F-06	<table border="1"><tr><td>F-06</td><td>S</td><td>D</td></tr><tr><td>MPX</td><td></td><td></td></tr></table>	F-06	S	D	MPX			3	1バイトデータの抽出	↑					34		
F-06	S	D															
MPX																	
F-06w	<table border="1"><tr><td>F-06w</td><td>S</td><td>D</td></tr><tr><td>MPX</td><td></td><td></td></tr></table>	F-06w	S	D	MPX			3	1ワードデータの抽出	↑					35		
F-06w	S	D															
MPX																	
F-07	<table border="1"><tr><td>F-07</td><td>n</td><td>D</td></tr><tr><td>DCML</td><td></td><td></td></tr></table>	F-07	n	D	DCML			3	10進定数(1バイト)の転送	↑					36		
F-07	n	D															
DCML																	
F-07w	<table border="1"><tr><td>F-07w</td><td>n</td><td>D</td></tr><tr><td>DCML</td><td></td><td></td></tr></table>	F-07w	n	D	DCML			3	10進定数(1ワード)の転送	↑					37		
F-07w	n	D															
DCML																	
F-08	<table border="1"><tr><td>F-08</td><td>n</td><td>D</td></tr><tr><td>OCT</td><td></td><td></td></tr></table>	F-08	n	D	OCT			3	8進定数(1バイト)の転送	↑					38		
F-08	n	D															
OCT																	
F-08w	<table border="1"><tr><td>F-08w</td><td>n</td><td>D</td></tr><tr><td>OCT</td><td></td><td></td></tr></table>	F-08w	n	D	OCT			3	8進定数(1ワード)の転送	↑					39		
F-08w	n	D															
OCT																	
F-09	<table border="1"><tr><td>F-09</td><td>S</td><td>D</td></tr><tr><td>INV</td><td></td><td></td></tr></table>	F-09	S	D	INV			3	8ビットデータの反転	↑					40		
F-09	S	D															
INV																	
F-09w	<table border="1"><tr><td>F-09w</td><td>S</td><td>D</td></tr><tr><td>INV</td><td></td><td></td></tr></table>	F-09w	S	D	INV			3	16ビットデータの反転	↑					41		
F-09w	S	D															
INV																	
F-09d	<table border="1"><tr><td>F-09d</td><td>S</td><td>D</td></tr><tr><td>INV</td><td></td><td></td></tr></table>	F-09d	S	D	INV			3	32ビットデータの反転	↑					42		
F-09d	S	D															
INV																	
F-10	<table border="1"><tr><td>F-10</td><td>S₁</td><td>S₂</td><td>D</td></tr><tr><td>ADD</td><td></td><td></td><td></td></tr></table>	F-10	S ₁	S ₂	D	ADD				4	レジスタ間(BCD2桁)の加算	↑	↕	↕	↕	↕	43
F-10	S ₁	S ₂	D														
ADD																	
F-10w	<table border="1"><tr><td>F-10w</td><td>S₁</td><td>S₂</td><td>D</td></tr><tr><td>ADD</td><td></td><td></td><td></td></tr></table>	F-10w	S ₁	S ₂	D	ADD				4	レジスタ間(BCD4桁)の加算	↑	↕	↕	↕	↕	45
F-10w	S ₁	S ₂	D														
ADD																	
F-10d	<table border="1"><tr><td>F-10d</td><td>S₁</td><td>S₂</td><td>D</td></tr><tr><td>ADD</td><td></td><td></td><td></td></tr></table>	F-10d	S ₁	S ₂	D	ADD				4	レジスタ間(BCD8桁)の加算	↑	↕	↕	↕	↕	47
F-10d	S ₁	S ₂	D														
ADD																	
Fc10	<table border="1"><tr><td>Fc10</td><td>S₁</td><td>n</td><td>D</td></tr><tr><td>ADD</td><td></td><td></td><td></td></tr></table>	Fc10	S ₁	n	D	ADD				4	レジスタ(BCD2桁)と定数(2桁)の加算	↑	↕	↕	↕	↕	48
Fc10	S ₁	n	D														
ADD																	
Fc10w	<table border="1"><tr><td>Fc10w</td><td>S₁</td><td>n</td><td>D</td></tr><tr><td>ADD</td><td></td><td></td><td></td></tr></table>	Fc10w	S ₁	n	D	ADD				4	レジスタ(BCD4桁)と定数(4桁)の加算	↑	↕	↕	↕	↕	49
Fc10w	S ₁	n	D														
ADD																	
Fc10d	<table border="1"><tr><td>Fc10d</td><td>S₁</td><td>n</td><td>D</td></tr><tr><td>ADD</td><td></td><td></td><td></td></tr></table>	Fc10d	S ₁	n	D	ADD				4	レジスタ(BCD8桁)と定数(4桁)の加算	↑	↕	↕	↕	↕	50
Fc10d	S ₁	n	D														
ADD																	
F-11	<table border="1"><tr><td>F-11</td><td>S₁</td><td>S₂</td><td>D</td></tr><tr><td>SUB</td><td></td><td></td><td></td></tr></table>	F-11	S ₁	S ₂	D	SUB				4	レジスタ間(BCD2桁)の減算	↑	↕	↕	↕	↕	51
F-11	S ₁	S ₂	D														
SUB																	
F-11w	<table border="1"><tr><td>F-11w</td><td>S₁</td><td>S₂</td><td>D</td></tr><tr><td>SUB</td><td></td><td></td><td></td></tr></table>	F-11w	S ₁	S ₂	D	SUB				4	レジスタ間(BCD4桁)の減算	↑	↕	↕	↕	↕	53
F-11w	S ₁	S ₂	D														
SUB																	
F-11d	<table border="1"><tr><td>F-11d</td><td>S₁</td><td>S₂</td><td>D</td></tr><tr><td>SUB</td><td></td><td></td><td></td></tr></table>	F-11d	S ₁	S ₂	D	SUB				4	レジスタ間(BCD8桁)の減算	↑	↕	↕	↕	↕	55
F-11d	S ₁	S ₂	D														
SUB																	

命令語	シンボル	語数	機	能	実行条件	フラグ				参照ページ
						ゼロ 07357	キャリー 07356	エラー 07355	オーバー 07354	
Fc11	$\overline{\text{Fc11 SUB}} \quad S_1 \quad n \quad D$	4	レジスタ(BCD2桁)と定数(2桁)の減算	↑	↑↓	↑↓	↑↓	↑↓	9・57	
Fc11w	$\overline{\text{Fc11w SUB}} \quad S_1 \quad n \quad D$	4	レジスタ(BCD4桁)と定数(4桁)の減算	↑	↑↓	↑↓	↑↓	↑↓	58	
Fc11d	$\overline{\text{Fc11d SUB}} \quad S_1 \quad n \quad D$	4	レジスタ(BCD8桁)と定数(4桁)の減算	↑	↑↓	↑↓	↑↓	↑↓	59	
F-12	$\overline{\text{F-12 CMP}} \quad S_1 \quad S_2$	3	レジスタ間(1バイト)の比較	ON	↑↓	↑↓	0	↑↓	60	
F-12w	$\overline{\text{F-12w CMP}} \quad S_1 \quad S_2$	3	レジスタ間(1ワード)の比較	ON	↑↓	↑↓	0	↑↓	61	
F-12d	$\overline{\text{F-12d CMP}} \quad S_1 \quad S_2$	3	レジスタ間(2ワード)の比較	ON	↑↓	↑↓	0	↑↓	62	
Fc12	$\overline{\text{Fc12 CMP}} \quad S_1 \quad n$	3	レジスタと8進定数(1バイト)の比較	ON	↑↓	↑↓	0	↑↓	63	
Fc12w	$\overline{\text{Fc12w CMP}} \quad S_1 \quad n$	3	レジスタと8進定数(1ワード)の比較	ON	↑↓	↑↓	0	↑↓	64	
Fx12	$\overline{\text{Fx12 CMP}} \quad S_1 \quad n$	3	レジスタと16進定数(1バイト)の比較	ON	↑↓	↑↓	0	↑↓	65	
Fx12w	$\overline{\text{Fx12w CMP}} \quad S_1 \quad n$	3	レジスタと16進定数(1ワード)の比較	ON	↑↓	↑↓	0	↑↓	66	
F-13	$\overline{\text{F-13 AND}} \quad S \quad D$	3	レジスタ間(1バイト)の論理積	↑					67	
F-13w	$\overline{\text{F-13w AND}} \quad S \quad D$	3	レジスタ間(1ワード)の論理積	↑					68	
F-13d	$\overline{\text{F-13d AND}} \quad S \quad D$	3	レジスタ間(2ワード)の論理積	↑					69	
Fc13	$\overline{\text{Fc13 AND}} \quad n \quad D$	3	レジスタと8進定数(1バイト)の論理積	↑					70	
Fc13w	$\overline{\text{Fc13w AND}} \quad n \quad D$	3	レジスタと8進定数(1ワード)の論理積	↑					71	
Fx13	$\overline{\text{Fx13 AND}} \quad n \quad D$	3	レジスタと16進定数(1バイト)の論理積	↑					72	
Fx13w	$\overline{\text{Fx13w AND}} \quad n \quad D$	3	レジスタと16進定数(1ワード)の論理積	↑					73	
F-14	$\overline{\text{F-14 OR}} \quad S \quad D$	3	レジスタ間(1バイト)の論理和	↑					74	
F-14w	$\overline{\text{F-14w OR}} \quad S \quad D$	3	レジスタ間(1ワード)の論理和	↑					75	
F-14d	$\overline{\text{F-14d OR}} \quad S \quad D$	3	レジスタ間(2ワード)の論理和	↑					76	
Fc14	$\overline{\text{Fc14 OR}} \quad n \quad D$	3	レジスタと8進定数(1バイト)の論理和	↑					77	
Fc14w	$\overline{\text{Fc14w OR}} \quad n \quad D$	3	レジスタと8進定数(1ワード)の論理和	↑					78	
Fx14	$\overline{\text{Fx14 OR}} \quad n \quad D$	3	レジスタと16進定数(1バイト)の論理和	↑					79	
Fx14w	$\overline{\text{Fx14w OR}} \quad n \quad D$	3	レジスタと16進定数(1ワード)の論理和	↑					80	
F-15	$\overline{\text{F-15 MUL}} \quad S_1 \quad S_2 \quad D$	4	レジスタ間(BCD4桁)の乗算	↑	0	0	↑↓	0	81	
F-15d	$\overline{\text{F-15d MUL}} \quad S_1 \quad S_2 \quad D$	4	レジスタ間(BCD8桁)の乗算	↑	0	0	↑↓	0	82	
Fc15	$\overline{\text{Fc15 MUL}} \quad S_1 \quad n \quad D$	4	レジスタ間(BCD4桁)とBCD定数(3桁)の乗算	↑	0	0	↑↓	0	83	
Fc15d	$\overline{\text{Fc15d MUL}} \quad S_1 \quad n \quad D$	4	レジスタ間(BCD8桁)とBCD定数(4桁)の乗算	↑	0	0	↑↓	0	84	
F-16	$\overline{\text{F-16 DIV}} \quad S_1 \quad S_2 \quad D$	4	レジスタ(BCD4桁)とレジスタ(BCD2桁)の除算	↑	0	0	↑↓	0	85	
F-16d	$\overline{\text{F-16d DIV}} \quad S_1 \quad S_2 \quad D$	4	レジスタ(BCD8桁)とレジスタ(BCD8桁)の除算	↑	0	0	↑↓	0	87	
Fc16	$\overline{\text{Fc16 DIV}} \quad S_1 \quad n \quad D$	4	レジスタ(BCD4桁)とBCD定数(2桁)の除算	↑	0	0	↑↓	0	88	
Fc16d	$\overline{\text{Fc16d DIV}} \quad S_1 \quad n \quad D$	4	レジスタ(BCD8桁)とBCD定数(4桁)の除算	↑	0	0	↑↓	0	89	

命令語	シンボル	語数	機能	実行条件	フラグ				参照ページ
					ゼロ 07357	キャリー 07356	エラー 07355	例外 07354	
F-17	$\overline{\text{F-17}} \begin{array}{ c c c } \hline \text{XNR} & \text{S} & \text{D} \\ \hline \end{array}$	3	レジスタ間(1バイト)の一致	↑					9・90
F-17w	$\overline{\text{F-17w}} \begin{array}{ c c c } \hline \text{XNR} & \text{S} & \text{D} \\ \hline \end{array}$	3	レジスタ間(1ワード)の一致	↑					91
F-17d	$\overline{\text{F-17d}} \begin{array}{ c c c } \hline \text{XNR} & \text{S} & \text{D} \\ \hline \end{array}$	3	レジスタ間(2ワード)の一致	↑					92
Fc17	$\overline{\text{Fc17}} \begin{array}{ c c c } \hline \text{XNR} & \text{n} & \text{D} \\ \hline \end{array}$	3	レジスタと8進定数(1バイト)の一致	↑					93
Fc17w	$\overline{\text{Fc17w}} \begin{array}{ c c c } \hline \text{XNR} & \text{n} & \text{D} \\ \hline \end{array}$	3	レジスタと8進定数(1ワード)の一致	↑					94
Fx17	$\overline{\text{Fx17}} \begin{array}{ c c c } \hline \text{XNR} & \text{n} & \text{D} \\ \hline \end{array}$	3	レジスタと16進定数(1バイト)の一致	↑					95
Fx17w	$\overline{\text{Fx17w}} \begin{array}{ c c c } \hline \text{XNR} & \text{n} & \text{D} \\ \hline \end{array}$	3	レジスタと16進定数(1ワード)の一致	↑					96
F-18	$\overline{\text{F-18}} \begin{array}{ c c c } \hline \text{XOR} & \text{S} & \text{D} \\ \hline \end{array}$	3	レジスタ間(1バイト)の排他的論理和	↑					97
F-18w	$\overline{\text{F-18w}} \begin{array}{ c c c } \hline \text{XOR} & \text{S} & \text{D} \\ \hline \end{array}$	3	レジスタ間(1ワード)の排他的論理和	↑					98
F-18d	$\overline{\text{F-18d}} \begin{array}{ c c c } \hline \text{XOR} & \text{S} & \text{D} \\ \hline \end{array}$	3	レジスタ間(2ワード)の排他的論理和	↑					99
Fc18	$\overline{\text{Fc18}} \begin{array}{ c c c } \hline \text{XOR} & \text{n} & \text{D} \\ \hline \end{array}$	3	レジスタと8進定数(1バイト)の排他的論理和	↑					100
Fc18w	$\overline{\text{Fc18w}} \begin{array}{ c c c } \hline \text{XOR} & \text{n} & \text{D} \\ \hline \end{array}$	3	レジスタと8進定数(1ワード)の排他的論理和	↑					101
Fx18	$\overline{\text{Fx18}} \begin{array}{ c c c } \hline \text{XOR} & \text{n} & \text{D} \\ \hline \end{array}$	3	レジスタと16進定数(1バイト)の排他的論理和	↑					102
Fx18w	$\overline{\text{Fx18w}} \begin{array}{ c c c } \hline \text{XOR} & \text{n} & \text{D} \\ \hline \end{array}$	3	レジスタと16進定数(1ワード)の排他的論理和	↑					103
F-20	$\overline{\text{MD}} \begin{array}{ c } \hline \text{(F-20)} \\ \hline \end{array}$	2	メンテナンスディスプレイ , , 入力情報 MD番号 (000 ~ 777) 出力指示端子 MDデータ (000 ~ 999) 拡張出力	出力表示 端子 ON					104
F-21	$\overline{\text{F-21}} \begin{array}{ c c c } \hline \text{SQRT} & \text{S} & \text{D} \\ \hline \end{array}$	3	レジスタ(BCD8桁)の平方根	↑	0	0	↕	0	105
F-22	$\overline{\text{F-22}} \begin{array}{ c c c } \hline \text{SIN} & \text{S} & \text{D} \\ \hline \end{array}$	3	三角関数(SIN)の演算	↑	0	↕	↕	↕	106
F-23	$\overline{\text{F-23}} \begin{array}{ c c c } \hline \text{COS} & \text{S} & \text{D} \\ \hline \end{array}$	3	三角関数(COS)の演算	↑	0	↕	↕	↕	107
F-24	$\overline{\text{F-24}} \begin{array}{ c c c } \hline \text{TAN} & \text{S} & \text{D} \\ \hline \end{array}$	3	三角関数(TAN)の演算	↑	0	↕	↕	↕	108
F-25	$\overline{\text{F-25}} \begin{array}{ c c c } \hline \text{ASIN} & \text{S} & \text{D} \\ \hline \end{array}$	3	三角関数(SIN ⁻¹)の演算	↑	0	↕	↕	↕	109
F-26	$\overline{\text{F-26}} \begin{array}{ c c c } \hline \text{ACOS} & \text{S} & \text{D} \\ \hline \end{array}$	3	三角関数(COS ⁻¹)の演算	↑	0	↕	↕	↕	110
F-27	$\overline{\text{F-27}} \begin{array}{ c c c } \hline \text{ATAN} & \text{S} & \text{D} \\ \hline \end{array}$	3	三角関数(TAN ⁻¹)の演算	↑	0	↕	↕	↕	111
F-28	$\overline{\text{F-28}} \begin{array}{ c c c } \hline \text{XY} \rightarrow & \text{S} & \text{D} \\ \hline \end{array}$	3	直交座標系(X,Y)データの極座標系(,)への変換	↑	0	0	↕	0	112
F-29	$\overline{\text{F-29}} \begin{array}{ c c c } \hline \rightarrow \text{XY} & \text{S} & \text{D} \\ \hline \end{array}$	3	極座標系(,)データの直交座標系(X,Y)への変換	↑	0	0	↕	0	113
F-30	$\overline{\text{F-30}} \begin{array}{ c } \hline \text{MCS} \\ \hline \end{array}$	1	マスターコントロールのセット	ON					114
F-31	$\overline{\text{F-31}} \begin{array}{ c } \hline \text{MCR} \\ \hline \end{array}$	1	マスターコントロールのリセット						114
F-32	$\overline{\text{F-32}} \begin{array}{ c c } \hline \text{SET} & \text{OUT} \\ \hline \end{array}$	2	セットコイル	ON					117
F-33	$\overline{\text{F-33}} \begin{array}{ c c } \hline \text{RST} & \text{OUT} \\ \hline \end{array}$	2	リセットコイル	ON					118
F-34	$\overline{\text{F-34}} \begin{array}{ c c c c } \hline \text{TSET} & \text{n}_1 & \text{n}_2 & \text{BIT} \\ \hline \end{array}$	4	時計の現在値との比較 (指定リレーのセット)	ON					120
F-35	$\overline{\text{F-35}} \begin{array}{ c c c c } \hline \text{TRST} & \text{n}_1 & \text{n}_2 & \text{BIT} \\ \hline \end{array}$	4	時計の現在値との比較 (指定リレーのリセット)	ON					121
F-36	$\overline{\text{F-36}} \begin{array}{ c c c c } \hline \text{TADD} & \text{S}_1 & \text{S}_2 & \text{D} \\ \hline \end{array}$	4	時計の加算	↑	↕	↕	↕	↕	122

JW-31CUH/H1ではプログラムできません。

命令語	シンボル	語数	機能	実行条件	フラグ				参照ページ
					ゼロ 07357	キャリー 07356	エラー 07355	オーバー 07354	
F-37		4	時計の減算	↑	↑↓	↑↓	↑↓	↑↓	9・123
F-38		2	時計現在値の転送	↑					124
F-40		1	END命令						125
F-41		1	ジャンプコントロールのセット	OFF					126
F-42		1	ジャンプコントロールのリセット						126
F-43		1	ビット反転(ACCの内容を反転)						128
F-44		1	ON時微分接点	↑					129
F-45		1	OFF時微分接点	↓					130
F-47		1	レベル演算条件セット						131
F-48		1	レベル演算条件リセット						131
F-49		1	条件END	OFF					132
F-50		3	4 16デコーダ	↑					133
F-51		3	16 4エンコーダ	↑					134
F-52		3	7SEGデコーダ	↑					135
F-53		3	BCD(4桁) BIN(16ビット)変換	↑	0	0	↑↓	0	136
F-54		3	BIN(16ビット) BCD(6桁)変換	↑					137
F-55		3	上位4ビットと下位4ビットの交換	↑					138
F-56		3	1バイトデータの10の補数	↑	0	0	↑↓	0	139
F-56w		3	1ワードデータの10の補数	↑	0	0	↑↓	0	140
F-56d		3	2ワードデータの10の補数	↑	0	0	↑↓	0	141
F-57		3	1バイトデータの2の補数	↑					142
F-57w		3	1ワードデータの2の補数	↑					143
F-57d		3	2ワードデータの2の補数	↑					144
F-58		4	ONビット数の合計	↑					145
F-60		2	両方向シフトレジスタ(1バイト) シフト方向指示入力 シフト入力 データ入力 リセット入力	シフト 入力 ↑	↑↓	↑↓	0	↑↓	146
F-60w		2	両方向シフトレジスタ(1ワード) シフト方向指示入力 シフト入力 データ入力 リセット入力	シフト 入力 ↑	↑↓	↑↓	0	↑↓	149
F-60d		2	両方向シフトレジスタ(2ワード) シフト方向指示入力 シフト入力 データ入力 リセット入力	シフト 入力 ↑	↑↓	↑↓	0	↑↓	150
F-61		2	非同期シフトレジスタ(1バイト) シフト方向指示入力 シフト入力	シフト 入力ON	0	↑↓	0	↑↓	151
F-61w		2	非同期シフトレジスタ(1ワード) シフト方向指示入力 シフト入力	シフト 入力ON	0	↑↓	0	↑↓	153

JW-31CUH/H1ではプログラムできません。

命令語	シンボル	語数	機能	実行条件	フラグ				参照ページ
					ゼロ 07357	キャリー 07356	エラー 07355	リセット 07354	
F-61d		2	非同期シフトレジスタ(2ワード) シフト方向指示入力 シフト入力	シフト 入力ON	0	↑↓	0	↑↓	9・154
F-62		2	BCD2桁のアップ・ダウンカウンタ アップ・ダウン指示入力 カウント入力 リセット入力	カウント 入力 ↑	↑↓	↑↓	↑↓	↑↓	155
F-62w		2	BCD4桁のアップ・ダウンカウンタ アップ・ダウン指示入力 カウント入力 リセット入力	カウント 入力 ↑	↑↓	↑↓	↑↓	↑↓	156
F-62d		2	BCD8桁のアップ・ダウンカウンタ アップ・ダウン指示入力 カウント入力 リセット入力	カウント 入力 ↑	↑↓	↑↓	↑↓	↑↓	157
F-63		2	バイナリ加算カウンタ(1バイト)	↑	↑↓	↑↓	0	↑↓	158
F-63w		2	バイナリ加算カウンタ(1ワード)	↑	↑↓	↑↓	0	↑↓	159
F-64		2	バイナリ減算カウンタ(1バイト)	↑	↑↓	↑↓	0	↑↓	160
F-64w		2	バイナリ減算カウンタ(1ワード)	↑	↑↓	↑↓	0	↑↓	161
F-65		2	BCD加算カウンタ(1バイト)	↑	↑↓	↑↓	↑↓	↑↓	162
F-65w		2	BCD加算カウンタ(1ワード)	↑	↑↓	↑↓	↑↓	↑↓	163
F-66		2	BCD減算カウンタ(1バイト)	↑	↑↓	↑↓	↑↓	↑↓	164
F-66w		2	BCD減算カウンタ(1ワード)	↑	↑↓	↑↓	↑↓	↑↓	165
F-67		3	桁シフト(上位シフト)	↑					166
F-68		3	桁シフト(下位シフト)	↑					167
F-69		3	桁転送	↑					168
F-70		4	nバイト一括転送	↑					169
F-70w		4	nワード一括転送	↑					170
F-71		4	8進定数(1バイト)一括転送	↑					171
F-71w		4	8進定数(1ワード)一括転送	↑					172
F-72		4	ファイル1のレジスタへのnバイト分配	↑					173
F-72w		4	ファイル1のレジスタへのnワード分配	↑					174
F-73		4	ファイル1のレジスタからのnバイト抽出	↑					175
F-73w		4	ファイル1のレジスタからのnワード抽出	↑					176
F-74		4	nバイト転送	↑					177
F-74w		4	nワード転送	↑					178
F-76		4	nバイト一括転送	↑					179
F-76w		4	nワード一括転送	↑					180
F-77		4	サムチェックコード生成	↑					181
F-78		4	データのチェック	↑	0	0	↑↓	0	182
F-79		4	1バイトデータの並べかえ	↑					183
F-79w		4	1ワードデータの並べかえ	↑					184

JW-31CUH/H1ではプログラムできません。

命令語	シンボル	語数	機能	実行条件	フラグ				参照ページ
					ゼロ 07357	キャリー 07356	エラー 07355	オーバー 07354	
F-80	$\overline{\text{F-80 IORF}}$ R,S	2	I/Oリフレッシュ	ON	0	↑↓	↑↓	↑↓	9・185
F-82	$\overline{\text{F-82 IORF}}$ SW	2	特殊I/Oのリフレッシュ	ON	0	↑↓	↑↓	↑↓	186
F-85	$\overline{\text{F-85 PRRD}}$ n ₁ SW,n ₂ D	4	特殊I/Oからの読出	↑	0	↑↓	↑↓	↑↓	187
F-86	$\overline{\text{F-86 PRWR}}$ n ₁ D SW,n ₂	4	特殊I/Oへの書込	↑	0	↑↓	↑↓	↑↓	187
F-90	$\overline{\text{F-90 REM}}$ n	2	リマーク n = 0000 ~ 3777						188
F-91	$\overline{\text{F-91 BCD8}}$ n ₁ n ₂ D	4	BCD定数(8桁)の転送	↑					189
F-97	$\overline{\text{F-97 DML8}}$ n ₁ n ₂ D	4	10進定数(8桁)の転送	↑					190
F-100	$\overline{\text{F-100 ADRS}}$ S D	3	間接アドレスの設定	↑					191
F-101	$\overline{\text{F-101 SEGM}}$ n file N D	4	間接アドレスの設定	↑					192
F-102	$\overline{\text{F-102 MRD}}$ n file N D	4	直接指定アドレスのレジスタからの読出 (1バイト)	↑					193
F-102w	$\overline{\text{F-102w MRD}}$ n file N D	4	直接指定アドレスのレジスタからの読出 (1ワード)	↑					194
F-103	$\overline{\text{F-103 MWR}}$ S n file N	4	直接指定アドレスのレジスタへの書込 (1バイト)	↑					195
F-103w	$\overline{\text{F-103w MWR}}$ S n file N	4	直接指定アドレスのレジスタへの書込 (1ワード)	↑					196
F-112	$\overline{\text{F-112 NCMP}}$ S ₁ S ₂ S ₃	4	nバイト一括比較	ON	↑↓	↑↓	0	↑↓	197
F-112w	$\overline{\text{F-112w NCMP}}$ S ₁ S ₂ S ₃	4	nワード一括比較	ON	↑↓	↑↓	0	↑↓	198
F-116	$\overline{\text{F-116 DIV}}$ S ₁ S ₂ D	4	レジスタ(BCD8桁)とレジスタ(BCD8桁) の除算(小数部4桁)	↑	0	0	↑↓	0	199
F-130	$\overline{\text{F-130 BIT}}$ S ₁ S ₂	3	ビット抽出(間接指定)	ON	0	↑↓	0	0	200
F-131	$\overline{\text{F-131 BIT}}$ n S	3	ビット抽出(直接指定)	ON	0	↑↓	0	0	201
F-132	$\overline{\text{F-132 S/R}}$ S D	3	ビットセット/リセット(間接指定) セット/リセット指示入力 入力条件	ON					202
F-133	$\overline{\text{F-133 S/R}}$ n D	3	ビットセット/リセット(直接指定) セット/リセット指示入力 入力条件	ON					203
F-140	$\overline{\text{F-140 LABL}}$ LBn	2	ラベルの設定 LB0000 ~ LB1377						204
F-141	$\overline{\text{F-141 JMP}}$ LBn	2	ラベルへジャンプ	ON					205
F-142	$\overline{\text{F-142 CALL}}$ LBn	2	ラベルをサブルーチンコール	↑					207
F-143	$\overline{\text{F-143 RET}}$	1	サブルーチンからのリターン						207
F-144	$\overline{\text{F-144 FOR}}$ n	2	ループ回数の設定	↑					209
F-145	$\overline{\text{F-145 NEXT}}$	1	ループの終了						209
F-146	$\overline{\text{F-146 FORR}}$ S	2	ループ回数のレジスタ設定	↑					211
F-147	$\overline{\text{F-147 EXIT}}$	1	ループの条件終了	OFF					212
F-148	$\overline{\text{F-148 CAL+}}$ LBn S	2	レジスタ設定ラベルをサブルーチンコール	↑					213
F-149	$\overline{\text{F-149 RETC}}$	1	サブルーチンからの条件リターン	OFF					214
F-151	$\overline{\text{F-151 JMP+}}$ LBn S	3	レジスタ設定ラベルへジャンプ	ON					215
F-153	$\overline{\text{F-153 -BIN}}$ S D	3	BCD(8桁) BIN(32ビット)変換	↑	0	0	↑↓	0	216
F-154	$\overline{\text{F-154 -BCD}}$ S D	3	BIN(32ビット) BCD(10桁)変換	↑					217

JW-31CUH/HIではプログラムできません。

命令語	シンボル	語数	機能	実行条件	フラグ				参照ページ
					ゼロ 07357	キャリー 07356	エラー 07355	オーバー 07354	
F-155		3	時(4桁), 分, 秒 秒(BCD8桁)	↑	0	0	↑↓	0	9・218
F-156		3	秒(BCD8桁) 時(4桁), 分, 秒(BCD)	↑	0	0	↑↓	0	219
F-160		4	両方向シフトレジスタ(nビット) シフト方向指示入力 シフト入力 データ入力 リセット入力 (シフトは1ビット) (S1)=0~256 (S2)=0~7	シフト 入力 ↑	↑↓	↑↓	0	↑↓	220
Fc160		4	両方向シフトレジスタ(nビット) シフト方向指示入力 シフト入力 データ入力 リセット入力 (シフトは1ビット) n1=0~377 n2=0~7	シフト 入力 ↑	↑↓	↑↓	0	↑↓	221
F-161		3	非同期シフトレジスタ(nバイト) シフト方向指示入力 シフト入力	シフト 入力ON ↑	0	↑↓	0	↑↓	222
F-161w		3	非同期シフトレジスタ(nワード) シフト方向指示入力 シフト入力	シフト 入力ON ↑	0	↑↓	0	↑↓	224
F-163		2	バイナリ加算(+2)カウンタ(1バイト)	↑	↑↓	↑↓	0	↑↓	226
F-163w		2	バイナリ加算(+2)カウンタ(1ワード)	↑	↑↓	↑↓	0	↑↓	227
F-164		2	バイナリ減算(-2)カウンタ(1バイト)	↑	↑↓	↑↓	0	↑↓	228
F-164w		2	バイナリ減算(-2)カウンタ(1ワード)	↑	↑↓	↑↓	0	↑↓	229
F-170		4	データ挿入(1バイト)	↑	0	0	↑↓	0	230
F-170w		4	データ挿入(1ワード)	↑	0	0	↑↓	0	231
F-171		4	データ削除(1バイト)	↑	0	0	↑↓	0	232
F-171w		4	データ削除(1ワード)	↑	0	0	↑↓	0	233
F-172		4	データ検索(1バイト)	↑	↑↓	↑↓	0	0	234
F-172w		4	データ検索(1ワード)	↑	↑↓	↑↓	0	0	235
F-173		4	データチェンジ(1バイト) モード指定 実行入力	実行入力 ↑	↑↓	↑↓	0	0	236
F-173w		4	データチェンジ(1ワード) モード指定 実行入力	実行入力 ↑	↑↓	↑↓	0	0	237
F-174		3	レジスタ間(1バイト)データ交換	↑					238
F-175		3	上位4ビットと下位4ビットの交換	↑					239
F-176		4	直接指定アドレスのレジスタからの読出 (256バイト)	↑					240
F-177		4	直接指定アドレスのレジスタへの書込 (256バイト)	↑					242

JW-31CUH/H1ではプログラムできません。

命令語	シンボル	語数	機能	実行条件	フラグ				参照ページ
					ゼロ 07357	キャリー 07356	エラー 07355	オーバー 07354	
F-180	$\overline{F-180}$ CP> S ₁ S ₂ BIT	4	レジスタ間(1バイト)の比較 >、リレー出力付)	ON	0	0	0	0	9・243
F-180w	$\overline{F-180w}$ CP> S ₁ S ₂ BIT	4	レジスタ間(1ワード)の比較 >、リレー出力付)	ON	0	0	0	0	244
Fc180	$\overline{Fc180}$ CP> S n BIT	4	レジスタと定数(1バイト)の比較 >、リレー出力付)	ON	0	0	0	0	245
Fc180w	$\overline{Fc180w}$ CP> S n BIT	4	レジスタと定数(1ワード)の比較 >、リレー出力付)	ON	0	0	0	0	246
F-181	$\overline{F-181}$ CP< S ₁ S ₂ BIT	4	レジスタ間(1バイト)の比較 <、リレー出力付)	ON	0	0	0	0	243
F-181w	$\overline{F-181w}$ CP< S ₁ S ₂ BIT	4	レジスタ間(1ワード)の比較 <、リレー出力付)	ON	0	0	0	0	244
Fc181	$\overline{Fc181}$ CP< S n BIT	4	レジスタと定数(1バイト)の比較 <、リレー出力付)	ON	0	0	0	0	245
Fc181w	$\overline{Fc181w}$ CP< S n BIT	4	レジスタと定数(1ワード)の比較 <、リレー出力付)	ON	0	0	0	0	246
F-182	$\overline{F-182}$ CP= S ₁ S ₂ BIT	4	レジスタ間(1バイト)の比較 =、リレー出力付)	ON	0	0	0	0	243
F-182w	$\overline{F-182w}$ CP= S ₁ S ₂ BIT	4	レジスタ間(1ワード)の比較 =、リレー出力付)	ON	0	0	0	0	244
Fc182	$\overline{Fc182}$ CP= S n BIT	4	レジスタと定数(1バイト)の比較 =、リレー出力付)	ON	0	0	0	0	245
Fc182w	$\overline{Fc182w}$ CP= S n BIT	4	レジスタと定数(1ワード)の比較 =、リレー出力付)	ON	0	0	0	0	246
F-183	$\overline{F-183}$ CP>= S ₁ S ₂ BIT	4	レジスタ間(1バイト)の比較 ≥、リレー出力付)	ON	0	0	0	0	243
F-183w	$\overline{F-183w}$ CP>= S ₁ S ₂ BIT	4	レジスタ間(1ワード)の比較 ≥、リレー出力付)	ON	0	0	0	0	244
Fc183	$\overline{Fc183}$ CP>= S n BIT	4	レジスタと定数(1バイト)の比較 ≥、リレー出力付)	ON	0	0	0	0	245
Fc183w	$\overline{Fc183w}$ CP>= S n BIT	4	レジスタと定数(1ワード)の比較 ≥、リレー出力付)	ON	0	0	0	0	246
F-184	$\overline{F-184}$ CP<= S ₁ S ₂ BIT	4	レジスタ間(1バイト)の比較 ≤、リレー出力付)	ON	0	0	0	0	243
F-184w	$\overline{F-184w}$ CP<= S ₁ S ₂ BIT	4	レジスタ間(1ワード)の比較 ≤、リレー出力付)	ON	0	0	0	0	244
Fc184	$\overline{Fc184}$ CP<= S n BIT	4	レジスタと定数(1バイト)の比較 ≤、リレー出力付)	ON	0	0	0	0	245
Fc184w	$\overline{Fc184w}$ CP<= S n BIT	4	レジスタと定数(1ワード)の比較 ≤、リレー出力付)	ON	0	0	0	0	246
F-185	$\overline{F-185}$ CP<> S ₁ S ₂ BIT	4	レジスタ間(1バイト)の比較 ≠、リレー出力付)	ON	0	0	0	0	243
F-185w	$\overline{F-185w}$ CP<> S ₁ S ₂ BIT	4	レジスタ間(1ワード)の比較 ≠、リレー出力付)	ON	0	0	0	0	244
Fc185	$\overline{Fc185}$ CP<> S n BIT	4	レジスタと定数(1バイト)の比較 ≠、リレー出力付)	ON	0	0	0	0	245
Fc185w	$\overline{Fc185w}$ CP<> S n BIT	4	レジスタと定数(1ワード)の比較 ≠、リレー出力付)	ON	0	0	0	0	246
F-202	$\overline{F-202}$ OPCH UN,C,ST file N n	4	オープンチャンネル(局番8進定数設定)	ON					247
F-203	$\overline{F-203}$ OPCH UN,C,ST file N n	4	オープンチャンネル(局番16進定数設定)	ON					247
F-204	$\overline{F-204}$ SEND n S	3	送信命令	↑	↑↓	↑↓	↑↓	↑↓	248
F-205	$\overline{F-205}$ RCV n D	3	受信命令	↑	↑↓	↑↓	↑↓	↑↓	249
F-206	$\overline{F-206}$ EOP1 UN1,CH ST1 UN2	4	オープンチャンネル1(階層通信設定)	ON					250
F-207	$\overline{F-207}$ EOP2 ST2 file N n	4	オープンチャンネル2(階層通信設定)	ON					250
F-210	$\overline{F-210}$ ADD S ₁ S ₂ D	4	レジスタ間のバイナリ加算 (8ビット+8ビット)	↑	↑↓	↑↓	0	↑↓	251
F-210w	$\overline{F-210w}$ ADD S ₁ S ₂ D	4	レジスタ間のバイナリ加算 (16ビット+16ビット)	↑	↑↓	↑↓	0	↑↓	252

命令語	シンボル	語数	機能	実行条件	フラグ				参照ページ
					ゼロ 07357	キャリー 07356	オーバー 07355	オーバー 07354	
F-210d	$\boxed{\text{F-210d}} \begin{array}{ c c c } \hline \text{ADD} & S_1 & S_2 & D \\ \hline \end{array}$	4	レジスタ間のバイナリ加算 (32ビット+32ビット)	↑	↑↓	↑↓	0	↑↓	9・253
Fc210	$\boxed{\text{Fc210}} \begin{array}{ c c c } \hline \text{ADD} & S_1 & n & D \\ \hline \end{array}$	4	レジスタと定数のバイナリ加算 (8ビット+8ビット)	↑	↑	↑	0	↑	254
Fc210w	$\boxed{\text{Fc210w}} \begin{array}{ c c c } \hline \text{ADD} & S_1 & n & D \\ \hline \end{array}$	4	レジスタと定数のバイナリ加算 (16ビット+16ビット)	↑	↑	↑	0	↑	255
Fc210d	$\boxed{\text{Fc210d}} \begin{array}{ c c c } \hline \text{ADD} & S_1 & n & D \\ \hline \end{array}$	4	レジスタと定数のバイナリ加算 (32ビット+16ビット)	↑	↑	↑	0	↑	256
F-211	$\boxed{\text{F-211}} \begin{array}{ c c c } \hline \text{SUB} & S_1 & S_2 & D \\ \hline \end{array}$	4	レジスタ間のバイナリ減算 (8ビット-8ビット)	↑	↑	↑	0	↑	257
F-211w	$\boxed{\text{F-211w}} \begin{array}{ c c c } \hline \text{SUB} & S_1 & S_2 & D \\ \hline \end{array}$	4	レジスタ間のバイナリ減算 (16ビット-16ビット)	↑	↑	↑	0	↑	258
F-211d	$\boxed{\text{F-211d}} \begin{array}{ c c c } \hline \text{SUB} & S_1 & S_2 & D \\ \hline \end{array}$	4	レジスタ間のバイナリ減算 (32ビット-32ビット)	↑	↑	↑	0	↑	259
Fc211	$\boxed{\text{Fc211}} \begin{array}{ c c c } \hline \text{SUB} & S_1 & n & D \\ \hline \end{array}$	4	レジスタと定数のバイナリ減算 (8ビット-8ビット)	↑	↑	↑	0	↑	260
Fc211w	$\boxed{\text{Fc211w}} \begin{array}{ c c c } \hline \text{SUB} & S_1 & n & D \\ \hline \end{array}$	4	レジスタと定数のバイナリ減算 (16ビット-16ビット)	↑	↑	↑	0	↑	261
Fc211d	$\boxed{\text{Fc211d}} \begin{array}{ c c c } \hline \text{SUB} & S_1 & n & D \\ \hline \end{array}$	4	レジスタと定数のバイナリ減算 (32ビット-16ビット)	↑	↑	↑	0	↑	262
F-212	$\boxed{\text{F-212}} \begin{array}{ c c c } \hline \text{WNDW} & S_1 & S_2 & S_3 \\ \hline \end{array}$	4	ウィンドウコンパレータ (1バイトレジスタ間)	ON	↑	↑	↑	↑	263
F-212w	$\boxed{\text{F-212w}} \begin{array}{ c c c } \hline \text{WNDW} & S_1 & S_2 & S_3 \\ \hline \end{array}$	4	ウィンドウコンパレータ (1ワードレジスタ間)	ON	↑	↑	↑	↑	264
F-212d	$\boxed{\text{F-212d}} \begin{array}{ c c c } \hline \text{WNDW} & S_1 & S_2 & S_3 \\ \hline \end{array}$	4	ウィンドウコンパレータ (2ワードレジスタ間)	ON	↑	↑	↑	↑	265
Fc212	$\boxed{\text{Fc212}} \begin{array}{ c c c } \hline \text{WNDW} & S_1 & n_1 & n_2 \\ \hline \end{array}$	4	ウィンドウコンパレータ (1バイト8進定数間)	ON	↑	↑	↑	↑	266
Fc212w	$\boxed{\text{Fc212w}} \begin{array}{ c c c } \hline \text{WNDW} & S_1 & n_1 & n_2 \\ \hline \end{array}$	4	ウィンドウコンパレータ (1ワード8進定数間)	ON	↑	↑	↑	↑	267
Fx212	$\boxed{\text{Fx212}} \begin{array}{ c c c } \hline \text{WNDW} & S_1 & n_1 & n_2 \\ \hline \end{array}$	4	ウィンドウコンパレータ (1バイト16進定数間)	ON	↑	↑	↑	↑	268
Fx212w	$\boxed{\text{Fx212w}} \begin{array}{ c c c } \hline \text{WNDW} & S_1 & n_1 & n_2 \\ \hline \end{array}$	4	ウィンドウコンパレータ (1ワード16進定数間)	ON	↑	↑	↑	↑	269
F-215	$\boxed{\text{F-215}} \begin{array}{ c c c } \hline \text{MUL} & S_1 & S_2 & D \\ \hline \end{array}$	4	レジスタ間のバイナリ乗算 (8ビット×8ビット)	↑	0	0	0	0	270
F-215w	$\boxed{\text{F-215w}} \begin{array}{ c c c } \hline \text{MUL} & S_1 & S_2 & D \\ \hline \end{array}$	4	レジスタ間のバイナリ乗算 (16ビット×16ビット)	↑	0	0	0	0	271
F-215d	$\boxed{\text{F-215d}} \begin{array}{ c c c } \hline \text{MUL} & S_1 & S_2 & D \\ \hline \end{array}$	4	レジスタ間のバイナリ乗算 (32ビット×32ビット)	↑	0	0	0	0	272
Fc215	$\boxed{\text{Fc215}} \begin{array}{ c c c } \hline \text{MUL} & S_1 & n & D \\ \hline \end{array}$	4	レジスタと定数のバイナリ乗算 (8ビット×8ビット)	↑	0	0	0	0	273
Fc215w	$\boxed{\text{Fc215w}} \begin{array}{ c c c } \hline \text{MUL} & S_1 & n & D \\ \hline \end{array}$	4	レジスタと定数のバイナリ乗算 (16ビット×16ビット)	↑	0	0	0	0	274
Fc215d	$\boxed{\text{Fc215d}} \begin{array}{ c c c } \hline \text{MUL} & S_1 & n & D \\ \hline \end{array}$	4	レジスタと定数のバイナリ乗算 (32ビット×16ビット)	↑	0	0	0	0	275
F-216	$\boxed{\text{F-216}} \begin{array}{ c c c } \hline \text{DIV} & S_1 & S_2 & D \\ \hline \end{array}$	4	レジスタ間のバイナリ除算 (8ビット÷8ビット)	↑	0	0	↑	0	276
F-216w	$\boxed{\text{F-216w}} \begin{array}{ c c c } \hline \text{DIV} & S_1 & S_2 & D \\ \hline \end{array}$	4	レジスタ間のバイナリ除算 (15ビット÷15ビット)	↑	0	0	↑	0	277
F-216d	$\boxed{\text{F-216d}} \begin{array}{ c c c } \hline \text{DIV} & S_1 & S_2 & D \\ \hline \end{array}$	4	レジスタ間のバイナリ除算 (31ビット÷31ビット)	↑	0	0	↑	0	278
Fc216	$\boxed{\text{Fc216}} \begin{array}{ c c c } \hline \text{DIV} & S_1 & n & D \\ \hline \end{array}$	4	レジスタと定数のバイナリ除算 (8ビット÷8ビット)	↑	0	0	↑	0	279
Fc216w	$\boxed{\text{Fc216w}} \begin{array}{ c c c } \hline \text{DIV} & S_1 & n & D \\ \hline \end{array}$	4	レジスタと定数のバイナリ除算 (15ビット÷15ビット)	↑	0	0	↑	0	280
Fc216d	$\boxed{\text{Fc216d}} \begin{array}{ c c c } \hline \text{DIV} & S_1 & n & D \\ \hline \end{array}$	4	レジスタと定数のバイナリ除算 (31ビット÷15ビット)	↑	0	0	↑	0	281
F-231	$\boxed{\text{F-231}} \text{MCRN}$	1	マスターコントロール ネスティング リセット						282
F-242	$\boxed{\text{F-242}} \text{JCRN}$	1	ジャンプコントロール ネスティング リセット						283
F-252	$\boxed{\text{F-252}} \begin{array}{ c c c } \hline \text{→ASC} & S & n & D \\ \hline \end{array}$	4	HEX ASCII変換	↑					284

命令語	シンボル	語数	機能	実行条件	フラグ				参照ページ			
					ゼロ 07357	キャリー 07356	エラー 07355	オーバー 07354				
F-253	<input type="checkbox"/> F-253 →HEX	S	n	D	4	ASCII HEX変換	↑	0	0	↑↓	0	9・285
F-260	<input type="checkbox"/> F-260 RTMR	S	D	BIT	4	現在値と設定値が、レジスタ指定可能なタイマ命令	ON	0	0	↑↓	0	286
Fc260	<input type="checkbox"/> Fc260 RTMR	n	D	BIT	4	現在値がレジスタ指定可能なタイマ命令(設定値はBCD定数)	ON	0	0	↑↓	0	287
F-261	<input type="checkbox"/> F-261 RCNT	S	D	BIT	4	現在値と設定値が、レジスタ指定可能なカウンタ命令 計数入力 リセット入力	計数入力 ↑	0	0	↑↓	0	288
Fc261	<input type="checkbox"/> Fc261 RCNT	n	D	BIT	4	現在値がレジスタ指定可能なカウンタ命令(設定値はBCD定数) 計数入力 リセット入力	計数入力 ↑	0	0	↑↓	0	289
F-263	<input type="checkbox"/> F-263 INC4	D			2	バイナリ加算(+4)カウンタ(1バイト)	↑	↑↓	↑↓	0	↑↓	290
F-263w	<input type="checkbox"/> F-263w INC4	D			2	バイナリ加算(+4)カウンタ(1ワード)	↑	↑↓	↑↓	0	↑↓	291
F-264	<input type="checkbox"/> F-264 DEC4	D			2	バイナリ減算(-4)カウンタ(1バイト)	↑	↑↓	↑↓	0	↑↓	292
F-264w	<input type="checkbox"/> F-264w DEC4	D			2	バイナリ減算(-4)カウンタ(1ワード)	↑	↑↓	↑↓	0	↑↓	293
F-310	<input type="checkbox"/> F-310 SADD	S ₁	S ₂	D	4	レジスタ間の符号付バイナリ加算 (31ビット+31ビット)	↑	↑↓	↑↓	↑↓	↑↓	294
F-311	<input type="checkbox"/> F-311 SSUB	S ₁	S ₂	D	4	レジスタ間の符号付バイナリ減算 (31ビット-31ビット)	↑	↑↓	↑↓	↑↓	↑↓	295
F-315	<input type="checkbox"/> F-315 SMUL	S ₁	S ₂	D	4	レジスタ間の符号付バイナリ乗算 (31ビット×31ビット)	↑	0	0	0	0	296
F-316	<input type="checkbox"/> F-316 SDIV	S ₁	S ₂	D	4	レジスタ間の符号付バイナリ除算 (31ビット÷31ビット)	↑	0	0	↑↓	0	297
NOP					1	無効命令						—

〔 2 〕 動作による分類

基本命令

分		類	命令語	参照ページ	
シーケンス命令			STR	8・2	
			STR NOT	2	
			AND	3	
			AND NOT	3	
			OR	3	
			OR NOT	4	
			AND STR	4	
			OR STR	4	
			OUT	2	
タイマ命令	減算タイマ	BCD	TMR	6	
			DTMR (BCD)	6	
		バイナリ	DTMR (BIN)	6	
	加算タイマ	BCD	UTMR (BCD)	6	
			バイナリ	UTMR (BIN)	6
カウンタ命令	減算カウンタ	BCD	CNT	7	
			DCNT (BCD)	7	
		バイナリ	DCNT (BIN)	7	
	加算カウンタ	BCD	UCNT (BCD)	7	
			バイナリ	UCNT (BIN)	7
メンテナンスディスプレイ			MD (F-20)	9	

応用命令

分		類	命令語	参照ページ
転送命令	レジスタ間の転送	1バイト	F-00	9・19
		1ワード		
		2ワード		
		nバイト	F-70	169
		nワード		
		nバイト (間接指定)	F-76	179
	nワード (間接指定)			
	nバイト (同一データ)	F-74	177	
	nワード (同一データ)			
	BCD定数の転送	2桁	F-01	22
		4桁		
		8桁	F-91	189
	10進定数の転送	1バイト	F-07	36
		1ワード		
		8桁	F-97	190
	8進定数の転送	1バイト	F-08	38
		1ワード		
		nバイト	F-71	171
	nワード			
	分配	1バイト	F-05	31
1ワード				
nバイト (ファイル1)		F-72	173	
nワード (ファイル1)				
抽出	1バイト	F-06	34	
	1ワード			
	nバイト (ファイル1)	F-73	175	
	nワード (ファイル1)			
桁転送	4ビット	F-69	168	
ファイルの読出	1バイト	F-102	193	
	1ワード			
	256バイト	F-176	240	
ファイルへの書込	1バイト	F-103	195	
	1ワード			
	256バイト	F-177	242	

分		類	命令語	参照ページ		
算 術 演 算 命 令	BCD 加算	レジスタ間	2桁 + 2桁 4桁 + 4桁 8桁 + 8桁	F-10	9・43	
		レジスタ と定数	2桁 + 2桁 4桁 + 4桁 8桁 + 4桁	Fc10	48	
			BCD 減算	レジスタ間	2桁 - 2桁 4桁 - 4桁 8桁 - 8桁	F-11
	レジスタ と定数			2桁 - 2桁 4桁 - 4桁 8桁 - 4桁	Fc11	57
		BCD 乗算		レジスタ間	4桁 × 4桁 8桁 × 8桁	F-15
			レジスタ と定数	4桁 × 3桁 8桁 × 4桁	Fc15	83
	BCD 除算			レジスタ間	4桁 ÷ 2桁 8桁 ÷ 8桁	F-16
		レジスタ と定数		8桁 ÷ 8桁 (小数部4桁)	F-116	199
			レジスタ と定数	4桁 ÷ 2桁 8桁 ÷ 4桁	Fc16	88
	バイナリ 加算			レジスタ間	8ビット + 8ビット 16ビット + 16ビット 32ビット + 32ビット	F-210
		レジスタ と定数		8ビット + 8ビット 16ビット + 16ビット 32ビット + 16ビット	Fc210	254
			レジスタ間 (符号付)	31ビット + 31ビット	F-310	294
	バイナリ 減算		レジスタ間	8ビット - 8ビット 16ビット - 16ビット 32ビット - 32ビット	F-211	257
		レジスタ と定数	8ビット - 8ビット 16ビット - 16ビット 32ビット - 16ビット	Fc211	260	
			レジスタ間 (符号付)	31ビット - 31ビット	F-311	295
	バイナリ 乗算		レジスタ間	8ビット × 8ビット 16ビット × 16ビット 32ビット × 32ビット	F-215	270
		レジスタ と定数	8ビット × 8ビット 16ビット × 16ビット 32ビット × 16ビット	Fc215	273	
			レジスタ間 (符号付)	31ビット × 31ビット	F-315	296

分		類	命令語	参照ページ		
算 術 演 算 命 令	バイナリ 除算	レジスタ間	8ビット ÷ 8ビット 15ビット ÷ 15ビット 31ビット ÷ 31ビット	F-216	9・276	
		レジスタ と定数	8ビット ÷ 8ビット 15ビット ÷ 15ビット 31ビット ÷ 15ビット	Fc216	279	
			レジスタ間 (符号付)	31ビット ÷ 31ビット	F-316	297
	論理積		レジスタ間	8ビット 16ビット 32ビット	F-13	67
		レジスタと 8進定数	8ビット 16ビット	Fc13	70	
			レジスタと 16進定数	8ビット 16ビット	Fx13	72
	論理和			レジスタ間	8ビット 16ビット 32ビット	F-14
		レジスタと 8進定数		8ビット 16ビット	Fc14	77
			レジスタと 16進定数	8ビット 16ビット	Fx14	79
	一致			レジスタ間	8ビット 16ビット 32ビット	F-17
		レジスタと 8進定数		8ビット 16ビット	Fc17	93
			レジスタと 16進定数	8ビット 16ビット	Fx17	95
	排他的 論理和			レジスタ間	8ビット 16ビット 32ビット	F-18
		レジスタと 8進定数		8ビット 16ビット	Fc18	100
			レジスタと 16進定数	8ビット 16ビット	Fx18	102
	反 転			レジスタ間	8ビット 16ビット 32ビット	F-09
		比較 命 令		レジスタ間	1バイト 1ワード 2ワード	F-12
			nバイト nワード		F-112	197
	レジスタと 8進定数		1バイト 1ワード		Fc12	63
	レジスタと 16進定数	1バイト 1ワード	Fx12	65		

分		類	命令語	参照ページ	
比較命令	比較 (リレー出力付)	レジスタ間	1バイト	F-180	9・243
			1ワード		
		レジスタと8進定数	1バイト	Fc180	245
			1ワード		
		レジスタ間	1バイト	F-181	243
			1ワード		
		レジスタと8進定数	1バイト	Fc181	245
			1ワード		
		レジスタ間	1バイト	F-182	243
			1ワード		
	レジスタと8進定数	1バイト	Fc182	245	
		1ワード			
	レジスタ間	1バイト	F-183	243	
		1ワード			
	レジスタと8進定数	1バイト	Fc183	245	
		1ワード			
	レジスタ間	1バイト	F-184	243	
		1ワード			
	レジスタと8進定数	1バイト	Fc184	245	
		1ワード			
レジスタ間	1バイト	F-185	243		
	1ワード				
レジスタと8進定数	1バイト	Fc185	245		
	1ワード				
ウィンドウコンパレータ	レジスタ間	1バイト	F-212	263	
		1ワード			
		2ワード			
	レジスタと8進定数	1バイト	Fc212	266	
		1ワード			
	レジスタと16進定数	1バイト	Fx212	268	
1ワード					
変換命令	BCD BIN変換	2桁 8ビット	F-03	27	
		4桁 16ビット			
		4桁 16ビット	F-53	136	
		8桁 32ビット	F-153	216	

分		類	命令語	参照ページ
変換命令	BIN BCD変換	8ビット 2桁	F-04	9・29
		16ビット 6桁		
		16ビット 6桁	F-54	137
		32ビット 10桁	F-154	217
	HEX ASCII変換		F-252	284
	ASCII HEX変換		F-253	285
	時・分・秒 秒変換		F-155	218
	秒 時・分・秒変換		F-156	219
	4 16デコーダ		F-50	133
	16 4エンコーダ		F-51	134
	7 SEGデコーダ		F-52	135
	10の補数	2桁	F-56	139
		4桁		
		8桁		
	2の補数	8ビット	F-57	142
16ビット				
32ビット				
ONビット数の合計		F-58	145	
極座標変換		F-28	112	
直交座標変換		F-29	113	
交換命令	データの交換	1バイト	F-02	24
		1ワード		
		2ワード		
	nバイト		F-174	238
上位4ビットと下位4ビットの交換	1バイト	F-55	138	
	nバイト	F-175	239	

分		類	命令語	参照ページ
データ処理命令	データの挿入	1バイト	F-170	9・230
		1ワード		
	データの削除	1バイト	F-171	232
		1ワード		
	データの検索	1バイト	F-172	234
		1ワード		
	データの変更	1バイト	F-173	236
		1ワード		
	データの並び替え	1バイト	F-79	183
		1ワード		
SIN関数		F-22	106	
COS関数		F-23	107	
TAN関数		F-24	108	
ASIN関数		F-25	109	
ACOS関数		F-26	110	
ATAN関数		F-27	111	
ビット処理命令	ビット反転		F-43	128
	ON時微分		F-44	129
	OFF時微分		F-45	130
	セットコイル		F-32	117
	リセットコイル		F-33	118
	ビット抽出	ビット抽出	間接指定	F-130
直接指定			F-131	201
ビットセット/リセット		間接指定	F-132	202
		直接指定	F-133	203

分		類	命令語	参照ページ	
タイマ/カウンタ命令	BCDアップ ダウンカウンタ	2桁	F-62	9・155	
		4桁			
		8桁			
	BCD加算 カウンタ	2桁	F-65	162	
		4桁			
	BCD減算 カウンタ	2桁	F-66	164	
		4桁			
	バイナリ 加算カウンタ	+ 1	1バイト	F-63	158
			1ワード		
		+ 2	1バイト	F-163	226
1ワード					
+ 4	1バイト	F-263	290		
	1ワード				
バイナリ 減算カウンタ	- 1	1バイト	F-64	160	
		1ワード			
	- 2	1バイト	F-164	228	
		1ワード			
- 4	1バイト	F-264	292		
	1ワード				
拡張 タイマ	減算タイマ (設定値、レジスタ指定)		F-260	286	
	減算タイマ (定数、レジスタ指定)		Fc260	287	
拡張 カウンタ	減算カウンタ (設定値、レジスタ指定)		F-261	288	
	減算カウンタ (定数、レジスタ指定)		Fc261	289	
シフト命令	両方向シフト レジスタ	8ビット	F-60	146	
		16ビット			
		32ビット			
	nビット (レジスタ指定)	nビット (レジスタ指定)	F-160	220	
		nビット (定数指定)	Fc160	221	
	非同期両方向 シフトレジスタ	1バイト	F-61	151	
		1ワード			
		2ワード	F-161	222	
		nワード			
	桁シフト(上位シフト)		F-67	166	
桁シフト(下位シフト)		F-68	167		

分 類		命 令 語	参 照 ペ ージ	
演 算 条 件 命 令	マスターコントロールのセット	F-30	9・114	
	マスターコントロールのリセット	F-31	114	
	マスターコントロール ネスティングリセット	F-231	282	
	ジャンプコントロールのセット	F-41	126	
	ジャンプコントロールのリセット	F-42	126	
	ジャンプコントロール ネスティングリセット	F-242	283	
	レベル演算条件のセット	F-47	131	
	レベル演算条件のリセット	F-48	131	
	エンド	無条件エンド	F-40	125
		条件エンド	F-49	132
分 岐 命 令	ラベル	F-140	204	
	ジャンプ	直接指定	F-141	205
		間接指定	F-151	215
	サブルーチン コール	直接指定	F-142	207
		間接指定	F-148	213
	サブルーチン からのリターン	無条件リターン	F-143	207
		条件リターン	F-149	214
	ループ回数の 設定	直接指定	F-144	209
		間接指定	F-146	211
	ループの終了	F-145	209	
ループの強制終了	F-147	212		

分 類		命 令 語	参 照 ペ ージ	
時 計 命 令	時計現在値との比較 (指定リレーのセット)	F-34	9・120	
	時計現在値との比較 (指定リレーのリセット)	F-35	121	
	時間の加算	F-36	122	
	時間の減算	F-37	123	
	時計現在値の転送	F-38	124	
通 信 命 令	オープンチャンネル命令	F-202	247	
		F-203		
	オープンチャンネル1 (階層通信設定)	F-206	250	
	オープンチャンネル2 (階層通信設定)	F-207	250	
	送信命令	F-204	248	
受信命令	F-205	249		
そ の 他 の 命 令	メンテナンスディスプレイ (MD)	F-20	104	
	平方根	F-21	105	
	データのサムチェック コードの生成	F-77	181	
	データのチェック	F-78	182	
	I/Oリフレッシュ	F-80	185	
	特殊I/O ユニット	リフレッシュ	F-82	186
		読出	F-85	187
		書込	F-86	187
	間接アドレス の設定	ファイル0のみ	F-100	191
		ファイル0～3	F-101	192
リマーク (コメント識別用命令)	F-90 (REM)	188		

[3] 命令語処理時間

命令語	処理時間 (μ s)			
	JW-31CUH JW-32CUH JW-33CUH		JW-31CUH1 JW-32CUH1 JW-33CUH1/2/3	
	1 語命令(*1)	2 語命令(*2)	1 語命令(*1)	2 語命令(*2)
STR	0.05	0.10	0.038	0.076
STR NOT	0.05	0.10	0.038	0.076
AND	0.05	0.10	0.038	0.076
AND NOT	0.05	0.10	0.038	0.076
OR	0.05	0.10	0.038	0.076
OR NOT	0.05	0.10	0.038	0.076
AND STR	0.05	-	0.038	-
OR STR	0.05	-	0.038	-
OUT	0.10	0.15	0.076	0.114
NOP	0.05	-	0.038	-

(*1) 1 語命令はリレー番号00000 ~ 15777、TMR/CNT接点番号0000 ~ 0777を使用する場合

(*2) 2 語命令はリレー番号20000 ~ 75777、TMR/CNT接点番号1000 ~ 1777を使用する場合

命令語	処理時間 (μ s)	
	JW-31CUH JW-32CUH JW-33CUH	JW-31CUH1 JW-32CUH1 JW-33CUH1/2/3
TMR	3.55	2.70
DTMR(BCD)	3.60	2.74
DTMR(BIN)	3.60	2.74
UTMR(BCD)	3.60	2.74
UTMR(BIN)	3.60	2.74
CNT	4.35	3.31
DCNT(BCD)	4.70	3.57
DCNT(BIN)	4.70	3.57
UCNT(BCD)	4.70	3.57
UCNT(BIN)	4.70	3.57
MD	1.60	1.22

命令語	処理時間 (μ s)			
	JW-31CUH JW-32CUH JW-33CUH		JW-31CUH1 JW-32CUH1 JW-33CUH1/2/3	
	実行時	非実行時	実行時	非実行時
F-00	1.60	0.60	1.22	0.46
F-00w	1.95	0.60	1.48	0.46
F-00d	2.50	0.60	1.90	0.46
F-01	1.40	0.60	1.06	0.46
F-01w	1.60	0.60	1.22	0.46
F-02	1.90	0.60	1.44	0.46
F-02w	2.50	0.60	1.90	0.46
F-02d	3.55	0.60	2.70	0.46
F-03	3.05	1.20	2.32	0.91
F-03w	4.25	1.20	3.23	0.91
F-04	2.50	0.60	1.90	0.46
F-04w	5.45	0.60	4.14	0.46
F-05	2.00	0.60	1.52	0.46
F-05w	2.30	0.60	1.75	0.46
F-06	2.00	0.60	1.52	0.46
F-06w	2.30	0.60	1.75	0.46
F-07	1.40	0.60	1.06	0.46
F-07w	1.60	0.60	1.22	0.46
F-08	1.40	0.60	1.06	0.46
F-08w	1.60	0.60	1.22	0.46
F-09	1.70	0.60	1.29	0.46
F-09w	2.10	0.60	1.60	0.46
F-09d	2.75	0.60	2.09	0.46

命令語	処理時間(μs)			
	JW-31CUH JW-32CUH JW-33CUH		JW-31CUH1 JW-32CUH1 JW-33CUH1/2/3	
	実行時	非実行時	実行時	非実行時
F-10	4.95	1.20	3.76	0.91
F-10w	7.20	1.20	5.47	0.91
F-10d	16.85	1.20	12.81	0.91
Fc10	4.70	1.20	3.57	0.91
Fc10w	6.55	1.20	4.98	0.91
Fc10d	15.40	1.20	11.70	0.91
F-11	4.85	1.20	3.69	0.91
F-11w	6.95	1.20	5.28	0.91
F-11d	15.20	1.20	11.55	0.91
Fc11	4.60	1.20	3.50	0.91
Fc11w	6.30	1.20	4.79	0.91
Fc11d	13.90	1.20	10.56	0.91
F-12	2.60	1.05	1.98	0.80
F-12w	3.35	1.05	2.55	0.80
F-12d	4.15	1.05	3.15	0.80
Fc12	2.50	1.05	1.90	0.80
Fc12w	3.15	1.05	2.39	0.80
Fx12	2.50	1.05	1.90	0.80
Fx12w	3.15	1.05	2.39	0.80
F-13	2.10	0.60	1.60	0.46
F-13w	2.75	0.60	2.09	0.46
F-13d	3.70	0.60	2.81	0.46
Fc13	2.00	0.60	1.52	0.46
Fc13w	2.55	0.60	1.94	0.46
Fx13	2.00	0.60	1.52	0.46
Fx13w	2.55	0.60	1.94	0.46
F-14	2.10	0.60	1.60	0.46
F-14w	2.75	0.60	2.09	0.46
F-14d	3.70	0.60	2.81	0.46
Fc14	2.00	0.60	1.52	0.46
Fc14w	2.55	0.60	1.94	0.46
Fx14	2.00	0.60	1.52	0.46
Fx14w	2.55	0.60	1.94	0.46
F-15	26.60	1.20	20.22	0.91
F-15d	101.30	1.20	76.99	0.91
Fc15	25.95	1.20	19.72	0.91
Fc15d	99.85	1.20	75.87	0.91
F-16	10.85	1.20	8.25	0.91
F-16d	51.95	1.20	39.48	0.91
Fc16	10.60	1.20	8.06	0.91
Fc16d	49.45	1.20	37.58	0.91
F-17	2.15	0.60	1.63	0.46
F-17w	2.85	0.60	2.17	0.46
F-17d	3.90	0.60	2.96	0.46
Fc17	2.05	0.60	1.56	0.46
Fc17w	2.65	0.60	2.01	0.46
Fx17	2.05	0.60	1.56	0.46
Fx17w	2.65	0.60	2.01	0.46
F-18	2.10	0.60	1.60	0.46
F-18w	2.75	0.60	2.09	0.46
F-18d	3.70	0.60	2.81	0.46
Fc18	2.00	0.60	1.52	0.46
Fc18w	2.55	0.60	1.94	0.46
Fx18	2.00	0.60	1.52	0.46
Fx18w	2.55	0.60	1.94	0.46
F-20	1.60	-	1.22	-
F-21	43.47	1.20	33.04	0.91
F-22	58.00	1.20	44.08	0.91
F-23	58.00	1.20	44.08	0.91
F-24	76.00	1.20	57.76	0.91

JW-31CUH/H1ではプログラムできません。

命令語	処理時間(μ s)			
	JW-31CUH JW-32CUH JW-33CUH		JW-31CUH1 JW-32CUH1 JW-33CUH1/2/3	
	実行時	非実行時	実行時	非実行時
F-25	98.00	1.20	74.48	0.91
F-26	109.00	1.20	82.84	0.91
F-27	44.50	1.20	33.82	0.91
F-28	146.25	1.20	111.15	0.91
F-29	229.75	1.20	174.61	0.91
F-30	1.95	-	1.48	-
F-31	1.20	-	0.91	-
F-32	0.20	-	0.15	-
F-33	0.20	-	0.15	-
F-34	2.25	0.45	1.71	0.34
F-35	2.25	0.45	1.71	0.34
F-36	9.70	1.20	7.37	0.91
F-37	10.15	1.20	7.71	0.91
F-38	1.95	0.60	1.48	0.46
F-40	1.25	-	0.95	-
F-41	2.15	-	1.63	-
F-42	1.30	-	0.99	-
F-43	0.60	-	0.46	-
F-44	0.80	-	0.61	-
F-45	0.90	-	0.68	-
F-47	0.50	-	0.38	-
F-48	0.50	-	0.38	-
F-49	1.55	1.00	1.18	0.76
F-50	1.95	0.60	1.48	0.46
F-51	2.30	0.60	1.75	0.46
F-52	2.00	0.60	1.52	0.46
F-53	4.25	1.20	3.23	0.91
F-54	5.45	0.60	4.14	0.46
F-55	1.80	0.60	1.37	0.46
F-56	3.70	1.20	2.81	0.91
F-56w	5.40	1.20	4.10	0.91
F-56d	13.00	1.20	9.88	0.91
F-57	1.75	0.60	1.33	0.46
F-57w	2.30	0.60	1.75	0.46
F-57d	3.10	0.60	2.36	0.46
F-58	1.80+1.20B (B=1~8) (*1)	0.60	1.37+0.91B (B=1~8) (*1)	0.46
F-60	3.97	3.02	3.02	2.30
F-60w	4.62	3.02	3.51	2.30
F-60d	5.62	3.02	4.27	2.30
F-61	1.92	1.20	1.46	0.91
F-61w	2.37	1.20	1.80	0.91
F-61d	5.37	1.20	4.08	0.91
F-62	4.40	1.95	3.34	1.48
F-62w	5.55	1.95	4.22	1.48
F-62d	7.75	1.95	5.89	1.48
F-63	2.15	1.20	1.63	0.91
F-63w	2.60	1.20	1.98	0.91
F-64	2.15	1.20	1.63	0.91
F-64w	2.60	1.20	1.98	0.91
F-65	3.05	1.20	2.32	0.91
F-65w	4.10	1.20	3.12	0.91
F-66	3.45	1.20	2.62	0.91
F-66w	4.75	1.20	3.61	0.91
F-67	2.68+0.80B (B=1~256) (*1)	0.60	2.04+0.61B (B=1~256) (*1)	0.46
F-68	2.78+0.80B (B=1~256) (*1)	0.60	2.11+0.61B (B=1~256) (*1)	0.46
F-69	1.75	0.60	1.33	0.46
F-70	4.03+0.35B (B=1~256) (*1)	0.60	3.06+0.27B (B=1~256) (*1)	0.46
F-70w	4.03+0.70W (W=1~256) (*1)	0.60	3.06+0.57W (W=1~256) (*1)	0.46
F-71	6.23+0.15B (B=1~1024)(*1)	0.60	4.74+0.11B (B=1~1024)(*1)	0.46
F-71w	6.43+0.30W (W=1~512) (*1)	0.60	4.89+0.22W (W=1~512) (*1)	0.46

JW-31CUH/H1ではプログラムできません。

(*1) B : バイト数 W : ワード数

命令語	処理時間(μs)			
	JW-31CUH JW-32CUH JW-33CUH		JW-31CUH1 JW-32CUH1 JW-33CUH1/2/3	
	実行時	非実行時	実行時	非実行時
F-72	3.70+0.55B (B=1~256) (*1)	0.60	2.81+0.42B (B=1~256) (*1)	0.46
F-72w	3.70+1.10W (W=1~256) (*1)	0.60	2.81+0.84W (W=1~256) (*1)	0.46
F-73	3.30+0.55B (B=1~256) (*1)	0.60	2.51+0.42B (B=1~256) (*1)	0.46
F-73w	3.30+1.10W (W=1~256) (*1)	0.60	2.51+0.84W (W=1~256) (*1)	0.46
F-74	3.68+0.15B (B=1~256) (*1)	0.60	2.80+0.11B (B=1~256) (*1)	0.46
F-74w	3.98+0.30W (W=1~256) (*1)	0.60	3.03+0.23W (W=1~256) (*1)	0.46
F-76	4.18+0.35B (B=1~256) (*1)	0.60	3.18+0.27B (B=1~256) (*1)	0.46
F-76w	4.18+0.70W (W=1~256) (*1)	0.60	3.18+0.53W (W=1~256) (*1)	0.46
F-77	2.11+0.45B (B=1~256) (*1)	0.60	1.60+0.34B (B=1~256) (*1)	0.46
F-78	2.83+0.45B (B=1~256) (*1)	1.20	2.15+0.34B (B=1~256) (*1)	0.91
F-79	10バイト…68 20バイト…167 50バイト…828 100バイト…3165 200バイト…12500 256バイト…20300	0.60	10バイト…52 20バイト…127 50バイト…629 100バイト…2405 200バイト…9500 256バイト…15428	0.46
F-79w	10ワード…90 20ワード…233 50ワード…1192 100ワード…4560 200ワード…17000 256ワード…28900	0.60	10ワード…68 20ワード…177 50ワード…906 100ワード…3466 200ワード…12920 256ワード…21964	0.46
F-80	27.95	1.05	21.24	0.80
F-82	特殊I/Oの状態による	1.05	特殊I/Oの状態による	0.80
F-85	特殊I/Oの状態による	1.20	特殊I/Oの状態による	0.91
F-86	特殊I/Oの状態による	1.20	特殊I/Oの状態による	0.91
F-90	0.10	-	0.08	-
F-91	1.90	0.60	1.44	0.46
F-97	4.05	0.60	3.08	0.46
F-100	2.28	0.60	1.73	0.46
F-101	2.05	0.60	1.56	0.46
F-102	2.40	0.60	1.82	0.46
F-102w	2.80	0.60	2.13	0.46
F-103	2.70	0.60	2.05	0.46
F-103w	3.10	0.60	2.36	0.46
F-112	2.38+1.25B (B=1~256) (*1)	1.05	1.81+0.95B (B=1~256) (*1)	0.80
F-112w	2.38+2.50W (W=1~256) (*1)	1.05	1.81+1.90W (W=1~256) (*1)	0.80
F-116	35.85	1.20	27.25	0.91
F-130	2.35	1.05	1.79	0.80
F-131	2.25	1.05	1.71	0.80
F-132	2.10	1.05	1.60	0.80
F-133	2.00	1.05	1.52	0.80
F-140	1.85	-	1.41	-
F-141	1.95	1.00	1.48	0.76
F-142	3.30	1.15	2.51	0.87
F-143	2.40	-	1.82	-
F-144	3.05	3.05	2.32	2.32
F-145	3.05	3.05	2.32	2.32
F-146	3.15	3.15	2.39	2.32
F-147	1.05	1.05	0.80	0.80
F-148	4.00	1.15	3.04	0.87
F-149	2.30	1.05	1.75	0.80
F-151	2.55	1.00	1.94	0.76
F-153	6.35	1.20	4.83	0.91
F-154	23.00	0.60	17.48	0.46
F-155	26.45	1.20	20.10	0.91
F-156	13.25	1.20	10.07	0.91
F-160	32.05	5.22	24.36	3.97
Fc160	31.65	4.82	24.05	3.66
F-161	3.45+0.95B (B=1~256) (*1) (*2)	1.20	2.62+0.72B (B=1~256) (*1) (*2)	0.91

JW-31CUH/H1ではプログラムできません。

(*1) B : バイト数 W : ワード数 (*2) シフト領域全データ 00 時

命令語	処理時間 (μs)			
	JW-31CUH JW-32CUH JW-33CUH		JW-31CUH1 JW-32CUH1 JW-33CUH1/2/3	
	実行時	非実行時	実行時	非実行時
F-161w	4.21+1.40W (W=1~256)(*1)(*2)	1.20	3.13+1.06W (W=1~256)(*1)(*2)	0.91
F-163	2.20	1.20	1.67	0.91
F-163w	2.65	1.20	2.01	0.91
F-164	2.15	1.20	1.63	0.91
F-164w	2.60	1.20	1.98	0.91
F-170	2.28+0.55B (B=1~256) (*1)	1.20	1.73+0.42B (B=1~256) (*1)	0.91
F-170w	2.58+1.10W (W=1~256) (*1)	1.20	1.96+0.84W (W=1~256) (*1)	0.91
F-171	2.28+0.55B (B=1~256) (*1)	1.20	1.73+0.42B (B=1~256) (*1)	0.91
F-171w	2.58+1.10W (W=1~256) (*1)	1.20	1.96+0.84W (W=1~256) (*1)	0.91
F-172	3.33+0.60B (B=1~256) (*1)	1.20	2.53+0.46B (B=1~256) (*1)	0.91
F-172w	3.83+0.85W (W=1~256) (*1)	1.20	2.91+0.65W (W=1~256) (*1)	0.91
F-173	3.43+1.35B (B=1~256) (*1)	1.20	2.61+1.03B (B=1~256) (*1)	0.91
F-173w	4.08+1.90W (W=1~256) (*1)	1.20	3.10+1.44W (W=1~256) (*1)	0.91
F-174	2.23+0.45B (B=1~1024) (*1)	0.60	1.69+0.34B (B=1~1024)(*1)	0.46
F-175	2.13+0.75B (B=1~1024) (*1)	0.60	1.62+0.57B (B=1~1024)(*1)	0.46
F-176	143.65	0.60	109.17	0.46
F-177	144.65	0.60	109.93	0.46
F-180	3.40	2.20	2.58	1.67
F-180w	4.05	2.20	3.08	1.67
Fc180	3.30	2.20	2.51	1.67
Fc180w	3.85	2.20	2.93	1.67
F-181	3.40	2.20	2.58	1.67
F-181w	4.05	2.20	3.08	1.67
Fc181	3.30	2.20	2.51	1.67
Fc181w	3.85	2.20	2.93	1.67
F-182	3.40	2.20	2.58	1.67
F-182w	4.05	2.20	3.08	1.67
Fc182	3.30	2.20	2.51	1.67
Fc182w	3.85	2.20	2.93	1.67
F-183	3.40	2.20	2.58	1.67
F-183w	4.05	2.20	3.08	1.67
Fc183	3.30	2.20	2.51	1.67
Fc183w	3.85	2.20	2.93	1.67
F-184	3.40	2.20	2.58	1.67
F-184w	4.05	2.20	3.08	1.67
Fc184	3.30	2.20	2.51	1.67
Fc184w	3.85	2.20	2.93	1.67
F-185	3.40	2.20	2.58	1.67
F-185w	4.05	2.20	3.08	1.67
Fc185	3.30	2.20	2.51	1.67
Fc185w	3.85	2.20	2.93	1.67
F-202	2.70	0.85	2.05	0.65
F-203	2.70	0.85	2.05	0.65
F-204	3.60	1.20	2.74	0.91
F-205	3.60	1.20	2.74	0.65
F-206	2.70	0.85	2.05	0.65
F-207	2.70	0.85	2.05	0.91
F-210	3.15	1.20	2.39	0.91
F-210w	3.95	1.20	3.00	0.91
F-210d	5.05	1.20	3.84	0.91
Fc210	3.05	1.20	2.32	0.91
Fc210w	3.70	1.20	2.81	0.91
Fc210d	4.40	1.20	3.34	0.91
F-211	3.00	1.20	2.28	0.91
F-211w	3.80	1.20	2.89	0.91
F-211d	5.00	1.20	3.80	0.91
Fc211	2.90	1.20	2.20	0.91
Fc211w	3.55	1.20	2.70	0.91
Fc211d	4.35	1.20	3.31	0.91
F-212	3.15	1.05	2.39	0.80

JW-31CUH/H1ではプログラムできません。

(*1) B : バイト数 W : ワード数 (*2) シフト領域全データ 00時

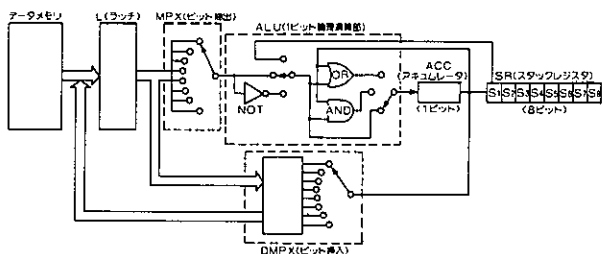
命令語	処理時間(μs)			
	JW-31CUH JW-32CUH JW-33CUH		JW-31CUH1 JW-32CUH1 JW-33CUH1/2/3	
	実行時	非実行時	実行時	非実行時
F-212w	4.15	1.05	3.15	0.80
F-212d	5.25	1.05	3.99	0.80
Fc212	2.95	1.05	2.20	0.80
Fc212w	3.20	1.05	2.43	0.80
Fx212	2.95	1.05	2.24	0.80
Fx212w	3.20	1.05	2.43	0.80
F-215	3.10	1.20	2.36	0.91
F-215w	4.05	1.20	3.08	0.91
F-215d	6.30	1.20	4.79	0.91
Fc215	3.00	1.20	2.28	0.91
Fc215w	3.80	1.20	2.89	0.91
Fc215d	5.65	1.20	4.29	0.91
F-216	4.05	1.20	3.08	0.91
F-216w	5.10	1.20	3.88	0.91
F-216d	6.60	1.20	5.02	0.91
Fc216	3.95	1.20	3.00	0.91
Fc216w	4.85	1.20	3.69	0.91
Fc216d	5.95	1.20	4.52	0.91
F-231	1.55	-	1.18	-
F-242	1.75	-	1.33	-
F-252	1.60+1.10B (B=1~1024) (*1)	0.60	1.22+0.84B (B=1~1024) (*1)	0.46
F-253	2.48+1.34B (B=1~1024) (*1)	1.20	1.89+1.01B (B=1~1024) (*1)	0.91
F-260	3.90	3.40	2.96	2.58
Fc260	3.90	3.40	2.96	2.58
F-261	4.00	3.50	3.04	2.66
Fc261	4.00	3.50	3.04	2.66
F-263	2.20	1.20	1.67	0.91
F-263w	2.65	1.20	2.01	0.91
F-264	2.15	1.20	1.63	0.91
F-264w	2.60	1.20	1.98	0.91
F-310	4.65	1.20	3.53	0.91
F-311	4.65	1.20	3.53	0.91
F-315	6.45	1.20	4.90	0.91
F-316	6.95	1.20	5.28	0.91

(*1) B : バイト数

第 8 章 基本命令の説明

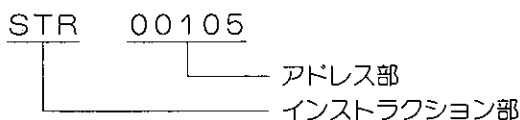
8-1 ビット処理部の動作

ビット処理とは、接点信号の論理演算のことで JW30H のビット処理部の概略ブロック図を示します。



(1) L(ラッチ)

ビット処理命令は、インストラクション部とアドレス部で構成します。

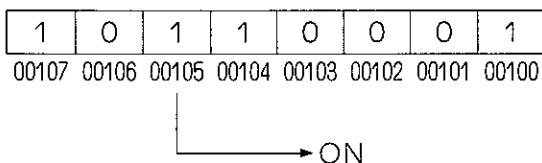


アドレス部はデータメモリのリレー領域(入出力リレー、補助リレー、特殊リレー等)のリレー番号を表わします。データメモリからリレーのON/OFF 情報を読み出す場合、そのリレー番号を含む1バイト(8ビット)の内容をまとめてL(ラッチ)に読み出します。

STR00105の場合、00100~00107の8ビットを読み出します。

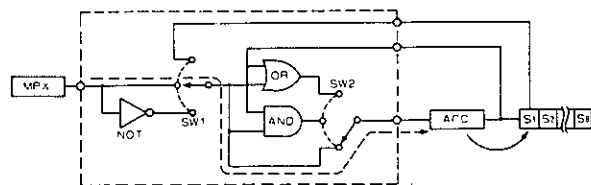
(2) MPX(マルチプレクサ)

L(ラッチ)に読み出した8ビットのうち、必要な1ビットを抽出します。STR00105の場合00100~00107から00105のON/OFF 情報を抽出します。



(3) ALU(1ビット論理演算部)

命令のインストラクション部の内容に従い論理演算を行います。



上図はSTR命令の場合の演算状態を示します。インストラクション部の内容により、SW1、SW2を切換えます。

(4) ACC(アキュムレータ)

ALUの演算結果を格納する1ビットのレジスタです。

(5) SR(スタックレジスタ)

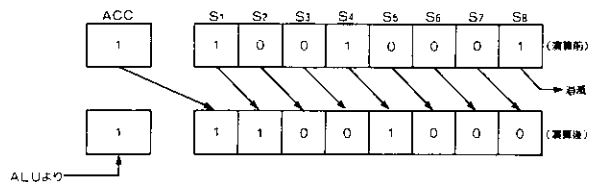
直並列回路の演算や、複数の入力条件をもつ応用命令の演算時に、演算の中間結果を記憶する8ビットのレジスタです。



① STR、STR NOT 命令実行時のSRの動き

- データメモリから読み出した1ビットのON/OFF情報がACCに入ります。(STR NOTでは反転後ACCに入ります。)

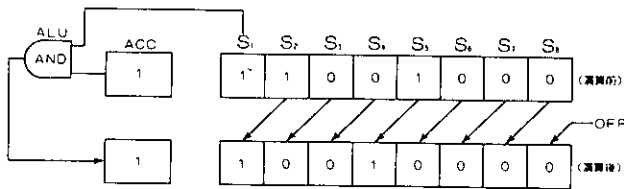
- それ以前にACCに入っていたON/OFF情報はS1に、S1の情報はS2に、以後S2→S3、S3→S4、S4→S5、S5→S6、S6→S7、S7→S8とシフトし、S8に入っていた情報は消滅します。



② AND STR、OR STR 命令実行時のSRの動き

- S1のON/OFF情報がALUに入り、ACCの内容との間でAND又はORの演算を行い、演算結果をACCに格納します。

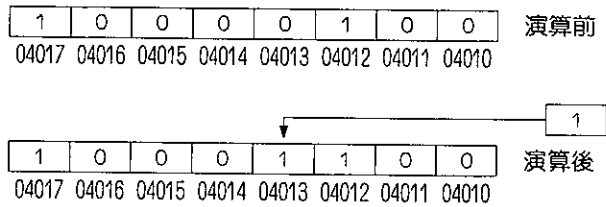
- 演算後不要となったS1のON/OFF情報は消滅し、S1にはS2の情報が、S2にはS3が、以後S3→S4、S4→S5、S5→S6、S6→S7、S7→S8とシフトし、S8にはOFFの情報が入ります。



(6) DMPX(デマルチプレクサ)

OUT命令では、L(ラッチ)に読み出した8ビットのうち、命令のアドレス部で示す1ビットを、演算結果(ACCの内容)に書き換え、データメモリに1バイト分転送します。

(OUT 04013で、演算結果がONの場合)



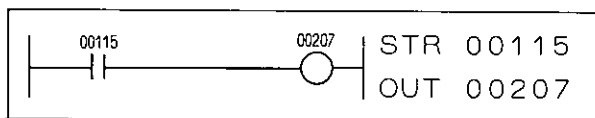
ビット処理部の動作は下記を参照してください。

8-2 各基本命令について

(1) STR/OUT

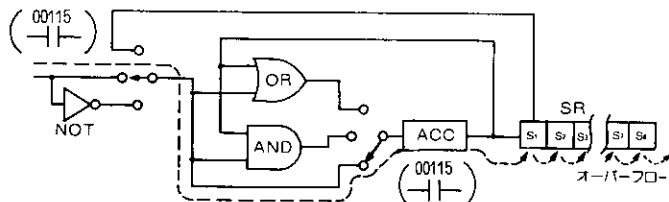
STR 指定したデータメモリの内容(ON/OFF状態)をアキュムレータ(ACC)に格納します。また、以前の内容をスタックレジスタ(SR)のS1にシフトします。

OUT アキュムレータ(ACC)の内容を指定したデータメモリへ転送します。



STR 00115

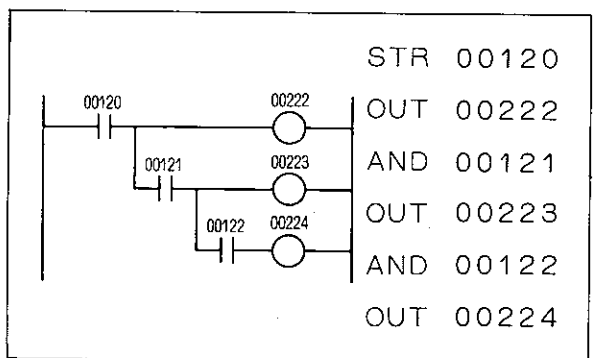
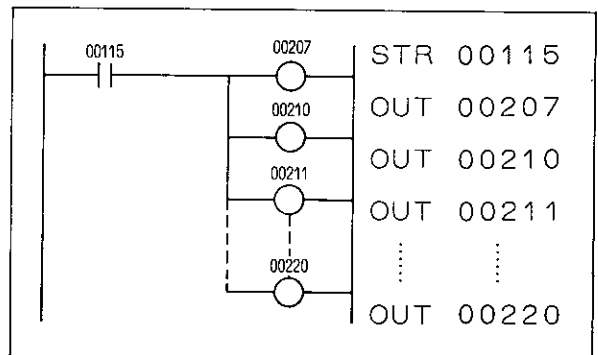
- L(ラッチ)…データメモリから(00110)~(00117) 8ビット読み出します。
- MPX ……L(ラッチ)内の8ビットから(00115) 1ビットを抽出します。
- ALU、ACC、SR…MPXの出力をそのままACCに書き込みます。また、以前の内容はSRのS1にシフトします。



OUT 00207

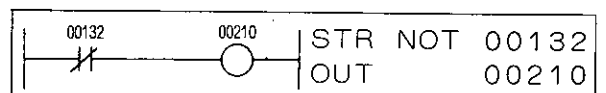
- L(ラッチ)…データメモリから(00200)~(00207) 8ビット読み出します。
- MPX ……OUT命令では関与しません。
- ALU、ACC、SR…ACC、SRの内容は不変です。
- DMPX ……L(ラッチ)内の8ビットの内(00207)の1ビットをACCの内容に書き換え(00200)~(00207)の8ビットをデータメモリに転送します。

参考 OUT命令の演算後もACCの内容は変化しないため、次のようなプログラムも有効です。



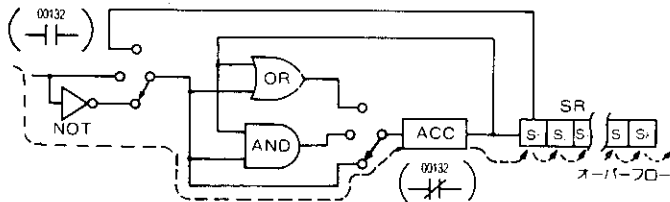
(2) STR NOT

- 指定したデータメモリの内容を反転してACCに格納します。また、以前の内容をSRのS1にシフトします。



STR -NOT 00132

- L(ラッチ)…データメモリから(00130)~(00137) 8ビットを読み出します。
- MPX………L(ラッチ)内の8ビットから(00132) 1ビットを抽出します。
- ALU、ACC、SR…MPXの出力を反転してACCに書き込みます。
また、以前の内容はSRのS1にシフトします。

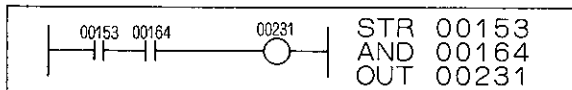


OUT 00210

データメモリの00210は 00132 の演算結果に書き換わります。

(3) AND

- 指定したデータメモリの内容とACCの内容をAND演算してその結果をACCに格納します。

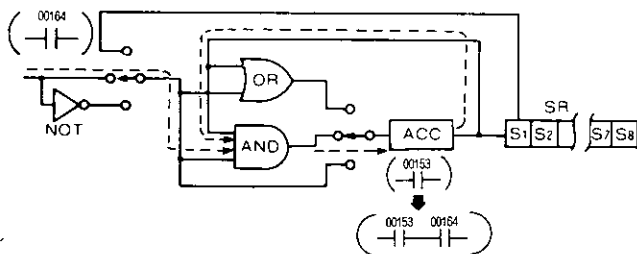


STR 00153

ACCにデータメモリの00153の内容を記憶します。

AND 00164

- L(ラッチ)…データメモリから00160~00167の8ビットを読み出します。
- MPX………L(ラッチ)内の8ビットから00164の1ビットを抽出します。
- ALU、ACC、SR…ACCの内容(00153)とMPXの出力(00164)のANDを演算し、ACCに書き込みます。SRの内容は保持します。



OUT 00231

データメモリの00231は 00153 00164 の演算結果に書き換わります。

(4) AND NOT

- 指定したデータメモリの内容を反転し、ACCの内容とAND演算し、その結果をACCに格納します。

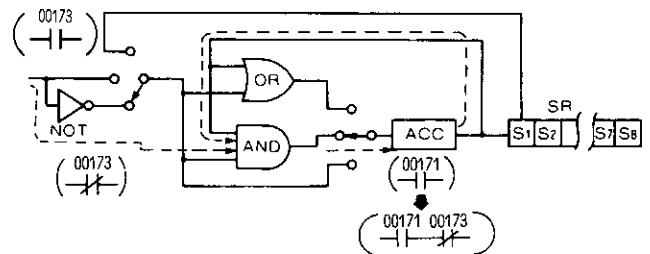


STR 00171

ACCにデータメモリの00171の内容を記憶します。

AND NOT 00173

- L(ラッチ)…データメモリから00170~00177の8ビットを読み出します。
- MPX………L(ラッチ)内の8ビットから00173の1ビットを抽出します。
- ALU、ACC、SR…ACCの内容(00171)と、MPXの出力(00173)の反転したものをAND演算しACCに書き込みます。SRの内容は保持します。

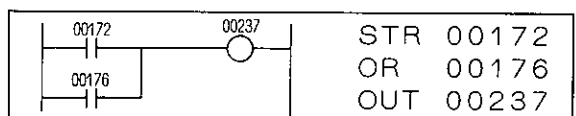


OUT 00235

データメモリの00235は 00171 00173 の演算結果に書き換わります。

(5) OR

- 指定したデータメモリの内容ACCの内容をOR演算し、その結果をACCに格納します。

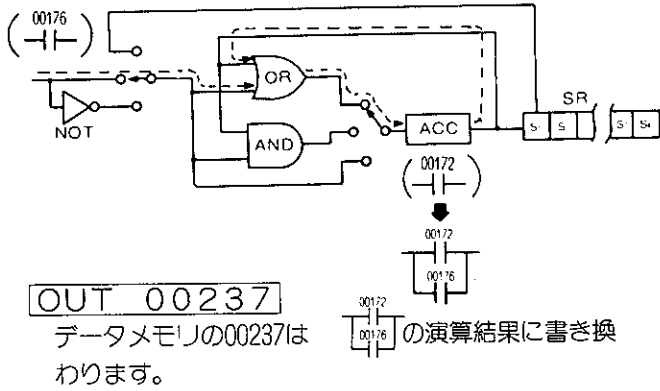


STR 00172

ACCにデータメモリの00172の内容を記憶します。

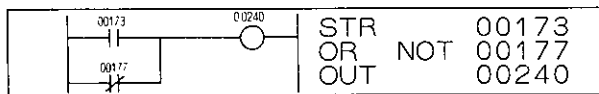
OR 00176

- L(ラッチ)…データメモリから00170~00177の8ビットを読み出します。
- MPX………L(ラッチ)内の8ビットから00176の1ビットを抽出します。
- ALU、ACC、SR…ACCの内容(00172)とMPXの出力(00176)のORを演算し、ACCに書き込みます。SRの内容は保持します。



(6) OR NOT

- 指定したデータメモリの内容を反転し、ACCの内容とOR演算し、その結果をACCに格納します。

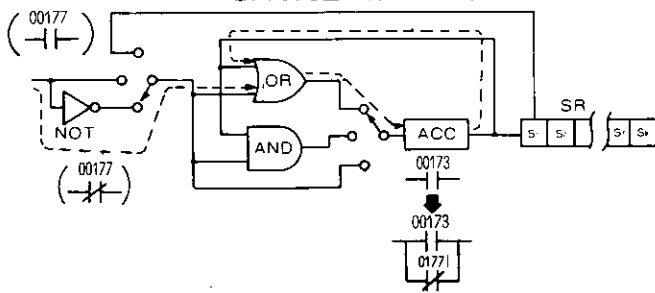


STR 00173

ACCにデータメモリの00173の内容を記憶します。

OR NOT 00177

- L(ラッチ)…データメモリから00170~00177の8ビットを読み出します。
- MPX…L(ラッチ)内の8ビットから00177の1ビットを抽出します。
- ALU、ACC、SR…ACCの内容(00173)とMPXの出力(00177)を反転したものをOR演算しACCに書き込みます。SRの内容は保持します。

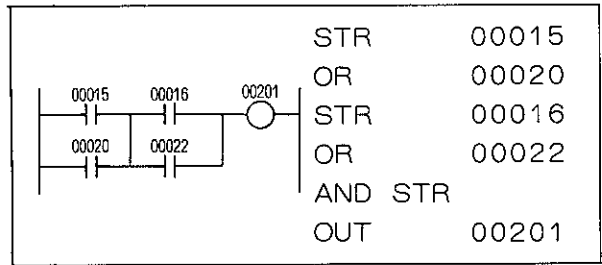


OUT 00240

データメモリの00240は
の演算結果に書き換
わります。

(7) AND STR

- スタックレジスタ(SR)のS1の内容とACCの内容をAND演算し、その結果をACCに格納します。



STR 00015

ACCにデータメモリの00015の内容を記憶します。

OR 00020

ACCには
の演算結果を記憶します。

STR 00016

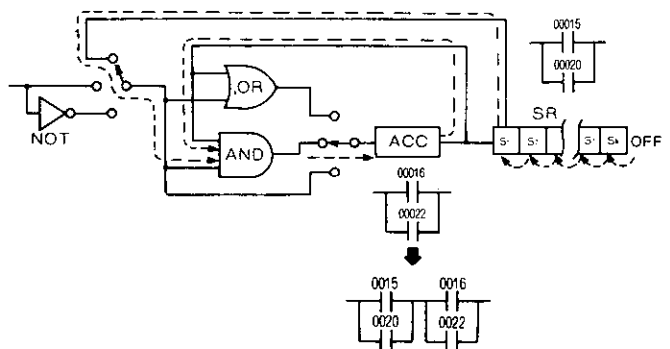
ACCに入っているそれ迄の演算結果
は、SRのS1に待避し、データメモリ00016の内容をACCに書き込みます。

OR 00022

ACCには
の演算結果を記憶します。

AND STR

- L(ラッチ)…AND STR命令の場合
関与しません。
- MPX…AND STR命令の場合
関与しません。
- ALU、ACC、SR…SRのS1の内容
とACCの内容
のANDを演算し、ACCに書き込みます。

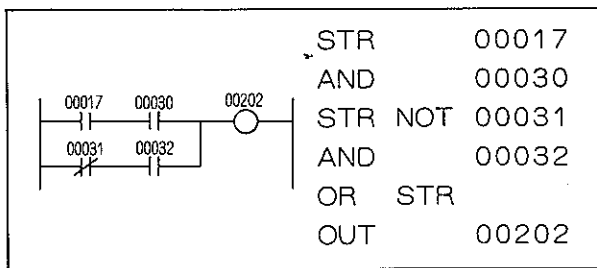


OUT 00201

データメモリの00201は
の演算結果に書
き換わります。

(8) OR STR

- スタックレジスタ (SR) の S1 の内容とACCの内容をOR演算し、その結果をACCに格納します。



STR 00017

ACCにデータメモリの00017の内容を記憶します。

AND 00030

ACCには $\overline{00017} \cdot 00030$ の演算結果を記憶します。

STR NOT 00031

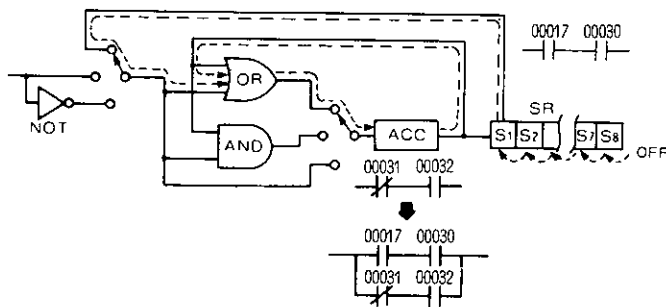
ACCに入っているそれ迄の演算結果 $\overline{00017} \cdot 00030$ はSRのS1に待避し、データメモリ00031の内容を反転してACCに書き込みます。

AND 00032

ACCには $\overline{00031} \cdot 00032$ の演算結果を記憶します。

OR STR

- L(ラッチ)…OR STR命令の場合
関与しません。
- MPX………OR STR命令の場合
関与しません。
- ALU、ACC、SR…SRのS1の内容 $\overline{00017} \cdot 00030$ とACCの内容 $\overline{00031} \cdot 00032$ をOR演算し、ACCに書き込みます。



OUT 00202

データメモリの00202は $\overline{00031} \cdot 00032$ の演算結果に書き換わります。

(9) TMR(タイマ命令)

TMR命令は、0.1秒クロックを内部クロックとし減算式、加算式及び、計数回路をBCD値又はバイナリ値で取り扱う5種類があります。

(1) TMR命令の種類

名称	演算方法	計数值	設定範囲
TMR	減算式	BCD	000~1999
DTMR (BCD)			0000~7999
DTMR (BIN)		バイナリ	00000~32767
UTMR (BCD)	加算式	BCD	0000~7999
UTMR (BIN)		バイナリ	00000~32767

(2) 減算式TMR命令

- スタート入力OFFの間、計数は行わず、現在値=設定値を維持し、TMR接点はOFFです。
- スタート入力ONになると0.1秒ごとに現在値は-1し、現在値が0になるとTMR接点はONし、スタート入力ONの間この状態を保持します。

スタート入力	現在値	TMR接点
OFF	設定値	OFF
ON(現在値>0)	0.1秒ごとに-1	OFF
ON(現在値=0)	0	ON

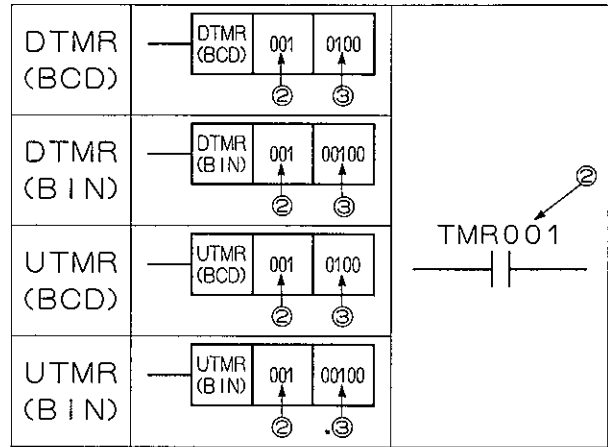
(3) 加算式TMR命令

- スタート入力OFFの間、計数は行わず、現在値=0を維持し、TMR接点はOFFです。
- スタート入力ONになると0.1秒ごとに現在値は+1し、現在値=設定値になるとTMR接点はONし、スタート入力ONの間この状態を保持します。

スタート入力	現在値	TMR接点
OFF	0	OFF
ON(現在値<設定値)	0.1秒ごとに+1	OFF
ON(現在値=設定値)	設定値	ON

(4) シンボルマーク

名称	TMRのシンボル	TMR接点のシンボル
TMR		



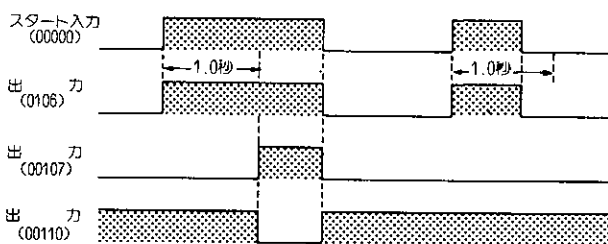
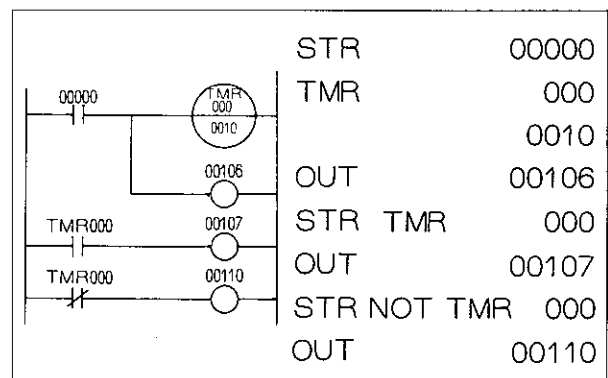
①スタート入力 (ONでスタート)

②TMR番号 0000~1777(oct).....CNT、MDと共通使用

ただし、DTMR、UTMRは000~777oct

③設定値 0.1秒単位
0.01秒単位*

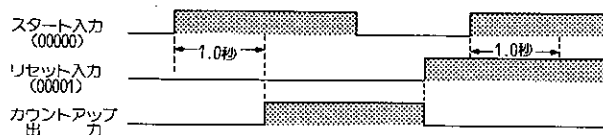
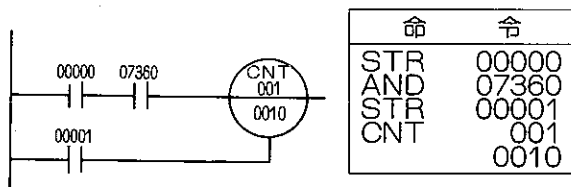
④精度 10msタイマ(設定値 $\pm 0.01s$) + スキャンタイム
100msタイマ(設定値 $\pm 0.1s$) + スキャンタイム



* TMR700~TMR777までを10ms単位のタイマにするときは、システムメモリの#227に設定します。ただしDTMR、UTMRは100msタイマとして働きます。(5・11ページ参照)

- TMR番号は、CNT、MDと共通使用です。CNT、MDに使用した番号は、TMRに使用しないでください。又、同一TMR番号の使用も避けてください。万一、同一番号を使用した場合、プログラマ等のプログラムチェックでエラー表示します。
- TMR接点はTMR番号と同じ番号を指定し、a接点、b接点を何個でも使用できます。
- TMRの現在値は、b0000~b3777の2048バイトに格納します。(2・6ページ参照)

- JW30Hの電源投入時、タイマはリセットします。従って、タイマのスタート入力(00000)がON状態で、JW30Hの電源が入っても、リセット機能が働き、現在値は設定値となります。
- タイマ命令はシステムメモリ(#201)にタイマリセット条件を設定すると停電時の状態を記憶できます。(5・8ページ参照)
- 接点07360(0.1秒クロック)とCNT命令を利用して停電記憶のタイマや、スタート条件とリセット条件の違うタイマを実現できます。



(10) CNT(カウンタ命令)

CNT命令は計数入力の立上りで1回計算する減算式、加算式及び計数回路をBCD値及びバイナリ値で取り扱う5種類があります。

(1) CNT命令の種類

名 称	演算方法	計数値	設定範囲
CNT	減算式	BCD	000~1999
DCNT (BCD)			0000~7999
DCNT (BIN)		バイナリ	00000~32767
UCNT (BCD)	加算式	BCD	0000~7999
UCNT (BIN)			バイナリ

(2) 減算式CNT命令

- リセット入力(00001)がONの間、計数入力(00000)がOFF→ONに変化しても計数は行わず、現在値=設定値を維持し、CNT接点はOFFです。
- リセット入力(00001)がOFFの間、計数入力(00000)がOFF→ONに変化するとに現在値は-1し、現在値が0になるとCNT接点はONし、リセット入力(00001)がOFFの間この状態を保持します。

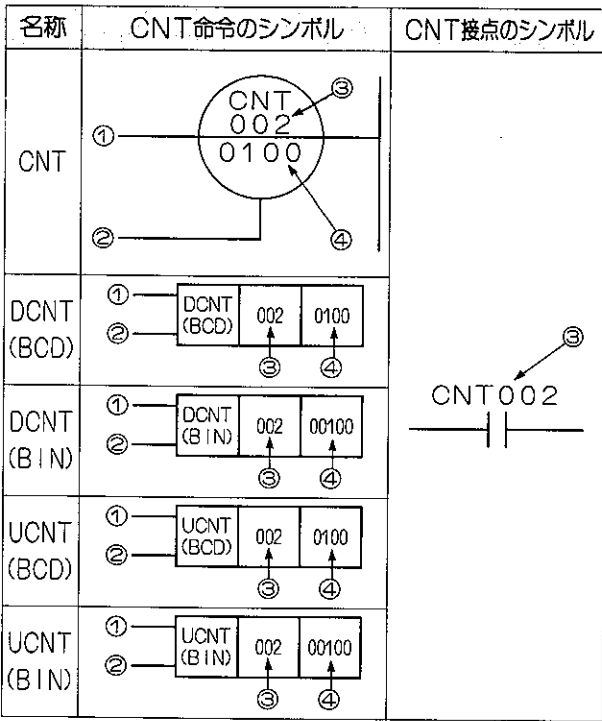
リセット入力	現 在 値	CNT接点
ON	設定値	OFF
OFF(現在値>0)	計数入力がOFF→ONとなることに-1	OFF
OFF(現在値=0)	0	ON

(3) 加算式CNT命令

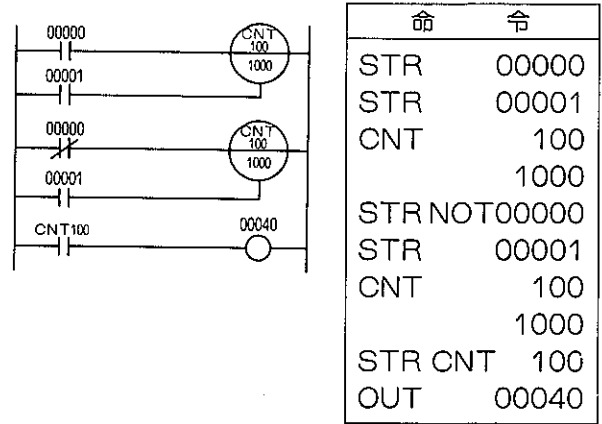
- リセット入力(00001)がONの間、計数入力(00000)がOFF→ONに変化しても計数は行わず、現在値=0を維持しCNT接点はOFFです。
- リセット入力(00001)がOFFの間、計数入力(00000)がOFF→ONに変化するとに現在値は+1し、現在値=設定値になるとCNT接点はONし、リセット入力(00001)がOFFの間この状態を保持します。

リセット入力	現 在 値	CNT接点
ON	0	OFF
OFF(現在値=0)	計数入力がOFF→ONとなることに+1	OFF
OFF(現在値=設定値)	設定値	ON

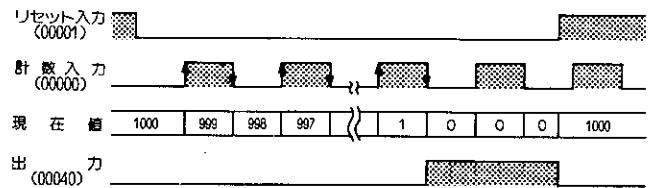
(4) シンボルマーク



(例) 計数入力の立上り、立下りで計数するカウンタ。

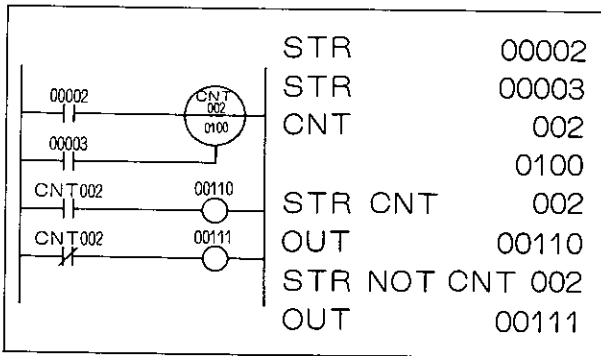


- 計数入力がOFF→ONに変化したとき、ON→OFFに変化したときのいずれの場合も減算するカウンタです。

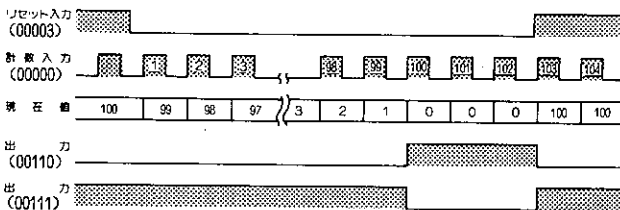


- ① 計 数 入 力 (OFF→ONを検知)
- ② リセット入力 (ONでリセット)
- ③ CNT 番号 0000~1777(oct).....TMR、MDと共通使用
ただし、DCNT、UCNTは000~777oct

④ 設 定 値



- CNT接点は、CNT番号と同じ番号を指定し、a接点、b接点を何個でも使用できます。
- カウントアップすると以後の入力を無視します。再び、計数をはじめるときはリセット入力を一旦ONした後、再びOFFにするか、プログラマ等により、強制リセット後、計数を開始してください。
- 計数入力と、リセット入力が同時ONの場合、リセットを優先します。
- CNTの現在値は、b0000~b3777の2048バイトに格納します。(2・6ページ参照)
- 停電時カウンタは現在値を記憶しています。ただしリセット入力が電源投入時ONとなる場合、現在値をリセットします。停電時にも現在値を記憶する必要がある場合、電源投入時OFFとなるリセット入力を加えてください。
- リセット入力はシステムメモリ(#202)にリセット条件を設定することにより「OFFでリセット」もできます。(5・8ページ参照)

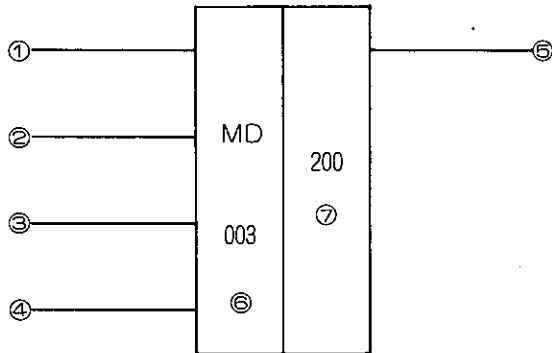


- CNT番号は、TMR、MDと共通使用です。TMR、MDに使用した番号は、CNTに使用しないでください。万一、同一番号を使用した場合、プログラマ等のプログラムチェックでエラー表示します。又、同一CNT番号を使用してもエラー表示しますが意図的に同一番号を使用する場合、この警告は無視してください。

(11) MD(メンテナンスディスプレイ)

MD(メンテナンスディスプレイ)命令は、被制御機器の動作状態の監視情報や、故障発生時の原因究明用情報をプログラマ等の周辺機器に表示したり、外部に出力する命令です。

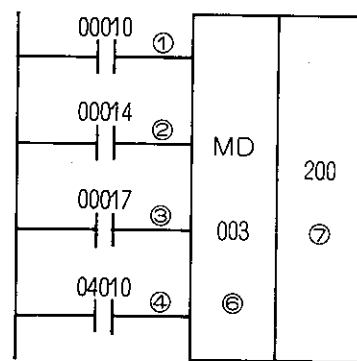
(1) シンボルの説明



① ② ③	入力情報	⑦のMDデータと共に外部に出力する接点情報で00000~15777, 20000~75777の各リレー、TMR・CNTの接点を使用できます。
④	出力指示条件	⑥で指定したMD番号のデータメモリ又はリレー領域に、①、②、③の接点情報および⑦のMDデータを出力するかどうか指示する入力で、00000~15777, 20000~75777の各リレー、TMR・CNTの接点を使用できます。ONのとき出力します。OFFになっても接点情報、MDデータは変化しません。

⑤	MD拡張出力	MD命令を同一出力指示条件で連続して使用するとき、それぞれのMD命令に④の条件をプログラムする必要はありません。 詳細は次項「(2)MD命令のプログラム手順」を参照してください。
⑥	MD番号	MD命令は①、②、③の接点情報、⑦のMDデータの各情報を格納するデータメモリ領域としてTMR、CNTの現在値格納領域(b0000~b1777)またはリレー領域(00000~15777)を使用します。 (1)TMR、CNT領域を使用するとき TMR、CNTと同じ000~777の番号でプログラムし、情報はプログラマ等でモニタします。 ②TMR、CNTで使用した番号と重複使用はできません。 (2)リレー領域を使用するとき バイトアドレスコ××××でプログラムします。たとえば 20000とプログラムすると、20000、20001の2バイトがMD用の領域となります。出力ユニットを装着している領域を使用すると、①、②、③の接点情報と、⑦のMDデータを外部出力(表示)できます。
⑦	MDデータ	BCDコードで000~999の任意の数値を使用できます。工程番号、リレー番号、外部機器番号等と関連付けてプログラムします。

(2) MD命令のプログラム手順



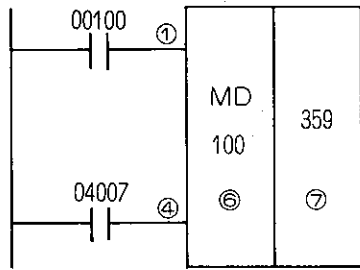
```

STR    00010  —①
STR    00014  —②
STR    00017  —③
STR    04010  —④  出力指示
MD     003    —⑥  MD番号
        200   —⑦  MDデータ
    
```

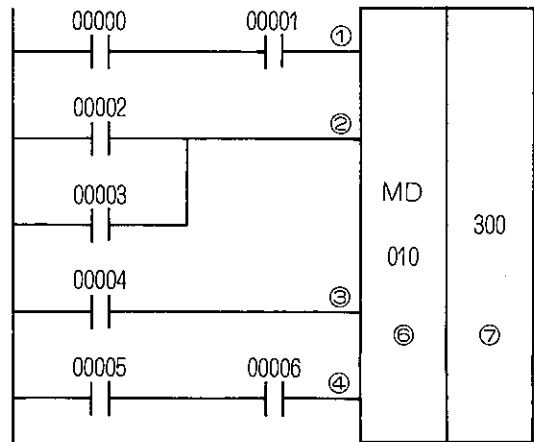
●入力情報をモニタ（外部出力）する必要のない場合、プログラムする必要はありません。

●入力情報、出力指示条件とも単一条件でない複雑な論理演算結果でもかまいません。

(例1)

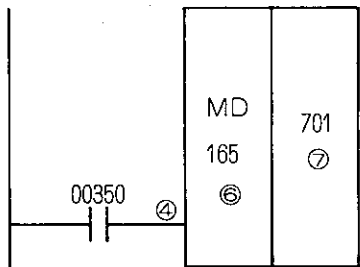


STR 00100 —① 入力情報
 STR 04007 —④ 出力指示
 MD 100 —⑥ MD番号
 359 —⑦ MDデータ



STR 00000) —①
 AND 00001) —① 入力情報
 STR 00002) —②
 OR 00003) —②
 STR 00004 —③
 STR 00005) —④ 出力指示
 AND 00006) —④
 MD 010 —⑥ MD番号
 300 —⑦ MDデータ

(例2)

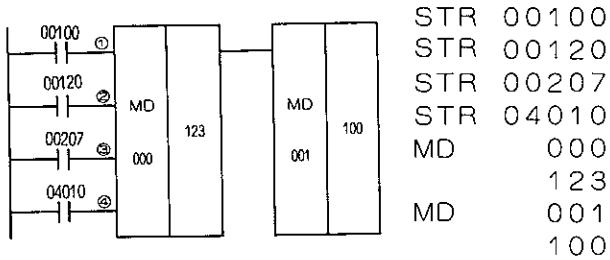


STR 00350 —④ 出力指示
 MD 165 —⑥ MD番号
 701 —⑦ MDデータ

MD命令演算時のスタックレジスタの推移

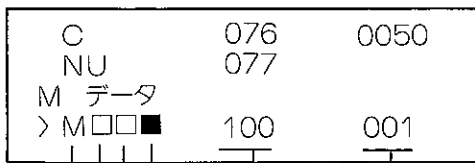
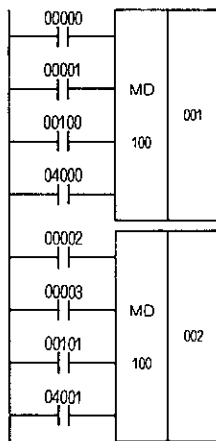
	アキュムレータ		スタックレジスタ		
	ACC	S1	S2	S3	
STR 00000					
AND 00001					
STR 00002					
OR 00003					
STR 00004					
STR 00005					
AND 00006					
MD 010 300	出力指示④	入力情報③	入力情報②	入力情報①	

- MD命令の演算実行後もアキュムレータおよびスタックレジスタの状態は変化しません。したがって同一出力指示条件でMD命令を連続使用するときには次のようにプログラムできます。



(3) MD情報モニタ

プログラマでMD情報をモニタすると次のように表示します。



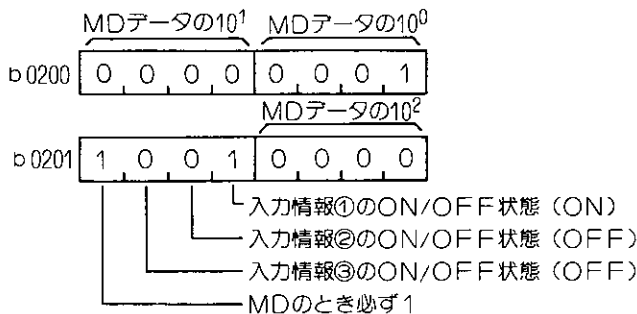
MDを示す
①の状態(ON)
②の状態(OFF)
③の状態(OFF)

プログラマの表示から次のような情報が得られます。

MDデータが001であるから

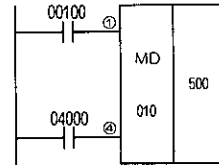
- 補助リレー04000がONで04001はOFF
- 表示中の入力情報は ①……00000 (ON)
②……00001 (OFF)
③……00100 (OFF)

参考 MD番号100のMD情報はデータメモリのb0200、b0201に格納しています。



入力情報①、②、③でプログラム上使用していないものがあるとき、モニタした場合の入力情報の表示に注意してください。

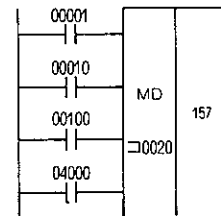
- 下図のような場合、MD 010の演算時入力情報①はスタックレジスタS1に、出力指示条件④はアキュムレータに格納しています。スタックレジスタS2、S3にはそれ以前の演算で使用した中間結果が残っているため、MD情報としては全く無意味なものです。



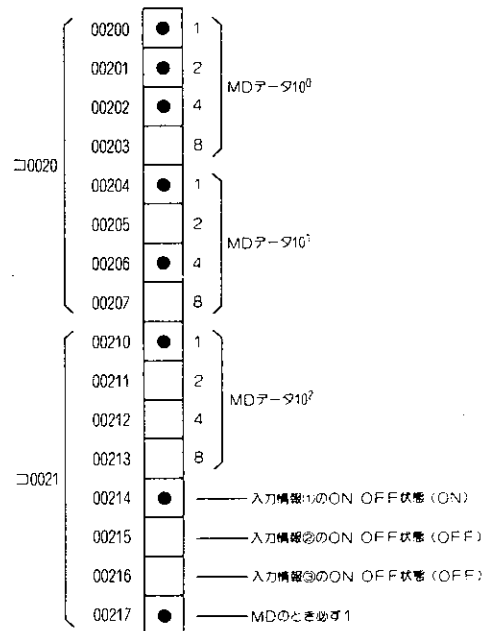
MD番号は本例のように重複使用できますが、出力指示条件が同時にONになった場合、プログラム順が後の方の出力指示条件が有効となります。(例では04001が有効)

(4) MD情報の外部出力

MD番号のかわりにデータメモリのリレー領域をバイトアドレスで指定すると、MD情報を外部に出力したり、データリンク機能を使って他のPCに伝送できます。



CO020と指定することで、CO020、CO021の2バイトにMD情報を出力します。CO020、CO021には出力ユニットを装着しておきます。



● EFGON

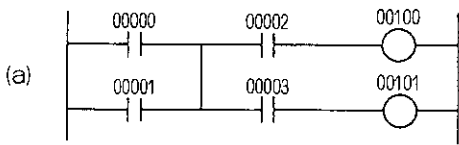
8-3 ラダー設計に関する留意事項

PCはプログラムメモリを順次読み出し、その内容に基づき演算を行う直列処理方式のため、リレー盤用のラダー図をそのまま適用できないことがあります。また、リレー盤では必要であった回り込み防止ダイオードが不要となったり、補助接点の使用数に制限が無い等の利点もあります。

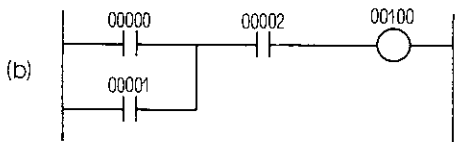
以下のリレー盤でのラダー設計とPCでのラダー設計の相違点を十分理解し、効率の良いラダー図を設計してください。

(1) リレー盤用ラダー図から書換えを必要とする回路

(例1)



(a)のラダー図は、このままではPCでは使用できません。



(b)の部分は

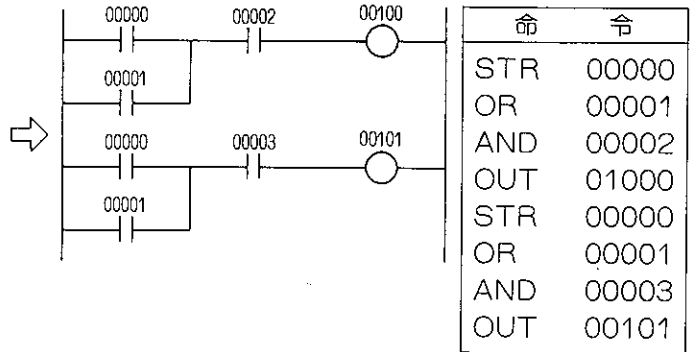
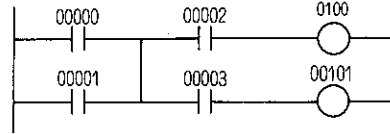
命 令	
STR	00000
OR	00001
AND	00002
OUT	00100

というプログラムで演算可能です。

(b)のプログラムを演算する場合のACC (アキュムレータ) の状態推移は、次のようになります。

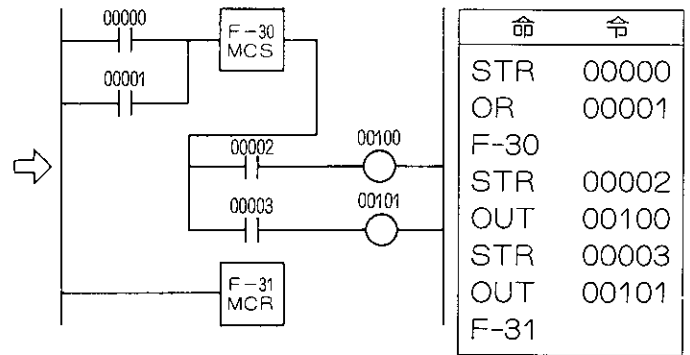
命 令	ACCの内容
STR 00000	00000
OR 00001	00000 00001 の演算結果
AND 00002	00000 00002 00001 の演算結果
OUT 00100	00000 00002 00001 の演算結果

したがってAND 00002まで演算すると $\frac{00000}{00001}$ の演算結果はすでに消滅して、これを00003に反映できません。PC用のラダー図として、次のように書き換えます。



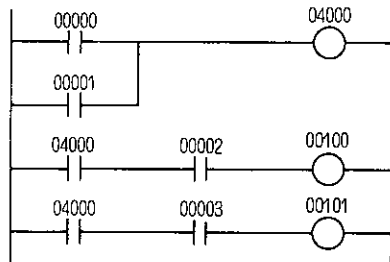
命 令	
STR	00000
OR	00001
AND	00002
OUT	01000
STR	00000
OR	00001
AND	00003
OUT	00101

または



命 令	
STR	00000
OR	00001
F-30	
STR	00002
OUT	00100
STR	00003
OUT	00101
F-31	

または



命 令	
STR	00000
OR	00001
OUT	04000
STR	04000
AND	00002
OUT	00100
STR	04000
AND	00003
OUT	00101

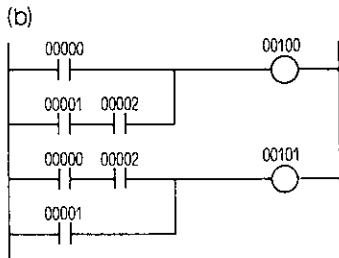
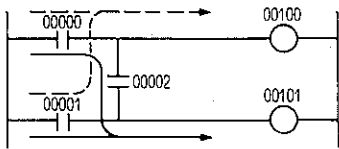
(4000は補助リレーとして使用)

F-30 (MCS)、F-31 (MCR)に関しては、第9章「応用命令の説明」を参照してください。

ACCにはプログラムの1命令を演算することに演算結果が0または1で入ります。

(例2)

(a)のリレー盤のラダー図は、00002に00000からと、00001からの両方向に電流が流れ、(b)のPC用に変えたラダー図と同様の動作をします。

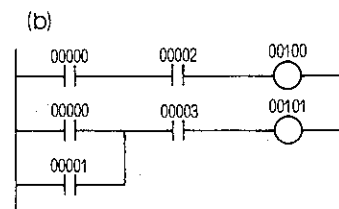
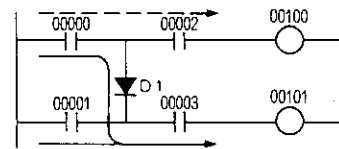


命 令	
STR	00000
STR	00001
AND	00002
OR STR	
OUT	00100
STR	00000
AND	00002
OR	00001
OUT	00101

PCでは(a)の00002のようにラダー図上の1つの接点シンボルに両方向に電流が流れるような考え方は成り立ちません。PCの演算はプログラムメモリをアドレス0からEND命令まで順次スキャンする方式のため、ラダー図上の同一接点シンボルを2度通るような処理は行いません。

(例3)

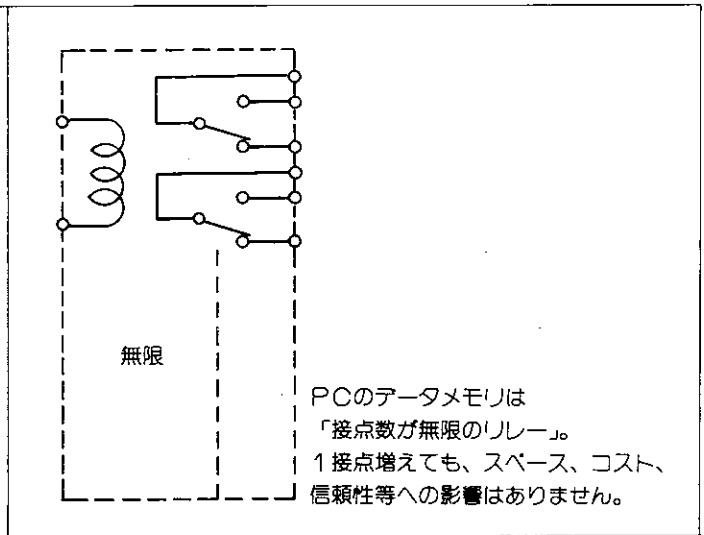
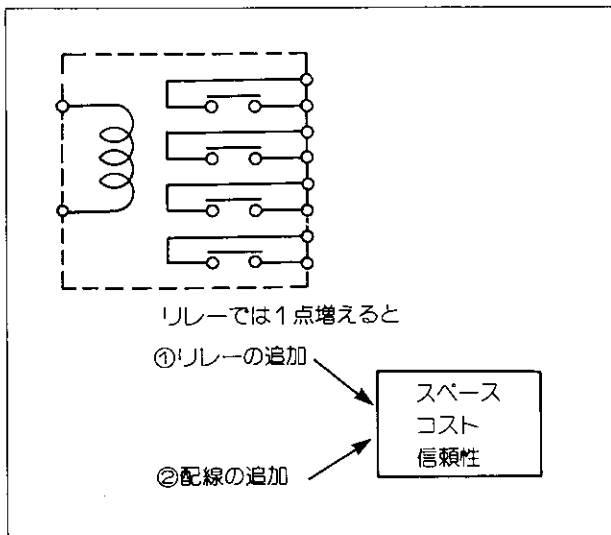
(a)のリレー盤の回路は回り込み防止ダイオードD1の働きにより、00001から00002には電流は流れず、(b)のPC用に変えたラダー図と同様の動作をします。



命 令	
STR	00000
AND	00002
OUT	00100
STR	00000
OR	00001
AND	00003
OUT	00101

PCでは(a)のD1のような回り込み防止ダイオードをプログラムできません。

(例1)、(例2)、(例3)はリレー盤では、接点数の少ないリレーが使用できることや、盤内の配線が簡単になるため、ごく一般的に使われるテクニックですが、PCにはデータメモリという「接点数が無限にあるリレー」を使用しているため、接点数を制約する努力は不要で、むしろ誰が見ても理解できるラダー図の設計ができます。



〔2〕 入出力一括処理方式

6・1ページ「運転サイクル」の説明のように、JW30Hでは毎スキャンサイクルに「入出力処理」というデータメモリと入出力ユニット間でデータ交換を行う処理があります。

入出力処理では、ベースユニットに装着した入出力ユニットをラック、スロット番号の若い方から順にスキャンし、

①入力ユニットであれば

入力ユニットに接続した外部接点のON/OFF状態をデータメモリに書き込みます。

②出力ユニットであれば

当該のデータメモリのON/OFF状態を読み出し、出力ユニットのラッチに書き込みます。

入出力処理で、ベースユニットに装着した全ての入出力ユニットに対して以上の処理を行った後、ユーザープログラム処理に入ります。

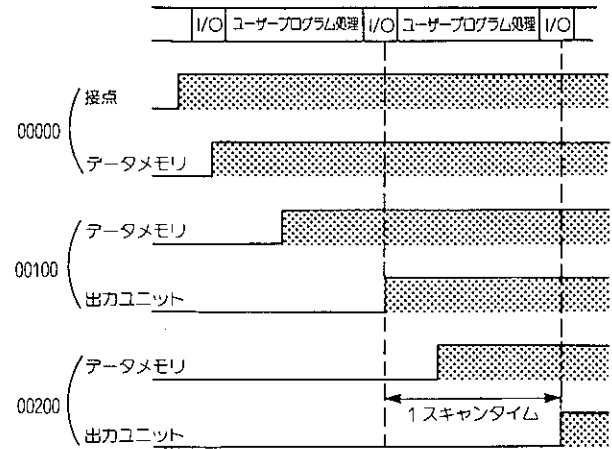
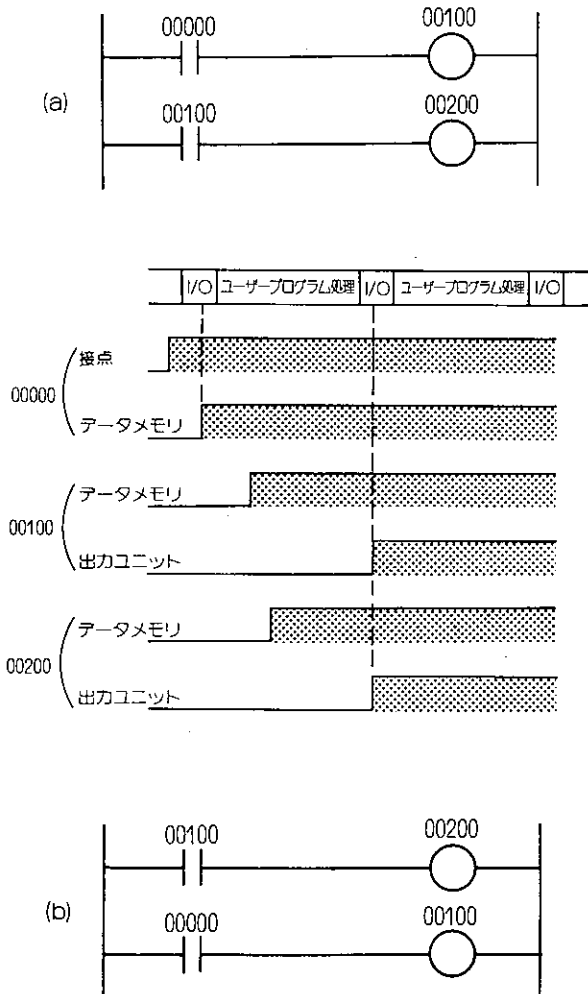
このように入出力ユニットに対する処理を一括して行うPCでは次の事項を念頭に置いてラダー設計をする必要があります。

- (1) 外部接点のON/OFF状態の変化は1スキャンに1度の入出力処理でデータメモリに取り込みます。したがって、ユーザープログラム処理中に外部機器のON/OFF状態が変化しても、そのスキャンサイクル中はデータメモリ（入力として割当てられているもの）の内容は変化しません。このため「入力レーシング現象」(6・5ページ「ユーザープログラム処理」参照)は発生しません。
- (2) 演算結果のON/OFF状態をデータメモリから出力ユニットに書き込むのは1スキャンに1度の入出力処理で行います。したがって演算結果を出力ユニットに出力するのは、次のスキャンの入出力処理となります。

(3) プログラム順序による影響

PCはプログラムの先頭からEND命令までを直列に演算し、これを何度も繰り返します。(サイクリック・スキニング方式)

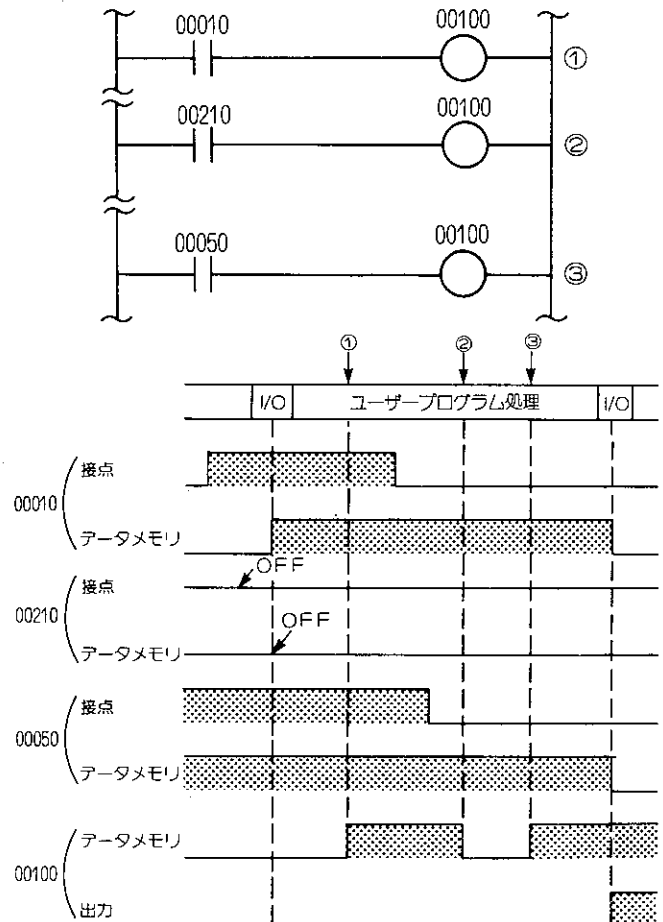
- (1) プログラム順を入れ替えると異なった動作をすることがあります。



(a)のプログラムでは、入力00000がONになると、出力00100、00200は同一スキャン内でONとなりますが、(b)では1スキャン遅れて00200がONになります。コイルの補助接点を使う場合、「コイルの前に書かれた補助接点の状態変化は、コイルの状態が変わった次のスキャンに生じる」ということを考慮してプログラムしてください。

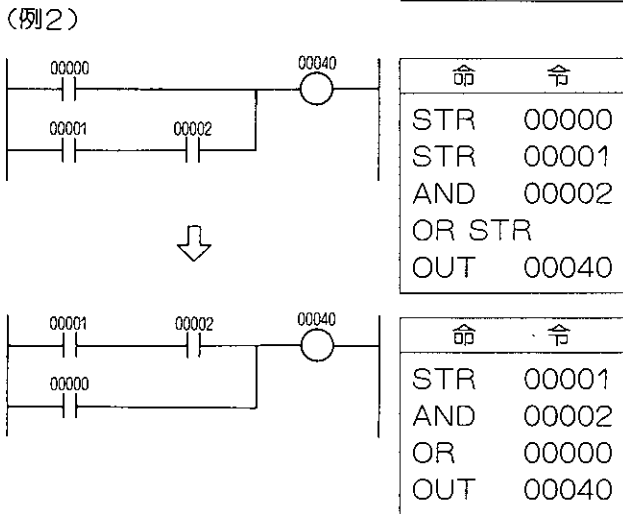
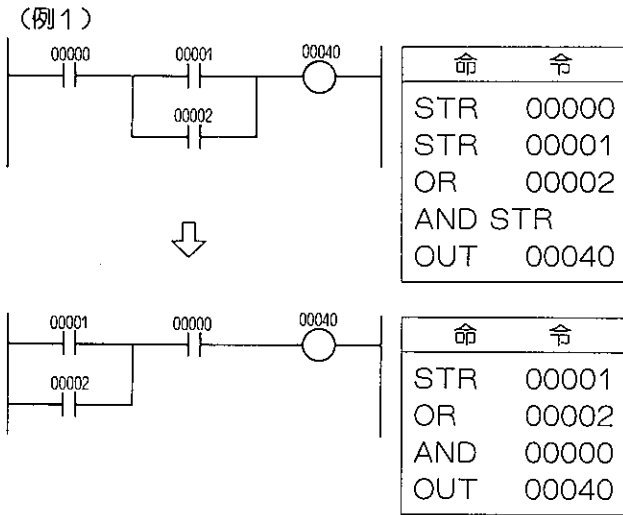
- (2) コイルの2重使用

同一のリレー番号をコイルとして複数回使用すると、それぞれのプログラム内容に応じデータメモリの内容は変化し、出力ユニットには一番最後に書かれたプログラムの演算結果をデータメモリから書き込みます。

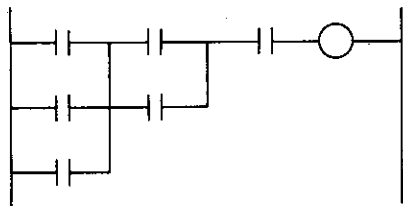


(4) プログラムの簡略化

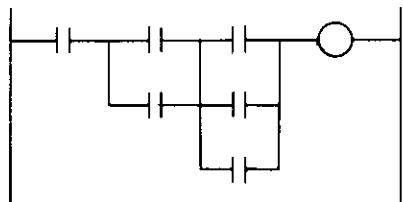
シーケンス回路によっては、回路を書き換えることによりプログラムを簡単にできます。



一般に左下りの回路を作るとプログラムが簡単になります。



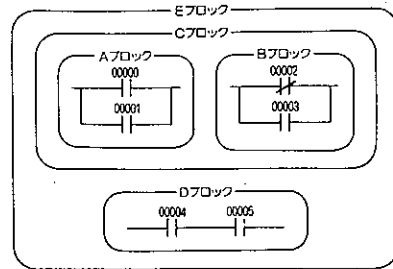
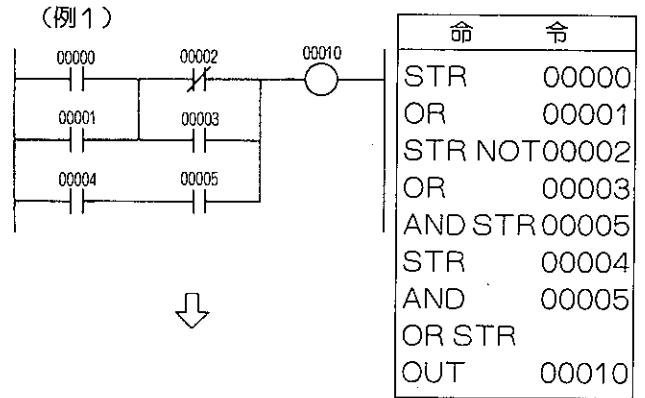
左下りの回路



右下りの回路

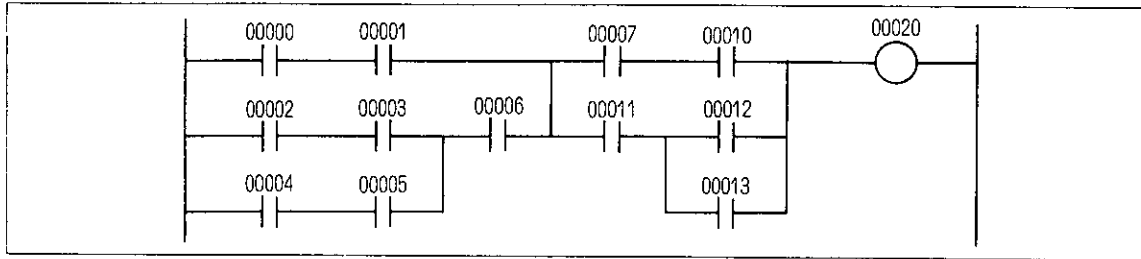
(5) 直並列回路のプログラム

直並列回路をプログラムする場合にはまず、小さなブロックに分割し、その小さなブロック毎にプログラムし、最終的に1つの大きなブロックになるようにします。



命 令	アキュムレータ ACC	スタックレジスタ S1
A [STR 00000 OR 00001	00000	直前のACCの状態
	00000 00001	
B [STR NOT 00002 OR 00003	00002	00000
	00002 00003	00000 00001
C [AND STR	00000 00002 00001 00003	
		00000 00002 00001 00003
D [STR 00004 AND 00005	00004	00000 00002 00001 00003
	00004 00005	00000 00002 00001 00003
E [OR STR	00000 00002 00001 00003 00004 00005	
		00000 00002 00001 00003 00004 00005
OUT 00010	00000 00002 00001 00003 00004 00005	

(例2)



命令	アキュムレータ ACC	スタックレジスタ		
		S1	S2	S3
STR 00000	00000	直前のACCの状態		
AND 00001	00000 00001			
STR 00002	00002	00000 00001		
AND 00003	00002 00003	00000 00001		
STR 00004	00004	00002 00003	00002 00001	
AND 00005	00004 00005	00002 00003	00002 00001	
OR STR	00002 00003 00004 00005	00000 00001		
AND 00006	00002 00003 00006 00004 00005	00000 00001		
OR STR	00000 00001 00002 00003 00006 00004 00005			
STR 00007	00007	00000 00001 00002 00003 00006 00004 00005		
AND 00010	00007 00010	00000 00001 00002 00003 00006 00004 00005		
STR 00011	00011	00007 00010	00000 00001 00002 00003 00006 00004 00005	
STR 00012	00012	00011	00007 00010	00000 00001 00002 00003 00006 00004 00005
OR 00013	00012 00013	00011	00007 00010	00000 00001 00002 00003 00006 00004 00005
AND STR	00011 00012 00013	00007 00010	00000 00001 00002 00003 00006 00004 00005	
OR STR	00007 00010 00011 00012 00013	00000 00001 00002 00003 00006 00004 00005		
AND STR	00000 00001 00007 00010 00002 00003 00006 00011 00012 00004 00005 00013			
OUT 00020	00000 00001 00007 00010 00002 00003 00006 00011 00012 00004 00005 00013			

第 9 章 応用命令の説明

9-1 応用命令に関する留意事項

(1) 数値の表現方法

(1) 2進数 (Binary Code)

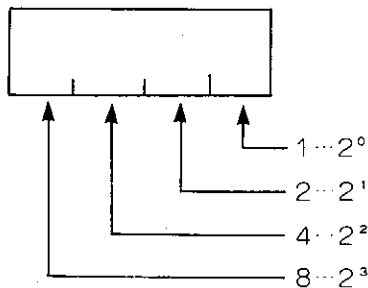
我々が日常使用している10進数では0~9の数字を使用します。ロジックの世界では0 (OFF) と1 (ON) の2つの状態しか存在しませんが、この0と1であらゆる数値を表現できます。

0と1で表現した数値を2進数といいます。

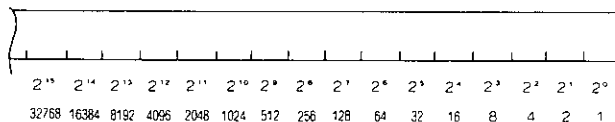
10進数では0、1、2、…8、9と数字が増えると、次に10と桁上げが起こりますが、2進数ではこの桁上げが0、1の次に10という形で起こります。したがって10 (イチゼロと読む) は10進数の2を意味します。以下同様に11→100、111→1000と桁上げが起こります。

10進数	0	1	2*	3	4*	5	6	7	8*
2進数	0	1	10	11	100	101	110	111	1000

*印のところで桁上げが起こっています。したがって2進数の各桁は次のような「重み」を持っていることになります。

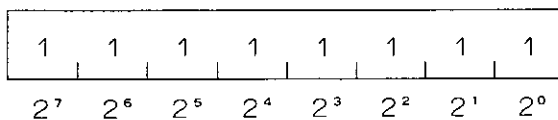


以下同様にして、各桁は 2^n の重みを持ちます。



2進数の各桁を「ビット」と呼びます。

JW30Hのレジスタは8ビットで構成しています。8ビットがすべて1のとき次のようになります。



それぞれのビットの重みを合計すると

$$2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 + 2^6 + 2^7 = 1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 = 255$$

すなわち8ビットで0~255(16ビットでは0~65535)の10進数を表現できます。

(2) 2進化10進数 (Binary Coded Decimal ...BCD)

10進数は0、1、2…9の次は10と桁上げが起こります。2進数にさらにこの9→10と同じような桁上げを付加したものを2進化10進数といいます。

10進数	2進数	BCD
0	0	0
1	1	1
2	10	10
3	11	11
4	100	100
5	101	101
6	110	110
7	111	111
8	1000	1000
9	1001	1001
10	1010	1 0000
11	1011	1 0001
12	1100	1 0010
99	1100011	1001 1001

すなわち4ビットごとに区切りを設け、4ビット内で1001以上のビットの組合せ(1010等)を禁止し、桁上げを起こします。したがって各4ビットは10進数で0~9の範囲の数値を取り得ます。

(3) 8進数と16進数

PCの内部では、数値はすべて2進数(バイナリーコード)又はBCDコードで処理します。しかし、プログラムの書き込みや、演算結果のモニタを2進数(0と1のビットパターン)で行うとキー操作や重み計算が面倒なため、プログラムに2進↔10進変換機能(BCD↔10進変換機能)を持たせ、10進数でプログラムの書き込み、モニタを可能にしています。ただし、PCをビット演算機能を中心に考えたとき、ビットパターンを直感的に連想できる他の数値表現方法の方が望ましい場合が多々あります。8進数および16進数は、ビットパターンとの相性がよくPCやコンピュータでよく使われます。

a、8進数

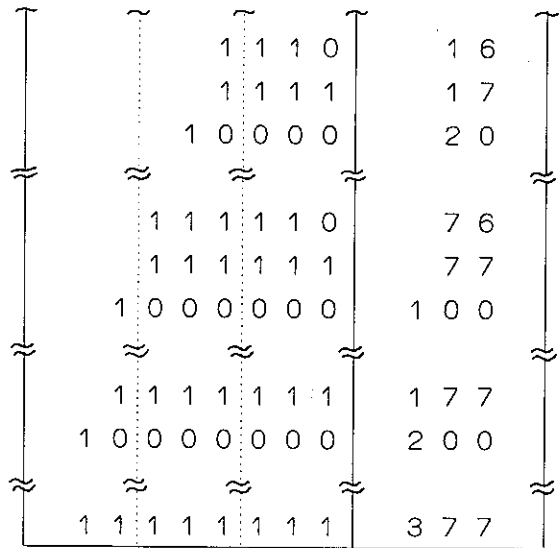
10進数では9→10、2進数では1→10と桁上げが起こりますが、8進数では7→10と桁上げが起こります。

10進数	2進数	8進数
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	10
9	1001	11
10	1010	12
11	1011	13
12	1100	14
13	1101	15
14	1110	16
15	1111	17
16	10000	20
...
62	111110	76
63	111111	77
64	1000000	100
65	1000001	101

桁上げ

桁上げ

桁上げ



レジスタは8ビットで構成していますので、0~377(8)の範囲を取り得ます。

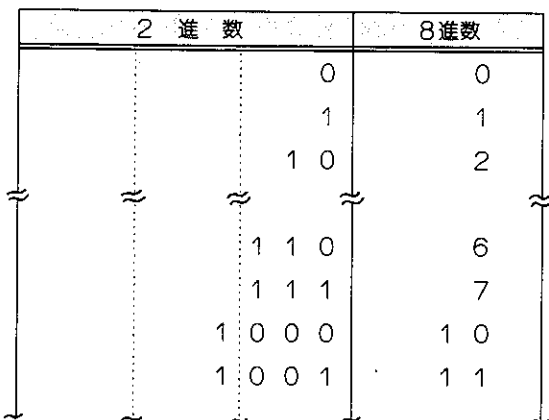
- データメモリのアドレス、システムメモリのアドレス、プログラムメモリのアドレスも8進法で表現します。

b、16進数

10進数では9→10と桁上げが起こりますが、16進数では9→A→B→C→D→E→FとなりF→10と桁上げします。

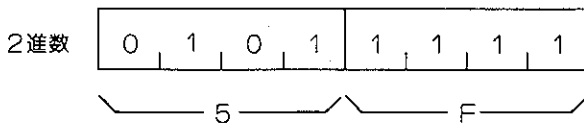
すなわち、0、1、2...7の次は8ではなく、10と桁上げが起こります。同様に17→20、77→100と桁上げが起こります。

- 8進数と2進数は次のように対応します。
2進数は3桁で0~7を表わし、111→1000と桁上げが起こります。
8進数は1桁で0~7の範囲をとり、7→10と桁上げが起こります。
2進数、8進数がともに7の次に桁上げが起こる性質から、2進数を3桁ごとに区切ると、これに1桁の8進数を当てはめることができます。



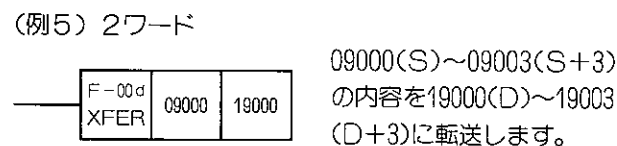
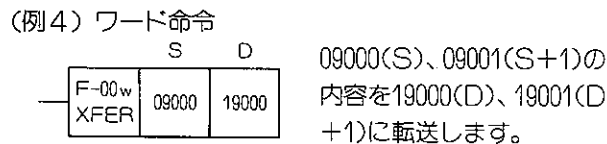
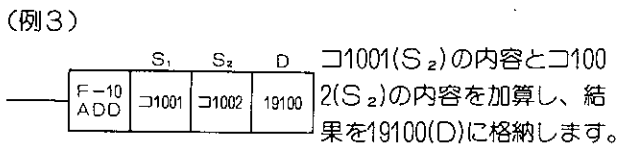
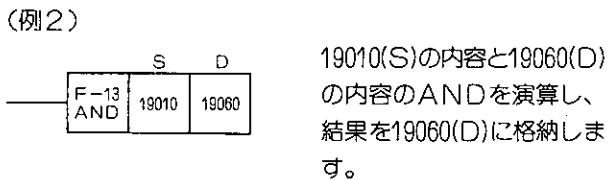
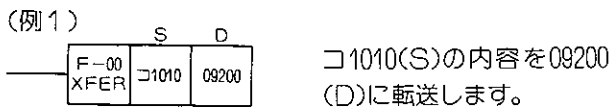
10進数	2進数	8進数	16進数
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
...
31	11111	37	1F
32	100000	40	20
...
255	11111111	377	FF

- 16進数と2進数は次のように対応します。
2進数を4ビットごとに区切り、これに16進数の1桁を割り当てます。

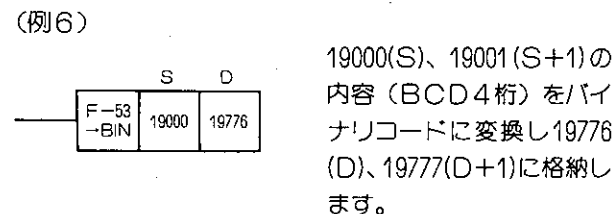


(2) ソースとデスティネーション

データ処理命令はバイト単位またはワード単位でデータメモリを扱います。演算前のデータが入っている方のレジスタをソース (Source—略号S) と呼び、演算結果を格納するレジスタをデスティネーション (Destination—略号D) と呼びます。



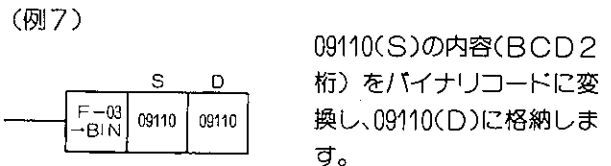
- ワード処理命令、2ワード処理命令は必ずソース、デスティネーションに偶数アドレスを設定してください。



- ソース、デスティネーションが2バイト以上のデータメモリを意味する命令ではSがコ1577のときS+1は、TMR・CNTの限時接点の領域(ファイルアドレスの001600~001777)に入ってしまいます。

また、Sがb1777のときS+1は09000、Sが09777のときS+1は19000となり、Sがコ7577のときS+1は、TMR・CNTの限時接点の領域(ファイルアドレスの035600~)に入っています。

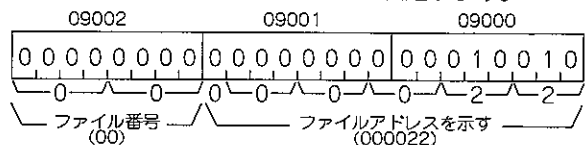
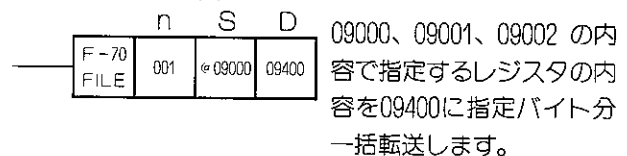
特に、TMR・CNTの限時接点、CPUの内部処理領域にデータの読出、書込を行なわないように注意してください。



- ソース側のレジスタの内容は演算実行後も変化しません。ただし、ソースとデスティネーションに同一レジスタの使用も可能です。この場合命令によってはソース(すなわちデスティネーション)の内容が変化します。

(3) 間接アドレス指定

本PCのデータ処理命令の中にはソース、デスティネーションに間接アドレスを指定できる命令があります。間接アドレス指定とはソース、デスティネーションに指定したレジスタ自身が演算を実行するのではなくそのレジスタを先頭とする3バイトの内容で指定するファイルアドレスのレジスタが演算を実行することをいいます。間接アドレス指定の場合、レジスタの前に@(アットマーク)を付加します。



ファイル番号：00~03, 10~2CHEX
ファイルアドレス：000000~177777oct

上記の例ではファイル0のファイルアドレス000022はコ0022ですので、結果的に@09000はコ0022を示します。

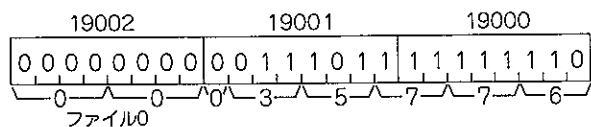
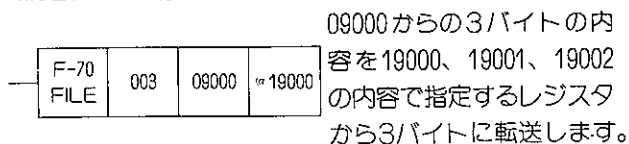


- 間接アドレス指定する場合は必ず偶数アドレスを設定してください。奇数アドレスを設定した場合は自動的にアドレスを-1した偶数アドレスと同じ動作となります。

@コ0001、@09121等は禁止

- ファイル0の036000以降は、CPUの内部処理に使用しており、使用禁止領域です。従って間接アドレス指定はできません。

(例2)……………禁止例



レジスタの状態が上記の場合、演算後ファイル0の035776~036000の3バイトにデータを転送します。この036000は使用禁止領域です。

- ファイル0の001600~001777、035600~035777はTMR/CNT限時接点エリアなので指定しないで下さい。
- ワード処理命令では間接指定するアドレスは偶数アドレスを設定してください。奇数アドレスを設定すると自動的にアドレスを-1した偶数アドレスと同じ動作となります。
- 間接アドレス指定したファイルアドレスが各ファイルの最終アドレスを越えると次の様な処理になります。

a) JW-32CUH/H1の場合

ファイル1の最終アドレス(037777)の次はファイル2の先頭アドレス(000000)となります。ファイル2の最終アドレス(177777)を越える場合、演算は実行されません。

b) JW-33CUH/H1の場合

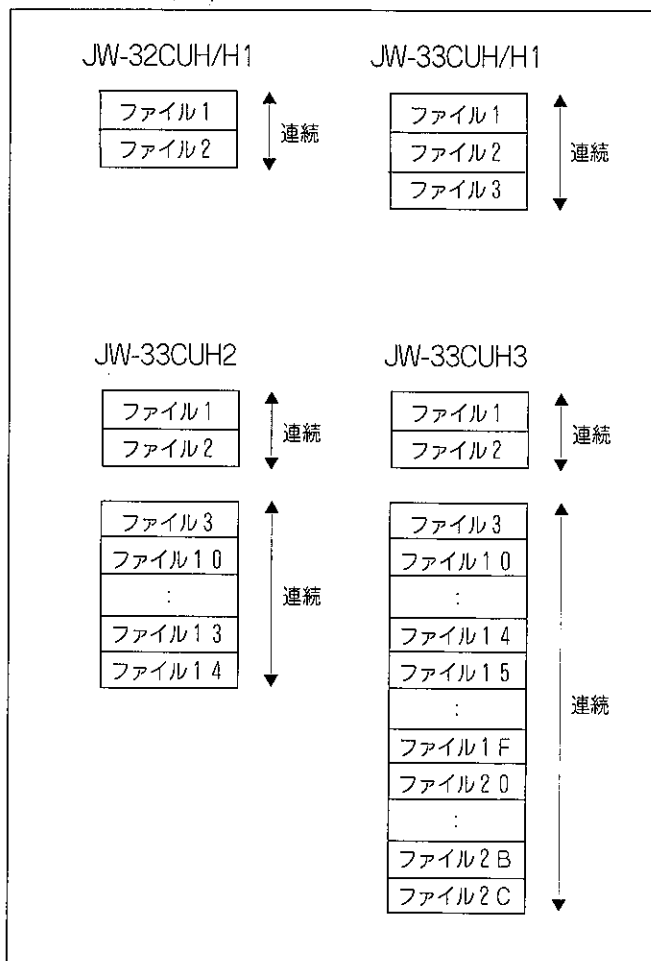
ファイル1の最終アドレスの次はファイル2の先頭アドレス、ファイル2の最終アドレスの次はファイル3の先頭アドレス(000000)となります。ファイル3の最終アドレス(177777)を越える場合、演算は実行されません。

c) JW-33CUH2の場合

ファイル1の最終アドレスの次は、ファイル2の先頭アドレスとなります。ファイル2の最終アドレスを越える場合、演算は実行されません。ファイル3の最終アドレスの次は、ファイル10の先頭アドレス(000000)、以降、ファイル13までは、最終アドレス(177777)を越える場合、その次のファイルの先頭アドレス(000000)となります。ファイル14の最終アドレス(177777)を越える場合、演算は実行されません。

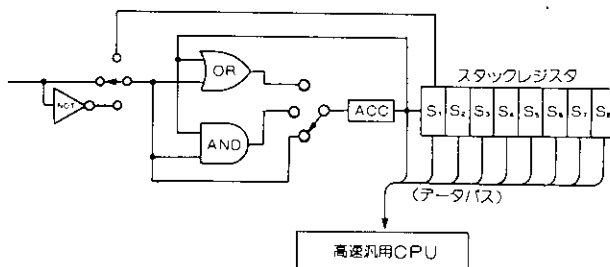
d) JW-33CUH3の場合

ファイル1の最終アドレスの次は、ファイル2の先頭アドレスとなります。ファイル2の最終アドレスを越える場合、演算は実行されません。ファイル3の最終アドレスの次は、ファイル10の先頭アドレス、以降、ファイル2Bまでは、最終アドレスを越える場合、その次のファイルの先頭アドレス(000000)となります。ファイル2Cの最終アドレス(177777)を越える場合、演算は実行されません。

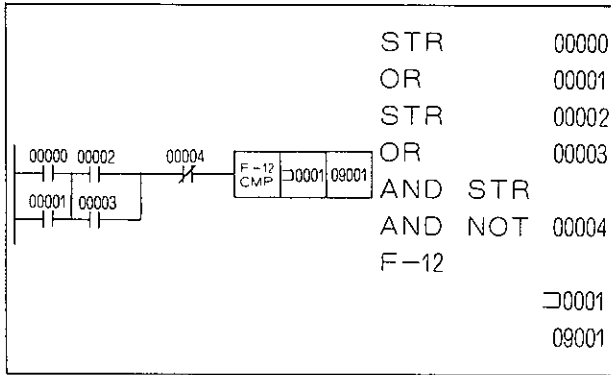


(4) 応用命令とスタックレジスタ

JW30HではTMR、CNT、MDの各命令とF00~F216の応用命令は高速汎用CPUで処理しています。これらの命令はACC(アキュムレータ)とSR(スタックレジスタ)の内容はデータバスを経由してCPUに送りこれを演算条件として実行します。



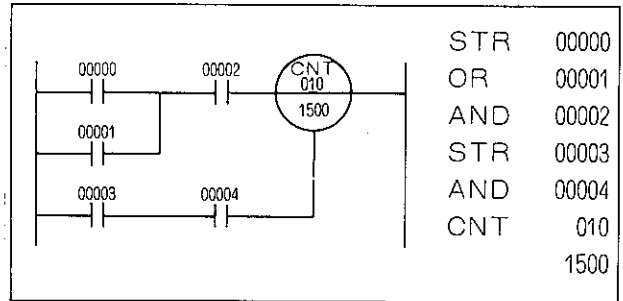
(例1) CNT5種類、MD、F-60、F-60W、F-60d、
F-61、F-61W、F-61d、F-62、F-62W、
F-62d、F-132、F-133を除く応用命令は、
ACCの内容のみを演算条件として実行します。



STR 00000
OR 00001
STR 00002
OR 00003
AND STR
AND NOT 00004
F-12
00001
09001

命令	アキュムレータ ACC	スタックレジスタ S ₁
STR 00000	00000	
OR 00001	00000 00001	
STR 00002	00002	00000 00001
OR 00003	00002 00003	00000 00001
AND STR	00000 00002 00001 00003	
AND NOT 00004	00000 00002 00004 00001 00003	
F-12	条件成立のとき演算	

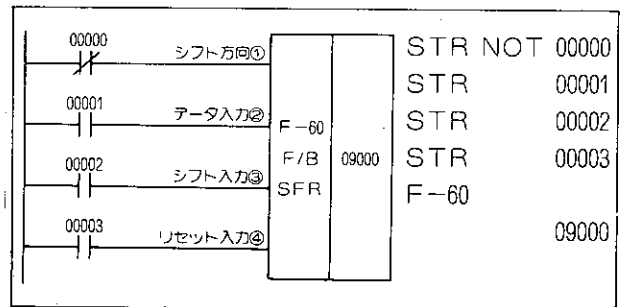
(例2) CNT命令の場合



STR 00000
OR 00001
AND 00002
STR 00003
AND 00004
CNT 010
1500

命令	アキュムレータ ACC	スタックレジスタ S ₁
STR 00000	00000	
OR 00001	00000 00001	
AND 00002	00000 00002 00001	
STR 00003	00003	00000 00002 00001
AND 00004	00003 00004	00000 00002 00001
CNT 010 1500	リセット入力	計数入力

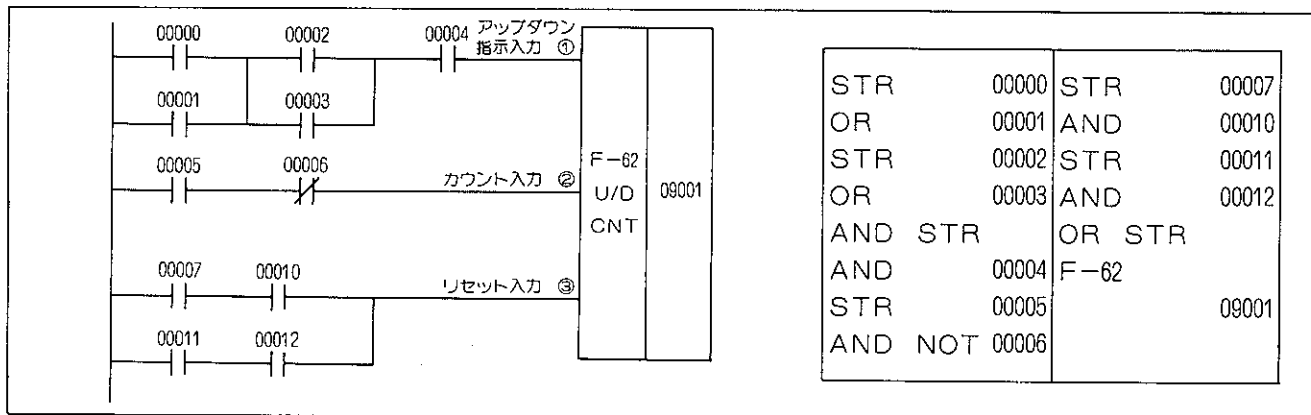
(例3) F-60ではACC、スタックレジスタ(S₁~S₃)
が演算条件となります。



STR NOT 00000
STR 00001
STR 00002
STR 00003
F-60
09000

命令	アキュムレータ ACC	スタックレジスタ		
		S ₁	S ₂	S ₃
STR NOT 00000	00000			
STR 00001	00001	00000		
STR 00002	00002	00001	00000	
STR 00003	00003	00002	00001	00000
F-60	リセット入力 ④	シフト入力 ③	データ入力 ②	シフト方向 ①

(例4) スタックの内容は複雑な直並列回路でもかまいません。



命 令	アキュムレータ ACC	スタックレジスタ		
		S ₁	S ₂	S ₃
STR 00000	00000			
OR 00001	00000 00001			
STR 00002	00002	00000 00001		
OR 00003	00002 00003	00000 00001		
AND STR	00000 00002 00001 00003			
AND 00004	00000 00002 00004 00001 00003			
STR 00005	00005	00000 00002 00004 00001 00003		
AND NOT 00006	00005 00006	00000 00002 00004 00001 00003		
STR 00007	00007	00005 00006	00000 00002 00004 00001 00003	
AND 00010	00007 00010	00005 00006	00000 00002 00004 00001 00003	
STR 00011	00011	00007 00010	00005 00006	00000 00002 00004 00001 00003
AND 00012	00011 00012	00007 00010	00005 00006	00000 00002 00004 00001 00003
OR STR	00007 00010 00011 00012	00005 00006	00000 00002 00004 00001 00003	
F-62	リセット入力 ③	カウント入力 ②	アップダウン指示入力 ①	

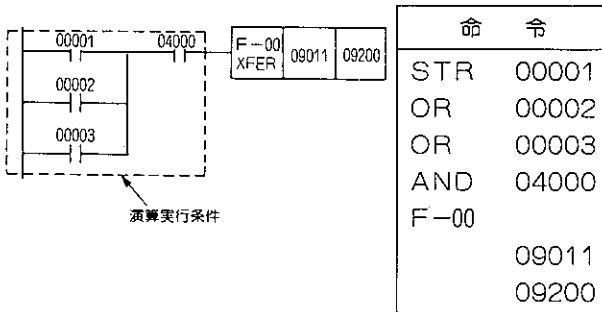
この例ではSTR00011演算時スタックレジスタを3段目(S₃)まで使います。

〔5〕 演算実行条件

- (1) 応用命令の演算実行条件（演算を実行するか否かの条件）は、1接点のON/OFFに限らず、複雑な直並列回路も可能です。

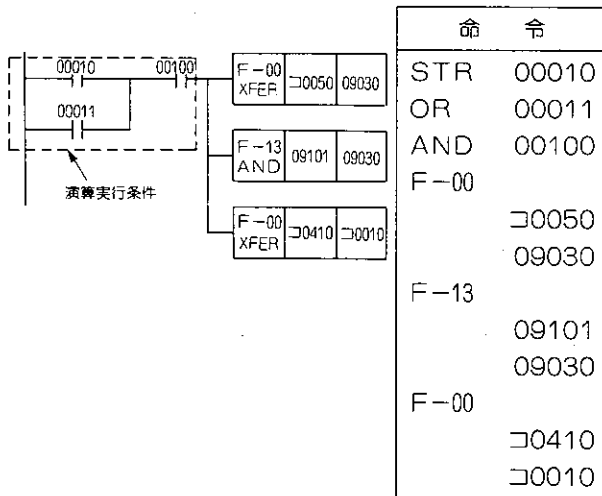
（9・4ページ「応用命令とレジスタ」参照）

（例）



- (2) 演算実行条件が共通の場合、次のように続けてプログラムできます。

（例）



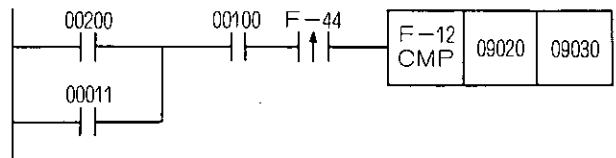
- 9・10ページ「倍長演算」の項を参照してください。

- (3) 応用命令には、演算実行条件が成立した場合の処理方式に次の2種類の形態があります。

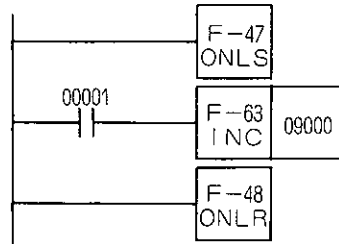
①	演算実行条件が成立している間、毎スキャンサイクル演算を実行するもの	F-12、F-61等
②	演算実行条件が成立した最初の1スキャンサイクルのみ演算を実行するもの	F-00、F-10等

②のグループの命令は、毎スキャンサイクルの当該命令演算時に、前のスキャンサイクルでの演算実行条件のON/OFF状態と、今回のスキャンサイクルの演算実行条件のON/OFF状態を比較し、前回OFF、今回ONの場合、演算実行条件がOFF→ONに変化したものとして実行します。

- ①のグループで、演算実行条件がOFF→ONの変化時のみ演算する必要がある場合、F-44（立上り微分命令）を使用してください。



- ②のグループで毎スキャンサイクル演算を実行する必要がある場合、F-47（レベル演算条件のセット）、F-48（レベル演算条件のリセット）を使用してください。



- (4) 演算実行条件が不成立の場合（演算実行条件がOFF→ONへの変化時のみ演算を実行する命令では、ON中の以降のスキャンサイクルも含まれます）、演算は実行せず、デスティネーション側のレジスタの内容は不変です。

またフラグに影響を与える命令の場合、フラグはクリアします。

- (5) JW30Hは、演算途中で電源をOFF（4.5V以下）にすると、その時点で演算を中止します。また、電源電圧が4.5V以上でも、スキャンサイクルのI/O処理でPF（パワーフェイル、停電）レベルを検知するとそのサイクルのEND命令で演算を中止します。

フラグに関しては次項「データ処理命令とフラグ」を参照してください。

〔6〕 データ処理命令とフラグ

- (1) フラグの種類

フラグ(Flag…旗)は、演算結果を以降のステップの演算に反映させるための信号で、JW30Hにはノンキャリアフラグ、エラーフラグ、キャリアフラグ、ゼロフラグの4種類のフラグがあり、データメモリの07354～07357の4ビットに割当てています。

ゼロフラグ	キャリアフラグ	エラーフラグ	ノンキャリアフラグ
07357	07356	07355	07354

- (2) フラグに影響を与える命令

F-10、F-60等の命令は演算結果に従いフラグをセットします。

③ スキャンサイクル中のフラグの推移

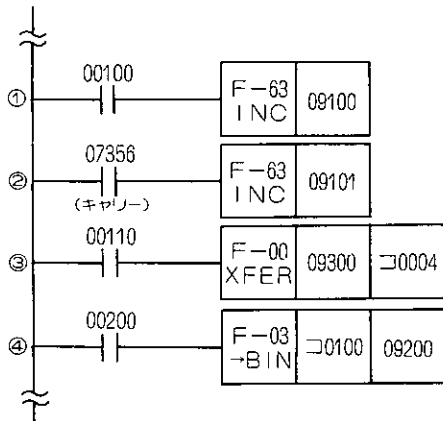
① 毎スキャンサイクルのユーザープログラム処理に先立ち、フラグはクリアします。6・1ページ「運転サイクル」を参照してください。

② フラグに影響を与える命令の処理に入ると、

- a. その命令の実行条件が成立しているとき
命令の演算結果によりフラグをセットします。
- b. その命令の実行条件が不成立のとき
フラグをクリアします。

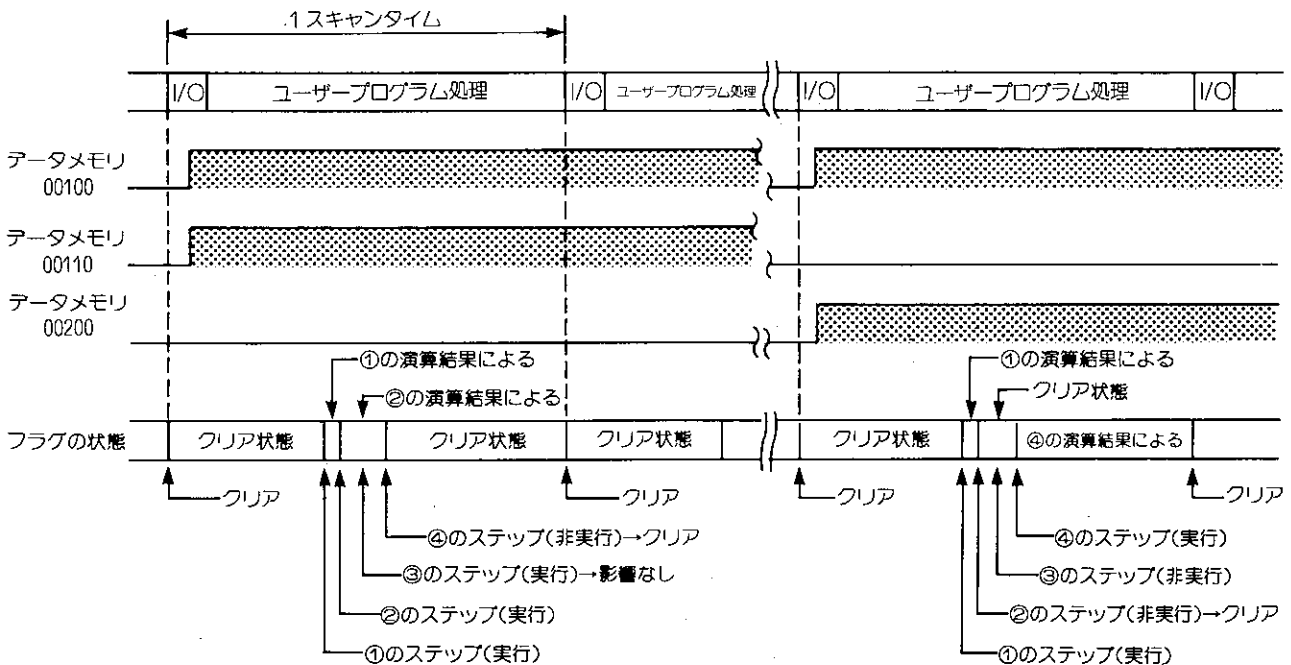
③ フラグに影響を与えない命令の処理では、実行・非実行にかかわらず、フラグの状態は変化しません。

(以前にフラグに影響のある命令なしとする。)



(以後フラグに影響のある命令なしとする。)

アドレス	命令	
01000	STR	00100
01001	F-63	
01002		09100
01003	STR	07356
01004	F-63	
01005		09101
01006	STR	00110
01007	F-00	
01010		09300
01011		C00004
01012	STR	00200
01013	F-03	
01014		C0100
01015		09200

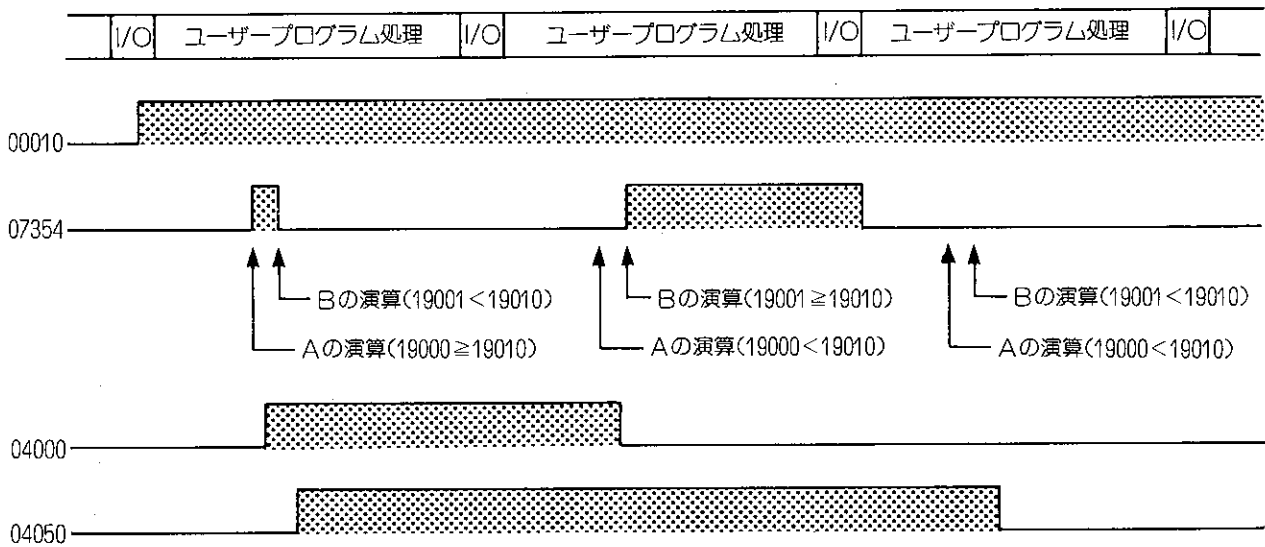
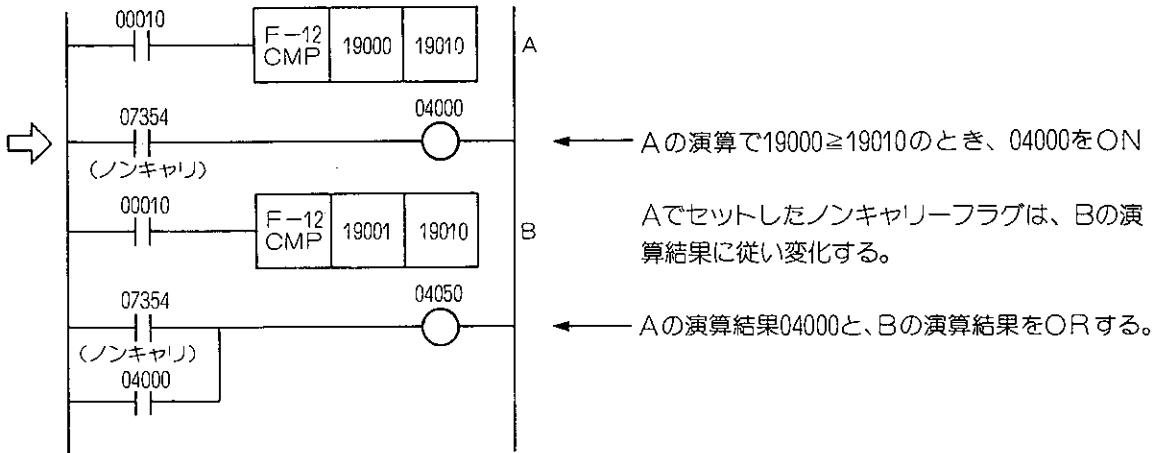


(4) フラグを保持する方法

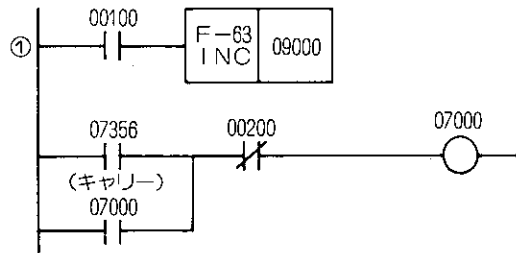
以上のように演算の結果セットしたフラグは、そのスキャンサイクル中、次にフラグに影響を与える命令の処理により変化したり、クリアしてしまいます。また次のスキャンサイクルに入るとユーザープログラム処理の前にクリアしてしまいます。

以下のように当該命令の直後にフラグの状態をコイル（補助リレー等）に書き込んでおくと次のスキャンサイクルの当該命令の演算まで保持できます。

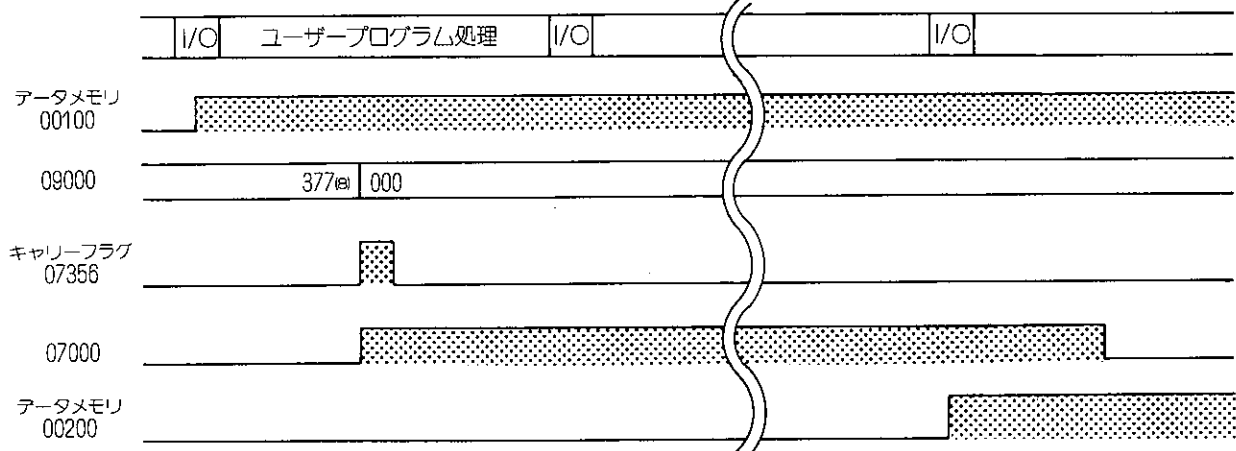
(例1) $19000 \geq 19010$ 又は $19001 \geq 19010$ のとき、04050をONにするプログラム



フラグの状態をプログラマ等の周辺装置でモニタしたり、外部に表示する場合は、(例1)のようにフラグの状態をコイルに書くだけでは1スキャンサイクルしか保持できないため、目で確認するのは困難です。このような場合、右図のようにフラグを自己保持してください。



00200をONにするまで、①の演算によるキャリーフラグ(07356)の状態を自己保持します。



(7) 倍長演算

(1) 倍長演算機能をもつ命令

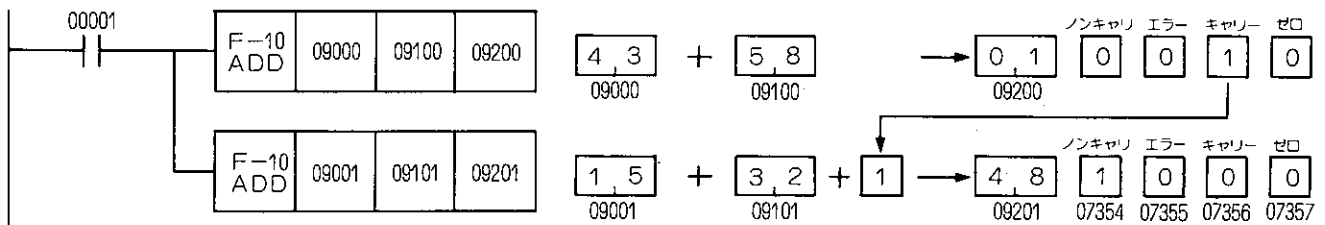
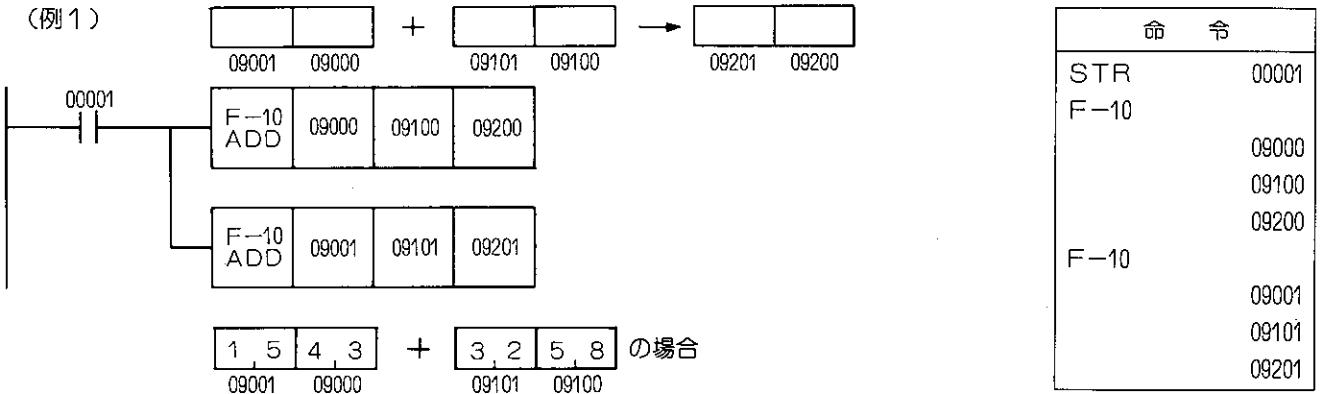
次の17種類の命令は、2バイト以上(ワード命令は4バイト以上)のデータの演算を可能とする倍長演算の機能があります。

- ① F-10、F-10w、F-10d レジスタ間の加算
- ② Fc10、Fc10w、Fc10d レジスタとBCD定数の加算
- ③ F-11、F-11w、F-11d レジスタ間の減算
- ④ Fc11、Fc11w、Fc11d レジスタとBCD定数の減算

- ⑤ F-12、F-12w、F-12d レジスタ間の比較
- ⑥ Fc12、Fc12w レジスタと定数の比較

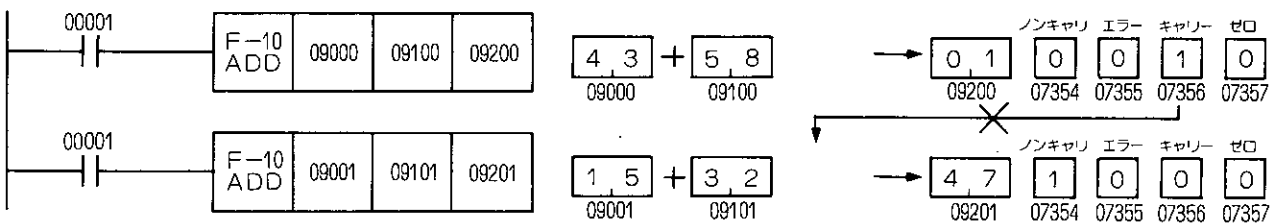
(2) 倍長演算時のプログラム

倍長演算は下の桁の演算により発生した桁上げ、桁下げ信号を次の桁の演算に自動的に反映させるもので、次のように演算実行条件に続けて下の桁からプログラムを書き込みます。

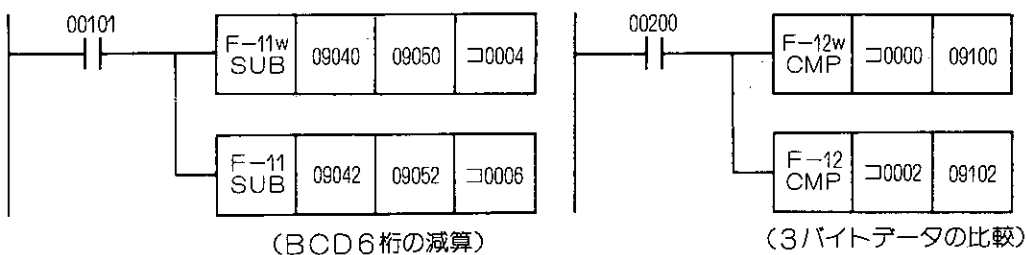


上の桁の演算時に下の桁のキャリーフラグを加算します。

【参考】 次のようにプログラムすると倍長演算になりません。



(例2) 3バイト以上の倍長演算も同様にして可能です。



③ 倍長演算時の内部処理

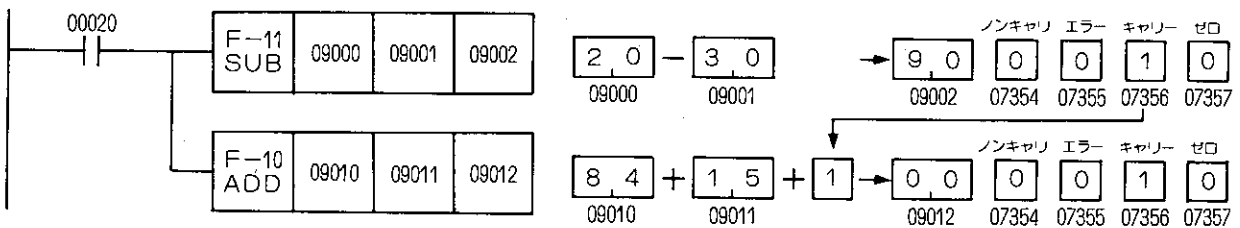
- ① 演算実行条件以後、最初に現われるF-10、Fc10、F-11、Fc11、F-12、Fc12及びそれぞれのワード(W)、2ワード(d)の各命令の演算時は、それ以前のフラグの状態を含めずに演算します。
- ② 共通演算実行条件中、次にF-10、Fc10、F-11、Fc11、F-12、Fc12及びそれぞれのワード(W)、2ワード(d)のいずれかの命令があると次のように演算します。
 - a、直前のキャリーフラグの状態を含めて演算を実行します。
 - b、ゼロフラグは、直前のゼロフラグの状態と、当該命令の演算によるゼロフラグの状態のANDをとり、いずれも1のときにゼロフラグをセットします。

F-10(W, d) Fc10(W, d)	直前のキャリーフラグの状態を加算
F-11(W, d) Fc11(W, d)	直前のキャリーフラグの状態を減算
F-12(W, d) Fc12(W, d)	直前のキャリーフラグの状態を減算

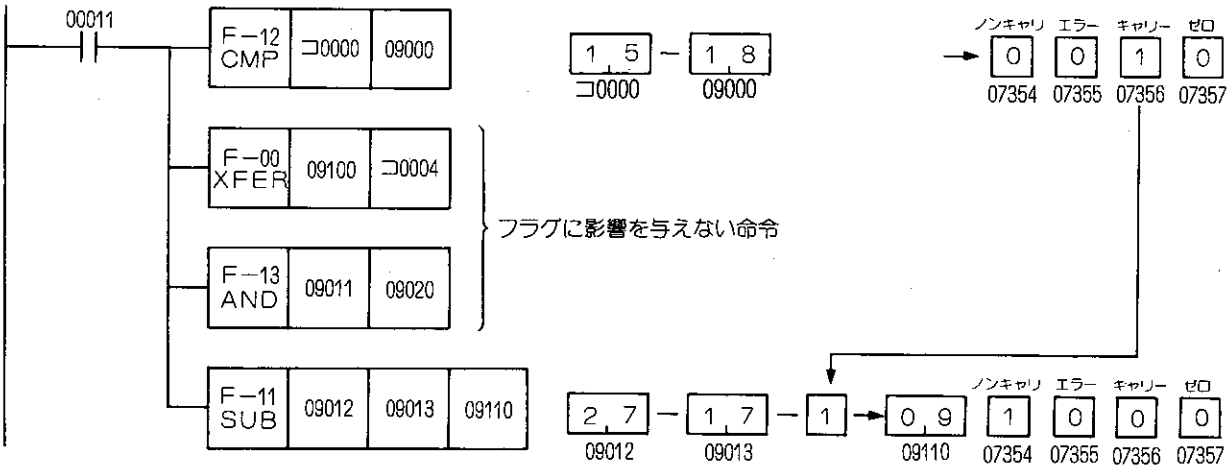
- F-12、Fc12命令はS1-S2又はS1-nの演算を行い、結果をフラグに格納します。

(4) 倍長演算に関する注意事項

- ① F-10、Fc10、F-11、Fc11、F-12、Fc12及びそれぞれのワード(W)、2ワード(d)は、共通演算条件の形式でプログラムしていると、異種命令間でもフラグを含んだ演算を行います。

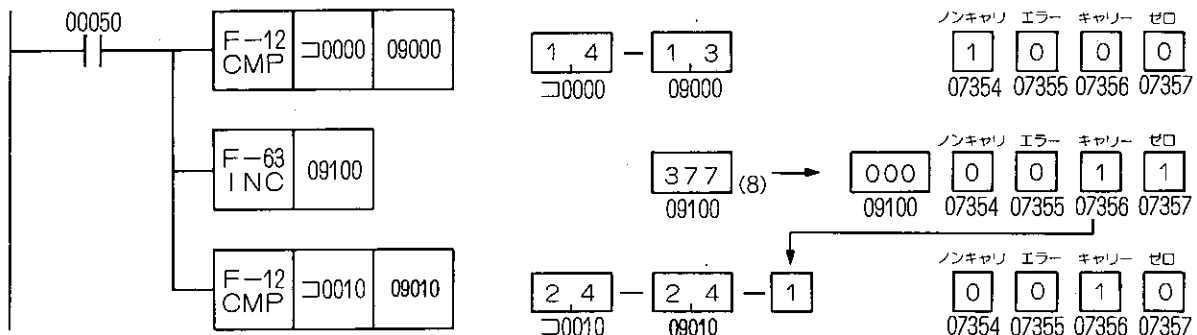


- ② F-10、Fc10、F-11、Fc11、F-12、Fc12及びそれぞれのワード(W)、2ワード(d)の間に、フラグに影響を与えない命令があっても倍長演算として実行します。



- 多数の命令が間に入る場合、特にご注意ください。

- ③ F-10、Fc10、F-11、Fc11、F-12、Fc12及びそれぞれのワード(W)、2ワード(d)の間に、フラグに影響を与える命令があると、その命令の演算によるフラグを含んだ演算を行います。



- ④ F-10、Fc10、F-11、Fc11及びそれぞれのワード(W)、2ワード(d)命令で、BCDコード以外を使用するとエラーフラグが立ち、それ以降の倍長演算は実行しません。

〔8〕符号付演算

(1) 符号付演算機能をもつ命令

次の4種類の命令は、符号付演算を可能とします。

- ① F-310 レジスタ間の符号付バイナリ加算 (31ビット+31ビット)
- ② F-311 レジスタ間の符号付バイナリ減算 (31ビット-31ビット)
- ③ F-315 レジスタ間の符号付バイナリ乗算 (31ビット×31ビット)
- ④ F-316 レジスタ間の符号付バイナリ除算 (31ビット÷31ビット)

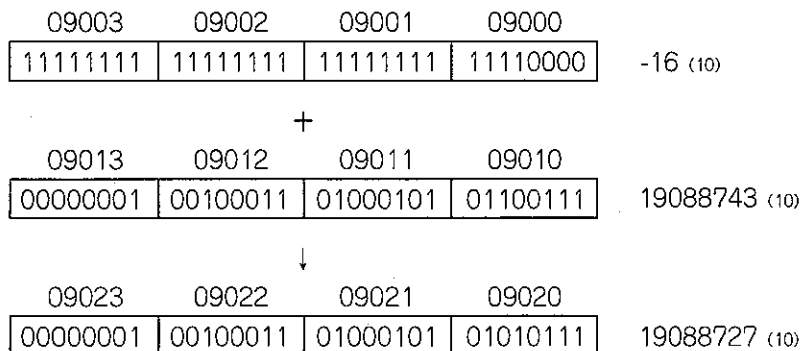
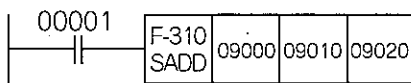
(2) 符号付演算で扱える数値の範囲

扱える数値の範囲は、2147483647~2147483648です。

負の数値は、2の補数表現をするものとします。

符号付き2進整数 (2の補数)	10進整数
01111111 11111111 11111111 11111111	2,147,483,647
01111111 11111111 11111111 11111110	2,147,483,646
⋮	
00000000 11111111 11111111 11111111	16,777,215
⋮	
00000000 00000000 11111111 11111111	65,535
⋮	
00000000 00000000 00000000 11111111	255
⋮	
00000000 00000000 00000000 00000010	2
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000000	0
11111111 11111111 11111111 11111111	-1
11111111 11111111 11111111 11111110	-2
⋮	
11111111 11111111 11111111 00000000	-256
⋮	
11111111 11111111 00000000 00000000	-65,536
⋮	
11111111 00000000 00000000 00000000	-16,777,216
⋮	
10000000 00000000 00000000 00000001	-2,147,483,647
10000000 00000000 00000000 00000000	-2,147,483,648

(例) 31ビット+31ビット



上記演算は -16+19088743=19088727 を示します。

[9] データメモリのブロックと基準アドレス

データメモリの256バイトを1ブロックとして分割したとき、その先頭アドレスを基準アドレスと呼びます。

基準アドレス	ブロック	範囲
コ0000	リレー	コ0000～コ0377
コ0400	〃	コ0400～コ0777
コ1000	〃	コ1000～コ1377
※ コ1400	〃	コ1400～コ1577
コ2000	〃	コ2000～コ2377
コ7400	〃	コ7400～コ7577
b 0000	TMR・CNTの現在値、MD情報	b 0000～b 0377
b 0400	〃	b 0400～b 0777
b 1000	〃	b 1000～b 1377
b 3400	〃	b 3400～b 3777
09000	レジスタ	09000～09377
09400	〃	09400～09777
19000	〃	19000～19377
99400	〃	99400～99777
E 0000	〃	E 0000～E 0377
E 0400	〃	E 0400～E 0777
E 1000	〃	E 1000～E 1377
E 7400	〃	E 7400～E 7777
000000	ファイル1のレジスタ	000000～000377
000400	〃	000400～000777
037000	〃	037000～037377
037400	〃	037400～037777

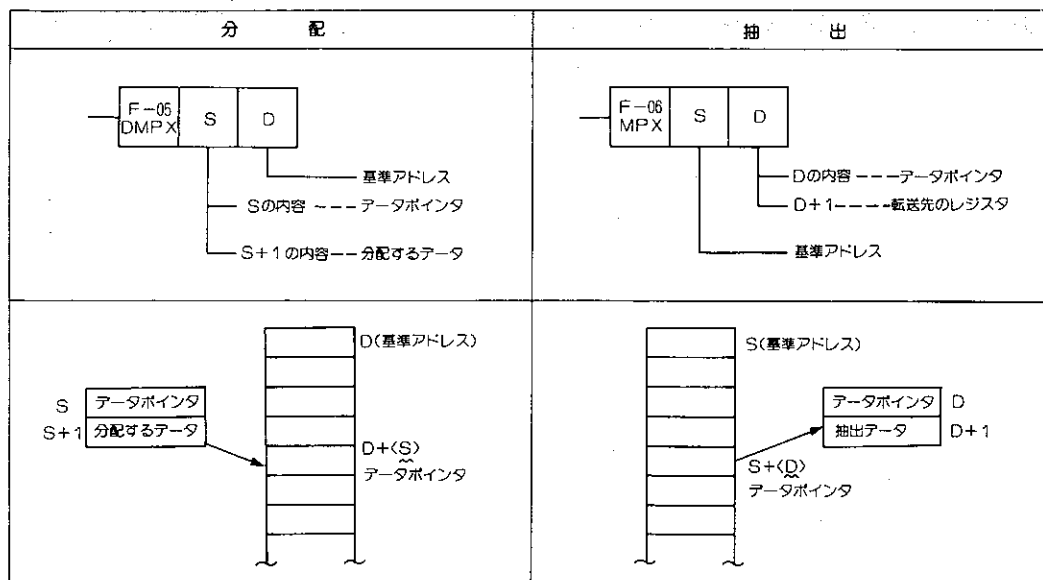
※ コ1400～コ1577のブロックは128バイトです。

次の各命令では基準アドレスを用います。

- ① F-05、F-05w(分配)
- ② F-06、F-06w(抽出)
- ③ F-72、F-72w(ファイルの1レジスタへの分配)
- ④ F-73、F-73w(ファイルの1レジスタからの抽出)

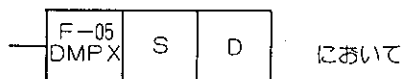
これらの命令はレジスタ間のデータ転送を行う命令ですが、(基準アドレス+データポインタ)で転送先のレジスタを指定できます。

a. F-05、F-06の場合



●基準アドレス

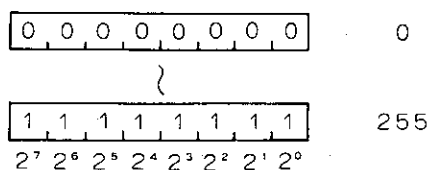
F-05のD、F-06のSが基準アドレスで、各ブロックの先頭アドレス(コ0000, …b0000, …09000, 09400, 19000, 19400等)を使用します。基準アドレスとしてブロックの先頭アドレス以外のアドレスもプログラム上設定できますが、JW30Hの演算ではそのアドレスの含まれるブロックの先頭アドレスを基準アドレスとして処理します。



Dの設定	演算上の基準アドレス
コ0200	コ0000
b0110	b0000
09005	09000

●データポインタ

F-05のS、F-06のDの内容がデータポインタとなります。S、Dは8ビットで構成していますので、0~255の値を取り出せます。

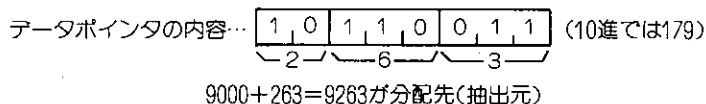


$$\begin{aligned}
 & \left[2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 \right] \\
 & = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 \\
 & = 255
 \end{aligned}$$

(基準アドレス+データポインタ)で各ブロック内の任意のデータメモリを分配先、抽出元とできます。データメモリのバイトアドレスは8進数で扱いますので、データポインタの内容も8進数と見なすと、対象のレジスタのアドレスが直接判断できます。

データポインタとなるレジスタの内容をF-63 (INC命令)で変化させたり、外部機器(デジタルスイッチ等)で指定することにより、分配先、抽出元を変化させられます。

基準アドレス.....09000



[10] ファイル番号を指定する命令

ファイル番号を付属語に指定する次の応用命令について、JW-33CUH2/H3は2桁のファイル番号(00~03, 10~2Chex)を指定できます。

- F-101 : 間接アドレスの設定
- F-102 : 直接指定アドレスのレジスタからの読出 (1バイト)
- F-102w : 直接指定アドレスのレジスタからの読出 (1ワード)
- F-103 : 直接指定アドレスのレジスタへの書込 (1バイト)
- F-103w : 直接指定アドレスのレジスタへの書込 (1ワード)
- F-176 : 直接指定アドレスのレジスタからの読出 (256バイト)
- F-177 : 直接指定アドレスのレジスタへの書込 (256バイト)
- F-202 : オープンチャンネル (局番 8 進数設定)
- F-203 : オープンチャンネル (局番 16 進数設定)
- F-207 : オープンチャンネル 2 (階層通信設定)

ラダーソフト (JW-52SP/92SP) を使用して、JW-70CUH/100CUHとJW-33CUH2/H3のプログラムを相互に変換すると、上記、応用命令で使用しているファイル番号 (4~7, 10~13) は右表のように変換されます。

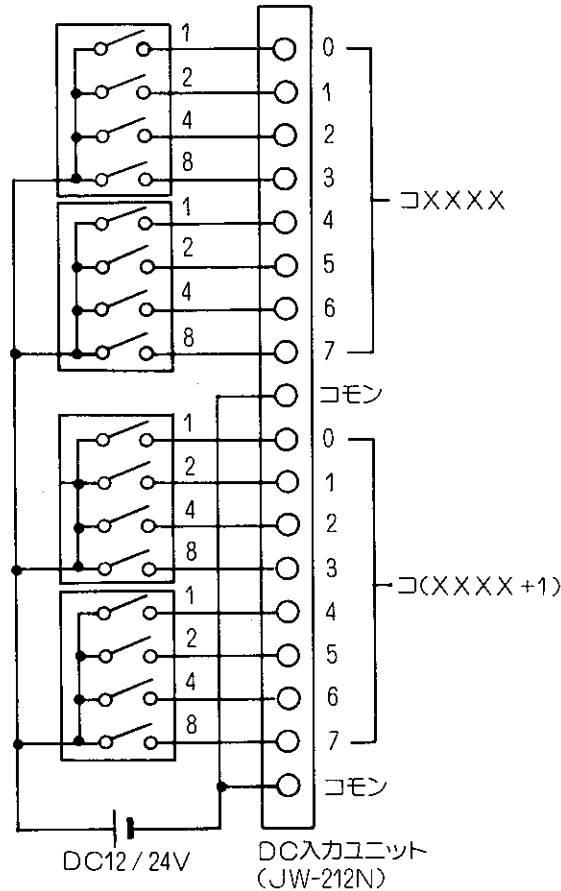
JW-70CUH/100CUH	JW-33CUH2/H3
ファイル 4	ファイル 1 0
ファイル 5	ファイル 1 1
ファイル 6	ファイル 1 2
ファイル 7	ファイル 1 3

[11] 数値信号の入出力方法

デジタルスイッチ等の外部機器から数値信号を読み込み JW30H のデータ処理命令で演算したり、演算結果を数字表示器に出力する場合の外部機器との接続例を示します。

(1) 数値信号の入力方法

a. デジタルスイッチとの接続



- DC入力ユニット (JW-212N) を使用すると、1 ユニット当たりBCD 4桁の信号を読み込みます。
- デジタルスイッチはリアルコードのものを使用します。

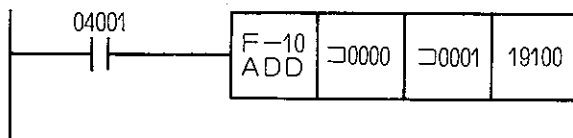
重み \ 数値	0	1	2	3	4	5	6	7	8	9
1		●		●		●		●		●
2			●	●			●	●		
4					●	●	●	●		
8									●	●

●印—スイッチON

- コンプリメンタルコードのデジタルスイッチを使用するときは、F-09 (INV命令) で反転させてください。
- 左図は2段端子台を1段で記載しています。

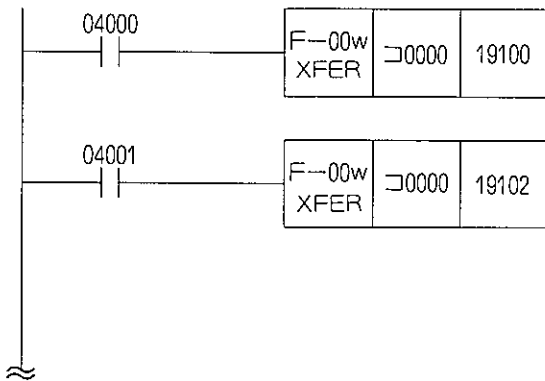
- 上記の接続で毎スキャンサイクルの入出力処理で、データメモリの入出力リレー領域に読み込みます。16ビットのデータはコXXXXの1バイト(8点)と、コ(XXXX+1)の1バイト(8点)としてデータ処理命令で直接指定できます。

(例)



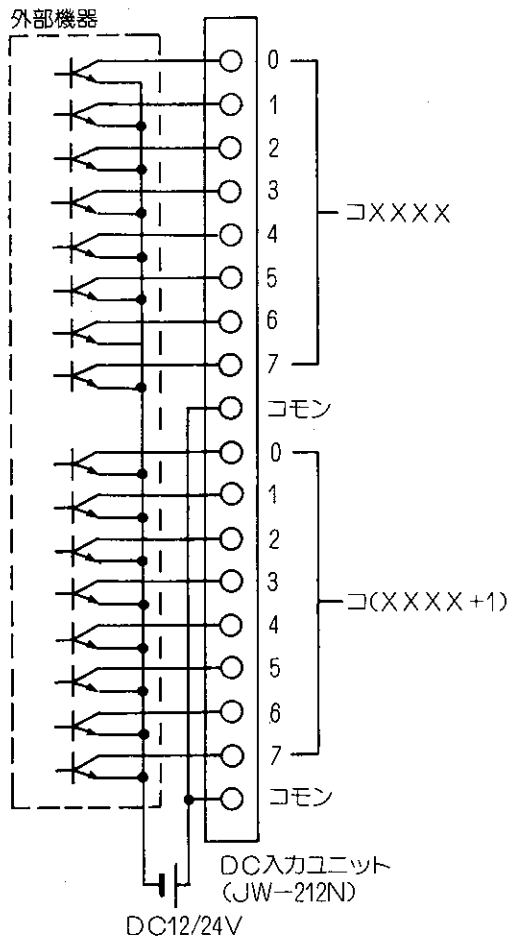
コ0000 (BCD 2桁) と、コ0001 (BCD 2桁) を加算し、レジスタ19100に格納。
また、転送命令により一旦レジスタ領域に転送後、データ処理命令にも使えます。

(例)



上記の例は、1組のデジタルスイッチで複数の設定値を読み込んでいます。

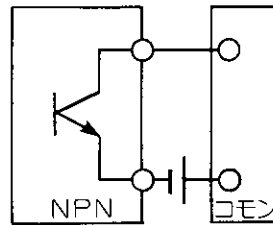
b. オープンコレクタ出力の外部機器との接続



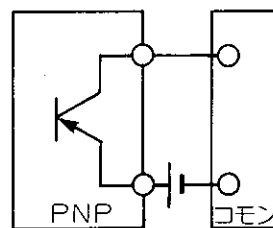
●上記の接続で毎スキャンサイクルの入出力処理で、データメモリの入出力リレー領域に読み込みます。デジタルスイッチの場合と同様にデータ処理命令で1バイト単位で使用します。

- 04000をONにするとコ0000, コ0001の2バイト(BCD4桁)を19100、19101に転送します。
- 04001をONにすると、コ0000, コ0001の2バイト(BCD4桁)を19102、19103に転送します。

- DC入力ユニット (JW-212N) を使用すると、1ユニット当たり16ビットの数値信号を読み込みます。
- 外部機器の出カトランジスタがNPNかPNPかで接続方法が異なります。



NPNトランジスタ

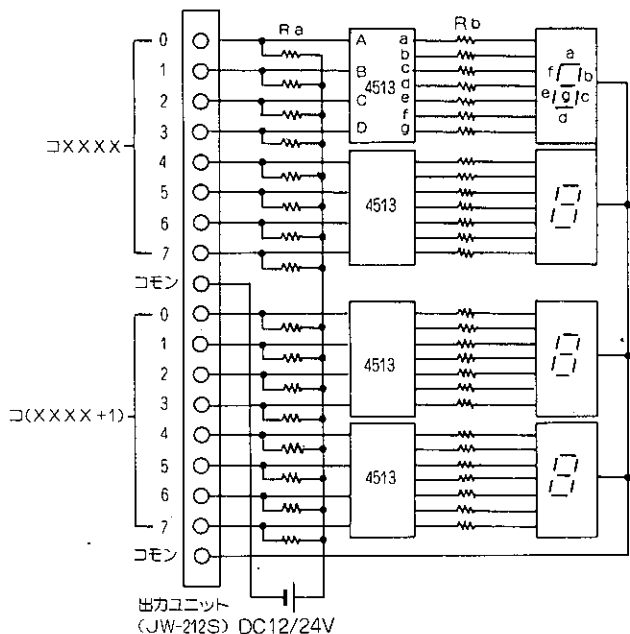


PNPトランジスタ

- JW-202N、JW-212N、JW-214N以外のDC入力ユニットは、NPN出力型の配線のみ可能です。PNP型の配線はできません。
- 左図は2段端子を1段で記載しています。

(2) 数値信号の出力方法

a. 数字表示器との接続(1)



●7セグメントLED数字表示器は、カソードコモンのも
を使用します。

●デコーダ・ドライバーICは、C-MOSMC4513相
当品を使用します。

デコーダ・ドライバーIC

V_{DD} — DC12~18V

V_{SS} — 0V

LE — 0V

RB1 — 0V

B1 — V_{DD}と同電位

LT — V_{DD}と同電位

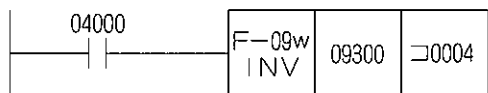
●R_aはプルアップ抵抗で5~10KΩとします。

●R_bは電流制限抵抗で、LED数字表示器のI_{FMAX}、
V_Fより算出します。

$$R_b = \frac{V_{DD} - V_F}{I_{FMAX}}$$

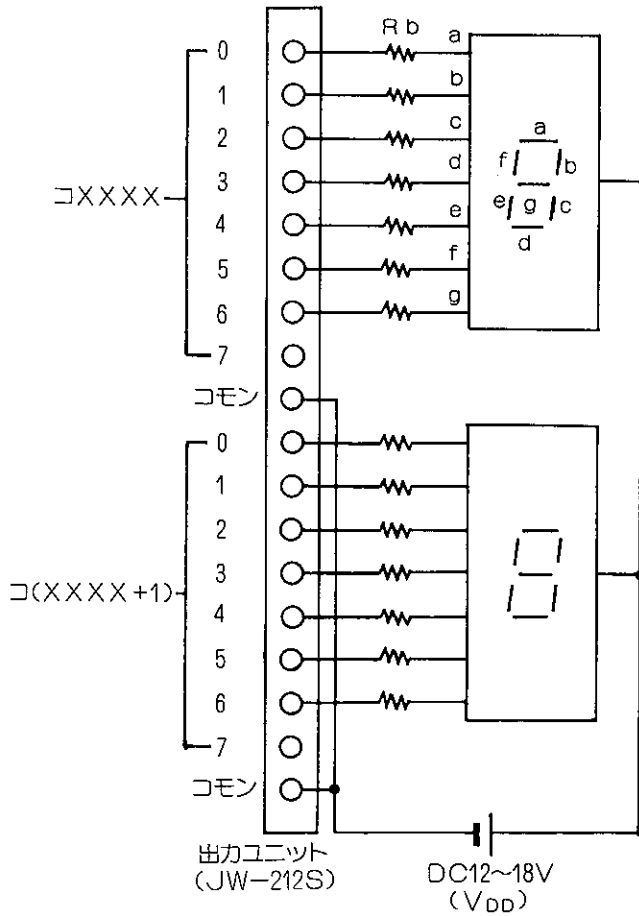
ただし4513の出力電流の制限からI_F<25mAと
してください

- 上記の表示回路は正論理で動作します。
出力するデータはF-09w(INV命令)で論理を反転
してからデータメモリの入出力リレー領域に転送する
必要があります。



- レジスタ09300, 09301の内容を論理反転し、
コ0004(数字表示器下2桁接続)、コ0005(数字
表示器上2桁接続)に格納。

b. 数字表示器との接続(F-52使用)

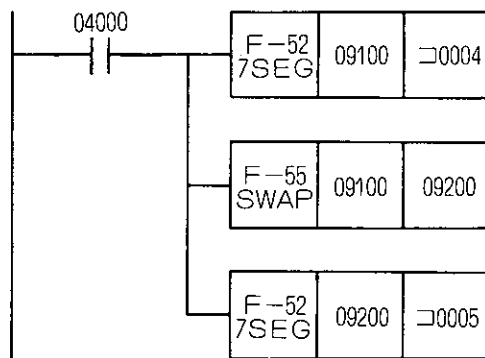


- F-52 (7SEGデコーダ命令)を使用すると、数字表示を簡単な配線で接続できます。
- 出力ユニットにJW-212Sを用いると2桁の数値が表示できます。
- 7セグメントLED数字表示器はアノードコモンのもので使用します。
- Rbは電流制限抵抗で次式で算出します。

$$R_b = \frac{V_{DD} - V_F - V_{ON}}{I_{FMAX}}$$

- V_{DD} — 電源電圧
- V_F — LED数字表示器の順電圧
- V_{ON} — 出力ユニットのON電圧(1Vで計算)

- 1バイトのBCD2桁の数値を表示する場合、次の様にプログラムします。



- レジスタ09100の下位4ビット(BCD2桁のうち下位1桁)を7セグメントデータに変換し、0004に出力
- レジスタ09100の上位4ビットと下位4ビットを交換し、レジスタ09200に格納
- レジスタ09200の下位4ビット(BCD2桁のうち上位1桁)を7セグメントデータに変換し、0005に出力

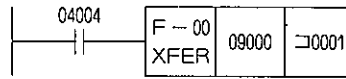
9-2 各応用命令について

**F-00
XFER**

1バイトデータの転送

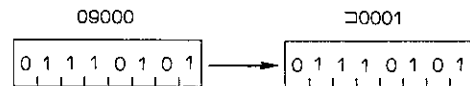
シンボル		(解説) 命令 <table border="1"> <tr><td>STR</td><td>04004</td></tr> <tr><td>F-00</td><td>09000</td></tr> <tr><td></td><td>C0001</td></tr> </table>	STR	04004	F-00	09000		C0001
STR	04004							
F-00	09000							
	C0001							
機能	レジスタSの内容(1バイトデータ)をレジスタDに転送する。							
演算内容	S→D							
Sの使用範囲	C0000~C1577 : @C0000~@C1574 C2000~C7577 : @C2000~@C7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 771#1 000000~037777 : 771#1 @000000~@037774							
Dの使用範囲	C0000~C1577 : @C0000~@C1574 C2000~C7577 : @C2000~@C7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 771#1 000000~037777 : 771#1 @000000~@037774							
演算条件	入力信号の立上り(OFF→ON)							
演算後	Sの内容	不変						
	Dの内容	レジスタSの内容						
	フラグ	不変						

(解説)



命令	
STR	04004
F-00	09000
	C0001

入力条件04004がOFF→ONの変化時に、レジスタ09000の内容をレジスタC0001に転送します。



- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

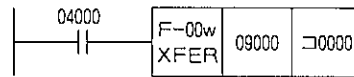
参考 下記のF命令は働きが類似しています。
 F-00、F-00w、F-00d、F-70、F-70w、
 F-74、F-74w、F-76、F-76w

**F-00w
XFER**

1ワードデータの転送

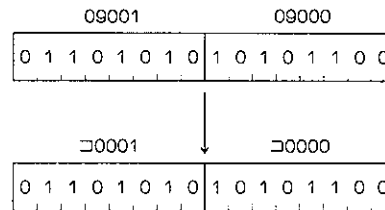
シンボル	<table border="1"> <tr> <td>F-00w</td> <td>S</td> <td>D</td> </tr> <tr> <td>XFER</td> <td></td> <td></td> </tr> </table>			F-00w	S	D	XFER		
F-00w	S	D							
XFER									
機能	レジスタS、S+1の内容(1ワードデータ)をレジスタD、D+1に転送する。								
演算内容	S、S+1→D、D+1								
Sの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 771M1 000000~037776 771M1 @000000~@037774								
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 771M1 000000~037776 771M1 @000000~@037774								
演算条件	入力信号の立上り(OFF→ON)								
演算後	S、S+1の内容	不変							
	Dの内容	レジスタSの内容							
	D+1の内容	レジスタS+1の内容							
	フラグ	不変							

(解説)



命令	
STR	04000
F-00w	09000
	コ0000

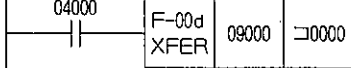
入力条件04000がOFF→ONの変化時に、レジスタ09000、09001の内容(1ワードデータ)をレジスタコ0000、コ0001に転送します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- 参考 下記のF命令は働きが類似しています。
F-00、F-00w、F-00d、F-70、F-70w、
F-74、F-74w、F-76、F-76w

F-00d
XFER

2ワードデータの転送

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-00d</td> <td style="padding: 2px;">S</td> <td style="padding: 2px;">D</td> </tr> </table>		F-00d	S	D	(解説)	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td colspan="2" style="text-align: center;">命令</td> </tr> <tr> <td style="padding: 2px;">STR</td> <td style="padding: 2px;">04000</td> </tr> <tr> <td style="padding: 2px;">F-00d</td> <td style="padding: 2px;">09000</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">C0000</td> </tr> </table>	命令		STR	04000	F-00d	09000		C0000																								
F-00d	S	D																																					
命令																																							
STR	04000																																						
F-00d	09000																																						
	C0000																																						
機能	レジスタS~S+3の内容(2ワードデータ)をレジスタD~D+3に転送する。																																						
演算内容	S~S+3→D~D+3		入力条件04000がOFF→ONの変化時に、レジスタ09000~09003の内容(2ワードデータ)をレジスタC0000~C0003に転送します。																																				
Sの使用範囲	<table border="0" style="font-family: monospace; font-size: small;"> <tr><td>C0000~C01574</td><td>:@C0000~@C01574</td></tr> <tr><td>C2000~C07574</td><td>:@C2000~@C07574</td></tr> <tr><td>b0000~b1774</td><td>:@b0000~@b1774</td></tr> <tr><td>b2000~b3774</td><td>:@b2000~@b3774</td></tr> <tr><td>09000~99774</td><td>:@09000~@99774</td></tr> <tr><td>E0000~E7774</td><td>:@E0000~@E7774</td></tr> <tr><td>774M1 000000~037774</td><td>774M1 @000000~@037774</td></tr> </table>	C0000~C01574	:@C0000~@C01574	C2000~C07574	:@C2000~@C07574	b0000~b1774	:@b0000~@b1774	b2000~b3774	:@b2000~@b3774	09000~99774	:@09000~@99774	E0000~E7774	:@E0000~@E7774	774M1 000000~037774	774M1 @000000~@037774	<table border="0" style="font-family: monospace; font-size: small;"> <tr><td>@C0000~@C01574</td><td></td></tr> <tr><td>@C2000~@C07574</td><td></td></tr> <tr><td>@b0000~@b1774</td><td></td></tr> <tr><td>@b2000~@b3774</td><td></td></tr> <tr><td>@09000~@99774</td><td></td></tr> <tr><td>@E0000~@E7774</td><td></td></tr> <tr><td>@000000~@037774</td><td></td></tr> </table>	@C0000~@C01574		@C2000~@C07574		@b0000~@b1774		@b2000~@b3774		@09000~@99774		@E0000~@E7774		@000000~@037774		<table border="1" style="font-family: monospace; font-size: small; text-align: center;"> <tr> <td style="width: 25%;">09003</td> <td style="width: 25%;">09002</td> <td style="width: 25%;">09001</td> <td style="width: 25%;">09000</td> </tr> <tr> <td>01110110</td> <td>011101010</td> <td>101011100</td> <td>001000001</td> </tr> </table>	09003	09002	09001	09000	01110110	011101010	101011100	001000001
C0000~C01574	:@C0000~@C01574																																						
C2000~C07574	:@C2000~@C07574																																						
b0000~b1774	:@b0000~@b1774																																						
b2000~b3774	:@b2000~@b3774																																						
09000~99774	:@09000~@99774																																						
E0000~E7774	:@E0000~@E7774																																						
774M1 000000~037774	774M1 @000000~@037774																																						
@C0000~@C01574																																							
@C2000~@C07574																																							
@b0000~@b1774																																							
@b2000~@b3774																																							
@09000~@99774																																							
@E0000~@E7774																																							
@000000~@037774																																							
09003	09002	09001	09000																																				
01110110	011101010	101011100	001000001																																				
Dの使用範囲	<table border="0" style="font-family: monospace; font-size: small;"> <tr><td>C0000~C01574</td><td>:@C0000~@C01574</td></tr> <tr><td>C2000~C07574</td><td>:@C2000~@C07574</td></tr> <tr><td>b0000~b1774</td><td>:@b0000~@b1774</td></tr> <tr><td>b2000~b3774</td><td>:@b2000~@b3774</td></tr> <tr><td>09000~99774</td><td>:@09000~@99774</td></tr> <tr><td>E0000~E7774</td><td>:@E0000~@E7774</td></tr> <tr><td>774M1 000000~037774</td><td>774M1 @000000~@037774</td></tr> </table>	C0000~C01574	:@C0000~@C01574	C2000~C07574	:@C2000~@C07574	b0000~b1774	:@b0000~@b1774	b2000~b3774	:@b2000~@b3774	09000~99774	:@09000~@99774	E0000~E7774	:@E0000~@E7774	774M1 000000~037774	774M1 @000000~@037774	<table border="0" style="font-family: monospace; font-size: small;"> <tr><td>@C0000~@C01574</td><td></td></tr> <tr><td>@C2000~@C07574</td><td></td></tr> <tr><td>@b0000~@b1774</td><td></td></tr> <tr><td>@b2000~@b3774</td><td></td></tr> <tr><td>@09000~@99774</td><td></td></tr> <tr><td>@E0000~@E7774</td><td></td></tr> <tr><td>@000000~@037774</td><td></td></tr> </table>	@C0000~@C01574		@C2000~@C07574		@b0000~@b1774		@b2000~@b3774		@09000~@99774		@E0000~@E7774		@000000~@037774		<table border="1" style="font-family: monospace; font-size: small; text-align: center;"> <tr> <td style="width: 25%;">C0003</td> <td style="width: 25%;">C0002</td> <td style="width: 25%;">C0001</td> <td style="width: 25%;">C0000</td> </tr> <tr> <td>01110110</td> <td>011101010</td> <td>101011100</td> <td>001000001</td> </tr> </table>	C0003	C0002	C0001	C0000	01110110	011101010	101011100	001000001
C0000~C01574	:@C0000~@C01574																																						
C2000~C07574	:@C2000~@C07574																																						
b0000~b1774	:@b0000~@b1774																																						
b2000~b3774	:@b2000~@b3774																																						
09000~99774	:@09000~@99774																																						
E0000~E7774	:@E0000~@E7774																																						
774M1 000000~037774	774M1 @000000~@037774																																						
@C0000~@C01574																																							
@C2000~@C07574																																							
@b0000~@b1774																																							
@b2000~@b3774																																							
@09000~@99774																																							
@E0000~@E7774																																							
@000000~@037774																																							
C0003	C0002	C0001	C0000																																				
01110110	011101010	101011100	001000001																																				
演算条件	入力信号の立上り(OFF→ON)																																						
演算後	S~S+3の内容	不変																																					
	D~D+3の内容	レジスタS~S+3の内容																																					
	フラグ	不変																																					

- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(C0011、19003等は禁止)

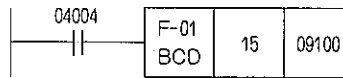
参考 下記のF命令は働きが類似しています。
 F-00、F-00w、F-00d、F-70、F-70w、
 F-74、F-74w、F-76、F-76w

**F-01
BCD**

BCD定数(2桁)の転送

シンボル		
機能	2桁のBCD定数nをレジスタDに転送する。	
演算内容	n → D	
nの使用範囲	00~99	
Dの使用範囲	コ0000~コ1577 ; @コ0000~@コ1574 コ2000~コ7577 ; @コ2000~@コ7574 b0000~b1777 ; @b0000~@b1774 b2000~b3777 ; @b2000~@b3774 09000~99777 ; @09000~@99774 E0000~E7777 ; @E0000~@E7774 777M1 000000~037777 ; 777M1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Dの内容	n(00~99)
	フラグ	不変

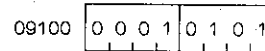
(解説)



命令	
STR	04004
F-01	15
	09100

入力条件04004がOFF→ONの変化時に、レジスタ09100にBCD定数15を転送します。

レジスタ09100は転送時、下の数値になります。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

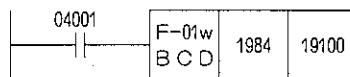
参考 下記のF命令は働きが類似しています。
F-01、F-01w、F-91

**F-01w
BCD**

BCD定数(4桁)の転送

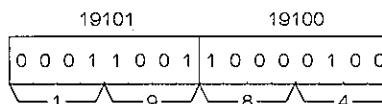
シンボル															
機能	4桁のBCD定数 n をレジスタ D、D+1 に転送する。														
演算内容	n → D、D+1														
n の使用範囲	0000~9999														
D の使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>77141 00000~03776</td> <td>77141 @00000~@03774</td> </tr> </table>	コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	77141 00000~03776	77141 @00000~@03774
コ0000~コ1576	@コ0000~@コ1574														
コ2000~コ7576	@コ2000~@コ7574														
b0000~b1776	@b0000~@b1774														
b2000~b3776	@b2000~@b3774														
09000~99776	@09000~@99774														
E0000~E7776	@E0000~@E7774														
77141 00000~03776	77141 @00000~@03774														
演算条件	入力信号の立上り(OFF→ON)														
演算後	D、D+1の内容	n													
	フラグ	不変													

(解説)



命令	
STR	04001
F-01w	1984
	19100

入力条件04001がOFF→ONの変化時に、レジスタ19100、19101にBCD定数1984を転送します。レジスタ19100、19101は転送時、下の数値になります。



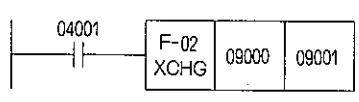
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

参考 下記のF命令は働きが類似しています。
F-01、F-01w、F-91

F-02 XCHG **1バイトデータの交換**
(eXCHAnGe)

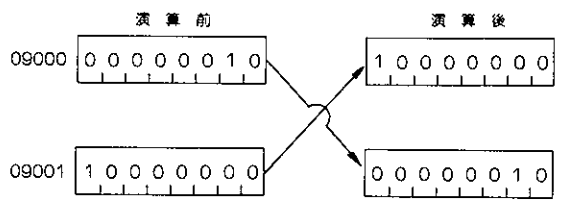
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-02</td><td style="padding: 2px;">D₁</td><td style="padding: 2px;">D₂</td></tr></table>			F-02	D ₁	D ₂
F-02	D ₁	D ₂				
機能	レジスタD ₁ の内容とレジスタD ₂ の内容を交換する。					
演算内容	D ₁ ↔ D ₂					
D ₁ の使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 774#1 000000~037777 : 774#1 @000000~@037774					
D ₂ の使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 774#1 000000~037777 : 774#1 @000000~@037774					
演算条件	入力信号の立上り(OFF→ON)					
演算後	D ₁ の内容	レジスタD ₂ の内容				
	D ₂ の内容	レジスタD ₁ の内容				
	フラグ	不変				

(解説)



命 令	
STR	04001
F-02	09000
	09001

入力条件04001がOFF→ONの変化時に、レジスタ09000の内容とレジスタ09001の内容を交換します。



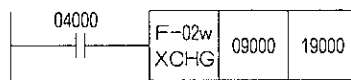
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

参考 下記のF命令は働きが類似しています。
F-02、F-02w、F-02d、F-174

F-02w XCHG 1ワードデータの交換
(eXCHanGe)

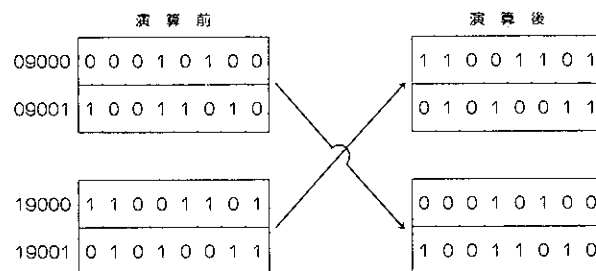
シンボル	F-02w XCHG		D ₁	D ₂
機能	レジスタD ₁ 、D ₁ +1の内容(1ワードデータ)とレジスタD ₂ 、D ₂ +1の内容(1ワードデータ)を交換する。			
演算内容	D、D ₁ +1 ↔ D ₂ 、D ₂ +1			
D ₁ の使用範囲	コ0000~コ1576 コ2000~コ7576 b0000~b1776 b2000~b3776 09000~99776 E0000~E7776 771#1 000000~037776	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774		
D ₂ の使用範囲	コ0000~コ1576 コ2000~コ7576 b0000~b1776 b2000~b3776 09000~99776 E0000~E7776 771#1 000000~037776	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774		
演算条件	入力信号の立上り(OFF→ON)			
演算後	D ₁ の内容	レジスタD ₂ の内容		
	D ₁ +1の内容	レジスタD ₂ +1の内容		
	D ₂ の内容	レジスタD ₁ の内容		
	D ₂ +1の内容	レジスタD ₁ +1の内容		
	フラグ	不変		

(解説)



命令	
STR	04000
F-02w	09000
	19000

入力条件04000がOFF→ONの変化時に、レジスタ09000、09001の内容(1ワードデータ)とレジスタ19000、19001の内容(1ワードデータ)を交換します。

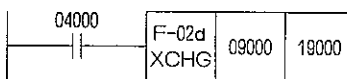


- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
 - D₁、D₂には必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- 参考** 下記のF命令は働きが類似しています。
F-02、F-02w、F-02d、F-174

F-02d 2ワードデータの交換
XCHG (eXCHanGe)

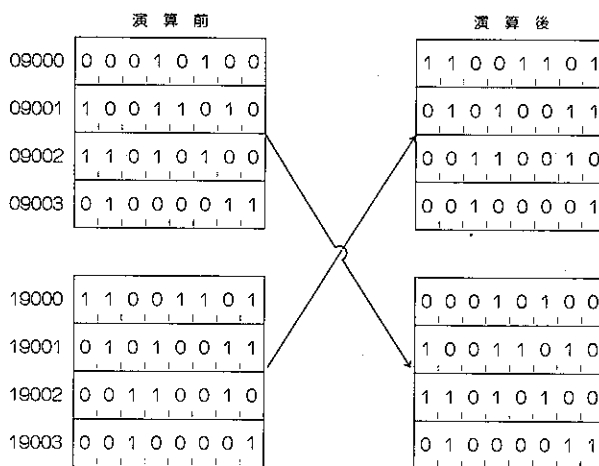
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-02d</td> <td style="padding: 2px;">D₁</td> <td style="padding: 2px;">D₂</td> </tr> <tr> <td style="padding: 2px;">XCHG</td> <td></td> <td></td> </tr> </table>		F-02d	D ₁	D ₂	XCHG		
F-02d	D ₁	D ₂						
XCHG								
機能	レジスタD ₁ ~D ₁ +3の内容(2ワードデータ)とレジスタD ₂ ~D ₂ +3の内容(2ワードデータ)を交換する。							
演算内容	D ₁ ~D ₁ +3 ↔ D ₂ ~D ₂ +3							
D ₁ の使用範囲	コ0000~コ1574 @コ0000~@コ1574 コ2000~コ7574 @コ2000~@コ7574 b0000~b1774 @b0000~@b1774 b2000~b3774 @b2000~@b3774 09000~99774 @09000~@99774 E0000~E7774 @E0000~@E7774 771#1 000000~037774 771#1 @000000~@037774							
D ₂ の使用範囲	コ0000~コ1574 @コ0000~@コ1574 コ2000~コ7574 @コ2000~@コ7574 b0000~b1774 @b0000~@b1774 b2000~b3774 @b2000~@b3774 09000~99774 @09000~@99774 E0000~E7774 @E0000~@E7774 771#1 000000~037774 771#1 @000000~@037774							
演算条件	入力信号の立上り(OFF→ON)							
演算後	D ₁ ~D ₁ +3の内容	レジスタD ₂ ~D ₂ +3の内容						
	D ₂ ~D ₂ +3の内容	レジスタD ₁ ~D ₁ +3の内容						
	フラグ	不変						

(解説)



命令	
STR	04000
F-02d	09000
	19000

入力条件04000がOFF→ONの変化時に、レジスタ09000~09003の内容(2ワードデータ)とレジスタ19000~19003の内容(2ワードデータ)を交換します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- D₁、D₂には必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

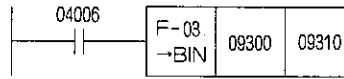
参考 下記のF命令は働きが類似しています。
 F-02、F-02w、F-02d、F-174

F-03
→BIN

BCD(2桁)→BIN(8ビット)変換

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-03</td><td>S</td><td>D</td></tr><tr><td>→BIN</td><td></td><td></td></tr></table>				F-03	S	D	→BIN		
F-03	S	D								
→BIN										
機能	レジスタSの内容(8ビットデータ)をBCDコードと見なしBinary(2進数)コードに変換して、レジスタDに格納する。									
演算内容	S→D									
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771M1 00000~03777 771M1 @00000~@03774									
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771M1 00000~03777 771M1 @00000~@03774									
演算条件	入力信号の立上り(OFF→ON)									
演算	Sの内容	不変								
	Dの内容	・演算結果 ・レジスタSの内容がBCDコードでない時不変								
後	フラグ	レジスタSの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354				
		BCDコード	0	0	0	0				
		BCDコードでない時	0	0	1	0				

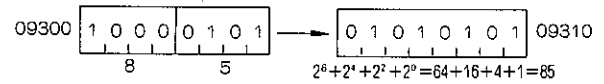
(解説)



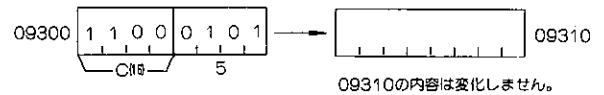
命令	
STR	04006
F-03	09300
	09310

入力条件04006がOFF→ONの変化時に、レジスタ09300の8ビットのデータをBCDコードと見なし、Binary(2進数)のコードに変換して、レジスタ09310に転送します。レジスタ09300の内容は不変です。09300の内容がBCDコード以外のおき09310の内容は変化せず、エラーフラグ(07355)が1になります。

●レジスタの内容とフラグの推移



ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
0	0	0	0



ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
0	0	1	0

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

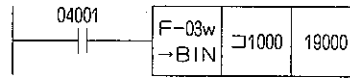
【参考】 下記のF命令は働きが類似しています。
F-03、F-03w、F-53、F-153

**F-03w
→BIN**

BCD(4桁)→BIN(16ビット)変換

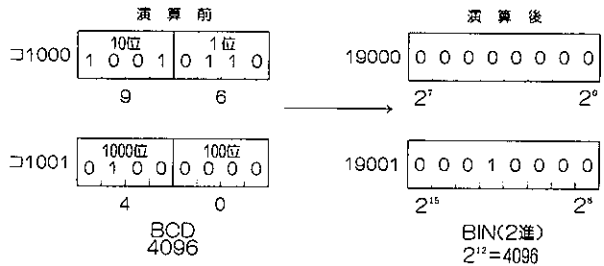
シンボル	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-03w →BIN</td> <td style="padding: 2px;">S</td> <td style="padding: 2px;">D</td> </tr> </table>					F-03w →BIN	S	D
F-03w →BIN	S	D						
機能	レジスタS、S+1の2バイトのBCD4桁データを2進に変換し、レジスタD、D+1の2バイトに格納する。							
演算内容	S、S+1→D、D+1							
Sの使用範囲	コ0000~コ1576 コ2000~コ7576 b0000~b1776 b2000~b3776 09000~99776 E0000~E7776 771M1 000000~037776		@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771M1 @000000~@037774					
Dの使用範囲	コ0000~コ1576 コ2000~コ7576 b0000~b1776 b2000~b3776 09000~99776 E0000~E7776 771M1 000000~037776		@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771M1 @000000~@037774					
演算条件	入力信号の立上り(OFF→ON)							
演算後	S、S+1の内容	不変						
	Dの内容	演算結果 (0~255)	レジスタS、S+1の内容がBCDコードでない時不変					
	D+1の内容	演算結果 (256~9999)						
フラグ	レジスタS、S+1の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354			
	BCDコード	0	0	0	0			
	BCDコードでない時	0	0	1	0			

(解説)



命 令	
STR	04001
F-03w	コ1000 19000

入力条件04001がOFF→ONの変化時に、レジスタコ1000とコ1001のBCD4桁データを2進に変換し、レジスタ19000と19001の2バイトに変換データを格納します。



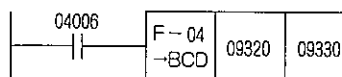
- F-53でプログラム作成するとモニタ時F-03wで表示します。
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

参考 下記のF命令は働きが類似しています。
F-03、F-03w、F-53、F-153

F-04 BIN(8ビット)→BCD(2桁)変換
→BCD

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>F-04</td><td>S</td><td>D</td></tr> <tr><td>-BCD</td><td></td><td></td></tr> </table>			F-04	S	D	-BCD		
F-04	S	D							
-BCD									
機能	レジスタSの内容(8ビットデータ)をBinary(2進数)コードと見なしBCDコードに変換してレジスタDに格納する。								
演算内容	S→D								
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774								
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774								
演算条件	入力信号の立上り(OFF→ON)								
演算後	Sの内容	不変							
	Dの内容	演算結果							
	フラグ	不変							

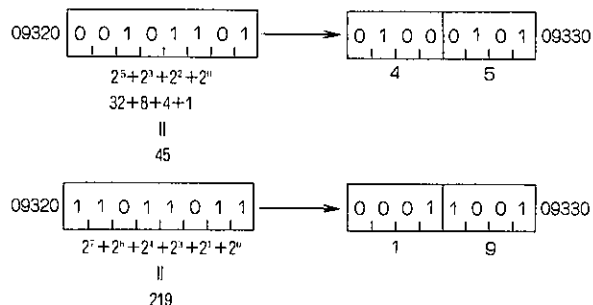
〔解説〕



命令	
STR	04006
F-04	09320
	09330

入力条件04006がOFF→ONの変化時に、レジスタ09320の8ビットのデータをBinary(2進数)のコードと見なし、BCDコードに変換してレジスタ09330に転送します。レジスタ09320の内容は不変です。

変換したBCD値が100を越える場合、100以上の数値は無視します。



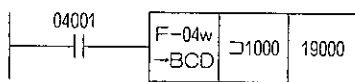
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 【参考】 下記のF命令は働きが類似しています。
F-04、F-04w、F-54、F-154

**F-04w
→BCD**

BIN(16ビット)→BCD(6桁)変換

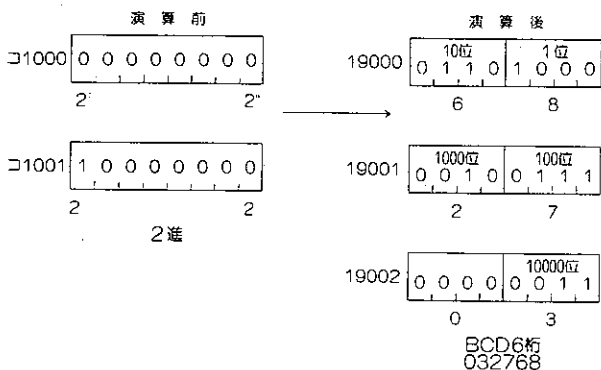
シンボル	<table border="1"> <tr> <td>F-04w →BCD</td> <td>S</td> <td>D</td> </tr> </table>			F-04w →BCD	S	D																		
F-04w →BCD	S	D																						
機能	レジスタS、S+1の2バイトの2進データをBCD6桁に変換し、レジスタD、D+1、D+2の3バイトに格納する。																							
演算内容	S、S+1→D、D+1、D+2																							
Sの使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>:</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>:</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>:</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>:</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>:</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>:</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771M1 000000~037776</td> <td>:</td> <td>771M1 @000000~@037774</td> </tr> </table>			コ0000~コ1576	:	@コ0000~@コ1574	コ2000~コ7576	:	@コ2000~@コ7574	b0000~b1776	:	@b0000~@b1774	b2000~b3776	:	@b2000~@b3774	09000~99776	:	@09000~@99774	E0000~E7776	:	@E0000~@E7774	771M1 000000~037776	:	771M1 @000000~@037774
コ0000~コ1576	:	@コ0000~@コ1574																						
コ2000~コ7576	:	@コ2000~@コ7574																						
b0000~b1776	:	@b0000~@b1774																						
b2000~b3776	:	@b2000~@b3774																						
09000~99776	:	@09000~@99774																						
E0000~E7776	:	@E0000~@E7774																						
771M1 000000~037776	:	771M1 @000000~@037774																						
Dの使用範囲	<table border="0"> <tr> <td>コ0000~コ1575</td> <td>:</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7575</td> <td>:</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1775</td> <td>:</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3775</td> <td>:</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99775</td> <td>:</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7775</td> <td>:</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771M1 000000~037775</td> <td>:</td> <td>771M1 @000000~@037774</td> </tr> </table>			コ0000~コ1575	:	@コ0000~@コ1574	コ2000~コ7575	:	@コ2000~@コ7574	b0000~b1775	:	@b0000~@b1774	b2000~b3775	:	@b2000~@b3774	09000~99775	:	@09000~@99774	E0000~E7775	:	@E0000~@E7774	771M1 000000~037775	:	771M1 @000000~@037774
コ0000~コ1575	:	@コ0000~@コ1574																						
コ2000~コ7575	:	@コ2000~@コ7574																						
b0000~b1775	:	@b0000~@b1774																						
b2000~b3775	:	@b2000~@b3774																						
09000~99775	:	@09000~@99774																						
E0000~E7775	:	@E0000~@E7774																						
771M1 000000~037775	:	771M1 @000000~@037774																						
演算条件	入力信号の立上り(OFF→ON)																							
演算後	S、S+1の内容	不変																						
	Dの内容	演算結果(1の位と10の位)																						
	D+1の内容	演算結果(100の位と1,000の位)																						
	D+2の内容	演算結果(10,000の位)																						
	フラグ	不変																						

(解説)



命 令	
STR	04001
F-04w	コ1000 コ19000

入力条件04001がOFF→ONの変化時に、レジスタコ1000とコ19001の2バイト2進データをBCD6桁に変換し、レジスタ19000から3バイトに変換データを格納します。



●F-54でプログラム作成するとモニタ時F-04wで表示します。

●コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

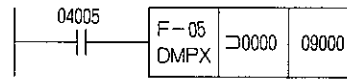
【参考】 下記のF命令は働きが類似しています。
F-04、F-04w、F-54、F-154

**F-05
DMPX**

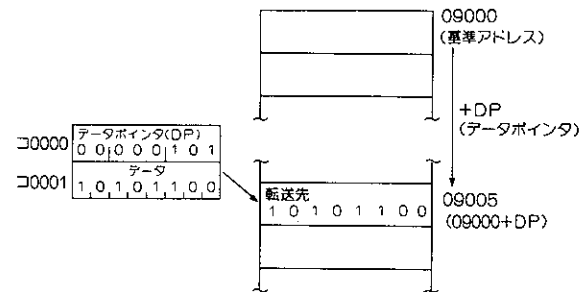
**1バイトデータの分配
(DeMultiPleXer)**

シンボル	<table border="1"> <tr> <td>F-05</td> <td>S</td> <td>D</td> </tr> <tr> <td>DMPX</td> <td></td> <td></td> </tr> </table>		F-05	S	D	DMPX		
F-05	S	D						
DMPX								
機能	レジスタS+1の内容をレジスタD(基準アドレス)からレジスタSの内容(データポインタ)だけ変位したレジスタに転送する。							
演算内容	$S+1 \rightarrow D + \langle S \rangle$ ↳データポインタ(DP) ↳基準アドレス							
Sの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 ファイル1 000000~037776 ファイル1 @000000~@037774							
Dの使用範囲	コ0000、コ0400 @コ0000、@コ0400 ----- コ7400 ----- @コ7400 b0000、b0400 @b0000、@b0400 ----- b3400 ----- @b3400 09000、09400 @09000、@09400 ----- 99400 ----- @99400 E0000、E0400 @E0000、@E0400 ----- E7400 ----- @E7400 ファイル1 000000 ファイル1 @000000 000400 @000400 037400 @037400							
演算条件	入力信号の立上り(OFF→ON)							
演算後	S, S+1の内容	不変						
	Dの内容	不変						
	D+⟨S⟩の内容	S+1のレジスタの内容						
	フラグ	不変						

〔解説〕



命 令	
STR	04005
F-05	コ0000
	09000



入力条件04005がOFF→ONの変化時に、以下の転送をします。
 コ0000+1すなわちコ0001にあるデータを、基準アドレス09000からデータポインタ、コ0000の内容(005₈)だけ変位したアドレス09005に転送します。
 データポインタは、8進数で000から377迄の値を取ります。従って、上記の例では、基準アドレスを09000とすると、データポインタを変えることにより、09000~09377の番地にデータの分配ができます。

- コ0734~コ0737は特殊領域です。
 (2・5ページ「特殊リレー」参照)
 (1)Sをコ0733とすると、S+1がコ0734になり、特殊領域に入ってしまいます。
 (2)Dにコ0400を使用するとき、Sの内容(データポインタ)を333~336に設定しないでください。
- D(基準アドレス)にブロックの先頭アドレス以外のアドレスもプログラム上設定できますが、PCの演算ではそのアドレスの含まれるブロックの先頭アドレスを基準アドレスとして処理します。

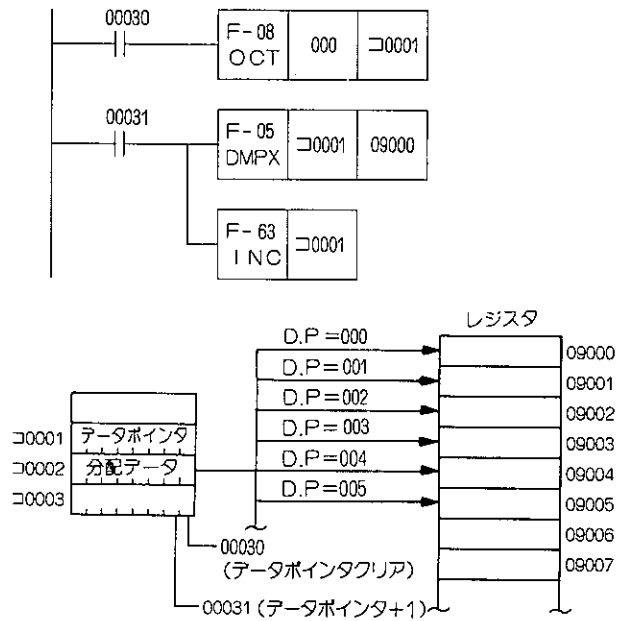
(例)

Dの設定	演算上の基準アドレス
コ0050	コ0000
b0210	b0000
09105	09000
033210	033000

9・13ページ「データメモリのブロックと基準アドレス」を参照してください。

〔参考〕 下記のF命令は働きが類似しています。
 F-05、F-05w、F-73、F-73w

【参考】 データポインタを変化させ、分配先を移動させる
プログラム例



①00030をOFF→ONにするとコ0001に8進定数000を転送します。(データポインタ000)

②00031をOFF→ONにすると、コ0002の内容を09000+000=09000に転送します。コ0001の内容はF-63により+1され001となります。

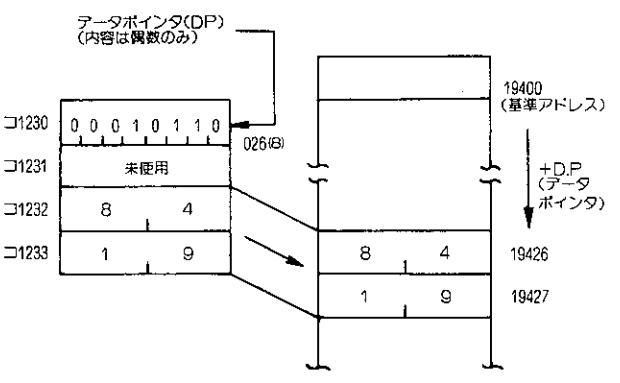
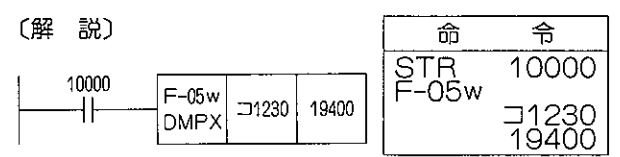
③00031を再びOFF→ONにすると、コ0002の内容を09000+001=09001に転送します。コ0001の内容はF-63により+1され002となります。

以後これと同様にして09377までのレジスタにコ0002の内容を分配します。

**F-05w
DMPX**

**1ワードデータの分配
(DeMultiPleXer)**

シンボル		
機能	レジスタS+2、S+3の内容をレジスタD(基準アドレス)からレジスタSの内容(データポイント)だけ変位したレジスタからの2バイトに転送する。	
演算内容	$S+2, S+3 \rightarrow D + \langle S \rangle, D + \langle S \rangle + 1$ 	
Sの使用範囲	コ0000~コ1574 @コ0000~@コ1574 コ2000~コ7574 @コ2000~@コ7574 b0000~b1774 @b0000~@b1774 b2000~b3774 @b2000~@b3774 09000~09774 @09000~@09774 E0000~E7774 @E0000~@E7774 ファイル1 000000~037774 ファイル1 @000000~@037774	
Dの使用範囲	コ0000、コ0400 @コ0000、@コ0400 ----- コ7400 ----- @コ7400 b0000、b0400 @b0000、@b0400 ----- b3400 ----- @b3400 09000、09400 @09000、@09400 ----- 99400 ----- @99400 E0000、E0400 @E0000、@E0400 ----- E7400 ----- @E7400 ファイル1 000000 ファイル1 @000000 000400 @000400 037400 @037400	
演算条件	入力信号の立上り(OFF→ON)	
演算後	S、S+1、S+2、S+3の内容	不変
	Dの内容	不変
	D+⟨S⟩の内容	S+2のレジスタの内容
	D+⟨S⟩+1の内容	S+3のレジスタの内容
	フラグ	不変



入力条件10000がOFF→ONの変化時に、以下の転送をします。
 コ1230+2、コ1230+3すなわちコ1232、コ1233にあるデータを基準アドレス19400からデータポインタコ1230の内容(026h)だけ変位したアドレス19426からの2バイトに転送します。
 コ1230の内容(データポインタ)は、ワードアドレスを設定します。従ってコ1230の内容は偶数を設定してください。(000~376)

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Sには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

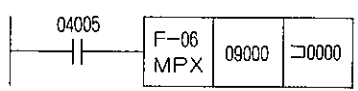
参考 下記のF命令は働きが類似しています。
 F-05、F-05w、F-72、F-72w

**F-06
MPX**

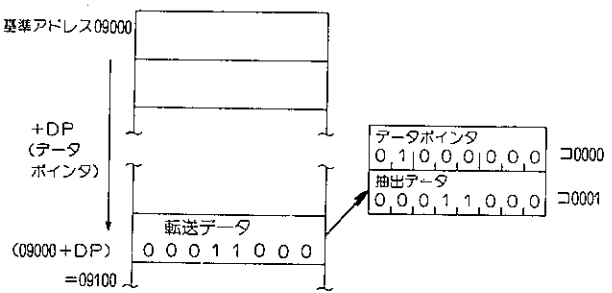
**1バイトデータの抽出
(MultiPlexer)**

シンボル	<table border="1"><tr><td>F-06</td><td>S</td><td>D</td></tr></table>	F-06	S	D
F-06	S	D		
機能	レジスタS(基準アドレス)からレジスタDの内容(データポインタ)だけ変位したレジスタの内容をレジスタD+1に転送する。			
演算内容	$S + \langle D \rangle \rightarrow D + 1$ ↳データポインタ(DP) ↳基準アドレス			
Sの使用範囲	コ0000、コ0400 @コ0000、@コ0400 ----- コ7400 ----- @コ7400 b0000、b0400 @b0000、@b0400 ----- b3400 ----- @b3400 09000、09400 @09000、@09400 ----- 99400 ----- @99400 E0000、E0400 @E0000、@E0400 ----- E7400 ----- @E7400 ファイル1 000000 ファイル1 @000000 000400 @000400 037400 @037400			
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 ファイル1 000000~037776 ファイル1 @000000~@037774			
演算条件	入力信号の立上り(OFF→ON)			
演算後	Sの内容	不変		
	Dの内容	不変(データポインタ)		
	D+1の内容	S+⟨D⟩のレジスタの内容		
	フラグ	不変		

(解説)



命 令	
STR	04005
F-06	09000 コ0000



入力条件04005がOFF→ONの変化時に、以下の転送をします。
 基準アドレス09000からコ0000のデータポインタの内容(100₈)だけ変位したアドレス09100の内容を、コ0000+1(コ0001)に転送します。
 データポインタは、8進数で000から377の値をとります。従って、データポインタを変えることにより、09000~09377の番地からデータの抽出ができます。

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
 (1)Dをコ0733とすると、D+1はコ0734になり、特殊領域に入ってしまう。
 (2)Sにコ0400を使用するとき、Dの内容(データポインタ)を333~336に設定しないでください。
- S(基準アドレス)にブロックの先頭アドレス以外のアドレスもプログラム上設定できますが、PCの演算ではそのアドレスの含まれるブロックの先頭アドレスを基準アドレスとして処理します。

(例)

Sの設定	演算上の基準アドレス
コ0051	コ0000
b0106	b0000
09023	09000
031257	031000

9・13ページ「データメモリのブロックと基準アドレス」を参照してください。

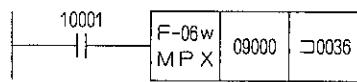
参考 下記のF命令は働きが類似しています。
 F-06、F-06w、F-73、F-73w

F-06w
MPX

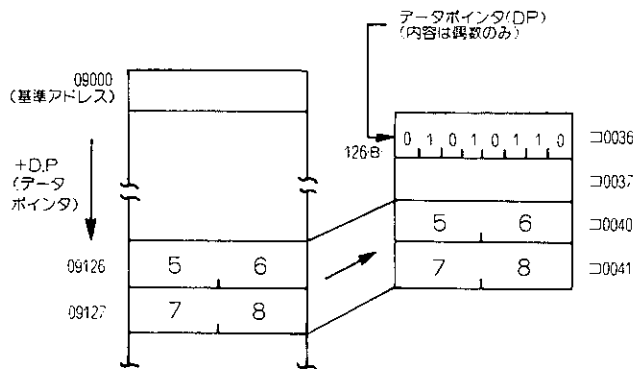
1ワードデータの抽出 (MultiPlexer)

シンボル	F-06w MPX		S	D
機能	レジスタS(基準アドレス)からレジスタDの内容(データポインタ)だけ変位したレジスタからの2バイトの内容をレジスタD+2、D+3に転送する。			
演算内容	$S+(D), S+(D)+1 \rightarrow D+2, D+3$ テータポインタ(DP) 基準アドレス			
Sの使用範囲	コ0000、コ0400 ----- コ7400 b0000、b0400 ----- b3400 09000、09400 ----- 99400 E0000、E0400 ----- E7400 ファイル1 000000 000400 ----- 037400	@コ0000、@コ0400 ----- @コ7400 @b0000、@b0400 ----- @b3400 @09000、@09400 ----- @99400 @E0000、@E0400 ----- @E7400 ファイル1 @000000 @000400 ----- @037400		
Dの使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 ファイル1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 ファイル1 @000000~@037774		
演算条件	入力信号の立上り(OFF→ON)			
演算後	Sの内容	不変		
	D、D+1の内容	不変		
	D+2の内容	S+(D)のレジスタの内容		
	D+3の内容	S+(D)+1のレジスタの内容		
	フラグ	不変		

(解説)



命令	
STR	10001
F-06w	
	09000
	コ0036



入力条件10001がOFF→ONの変化時に以下の転送をします。

基準アドレス09000からコ0036のデータポインタの内容(126bit)だけ変位したアドレス09126と09127の内容をコ0040(コ0036+2)、コ0041(コ0036+3)に転送します。

コ0036の内容はワードアドレスを設定します。従って、偶数を設定してください。(000~376)

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

参考 下記のF命令は働きが類似しています。
F-06、F-06w、F-73、F-73w

**F-07
DCML**

10進定数(1バイト)の転送

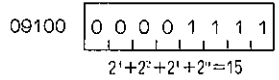
シンボル		
機能	10進定数 n をレジスタ D に転送する。	
演算内容	n → D	
n の使用範囲	000~255	
D の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 00000~03777 771#1 @00000~@03774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Dの内容	n (000~255)
	フラグ	不変

(解説)



命 令	
STR	04004
F-07	015 09100

入力条件04004がOFF→ONの変化時に、レジスタ09100に10進定数15を転送します。レジスタ09100は転送時、バイナリコードで下の数値になります。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

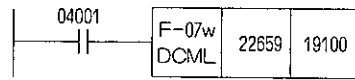
参考 下記のF命令は働きが類似しています。
F-07、F-07w、F-97

**F-07w
DCML**

**10進定数(1ワード)の転送
(DeCiMaL)**

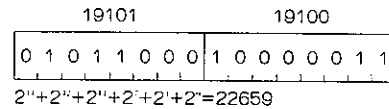
シンボル		
機能	10進定数nをレジスタD、D+1に転送する。	
演算内容	n → D、D+1	
nの使用範囲	00000~65535	
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 771#1 00000~03776 771#1 @00000~@03774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	D、D+1の内容	n
	フラグ	不変

(解説)



命令	
STR	04001
F-07w	22659 19100

入力条件04001がOFF→ONの変化時に、レジスタ19100、19101に10進定数22659を転送します。レジスタ19100、19101は転送時、下の数値になります。



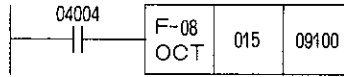
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
 - Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- 参考** 下記のF命令は働きが類似しています。
F-07、F-07w、F-97

**F-08
OCT**

8進定数(1バイト)の転送

シンボル		
機能	8進定数 n をレジスタ D に転送する。	
演算内容	n → D	
n の使用範囲	000~377(8)	
D の使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 771#1 000000~037777 : 771#1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Dの内容	n(000~377)
	フラグ	不変

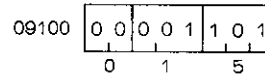
(解説)



命令	
STR	04004
F-08	015 09100

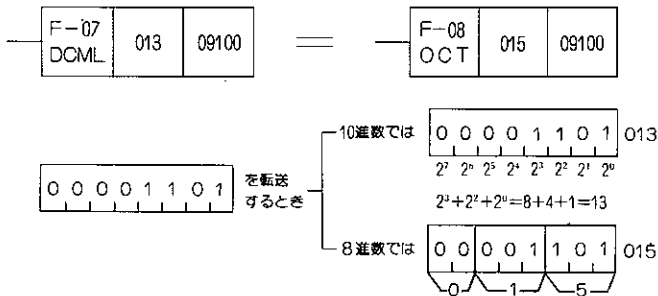
入力条件04004がOFF→ONの変化時に、レジスタ09100に8進定数015を転送します。

レジスタ09100は転送時、下の数値になります。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

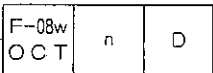
参考 F-07(10進定数の転送)とF-08(8進定数の転送)は、プログラム上10進数、8進数を用いる違いはありますが、転送後のレジスタの内容はともにバイナリコードとなります。



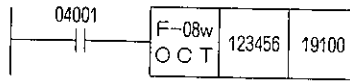
F-08は、F-05(分配)、F-06(抽出)等のデータポイントのプリセット等に使用するとデータメモリのアドレス(8進数)が直感的に把握できます。

参考 下記のF命令は働きが類似しています。
F-08、F-08w、F-71、F-71w

F-08w
OCT **8進定数(1ワード)の転送**
(OCTal)

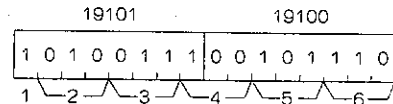
シンボル		
機能	8進定数 n をレジスタ D、D+1 に転送する。	
演算内容	n → D、D+1	
n の使用範囲	000000~177777	
D の使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 ファイル1 000000~037776 : ファイル1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	D、D+1の内容	n
	フラグ	不変

(解説)



命令	
STR	04001
F-08w	123456 19100

入力条件04001がOFF→ONの変化時に、レジスタ19100、19101に8進定数123456を転送します。レジスタ19100、19101は転送時、下の数値になります。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

参考 下記のF命令は働きが類似しています。

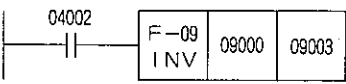
F-08、F-08w、F-71、F-71w

**F-09
INV**

**8ビットデータの反転
(INVerter)**

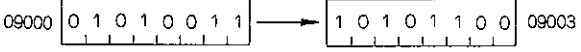
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-09</td><td>S</td><td>D</td></tr><tr><td>INV</td><td></td><td></td></tr></table>			F-09	S	D	INV		
F-09	S	D							
INV									
機能	レジスタSの内容を反転してレジスタDに格納する。								
演算内容	S→D								
Sの使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~99777 @090000~@99774 E00000~E7777 @E00000~@E7774 771#1 000000~037777 771#1 @000000~@037774								
Dの使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~99777 @090000~@99774 E00000~E7777 @E00000~@E7774 771#1 000000~037777 771#1 @000000~@037774								
演算条件	入力信号の立上り(OFF→ON)								
演算後	Sの内容	不変							
	Dの内容	レジスタSの内容の反転データ							
	フラグ	不変							

(解説)



命令	
STR	04002
F-09	09000
	09003

入力条件04002がOFF→ONの変化時に、レジスタ09000の8ビットの内容を反転させ、その内容をレジスタ09003に格納します。
レジスタ09000の内容は不変です。



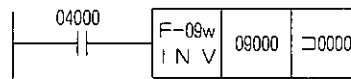
●コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

**F-09w
INV**

**16ビットデータの反転
(INVerter)**

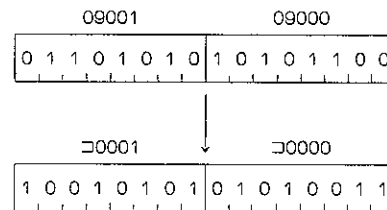
シンボル															
機能	レジスタS、S+1の内容(16ビットデータ)を反転してレジスタD、D+1に格納する。														
演算内容	$\overline{S, S+1} \rightarrow D, D+1$														
Sの使用範囲	<table border="0"> <tr><td>コ0000~コ1576</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7576</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1776</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3776</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99776</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7776</td><td>@E0000~@E7774</td></tr> <tr><td>77771 000000~037776</td><td>77771 @000000~@037774</td></tr> </table>	コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	77771 000000~037776	77771 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574														
コ2000~コ7576	@コ2000~@コ7574														
b0000~b1776	@b0000~@b1774														
b2000~b3776	@b2000~@b3774														
09000~99776	@09000~@99774														
E0000~E7776	@E0000~@E7774														
77771 000000~037776	77771 @000000~@037774														
Dの使用範囲	<table border="0"> <tr><td>コ0000~コ1576</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7576</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1776</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3776</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99776</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7776</td><td>@E0000~@E7774</td></tr> <tr><td>77771 000000~037776</td><td>77771 @000000~@037774</td></tr> </table>	コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	77771 000000~037776	77771 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574														
コ2000~コ7576	@コ2000~@コ7574														
b0000~b1776	@b0000~@b1774														
b2000~b3776	@b2000~@b3774														
09000~99776	@09000~@99774														
E0000~E7776	@E0000~@E7774														
77771 000000~037776	77771 @000000~@037774														
演算条件	入力信号の立上り(OFF→ON)														
演算後	S、S+1の内容	不変													
	Dの内容	レジスタSの内容の反転データ													
	D+1の内容	レジスタS+1の内容の反転データ													
	フラグ	不変													

(解説)



命 令	
STR	04000
F-09w	09000 コ0000

入力条件04000がOFF→ONの変化時にレジスタ09000、09001の16ビットの内容を反転させ、その内容をレジスタコ0000、コ0001に格納します。レジスタ09000、09001の内容は不変です。



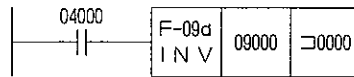
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

**F-09d
INV**

**32ビットデータの反転
(INVerter)**

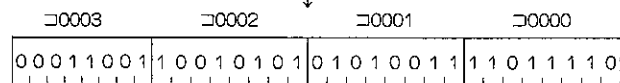
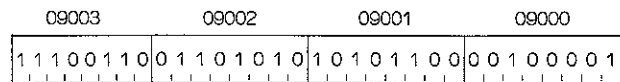
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-09d</td><td style="padding: 2px;">S</td><td style="padding: 2px;">D</td></tr></table>		F-09d	S	D											
F-09d	S	D														
機能	レジスタS~S+3の内容(32ビットデータ)を反転してレジスタD~D+3に格納する。															
演算内容	$\overline{S \sim S+3} \rightarrow D \sim D+3$															
Sの使用範囲	<table style="font-family: monospace; font-size: small;"> <tr><td>コ0000~コ1574</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7574</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1774</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3774</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99774</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7774</td><td>@E0000~@E7774</td></tr> <tr><td>771M1 000000~037774</td><td>771M1 @000000~@037774</td></tr> </table>		コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	771M1 000000~037774	771M1 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574															
コ2000~コ7574	@コ2000~@コ7574															
b0000~b1774	@b0000~@b1774															
b2000~b3774	@b2000~@b3774															
09000~99774	@09000~@99774															
E0000~E7774	@E0000~@E7774															
771M1 000000~037774	771M1 @000000~@037774															
Dの使用範囲	<table style="font-family: monospace; font-size: small;"> <tr><td>コ0000~コ1574</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7574</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1774</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3774</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99774</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7774</td><td>@E0000~@E7774</td></tr> <tr><td>771M1 000000~037774</td><td>771M1 @000000~@037774</td></tr> </table>		コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	771M1 000000~037774	771M1 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574															
コ2000~コ7574	@コ2000~@コ7574															
b0000~b1774	@b0000~@b1774															
b2000~b3774	@b2000~@b3774															
09000~99774	@09000~@99774															
E0000~E7774	@E0000~@E7774															
771M1 000000~037774	771M1 @000000~@037774															
演算条件	入力信号の立上り(OFF→ON)															
演算後	S~S+3の内容	不変														
	D~D+3の内容	レジスタS~S+3の内容の反転データ														
	フラグ	不変														

〔解説〕



命 令	
STR	04000
F-09d	09000
	コ0000

入力条件04000がOFF→ONの変化時にレジスタ09000~09003の32ビットの内容を反転させ、その内容をレジスタコ0000~コ0003に格納します。レジスタ09000~09003の内容は不変です。



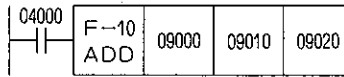
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

**F-10
ADD**

**レジスタ間(BCD2桁)の加算
(ADD)**

シンボル	<table border="1"> <tr> <td>F-10 ADD</td> <td>S₁</td> <td>S₂</td> <td>D</td> </tr> </table>				F-10 ADD	S ₁	S ₂	D
F-10 ADD	S ₁	S ₂	D					
機能	レジスタS ₁ の内容とレジスタS ₂ の内容を加算(BCD2桁加算)してレジスタDに格納する。							
演算内容	S ₁ +S ₂ →D							
S ₁ の使用範囲	コ0000~コ15777 @コ00000~@コ1574 コ2000~コ75777 @コ20000~@コ7574 b00000~b17777 @b00000~@b1774 b20000~b37777 @b20000~@b3774 090000~997777 @090000~@99774 E00000~E77777 @E00000~@E7774 771#1 000000~037777 771#1 @000000~@037774							
S ₂ の使用範囲	コ00000~コ15777 @コ00000~@コ1574 コ20000~コ75777 @コ20000~@コ7574 b00000~b17777 @b00000~@b1774 b20000~b37777 @b20000~@b3774 090000~997777 @090000~@99774 E00000~E77777 @E00000~@E7774 771#1 000000~037777 771#1 @000000~@037774							
Dの使用範囲	コ00000~コ15777 @コ00000~@コ1574 コ20000~コ75777 @コ20000~@コ7574 b00000~b17777 @b00000~@b1774 b20000~b37777 @b20000~@b3774 090000~997777 @090000~@99774 E00000~E77777 @E00000~@E7774 771#1 000000~037777 771#1 @000000~@037774							
演算条件	入力信号の立上り(OFF→ON)							
演算	S ₁ の内容	不変						
	S ₂ の内容	不変						
後	Dの内容	●演算結果(下位2桁) ●レジスタS ₁ 、S ₂ の内容がBCDコードでないとき不変						
	フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354		
		0	1	0	0	1		
		1~99	0	0	0	1		
		100	1	1	0	0		
		101以上	0	1	0	0		
		S ₁ 、S ₂ の内容がBCDコードでない時	0	0	1	0		

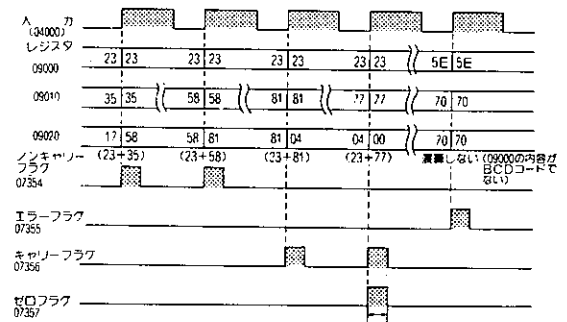
(解説)



命令	
STR	04000
F-10	09000
	09010
	09020

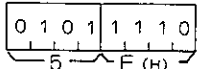
入力条件04000がOFF→ONの変化時に、レジスタ09000の内容とレジスタ09010の内容を加算して、レジスタ09020に格納します。レジスタ09000、09010の内容は不変です。

●演算結果とフラグの推移



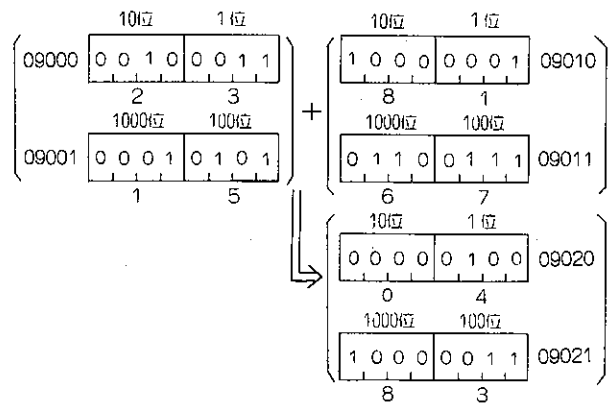
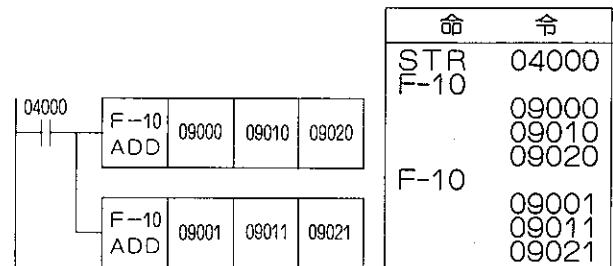
1スキャンタイム以内
プログラム中、フラグに影響を与える命令まで有効

- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクルの中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- S₁、S₂の内容がBCDコード以外の数値の場合、エラーフラグ(07355)をONし、加算は行いません。

(例) S₁ 0 1 0 1 1 1 1 0 1110はBCDでは禁止のコードです。


参考 BCDで3桁以上の加算をする場合、F-10命令を続けて設定します。

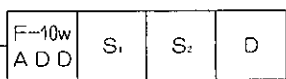
連続してF-10命令を設定すると、2つ目以降のF-10命令では、キャリーフラグ(07356)の内容も加算します。STR命令に続く最初のF-10命令はキャリーフラグ(07356)の内容を加算しません。



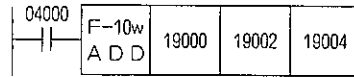
- 上記の演算は1523+6781=8304を示しています。
- 下の桁から順次プログラムしていくと、桁上げの情報が上位桁に入ってきます。
(9・10ページ「倍長演算」参照)

**F-10w
ADD**

**レジスタ間(BCD4桁)の加算
(ADD)**

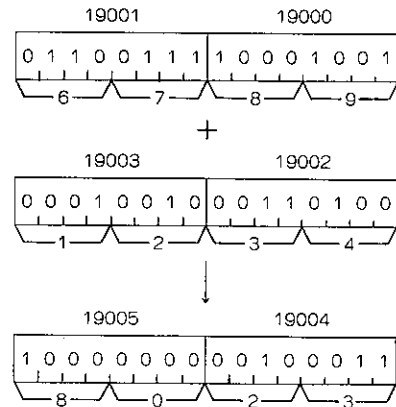
シンボル																		
機能	レジスタS ₁ 、S ₁ +1の内容とレジスタS ₂ 、S ₂ +1の内容を加算(BCD4桁加算)してレジスタD、D+1に格納する。																	
演算内容	(S ₁ 、S ₁ +1)+(S ₂ 、S ₂ +1)→D、D+1																	
S ₁ の使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771#1 000000~037776</td> <td>771#1 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771#1 000000~037776	771#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																	
コ2000~コ7576	@コ2000~@コ7574																	
b0000~b1776	@b0000~@b1774																	
b2000~b3776	@b2000~@b3774																	
09000~99776	@09000~@99774																	
E0000~E7776	@E0000~@E7774																	
771#1 000000~037776	771#1 @000000~@037774																	
S ₂ の使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771#1 000000~037776</td> <td>771#1 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771#1 000000~037776	771#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																	
コ2000~コ7576	@コ2000~@コ7574																	
b0000~b1776	@b0000~@b1774																	
b2000~b3776	@b2000~@b3774																	
09000~99776	@09000~@99774																	
E0000~E7776	@E0000~@E7774																	
771#1 000000~037776	771#1 @000000~@037774																	
Dの使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771#1 000000~037776</td> <td>771#1 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771#1 000000~037776	771#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																	
コ2000~コ7576	@コ2000~@コ7574																	
b0000~b1776	@b0000~@b1774																	
b2000~b3776	@b2000~@b3774																	
09000~99776	@09000~@99774																	
E0000~E7776	@E0000~@E7774																	
771#1 000000~037776	771#1 @000000~@037774																	
演算条件	入力信号の立上り(OFF→ON)																	
演算	S ₁ 、S ₁ +1の内容	不変																
	S ₂ 、S ₂ +1の内容	不変																
	Dの内容	演算結果(下位2桁)	S ₁ 、S ₁ +1、S ₂ 、S ₂ +1がBCDコードでない時不変															
	D+1の内容	演算結果(上位2桁)																
後フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354													
	0	1	0	0	1													
	1~9999	0	0	0	1													
	10000	1	1	0	0													
	10001以上	0	1	0	0													
BCD以外の時	0	0	1	0														

(解説)



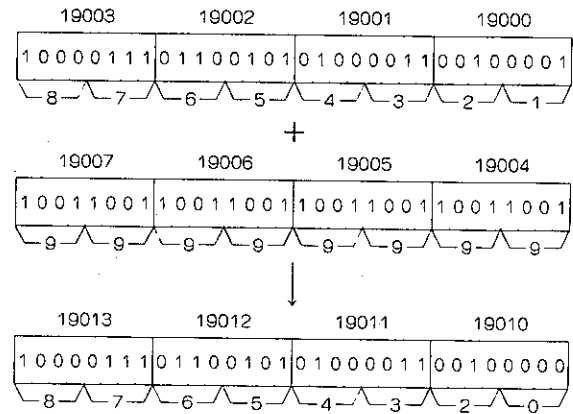
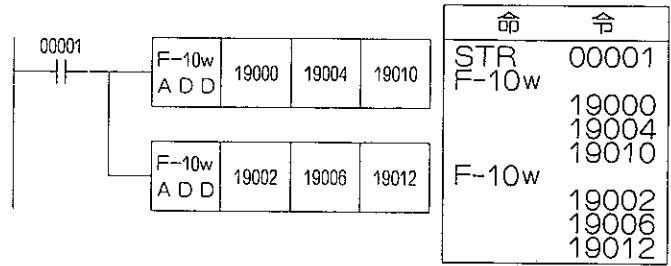
命令	
STR	04000
F-10w	19000
	19002
	19004

入力条件04000がOFF→ONの変化時に、レジスタ19000、19001の内容(BCD4桁)とレジスタ19002、19003の内容(BCD4桁)を加算してレジスタ19004、19005に格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクルの中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

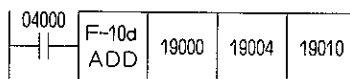
参考 F-10w命令もF-10命令と同様に倍長演算が可能です。
従ってBCD8桁以上の加算をする場合、F-10w命令を続けて設定します。



F-10d ADD レジスタ間(BCD8桁)の加算 (ADD)

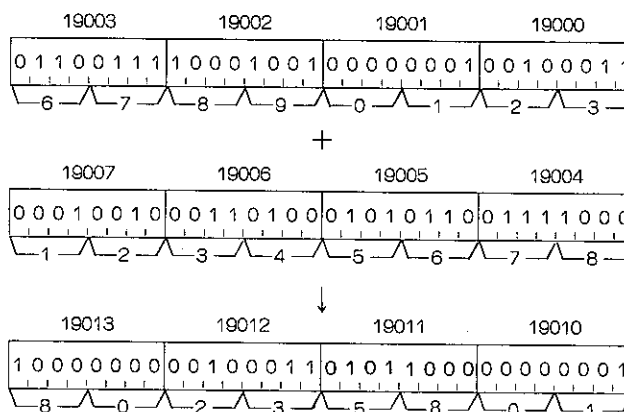
シンボル						
機能	レジスタS ₁ ~S ₁ +3の内容とレジスタS ₂ ~S ₂ +3の内容を加算(BCD8桁加算)してレジスタD~D+3に格納する。					
演算内容	(S ₁ ~S ₁ +3)+(S ₂ ~S ₂ +3)→D~D+3					
S ₁ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 ファイル000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 ファイル@000000~@037774				
S ₂ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 ファイル000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 ファイル@000000~@037774				
Dの使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 ファイル000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 ファイル@000000~@037774				
演算条件	入力信号の立上り(OFF→ON)					
演算後	S ₁ ~S ₁ +3の内容	不変				
	S ₂ ~S ₂ +3の内容	不変				
演算後	D~D+3の内容	演算結果(BCD8桁)		S ₁ ~S ₁ +3, S ₂ ~S ₂ +3がBCDコードでない時不変		
		演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
	フ	0	1	0	0	1
	ラ	1~99999999	0	0	0	1
	グ	100000000	1	1	0	0
		100000001以上	0	1	0	0
	BCD以外の時	0	0	1	0	

(解説)



命令	
STR	04000
F-10d	19000
	19004
	19010

入力条件04000がOFF→ONの変化時に、レジスタ19000~19003の内容(BCD8桁)とレジスタ19004~19007の内容(BCD8桁)を加算してレジスタ19010~19013に格納します。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- S₁、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクルの中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

- S₁~S₁+3、S₂~S₂+3の内容がBCDコード以外の数値の場合、エラーフラグ(07355)がONし、加算は行いません。(D~D+3は不変です。)

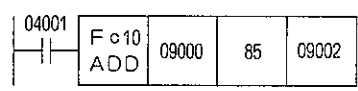
【参考】 F-10d命令もF-10W命令と同様に倍長演算が可能です。従ってBCD16桁以上の加算をする場合、F-10d命令をつけて設定できます。

**Fc10
ADD**

**レジスタ(BCD2桁)と定数(2桁)の加算
(ADD)**

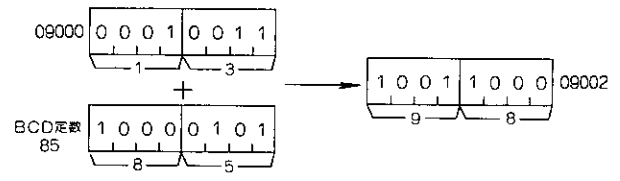
シンボル	— Fc10 ADD S _i n D				
機能	レジスタS _i の内容と2桁のBCD定数nを加算してレジスタDに格納する。				
演算内容	S _i + n → D				
S _i の使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 ファイル1 00000~03777 : ファイル1 @00000~@03774				
nの使用範囲	00~99				
Dの使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 ファイル1 00000~03777 : ファイル1 @00000~@03774				
演算条件	入力信号の立上り(OFF→ON)				
演算	S _i の内容	不変			
	Dの内容	●演算結果(下位2桁) ●レジスタS _i の内容がBCDコードでないとき不変			
後フラグ	演算結果	ゼロ	キャリー	エラー	ノンキャリー
		07357	07356	07355	07354
	0	1	0	0	1
	1~99	0	0	0	1
	100	1	1	0	0
101以上	0	1	0	0	
S _i 内容がBCDでない時	0	0	1	0	

(解説)



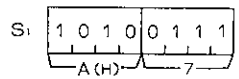
命令	
STR	04001
Fc10	09000
	85
	09002

入力条件04001がOFF→ONの変化時に、レジスタ09000の内容とBCD定数85を加算して、レジスタ09002に格納します。タイミング関係はF-10と同様です。



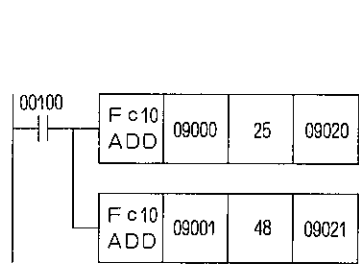
- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)
- S_iの内容がBCDコード以外の数値の場合、エラーフラグ(07355)がONし、加算は行いません。

(例)

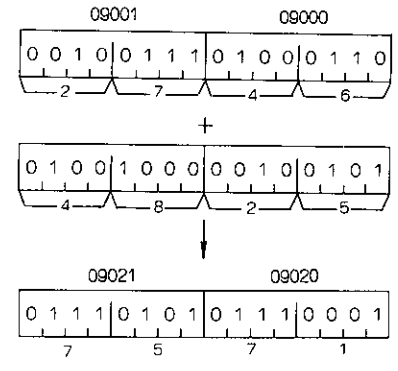


1010はBCDでは禁止のコードです。

【参考】 F-10と同様にBCD3桁以上の加算が可能です。



命令	
STR	00100
Fc10	09000
	25
	09020
Fc10	09001
	48
	09021

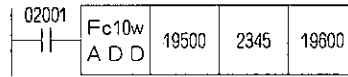


**Fc10w
ADD**

**レジスタ(BCD4桁)と定数(4桁)の加算
(ADD)**

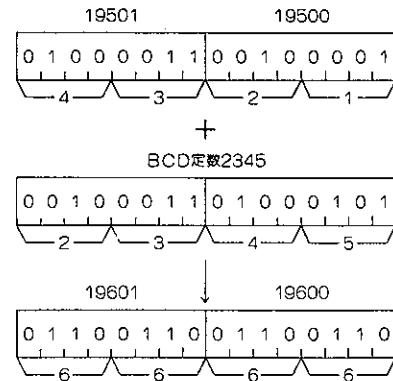
シンボル	Fc10w ADD S _i n D				
機能	レジスタS _i 、S _i +1の内容(BCD4桁)と4桁のBCD定数nを加算してレジスタD、D+1に格納する。				
演算内容	(S _i 、S _i +1)+n→D、D+1				
S _i の使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 774#1 000000~037776 774#1 @000000~@037774				
nの使用範囲	0000~9999				
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 774#1 000000~037776 774#1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	S _i 、S _i +1の内容	不変			
	Dの内容	演算結果(下位2桁)	S _i 、S _i +1の内容がBCDコードでない時不変		
	D+1の内容	演算結果(上位2桁)			
フラグ	演算結果	ゼロ	キャリー	エラー	ノキャリー
		07357	07356	07355	07354
	0	1	0	0	1
	1~9999	0	0	0	1
	10000	1	1	0	0
10000以上	0	1	0	0	
BCD以外の時	0	0	1	0	

(解説)



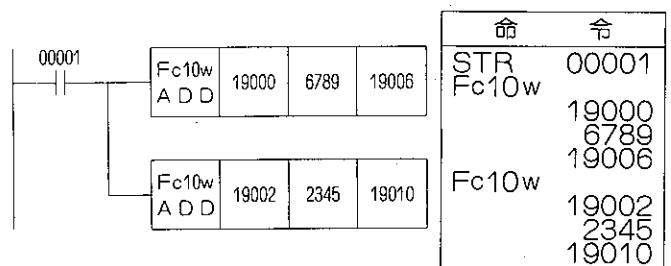
命 令	
STR	02001
Fc10w	19500
	2345
	19600

入力条件02001がOFF→ONの変化時、レジスタ19500、19501の内容(BCD4桁)とBCD定数2345を加算してレジスタ19600、19601に格納します。

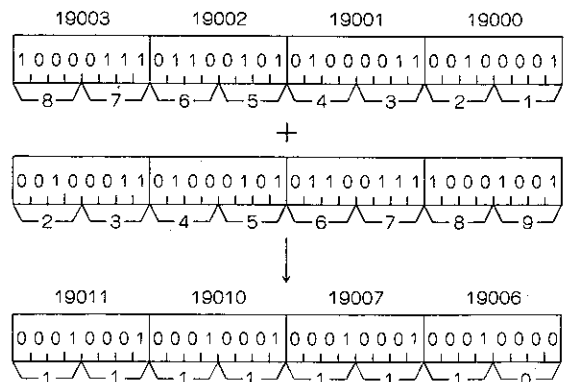


- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- S_i、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)
- S_iの内容がBCDコード以外の数値の場合エラーフラグ(07355)がONし、加算は行いません。

【参考】 Fc10w命令もFc10命令と同様に倍長演算が可能です。従ってBCD8桁以上の加算をする場合、Fc10w命令を続けて設定します。



命 令	
STR	00001
Fc10w	19000
	6789
	19006
Fc10w	19002
	2345
	19010

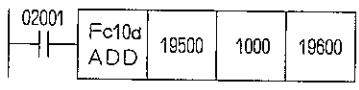


**Fc10d
ADD**

**レジスタ(BCD8桁)と定数(4桁)の加算
(ADD)**

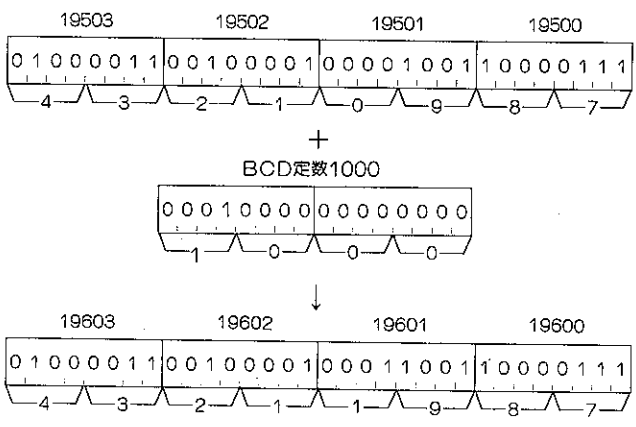
シンボル	— Fc10d ADD S _i n D			
機能	レジスタS _i ~S _i +3の内容(BCD8桁)と4桁のBCD定数nを加算してレジスタD~D+3に格納する。			
演算内容	(S _i ~S _i +3)+n→D~D+3			
S _i の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774		
nの使用範囲	0000~9999			
Dの使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774		
演算条件	入力信号の立上り(OFF→ON)			
演算後	S _i ~S _i +3の内容	不変		
	D~D+3の内容	演算結果(BCD8桁)	S _i ~S _i +3の内容がBCDコードでない時不変	
	演算結果	ゼロ 07357	キャリー 07356	エラー 07355
	0	1	0	0
	1~99999999	0	0	0
	100000000	1	1	0
	100000001以上	0	1	0
	BCD以外の時	0	0	1

(解説)



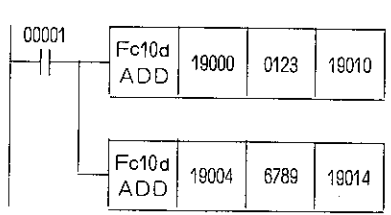
命令	
STR	02001
Fc10d	19500
	1000
	19600

入力条件02001がOFF→ONの変化時、レジスタ19500~19503の内容(BCD8桁)とBCD定数1000を加算してレジスタ19600~19603に格納します。

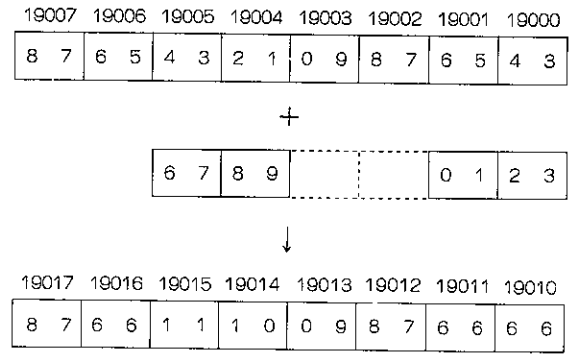


- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- S_i、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)
- S_i~S_i+3の内容がBCDコード以外の数値の場合、エラーフラグ(07355)がONし、加算は行いません。(D~D+3は不変です。)

参考 Fc10d 命令もFc10命令と同様に倍長演算が可能です。従ってBCD16桁以上の加算をする場合、Fc10d 命令を続けて設定します。



命令	
STR	00001
Fc10d	19000
	0123
	19010
Fc10d	19004
	6789
	19014



**F-11
SUB**

**レジスタ間(BCD2桁)の減算
(SUBtract)**

シンボル	<table border="1"> <tr> <td>F-11 SUB</td> <td>S₁</td> <td>S₂</td> <td>D</td> </tr> </table>				F-11 SUB	S ₁	S ₂	D																					
F-11 SUB	S ₁	S ₂	D																										
機能	レジスタS ₁ の内容からレジスタS ₂ の内容を減算(BCD2桁減算)してレジスタDに格納する。																												
演算内容	S ₁ - S ₂ → D																												
S ₁ の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771M1 000000~037777 771M1 @000000~@037774																												
S ₂ の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771M1 000000~037777 771M1 @000000~@037774																												
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771M1 000000~037777 771M1 @000000~@037774																												
演算条件	入力信号の立上り(OFF→ON)																												
演算	S ₁ の内容	不変																											
	S ₂ の内容	不変																											
後	Dの内容	● 演算結果 ● レジスタS ₁ 、S ₂ の内容がBCDコードでないとき不変																											
	フラグ	<table border="1"> <tr> <th>演算結果</th> <th>ゼロ 07357</th> <th>キャリー 07356</th> <th>エラー 07355</th> <th>ノキャリー 07354</th> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>1~99</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>負の数値</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>S₁、S₂がBCDでない時</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> </table>	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354	0	1	0	0	1	1~99	0	0	0	1	負の数値	0	1	0	0	S ₁ 、S ₂ がBCDでない時	0	0	1	0		
演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354																									
0	1	0	0	1																									
1~99	0	0	0	1																									
負の数値	0	1	0	0																									
S ₁ 、S ₂ がBCDでない時	0	0	1	0																									

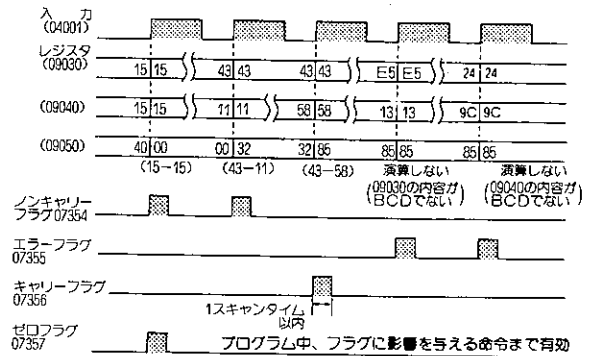
(解説)



命令	
STR	04001
F-11	09030
	09040
	09050

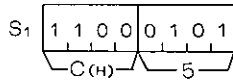
入力条件04001がOFF→ONの変化時に、レジスタ09030の内容から、レジスタ09040の内容を減算して、レジスタ09050に格納します。レジスタ09030、09040の内容は不変です。

演算結果とフラグの推移



- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- (S₁の内容)×(S₂の内容)の演算を行うと、答は100の補数で得られます。
(例) 23-85=-62は、62の100の補数38が答となります。
(123-85=38と考えてください)

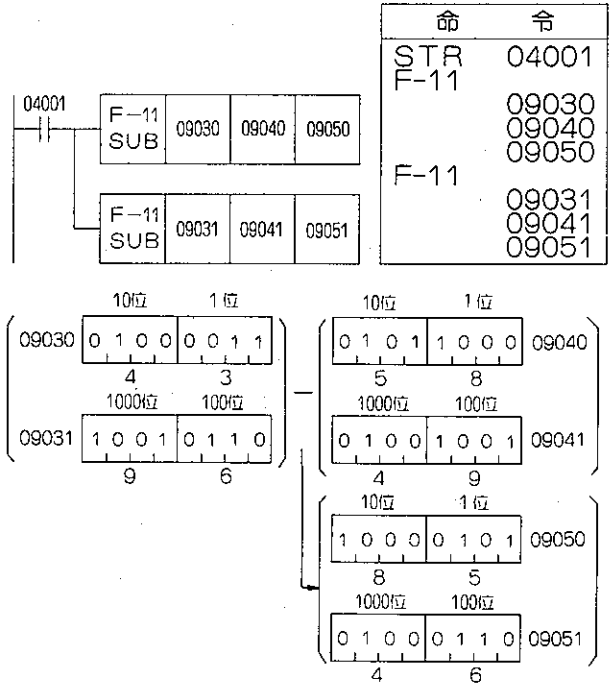
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- S₁、S₂の内容がBCDコード以外の数値の場合、エラーフラグ(07355)がONし、減算は行いません。
(Dの内容は不変です。)



1100はBCDでは禁止のコードです。

参考 3桁以上のBCD減算する場合、F-11命令を続けて設定します。

連続して、F-11命令を設定すると、2つ目以降のF-11命令では、キャリーフラグ(07356)の内容も減算します。STR命令に続く最初のF-11命令は、キャリーフラグ(07356)の内容を減算しません。



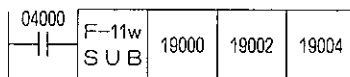
- 上記の演算は、9643-4958=4685を示しています。
- 下の桁から順次プログラムしていくと、桁下げの情報が上位桁に入ってきます。
(9・10ページ「倍長演算」参照)

**F-11w
SUB**

**レジスタ間(BCD4桁)の減算
(SUBtract)**

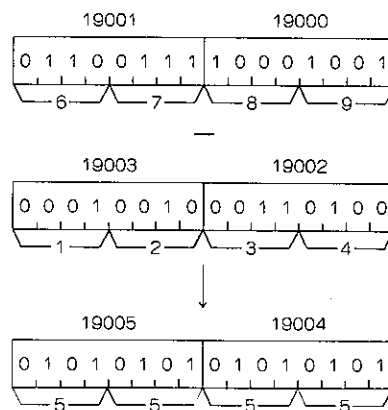
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-11w SUB</td> <td style="padding: 2px;">S₁</td> <td style="padding: 2px;">S₂</td> <td style="padding: 2px;">D</td> </tr> </table>				F-11w SUB	S ₁	S ₂	D										
F-11w SUB	S ₁	S ₂	D															
機能	レジスタS ₁ 、S ₁ +1の内容からレジスタS ₂ 、S ₂ +1の内容を減算(BCD4桁減算)してレジスタD、D+1に格納する。																	
演算内容	(S ₁ 、S ₁ +1)-(S ₂ 、S ₂ +1)→D、D+1																	
S ₁ の使用範囲	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border-right: 1px dashed black; padding: 2px;">コ0000~コ1576</td> <td style="width: 50%; padding: 2px;">@コ0000~@コ1574</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">コ2000~コ7576</td> <td style="padding: 2px;">@コ2000~@コ7574</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">b0000~b1776</td> <td style="padding: 2px;">@b0000~@b1774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">b2000~b3776</td> <td style="padding: 2px;">@b2000~@b3774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">09000~99776</td> <td style="padding: 2px;">@09000~@99774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">E0000~E7776</td> <td style="padding: 2px;">@E0000~@E7774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">771#1 000000~037776</td> <td style="padding: 2px;">771#1 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771#1 000000~037776	771#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																	
コ2000~コ7576	@コ2000~@コ7574																	
b0000~b1776	@b0000~@b1774																	
b2000~b3776	@b2000~@b3774																	
09000~99776	@09000~@99774																	
E0000~E7776	@E0000~@E7774																	
771#1 000000~037776	771#1 @000000~@037774																	
S ₂ の使用範囲	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border-right: 1px dashed black; padding: 2px;">コ0000~コ1576</td> <td style="width: 50%; padding: 2px;">@コ0000~@コ1574</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">コ2000~コ7576</td> <td style="padding: 2px;">@コ2000~@コ7574</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">b0000~b1776</td> <td style="padding: 2px;">@b0000~@b1774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">b2000~b3776</td> <td style="padding: 2px;">@b2000~@b3774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">09000~99776</td> <td style="padding: 2px;">@09000~@99774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">E0000~E7776</td> <td style="padding: 2px;">@E0000~@E7774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">771#1 000000~037776</td> <td style="padding: 2px;">771#1 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771#1 000000~037776	771#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																	
コ2000~コ7576	@コ2000~@コ7574																	
b0000~b1776	@b0000~@b1774																	
b2000~b3776	@b2000~@b3774																	
09000~99776	@09000~@99774																	
E0000~E7776	@E0000~@E7774																	
771#1 000000~037776	771#1 @000000~@037774																	
Dの使用範囲	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border-right: 1px dashed black; padding: 2px;">コ0000~コ1576</td> <td style="width: 50%; padding: 2px;">@コ0000~@コ1574</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">コ2000~コ7576</td> <td style="padding: 2px;">@コ2000~@コ7574</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">b0000~b1776</td> <td style="padding: 2px;">@b0000~@b1774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">b2000~b3776</td> <td style="padding: 2px;">@b2000~@b3774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">09000~99776</td> <td style="padding: 2px;">@09000~@99774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">E0000~E7776</td> <td style="padding: 2px;">@E0000~@E7774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">771#1 000000~037776</td> <td style="padding: 2px;">771#1 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771#1 000000~037776	771#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																	
コ2000~コ7576	@コ2000~@コ7574																	
b0000~b1776	@b0000~@b1774																	
b2000~b3776	@b2000~@b3774																	
09000~99776	@09000~@99774																	
E0000~E7776	@E0000~@E7774																	
771#1 000000~037776	771#1 @000000~@037774																	
演算条件	入力信号の立上り(OFF→ON)																	
演算	S ₁ 、S ₁ +1の内容	不変																
	S ₂ 、S ₂ +1の内容	不変																
算	Dの内容	演算結果(下位2桁)		S ₁ 、S ₁ +1、S ₂ 、S ₂ +1がBCDコードでない時不変														
	D+1の内容	演算結果(上位2桁)																
後フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354													
	0	1	0	0	1													
	1~9999	0	0	0	1													
	負の数値	0	1	0	0													
	BCD以外の時	0	0	1	0													

(解説)



命 令	
STR	04000
F-11w	19000
	19002
	19004

入力条件04000がOFF→ONの変化時、レジスタ19000、19001の内容(BCD4桁)からレジスタ19002、19003の内容(BCD4桁)を減算してレジスタ19004、19005に格納します。

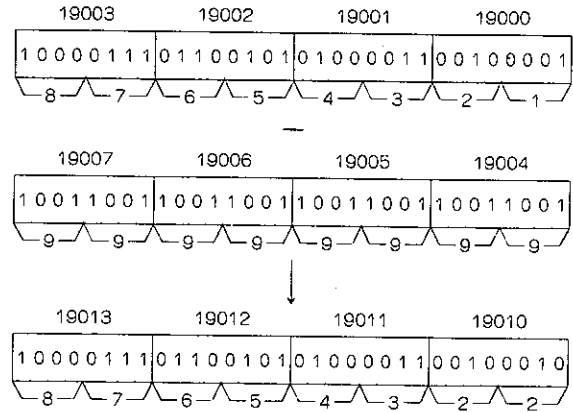
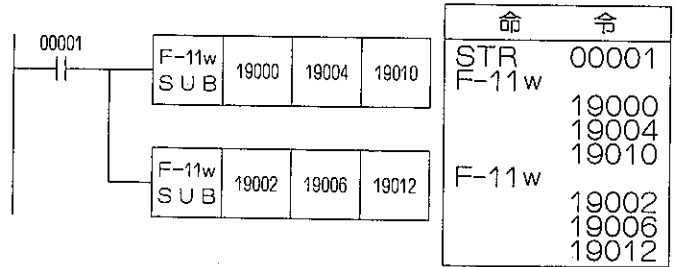


- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S₁、S₂、Dには必ず偶数アドレスを設定してください。(C0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- (S₁、S₁+1の内容) << (S₂、S₂+1の内容) の演算を行なうと答えは10000の補数で得られます。

(例) 2578-7890=-5312
 は5312の10000の補数4688が答となります。
 (12578-7890=4688と考えてください。)

【参考】 F-11w 命令も F-11 命令と同様に倍長演算が可能です。

従ってBCD16桁以上の減算をする場合、F-11w命令を続けて設定します。



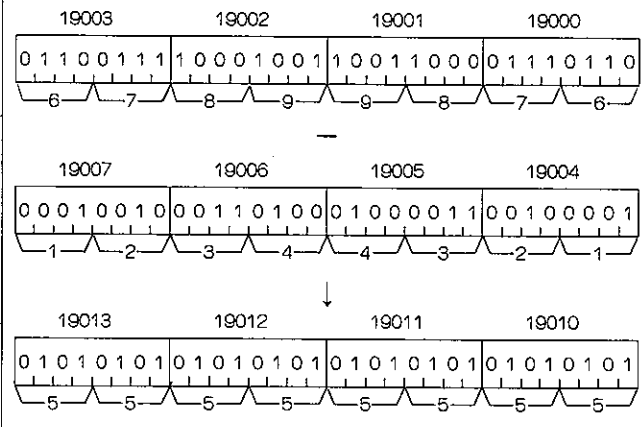
F-11d SUB レジスタ間(BCD8桁)の減算 (SUBtract)

シンボル	<table border="1"> <tr> <td>F-11d SUB</td> <td>S₁</td> <td>S₂</td> <td>D</td> </tr> </table>				F-11d SUB	S ₁	S ₂	D
F-11d SUB	S ₁	S ₂	D					
機能	レジスタS ₁ ~S ₁ +3の内容からレジスタS ₂ ~S ₂ +3の内容を減算(BCD8桁減算)してレジスタD~D+3に格納する。							
演算内容	(S ₁ ~S ₁ +3)-(S ₂ ~S ₂ +3)→D~D+3							
S ₁ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774						
S ₂ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774						
Dの使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774						
演算条件	入力信号の立上り(OFF→ON)							
演算後	S ₁ ~S ₁ +3の内容	不変						
	S ₂ ~S ₂ +3の内容	不変						
	D~D+3の内容	演算結果(BCD8桁)		S ₁ ~S ₁ +3, S ₂ ~S ₂ +3がBCDコードでない時不変				
		演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354		
		0	1	0	0	1		
	1~99999999	0	0	0	1			
	負の数値	0	1	0	0			
	BCD以外の時	0	0	1	0			

(解説)

命令	
STR	04000
F-11d	19000
	19004
	19010

入力条件04000がOFF→ONの変化時、レジスタ19000~19003の内容(BCD8桁)からレジスタ19004~19007の内容(BCD8桁)を減算してレジスタ19010~19013に格納します。



- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S₁、S₂、Dには必ず偶数アドレスを設定してください。(C0011、19003等は禁止)

- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

- (S₁、S₁+3の内容) << (S₂、S₂+3の内容) の演算を行なうと答えは100000000の補数で得られます。
(例) 25780000-78900000=-53120000
は53120000の100000000の補数46880000が答となります。

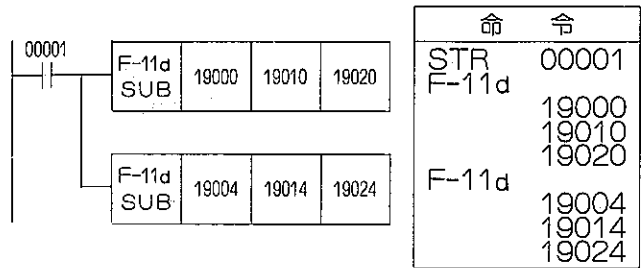
$$(125780000 - 78900000 = 46880000)$$

とを考えてください。

- S₁~S₁+3、S₂~S₂+3の内容がBCDコード以外の場合、エラーフラグ (07355) がONし、減算は行いません。
(D~D+3の内容は不変です。)

参考 F-11d 命令もF-11 命令と同様に倍長演算が可能です。

従ってBCD16桁以上の減算をする場合、F-11d 命令を続けて設定します。



19007	19006	19005	19004	19003	19002	19001	19000
8	7	6	5	4	3	2	1
0	9	8	7	6	5	4	3

19017	19016	19015	19014	19013	19012	19011	19010
9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9



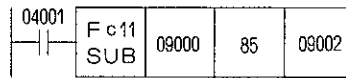
19027	19026	19025	19024	19023	19022	19021	19020
8	7	6	5	4	3	2	1
0	9	8	7	6	5	4	4

Fc11
SUB

レジスタ(BCD2桁)と定数(2桁)の減算
(SUBtract)

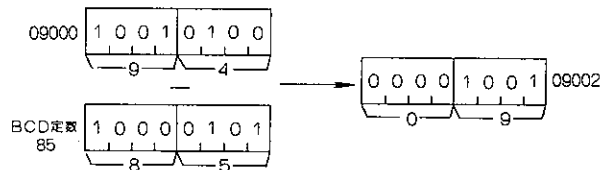
シンボル	Fc11 SUB S _i n D					
機能	レジスタS _i の内容から2桁のBCD定数nを減算してレジスタDに格納する。					
演算内容	S _i → n → D					
S _i の使用範囲	コ0000~コ15777 : @コ0000~@コ15774 コ2000~コ75777 : @コ2000~@コ75774 b0000~b17777 : @b0000~@b17774 b2000~b37777 : @b2000~@b37774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 ファイル000000~037777 : ファイル@000000~@037774					
nの使用範囲	00~99					
Dの使用範囲	コ0000~コ15777 : @コ0000~@コ15774 コ2000~コ75777 : @コ2000~@コ75774 b0000~b17777 : @b0000~@b17774 b2000~b37777 : @b2000~@b37774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 ファイル000000~037777 : ファイル@000000~@037774					
演算条件	入力信号の立上り(OFF→ON)					
演算	S _i の内容	不変				
	Dの内容	●演算結果 ●レジスタS _i の内容がBCDコードでない時不変				
後	フラグ	演算結果	ゼロ	キャリー	エラー	ノンキャリー
		0	07357	07356	07355	07354
		1~99	0	0	0	1
		負の数値	0	1	0	0
	S _i がBCDでない時	0	0	1	0	

(解説)

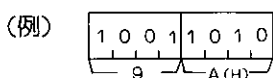


命令	
STR	04001
Fc11	09000
	85
	09002

入力条件04001がOFF→ONの変化時に、レジスタ09000の内容からBCD定数85を減算して、レジスタ09002に格納します。
タイミング関係はF-11と同様です。

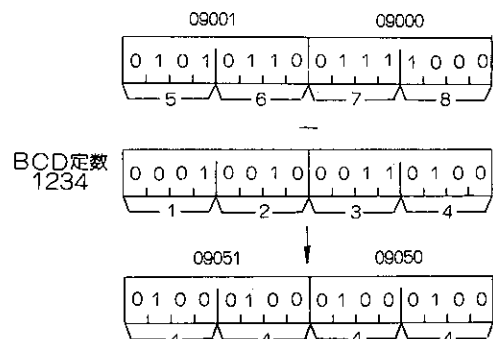
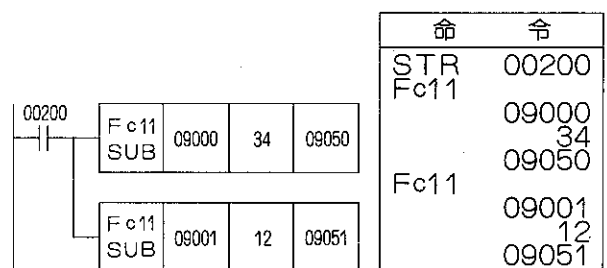


- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- (S_iの内容) <nの演算を行うと、答は100の補数で得られます。
(例) 23-85=-62は、62の100の補数38が答となります。
(123-85=38と考えてください)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- S_iの内容がBCDコード以外の数値の場合、エラーフラグ(07355)がONし、減算は行いません。
(Dの内容は不変です)



1010はBCDでは禁止のコードです。

【参考】 F-11と同様にBCD3桁以上の減算が可能です。

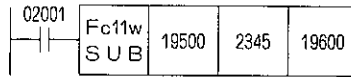


**Fc11w
SUB**

**レジスタ(BCD4桁)と定数(4桁)の減算
(SUBtract)**

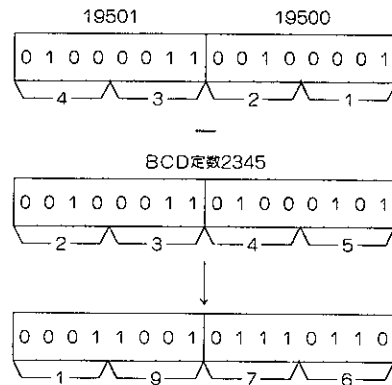
シンボル	— Fc11w SUB S _i n D				
機能	レジスタS _i 、S _i +1の内容(BCD4桁)から4桁のBCD定数nを減算してレジスタD、D+1に格納する。				
演算内容	(S _i 、S _i +1) - n → D、D+1				
S _i の使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 774#1 000000~037776 : 774#1 @000000~@037774				
nの使用範囲	0000~9999				
Dの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 774#1 000000~037776 : 774#1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	S _i 、S _i +1の内容	不変			
	Dの内容	演算結果(下位2桁)	S _i 、S _i +1の内容がBCDコードでない時不変		
	D+1の内容	演算結果(上位2桁)			
フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
	0	1	0	0	1
	1~9999	0	0	0	1
	負の数値	0	1	0	0
	BCD以外の時	0	0	1	0

(解説)



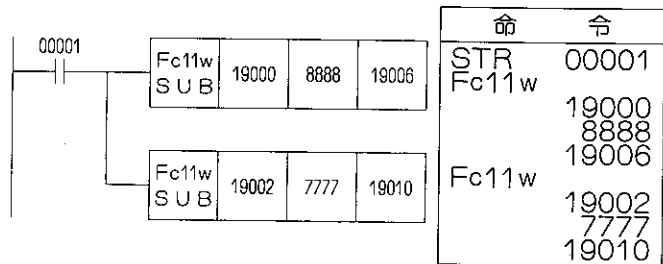
命令	
STR	02001
Fc11w	19500
	2345
	19600

入力条件02001がOFF→ONの変化時、レジスタ19500、19501の内容(BCD4桁)からBCD定数2345を減算してレジスタ19600、19601に格納します。

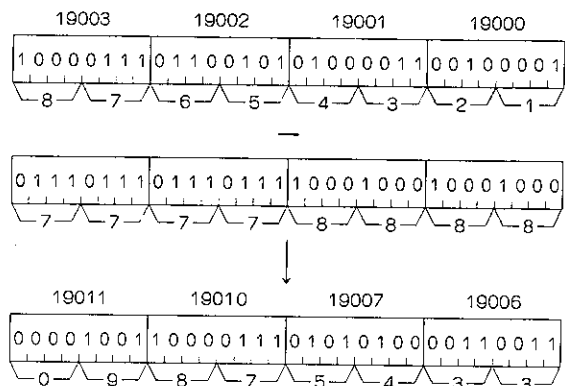


- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S_i、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- (S_i、S_i+1の内容) < nの演算を行なうと答は10000の補数で得られます。
(例) 4568 - 7890 = -3322
は3322の10000の補数6678が答となります。
(14568 - 7890 = 6678と考えてください。)

参考 Fc11w命令もFc11命令と同様に倍長演算が可能です。
従ってBCD9桁以上の減算をする場合、Fc11w命令を続けて設定します。



命令	
STR	00001
Fc11w	19000
	8888
	19006
Fc11w	19002
	7777
	19010

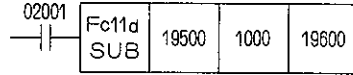


**Fc11d
SUB**

**レジスタ(BCD8桁)と定数(4桁)の減算
(SUBtract)**

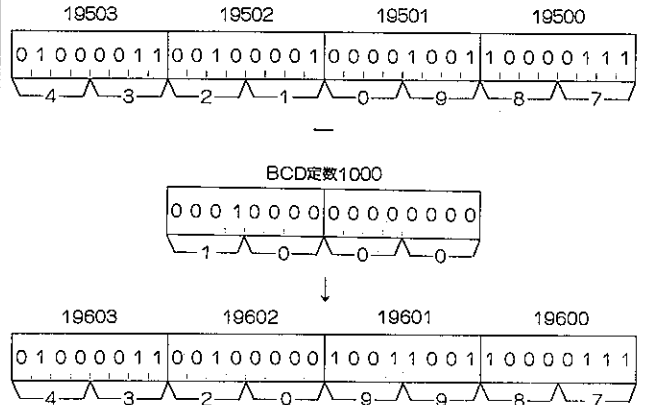
シンボル	— Fc11d SUB S _i n D				
機能	レジスタS _i ~S _i +3の内容(BCD8桁)から4桁のBCD定数nを減算してレジスタD~D+3に格納する。				
演算内容	(S _i ~S _i +3)-n→D~D+3				
S _i の使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 771M1 000000~037774 : 771M1 @000000~@037774				
nの使用範囲	0000~9999				
Dの使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 771M1 000000~037774 : 771M1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	S _i ~S _i +3の内容	不変			
	D~D+3の内容	演算結果(BCD8桁)	S _i ~S _i +3の内容がBCDコードでない時不変		
フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
	0	1	0	0	1
	1~99999999	0	0	0	1
	負の数値	0	1	0	0
	BCD以外の時	0	0	1	0

(解説)



命令	
STR	02001
Fc11d	19500
	1000
	19600

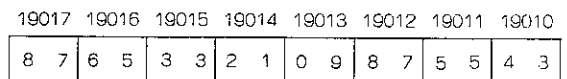
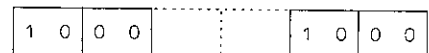
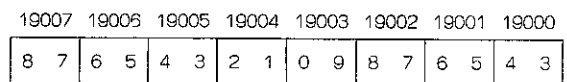
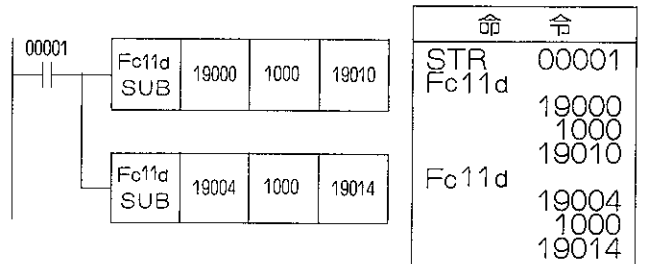
入力条件02001がOFF→ONの変化時、レジスタ19500~19503の内容(BCD8桁)からBCD定数1000を減算してレジスタ19600~19603に格納します。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- S_i、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)
- (S_i、S_i+3の内容) <n>の演算を行なうと答は100000000の補数で得られます。

(例) 4568-7890=-3322は3322の10000000の補数6678が答となります。
(100004568-7890=99996678と考えてください。)

参考 Fc11d 命令もFc11命令と同様に倍長演算が可能です。従ってBCD16桁以上の減算をする場合、Fc11d命令を続けて設定します。



**F-12
CMP**

**レジスタ間(1バイト)の比較
(CoMPare)**

シンボル	— F-12 CMP S ₁ S ₂					
機能	レジスタS ₁ の内容とレジスタS ₂ の内容を大小比較する。					
演算内容	S ₁ <=> S ₂ → フラグ					
S ₁ の使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 09000~99777 @09000~@99774 E00000~E77777 @E00000~@E77774 771M1 000000~037777 771M1 @000000~@037774					
S ₂ の使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 09000~99777 @09000~@99774 E00000~E77777 @E00000~@E77774 771M1 000000~037777 771M1 @000000~@037774					
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)					
演算後	S ₁ の内容	不変				
	S ₂ の内容	不変				
	フラグ	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
		S ₁ > S ₂	0	0	0	1
	S ₁ = S ₂	1	0	0	1	
	S ₁ < S ₂	0	1	0	0	

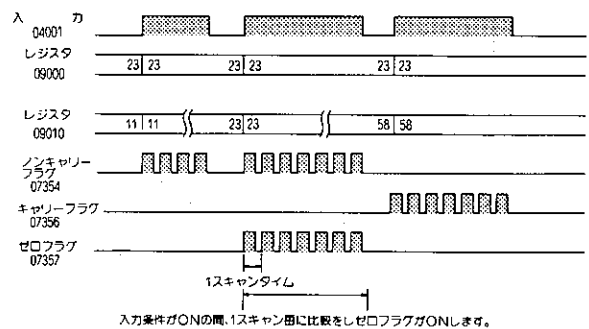
(解説)



命 令	
STR	04003
F-12	09000 09010

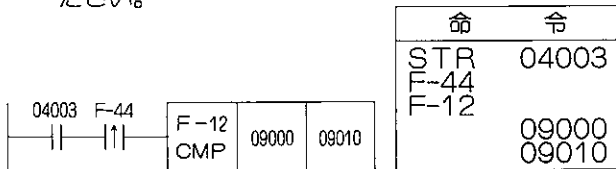
入力条件04003がONの時、レジスタ09000とレジスタ09010の内容を大小比較して、その結果をノンキャリーフラグ(07354)、キャリーフラグ(07356)とゼロフラグ(07357)に設定します。この時、レジスタ09000とレジスタ09010の内容は不変です。

●レジスタの内容とフラグの推移



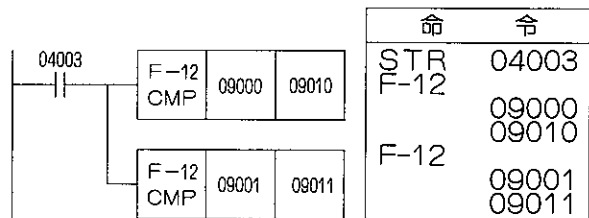
- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 入力信号がONの間、毎スキャンサイクル演算します。(9・7ページ「演算実行条件」参照)
- エラーフラグ(07355)は常に「0」となります。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

【参考】 入力条件のOFF→ONの変化時にのみ、大小比較をする場合は、入力条件に微分命令を組合せてください。



命 令	
STR	04003
F-44	
F-12	09000 09010

【参考】 2バイト以上のデータの大小比較をする場合は、加算・減算(F-10・F-11)の場合と同様に、下位の数値から比較するようにプログラムします。連続して、F-12命令を設定すると、2つ目以降のF-12命令では、キャリーフラグ(07356)の内容も比較対象に入ります。(STR命令に続く最初のF-12命令では、キャリーフラグ(07356)の内容は比較対象から除外します。)



命 令	
STR	04003
F-12	09000 09010
F-12	09001 09011

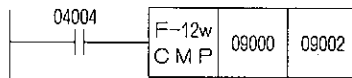
下の桁から、順次プログラムしていくと、桁下げの情報が上位桁に入ってきます。9・10ページ「倍長演算」を参照してください。

【参考】 下記のF命令は働きが類似しています。F-12、F-12w、F-12d、Fc12、Fc12w、F-112、F-112w

F-12w CMP レジスタ間(1ワード)の比較 (COMPare)

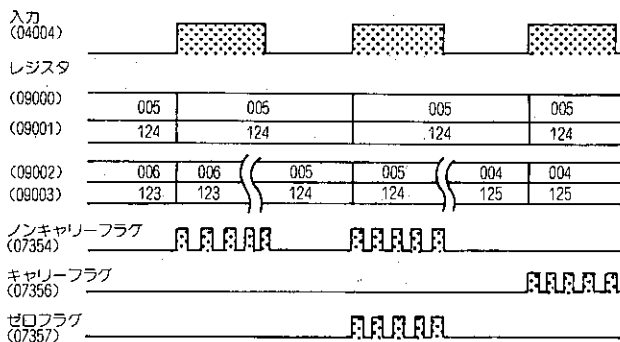
シンボル	F-12w CMP S ₁ S ₂				
機能	レジスタS ₁ 、S ₁ +1の内容(1ワードデータ)とレジスタS ₂ 、S ₂ +1の内容(1ワードデータ)を大小比較する。				
演算内容	S ₁ 、S ₁ +1 <=> S ₂ 、S ₂ +1 → フラグ				
S ₁ の使用範囲	コ0000~コ1576 コ2000~コ7576 b0000~b1776 b2000~b3776 09000~99776 E0000~E7776 77411 000000~037776	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 77411 @000000~@037774			
S ₂ の使用範囲	コ0000~コ1576 コ2000~コ7576 b0000~b1776 b2000~b3776 09000~99776 E0000~E7776 77411 000000~037776	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 77411 @000000~@037774			
演算条件	入力信号がONの時 (OFF→ONの変化時に限定されない)				
演算後	S ₁ 、S ₁ +1の内容	不変			
	S ₂ 、S ₂ +1の内容	不変			
フラグ	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
	S ₁ 、S ₁ +1 > S ₂ 、S ₂ +1	0	0	0	1
	S ₁ 、S ₁ +1 = S ₂ 、S ₂ +1	1	0	0	1
	S ₁ 、S ₁ +1 < S ₂ 、S ₂ +1	0	1	0	0

(解説)



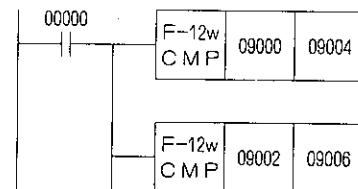
命令	
STR	04004
F-12w	09000 09002

入力条件04004がONの時レジスタ09000、09001の内容(1ワードデータ)とレジスタ09002、09003の内容(1ワードデータ)を大小比較して、その結果をノンキャリーフラグ(07354)、キャリーフラグ(07356)、ゼロフラグ(07357)に設定します。
この時レジスタ09000、09001、09002、09003の内容は不変です。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- S₁、S₂には必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)
- 入力信号がONの間、毎スキャンサイクル演算します。(9・7ページ「演算実行条件」参照)

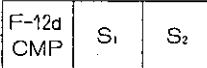
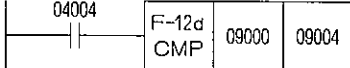
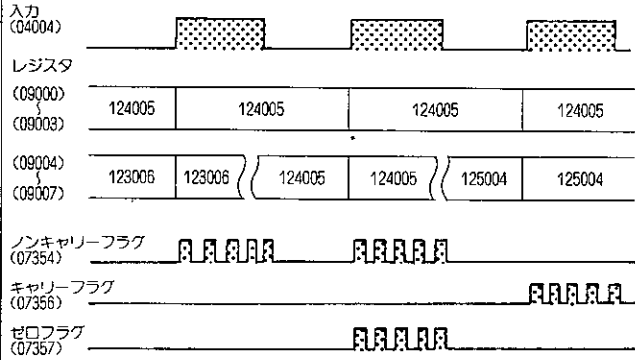
参考 F-12w命令を連続して使用すると4バイト以上のデータの大小比較ができます。



命令	
STR	00000
F-12w	09000 09004
F-12w	09002 09006

参考 下記のF命令は働きが類似しています。
F-12、F-12w、F-12d、Fc12、Fc12w、
F-112、F-112w

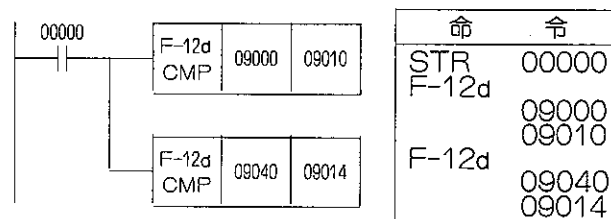
F-12d CMP レジスタ間(2ワード)の比較 (CoMPare)

シンボル			(解説) 		<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>04004</td></tr> <tr><td>F-12d</td><td>09000</td></tr> <tr><td>F-12d</td><td>09004</td></tr> </table>	命 令		STR	04004	F-12d	09000	F-12d	09004
命 令													
STR	04004												
F-12d	09000												
F-12d	09004												
機能	レジスタS ₁ ~S ₁ +3の内容(2ワードデータ)とレジスタS ₂ ~S ₂ +3の内容(2ワードデータ)を大小比較する。		入力条件04004がONの時レジスタ09000~09003の内容(2ワードデータ)とレジスタ09004~09007の内容(2ワードデータ)を大小比較して、その結果をノンキャリーフラグ(07354)、キャリーフラグ(07356)ゼロフラグ(07357)に設定します。 この時レジスタ09000~09007の内容は不変です。										
演算内容	S ₁ ~S ₁ +3 <=> S ₂ ~S ₂ +3 → フラグ												
S ₁ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774											
S ₂ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774											
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)												
演算後	S ₁ ~S ₁ +3の内容	不変											
	S ₂ ~S ₂ +3の内容	不変											
	フラグ	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354							
		S ₁ ~S ₁ +3 > S ₂ ~S ₂ +3	0	0	0	1							
	S ₁ ~S ₁ +3 = S ₂ ~S ₂ +3	1	0	0	1								
	S ₁ ~S ₁ +3 < S ₂ ~S ₂ +3	0	1	0	0								

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- S₁、S₂には必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)
- 入力信号がONの間、毎スキャンサイクル演算します。(9・7ページ「演算実行条件」参照)

参考 下記のF命令は働きが類似しています。
 F-12、F-12w、F-12d、Fc12、Fc12w、F-112、F-112w

参考 F-12d 命令を連続して使用すると8バイト以上のデータの大小比較ができます。

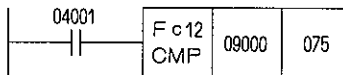


**Fc12
CMP**

レジスタと8進定数(1バイト)の比較
(CoMPare)

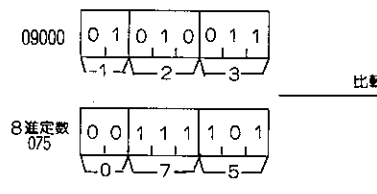
シンボル						
機能	レジスタSiの内容と8進定数nを大小比較する。					
演算内容	Si <=> n → フラグ					
Siの使用範囲	コ00000~コ15777 : @コ00000~@コ15774 コ20000~コ75777 : @コ20000~@コ75774 b00000~b17777 : @b00000~@b17774 b20000~b37777 : @b20000~@b37774 090000~997777 : @090000~@997774 E00000~E77777 : @E00000~@E77774 777777 000000~037777 : 777777 @000000~@037774					
nの使用範囲	000~377					
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)					
演算後	Siの内容	不変				
	フラグ	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
		Si > n	0	0	0	1
		Si = n	1	0	0	1
		Si < n	0	1	0	0

(解説)



命令	
STR Fc12	04001
	09000
	075

入力条件04001がONの時に、レジスタ09000の内容と8進定数075を大小比較して、その結果をノンキャリーフラグ(07354)、キャリーフラグ(07356)、ゼロフラグ(07357)に設定します。この時レジスタ09000の内容は不変です。

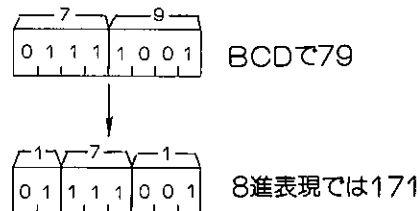


ゼロ	キャリー	エラー	ノンキャリー
07357	07356	07355	07354
0	0	0	1

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 入力信号がONの間、毎スキャンサイクル演算します。(9・7ページ「演算実行条件」参照)
- エラーフラグ(07355)は常に「0」となります。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

参考 下記のF命令は働きが類似しています。
F-12、F-12w、F-12d、Fc12、Fc12w、
F-112、F-112w

参考 Fc12はプログラム書き込み時に8進数を用います。8進数は、あらゆるビットパターンを数値で表現でき、面倒な重み計算も不要です。BCD定数と比較する場合、BCD定数を8進数に変換し、プログラムを書き込んでください。



**Fc12w
CMP**

レジスタと8進定数(1ワード)の比較
(COMPare)

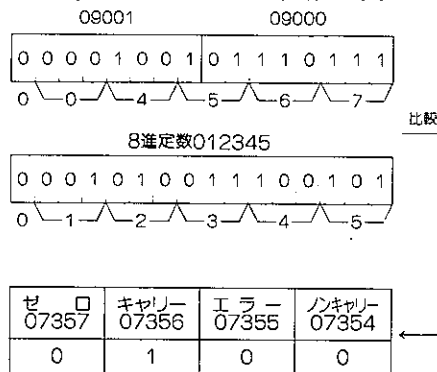
シンボル	Fc12w CMP S _i n				
機能	レジスタS _i 、S _i +1の内容(1ワードデータ)と8進定数nを大小比較する。				
演算内容	S _i 、S _i +1 <=> n → フラグ				
S _i の使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 771#1 000000~037776 771#1 @000000~@037774				
nの使用範囲	000000~177777				
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)				
演算後	S _i 、S _i +1の内容	不変			
	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
	S _i 、S _i +1 > n	0	0	0	1
	S _i 、S _i +1 = n	1	0	0	1
	S _i 、S _i +1 < n	0	1	0	0

(解説)



命令	
STR	02000
Fc12w	09000 012345

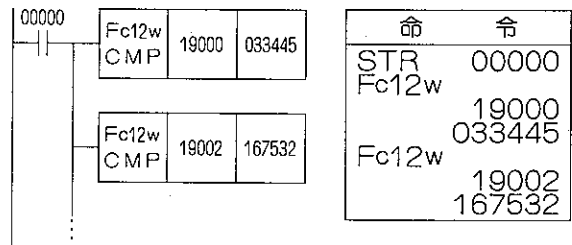
入力条件02000がONの時、レジスタ09000、09001の内容(1ワードデータ)と8進定数012345を大小比較して、その結果をノンキャリーフラグ(07354)、キャリーフラグ(07356)、ゼロフラグ(07357)に設定します。この時、レジスタ09000、09001の内容は不変です。タイミング関係はF-12wと同様です。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- S_iには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)
- 入力信号がONの間、毎スキャンサイクル演算します。(9・7ページ「演算実行条件」参照)

参考) 下記のF命令は働きが類似しています。
F-12、F-12w、F-12d、Fc12、Fc12w、F-112、F-112w

参考) Fc12w命令を連続して使用すると4バイト以上のデータの大小比較ができます。

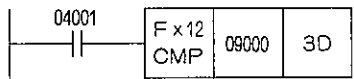


**Fx12
CMP**

**レジスタと16進定数(1バイト)の比較
(CoMPare)**

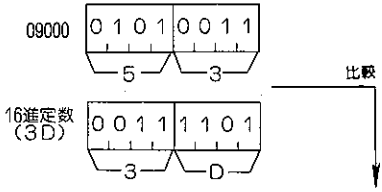
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F x12 CMP</td><td style="padding: 2px;">S_i</td><td style="padding: 2px;">n</td></tr></table>					F x12 CMP	S _i	n
F x12 CMP	S _i	n						
機能	レジスタS _i の内容と16進定数nを大小比較する。							
演算内容	S _i <=> n → フラグ							
S _i の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 77IM1 000000~037777 77IM1 @000000~@037774							
nの使用範囲	00~FF							
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)							
演算後	S _i の内容	不変						
	フラグ	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354		
	S _i > n		0	0	0	1		
	S _i = n		1	0	0	1		
S _i < n		0	1	0	0			

(解説)



命令	
STR	04001
Fx12	09000
	3D

入力条件04001がONの時に、レジスタ09000の内容と16進定数3Dを大小比較して、その結果をノンキャリーフラグ(07354)、キャリーフラグ(07356)、ゼロフラグ(07357)に設定します。この時レジスタ09000の内容は不変です。



ゼロ	キャリー	エラー	ノンキャリー
07357	07356	07355	07354
0	0	0	1

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合、必ず偶数アドレスを設定してください。(@コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- 入力信号がONの間、毎スキャンサイクル演算が実行されます。(9・7ページ「演算実行条件」参照)
- エラーフラグ(07355)は常に"0"となります。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)

参考 下記のF命令は働きが類似しています。
 F-12、F-12w、F-12d、Fc12、Fc12w、
 Fx12、Fx12w、F-112、F-112w

**Fx12w
CMP**

**レジスタと16進定数(1ワード)の比較
(COMPare)**

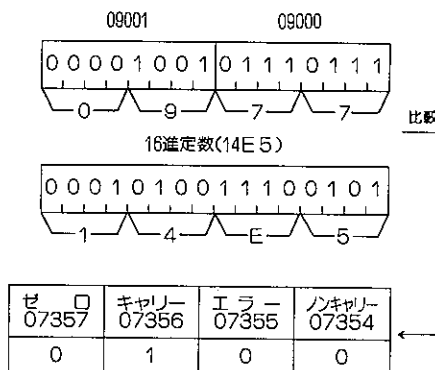
シンボル	<table border="1"> <tr> <td>Fx12w CMP</td> <td>S_i</td> <td>n</td> </tr> </table>					Fx12w CMP	S _i	n
Fx12w CMP	S _i	n						
機能	レジスタS _i 、S _i +1の内容(1ワードデータ)と16進定数nを大小比較する。							
演算内容	S _i 、S _i +1 <=> n → フラグ							
S _i の使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 771#1 000000~037776 771#1 @000000~@037774							
nの使用範囲	0000~FFFF							
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)							
演算後	S _i 、S _i +1の内容	不変						
	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354			
	S _i 、S _i +1 > n	0	0	0	1			
	S _i 、S _i +1 = n	1	0	0	1			
	S _i 、S _i +1 < n	0	1	0	0			

(解説)



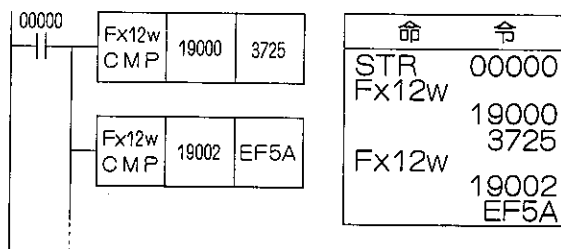
命令	
STR	02000
Fx12w	09000
	14E5

入力条件02000がONの時、レジスタ09000、09001の内容(1ワードデータ)と16進定数14E5を大小比較して、その結果をノンキャリーフラグ(07354)、キャリーフラグ(07356)、ゼロフラグ(07357)に設定します。この時、レジスタ09000、09001の内容は不変です。タイミング関係はF-12wと同様な動きをします。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- S_iには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)
- 入力信号がONの間、毎スキャンサイクル演算が実行されます。(9・7ページ「演算実行条件」参照)

参考 Fx12w命令を連続して使用すると4バイト以上のデータの大小比較ができます。

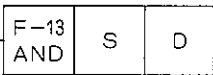


命令	
STR	00000
Fx12w	19000
	3725
Fx12w	19002
	EF5A

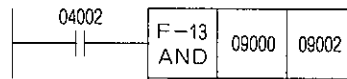
参考 下記のF命令は動きが類似しています。
F-12、F-12w、F-12d、Fc12、Fc12w、Fx12、Fx12w、F-112、F-112w

F-13
AND

レジスタ間(1バイト)の論理積
(AND)

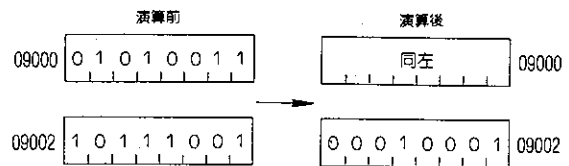
シンボル			
機能	レジスタSの内容(8ビットデータ)とレジスタDの内容(8ビットデータ)の論理積をとり、レジスタDに格納する。		
演算内容	S AND D		
Sの使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 Q9000~Q9977 : @Q9000~@Q9974 E0000~E7777 : @E0000~@E7774 771#1 000000~037777 : 771#1 @000000~@037774		
Dの使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 Q9000~Q9977 : @Q9000~@Q9974 E0000~E7777 : @E0000~@E7774 771#1 000000~037777 : 771#1 @000000~@037774		
演算条件	入力信号の立上り(OFF→ON)		
演算後	Sの内容	不変	
	Dの内容	演算結果	
	フラグ	不変	

〔解説〕



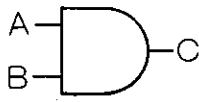
命 令	
STR	04002
F-13	09000
	09002

入力条件04002がOFF→ONの変化時に、レジスタ09000の8ビットの内容とレジスタ09002の8ビットの内容の論理積(AND)をとり、レジスタ09002に格納します。レジスタ09000の内容は不変です。



●コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

ANDの真理値表

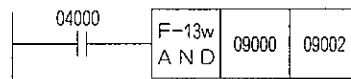
シンボル	A	B	C
	0	0	0
	1	0	0
	0	1	0
	1	1	1

**F-13w
AND**

**レジスタ間(1ワード)の論理積
(AND)**

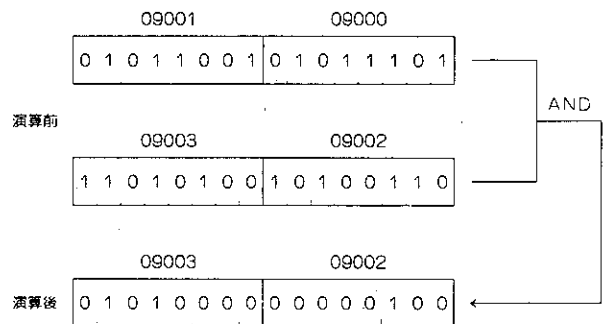
シンボル		
機能	レジスタS、S+1の内容(16ビットデータ)とレジスタD、D+1の内容(16ビットデータ)の論理積をとり、レジスタD、D+1に格納する。	
演算内容	S、S+1 \cap D、D+1 \rightarrow D、D+1	
Sの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 771#1 000000~037776 771#1 @000000~@037774	
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 771#1 000000~037776 771#1 @000000~@037774	
演算条件	入力信号の立上り(OFF \rightarrow ON)	
演算後	S、S+1の内容	不変
	D、D+1の内容	演算結果
	フラグ	不変

〔解説〕



命 令	
STR	04000
F-13w	09000 09002

入力条件04000がOFF \rightarrow ONの変化時、レジスタ09000、09001の内容(16ビットデータ)とレジスタ09002、09003の内容(16ビットデータ)の論理積をとり、レジスタ09002、09003に格納します。
レジスタ09000、09001の内容は不変です。



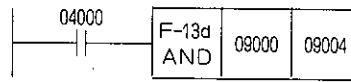
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

**F-13d
AND**

**レジスタ間(2ワード)の論理積
(AND)**

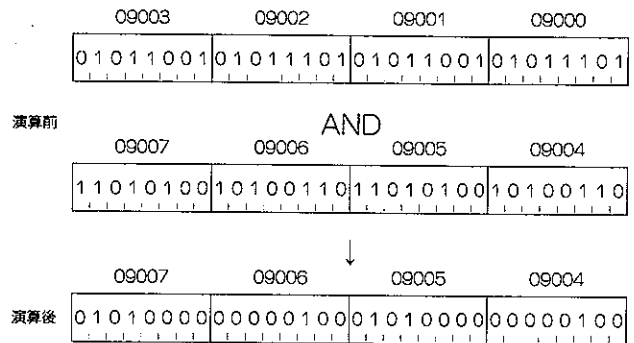
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-13d AND</td><td style="padding: 2px;">S</td><td style="padding: 2px;">D</td></tr></table>		F-13d AND	S	D											
F-13d AND	S	D														
機能	レジスタS~S+3の内容(32ビットデータ)とレジスタD~D+3の内容(32ビットデータ)の論理積をとり、レジスタD~D+3に格納する。															
演算内容	S~S+3∩D~D+3→D~D+3															
Sの使用範囲	<table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 50%; border-right: 1px dashed black;">C0000~C1574</td><td style="width: 50%;">@C0000~@C1574</td></tr> <tr><td style="border-right: 1px dashed black;">C2000~C7574</td><td>@C2000~@C7574</td></tr> <tr><td style="border-right: 1px dashed black;">b0000~b1774</td><td>@b0000~@b1774</td></tr> <tr><td style="border-right: 1px dashed black;">b2000~b3774</td><td>@b2000~@b3774</td></tr> <tr><td style="border-right: 1px dashed black;">09000~99774</td><td>@09000~@99774</td></tr> <tr><td style="border-right: 1px dashed black;">E0000~E7774</td><td>@E0000~@E7774</td></tr> <tr><td style="border-right: 1px dashed black;">77777@000000~@037774</td><td>77777@000000~@037774</td></tr> </table>		C0000~C1574	@C0000~@C1574	C2000~C7574	@C2000~@C7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	77777@000000~@037774	77777@000000~@037774
C0000~C1574	@C0000~@C1574															
C2000~C7574	@C2000~@C7574															
b0000~b1774	@b0000~@b1774															
b2000~b3774	@b2000~@b3774															
09000~99774	@09000~@99774															
E0000~E7774	@E0000~@E7774															
77777@000000~@037774	77777@000000~@037774															
Dの使用範囲	<table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 50%; border-right: 1px dashed black;">C0000~C1574</td><td style="width: 50%;">@C0000~@C1574</td></tr> <tr><td style="border-right: 1px dashed black;">C2000~C7574</td><td>@C2000~@C7574</td></tr> <tr><td style="border-right: 1px dashed black;">b0000~b1774</td><td>@b0000~@b1774</td></tr> <tr><td style="border-right: 1px dashed black;">b2000~b3774</td><td>@b2000~@b3774</td></tr> <tr><td style="border-right: 1px dashed black;">09000~99774</td><td>@09000~@99774</td></tr> <tr><td style="border-right: 1px dashed black;">E0000~E7774</td><td>@E0000~@E7774</td></tr> <tr><td style="border-right: 1px dashed black;">77777@000000~@037774</td><td>77777@000000~@037774</td></tr> </table>		C0000~C1574	@C0000~@C1574	C2000~C7574	@C2000~@C7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	77777@000000~@037774	77777@000000~@037774
C0000~C1574	@C0000~@C1574															
C2000~C7574	@C2000~@C7574															
b0000~b1774	@b0000~@b1774															
b2000~b3774	@b2000~@b3774															
09000~99774	@09000~@99774															
E0000~E7774	@E0000~@E7774															
77777@000000~@037774	77777@000000~@037774															
演算条件	入力信号の立上り(OFF→ON)															
演算後	S~S+3の内容	不変														
	D~D+3の内容	演算結果														
	フラグ	不変														

(解説)



命令	
STR	04000
F-13d	09000
	09004

入力条件04000がOFF→ONの変化時、レジスタ09000~09003の内容(32ビットデータ)とレジスタ09004~09007の内容(32ビットデータ)の論理積をとり、レジスタ09004~09007に格納します。
レジスタ09000~09003の内容は不変です。



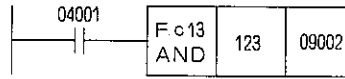
- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(C0011、19003等は禁止)

**Fc13
AND**

**レジスタと8進定数(1バイト)の論理積
(AND)**

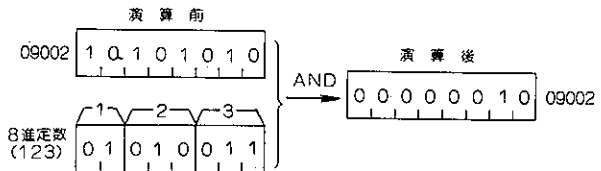
シンボル		
機能	8進定数nとレジスタDの内容の論理積をとり、レジスタDに格納する。	
演算内容	$n \cap D \rightarrow D$	
nの使用範囲	000~377	
Dの使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 77771 000000~03777 : 77771 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Dの内容	演算結果
	フラグ	不変

(解説)



命 令	
STR	04001
Fc13	123 09002

入力条件04001がOFF→ONの変化時に、8進定数123とレジスタ09002の内容の論理積をとり、レジスタ09002に格納します。




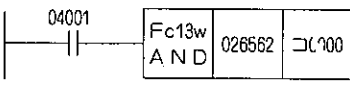
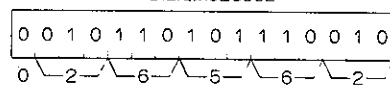
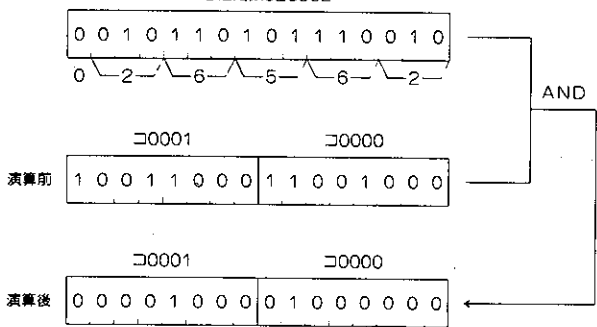
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

ANDの真理値表

シンボル	A	B	C
	0	0	0
	1	0	0
	0	1	0
	1	1	1

Fc13w AND

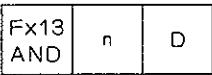
レジスタと8進定数(1ワード)の論理積 (AND)

シンボル			<p>(解説)</p> 	<table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>04001</td> </tr> <tr> <td>Fc13w</td> <td>026562 コ0000</td> </tr> </tbody> </table>	命 令		STR	04001	Fc13w	026562 コ0000
命 令										
STR	04001									
Fc13w	026562 コ0000									
機能	8進定数 n とレジスタ D、D+1 の内容 (16ビットデータ) の論理積をとり、レジスタ D、D+1 に格納する。		<p>入力条件 04001 が OFF → ON の変化時に 8進定数 026562 とレジスタ コ0000、コ0001 の内容 (16ビットデータ) の論理積をとり、レジスタ コ0000、コ0001 に格納します。</p>							
演算内容	$n \cap D, D+1 \rightarrow D, D+1$									
n の使用範囲	000000 ~ 177777		<p>8進定数 026562</p> 							
D の使用範囲	コ0000 ~ コ1576 : @コ0000 ~ @コ1574 コ2000 ~ コ7576 : @コ2000 ~ @コ7574 b0000 ~ b1776 : @b0000 ~ @b1774 b2000 ~ b3776 : @b2000 ~ @b3774 09000 ~ 99776 : @09000 ~ @99774 E0000 ~ E7776 : @E0000 ~ @E7774 ファイル 000000 ~ 037776 : ファイル @000000 ~ @037774									
演算条件	入力信号の立上り (OFF → ON)		<p>AND</p> 							
演算後	D、D+1の内容	演算結果								
演算後	フ ラ グ	不変								

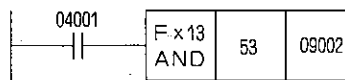
- コ0734 ~ コ0737 は特殊領域です。
(2・5 ページ「特殊リレー」参照)
- D には必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

**Fx13
AND**

**レジスタと16進定数(1バイト)の論理積
(AND)**

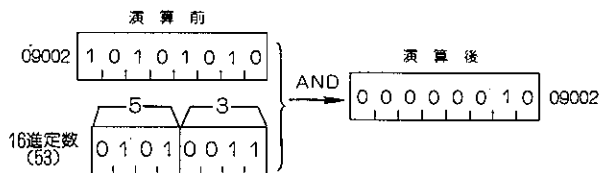
シンボル		
機能	16進定数nとレジスタDの内容の論理積をとり、レジスタDに格納する。	
演算内容	$n \cap D \rightarrow D$	
nの使用範囲	00~FF	
Dの使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~09777 : @09000~@09774 E0000~E7777 : @E0000~@E7774 77771 000000~037777 : 77771 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Dの内容	演算結果
	フラグ	不変

〔解説〕



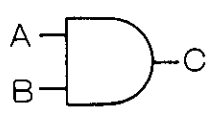
命 令	
STR	04001
Fx13	53
	09002

入力条件04001がOFF→ONの変化時に、16進定数53とレジスタ09002の内容の論理積をとり、レジスタ09002に格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

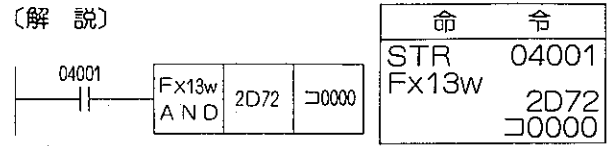
ANDの真理値表

シンボル	A	B	C
	0	0	0
	1	0	0
	0	1	0
	1	1	1

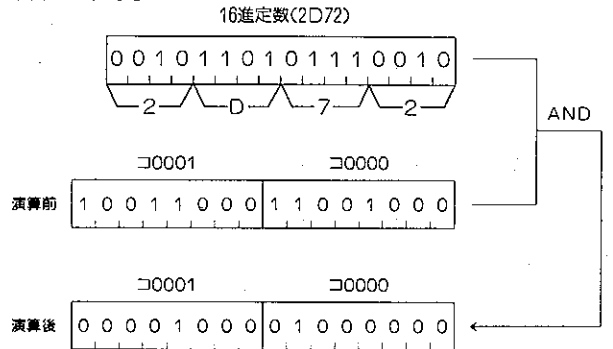
**Fx13w
AND**

**レジスタと16進定数(1ワード)の論理積
(AND)**

シンボル	<table border="1"> <tr> <td>Fx13w AND</td> <td>n</td> <td>D</td> </tr> </table>		Fx13w AND	n	D	<p>(解説)</p>										
Fx13w AND	n	D														
機能	16進定数nとレジスタD、D+1の内容(16ビットデータ)の論理積をとり、レジスタD、D+1に格納する。															
演算内容	n ∩ D、D+1 → D、D+1		<p>命令</p> <table border="1"> <tr> <td>STR</td> <td>04001</td> </tr> <tr> <td>Fx13w</td> <td>2D72 C0000</td> </tr> </table>	STR	04001	Fx13w	2D72 C0000									
STR	04001															
Fx13w	2D72 C0000															
nの使用範囲	0000~FFFF															
Dの使用範囲	<table border="1"> <tr> <td>C0000~C1576</td> <td>@C0000~@C1574</td> </tr> <tr> <td>C2000~C7576</td> <td>@C2000~@C7574</td> </tr> <tr> <td>B0000~B1776</td> <td>@B0000~@B1774</td> </tr> <tr> <td>B2000~B3776</td> <td>@B2000~@B3774</td> </tr> <tr> <td>09000~09776</td> <td>@09000~@09774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771#1 000000~037776</td> <td>771#1 @000000~@037774</td> </tr> </table>		C0000~C1576	@C0000~@C1574	C2000~C7576	@C2000~@C7574	B0000~B1776	@B0000~@B1774	B2000~B3776	@B2000~@B3774	09000~09776	@09000~@09774	E0000~E7776	@E0000~@E7774	771#1 000000~037776	771#1 @000000~@037774
C0000~C1576	@C0000~@C1574															
C2000~C7576	@C2000~@C7574															
B0000~B1776	@B0000~@B1774															
B2000~B3776	@B2000~@B3774															
09000~09776	@09000~@09774															
E0000~E7776	@E0000~@E7774															
771#1 000000~037776	771#1 @000000~@037774															
演算条件	入力信号の立上り(OFF→ON)															
演算後	D、D+1の内容	演算結果														
	フラグ	不変														



入力条件04001がOFF→ONの変化時に16進定数2D72とレジスタC0000、C0001の内容(16ビットデータ)の論理積をとり、レジスタC0000、C0001に格納します。



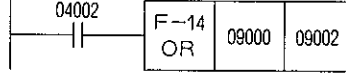
- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(C0011、19003等は禁止)

**F-14
OR**

**レジスタ間(1バイト)の論理和
(OR)**

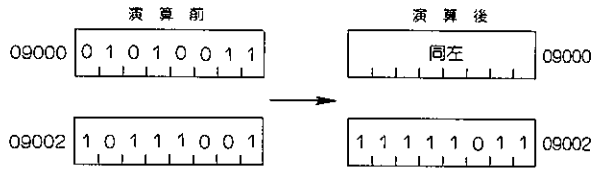
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">F-14 OR</td> <td style="text-align: center;">S</td> <td style="text-align: center;">D</td> </tr> </table>			F-14 OR	S	D
F-14 OR	S	D				
機能	レジスタSの内容(8ビットデータ)とレジスタDの内容(8ビットデータ)の論理和をとり、レジスタDに格納する。					
演算内容	SUD→D					
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 774#1 000000~037777 774#1 @000000~@037774					
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 774#1 000000~037777 774#1 @000000~@037774					
演算条件	入力信号の立上り(OFF→ON)					
演算後	Sの内容	不変				
	Dの内容	演算結果				
	フラグ	不変				

(解説)



命令	
STR	04002
F-14	09000 09002

入力条件04002がOFF→ONの変化時に、レジスタ09000の8ビットの内容とレジスタ09002の8ビットの内容の論理和(OR)をとり、レジスタ09002に格納します。レジスタ09000の内容は不変です。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)

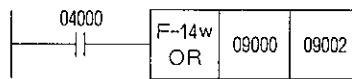
ORの真理値表

シンボル	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	1

F-14w OR レジスタ間(1ワード)の論理和 (OR)

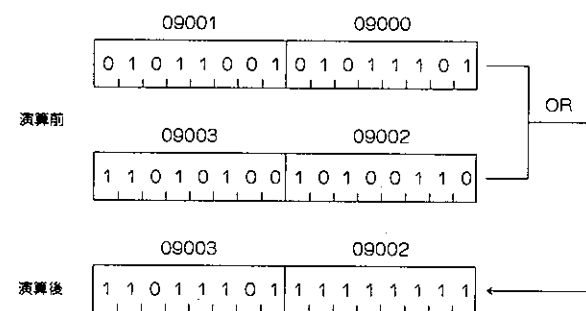
シンボル	<table border="1"> <tr> <td>F-14w</td> <td>S</td> <td>D</td> </tr> <tr> <td>OR</td> <td></td> <td></td> </tr> </table>			F-14w	S	D	OR										
F-14w	S	D															
OR																	
機能	レジスタS、S+1の内容(16ビットデータ)とレジスタD、D+1の内容(16ビットデータ)の論理和をとり、レジスタD、D+1に格納する。																
演算内容	S、S+1 U D、D+1 → D、D+1																
Sの使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>774#1 000000~037776</td> <td>774#1 @000000~@037774</td> </tr> </table>			コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	774#1 000000~037776	774#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																
コ2000~コ7576	@コ2000~@コ7574																
b0000~b1776	@b0000~@b1774																
b2000~b3776	@b2000~@b3774																
09000~99776	@09000~@99774																
E0000~E7776	@E0000~@E7774																
774#1 000000~037776	774#1 @000000~@037774																
Dの使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>774#1 000000~037776</td> <td>774#1 @000000~@037774</td> </tr> </table>			コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	774#1 000000~037776	774#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																
コ2000~コ7576	@コ2000~@コ7574																
b0000~b1776	@b0000~@b1774																
b2000~b3776	@b2000~@b3774																
09000~99776	@09000~@99774																
E0000~E7776	@E0000~@E7774																
774#1 000000~037776	774#1 @000000~@037774																
演算条件	入力信号の立上り(OFF→ON)																
演算後	S、S+1の内容	不変															
	D、D+1の内容	演算結果															
	フラグ	不変															

(解説)



命令	
STR	04000
F-14w	09000
	09002

入力条件04000がOFF→ONの変化時に、レジスタ09000、09001の内容(16ビットデータ)とレジスタ09002、09003の内容(16ビットデータ)の論理和をとり、レジスタ09002、09003に格納します。レジスタ09000、09001の内容は不変です。



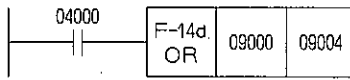
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

**F-14d
OR**

**レジスタ間(2ワード)の論理和
(OR)**

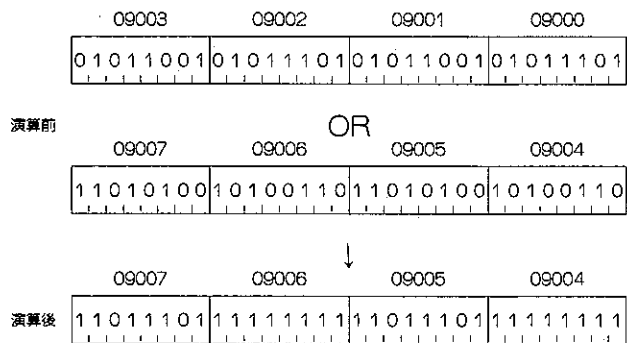
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-14d OR</td><td style="padding: 2px;">S</td><td style="padding: 2px;">D</td></tr></table>		F-14d OR	S	D
F-14d OR	S	D			
機能	レジスタS~S+3の内容(32ビットデータ)とレジスタD~D+3の内容(32ビットデータ)の論理和をとり、レジスタD~D+3に格納する。				
演算内容	S~S+3UD~D+3→D~D+3				
Sの使用範囲	コ0000~コ1574 ; @コ0000~@コ1574 コ2000~コ7574 ; @コ2000~@コ7574 b0000~b1774 ; @b0000~@b1774 b2000~b3774 ; @b2000~@b3774 09000~99774 ; @09000~@99774 E0000~E7774 ; @E0000~@E7774 771M1 000000~037774 ; 771M1 @000000~@037774				
Dの使用範囲	コ0000~コ1574 ; @コ0000~@コ1574 コ2000~コ7574 ; @コ2000~@コ7574 b0000~b1774 ; @b0000~@b1774 b2000~b3774 ; @b2000~@b3774 09000~99774 ; @09000~@99774 E0000~E7774 ; @E0000~@E7774 771M1 000000~037774 ; 771M1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	S~S+3の内容	不変			
	D~D+3の内容	演算結果			
	フラグ	不変			

(解説)



命令	
STR	04000
F-14d	09000 09004

入力条件04000がOFF→ONの変化時に、レジスタ09000~09003の内容(32ビットデータ)とレジスタ09004~09007の内容(32ビットデータ)の論理和をとり、レジスタ09004~09007に格納します。レジスタ09000~09003の内容は不変です。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

Fc14
OR

レジスタと8進定数(1バイト)の論理和
(OR)

シンボル			<p>(解説)</p>	<table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>04001</td> </tr> <tr> <td>Fc14</td> <td>123</td> </tr> <tr> <td></td> <td>09002</td> </tr> </tbody> </table>	命 令		STR	04001	Fc14	123		09002
命 令												
STR	04001											
Fc14	123											
	09002											
機能	8進定数nとレジスタDの内容の論理和をとりレジスタDに格納する。		<p>入力条件04001がOFF→ONの変化時に、8進定数123とレジスタ09002の内容の論理和(OR)をとり、レジスタ09002に格納します。</p>									
演算内容	nUD→D											
nの使用範囲	000~377		<p>演算前</p>									
Dの使用範囲	コ0000~コ1577! @コ0000~@コ1574 コ2000~コ7577! @コ2000~@コ7574 b0000~b1777! @b0000~@b1774 b2000~b3777! @b2000~@b3774 09000~99777! @09000~@99774 E0000~E7777! @E0000~@E7774 771#1 00000~03777! 771#1 @00000~@03774											
演算条件	入力信号の立上り(OFF→ON)		<table border="1"> <tr> <td>演算後</td> <td>Dの内容</td> <td>演算結果</td> </tr> <tr> <td></td> <td>フラグ</td> <td>不変</td> </tr> </table>	演算後	Dの内容	演算結果		フラグ	不変			
演算後	Dの内容	演算結果										
	フラグ	不変										

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

ORの真理値表

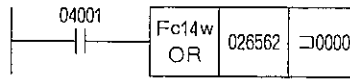
シンボル	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	1

**Fc14w
OR**

レジスタと8進定数(1ワード)の論理和
(OR)

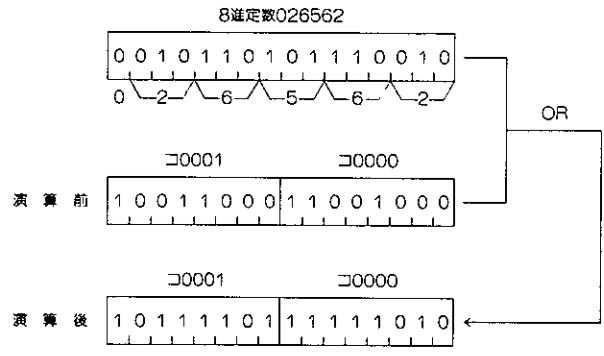
シンボル		
機能	8進定数 n とレジスタ D、D+1 の内容 (16ビットデータ) の論理和をとり、レジスタ D、D+1 に格納する。	
演算内容	n U D、D+1 → D、D+1	
n の使用範囲	000000~177777	
D の使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 77771 000000~037776 : 77771 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	D、D+1の内容	演算結果
	フラグ	不変

(解説)



命令	
STR	04001
Fc14w	026562 コ0000

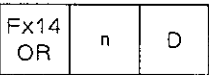
入力条件04001がOFF→ONの変化時、8進定数026562とレジスタコ0000、コ0001の内容(16ビットデータ)の論理和をとり、レジスタコ0000、コ0001に格納します。



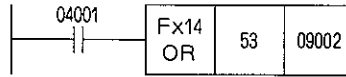
- コ0734~コ0737は特殊領域です。
(2・5 ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

**Fx14
OR**

**レジスタと16進定数(1バイト)の論理和
(OR)**

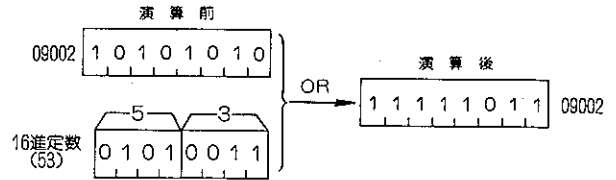
シンボル		
機能	16進定数nとレジスタDの内容の論理和をとりレジスタDに格納する。	
演算内容	nUD→D	
nの使用範囲	00~FF	
Dの使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 771#1 00000~03777 : 771#1 @00000~@03774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Dの内容	演算結果
	フラグ	不変

〔解説〕



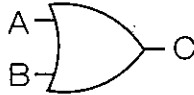
命 令	
STR	04001
Fx14	53
	09002

入力条件04001がOFF→ONの変化時に、16進定数53とレジスタ09002の内容の論理和 (OR)をとり、レジスタ09002に格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

ORの真理値表

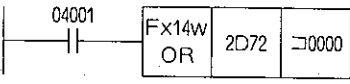
シンボル	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	1

**Fx14w
OR**

**レジスタと16進定数(1ワード)の論理和
(OR)**

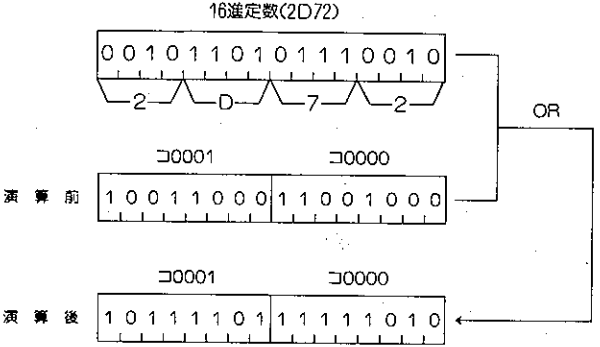
シンボル		
機能	16進定数nとレジスタD、D+1の内容(16ビットデータ)の論理和をとり、レジスタD、D+1に格納する。	
演算内容	nUD、D+1→D、D+1	
nの使用範囲	0000~FFFF	
Dの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 774M1 000000~037776 : 774M1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	D、D+1の内容	演算結果
	フラグ	不変

(解説)



命令	
STR	04001
Fx14w	2D72 コ0000

入力条件04001がOFF→ONの変化時、16進定数2D72とレジスタコ0000、コ0001の内容(16ビットデータ)の論理和をとり、レジスタコ0000、コ0001に格納します。



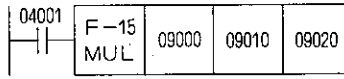
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

**F-15
MUL**

**レジスタ間(BCD4桁)の乗算
(MULtiply)**

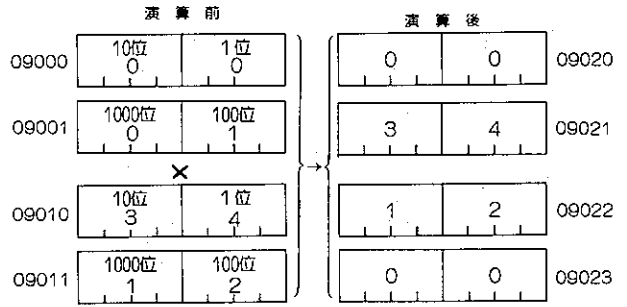
シンボル	<table border="1" style="display: inline-table;"> <tr> <td style="text-align: center;">F-15 MUL</td> <td style="text-align: center;">S₁</td> <td style="text-align: center;">S₂</td> <td style="text-align: center;">D</td> </tr> </table>				F-15 MUL	S ₁	S ₂	D										
F-15 MUL	S ₁	S ₂	D															
機能	レジスタS ₁ 、S ₁ +1の内容(BCD4桁)とレジスタS ₂ 、S ₂ +1の内容(BCD4桁)を乗算してレジスタDから4バイトに格納する。																	
演算内容	(S ₁ 、S ₁ +1)×(S ₂ 、S ₂ +1) →D、D+1、D+2、D+3																	
S ₁ の使用範囲	<table border="0" style="width: 100%;"> <tr> <td style="width: 50%;">コ0000~コ1576</td> <td style="width: 50%;">@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~09776</td> <td>@09000~@09774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>774#1 000000~037776</td> <td>774#1 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~09776	@09000~@09774	E0000~E7776	@E0000~@E7774	774#1 000000~037776	774#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																	
コ2000~コ7576	@コ2000~@コ7574																	
b0000~b1776	@b0000~@b1774																	
b2000~b3776	@b2000~@b3774																	
09000~09776	@09000~@09774																	
E0000~E7776	@E0000~@E7774																	
774#1 000000~037776	774#1 @000000~@037774																	
S ₂ の使用範囲	<table border="0" style="width: 100%;"> <tr> <td style="width: 50%;">コ0000~コ1576</td> <td style="width: 50%;">@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~09776</td> <td>@09000~@09774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>774#1 000000~037776</td> <td>774#1 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~09776	@09000~@09774	E0000~E7776	@E0000~@E7774	774#1 000000~037776	774#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																	
コ2000~コ7576	@コ2000~@コ7574																	
b0000~b1776	@b0000~@b1774																	
b2000~b3776	@b2000~@b3774																	
09000~09776	@09000~@09774																	
E0000~E7776	@E0000~@E7774																	
774#1 000000~037776	774#1 @000000~@037774																	
Dの使用範囲	<table border="0" style="width: 100%;"> <tr> <td style="width: 50%;">コ0000~コ1574</td> <td style="width: 50%;">@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7574</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1774</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3774</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~09774</td> <td>@09000~@09774</td> </tr> <tr> <td>E0000~E7774</td> <td>@E0000~@E7774</td> </tr> <tr> <td>774#1 000000~037774</td> <td>774#1 @000000~@037774</td> </tr> </table>				コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~09774	@09000~@09774	E0000~E7774	@E0000~@E7774	774#1 000000~037774	774#1 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																	
コ2000~コ7574	@コ2000~@コ7574																	
b0000~b1774	@b0000~@b1774																	
b2000~b3774	@b2000~@b3774																	
09000~09774	@09000~@09774																	
E0000~E7774	@E0000~@E7774																	
774#1 000000~037774	774#1 @000000~@037774																	
演算条件	入力信号の立上り(OFF→ON)																	
演算後	S ₁ 、S ₁ +1の内容	不変																
	S ₂ 、S ₂ +1の内容	不変																
	Dの内容	演算結果 (1の位と10の位)	レジスタS ₁ 、 S ₁ +1、S ₂ 、 S ₂ +1の内容 がBCDコードでない時不 変															
	D+1の内容	演算結果 (100の位と1,000の位)																
	D+2の内容	演算結果(10,000の 位と100,000の位)																
D+3の内容	演算結果(1,000,000の 位と10,000,000の位)																	
フラグ	レジスタS ₁ 、S ₁ +1、 S ₂ 、S ₂ +1の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354													
	BCDコード	0	0	0	0													
	BCDコードでない時			1	0													

(解説)



命 令	
STR	04001
F-15	09000
	09010
	09020

入力条件04001がOFF→ONの変化時に、レジスタ09000、09001のBCD4桁とレジスタ09010、09011のBCD4桁を乗算をして、レジスタ09020からの4バイトに格納します。



上記の演算は100×1234=123400を示しています。

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

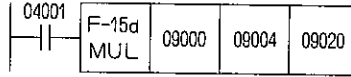
- S₁、S₁+1、S₂、S₂+1の内容がBCDコード以外の数値の場合、エラーフラグ(07355)がONし、乗算を行いません。(D~D+3の内容は不変です。)

**F-15d
MUL**

**レジスタ間(BCD8桁)の乗算
(MULtiply)**

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-15d MUL</td> <td style="padding: 2px;">S₁</td> <td style="padding: 2px;">S₂</td> <td style="padding: 2px;">D</td> </tr> </table>				F-15d MUL	S ₁	S ₂	D										
F-15d MUL	S ₁	S ₂	D															
機能	レジスタS ₁ ~S ₁ +3の内容(BCD8桁)とレジスタS ₂ ~S ₂ +3の内容(BCD8桁)を乗算してレジスタDから8バイトに格納する。																	
演算内容	(S ₁ ~S ₁ +3) × (S ₂ ~S ₂ +3) →D~D+7																	
S ₁ の使用範囲	<table border="0" style="width: 100%; font-size: small;"> <tr> <td>コ0000~コ1574</td><td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7574</td><td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1774</td><td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3774</td><td>@b2000~@b3774</td> </tr> <tr> <td>09000~09774</td><td>@09000~@09774</td> </tr> <tr> <td>E0000~E7774</td><td>@E0000~@E7774</td> </tr> <tr> <td>774M1 000000~037774</td><td>774M1 @000000~@037774</td> </tr> </table>				コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~09774	@09000~@09774	E0000~E7774	@E0000~@E7774	774M1 000000~037774	774M1 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																	
コ2000~コ7574	@コ2000~@コ7574																	
b0000~b1774	@b0000~@b1774																	
b2000~b3774	@b2000~@b3774																	
09000~09774	@09000~@09774																	
E0000~E7774	@E0000~@E7774																	
774M1 000000~037774	774M1 @000000~@037774																	
S ₂ の使用範囲	<table border="0" style="width: 100%; font-size: small;"> <tr> <td>コ0000~コ1574</td><td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7574</td><td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1774</td><td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3774</td><td>@b2000~@b3774</td> </tr> <tr> <td>09000~09774</td><td>@09000~@09774</td> </tr> <tr> <td>E0000~E7774</td><td>@E0000~@E7774</td> </tr> <tr> <td>774M1 000000~037774</td><td>774M1 @000000~@037774</td> </tr> </table>				コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~09774	@09000~@09774	E0000~E7774	@E0000~@E7774	774M1 000000~037774	774M1 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																	
コ2000~コ7574	@コ2000~@コ7574																	
b0000~b1774	@b0000~@b1774																	
b2000~b3774	@b2000~@b3774																	
09000~09774	@09000~@09774																	
E0000~E7774	@E0000~@E7774																	
774M1 000000~037774	774M1 @000000~@037774																	
Dの使用範囲	<table border="0" style="width: 100%; font-size: small;"> <tr> <td>コ0000~コ1570</td><td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7570</td><td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1770</td><td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3770</td><td>@b2000~@b3774</td> </tr> <tr> <td>09000~09770</td><td>@09000~@09774</td> </tr> <tr> <td>E0000~E7770</td><td>@E0000~@E7774</td> </tr> <tr> <td>774M1 000000~037770</td><td>774M1 @000000~@037774</td> </tr> </table>				コ0000~コ1570	@コ0000~@コ1574	コ2000~コ7570	@コ2000~@コ7574	b0000~b1770	@b0000~@b1774	b2000~b3770	@b2000~@b3774	09000~09770	@09000~@09774	E0000~E7770	@E0000~@E7774	774M1 000000~037770	774M1 @000000~@037774
コ0000~コ1570	@コ0000~@コ1574																	
コ2000~コ7570	@コ2000~@コ7574																	
b0000~b1770	@b0000~@b1774																	
b2000~b3770	@b2000~@b3774																	
09000~09770	@09000~@09774																	
E0000~E7770	@E0000~@E7774																	
774M1 000000~037770	774M1 @000000~@037774																	
演算条件	入力信号の立上り(OFF→ON)																	
演算後	S ₁ ~S ₁ +3の内容	不変																
	S ₂ ~S ₂ +3の内容	不変																
演算後	D~D+7の内容	演算結果		レジスタS ₁ ~S ₁ +3、S ₂ ~S ₂ +3の内容がBCDコードでない時不変														
			MSB		LSB													
		D	10 ¹		10 ⁰													
		D+1	10 ³		10 ²													
		D+2	10 ⁵		10 ⁴													
		D+3	10 ⁷		10 ⁶													
		D+4	10 ⁹		10 ⁸													
		D+5	10 ¹¹		10 ¹⁰													
D+6	10 ¹³	10 ¹²																
D+7	10 ¹⁵	10 ¹⁴																
フラグ	レジスタS ₁ ~S ₁ +3、S ₂ ~S ₂ +3の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354													
	BCDコード	0	0	0	0													
	BCDコードでない時	0	0	1	0													

(解説)



命 令	
STR	04001
F-15d	09000
	09004
	09020

入力条件04001がOFF→ONの変化時に、レジスタ09000~09003のBCD8桁とレジスタ09004~09007のBCD8桁を乗算をして、レジスタ09020からの8バイトに格納します。

演 算 前		演 算 後							
09000	<table border="1" style="font-size: x-small;"> <tr><td>10¹</td><td>10⁰</td></tr> <tr><td>0</td><td>0</td></tr> </table>	10 ¹	10 ⁰	0	0	09020	<table border="1" style="font-size: x-small;"> <tr><td>0</td><td>0</td></tr> </table>	0	0
10 ¹	10 ⁰								
0	0								
0	0								
09001	<table border="1" style="font-size: x-small;"> <tr><td>10³</td><td>10²</td></tr> <tr><td>0</td><td>1</td></tr> </table>	10 ³	10 ²	0	1	09021	<table border="1" style="font-size: x-small;"> <tr><td>0</td><td>0</td></tr> </table>	0	0
10 ³	10 ²								
0	1								
0	0								
09002	<table border="1" style="font-size: x-small;"> <tr><td>10⁵</td><td>10⁴</td></tr> <tr><td>3</td><td>4</td></tr> </table>	10 ⁵	10 ⁴	3	4	09022	<table border="1" style="font-size: x-small;"> <tr><td>0</td><td>1</td></tr> </table>	0	1
10 ⁵	10 ⁴								
3	4								
0	1								
09003	<table border="1" style="font-size: x-small;"> <tr><td>10⁷</td><td>10⁶</td></tr> <tr><td>1</td><td>2</td></tr> </table>	10 ⁷	10 ⁶	1	2	09023	<table border="1" style="font-size: x-small;"> <tr><td>6</td><td>8</td></tr> </table>	6	8
10 ⁷	10 ⁶								
1	2								
6	8								
×									
09004	<table border="1" style="font-size: x-small;"> <tr><td>10¹</td><td>10⁰</td></tr> <tr><td>0</td><td>0</td></tr> </table>	10 ¹	10 ⁰	0	0	09024	<table border="1" style="font-size: x-small;"> <tr><td>8</td><td>0</td></tr> </table>	8	0
10 ¹	10 ⁰								
0	0								
8	0								
09005	<table border="1" style="font-size: x-small;"> <tr><td>10³</td><td>10²</td></tr> <tr><td>0</td><td>1</td></tr> </table>	10 ³	10 ²	0	1	09025	<table border="1" style="font-size: x-small;"> <tr><td>2</td><td>7</td></tr> </table>	2	7
10 ³	10 ²								
0	1								
2	7								
09006	<table border="1" style="font-size: x-small;"> <tr><td>10⁵</td><td>10⁴</td></tr> <tr><td>3</td><td>4</td></tr> </table>	10 ⁵	10 ⁴	3	4	09026	<table border="1" style="font-size: x-small;"> <tr><td>5</td><td>2</td></tr> </table>	5	2
10 ⁵	10 ⁴								
3	4								
5	2								
09007	<table border="1" style="font-size: x-small;"> <tr><td>10⁷</td><td>10⁶</td></tr> <tr><td>1</td><td>2</td></tr> </table>	10 ⁷	10 ⁶	1	2	09027	<table border="1" style="font-size: x-small;"> <tr><td>0</td><td>1</td></tr> </table>	0	1
10 ⁷	10 ⁶								
1	2								
0	1								

上記計算は
12340100×12340100=152278068010000
を示しています。

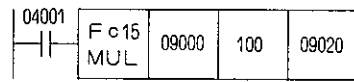
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- S₁~S₁+3、S₂~S₂+3の内容がBCDコード以外の数値の場合、エラーフラグ(07355)がONし、乗算を行いません。(D~D+7の内容は不変です。)
- S₁、S₂、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)

**Fc15
MUL**

**レジスタ(BCD4桁)とBCD定数(3桁)の乗算
(MULTiply)**

シンボル					
機能	レジスタSi、Si+1の内容(BCD4桁)と3桁のBCD定数nを乗算してレジスタDからの4バイトに格納する。				
演算内容	$(S_i, S_{i+1}) \times n \rightarrow D, D+1, D+2, D+3$				
Siの使用範囲	コ0000~コ1574 @コ0000~@コ1574 コ2000~コ7574 @コ2000~@コ7574 b0000~b1774 @b0000~@b1774 b2000~b3774 @b2000~@b3774 09000~99774 @09000~@99774 E0000~E7774 @E0000~@E7774 771#1 000000~037774 771#1 @000000~@037774				
nの使用範囲	000~999				
Dの使用範囲	コ0000~コ1574 @コ0000~@コ1574 コ2000~コ7574 @コ2000~@コ7574 b0000~b1774 @b0000~@b1774 b2000~b3774 @b2000~@b3774 09000~99774 @09000~@99774 E0000~E7774 @E0000~@E7774 771#1 000000~037774 771#1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	Si、Si+1の内容	不変			
	Dの内容	演算結果 (1の位と10の位)		レジスタSi、 Si+1の内容 がBCDコー ドでない時不 変	
	D+1の内 容	演算結果 (100位と1,000の位)			
	D+2の内 容	演算結果(10,000の位 と100,000の位)			
	D+3の内 容	演算結果(1,000,000位 と10,000,000の位)			
フラグ	レジスタSi、 Si+1の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
	BCDコード	0	0	0	0
	BCDコードで ない時			1	

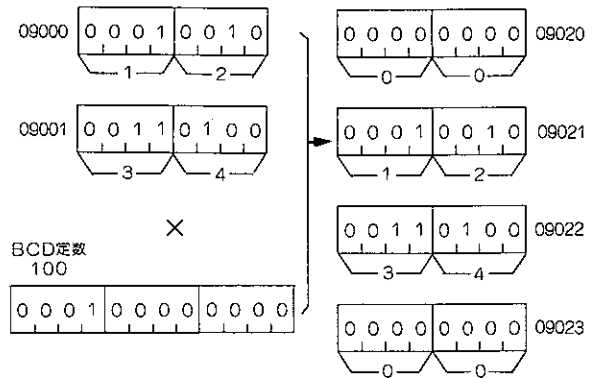
(解説)



命 令	
STR	04001
Fc15	09000
	100
	09020

入力条件04001がOFF→ONの変化時に、レジスタ09000、09001のBCD4桁とBCD定数100(3桁)の乗算をして、レジスタ09020から4バイトに格納します。

$3412 \times 100 = 341200$



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

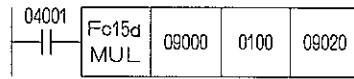
- Si、Si+1の内容がBCDコード以外の場合、エラーフラグ(07355)がONし、乗算は行いません。
(D~D+3の内容は不変です。)

**Fc15d
MUL**

**レジスタ(BCD8桁)とBCD定数(4桁)の乗算
(MULtiply)**

シンボル	Fc15d MUL S _i n D																										
機能	レジスタS _i ~S _i +3の内容(BCD8桁)と4桁のBCD定数nを乗算してレジスタDからの8/バイトに格納する。																										
演算内容	(S~S+3)×n→D~D+5																										
S _i の使用範囲	コ0000~コ1574 @コ0000~@コ1574 コ2000~コ7574 @コ2000~@コ7574 b0000~b1774 @b0000~@b1774 b2000~b3774 @b2000~@b3774 09000~99774 @09000~@99774 E0000~E7774 @E0000~@E7774 771#1 000000~037774 771#1 @000000~@037774																										
nの使用範囲	0000~9999																										
Dの使用範囲	コ0000~コ1570 @コ0000~@コ1574 コ2000~コ7570 @コ2000~@コ7574 b0000~b1770 @b0000~@b1774 b2000~b3770 @b2000~@b3774 09000~99770 @09000~@99774 E0000~E7770 @E0000~@E7774 771#1 000000~037770 771#1 @000000~@037774																										
演算条件	入力信号の立上り(OFF→ON)																										
演算後	S _i ~S _i +3の内容	不変																									
	D~D+5の内容	演算結果 <table border="1"> <tr><td>D</td><td>10¹</td><td>10⁰</td></tr> <tr><td>D+1</td><td>10³</td><td>10²</td></tr> <tr><td>D+2</td><td>10⁵</td><td>10⁴</td></tr> <tr><td>D+3</td><td>10⁷</td><td>10⁶</td></tr> <tr><td>D+4</td><td>10⁹</td><td>10⁸</td></tr> <tr><td>D+5</td><td>10¹¹</td><td>10¹⁰</td></tr> <tr><td>D+6</td><td>0</td><td>0</td></tr> <tr><td>D+7</td><td>0</td><td>0</td></tr> </table>	D	10 ¹	10 ⁰	D+1	10 ³	10 ²	D+2	10 ⁵	10 ⁴	D+3	10 ⁷	10 ⁶	D+4	10 ⁹	10 ⁸	D+5	10 ¹¹	10 ¹⁰	D+6	0	0	D+7	0	0	レジスタS _i ~S _i +3の内容がBCDコードでない時不変
D	10 ¹	10 ⁰																									
D+1	10 ³	10 ²																									
D+2	10 ⁵	10 ⁴																									
D+3	10 ⁷	10 ⁶																									
D+4	10 ⁹	10 ⁸																									
D+5	10 ¹¹	10 ¹⁰																									
D+6	0	0																									
D+7	0	0																									
フラグ	レジスタS _i ~S _i +3の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354																						
	BCDコード1	0	0	0	0																						
	BCDコードでない時	0	0	1	0																						

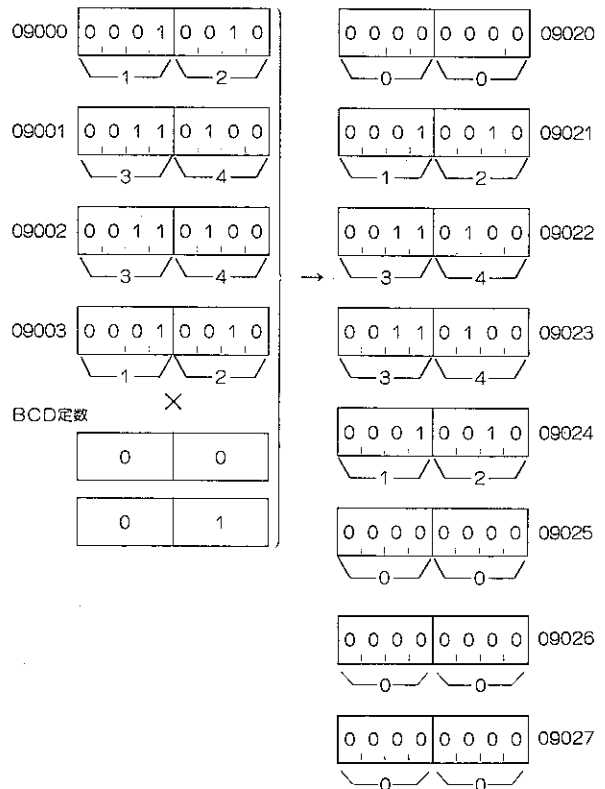
(解 説)



命 令	
STR	04001
Fc15d	09000
	0100
	09020

入力条件04001がOFF→ONの変化時に、レジスタ09000~09003のBCD8桁とBCD定数0100(4桁)の乗算をして、レジスタ09020から8バイトに格納します。

3412×100=341200



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

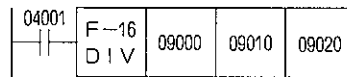
- S_i~S_i+3の内容がBCDコード以外の場合、エラーフラグ(07355)がONし、乗算は行いません。(D~D+5の内容は不変です。)

**F-16
DIV**

**レジスタ(BCD4桁)とレジスタ(BCD2桁)の除算
(DIVide)**

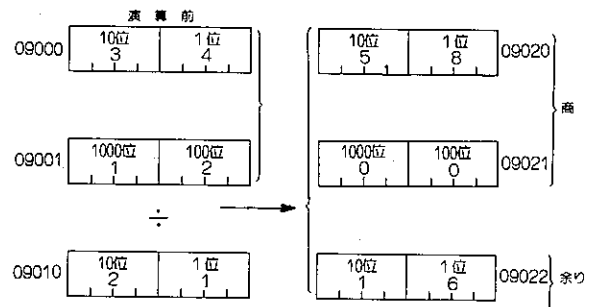
シンボル					
機能	レジスタS ₁ 、S ₁ +1の内容(BCD4桁)をレジスタS ₂ の内容(BCD2桁)で除算し、レジスタDからの2バイトに商を3バイト目に余を格納する。				
演算内容	(S ₁ 、S ₁ +1)÷S ₂ →D、D+1、D+2				
S ₁ の使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~09776 : @09000~@09774 E0000~E776 : @E0000~@E774 771M1 000000~03776 : 771M1 @000000~@03774				
S ₂ の使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~09777 : @09000~@09774 E0000~E777 : @E0000~@E774 771M1 000000~03777 : 771M1 @000000~@03774				
Dの使用範囲	コ0000~コ1575 : @コ0000~@コ1574 コ2000~コ7575 : @コ2000~@コ7574 b0000~b1775 : @b0000~@b1774 b2000~b3775 : @b2000~@b3774 09000~09775 : @09000~@09774 E0000~E775 : @E0000~@E774 771M1 000000~03775 : 771M1 @000000~@03774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	S ₁ 、S ₁ +1の内容	不変			
	S ₂ の内容	不変			
	Dの内容	演算結果の商 (1の位と10の位)	レジスタS ₁ 、S ₁ +1、S ₂ の内容がBCDコードでない時、S ₂ の内容が00の時不変		
	D+1の内容	演算結果の商 (100の位と1,000の位)			
D+2の内容	演算結果の除				
フラグ	レジスタS ₁ 、S ₁ +1、S ₂ の内容 BCDコード ●BCDコードでない時 ●S ₂ の内容が00の時	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
		0	0	0	0
				1	0

(解説)



命 令	
STR	04001
F-16	09000
	09010
	09020

入力条件04001がOFF→ONの変化時に、レジスタ09000、09001のBCD4桁をレジスタ09010のBCD2桁で除算をし、レジスタ09020からの2バイトに商を入れ、3バイト目に余りを入れます。



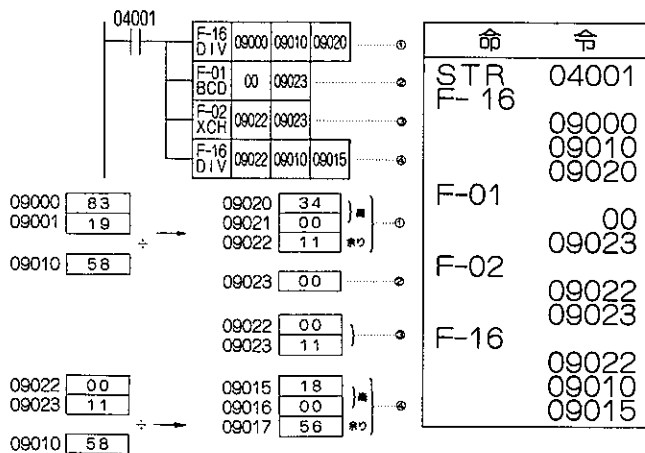
上記の演算は1234÷21=58余り16を示しています。

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- S₁、S₁+1、S₂の内容がBCDコード以外の数値の場合、エラーフラグ(07355)がONし、除算を行いません。(D、D+1、D+2の内容は不変です。)

- 分子(分母(S₁<S₂、S₁+1=0)の時、演算結果の商(D、D+1の内容)は0となり、余り(D+2の内容)は、分子(S₁の内容)となります。例えば20÷30を実行すると、答えは0余り20となります。

参考 小数点以下2桁を求めるときは次のようなプログラムを組んでください。

例 $1983 \div 58 = 34.18$ 余り0.56



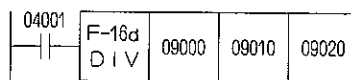
- ① 入力条件04001がOFF→ONのとき、レジスタ09000、09001の内容をレジスタ09010の内容で除算し結果は09020、09021に商を格納し、09022に余りを格納します。
- ② 09023に00のデータを入れ、09022と09023の内容を交換し、余りを、千、百の位に変換します。
- ③ ②のデータを再度09010の内容で除算し、09015、09016に商を09017に余りを格納します。09015に格納したデータが小数点以下の2桁になります。

**F-16d
DIV**

**レジスタ(BCD8桁)とレジスタ(BCD8桁)の除算
(DIVide)**

シンボル					
機能	レジスタS ₁ ~S ₁ +3の内容(BCD8桁)をレジスタS ₂ ~S ₂ +3の内容(BCD8桁)で除算し、レジスタD~D+3に商をD+4~D+7に余を格納する。				
演算内容	(S ₁ ~S ₁ +3)÷S ₂ , S ₂ +1→D~D+7				
S ₁ の使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 774#1 000000~037774 : 774#1 @000000~@037774				
S ₂ の使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 774#1 000000~037774 : 774#1 @000000~@037774				
Dの使用範囲	コ0000~コ1570 : @コ0000~@コ1574 コ2000~コ7570 : @コ2000~@コ7574 b0000~b1770 : @b0000~@b1774 b2000~b3770 : @b2000~@b3774 09000~99770 : @09000~@99774 E0000~E7770 : @E0000~@E7774 774#1 000000~037770 : 774#1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	S ₁ ~S ₁ +3の内容	不変			
	S ₂ ~S ₂ +3の内容	不変			
	D~D+3の内容	演算結果の商(BCD8桁)	レジスタS ₁ ~S ₁ +3, S ₂ ~S ₂ +3の内容がBCDコードでない時、S ₂ ~S ₂ +3の内容が0の時不変		
	D+4~D+7の内容	演算結果の余(BCD8桁)			
フラグ	レジスタS ₁ ~S ₁ +3, S ₂ ~S ₂ +3の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
	BCDコード	0	0	0	0
	●BCDコードでない時 ●S ₂ の内容が0の時			1	

(解説)



命令	
STR	04001
F-16d	09000
	09010
	09020

入力条件04001がOFF→ONの変化時に、レジスタ09000~09003のBCD8桁をレジスタ09010~09013のBCD4桁で除算をし、レジスタ09020~09023に商を09024~09027に余りを格納します。

演算前		演算後	
09000	$\begin{matrix} 10^1 & 10^0 \\ 7 & 8 \end{matrix}$	$\begin{matrix} 10^1 & 10^0 \\ 5 & 7 \end{matrix}$	09020
09001	$\begin{matrix} 10^3 & 10^2 \\ 5 & 6 \end{matrix}$	$\begin{matrix} 10^3 & 10^2 \\ 2 & 8 \end{matrix}$	09021
09002	$\begin{matrix} 10^5 & 10^4 \\ 3 & 4 \end{matrix}$	$\begin{matrix} 10^5 & 10^4 \\ 0 & 0 \end{matrix}$	09022
09003	$\begin{matrix} 10^7 & 10^6 \\ 1 & 2 \end{matrix}$	$\begin{matrix} 10^7 & 10^6 \\ 0 & 0 \end{matrix}$	09023
÷ →			
09010	$\begin{matrix} 10^1 & 10^0 \\ 2 & 1 \end{matrix}$	$\begin{matrix} 10^1 & 10^0 \\ 8 & 1 \end{matrix}$	09024
09011	$\begin{matrix} 10^3 & 10^2 \\ 4 & 3 \end{matrix}$	$\begin{matrix} 10^3 & 10^2 \\ 0 & 5 \end{matrix}$	09025
09012	$\begin{matrix} 10^5 & 10^4 \\ 0 & 0 \end{matrix}$	$\begin{matrix} 10^5 & 10^4 \\ 0 & 0 \end{matrix}$	09026
09013	$\begin{matrix} 10^7 & 10^6 \\ 0 & 0 \end{matrix}$	$\begin{matrix} 10^7 & 10^6 \\ 0 & 0 \end{matrix}$	09027

上記の演算は12345678÷4321=2857
余りは581を示しています。

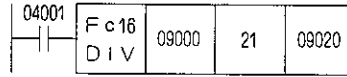
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- S₁~S₁+3, S₂~S₂+3の内容がBCDコード以外の数値の場合、エラーフラグ(07355)がONし、除算を行いません。(D~D+7の内容は不変です。)
- 分子(分母(S₁~S₁+3(S₂~S₂+3)の時、演算結果の商(D~D+3の内容)は0となり、余り(D+4~D+7の内容)は、分子(S₁~S₁+3の内容)となります。例えば20÷30を実行すると、答は0余り20となります。
- S₁, S₂, Dには必ず偶数アドレスを設定してください。(コ0011, 19003等は禁止)

**Fc16
DIV**

**レジスタ(BCD4桁)とBCD定数(2桁)の除算
(DIVide)**

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="text-align: center;">Fc16 DIV</td><td style="text-align: center;">S_i</td><td style="text-align: center;">n</td><td style="text-align: center;">D</td></tr></table>				Fc16 DIV	S _i	n	D												
Fc16 DIV	S _i	n	D																	
機能	レジスタS _i 、S _i +1の内容(BCD4桁)を2桁のBCD定数nで除算し、レジスタDから2バイトに商を3バイト目に余を格納する。																			
演算内容	$(S_i, S_{i+1}) \div n \rightarrow D, D+1, D+2$																			
S _i の使用範囲	<table border="0" style="font-size: small;"> <tr><td>コ0000~コ1576</td><td>コ2000~コ7576</td><td>コ0000~コ1574</td><td>コ2000~コ7574</td></tr> <tr><td>b0000~b1776</td><td>b2000~b7776</td><td>b0000~b1774</td><td>b2000~b7774</td></tr> <tr><td>09000~099776</td><td>E0000~E7776</td><td>09000~099774</td><td>E0000~E7774</td></tr> <tr><td>774#1 000000~037776</td><td>774#1 @000000~@037774</td><td></td><td></td></tr> </table>				コ0000~コ1576	コ2000~コ7576	コ0000~コ1574	コ2000~コ7574	b0000~b1776	b2000~b7776	b0000~b1774	b2000~b7774	09000~099776	E0000~E7776	09000~099774	E0000~E7774	774#1 000000~037776	774#1 @000000~@037774		
コ0000~コ1576	コ2000~コ7576	コ0000~コ1574	コ2000~コ7574																	
b0000~b1776	b2000~b7776	b0000~b1774	b2000~b7774																	
09000~099776	E0000~E7776	09000~099774	E0000~E7774																	
774#1 000000~037776	774#1 @000000~@037774																			
nの使用範囲	00~99																			
Dの使用範囲	<table border="0" style="font-size: small;"> <tr><td>コ0000~コ1575</td><td>コ2000~コ7575</td><td>コ0000~コ1574</td><td>コ2000~コ7574</td></tr> <tr><td>b0000~b1775</td><td>b2000~b7775</td><td>b0000~b1774</td><td>b2000~b7774</td></tr> <tr><td>09000~099775</td><td>E0000~E7775</td><td>09000~099774</td><td>E0000~E7774</td></tr> <tr><td>774#1 000000~037775</td><td>774#1 @000000~@037774</td><td></td><td></td></tr> </table>				コ0000~コ1575	コ2000~コ7575	コ0000~コ1574	コ2000~コ7574	b0000~b1775	b2000~b7775	b0000~b1774	b2000~b7774	09000~099775	E0000~E7775	09000~099774	E0000~E7774	774#1 000000~037775	774#1 @000000~@037774		
コ0000~コ1575	コ2000~コ7575	コ0000~コ1574	コ2000~コ7574																	
b0000~b1775	b2000~b7775	b0000~b1774	b2000~b7774																	
09000~099775	E0000~E7775	09000~099774	E0000~E7774																	
774#1 000000~037775	774#1 @000000~@037774																			
演算条件	入力信号の立上り(OFF→ON)																			
演算後	S _i 、S _i +1の内容	不変																		
	Dの内容	演算結果の商 (1の位と10の位)	レジスタS _i 、S _i +1の内容がBCDコードでない時、nが00の時不変																	
	D+1の内容	演算結果の商 (100位と1,000の位)																		
	D+2の内容	演算結果の余																		
フラグ	レジスタS _i 、S _i +1、nの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354															
	BCDコード	0	0	0	0															
	●BCDコードでない時 ●nが00の時	0	0	1	0															

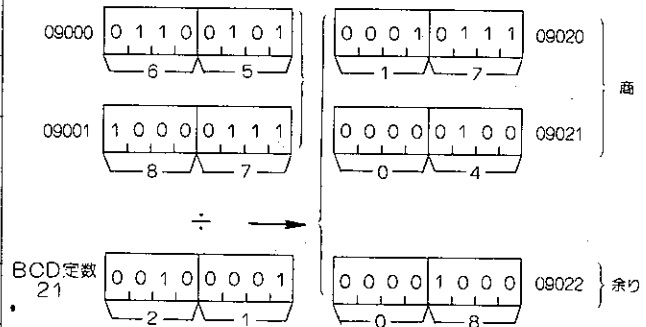
(解説)



命令	
STR Fc16	04001
	09000
	21
	09020

入力条件04001がOFF→ONの変化時に、レジスタ09000、09001のBCD4桁をBCD定数21で除算をし、レジスタ09020から2バイトに商を入れ、3バイト目に余りを格納します。

$8765 \div 21 = 417 \dots 8$



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)
- S_i、S_i+1の内容がBCDコード以外の数値の場合、エラーフラグ(07355)がONし、除算を行いません。(D、D+1、D+2の内容は不変です。)

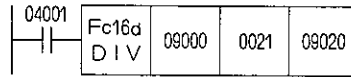
- 分子<分母(S_i<n、S_i+1=0)の時、演算結果の商(D、D+1の内容)は0となり、余り(D+2の内容)は、分子(S_iの内容)となります。例えば、20÷30を実行すると、商は0余り20となります。

**Fc16d
DIV**

**レジスタ(BCD8桁)とBCD定数(4桁)の除算
(DIVide)**

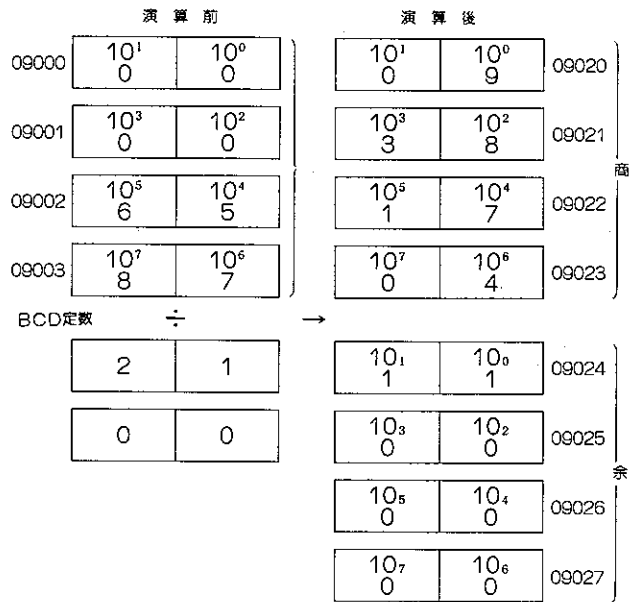
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">Fc16d DIV</td> <td style="padding: 2px;">S_i</td> <td style="padding: 2px;">n</td> <td style="padding: 2px;">D</td> </tr> </table>				Fc16d DIV	S _i	n	D										
Fc16d DIV	S _i	n	D															
機能	レジスタS _i ~S _i +3の内容(BCD8桁)と4桁のBCD定数nで除算し、レジスタD~D+3に商をD+4~D+7に余を格納する。																	
演算内容	(S _i ~S _i +3) ÷ n → D~D+7																	
S _i の使用範囲	<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="border-right: 1px dashed black; padding: 2px;">C0000~C1574</td> <td style="padding: 2px;">@C0000~@C1574</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">C2000~C7574</td> <td style="padding: 2px;">@C2000~@C7574</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">b0000~b1774</td> <td style="padding: 2px;">@b0000~@b1774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">b2000~b3774</td> <td style="padding: 2px;">@b2000~@b3774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">09000~99774</td> <td style="padding: 2px;">@09000~@99774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">E0000~E7774</td> <td style="padding: 2px;">@E0000~@E7774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">774#1 000000~037774</td> <td style="padding: 2px;">774#1 @000000~@037774</td> </tr> </table>				C0000~C1574	@C0000~@C1574	C2000~C7574	@C2000~@C7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	774#1 000000~037774	774#1 @000000~@037774
C0000~C1574	@C0000~@C1574																	
C2000~C7574	@C2000~@C7574																	
b0000~b1774	@b0000~@b1774																	
b2000~b3774	@b2000~@b3774																	
09000~99774	@09000~@99774																	
E0000~E7774	@E0000~@E7774																	
774#1 000000~037774	774#1 @000000~@037774																	
nの使用範囲	0000~9999																	
Dの使用範囲	<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="border-right: 1px dashed black; padding: 2px;">C0000~C1570</td> <td style="padding: 2px;">@C0000~@C1574</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">C2000~C7570</td> <td style="padding: 2px;">@C2000~@C7574</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">b0000~b1770</td> <td style="padding: 2px;">@b0000~@b1774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">b2000~b3770</td> <td style="padding: 2px;">@b2000~@b3774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">09000~99770</td> <td style="padding: 2px;">@09000~@99774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">E0000~E7770</td> <td style="padding: 2px;">@E0000~@E7774</td> </tr> <tr> <td style="border-right: 1px dashed black; padding: 2px;">774#1 000000~037770</td> <td style="padding: 2px;">774#1 @000000~@037774</td> </tr> </table>				C0000~C1570	@C0000~@C1574	C2000~C7570	@C2000~@C7574	b0000~b1770	@b0000~@b1774	b2000~b3770	@b2000~@b3774	09000~99770	@09000~@99774	E0000~E7770	@E0000~@E7774	774#1 000000~037770	774#1 @000000~@037774
C0000~C1570	@C0000~@C1574																	
C2000~C7570	@C2000~@C7574																	
b0000~b1770	@b0000~@b1774																	
b2000~b3770	@b2000~@b3774																	
09000~99770	@09000~@99774																	
E0000~E7770	@E0000~@E7774																	
774#1 000000~037770	774#1 @000000~@037774																	
演算条件	入力信号の立上り(OFF→ON)																	
演算後	S _i ~S _i +3の内容	不変																
	D~D+3の内容	演算結果の商(BCD8桁)	レジスタS _i ~S _i +3の内容がBCDコードでない時、nが00の時不変															
	D+4~D+7の内容	演算結果の余(BCD8桁)																
フラグ	レジスタS _i ~S _i +3, nの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354													
	BCDコード ●BCDコードでない時 ●nが00の時	0	0	0 1	0													

(解説)



命 令	
STR	04001
Fc16d	09000
	0021
	09020

入力条件04001がOFF→ONの変化時に、レジスタ09000~09003のBCD8桁をBCD定数0021で除算をし、レジスタ09020~09023に商を入れ09024~09027に余りを格納します。



上記の演算は87650000 ÷ 21 = 4173809
余りは11を示しています。

- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- S_i~S_i+3の内容がBCDコード以外の数値の場合、エラーフラグ(07355)がONし、除算を行いません。(D~D+7の内容は不変です。)
- 分子(分母(S_i~S_i+3 (n)の時、演算結果の商(D~D+7の内容)は0となり、余り(D~D+7の内容)は、分子(S_iの内容)となります。
例えば、20 ÷ 30を実行すると、答は0余り20となります。
- S_i、Dには必ず偶数アドレスを設定してください。
(C0011、19003等は禁止)

**F-17
XNR**

**レジスタ間(1バイト)の一致
(eXclusive NoR)**

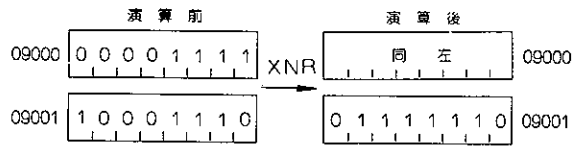
シンボル	— F-17 XNR S D		
機能	レジスタSの内容とレジスタDの内容の否定排他的論理和をとりレジスタDに格納する。		
演算内容	$S \oplus D \rightarrow D$		
Sの使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 Q90000~Q99777 @Q90000~@Q99774 E00000~E77777 @E00000~@E77774 774M1 000000~037777 774M1 @000000~@037774		
Dの使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 Q90000~Q99777 @Q90000~@Q99774 E00000~E77777 @E00000~@E77774 774M1 000000~037777 774M1 @000000~@037774		
演算条件	入力信号の立上り(OFF→ON)		
演算後	Sの内容	不変	
	Dの内容	演算結果	
	フラグ	不変	

(解説)



命 令	
STR	04001
F-17	09000 09001

入力条件04001がOFF→ONの変化時に、レジスタ09000の内容とレジスタ09001の内容の否定排他的論理和(exclusive NOR)をとり、レジスタ09001に格納します。
レジスタ09000の内容は不変です。



09000と09001で一致したビット(0と0、1と1)は1に、不一致のビット(0と1)は0になります。

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

Exclusive NOR 真理値表

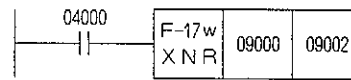
シンボル	A	B	C
	0	0	1
	1	0	0
	0	1	0
	1	1	1

F-17w
XNR

レジスタ間(1ワード)の一致
(eXclusive NoR)

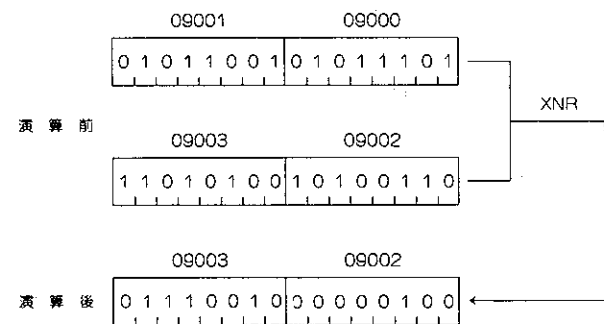
シンボル		
機能	レジスタS、S+1の内容(16ビットデータ)とレジスタD、D+1の内容(16ビットデータ)の否定排他的論理和をとり、レジスタD、D+1に格納する。	
演算内容	$S, S+1 \oplus D, D+1 \rightarrow D, D+1$	
Sの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 ファイル1 000000~037776 : ファイル1 @000000~@037774	
Dの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 ファイル1 000000~037776 : ファイル1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	S,S+1の内容	不変
	D,D+1の内容	演算結果
	フラグ	不変

(解説)



命令	
STR	04000
F-17w	09000 09002

入力条件04000がOFF→ONの変化時に、レジスタ09000、09001の内容(16ビットデータ)とレジスタ09002、09003の内容(16ビットデータ)の否定排他的論理和をとり、レジスタ09002、09003に格納します。レジスタ09000、09001の内容は不変です。



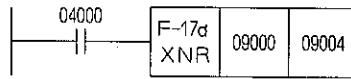
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

**F-17d
XNR**

**レジスタ間(2ワード)の一致
(eXclusive NoR)**

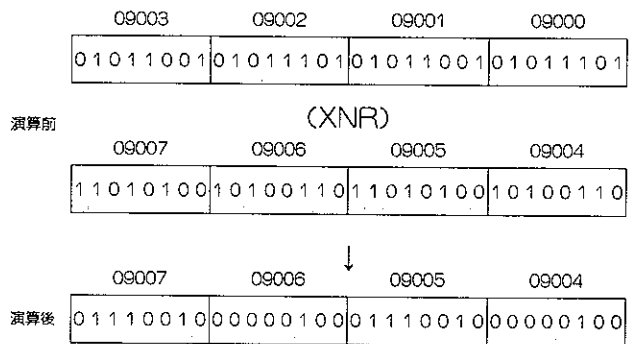
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-17d XNR</td> <td style="padding: 2px;">S</td> <td style="padding: 2px;">D</td> </tr> </table>		F-17d XNR	S	D											
F-17d XNR	S	D														
機能	レジスタS~S+3の内容(32ビットデータ)とレジスタD~D+3の内容(32ビットデータ)の否定排他的論理和をとり、レジスタD~D+3に格納する。															
演算内容	$S \sim S+3 \oplus D \sim D+3 \rightarrow D \sim D+3$															
Sの使用範囲	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border-right: 1px dashed black;">C0000~C1574</td> <td style="width: 50%;">@C0000~@C1574</td> </tr> <tr> <td style="border-right: 1px dashed black;">C2000~C7574</td> <td>@C2000~@C7574</td> </tr> <tr> <td style="border-right: 1px dashed black;">b0000~b1774</td> <td>@b0000~@b1774</td> </tr> <tr> <td style="border-right: 1px dashed black;">b2000~b3774</td> <td>@b2000~@b3774</td> </tr> <tr> <td style="border-right: 1px dashed black;">09000~99774</td> <td>@09000~@99774</td> </tr> <tr> <td style="border-right: 1px dashed black;">E0000~E7774</td> <td>@E0000~@E7774</td> </tr> <tr> <td style="border-right: 1px dashed black;">77#1 000000~037774</td> <td>77#1 @000000~@037774</td> </tr> </table>		C0000~C1574	@C0000~@C1574	C2000~C7574	@C2000~@C7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	77#1 000000~037774	77#1 @000000~@037774
C0000~C1574	@C0000~@C1574															
C2000~C7574	@C2000~@C7574															
b0000~b1774	@b0000~@b1774															
b2000~b3774	@b2000~@b3774															
09000~99774	@09000~@99774															
E0000~E7774	@E0000~@E7774															
77#1 000000~037774	77#1 @000000~@037774															
Dの使用範囲	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border-right: 1px dashed black;">C0000~C1574</td> <td style="width: 50%;">@C0000~@C1574</td> </tr> <tr> <td style="border-right: 1px dashed black;">C2000~C7574</td> <td>@C2000~@C7574</td> </tr> <tr> <td style="border-right: 1px dashed black;">b0000~b1774</td> <td>@b0000~@b1774</td> </tr> <tr> <td style="border-right: 1px dashed black;">b2000~b3774</td> <td>@b2000~@b3774</td> </tr> <tr> <td style="border-right: 1px dashed black;">09000~99774</td> <td>@09000~@99774</td> </tr> <tr> <td style="border-right: 1px dashed black;">E0000~E7774</td> <td>@E0000~@E7774</td> </tr> <tr> <td style="border-right: 1px dashed black;">77#1 000000~037774</td> <td>77#1 @000000~@037774</td> </tr> </table>		C0000~C1574	@C0000~@C1574	C2000~C7574	@C2000~@C7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	77#1 000000~037774	77#1 @000000~@037774
C0000~C1574	@C0000~@C1574															
C2000~C7574	@C2000~@C7574															
b0000~b1774	@b0000~@b1774															
b2000~b3774	@b2000~@b3774															
09000~99774	@09000~@99774															
E0000~E7774	@E0000~@E7774															
77#1 000000~037774	77#1 @000000~@037774															
演算条件	入力信号の立上り(OFF→ON)															
演算後	S~S+3の内容	不変														
	D~D+3の内容	演算結果														
	フラグ	不変														

(解説)



命 令	
STR	04000
F-17d	09000
	09004

入力条件04000がOFF→ONの変化時に、レジスタ09000~09003の内容(32ビットデータ)とレジスタ09004~09007の内容(32ビットデータ)の否定排他的論理和をとり、レジスタ09004~09007に格納します。レジスタ09000~09003の内容は不変です。



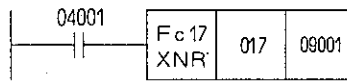
- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(C0011、19003等は禁止)

Fc17
XNR

レジスタと8進定数(1バイト)の一致 (eXclusive NoR)

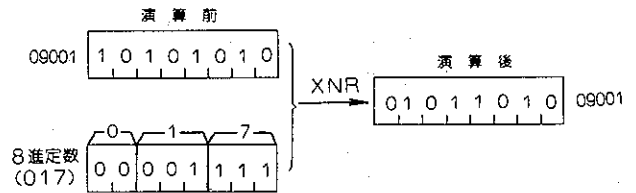
シンボル		
機能	8進定数nとレジスタDの内容の否定排他的論理和をとり、レジスタDに格納する。	
演算内容	$\bar{n} \oplus D \rightarrow D$	
nの使用範囲	000~377	
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 774#1 00000~03777 774#1 @00000~@03774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Dの内容	演算結果
	フラグ	不変

〔解説〕



命 令	
STR	04001
Fc17	017
	09001

入力条件04001がOFF→ONの変化時に、8進定数017とレジスタ09001の内容の否定排他的論理和(exclusive NOR)をとり、レジスタ09001に格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

Exclusive NOR 真理値表

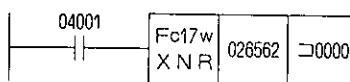
シンボル	A	B	C
	0	0	1
	1	0	0
	0	1	0
	1	1	1

**Fc17w
XNR**

レジスタと8進定数(1ワード)の一致
(eXclusive NOR)

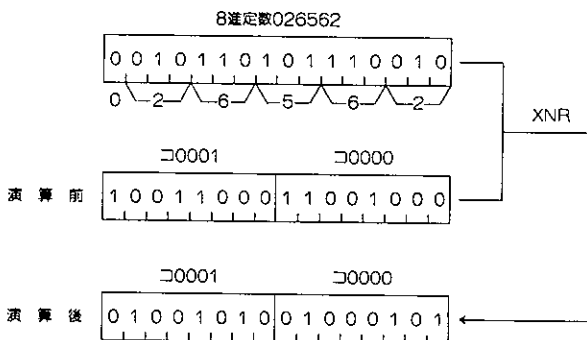
シンボル	$\overline{\text{Fc17w XNR}} \quad n \quad D$														
機能	8進定数nとレジスタD、D+1の内容(16ビットデータ)の否定排他的論理和をとり、レジスタD、D+1に格納する。														
演算内容	$n \oplus D, D+1 \rightarrow D, D+1$														
nの使用範囲	000000~177777														
Dの使用範囲	<table border="0"> <tr> <td>c00000~c15776</td> <td>@c00000~@c15774</td> </tr> <tr> <td>c20000~c75776</td> <td>@c20000~@c75774</td> </tr> <tr> <td>b00000~b17776</td> <td>@b00000~@b17774</td> </tr> <tr> <td>b20000~b37776</td> <td>@b20000~@b37774</td> </tr> <tr> <td>090000~997776</td> <td>@090000~@997774</td> </tr> <tr> <td>E00000~E77776</td> <td>@E00000~@E77774</td> </tr> <tr> <td>777776~037776</td> <td>777776 @000000~@037774</td> </tr> </table>	c00000~c15776	@c00000~@c15774	c20000~c75776	@c20000~@c75774	b00000~b17776	@b00000~@b17774	b20000~b37776	@b20000~@b37774	090000~997776	@090000~@997774	E00000~E77776	@E00000~@E77774	777776~037776	777776 @000000~@037774
c00000~c15776	@c00000~@c15774														
c20000~c75776	@c20000~@c75774														
b00000~b17776	@b00000~@b17774														
b20000~b37776	@b20000~@b37774														
090000~997776	@090000~@997774														
E00000~E77776	@E00000~@E77774														
777776~037776	777776 @000000~@037774														
演算条件	入力信号の立上り(OFF→ON)														
演算後	D、D+1の内容	演算結果													
	フラグ	不変													

(解説)



命 令	
STR	04001
Fc17w	026562
	C0000

入力条件04001がOFF→ONの変化時、8進定数026562とレジスタC0000、C0001の内容(16ビットデータ)の否定排他的論理和をとり、レジスタC0000、C0001に格納します。



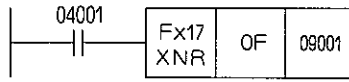
- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(C0011、19003等は禁止)

**Fx17
XNR**

**レジスタと16進定数(1バイト)の一致
(eXclusive NoR)**

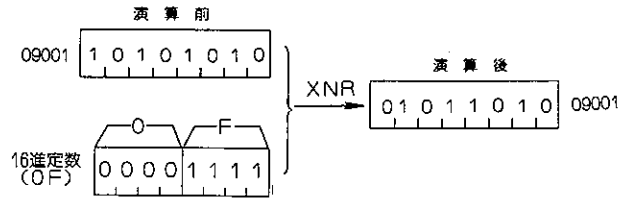
シンボル		
機能	16進定数 n とレジスタ D の内容の否定排他的論理和をとり、レジスタ D に格納する。	
演算内容	$\bar{n} \oplus D \rightarrow D$	
n の使用範囲	00~FF	
D の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 77イ#1 000000~037777 77イ#1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Dの内容	演算結果
	フラグ	不変

(解説)



命令	
STR	04001
Fx17	OF
	09001

入力条件04001がOFF→ONの変化時に、16進定数0Fとレジスタ09001の内容の否定排他的論理和(exclusive NOR)をとり、レジスタ09001に格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

Exclusive NOR 真理値表

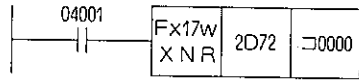
シンボル	A	B	C
	0	0	1
	1	0	0
	0	1	0
	1	1	1

**Fx17w
XNR**

**レジスタと16進定数(1ワード)の一致
(eXclusive NOR)**

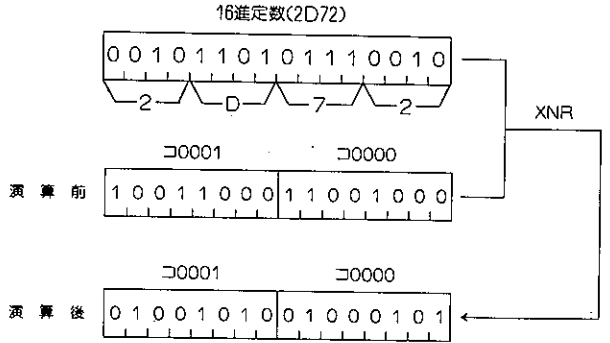
シンボル		
機能	16進定数nとレジスタD、D+1の内容(16ビットデータ)の否定排他的論理和をとり、レジスタD、D+1に格納する。	
演算内容	$n \oplus D, D+1 \rightarrow D, D+1$	
nの使用範囲	0000~FFFF	
Dの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 771#1 000000~037776 : 771#1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	D、D+1の内容	演算結果
	フラグ	不変

〔解説〕



命 令	
STR	04001
Fx17w	2D72
	コ0000

入力条件04001がOFF→ONの変化時、16進定数2D72とレジスタコ0000、コ0001の内容(16ビットデータ)の否定排他的論理和をとり、レジスタコ0000、コ0001に格納します。



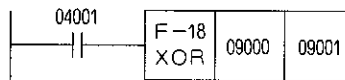
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

**F-18
XOR**

**レジスタ間(1バイト)の排他的論理和
(eXclusive OR)**

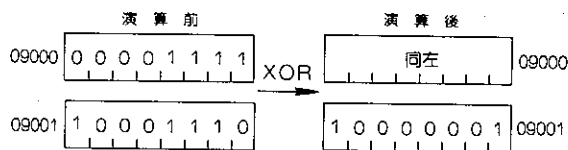
シンボル			
機能	レジスタSの内容とレジスタDの内容の排他的論理和をとり、レジスタDに格納する。		
演算内容	$S \oplus D \rightarrow D$		
Sの使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~997777 @0900000~@997774 E00000~E77777 @E000000~@E77774 771#1 000000~037777 771#1 @000000~@037774		
Dの使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~997777 @0900000~@997774 E00000~E77777 @E000000~@E77774 771#1 000000~037777 771#1 @000000~@037774		
演算条件	入力信号の立上り(OFF→ON)		
演算後	Sの内容	不変	
	Dの内容	演算結果	
	フラグ	不変	

(解説)



命 令	
STR	04001
F-18	09000 09001

入力条件04001がOFF→ONの変化時に、レジスタ09000の内容とレジスタ09001の内容の排他的論理和(exclusive OR)をとり、レジスタ09001に格納します。
レジスタ09000の内容は不変です。



09000と09001で不一致のビット(0と1)は1に、一致のビット(0と0、1と1)は0になります。

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

Exclusive OR 真理値表

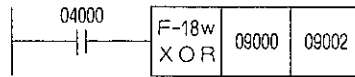
シンボル	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	0

**F-18w
XOR**

**レジスタ間(1ワード)の排他的論理和
(eXclusive OR)**

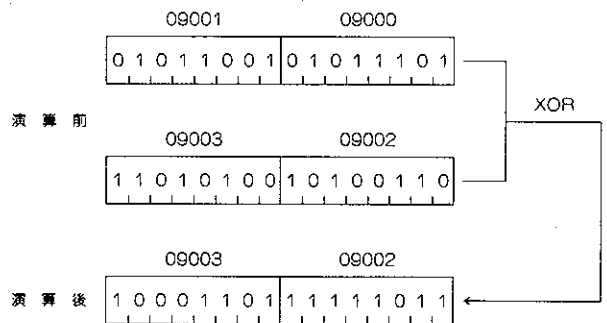
シンボル		
機能	レジスタS、S+1の内容(16ビットデータ)とレジスタD、D+1の内容(16ビットデータ)の排他的論理和をとり、レジスタD、D+1に格納する。	
演算内容	S、S+1 ⊕ D、D+1 → D、D+1	
Sの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 774#1 000000~037776 774#1 @000000~@037774	
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 774#1 000000~037776 774#1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	S、S+1の内容	不変
	D、D+1の内容	演算結果
	フラグ	不変

(解説)



命令	
STR	04000
F-18w	09000 09002

入力条件04000がOFF→ONの変化時に、レジスタ09000、09001の内容(16ビットデータ)とレジスタ09002、09003の内容(16ビットデータ)の排他的論理和をとり、レジスタ09002、09003に格納します。レジスタ09000、09001の内容は不変です。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

**F-18d
XOR**

**レジスタ間(2ワード)の排他的論理和
(eXclusive OR)**

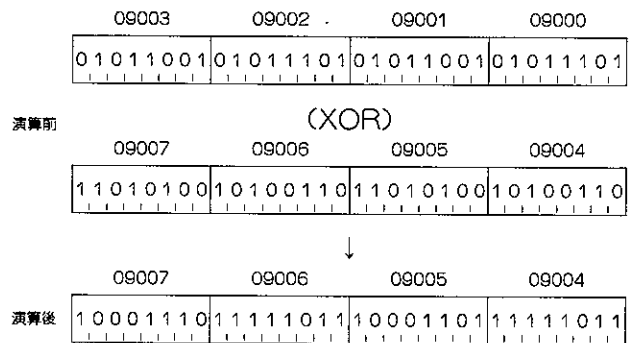
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-18d XOR</td> <td style="padding: 2px;">S</td> <td style="padding: 2px;">D</td> </tr> </table>		F-18d XOR	S	D
F-18d XOR	S	D			
機能	レジスタS~S+3の内容(32ビットデータ)とレジスタD~D+3の内容(32ビットデータ)の排他的論理和をとり、レジスタD~D+3に格納する。				
演算内容	S~S+3 ⊕ D~D+3 → D~D+3				
Sの使用範囲	コ0000~コ1574 ; @コ0000~@コ1574 コ2000~コ7574 ; @コ2000~@コ7574 b0000~b1774 ; @b0000~@b1774 b2000~b3774 ; @b2000~@b3774 09000~99774 ; @09000~@99774 E0000~E7774 ; @E0000~@E7774 ファイル1 000000~037774 ; ファイル1 @000000~@037774				
Dの使用範囲	コ0000~コ1574 ; @コ0000~@コ1574 コ2000~コ7574 ; @コ2000~@コ7574 b0000~b1774 ; @b0000~@b1774 b2000~b3774 ; @b2000~@b3774 09000~99774 ; @09000~@99774 E0000~E7774 ; @E0000~@E7774 ファイル1 000000~037774 ; ファイル1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	S~S+3の内容	不変			
	D~D+3の内容	演算結果			
	フラグ	不変			

(解説)



命 令	
STR	04000
F-18d	09000
	09004

入力条件04000がOFF→ONの変化時に、レジスタ09000~09003の内容(32ビットデータ)とレジスタ09004~09007の内容(32ビットデータ)の排他的論理和をとり、レジスタ09004~09007に格納します。レジスタ09000~09003の内容は不変です。



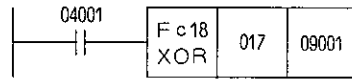
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

**Fc18
XOR**

レジスタと8進定数(1バイト)の排他的論理和
(eXclusive OR)

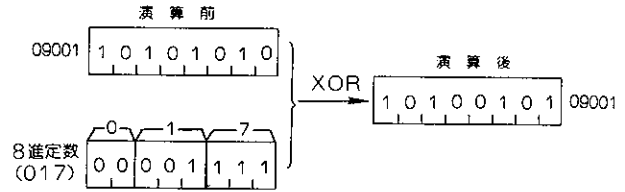
シンボル		
機能	8進定数nとレジスタDの内容の排他的論理和をとりレジスタDに格納する。	
演算内容	$n \oplus D \rightarrow D$	
nの使用範囲	000~377	
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 77777 000000~037777 77777 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Dの内容	演算結果
	フラグ	不変

〔解説〕



命 令	
STR	04001
Fc18	017 09001

入力条件04001がOFF→ONの変化時に、8進定数017とレジスタ09001の内容の排他的論理和(exclusive OR)をとり、レジスタ09001に格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

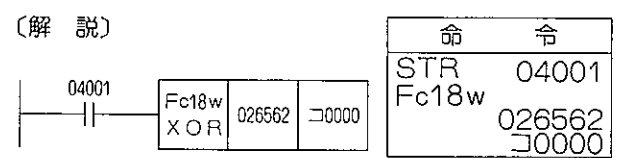
Exclusive OR 真理値表

シンボル	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	0

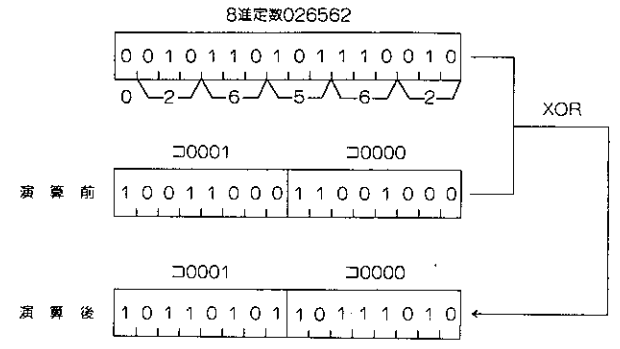
Fc18w XOR

レジスタと8進定数(1ワード)の排他的論理和 (eXclusive OR)

シンボル															
機能	8進定数 n とレジスタ D、D+1 の内容 (16ビットデータ) の排他的論理和をとり、レジスタ D、D+1 に格納する。														
演算内容	$n \oplus D, D+1 \rightarrow D, D+1$														
n の使用範囲	000000~177777														
D の使用範囲	<table style="font-size: small; border: none;"> <tr><td>コ0000~コ1576</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7576</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1776</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3776</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99776</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7776</td><td>@E0000~@E7774</td></tr> <tr><td>771M1 000000~037776</td><td>771M1 @000000~@037774</td></tr> </table>	コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771M1 000000~037776	771M1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574														
コ2000~コ7576	@コ2000~@コ7574														
b0000~b1776	@b0000~@b1774														
b2000~b3776	@b2000~@b3774														
09000~99776	@09000~@99774														
E0000~E7776	@E0000~@E7774														
771M1 000000~037776	771M1 @000000~@037774														
演算条件	入力信号の立上り(OFF→ON)														
演算後	D、D+1の内容	演算結果													
	フラグ	不変													



入力条件04001がOFF→ONの変化時に8進定数026562とレジスタコ0000、コ0001の内容(16ビットデータ)の排他的論理和をとり、レジスタコ0000、コ001に格納します。



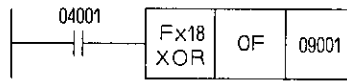
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

**Fx18
XOR**

**レジスタと16進定数(1バイト)の排他的論理和
(eXclusive OR)**

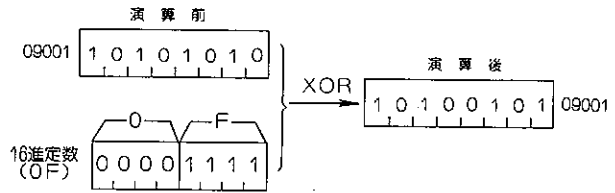
シンボル		
機能	16進定数nとレジスタDの内容の排他的論理和をとりレジスタDに格納する。	
演算内容	$n \oplus D \rightarrow D$	
nの使用範囲	00~FF	
Dの使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 777771 000000~037777 : 777771 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Dの内容	演算結果
	フラグ	不変

(解説)



命令	
STR	04001
Fx18	0F 09001

入力条件04001がOFF→ONの変化時に、16進定数0Fとレジスタ09001の内容の排他的論理和(exclusive OR)をとり、レジスタ09001に格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

Exclusive OR 真理値表

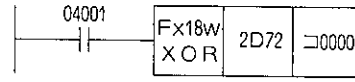
シンボル	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	0

**Fx18w
XOR**

**レジスタと16進定数(1ワード)の排他的論理和
(eXclusive OR)**

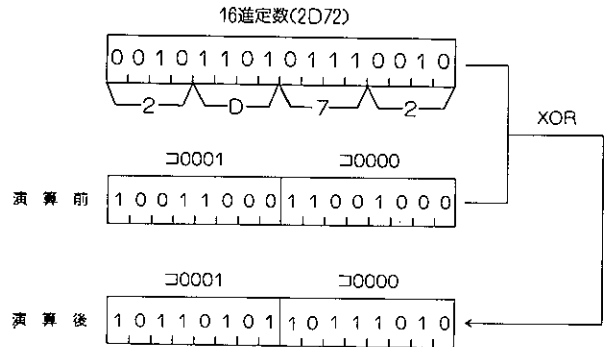
シンボル		
機能	16進定数nとレジスタD、D+1の内容(16ビットデータ)の排他的論理和をとり、レジスタD、D+1に格納する。	
演算内容	$n \oplus D, D+1 \rightarrow D, D+1$	
nの使用範囲	0000~FFFF	
Dの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 771W1 000000~037776 : 771W1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	D、D+1の内容	演算結果
	フラグ	不変

(解説)



命 令	
STR	04001
Fx18w	2D72 コ0000

入力条件04001がOFF→ONの変化時に16進定数2D72とレジスタコ0000、コ0001の内容(16ビットデータ)の排他的論理和をとり、レジスタコ0000、コ001に格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

F-20
(MD)

メンテナンスディスプレイ

F-20命令は、MD命令と同機能です。8・9ページ「MD
(メンテナンスディスプレイ)」を参照してください。

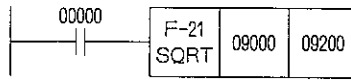
**F-21
SQRT**

**レジスタ(BCD8桁)の平方根
(SQuare RooT)**

(この命令はJW-31CUH/H1ではプログラムできません。)

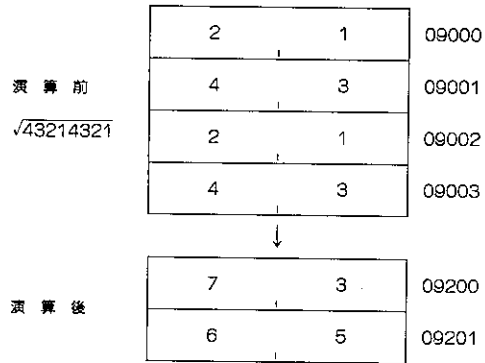
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-21 SQRT</td><td style="padding: 2px;">S</td><td style="padding: 2px;">D</td></tr></table>				F-21 SQRT	S	D
F-21 SQRT	S	D					
機能	レジスタS~S+3の内容(BCD8桁)の平方根を求め結果をレジスタDへ格納する。 ●小数点以下は切捨てる						
演算内容	$\sqrt{(S \sim S+3)} \rightarrow D, D+1$						
Sの使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 ファイル1 000000~037774 : ファイル1 @000000~@037774						
Dの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 ファイル1 000000~037776 : ファイル1 @000000~@037774						
演算条件	入力信号の立上り(OFF→ON)						
演算後	S ₁ ~S ₁ +3の内容	不変					
	Dの内容	演算結果					
	レジスタS ₁ ~S ₁ +3の内容がBCDコードでない時	レジスタS ₁ ~S ₁ +3の内容がBCDコードでない時不変					
フラグ	レジスタS ₁ ~S ₁ +3の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354		
	BCDコード	0	0	0	0		
	BCDコードでない時	0	0	1	0		

〔解説〕



命 令	
STR	00000
F-21	09000
	09200

入力条件00000がOFF→ONの変化時に、レジスタ09000~09003のBCD8桁の平方根を求め、レジスタ09200、09201へ格納します。



演算結果小数点以下は切り捨てる

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)
- S₁~S₁+3の内容がBCDコード以外の数値の場合、エラーフラグ(07355)がONし、演算を行いません。(D、D+1の内容は不変です。)
- S、Dのアドレスはかならず偶数アドレスを設定してください。(コ0001、19003等は禁止)

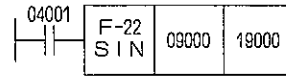
F-22
SIN

三角関数(SIN)の演算

(この命令はJW-31CUH/H1ではプログラムできません。)

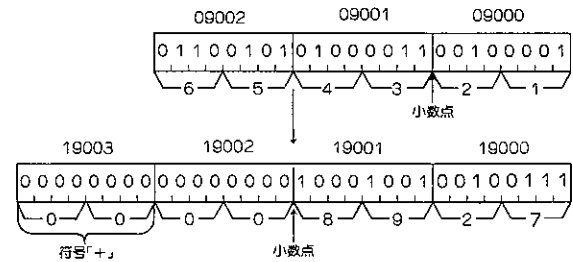
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="text-align: center;">F-22</td><td style="text-align: center;">S</td><td style="text-align: center;">D</td></tr><tr><td style="text-align: center;">SIN</td><td></td><td></td></tr></table>				F-22	S	D	SIN		
F-22	S	D								
SIN										
機能	レジスタS~S+2の内容(BCD6桁)の正弦(SIN)を求め、レジスタD~D+3にBCD8桁で格納する。									
演算内容	SIN(S~S+2)→D~D+3									
Sの使用範囲	コ0000~コ1574 @コ0000~@コ1574 コ2000~コ7574 @コ2000~@コ7574 b0000~b1774 @b0000~@b1774 b2000~b3774 @b2000~@b3774 09000~99774 @09000~@99774 E0000~E7774 @E0000~@E7774 774M1 000000~037774 774M1 @000000~@037774									
Dの使用範囲	コ0000~コ1574 @コ0000~@コ1574 コ2000~コ7574 @コ2000~@コ7574 b0000~b1774 @b0000~@b1774 b2000~b3774 @b2000~@b3774 09000~99774 @09000~@99774 E0000~E7774 @E0000~@E7774 774M1 000000~037774 774M1 @000000~@037774									
演算条件	入力信号の立上り(OFF→ON)									
演算前	Sの内容	角度の小数部 (BCD2桁)	角度の範囲は 0~9999.99°							
	S+1、S+2の内容	角度の整数部 (BCD4桁)								
演算中	S~S+2の内容	不変								
	D、D+1の内容	演算結果の小数部 (BCD4桁)	●演算結果の範囲は -1.0000~1.0000 ●S~S+2の内容がBCDでない時不変							
	D+2の内容	演算結果の整数部 (BCD2桁)								
D+3の内容	演算結果の符号 [00-正(+)] [80-負(-)] (BCD2桁)									
演算後	フ ラ グ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354				
		正(+)	0	0	0	1				
		負(-)	0	1	0	0				
		S~S+2の内容がBCDでない時	0	0	1	0				

(解説)



命 令	
STR	04001
F-22	09000 19000

入力条件04001がOFF→ONの変化時に、レジスタ09000~09002のBCD6桁データの正弦(SIN)を求め、演算結果をレジスタ19000~19003に格納します。



上記の演算は、 $SIN 6543.21^\circ \approx 0.8927$ を示しています。

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- 演算結果は、小数第5位を四捨五入します。

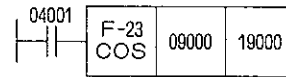
**F-23
COS**

三角関数(COS)の演算

(この命令はJW-31CUH/H1ではプログラムできません。)

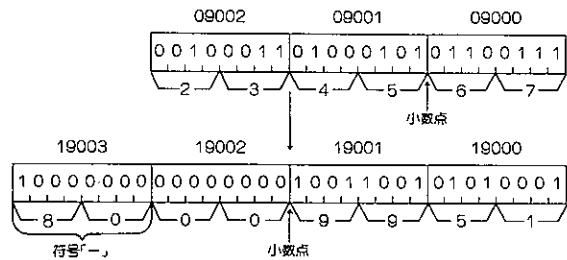
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-23</td><td>S</td><td>D</td></tr><tr><td>COS</td><td></td><td></td></tr></table>				F-23	S	D	COS		
F-23	S	D								
COS										
機能	レジスタS~S+2の内容(BCD6桁)の余弦(COS)を求め、レジスタD~D+3にBCD8桁で格納する。									
演算内容	COS(S~S+2)→D~D+3									
Sの使用範囲	コ0000~コ1575 @コ0000~@コ1574 コ2000~コ7575 @コ2000~@コ7574 b0000~b1775 @b0000~@b1774 b2000~b3775 @b2000~@b3774 09000~99775 @09000~@99774 E0000~E7775 @E0000~@E7774 77441 000000~037775 77441 @000000~@037774									
Dの使用範囲	コ0000~コ1574 @コ0000~@コ1574 コ2000~コ7574 @コ2000~@コ7574 b0000~b1774 @b0000~@b1774 b2000~b3774 @b2000~@b3774 09000~99774 @09000~@99774 E0000~E7774 @E0000~@E7774 77441 000000~037774 77441 @000000~@037774									
演算条件	入力信号の立上り(OFF→ON)									
演算前	Sの内容	角度の小数部 (BCD2桁)	角度の範囲は 0~9999.99°							
	S+1、S+2の内容	角度の整数部 (BCD4桁)								
演算中	S~S+2の内容	不変								
	D、D+1の内容	演算結果の小数部 (BCD4桁)	●演算結果の範囲は -1.0000~1.0000							
	D+2の内容	演算結果の整数部 (BCD2桁)	●S~S+2の内容が BCDでない時不変							
	D+3の内容	演算結果の符号 [00-正(+)] [80-負(-)] (BCD2桁)								
演算後	フ ラ グ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354				
		正(+)	0	0	0	1				
		負(-)	0	1	0	0				
	S~S+2の内容 がBCDでない時	0	0	1	0					

(解説)



命 令	
STR	04001
F-23	09000
	19000

入力条件04001がOFF→ONの変化時に、レジスタ09000~09002のBCD6桁データの余弦(COS)を求め、演算結果をレジスタ19000~19003に格納します。



上記の演算は、 $\text{COS}2345.67^\circ \approx -0.9951$ を示しています。

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- 演算結果は、小数第5位を四捨五入します。

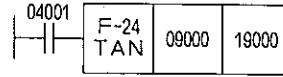
F-24 TAN

三角関数(TAN)の演算

(この命令はJW-31CUH/H1ではプログラムできません。)

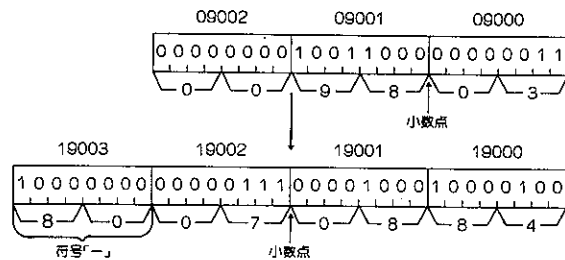
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="text-align: center;">F-24 TAN</td><td style="text-align: center;">S</td><td style="text-align: center;">D</td></tr></table>					F-24 TAN	S	D																		
F-24 TAN	S	D																								
機能	レジスタS~S+2の内容(BCD6桁)の正接(TAN)を求め、レジスタD~D+3にBCD8桁で格納する。																									
演算内容	TAN(S~S+2)→D~D+3																									
Sの使用範囲	<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">C0000~C1575</td> <td style="width: 33%;">@C0000~@C1574</td> <td style="width: 33%;">@C0000~@C1574</td> </tr> <tr> <td>C2000~C7575</td> <td>@C2000~@C7574</td> <td>@C2000~@C7574</td> </tr> <tr> <td>b0000~b1775</td> <td>@b0000~@b1774</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3775</td> <td>@b2000~@b3774</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99775</td> <td>@09000~@99774</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7775</td> <td>@E0000~@E7774</td> <td>@E0000~@E7774</td> </tr> <tr> <td>774#1 00000~03775</td> <td>774#1 @00000~@03774</td> <td>774#1 @00000~@03774</td> </tr> </table>					C0000~C1575	@C0000~@C1574	@C0000~@C1574	C2000~C7575	@C2000~@C7574	@C2000~@C7574	b0000~b1775	@b0000~@b1774	@b0000~@b1774	b2000~b3775	@b2000~@b3774	@b2000~@b3774	09000~99775	@09000~@99774	@09000~@99774	E0000~E7775	@E0000~@E7774	@E0000~@E7774	774#1 00000~03775	774#1 @00000~@03774	774#1 @00000~@03774
C0000~C1575	@C0000~@C1574	@C0000~@C1574																								
C2000~C7575	@C2000~@C7574	@C2000~@C7574																								
b0000~b1775	@b0000~@b1774	@b0000~@b1774																								
b2000~b3775	@b2000~@b3774	@b2000~@b3774																								
09000~99775	@09000~@99774	@09000~@99774																								
E0000~E7775	@E0000~@E7774	@E0000~@E7774																								
774#1 00000~03775	774#1 @00000~@03774	774#1 @00000~@03774																								
Dの使用範囲	<table border="0" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">C0000~C1574</td> <td style="width: 33%;">@C0000~@C1574</td> <td style="width: 33%;">@C0000~@C1574</td> </tr> <tr> <td>C2000~C7574</td> <td>@C2000~@C7574</td> <td>@C2000~@C7574</td> </tr> <tr> <td>b0000~b1774</td> <td>@b0000~@b1774</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3774</td> <td>@b2000~@b3774</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99774</td> <td>@09000~@99774</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7774</td> <td>@E0000~@E7774</td> <td>@E0000~@E7774</td> </tr> <tr> <td>774#1 00000~03774</td> <td>774#1 @00000~@03774</td> <td>774#1 @00000~@03774</td> </tr> </table>					C0000~C1574	@C0000~@C1574	@C0000~@C1574	C2000~C7574	@C2000~@C7574	@C2000~@C7574	b0000~b1774	@b0000~@b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	@b2000~@b3774	09000~99774	@09000~@99774	@09000~@99774	E0000~E7774	@E0000~@E7774	@E0000~@E7774	774#1 00000~03774	774#1 @00000~@03774	774#1 @00000~@03774
C0000~C1574	@C0000~@C1574	@C0000~@C1574																								
C2000~C7574	@C2000~@C7574	@C2000~@C7574																								
b0000~b1774	@b0000~@b1774	@b0000~@b1774																								
b2000~b3774	@b2000~@b3774	@b2000~@b3774																								
09000~99774	@09000~@99774	@09000~@99774																								
E0000~E7774	@E0000~@E7774	@E0000~@E7774																								
774#1 00000~03774	774#1 @00000~@03774	774#1 @00000~@03774																								
演算条件	入力信号の立上り(OFF→ON)																									
演算前	Sの内容	角度の小数部(BCD2桁)		角度の範囲は 0~9999.99°																						
	S+1, S+2の内容	角度の整数部(BCD4桁)																								
演算中	S~S+2の内容	不変		●演算結果の範囲は -99.9999~99.9999 ●S~S+2の内容がBCDでない時不変																						
	D, D+1の内容	演算結果の小数部(BCD4桁)																								
	D+2の内容	演算結果の整数部(BCD2桁)																								
	D+3の内容	演算結果の符号 [00-正(+)] [80-負(-)] (BCD2桁)																								
演算後	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354																					
	正(+)	0	0	0	1																					
	負(-)	0	1	0	0																					
	+100以上、または-100以下	0	0	1	0																					
	S~S+2の内容がBCDでない時	0	0	1	0																					

(解説)



命令	
STR	04001
F-24	09000 19000

入力条件04001がOFF→ONの変化時に、レジスタ09000~09002のBCD6桁データの正接(TAN)を求め、演算結果をレジスタ19000~19003に格納します。



上記の演算は、 $TAN98.03^\circ \approx -7.0884$ を示しています。

- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- 演算結果が100.0000以上または-100.0000以下になるSの内容の場合、エラーフラグをONにして演算しません。
- 演算結果は、小数第5位を四捨五入します。

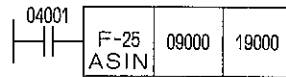
**F-25
ASIN**

三角関数(SIN⁻¹)の演算

(この命令はJW-31CUH/H1ではプログラムできません。)

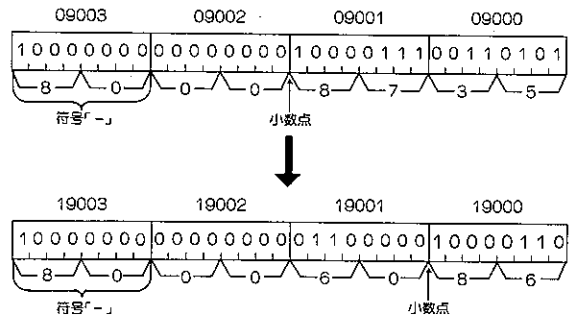
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-25</td><td style="padding: 2px;">ASIN</td><td style="padding: 2px;">S</td><td style="padding: 2px;">D</td></tr></table>					F-25	ASIN	S	D										
F-25	ASIN	S	D																
機能	レジスタS~S+3の内容(BCD8桁)の逆正弦(SIN ⁻¹)を求め、レジスタD~D+3にBCD8桁で格納する。																		
演算内容	SIN ⁻¹ (S~S+3)→D~D+3																		
Sの使用範囲	<table border="0" style="width: 100%; font-size: small;"> <tr><td>コ0000~コ1574</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7574</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1774</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3774</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99774</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7774</td><td>@E0000~@E7774</td></tr> <tr><td>771M1 000000~037774</td><td>771M1 @000000~@037774</td></tr> </table>					コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	771M1 000000~037774	771M1 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																		
コ2000~コ7574	@コ2000~@コ7574																		
b0000~b1774	@b0000~@b1774																		
b2000~b3774	@b2000~@b3774																		
09000~99774	@09000~@99774																		
E0000~E7774	@E0000~@E7774																		
771M1 000000~037774	771M1 @000000~@037774																		
Dの使用範囲	<table border="0" style="width: 100%; font-size: small;"> <tr><td>コ0000~コ1574</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7574</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1774</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3774</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99774</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7774</td><td>@E0000~@E7774</td></tr> <tr><td>771M1 000000~037774</td><td>771M1 @000000~@037774</td></tr> </table>					コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	771M1 000000~037774	771M1 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																		
コ2000~コ7574	@コ2000~@コ7574																		
b0000~b1774	@b0000~@b1774																		
b2000~b3774	@b2000~@b3774																		
09000~99774	@09000~@99774																		
E0000~E7774	@E0000~@E7774																		
771M1 000000~037774	771M1 @000000~@037774																		
演算条件	入力信号の立上り(OFF→ON)																		
演算前	S、S+1の内容	SIN ⁻¹ データの小数部(BCD4桁)	SIN ⁻¹ データの範囲は -1.0000~1.0000																
	S+2の内容	SIN ⁻¹ データの整数部(BCD2桁)																	
	S+3の内容	SIN ⁻¹ データの符号 [00-正(+)] (BCD2桁) [80-負(-)] (BCD2桁)																	
演算中	S~S+3の内容	不変																	
	Dの内容	演算結果の小数部(BCD2桁)	演算結果の範囲は -90.00°~90.00° ● S~S+3の内容がBCDでない時不変																
	D+1、D+2の内容	演算結果の整数部(BCD4桁)																	
	D+3の内容	演算結果の符号 [00-正(+)] (BCD2桁) [80-負(-)] (BCD2桁)																	
演算後	フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354													
		正(+)	0	0	0	1													
		負(-)	0	1	0	0													
	S~S+3の内容がBCDでない時	0	0	1	0														
	S~S+3の内容が+1より大きいまたは-1より小さい時	0	0	1	0														

(解説)



命令	
STR	04001
F-25	09000
	19000

入力条件04001がOFF→ONの変化時に、レジスタ09000~09003のBCD8桁データの逆正弦(SIN⁻¹)を求め、演算結果をレジスタ19000~19003に格納します。



上記の演算は、SIN⁻¹(-0.8735) ≃ -60.87°を示しています。

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- 演算結果は、小数第3位を四捨五入します。

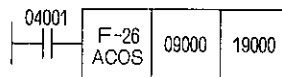
**F-26
ACOS**

三角関数(COS⁻¹)の演算

(この命令はJW-31CUH/H1ではプログラムできません。)

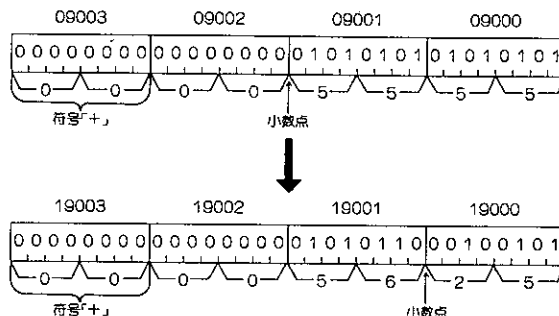
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-26</td><td>S</td><td>D</td></tr><tr><td>ACOS</td><td></td><td></td></tr></table>				F-26	S	D	ACOS		
F-26	S	D								
ACOS										
機能	レジスタS~S+3の内容(BCD8桁)の逆余弦(COS ⁻¹)を求め、レジスタD~D+3にBCD8桁で格納する。									
演算内容	COS ⁻¹ (S~S+3)→D~D+3									
Sの使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 771M1 000000~037774 : 771M1 @000000~@037774									
Dの使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 771M1 000000~037774 : 771M1 @000000~@037774									
演算条件	入力信号の立上り(OFF→ON)									
演算前	S、S+1の内容	COS ⁻¹ データの小数部(BCD4桁)	COS ⁻¹ データの範囲は-1.0000~1.0000							
	S+2の内容	COS ⁻¹ データの整数部(BCD2桁)								
	S+3の内容	COS ⁻¹ データの符号 [00-正(+)] [80-負(-)] (BCD2桁)								
演算	S~S+3の内容	不変								
	Dの内容	演算結果の小数部(BCD2桁)	演算結果の範囲は0~180.00° ●S~S+3の内容がBCDでない時不変							
	D+1、D+2の内容	演算結果の整数部(BCD4桁)								
D+3の内容	演算結果の符号 [00-正(+)] [80-負(-)] (BCD2桁)									
後	フ ラ グ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354				
		正(+)	0	0	0	1				
		負(-)	0	1	0	0				
		S~S+3の内容がBCDでない時	0	0	1	0				
		S~S+3の内容が+1より大きいまたは-1より小さい時	0	0	1	0				

(解説)



命 令	
STR	04001
F-26	09000
	19000

入力条件04001がOFF→ONの変化時に、レジスタ09000~09003のBCD8桁データの逆余弦(COS⁻¹)を求め、演算結果をレジスタ19000~19003に格納します。



上記の演算は、COS⁻¹(0.5555)≒56.25°を示しています。

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- 演算結果は、小数第3位を四捨五入します。

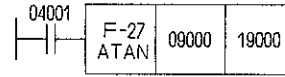
**F-27
ATAN**

三角関数(TAN⁻¹)の演算

(この命令はJW-31CUH/H1ではプログラムできません。)

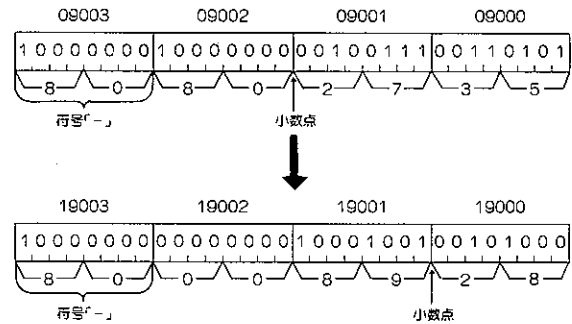
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-27</td><td>S</td><td>D</td></tr></table>				F-27	S	D											
F-27	S	D																
機能	レジスタS~S+3の内容(BCD8桁)の逆正接(TAN ⁻¹)を求め、レジスタD~D+3にBCD8桁で格納する。																	
演算内容	TAN ⁻¹ (S~S+3)→D~D+3																	
Sの使用範囲	<table border="0"> <tr><td>コ0000~コ1574</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7574</td><td>@コ2000~@コ7574</td></tr> <tr><td>ボ0000~ボ1774</td><td>@ボ0000~@ボ1774</td></tr> <tr><td>ボ2000~ボ3774</td><td>@ボ2000~@ボ3774</td></tr> <tr><td>ボ9000~ボ9774</td><td>@ボ9000~@ボ9774</td></tr> <tr><td>エ0000~エ7774</td><td>@エ0000~@エ7774</td></tr> <tr><td>77441 000000~037774</td><td>77441 @000000~@037774</td></tr> </table>				コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	ボ0000~ボ1774	@ボ0000~@ボ1774	ボ2000~ボ3774	@ボ2000~@ボ3774	ボ9000~ボ9774	@ボ9000~@ボ9774	エ0000~エ7774	@エ0000~@エ7774	77441 000000~037774	77441 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																	
コ2000~コ7574	@コ2000~@コ7574																	
ボ0000~ボ1774	@ボ0000~@ボ1774																	
ボ2000~ボ3774	@ボ2000~@ボ3774																	
ボ9000~ボ9774	@ボ9000~@ボ9774																	
エ0000~エ7774	@エ0000~@エ7774																	
77441 000000~037774	77441 @000000~@037774																	
Dの使用範囲	<table border="0"> <tr><td>コ0000~コ1574</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7574</td><td>@コ2000~@コ7574</td></tr> <tr><td>ボ0000~ボ1774</td><td>@ボ0000~@ボ1774</td></tr> <tr><td>ボ2000~ボ3774</td><td>@ボ2000~@ボ3774</td></tr> <tr><td>ボ9000~ボ9774</td><td>@ボ9000~@ボ9774</td></tr> <tr><td>エ0000~エ7774</td><td>@エ0000~@エ7774</td></tr> <tr><td>77441 000000~037774</td><td>77441 @000000~@037774</td></tr> </table>				コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	ボ0000~ボ1774	@ボ0000~@ボ1774	ボ2000~ボ3774	@ボ2000~@ボ3774	ボ9000~ボ9774	@ボ9000~@ボ9774	エ0000~エ7774	@エ0000~@エ7774	77441 000000~037774	77441 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																	
コ2000~コ7574	@コ2000~@コ7574																	
ボ0000~ボ1774	@ボ0000~@ボ1774																	
ボ2000~ボ3774	@ボ2000~@ボ3774																	
ボ9000~ボ9774	@ボ9000~@ボ9774																	
エ0000~エ7774	@エ0000~@エ7774																	
77441 000000~037774	77441 @000000~@037774																	
演算条件	入力信号の立上り(OFF→ON)																	
演算前	S、S+1の内容	TAN ⁻¹ データの小数部(BCD4桁)	TAN ⁻¹ データの範囲は -98.9999~98.9999															
	S+2の内容	TAN ⁻¹ データの整数部(BCD2桁)																
	S+3の内容	TAN ⁻¹ データの符号 [00-正(+)] [80-負(-)] (BCD2桁)																
演算	S~S+3の内容	不変																
	Dの内容	演算結果の小数部(BCD2桁)	演算結果の範囲は -89.42~89.42 ● S~S+3の内容がBCDでない時不変															
	D+1、D+2の内容	演算結果の整数部(BCD4桁)																
D+3の内容	演算結果の符号 [00-正(+)] [80-負(-)] (BCD2桁)																	
演算後	フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354												
		正(+)	0	0	0	1												
		負(-)	0	1	0	0												
		S~S+3の内容がBCDでない時	0	0	1	0												
		S~S+3の内容が+99.0000以上、または、-99.0000以下の時	0	0	1	0												

(解説)



命令	
STR	04001
F-27	09000
	19000

入力条件04001がOFF→ONの変化時に、レジスタ09000~09003のBCD8桁データの逆正接(TAN⁻¹)を求め、演算結果をレジスタ19000~19003に格納します。



上記の演算は、TAN⁻¹(-80.2735) ≃ -89.29°を示しています。

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- 演算結果は、小数第3位を四捨五入します。

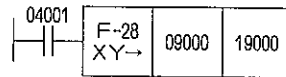
F-28
XY→

直交座標系(X、Y)データの極座標系(r、θ)変換

(この命令はJW-31CUH/H1ではプログラムできません。)

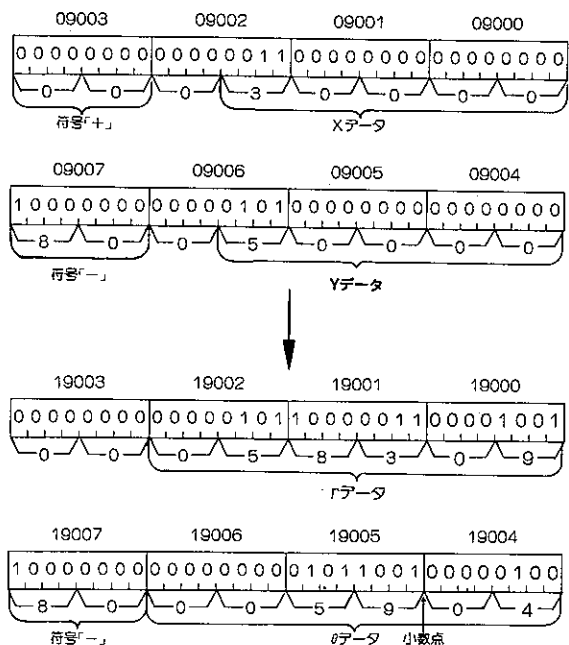
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-28</td><td style="padding: 2px;">S</td><td style="padding: 2px;">D</td></tr><tr><td style="padding: 2px;">XY→</td><td></td><td></td></tr></table>				F-28	S	D	XY→																																																																																																																																																																								
F-28	S	D																																																																																																																																																																														
XY→																																																																																																																																																																																
機能	レジスタS~S+3とレジスタS+4~S+7の直交座標(X、Y)を極座標(r、θ)に変換し、レジスタD~D+3とレジスタD+4~D+7に格納する。																																																																																																																																																																															
演算内容	X(S~S+3)、Y(S+4~S+7) →r(D~D+3)、θ(D+4~D+7)																																																																																																																																																																															
Sの使用範囲	<table border="0" style="font-size: small;"> <tr><td>U</td><td>0</td><td>0</td><td>0</td><td>0</td><td>~</td><td>U</td><td>1</td><td>5</td><td>7</td><td>0</td><td>;</td><td>@</td><td>0</td><td>0</td><td>0</td><td>0</td><td>~</td><td>@</td><td>1</td><td>5</td><td>7</td><td>4</td></tr> <tr><td>U</td><td>2</td><td>0</td><td>0</td><td>0</td><td>~</td><td>U</td><td>7</td><td>7</td><td>7</td><td>0</td><td>;</td><td>@</td><td>2</td><td>0</td><td>0</td><td>0</td><td>~</td><td>@</td><td>7</td><td>7</td><td>7</td><td>4</td></tr> <tr><td>b</td><td>0</td><td>0</td><td>0</td><td>0</td><td>~</td><td>b</td><td>1</td><td>3</td><td>7</td><td>0</td><td>;</td><td>@</td><td>b</td><td>0</td><td>0</td><td>0</td><td>~</td><td>@</td><td>b</td><td>3</td><td>7</td><td>4</td></tr> <tr><td>b</td><td>2</td><td>0</td><td>0</td><td>0</td><td>~</td><td>b</td><td>3</td><td>7</td><td>7</td><td>0</td><td>;</td><td>@</td><td>b</td><td>2</td><td>0</td><td>0</td><td>~</td><td>@</td><td>b</td><td>3</td><td>7</td><td>4</td></tr> <tr><td>E</td><td>0</td><td>0</td><td>0</td><td>0</td><td>~</td><td>E</td><td>9</td><td>9</td><td>7</td><td>0</td><td>;</td><td>@</td><td>E</td><td>0</td><td>0</td><td>0</td><td>~</td><td>@</td><td>E</td><td>9</td><td>9</td><td>7</td><td>4</td></tr> <tr><td>E</td><td>0</td><td>0</td><td>0</td><td>0</td><td>~</td><td>E</td><td>7</td><td>7</td><td>7</td><td>0</td><td>;</td><td>@</td><td>E</td><td>0</td><td>0</td><td>0</td><td>~</td><td>@</td><td>E</td><td>7</td><td>7</td><td>7</td><td>4</td></tr> <tr><td>77</td><td>#</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>~</td><td>77</td><td>#</td><td>1</td><td>@</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>~</td><td>77</td><td>#</td><td>1</td><td>@</td><td>0</td><td>3</td><td>7</td><td>7</td><td>4</td></tr> </table>				U	0	0	0	0	~	U	1	5	7	0	;	@	0	0	0	0	~	@	1	5	7	4	U	2	0	0	0	~	U	7	7	7	0	;	@	2	0	0	0	~	@	7	7	7	4	b	0	0	0	0	~	b	1	3	7	0	;	@	b	0	0	0	~	@	b	3	7	4	b	2	0	0	0	~	b	3	7	7	0	;	@	b	2	0	0	~	@	b	3	7	4	E	0	0	0	0	~	E	9	9	7	0	;	@	E	0	0	0	~	@	E	9	9	7	4	E	0	0	0	0	~	E	7	7	7	0	;	@	E	0	0	0	~	@	E	7	7	7	4	77	#	1	0	0	0	0	0	0	0	~	77	#	1	@	0	0	0	0	0	0	0	~	77	#	1	@	0	3	7	7	4
U	0	0	0	0	~	U	1	5	7	0	;	@	0	0	0	0	~	@	1	5	7	4																																																																																																																																																										
U	2	0	0	0	~	U	7	7	7	0	;	@	2	0	0	0	~	@	7	7	7	4																																																																																																																																																										
b	0	0	0	0	~	b	1	3	7	0	;	@	b	0	0	0	~	@	b	3	7	4																																																																																																																																																										
b	2	0	0	0	~	b	3	7	7	0	;	@	b	2	0	0	~	@	b	3	7	4																																																																																																																																																										
E	0	0	0	0	~	E	9	9	7	0	;	@	E	0	0	0	~	@	E	9	9	7	4																																																																																																																																																									
E	0	0	0	0	~	E	7	7	7	0	;	@	E	0	0	0	~	@	E	7	7	7	4																																																																																																																																																									
77	#	1	0	0	0	0	0	0	0	~	77	#	1	@	0	0	0	0	0	0	0	~	77	#	1	@	0	3	7	7	4																																																																																																																																																	
Dの使用範囲	<table border="0" style="font-size: small;"> <tr><td>U</td><td>0</td><td>0</td><td>0</td><td>0</td><td>~</td><td>U</td><td>1</td><td>5</td><td>7</td><td>0</td><td>;</td><td>@</td><td>0</td><td>0</td><td>0</td><td>0</td><td>~</td><td>@</td><td>1</td><td>5</td><td>7</td><td>4</td></tr> <tr><td>U</td><td>2</td><td>0</td><td>0</td><td>0</td><td>~</td><td>U</td><td>7</td><td>7</td><td>7</td><td>0</td><td>;</td><td>@</td><td>2</td><td>0</td><td>0</td><td>0</td><td>~</td><td>@</td><td>7</td><td>7</td><td>7</td><td>4</td></tr> <tr><td>b</td><td>0</td><td>0</td><td>0</td><td>0</td><td>~</td><td>b</td><td>1</td><td>3</td><td>7</td><td>0</td><td>;</td><td>@</td><td>b</td><td>0</td><td>0</td><td>0</td><td>~</td><td>@</td><td>b</td><td>3</td><td>7</td><td>4</td></tr> <tr><td>b</td><td>2</td><td>0</td><td>0</td><td>0</td><td>~</td><td>b</td><td>3</td><td>7</td><td>7</td><td>0</td><td>;</td><td>@</td><td>b</td><td>2</td><td>0</td><td>0</td><td>~</td><td>@</td><td>b</td><td>3</td><td>7</td><td>4</td></tr> <tr><td>E</td><td>0</td><td>0</td><td>0</td><td>0</td><td>~</td><td>E</td><td>9</td><td>9</td><td>7</td><td>0</td><td>;</td><td>@</td><td>E</td><td>0</td><td>0</td><td>0</td><td>~</td><td>@</td><td>E</td><td>9</td><td>9</td><td>7</td><td>4</td></tr> <tr><td>E</td><td>0</td><td>0</td><td>0</td><td>0</td><td>~</td><td>E</td><td>7</td><td>7</td><td>7</td><td>0</td><td>;</td><td>@</td><td>E</td><td>0</td><td>0</td><td>0</td><td>~</td><td>@</td><td>E</td><td>7</td><td>7</td><td>7</td><td>4</td></tr> <tr><td>77</td><td>#</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>~</td><td>77</td><td>#</td><td>1</td><td>@</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>~</td><td>77</td><td>#</td><td>1</td><td>@</td><td>0</td><td>3</td><td>7</td><td>7</td><td>4</td></tr> </table>				U	0	0	0	0	~	U	1	5	7	0	;	@	0	0	0	0	~	@	1	5	7	4	U	2	0	0	0	~	U	7	7	7	0	;	@	2	0	0	0	~	@	7	7	7	4	b	0	0	0	0	~	b	1	3	7	0	;	@	b	0	0	0	~	@	b	3	7	4	b	2	0	0	0	~	b	3	7	7	0	;	@	b	2	0	0	~	@	b	3	7	4	E	0	0	0	0	~	E	9	9	7	0	;	@	E	0	0	0	~	@	E	9	9	7	4	E	0	0	0	0	~	E	7	7	7	0	;	@	E	0	0	0	~	@	E	7	7	7	4	77	#	1	0	0	0	0	0	0	0	~	77	#	1	@	0	0	0	0	0	0	0	~	77	#	1	@	0	3	7	7	4
U	0	0	0	0	~	U	1	5	7	0	;	@	0	0	0	0	~	@	1	5	7	4																																																																																																																																																										
U	2	0	0	0	~	U	7	7	7	0	;	@	2	0	0	0	~	@	7	7	7	4																																																																																																																																																										
b	0	0	0	0	~	b	1	3	7	0	;	@	b	0	0	0	~	@	b	3	7	4																																																																																																																																																										
b	2	0	0	0	~	b	3	7	7	0	;	@	b	2	0	0	~	@	b	3	7	4																																																																																																																																																										
E	0	0	0	0	~	E	9	9	7	0	;	@	E	0	0	0	~	@	E	9	9	7	4																																																																																																																																																									
E	0	0	0	0	~	E	7	7	7	0	;	@	E	0	0	0	~	@	E	7	7	7	4																																																																																																																																																									
77	#	1	0	0	0	0	0	0	0	~	77	#	1	@	0	0	0	0	0	0	0	~	77	#	1	@	0	3	7	7	4																																																																																																																																																	
演算条件	入力信号の立上り(OFF→ON)																																																																																																																																																																															
演算前	S~S+2の内容	Xデータの整数部 (BCD5桁)		Xデータの範囲は -99999~99999																																																																																																																																																																												
	S+3の内容	Xデータの符号 (BCD2桁) [00-正(+)] [80-負(-)]																																																																																																																																																																														
	S+4~S+6の内容	Yデータの整数部 (BCD5桁)		Yデータの範囲は -99999~99999																																																																																																																																																																												
	S+7の内容	Yデータの符号 (BCD2桁) [00-正(+)] [80-負(-)]																																																																																																																																																																														
演算後	S~S+7の内容	不変																																																																																																																																																																														
	D~D+3の内容	rデータの整数部 (BCD8桁)		●θデータの範囲は -179.99°~ 180.00° ●X、YデータがBCDコードでない時不変																																																																																																																																																																												
	D+4の内容	θデータの小数部 (BCD2桁)																																																																																																																																																																														
	D+5、D+6の内容	θデータの整数部 (BCD3桁)		●X、YデータがBCDコードでない時不変																																																																																																																																																																												
D+7の内容	θデータの符号(BCD2桁) [00-正(+)] [80-負(-)]																																																																																																																																																																															
フラグ	レジスタSの内容	ゼロ	キャリー	エラー	ノキャリー																																																																																																																																																																											
		07357	07356	07355	07354																																																																																																																																																																											
フラグ	X、YデータがBCDの時	0	0	0	0																																																																																																																																																																											
	X、YデータがBCDでない時	0	0	1	0																																																																																																																																																																											

〔解説〕



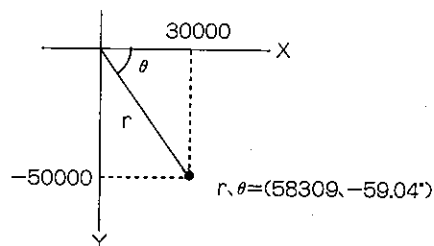
命令	
STR	04001
F-28	09000
	19000

入力条件04001がOFF→ONの変化時に、レジスタ09000~09007の直交座標(X、Y)データを極座標(r、θ)データに変換し、レジスタ19000~19007に格納します。



上記は次の演算を示します。

$X(30000)、Y(-50000) \rightarrow r(58309)、\theta(-59.04^\circ)$

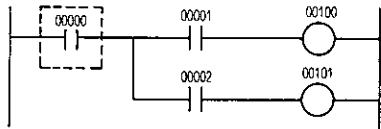


- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- S+3、S+7のBCD上位桁は無視します。

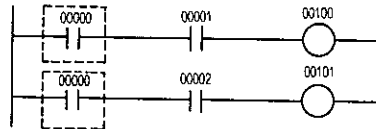
F-30 MCS マスターコントロールセット
(Master Control Set)

F-31 MCR マスターコントロールリセット
(Master Control Reset)

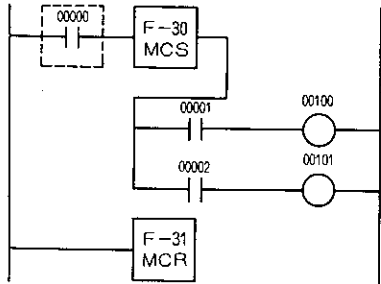
MCS、MCRは、共通演算条件以後の回路が複数の出力に分岐している場合に使用します。



(1) リレー盤の場合



(2) MCS、MCRを使わない場合



(3) MCS、MCRを使用した場合

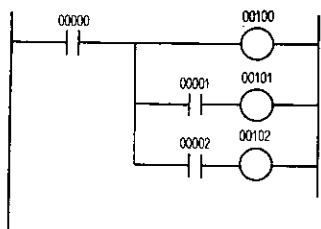
MCS→	STR	00000
	F-30	
	STR	00001
	OUT	00100
	STR	00002
	OUT	00101
MCR→	F-31	

F-30(MCS)を使用するとそれまでのACC (アキュムレータ)の内容をCPU内部のレジスタに記憶し、F-31(MCR)までの各命令の演算はCPU内部レジスタの内容とANDしたものとなります。F-31(MCR)は、このANDする範囲の終了を意味し

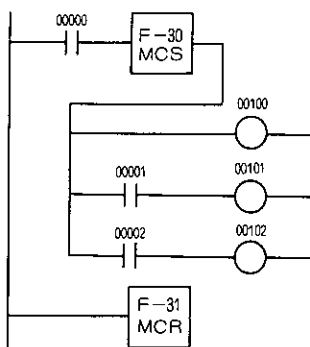
ます。
内の共通演算条件が複雑な場合や、共通演算条件に続く演算の分岐が多い場合、プログラムを簡略できます。

- F-30(MCS)で派生した母線に、直接OUT、TMR、CNTの各命令及び応用命令は接続できません。

(1) リレー盤の場合

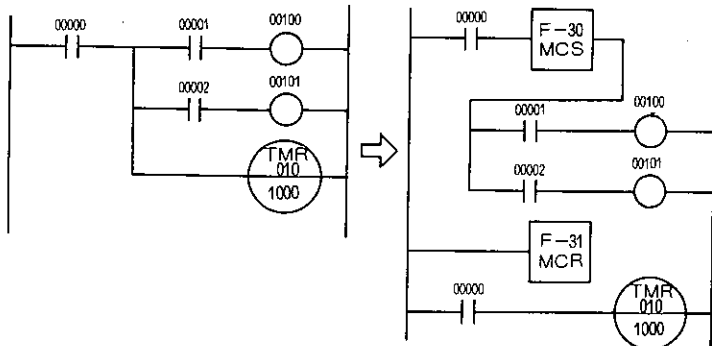


(2) MCS、MCRで禁止のプログラム

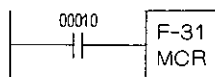


命 令	
STR	00000
F-30	
OUT	00100
STR	00001
OUT	00101
STR	00002
OUT	00102
F-31	

次のようにプログラムしてください。

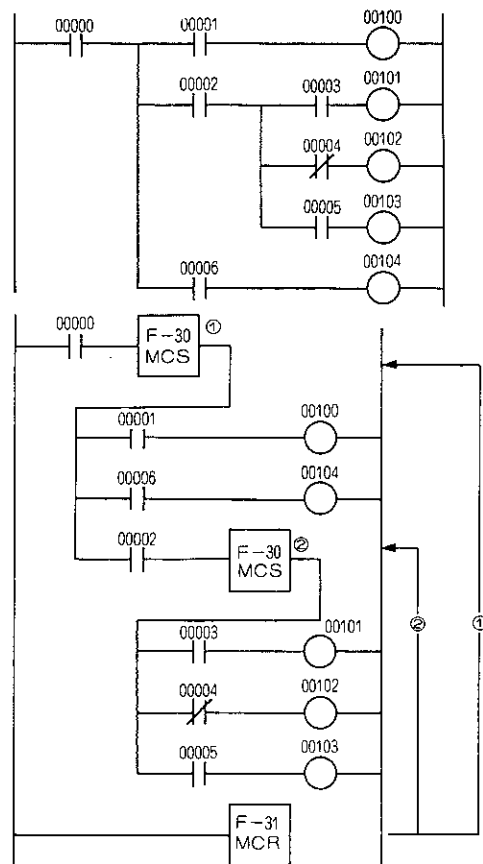


- F-31 (MCR) は無条件命令です。



のようなプログラムはできません。

MCS、MCRの間にさらにMCSを使用できます。



左図のリレー盤のラダー図はMCS、MCRを用いて次のようにプログラムできます。ただし本例のようにプログラム順を入れ換える必要がある場合があります。(※印部)

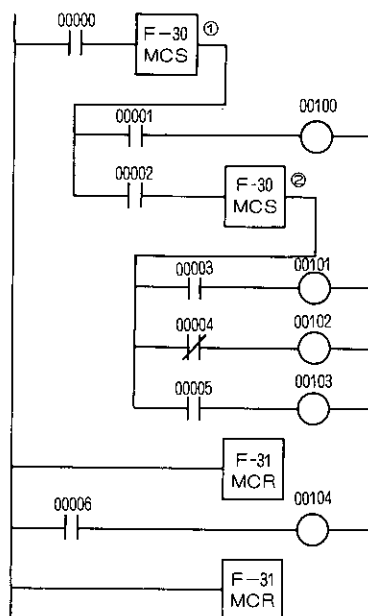
命 令	
STR	00000
F-30	
STR	00001
OUT	00100
STR	00006
OUT	00104
STR	00002
F-30	
STR	00003
OUT	00101
STR NOT	00004
OUT	00102
STR	00005
OUT	00103
F-31	

- F-31(MCR)はそれ以前のF-30(MCS)…左図の場合①、②…の終了を意味します。

- 次のようにプログラムすると、初期の回路にはなりません。

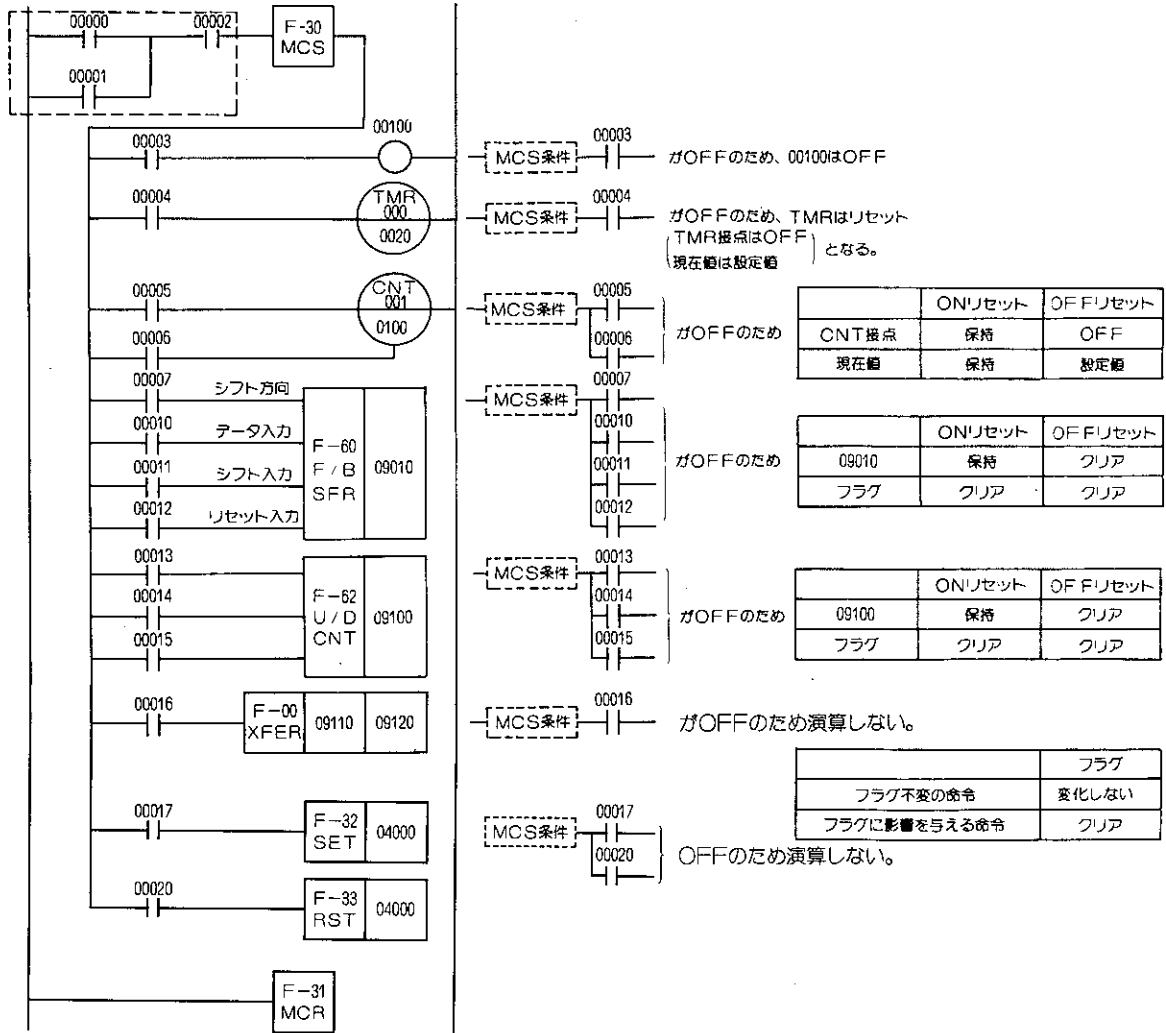
命 令	
MCS →	STR 00000
	F-30
	STR 00001
	OUT 00100
	STR 00002
MCS →	F-30
	STR 00003
	OUT 00101
	STR NOT 00004
	OUT 00102
	STR 00005
	OUT 00103
MCR →	F-31
	STR 00006
	OUT 00104
MCR →	F-31

このMCRは無意味なものです。
このMCRで①、②のMCSは終了します。



- MCS、MCR*の間にMCSを何度でも使用できますが、すべてのMCSの範囲は、*のMCRで終了します。上記の様なネスティングを行いたい場合はF-231(MCRN)「マスターコントロールネスティングリセット」命令をF-31(MCR)のかわりに使用します。

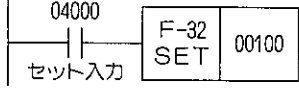
MCSの条件(点線内)がOFFのとき、MCSとMC Rの間にある命令は次のように処理します。



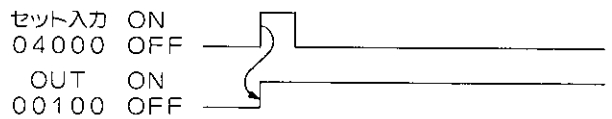
- CNT、F-60、F-62の各命令はシステムメモリ #202でリセット条件をONリセット、OFFリセットのいずれかに設定できます。OFFリセットの場合、MCSによりリセットします。

F-32
SET

セットコイル

シンボル	— [F-32 SET] [OUT]		<p>(解説)</p>  <table border="1" data-bbox="1181 264 1449 409"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>04000</td> </tr> <tr> <td>F-32</td> <td>00100</td> </tr> </tbody> </table>	命 令		STR	04000	F-32	00100
命 令									
STR	04000								
F-32	00100								
機能	セット入力ONになった時に、F-32で指定したOUTをONにする。								
演算内容	F-32で指定したOUTをON								
OUTの使用範囲	00000～15777 20000～75777								
演算条件	セット入力ONの時 (OFF→ONの変化時に限定されない)								
演算後	OUTの内容	ON							
	フラグ	不変							

セット入力04000がONの時に、OUT00100がONになります。
ONになったOUT00100は、セット入力04000がOFFになってもONのまま保持します。
セット入力04000がOFFの時は、OUT00100の状態は変化しません。



- F-32(SET)命令がF-30(MCS)命令の中にあるとき、ONにしたいOUTは、F-30命令がOFFになってもONのまま保持します。
- F-32(SET)命令を使用すると1個のOUTを複数の回路上で制御できます。
- F-32(SET)命令で指定したOUTがキープ指定領域のときは、復電後も停電前の状態を保持します。また、指定したOUTがキープ指定領域以外のときは、復電時にリセットします。
- F-32(SET)命令で指定されたOUTが本PC停止時に出力保持を行う領域内のときは、本体停止時に停止前の状態を保持します。また、指定されたOUTが本体停止時に出力保持を行う領域以外のときは、本体停止時にリセットされます。5・12ページ「システムメモリ #232, #233, #252, #253」を参照ください。
- F-32(SET)命令は、次項のF-33命令とペアで使用してください。
- MCS(F-30)とMCR(F-31)の間にあるF-32(SET)命令とF-33(RST)命令はMCS(F-30)の演算条件がOFFのとき動作しません。

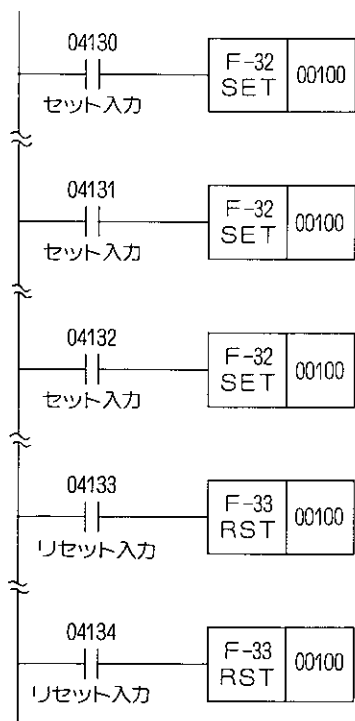
F-33 RST

リセットコイル

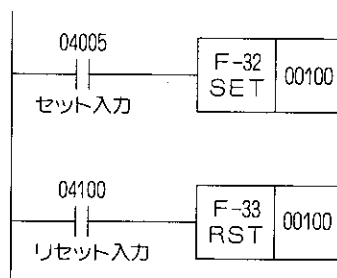
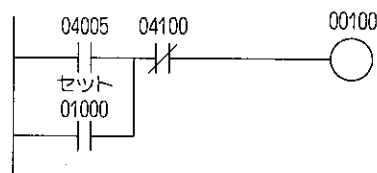
シンボル			(解説)		<table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>04002</td> </tr> <tr> <td>F-33</td> <td>00110</td> </tr> </tbody> </table>	命 令		STR	04002	F-33	00110
命 令											
STR	04002										
F-33	00110										
機能	リセット入力がONになった時に、F-33で指定したOUTをOFFにする。										
演算内容	F-33で指定したOUTをOFF										
OUTの使用範囲	00000~15777 20000~75777										
演算条件	リセット入力がONの時(OFF→ONの変化時に限定されない)										
演算後	OUTの内容	OFF									
	フラグ	不変									

● F-32(SET)命令とF-33(RST)命令を使用すると1個のOUTを複数の条件により制御できます。

● F-32(SET)命令とF-33(RST)命令はペアで使用すると便利です。F-32命令とF-33命令をペアで使用すると自己保持回路等の簡略化ができます。



リセット優先自己保持回路

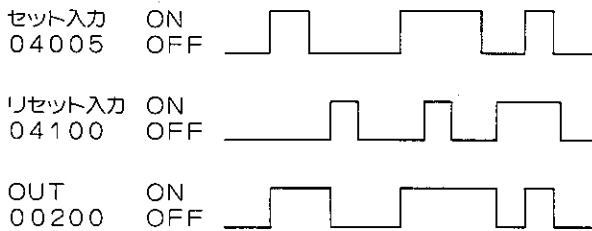
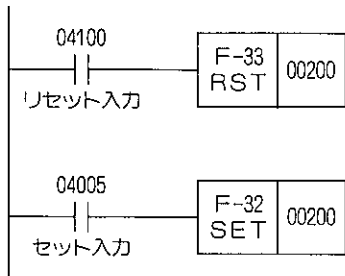
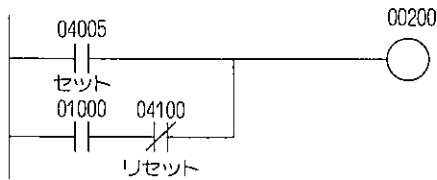


セット入力 ON
04005 OFF

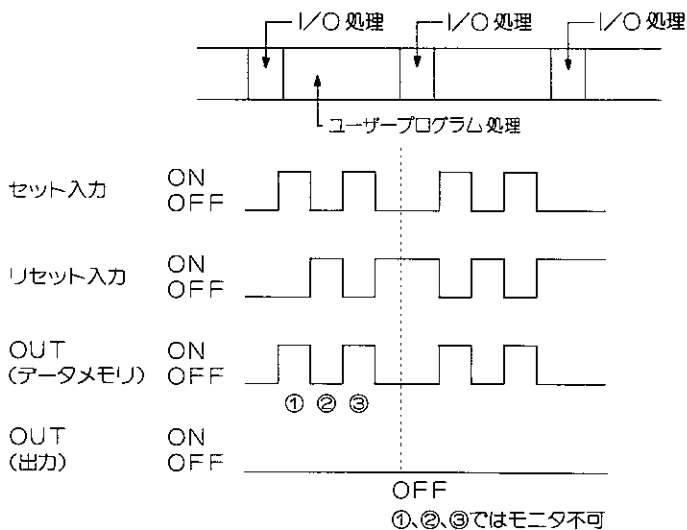
リセット入力 ON
04100 OFF

OUT ON
00100 OFF

セット優先自己保持回路



- セット入力とリセット入力が1スキャン内で複数回ON/OFFする場合、OUTとして使用しているデータメモリは1スキャン周期内でON/OFFを繰り返しません。ただし、出力ユニットの出力端子はI/O処理直前のOUTの結果(ON又はOFF)を出力します。



ユーザープログラム処理中のデータメモリのモニタは、I/O処理直前の結果で出力します。

- F-33(RST)命令で指定したOUTがキープ指定領域内のときは、復電後も停電前の状態を保持します。また、指定したOUTがキープ指定領域以外のときは、復電時にリセットします。
- F-33(RST)命令で指定したOUTがJW30H停止時に出力保持を行う領域内のときは、停止前の状態を保持します。また、指定したOUTがJW30H停止時に出力保持を行う領域以外のときは、停止時にリセットします。5・12ページ「システムメモリ #232, #233, #252, #253」を参照ください。

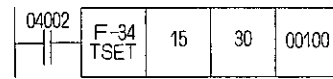
F-34 TSET

時計の現在値との比較(指定リレーのセット)

(この命令はJW-31CUH/H1ではプログラムできません。)

シンボル	— F-34 TSET n ₁ n ₂ BIT		
機能	定数n ₁ (時)、n ₂ (分)と時計の現在値とを比較し、一致すると指定したBIT(リレー)をセット(ON)にする。		
演算内容	n ₁ 、n ₂ (=)時計の現在値を比較 比較結果が一致でリレーをON		
n ₁ の使用範囲	00~23(10進)		
n ₂ の使用範囲	00~59(10進)		
BITの使用範囲	00000~15777, 20000~75777		
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)		
演算後	n ₁ の内容	不変	
	n ₂ の内容	不変	
	フラグ	不変	

(解説)



命 令	
STR	04002
F-34	15 30 00100

入力条件04002がONの時、指定した時計時刻と、時計の現在値が一致したときリレー00100がONになります。

ONになったリレー00100は、入力条件04002がOFFになってもONのまま保持します。

時計の現在値と指定時刻が一致しないときは、リレー00100の状態は変化しません。

指定時刻 ≥ 時計の現在値 → NOP

指定時刻 = 時計の現在値 → 指定リレーON

- 本命令はコントロールがJW-31CUH/H1のとき、使用できません。JW-31CUH/H1には時計機能がありません。
- F-34(TSET)命令がF-30(MCS)命令の中にあるとき、ONにしたリレーは、F-30命令がOFFになってもONのまま保持します。
- F-34(TSET)命令を使用すると1個のリレーを複数の回路上で制御できます。
- F-34(TSET)命令で指定したリレーがキープ指定領域内のときは、復電後も停電前の状態を保持します。また、指定したリレーがキープ指定領域以外のときは、復電時リセットします。

- F-34(TSET)命令で指定したリレーがJW30H停止時に出力保持を行う領域内のときは、停止前の状態を保持します。また、指定したリレーがJW30H停止時に出力保持を行う領域以外のときは、停止時にリセットします。
- F-34(TSET)命令は、次頁のF-35(TRST)命令とペアで使用してください。
- F-30(MCS)命令とF-31(MCR)命令の間にあるF-34(TSET)命令とF-35(TRST)命令は、F-30(MCS)命令の演算条件がOFFのとき動作しません。

**F-35
TRST**

時計の現在値との比較(指定リレーのリセット)

(この命令はJW-31CUH/H1ではプログラムできません。)

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-35 TRST</td> <td style="padding: 2px;">n₁</td> <td style="padding: 2px;">n₂</td> <td style="padding: 2px;">BIT</td> </tr> </table>			F-35 TRST	n ₁	n ₂	BIT	<p>(解 説)</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="margin-right: 10px;"> <table border="1" style="font-size: small;"> <tr><td style="padding: 2px;">04003</td></tr> </table> </div> <table border="1" style="font-size: small;"> <tr> <td style="padding: 2px;">F-35 TRST</td> <td style="padding: 2px;">09</td> <td style="padding: 2px;">15</td> <td style="padding: 2px;">00110</td> </tr> </table> </div> <table border="1" style="font-size: small; margin-left: 20px;"> <thead> <tr><th colspan="2">命 令</th></tr> </thead> <tbody> <tr><td style="padding: 2px;">STR</td><td style="padding: 2px;">04003</td></tr> <tr><td style="padding: 2px;">F-35</td><td style="padding: 2px;">09 15 00110</td></tr> </tbody> </table> <p>入力条件04003がONの時、指定した時計時刻と、時計の現在値が一致したときリレー00110がOFFになります。 OFFになったリレーは、入力条件04003がOFFになってもOFFのまま保持します。 時計の現在値と指定時刻が一致しないときは、リレー00110の状態は変化しません。</p> <p>指定時刻 ≥ 時計の現在値 → NOP 指定時刻 = 時計の現在値 → 指定リレーOFF</p>	04003	F-35 TRST	09	15	00110	命 令		STR	04003	F-35	09 15 00110
F-35 TRST	n ₁	n ₂	BIT																
04003																			
F-35 TRST	09	15	00110																
命 令																			
STR	04003																		
F-35	09 15 00110																		
機能	定数n ₁ (時)、n ₂ (分)と時計の現在値とを比較し、一致すると指定したBIT(リレー)をセット(OFF)にする。																		
演算内容	n ₁ 、n ₂ <=> 時計の現在値を比較 比較結果が一致でリレーをリセット																		
n ₁ の使用範囲	00~23(10進)																		
n ₂ の使用範囲	00~59(10進)																		
BITの使用範囲	00000~15777, 20000~75777																		
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)																		
演算後	n ₁ の内容	不変																	
	n ₂ の内容	不変																	
	フラグ	不変																	

- 本命令はコントロールユニットがJW-31CUH/H1のとき、使用できません。JW-31CUH/H1には時計機能がありません。
- F-35(TRST)命令で指定したリレーがキープ指定領域内のときは、復電後も停電前の状態を保持します。また、指定したリレーがキープ指定領域以外のときは、復電時リセットします。
- F-35(TSET)命令で指定したリレーがJW30H停止時に出力保持を行う領域内のときは、停止前の状態を保持します。また、指定したリレーがJW30H停止時に出力保持を行う領域以外のときは、停止時にリセットします。

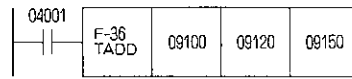
F-36 TADD

時計の加算

(この命令はJW-31CUH/H1ではプログラムできません。)

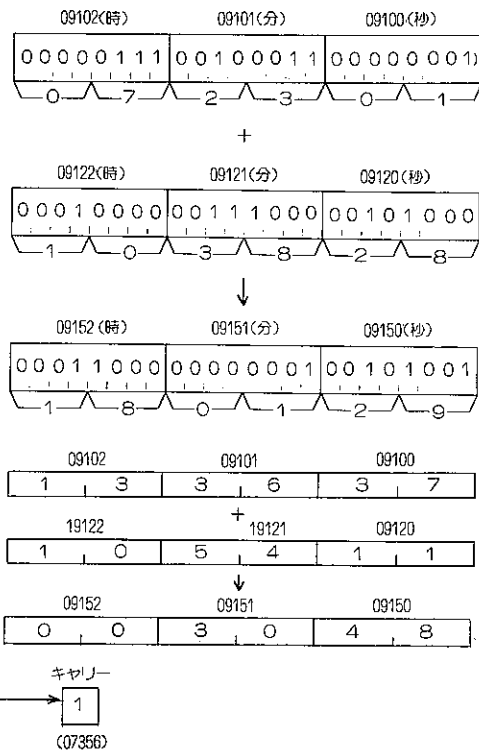
シンボル	— F-36 TADD S ₁ S ₂ D					
機能	レジスタS ₁ ~S ₁ +2の内容とレジスタS ₂ ~S ₂ +2の内容を時計の時、分、秒として加算し、レジスタD~D+2に格納する。					
演算内容	(S ₁ ~S ₁ +2)+(S ₂ ~S ₂ +2) → D~D+2					
S ₁ の使用範囲	コ0000~コ1575 @コ0000~@コ1574 コ2000~コ7575 @コ2000~@コ7574 b0000~b1775 @b0000~@b1774 b2000~b3775 @b2000~@b3774 09000~99775 @09000~@99774 E0000~E7775 @E0000~@E7774 774M1 000000~037775 774M1 @000000~@037774					
S ₂ の使用範囲	コ0000~コ1575 @コ0000~@コ1574 コ2000~コ7575 @コ2000~@コ7574 b0000~b1775 @b0000~@b1774 b2000~b3775 @b2000~@b3774 09000~99775 @09000~@99774 E0000~E7775 @E0000~@E7774 774M1 000000~037775 774M1 @000000~@037774					
Dの使用範囲	コ0000~コ1575 @コ0000~@コ1574 コ2000~コ7575 @コ2000~@コ7574 b0000~b1775 @b0000~@b1774 b2000~b3775 @b2000~@b3774 09000~99775 @09000~@99774 E0000~E7775 @E0000~@E7774 774M1 000000~037775 774M1 @000000~@037774					
演算条件	入力信号の立上がり(OFF→ON)					
演算後	S ₁ ~S ₁ +2の内容	不変				
	S ₂ ~S ₂ +2の内容	不変				
	D~D+2の内容	演算結果				
	フラグ	演算結果	ゼロ	キャリー	エラー	ノキャリー
		0	07357	07356	07355	07354
000001~235959		0	0	0	1	
000000(明日)		1	1	0	0	
000000以上	0	1	0	0		
時刻以外	0	0	1	0		

(解説)



命令	
STR	04001
F-36	09100
	09120
	09150

入力条件04001がOFF→ONの変化時に、レジスタ09100、09101、09102の内容(BCD6桁)とレジスタ09120、09121、09122の内容(BCD6桁)を加算してレジスタ09150、09151、09152に格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 24時間を越えて明日の時間になると、キャリーフラグ07356がONになります。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

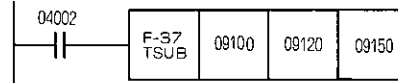
F-37 TSUB

時計の減算

(この命令はJW-31CUH/H1ではプログラムできません。)

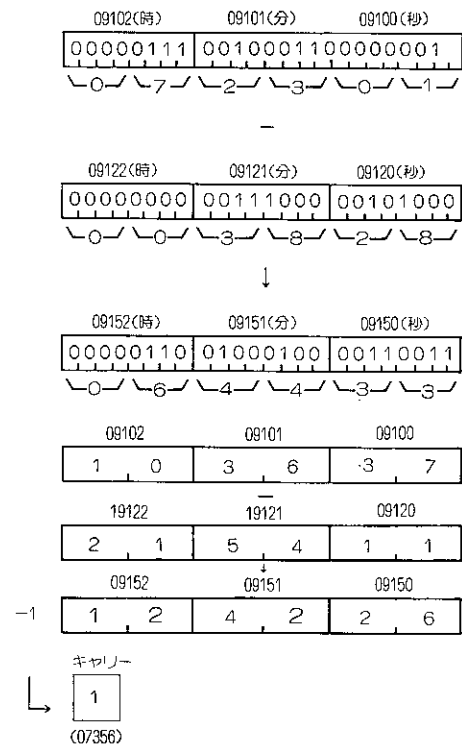
シンボル	— F-37 TSUB S ₁ S ₂ D																	
機能	レジスタS ₁ ~S ₁ +2の内容からレジスタS ₂ ~S ₂ +2の内容を時計の時、分、秒として減算し、レジスタD~D+2に格納する。																	
演算内容	$(S_1 \sim S_1 + 2) - (S_2 \sim S_2 + 2) \rightarrow D \sim D + 2$																	
S ₁ の使用範囲	<table border="0"> <tr><td>コ0000~コ1575</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7575</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1775</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3775</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99775</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7775</td><td>@E0000~@E7774</td></tr> <tr><td>774M1 000000~037775</td><td>774M1 @000000~@037774</td></tr> </table>				コ0000~コ1575	@コ0000~@コ1574	コ2000~コ7575	@コ2000~@コ7574	b0000~b1775	@b0000~@b1774	b2000~b3775	@b2000~@b3774	09000~99775	@09000~@99774	E0000~E7775	@E0000~@E7774	774M1 000000~037775	774M1 @000000~@037774
コ0000~コ1575	@コ0000~@コ1574																	
コ2000~コ7575	@コ2000~@コ7574																	
b0000~b1775	@b0000~@b1774																	
b2000~b3775	@b2000~@b3774																	
09000~99775	@09000~@99774																	
E0000~E7775	@E0000~@E7774																	
774M1 000000~037775	774M1 @000000~@037774																	
S ₂ の使用範囲	<table border="0"> <tr><td>コ0000~コ1575</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7575</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1775</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3775</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99775</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7775</td><td>@E0000~@E7774</td></tr> <tr><td>774M1 000000~037775</td><td>774M1 @000000~@037774</td></tr> </table>				コ0000~コ1575	@コ0000~@コ1574	コ2000~コ7575	@コ2000~@コ7574	b0000~b1775	@b0000~@b1774	b2000~b3775	@b2000~@b3774	09000~99775	@09000~@99774	E0000~E7775	@E0000~@E7774	774M1 000000~037775	774M1 @000000~@037774
コ0000~コ1575	@コ0000~@コ1574																	
コ2000~コ7575	@コ2000~@コ7574																	
b0000~b1775	@b0000~@b1774																	
b2000~b3775	@b2000~@b3774																	
09000~99775	@09000~@99774																	
E0000~E7775	@E0000~@E7774																	
774M1 000000~037775	774M1 @000000~@037774																	
Dの使用範囲	<table border="0"> <tr><td>コ0000~コ1575</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7575</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1775</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3775</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99775</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7775</td><td>@E0000~@E7774</td></tr> <tr><td>774M1 000000~037775</td><td>774M1 @000000~@037774</td></tr> </table>				コ0000~コ1575	@コ0000~@コ1574	コ2000~コ7575	@コ2000~@コ7574	b0000~b1775	@b0000~@b1774	b2000~b3775	@b2000~@b3774	09000~99775	@09000~@99774	E0000~E7775	@E0000~@E7774	774M1 000000~037775	774M1 @000000~@037774
コ0000~コ1575	@コ0000~@コ1574																	
コ2000~コ7575	@コ2000~@コ7574																	
b0000~b1775	@b0000~@b1774																	
b2000~b3775	@b2000~@b3774																	
09000~99775	@09000~@99774																	
E0000~E7775	@E0000~@E7774																	
774M1 000000~037775	774M1 @000000~@037774																	
演算条件	入力信号の上上がり(OFF→ON)																	
演算後	S ₁ ~S ₁ +2の内容	不変																
	S ₂ ~S ₂ +2の内容	不変																
	D~D+2の内容	演算結果																
	フラグ	演算結果	ゼロ	キャリー	エラー	ノキャリー												
		0	07357	07356	07355	07354												
000001~235959		0	0	0	1													
負の数値		0	1	0	0													
時刻以外	0	0	1	0														

(解説)



命令	
STR	04002
F-37	09100
	09120
	09150

入力条件04002がOFF→ONの変化時に、レジスタ09100、09101、09102の内容(BCD6桁)とレジスタ09120、09121、09122の内容(BCD6桁)を減算してレジスタ09150、09151、09152に格納します。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 0時間を越えて昨日の時間になると、キャリーフラグ07356がONになります。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

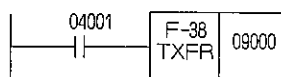
F-38 TXFR

時計現在値の転送

(この命令はJW-31CUH/H1ではプログラムできません。)

シンボル		
機能	時計の現在値(時、分、秒)をレジスタ D~D+2に転送する。	
演算内容	時計の現在値→D~D+2	
Dの使用範囲	コ0000~コ1575 @コ0000~@コ1574 コ2000~コ7575 @コ2000~@コ7574 b0000~b1775 @b0000~@b1774 b2000~b3775 @b2000~@b3774 09000~99775 @09000~@99774 E0000~E7775 @E0000~@E7774 771M1 000000~037775 771M1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Dの内容	時計データ
	フラグ	

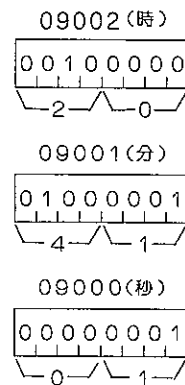
〔解説〕



命令	
STR	04001
F-38	
	09000

入力条件04001がOFF→ONの変化時に、時計の現在値をレジスタ09000、09001、09002に転送します。

時計の現在値
20時41分01秒→



- 本命令はコントロールユニットがJW-31CUH/H1のとき、使用できません。JW-31CUH/H1には時計機能がありません。
- コ0734~コ0737は特殊領域です。
(9・7ページ「データ処理命令とフラグ」参照)

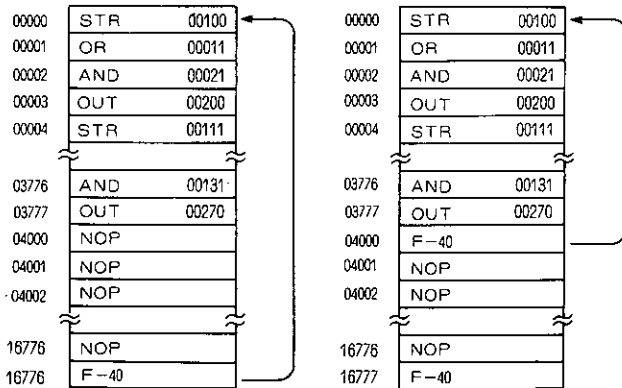
F-40 END

エンド命令 (END)

F-40(エンド命令)はプログラムの終了を意味します。END命令はメモリをクリアすると、プログラムメモリの最終アドレスに自動的に書き込みますので、次のような場合を除き特に書き込む必要はありません。

(1) スキャンタイムを早くする場合

スキャンタイムは入出力処理時間+ユーザープログラム処理時間となります。ユーザープログラム処理時間はプログラムアドレス00000からEND命令までの全命令の処理時間の合計です。メモリのクリアで自動的に書き込めるEND命令の位置は、使用しているコントロールユニットがJW-31CUH1のときは、16777(7680語目)となります。設計完了したラダー図をプログラムで書き込んだとき、その最終アドレスがたとえば03777(2048語目)とすると、04000~16776まではNOP命令、16777にEND命令が存在し、このNOPの処理時間(1語当り0.038 μ s)を空費します。少しでも演算時間を短縮するには、04000にF-40を書き込むと以下のNOP命令を処理せずユーザープログラムの演算を終了し、次のスキャンサイクルに移ります。



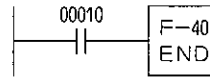
(a)メモリクリアによるEND(16777)のみ (b)04000にF-40(END)を挿入

(2) 試運転でプログラムを部分的に実行させる場合

シーケンス動作の区切毎にF-40を挿入する事でプログラムを部分的に実行させ、OKであればF-40を削除します。

- (1)、(2)でEND命令を書き込むと、F-40が複数個存在することがあります。このような場合、最初のF-40でユーザープログラムの演算を終了します。本運転の前にF-40の位置を検索して確認してください。

- F-40(END)は無条件命令です。



のようなプログラムはできません。

- F-40は優先度が一番高い命令ですが、F-141(JMP)とF-140(LABL)間、または、F-142(CALL)とF-140(LABL)間にEND命令がある場合、F-141、F-142を実行すると、そのEND命令は無視します。

F-41
JCS

ジャンプコントロールセット

(Jump Control Set)

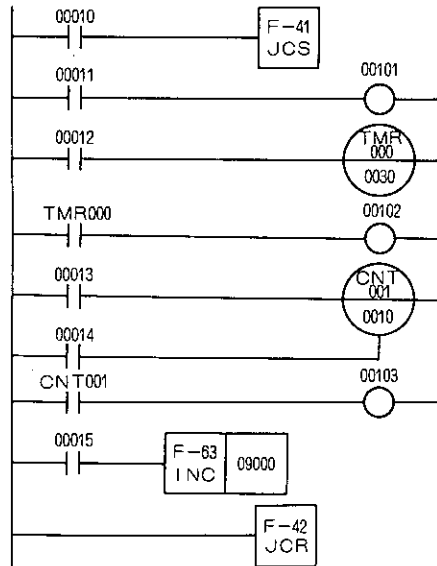
F-42
JCR

ジャンプコントロールリセット

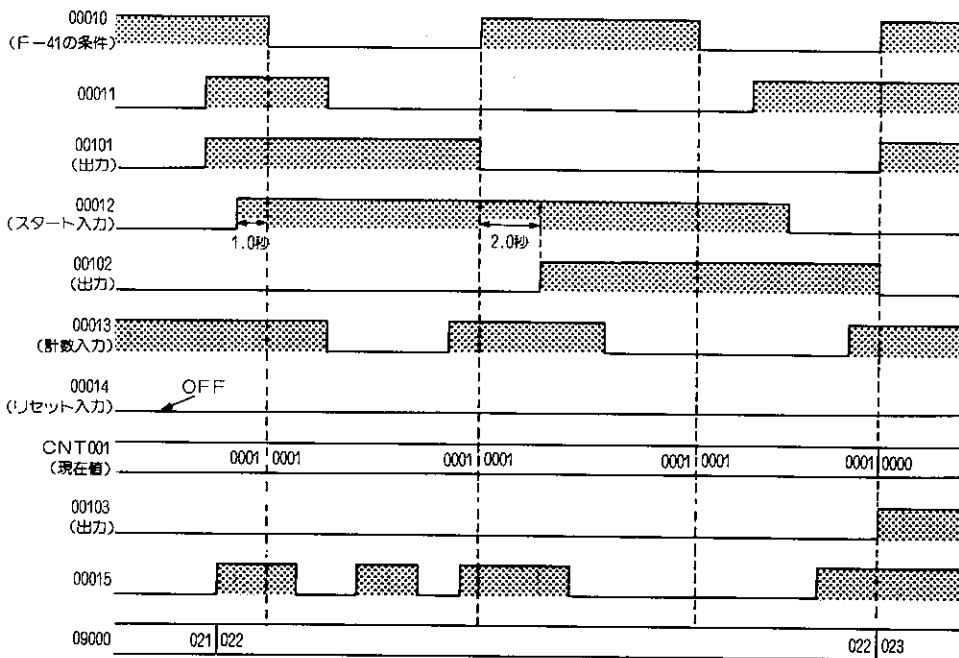
(Jump Control Reset)

F-41 (JCS)の条件がOFFの時、F-42 (JCR) までにあるEND命令を除くすべての命令は実行しません。したがってOUT命令、TMR・CNT・MD

命令、応用命令等の演算結果をデータメモリに書き込む命令があってもデータメモリの内容は変化せず、JCSの条件がONの時の状態を保持します。

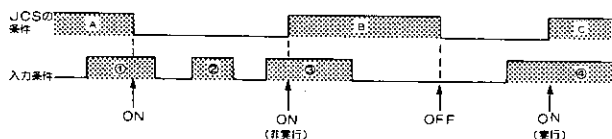


命 令	
STR	00010
F-41	
STR	00011
OUT	00101
STR	00012
TMR	000
	0030
STR TMR	000
OUT	00102
STR	00013
STR	00014
CNT	001
	0010
STR CNT	001
OUT	00103
STR	00015
F-63	
	09000
F-42	



- TMRの内部クロック(0.1秒クロック)、CNTの計数入力および応用命令の入力条件(入力条件のOFF→ONで演算を実行するもの)と、F-41 (JCS)の条件のON/OFFのタイミングに注意してください。

- ①の立上りでは、JCSの条件ONのため、演算します。
 - ②の立上りでは、JCSの条件OFFのため、演算しません。
 - ③の立上りでは、JCSの条件OFFのため、演算しません。
- ③がONの間にJCSの条件がONとなりますが、④のJCSの条件がON→OFFとなるときの入力条件がON、⑤のJCSの条件がOFF→



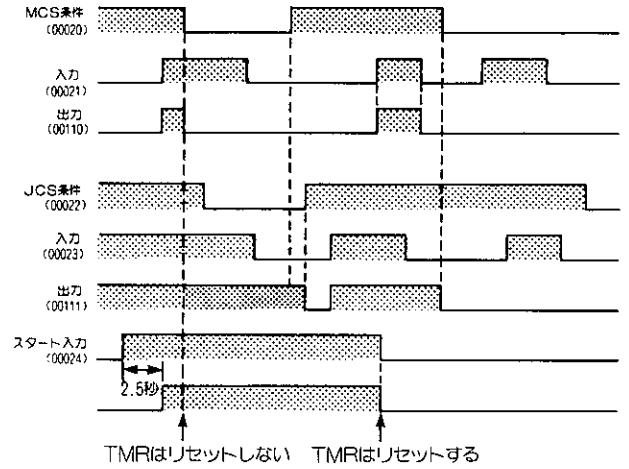
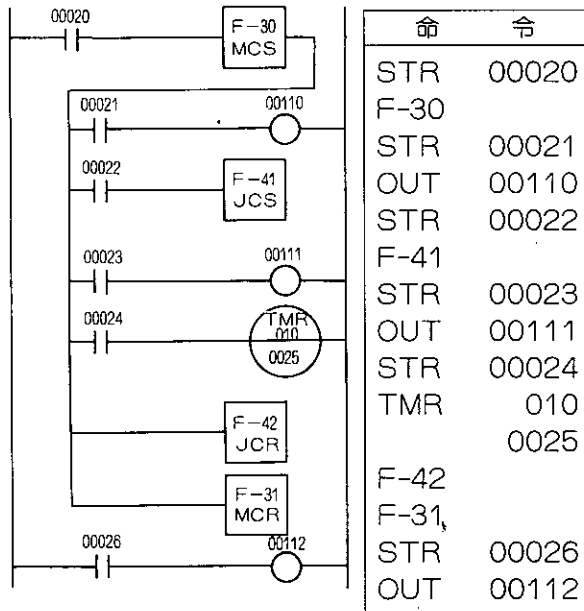
ONとなるときの入力条件もONのため、入力条件がOFF→ONに変化したとは見なさず演算しません。

④の立上りでは、JCSの条件がOFFのため、演算しません。

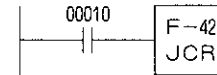
④がONの間にJCSの条件がONとなります。

⑤のJCSの条件がON→OFFとなるときの入力条件がOFF、⑥のJCSの条件がOFF→ONとなるときの入力条件はONと変化しているため、⑥のJCS条件がOFF→ONとなった直後に演算します。

- F-41(JCS)とF-42(JCR)の間に、F-40(END命令)があるとき、JCSの条件のON/OFFにかかわらずEND命令を実行し、ユーザープログラムの演算は終了し、次のスキャンサイクルに移ります。
- F-30(MCS)とF-31(MCR)の間に、F-41(JCS)、F-42(JCR)を入れ子構造でプログラムできます。ただし、MCSの条件がOFFになると、JCS～JCR間の命令は、JCSの条件のON/OFFにかかわらず非実行となります。



- F-41(JCS)とF-42(JCR)の間に、さらにF-41、F-42は入れられません。このようなプログラムを書き込むと、プログラムチェックの際、ハンディプログラマ等では「JCS ERROR」と表示します。上記の様なネスティングを行いたい場合は、F-242(JCRN)「ジャンプコントロールネスティングリセット」命令をF-42(JCR)のかわりに使用します。
- F-42(JCR)は無条件命令です。

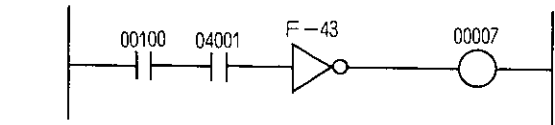


のようなプログラムはできません。

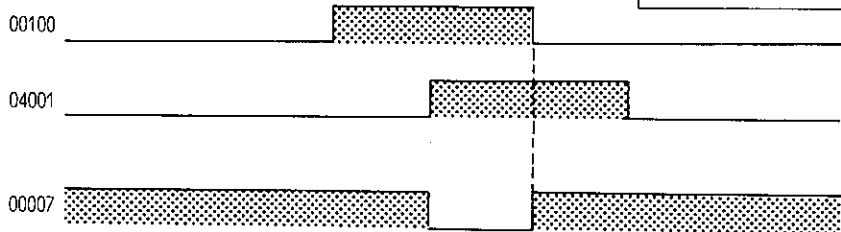
- F-41(JCS)とF-42(JCR)の間に立上りで演算する応用命令を使用する場合、F-41(JCS)の入力条件と違った条件にしてください。同一の条件を使用した場合、演算しません。

F-43 ビット反転 (ComPLement)

F-43は直前のACC(アキュムレータ)の内容を反転する命令です。

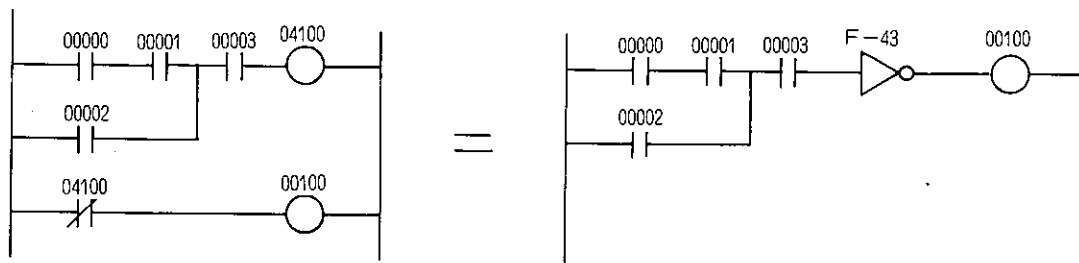


命 令	
STR	00100
AND	04001
F-43	
OUT	00007

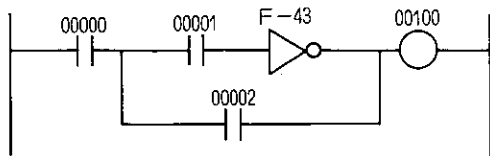


STR命令からF-43命令までの演算結果を反転し出力レリ-00007に出力します。

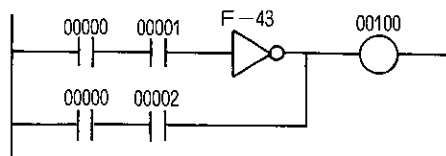
F-43を使用すると補助接点を使うことなく、反転出力が得られます。



- F-43命令の条件は、1接点でも複数の接点でも構いません。
- F-43は直前のACCの内容を反転する命令のため、次の(a)と(b)のプログラムでは同じ演算結果が得られませんので注意してください。



(a)

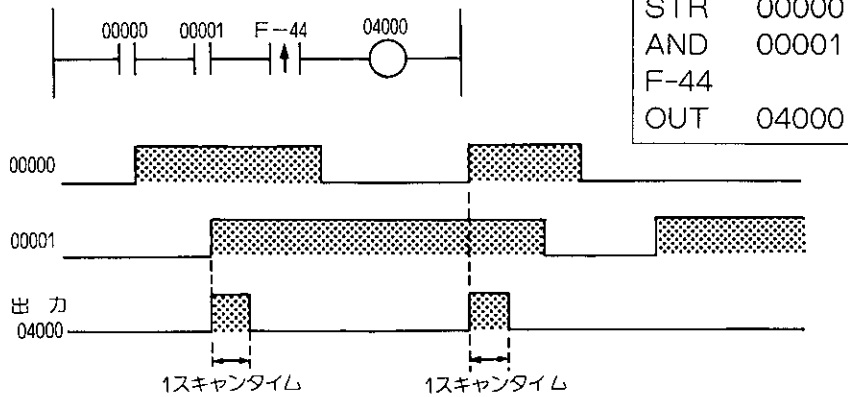


(b)

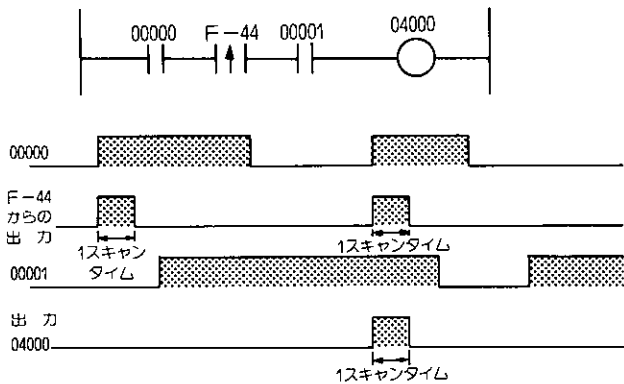
F-44
—↑—

ON時微分

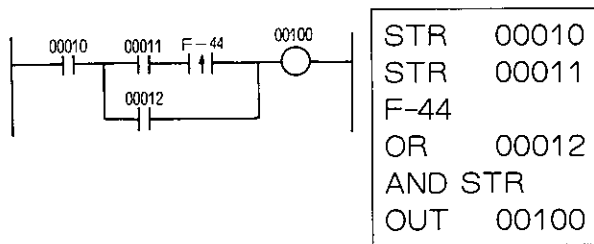
F-44命令の直前のACC(アキュムレータ)の状態がOFF→ONと変化した時に1スキャンタイムのパルスを発生します。



- 上記ラダー図でF-44のプログラム順序を変えると、結果が変わりますので、注意してください。(F-45の場合も同様です。)

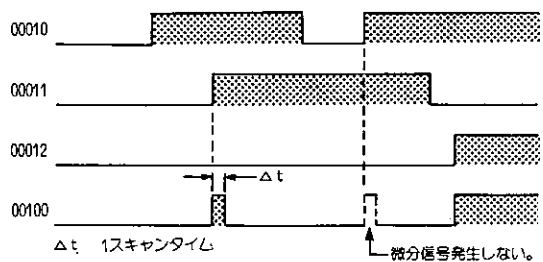


- F-44命令の条件は、1接点でも複数の接点でも構いません。



	アキュムレータ ACC	スタックレジスタ S1
STR 00010	—00010— —↑—	
STR 00011	—00011— —↑—	—00010— —↑—
F-44	—00011— F-44 —↑—	—00010— —↑—
OR 00012	—00011— F-44 —↑— —00012— —↑—	—00010— —↑—
AND STR	—00010— 00011 F-44 —↑— —↑— —↑—	
OUT 00100	—00010— 00011 F-44 —↑— —↑— —↑—	

00011がOFF→ONに変化したスキャンサイクルのみACCがON



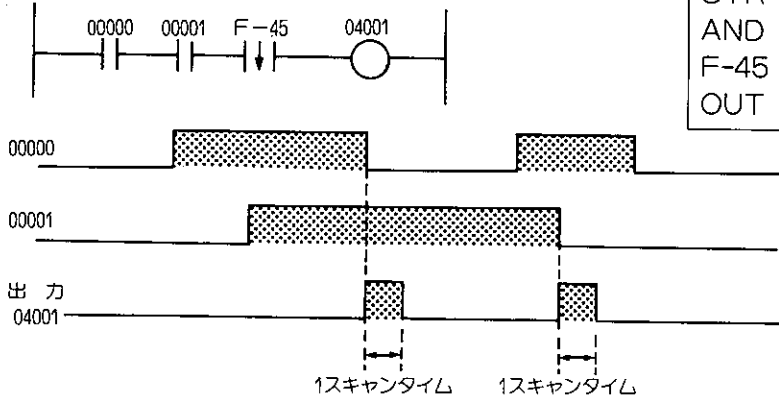
上記の例の場合、AND STR命令で00010とのANDを演算するため、00011がONのとき00010がOFF→ONとなっても微分信号は発生しません。

- F-44命令はF-47(レベル演算条件セット)とF-48(レベル演算条件リセット)の間に入れても1スキャンしか演算しません。(9・131ページF-47、F-48参照)

F-45

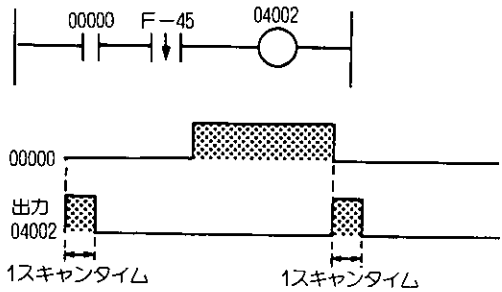
OFF時微分

F-45命令の直前のACC(アキュムレータ)の状態が、ON→OFFと変化した時に、1スキャンタイムのパルスが発生します。

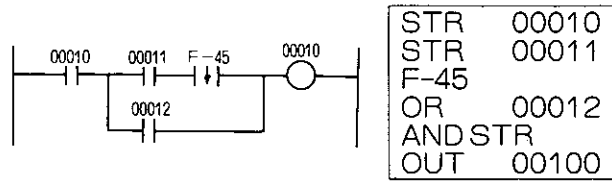


命 令	
STR	00000
AND	00001
F-45	
OUT	04001

- 微力命令(F-44、F-45)は、プログラム中何回使っても構いません。
- OFF時微力命令を使用すると、プログラム書込(F-45命令の書込又は挿入、削除などによりF-45命令のプログラムアドレスが移動する場合)直後の運転時に1スキャンタイムのパルスが発生する場合があります。



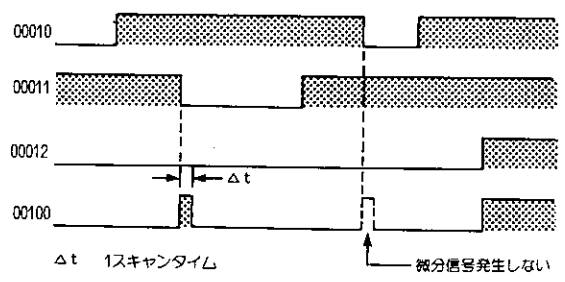
プログラム書込直後の運転開始時に、入力(00000)がOFF状態の場合、出力(04002)がONとなります。



STR	00010
STR	00011
F-45	
OR	00012
AND STR	
OUT	00010

	アキュムレータ ACC	スタックレジスタ S1
STR 00010	00010	
STR 00011	00011	00010
F-45	00011 F-45	00010
OR 00012	00011 F-45 00012	00010
AND STR	00010 00011 F-45 00012	
OUT 00100	00010 00011 F-45 00012	

00011がON→OFFに変化したスキャンサイクルのみACCがON



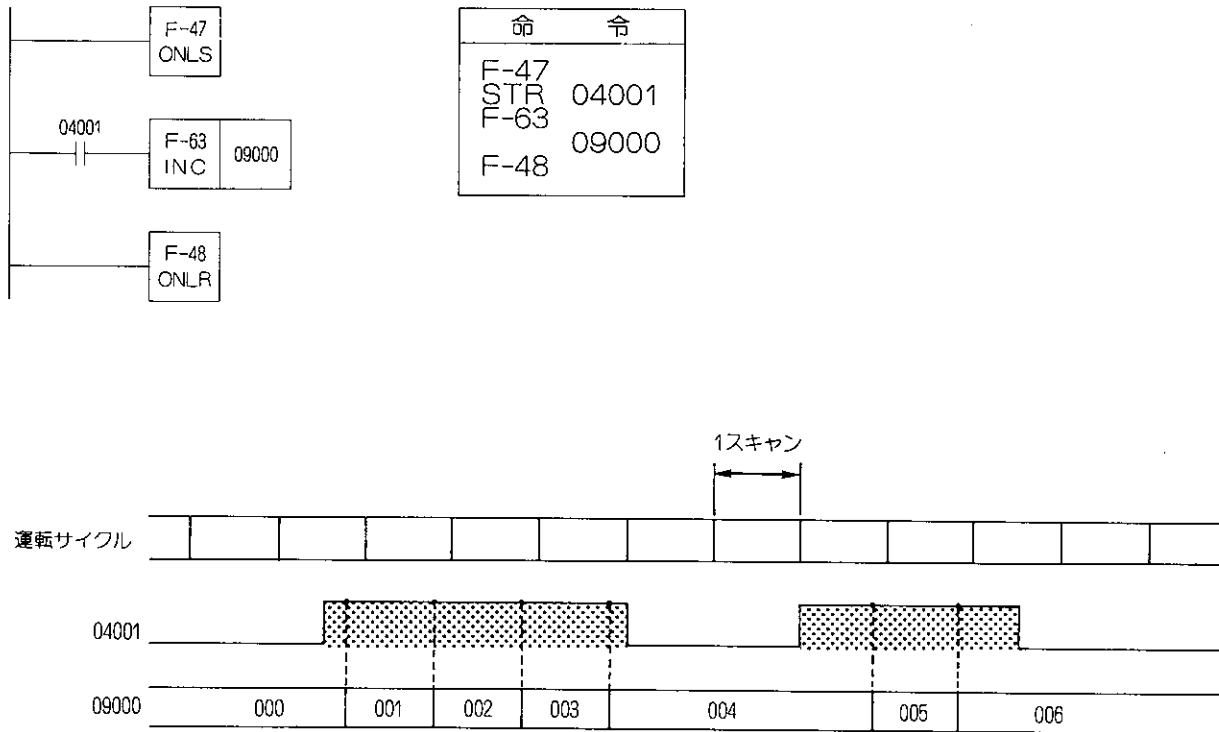
上記の例の場合、AND STR命令で00010とのANDを演算するため、00011がONのときに00010がON→OFFとなっても微分信号は発生しません。

- F-45命令はF-47(レベル演算条件セット)とF-48(レベル演算条件リセット)の間に入れても1スキャンしか演算しません。(次頁F-47、F-48参照)
- 本命令は電源投入時と運転開始直後の1スキャンの間演算しません。

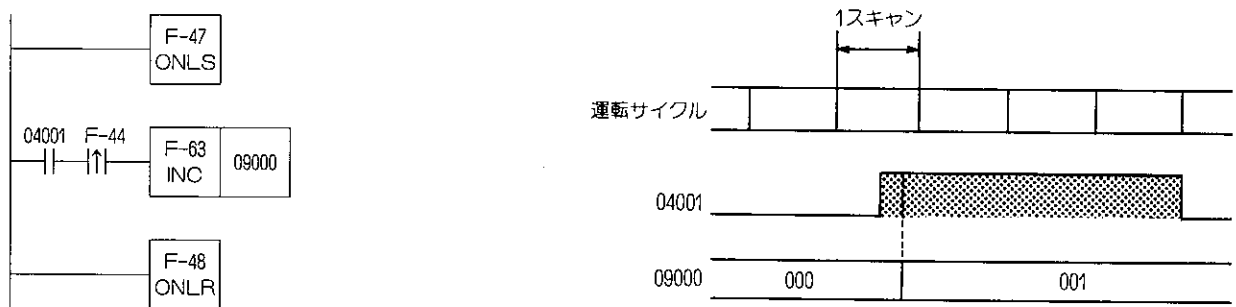
F-47 ONLS レベル演算条件セット
(ON Level Set)

F-48 ONLR レベル演算条件リセット
(ON Level Reset)

F-47(ONLS)と、F-48(ONLR)の間の命令の
立上り演算条件をレベル演算条件(ONで演算)
に設定します。



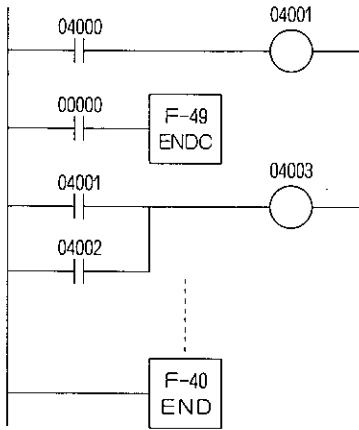
- F-47(ONLS)とF-48(ONLR)の中にさらにF-47は入
れられません。
- 微分命令(F-44、F-45)が含まれている回路では04001の立上り
時の1スキャンだけ演算します。(F-44の例)



**F-49
ENDC**

条件エンド

F-49の条件がOFFの時、シーケンス演算を終了
します。



アドレス	命 令
.....
00010	STR 04000
00011	OUT 04001
00012	STR 00000
00013	F-49
00014	STR 04001
00015	OR 04002
00016	OUT 04003
.....
36777	F-40

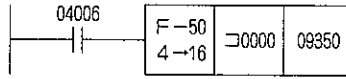
- 入力条件00000がONの時
F-40命令(アドレス36777)までの命令を実行します。
- 入力条件00000がOFFの時
アドレス00014以降の命令は実行しません。

F-50
4→16

4→16デコーダ

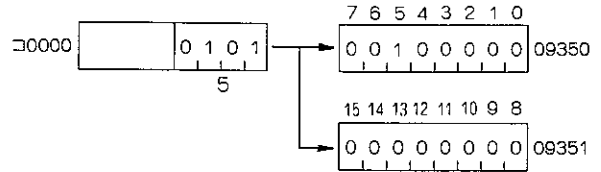
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-50 4→16</td><td style="padding: 2px;">S</td><td style="padding: 2px;">D</td></tr></table>			F-50 4→16	S	D											
F-50 4→16	S	D															
機能	レジスタSの下位4ビットのデータをデコードし、レジスタD、D+1の2バイトに16ビットのデータとして格納する。																
演算内容	S→D、D+1																
Sの使用範囲	<table border="0" style="font-family: monospace; font-size: small;"> <tr><td>コ0000~コ1577</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7577</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1777</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3777</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99777</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7777</td><td>@E0000~@E7774</td></tr> <tr><td>771#1 000000~037777</td><td>771#1 @000000~@037774</td></tr> </table>			コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	771#1 000000~037777	771#1 @000000~@037774
コ0000~コ1577	@コ0000~@コ1574																
コ2000~コ7577	@コ2000~@コ7574																
b0000~b1777	@b0000~@b1774																
b2000~b3777	@b2000~@b3774																
09000~99777	@09000~@99774																
E0000~E7777	@E0000~@E7774																
771#1 000000~037777	771#1 @000000~@037774																
Dの使用範囲	<table border="0" style="font-family: monospace; font-size: small;"> <tr><td>コ0000~コ1576</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7576</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1776</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3776</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99776</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7776</td><td>@E0000~@E7774</td></tr> <tr><td>771#1 000000~037776</td><td>771#1 @000000~@037774</td></tr> </table>			コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771#1 000000~037776	771#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																
コ2000~コ7576	@コ2000~@コ7574																
b0000~b1776	@b0000~@b1774																
b2000~b3776	@b2000~@b3774																
09000~99776	@09000~@99774																
E0000~E7776	@E0000~@E7774																
771#1 000000~037776	771#1 @000000~@037774																
演算条件	入力信号の立上り(OFF→ON)																
演算後	Sの内容	不変															
	Dの内容	演算結果(0~7)															
	D+1の内容	演算結果(8~15)															
	フラグ	不変															

(解説)



命令	
STR	04006
F-50	コ0000
	09350

入力条件4006がOFF→ONの変化時に、レジスタコ0000の下位4ビットのデータをデコードし、レジスタ09350と09351の2バイトに16ビットのデータとして格納します。



下位4ビットの数値0~15に相当するビットの位置のみがONし、その他のビットはOFFとなります。

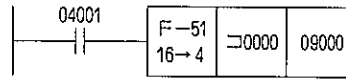
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Sの上位4ビットは演算上無視します。

F-51
16→4

16→4エンコーダ

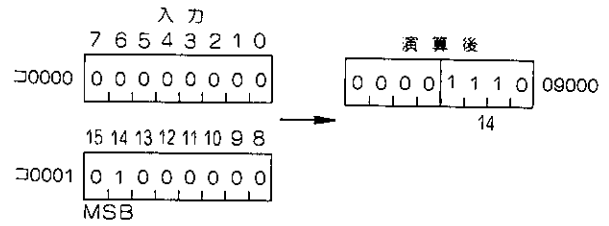
シンボル	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">F-51 16→4</td> <td style="text-align: center;">S</td> <td style="text-align: center;">D</td> </tr> </table>			F-51 16→4	S	D																		
F-51 16→4	S	D																						
機能	レジスタS、S+1の2バイトのデータをエンコードし、レジスタDに格納する。																							
演算内容	S、S+1→D																							
Sの使用範囲	<table border="0" style="font-size: small;"> <tr> <td>コ00000~コ15776</td> <td>:</td> <td>@コ00000~@コ15774</td> </tr> <tr> <td>コ20000~コ75776</td> <td>:</td> <td>@コ20000~@コ75774</td> </tr> <tr> <td>b00000~b17776</td> <td>:</td> <td>@b00000~@b17774</td> </tr> <tr> <td>b20000~b37776</td> <td>:</td> <td>@b20000~@b37774</td> </tr> <tr> <td>090000~997776</td> <td>:</td> <td>@090000~@997774</td> </tr> <tr> <td>E00000~E77776</td> <td>:</td> <td>@E00000~@E77774</td> </tr> <tr> <td>777771 000000~037776</td> <td>:</td> <td>777771 @000000~@037774</td> </tr> </table>			コ00000~コ15776	:	@コ00000~@コ15774	コ20000~コ75776	:	@コ20000~@コ75774	b00000~b17776	:	@b00000~@b17774	b20000~b37776	:	@b20000~@b37774	090000~997776	:	@090000~@997774	E00000~E77776	:	@E00000~@E77774	777771 000000~037776	:	777771 @000000~@037774
コ00000~コ15776	:	@コ00000~@コ15774																						
コ20000~コ75776	:	@コ20000~@コ75774																						
b00000~b17776	:	@b00000~@b17774																						
b20000~b37776	:	@b20000~@b37774																						
090000~997776	:	@090000~@997774																						
E00000~E77776	:	@E00000~@E77774																						
777771 000000~037776	:	777771 @000000~@037774																						
Dの使用範囲	<table border="0" style="font-size: small;"> <tr> <td>コ00000~コ15777</td> <td>:</td> <td>@コ00000~@コ15774</td> </tr> <tr> <td>コ20000~コ75777</td> <td>:</td> <td>@コ20000~@コ75774</td> </tr> <tr> <td>b00000~b17777</td> <td>:</td> <td>@b00000~@b17774</td> </tr> <tr> <td>b20000~b37777</td> <td>:</td> <td>@b20000~@b37774</td> </tr> <tr> <td>090000~997777</td> <td>:</td> <td>@090000~@997774</td> </tr> <tr> <td>E00000~E77777</td> <td>:</td> <td>@E00000~@E77774</td> </tr> <tr> <td>777771 000000~037777</td> <td>:</td> <td>777771 @000000~@037774</td> </tr> </table>			コ00000~コ15777	:	@コ00000~@コ15774	コ20000~コ75777	:	@コ20000~@コ75774	b00000~b17777	:	@b00000~@b17774	b20000~b37777	:	@b20000~@b37774	090000~997777	:	@090000~@997774	E00000~E77777	:	@E00000~@E77774	777771 000000~037777	:	777771 @000000~@037774
コ00000~コ15777	:	@コ00000~@コ15774																						
コ20000~コ75777	:	@コ20000~@コ75774																						
b00000~b17777	:	@b00000~@b17774																						
b20000~b37777	:	@b20000~@b37774																						
090000~997777	:	@090000~@997774																						
E00000~E77777	:	@E00000~@E77774																						
777771 000000~037777	:	777771 @000000~@037774																						
演算条件	入力信号の立上り(OFF→ON)																							
演算後	S、S+1の内容	不変																						
	Dの内容	演算結果																						
	フラグ	不変																						

(解説)

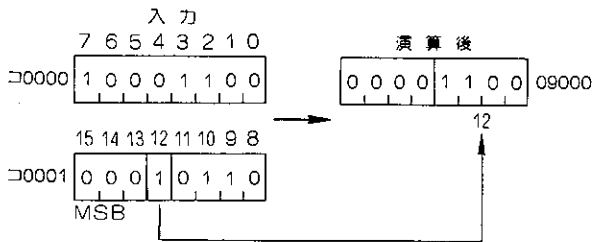


命 令	
STR	04001
F-51	コ0000 コ09000

入力条件04001がOFF→ONの変化時に、レジスタコ0000とコ09000の2バイトのデータをエンコードし、レジスタ9000に格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 演算後、D(例の場合09000)の上位4ビットは常に0になります。
- エンコーダの入力はMSB側を優先します。

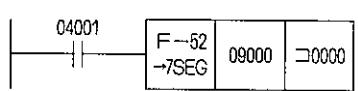


F-52
→7SEG

7SEGデコーダ

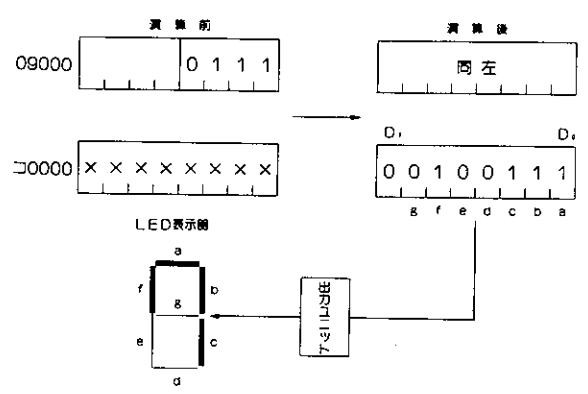
シンボル		
機能	レジスタSの下位4ビットのデータを7セグメントの表示データにデコードする。	
演算内容	S→D	
Sの使用範囲	コ00000~コ15777 : @コ00000~@コ15774 コ20000~コ75777 : @コ20000~@コ75774 b00000~b17777 : @b00000~@b17774 b20000~b37777 : @b20000~@b37774 090000~997777 : @090000~@997774 E00000~E77777 : @E00000~@E77774 77IM1 000000~037777 : 77IM1 @000000~@037774	
Dの使用範囲	コ00000~コ15777 : @コ00000~@コ15774 コ20000~コ75777 : @コ20000~@コ75774 b00000~b17777 : @b00000~@b17774 b20000~b37777 : @b20000~@b37774 090000~997777 : @090000~@997774 E00000~E77777 : @E00000~@E77774 77IM1 000000~037777 : 77IM1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Sの内容	不変
	Dの内容	演算結果 (「7セグメントデコーダ表」参照)
	フラグ	不変

(解説)



命令	
STR	04001
F-52	09000 00000

入力条件04001がOFF→ONの変化時に、レジスタ09000の内容(下4ビット)が7セグメントの表示データにデコードします。入力データと表示出力の関係は「7セグメントデコーダ表」をご覧ください。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 出力データD₀~D₆は7セグメント表示器のa~gに対応しています。D₇の出力は常に「0」です。

7セグメント デコーダ表



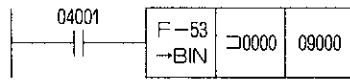
入力データ	出力データ	表示出力
	g f e d c b a	
00000000	00111111	0
00000001	00000110	1
00000010	01011011	2
00000011	01001111	3
00000100	01100110	4
00000101	01101101	5
00000110	01111101	6
00000111	00100111	7
00001000	01111111	8
00001001	01101111	9
00001010	01110111	A
00001011	01111100	b
00001100	00111001	c
00001101	01011110	d
00001110	01111001	e
00001111	01110001	f

**F-53
→BIN**

BCD(4桁)→BIN(16ビット)変換

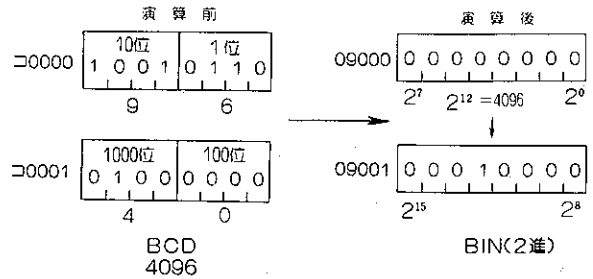
シンボル	F-53 →BIN S D				
機能	レジスタS、S+1の2バイトのBCD 4桁データを2進に変換し、レジスタD、D+1の2バイトに格納する。				
演算内容	S、S+1→D、D+1				
Sの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 771M1 000000~037776 : 771M1 @000000~@037774				
Dの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 771M1 000000~037776 : 771M1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	S、S+1の内容	不変			
	Dの内容	演算結果 (0~255)	レジスタS、S+1の内容がBCDコードでない時不変		
	D+1の内容	演算結果 (256~9999)			
フラグ	レジスタS、S+1の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
	BCDコード	0	0	0	0
	BCDコードでない時	0	0	1	0

(解説)



命令	
STR	04001
F-53	コ0000 09000

入力条件04001がOFF→ONの変化時に、レジスタコ0000とコ0001のBCD 4桁データを2進に変換し、レジスタ09000と09001の2バイトに変換データを格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- F-53でプログラムを作成するとモニタ時F-03wで表示します。

参考 下記のF命令は働きが類似しています。

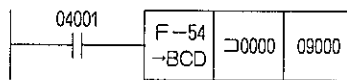
F-03、F-03w、F-53、F-153

**F-54
→BCD**

BIN(16ビット)→BCD(6桁)変換

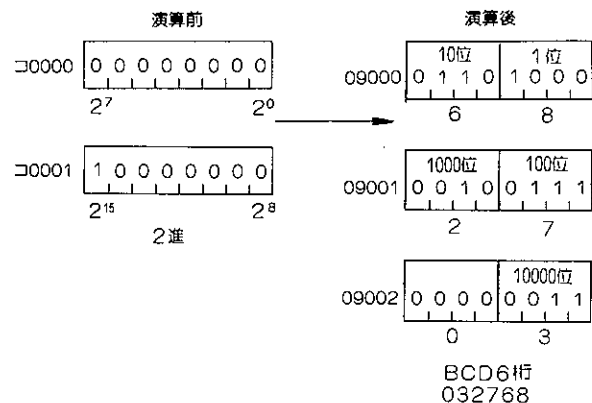
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-54 →BCD</td><td style="padding: 2px;">S</td><td style="padding: 2px;">D</td></tr></table>		F-54 →BCD	S	D											
F-54 →BCD	S	D														
機能	レジスタS、S+1の2バイトの2進データをBCD6桁に変換し、レジスタD、D+1、D+2の3バイトに格納する。															
演算内容	S、S+1→D、D+1、D+2															
Sの使用範囲	<table style="font-size: small; border-collapse: collapse;"> <tr><td>コ0000~コ1576</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7576</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1776</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3776</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99776</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7776</td><td>@E0000~@E7774</td></tr> <tr><td>771M1 000000~037776</td><td>771M1 @000000~@037774</td></tr> </table>		コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771M1 000000~037776	771M1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574															
コ2000~コ7576	@コ2000~@コ7574															
b0000~b1776	@b0000~@b1774															
b2000~b3776	@b2000~@b3774															
09000~99776	@09000~@99774															
E0000~E7776	@E0000~@E7774															
771M1 000000~037776	771M1 @000000~@037774															
Dの使用範囲	<table style="font-size: small; border-collapse: collapse;"> <tr><td>コ0000~コ1575</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7575</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1775</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3775</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99775</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7775</td><td>@E0000~@E7774</td></tr> <tr><td>771M1 000000~037775</td><td>771M1 @000000~@037774</td></tr> </table>		コ0000~コ1575	@コ0000~@コ1574	コ2000~コ7575	@コ2000~@コ7574	b0000~b1775	@b0000~@b1774	b2000~b3775	@b2000~@b3774	09000~99775	@09000~@99774	E0000~E7775	@E0000~@E7774	771M1 000000~037775	771M1 @000000~@037774
コ0000~コ1575	@コ0000~@コ1574															
コ2000~コ7575	@コ2000~@コ7574															
b0000~b1775	@b0000~@b1774															
b2000~b3775	@b2000~@b3774															
09000~99775	@09000~@99774															
E0000~E7775	@E0000~@E7774															
771M1 000000~037775	771M1 @000000~@037774															
演算条件	入力信号の立上り(OFF→ON)															
演算後	S、S+1の内容	不変														
	Dの内容	演算結果(1の位と10の位)														
	D+1の内容	演算結果(100の位と1,000の位)														
	D+2の内容	演算結果(10,000の位)														
	フラグ	不変														

(解説)



命 令	
STR	04001
F-54	コ0000
	9000

入力条件04001がOFF→ONの変化時に、レジスタコ0000とコ0001の2バイト2進データをBCD6桁に変換し、レジスタ09000からの3バイトに変換データを格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- F-54でプログラムを作成するとモニタ時F-04wで表示します。

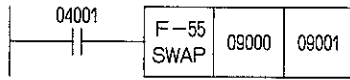
参考 下記のF命令は働きが類似しています。
F-04、F-04w、F-54、F-154

F-55 SWAP

上位4ビットと下位4ビットの交換 (SWAP)

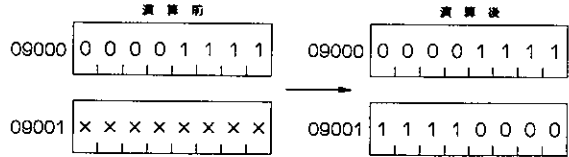
シンボル		
機能	レジスタSの内容の上下4ビットずつを交換し、レジスタDに格納する。	
演算内容	S → D	
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~09977 @09000~@09974 E0000~E7777 @E0000~@E7774 771#1 00000~03777 771#1 @00000~@03774	
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~09977 @09000~@09974 E0000~E7777 @E0000~@E7774 771#1 00000~03777 771#1 @00000~@03774	
演算条件	入力信号の立上り(OFF → ON)	
演算後	Sの内容	不変
	Dの内容	演算結果
	フラグ	不変

(解説)



命令	
STR	04001
F-55	09000
	09001

入力条件04001がOFF → ONの変化時に、レジスタ09000の上下4ビットずつを交換し、レジスタ09001に格納します。
レジスタ09000の内容は不変です。



●コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

参考 F-55命令は次のようなときに有効です。
F-52命令(7SEGデコーダ)は、下4ビットを7セグメントデータにデコードします。多桁の表示をするとき、F-55命令により上4ビットと下4ビットを交換し、再度F-52を使用します。

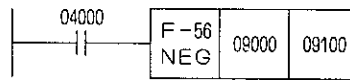
参考 下記のF命令は働きが類似しています。
F-55、F-175

**F-56
NEG**

1バイトデータの10の補数

シンボル	— F-56 NEG S D				
機能	レジスタSの内容(1バイトデータ)を2桁のBCDコードと見なし、その値の10の補数を取り、レジスタDに格納する。				
演算内容	100-S→D				
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774				
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	Sの内容	不変			
	Dの内容	●演算結果 ●レジスタSの内容がBCDコードでない時不変			
フラグ	レジスタSの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
	BCDコード	0	0	0	0
	BCDコードでない時	0	0	1	0

(解説)



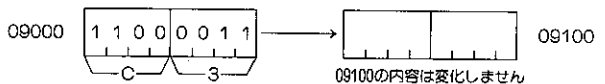
命令	
STR	04000
F-56	09000
	09100

入力条件04000がOFF→ONの変化時に、レジスタ09000の内容を2桁のBCDコードと見なし、その値の10の補数を取りレジスタ09100に格納します。09000の内容がBCDコード以外の場合、09100の内容は変化せず、エラーフラグ(07355)が1になります。



100-58=42

ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
0	0	0	0



ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
0	0	1	0

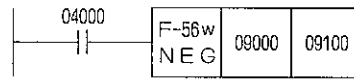
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

**F-56w
NEG**

1ワードデータの10の補数

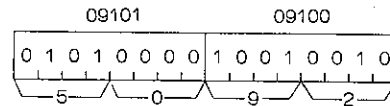
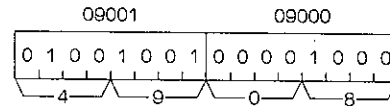
シンボル	<table border="1"> <tr> <td>F-56w NEG</td> <td>S</td> <td>D</td> </tr> </table>				F-56w NEG	S	D											
F-56w NEG	S	D																
機能	レジスタS、S+1の内容(1ワードデータ)を4桁のBCDコードと見なし、その値の10の補数を取り、レジスタD、D+1に格納する。																	
演算内容	10000-(S、S+1)→D、D+1																	
Sの使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>77#1 000000~037776</td> <td>77#1 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	77#1 000000~037776	77#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																	
コ2000~コ7576	@コ2000~@コ7574																	
b0000~b1776	@b0000~@b1774																	
b2000~b3776	@b2000~@b3774																	
09000~99776	@09000~@99774																	
E0000~E7776	@E0000~@E7774																	
77#1 000000~037776	77#1 @000000~@037774																	
Dの使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>77#1 000000~037776</td> <td>77#1 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	77#1 000000~037776	77#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																	
コ2000~コ7576	@コ2000~@コ7574																	
b0000~b1776	@b0000~@b1774																	
b2000~b3776	@b2000~@b3774																	
09000~99776	@09000~@99774																	
E0000~E7776	@E0000~@E7774																	
77#1 000000~037776	77#1 @000000~@037774																	
演算条件	入力信号の立上り(OFF→ON)																	
演算後	S、S+1の内容	不変																
	Dの内容	演算結果(下位)	レジスタS、S+1の内容がBCDコードでない時不変															
	D+1の内容	演算結果(上位)																
フラグ	レジスタS、S+1の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354													
	BCDコード	0	0	0	0													
	BCDコードでない時	0	0	1	0													

(解説)



命 令	
STR	04000
F-56w	09000
	09100

入力条件04000がOFF→ONの変化時に、レジスタ09000、09001の内容を4桁のBCDコードと見なし、その値の10の補数を取りレジスタ09100、09101に格納します。



10000 - 4908 = 5092

ゼロ	キャリー	エラー	ノキャリー
07357	07356	07355	07354
0	0	0	0

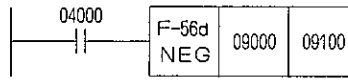
- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

**F-56d
NEG**

2ワードデータの10の補数

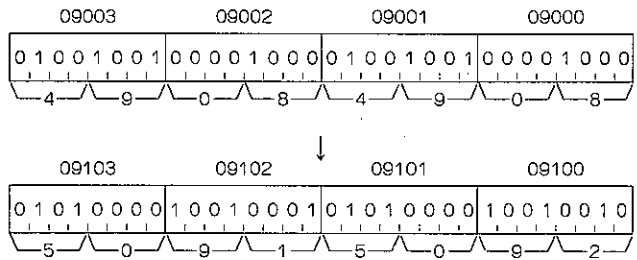
シンボル	<table border="1"> <tr> <td>F-56d NEG</td> <td>S</td> <td>D</td> </tr> </table>				F-56d NEG	S	D																		
F-56d NEG	S	D																							
機能	レジスタS~S+3の内容(2ワードデータ)を8桁のBCDコードと見なし、その値の10の補数を取り、レジスタD~D+3に格納する。																								
演算内容	100000000-(S~S+3)→D~D+3																								
Sの使用範囲	<table border="0"> <tr> <td>コ0000~コ1574</td> <td>:</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7574</td> <td>:</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1774</td> <td>:</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3774</td> <td>:</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99774</td> <td>:</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7774</td> <td>:</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771M1 000000~037774</td> <td>:</td> <td>771M1 @000000~@037774</td> </tr> </table>				コ0000~コ1574	:	@コ0000~@コ1574	コ2000~コ7574	:	@コ2000~@コ7574	b0000~b1774	:	@b0000~@b1774	b2000~b3774	:	@b2000~@b3774	09000~99774	:	@09000~@99774	E0000~E7774	:	@E0000~@E7774	771M1 000000~037774	:	771M1 @000000~@037774
コ0000~コ1574	:	@コ0000~@コ1574																							
コ2000~コ7574	:	@コ2000~@コ7574																							
b0000~b1774	:	@b0000~@b1774																							
b2000~b3774	:	@b2000~@b3774																							
09000~99774	:	@09000~@99774																							
E0000~E7774	:	@E0000~@E7774																							
771M1 000000~037774	:	771M1 @000000~@037774																							
Dの使用範囲	<table border="0"> <tr> <td>コ0000~コ1574</td> <td>:</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7574</td> <td>:</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1774</td> <td>:</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3774</td> <td>:</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99774</td> <td>:</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7774</td> <td>:</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771M1 000000~037774</td> <td>:</td> <td>771M1 @000000~@037774</td> </tr> </table>				コ0000~コ1574	:	@コ0000~@コ1574	コ2000~コ7574	:	@コ2000~@コ7574	b0000~b1774	:	@b0000~@b1774	b2000~b3774	:	@b2000~@b3774	09000~99774	:	@09000~@99774	E0000~E7774	:	@E0000~@E7774	771M1 000000~037774	:	771M1 @000000~@037774
コ0000~コ1574	:	@コ0000~@コ1574																							
コ2000~コ7574	:	@コ2000~@コ7574																							
b0000~b1774	:	@b0000~@b1774																							
b2000~b3774	:	@b2000~@b3774																							
09000~99774	:	@09000~@99774																							
E0000~E7774	:	@E0000~@E7774																							
771M1 000000~037774	:	771M1 @000000~@037774																							
演算条件	入力信号の立上り(OFF→ON)																								
演算後	S~S+1の内容	不変																							
	Dの内容	演算結果	S~S+3の内容がBCDコードでない時不変																						
	フラグ	レジスタS~S+3の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354																			
	BCDコード	0	0	0	0																				
	BCDコードでない時	0	0	1	0																				

(解説)



命令	
STR	04000
F-56d	09000
	09100

入力条件04000がOFF→ONの変化時に、レジスタ09000~09003の内容を8桁のBCDコードと見なし、その値の10の補数を取りレジスタ09100~09103に格納します。



$$100000000 - 49084908 = 50915092$$

ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
0	0	0	0

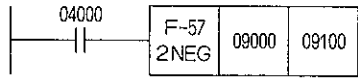
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

**F-57
2NEG**

1バイトデータの2の補数

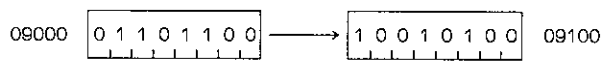
シンボル	<table border="1"> <tr> <td>F-57</td> <td>S</td> <td>D</td> </tr> <tr> <td>2NEG</td> <td></td> <td></td> </tr> </table>		F-57	S	D	2NEG		
F-57	S	D						
2NEG								
機能	レジスタSの内容(8ビットデータ)の2の補数を取りレジスタDに格納する。							
演算内容	0-S→D							
Sの使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 Q90000~Q99777 @Q90000~@Q99774 E00000~E77777 @E00000~@E77774 7741b1 000000~037777 7741b1 @000000~@037774							
Dの使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 Q90000~Q99777 @Q90000~@Q99774 E00000~E77777 @E00000~@E77774 7741b1 000000~037777 7741b1 @000000~@037774							
演算条件	入力信号の立上り(OFF→ON)							
演算後	Sの内容	不変						
	Dの内容	演算結果						
	フラグ	不変						

(解説)



命令	
STR	04000
F-57	09000
	09100

入力条件04000がOFF→ONの変化時に、レジスタ09000の内容(8ビットデータ)の2の補数を取り、レジスタ09100に格納します。

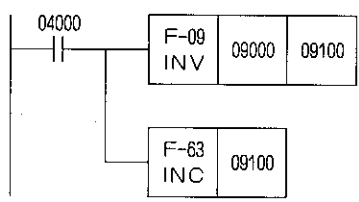


(2の補数の作り方)

- i) すべてのビットを反転する(0なら1、1なら0とする)
- ii) i)を施した数に+1する。

●コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

参考 上記の場合、下記命令と同じ動作となります。

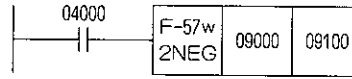


**F-57w
2NEG**

1ワードデータの2の補数

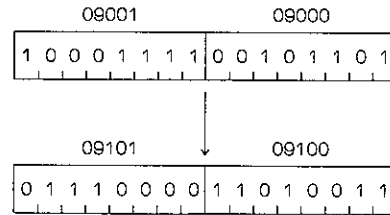
シンボル	<table border="1"> <tr> <td>F-57w</td> <td>S</td> <td>D</td> </tr> <tr> <td>2NEG</td> <td></td> <td></td> </tr> </table>	F-57w	S	D	2NEG										
F-57w	S	D													
2NEG															
機能	レジスタS、S+1の内容(1ワードデータ)の2の補数を取り、レジスタD、D+1に格納する。														
演算内容	0-(S、S+1)→D、D+1														
Sの使用範囲	<table border="0"> <tr><td>コ0000~コ1576</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7576</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1776</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3776</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99776</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7776</td><td>@E0000~@E7774</td></tr> <tr><td>77771 000000~037776</td><td>77771 @000000~@037774</td></tr> </table>	コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	77771 000000~037776	77771 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574														
コ2000~コ7576	@コ2000~@コ7574														
b0000~b1776	@b0000~@b1774														
b2000~b3776	@b2000~@b3774														
09000~99776	@09000~@99774														
E0000~E7776	@E0000~@E7774														
77771 000000~037776	77771 @000000~@037774														
Dの使用範囲	<table border="0"> <tr><td>コ0000~コ1576</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7576</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1776</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3776</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99776</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7776</td><td>@E0000~@E7774</td></tr> <tr><td>77771 000000~037776</td><td>77771 @000000~@037774</td></tr> </table>	コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	77771 000000~037776	77771 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574														
コ2000~コ7576	@コ2000~@コ7574														
b0000~b1776	@b0000~@b1774														
b2000~b3776	@b2000~@b3774														
09000~99776	@09000~@99774														
E0000~E7776	@E0000~@E7774														
77771 000000~037776	77771 @000000~@037774														
演算条件	入力信号の立上り(OFF→ON)														
演算後	S、S+1の内容	不変													
	Dの内容	演算結果(下位)													
	D+1の内容	演算結果(上位)													
	フラグ	不変													

〔解説〕



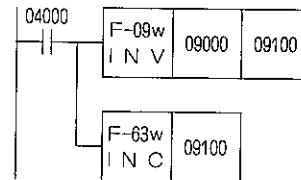
命令	
STR	04000
F-57w	09000
	09100

入力条件04000がOFF→ONの変化時に、レジスタ09000、09001の内容(16ビットデータ)の2の補数を取り、レジスタ09100、09101に格納する。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

〔参考〕 上記の場合、下記命令と同じ動作となります。

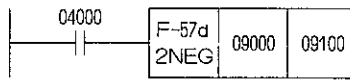


F-57d
2NEG

2ワードデータの2の補数

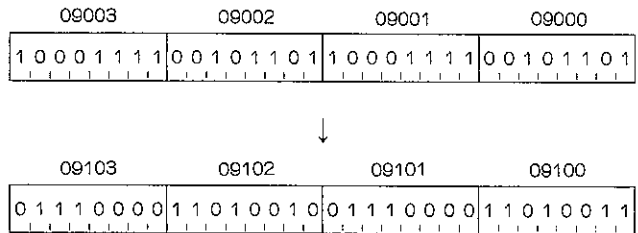
シンボル	$\overline{04000}$ F-57d 2NEG S D	
機能	レジスタS~S+3の内容(2ワードデータ)の2の補数を取り、レジスタD~D+3に格納する。	
演算内容	$0 - (S \sim S+3) \rightarrow D \sim D+3$	
Sの使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 ファイル000000~037774 : ファイル@000000~@037774	
Dの使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 ファイル000000~037774 : ファイル@000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	S、S+1の内容	不変
	Dの内容	演算結果(下位)
	D+1の内容	演算結果(上位)
	フラグ	不変

(解説)



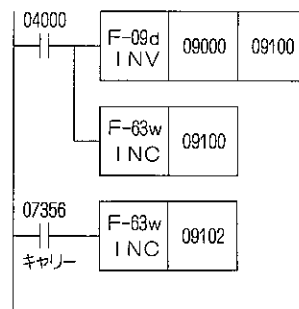
命令	
STR	04000
F-57d	09000
	09100

入力条件04000がOFF→ONの変化時に、レジスタ09000~09003の内容(32ビットデータ)の2の補数を取り、レジスタ09100~09103に格納する。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)

参考 上記の場合、下記命令と同じ動作となります。

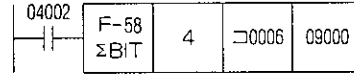


F-58
ΣBIT

ONビット数の合計

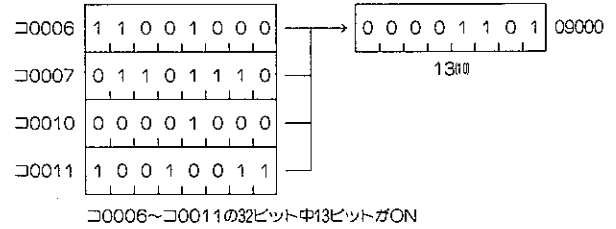
シンボル	Σ BIT	n	S	D
機能	レジスタSを先頭とするnバイトのレジスタ中のONビット数をレジスタDに格納する。			
演算内容	ONビット数→D			
nの使用範囲	0~7(0とすると8/バイトとなる)			
Sの使用範囲	コ00000~コ15777 ! @コ00000~@コ15774 コ20000~コ75777 ! @コ20000~@コ75774 b00000~b17777 ! @b00000~@b17774 b20000~b37777 ! @b20000~@b37774 090000~997777 ! @090000~@997774 E00000~E77777 ! @E00000~@E77774 771#1 000000~037777 ! 771#1 @000000~@037774			
Dの使用範囲	コ00000~コ15777 ! @コ00000~@コ15774 コ20000~コ75777 ! @コ20000~@コ75774 b00000~b17777 ! @b00000~@b17774 b20000~b37777 ! @b20000~@b37774 090000~997777 ! @090000~@997774 E00000~E77777 ! @E00000~@E77774 771#1 000000~037777 ! 771#1 @000000~@037774			
演算条件	入力信号の立上り(OFF→ON)			
演算後	S, S+1, …… S+n-1の内容	不変		
	Dの内容	演算結果		
	フラグ	不変		

(解説)



命令	
STR	04002
F-58	4
	コ0006
	09000

入力条件04002がOFF→ONの変化時、レジスタコ0006を先頭とする4/バイトのレジスタ中のONビット数をレジスタ09000に格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

F-60
SFR

両方向シフトレジスタ(1バイト)
(Forward/Backward ShiFt Register)

シンボル		①シフト方向指示入力 ②データ入力 ③シフト入力 ④リセット入力				
機能	レジスタDの8ビットデータをシフト方向指示入力①に従って上位ビット、又は下位ビットへシフトする。					
演算内容	<ul style="list-style-type: none"> シフト方向指示入力①がONの場合 シフト方向指示入力①がOFFの場合 					
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774					
演算条件	リセット入力④がOFFの時、シフト入力③の立上り(OFF→ON)でシフト					
演算後	Dの内容	<ul style="list-style-type: none"> リセット入力④がOFFの時、演算結果 リセット入力④がONの時、全ビットOFF 				
	フラグ	リセット入力④	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
		OFF	0又は1	0又は1	0	1又は0
ON	0	0	0	0		

●コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

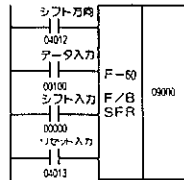
●リセット入力④はシステムメモリ(#202)にリセット条件を設定することにより「OFFでリセット」もできます。

●フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。

(9・7ページ「データ処理命令とフラグ」参照)

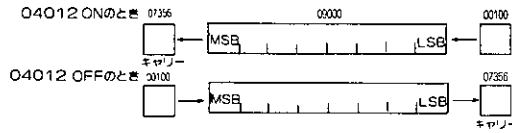
参考 下記のF命令は働きが類似しています。
F-60、F-60w、F-60d、F-160、Fc-160

(解説)



命 令	
STR	04012
STR	00100
STR	00000
STR	04013
T-60	09000

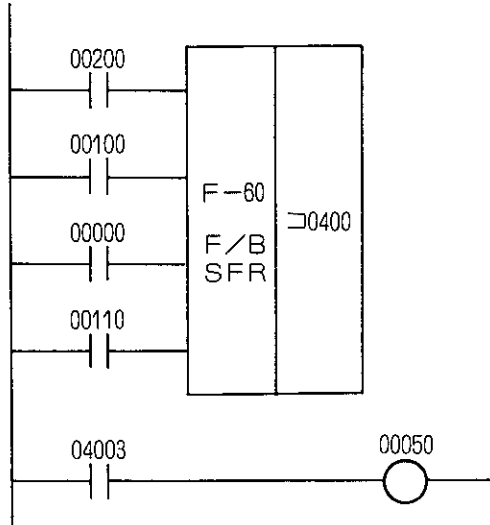
シフト入力00000のOFF→ONの変化時、シフト方向指示入力04012の状態により、次のようにシフトします。



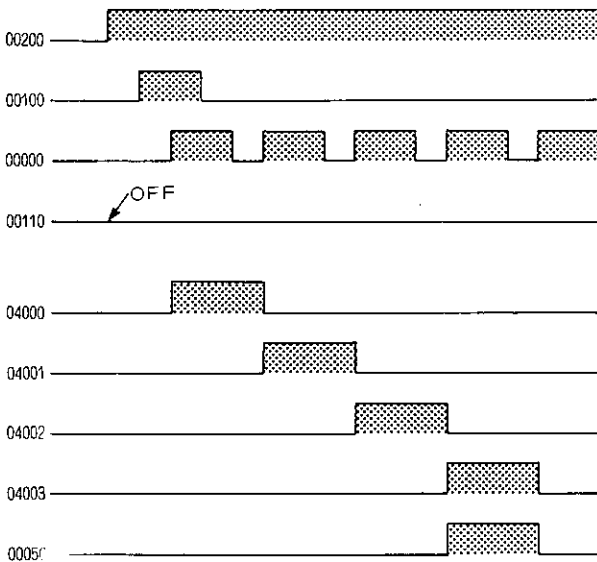
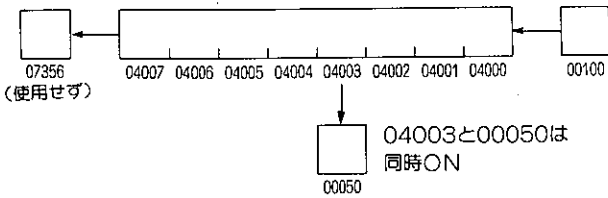
入 力 条 件	09000 (演算前)								09000 (演算後)								ゼ 〇	キ ャ リ ー	ノ ン キ ャ リ ー
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	07357	07356	07354
04012 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●
00100 ○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●
00000 ↑	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○
04013 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	○
04012 ●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●
00100 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●
00000 ↓	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○
04013 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●	○
04012 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●
00100 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●
00000 ↓	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
04013 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
4013 ●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

● エラーフラグ(07355)は常にOFFとなります。 ○ OFF ● ON

参考 Dにコ××××の領域を使用すると、nビット(n < 8) のシフトレジスタを構成できます。

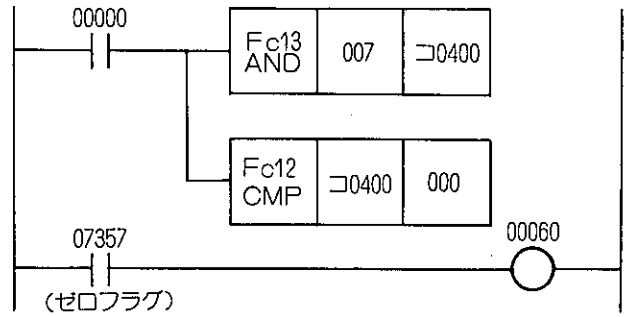


(00200がONの場合)



(注1) 04004~04007にもデータをシフトします。

(注2) ゼロフラグは04000~04007が全て0のとき1となります。04000~04002が0であることを確認するときは、次のプログラムを追加します。



000000111 とANDすることで04003~04007をマスク(すべて0にする)しています。

**F-60w
SFR**

両方向シフトレジスタ(1ワード)
(Forward/Backward Shift Register)

シンボル		①シフト方向指示入力 ②データ入力 ③シフト入力 ④リセット入力														
機能	レジスタD、D+1の16ビットデータをシフト方向指示入力①に従って上位ビット、又は下位ビットへシフトする。															
演算内容	●シフト方向指示入力①がONの場合 ●シフト方向指示入力①がOFFの場合 															
Dの使用範囲	コ0000～コ1576 ; @コ0000～@コ1574 コ2000～コ7576 ; @コ2000～@コ7574 b0000～b1776 ; @b0000～@b1774 b2000～b3776 ; @b2000～@b3774 09000～99776 ; @09000～@99774 E0000～E7776 ; @E0000～@E7774 ファイル1 000000～037776 ; ファイル1 @000000～@037774															
演算条件	リセット入力④がOFFの時、シフト入力③の立上り(OFF→ON)でシフト															
演算後	D、D+1の内容	●リセット入力④がOFFの時、演算結果 ●リセット入力④がONの時、全ビットOFF														
	フラグ	<table border="1"> <tr> <th>リセット入力④</th> <th>ゼロ 07357</th> <th>キャリー 07356</th> <th>エラー 07355</th> <th>ノンキャリー 07354</th> </tr> <tr> <td>OFF</td> <td>0又は1</td> <td>0又は1</td> <td rowspan="2">0</td> <td>1又は0</td> </tr> <tr> <td>ON</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>	リセット入力④	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354	OFF	0又は1	0又は1	0	1又は0	ON	0	0	0
		リセット入力④	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354										
OFF	0又は1	0又は1	0	1又は0												
ON	0	0		0												

- コ0734～コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
 - Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
 - フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
 - リセット入力④はシステムメモリ(#202)の設定で「OFFでリセット」とすることもできます。
- 【参考】 下記のF命令は働きが類似しています。
F-60、F-60w、F-60d、F-160、Fc-160

(解説)

00000	F-60w SFR	09000
00001		
00002		
00003		

命令	
STR	00000
STR	00001
STR	00002
STR	00003
F-60w	09000

00000(①)ON-----MSB方向へシフト
 00001(②)ON-----データ入力ON
 00002(③)OFF→ON-----シフト指示
 00003(④)OFF-----リセット機能なし

入力条件が上記の場合、演算結果は次のようになります。

09001	09000
演算前	1 0 0 1 0 1 1 0 1 0 1 0 0 1 0 0
	↓
09001	09000
演算後	0 0 1 0 1 1 0 1 0 1 0 0 1 0 0 1

ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
0	1	0	0

F-60d SFR

両方向シフトレジスタ(2ワード)
(Forward/Backward Shift Register)

シンボル	<p>①シフト方向指示入力 ②データ入力 ③シフト入力 ④リセット入力</p>														
機能	レジスタD~D+3の32ビットデータをシフト方向指示入力①に従って上位ビット、又は下位ビットへシフトする。														
演算内容	<p>●シフト方向指示入力①がONの場合</p> <p>●シフト方向指示入力①がOFFの場合</p>														
Dの使用範囲	<p>コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 ファイル000000~037774 : ファイル@000000~@037774</p>														
演算条件	リセット入力④がOFFの時、シフト入力③の立上り(OFF→ON)でシフト														
演算内容	<p>D、D+1の</p> <ul style="list-style-type: none"> ●リセット入力④がOFFの時、演算結果 ●リセット入力④がONの時、全ビットOFF 														
算後フラグ	<table border="1"> <tr> <td>リセット入力④</td> <td>ゼロ 07357</td> <td>キャリー 07356</td> <td>エラー 07355</td> <td>ノンキャリー 07354</td> </tr> <tr> <td>OFF</td> <td>0又は1</td> <td>0又は1</td> <td rowspan="2">0</td> <td>1又は0</td> </tr> <tr> <td>ON</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>	リセット入力④	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354	OFF	0又は1	0又は1	0	1又は0	ON	0	0	0
リセット入力④	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354											
OFF	0又は1	0又は1	0	1又は0											
ON	0	0		0											

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- リセット入力④はシステムメモリ(#202)の設定で「OFFでリセット」とすることもできます。

参考 下記のF命令は働きが類似しています。
F-60、F-60w、F-60d、F-160、Fc-160

(解説)

命令	
STR	00000
STR	00001
STR	00002
STR	00003
F-60d	09000

00000(①)ON MSB方向へシフト
 00001(②)ON データ入力ON
 00002(③)OFF→ON シフト指示
 00003(④)OFF リセット機能なし
 入力条件が上記の場合、演算結果は次のようになります。

演算前

09003	09002	09001	09000
10010110	10100100	10010110	10100100

演算後

09003	09002	09001	09000
00101101	01001001	00101101	01001001

ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
0	1	0	0

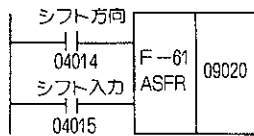
F-61 ASFR

非同期両方向シフトレジスタ(1バイト)
(Asynchronous ShiFt Register)

シンボル			①シフト方向指示入力 ②シフト入力														
機能	シフト方向指示入力①に従って、レジスタD-1(①ON)またはレジスタD+1(①OFF)の1バイトデータをレジスタDにシフトする。																
演算内容	<ul style="list-style-type: none"> シフト方向指示入力ONのとき D-1 → D シフト方向指示入力OFFのとき D+1 → D 																
Dの使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>77771 000000~037776</td> <td>77771 @000000~@037774</td> </tr> </table>			コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	77771 000000~037776	77771 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																
コ2000~コ7576	@コ2000~@コ7574																
b0000~b1776	@b0000~@b1774																
b2000~b3776	@b2000~@b3774																
09000~99776	@09000~@99774																
E0000~E7776	@E0000~@E7774																
77771 000000~037776	77771 @000000~@037774																
演算条件	レジスタDの内容が0の時、シフト入力②がONでシフト(OFF→ONの変化時に限定されない)																
	① ON		① OFF														
	演算前	演算後	演算前														
レジスタ	D-1の内容	0	同左														
	Dの内容	D1	D2														
	D+1の内容	同左	0														
フラグ	ノンキャリー 07354	1 (D1=0) 0 (D1≠0)	1 (D2=0) 0 (D2≠0)														
	エラー 07355	0	0														
	キャリー 07356	0 (D1=0) 1 (D1≠0)	0 (D2=0) 1 (D2≠0)														
	ゼロ 07357	0	0														

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- シフト入力②がONの間、毎スキャンサイクル演算を実行します。
- 演算を実行すると、シフトしたレジスタ(D-1又はD+1)の内容はクリアします。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)
- Dの内容が0でないとき、演算は実行しません。
- D-1又はD+1から0以外のデータがシフトした場合だけ、キャリーフラグ(07356)がONします。
- 【参考】 下記のF命令は働きが類似しています。
F-61、F-61w、F-61d、F-161、F-161w

(解説)



命 令	
STR	04014
STR	04015
F-61	09020

シフト入力04015がONの間、シフト方向指示入力04014の状態により、次のように1バイト単位でデータがシフトします。



- 演算前09020の内容が0でないとき、シフトしません。
- シフトしたレジスタ(09017または09021)の内容はクリアします。

入力条件	演算前		演算後		ぜ	キャリー	ノンキャリー												
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	07357	07356	07354
04014 ● 04015 ●	9017	9020	○	○	●	○	○	○	○	○	○	○	○	○	○	○	○	○	●
04014 ● 04015 ●	9017	9020	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○
04014 ○ 04015 ●	9017	9020	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○
04014 ○ 04015 ●	9017	9020	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●

• エラーフラグ(07355)は常にOFFとなります。

○ OFF ● ON

F-61w ASFR 非同期両方向シフトレジスタ(1ワード) (Asynchronous Shift Register)

シンボル		①シフト方向指示入力 ②シフト入力																																																													
機能	シフト方向指示入力①に従って、レジスタD-2、D-1 (①ON) または、レジスタD+2、D+3 (①OFF)の1ワードデータをレジスタD、D+1にシフトする。																																																														
演算内容	<ul style="list-style-type: none"> ●シフト方向指示入力ONのとき D-2、D-1→D、D+1 ●シフト方向指示入力OFFのとき D+2、D+3→D、D+1 																																																														
Dの使用範囲	コ0000~コ1574、@コ0000~@コ1574 コ2000~コ7574、@コ2000~@コ7574 b0000~b1774、@b0000~@b1774 b2000~b3774、@b2000~@b3774 09000~99774、@09000~@99774 E0000~E7774、@E0000~@E7774 <small>774#1</small> 000000~037774、 <small>774#1</small> @000000~@037774																																																														
演算条件	レジスタD、D+1の内容が0の時、シフト入力②がONでシフト(OFF→ONの変化時に限定されない)																																																														
	<table border="1"> <thead> <tr> <th></th> <th colspan="2">① ON</th> <th colspan="2">① OFF</th> <th colspan="2">① ON/OFF</th> </tr> <tr> <th></th> <th>演算前</th> <th>演算後</th> <th>演算前</th> <th>演算後</th> <th>演算前</th> <th>演算後</th> </tr> </thead> <tbody> <tr> <td rowspan="3">レジスタ</td> <td>D-2,D-1の内容</td> <td>D1</td> <td>0</td> <td>D1</td> <td>同左</td> <td>D1</td> <td>同左</td> </tr> <tr> <td>D,D+1の内容</td> <td>0</td> <td>D1</td> <td>0</td> <td>D2</td> <td>0以外</td> <td>同左</td> </tr> <tr> <td>D+2,D+3の内容</td> <td>D2</td> <td>同左</td> <td>D2</td> <td>0</td> <td>D2</td> <td>同左</td> </tr> <tr> <td rowspan="4">フラグ</td> <td>ノンキャリー 07354</td> <td>1 (D1=0) 0 (D1≠0)</td> <td></td> <td>1 (D2=0) 0 (D2≠0)</td> <td></td> <td>1</td> </tr> <tr> <td>エラー 07355</td> <td>0</td> <td></td> <td>0</td> <td></td> <td>0</td> </tr> <tr> <td>キャリー 07356</td> <td>0 (D1=0) 1 (D1≠0)</td> <td></td> <td>0 (D2=0) 1 (D2≠0)</td> <td></td> <td>0</td> </tr> <tr> <td>ゼロ 07357</td> <td>0</td> <td></td> <td>0</td> <td></td> <td>0</td> </tr> </tbody> </table>			① ON		① OFF		① ON/OFF			演算前	演算後	演算前	演算後	演算前	演算後	レジスタ	D-2,D-1の内容	D1	0	D1	同左	D1	同左	D,D+1の内容	0	D1	0	D2	0以外	同左	D+2,D+3の内容	D2	同左	D2	0	D2	同左	フラグ	ノンキャリー 07354	1 (D1=0) 0 (D1≠0)		1 (D2=0) 0 (D2≠0)		1	エラー 07355	0		0		0	キャリー 07356	0 (D1=0) 1 (D1≠0)		0 (D2=0) 1 (D2≠0)		0	ゼロ 07357	0		0		0
	① ON		① OFF		① ON/OFF																																																										
	演算前	演算後	演算前	演算後	演算前	演算後																																																									
レジスタ	D-2,D-1の内容	D1	0	D1	同左	D1	同左																																																								
	D,D+1の内容	0	D1	0	D2	0以外	同左																																																								
	D+2,D+3の内容	D2	同左	D2	0	D2	同左																																																								
フラグ	ノンキャリー 07354	1 (D1=0) 0 (D1≠0)		1 (D2=0) 0 (D2≠0)		1																																																									
	エラー 07355	0		0		0																																																									
	キャリー 07356	0 (D1=0) 1 (D1≠0)		0 (D2=0) 1 (D2≠0)		0																																																									
	ゼロ 07357	0		0		0																																																									

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
 - Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
 - 入力信号がONの間、毎スキャンサイクル演算を実行します。(9・7ページ「演算実行条件」参照)
- 参考 下記のF命令は働きが類似しています。
F-61、F-61w、F-61d、F-161、F-161w

(解説)

命令	
STR	00000
STR	00001
F-61w	09004

00000(①)ON……………09002、09003よりシフト
 00001(②)ON……………シフト指示
 09004、09005の内容 0000

入力条件が上記の場合、演算結果は次のようになります。

	演算前					演算後				
09000										09000
09001	1	2	3	4		1	2	3	4	09001
09002	5	6	7	8		0	0	0	0	09002
09003										09003
09004	0	0	0	0		5	6	7	8	09004
09005										09005
09006	9	8	7	6		9	8	7	6	09006
09007										09007
09010	5	4	3	2		5	4	3	2	09010
09011										09011

キャリーフラグ(07356)のみONします。

**F-61d
ASFR**

非同期両方向シフトレジスタ(2ワード)
(Asynchronous Shift Register)

シンボル	① F-61d ASFR D	①シフト方向指示入力 ②シフト入力					
機能	シフト方向指示入力①に従って、レジスタD-4~D-1(①ON)または、レジスタD+4~D+7(①OFF)の2ワードデータをレジスタD~D+3にシフトする。						
演算内容	<ul style="list-style-type: none"> ●シフト方向指示入力ONのとき D-4~D-1→D~D+3 ●シフト方向指示入力OFFのとき D+4~D+7→D~D+3 						
Dの使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 771#1 000000~037774 : 771#1 @000000~@037774						
演算条件	レジスタD~D+3の内容が0の時、シフト入力②がONでシフト(OFF→ONの変化時に限定されない)						
	① ON	① OFF	① ON/OFF				
	演算前	演算後	演算前	演算後	演算前	演算後	
レジスタ	D-4~D-1の内容	D1	0	D1	同左	D1	同左
	D~D+3の内容	0	D1	0	D2	0以外	同左
	D+4~D+7の内容	D2	同左	D2	0	D2	同左
フラグ	ノンキャリー 07354	1 (D1=0) 0 (D1≠0)		1 (D2=0) 0 (D2≠0)		1	
	エラー 07355	0		0		0	
	キャリー 07356	0 (D1=0) 1 (D1≠0)		0 (D2=0) 1 (D2≠0)		0	
	ゼロ 07357	0		0		0	

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- 入力信号がONの間、毎スキャンサイクル演算を実行します。(9・7ページ「演算実行条件」参照)

参考 下記のF命令は働きが類似しています。
F-61、F-61w、F-61d、F-161、F-161w

(解説)

00000	F-61d ASFR	09004
-------	------------	-------

命 令	
STR	00000
STR	00001
F-61d	09004

00000(①)ON.....09000~09003よりシフト
00001(②)ON.....シフト指示
09004、09005の内容0000

入力条件が上記の場合、演算結果は次のようになります。

	演算前	演算後	
09000	1 2 3 4	0 0 0 0	09000
09001			09001
09002	5 6 7 8	0 0 0 0	09002
09003			09003
09004	0 0 0 0	1 2 3 4	09004
09005			09005
09006	0 0 0 0	5 6 7 8	09006
09007			09007
09010	9 8 7 6	9 8 7 6	09010
09011			09011
09012	5 4 3 2	5 4 3 2	09012
09013			09013

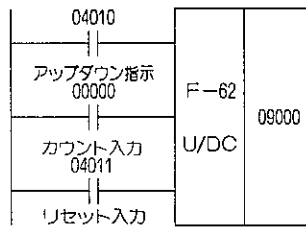
キャリーフラグ(07356)のみONします。

**F-62
U/DC**

**BCD2桁のアップ・ダウンカウンタ
(Up/Down Counter)**

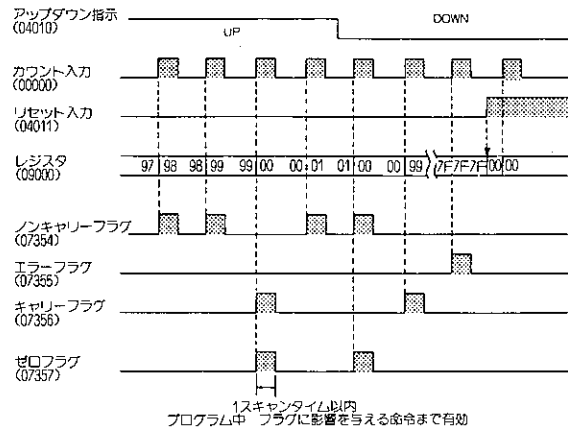
シンボル		①アップ・ダウン指示入力 ②カウント入力 ③リセット入力																																																	
機能	アップ・ダウン指示入力①に従ってレジスタDの内容(BCD2桁)を加算(①ON)または減算(①OFF)する。																																																		
演算内容	アップダウン指示入力①ONのとき $\langle D \rangle + 1 \rightarrow D$ アップダウン指示入力①OFFのとき $\langle D \rangle - 1 \rightarrow D$																																																		
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 7F401 000000~037777 7F401 @000000~@037774																																																		
演算条件	リセット入力③がOFFの時、カウント入力②の立上り(OFF→ON)																																																		
演算後	Dの内容	<ul style="list-style-type: none"> ●リセット入力③がOFFの時 演算結果(BCDコード) ●リセット入力③がONの時 全ビットOFF 																																																	
	フラグ	<table border="1"> <thead> <tr> <th>アップ・ダウン指示入力①</th> <th>演算結果</th> <th>ゼロ 07357</th> <th>キャリー 07356</th> <th>エラー 07355</th> <th>ノキャリー 07354</th> </tr> </thead> <tbody> <tr> <td rowspan="3">ON</td> <td>99+1 →00</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>00~98+1 →01~99</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>BCD以外の数値</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td rowspan="3">OFF</td> <td>00-1 →99</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>01-1 →00</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>02~99-1 →01~98</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td></td> <td>BCD以外の数値</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td></td> <td>リセット入力③ONの時</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	アップ・ダウン指示入力①	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354	ON	99+1 →00	1	1	0	0	00~98+1 →01~99	0	0	0	1	BCD以外の数値	0	0	1	0	OFF	00-1 →99	0	1	0	0	01-1 →00	1	0	0	1	02~99-1 →01~98	0	0	0	1		BCD以外の数値	0	0	1	0		リセット入力③ONの時	0	0	0
アップ・ダウン指示入力①	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354																																														
ON	99+1 →00	1	1	0	0																																														
	00~98+1 →01~99	0	0	0	1																																														
	BCD以外の数値	0	0	1	0																																														
OFF	00-1 →99	0	1	0	0																																														
	01-1 →00	1	0	0	1																																														
	02~99-1 →01~98	0	0	0	1																																														
	BCD以外の数値	0	0	1	0																																														
	リセット入力③ONの時	0	0	0	0																																														

(解説)



命令	
STR	04010
STR	00000
STR	04011
F-62	09000

リセット入力04011がOFFで計数可能となります。(ONリセットに設定時)
アップダウン指示入力04010がONの時加算、OFFの時減算カウンタとして動作します。他の命令により09000の内容がBCD以外のコードになったとき、エラーフラグ(07355)がONし、カウント動作は実行しません。(例では7F)



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- リセット入力③はシステムメモリ(#202)の設定により「OFF」でリセットすることもできます。

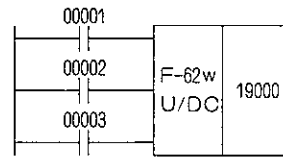
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

F-62w
U/DC

BCD 4桁のアップ・ダウンカウンタ (Up/Down Counter)

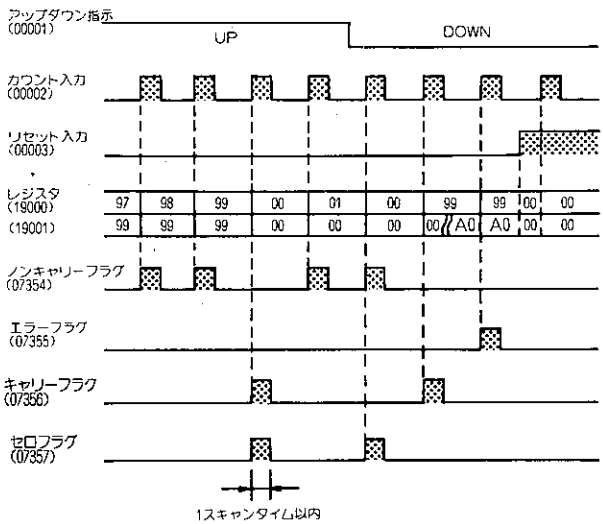
シンボル			①アップ・ダウン指示入力 ②カウント入力 ③リセット入力					
機能	アップダウン指示入力①に従ってレジスタD、D+1の内容(BCD4桁)を加算(①ON)または減算(①OFF)する。							
演算内容	アップダウン指示入力①ONのとき $\langle D, D+1 \rangle + 1 \rightarrow D, D+1$ アップダウン指示入力①OFFのとき $\langle D, D+1 \rangle - 1 \rightarrow D, D+1$							
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 77771 000000~037776 77771 @000000~@037774							
演算条件	リセット入力③がOFFのとき、カウント入力②の立上り(OFF→ON)							
演算後	Dの内容	演算結果(下2桁)	リセット入力③ONの時、全ビットOFF					
	D+1の内容	演算結果(上2桁)						
	フラグ	アップ・ダウン指示入力①	ON	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
			OFF	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
		ON	9999+1	1	1	0	0	
			0000~9998+1	0	0	0	1	
			BCD以外の数値	0	0	1	0	
OFF	0000-1	0	1	0	0			
	0001-1	1	0	0	1			
	0002~9999-1	0	0	0	1			
BCD以外の数値	0	0	1	0				
リセット入力③ONの時		0	0	0	0			

〔解説〕



命令	
STR	00001
STR	00002
STR	00003
F-62w	19000

リセット入力00003がOFFで計数可能となります。
(ONリセットに設定時)
アップダウン指示入力00001がONの時加算、OFFの時減算カウンタとして動作します。レジスタ19000または19001の内容がBCD以外のコードの時、エラーフラグ(07355)がONし、カウント動作は実行しません。



1スキャンタイム以内

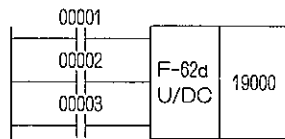
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)
- リセット入力③はシステムメモリ(#202)の設定により「OFF」でリセットすることもできます。

F-62d
U/DC

BCD8桁のアップ・ダウンカウンタ (Up/Down Counter)

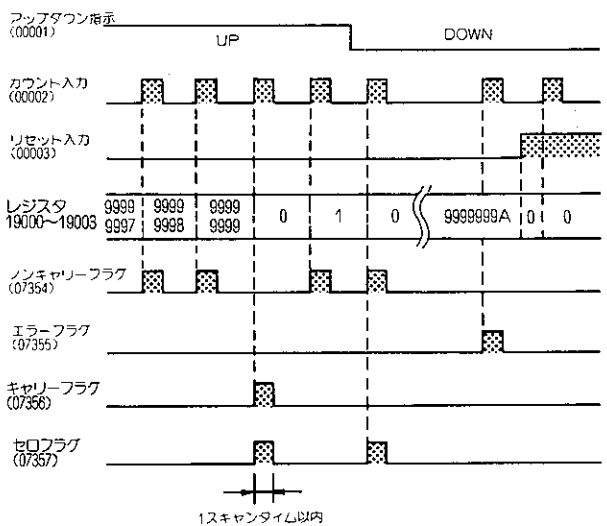
シンボル	① ② ③		①アップ・ダウン指示入力 ②カウント入力 ③リセット入力														
機能	アップダウン指示入力①に従ってレジスタD~D+3の内容(BCD8桁)を加算(①ON)または減算(①OFF)する。																
演算内容	アップダウン指示入力①ONのとき $\langle D \sim D+3 \rangle + 1 \rightarrow D \sim D+3$ アップダウン指示入力①OFFのとき $\langle D \sim D+3 \rangle - 1 \rightarrow D \sim D+3$																
Dの使用範囲	<table border="0"> <tr><td>コ0000~コ1574</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7574</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1774</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3774</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99774</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7774</td><td>@E0000~@E7774</td></tr> <tr><td>771#1 000000~037774</td><td>771#1 @000000~@037774</td></tr> </table>			コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	771#1 000000~037774	771#1 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																
コ2000~コ7574	@コ2000~@コ7574																
b0000~b1774	@b0000~@b1774																
b2000~b3774	@b2000~@b3774																
09000~99774	@09000~@99774																
E0000~E7774	@E0000~@E7774																
771#1 000000~037774	771#1 @000000~@037774																
演算条件	リセット入力③がOFFのとき、カウント入力②の立上り(OFF→ON)																
演算後	D~D+3の内容	演算結果(8桁)	リセット入力③ONの時、全ビットOFF														
	アップ ON	アップ・ダウン指示入力①	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354										
		99999999+1	1	1	0	0											
		00000000~99999998+1	0	0	0	1											
	BCD以外の数値	0	0	1	0												
	ダウン OFF	アップ・ダウン指示入力①	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354										
		00000000-1	0	1	0	0											
00000001-1		1	0	0	1												
00000002~99999999-1	0	0	0	1													
BCD以外の数値	0	0	1	0													
リセット入力③ONの時	0	0	0	0													

(解説)



命令	
STR	00001
STR	00002
STR	00003
F-62d	19000

リセット入力00003がOFFで計数可能となります。(ON/リセットに設定時)
アップダウン指示入力00001がONの時加算、OFFの時減算カウンタとして動作します。レジスタ19000~19003の内容がBCD以外のコードの時、エラーフラグ(07355)がONし、カウント動作は実行しません。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)
- D~D+3の内容がBCDコード以外の場合、エラーフラグ(07355)がONし、演算は実行しません。(D~D+3の内容は不変)
- リセット入力③はシステムメモリ(#202)の設定により「OFF」でリセットすることもできます。

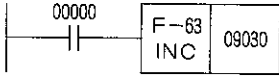
【参考】 下記のF命令は働きが類似しています。
F-62、F-62w、F-62d、F-65、F-65w、F-66、F-66w

**F-63
INC**

**加算カウンタ(1バイト)
(INCRement)**

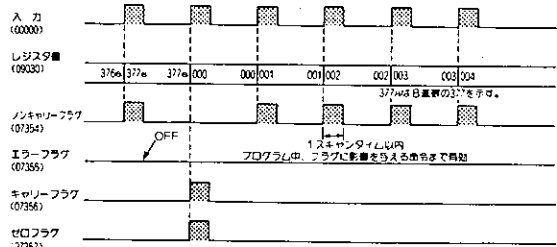
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-63 INC</td><td style="padding: 2px;">D</td></tr></table>		F-63 INC	D																		
F-63 INC	D																					
機能	レジスタDの内容(バイナリーデータ)を加算カウントする。																					
演算内容	$\langle D \rangle + 1 \rightarrow D$																					
Dの使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~09777 : @09000~@09774 E0000~E7777 : @E0000~@E7774 ファイル1 000000~037777 : ファイル1 @000000~@037774																					
演算条件	入力信号の立上り(OFF→ON)																					
演算後	Dの内容	演算結果 (バイナリーコード)																				
	フラグ	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>演算結果</td> <td>ゼロ</td> <td>キャリー</td> <td>エラー</td> <td>ノキャリー</td> </tr> <tr> <td>377→000 (注2)</td> <td>07357</td> <td>07356</td> <td>07355</td> <td>07354</td> </tr> <tr> <td>上記以外</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td></td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> </table>	演算結果	ゼロ	キャリー	エラー	ノキャリー	377→000 (注2)	07357	07356	07355	07354	上記以外	1	1	0	0		0	0	0	1
		演算結果	ゼロ	キャリー	エラー	ノキャリー																
377→000 (注2)	07357	07356	07355	07354																		
上記以外	1	1	0	0																		
	0	0	0	1																		

(解説)



命 令	
STR	00000
F-63	09030

入力条件00000のOFF→ONを検知して、加算カウントします。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dの内容はバイナリーコードです。10進表現では000~255、8進表現では000~377(8)と見なせます。

- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

参考 下記のF命令は働きが類似しています。
F-63、F-63w、F-163、F-163w、F-263、F-263w

F-63w 加算カウンタ(1ワード)
INC (INCrement)

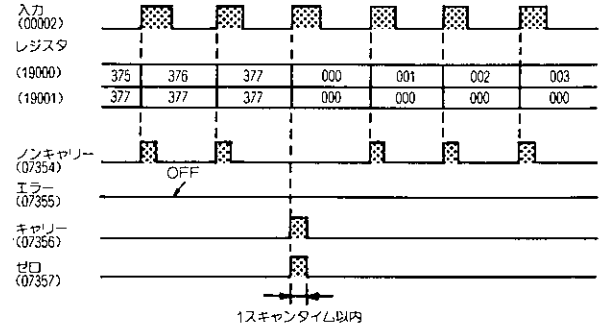
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-63w</td><td style="padding: 2px;">INC</td><td style="padding: 2px;">D</td></tr></table>					F-63w	INC	D
F-63w	INC	D						
機能	レジスタD、D+1の内容(バイナリデータ)を加算カウントする。							
演算内容	$\langle D, D+1 \rangle + 1 \rightarrow D, D+1$							
Dの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 77776~77776 : @77776~@77774							
演算条件	入力信号の立上り(OFF→ON)							
演算後	Dの内容	演算結果(下位)						
	D+1の内容	演算結果(上位)						
	フラグ	演算結果(8進)	ゼロ	キャリー	エラー	ノンキャリー		
	17777-00000	1	1	0	0			
	上記以外	0	0	0	1			

(解説)



命 令	
STR	00002
F-63w	19000

入力条件00002がOFF→ONの変化を検知して加算カウントします。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

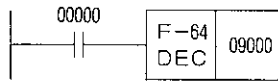
参考 下記のF命令は働きが類似しています。
 F-63、F-63w、F-163、F-163w、F-263、F-263w

**F-64
DEC**

**減算カウンタ(1バイト)
(DECrement)**

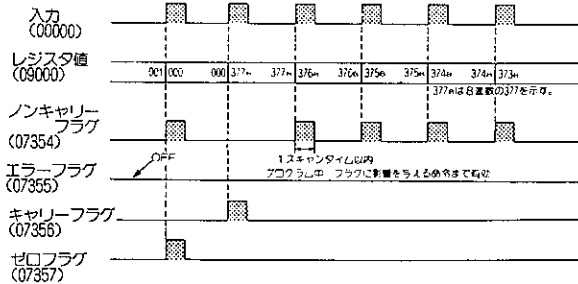
シンボル	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-64 DEC</td> <td style="padding: 2px;">D</td> </tr> </table>					F-64 DEC	D
F-64 DEC	D						
機能	レジスタDの内容(バイナリーデータ)を減算カウントする。						
演算内容	$\langle D \rangle - 1 \rightarrow D$						
Dの使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 ファイル1 000000~037777 : ファイル1 @000000~@037774						
演算条件	入力信号の立上り(OFF→ON)						
演算後	Dの内容	演算結果(バイナリーコード)					
	フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354	
		001 _(e) →000 _(e)	1	0	0	1	
		000 _(e) →377 _(e)	0	1	0	0	
上記以外	0	0	0	1			

(解説)



命令	
STR	00000
F-64	09000

入力条件00000のOFF→ONの変化を検知して減算カウントします。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dの内容はバイナリーコードです。10進表現では000~255、8進表現では000~377_(e)と見なせます。

- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

参考 下記のF命令は働きが類似しています。
F-64、F-64w、F-164、F-164w、F-264、F-264w

F-64w
DEC

減算カウンタ(1ワード) (DECrement)

シンボル					
機能	レジスタD、D+1の内容(バイナリデータ)を減算カウントする。				
演算内容	$\langle D, D+1 \rangle - 1 \rightarrow D, D+1$				
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 ファイル000000~037776 ファイル@000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	Dの内容	演算結果(下位)			
	D+1の内容	演算結果(上位)			
フラグ	演算結果(8進)	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
	000001-000000	1	0	0	1
	000000-177777	0	1	0	0
	上記以外	0	0	0	1

(解説)

命令	
STR	00002
F-64w	19000

入力条件00002がOFF→ONの変化を検知して減算カウントします。

1スキャンタイム以内

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

参考 下記のF命令は働きが類似しています。
F-64、F-64w、F-164、F-164w、F-264、F-264w

F-65 BCDI BCD加算カウンタ(1バイト)
(BCD Increment)

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-65 BCDI</td><td>D</td></tr></table>		F-65 BCDI	D																							
F-65 BCDI	D																										
機能	レジスタDの内容(BCDデータ)を加算カウントする。																										
演算内容	$\langle D \rangle + 1 \rightarrow D$																										
Dの使用範囲	<table border="0"> <tr><td>コ0000~コ1577</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7577</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1777</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3777</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99777</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7777</td><td>@E0000~@E7774</td></tr> <tr><td>771M1 000000~037777</td><td>771M1 @000000~@037774</td></tr> </table>		コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	771M1 000000~037777	771M1 @000000~@037774											
コ0000~コ1577	@コ0000~@コ1574																										
コ2000~コ7577	@コ2000~@コ7574																										
b0000~b1777	@b0000~@b1774																										
b2000~b3777	@b2000~@b3774																										
09000~99777	@09000~@99774																										
E0000~E7777	@E0000~@E7774																										
771M1 000000~037777	771M1 @000000~@037774																										
演算条件	入力信号の立上り(OFF→ON)																										
演算後	Dの内容	<table border="1"> <tr> <th>演算結果 (BCDコード)</th> <th colspan="4">レジスタDがBCDコードでない時不変</th> </tr> <tr> <td>演算結果</td> <td>ゼロ 07357</td> <td>キャリー 07356</td> <td>エラー 07355</td> <td>ノンキャリー 07354</td> </tr> <tr> <td>99→00(注2)</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>上記以外</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>BCDコード以外</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> </table>	演算結果 (BCDコード)	レジスタDがBCDコードでない時不変				演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354	99→00(注2)	1	1	0	0	上記以外	0	0	0	1	BCDコード以外	0	0	1	0
	演算結果 (BCDコード)	レジスタDがBCDコードでない時不変																									
	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354																						
99→00(注2)	1	1	0	0																							
上記以外	0	0	0	1																							
BCDコード以外	0	0	1	0																							
フラグ	<table border="1"> <tr> <th>演算結果</th> <th>ゼロ 07357</th> <th>キャリー 07356</th> <th>エラー 07355</th> <th>ノンキャリー 07354</th> </tr> <tr> <td>99→00(注2)</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>上記以外</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>BCDコード以外</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> </table>		演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354	99→00(注2)	1	1	0	0	上記以外	0	0	0	1	BCDコード以外	0	0	1	0					
演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354																							
99→00(注2)	1	1	0	0																							
上記以外	0	0	0	1																							
BCDコード以外	0	0	1	0																							
<p>(解説)</p> <p>入力条件00000がOFF→ONの変化を検知してレジスタ09000の内容を加算カウント(+1)します。</p>																											
<table border="1" style="float: right;"> <tr><th colspan="2">命令</th></tr> <tr><td>STR</td><td>00000</td></tr> <tr><td>F-65</td><td>09000</td></tr> </table>			命令		STR	00000	F-65	09000																			
命令																											
STR	00000																										
F-65	09000																										

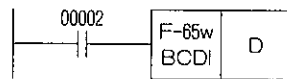
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dの内容はBCDコードです。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- Dの内容がBCDコード以外の場合、エラーフラグ(07355)がONし、演算は実行しません。(Dの内容は不変)

参考 下記のF命令は働きが類似しています。
F-62、F-62w、F-62d、F-65、F-65w

F-65w BCDI BCD加算カウンタ(1ワード)
(BCD Increment)

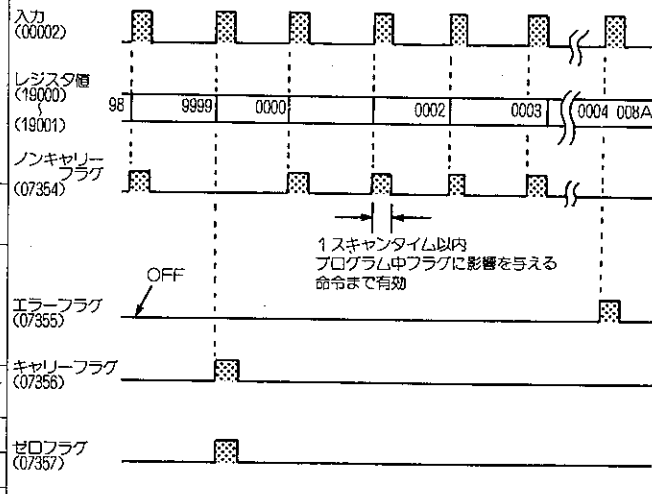
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-65w BCDI</td><td style="padding: 2px;">D</td></tr></table>			F-65w BCDI	D
F-65w BCDI	D				
機能	レジスタD、D+1の内容(BCDデータ)を加算カウントする。				
演算内容	$\langle D, D+1 \rangle + 1 \rightarrow D, D+1$				
Dの使用範囲	コ0000~コ1576: @コ0000~@コ1574 コ2000~コ7576: @コ2000~@コ7574 b0000~b1776: @b0000~@b1774 b2000~b3776: @b2000~@b3774 09000~99776: @09000~@99774 E0000~E7776: @E0000~@E7774 774#1 000000~037776: 774#1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	Dの内容	演算結果(下位)	レジスタD、D+1の内容がBCDコードでない時不変		
	D+1の内容	演算結果(上位)			
	フラグ	演算結果 BCD	ゼロ 07357	キャリー 07356	エラー 07355
	9999→0000	1	1	0	0
	上記以外	0	0	0	1
	BCDコード以外	0	0	1	0

(解説)



命令	
STR	00002
F-65w	19000

入力条件00002がOFF→ONの変化を検知してレジスタ19000、19001の内容を加算カウントします。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- D、D+1の内容がBCDコード以外の場合エラーフラグ(07355)がONし、演算は実行しません。(D、D+1の内容は不変)

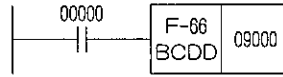
参考 下記のF命令は働きが類似しています。
F-62、F-62w、F-62d、F-65、F-65w

**F-66
BCDD**

**BCD減算カウンタ(1バイト)
(BCD Decrement)**

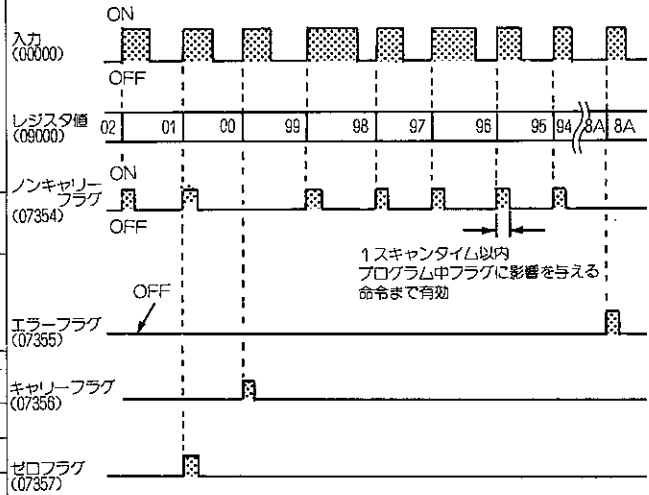
シンボル	F-66 BCDD D					
機能	レジスタDの内容(BCDデータ)を減算カウントする。					
演算内容	$\langle D \rangle - 1 \rightarrow D$					
Dの使用範囲	コ0000~コ1577 コ2000~コ7577 b0000~b1777 b2000~b3777 09000~99777 E0000~E7777 77771 000000~037777	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 77771 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)					
演算後	Dの内容	演算結果 (BCDコード)	レジスタDの内容がBCDコードでないとき不変			
	フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
		01→00	1	0	0	1
		00→99	0	1	0	0
		上記以外	0	0	0	1
BCDコード以外	0	0	1	0		

(解説)



命令	
STR	00000
F-66	09000

入力条件00000がOFF→ONの変化時にレジスタ09000の内容を減算カウント(-1)します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dの内容はBCDコードです。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- Dの内容がBCDコード以外の場合エラーフラグ(07355)がONし、演算は実行しません。(Dの内容は不変)

参考 下記のF命令は働きが類似しています。
F-62、F-62w、F-62d、F-66、F-66w

F-66w
BCDD

BCD減算カウンタ(1ワード)
(BCD Decrement)

シンボル	F-66w BCDD		D		
機能	レジスタD、D+1の内容(BCDデータ)を減算カウントする。				
演算内容	$\langle D, D+1 \rangle - 1 \rightarrow D, D+1$				
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 771#1 000000~037776 771#1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	Dの内容	演算結果(下位)		レジスタD、D+1の内容がBCDコードでない時不変	
	D+1の内容	演算結果(上位)			
フラグ	演算結果(8進)	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
	0001→0000	1	0	0	1
	0000→9999	0	1	0	0
	上記以外	0	0	0	1
	BCDコード以外	0	0	1	0

(解説)

00002

F-66w
BCDD

19000

命令	
STR	00002
F-66w	19000

入力条件00002がOFF→ONの変化を検知して減算カウントします。

1スキャンタイム以内プログラム中フラグに影響を与える命令まで有効

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- D、D+1の内容がBCDコード以外の場合エラーフラグ(07355)がONし、演算は実行しません。(Dの内容は不変)

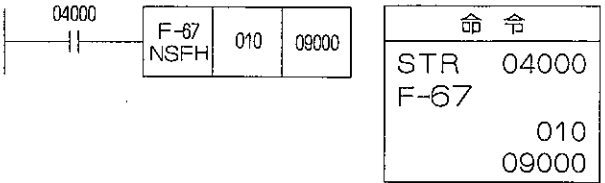
参考 下記のF命令は働きが類似しています。
F-62、F-62w、F-62d、F-66、F-66w

**F-67
NSFH**

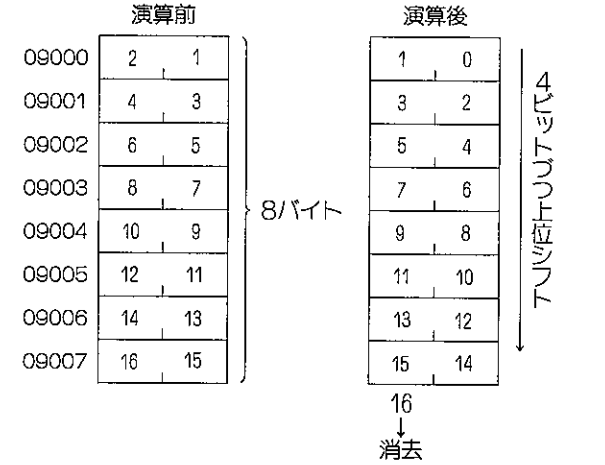
桁シフト(上位)

シンボル	<table border="1" style="display: inline-table;"> <tr> <td style="text-align: center;">F-67 NSFH</td> <td style="text-align: center;">n</td> <td style="text-align: center;">D</td> </tr> </table>			F-67 NSFH	n	D
F-67 NSFH	n	D				
機能	レジスタDを先頭とするnバイトのデータを上位に4ビットづつアドレスの大きい方にシフトする。					
演算内容	D~D+n-1を上位へ4ビットづつシフト					
nの使用範囲	000~377 (OCT) (000とすると256バイトとなる)					
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774					
演算条件	入力信号の立上り					
演算後	Dの内容	演算結果(シフト結果)				
	フラグ	不変				

(解説)



入力条件04000がOFF→ONの変化時に、レジスタ09000~09007の8バイトのデータを4ビットづつ上位へシフトします。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 演算後、先頭レジスタDの低位4ビットに0を格納します。
また、シフト後nバイトのレジスタの上位4ビットのデータは消去(クリア)します。

**F-68
NSFL**

桁シフト(下位)

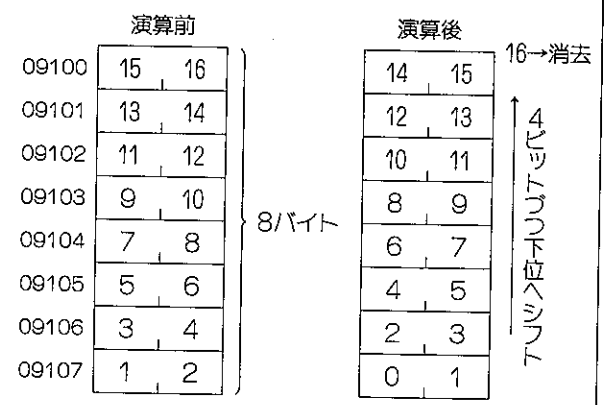
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-68 NSFL</td><td style="padding: 2px;">n</td><td style="padding: 2px;">D</td></tr></table>			F-68 NSFL	n	D											
F-68 NSFL	n	D															
機能	レジスタDを先頭とするnバイトのデータを下位に4ビットづつアドレスの小さい方にシフトする。																
演算内容	D~D+n-1を下位へ4ビットづつシフト																
nの使用範囲	000~377 (OCT) (000とすると256バイトとなる)																
Dの使用範囲	<table style="font-family: monospace; font-size: small;"> <tr><td>コ0000~コ1577</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7577</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1777</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3777</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99777</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7777</td><td>@E0000~@E7774</td></tr> <tr><td>771#1 000000~037777</td><td>771#1 @000000~@037774</td></tr> </table>			コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	771#1 000000~037777	771#1 @000000~@037774
コ0000~コ1577	@コ0000~@コ1574																
コ2000~コ7577	@コ2000~@コ7574																
b0000~b1777	@b0000~@b1774																
b2000~b3777	@b2000~@b3774																
09000~99777	@09000~@99774																
E0000~E7777	@E0000~@E7774																
771#1 000000~037777	771#1 @000000~@037774																
演算条件	入力信号の立上り																
演算後	Dの内容	演算結果(シフト結果)															
	フラグ	不変															

〔解説〕

04010
—
F-68 NSFL
010
09100

命 令	
STR	04010
F-67	010
	09100

入力条件04010がOFF→ONの変化時に、レジスタ09100~09107の8バイトのデータを4ビットづつ下位へシフトします。



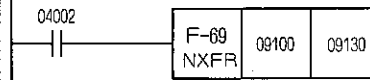
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 演算後、D+n-1のレジスタの上位4ビットに0を格納します。
また、シフト後Dのレジスタの下位4ビットのデータは消去(クリア)します。

**F-69
NXFR**

桁転送

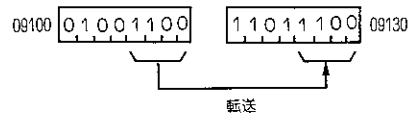
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-69 NXFR</td> <td style="padding: 2px;">S</td> <td style="padding: 2px;">D</td> </tr> </table>	F-69 NXFR	S	D
F-69 NXFR	S	D		
機能	レジスタSの下位4ビットをレジスタDの下位4ビットに転送する。			
演算内容	Sの下位4ビット→Dの下位4ビット			
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 ファイル 000000~037777 ファイル @000000~@037774			
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 ファイル 000000~037777 ファイル @000000~@037774			
演算条件	入力信号の立上り			
演算後	Sの内容	不変		
	Dの内容	レジスタSの下位4ビットの内容 上位は不変		
	フラグ	不変		

(解説)



命 令	
STR	04002
F-69	09100
	09130

入力条件04002がOFF→ONの変化時に、レジスタ09100の下位4ビットの内容をレジスタ09130の下位4ビットに転送します。



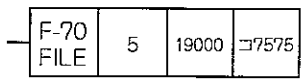
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

F-70 FILE

nバイト一括転送 (FILE)

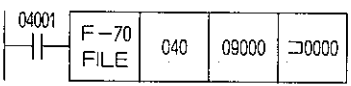
シンボル	F-70 FILE n S D			
機能	レジスタSからS+n-1までのnバイト(8進数)のデータをレジスタDからD+n-1までのnバイトに一括転送する。			
演算内容	S, ..., S+n-1 → D, ..., D+n-1			
nの使用範囲	000~377(8) (000とすると256/バイトとなる)			
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 7777M1 000000~037777 7777M1 @000000~@037774			
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 7777M1 000000~037777 7777M1 @000000~@037774			
演算条件	入力信号の立上り(OFF→ON)			
演算後	S...S+n-1の内容	不変		
	Dの内容	レジスタSの内容		
	D+1の内容	レジスタS+1の内容		
	D+n-1の内容	レジスタS+n-1の内容		
フラグ	不変			

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- ファイル0のアドレス001600~001777と035600以降に転送しないようにしてください。



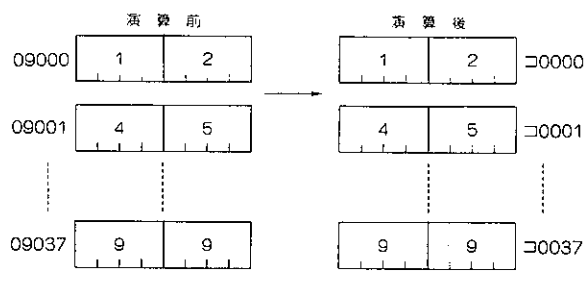
上記のようにプログラムすると035600と035601に19003と19004の内容が転送されてしまいます。
(9・3ページ「ソースとデスティネーション」参照)

(解説)

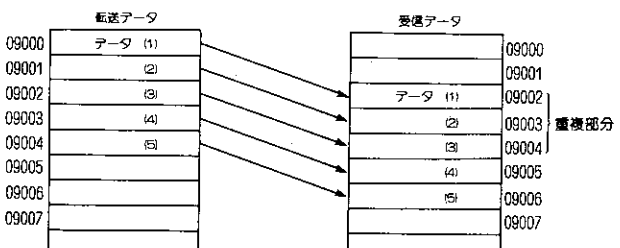
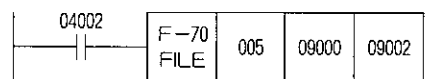


命令	
STR	04001
F-70	040
	09000
	コ0000

入力条件04001がOFF→ONの変化時に、レジスタ09000から09037までの040(8/バイト(10進数で32/バイト))のデータをレジスタコ0000からコ0037までの32/バイトに一括転送します。レジスタ09000から09037までの内容は不変です。



- 転送元、転送先が重複するようなn、S、Dの設定もできます。



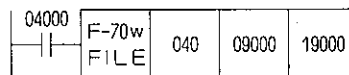
参考 下記のF命令は働きが類似しています。
F-00、F-00w、F-00d、F-70、F-70w、F-74、F-74w、F-76、F-76w

F-70w
FILE

nワード一括転送
(FILE)

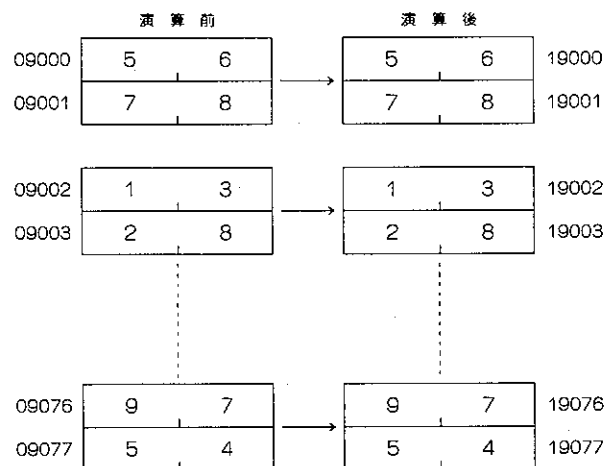
シンボル	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">F-70w</td> <td style="text-align: center;">n</td> <td style="text-align: center;">S</td> <td style="text-align: center;">D</td> </tr> <tr> <td style="text-align: center;">FILE</td> <td></td> <td></td> <td></td> </tr> </table>				F-70w	n	S	D	FILE			
F-70w	n	S	D									
FILE												
機能	レジスタSからS+2n-1までのnワードのデータをレジスタDからD+2n-1までのnワードに一括転送する。											
演算内容	S、S+1、…S+2n-1→ D、D+1、…D+2n-1											
nの使用範囲	000~377(8) (000とすると256ワードとなる)											
Sの使用範囲	コ0000~コ1576 ; @コ0000~@コ1574 コ2000~コ7576 ; @コ2000~@コ7574 b0000~b1776 ; @b0000~@b1774 b2000~b3776 ; @b2000~@b3774 09000~99776 ; @09000~@99774 E0000~E7776 ; @E0000~@E7774 774#1 000000~037776 ; 774#1 @000000~@037774											
Dの使用範囲	コ0000~コ1576 ; @コ0000~@コ1574 コ2000~コ7576 ; @コ2000~@コ7574 b0000~b1776 ; @b0000~@b1774 b2000~b3776 ; @b2000~@b3774 09000~99776 ; @09000~@99774 E0000~E7776 ; @E0000~@E7774 774#1 000000~037776 ; 774#1 @000000~@037774											
演算条件	入力信号の立上り(OFF→ON)											
演算後	S、…S+2n-1の内容	不変										
	Dの内容	レジスタSの内容										
	D+1の内容	レジスタS+1の内容										
	D+2n-1の内容	レジスタS+2n-1の内容										
フラグ	不変											

(解説)



命令	
STR	04000
F-70w	040
	09000
	19000

入力条件04000がOFF→ONの変化時に、レジスタ09000から9077までの040₈ワード（10進数で32ワード）のデータをレジスタ19000から19077までの32ワードに一括転送します。レジスタ09000から09077までの内容は不変です。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

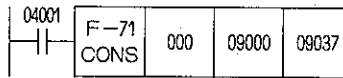
参考 下記のF命令は働きが類似しています。
F-00、F-00w、F-00d、F-70、F-70w、F-74、
F-74w、F-76、F-76w

**F-71
CONS**

**8進定数(1バイト)一括転送
(CONStant)**

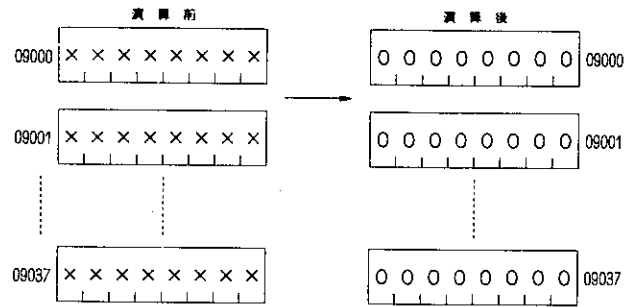
シンボル	$\overline{\text{F-71}} \text{ CONS } n \text{ D}_1 \text{ D}_2$														
機能	レジスタD ₁ からレジスタD ₂ に8進定数nを一括転送する。														
演算内容	$n \rightarrow D_1, \dots, D_2$														
nの使用範囲	000~377(8)														
D ₁ の使用範囲	<table border="0"> <tr><td>コ0000~コ1577</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7577</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1777</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3777</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99777</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7777</td><td>@E0000~@E7774</td></tr> <tr><td>ファイル1 000000~037777</td><td>ファイル1 @000000~@037774</td></tr> </table>	コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	ファイル1 000000~037777	ファイル1 @000000~@037774
コ0000~コ1577	@コ0000~@コ1574														
コ2000~コ7577	@コ2000~@コ7574														
b0000~b1777	@b0000~@b1774														
b2000~b3777	@b2000~@b3774														
09000~99777	@09000~@99774														
E0000~E7777	@E0000~@E7774														
ファイル1 000000~037777	ファイル1 @000000~@037774														
D ₂ の使用範囲	<table border="0"> <tr><td>コ0000~コ1577</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7577</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1777</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3777</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99777</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7777</td><td>@E0000~@E7774</td></tr> <tr><td>ファイル1 000000~037777</td><td>ファイル1 @000000~@037774</td></tr> </table>	コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	ファイル1 000000~037777	ファイル1 @000000~@037774
コ0000~コ1577	@コ0000~@コ1574														
コ2000~コ7577	@コ2000~@コ7574														
b0000~b1777	@b0000~@b1774														
b2000~b3777	@b2000~@b3774														
09000~99777	@09000~@99774														
E0000~E7777	@E0000~@E7774														
ファイル1 000000~037777	ファイル1 @000000~@037774														
演算条件	入力信号の立上り(OFF→ON)														
演算後	D ₁ の内容 D ₁ +1の内容 : D ₂ -1の内容 D ₂ の内容	定数 n													
	フラグ	不変													

(解説)



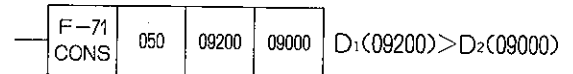
命令	
STR	04001
F-71	000
	09000
	09037

入力条件04001がOFF→ONの変化時に、レジスタ09000から09037に8進定数000を一括転送します。



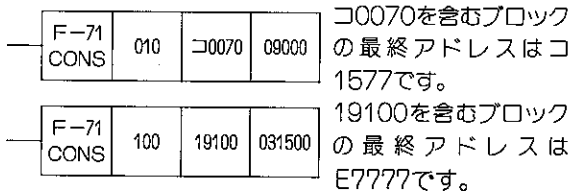
- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 次表の①~⑥のブロックをまたがるようなD₁、D₂の設定をすると演算を実行しません。

●D₁ > D₂となるアドレスを設定すると、演算しません。



ブロック	範囲
①	コ0000 ~ コ1577
②	コ2000 ~ コ7577
③	b0000 ~ b1777
④	b2000 ~ b3777
⑤	09000 ~ E7777
⑥	ファイル1 000000 ~ 037777

参考 下記のF命令は働きが類似しています。
F-08、F-08w、F-71、F-71w

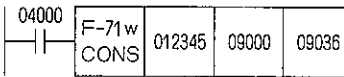


**F-71w
CONS**

**8進定数(1ワード)一括転送
(CONSTant)**

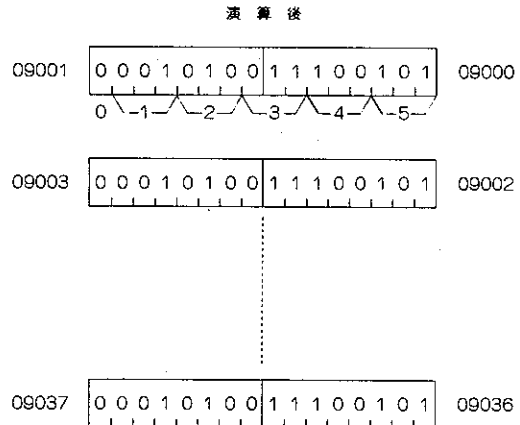
シンボル	F-71w CONS	n	D_1	D_2
機能	レジスタ D_1 、 D_1+1 からレジスタ D_2 、 D_2+1 に8進定数 n を一括転送する。			
演算内容	$n \rightarrow (D_1, D_1+1), \dots, (D_2, D_2+1)$			
n の使用範囲	000000~177777(8)			
D_1 の使用範囲	$\text{C}0000 \sim \text{C}1576$ $\text{@C}0000 \sim \text{@C}1574$ $\text{C}2000 \sim \text{C}7576$ $\text{@C}2000 \sim \text{@C}7574$ $\text{b}0000 \sim \text{b}1776$ $\text{@b}0000 \sim \text{@b}1774$ $\text{b}2000 \sim \text{b}3776$ $\text{@b}2000 \sim \text{@b}3774$ $09000 \sim 99776$ $\text{@}09000 \sim \text{@}99774$ $\text{E}0000 \sim \text{E}7776$ $\text{@E}0000 \sim \text{@E}7774$ 77776 77776 $\text{@}00000 \sim \text{@}03774$			
D_2 の使用範囲	$\text{C}0000 \sim \text{C}1576$ $\text{@C}0000 \sim \text{@C}1574$ $\text{C}2000 \sim \text{C}7576$ $\text{@C}2000 \sim \text{@C}7574$ $\text{b}0000 \sim \text{b}1776$ $\text{@b}0000 \sim \text{@b}1774$ $\text{b}2000 \sim \text{b}3776$ $\text{@b}2000 \sim \text{@b}3774$ $09000 \sim 99776$ $\text{@}09000 \sim \text{@}99774$ $\text{E}0000 \sim \text{E}7776$ $\text{@E}0000 \sim \text{@E}7774$ 77776 77776 $\text{@}00000 \sim \text{@}03774$			
演算条件	入力信号の立上がり(OFF→ON)			
演算後	D_1, D_1+1 D_1+2, D_1+3 ⋮ D_2-2, D_2-1 D_2, D_2+1	定数 n		
	フラグ	不変		

(解説)



命令	
STR	04000
F-71w	012345 09000 09036

入力条件04000がOFF→ONの変化時、レジスタ09000、09001から、09036、09037に8進定数012345を一括転送します。



- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(C0011、19003等は禁止)

参考 下記のF命令は働きが類似しています。
F-08、F-08w、F-71、F-71w

**F-72
DMPX**

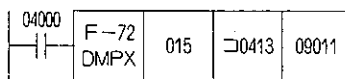
**ファイル1のレジスタへのnバイト分配
(DeMultiPleXer)**

(この命令はJW-31CUH/H1ではプログラムできません。)

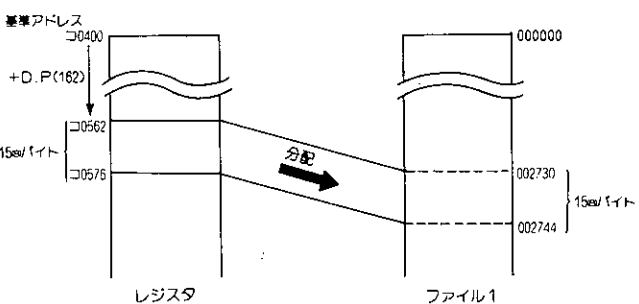
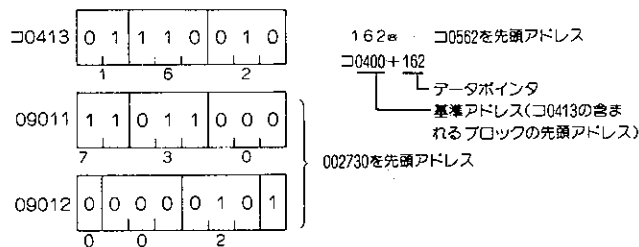
シンボル	$\overline{\text{F-72 DMPX}}$ n S D	
機能	レジスタSが含まれるデータメモリブロックの先頭アドレス(基準アドレス)からSの内容(データポインタ)だけ変位したレジスタを先頭とするnバイトのレジスタ群の内容を、レジスタD、D+1の内容でアドレス指定されるファイル1のレジスタへ転送する。	
演算内容	$X + \langle S \rangle \dots X + \langle S \rangle + n - 1$ $\rightarrow \langle D, D + 1 \rangle \dots \langle D, D + 1 \rangle + n - 1$ $X \dots S$ が含まれるデータメモリブロックの先頭アドレス(基準アドレス) $\langle S \rangle \dots$ データポインタ	
nの使用範囲	000~377(8) (000とすると256/バイトとなる)	
Sの使用範囲	$\text{C}00000 \sim \text{C}15777$: @ $\text{C}00000 \sim \text{C}15774$ $\text{C}20000 \sim \text{C}75777$: @ $\text{C}20000 \sim \text{C}75774$ $\text{b}00000 \sim \text{b}17777$: @ $\text{b}00000 \sim \text{b}17774$ $\text{b}20000 \sim \text{b}37777$: @ $\text{b}20000 \sim \text{b}37774$ $\text{0}90000 \sim \text{0}99777$: @ $\text{0}90000 \sim \text{0}99774$ $\text{E}00000 \sim \text{E}77777$: @ $\text{E}00000 \sim \text{E}77774$ ファイル1 000000~037777 : ファイル1 @000000~@037774	
Dの使用範囲	$\text{C}00000 \sim \text{C}15776$: @ $\text{C}00000 \sim \text{C}15774$ $\text{C}20000 \sim \text{C}75776$: @ $\text{C}20000 \sim \text{C}75774$ $\text{b}00000 \sim \text{b}17776$: @ $\text{b}00000 \sim \text{b}17774$ $\text{b}20000 \sim \text{b}37776$: @ $\text{b}20000 \sim \text{b}37774$ $\text{0}90000 \sim \text{0}99776$: @ $\text{0}90000 \sim \text{0}99774$ $\text{E}00000 \sim \text{E}77776$: @ $\text{E}00000 \sim \text{E}77774$ ファイル1 000000~037776 : ファイル1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	ファイル1以外のレジスタ	不変
	$\langle D, D + 1 \rangle$	レジスタ $X + \langle S \rangle$ の内容
	$\langle D, D + 1 \rangle + 1$	レジスタ $X + \langle S \rangle + 1$ の内容
	$\langle D, D + 1 \rangle + n - 1$	レジスタ $X + \langle S \rangle + n - 1$ の内容
フラグ	不変	

(解説)

命令	
STR	04000
F-72	015
	C0413
	09011



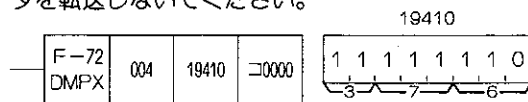
入力条件04000がOFF→ONの変化時に、C0400(レジスタC0413が含まれるデータメモリブロックの先頭アドレス…基準アドレス)からレジスタC0413の内容(データポインタ)だけ変位したレジスタを先頭とする015(8)バイトのレジスタ群の内容を、レジスタ09011と09012の内容でアドレス指定されるファイル1のレジスタを先頭とする015(8)バイトのレジスタ群に転送します。



- C0734~C0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- レジスタ側の基準アドレスは、C0000、C0400、C1000……19000、19400で、Sに含まれるデータメモリブロックの先頭アドレスとなります。(9・13ページ「データメモリのブロックと基準アドレス」参照)

参考 下記のF命令は働きが類似しています。
F-05、F-05w、F-72、F-72w

- 基準アドレスとしてC1400を使用する場合はファイル0のアドレス001600~001700のデータを、基準アドレスとして19400を使用する場合はファイル0のアドレス006000以降のデータを転送しないでください。



上記のプログラムの場合、先頭アドレスが19776となり、ファイル0のアドレス006000と006001のデータが転送されてしまいます。(9・3ページ「ソースとデスティネーション」参照)

- ファイル1のレジスタの最終アドレス(037777)を越えるとファイル2の先頭アドレス(000000)へ移ります。

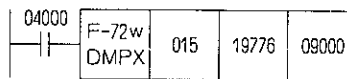
**F-72w
DMPX**

ファイル1のレジスタへのnワード分配 (DeMultiPleXer)

(この命令はJW-31CUH/H1ではプログラムできません。)

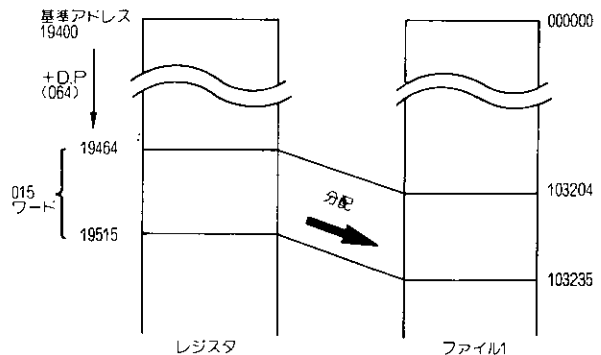
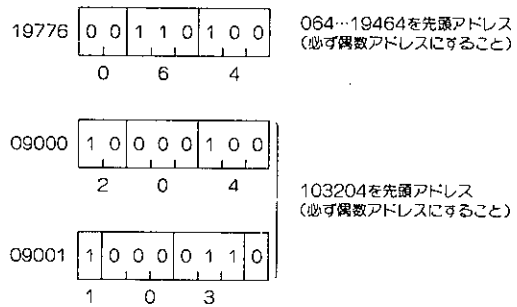
シンボル	<table border="1"><tr><td>F-72w</td><td>n</td><td>S</td><td>D</td></tr></table>	F-72w	n	S	D										
F-72w	n	S	D												
機能	レジスタSが含まれるデータメモリアドレスの先頭アドレス(基準アドレス)からSの内容(データポインタ)だけ変位したレジスタを先頭とするnワードのレジスタ群の内容をレジスタD、D+1の内容でアドレス指定されるファイル1のレジスタへ転送する。														
演算内容	$X + \langle S \rangle, \dots, X + \langle S \rangle + 2n - 1$ $\rightarrow \langle D, D+1 \rangle, \dots, \langle D, D+1 \rangle + 2n - 1$ X...Sが含まれるデータメモリアドレスの先頭アドレス(基準アドレス) $\langle S \rangle$...データポインタ														
nの使用範囲	000~377(8) (000とすると256ワードとなる)														
Sの使用範囲	<table border="0"> <tr><td>コ00000~コ15776</td><td>@コ00000~@コ15774</td></tr> <tr><td>コ20000~コ75776</td><td>@コ20000~@コ75774</td></tr> <tr><td>b00000~b17776</td><td>@b00000~@b17774</td></tr> <tr><td>b20000~b37776</td><td>@b20000~@b37774</td></tr> <tr><td>090000~997776</td><td>@090000~@997774</td></tr> <tr><td>E00000~E77776</td><td>@E00000~@E77774</td></tr> <tr><td>771#1 000000~037776</td><td>771#1 @000000~@037774</td></tr> </table>	コ00000~コ15776	@コ00000~@コ15774	コ20000~コ75776	@コ20000~@コ75774	b00000~b17776	@b00000~@b17774	b20000~b37776	@b20000~@b37774	090000~997776	@090000~@997774	E00000~E77776	@E00000~@E77774	771#1 000000~037776	771#1 @000000~@037774
コ00000~コ15776	@コ00000~@コ15774														
コ20000~コ75776	@コ20000~@コ75774														
b00000~b17776	@b00000~@b17774														
b20000~b37776	@b20000~@b37774														
090000~997776	@090000~@997774														
E00000~E77776	@E00000~@E77774														
771#1 000000~037776	771#1 @000000~@037774														
Dの使用範囲	<table border="0"> <tr><td>コ00000~コ15776</td><td>@コ00000~@コ15774</td></tr> <tr><td>コ20000~コ75776</td><td>@コ20000~@コ75774</td></tr> <tr><td>b00000~b17776</td><td>@b00000~@b17774</td></tr> <tr><td>b20000~b37776</td><td>@b20000~@b37774</td></tr> <tr><td>090000~997776</td><td>@090000~@997774</td></tr> <tr><td>E00000~E77776</td><td>@E00000~@E77774</td></tr> <tr><td>771#1 000000~037776</td><td>771#1 @000000~@037774</td></tr> </table>	コ00000~コ15776	@コ00000~@コ15774	コ20000~コ75776	@コ20000~@コ75774	b00000~b17776	@b00000~@b17774	b20000~b37776	@b20000~@b37774	090000~997776	@090000~@997774	E00000~E77776	@E00000~@E77774	771#1 000000~037776	771#1 @000000~@037774
コ00000~コ15776	@コ00000~@コ15774														
コ20000~コ75776	@コ20000~@コ75774														
b00000~b17776	@b00000~@b17774														
b20000~b37776	@b20000~@b37774														
090000~997776	@090000~@997774														
E00000~E77776	@E00000~@E77774														
771#1 000000~037776	771#1 @000000~@037774														
演算条件	入力信号の立上り(OFF→ON)														
演算後	ファイル1以外のレジスタ	不変													
	$\langle D, D+1 \rangle$	レジスタX+ $\langle S \rangle$ の内容													
	$\langle D, D+1 \rangle + 1$	レジスタX+ $\langle S \rangle + 1$ の内容													
	$\langle D, D+1 \rangle + 2n - 1$	レジスタX+ $\langle S \rangle + 2n - 1$ の内容													
フラグ	不変														

(解説)



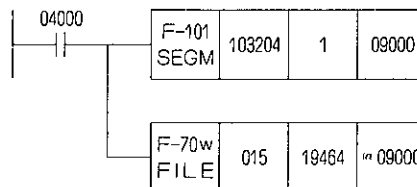
命令	
STR	04000
F-72w	015
	19776
	09000

入力条件04000がOFF→ONの変化時に19400(レジスタ19776が含まれるデータメモリアドレスの先頭アドレス...基準アドレス)からレジスタ19776の内容(データポインタ)だけ変位したレジスタを先頭とする015ワードのレジスタ群の内容をレジスタ09000と09001の内容でアドレス指定されるファイル1のレジスタを先頭とする015ワードのレジスタ群へ転送します。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- 基準アドレスに関しては(9・13ページ「データメモリのブロックと基準アドレス」の項をご参照ください)。

参考 レジスタが上記内容のとき、次の命令と同じ動作となります。



参考 下記のF命令は働きが類似しています。
F-05、F-05w、F-72、F-72w

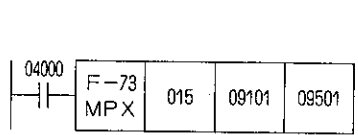
**F-73
MPX**

**ファイル1のレジスタからのnバイト抽出
(MultiPleXer)**

(この命令はJW-31CUH/H1ではプログラムできません。)

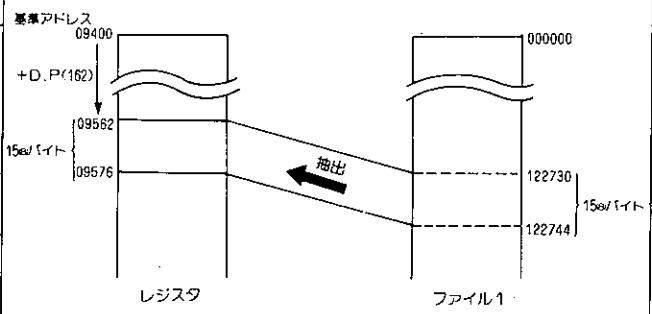
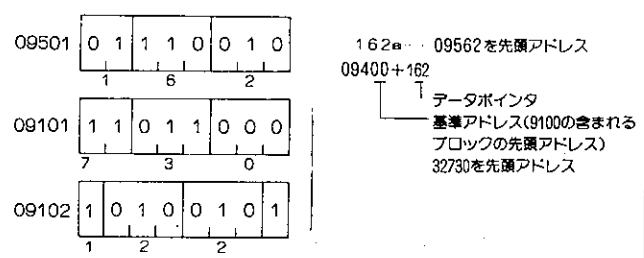
シンボル	$\overline{\text{F-73 MPX}}$ n S D														
機能	レジスタS、S+1の内容でアドレス指定されるファイル1のレジスタを先頭とするnバイトのレジスタ群の内容をレジスタDに含まれるデータメモリブロックの先頭アドレス(基準アドレス)からDの内容(データポインタ)だけ変位したレジスタを先頭とするnバイトのレジスタ群へ転送する。														
演算内容	$\langle S, S+1 \rangle \dots \langle S, S+1 \rangle + n - 1$ $\rightarrow X + \langle D \rangle, \dots X + \langle D \rangle + n - 1$ X...Dが含まれるデータメモリブロックの先頭アドレス(基準アドレス) $\langle D \rangle \dots$ データポインタ														
nの使用範囲	000~377(8) (000とすると256バイトとなる)														
Sの使用範囲	<table border="0"> <tr><td>コ0000~コ1576</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7576</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1776</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3776</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99776</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7776</td><td>@E0000~@E7774</td></tr> <tr><td>ファイル1 000000~037776</td><td>ファイル1 @000000~@037774</td></tr> </table>	コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	ファイル1 000000~037776	ファイル1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574														
コ2000~コ7576	@コ2000~@コ7574														
b0000~b1776	@b0000~@b1774														
b2000~b3776	@b2000~@b3774														
09000~99776	@09000~@99774														
E0000~E7776	@E0000~@E7774														
ファイル1 000000~037776	ファイル1 @000000~@037774														
Dの使用範囲	<table border="0"> <tr><td>コ0000~コ1577</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7577</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1777</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3777</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99777</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7777</td><td>@E0000~@E7774</td></tr> <tr><td>ファイル1 000000~037777</td><td>ファイル1 @000000~@037774</td></tr> </table>	コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	ファイル1 000000~037777	ファイル1 @000000~@037774
コ0000~コ1577	@コ0000~@コ1574														
コ2000~コ7577	@コ2000~@コ7574														
b0000~b1777	@b0000~@b1774														
b2000~b3777	@b2000~@b3774														
09000~99777	@09000~@99774														
E0000~E7777	@E0000~@E7774														
ファイル1 000000~037777	ファイル1 @000000~@037774														
演算条件	入力信号の立上り(OFF→ON)														
演算後	ファイル1のレジスタ	不変													
	X + <D>	ファイル1のレジスタ<S, S+1>の内容													
	X + <D> + 1	ファイル1のレジスタ<S, S+1> + 1の内容													
	X + <D> + n - 1	ファイル1のレジスタ<S, S+1> + n + 1の内容													
フラグ	不変														

(解説)



命令	
STR	04000
F-73	015
	09101
	09501

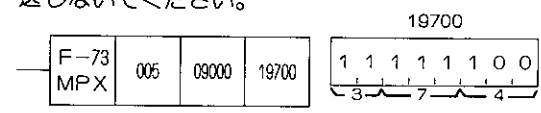
入力条件04000がOFF→ONの変化時に、レジスタ09101、09102の内容でアドレス指定されるファイル1のレジスタを先頭とする015₁₀バイトのレジスタ群の内容を、09400(09501が含まれるデータメモリブロックの先頭アドレス...基準アドレス)から09501の内容(データポインタ)だけ変位したレジスタを先頭とする015₁₀バイトのレジスタ群へ転送します。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- レジスタの基準アドレスはコ0000、コ0400、コ1000、.....19000、19400で、Dの含まれるデータメモリブロックの先頭アドレスとなります。(9・13ページ「データメモリのブロックと基準アドレス」参照)
- ファイル1のレジスタの最終アドレス(037777)を越えるとファイル2の先頭アドレス(000000)へ移ります。

参考) 下記のF命令は働きが類似しています。
F-06、F-06w、F-73、F-73w

- 基準アドレスとしてコ1400を使用する場合は、ファイル0のアドレス001600~001777に、基準アドレスとして19400を使用する場合、ファイル0のアドレス006000以降にデータを転送しないでください。



上記のプログラムの場合、先頭アドレスが19774となり、ファイル0のアドレス006000にデータが転送されます。(9・3ページ「ソースとデスティネーション」参照)

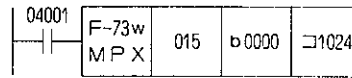
**F-73w
MPX**

**ファイル1のレジスタからのnワード抽出
(MultiPleXer)**

(この命令はJW-31CUH/H1ではプログラムできません。)

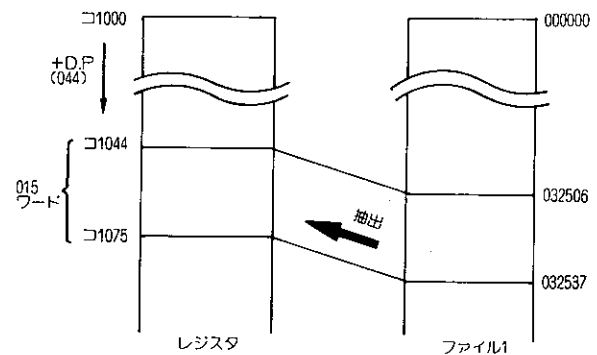
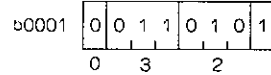
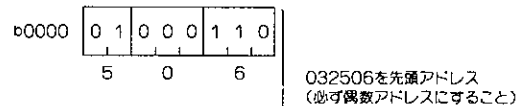
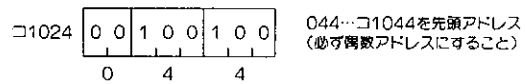
シンボル	$\overline{\text{F-73w MPX}}$ n S D	
機能	レジスタS、S+1の内容でアドレス指定されるファイル1のレジスタを先頭とするnワードのレジスタ群の内容をレジスタDが含まれるデータメモリブロックの先頭アドレス(基準アドレス)からDの内容だけ変位したレジスタを先頭とするnワードのレジスタ群へ転送する。	
演算内容	$\langle S, S+1 \rangle, \dots, \langle S, S+1 \rangle + 2n - 1$ $\rightarrow X + \langle D \rangle, \dots, X + \langle D \rangle + 2n - 1$ $X \dots D$ が含まれるデータメモリブロックの先頭アドレス(基準アドレス) $\langle D \rangle \dots$ データポインタ	
nの使用範囲	000~377(8) (000とすると256ワードとなる)	
Sの使用範囲	$\text{C}00000 \sim \text{C}15776$: @ $\text{C}00000 \sim \text{C}15774$ $\text{C}20000 \sim \text{C}75776$: @ $\text{C}20000 \sim \text{C}75774$ $\text{b}00000 \sim \text{b}17776$: @ $\text{b}00000 \sim \text{b}17774$ $\text{b}20000 \sim \text{b}37776$: @ $\text{b}20000 \sim \text{b}37774$ $090000 \sim 997776$: @ $090000 \sim 997774$ $\text{E}00000 \sim \text{E}77776$: @ $\text{E}00000 \sim \text{E}77774$ ファイル1 000000~037776 : ファイル1 @000000~@037774	
Dの使用範囲	$\text{C}00000 \sim \text{C}15776$: @ $\text{C}00000 \sim \text{C}15774$ $\text{C}20000 \sim \text{C}75776$: @ $\text{C}20000 \sim \text{C}75774$ $\text{b}00000 \sim \text{b}17776$: @ $\text{b}00000 \sim \text{b}17774$ $\text{b}20000 \sim \text{b}37776$: @ $\text{b}20000 \sim \text{b}37774$ $090000 \sim 997776$: @ $090000 \sim 997774$ $\text{E}00000 \sim \text{E}77776$: @ $\text{E}00000 \sim \text{E}77774$ ファイル1 000000~037776 : ファイル1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	ファイル1のレジスタ	不変
	$X + \langle D \rangle$	ファイル1の $\langle S, S+1 \rangle$ の内容
	$X + \langle D \rangle + 1$	ファイル1の $\langle S, S+1 \rangle + 1$ の内容
	\vdots $X + \langle D \rangle + 2n - 1$	\vdots ファイル1の $\langle S, S+1 \rangle + 2n - 1$ の内容
フラグ	不変	

〔解説〕



命令	
STR	04001
F-73w	015
	b0000
	C1024

入力条件04001がOFF→ONの変化時にレジスタb0000、b0001の内容でアドレス指定されるファイル1のレジスタを先頭とする015₈ワードのレジスタ群の内容をC1000(レジスタC1024が含まれるデータメモリブロックの先頭アドレス…基準アドレス)からレジスタC1024の内容(データポインタ)だけ変位したレジスタを先頭とするnワードのレジスタ群へ転送する。



- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(C0011、19003等は禁止)
- 基準アドレスに関しては(9・13ページ「データメモリのブロックと基準アドレス」の項をご参照ください)。

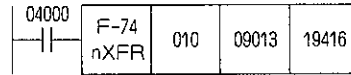
参考 下記のF命令は働きが類似しています。
F-06、F-06w、F-73、F-73w

F-74
nXFR

nバイト転送

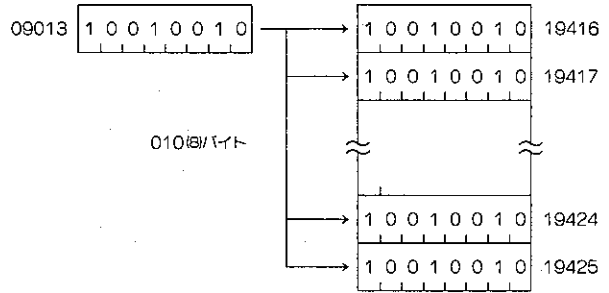
シンボル		
機能	レジスタDを先頭とするnバイトのレジスタにレジスタSの内容を転送する。	
演算内容	S→D、…D+n-1	
nの使用範囲	000~377(8) (000とすると256/バイトとなる)	
Sの使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~997777 @090000~@997774 E00000~E77777 @E00000~@E77774 777771 000000~037777 777771 @000000~@037774	
Dの使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~997777 @090000~@997774 E00000~E77777 @E00000~@E77774 777771 000000~037777 777771 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Sの内容	不変
	Dの内容	レジスタSの内容
	D+1の内容	
	D+n-1の内容	
フラグ	不変	

〔解説〕



命 令	
STR	04000
F-74	010 09013 19416

入力条件04000がOFF→ONの変化時、レジスタ19416を先頭とする010₈/バイトのレジスタにレジスタ09013の内容を転送します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

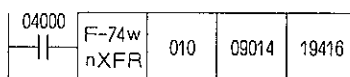
【参考】 下記のF命令は働きが類似しています。
F-00、F-00w、F-00d、F-70、F-70w、
F-74、F-74w、F-76、F-76w

F-74w
nXFR

nワード転送

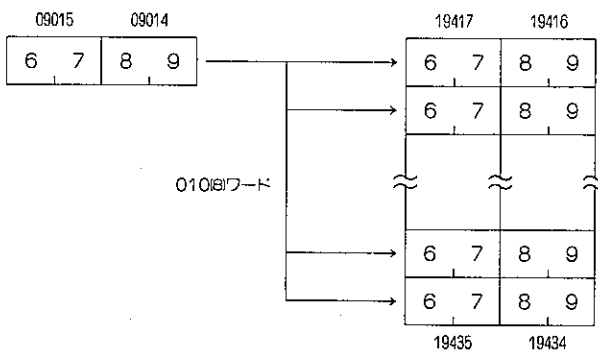
シンボル		
機能	レジスタD、D+1を先頭とするnワードのレジスタにレジスタS、S+1の内容を転送する。	
演算内容	S、S+1→D、D+1、…D+2n-2、D+2n-1	
nの使用範囲	000~377(8) (000とすると256ワードとなる)	
Sの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 774#1 000000~037776 : 774#1 @000000~@037774	
Dの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 774#1 000000~037776 : 774#1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	S、S+1の内容	不変
	Dの内容	レジスタSの内容
	D+1の内容	レジスタS+1の内容
	⋮	⋮
	D+2n-2 D+2n-1	レジスタSの内容 レジスタS+1の内容
フラグ	不変	

〔解説〕



命 令	
STR	04000
F-74w	010 09014 19416

入力条件04000がOFF→ONの変化時、レジスタ19416、19417を先頭とする010ワードのレジスタにレジスタ09014、09015の内容を転送します。



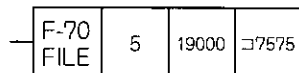
- コ0734~コ9737は特殊領域です。
(2・5ページ「特殊リレー」参照)
 - S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- 【参考】 下記のF命令は働きが類似しています。
F-00、F-00w、F-00d、F-70、F-70w、F-74、
F-74w、F-76、F-76w

F-76 FILR

nバイト一括転送 (FILR)

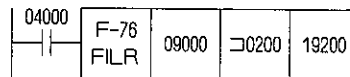
シンボル	— F-76 FILR S ₁ S ₂ D			
機能	レジスタS ₂ からS ₂ +(S ₁)-1までのS ₁ 内容のバイト(8進数)のデータをレジスタDからD+(S ₁)-1までのS ₁ 内容のバイトに一括転送する。			
演算内容	S ₂ …S ₂ +(S ₁)-1→D…D+(S ₁)-1			
S ₁ の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 774#1 000000~037777 774#1 @000000~@037774			
S ₂ の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 774#1 000000~037777 774#1 @000000~@037774			
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 774#1 000000~037777 774#1 @000000~@037774			
演算条件	入力信号の立上り(OFF→ON)			
演算後	S ₁ の内容	不変		
	S ₂ …S ₂ +(S ₁)-1の内容	不変		
	Dの内容	レジスタS ₂ の内容		
	D+1の内容	レジスタS ₂ +1の内容		
	⋮	⋮		
	D+(S ₁)-1の内容	レジスタS ₂ +(S ₁)-1の内容		
フラグ	不変			

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合必ず偶数アドレスを設定してください。(@コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- ファイル0のアドレス001600~001777と035600以降に転送しないようにしてください。



上記のようにプログラムすると035600と035601に19003と19004の内容が転送されてしまいます。(9・3ページ「ソースとディスティネーション」参照)

(解説)



命 令	
STR	04000
F-76	09000
	コ0200
	19200

入力条件04000がOFF→ONの変化時に、レジスタコ200から、レジスタ09000の内容のバイト分をDから始まる領域に一括転送する。

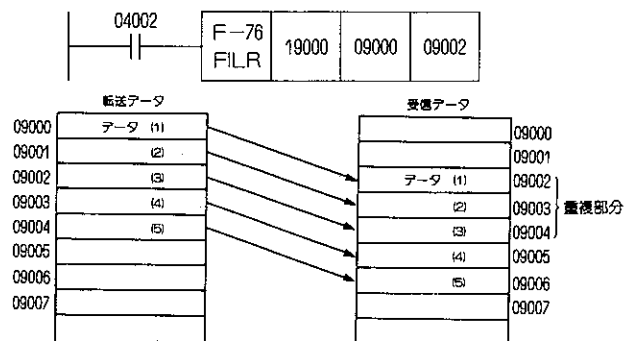
09000の内容が012₍₁₀₎バイト(8進数では014₍₈₎)の時は

	演算前		演算後	
12バイト	コ0200	0 1	19200	0 1
	コ0201	2 3	19201	2 3
	コ0202	4 5	19202	4 5
	コ0203	6 7	19203	6 7
	コ0204	8 9	19204	8 9
	コ0205	1 1	19205	1 1
	コ0206	2 2	19206	2 2
	コ0207	3 3	19207	3 3
	コ0210	4 4	19210	4 4
	コ0211	5 5	19211	5 5
	コ0212	6 6	19212	6 6
	コ0213	7 7	19213	7 7

09000の内容が003₍₁₀₎バイト(8進数では003₍₈₎)の時は

	演算前		演算後	
3バイト	コ200	0 1	19200	0 1
	コ201	2 3	19201	2 3
	コ202	4 5	19202	4 5

- S₁の内容は000~377₍₈₎です。000にすると256バイトとなります。
- 転送元、転送先が重複するようなS₁、S₂、Dの設定も可能です。



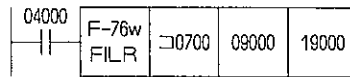
参考 下記のF命令は働きが類似しています。
 F-00、F-00w、F-00d、F-70、F-70w
 F-76、F-76w、F-74、F-74w

**F-76w
FILR**

**nワード一括転送
(FILR)**

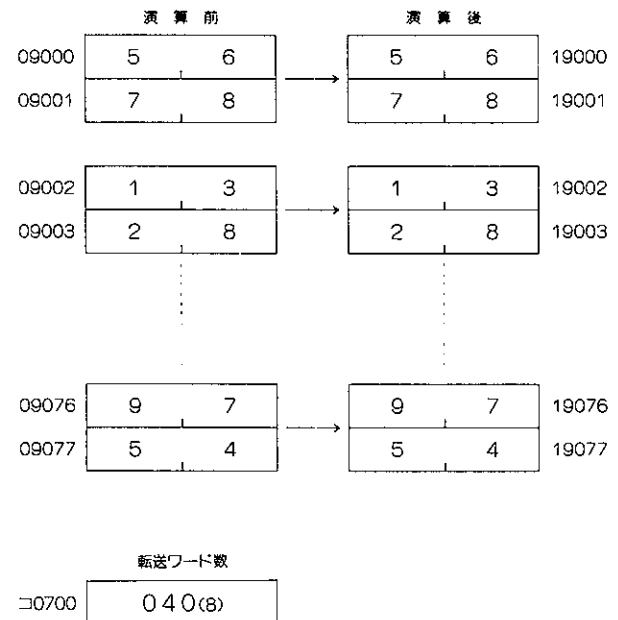
シンボル	<table border="1" style="display: inline-table;"> <tr> <td style="text-align: center;">F-76w FILR</td> <td style="text-align: center;">S₁</td> <td style="text-align: center;">S₂</td> <td style="text-align: center;">D</td> </tr> </table>				F-76w FILR	S ₁	S ₂	D
F-76w FILR	S ₁	S ₂	D					
機能	レジスタS ₂ からS ₂ +2(S ₁)-1までのS ₁ 内容のワードのデータをレジスタDからD+2(S ₁)-1までのS ₁ 内容ワードに一括転送する。							
演算内容	S ₂ , S ₂ +1, …, S ₂ +2(S ₁)-1 → D, D+1, …, D+2(S ₁)-1							
S ₁ の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774							
S ₂ の使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 771#1 000000~037776 771#1 @000000~@037774							
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 771#1 000000~037776 771#1 @000000~@037774							
演算条件	入力信号の立上り(OFF→ON)							
演算後	S ₁ の内容	不変						
	S ₂ …S ₂ +2(S ₁)-1の内容	不変						
	Dの内容	レジスタS ₂ の内容						
	D+1の内容	レジスタS ₂ +1の内容						
	⋮	⋮						
D+2(S ₁)-1の内容	レジスタS ₂ +2(S ₁)-1の内容							
フラグ	不変							

(解説)



命令	
STR	04000
F-76w	コ0700
	09000
	19000

コ0700の内容が040₍₈₎ワードのときに、レジスタ09000から9077までの040₍₈₎ワード(10進数で32ワード)のデータをレジスタ19000から19077までの32ワードに一括転送します。レジスタ09000から09077までの内容は不変です。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- S₁の内容は000~377₍₈₎です。000にすると256ワードとなります。

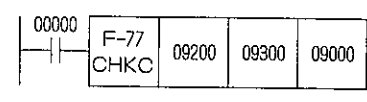
参考 下記のF命令は働きが類似しています。
F-00、F-00w、F-00d、F-70、F-70w
F-76、F-76w、F-74、F-74w

**F-77
CHKC**

**データサムチェックコードの生成
(CHecK Code)**

シンボル	<table border="1"> <tr> <td>F-77 CHKC</td> <td>S₁</td> <td>S₂</td> <td>D</td> </tr> </table>				F-77 CHKC	S ₁	S ₂	D
F-77 CHKC	S ₁	S ₂	D					
機能	レジスタS ₂ からS ₂ +(S ₁)-1まで、(レジスタS ₁ の内容のバイト分)のサムチェックコードを作成し、レジスタDに格納する。							
演算内容	0 - Σ(S ₂ ~S ₂ +(S ₁)-1) → D							
S ₁ の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774							
S ₂ の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774							
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774							
演算条件	入力信号の立上り(OFF→ON)							
演算後	S ₁ の内容	不変						
	S ₂ ~S ₂ +n-1の内容	不変						
	Dの内容	演算結果						
	フラグ	不変						

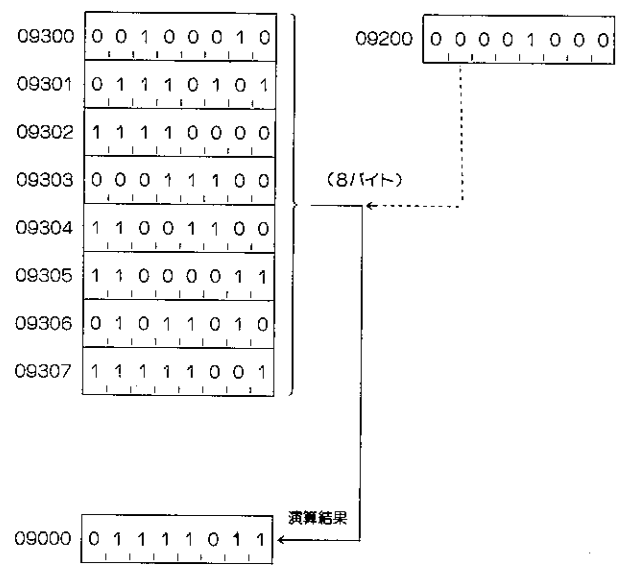
(解説)



命令	
STR	00000
F-77	09200
	09300
	09000

入力条件00000がOFF→ONの変化時にレジスタ09300から09200の内容バイト分のサムチェックコードを計算し、レジスタ09000に格納します。

①レジスタ09200の値が8バイトのとき
レジスタ09300~09307のチェックコードを生成しレジスタ09000へ格納する。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- S₁の内容は000~377₍₈₎です。000にすると256バイトとなります。

• サムチェックの求め方(解説の例)

```

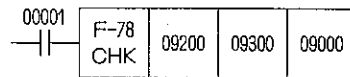
22(H)
75
F0
1C
CC
C3
5A
+ F9
-----
485
  85  下2桁の補数を計算
-----
7B   2の補数(100(H)-85(H))
よってサムチェック値は7B(H)です。
  
```

**F-78
CHK**

**データのチェック
(CHekK)**

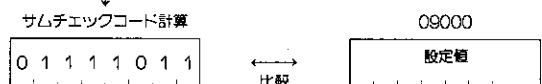
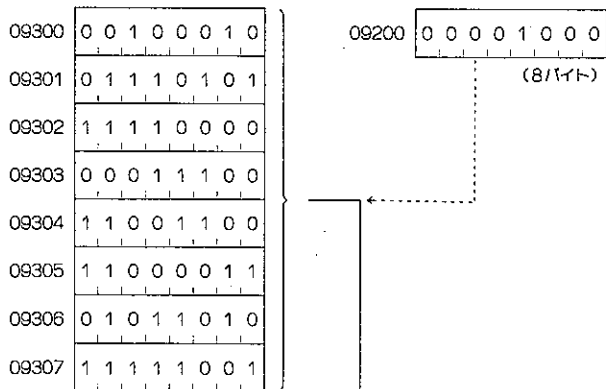
シンボル	— F-78 CHK S ₁ S ₂ S ₃				
機能	レジスタS ₂ からS ₂ +(S ₁)-1まで(レジスタS ₁ の内容のバイト分)のサムチェックコードを作成し、あらかじめF-77で作成しておいたチェックコードが格納されているレジスタS ₃ と比較し、フラグを変化させる。				
演算内容	$(0 - \sum(S_2 \sim S_2 + (S_1) - 1)) \xrightarrow{\text{比較}} S_3$ →フラグ				
S ₁ の使用範囲	コ0000~コ1577 コ2000~コ7577 b0000~b1777 b2000~b3777 09000~99777 E0000~E7777 ファイル000000~037777	@	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~@99774 @E0000~@E7774 ファイル@000000~@037774		
S ₂ の使用範囲	コ0000~コ1577 コ2000~コ7577 b0000~b1777 b2000~b3777 09000~99777 E0000~E7777 ファイル000000~037777	@	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~@99774 @E0000~@E7774 ファイル@000000~@037774		
S ₃ の使用範囲	コ0000~コ1577 コ2000~コ7577 b0000~b1777 b2000~b3777 09000~99777 E0000~E7777 ファイル000000~037777	@	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~@99774 @E0000~@E7774 ファイル@000000~@037774		
演算条件	入力信号の立上り (OFF→ON)				
演算後	S ₁ の内容	不変			
	S ₂ の内容	不変			
	S ₃ の内容	不変			
フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
	エラー無し	0	0	0	0
	サムチェックエラー	0	0	1	0

〔解説〕



命令	
STR F-78	00001
	09200
	09300
	09000

入力条件00001がOFF→ONの変化時に、レジスタ09300から09200の内容バイト分のサムチェックを計算し、あらかじめF-77で作成しておいたチェックコードが格納されているレジスタ09000と比較します。



比較の結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
同じ	0	0	0	0
異なる	0	0	1	0

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合、必ず偶数アドレスを設定してください。(@コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)
- S₁の内容は000~377₍₈₎です。000にすると256バイトとなります。

F-79 SORT

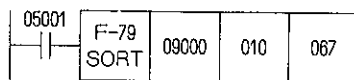
レジスタ(1バイト)データの並べかえ

(この命令はJW-31CUH/H1ではプログラムできません。)

シンボル	— F-79 SORT S n ₁ n ₂			
機能	レジスタSからS+n ₁ -1までの内容を小さい方から順に並べかえます。n ₂ は比較データのマスク値です。			
演算内容	S < S+1 ~ S+n ₁ -2 < S+n ₁ -1			
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 ファイル000000~037777 ファイル@000000~@037774			
n ₁ の使用範囲	000~377 ₍₈₎ (000にすると256バイトとなります)			
n ₂ の使用範囲	000~377 ₍₈₎			
演算条件	入力信号の立上り(OFF→ON)			
演算後	S~S+n ₁ -1の内容	演算結果(小さい方からの順)		
	フラグ	不変		

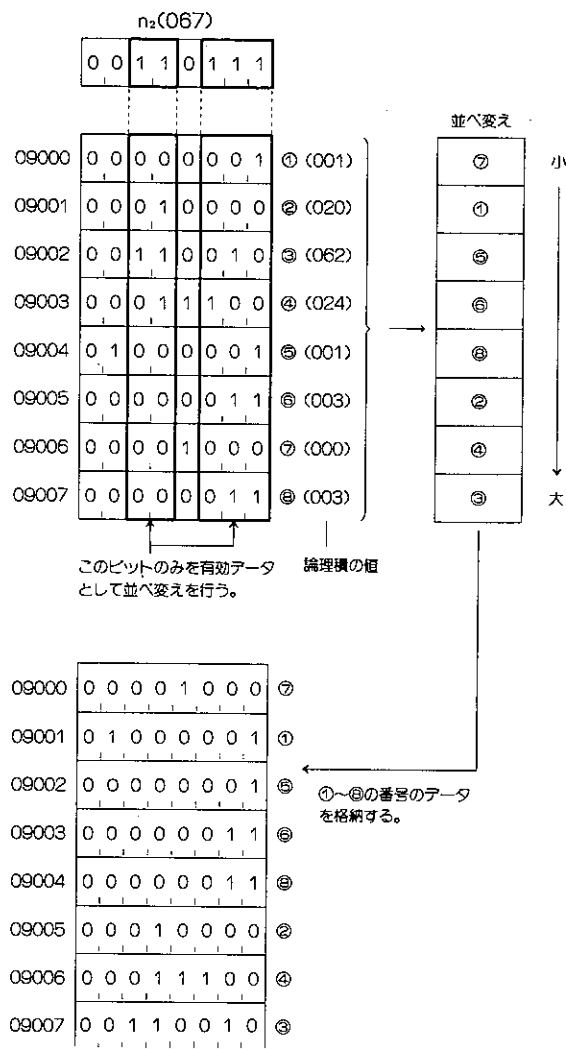
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合、必ず偶数アドレスを設定してください。(＠コ0001、＠b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- n₂が000と377は同じ動作をし、全ビットが有効となります。
- データ交換バイト数が多いとき演算時間が長くなります。演算時間は7・20ページ「命令語処理時間」を参照ください。

〔解説〕



命令	
STR	05001
F-79	09000
	010
	067

入力条件05001がOFF→ONの変化時に、レジスタ09000~09007の8バイト(n₁内容)をn₂の内容067₍₈₎で有効ビットを論理積(AND)し、その結果で大小比較したあと、小さい順にレジスタ09000から並べて行きます。



データそのものは変化しません。

F-79w SORT

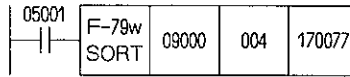
レジスタ(1ワード)データの並べかえ

(この命令はJW-31CUH/H1ではプログラムできません。)

シンボル	F-79w SORT				S	n_1	n_2
機能	レジスタSからS+2(n_1-1)までの内容を小さい方から順に並べかえます。 n_2 は比較データのマスク値です。						
演算内容	$S < S+1 \sim S+n_1-2 < S+n_1-1$						
Sの使用範囲	C0000~C1577		@C0000~@C1574				
	C2000~C7577		@C2000~@C7574				
	b0000~b1777		@b0000~@b1774				
	b2000~b3777		@b2000~@b3774				
	09000~99777		@09000~@99774				
	E0000~E7777		@E0000~@E7774				
	771#1 000000~037777		771#1 @000000~@037774				
n_1 の使用範囲	000~377(n_1) (000にすると256ワードとなります)						
n_2 の使用範囲	000000~177777						
演算条件	入力信号の立上り(OFF→ON)						
演算後	S+2(n_1-1)の内容	演算結果(小さい方からの順)					
フラグ	不変						

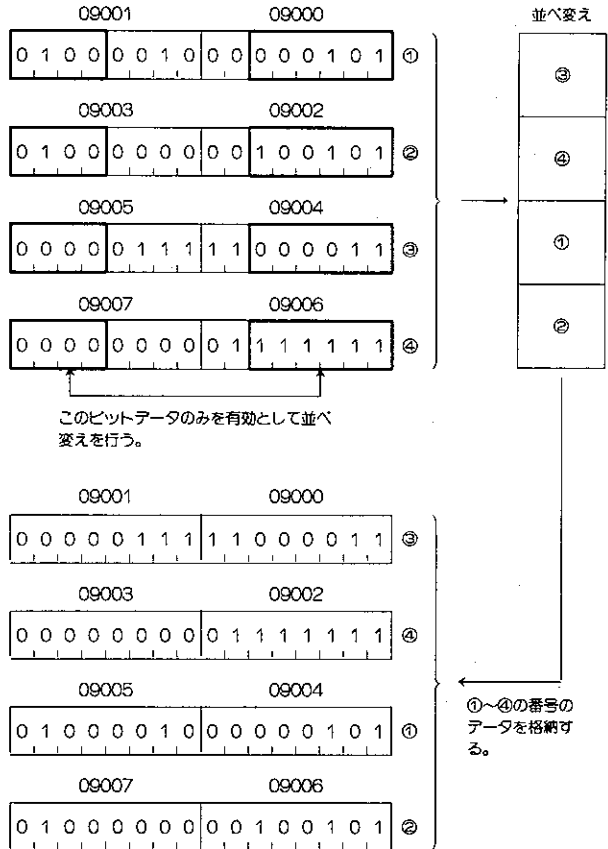
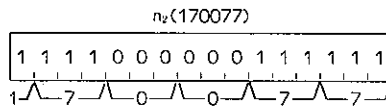
- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合、必ず偶数アドレスを設定してください。(C0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- Sには必ず偶数アドレスを設定してください。
(C0011、19003等は禁止)
- n_2 が000000と177777は同じ動作をし、全ビットが有効となります。
- データ交換バイト数が多いとき演算時間が長くなります。演算時間は7・20ページ「命令語処理時間」を参照ください。

(解説)



命令	
STR	05001
F-79w	09000
	004
	170077

入力条件05001がOFF→ONの変化時に、レジスタ09000~09007の4ワード(n_1 内容)を n_2 の内容170077(8)で有効ビットを論理積(AND)し、その結果で大小比較したあと、小さい順にレジスタ09000から並べて行きます。



データそのものは変化しません。

F-80
IORF

I/Oリフレッシュ
(I/O ReFresh)

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-80</td><td>RACK-SLOT</td></tr><tr><td>IORF</td><td></td></tr></table>		F-80	RACK-SLOT	IORF		<p>(解説)</p> <div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> <table border="1" style="font-size: small;"> <tr><td>04000</td><td> </td><td> </td></tr> <tr><td></td><td>└─┘</td><td></td></tr> </table> </div> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>F-80</td><td>RACK-SLOT</td></tr><tr><td>IORF</td><td>0 2</td></tr> </table> </div>		04000				└─┘		F-80	RACK-SLOT	IORF	0 2																														
F-80	RACK-SLOT																																															
IORF																																																
04000																																																
	└─┘																																															
F-80	RACK-SLOT																																															
IORF	0 2																																															
機能	RACK(ラック)番号とSLOT(スロット)番号で指定した入出力ユニットのすべてのデータとPCデータメモリ間でデータ交換を行う。		<table border="1" style="font-size: small; text-align: center;"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>04000</td></tr> <tr><td>F-80</td><td></td></tr> <tr><td>R</td><td>0</td></tr> <tr><td>S</td><td>2</td></tr> </table>		命 令		STR	04000	F-80		R	0	S	2																																		
命 令																																																
STR	04000																																															
F-80																																																
R	0																																															
S	2																																															
演算内容	入力ユニット→データメモリ 出力ユニット←データメモリ		<p>入力条件04000がONの時、ラック番号0、スロット番号2に実装しているユニットのデータとデータメモリ間でデータ交換を行います。</p>																																													
RACKの使用範囲	0~7																																															
SLOTの使用範囲	0~7																																															
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)																																															
演算後	<table border="1" style="font-size: x-small;"> <tr> <td>入力ユニット</td> <td>データメモリの更新</td> <td colspan="4">バス異常の時データメモリ又は出力状態は更新されない。</td> </tr> <tr> <td>出力ユニット</td> <td>出力状態の更新</td> <td colspan="4"></td> </tr> <tr> <td rowspan="5">フ ラ グ</td> <td>I/Oリフレッシュ後</td> <td>ゼロ</td> <td>キャリー</td> <td>エラー</td> <td>ノキャリー</td> </tr> <tr> <td>バス異常</td> <td>07357</td> <td>07356</td> <td>07355</td> <td>07354</td> </tr> <tr> <td>バス異常</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>ユニット無し</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>転送完了</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>非実行時</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>	入力ユニット			データメモリの更新	バス異常の時データメモリ又は出力状態は更新されない。				出力ユニット	出力状態の更新					フ ラ グ	I/Oリフレッシュ後	ゼロ	キャリー	エラー	ノキャリー	バス異常	07357	07356	07355	07354	バス異常	0	0	1	0	ユニット無し	0	0	0	1	転送完了	0	1	0	0	非実行時	0	0	0	0		
入力ユニット	データメモリの更新	バス異常の時データメモリ又は出力状態は更新されない。																																														
出力ユニット	出力状態の更新																																															
フ ラ グ	I/Oリフレッシュ後	ゼロ	キャリー	エラー	ノキャリー																																											
	バス異常	07357	07356	07355	07354																																											
	バス異常	0	0	1	0																																											
	ユニット無し	0	0	0	1																																											
	転送完了	0	1	0	0																																											
非実行時	0	0	0	0																																												

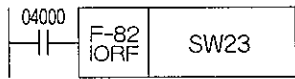
- 本命令はプログラム演算中に何回でも使用できます。またRACK、SLOTの番号を同一にする必要はありません。
- 本命令でI/Oリフレッシュしている入出力ユニットは通常のI/Oサイクルでもデータのリフレッシュを行います。
- 本命令でリフレッシュしている入出力リレーをプログラマ(JW-13PG)等で強制セット/リセットできません。(強制セット/リセットについてはプログラマの「取扱説明書」を参照してください。)
- 入力割込用(システムメモリ#240~#243で設定)で使用している入力ユニットのアドレスはI/Oリフレッシュに使用しないでください。
- スロット番号(SLOT)の上限は使用するベースユニットによって異なります。
- 非実行時とは入力条件OFFのときです。全てのフラグがOFFになります。

**F-82
IORF**

**特殊I/Oのリフレッシュ
(I/O ReFresh)**

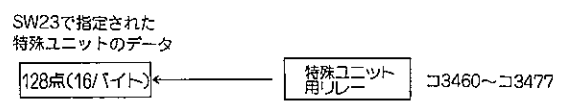
シンボル	— F-82 IORF SW					
機能	SW(ラック番号とユニットNo.スイッチで決まる番号)で指定した特殊I/OユニットとPCのデータメモリ筒でデータメモリ(16バイト)と制御リレーの交換を行う。					
演算内容	特殊入力ユニット→データメモリ 特殊出力ユニット←データメモリ					
SWの設定	00~37oct (上位桁 ラック番号(0~3) 下位桁 ユニットNo.スイッチ(0~7))					
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)					
演算後	特殊入力ユニット	データメモリの更新	バス異常又は特殊I/O異常のときデータメモリ及び出力状態は更新されない			
	特殊出力ユニット	出力状態の更新				
フラグ	演算後	ゼロ	キャリー	エラー	ノンキャリー	
	特殊ユニット異常又はバス異常	07357	07356	07355	07354	
	転送完了	0	0	1	0	
	転送データなし又は特殊ユニット無し	0	1	0	0	
	非実行時	0	0	0	0	1

(解説)



命 令	
STR	04000
F-82	SW23

入力条件04000がONの時、ラック2,ユニットNo.スイッチ3の特殊ユニットのデータ(16バイト)とPCのデータのリフレッシュを行います。



- 本命令はプログラム演算中に何回でも使用できます。
- 本命令は、特殊I/Oユニットと特殊ユニット用リレーとのデータ交換用で転送データバイト数は16バイトです。
- 本命令でデータのリフレッシュしている特殊I/Oユニットは通常のI/Oサイクルでもデータのリフレッシュを行います。

- 特殊I/Oユニットのデータ変換が完了していない時に本命令を実行するとノンキャリーフラグ(07354)がONします。
- バス異常はI/Oリフレッシュ中に、ノイズ等による異常データやバス異常を検知したときエラーフラグ(07355)がONします。
- 非実行時とは入力条件OFFのときです。全てのフラグがOFFになります。
- 特殊I/O異常は、特殊I/Oユニットから出力する異常信号です。ユニットによっては本信号がありません。
- リモートI/O子局(JW-21RS)に実装の特殊I/Oユニットのリフレッシュはできません。

**F-85
PRRD**

特殊I/Oからの読出し

**F-86
PRWR**

特殊I/Oへの書込み

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>F-85 PRRD</td> <td>n₁</td> <td>SW,n₂</td> <td>D</td> </tr> </table>				F-85 PRRD	n ₁	SW,n ₂	D	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>F-86 PRWR</td> <td>n₁</td> <td>D</td> <td>SW,n₂</td> </tr> </table>				F-86 PRWR	n ₁	D	SW,n ₂
F-85 PRRD	n ₁	SW,n ₂	D													
F-86 PRWR	n ₁	D	SW,n ₂													
機能	SW(ラック番号とユニットNo.スイッチで決まる番号)で示される特殊ユニットの特殊I/O専用F命令領域 (n ₂) のバイト数 (n ₁) をレジスタDを先頭とするD+n ₂ -1に読出す。				レジスタDを先頭とするn ₁ バイトの内容をSW(ラック番号とユニットNo.スイッチで決まる番号)で示される特殊ユニット特殊I/O専用F命令領域 (n ₂) へ転送する。											
n ₁ の使用範囲	000~377oct (000とすると256バイトとなる)				000~377oct (000とすると256バイトとなる)											
SWの使用範囲	00~37 (上位桁 ラック番号(0~3) 下位桁 ユニットNo.スイッチ(0~7))				00~37 (上位桁 ラック番号(0~3) 下位桁 ユニットNo.スイッチ(0~7))											
特殊I/O専用F命令領域 (n ₂)	0 ~ 3				0 ~ 3											
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774				コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774											
演算条件	入力信号の立上り (OFF→ON)				入力信号の立上り (OFF→ON)											
演算後	n ₁ の内容	不変				同左										
	SWの内容	不変				同左										
	n ₂ の内容	不変				同左										
	Dの内容	n ₂ の内容				不変										
フラグ		ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354	同左										
	特殊ユニットからの応答なし	0	0	1	0											
	転送待ちのとき	0	0	0	1											
	転送完了時	0	1	0	0											
上記以外	0	0	0	0												

- F-85、F-86は1部の特殊I/Oユニットに限り使用する命令です。(JW-21SU, JW-21PS, JW-21DU/22DU)
使用する必要があるユニットは、そのユニットの「取扱説明書、マニュアル」に使用方法を記載しています。
- この命令を必要とするユニット以外の他のユニットには使用しないでください。使用すると誤動作の原因になることがあります。
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 特殊I/O専用F命令領域 (n₂) は各ブロックを0~3の数値で設定します。
 特殊I/O専用F命令Aブロック→0
 特殊I/O専用F命令Bブロック→1
 特殊I/O専用F命令Cブロック→2
 特殊I/O専用F命令Dブロック→3
 なお各ブロックは256バイトあります。

F-90 REM リマーク
(REMark)

シンボル	F-90 REM	n	【解説】 ・F-90用のシンボル・コメントは、多機能プログラマ、またはラダーソフトで「シンボル・コメント設定」にて登録します。 (シンボル：半角16文字、コメント：半角28文字) ・ラダー印字時は、シンボル・コメント内容を印字し、F-90命令は印字しません。 また、シンボル内容の1文字目に@ (アットマーク) を登録すると改ページとなり、シンボル・コメント内容は印字しません。 ・命令語印字時は、F-90命令、シンボル・コメント内容ともに印字します。 また、シンボル内容の1文字目に@ (アットマーク) を登録しても改ページは行われず、登録内容を印字します。
機能	多機能プログラマ (JW-50PG)、またはラダーソフト (JW-50SP/52SP/92SP) でラダー、命令語印字時、行コメントの印字を行う。		
演算内容	演算しない (NOPと同じ)		
nの使用範囲	0000~3777(8)		
演算後	フラグ等データメモリは不変		

【使用例】

<p>(1) ラダープログラミング</p>	<p>(2) シンボル・コメント設定</p> <table border="0"> <tr> <td></td> <td>アドレス</td> <td>シンボル</td> <td>コメント</td> </tr> <tr> <td>F-90</td> <td>0010</td> <td>No.10</td> <td>異常処理監視部</td> </tr> <tr> <td>F-90</td> <td>0011</td> <td>@</td> <td></td> </tr> </table> <p>↑ ラダー印字の時、改ページ</p>		アドレス	シンボル	コメント	F-90	0010	No.10	異常処理監視部	F-90	0011	@	
	アドレス	シンボル	コメント										
F-90	0010	No.10	異常処理監視部										
F-90	0011	@											



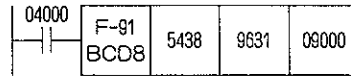
<p>(3) ラダー印字</p>	<p>(4) 命令語印字</p> <table border="0"> <tr> <td>STR</td> <td>00000</td> <td></td> </tr> <tr> <td>OUT</td> <td>00400</td> <td></td> </tr> <tr> <td>F-90[REM]</td> <td></td> <td></td> </tr> <tr> <td>STR</td> <td>0010</td> <td>No.10: 異常処理監視部</td> </tr> <tr> <td>OR</td> <td>00100</td> <td></td> </tr> <tr> <td>AND</td> <td>00102</td> <td></td> </tr> <tr> <td>OUT</td> <td>00401</td> <td></td> </tr> </table>	STR	00000		OUT	00400		F-90[REM]			STR	0010	No.10: 異常処理監視部	OR	00100		AND	00102		OUT	00401	
STR	00000																					
OUT	00400																					
F-90[REM]																						
STR	0010	No.10: 異常処理監視部																				
OR	00100																					
AND	00102																					
OUT	00401																					

**F-91
BCD8**

BCD定数(8桁)の転送

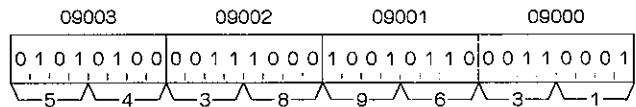
シンボル															
機能	8桁のBCD定数 n_1 、 n_2 (n_1 は上位4桁、 n_2 は下位4桁)をレジスタD~D+3に転送する。														
演算内容	$n_1 \rightarrow D+3, D+2$ $n_2 \rightarrow D+1, D$														
n_1 の使用範囲	0000~9999 (FFFF(16進数)まで可能)														
n_2 の使用範囲	0000~9999 (FFFF(16進数)まで可能)														
Dの使用範囲	$\text{C}0000 \sim \text{C}1574 ; @\text{C}0000 \sim @\text{C}1574$ $\text{C}2000 \sim \text{C}7574 ; @\text{C}2000 \sim @\text{C}7574$ $\text{b}0000 \sim \text{b}1774 ; @\text{b}0000 \sim @\text{b}1774$ $\text{b}2000 \sim \text{b}3774 ; @\text{b}2000 \sim @\text{b}3774$ $09000 \sim 99774 ; @09000 \sim @99774$ $\text{E}0000 \sim \text{E}7774 ; @\text{E}0000 \sim @\text{E}7774$ $771\#1 000000 \sim 037774 ; 771\#1 @000000 \sim @037774$														
演算条件	入力信号の立上り(OFF→ON)														
演算後	演算結果 <table border="1"> <tr> <td>D</td> <td>10^1</td> <td>10^0</td> <td rowspan="2">} n_2</td> </tr> <tr> <td>D+1</td> <td>10^3</td> <td>10^2</td> </tr> <tr> <td>D+2</td> <td>10^5</td> <td>10^4</td> <td rowspan="2">} n_1</td> </tr> <tr> <td>D+3</td> <td>10^7</td> <td>10^6</td> </tr> </table>	D	10^1	10^0	} n_2	D+1	10^3	10^2	D+2	10^5	10^4	} n_1	D+3	10^7	10^6
	D	10^1	10^0	} n_2											
D+1	10^3	10^2													
D+2	10^5	10^4	} n_1												
D+3	10^7	10^6													
フラグ	不変														

(解説)



命令	
STR	04000
F-91	5438 9631 09000

入力条件04000がOFF→ONの変化時にレジスタ09000~09003にBCD定数54389631を転送します。レジスタ09000~09003は転送時コードで下記の数値になります。



- $\text{C}0734 \sim \text{C}0737$ は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合、必ず偶数アドレスを設定してください。(@ $\text{C}0001$ 、@ $\text{b}0173$ 等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- 定数転送00000000~FFFFFFFFが書込可能なのはハンディプログラマや多機能プログラマを使用した場合です。

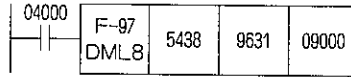
参考 下記のF命令は働きが類似しています。
F-01、F-01w、F-91

**F-97
DML8**

10進定数(8桁)の転送

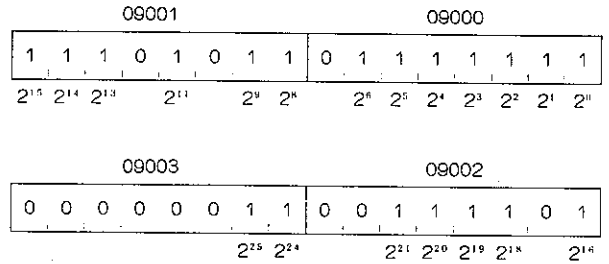
シンボル	F-97 DML8 n_1 n_2 D		
機能	8桁の10進定数 n_1 、 n_2 ($n_1 \times 10000 + n_2$)をレジスタD~D+3に転送する。		
演算内容	$n_1 \times 10000 + n_2 \rightarrow D \sim D+3$		
n_1 の使用範囲	0000~9999		
n_2 の使用範囲	0000~9999		
Dの使用範囲	コ0000~コ1574 @コ0000~@コ1574 コ2000~コ7574 @コ2000~@コ7574 b0000~b1774 @b0000~@b1774 b2000~b3774 @b2000~@b3774 09000~99774 @09000~@99774 E0000~E7774 @E0000~@E7774 771#1 000000~037774 771#1 @000000~@037774		
演算条件	入力信号の立上り(OFF→ON)		
演算後	D~D+3の内容	演算結果 10進定数 (00000000~99999999)	
	フラグ	不変	

(解説)



命 令	
STR	04000
F-97	5438 9631 09000

入力条件04000がOFF→ONの変化時にレジスタ09000~09003に10進定数54389631を転送します。レジスタ09000~09003は転送時バイナリコードで下記の数値になります。



$$1+2+4+8+16+32+64+256+512+2048+8192+16384+32768+65536+262144+524288+1048576+2097152+16777216+33554432=54389631$$

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合、必ず偶数アドレスを設定してください。(コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- バイナリコードの各ビットの重みは下記のとおりです。

参考 下記のF命令は働きが類似しています。
F-07、F-07w、F-97

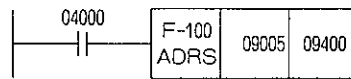
	7	6	5	4	3	2	1	0
D	128 (2 ⁷)	64 (2 ⁶)	32 (2 ⁵)	16 (2 ⁴)	8 (2 ³)	4 (2 ²)	2 (2 ¹)	1 (2 ⁰)
D+1	32768 (2 ¹⁵)	16384 (2 ¹⁴)	8192 (2 ¹³)	4096 (2 ¹²)	2048 (2 ¹¹)	1024 (2 ¹⁰)	512 (2 ⁹)	256 (2 ⁸)
D+2	8388608 (2 ²³)	4194304 (2 ²²)	2097152 (2 ²¹)	1048576 (2 ²⁰)	524288 (2 ¹⁹)	262144 (2 ¹⁸)	131072 (2 ¹⁷)	65536 (2 ¹⁶)
D+3	—	—	—	—	—	67108864 (2 ²⁶)	33554432 (2 ²⁵)	16777216 (2 ²⁴)

**F-100
ADRS**

間接アドレスの設定

シンボル	— F-100 ADRS S D		
機能	レジスタSのファイルアドレスをレジスタD、D+1、D+2に設定する。		
演算内容	S→D、D+1、D+2		
Sの使用範囲	コ00000~コ15777 : @コ00000~@コ15774 コ20000~コ75777 : @コ20000~@コ75774 b00000~b17777 : @b00000~@b17774 b20000~b37777 : @b20000~@b37774 090000~997777 : @090000~@997774 E00000~E77777 : @E00000~@E77774 ファイル000000~037777 : ファイル000000~@037774		
Dの使用範囲	コ00000~コ15774 : @コ00000~@コ15774 コ20000~コ75774 : @コ20000~@コ75774 b00000~b17774 : @b00000~@b17774 b20000~b37774 : @b20000~@b37774 090000~997774 : @090000~@997774 E00000~E77774 : @E00000~@E77774 ファイル000000~037774 : ファイル000000~@037774		
演算条件	入力信号の立上り(OFF→ON)		
演算後	Sの内容	不変	
	Dの内容	レジスタSのファイルアドレス(下位)	
	D+1の内容	レジスタSのファイルアドレス(上位)	
	D+2の内容	レジスタSのファイル番号	
	フラグ	不変	

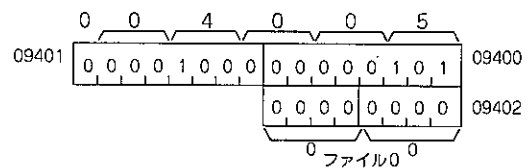
(解説)



命令	
STR	04000
F-100	09005 09400

入力条件04000がOFF→ONの変化時に、レジスタ09005のファイルアドレスをレジスタ09400、09401、09402に設定する。

●演算後のレジスタ



ファイル0のファイルアドレス004005はレジスタ9005を示します。

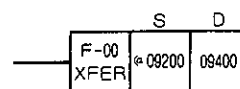
この設定したファイルアドレスは間接アドレス指定時の直接アドレスとなります。(≠09400)

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 演算前のレジスタSの内容は、演算に関与しません。
- Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)

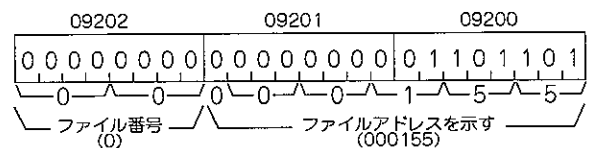
参考 間接アドレスとは

データ処理命令の中にはソース、デスティネーションに間接アドレスを指定できる命令があります。間接アドレス指定とはソース、デスティネーションに指定したレジスタ自身が演算を実行するのではなくそのレジスタを先頭とする3バイトの内容で指定されるファイルアドレスのレジスタが演算を実行することをいいます。間接アドレス指定の場合、レジスタの前に@ (アットマーク) を付加します。

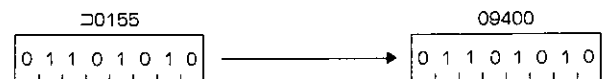
(例1)



09200、09201、09202の内容で指定されるレジスタの内容を09400に転送されます。



上例では、ファイル0のファイルアドレス000155はコ0155ですので結果的に@09200はコ0155を示します。

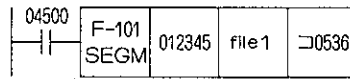


F-101 SEGM

間接アドレスの設定

シンボル	F-101 SEGM	n	fileN	D
機能	間接アドレス指定に使用するファイルアドレスnをレジスタD、D+1に設定する。 ファイル番号NをレジスタD+2に設定する。			
演算内容	n → D、D+1 fileN → D+2			
nの使用範囲	000000~177777(8)			
Nの使用範囲	0~3, 10~2C(H)			
Dの使用範囲	コ0000~コ1574 @コ0000~@コ1574 コ2000~コ7574 @コ2000~@コ7574 b0000~b1774 @b0000~@b1774 b2000~b3774 @b2000~@b3774 09000~99774 @09000~@99774 E0000~E7774 @E0000~@E7774 ファイル1 000000~037774 ファイル1 @000000~@037774			
演算条件	入力信号の立上り(OFF→ON)			
演算後	Dの内容	n(下位)		
	D+1の内容	n(上位)		
	D+2の内容	ファイル番号(000~007)		
	フラグ	不変		

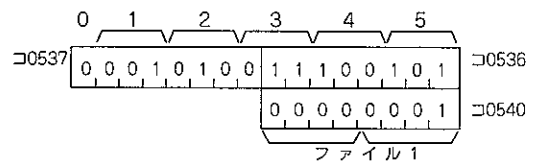
(解説)



命令	
STR	04500
F-101	012345 file1 コ0536

入力条件04500がOFF→ONの変化時、ファイル1のファイルアドレス012345をレジスタコ0536、コ0537、コ0540に設定します。

●演算後のレジスタ

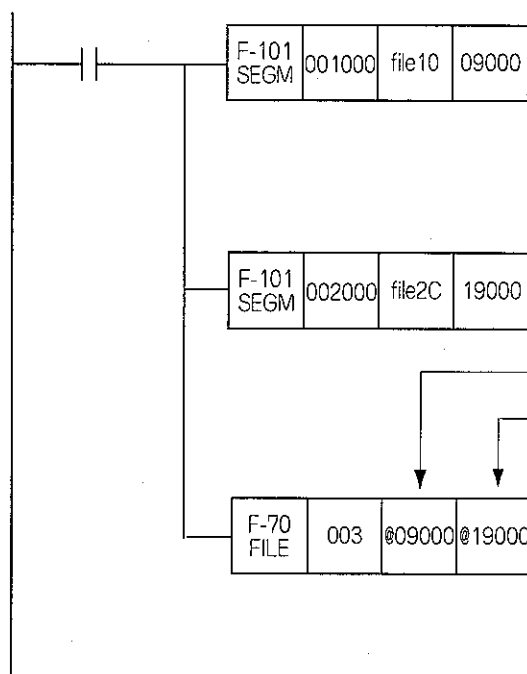


この設定したファイルアドレスは、間接アドレス指定時の直接アドレスとなります。(コ0536)

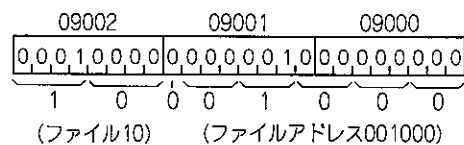
● コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

● Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)

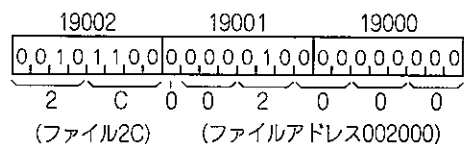
(例) ファイル10の001000からの3バイトをファイル2Cの002000からの3バイトに転送するプログラム(JW-33CUH3の場合)



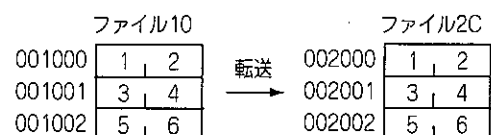
・間接アドレスの設定 (F-70のソース側)



・間接アドレスの設定 (F-70のデスティネーション側)



・3バイトデータの転送



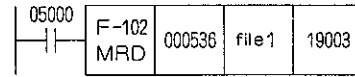
**F-102
MRD**

直接指定アドレスのレジスタからの読出(1バイト)

(この命令はJW-31CUH/H1ではプログラムできません。)

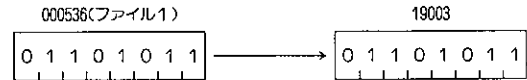
シンボル	$\text{F-102 MRD } n \text{ fileN } D$	
機能	ファイルNの、ファイルアドレスnのレジスタの内容をレジスタDに転送する。	
演算内容	$n \cdot \text{fileN} \rightarrow D$	
nの使用範囲	000000~177777(8)	
Nの使用範囲	0~3, 10~2C(H)	
Dの使用範囲	コ00000~コ15777 : @コ00000~@コ15774 コ20000~コ75777 : @コ20000~@コ75774 b00000~b17777 : @b00000~@b17774 b20000~b37777 : @b20000~@b37774 090000~997777 : @090000~@997774 E00000~E77777 : @E00000~@E77774 ファイル1 000000~037777 : ファイル1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	nの内容	不変
	Dの内容	レジスタnの内容
	フラグ	不変

(解説)



命 令	
STR	05000
F-102	000536 file1 19003

入力条件05000がOFF→ONの変化時に、ファイル1のレジスタ000536の内容をレジスタ19003に転送します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Nとnの関係

N	nの使用範囲
0	000000~035777
1	000000~037777
2	000000~177777
3	000000~177777
10~2C	000000~177777

参考 下記のF命令は働きが類似しています。
F-102、F-102w、F-176

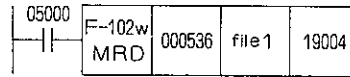
**F-102w
MRD**

直接指定アドレスのレジスタからの読出(1ワード)

(この命令はJW-31CUH/H1ではプログラムできません。)

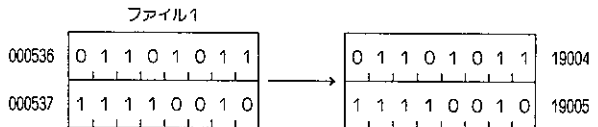
シンボル		
機能	ファイルNのファイルアドレスn、n+1のレジスタの内容をレジスタD、D+1に転送する。	
演算内容	n・fileN、n+1・fileN→D、D+1	
nの使用範囲	000000~177776(8)	
Nの使用範囲	0~3, 10~2C(H)	
Dの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 77M1 000000~037776 : 77M1 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	n+1の内容	不変
	Dの内容	レジスタnの内容
	D+1の内容	レジスタn+1の内容
	フラグ	不変

(解説)



命 令	
STR	05000
F-102w	000536
	file1
	19004

入力条件05000がOFF→ONの変化時に、ファイル1のレジスタ000536、000537の内容(1ワードデータ)をレジスタ19004、19005に転送します。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- nには必ず偶数アドレスを設定してください。(000003、177777等は禁止)
- Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- Nとnの関係

N	nの使用範囲
0	000000~035776
1	000000~037776
2	000000~177776
3	000000~177776
10~2C	000000~177776

参考 下記のF命令は働きが類似しています。
F-102、F-102w、F-176

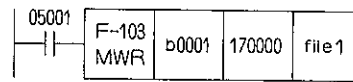
**F-103
MWR**

直接指定アドレスのレジスタへの書込(1バイト)

(この命令はJW-31CUH/H1ではプログラムできません。)

シンボル	$\text{F-103 MWR } S \ n \ \text{fileN}$	
機能	レジスタSの内容をファイルNのファイルアドレスnのレジスタに転送する。	
演算内容	$S \rightarrow n \cdot \text{fileN}$	
Sの使用範囲	$\text{C}00000 \sim \text{C}15777 \quad \text{!} \quad \text{C}00000 \sim \text{C}15777$ $\text{C}20000 \sim \text{C}75777 \quad \text{!} \quad \text{C}20000 \sim \text{C}75777$ $\text{b}00000 \sim \text{b}17777 \quad \text{!} \quad \text{b}00000 \sim \text{b}17777$ $\text{b}20000 \sim \text{b}37777 \quad \text{!} \quad \text{b}20000 \sim \text{b}37777$ $\text{O}90000 \sim \text{O}99777 \quad \text{!} \quad \text{O}90000 \sim \text{O}99777$ $\text{E}00000 \sim \text{E}77777 \quad \text{!} \quad \text{E}00000 \sim \text{E}77777$ $\text{77fileN } 000000 \sim 037777 \quad \text{!} \quad \text{77fileN } @000000 \sim @037777$	
nの使用範囲	000000~177777(8)	
Nの使用範囲	0~3, 10~2C(H)	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Sの内容	不変
	nの内容	レジスタSの内容
	フラグ	不変

〔解説〕



命 令	
STR	05001
F-103	b0001
	170000
	file1

入力条件05001がOFF→ONの変化時にレジスタb0001の内容をファイル1のレジスタ170000に転送します。



- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Nとnの関係

N	nの使用範囲
0	000000~035777
1	000000~037777
2	000000~177777
3	000000~177777
10~2C	000000~177777

参考 下記のF命令は働きが類似しています。
F-103、F-103w、F-177

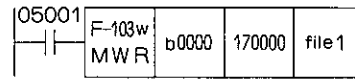
F-103w
MWR

直接指定アドレスのレジスタへの書込(1ワード)

(この命令はJW-31CUH/H1ではプログラムできません。)

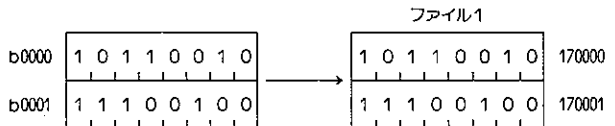
シンボル	$\overline{\text{F-103w MWR}}$ S n fileN	
機能	レジスタS、S+1の内容をファイルNのファイルアドレスn、n+1のレジスタに転送する。	
演算内容	S、S+1 → n・fileN、n+1・fileN	
Sの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 ファイル000000~037776 ファイル000000~@037774	
nの使用範囲	000000~177776(8)	
Nの使用範囲	0~3, 10~2C(H)	
演算条件	入力信号の立上り(OFF→ON)	
演算後	S、S+1の内容	不変
	nの内容	レジスタSの内容
	n+1の内容	レジスタS+1の内容
	フラグ	不変

(解説)



命令	
STR	05001
F-103w	b0000 170000 file1

入力条件05001がOFF→ONの変化時にレジスタb0000、b0001の内容(1ワードデータ)をファイル1のレジスタ170000、170001に転送します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Sには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- nには必ず偶数アドレスを設定してください。
(000003、177777等は禁止)
- Nとnの関係

N	nの使用範囲
0	000000~035776
1	000000~037776
2	000000~177776
3	000000~177776
10~2C	000000~177776

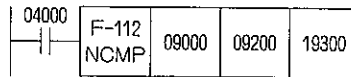
参考 下記のF命令は働きが類似しています。
F-103、F-103w、F-177

F-112
NCMP

nバイト一括比較(1バイトレジスタ間)

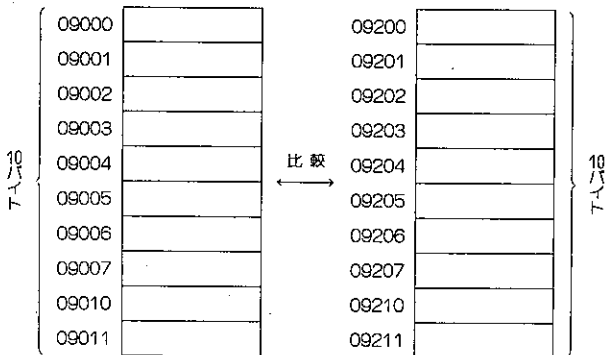
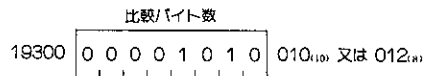
シンボル	— F-112 NCMP S ₁ S ₂ S ₃					
演 能	レジスタS ₁ からS ₃ の内容で示されるバイト数のデータと、レジスタS ₂ からS ₃ の内容で示されるバイト数のデータを大小比較する。					
演算内容	比較結果→フラグ					
S ₁ の使用範囲	コ0000~コ1577 コ2000~コ7577 b0000~b1777 b2000~b3777 09000~99777 E0000~E7777 771#1 000000~037777	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774				
S ₂ の使用範囲	コ0000~コ1577 コ2000~コ7577 b0000~b1777 b2000~b3777 09000~99777 E0000~E7777 771#1 000000~037777	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774				
S ₃ の使用範囲	コ0000~コ1577 コ2000~コ7577 b0000~b1777 b2000~b3777 09000~99777 E0000~E7777 771#1 000000~037777	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774				
演算条件	入力信号がONの時 (OFF→ONの変化時に限定されない)					
演算	S ₁ の内容	不変				
	S ₂ の内容	不変				
	S ₃ の内容	不変				
後ラ	フ	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
	ラ	S ₁ ~S ₁ +(S ₃)-1>S ₂ ~S ₂ +(S ₃)-1	0	0	0	1
	グ	S ₁ ~S ₁ +(S ₃)-1=S ₂ ~S ₂ +(S ₃)-1	1	0	0	1
	グ	S ₁ ~S ₁ +(S ₃)-1<S ₂ ~S ₂ +(S ₃)-1	0	1	0	0

(解 説)



命 令	
STR	04000
F-112	09000
	09200
	19300

レジスタ19300の内容が010₍₁₀₎(012₍₈₎)とすると、入力条件04000がONの時に、レジスタ09000~09011の内容(10/バイト)と09200~09211の内容(10/バイト)を大小比較して、その結果をノンキャリーフラグ(07354)、キャリーフラグ(07356)ゼロフラグ(07357)に設定します。



- 09000~09011 > 09200~09211 → ノンキャリー
- 09000~09011 = 09200~09211 → ノンキャリー
ゼロ
- 09000~09011 < 09200~09211 → キャリー

レジスタS₃の内容を"000"とすると256バイトの比較となります。

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合必ず偶数アドレスを設定してください。
(@コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- 入力信号がONの間、毎スキャンサイクル演算が実行されます。(9・7ページ「演算実行条件」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。
(9・7ページ「データ処理命令とフラグ」参照)

- S₃の内容は000~377₍₈₎です。000にすると256バイトとなります。
- 本命令は倍長演算できません。

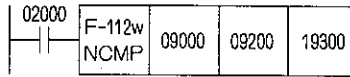
参考 下記のF命令は働きが類似しています。
F-12、F-12w、F-12d、F-112、F-112w
Fc12、Fc12w

F-112w
NCMP

nワード一括比較

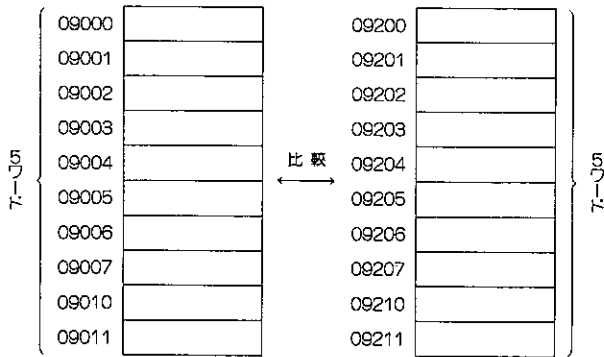
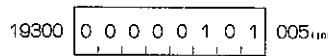
シンボル	— F-112w NCMP S ₁ S ₂ S ₃				
機能	レジスタS ₁ からS ₃ の内容で示されるワード数のデータと、レジスタS ₂ からS ₃ の内容で示されるワード数のデータを大小比較する。				
演算内容	比較結果→フラグ				
S ₁ の使用範囲	コ0000~コ1576 コ2000~コ7576 b0000~b1776 b2000~b3776 09000~99776 E0000~E7776 771#1 000000~037776	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774			
S ₂ の使用範囲	コ0000~コ1576 コ2000~コ7576 b0000~b1776 b2000~b3776 09000~99776 E0000~E7776 771#1 000000~037776	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774			
S ₃ の使用範囲	コ0000~コ1576 コ2000~コ7576 b0000~b1776 b2000~b3776 09000~99776 E0000~E7776 771#1 000000~037776	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774			
演算条件	入力信号がONの時 (OFF→ONの変化時に限定されない)				
演算	S ₁ 、S ₁ +1の内容	不変			
	S ₂ 、S ₂ +1の内容	不変			
	S ₃ 、S ₃ +1の内容	不変			
後ラゲ	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
	S _r ~S _r +2(S _r)-1>S _r ~S _r +2(S _r)-1	0	0	0	1
	S _r ~S _r +2(S _r)-1=S _r ~S _r +2(S _r)-1	1	0	0	1
	S _r ~S _r +2(S _r)-1<S _r ~S _r +2(S _r)-1	0	1	0	0

(解説)



命令	
STR	02000
F-112w	09000
	09200
	19300

レジスタ19300の内容が005₍₁₀₎(005₍₈₎)とすると、入力条件02000がONの時に、レジスタ09000~09011の内容(5ワード)と09200~09211の内容(5ワード)を大小比較して、その結果をノンキャリーフラグ(07354)、キャリーフラグ(07356)ゼロフラグ(07357)に設定します。



09000~09011 > 09200~09211 → ノンキャリー
 09000~09011 = 09200~09211 → ノンキャリー
 09000~09011 < 09200~09211 → キャリー

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- S₁、S₂には必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- 入力信号がONの間、毎スキャンサイクル演算が実行されます。(9・7ページ「演算実行条件」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)

- S₃の内容は000~377₍₈₎です。000にすると256ワードとなります。
- 本命令は倍長演算できません。

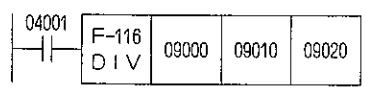
参考 下記のF命令は働きが類似しています。
 F-12、F-12w、F-12d、F-112、F-112w
 Fc12、Fc12w

F-116
DIV

レジスタ(BCD8桁)とレジスタ(BCD8桁)の除算
(DIVide) (商は整数部8桁、小数部4桁)

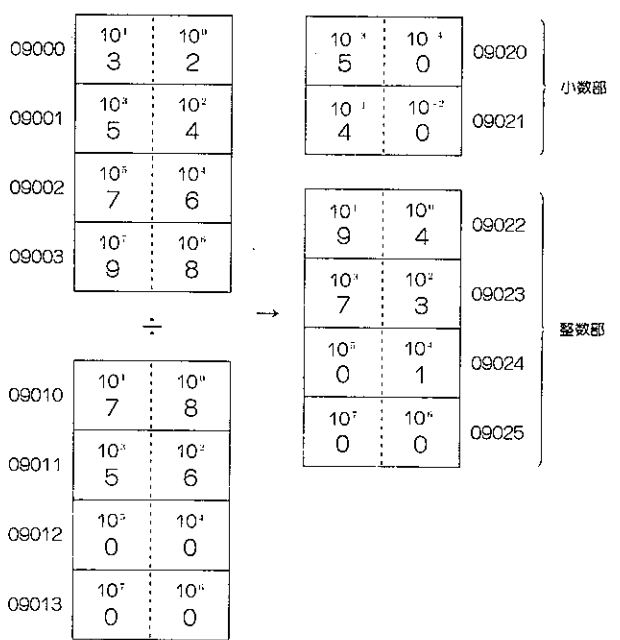
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">F-116 DIV</td> <td style="text-align: center;">S₁</td> <td style="text-align: center;">S₂</td> <td style="text-align: center;">D</td> </tr> </table>				F-116 DIV	S ₁	S ₂	D
F-116 DIV	S ₁	S ₂	D					
機能	レジスタS ₁ ~S ₁ +3の内容(BCD8桁)をレジスタS ₂ ~S ₂ +3の内容(BCD8桁)で除算し、レジスタDからの6バイトに小数4桁の商と整数8桁の商を格納する。							
演算内容	(S ₁ ~S ₁ +3)÷(S ₂ ~S ₂ +3)→D~D+5							
S ₁ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 77#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 77#1 @000000~@037774						
S ₂ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 77#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 77#1 @000000~@037774						
Dの使用範囲	コ0000~コ1572 コ2000~コ7572 b0000~b1772 b2000~b3772 09000~99772 E0000~E7772 77#1 000000~037772	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 77#1 @000000~@037774						
演算条件	入力信号の立上り(OFF→ON)							
演算	S ₁ ~S ₁ +3の内容	不変						
	S ₂ ~S ₂ +3の内容	不変						
算	D~D+1の内容	演算結果の商 (小数部4桁)	レジスタS ₁ ~S ₁ +3、 S ₂ ~S ₂ +3の内容が BCDコードでない					
	D+2~D+5の内容	演算結果の商 (整数部8桁)	時、S ₂ ~S ₂ +3の内容 が00の時不変					
後	フラグ	レジスタS ₁ ~S ₁ +3、 S ₂ ~S ₂ +3の内容 BCDコード ●BCDコードでない時 ●S ₂ の内容が00の時	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354		
		0	0	0	0	1		

(解説)



命 令	
STR	04001
F-116	09000
	09010
	09020

入力条件04001がOFF→ONの変化時に、レジスタ09000~09003のBCD8桁をレジスタ09010~09013のBCD8桁で除算をし、レジスタ09020からの2バイトに小数部4桁の商を入れ、09022~09025に整数部8桁を格納します。



上記の演算は98765432÷5678=17394.4050を示しています。

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)
- S₁~S₁+3、S₂~S₂+3の内容がBCDコード以外の数値の場合、エラーフラグ(07355)がONし、除算は実行しません。(D~D+5の内容は不変です。)

- 分子<分母(S₁~S₁+3<S₂~S₂+3)の時、演算結果の商(D+2~D+5の内容)は0となります。D、D+1は小数点5桁以下は切り捨てた値になります。
- S₁、S₂、Dには必ず偶数アドレスを設定してください。(コ0001、19003等は禁止)

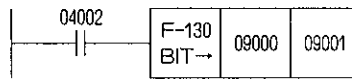
参考 下記のF命令は働きが類似しています。
F-16、F-16w、Fc16、Fc16w、F-116

F-130
BIT→

ビット抽出(間接指定)

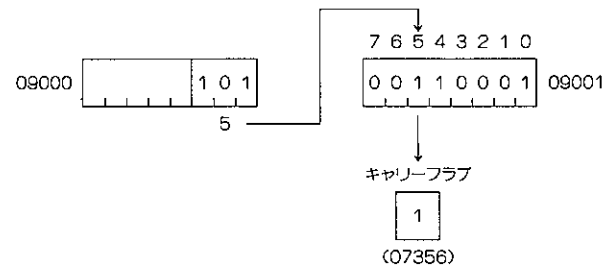
シンボル	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">F-130 BIT→</td> <td style="text-align: center;">S₁</td> <td style="text-align: center;">S₂</td> </tr> </table>					F-130 BIT→	S ₁	S ₂
F-130 BIT→	S ₁	S ₂						
機能	レジスタS ₁ の内容で指定するレジスタS ₂ のビット内容をキャリーフラグ(07356)に転送する。							
演算内容	S ₂ のビット<S _i >→キャリーフラグ							
S ₁ の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774							
S ₂ の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774							
演算条件	入力信号がONの時 (OFF→ONの変化時に限定されない)							
演算後	S ₁ の内容	不変						
	S ₂ の内容	不変						
フラグ	指定ビットの状態	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354			
	0(OFF)	0	0	0	0			
	1(ON)	0	1	0	0			

(解説)



命 令	
STR	04002
F-130	09000 09001

入力条件04002がONの時、レジスタ09000の下位3ビットで指定するレジスタ09001のビット内容をキャリーフラグ(07356)に転送します。



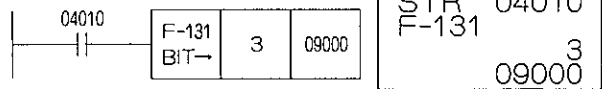
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 入力信号がONの間、毎スキャンサイクル演算を実行します。(9・7ページ「演算実行条件」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

F-131
BIT→

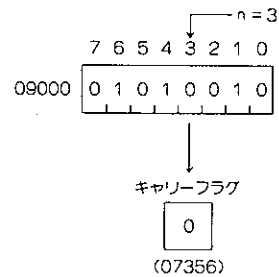
ビット抽出(直接指定)

シンボル	— F-131 BIT→ n S					
機能	レジスタSのビットnの内容をキャリアフラグに転送する。					
演算内容	Sのビットn→キャリアフラグ					
nの使用範囲	0~7					
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774					
演算内容	入力信号がONの時 (OFF→ONの変化時に限定されない)					
演算後	Sの内容	不変				
	フラグ	指定ビットの状態	ゼロ 07357	キャリア 07356	エラー 07355	ノキャリア 07354
		0(OFF)	0	0	0	0
1(ON)	0	1	0	0		

(解説)



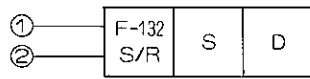
入力条件04010がONの時、レジスタ09000のビット3の内容をキャリアフラグ(07356)に転送します。



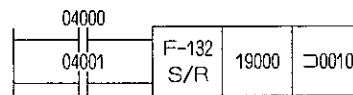
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 入力信号がONの間、毎スキャンサイクル演算を実行します。(9・7ページ「演算実行条件」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

F-132
S/R

ビットセット/リセット(間接指定)
(Set/Reset)

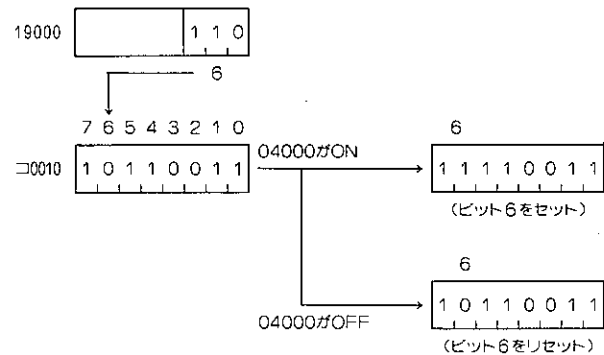
シンボル	 <p>①セット/リセット指示入力 ②入力条件</p>	
機能	レジスタSの内容で指定されるレジスタDのビットを、セット/リセット指示入力①に従ってセットまたはリセットする。	
演算内容	①の状態→Dのビット<S>	
Sの使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~997777 @090000~@997774 E00000~E77777 @E00000~@E77774 771#1 000000~037777 771#1 @000000~@037774	
Dの使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~997777 @090000~@997774 E00000~E77777 @E00000~@E77774 771#1 000000~037777 771#1 @000000~@037774	
演算条件	入力条件②がONの時 (OFF→ONの変化時に限定されない)	
演算後	Sの内容	不変
	Dの内容	指定ビットのみ変化
	フラグ	不変

(解説)



命 令	
STR	04000
STR	04001
F-132	19000
	コ0010

入力条件04001がONの時、レジスタ19000の下位3ビットで指定されるレジスタコ0010のビットを、04000がON時セット、OFF時リセットします。

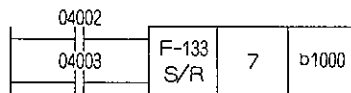


- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 入力信号がONの間、毎スキャンサイクル演算を実行します。(9・7ページ「演算実行条件」参照)

F-133 S/R ビットセット/リセット(直接指定)
(Set/Reset)

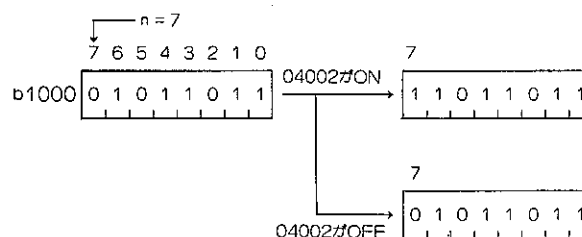
シンボル	<p>①セット/リセット指示入力 ②入力条件</p>	
機能	レジスタDのビットnをセット/リセット指示入力①に従ってセットまたはリセットする。	
演算内容	①の状態→Dのビットn	
nの使用範囲	0~7	
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~09777 @09000~@09774 E0000~E7777 @E0000~@E7774 77#1 000000~037777 77#1 @000000~@037774	
演算条件	入力条件②がONの時 (OFF→ONの変化時に限定されない)	
演算後	Dの内容	指定ビットのみ変化
	フラグ	不変

(解説)



命 令	
STR	04002
STR	04003
F-133	7
	b1000

入力条件04003がONの時、レジスタb1000のビット7を04002がON時セット、OFF時リセットします。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 入力信号がONの間、毎スキャンサイクル演算を実行します。(9・7ページ「演算実行条件」参照)

F-140 ラベルの設定 LABL (LABeL)

シンボル	F-140 LABL LBn
機能	F-141(JMP) 命令のジャンプ先 F-142(CALL) 命令のサブルーチン先 F-148(CAL+) 命令のサブルーチン先 F-151(JMP+) 命令のジャンプ先 タイマ割込のサブルーチン先 入力割込のサブルーチン先
nの使用範囲	0000~1377(8)

を示す。

(使用例)

アドレス	命 令
00000	STR 00000
00001	OR 00001
00002	OUT 04000
00003	F-140
00004	LB0002
00005	STR 04000
00006	F-00
00007	09000
00010	09001
00011	STR 00002
00012	AND 04000
00013	OUT 04001
00014	F-140
00015	LB0005

F-140は、ジャンプ先、サブルーチン先を表わすラベルで、実際に演算を実行するものではありません。従ってF-140を実行後、データメモリは保持しています。

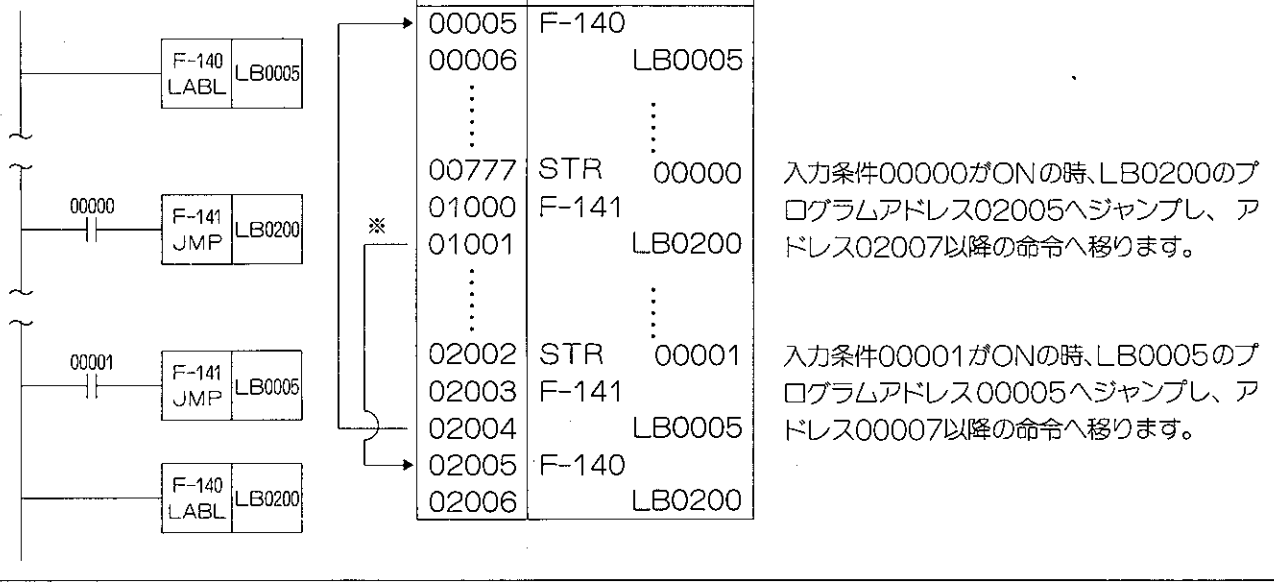
- ラベル番号 (LB0000~LB1377) は、任意に選択できますが、同じ番号を2度使用できません。
- ラベル番号 (LB1353~LB1357) はタイマ割込用に使用します。使い方はシステムメモリ #240の解説と、サブルーチンコールのリターン命令 (F-143) の使い方を参照してください。
- ラベル番号 (LB1360~LB1367) は入力割込用に使用します。使い方はシステムメモリ #241~#242の解説と、サブルーチンコールのリターン命令 (F-143) の使い方を参照してください。

F-141 JMP

ラベルへジャンプ (JuMP)

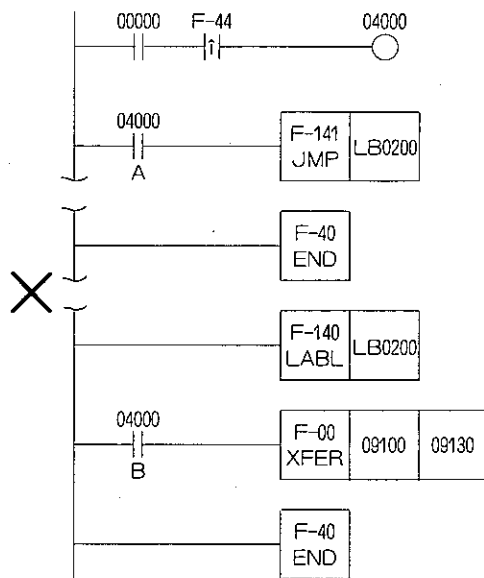
シンボル	F-141 JMP LBn
機能	プログラムの実行をLBn(F-140)のプログラムアドレスへ移す。
nの使用範囲	0000~1377(回)
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)

(解説)

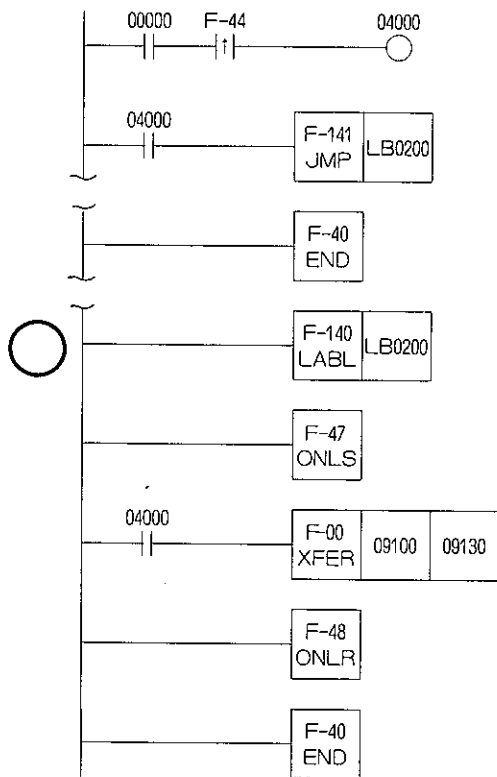


- F-141命令実行後、データメモリの内容は変化しません。
- ※部分は、F-41(JCS)、F-42(JCR)を使用しても実行しますが、F-141(JMP)を使用するとF-140(LABL)までの命令を処理しないため演算時間が短縮できます。
- F-141命令のラベル番号は任意に同一番号を何度でも使用できます。
- F-141命令は、ジャンプ先アドレスまでの命令を実行しないため、ジャンプ先アドレス以前にF-40(END命令)が存在してもF-40は無視します。
- ジャンプ先ラベル(F-140)は必ず設定してください。ジャンプ先ラベル番号が存在しないと誤動作します。
- JW-33CUH2/H3でファイル9(100000~176777)のプログラムを使用する時、F-141(JMP)命令で、ファイル8からファイル9にジャンプできます。(4・2ページ参照)
- F-140(LABL)、F-141(JMP)命令で下記のプログラム(プログラム例1、プログラム例2)を組んだとき、両方も動作は同じですが、プログラム例1についてはF-00(XFER)命令が動作しません。
- 動作は、プログラム例1、プログラム例2ともに接点04000(Aの接点)がONになるとF-141を実行し、F-141で指定したジャンプ先までジャンプします。ジャンプ後、次の接点04000(Bの接点)がONのときF-00を実行するプログラム例です。

プログラム例 1



プログラム例 2



- プログラム例1では、接点04000(Aの接点)がONになった最初のサイクルでF-141を実行し、F-141で指定したジャンプ先までジャンプし、ジャンプ後の次の接点04000(Bの接点)がONになっているためF-00を実行しますが、接点04000(Aの接点)がONになった2回目のサイクルでは接点04000がONになってジャンプしてもF-00は実行しません。これは1スキャン前のACC(アキュムレータ)の内容と現在のACCの内容がともにONのためF-00は立ち上がりと認識しないためです。

- プログラム例2では、接点04000(Aの接点)がONになった2回目のサイクルでもジャンプ後のF-00は動作を実行します。これはジャンプ後の命令をONのときに実行するようにレベル演算条件(F-47、F-48)にしているためです。以上のことにより毎演算サイクル、ジャンプ後のプログラムを実行させるにはプログラム例2のようにしてください。

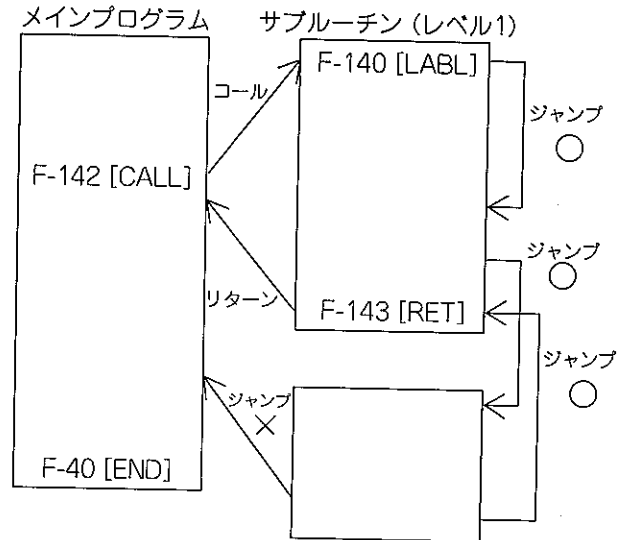
【参考】 F-00は1スキャン前のACCの内容と現在のACCに格納している内容を比較し、その結果、立ち上がりと認識したときに実行する命令です。

- (9・7ページ「演算実行条件」参照)
- ラベル番号(LB1354~LB1367)はタイマ割込(#240で設定)と入力割込(#241~#242で設定)で使用します。

●F-142 (CALL)/F-143 (RET) の多重使用 (ネスティング) に使用する場合

F-141 (JMP) は同一サブルーチン内へのジャンプ、およびジャンプ先からジャンプ元のサブルーチンに必ず処理が戻る場合に限り使用可能です。

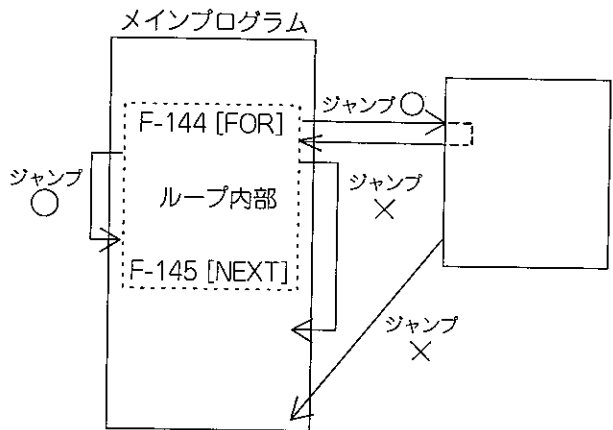
【例】



●F-144 (FOR)/F-145 (NEXT) の多重使用 (ネスティング) に使用する場合

F-141 (JMP) は同一のループ内でのジャンプ、およびジャンプ先からジャンプ元のループに必ず処理が戻る場合に限り使用可能です。

【例】

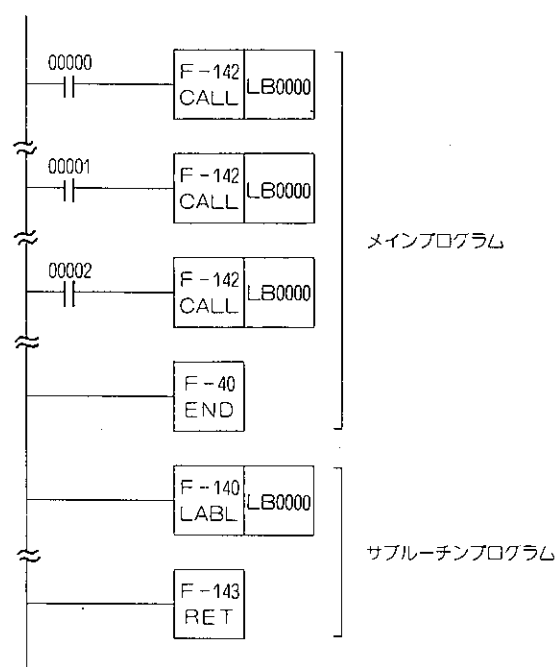


F-142 ラベルをサブルーチンコール
CALL (CALL)

F-143 サブルーチンからのリターン
RET (Return)

シンボル	F-142 CALL	LBn	F-143 RET
機能	プログラムの実行をLBn(F-140)のサブルーチンに移し、F-143命令で戻る。		
nの使用範囲	0000~1377 ^(a)		
演算条件	入力信号の立上り(OFF→ON)		

(解説)



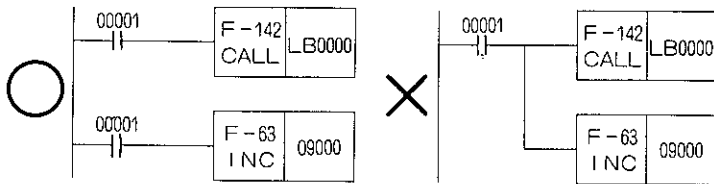
アドレス	命 令
00010	STR 00000
00011	F-142
00012	LB0000
00100	STR 00001
00101	F-142
00102	LB0000
00200	STR 00002
00201	F-142
00202	LB0000
00300	F-40
00301	F-140
00302	LB0000
00315	F-143

F-142(CALL)、F-143(RET)命令は、プログラム中で何度も実行する必要がある部分があるときに使用するとステップ数を縮めるだけでなく、プログラムの構造自体を組織化できます。

上記例の場合、入力条件00000がOFF→ONの変化時、メインプログラムからアドレス00303~00314のサブルーチンプログラムの実行に移り、F-143命令でメインプログラムのアドレス00013以降の命令を実行します。

- サブルーチンコールでは、F-148(CAL+)やF-149(RETC)も使用できます。
- F-143(RET)命令はタイマ割込(#240の設定)や入力割込(#241~#242の設定)でサブルーチンプログラムを作る時にも使用します。
- タイマ割込のラベル番号はLB1354~LB1357を使用します。
- 入力割込のラベル番号はLB1360~LB1367を使用します。
- 必ず、サブルーチンプログラムの前にメインプログラムを作成し、その最終アドレスにF-40(END命令)を入れてください。
- F-142は割込プログラム内には使用できません。
- 次の各命令をサブルーチンプログラムに入れることはできません。
 TMR、CNT、F-30(MCS)、F-31(MCR)、F-40(END)、F-41(JCS)、F-42(JCR)、F-44(↑↑)、F-45(↓↓)、F-47(ONLS)、F-48(ONLR)
- サブルーチンがコールされたとき、サブルーチン内の命令は、レベル演算条件(ONで実行)となります。
- F-143はF-142へ命令の実行を移行するのではなく、F-142の次のステップへ命令の実行を移行します。
- JW-33CUH2/H3でファイル9(100000~176777)のプログラムを使用する時、F-142(CALL)命令で、ファイル8からファイル9をサブルーチンプログラムとして使用できます。(4・2ページ参照)

●F-142 (CALL) 命令の次ステップは接点入力で始めてください。



1つの接点でF-142と次ステップの命令を駆動させるとF-143 (RET) 命令から戻ったとき次ステップ入力サブルーチン内の状態によって決まるため正常に動作しなくなります。

●F-142 (CALL)、F-143 (RET) 命令で右記のプログラム (プログラム例1、プログラム例2) を組んだとき、両方とも動作は同じですが、プログラム例1についてはF-63 (INC) 命令が接点00070の条件で動作を行いません。

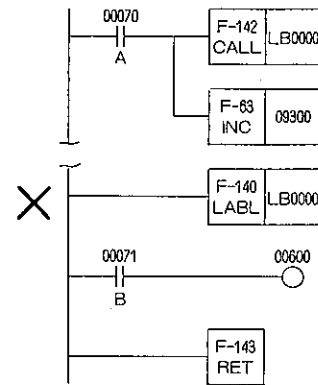
●動作は、プログラム例1、プログラム例2ともに接点00070 (Aの接点) がONになるとF-142を実行した後、F-142で指定しているサブルーチンへ命令の実行が移動します。F-143でF-142の次のステップのF-63へ戻り、F-63を実行する例です。

●プログラム例1では、接点00070 (Aの接点) がONになるとF-142を実行し、F-142で指定しているサブルーチンへ移行し、F-143まで命令を実行した後、F-142の次のステップ (F-63) へ命令の実行が移行します。このときF-63は接点00070 (Bの接点) の条件で実行します。これはF-143でF-142の次のステップへ戻ったとき、F-63の実行条件はF-63の1つ前のACC (アキュムレータ) に書き込まれた状態 (この例の場合はサブルーチン内の最後にACCに書き込まれた状態) で実行するためF-63はプログラムどおりの動作を行いません。

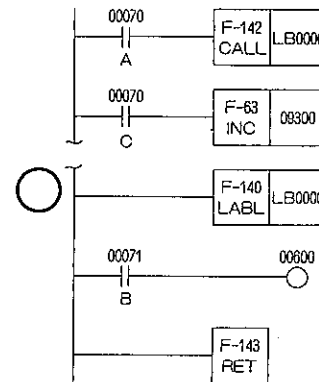
●プログラム例2では、F-143でF-142の次のステップへ戻ってもF-63はサブルーチン内の最後にACCに書き込まれた状態で行うのではなく、戻った後の接点00070 (Cの接点) の条件で実行するためプログラムどおりの動作を行います。

以上のことによりF-142の次のステップは接点入力で始まるプログラム例2のようにしてください。

プログラム例1

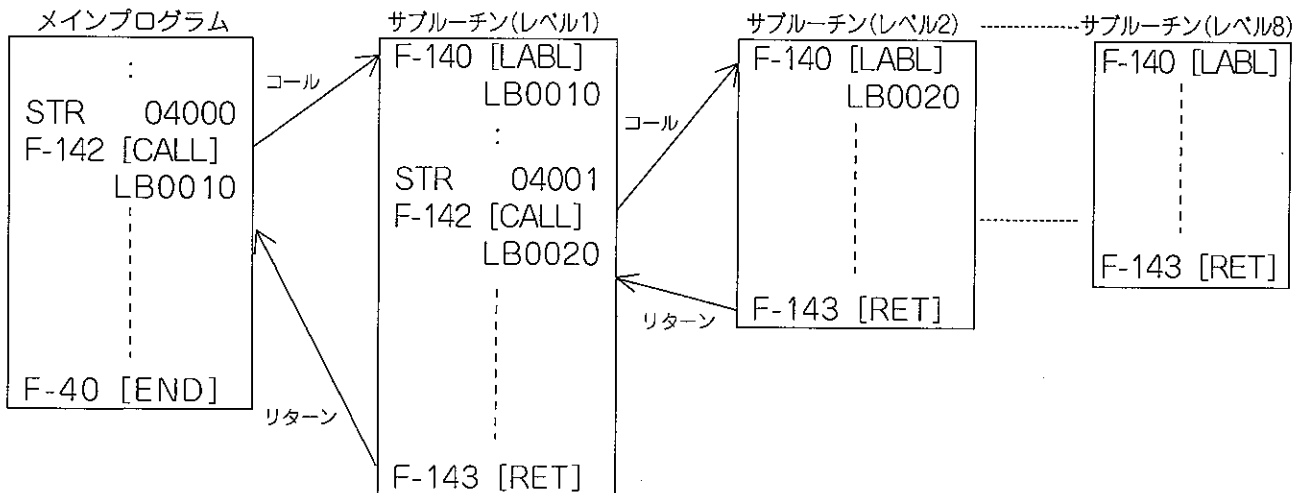


プログラム例2



●F-142 (CALL)/F-143 (RET) の多重使用 (ネスティング) は最大8レベルまで可能です。

[例]



【参考】 下記のF命令は働きが類似しています。

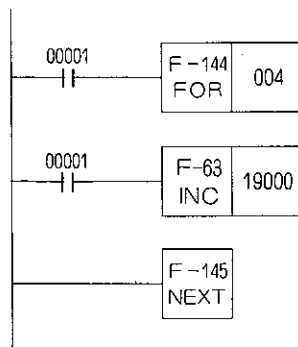
F-142、F-148 | F-143、F-149

F-144 ループ回数の設定
FOR (FOR)

F-145 ループの終了
NEXT (NEXT)

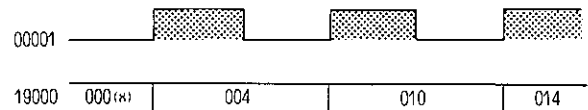
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-144</td><td style="padding: 2px;">n</td></tr><tr><td style="padding: 2px;">FOR</td><td style="padding: 2px;"></td></tr></table> <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-145</td></tr><tr><td style="padding: 2px;">NEXT</td></tr></table>	F-144	n	FOR		F-145	NEXT
F-144	n						
FOR							
F-145							
NEXT							
機能	F-144(FOR)とF-145(NEXT)間のプログラムをn回くり返す。						
nの使用範囲	000~377 ₈ (000とすると256回くり返す)						
演算条件	入力信号の立上り(OFF→ON)						

(解説)

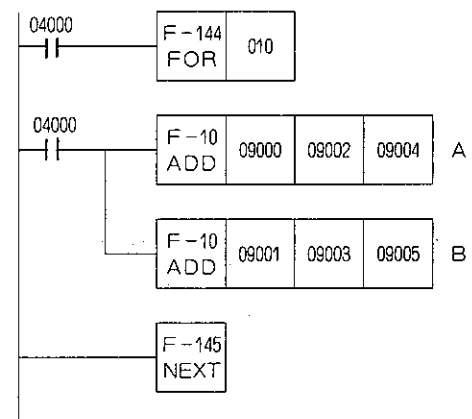


命 令	
STR	00001
F-144	004
STR	00001
F-63	19000
F-145	

入力条件00001がOFF→ONの変化時に、F-63 (INC)命令を4回実行します。



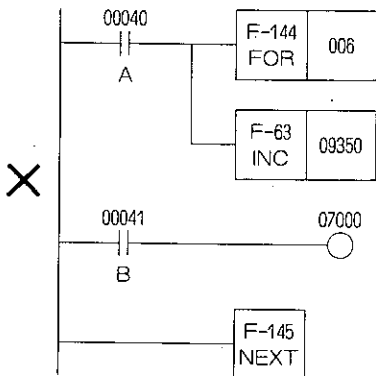
- F-144 (FOR)とF-145 (NEXT)は必ず一対で使用してください。
- F-144 (FOR)命令を実行時、F-144 (FOR)とF-145 (NEXT)の間の命令はレベル演算条件(ONで実行)となります。
- 次の各命令をF-144 (FOR)とF-145 (NEXT)の間に入れられません。
TMR、CNT、F-30 (MCS)、F-31 (MCR)、F-40 (END)、F-41 (JCS)、F-42 (JCR)、F-44 (↑↑)、F-45 (↓↓)、F-47 (ONLS)、F-48 (ONLR)
- 非実行時、F-144 (FOR)とF-145 (NEXT)の間のデータメモリの内容は変化しません。
- F-144 (FOR)/F-145 (NEXT)間で実行する命令数は極力少なくしてください。演算時間を考慮に入れて設計してください。



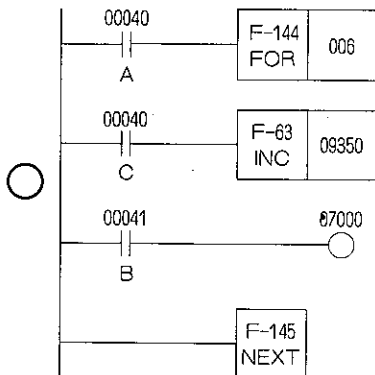
F-10命令は倍長演算が可能ですが、Bの命令の演算フラグがAの命令に影響を与えることはありません。

- F-144 (FOR)、F-145 (NEXT) 命令で下記のプログラム(プログラム例1、プログラム例2)を組んだとき、両方とも動作は同じですが、プログラム例1についてはF-63 (INC) 命令が動作しません。
- 動作は、プログラム例1、プログラム例2ともに接点00040 (Aの接点) がONになると、F-144の次のステップからF-145の間のプログラムをF-144で指定した回数だけ繰返し実行します。

プログラム例1



プログラム例2

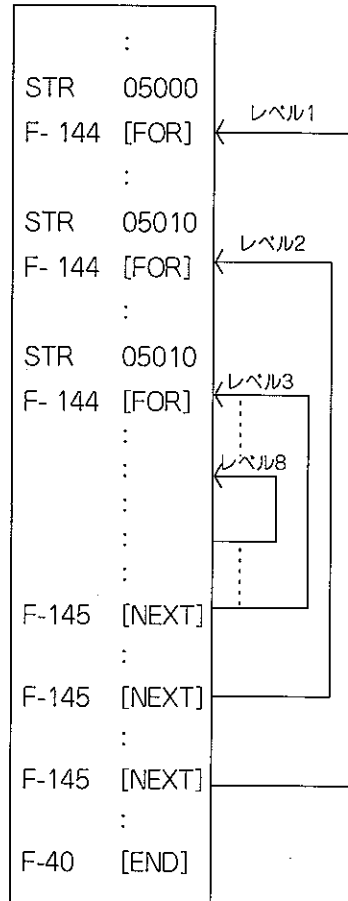


- プログラム例1では、接点00040 (Aの接点) がONになるとF-144とF-63を実行した後、F-144の次のステップからF-145の間のプログラムをF-144で指定した回数だけ繰返し実行しますが、このときF-63は接点00041 (Bの接点) の条件で実行します。これはF-145でF-144の1つ前のステップへ戻ったとき、F-63の実行条件はF-63の1つ前のACC (アキュムレータ) に書き込まれた状態 (この例の場合はF-145の1つ前にACCに書き込まれた状態) で実行するため、プログラムどおりの動作を行いません。
- プログラム例2では、F-145でF-144の1つ前のステップへ戻ってもF-63はF-145を実行する前にACCに書き込まれた状態で実行するのではなく、接点00040 (Cの接点) の条件で実行するためプログラムどおりの動作を行います。

以上のことによりF-144の次のステップは例2のように接点入力が始まるプログラムにしてください。

- ループ回数を変化させたいときF-146 (FORR) 命令を、また途中でループ演算をぬけ出すときはF-147 (EXIT) 命令をご使用いただけます。
- F-144 (FOR) / F-145 (NEXT) の多重使用 (ネスティング) は最大8レベルまで可能です。

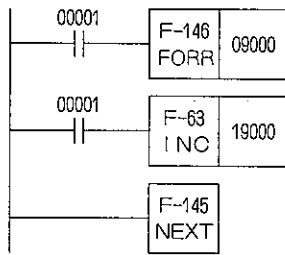
[例]



F-146 FORR (FORR) ループ回数のレジスタ設定

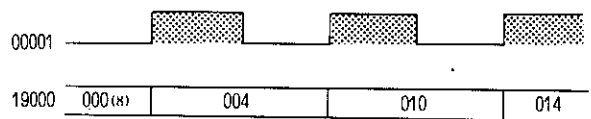
シンボル	F-146 FORR S	
機能	F-146(FORR)とF-145(NEXT)間のプログラムをSの内容の回数くり返す。	
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 77441 000000~037777 77441 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Sの内容	不変
	フラグ	不変

(解説)

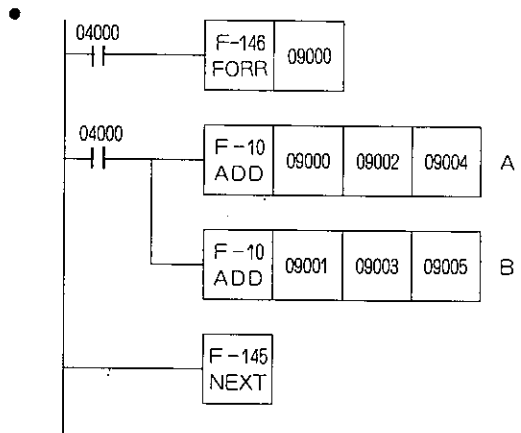


命令	
STR	00001
F-146	09000
F-63	19000
F-145	19000

入力条件00001がOFF→ONの変化時に、F-63(INC)命令をレジスタ09000の内容の回数実行します。(下記はレジスタ09000の内容が4の設定例です。)



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- Sの設定内容は000~377(8)です。000にすると256回くり返します。
- F-146(FORR)とF-145(NEXT)は必ず一対でご使用ください。
- F-146(FORR)命令を実行時、F-146(FORR)とF-145(NEXT)の間の命令はレベル演算条件(ONで実行)となります。
- 次の各命令をF-146(FORR)とF-145(NEXT)の間に入れることはできません。
TMR、CNT、F-30(MCS)、F-31(MCR)、F-40(END)、F-41(JCS)、F-42(JCR)、F-44(↑↑)、F-45(↑↑)、F-47(ONLS)、F-48(ONLR)
- F-146(FORR)/F-145(NEXT)の多重使用(ネスタイング)は最大8レベルまで可能です。使用方法はF-144(FOR)と同様です。(9・210ページ参照)
- 非実行時、F-146(FORR)とF-145(NEXT)の間のデータメモリの内容は変化しません。
- F-146(FORR)/F-145(NEXT)間で実行する命令数は極力少なくしてください。演算する場合の演算時間を考慮に入れて設計してください。



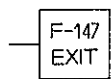
F-10命令は倍長演算が可能ですが、Bの命令の演算フラグがAの命令に影響を与えることはありません。

- その他の注意はF-144(FOR)を参照ください。
- ループ回数の途中で演算終了させるときはF-147(EXIT)をご使用ください。

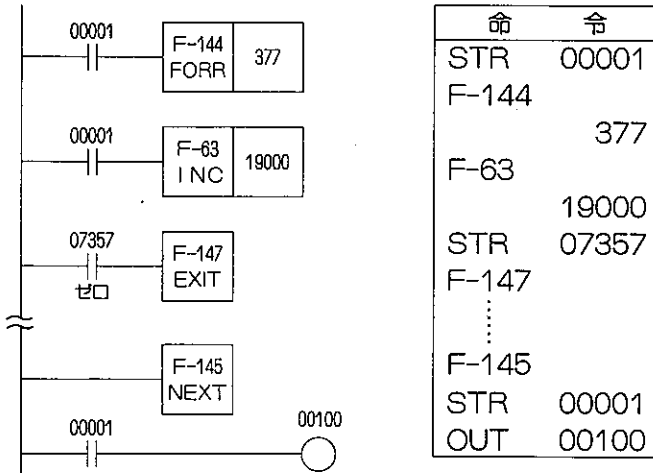
参考 下記のF命令は働きが類似しています。
F-144、F-146

**F-147
EXIT**

ループの条件終了

シンボル	
機能	F-144(FOR)又はF-146(FORR)とF-145(NEXT)間のループ回数の途中で終了させ、F-145(NEXT)の次のステップより演算を実行します。
演算条件	入力信号がOFFの時(ON→OFFの変化に限定されない)

(解説)



- 入力条件07357がOFFするとループ途中であってもF-147(EXIT)とF-145(NEXT)間の命令はNOPとして実行し、F-145でループを終了します。そしてF-145の次令より実行します。
- 入力条件07357がONのときは、F-147(EXIT)は何の影響もしません。

- F-144(FOR)とF-145(NEXT)間又はF-146(FORR)とF-145(NEXT)間でかならずお使いください。それ以外の所で使用すると演算内容が正常でなくなったりします。
- F-147は同一ループ内で複数回使用することができます。

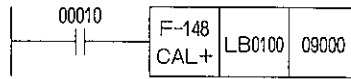
**F-148
CAL+**

**レジスタ設定ラベルをサブルーチンコール
(CAL+)**

シンボル	F-148 CAL+ LBn S	
機能	プログラムの実行をLBn とレジスタ Sの内容を加算した値のラベル (F-140)のサブルーチンに移しF-143又はF-149命令で戻る。	
LBnの使用範囲	0000~1370 ⁽⁸⁾ (最下位桁は0とみなす)	
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 77141 000000~037777 77141 @000000~@037774	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Sの内容	不変
	フラグ	不変

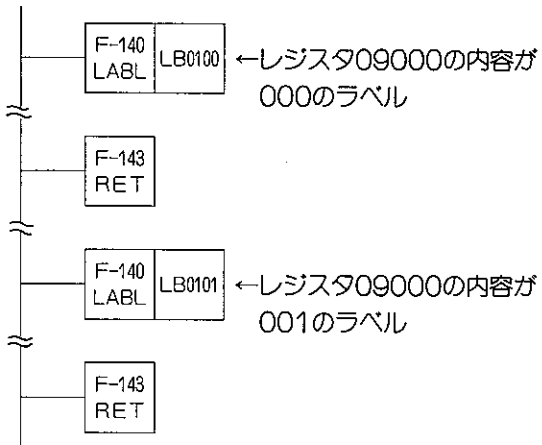
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- レジスタSの設定内容は000~007⁽⁸⁾です。000⁽⁸⁾に設定するとLBn+0=LBnとなりLBnへサブルーチンコールします。
- LBnの最下位桁の値は“0”とみなす。
(LBn0001はLB0000として処理します。)
- サブルーチンがコールされたとき、サブルーチン内の命令は、レベル演算条件(ONで実行)となります。
- 次の各命令をサブルーチンプログラムに入れることはできません。
TMR、CNT、F-30(MCS)、F-31(MCR)、F-40(END)、F-41(JCS)、F-42(JCR)、F-44(↑↑)、F-45(↑↑)、F-47(ONLS)、F-48(ONLR)
- 必ず、サブルーチンプログラムの前にメインプログラムを作成し、その最終アドレスにF-40(END命令)を入れてください。
- サブルーチン先にラベル番号が存在しないと誤動作します。
- F-148は割込プログラム内には使用できません。

(解説)



命 令	
STR	00010
F-148	LB0100
	09000

入力条件00010がOFF→ONしたときにLB0100とレジスタ09000の内容を加算した値のラベルへサブルーチンコールします。



- F-148 (CAL+) / F-143 (RET) 又はF-149 (RETC)の多重使用(ネスティング)は最大8レベルまで可能です。使用方法はF-142 (CALL)と同様です。(9・208ページ参照)
- その他の注意はF-142 (CALL)を参照ください。

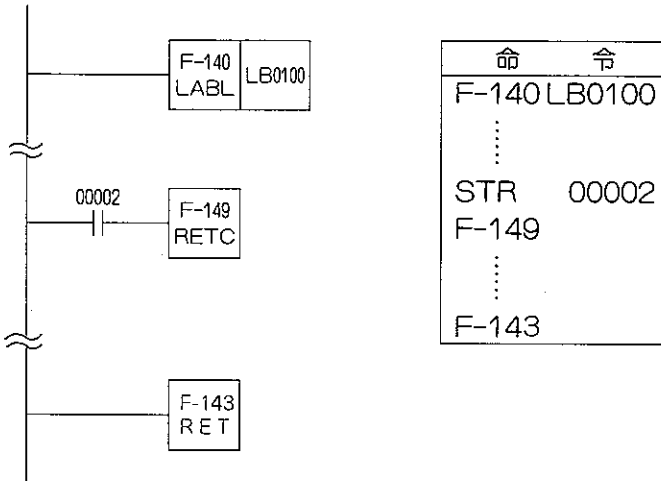
参考 下記のF命令は働きが類似しています。
F-142、F-148

F-149 RETC

サブルーチンからの条件終了

シンボル	F-149 RETC
機能	F-142(CALL)又はF-148(CAL@)でコールしたサブルーチン内から強制的にメインプログラムに戻る。
演算条件	入力信号がOFFのとき(ON→OFFの変化に限定されない)

(解説)



- 入力条件00002がOFFするとF-149(RETC)とF-143(RET)間の命令はNOPとして実行し、F-143でメインプログラムに戻ります。
- 入力条件00002がONのときはF-149(RETC)は何の影響もありません。

- F-140 (LABL) と-143 (RET) 間でかならずお使いください。それ以外の所で使用すると演算が正常に実行されずに誤動作します。
- F-149は同一サブルーチン内で複数回使用できません。

**F-151
JMP+**

**レジスタ設定ラベルへジャンプ
(JuMP+)**

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-151 JMP+</td> <td style="padding: 2px;">LBn</td> <td style="padding: 2px;">S</td> </tr> </table>		F-151 JMP+	LBn	S
F-151 JMP+	LBn	S			
機能	プログラムの実行をLBnとレジスタSの内容を加算した値のラベル(F-140)のプログラムに移る。				
LBnの使用範囲	0000~1370(最下位桁は0とみなす)				
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 ファイル1 000000~037777 ファイル1 @000000~@037774				
演算条件	入力信号がONの時 (OFF→ONの変化時に限定されない)				
演算後	Sの内容	不変			
	フラグ	不変			

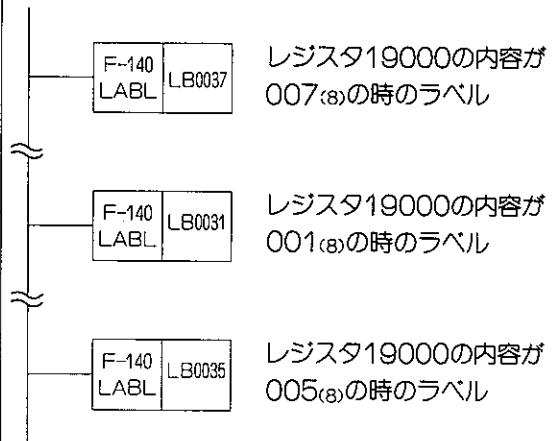
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- レジスタSの設定内容は000~007₍₈₎です。000₍₈₎に設定するとLBn+0=LBnとなりラベルLBn(F-140)のプログラムへ移ります。
- LBnの最下位桁の値は"0"とみなします。
LB0031はLB0030となります。
- F-151命令実行後、データメモリの内容は変化しません。
- F-151命令のラベル番号は任意に同一番号を何度でも使用できます。
- F-151命令は、ジャンプ先アドレスまでの命令を実行しないため、ジャンプ先アドレス以前にF-40(END命令)が存在してもF-40は無視されます。
- ジャンプ先ラベル(F-140)は必ず設定してください。ジャンプ先ラベル番号が存在しないと誤動作します。
- F-142(CALL)/F-143(RET)の多重使用(ネスティング)に使用する場合
F-151(JMP+)は同一サブルーチン内へのジャンプ、およびジャンプ先からジャンプ元のサブルーチンに必ず処理が戻る場合に限り使用可能です。使用方法はF-141(JMP)と同様です。(9・206ページ参照)

(解説)



命 令	
STR	04001
F-151	LB0030 19000

入力条件04001がONの時、LB0030とレジスタ19000の内容を加算した値のラベル(F-140)のプログラムへ移る。



- F-144(FOR)/F-145(NEXT)の多重使用(ネスティング)に使用する場合
F-151(JMP+)は同一のループ内でのジャンプおよびジャンプ先からジャンプ元のループに必ず処理が戻る場合に限り使用可能です。使用方法はF-141(JMP)と同様です。(9・206ページ参照)
- その他の注意はF-141(JMP)を参照ください。

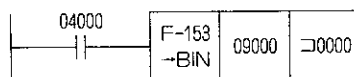
参考 下記のF命令は働きが類似しています。
F-141、F-151

F-153
→BIN

BCD(8桁)→BIN(32ビット)変換

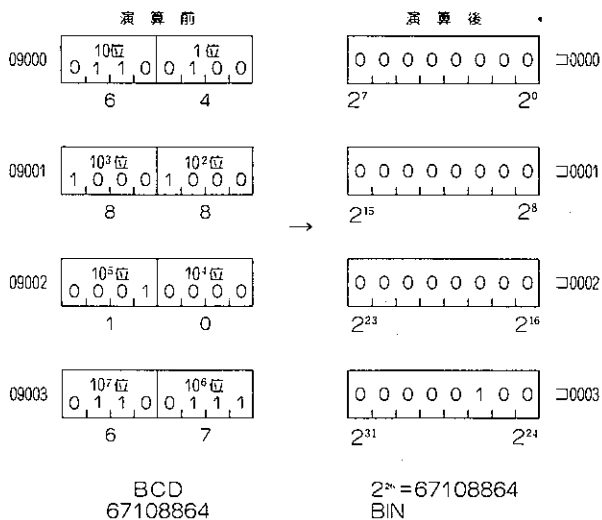
シンボル	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">F-153 →BIN</td> <td style="text-align: center;">S</td> <td style="text-align: center;">D</td> </tr> </table>					F-153 →BIN	S	D
F-153 →BIN	S	D						
機能	レジスタSを先頭とする4バイトのBCD8桁データをバイナリコードに変換し、レジスタDを先頭とする4バイトに格納する。							
演算内容	S, S+1, S+2, S+3 → D, D+1, D+2, D+3							
Sの使用範囲	<table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 77771 000000~037774 </td> <td style="width: 50%; vertical-align: top;"> @コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 77771 @000000~@037774 </td> </tr> </table>					コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 77771 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 77771 @000000~@037774	
コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 77771 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 77771 @000000~@037774							
Dの使用範囲	<table border="0" style="width: 100%;"> <tr> <td style="width: 50%; vertical-align: top;"> コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 77771 000000~037774 </td> <td style="width: 50%; vertical-align: top;"> @コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 77771 @000000~@037774 </td> </tr> </table>					コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 77771 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 77771 @000000~@037774	
コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 77771 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 77771 @000000~@037774							
演算条件	入力信号の立上り(OFF→ON)							
演算後	S, S+1, S+2, S+3の内容	不変						
	Dの内容	演算結果 (0~255)	S, S+1, S+2, S+3の内容がBCDコードでない時不変					
	D+1の内容	演算結果 (256~65025)						
	D+2の内容	演算結果 (65026~16777215)						
	D+3の内容	演算結果 (16777216~99999999)						
フラグ	S, S+1, S+2, S+3の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354			
	BCDコード	0	0	0	0			
	BCDコードでない時	0	0	1	0			

(解説)



命令	
STR	04000
F-153	09000
	00000

入力条件04000がOFF→ONの変化時に、レジスタ09000を先頭とする4バイトのBCD8桁データをバイナリコードに変換してレジスタ00000を先頭とする4バイトに変換データを格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

参考 下記のF命令は働きが類似しています。
F-03、F-03w、F-53、F-153

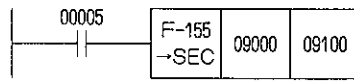
F-155
→SEC

時(BCD4桁)、分(BCD2桁)、秒(BCD2桁)→秒(BCD8桁)変換

(この命令はJW-31CUH/H1ではプログラムできません。)

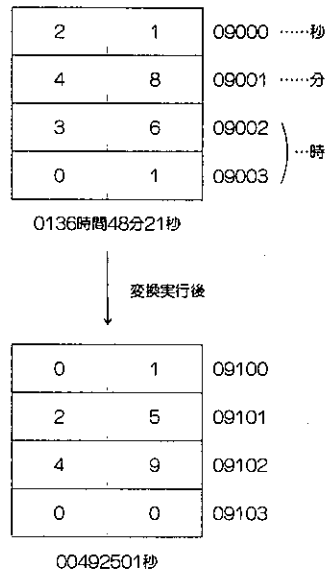
シンボル	— F-155 S D →SEC					
機能	レジスタS(秒)、S+1(分)、S+2(時下2桁)、S+3(時上2桁)の時分秒の時間データ(4バイト)を秒データに変換しD~D+3の4バイトに格納する。					
演算内容	<S(秒)、S+1(分)、S+2、S+3(時)> →D~D+3(秒) 最大値9999時59分59秒 →35999999秒					
Sの使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 774H1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 774H1 @000000~@037774				
Dの使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 774H1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 774H1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)					
演算	S~S+3の内容	不変				
	D~D+3の内容	演算結果(秒換算)	レジスタS~S+3の内容がBCDコードでない時又は分、秒が60以上のとき不変			
後	フラグ	レジスタS~S+3の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
		BCDコード	0	0	0	0
		●BCDコードでない時 ●分・秒データが60以上の時	0	0	1	0

〔解説〕



命令	
STR	00005
F-155	09000
	09100

入力条件00005がOFF→ON変化時、レジスタ09000、09001、09002、09003に格納されている時、分、秒の時間データを秒データに変換し、レジスタ09100、09101、09102、09103に格納します。(BCDコード)



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- S~S+3の内容がBCDコード以外の場合及び分、秒データが60以上のとき、エラーフラグ(07355)をONし演算を実行しません。
(D~D+3の内容は不変)

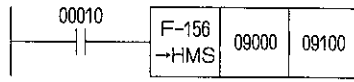
F-156
→HMS

秒(BCD8桁)→時(BCD4桁)、分(BCD2桁)、秒(BCD2桁)変換

(この命令はJW-31CUH/H1ではプログラムできません。)

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-156</td><td>S</td><td>D</td></tr><tr><td>→HMS</td><td></td><td></td></tr></table>					F-156	S	D	→HMS					
F-156	S	D												
→HMS														
機能	レジスタS~S+3に格納されている秒データ(BCD8桁)を時(BCD4桁)、分(BCD2桁)、秒(BCD2桁)をレジスタD~D+3に格納します。													
演算内容	〈S~S+3〉→D(秒)、D+1(時)、D+2とD+3(時) 最大値35999999 →9999時59分59秒													
Sの使用範囲	コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	ファイル000000~037774	ファイル@000000~@037774
Dの使用範囲	コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	ファイル000000~037774	ファイル@000000~@037774
演算条件	入力信号の立上り(OFF→ON)													
演算	S~S+3の内容	不変												
	D~D+3の内容	演算結果 (時、分、秒)	S~S+3がBCDコードでないとき又は最大値を越えるときD~D+3は不変											
後フラグ	S~S+3の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354									
	BCDコード ●BCDコード以外 ●最大値を越える時	0	0	0	0									

〔解説〕



命 令	
STR	00010
F-156	09000
	09100

入力条件00010がOFF→ON変化時レジスタ09000、09001、09002、09003に格納されている秒データを時間、分、秒に変換し、レジスタ09100、09101、09102、09103に格納します。(データはBCDコード)

7	8	09000	秒
5	6	09001	
3	4	09002	
1	2	09003	

12345678秒

変換実行後

1	8	09100	……秒
2	1	09101	……分
2	9	09102	時
3	4	09103	

3429時間21分18秒

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- S、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- S~S+3の内容がBCDコード以外、又は最大値を越える場合、エラーフラグ(07355)をONし演算を実行しません。
(D~D+3の内容は不変)

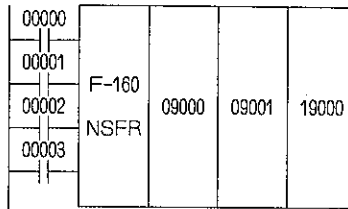
**F-160
NSFR**

**nビットシフトレジスタ
(N bit ShiFt Register)**

シンボル	<p>①シフト方向指示入力 ②データ入力 ③シフト入力 ④リセット入力</p>															
機能	レジスタDのレジスタS ₂ の内容のビットよりシフト方向指示入力①に従ってレジスタS ₁ の内容のビット領域で上位ビット、又は下位へ1ビットシフトする。															
演算内容	<ul style="list-style-type: none"> ●シフト方向指示入力①がONの場合 ●シフト方向指示入力①がOFFの場合 															
S ₁ の使用範囲	<table border="0"> <tr> <td>コ0000~コ1577</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7577</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1777</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3777</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99777</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7777</td> <td>@E0000~@E7774</td> </tr> <tr> <td>ファイル1 000000~037777</td> <td>ファイル1 @000000~@037774</td> </tr> </table>	コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	ファイル1 000000~037777	ファイル1 @000000~@037774	
コ0000~コ1577	@コ0000~@コ1574															
コ2000~コ7577	@コ2000~@コ7574															
b0000~b1777	@b0000~@b1774															
b2000~b3777	@b2000~@b3774															
09000~99777	@09000~@99774															
E0000~E7777	@E0000~@E7774															
ファイル1 000000~037777	ファイル1 @000000~@037774															
S ₂ の使用範囲	<table border="0"> <tr> <td>コ0000~コ1577</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7577</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1777</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3777</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99777</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7777</td> <td>@E0000~@E7774</td> </tr> <tr> <td>ファイル1 000000~037777</td> <td>ファイル1 @000000~@037774</td> </tr> </table>	コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	ファイル1 000000~037777	ファイル1 @000000~@037774	
コ0000~コ1577	@コ0000~@コ1574															
コ2000~コ7577	@コ2000~@コ7574															
b0000~b1777	@b0000~@b1774															
b2000~b3777	@b2000~@b3774															
09000~99777	@09000~@99774															
E0000~E7777	@E0000~@E7774															
ファイル1 000000~037777	ファイル1 @000000~@037774															
Dの使用範囲	<table border="0"> <tr> <td>コ0000~コ1577</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7577</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1777</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3777</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99777</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7777</td> <td>@E0000~@E7774</td> </tr> <tr> <td>ファイル1 000000~037777</td> <td>ファイル1 @000000~@037774</td> </tr> </table>	コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	ファイル1 000000~037777	ファイル1 @000000~@037774	
コ0000~コ1577	@コ0000~@コ1574															
コ2000~コ7577	@コ2000~@コ7574															
b0000~b1777	@b0000~@b1774															
b2000~b3777	@b2000~@b3774															
09000~99777	@09000~@99774															
E0000~E7777	@E0000~@E7774															
ファイル1 000000~037777	ファイル1 @000000~@037774															
演算条件	リセット入力④がOFFのときシフト入力③の立上り(OFF→ON)でシフト															
演算後	指令領域の内容	<ul style="list-style-type: none"> ●リセット入力④がOFFの時演算結果 ●リセット入力④がONの時は領域内ビットOFF 														
	フラグ	<table border="1"> <tr> <td>リセット入力④</td> <td>ゼロ 07357</td> <td>キャリー 07356</td> <td>エラー 07355</td> <td>ノキャリー 07354</td> </tr> <tr> <td>OFF</td> <td>1又は0</td> <td>0又は1</td> <td>0</td> <td>0又は1</td> </tr> <tr> <td>ON</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>	リセット入力④	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354	OFF	1又は0	0又は1	0	0又は1	ON	0	0	0
リセット入力④	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354												
OFF	1又は0	0又は1	0	0又は1												
ON	0	0	0	0												

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合、必ず偶数アドレスを設定してください。(コ0001、@b0173等は禁止)
- 間接アドレスの使用法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。

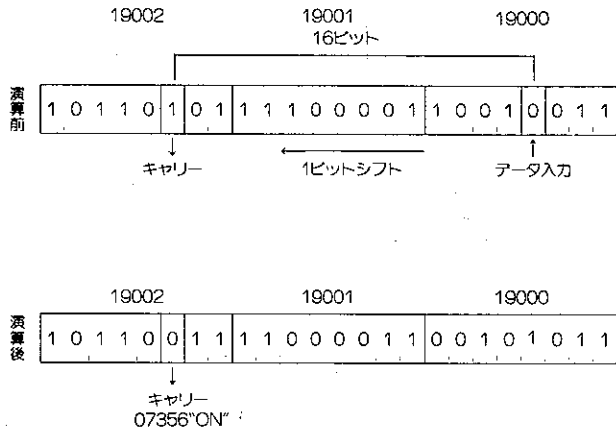
(解説)



命 令	
STR	00000
STR	00001
STR	00002
STR	00003
F-160	09000
	09001
	19000

レジスタ09000の内容が016₍₁₀₎ビット
レジスタ09001の内容が003₍₁₀₎ビット
方向指示入力

00000→ON データ入力 リセット入力
00001→ON 00003→OFF
のときシフト入力00002がOFF→ONに変化したとするとときの動作。



- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)
- S₁、S₂、Dで設定したシフト領域が、ファイル番号0のファイルアドレス001600~001677と036000以降に入らないようにしてください。(9・3ページ「ソースとデスティネーション」参照)
- S₁の内容は000~255₍₁₀₎です。000にすると256ビットとなります。
- S₂の内容は0~7₍₁₀₎です。

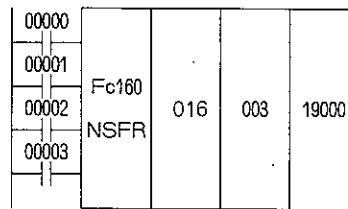
参考 下記のF命令は働きが類似しています。
F-60、F-60w、F-60d、F-160、Fc160

**Fc160
NSFR**

**nビットシフトレジスタ
(N bit ShiFt Register)**

シンボル																
機能	レジスタDのn ₂ ビットよりシフト方向指示入力①に従ってn ₁ のビット領域で上位ビット、又は下位へ1ビットシフトする。															
演算内容	<p>●シフト方向指示入力①がONの場合 (MSB) n₁ビット n₂のビット (LSB) D+1 レジスタ D キャリア 07356 ← 1ビットシフト → データ入力</p> <p>●シフト方向指示入力①がOFFの場合 (MSB) n₁ビット n₂のビット (LSB) D+1 レジスタ D データ入力 → 1ビットシフト → キャリア 07356</p>															
n ₁ の使用範囲	000~255 ₍₁₀₎ (000にすると256ビットとなる)															
n ₂ の使用範囲	000~007 ₍₁₀₎															
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 ファイル1 000000~037777 ファイル1 @000000~@037774															
演算条件	リセット入力④がOFFのときシフト入力③の立上り(OFF→ON)でシフト															
指令領域の内容	<ul style="list-style-type: none"> ●リセット入力④がOFFの時は演算結果 ●リセット入力④がONの時は領域内ビットOFF 															
フラグ	<table border="1"> <tr> <td>リセット入力④</td> <td>ゼロ 07357</td> <td>キャリア 07356</td> <td>エラー 07355</td> <td>ノキャリア 07354</td> </tr> <tr> <td>OFF</td> <td>1又は0</td> <td>0又は1</td> <td>0</td> <td>0又は1</td> </tr> <tr> <td>ON</td> <td>0</td> <td>0</td> <td></td> <td>0</td> </tr> </table>	リセット入力④	ゼロ 07357	キャリア 07356	エラー 07355	ノキャリア 07354	OFF	1又は0	0又は1	0	0又は1	ON	0	0		0
リセット入力④	ゼロ 07357	キャリア 07356	エラー 07355	ノキャリア 07354												
OFF	1又は0	0又は1	0	0又は1												
ON	0	0		0												

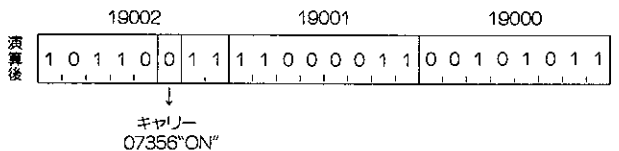
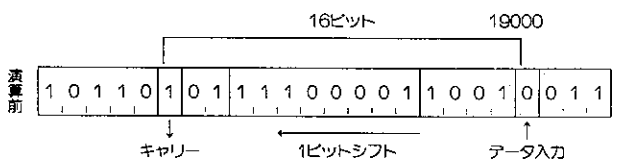
(解説)



命 令	
STR	00000
STR	00001
STR	00002
STR	00003
Fc160	016
	003
	19000

シフト領域が016₍₁₀₎ビット
データ入力が003₍₁₀₎ビット
方向指示入力

データ入力 リセット入力
00000→ON 00001→ON 00003→OFF
のときシフト入力00002がOFF→ONに変化したとすると



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合、必ず偶数アドレスを設定してください。(コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。
(9・7ページ「データ処理命令とフラグ」参照)

- Dで設定したシフト領域が、ファイル番号0のファイルアドレス001600~001677と036000以降に入らないようにしてください。
(9・3ページ「ソースとデスティネーション」参照)

参考 下記のF命令は働きが類似しています。
F-60、F-60w、F-60d、F-160、Fc160

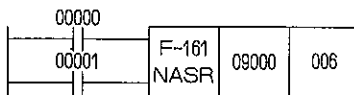
**F-161
NASR**

非同期両方向シフトレジスタ(Nバイト)
(N byte Asynchronous Shift Register)

シンボル		①シフト方向指示入力 ②シフト入力				
機能	シフト方向指示入力①に従ってレジスタDから始まる $D+(n-1)$ (①ON)又は $D-(n-1)$ (①OFF)の領域内でデータが00のレジスタの直前のレジスタの1バイトデータをデータが00のレジスタにシフトする。					
演算内容	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>●シフト方向指示①がONのとき レジスタDからレジスタD+n方向へシフトします。</p> </div> <div style="text-align: center;"> <p>●シフト方向指示がOFFのとき レジスタDからレジスタD-n方向へシフトします。</p> </div> </div>					
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 77411.000000~037777 77411.@00000~@037774					
nの使用範囲	000(8)~377(8)バイト(000にすると256/バイトとなる)					
演算条件	シフト入力②がONでシフト(OFF→ONの変化時に限定されない)					
	① ON	① OFF	① ON/OFF			
	演算前	演算後	演算前	演算後	演算前	演算後
データが0の直前のレジスタの内容	DATA1	0	DATA1	同左	DATA1	同左
データが0のレジスタの内容	0	DATA1	0	DATA2	0以外	同左
データが0の直後のレジスタの内容	DATA2	同左	DATA2	0	DATA2	同左
フラグ (演算後)	ノンキャリー 07354	D+(n-1)=0の時 1 D+(n-1)≠0の時 0	D-(n-1)=0の時 1 D-(n-1)≠0の時 0	1		
	エラー 07355	0	0	0		
	キャリー 07356	D+(n-1)=0の時 0 D+(n-1)≠0の時 1	D-(n-1)=0の時 0 D-(n-1)≠0の時 1	0		
	ゼロ 07357	0	0	0		

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 演算が実行されると、シフトしたレジスタDの内容はクリアされます。
- レジスタ領域が0でないとき、演算は実行されません。
- $D+(n-1)$ 又は $D-(n-1)$ に0以外のデータがシフトされた場合だけ、キャリーフラグ(07356)がONします。
- シフト入力②がONの間、毎スキャンサイクル演算が実行されます。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- Dとnで設定したシフト領域が、ファイル番号0のファイルアドレスの000000以下と001600~001677の範囲内及び036000を越えないようにしてください。
(9・3ページ「ソースとディスティネーション」参照)

〔解説〕

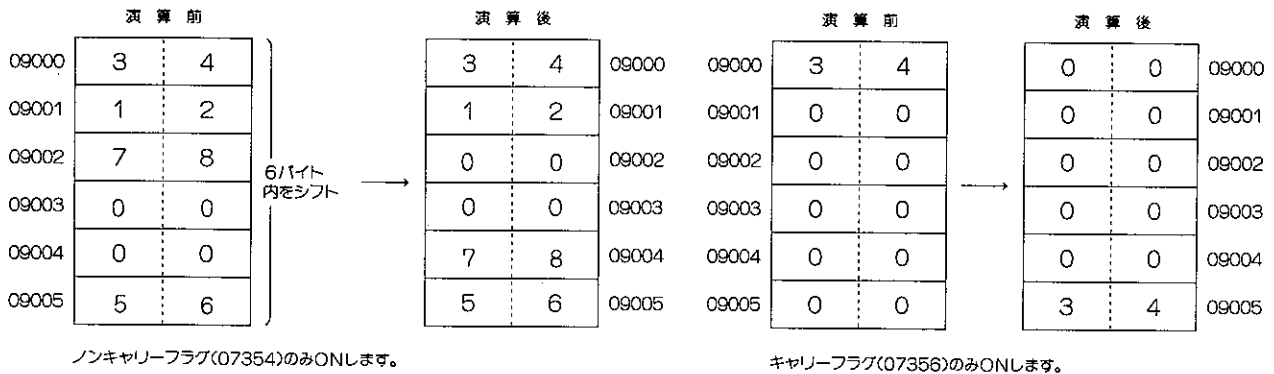


命 令	
STR	00000
STR	00001
F-161	09000
	006

00000(①) ON……………シフト指示
 00001(②) ON……………09000~09005をシフト

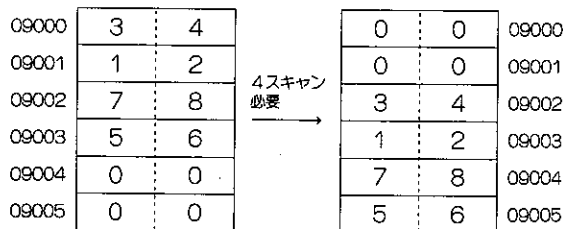
①09004、09005の内容 0000
 入力条件が上記の場合、演算結果は次のようになります。

②09001~09005の内容が 000 のときDのデータは
 1演算で09005にシフトします。

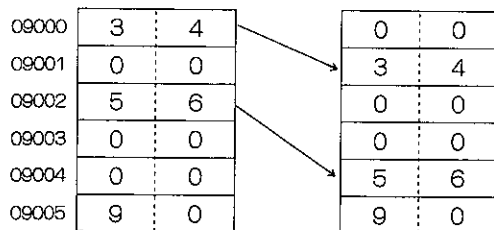


- ①の例で4バイト全てをシフトするには4スキャン必要です。

【参考】 下記のF命令は働きが類似しています。
 F-61、F-61w、F-61d、F-161、F-161w



- シフトするレジスタの直前データが“00”のとき1スキャンで複数データがシフトします。



**F-161w
NASR**

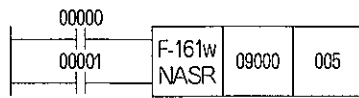
非同期両方向シフトレジスタ(Nワード)
(N byte Asynchronous Shift Register)

シンボル	①— F-161w NASR D n ②—	①シフト方向指示入力 ②シフト入力																												
機能	レジスタDから始まるD+2(n-1)(①ON)又はD-2(n+1)(①OFF)の領域内でデータが0000のレジスタの直前のレジスタの1ワードデータをデータが0000のレジスタにシフトする。																													
演算内容	<p>●シフト方向指示①がONのとき レジスタDからレジスタD+2n方向へシフトします。</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>D+1</td><td>D</td></tr> <tr><td>D+3</td><td>D+2</td></tr> <tr><td>D+5</td><td>D+4</td></tr> <tr><td>...</td><td>...</td></tr> <tr><td>D+2(n-3)+1</td><td>D+2(n-3)</td></tr> <tr><td>D+2(n-2)+1</td><td>D+2(n-2)</td></tr> <tr><td>D+2(n-1)+1</td><td>D+2(n-1)</td></tr> </table> <p style="text-align: center;">シフト ↓</p> <p>●シフト方向指示がOFFのとき レジスタDからレジスタD-2n方向へシフトします。</p> <table border="1" style="display: inline-table;"> <tr><td>D-2(n-1)+1</td><td>D-2(n-1)</td></tr> <tr><td>D-2(n-2)+1</td><td>D-2(n-2)</td></tr> <tr><td>D-2(n-3)+1</td><td>D-2(n-3)</td></tr> <tr><td>...</td><td>...</td></tr> <tr><td>D-4</td><td>D-3</td></tr> <tr><td>D-2</td><td>D-1</td></tr> <tr><td>D+1</td><td>D</td></tr> </table> <p style="text-align: center;">シフト ↑</p>		D+1	D	D+3	D+2	D+5	D+4	D+2(n-3)+1	D+2(n-3)	D+2(n-2)+1	D+2(n-2)	D+2(n-1)+1	D+2(n-1)	D-2(n-1)+1	D-2(n-1)	D-2(n-2)+1	D-2(n-2)	D-2(n-3)+1	D-2(n-3)	D-4	D-3	D-2	D-1	D+1	D
D+1	D																													
D+3	D+2																													
D+5	D+4																													
...	...																													
D+2(n-3)+1	D+2(n-3)																													
D+2(n-2)+1	D+2(n-2)																													
D+2(n-1)+1	D+2(n-1)																													
D-2(n-1)+1	D-2(n-1)																													
D-2(n-2)+1	D-2(n-2)																													
D-2(n-3)+1	D-2(n-3)																													
...	...																													
D-4	D-3																													
D-2	D-1																													
D+1	D																													
Dの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 ファイル1 000000~037776 : ファイル1 @000000~@037774																													
nの使用範囲	000(8)~377(8)ワード(000にすると256ワードとなる)																													
演算条件	シフト入力②がONでシフト(OFF→ONの変化時に限定されない)																													
	① ON	① OFF	① ON/OFF																											
	演算前	演算後	演算前	演算後	演算前	演算後																								
データが0の直前のレジスタの内容	DATA1	0	DATA1	同左	DATA1	同左																								
データが0のレジスタの内容	0	DATA1	0	DATA2	0以外	同左																								
データが0の直後のレジスタの内容	DATA2	同左	DATA2	0	DATA2	同左																								
フ ラ グ (演算後)	ノンキャリア 07354	D+2(n-1)+1, D+2(n-1)=0の時 1 D+2(n-1)+1, D+2(n-1)≠0の時 0	D-2(n-1)+1, D-2(n-1)=0の時 1 D-2(n-1)+1, D-2(n-1)≠0の時 0	1																										
	エラー 07355	0		0																										
	キャリア 07356	D+2(n-1)+1, D+2(n-1)=0の時 0 D+2(n-1)+1, D+2(n-1)≠0の時 1	D-2(n-1)+1, D-2(n-1)=0の時 0 D-2(n-1)+1, D-2(n-1)≠0の時 1	0																										
	ゼロ 07357	0		0																										

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 演算が実行されると、シフトしたレジスタD、D+1の内容はクリアされます。
- レジスタ領域内が0000でないとき、演算は実行されません。
- D+2(n-1)+1, D+2(n-1)又はD+2(n-1)+1, D+2(n-1)にデータがシフトされた場合だけ、キャリアフラグ(07356)がONします。
- シフト入力②がONの間、毎スキャンサイクル演算が実行されます。
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。
(9・7ページ「データ処理命令とフラグ」参照)
- Dとnで設定したシフト領域が、ファイル番号0のファイルアドレスの000000以下と001600~001677の範囲内及び036000を越えないようにしてください。
(9・3ページ「ソースとディスティネーション」参照)

参考 下記のF命令は働きが類似しています。
F-61、F-61w、F-61d、F-161、F-161w

〔解説〕



命 令	
STR	00000
STR	00001
F-161w	09000
	005

00000(①)ON.....09000~09011をシフト

00001(②)ON.....シフト指示

①09004、09005の内容 0000

入力条件が上記の場合、演算結果は次のようになります。

演 算 前		演 算 後	
09000	1 2 3 4	1 2 3 4	09000
09001			09001
09002	5 6 7 8	0 0 0 0	09002
09003			09003
09004	0 0 0 0	5 6 7 8	09004
09005			09005
09006	9 8 7 6	9 8 7 6	09006
09007			09007
09010	5 4 3 2	5 4 3 2	09010
09011			09011

ノンキャリーフラグ(07354)のみONします。

②09002~09011の内容が000のときDのデータは

1演算で09010、09011にシフトします。

演 算 前		演 算 後	
09000	1 2 3 4	0 0 0 0	09000
09001			09001
09002	0 0 0 0	0 0 0 0	09002
09003			09003
09004	0 0 0 0	0 0 0 0	09004
09005			09005
09006	0 0 0 0	0 0 0 0	09006
09007			09007
09010	0 0 0 0	1 2 3 4	09010
09011			09011

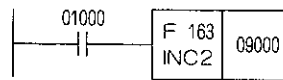
キャリーフラグ(07356)のみONします。

**F-163
INC2**

加算(+2)カウンタ(1バイト)

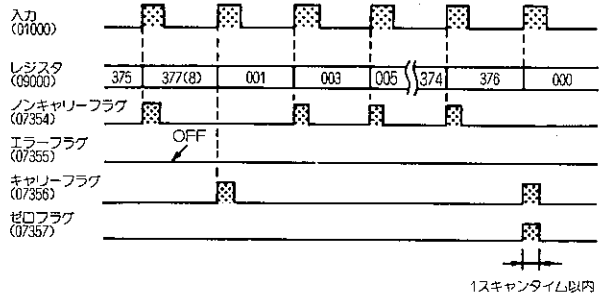
シンボル	<table border="1"> <tr> <td>F-163</td> <td>D</td> </tr> <tr> <td>INC2</td> <td></td> </tr> </table>		F-163	D	INC2																	
F-163	D																					
INC2																						
機能	レジスタDの内容(バイナリデータ)を加算(+2)カウントする。																					
演算内容	$\langle D \rangle + 2 \rightarrow D$																					
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 77471 00000~03777 77471 @00000~@03774																					
演算条件	入力信号の立上り(OFF→ON)																					
演算後	Dの内容	演算結果																				
	演算結果	<table border="1"> <tr> <th>演算結果</th> <th>ゼロ 07357</th> <th>キャリー 07356</th> <th>エラー 07355</th> <th>ノンキャリー 07354</th> </tr> <tr> <td>376₁₆→000</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>377₁₆→001</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>上記以外</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> </table>	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354	376 ₁₆ →000	1	1	0	0	377 ₁₆ →001	0	1	0	0	上記以外	0	0	0	1
	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354																	
	376 ₁₆ →000	1	1	0	0																	
377 ₁₆ →001	0	1	0	0																		
上記以外	0	0	0	1																		
フラグ																						

(解説)



命令	
STR	01000
F-163	09000

入力条件 01000 が OFF → ON の変化時、レジスタ 09000 の内容(バイナリデータ)を+2します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

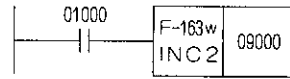
参考 下記のF命令は働きが類似しています。
F-63、F-63w、F-163、F-163w、F-263、F-263w

**F-163w
INC2**

加算(+2)カウンタ(1ワード)

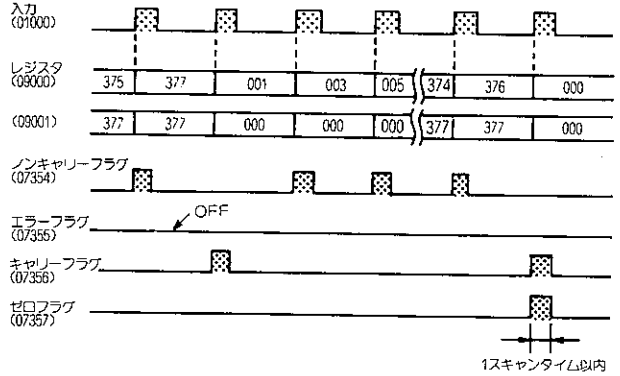
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-163w INC2</td><td style="padding: 2px;">D</td></tr></table>		F-163w INC2	D												
F-163w INC2	D															
機能	レジスタD、D+1の内容(バイナリデータ)を加算(+2)カウントする。															
演算内容	$\langle D, D+1 \rangle + 2 \rightarrow D, D+1$															
Dの使用範囲	<table style="font-size: small; border-collapse: collapse;"> <tr> <td style="padding: 2px;">C0000~C1576</td> <td style="padding: 2px;">@C0000~@C1574</td> </tr> <tr> <td style="padding: 2px;">C2000~C7576</td> <td style="padding: 2px;">@C2000~@C7574</td> </tr> <tr> <td style="padding: 2px;">b0000~b1776</td> <td style="padding: 2px;">@b0000~@b1774</td> </tr> <tr> <td style="padding: 2px;">b2000~b3776</td> <td style="padding: 2px;">@b2000~@b3774</td> </tr> <tr> <td style="padding: 2px;">09000~99776</td> <td style="padding: 2px;">@09000~@99774</td> </tr> <tr> <td style="padding: 2px;">E0000~E7776</td> <td style="padding: 2px;">@E0000~@E7774</td> </tr> <tr> <td style="padding: 2px;">771M1 000000~037776</td> <td style="padding: 2px;">771M1 @000000~@037774</td> </tr> </table>		C0000~C1576	@C0000~@C1574	C2000~C7576	@C2000~@C7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771M1 000000~037776	771M1 @000000~@037774
C0000~C1576	@C0000~@C1574															
C2000~C7576	@C2000~@C7574															
b0000~b1776	@b0000~@b1774															
b2000~b3776	@b2000~@b3774															
09000~99776	@09000~@99774															
E0000~E7776	@E0000~@E7774															
771M1 000000~037776	771M1 @000000~@037774															
演算条件	入力信号の立上り(OFF→ON)															
演算後	Dの内容	演算結果(下位)														
	D+1の内容	演算結果(上位)														
フラグ	演算結果	ゼロ 07357														
		キャリー 07356														
		エラー 07355														
		ノンキャリー 07354														
	17776-000000	1														
	17777-000001	0														
	上記以外	0														

(解説)



命令	
STR	01000
F-163w	09000

入力条件01000がOFF→ONの変化時、レジスタ09000、09001の内容(16ビットのバイナリデータ)を+2します。



- C0734~C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(C0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

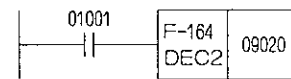
参考 下記のF命令は働きが類似しています。
F-63、F-63w、F-163、F-163w、F-263、F-263w

**F-164
DEC2**

減算(-2)カウンタ(1バイト)

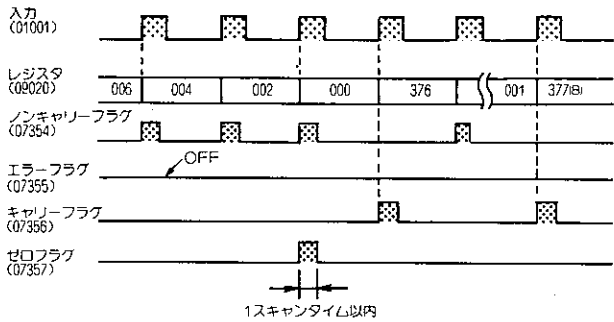
シンボル						
機能	レジスタDの内容(バイナリデータ)を減算(-2)カウントする。					
演算内容	〈D〉-2→D					
Dの使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 771M1 000000~037777 : 771M1 @000000~@037774					
演算条件	入力信号の立上り(OFF→ON)					
演算後	Dの内容	演算結果				
	フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
		002→000	1	0	0	1
		001→377 ^(M) 000→376 ^(M)	0	1	0	0
上記以外	0	0	0	1		

(解説)



命令	
STR	01001
F-164	09020

入力条件01001がOFF→ONの変化時、レジスタ09020の内容(バイナリデータ)を-2します。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

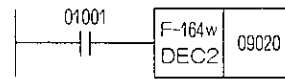
参考 下記のF命令は働きが類似しています。
F-64、F-64w、F-164、F-164w、F-264、F-264w

**F-164w
DEC2**

減算(-2)カウンタ(1ワード)

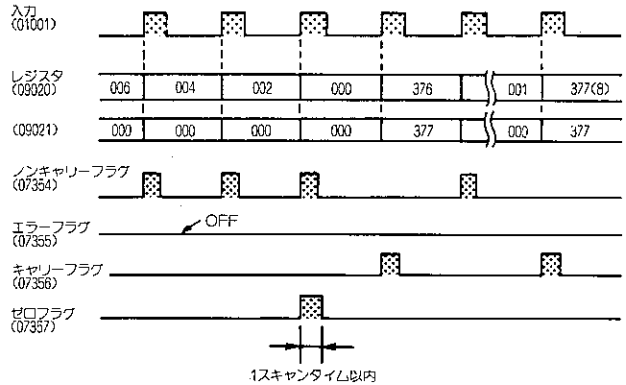
シンボル	F-164w DEC2 D					
機能	レジスタD、D+1の内容(バイナリデータ)を減算(-2)カウントする。					
演算内容	$\langle D, D+1 \rangle - 2 \rightarrow D, D+1$					
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 771#1 000000~037776 771#1 @000000~@037774					
演算条件	入力信号の立上り(OFF→ON)					
演算後	Dの内容	演算結果(下位)				
	D+1の内容	演算結果(上位)				
フラグ		演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
		000002-000000	1	0	0	1
		000001-177777 000000-177776	0	1	0	0
		上記以外	0	0	0	1

(解説)



命令	
STR	01001
F-164w	09020

入力条件01001がOFF→ONの変化時、レジスタ09020、09021の内容(16ビットのバイナリデータ)を-2します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

参考 下記のF命令は働きが類似しています。
F-64、F-64w、F-164、F-164w、F-264、F-264w

F-170 INS

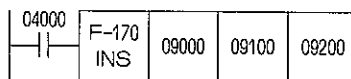
データの挿入(1バイト) (INSert)

(この命令はJW-31CUH/H1ではプログラムできません。)

シンボル	F-170 INS			S	D ₁	D ₂
機能	レジスタD ₁ を先頭アドレスとし、D ₂ の内容をバイト数とするシフトレジスタにおいてD ₁ +(D ₂ +1)のアドレスにレジスタSの内容を挿入する。					
演算内容						
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 ファイル 000000~037777 ファイル @000000~@037774					
D ₁ の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 ファイル 000000~037777 ファイル @000000~@037774					
D ₂ の使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 ファイル 000000~037776 ファイル @000000~@037774					
演算条件	入力信号の立上り(OFF→ON)					
演算後	Sの内容		不変			
	D ₂ 、D ₂ +1の内容		不変			
	D ₁ ~D ₁ +(D ₂)-1の内容		1バイトシフトしSの内容を挿入	終了アドレスの内容が"000"でない時又D ₂ ≤ D ₂ +1の内容の時不変		
	フラグ	D ₂ 、D ₂ +1の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
	●終了アドレス内容000			0		
	●終了アドレス内容が000でない又はD ₂ ≤ D ₂ +1の内容	0	0	1	0	

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合、必ず偶数アドレスを設定してください。(@コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。

(解説)



命令	
STR	04000
F-170	09000
	09100
	09200

- 入力条件04000がOFF→ON時に演算します。
- レジスタ09000の内容(55)が挿入データです。
- 先頭アドレスはレジスタ09100です。
- 挿入アドレスはレジスタ09100から09201の内容(005₍₈₎)を加算したアドレスです。
- 終了アドレスは09100から09200の内容(010₍₁₀₎バイト)の位置です。バイト数に先頭アドレスも含まれます。
- 終了アドレスの内容が000のとき挿入アドレスからデータを1バイトシフトし09000の内容(55)を挿入アドレスに格納します。

09000 55

09200 010₍₁₀₎ → 09100から09100を含む10バイトエリア
09201 005₍₈₎ → 09100から挿入するアドレス

上記設定の時 (09100はアドレス00に相当)

		演算前		演算後	
10 バイト (挿入アドレス)	(先頭アドレス) 09100	12		12	09100
	09101	34		34	09101
	09102	00		00	09102
	09103	78		78	09103
	09104	90		90	09104
	09105	09		55	09105
	09106	87		09	09106
	09107	65		87	09107
	09110	00		65	09110
	(終了アドレス) 09111	00		00	09111
			1バイトシフト		

- フラグの状態は、そのスキャンサイクルの中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)
- D₁、D₂で設定したシフトレジスタ領域が、ファイル番号0のファイルアドレス001600~001677と036000以降に入らないようにしてください。(9・3ページ「ソースとディスティネーション」参照)
- レジスタD₂の内容は000~377₍₈₎です。000にすると256/バイトになります。
- レジスタD₂+1の内容は000~377₍₈₎です。000は先頭アドレスになります。
- 最終アドレスのデータが000でない時又はD₂ ≤ D₂+1の内容のときエラーフラグ(07355)がONし演算しません。(D₂=000は256/バイトです。)
- 終了アドレス内容が000でないときは中間が000であっても演算しません。中間の000はデータとして処理します。

**F-170w
INS**

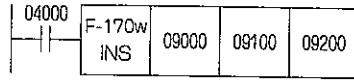
**データの挿入(1ワード)
(INSert)**

(この命令はJW-31CUH/H1ではプログラムできません。)

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="text-align: center;">F-170w INS</td><td style="text-align: center;">S</td><td style="text-align: center;">D₁</td><td style="text-align: center;">D₂</td></tr></table>				F-170w INS	S	D ₁	D ₂
F-170w INS	S	D ₁	D ₂					
機能	レジスタD ₁ を先頭アドレスとし、D ₂ の内容をワード数とするシフトレジスタにおいてD ₁ +2(D ₂ +1)、D ₁ +2(D ₂ +1)+1のアドレスにレジスタSの内容を挿入する。							
演算内容								
Sの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 ファイル000000~037776 ファイル@000000~@037774							
D ₁ の使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 ファイル000000~037776 ファイル@000000~@037774							
D ₂ の使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 ファイル000000~037776 ファイル@000000~@037774							
演算条件	入力信号の立上り(OFF→ON)							
演算後	Sの内容	不変						
	D ₁ 、D ₂ +1の内容	不変						
	D ₁ ~D ₁ +2((D ₂)-1)の内容	1ワードシフトしSの内容を挿入	終了アドレスの内容が"000"でない時又はD ₂ ≤D ₂ +1の内容時不変					
	D ₂ 、D ₂ +1の内容	ゼロ 07357	キャリー 07356	エラー 0735	ノキャリー 07354			
フラグ	●終了アドレス内容000	0	0	0	0			
フラグ	●終了アドレス内容が000でない又はD ₂ ≤D ₂ +1の内容時	0	0	1	0			

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合、必ず偶数アドレスを設定してください。(コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。

【解説】



命 令	
STR	04000
F-170w	09000
	09100
	09200

- 入力条件04000がOFF→ONの変化時に演算します。
- レジスタ09000、09001の内容(1055)が挿入データです。先頭アドレスはレジスタ09100です。
- 挿入アドレスはレジスタ09100から09200の内容(006₍₈₎ワード)を加算したアドレスです。
- 終了アドレスは09105から09200の内容(010₍₁₀₎ワード)の位置です。ワード数に先頭アドレスを含めます。
- 終了アドレスの内容が000のとき挿入アドレスからデータを1ワードシフトし09000の内容(1055)を挿入アドレスに格納します。

09000	55 (下位)	
09001	01 (上位)	
09200	006 ₍₁₀₎	→ 09100から09100を含む6ワード
09201	002 ₍₈₎	→ 09100から挿入するアドレス

上記設定の時 (09100はアドレス00に相当)

	演算前	演算後	
(先頭アドレス) 09100	12	12	09100
09101	34	34	09101
09102	00	00	09102
09103	78	78	09103
09104	90	55	09104
(挿入アドレス) 09105	09	01	09105
09106	87	90	09106
09107	65	09	09107
09110	00	87	09110
09111	10	65	09111
(終了アドレス) 09112	00	00	09112
09113	00	10	09113

1ワードシフト

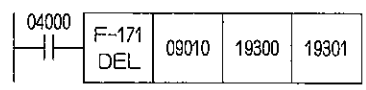
- フラグの状態は、そのスキャンサイクルの中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)
- S、D₁には必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- D₁、D₂で設定したシフトレジスタ領域が、ファイル番号0のファイルアドレス001600~001677と036000以降に入らないようにしてください。(9・3ページ「ソースとディスティネーション」参照)
- レジスタD₂の内容は000~377₍₈₎です。000にすると256ワードになります。
- レジスタD₂+1の内容は000~377₍₈₎です。000は先頭アドレスになります。
- 最終アドレスのデータが000でない時、又はD₂≤D₂+1の内容の時エラーフラグ(07355)がONし演算しません。(D₂=000は256ワード)
- 終了アドレス内容が000でないときは中間が000であっても演算しません。中間の000はデータとして処理します。

F-171 DEL データの削除(1バイト) (DELete)

(この命令はJW-31CUH/H1ではプログラムできません。)

シンボル	F-171 DEL S ₁ S ₂ S ₃																	
機能	レジスタS ₁ を先頭アドレスとしS ₂ の内容をバイト数とするシフトレジスタにおいてS ₁ +(S ₃)のアドレスデータを削除する。それ以後のデータが全てシフトします。																	
演算内容																		
S ₁ の内容	<table border="1"> <tr><td>コ0000~コ1577</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7577</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1777</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3777</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99777</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7777</td><td>@E0000~@E7774</td></tr> <tr><td>771#1 000000~037777</td><td>771#1 @000000~@037774</td></tr> </table>				コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	771#1 000000~037777	771#1 @000000~@037774
コ0000~コ1577	@コ0000~@コ1574																	
コ2000~コ7577	@コ2000~@コ7574																	
b0000~b1777	@b0000~@b1774																	
b2000~b3777	@b2000~@b3774																	
09000~99777	@09000~@99774																	
E0000~E7777	@E0000~@E7774																	
771#1 000000~037777	771#1 @000000~@037774																	
S ₂ の内容	<table border="1"> <tr><td>コ0000~コ1577</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7577</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1777</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3777</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99777</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7777</td><td>@E0000~@E7774</td></tr> <tr><td>771#1 000000~037777</td><td>771#1 @000000~@037774</td></tr> </table>				コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	771#1 000000~037777	771#1 @000000~@037774
コ0000~コ1577	@コ0000~@コ1574																	
コ2000~コ7577	@コ2000~@コ7574																	
b0000~b1777	@b0000~@b1774																	
b2000~b3777	@b2000~@b3774																	
09000~99777	@09000~@99774																	
E0000~E7777	@E0000~@E7774																	
771#1 000000~037777	771#1 @000000~@037774																	
S ₃ の内容	<table border="1"> <tr><td>コ0000~コ1577</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7577</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1777</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3777</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99777</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7777</td><td>@E0000~@E7774</td></tr> <tr><td>771#1 000000~037777</td><td>771#1 @000000~@037774</td></tr> </table>				コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	771#1 000000~037777	771#1 @000000~@037774
コ0000~コ1577	@コ0000~@コ1574																	
コ2000~コ7577	@コ2000~@コ7574																	
b0000~b1777	@b0000~@b1774																	
b2000~b3777	@b2000~@b3774																	
09000~99777	@09000~@99774																	
E0000~E7777	@E0000~@E7774																	
771#1 000000~037777	771#1 @000000~@037774																	
演算条件	入力信号の立上り(OFF→ON)																	
演算後	S ₂ の内容	不変																
	S ₃ の内容	不変																
フラグ	S ₁ ~S ₁ +(S ₂)-1の内容	削除アドレスのデータを消去し1バイトのデータシフト		S ₂ ≤ S ₃ の内容の時不変														
	S ₂ , S ₃ の内容	ゼロ 07357	キャリー 07356	エラー 07355														
	S ₂ > S ₃ の時	0	0	0														
	S ₂ ≤ S ₃ の時	0	0	1														

(解説)



命令	
STR	04000
F-171	09010
	19300
	19301

- 入力条件04000がOFF→ON時に演算します。
- 先頭アドレスはレジスタ09010です。
- 削除アドレスは09010から19301の内容(005₍₈₎)を加算したアドレスです。
- 終了アドレスは09010から19300の内容(010₍₁₀₎)の位置です。バイト数に先頭アドレスも含まれます。
- 演算行時削除アドレスのデータを消し削除アドレス以後終了アドレスまでのデータを1シフトします。終了アドレスのデータは000となります。

19300	010 ₍₁₀₎	→ 09010から09010を含む10バイトエリア
19301	005 ₍₈₎	→ 09010からアドレス位置(000のとき先頭アドレス)

	演算前	演算後	
(先頭アドレス) 09010	23	23	09010
09011	34	34	09011
09012	45	45	09012
09013	67	67	09013
09014	89	89	09014
(削除アドレス) 09015	11	22	09015
09016	22	33	09016
09017	33	44	09017
09020	44	55	09020
(終了アドレス) 09021	55	00	09021

1バイトシフト

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合、必ず偶数アドレスを設定してください。(@コ0001, @b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- フラグの状態は、そのスキャンサイクルの中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)

- S₂, S₃で設定したレジスタ範囲が、ファイル番号0のファイルアドレス001600~001677と036000以降に入らないようにしてください。(9・3ページ「ソースとディスティネーション」参照)
- レジスタS₂の内容は000~377₍₈₎です。000にすると256/バイトになります。
- レジスタS₃の内容は000~377₍₈₎です。000にすると先頭アドレスになります。
- レジスタ内容がS₂ ≤ S₃の時エラーフラグ(07355)がONし演算しません。(S₂=000は256/バイト)
- 演算完了で終了アドレスデータは000となります。

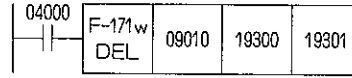
**F-171w
DEL**

**データの削除(1ワード)
(DELeTe)**

(この命令はJW-31CUH/H1ではプログラムできません。)

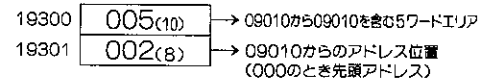
シンボル	F-171w DEL S ₁ S ₂ S ₃				
機能	レジスタS ₁ を先頭アドレスとしS ₂ の内容をワード数とするシフトレジスタにおいてS ₂ +2(S ₃)、S ₂ +2(S ₃)+1のアドレスデータを削除する。それ以後のデータが全てシフトします。				
演算内容					
S ₁ の内容	コ0000~コ1576 コ2000~コ7576 b0000~b1776 b2000~b3776 09000~99776 E0000~E7776 771#1 000000~037776	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774			
S ₂ の内容	コ0000~コ1577 コ2000~コ7577 b0000~b1777 b2000~b3777 09000~99777 E0000~E7777 771#1 000000~037777	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774			
S ₃ の内容	コ0000~コ1577 コ2000~コ7577 b0000~b1777 b2000~b3777 09000~99777 E0000~E7777 771#1 000000~037777	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774			
演算条件	入力信号の立上り(OFF→ON)				
演算後	S ₂ の内容	不変			
	S ₃ の内容	不変			
	S ₁ ~S ₁ +2(S ₃)-1の内容	削除アドレスのデータを消去し1バイトのデータシフト	S ₂ ≤ S ₃ の内容の時不変		
フラグ	S ₂ 、S ₃ の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
	S ₂ > S ₃ の時	0	0	0	0
	S ₂ ≤ S ₃ の時	0	0	1	0

(解説)



命令	
STR	04000
F-171w	09010
	19300
	19301

- 入力条件04000がOFF→ON時に演算します。
- 先頭アドレスはレジスタ09010です。
- 削除アドレスは09010から19301の内容(005₍₈₎ワード)を加算したアドレスです。
- 終了アドレスは09010から19300の内容(010₍₁₀₎ワード)の位置です。ワード数に先頭アドレスも含まれます。
- 演算行時削除アドレスのデータを消し削除アドレス以後終了アドレスまでのデータを1ワードシフトします。終了アドレスのワードデータは000となります。



	演算前	演算後	
(先頭アドレス) 09010	23	23	09010
09011	34	34	09011
09012	45	45	09012
09013	67	67	09013
(削除アドレス) 09014	89	22	09014
09015	11	33	09015
09016	22	44	09016
09017	33	55	09017
(終了アドレス) 09020	44	00	09020
09021	55	00	09021

1ワードシフト

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合、必ず偶数アドレスを設定してください。(コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- フラグの状態は、そのスキャンサイクルの中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)

- Sには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- S₂、S₃で設定したレジスタ範囲が、ファイル番号0のファイルアドレス001600~001677と036000以降に入らないようにしてください。(9・3ページ「ソースとディスティネーション」参照)
- レジスタS₂の内容は000~377₍₈₎です。000にすると256ワードになります。
- レジスタS₃の内容は000~377₍₈₎です。000にすると先頭アドレスになります。
- レジスタ内容がS₂ ≤ S₃の時エラーフラグ(07355)がONし演算しません。
- 演算完了で終了アドレスデータは000となります。

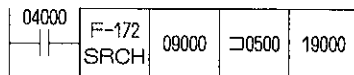
**F-172
SRCH**

データの検索(1バイト)
(Sea RCH)

(この命令はJW-31CUH/H1ではプログラムできません。)

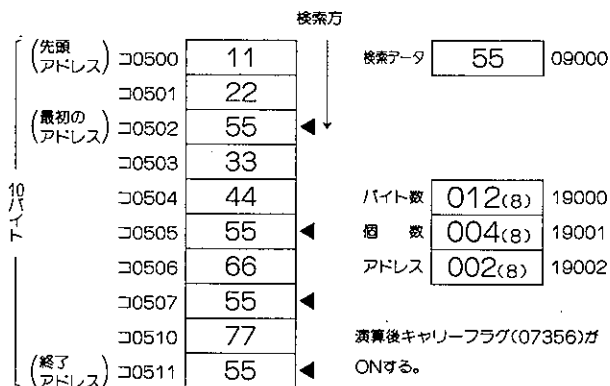
シンボル	— F-172 SRCH S D ₁ D ₂				
機能	レジスタD ₁ を先頭アドレスとしD ₂ の内容の数で指示する領域内で検索を行います。レジスタSの内容が検索データです。データ検索して最初の一致アドレス(D ₁ を基準とした)をD ₂ +2へ格納します。一致したデータ個数をD ₂ +1へ格納する。				
演算内容					
Sの使用範囲	コ00000~コ15777 : @コ00000~@コ15774 コ20000~コ75777 : @コ20000~@コ75774 b00000~b17777 : @b00000~@b17774 b20000~b37777 : @b20000~@b37774 090000~99777 : @090000~@99774 E00000~E7777 : @E00000~@E7774 ファイル000000~037777 : ファイル000000~@037774				
D ₁ の使用範囲	コ00000~コ15777 : @コ00000~@コ15774 コ20000~コ75777 : @コ20000~@コ75774 b00000~b17777 : @b00000~@b17774 b20000~b37777 : @b20000~@b37774 090000~99777 : @090000~@99774 E00000~E7777 : @E00000~@E7774 ファイル000000~037777 : ファイル000000~@037774				
D ₂ の使用範囲	コ00000~コ15775 : @コ00000~@コ15774 コ20000~コ75775 : @コ20000~@コ75774 b00000~b17775 : @b00000~@b17774 b20000~b37775 : @b20000~@b37774 090000~99775 : @090000~@99774 E00000~E7775 : @E00000~@E7774 ファイル000000~037775 : ファイル000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	Sの内容	不変			
	D ₁ ~D ₁ +(D ₂)-1の内容	不変			
演算後	D ₂ ~D ₂ +2の内容	演算結果	検索データ無しと256個の時D ₂ +1は000となる		
	フラグ	検索データ	ゼロ 07357	キャリー 07356	エラー 07355
			有	0	1
		無	1	0	0

(解説)



命 令	
STR	04000
F-172	09000
	コ0500
	19000

- 入力条件04000がOFF→ON変化時に演算します。
- 09000の内容(55)が検索データです。
- 先頭アドレスはコ0500です。
- 終了アドレスはコ0500からレジスタ19000の内容(012(8)バイト)の位置です。バイト数に先頭アドレスも含めます。
- レジスタ19001に検索個数を格納します。
- レジスタ19002に最初の一致アドレス(D₁を基準とした)を格納します。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- フラグの状態は、そのスキャンサイクルの中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)
- D₁、D₂で設定したレジスタ範囲が、ファイル番号0のファイルアドレス001600~001677と036000以降に入らないようにしてください。(9・3ページ「ソースとディスティネーション」参照)

- レジスタD₂の内容は000~377(8)です。000にすると256/バイトになります。
- レジスタD₂+1の内容は000~377(8)です。000はデータ無し又は256個のときです。キャリーフラグ(07356)でデータ有無を判別してください。
- レジスタD₂+2の内容は000~377(8)で、D₁を基準としたバイト数が入ります。000は先頭アドレスを示します。

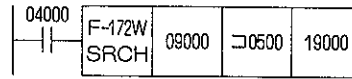
F-172w
SRCH

データの検索(1ワード)
(SeaRCH)

(この命令はJW-31CUH/H1ではプログラムできません。)

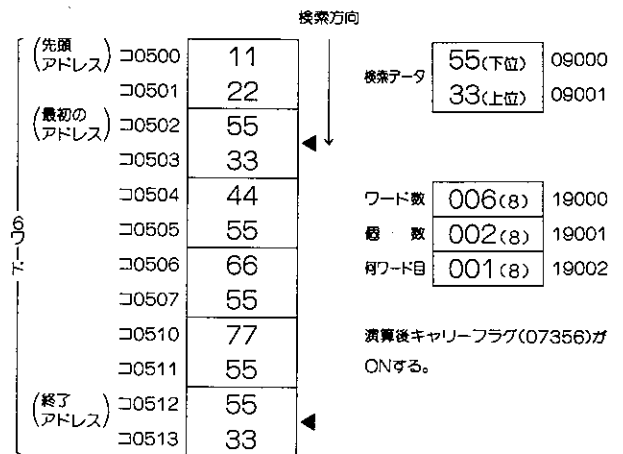
シンボル	— <table border="1" style="display: inline-table;"><tr><td>F-172W</td><td>S</td><td>D₁</td><td>D₂</td></tr></table>				F-172W	S	D ₁	D ₂										
F-172W	S	D ₁	D ₂															
機能	レジスタD ₁ を先頭アドレスとしD ₂ の内容のワード数で指示する領域内で検索を行います。レジスタS、S+1の内容が検索データです。データ検索して最初の一一致アドレス(D ₁ を基準とした)何ワード目であるかをD ₂ +2へ格納し、一致したデータ個数をD ₂ +1へ格納する。																	
演算内容																		
Sの使用範囲	<table border="1"> <tr><td>コ00000~コ15776</td><td>@コ00000~@コ15774</td></tr> <tr><td>コ20000~コ75776</td><td>@コ20000~@コ75774</td></tr> <tr><td>b00000~b17776</td><td>@b00000~@b17774</td></tr> <tr><td>b20000~b37776</td><td>@b20000~@b37774</td></tr> <tr><td>090000~997776</td><td>@090000~@997774</td></tr> <tr><td>E00000~E77776</td><td>@E00000~@E77774</td></tr> <tr><td>774#1 000000~037776</td><td>774#1 @000000~@037774</td></tr> </table>				コ00000~コ15776	@コ00000~@コ15774	コ20000~コ75776	@コ20000~@コ75774	b00000~b17776	@b00000~@b17774	b20000~b37776	@b20000~@b37774	090000~997776	@090000~@997774	E00000~E77776	@E00000~@E77774	774#1 000000~037776	774#1 @000000~@037774
コ00000~コ15776	@コ00000~@コ15774																	
コ20000~コ75776	@コ20000~@コ75774																	
b00000~b17776	@b00000~@b17774																	
b20000~b37776	@b20000~@b37774																	
090000~997776	@090000~@997774																	
E00000~E77776	@E00000~@E77774																	
774#1 000000~037776	774#1 @000000~@037774																	
D ₁ の使用範囲	<table border="1"> <tr><td>コ00000~コ15776</td><td>@コ00000~@コ15774</td></tr> <tr><td>コ20000~コ75776</td><td>@コ20000~@コ75774</td></tr> <tr><td>b00000~b17776</td><td>@b00000~@b17774</td></tr> <tr><td>b20000~b37776</td><td>@b20000~@b37774</td></tr> <tr><td>090000~997776</td><td>@090000~@997774</td></tr> <tr><td>E00000~E77776</td><td>@E00000~@E77774</td></tr> <tr><td>774#1 000000~037776</td><td>774#1 @000000~@037774</td></tr> </table>				コ00000~コ15776	@コ00000~@コ15774	コ20000~コ75776	@コ20000~@コ75774	b00000~b17776	@b00000~@b17774	b20000~b37776	@b20000~@b37774	090000~997776	@090000~@997774	E00000~E77776	@E00000~@E77774	774#1 000000~037776	774#1 @000000~@037774
コ00000~コ15776	@コ00000~@コ15774																	
コ20000~コ75776	@コ20000~@コ75774																	
b00000~b17776	@b00000~@b17774																	
b20000~b37776	@b20000~@b37774																	
090000~997776	@090000~@997774																	
E00000~E77776	@E00000~@E77774																	
774#1 000000~037776	774#1 @000000~@037774																	
D ₂ の使用範囲	<table border="1"> <tr><td>コ00000~コ15775</td><td>@コ00000~@コ15774</td></tr> <tr><td>コ20000~コ75775</td><td>@コ20000~@コ75774</td></tr> <tr><td>b00000~b17775</td><td>@b00000~@b17774</td></tr> <tr><td>b20000~b37775</td><td>@b20000~@b37774</td></tr> <tr><td>090000~997775</td><td>@090000~@997774</td></tr> <tr><td>E00000~E77775</td><td>@E00000~@E77774</td></tr> <tr><td>774#1 000000~037775</td><td>774#1 @000000~@037774</td></tr> </table>				コ00000~コ15775	@コ00000~@コ15774	コ20000~コ75775	@コ20000~@コ75774	b00000~b17775	@b00000~@b17774	b20000~b37775	@b20000~@b37774	090000~997775	@090000~@997774	E00000~E77775	@E00000~@E77774	774#1 000000~037775	774#1 @000000~@037774
コ00000~コ15775	@コ00000~@コ15774																	
コ20000~コ75775	@コ20000~@コ75774																	
b00000~b17775	@b00000~@b17774																	
b20000~b37775	@b20000~@b37774																	
090000~997775	@090000~@997774																	
E00000~E77775	@E00000~@E77774																	
774#1 000000~037775	774#1 @000000~@037774																	
演算条件	入力信号の立上り(OFF→ON)																	
演算後	S、S+1の内容	不変																
	D ₁ ~D ₁ +2((D ₂ -1)の内容)	不変																
後	D ₂ ~D ₂ +2の内容	演算結果	検索データ無しと256個の時D ₂ +1は000となる															
	フラグ	検索データ	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354												
		有	0	1	0	0												
無	1	0																

(解説)



命令	
STR	04000
F-172W	
	09000
	00500
	19000

- 入力条件04000がOFF→ON変化時に演算します。
- 09000、09001の内容(3355)が検索データです。
- 先頭アドレスは00500です。
- 終了アドレスは00500からレジスタ19000の内容(006(8)ワード)の位置です。バイト数に先頭アドレスも含まれます。
- レジスタ19001に検索個数を格納します。
- レジスタ19002に最初の一一致アドレス(D₁を基準とした)を何ワード目であるかを格納します。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- フラグの状態は、そのスキャンサイクルの中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)
- S、D₁には必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)

- D₁、D₂で設定したレジスタ範囲が、ファイル番号0のファイルアドレス001600~001677と036000以降に入らないようにしてください。(9・3ページ「ソースとディスティネーション」参照)
- レジスタD₂の内容は000~377(8)です。000にすると256ワードになります。
- レジスタD₂+1の内容は000~377(8)です。000はデータ無し又は256個のときです。キャリーフラグ(07356)でデータ有無を判別してください。
- レジスタD₂+2の内容は000~377(8)で、D₁を基準とした何ワード目であるの値が格納されます。

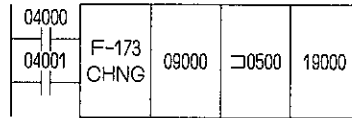
**F-173
CHNG**

**データチェンジ(1バイト)
(CHaNG)**

(この命令はJW-31CUH/H1ではプログラムできません。)

シンボル	① F-173 ② CHNG	S	D ₁	D ₂	①モード指定 ②実行入力														
機能	<p>レジスタD₁を先頭アドレスとしレジスタD₂の内容の数で指定する領域内で検索を行ないます。レジスタSの内容が検索データです。データ検索して最初の一致アドレス(D₁を基準として)何バイト目にあるかをD₂+2へ格納します。一致したデータ個数をD₂+1へ格納します。書き換えたいデータはレジスタS+1で、モード指定①入力の条件で書き換わります。</p> <ul style="list-style-type: none"> ●モード指定①入力OFFの時 最初の一致アドレスのみデータを書き換える。 ●モード指定①入力ONの時 全ての一致アドレスのデータを書き換える。 																		
演算条件																			
Sの使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>ファイル000000~037776</td> <td>ファイル@000000~@037774</td> </tr> </table>					コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	ファイル000000~037776	ファイル@000000~@037774
コ0000~コ1576	@コ0000~@コ1574																		
コ2000~コ7576	@コ2000~@コ7574																		
b0000~b1776	@b0000~@b1774																		
b2000~b3776	@b2000~@b3774																		
09000~99776	@09000~@99774																		
E0000~E7776	@E0000~@E7774																		
ファイル000000~037776	ファイル@000000~@037774																		
D ₁ の使用範囲	<table border="0"> <tr> <td>コ0000~コ1577</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7577</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1777</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3777</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99777</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7777</td> <td>@E0000~@E7774</td> </tr> <tr> <td>ファイル000000~037777</td> <td>ファイル@000000~@037774</td> </tr> </table>					コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	ファイル000000~037777	ファイル@000000~@037774
コ0000~コ1577	@コ0000~@コ1574																		
コ2000~コ7577	@コ2000~@コ7574																		
b0000~b1777	@b0000~@b1774																		
b2000~b3777	@b2000~@b3774																		
09000~99777	@09000~@99774																		
E0000~E7777	@E0000~@E7774																		
ファイル000000~037777	ファイル@000000~@037774																		
D ₂ の使用範囲	<table border="0"> <tr> <td>コ0000~コ1575</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7575</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1775</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3775</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99775</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7775</td> <td>@E0000~@E7774</td> </tr> <tr> <td>ファイル000000~037775</td> <td>ファイル@000000~@037774</td> </tr> </table>					コ0000~コ1575	@コ0000~@コ1574	コ2000~コ7575	@コ2000~@コ7574	b0000~b1775	@b0000~@b1774	b2000~b3775	@b2000~@b3774	09000~99775	@09000~@99774	E0000~E7775	@E0000~@E7774	ファイル000000~037775	ファイル@000000~@037774
コ0000~コ1575	@コ0000~@コ1574																		
コ2000~コ7575	@コ2000~@コ7574																		
b0000~b1775	@b0000~@b1774																		
b2000~b3775	@b2000~@b3774																		
09000~99775	@09000~@99774																		
E0000~E7775	@E0000~@E7774																		
ファイル000000~037775	ファイル@000000~@037774																		
演算条件	入力信号(②実行入力)の立上り(OFF→ON)																		
演算後	S, S+1の内容	不変																	
	D ₂ の内容	不変																	
	D ₂ +1, D ₂ +2の内容	演算結果	検索データ無しと256個の時D ₂ +1は000となる。																
	フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354													
		有	0	1	0	0													
		無	1	0	0	0													

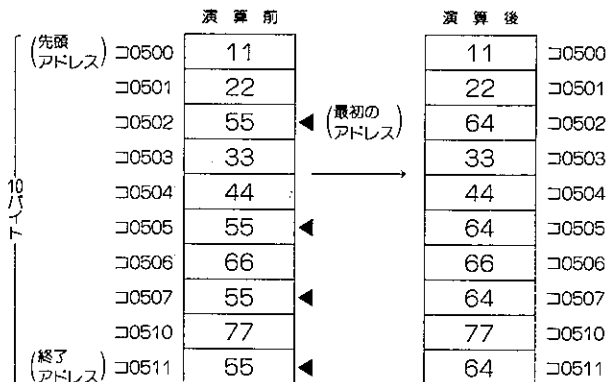
(解説)



命 令	
STR	04000
STR	04001
F-173	
	09000
	00500
	19000

- 実行入力条件04001がOFF→ON変化時に演算します。
- レジスタ09000の内容(55)が検索データです。
- 先頭アドレスは00500です。
- 終了アドレスは、00500からレジスタ19000の内容(010(10)バイト)の位置です。数に先頭アドレスも含まれます。
- レジスタ19001に検索個数を格納します。
- レジスタ19002に初の一致アドレス(D₁を基準としたバイト数)を格納します。
- 書き換えデータはレジスタ09001の内容(64)です。
- 下記の例では04000がONの動作です。

(検索データ)	09000	55	バイト数	010(10)	19000
(書き換えデータ)	09001	64	個数	004(10)	19001
			アドレス	002(8)	19002



演算後キャリーフラグ(07356)がONする。

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合、必ず偶数アドレスを設定してください。(コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。

- フラグの状態は、そのスキャンサイクルの中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)
- D₁、D₂で設定したレジスタ範囲が、ファイル番号0のファイルアドレス001600~001677と036000以降に入らないようにしてください。(9・3ページ「ソースとディスティネーション」参照)
- レジスタD₂の内容は000~377(8)です。000にすると256バイトになります。
- レジスタD₂+1の内容は000~377(8)です。000はデータ無し又は256個のときです。キャリーフラグ(07356)でデータの有無を判別してください。
- レジスタD₂+2の内容は000~377(8)でD₁を基準としたバイト数が入ります。000は先頭アドレスを示します。

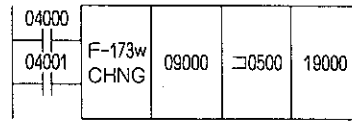
**F-173w
CHNG**

**データチェンジ(1ワード)
(CHaNG)**

(この命令はJW-31CUH/H1ではプログラムできません。)

シンボル					
機能	<p>レジスタD₁を先頭アドレスとしレジスタD₂の内容の数で指定する領域内で検索を行ないます。レジスタS、S+1の内容が検索データです。データ検索して最初の一致アドレス(D₁を基準として)何ワード目にあるかをD₂+2へ格納します。一致したデータ個数をD₂+1へ格納します。書き換えたいデータはレジスタS+2、S+3で、モード指定①入力の条件で書き換わります。</p> <ul style="list-style-type: none"> ●モード指定①入力OFFの時 最初の一致アドレスのみデータを書き換える。 ●モード指定①入力ONの時 全ての一致アドレスのデータを書き換える。 				
演算条件					
Sの使用範囲	<p>コ0000~コ1574 @コ0000~@コ1574 コ2000~コ7574 @コ2000~@コ7574 b0000~b1774 @b0000~@b1774 b2000~b3774 @b2000~@b3774 09000~99774 @09000~@99774 E0000~E7774 @E0000~@E7774 ファイル1 000000~037774 ファイル1 @000000~@037774</p>				
D ₁ の使用範囲	<p>コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 ファイル1 000000~037776 ファイル1 @000000~@037774</p>				
D ₂ の使用範囲	<p>コ0000~コ1575 @コ0000~@コ1574 コ2000~コ7575 @コ2000~@コ7574 b0000~b1775 @b0000~@b1774 b2000~b3775 @b2000~@b3774 09000~99775 @09000~@99774 E0000~E7775 @E0000~@E7774 ファイル1 000000~037775 ファイル1 @000000~@037774</p>				
演算条件	入力信号(②実行入力)の立上り(OFF→ON)				
演算後	S~S+3の内容	不変			
	D ₂ の内容	不変			
フラグ	D ₂ +1、D ₂ +2の内容	演算結果	検索データ無しと256個の時D ₂ +1は000となる。		
	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
フラグ	有	0	1	0	0
	無	1	0	0	0

(解説)



命 令	
STR	04000
STR	04001
F-173w	09000
	コ0500
	19000

- 実行入力条件04001がOFF→ON変化時に演算します。
- レジスタ09000、09001の内容(3355)が検索データです。
- 先頭アドレスはコ0500です。
- 終了アドレスは、コ0500からレジスタ19000の内容(005₍₁₀₎ワード)の位置です。ワード数に先頭アドレスも含まれます。
- レジスタ19001に検索個数を格納します。
- レジスタ19002に初の一致アドレスD₁を基準としたワード数を格納します。
- 書き換えデータはレジスタ09002、09003の内容(1964)です。
- 下記の例では04000がONの動作です。

(検索データ) 09000	55(下位)	ワード数	005(10)	19000
09001	33(上位)	個数	001(10)	19001
(書換データ) 09002	64(下位)	ワード目	001(8)	19002
09003	19(上位)			

	演算前		演算後	
ワード目	(先頭アドレス) コ0500	11	11	コ0500
	コ0501	22	22	コ0501
	コ0502	55	64	コ0502
	コ0503	33	19	コ0503
	コ0504	44	44	コ0504
	コ0505	55	55	コ0505
	コ0506	66	66	コ0506
	コ0507	55	55	コ0507
	コ0510	77	77	コ0510
	(終了アドレス) コ0511	55	55	コ0511

- フラグの状態は、そのスキャンサイクルの中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)
- S、D₁には必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- D₁、D₂で設定したレジスタ範囲が、ファイル番号0のファイルアドレス001600~001677と036000以降に入らないようにしてください。(9・3ページ「ソースとディスティネーション」参照)
- レジスタD₂の内容は000~377₍₈₎です。000にすると256ワードになります。
- レジスタD₂+1の内容は000~377₍₈₎です。000はデータ無し又は256個のときです。キャリーフラグ(07356)でデータの有無を判別してください。
- レジスタD₂+2の内容は000~377₍₈₎でD₁を基準としたバイト数が入ります。000は先頭アドレスを示します。

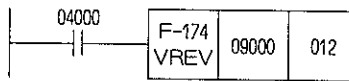
- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合、必ず偶数アドレスを設定してください。(@コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。

F-174
VREV

レジスタ間(1バイト)データ交換
(Vartical REVerse)

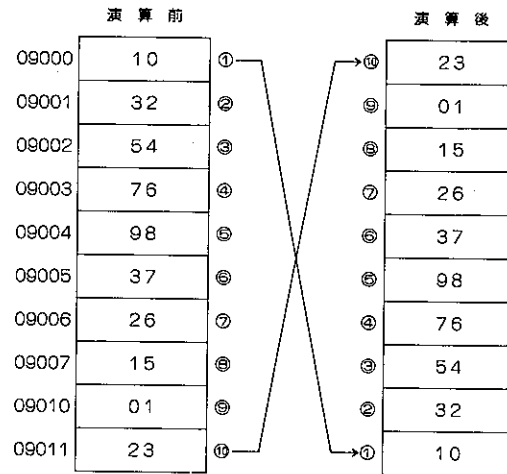
シンボル	$\overline{F-174}$ VREV D n														
機能	レジスタDを先頭とするnバイトのレジスタ範囲内で最下位アドレスの内容と最上位アドレスの内容を交換する。														
演算内容	<p style="text-align: center;">データ交換</p>														
Dの使用範囲	<table border="0"> <tr><td>コ0000~コ1577</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7577</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1777</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3777</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99777</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7777</td><td>@E0000~@E7774</td></tr> <tr><td>771#1 000000~037777</td><td>771#1 @000000~@037774</td></tr> </table>	コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	771#1 000000~037777	771#1 @000000~@037774
コ0000~コ1577	@コ0000~@コ1574														
コ2000~コ7577	@コ2000~@コ7574														
b0000~b1777	@b0000~@b1774														
b2000~b3777	@b2000~@b3774														
09000~99777	@09000~@99774														
E0000~E7777	@E0000~@E7774														
771#1 000000~037777	771#1 @000000~@037774														
nの使用範囲	0000~1777(8) (0000にすると1024バイトとなる)														
演算条件	入力信号の立上り(OFF→ON)														
演算後	D~D+n-1の内容	演算結果													
	フラグ	不変													

(解説)



命令	
STR	04000
F-174	09000
	0012

入力条件04000がOFF→ONの変化時にレジスタ09000~09011の10バイトの内容(1バイトデータ)を最下位アドレスと最上位アドレスから順に入換えます。



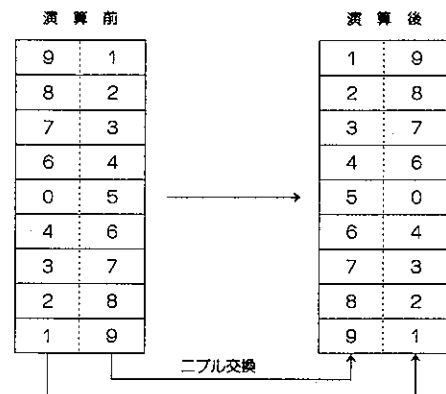
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- n、Dで設定したレジスタ範囲が、ファイル番号0のファイルアドレス001600~001777と035600以降に入らないようにしてください。
- データ交換バイト数が多い時、演算時間が長くなります。演算時間は7・1ページ「命令語一覧表」を参照下さい。

参考 下記のF命令は働きが類似しています。
F-02、F-02w、F-02d、F-174

**F-175
NSWP**

**レジスタの上位4ビットと下位4ビット交換
(Nbyte SWaP)**

シンボル	<table border="1"> <tr> <td>F-175 NSWP</td> <td>D</td> <td>n</td> </tr> </table>		F-175 NSWP	D	n	<p>(解説)</p>	<table border="1"> <tr> <th colspan="2">命令</th> </tr> <tr> <td>STR</td> <td>04000</td> </tr> <tr> <td>F-175</td> <td>09000</td> </tr> <tr> <td></td> <td>0011</td> </tr> </table>	命令		STR	04000	F-175	09000		0011	
F-175 NSWP	D	n														
命令																
STR	04000															
F-175	09000															
	0011															
機能	レジスタDを先頭とするnバイトのレジスタ範囲内でデータ上位4ビットと下位4ビットの内容を交換する。		<p>入力条件04000がOFF→ONの変化時にレジスタ09000~09010の9バイトの内容(1バイトデータ)のニブル交換を行う。</p>													
演算内容	D~D+n-1のニブル交換															
Dの使用範囲	<table border="0"> <tr> <td>コ0000~コ1577</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7577</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1777</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3777</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99777</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7777</td> <td>@E0000~@E7774</td> </tr> <tr> <td>ファイル000000~037777</td> <td>ファイル@000000~@037774</td> </tr> </table>		コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	ファイル000000~037777	ファイル@000000~@037774
コ0000~コ1577	@コ0000~@コ1574															
コ2000~コ7577	@コ2000~@コ7574															
b0000~b1777	@b0000~@b1774															
b2000~b3777	@b2000~@b3774															
09000~99777	@09000~@99774															
E0000~E7777	@E0000~@E7774															
ファイル000000~037777	ファイル@000000~@037774															
nの使用範囲	0000~1777(8) (0000にすると1024バイトとなる)															
演算条件	入力信号の立上り(OFF→ON)															
演算後	D~D+n-1の内容	演算結果														
	フラグ	不変														



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- n、Dで設定したレジスタ範囲が、ファイル番号0のファイルアドレス001600~001777と036000以降に入らないようにしてください。
- ニブル交換するバイト数が多い時、演算時間が長くなります。演算時間は7・1ページ「命令語一覧表」を参照下さい。

参考 下記のF命令は働きが類似しています。
F-55、F-175

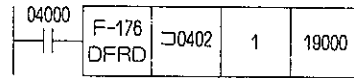
F-176
DFRD

指定アドレスのレジスタからの読出(256バイト)
(Direct File Read)

(この命令はJW-31CUH/H1ではプログラムできません。)

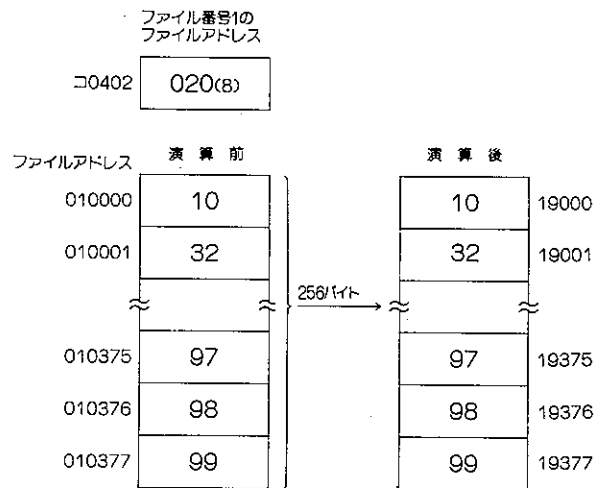
シンボル	F-176 DFRD S file N D			
機能	レジスタSの内容(ブロック番号)で指定するファイル番号N内のファイルアドレスを先頭とし256バイトのデータをレジスタDを先頭アドレスとする領域へ転送する。			
演算内容	File N、S→Dへ256バイト転送			
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 ファイル 000000~037777 ファイル @000000~@037774			
Nの使用範囲	0~3, 10~2C(H)			
Dの使用範囲	コ0000~コ1200 @コ0000~@コ1574 コ2000~コ7200 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 ファイル 000000~037776 ファイル @000000~@037774			
演算条件	入力信号の立上り(OFF→ON)			
演算後	S~S+255の内容	不変		
	D~D+255の内容	演算結果		
	フラグ	不変		

〔解説〕



命令	
STR	04000
F-176	コ0402
	1
	19000

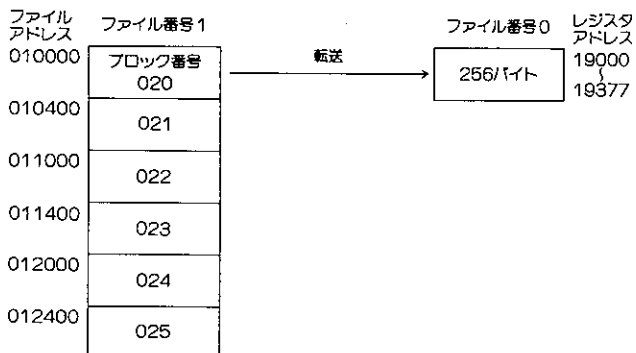
入力条件04000がOFF→ONの変化時ファイル番号1のブロック番号020(8) (レジスタ コ0402の内容)からレジスタ19000を先頭とする256バイトにデータをブロック転送します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- Dで設定する先頭アドレスにご注意ください。そのアドレス以後256バイトが転送領域になります。
- Sの内容はファイルレジスタ64Kバイトを256バイト単位で区切り割付けられた000~377(8)のブロック番号を使用します。(次ページ参照)
本命令では256バイトのブロック転送します。

- ファイル番号Nの使用範囲はコントロールユニットの種類によって異なります。(2・1ページ「ファイル番号とファイルアドレス」参照)

〔参考〕 下記のF命令は働きが類似しています。
F-102、F-102w、F-176



ブロック番号とファイルアドレス(先頭アドレス)一覧表

ブロック番号	ファイルアドレス	ブロック番号	ファイルアドレス	ブロック番号	ファイルアドレス	ブロック番号	ファイルアドレス
000	000000	100	040000	200	100000	300	140000
001	000400	101	040400	201	100400	301	140400
002	001000	102	041000	202	101000	302	141000
003	001400	103	041400	203	101400	303	141400
004	002000	104	042000	204	102000	304	142000
005	002400	105	042400	205	102400	305	142400
006	003000	106	043000	206	103000	306	143000
007	003400	107	043400	207	103400	307	143400
010	004000	110	044000	210	104000	310	144000
011	004400	111	044400	211	104400	311	144400
012	005000	112	045000	212	105000	312	145000
013	005400	113	045400	213	105400	313	145400
014	006000	114	046000	214	106000	314	146000
015	006400	115	046400	215	106400	315	146400
016	007000	116	047000	216	107000	316	147000
017	007400	117	047400	217	107400	317	147400
020	010000	120	050000	220	110000	320	150000
021	010400	121	050400	221	110400	321	150400
022	011000	122	051000	222	111000	322	151000
023	011400	123	051400	223	111400	323	151400
024	012000	124	052000	224	112000	324	152000
025	012400	125	052400	225	112400	325	152400
026	013000	126	053000	226	113000	326	153000
027	013400	127	053400	227	113400	327	153400
030	014000	130	054000	230	114000	330	154000
031	014400	131	054400	231	114400	331	154400
032	015000	132	055000	232	115000	332	155000
033	015400	133	055400	233	115400	333	155400
034	016000	134	056000	234	116000	334	156000
035	016400	135	056400	235	116400	335	156400
036	017000	136	057000	236	117000	336	157000
037	017400	137	057400	237	117400	337	157400
040	020000	140	060000	240	120000	340	160000
041	020400	141	060400	241	120400	341	160400
042	021000	142	061000	242	121000	342	161000
043	021400	143	061400	243	121400	343	161400
044	022000	144	062000	244	122000	344	162000
045	022400	145	062400	245	122400	345	162400
046	023000	146	063000	246	123000	346	163000
047	023400	147	063400	247	123400	347	163400
050	024000	150	064000	250	124000	350	164000
051	024400	151	064400	251	124400	351	164400
052	025000	152	065000	252	125000	352	165000
053	025400	153	065400	253	125400	353	165400
054	026000	154	066000	254	126000	354	166000
055	026400	155	066400	255	126400	355	166400
056	027000	156	067000	256	127000	356	167000
057	027400	157	067400	257	127400	357	167400
060	030000	160	070000	260	130000	360	170000
061	030400	161	070400	261	130400	361	170400
062	031000	162	071000	262	131000	362	171000
063	031400	163	071400	263	131400	363	171400
064	032000	164	072000	264	132000	364	172000
065	032400	165	072400	265	132400	365	172400
066	033000	166	073000	266	133000	366	173000
067	033400	167	073400	267	133400	367	173400
070	034000	170	074000	270	134000	370	174000
071	034400	171	074400	271	134400	371	174400
072	035000	172	075000	272	135000	372	175000
073	035400	173	075400	273	135400	373	175400
074	036000	174	076000	274	136000	374	176000
075	036400	175	076400	275	136400	375	176400
076	037000	176	077000	276	137000	376	177000
077	037400	177	077400	277	137400	377	177400

←ファイル0
使用禁止

↓
ファイル1
以後
使用禁止

ファイル0
使用禁止
↓
ファイル0
以後
使用禁止

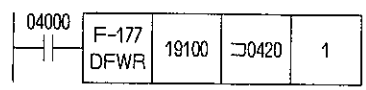
F-177
DFWR

指定アドレスのレジスタへ書込(256バイト)
(Direct File WRite)

(この命令はJW-31CUH/H1ではプログラムできません。)

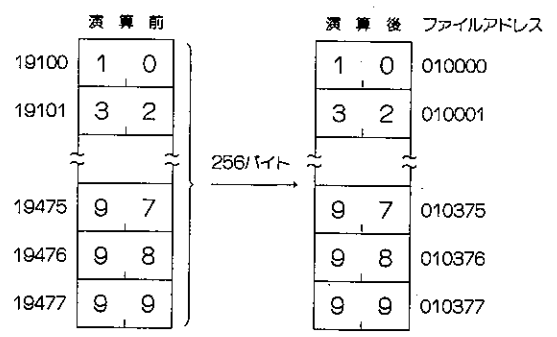
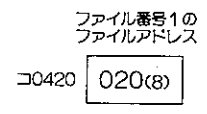
シンボル	F-177 DFWR	S	D	file N
機能	レジスタSを先頭アドレスとし256バイトのデータをレジスタDの内容(ブロック番号)で指定するファイル番号内のファイルアドレスを先頭とする領域へ転送する。			
演算内容	File O, S → File N, Dへ256バイト転送			
Sの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 ファイル1 000000~037776 ファイル1 @000000~@037774			
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 ファイル1 000000~037777 ファイル1 @000000~@037774			
Nの使用範囲	0~3, 10~2C(H)			
演算条件	入力信号の立上り(OFF→ON)			
演算後	S~S+255の内容	不変		
	D~D+255の内容	演算結果		
	フラグ	不変		

(解説)



命 令	
STR	04000
F-177	19100
	コ0420
	1

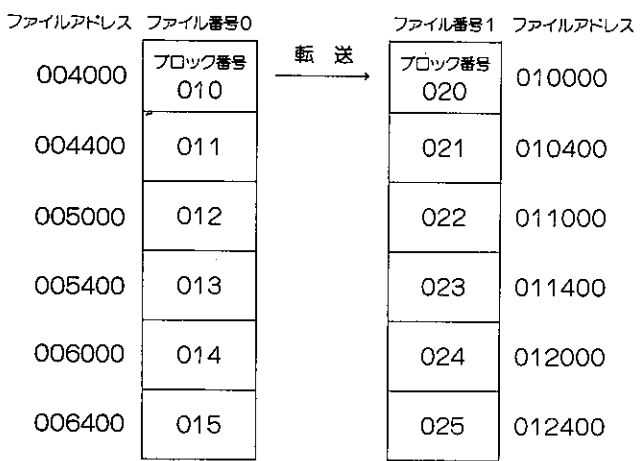
入力条件04000がOFF→ONの変化時、レジスタ19100を先頭アドレスとする256バイトデータを、ファイル番号1のブロック番号020(8)(レジスタコ0420の内容)で表わすファイルアドレスを先頭とする領域へブロック転送します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

- Sには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- Dの内容はファイルレジスタ64Kバイトを256バイト単位で区切り割付けられた000~377(8)ブロック番号を使用します。(前ページ参照)
本命令では256バイトのブロック転送します。
- ファイル番号Nの使用範囲はコントロールユニットの種類によって異なります。(2・1ページ「ファイル番号とファイルアドレス」参照)
- Dのブロック番号設定ではファイル番号0の003(8)と073(8)以後は使用しないでください。

参考 下記のF命令は働きが類似しています。
F-103、F-103w、F-177



レジスタ間 (1バイト) の比較 (リレー出力付)

F-180
CP>

(ComPare >)

F-183
CP>=

(ComPare >=)

F-181
CP<

(ComPare <)

F-184
CP<=

(ComPare <=)

F-182
CP=

(ComPare =)

F-185
CP<>

(ComPare <>)

シンボル	F-180 CP> S ₁ S ₂ BIT	F-183 CP>= S ₁ S ₂ BIT
	F-181 CP< S ₁ S ₂ BIT	F-184 CP<= S ₁ S ₂ BIT
	F-182 CP= S ₁ S ₂ BIT	F-185 CP<> S ₁ S ₂ BIT

機能 レジスタS₁の内容とS₂の内容を大小比較し、比較結果が成立時にBITをONする。

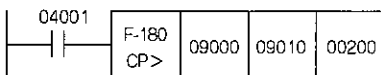
演算内容	F-180 S ₁ >S ₂ →BIT ON	F-183 S ₁ ≥S ₂ →BIT ON
	F-181 S ₁ <S ₂ →BIT ON	F-184 S ₁ ≤S ₂ →BIT ON
	F-182 S ₁ =S ₂ →BIT ON	F-185 S ₁ ≠S ₂ →BIT ON

S ₁ の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @02000~@03774 09000~99777 @E9000~@99774 E0000~E7777 @E0000~@E7774 774#1 000000~037777 774#1 @000000~@037774	S ₂ の使用範囲	コ0000~コ1577! @コ0000~@コ1574 コ2000~コ7577! @コ2000~@コ7574 b0000~b1777! @b0000~@b1774 b2000~b3777! @b2000~@b3774 09000~99777! @09000~@99774 E0000~E7777! @E0000~@E7774 774#1 000000~037777 774#1 @000000~@037774
----------------------	---	----------------------	---

BITの使用範囲	00000~15777、20000~75777	演算条件	入力信号がONの時 (OFF→ONの変化時に限定されない)
----------	-------------------------	------	----------------------------------

演算後	S ₁ ,S ₂ の内容	不変			
	BITの内容	F-180	S ₁ >S ₂ の時、ON S ₁ ≤S ₂ の時、OFF	F-183	S ₁ ≥S ₂ の時、ON S ₁ <S ₂ の時、OFF
		F-181	S ₁ <S ₂ の時、ON S ₁ ≥S ₂ の時、OFF	F-184	S ₁ ≤S ₂ の時、ON S ₁ >S ₂ の時、OFF
		F-182	S ₁ =S ₂ の時、ON S ₁ ≠S ₂ の時、OFF	F-185	S ₁ ≠S ₂ の時、ON S ₁ =S ₂ の時、OFF
フラグ	ゼロ 07357 0	キャリー 07356 0	エラー 07355 0	ノキャリー 07354 0	

〔解説〕



命令	
STR	04001
F-180	09000 09010 00200

入力条件04001がONの時に、レジスタ09000とレジスタ09010の内容を比較して、09000>09010の場合にリレー00200がONします。入力条件04001がOFFの時、および09000≤09010の場合にはリレー00200はOFFになります。

レジスタ間（1ワード）の比較（リレー出力付）

F-180w CP>	(ComPare >)	F-183w CP>=	(ComPare >=)
F-181w CP<	(ComPare <)	F-184w CP<=	(ComPare <=)
F-182w CP=	(ComPare =)	F-185w CP<>	(ComPare <>)

シンボル	<table border="1"> <tr><td>F-180w</td><td>S₁</td><td>S₂</td><td>BIT</td></tr> <tr><td>F-181w</td><td>S₁</td><td>S₂</td><td>BIT</td></tr> <tr><td>F-182w</td><td>S₁</td><td>S₂</td><td>BIT</td></tr> </table>		F-180w	S ₁	S ₂	BIT	F-181w	S ₁	S ₂	BIT	F-182w	S ₁	S ₂	BIT	<table border="1"> <tr><td>F-183w</td><td>S₁</td><td>S₂</td><td>BIT</td></tr> <tr><td>F-184w</td><td>S₁</td><td>S₂</td><td>BIT</td></tr> <tr><td>F-185w</td><td>S₁</td><td>S₂</td><td>BIT</td></tr> </table>		F-183w	S ₁	S ₂	BIT	F-184w	S ₁	S ₂	BIT	F-185w	S ₁	S ₂	BIT
	F-180w	S ₁	S ₂	BIT																								
F-181w	S ₁	S ₂	BIT																									
F-182w	S ₁	S ₂	BIT																									
F-183w	S ₁	S ₂	BIT																									
F-184w	S ₁	S ₂	BIT																									
F-185w	S ₁	S ₂	BIT																									
機能	レジスタS ₁ , S ₁ +1の内容(1ワードデータ)とレジスタS ₂ , S ₂ +1の内容(1ワードデータ)を大小比較し、比較結果が成立時にBITをONする。																											
演算内容	F-180w	S ₁ , S ₁ +1 > S ₂ , S ₂ +1 → BIT ON	F-183w	S ₁ , S ₁ +1 ≥ S ₂ , S ₂ +1 → BIT ON																								
	F-181w	S ₁ , S ₁ +1 < S ₂ , S ₂ +1 → BIT ON	F-184w	S ₁ , S ₁ +1 ≤ S ₂ , S ₂ +1 → BIT ON																								
	F-182w	S ₁ , S ₁ +1 = S ₂ , S ₂ +1 → BIT ON	F-185w	S ₁ , S ₁ +1 ≠ S ₂ , S ₂ +1 → BIT ON																								
S ₁ の使用範囲	コ0000~コ1576, @コ0000~@コ1574 コ2000~コ7576, @コ2000~@コ7574 b0000~b1776, @b0000~@コ1774 b2000~b3776, @b2000~@コ3774 09000~99776, @E9000~@99774 E0000~E7776, @E0000~@E7774 774#1 000000~037776, 774#1 @000000~@037774		S ₂ の使用範囲 コ0000~コ1576, @コ0000~@コ1574 コ2000~コ7576, @コ2000~@コ7574 b0000~b1776, @b0000~@コ1774 b2000~b3776, @b2000~@コ3774 09000~99776, @09000~@99774 E0000~E7776, @E0000~@E7774 774#1 000000~037776, 774#1 @000000~@037774																									
BITの使用範囲	00000~15777, 20000~75777		演算条件	入力信号がONの時 (OFF→ONの変化時に限定されない)																								
演算後	S ₁ , S ₁ +1 S ₂ , S ₂ +1の内容	不変																										
	BITの内容	F-180w	S ₁ , S ₁ +1 > S ₂ , S ₂ +1の時、ON S ₁ , S ₁ +1 ≤ S ₂ , S ₂ +1の時、OFF	F-183w	S ₁ , S ₁ +1 ≥ S ₂ , S ₂ +1の時、ON S ₁ , S ₁ +1 < S ₂ , S ₂ +1の時、OFF																							
		F-181w	S ₁ , S ₁ +1 < S ₂ , S ₂ +1の時、ON S ₁ , S ₁ +1 ≥ S ₂ , S ₂ +1の時、OFF	F-184w	S ₁ , S ₁ +1 ≤ S ₂ , S ₂ +1の時、ON S ₁ , S ₁ +1 > S ₂ , S ₂ +1の時、OFF																							
		F-182w	S ₁ , S ₁ +1 = S ₂ , S ₂ +1の時、ON S ₁ , S ₁ +1 ≠ S ₂ , S ₂ +1の時、OFF	F-185w	S ₁ , S ₁ +1 ≠ S ₂ , S ₂ +1の時、ON S ₁ , S ₁ +1 = S ₂ , S ₂ +1の時、OFF																							
フラグ	ゼロ 07357 0	キャリー 07356 0	エラー 07355 0	ノキャリー 07354 0																								
〔解説〕		<table border="1"> <tr><th colspan="2">命令</th></tr> <tr><td>STR</td><td>04201</td></tr> <tr><td>F-180w</td><td>09100</td></tr> <tr><td></td><td>09110</td></tr> <tr><td></td><td>00100</td></tr> </table>		命令		STR	04201	F-180w	09100		09110		00100	入力条件04201がONの時に、レジスタ09100、09101の内容とレジスタ09110、09111の内容を比較して、(09100, 09101) > (09110, 09111) の場合にリレー00100がONします。入力条件04201がOFFの時、および(09100, 09101) ≤ (09110, 09111)の場合には、リレー00100はOFFになります。														
命令																												
STR	04201																											
F-180w	09100																											
	09110																											
	00100																											

レジスタと定数（1バイト）の比較（リレー出力付）

Fc180 CP>	(ComPare >)	Fc183 CP>=	(ComPare >=)
Fc181 CP<	(ComPare <)	Fc184 CP<=	(ComPare <=)
Fc182 CP=	(ComPare =)	Fc185 CP<>	(ComPare <>)

シンボル	<table border="1"> <tr><td>Fc180</td><td>S</td><td>n</td><td>BIT</td></tr> <tr><td>Fc181</td><td>S</td><td>n</td><td>BIT</td></tr> <tr><td>Fc182</td><td>S</td><td>n</td><td>BIT</td></tr> </table>		Fc180	S	n	BIT	Fc181	S	n	BIT	Fc182	S	n	BIT	<table border="1"> <tr><td>Fc183</td><td>S</td><td>n</td><td>BIT</td></tr> <tr><td>Fc184</td><td>S</td><td>n</td><td>BIT</td></tr> <tr><td>Fc185</td><td>S</td><td>n</td><td>BIT</td></tr> </table>		Fc183	S	n	BIT	Fc184	S	n	BIT	Fc185	S	n	BIT
Fc180	S	n	BIT																									
Fc181	S	n	BIT																									
Fc182	S	n	BIT																									
Fc183	S	n	BIT																									
Fc184	S	n	BIT																									
Fc185	S	n	BIT																									
機能	レジスタSの内容とnの内容を大小比較し、比較結果が成立時にBITをONする。																											
演算内容	Fc180	S > n → BIT ON	Fc183	S ≥ n → BIT ON																								
	Fc181	S < n → BIT ON	Fc184	S ≤ n → BIT ON																								
	Fc182	S = n → BIT ON	Fc185	S ≠ n → BIT ON																								
Sの使用範囲	コ0000~コ15777 @コ0000~@コ1574 コ2000~コ75777 @コ2000~@コ7574 b0000~b17777 @b0000~@b1774 b2000~b37777 @b2000~@b3774 09000~997777 @09000~@99774 E0000~E77777 @E0000~@E7774 774#1 000000~037777 774#1 @000000~@037774		nの使用範囲	000~377 (8)																								
BITの使用範囲	00000~15777、20000~75777		演算条件	入力信号がONの時 (OFF→ONの変化時に限定されない)																								
演算後	Sの内容	不変																										
	BITの内容	Fc180	S > nの時、ON S ≤ nの時、OFF	Fc183	S ≥ nの時、ON S < nの時、OFF																							
		Fc181	S < nの時、ON S ≥ nの時、OFF	Fc184	S ≤ nの時、ON S > nの時、OFF																							
		Fc182	S = nの時、ON S ≠ nの時、OFF	Fc185	S ≠ nの時、ON S = nの時、OFF																							
フラグ	ゼロ 07357 0	キャリー 07356 0	エラー 07355 0	リフキャリー 07354 0																								
〔解説〕		<table border="1"> <tr><th colspan="2">命令</th></tr> <tr><td>STR</td><td>05001</td></tr> <tr><td>Fc180</td><td>19000</td></tr> <tr><td></td><td>012</td></tr> <tr><td></td><td>00300</td></tr> </table>		命令		STR	05001	Fc180	19000		012		00300	入力条件05001がONの時に、レジスタ19000の内容と8進定数012を比較して、(19000) > 012の場合にリレー00300がONします。 入力条件05001がOFFの時、および(19000) ≤ 012の場合にはリレー00300はOFFになります。														
命令																												
STR	05001																											
Fc180	19000																											
	012																											
	00300																											

レジスタと定数（1ワード）の比較（リレー出力付）

Fc180w
CP>

(ComPare >)

Fc183w
CP>=

(ComPare >=)

Fc181w
CP<

(ComPare <)

Fc184w
CP<=

(ComPare <=)

Fc182w
CP=

(ComPare =)

Fc185w
CP<>

(ComPare <>)

シンボル	Fc180w CP>	S	n	BIT	Fc183w CP>=	S	n	BIT
	Fc181w CP<	S	n	BIT	Fc184w CP<=	S	n	BIT
	Fc182w CP=	S	n	BIT	Fc185w CP<>	S	n	BIT

機能 レジスタS、S+1の内容（1ワードデータ）と定数nの内容を大小比較し、比較結果が成立時にBITをONする。

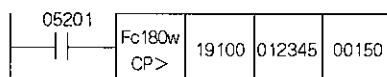
演算内容	Fc180w	$S, S+1 > n \rightarrow \text{BIT ON}$	Fc183w	$S, S+1 \geq n \rightarrow \text{BIT ON}$
	Fc181w	$S, S+1 < n \rightarrow \text{BIT ON}$	Fc184w	$S, S+1 \leq n \rightarrow \text{BIT ON}$
	Fc182w	$S, S+1 = n \rightarrow \text{BIT ON}$	Fc185w	$S, S+1 \neq n \rightarrow \text{BIT ON}$

Sの使用範囲	コ0000~コ1576 ! @コ0000~@コ1574 コ2000~コ7576 ! @コ2000~@コ7574 b0000~b1776 ! @b0000~@b1774 b2000~b3776 ! @b2000~@b3774 09000~99776 ! @09000~@99774 E0000~E7776 ! @E0000~@E7774 771M1 000000~037776 ! 771M1 @000000~@037774	nの使用範囲	000000~177777 (8)
--------	---	--------	-------------------

BITの使用範囲	00000~15777、20000~75777	演算条件	入力信号がONの時 (OFF→ONの変化時に限定されない)
----------	-------------------------	------	----------------------------------

演算後	S、S+1の内容	不変			
	BITの内容	Fc180w	$S, S+1 > n$ の時、ON $S, S+1 \leq n$ の時、OFF	Fc183w	$S, S+1 \geq n$ の時、ON $S, S+1 < n$ の時、OFF
		Fc181w	$S, S+1 < n$ の時、ON $S, S+1 \geq n$ の時、OFF	Fc184w	$S, S+1 \leq n$ の時、ON $S, S+1 > n$ の時、OFF
		Fc182w	$S, S+1 = n$ の時、ON $S, S+1 \neq n$ の時、OFF	Fc185w	$S, S+1 \neq n$ の時、ON $S, S+1 = n$ の時、OFF
フラグ	ゼロ 07357 0	キャリー 07356 0	エラー 07355 0	ノキャリー 07354 0	

〔解説〕



命令	
STR	05201
Fc180w	19100 012345 00150

入力条件05201がONの時に、レジスタ(19100、19101)の内容と8進定数012345を比較して、(19100、19101)>012345の場合にリレー00150がONします。
入力条件05201がOFFの時、および(19100、19101)≤012345の場合にはリレー00150はOFFになります。

F-202
OPCH

オープンチャンネル(1階層：局番8進数設定)

F-203
OPCH

オープンチャンネル(1階層：局番16進数設定)

シンボル	— <table border="1"> <tr> <td>F-202 OPCH</td> <td>UN CH-ST</td> <td>file N</td> <td>n</td> </tr> </table>	F-202 OPCH	UN CH-ST	file N	n	— <table border="1"> <tr> <td>F-203 OPCH</td> <td>UN CH-ST</td> <td>file N</td> <td>n</td> </tr> </table>	F-203 OPCH	UN CH-ST	file N	n
F-202 OPCH	UN CH-ST	file N	n							
F-203 OPCH	UN CH-ST	file N	n							
機能	サテライトネット (JW-22CM、ZW/JW-20CM)、SUMINET-3200 (ZW-30CM) を使用した PC間データ交換用の相手局指定命令です。F-204 (SEND) 又はF-205(RCV) 命令と組合せて使用します。									
UNの使用範囲	0~6 : JW-22CMのユニット番号									
CHの使用範囲	0~3 : 指定ユニット番号に対するチャンネル番号									
STの使用範囲	000~377 ⁽⁸⁾ 00~FF ⁽¹⁶⁾ : 通信相手局の局番									
Nの使用範囲	0~3, 10~2C ^(H) : 通信相手局のファイル番号									
nの使用範囲	000000~177777 : 通信相手局のファイルアドレス (データ先頭アドレス)									
演算条件	入力信号がONの時 (OFF→ONの変化時に限定されない)									
フラグ	不変									

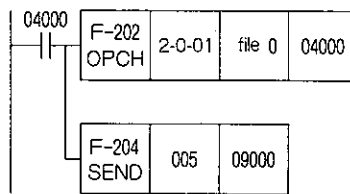
- CH0、CH1、CH2、CH3と分けるとPCプログラム中で4回同一ユニット番号のJW-22CMを使用できます。
- ファイルアドレス、ファイル番号については2・1ページを参照ください。
- 本命令はかならずF-204(SEND)命令 又は、F-205(RCV)命令と併用してください。
- F-202のSTは8進定数を使用します。
F-203のSTは16進定数を使用します。

F-204 SEND

送信命令

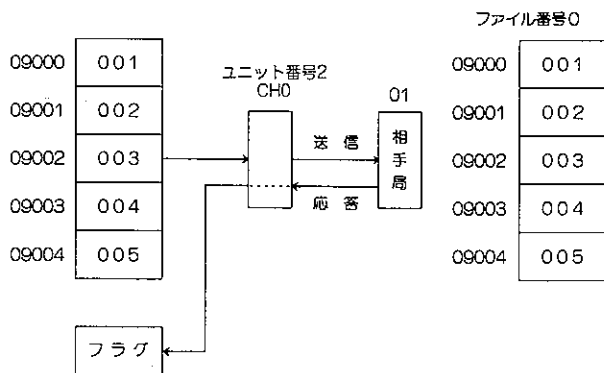
シンボル	F-204 SEND n S					
機能	サテライトネット間データ通信の送信 先頭アドレスと送信バイト数を指定					
演算内容	S~S+n-1→指定局					
nの使用範囲	000~377 ^(a) (000にすると256バイトとなる)					
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 77IM1 000000~037777 77IM1 @000000~@037774					
演算条件	入力信号の立上り(OFF→ON)					
演算後	Sの内容	不変				
		通信内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
	ポートからの応答なし	0	0	1	0	
	通信渋滞	0	0	0	1	
	通信中 (相手局からの応答待ち)	1	0	0	1	
	正常終了	0	1	0	0	
	異常終了(通信エラー)	0	1	1	0	
	相手局書込禁止	1	1	1	0	

(解説)



命令	
STR	04000
F-202	2-0-01 file 0 04000
F-204	005 09000

入力条件04000がOFF→ONの変化時にレジスタ09000~09004の内容(5バイトデータ)をユニット番号2のCHOを通してサテライトネット01局のファイル番号0、ファイルアドレス04000(レジスタ09000)に送信します。

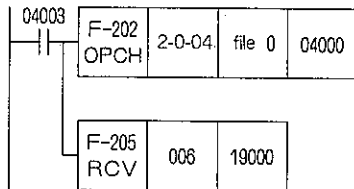


- この命令の使い方は、ネットワークユニットJW-22CMのユーザズマニュアルを参照してください。
- 間接アドレス指定する場合必ず偶数アドレスを設定してください。
(@コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ“間接アドレス指定”の頁を参照してください。
- 1階層通信の場合、必ずF-202又はF-203と併用して下さい。2階層通信の場合、必ず、F-206、F-207と併用して下さい。

F-205 RCV 受信命令

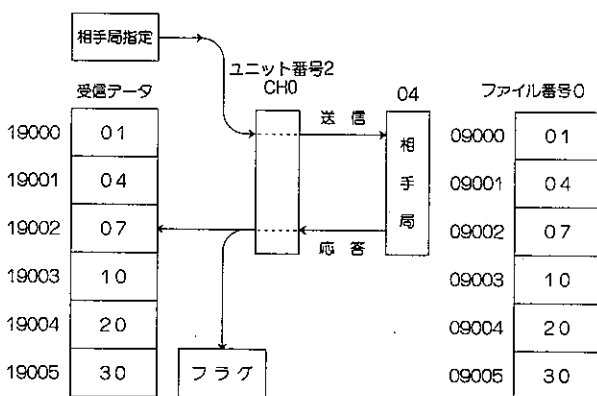
シンボル	— F-205 RCV n D					
機能	サテライトネット間データ通信の受信 先頭アドレスと受信バイト数を指定。					
演算内容	指定局→D~D+n-1					
nの使用範囲	000~377 ₍₈₎ (000にすると256バイトとなる)					
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 ファイル1 000000~037777 ファイル1 @000000~@037774					
演算条件	入力信号の立上り(OFF→ON)					
演算後	Dの内容	演算結果				
		通信内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
	フ	ポートからの応答なし	0	0	1	0
	ラ	通信渋滞	0	0	0	1
		通信中 (相手局からの応答待ち)	1	0	0	1
	グ	正常終了	0	1	0	0
		異常終了(通信エラー)	0	1	1	0

(解説)



命令	
STR	04000
F-202	2-0-04 file 0 04000
F-205	006 09000

入力条件04003がOFF→ONの変化時にユニット番号2のCH0を通してサテライトネット04局のファイル番号0、ファイルアドレス04000から6バイトのデータを読み出します。読み出したデータはレジスタ19000から19005に格納します。



- この命令の使い方は、ネットワークユニットJW-22CMのユーザズマニュアルを参照してください。
- 間接アドレス指定する場合必ず偶数アドレスを設定してください。
(@コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ“間接アドレス指定”の頁を参照してください。
- 1階層通信の場合、必ずF-202又はF-203と併用してください。2階層通信の場合、必ず、F-206、F-207と併用してください。

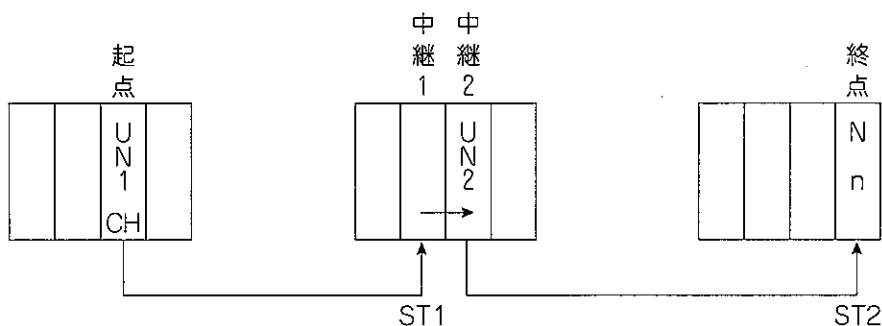
F-206 オープンチャンネル 1 (階層通信設定)

EOP1

F-207 オープンチャンネル 2 (階層通信設定)

EOP2

シンボル	— <table border="1"><tr><td>F-206 EOP1</td><td>UNI, CH</td><td>ST1</td><td>UN2</td></tr></table>				F-206 EOP1	UNI, CH	ST1	UN2	— <table border="1"><tr><td>F-207 EOP2</td><td>ST2</td><td>file N</td><td>n</td></tr></table>				F-207 EOP2	ST2	file N	n
F-206 EOP1	UNI, CH	ST1	UN2													
F-207 EOP2	ST2	file N	n													
機能	サテライトネット (JW-22CM、ZW/JW-20CM)、SUMINET-3200 (ZW-30CM) を使用した PC間データ交換用の相手局指定命令です。F-206 (EOP1) は、F-207 (EOP2)、F-204 (SEND)、又はF-205 (RCV) 命令と組み合わせて使用します。															
UN1の使用範囲	0~6		: SEND、RECEIVE命令の起点となるユニットのユニット番号													
UN2の使用範囲	0~7		: SEND、RECEIVEの中継局2となるユニットのユニット番号 (注) 中継局2がJW50H/70H/100Hのとき、スロット番号となる													
CHの使用範囲	0~3		: SEND、RECEIVE命令の起点となるユニットのチャンネル番号													
ST1の使用範囲	00~77		: SEND、RECEIVE命令の中継局1の局番													
ST2の使用範囲	00~77		: SEND、RECEIVE命令の終点局の局番													
Nの使用範囲	0~3, 10~2C(H)		: SEND、RECEIVE命令の終点局でのファイル番号													
nの使用範囲	000000~177777		: SEND、RECEIVE命令の終点局でのファイルアドレス													
演算条件	入力信号がONの時 (OFF→ONの変化時に限定されない)															
フラグ	不変															

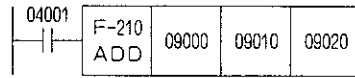


- F-206、F-207、F-204の3命令又はF-206、F-207、F-205の3命令は必ず併用してください。

F-210 ADD レジスタ間のバイナリ加算(8ビット+8ビット) (ADD)

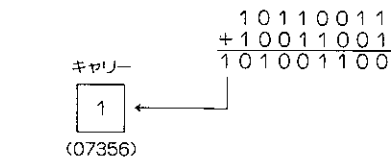
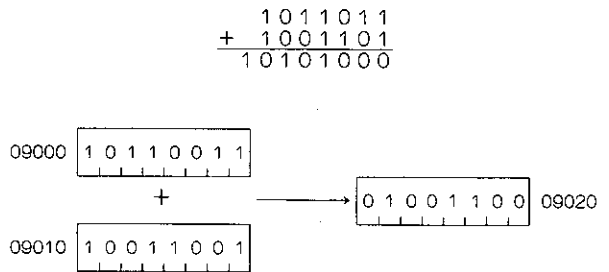
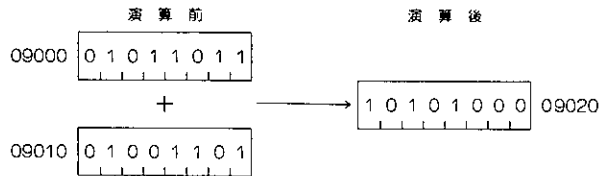
シンボル	— F-210 ADD S ₁ S ₂ D					
機能	レジスタS ₁ の内容とレジスタS ₂ の内容をバイナリ加算してレジスタDに格納する。					
演算内容	S ₁ +S ₂ →D					
S ₁ の使用範囲	コ0000~コ1577 コ2000~コ7577 b0000~b1777 b2000~b3777 09000~99777 E0000~E7777 774M1 000000~037777	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 774M1 @000000~@037774				
S ₂ の使用範囲	コ0000~コ1577 コ2000~コ7577 b0000~b1777 b2000~b3777 09000~99777 E0000~E7777 774M1 000000~037777	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 774M1 @000000~@037774				
Dの使用範囲	コ0000~コ1577 コ2000~コ7577 b0000~b1777 b2000~b3777 09000~99777 E0000~E7777 774M1 000000~037777	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 774M1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)					
演算	S ₁ の内容	不変				
	S ₂ の内容	不変				
後	Dの内容	演算結果				
	フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
		0	1	0	0	1
		001~377 ₍₈₎	0	0	0	1
		400 ₍₈₎	1	1	0	0
401 ₍₈₎ 以上	0	1	0	0		

(解説)



命令	
STR	04001
F-210	09000
	09010
	09020

入力条件04001がOFF→ONの変化時に、レジスタ09000の内容とレジスタ09010の内容をバイナリ加算して、結果をレジスタ09020に格納します。



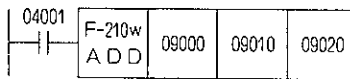
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

**F-210w
ADD**

**レジスタ間のバイナリ加算(16ビット+16ビット)
(ADD)**

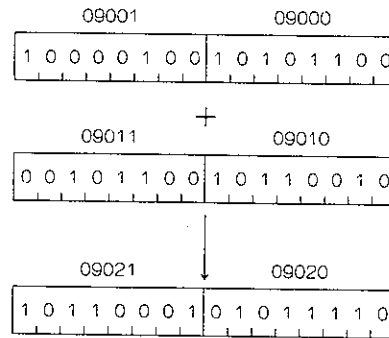
シンボル	<table border="1" style="display: inline-table;"> <tr> <td style="text-align: center;">F-210w ADD</td> <td style="text-align: center;">S₁</td> <td style="text-align: center;">S₂</td> <td style="text-align: center;">D</td> </tr> </table>					F-210w ADD	S ₁	S ₂	D										
F-210w ADD	S ₁	S ₂	D																
機能	レジスタS ₁ 、S ₁ +1の内容とレジスタS ₂ 、S ₂ +1の内容をバイナリ加算してレジスタD、D+1に格納する。																		
演算内容	(S ₁ 、S ₁ +1)+(S ₂ 、S ₂ +1)→D、D+1																		
S ₁ の使用範囲	<table border="0" style="width: 100%;"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771#1 000000~037776</td> <td>771#1 @000000~@037774</td> </tr> </table>					コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771#1 000000~037776	771#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																		
コ2000~コ7576	@コ2000~@コ7574																		
b0000~b1776	@b0000~@b1774																		
b2000~b3776	@b2000~@b3774																		
09000~99776	@09000~@99774																		
E0000~E7776	@E0000~@E7774																		
771#1 000000~037776	771#1 @000000~@037774																		
S ₂ の使用範囲	<table border="0" style="width: 100%;"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771#1 000000~037776</td> <td>771#1 @000000~@037774</td> </tr> </table>					コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771#1 000000~037776	771#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																		
コ2000~コ7576	@コ2000~@コ7574																		
b0000~b1776	@b0000~@b1774																		
b2000~b3776	@b2000~@b3774																		
09000~99776	@09000~@99774																		
E0000~E7776	@E0000~@E7774																		
771#1 000000~037776	771#1 @000000~@037774																		
Dの使用範囲	<table border="0" style="width: 100%;"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771#1 000000~037776</td> <td>771#1 @000000~@037774</td> </tr> </table>					コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771#1 000000~037776	771#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																		
コ2000~コ7576	@コ2000~@コ7574																		
b0000~b1776	@b0000~@b1774																		
b2000~b3776	@b2000~@b3774																		
09000~99776	@09000~@99774																		
E0000~E7776	@E0000~@E7774																		
771#1 000000~037776	771#1 @000000~@037774																		
演算条件	入力信号の立上り(OFF→ON)																		
演算	S ₁ 、S ₁ +1の内容	不変																	
	S ₂ 、S ₂ +1の内容	不変																	
後フラグ	Dの内容	演算結果(下位)																	
	D+1の内容	演算結果(上位)																	
後フラグ	演算結果(8進)	ゼロ 07357	キャリー 07356	エラー 07355	ハキャリー 07354														
	0	1	0	0	1														
	000001~17777	0	0	0	1														
	200000	1	1	0	0														
200001以上	0	1	0	0															

(解説)



命 令	
STR	04001
F-210w	ADD
	09000
	09010
	09020

入力条件04001がOFF→ONの変化時に、レジスタ09000、09001の内容とレジスタ09010、09011の内容をバイナリ加算して結果をレジスタ09020、09021に格納します。



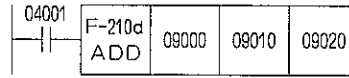
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S₁、S₂、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

**F-210d
ADD**

**レジスタ間のバイナリ加算(32ビット+32ビット)
(ADD)**

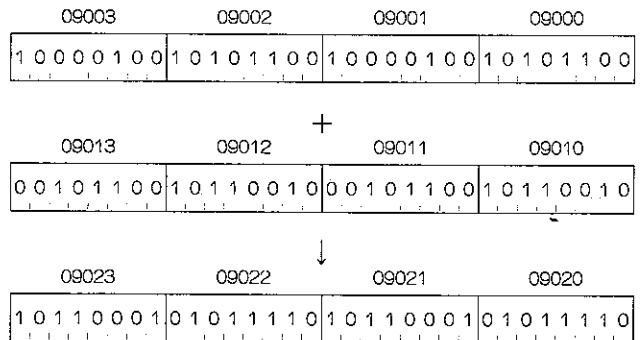
シンボル					
機能	レジスタS ₁ ~S ₁ +3の内容とレジスタS ₂ ~S ₂ +3の内容をバイナリ加算してレジスタD~D+3に格納する。				
演算内容	(S ₁ ~S ₁ +3)+(S ₂ ~S ₂ +3)→D~D+3				
S ₁ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774			
S ₂ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774			
Dの使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774			
演算条件	入力信号の立上り(OFF→ON)				
演算後	S ₁ ~S ₁ +3の内容	不変			
	S ₂ ~S ₂ +3の内容	不変			
	D~D+3の内容	演算結果 (バイナリ32ビット)			
		演算結果(8進)	ゼロ 07357	キャリー 07356	エラー 07355
	0	1	0	0	1
	0~3777777777 ⁽⁸⁾	0	0	0	1
	4000000000 ⁽⁸⁾	1	1	0	0
	40000000001 ⁽⁸⁾ 以上	0	1	0	0

(解説)



命令	
STR	04001
F-210d	09000 09010 09020

入力条件04001がOFF→ONの変化時に、レジスタ09000~09003の内容とレジスタ09010~09013の内容をバイナリ加算して結果をレジスタ09020~09023に格納します。



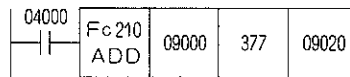
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S₁、S₂、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

Fc210
ADD

レジスタと定数のバイナリ加算(8ビット+8ビット)
(ADD)

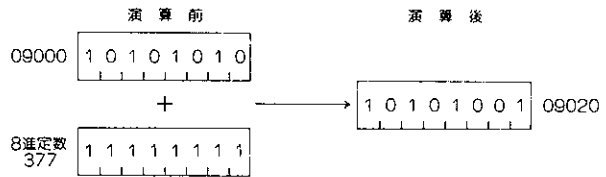
シンボル	<table border="1"> <tr> <td>Fc210 ADD</td> <td>S_i</td> <td>n</td> <td>D</td> </tr> </table>				Fc210 ADD	S _i	n	D
Fc210 ADD	S _i	n	D					
機能	レジスタS _i の内容と8進定数nの内容をバイナリ加算してレジスタDに格納する。							
演算内容	S _i +n→D							
S _i の使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~09977 : @09000~@09974 E0000~E7777 : @E0000~@E7774 771#1 000000~037777 : 771#1 @000000~@037774							
nの使用範囲	000~377(8)							
Dの使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~09977 : @09000~@09974 E0000~E7777 : @E0000~@E7774 771#1 000000~037777 : 771#1 @000000~@037774							
演算条件	入力信号の立上り(OFF→ON)							
演算後	S _i の内容	不変						
	Dの内容	演算結果						
フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354			
	0	1	0	0	1			
	001~377 ₍₈₎	0	0	0	1			
	400 ₍₈₎	1	1	0	0			
401 ₍₈₎ 以上	0	1	0	0				

(解説)



命 令	
STR	04000
Fc210	09000
	377
	09020

入力条件04000がOFF→ONの変化時に、レジスタ09000の内容と8進定数377をバイナリ加算して結果をレジスタ09020に格納します。



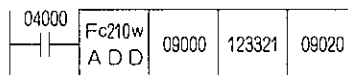
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

**Fc210w
ADD**

**レジスタと定数のバイナリ加算(16ビット+16ビット)
(ADD)**

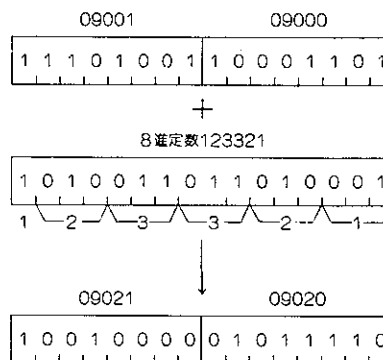
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">Fc210w ADD</td> <td style="padding: 2px;">S_i</td> <td style="padding: 2px;">n</td> <td style="padding: 2px;">D</td> </tr> </table>					Fc210w ADD	S _i	n	D										
Fc210w ADD	S _i	n	D																
機能	レジスタS _i 、S _i +1の内容と8進定数nの内容をバイナリ加算してレジスタD、D+1に格納する。																		
演算内容	(S _i 、S _i +1)+n→D、D+1																		
S _i の使用範囲	<table style="font-size: small; border-collapse: collapse;"> <tr> <td style="padding: 2px;">コ0000~コ1576</td> <td style="padding: 2px;">@コ0000~@コ1574</td> </tr> <tr> <td style="padding: 2px;">コ2000~コ7576</td> <td style="padding: 2px;">@コ2000~@コ7574</td> </tr> <tr> <td style="padding: 2px;">b0000~b1776</td> <td style="padding: 2px;">@b0000~@b1774</td> </tr> <tr> <td style="padding: 2px;">b2000~b3776</td> <td style="padding: 2px;">@b2000~@b3774</td> </tr> <tr> <td style="padding: 2px;">09000~99776</td> <td style="padding: 2px;">@09000~@99774</td> </tr> <tr> <td style="padding: 2px;">E0000~E7776</td> <td style="padding: 2px;">@E0000~@E7774</td> </tr> <tr> <td style="padding: 2px;">771#1 000000~037776</td> <td style="padding: 2px;">771#1 @000000~@037774</td> </tr> </table>					コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771#1 000000~037776	771#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																		
コ2000~コ7576	@コ2000~@コ7574																		
b0000~b1776	@b0000~@b1774																		
b2000~b3776	@b2000~@b3774																		
09000~99776	@09000~@99774																		
E0000~E7776	@E0000~@E7774																		
771#1 000000~037776	771#1 @000000~@037774																		
nの使用範囲	000000~177777(8)																		
Dの使用範囲	<table style="font-size: small; border-collapse: collapse;"> <tr> <td style="padding: 2px;">コ0000~コ1576</td> <td style="padding: 2px;">@コ0000~@コ1574</td> </tr> <tr> <td style="padding: 2px;">コ2000~コ7576</td> <td style="padding: 2px;">@コ2000~@コ7574</td> </tr> <tr> <td style="padding: 2px;">b0000~b1776</td> <td style="padding: 2px;">@b0000~@b1774</td> </tr> <tr> <td style="padding: 2px;">b2000~b3776</td> <td style="padding: 2px;">@b2000~@b3774</td> </tr> <tr> <td style="padding: 2px;">09000~99776</td> <td style="padding: 2px;">@09000~@99774</td> </tr> <tr> <td style="padding: 2px;">E0000~E7776</td> <td style="padding: 2px;">@E0000~@E7774</td> </tr> <tr> <td style="padding: 2px;">771#1 000000~037776</td> <td style="padding: 2px;">771#1 @000000~@037774</td> </tr> </table>					コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771#1 000000~037776	771#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																		
コ2000~コ7576	@コ2000~@コ7574																		
b0000~b1776	@b0000~@b1774																		
b2000~b3776	@b2000~@b3774																		
09000~99776	@09000~@99774																		
E0000~E7776	@E0000~@E7774																		
771#1 000000~037776	771#1 @000000~@037774																		
演算条件	入力信号の立上り(OFF→ON)																		
演算	S _i 、S _i +1の内容	不変																	
	Dの内容	演算結果(下位)																	
	D+1の内容	演算結果(上位)																	
後	演算結果(8進)	ゼロ 07357	キャリー 07356	エラー 07355	ハキャリー 07354														
	0	1	0	0	1														
	000001~17777	0	0	0	1														
	200000	1	1	0	0														
	200001以上	0	1	0	0														

(解説)



命令	
STR	04000
Fc210w	
	09000
	123321
	09020

入力条件04000がOFF→ONの変化時に、レジスタ09000、09001の内容と8進定数123321をバイナリ加算して結果をレジスタ09020、09021に格納します。



- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S_i、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

**Fc210d
ADD**

**レジスタと定数のバイナリ加算(32ビット+16ビット)
(ADD)**

シンボル					
機能	レジスタ $S_1 \sim S_1+3$ の内容と8進定数 n の内容をバイナリ加算してレジスタ $D \sim D+3$ に格納する。				
演算内容	$(S_1 \sim S_1+3) + n \rightarrow D \sim D+3$				
S_1 の使用範囲	$\begin{matrix} \text{C}0000 \sim \text{C}1574 & @\text{C}0000 \sim @\text{C}1574 \\ \text{C}2000 \sim \text{C}7574 & @\text{C}2000 \sim @\text{C}7574 \\ \text{b}0000 \sim \text{b}1774 & @\text{b}0000 \sim @\text{b}1774 \\ \text{b}2000 \sim \text{b}3774 & @\text{b}2000 \sim @\text{b}3774 \\ \text{O}9000 \sim \text{O}9774 & @\text{O}9000 \sim @\text{O}9774 \\ \text{E}0000 \sim \text{E}7774 & @\text{E}0000 \sim @\text{E}7774 \\ \text{7777} \text{H} \text{000000} \sim \text{O}37774 & \text{7777} \text{H} \text{@} \text{000000} \sim @\text{O}37774 \end{matrix}$				
n の使用範囲	000000 \sim 177777 $_{(8)}$				
D の使用範囲	$\begin{matrix} \text{C}0000 \sim \text{C}1574 & @\text{C}0000 \sim @\text{C}1574 \\ \text{C}2000 \sim \text{C}7574 & @\text{C}2000 \sim @\text{C}7574 \\ \text{b}0000 \sim \text{b}1774 & @\text{b}0000 \sim @\text{b}1774 \\ \text{b}2000 \sim \text{b}3774 & @\text{b}2000 \sim @\text{b}3774 \\ \text{O}9000 \sim \text{O}9774 & @\text{O}9000 \sim @\text{O}9774 \\ \text{E}0000 \sim \text{E}7774 & @\text{E}0000 \sim @\text{E}7774 \\ \text{7777} \text{H} \text{000000} \sim \text{O}37774 & \text{7777} \text{H} \text{@} \text{000000} \sim @\text{O}37774 \end{matrix}$				
演算条件	入力信号の立ち上り (OFF \rightarrow ON)				
演算後	$S_1 \sim S_1+3$ の内容	不変			
	$D \sim D+3$ の内容	演算結果(バイナリ32ビット)			
	演算結果(8進)	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
フ	0	1	0	0	1
ラ	1 \sim 3777777777 $_{(8)}$	0	0	0	1
	4000000000 $_{(8)}$	1	1	0	0
グ	4000000001 $_{(8)}$	0	1	0	0

(解説)

命令	
STR	04000
Fc210d	09000
	123321
	09020

入力条件04000がOFF \rightarrow ONの変化時に、レジスタ09000 \sim 09003の内容と8進定数123321をバイナリ加算して結果をレジスタ09020 \sim 09023に格納します。

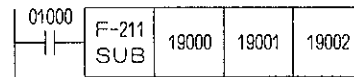
- C0734 \sim C0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S_1 、 D には必ず偶数アドレスを設定してください。
(C0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

**F-211
SUB**

**レジスタ間のバイナリ減算(8ビット-8ビット)
(SUBtract)**

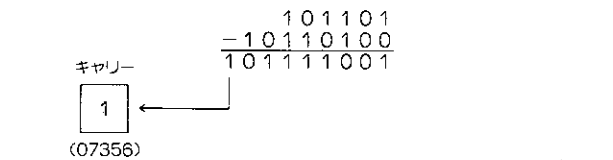
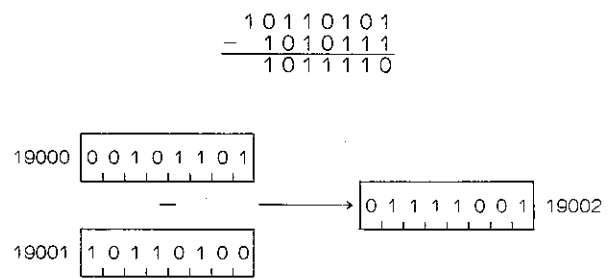
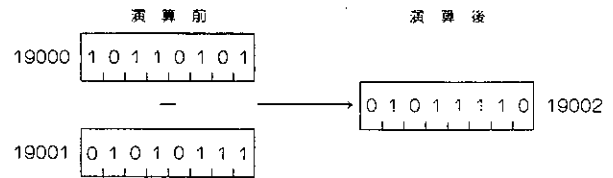
シンボル					
機能	レジスタS ₁ の内容からレジスタS ₂ の内容をバイナリ減算してレジスタDに格納する。				
演算内容	S ₁ - S ₂ → D				
S ₁ の使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~99777 @090000~@99774 E00000~E7777 @E00000~@E7774 771M1 000000~037777 771M1 @000000~@037774				
S ₂ の使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~99777 @090000~@99774 E00000~E7777 @E00000~@E7774 771M1 000000~037777 771M1 @000000~@037774				
Dの使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~99777 @090000~@99774 E00000~E7777 @E00000~@E7774 771M1 000000~037777 771M1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	S ₁ の内容	不変			
	S ₂ の内容	不変			
	Dの内容	演算結果			
フラグ	演算結果	ゼロ	キャリー	エラー	ノキャリー
	0	1	0	0	1
	1~377(※)	0	0	0	1
	負の数値	0	1	0	0

(解説)



命令	
STR	01000
F-211	19000
	19001
	19002

入力条件01000がOFF→ONの変化時に、レジスタ19000の内容からレジスタ19001の内容をバイナリ減算してレジスタ19002に格納します。



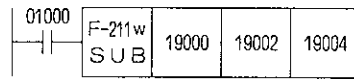
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

F-211w
SUB

レジスタ間のバイナリ減算(16ビット-16ビット)
(SUBtract)

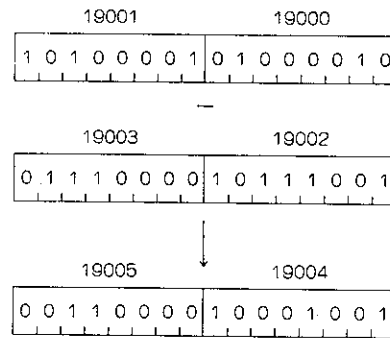
シンボル					
機能	レジスタS ₁ 、S ₁ +1の内容からレジスタS ₂ 、S ₂ +1の内容をバイナリ減算してレジスタD、D+1に格納する。				
演算内容	(S ₁ 、S ₁ +1)-(S ₂ 、S ₂ +1)→D、D+1				
S ₁ の使用範囲	コ00000~コ15776 : @コ00000~@コ15774 コ20000~コ75776 : @コ20000~@コ75774 b00000~b17776 : @b00000~@b17774 b20000~b37776 : @b20000~@b37774 090000~99776 : @090000~@99774 E00000~E7776 : @E00000~@E7774 771#1 000000~037776 : 771#1 @000000~@037774				
S ₂ の使用範囲	コ00000~コ15776 : @コ00000~@コ15774 コ20000~コ75776 : @コ20000~@コ75774 b00000~b17776 : @b00000~@b17774 b20000~b37776 : @b20000~@b37774 090000~99776 : @090000~@99774 E00000~E7776 : @E00000~@E7774 771#1 000000~037776 : 771#1 @000000~@037774				
Dの使用範囲	コ00000~コ15776 : @コ00000~@コ15774 コ20000~コ75776 : @コ20000~@コ75774 b00000~b17776 : @b00000~@b17774 b20000~b37776 : @b20000~@b37774 090000~99776 : @090000~@99774 E00000~E7776 : @E00000~@E7774 771#1 000000~037776 : 771#1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算	S ₁ 、S ₁ +1の内容	不変			
	S ₂ 、S ₂ +1の内容	不変			
	Dの内容	演算結果(下位)			
	D+1の内容	演算結果(上位)			
後フラグ	演算結果(8進)	ゼロ	キャリー	エラー	ノキャリー
	0	07357	07356	07355	07354
	1~177777	0	0	0	1
負の数値	0	1	0	0	

(解説)



命 令	
STR	01000
F-211w	
	19000
	19002
	19004

入力条件01000がOFF→ONの変化時に、レジスタ19000、19001の内容からレジスタ19002、19003の内容をバイナリ減算して結果をレジスタ19004、19005に格納します。



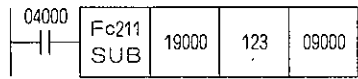
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S₁、S₂、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

**Fc211
SUB**

**レジスタと定数のバイナリ減算(8ビット-8ビット)
(SUBtract)**

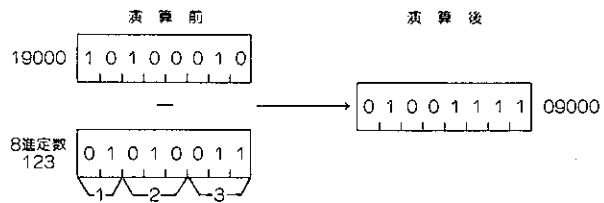
シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">Fc211 SUB</td> <td style="padding: 2px;">S_i</td> <td style="padding: 2px;">n</td> <td style="padding: 2px;">D</td> </tr> </table>				Fc211 SUB	S _i	n	D
Fc211 SUB	S _i	n	D					
機能	レジスタS _i の内容から8進定数nをバイナリ減算してレジスタDに格納する。							
演算内容	S _i - n → D							
S _i の使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~997777 @090000~@997774 E00000~E77777 @E00000~@E77774 771#1 000000~037777 771#1 @000000~@037774							
nの使用範囲	000~377(8)							
Dの使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~997777 @090000~@997774 E00000~E77777 @E00000~@E77774 771#1 000000~037777 771#1 @000000~@037774							
演算条件	入力信号の立上り(OFF→ON)							
演算後	S _i の内容	不変						
	Dの内容	演算結果						
フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354			
	0	1	0	0	1			
	1~377(8)	0	0	0	1			
	負の数値	0	1	0	0			

(解説)



命令	
STR	04000
Fc211	19000
	123
	09000

入力条件04000がOFF→ONの変化時に、レジスタ19000の内容から8進定数123をバイナリ減算して結果をレジスタ09000に格納します。



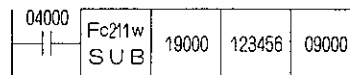
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

Fc211w
SUB

レジスタと定数のバイナリ減算(16ビット-16ビット)
(SUBtract)

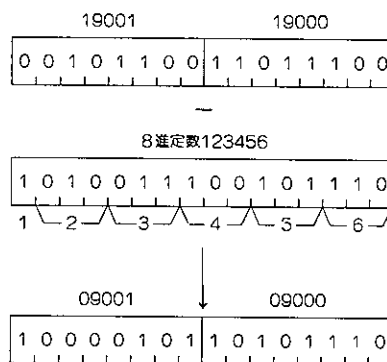
シンボル	— Fc211w SUB S _i n D				
機能	レジスタS _i 、S _i +1の内容から8進定数nをバイナリ減算して、レジスタD、D+1に格納する。				
演算内容	(S _i , S _i +1) - n → D, D+1				
S _i の使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 771#1 000000~037776 : 771#1 @000000~@037774				
nの使用範囲	000000~177777(8)				
Dの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 771#1 000000~037776 : 771#1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	S _i , S _i +1の内容	不変			
	Dの内容	演算結果(下位)			
	D+1の内容	演算結果(上位)			
フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
	0	1	0	0	1
	1~177777(8)	0	0	0	1
	負の数値	0	1	0	0

(解説)



命 令	
STR	04000
Fc211w	19000 123456 09000

入力条件04000がOFF→ONの変化時に、レジスタ19000、19001の内容から8進定数123456をバイナリ減算して、結果をレジスタ09000、09001に格納します。



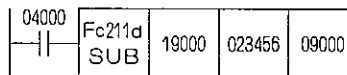
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S_i、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

Fc211d
SUB

レジスタと定数のバイナリ減算(32ビット-16ビット)
(SUBtract)

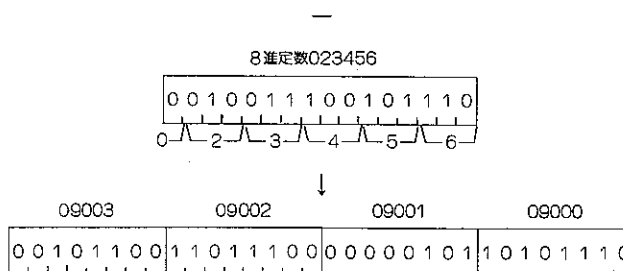
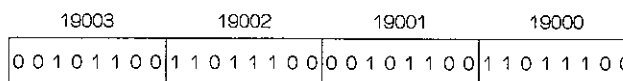
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>Fc211d</td><td>S_i</td><td>n</td><td>D</td></tr> </table>				Fc211d	S _i	n	D
Fc211d	S _i	n	D					
機能	レジスタS _i ~S _i +3の内容から8進定数nをバイナリ減算して、レジスタD~D+3に格納する。							
演算内容	(S _i ~S _i +3) - n → D~D+3							
S _i の使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 77IM1 000000~037774 : 77IM1 @000000~@037774							
nの使用範囲	000000~177777(8)							
Dの使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 77IM1 000000~037774 : 77IM1 @000000~@037774							
演算条件	入力信号の立上り(OFF→ON)							
演算後	S _i ~S _i +3の内容	不変						
	D~D+3の内容	演算結果(バイナリ32ビット)						
フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354			
	0	1	0	0	1			
	1~3777777777 _(n)	0	0	0	1			
	負の数値	0	1	0	0			

(解説)



命令	
STR	04000
Fc211d	19000
	023456
	09000

入力条件04000がOFF→ONの変化時に、レジスタ19000~19003の内容から8進定数023456をバイナリ減算して、結果をレジスタ09000~09003に格納します。



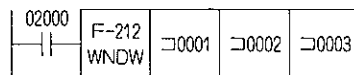
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S_i、Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

F-212
WNDW

ウィンドウコンパレータ(1バイトレジスタ間)

シンボル						
機能	レジスタS ₁ とレジスタS ₂ 、レジスタS ₃ の内容を比較し、比較結果をフラグに格納する。					
演算内容	比較結果→フラグ					
S ₁ の使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~997777 @090000~@997774 E00000~E77777 @E00000~@E77774 771#1 000000~037777 771#1 @000000~@037774					
S ₂ の使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~997777 @090000~@997774 E00000~E77777 @E00000~@E77774 771#1 000000~037777 771#1 @000000~@037774					
S ₃ の使用範囲	コ00000~コ15777 @コ00000~@コ15774 コ20000~コ75777 @コ20000~@コ75774 b00000~b17777 @b00000~@b17774 b20000~b37777 @b20000~@b37774 090000~997777 @090000~@997774 E00000~E77777 @E00000~@E77774 771#1 000000~037777 771#1 @000000~@037774					
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)					
演算後	S ₁ の内容	不変				
	S ₂ の内容	不変				
	S ₃ の内容	不変				
	フラグ	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
		S ₂ < S ₁	0	1	0	0
S ₂ > S ₁		1	0	0	0	
S ₂ < S ₃		0	0	0	1	
S ₁ < S ₂	0	0	1	0		

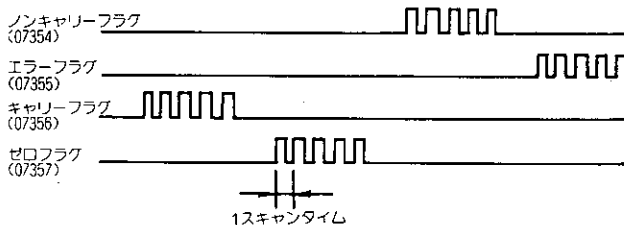
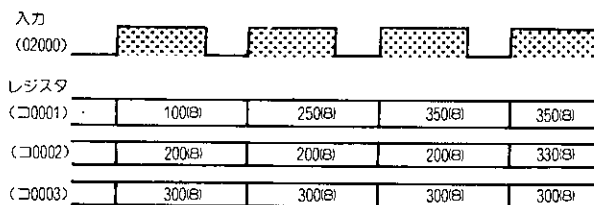
(解説)



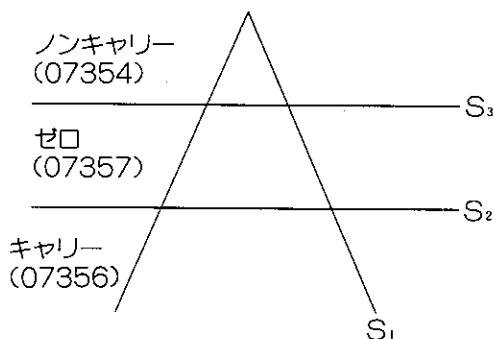
命令	
STR	02000
F-212	コ0001 コ0002 コ0003

入力条件02000がONの時、レジスタコ0001の内容がコ0001 < コ0002か、コ0002 ≤ コ0001 ≤ コ0003か、コ0003 < コ0001かなどの範囲に入っているかを演算し、その結果を、キャリーフラグ(07356)、ゼロフラグ(07357)、ノンキャリーフラグ(07354)に設定します。

コ0002 ≤ コ0003の条件の場合のみ演算し、コ0003 < コ0002の場合は、演算を中止し、エラーフラグ(07355)をONします。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 入力信号がONの間、毎スキャンサイクル演算を実行します。(9・7ページ「演算実行条件」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

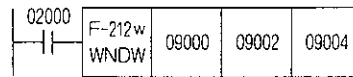


**F-212w
WNDW**

ウィンドウコンパレータ(1ワードレジスタ間)

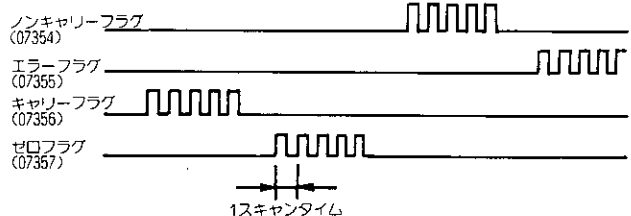
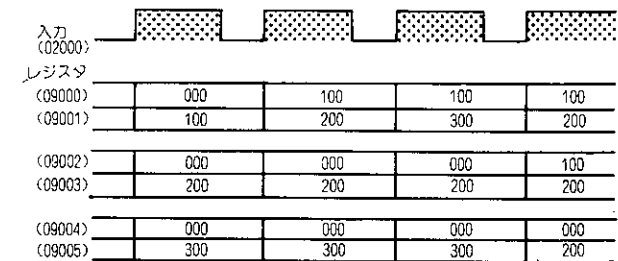
シンボル	— F-212w WNDW S ₁ S ₂ S ₃					
機能	レジスタS ₁ 、S ₁ +1とレジスタS ₂ 、S ₂ +1、レジスタS ₃ 、S ₃ +1の内容を比較し、比較結果をフラグに格納する。					
演算内容	比較結果→フラグ					
S ₁ の使用範囲	C0000~C1576 C2000~C7576 b0000~b1776 b2000~b3776 09000~99776 E0000~E7776 771#1 000000~037776		@C00000~@C1574 @C20000~@C7574 @b00000~@b1774 @b20000~@b3774 @090000~@99774 @E00000~@E7774 771#1 @000000~@037774			
S ₂ の使用範囲	C0000~C1576 C2000~C7576 b0000~b1776 b2000~b3776 09000~99776 E0000~E7776 771#1 000000~037776		@C00000~@C1574 @C20000~@C7574 @b00000~@b1774 @b20000~@b3774 @090000~@99774 @E00000~@E7774 771#1 @000000~@037774			
S ₃ の使用範囲	C0000~C1576 C2000~C7576 b0000~b1776 b2000~b3776 09000~99776 E0000~E7776 771#1 000000~037776		@C00000~@C1574 @C20000~@C7574 @b00000~@b1774 @b20000~@b3774 @090000~@99774 @E00000~@E7774 771#1 @000000~@037774			
演算条件	入力信号がONの時 (OFF→ONの変化時に限定されない)					
演算	S ₁ 、S ₁ +1の内容	不変				
	S ₂ 、S ₂ +1の内容	不変				
	S ₃ 、S ₃ +1の内容	不変				
後	フラグ	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
		S ₁ < S ₁ +1	0	1	0	0
		S ₁ +1 < S ₁	1	0	0	0
		S ₁ +1 < S ₁ +1	0	0	0	1
		S ₁ < S ₁ +1	0	0	1	0

(解説)



命令	
STR	02000
F-212w	
	09000
	09002
	09004

入力条件02000がONの時、レジスタ09000、09001の1ワードデータの内容(09000、09001)が(09000、09001)<(09002、09003)か(09002、09003)≤(09000、09001)≤(09004、09005)か(09004、09005)<(09000、09001)かのどの範囲に入っているかを演算し、その結果をキャリーフラグ(07356)、ゼロフラグ(07357)、ノンキャリーフラグ(07354)に設定します。
(09002、09003)≤(09004、09005)の条件の場合のみ演算し、(09004、09005)<(09002、09003)の場合は演算を中止し、エラーフラグ(07355)をONします。



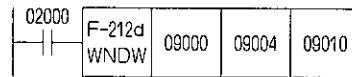
- C0734~C0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- S₁、S₂、S₃には必ず偶数アドレスを設定してください。(C0011、19003等は禁止)
- 入力信号がONの間、毎スキャンサイクル演算を実行します。(9・7ページ「演算実行条件」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

F-212d
WNDW

ウィンドウコンパレータ(2ワードレジスタ間)

シンボル	F-212d WNDW S ₁ S ₂ S ₃					
機能	レジスタS ₁ ~S ₁ +3とレジスタS ₂ ~S ₂ +3、レジスタS ₃ ~S ₃ +3の内容を比較し、比較結果をフラグに格納する。					
演算内容	比較結果→フラグ					
S ₁ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774				
S ₂ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774				
S ₃ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774				
演算条件	入力信号がONの時 (OFF→ONの変化時に限定されない)					
演算後	S ₁ ~S ₁ +3の内容	不変				
	S ₂ ~S ₂ +3の内容	不変				
	S ₃ ~S ₃ +3の内容	不変				
	フ	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
	フ	S ₁ ~S ₁ +3 < S ₂ ~S ₂ +3	0	1	0	0
グ	S ₂ ~S ₂ +3 ≤ S ₁ ~S ₁ +3 ≤ S ₃ ~S ₃ +3	1	0	0	0	
グ	S ₁ ~S ₁ +3 < S ₁ ~S ₁ +3	0	0	0	1	
グ	S ₁ ~S ₁ +3 < S ₂ ~S ₂ +3	0	0	1	0	

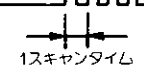
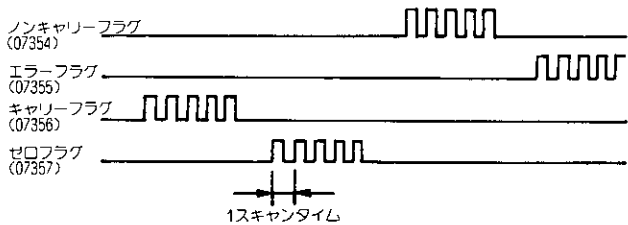
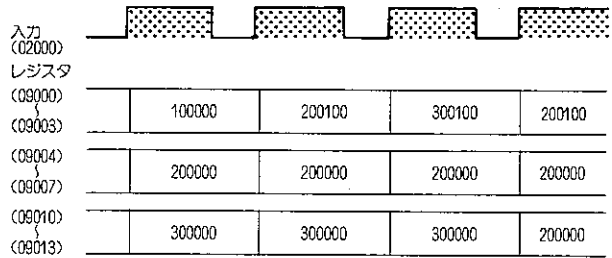
(解説)



命令	
STR	02000
F-212d	09000
	09004
	09010

入力条件02000がONの時、レジスタ09000~09003の2ワードデータの内容(09000~09003)が(09000~09003) < (09004~09007) か (09004~09007) ≤ (09000~09003) ≤ (09010~09013) か (09010~09013) < (09000~09003) かのどの範囲に入っているかを演算し、その結果をキャリーフラグ(07354)に設定します。

(09004~09007) ≤ (09010~09013) の条件の場合のみ演算し、(09010~09013) < (09004~09007) の場合は演算を中止し、エラーフラグ(07355)をONします。



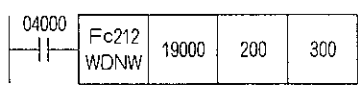
- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- S₁、S₂、S₃には必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- 入力信号がONの間、毎スキャンサイクル演算を実行します。(9・7ページ「演算実行条件」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

**Fc212
WNDW**

ウィンドウコンパレータ(1バイト8進定数間)

シンボル					
機能	レジスタS ₁ の内容と8進定数n ₁ 、n ₂ を比較し、比較結果をフラグに設定する。				
演算内容	比較結果→フラグ				
S ₁ の使用範囲	コ00000~コ15777 : @コ00000~@コ15774 コ20000~コ75777 : @コ20000~@コ75774 b00000~b17777 : @b00000~@b17774 b20000~b37777 : @b20000~@b37774 090000~997777 : @090000~@997774 E00000~E77777 : @E00000~@E77774 771#1 000000~037777 : 771#1 @000000~@037774				
n ₁ の使用範囲	000~377(8)				
n ₂ の使用範囲	000~377(8)				
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)				
演算後	S ₁ の内容	不変			
	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
	S ₁ < n ₁	0	1	0	0
	n ₁ ≤ S ₁ ≤ n ₂	1	0	0	0
	n ₂ < S ₁	0	0	0	1
n ₂ < n ₁	0	0	1	0	

(解説)

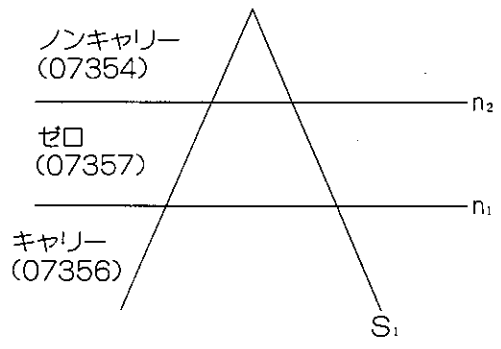


命令	
STR	04000
Fc212	19000
	200
	300

入力条件04000がONの時、レジスタ19000の内容(19000)が(19000) < 200か 200 ≤ (19000) ≤ 300か 300 < (19000) かのどの範囲に入っているかを演算し、その結果をキャリーフラグ(07356)、ゼロフラグ(07357)、ノンキャリーフラグ(07354)に設定します。

19000の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
150	0	1	0	0
250	1	0	0	0
350	0	0	0	1

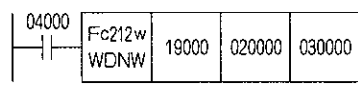
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 入力信号がONの間、毎スキャンサイクル演算を実行します。(9・7ページ「演算実行条件」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)



**Fc212w
WNDW**

ウィンドウコンパレータ(1ワード8進定数間)

シンボル	<table border="1"> <tr> <td>Fc212w WNDW</td> <td>S_i</td> <td>n₁</td> <td>n₂</td> </tr> </table>				Fc212w WNDW	S _i	n ₁	n ₂	(解説)	<table border="1"> <tr> <th colspan="2">命 令</th> </tr> <tr> <td>STR</td> <td>04000</td> </tr> <tr> <td>Fc212w</td> <td>19000</td> </tr> <tr> <td></td> <td>020000</td> </tr> <tr> <td></td> <td>030000</td> </tr> </table>	命 令		STR	04000	Fc212w	19000		020000		030000
Fc212w WNDW	S _i	n ₁	n ₂																	
命 令																				
STR	04000																			
Fc212w	19000																			
	020000																			
	030000																			
機能	レジスタS _i 、S _i +1の内容(1ワードデータ)と8進定数n ₁ 、n ₂ を比較し、比較結果をフラグに設定する。																			
演算内容	比較結果→フラグ																			
S _i の使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771#1 000000~037776</td> <td>771#1 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771#1 000000~037776	771#1 @000000~@037774		
コ0000~コ1576	@コ0000~@コ1574																			
コ2000~コ7576	@コ2000~@コ7574																			
b0000~b1776	@b0000~@b1774																			
b2000~b3776	@b2000~@b3774																			
09000~99776	@09000~@99774																			
E0000~E7776	@E0000~@E7774																			
771#1 000000~037776	771#1 @000000~@037774																			
n ₁ の使用範囲	000000~177777(8)																			
n ₂ の使用範囲	000000~177777(8)																			
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)																			
演算後	S _i 、S _i +1の内容	不変																		
	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354															
	S _i 、S _i +1 < n ₁	0	1	0	0															
	n ₁ ≤ S _i 、S _i +1 ≤ n ₂	1	0	0	0															
	n ₂ < S _i 、S _i +1	0	0	0	1															
	n ₂ < n ₁	0	0	1	0															



入力条件04000がONの時、レジスタ19000、19001の1ワードデータの内容(19000、19001)が、(19000、19001) < 020000か020000 ≤ (19000、19001) ≤ 030000か030000 < (19000、19001)かのどの範囲に入っているかを演算し、その結果をキャリーフラグ(07356)、ゼロフラグ(07357)、ノンキャリーフラグ(07354)に設定します。

19000、19001の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
015000	0	1	0	0
025000	1	0	0	0
035000	0	0	0	1

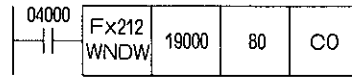
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 入力信号がONの間、毎スキャンサイクル演算を実行します。(9・7ページ「演算実行条件」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。
(9・7ページ「データ処理命令とフラグ」参照)

**Fx212
WNDW**

ウィンドウコンパレータ(1バイト16進定数間)

シンボル	— Fx212 WNDW S ₁ n ₁ n ₂					
機能	レジスタS ₁ の内容と16進定数n ₁ 、n ₂ を比較し、比較結果をフラグに設定する。					
演算内容	比較結果→フラグ					
S ₁ の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~09777 @09000~@09774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774					
n ₁ の使用範囲	00~FF					
n ₂ の使用範囲	00~FF					
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)					
演算後	S ₁ の内容	不変				
	フラグ	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
		S ₁ < n ₁	0	1	0	0
		n ₁ ≤ S ₁ ≤ n ₂	1	0	0	0
		n ₂ < S ₁	0	0	0	1
		n ₂ < n ₁	0	0	1	0

(解説)

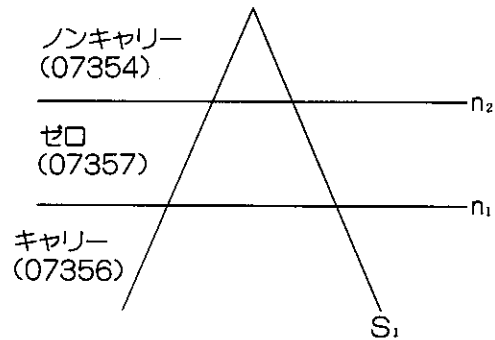


命令	
STR	04000
Fx212	19000
	80
	CO

入力条件 04000 が ON 時、レジスタ 19000 の内容 (19000) が (19000) < 80 か 80 ≤ (19000) ≤ C0 か C0 < (19000) かのどの範囲に入っているかを演算し、その結果をキャリーフラグ (07356)、ゼロフラグ (07357)、ノンキャリーフラグ (07354) に設定します。

19000の 内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
70	0	1	0	0
90	1	0	0	0
D0	0	0	0	1

- コ0734~コ0737は特殊領域です。
(2・5 ページ「特殊リレー」参照)
- 間接アドレス指定する場合必ず偶数アドレスを設定してください。
(@コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3 ページ「間接アドレス指定」の項をご参照ください。
- 入力信号がONの間、毎スキャンサイクル演算が実行されます。(9・7 ページ「演算実行条件」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。
(9・7 ページ「データ処理命令とフラグ」参照)

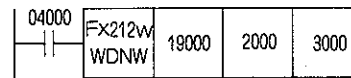


**Fx212w
WNDW**

ウィンドウコンパレータ(1ワード16進定数間)

シンボル	<table border="1"> <tr> <td>Fx212w WNDW</td> <td>S_i</td> <td>n₁</td> <td>n₂</td> </tr> </table>				Fx212w WNDW	S _i	n ₁	n ₂										
Fx212w WNDW	S _i	n ₁	n ₂															
機能	レジスタS _i 、S _i +1の内容(1ワードデータ)と16進定数n ₁ 、n ₂ を比較し、比較結果をフラグに設定する。																	
演算内容	比較結果→フラグ																	
S _i の使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771M1 000000~037776</td> <td>771M1 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	771M1 000000~037776	771M1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																	
コ2000~コ7576	@コ2000~@コ7574																	
b0000~b1776	@b0000~@b1774																	
b2000~b3776	@b2000~@b3774																	
09000~99776	@09000~@99774																	
E0000~E7776	@E0000~@E7774																	
771M1 000000~037776	771M1 @000000~@037774																	
n ₁ の使用範囲	0000~FFFF																	
n ₂ の使用範囲	0000~FFFF																	
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)																	
演算後	S _i 、S _i +1の内容	不変																
	レジスタの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354													
	S _i 、S _i +1 < n ₁	0	1	0	0													
	n ₁ ≤ S _i 、S _i +1 ≤ n ₂	1	0	0	0													
	n ₂ < S _i 、S _i +1	0	0	0	1													
n ₂ < n ₁	0	0	1	0														

(解説)



命 令	
STR	04000
Fx212w	19000
	2000
	3000

入力条件04000がONの時、レジスタ19000、19001の1ワードデータの内容(19000、19001)が、(19000、19000) < 2000か2000 ≤ (19000、19001) ≤ 3000か3000 < (19000、19001)かのどの範囲に入っているかを演算し、その結果をキャリーフラグ(07356)、ゼロフラグ(07357)、ノンキャリーフラグ(07354)に設定します。

19000、19001 の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
1500	0	1	0	0
2500	1	0	0	0
3500	0	0	0	1

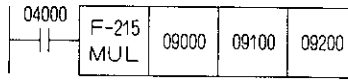
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- 間接アドレス指定する場合必ず偶数アドレスを設定してください。
(@コ0001、@b0173等は禁止)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- 入力信号がONの間、毎スキャンサイクル演算が実行されます。(9・7ページ「演算実行条件」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。
(9・7ページ「データ処理命令とフラグ」参照)

**F-215
MUL**

**レジスタ間のバイナリ乗算(8ビット×8ビット)
(MULtiply)**

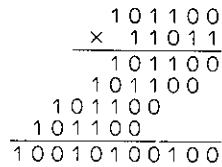
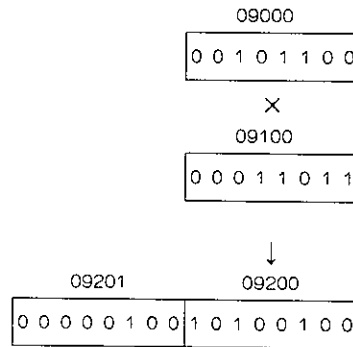
シンボル	<table border="1"> <tr> <td>F-215 MUL</td> <td>S₁</td> <td>S₂</td> <td>D</td> </tr> </table>				F-215 MUL	S ₁	S ₂	D																																						
F-215 MUL	S ₁	S ₂	D																																											
機能	レジスタS ₁ の内容とレジスタS ₂ の内容をバイナリ乗算してレジスタD、D+1に格納する。																																													
演算内容	S ₁ × S ₂ → D、D+1																																													
S ₁ の使用範囲	<table border="0"> <tr><td>コ0000</td><td>～</td><td>コ1577</td><td> @コ0000</td><td>～</td><td>@コ1574</td></tr> <tr><td>コ2000</td><td>～</td><td>コ7577</td><td> @コ2000</td><td>～</td><td>@コ7574</td></tr> <tr><td>ボ0000</td><td>～</td><td>ボ1777</td><td> @ボ0000</td><td>～</td><td>@ボ1774</td></tr> <tr><td>ボ2000</td><td>～</td><td>ボ3777</td><td> @ボ2000</td><td>～</td><td>@ボ3774</td></tr> <tr><td>ヨ9000</td><td>～</td><td>ヨ9777</td><td> @ヨ9000</td><td>～</td><td>@ヨ9774</td></tr> <tr><td>エ0000</td><td>～</td><td>エ7777</td><td> @エ0000</td><td>～</td><td>@エ7774</td></tr> <tr><td>771#1</td><td>000000</td><td>～037777</td><td> 771#1 @000000</td><td>～</td><td>@037774</td></tr> </table>				コ0000	～	コ1577	@コ0000	～	@コ1574	コ2000	～	コ7577	@コ2000	～	@コ7574	ボ0000	～	ボ1777	@ボ0000	～	@ボ1774	ボ2000	～	ボ3777	@ボ2000	～	@ボ3774	ヨ9000	～	ヨ9777	@ヨ9000	～	@ヨ9774	エ0000	～	エ7777	@エ0000	～	@エ7774	771#1	000000	～037777	771#1 @000000	～	@037774
コ0000	～	コ1577	@コ0000	～	@コ1574																																									
コ2000	～	コ7577	@コ2000	～	@コ7574																																									
ボ0000	～	ボ1777	@ボ0000	～	@ボ1774																																									
ボ2000	～	ボ3777	@ボ2000	～	@ボ3774																																									
ヨ9000	～	ヨ9777	@ヨ9000	～	@ヨ9774																																									
エ0000	～	エ7777	@エ0000	～	@エ7774																																									
771#1	000000	～037777	771#1 @000000	～	@037774																																									
S ₂ の使用範囲	<table border="0"> <tr><td>コ0000</td><td>～</td><td>コ1577</td><td> @コ0000</td><td>～</td><td>@コ1574</td></tr> <tr><td>コ2000</td><td>～</td><td>コ7577</td><td> @コ2000</td><td>～</td><td>@コ7574</td></tr> <tr><td>ボ0000</td><td>～</td><td>ボ1777</td><td> @ボ0000</td><td>～</td><td>@ボ1774</td></tr> <tr><td>ボ2000</td><td>～</td><td>ボ3777</td><td> @ボ2000</td><td>～</td><td>@ボ3774</td></tr> <tr><td>ヨ9000</td><td>～</td><td>ヨ9777</td><td> @ヨ9000</td><td>～</td><td>@ヨ9774</td></tr> <tr><td>エ0000</td><td>～</td><td>エ7777</td><td> @エ0000</td><td>～</td><td>@エ7774</td></tr> <tr><td>771#1</td><td>000000</td><td>～037777</td><td> 771#1 @000000</td><td>～</td><td>@037774</td></tr> </table>				コ0000	～	コ1577	@コ0000	～	@コ1574	コ2000	～	コ7577	@コ2000	～	@コ7574	ボ0000	～	ボ1777	@ボ0000	～	@ボ1774	ボ2000	～	ボ3777	@ボ2000	～	@ボ3774	ヨ9000	～	ヨ9777	@ヨ9000	～	@ヨ9774	エ0000	～	エ7777	@エ0000	～	@エ7774	771#1	000000	～037777	771#1 @000000	～	@037774
コ0000	～	コ1577	@コ0000	～	@コ1574																																									
コ2000	～	コ7577	@コ2000	～	@コ7574																																									
ボ0000	～	ボ1777	@ボ0000	～	@ボ1774																																									
ボ2000	～	ボ3777	@ボ2000	～	@ボ3774																																									
ヨ9000	～	ヨ9777	@ヨ9000	～	@ヨ9774																																									
エ0000	～	エ7777	@エ0000	～	@エ7774																																									
771#1	000000	～037777	771#1 @000000	～	@037774																																									
Dの使用範囲	<table border="0"> <tr><td>コ0000</td><td>～</td><td>コ1576</td><td> @コ0000</td><td>～</td><td>@コ1574</td></tr> <tr><td>コ2000</td><td>～</td><td>コ7576</td><td> @コ2000</td><td>～</td><td>@コ7574</td></tr> <tr><td>ボ0000</td><td>～</td><td>ボ1776</td><td> @ボ0000</td><td>～</td><td>@ボ1774</td></tr> <tr><td>ボ2000</td><td>～</td><td>ボ3776</td><td> @ボ2000</td><td>～</td><td>@ボ3774</td></tr> <tr><td>ヨ9000</td><td>～</td><td>ヨ9776</td><td> @ヨ9000</td><td>～</td><td>@ヨ9774</td></tr> <tr><td>エ0000</td><td>～</td><td>エ7776</td><td> @エ0000</td><td>～</td><td>@エ7774</td></tr> <tr><td>771#1</td><td>000000</td><td>～037776</td><td> 771#1 @000000</td><td>～</td><td>@037774</td></tr> </table>				コ0000	～	コ1576	@コ0000	～	@コ1574	コ2000	～	コ7576	@コ2000	～	@コ7574	ボ0000	～	ボ1776	@ボ0000	～	@ボ1774	ボ2000	～	ボ3776	@ボ2000	～	@ボ3774	ヨ9000	～	ヨ9776	@ヨ9000	～	@ヨ9774	エ0000	～	エ7776	@エ0000	～	@エ7774	771#1	000000	～037776	771#1 @000000	～	@037774
コ0000	～	コ1576	@コ0000	～	@コ1574																																									
コ2000	～	コ7576	@コ2000	～	@コ7574																																									
ボ0000	～	ボ1776	@ボ0000	～	@ボ1774																																									
ボ2000	～	ボ3776	@ボ2000	～	@ボ3774																																									
ヨ9000	～	ヨ9776	@ヨ9000	～	@ヨ9774																																									
エ0000	～	エ7776	@エ0000	～	@エ7774																																									
771#1	000000	～037776	771#1 @000000	～	@037774																																									
演算条件	入力信号の立上り(OFF→ON)																																													
演算後	S ₁ の内容	不変																																												
	S ₂ の内容	不変																																												
	Dの内容	演算結果(下位)																																												
	D+1の内容	演算結果(上位)																																												
フラグ	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354																																										
	0	0	0	0																																										

(解説)



命令	
STR	04000
F-215	09000
	09100
	09200

入力条件04000がOFF→ONの変化時に、レジスタ09000の内容とレジスタ09100の内容をバイナリ乗算して結果をレジスタ09200と09201に格納します。



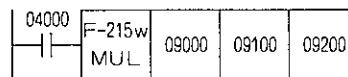
- コ0734～コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

**F-215w
MUL**

**レジスタ間のバイナリ乗算(16ビット×16ビット)
(MULtiply)**

シンボル	<table border="1"> <tr> <td>F-215w MUL</td> <td>S₁</td> <td>S₂</td> <td>D</td> </tr> </table>				F-215w MUL	S ₁	S ₂	D										
F-215w MUL	S ₁	S ₂	D															
機能	レジスタS ₁ 、S ₁ +1の内容とレジスタS ₂ 、S ₂ +1の内容をバイナリ乗算して、レジスタD、D+1、D+2、D+3に格納する。																	
演算内容	(S ₁ 、S ₁ +1)×(S ₂ 、S ₂ +1)→ D、D+1、D+2、D+3																	
S ₁ の使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>77771 000000~037776</td> <td>77771 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	77771 000000~037776	77771 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																	
コ2000~コ7576	@コ2000~@コ7574																	
b0000~b1776	@b0000~@b1774																	
b2000~b3776	@b2000~@b3774																	
09000~99776	@09000~@99774																	
E0000~E7776	@E0000~@E7774																	
77771 000000~037776	77771 @000000~@037774																	
S ₂ の使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1776</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3776</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99776</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7776</td> <td>@E0000~@E7774</td> </tr> <tr> <td>77771 000000~037776</td> <td>77771 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	b0000~b1776	@b0000~@b1774	b2000~b3776	@b2000~@b3774	09000~99776	@09000~@99774	E0000~E7776	@E0000~@E7774	77771 000000~037776	77771 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																	
コ2000~コ7576	@コ2000~@コ7574																	
b0000~b1776	@b0000~@b1774																	
b2000~b3776	@b2000~@b3774																	
09000~99776	@09000~@99774																	
E0000~E7776	@E0000~@E7774																	
77771 000000~037776	77771 @000000~@037774																	
Dの使用範囲	<table border="0"> <tr> <td>コ0000~コ1574</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7574</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1774</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3774</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99774</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7774</td> <td>@E0000~@E7774</td> </tr> <tr> <td>77771 000000~037774</td> <td>77771 @000000~@037774</td> </tr> </table>				コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	77771 000000~037774	77771 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																	
コ2000~コ7574	@コ2000~@コ7574																	
b0000~b1774	@b0000~@b1774																	
b2000~b3774	@b2000~@b3774																	
09000~99774	@09000~@99774																	
E0000~E7774	@E0000~@E7774																	
77771 000000~037774	77771 @000000~@037774																	
演算条件	入力信号の立上り(OFF→ON)																	
演算後	S ₁ 、S ₁ +1の内容	不変																
	S ₂ 、S ₂ +1の内容	不変																
	Dの内容	演算結果(下位)																
	D+1の内容	演算結果																
フラグ	D+2の内容	演算結果																
	D+3の内容	演算結果(上位)																
	ゼロ	キャリー	エラー	ノンキャリー														
	07357	07356	07355	07354														
	0	0	0	0														

(解説)



命令	
STR	04000
F-215w	09000
	09100
	09200

入力条件04000がOFF→ONの変化時に、レジスタ09000、09001の内容(16ビットデータ)とレジスタ09100、09101の内容(16ビットデータ)をバイナリ乗算して結果をレジスタ09200、09201、09202、09203に格納します。



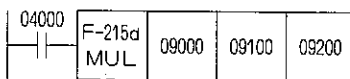
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S₁、S₂、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)

**F-215d
MUL**

**レジスタ間のバイナリ乗算(32ビット×32ビット)
(MULtiple)**

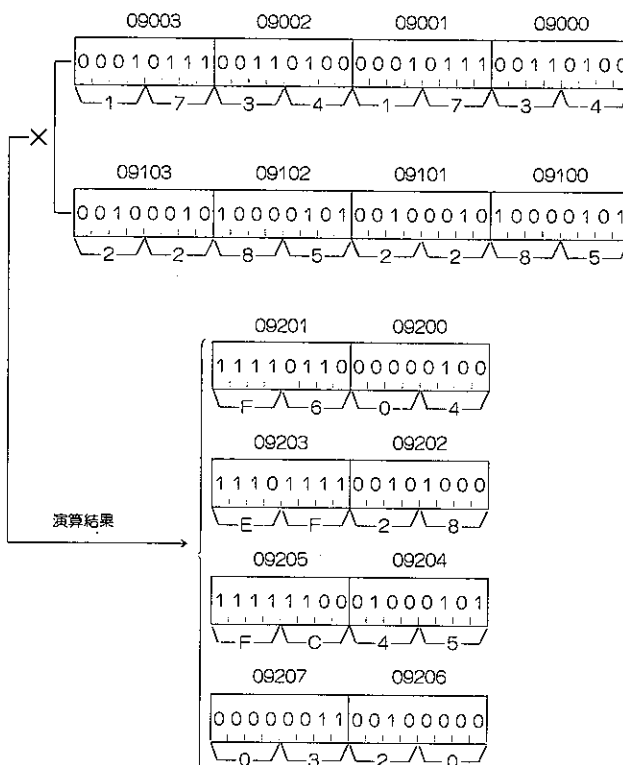
シンボル	<table border="1"> <tr> <td>F-215d MUL</td> <td>S₁</td> <td>S₂</td> <td>D</td> </tr> </table>				F-215d MUL	S ₁	S ₂	D										
F-215d MUL	S ₁	S ₂	D															
機能	レジスタS ₁ ~S ₁ +3の内容とレジスタS ₂ ~S ₂ +3の内容をバイナリ乗算して、レジスタD~D+7に格納する。																	
演算内容	(S ₁ ~S ₁ +3)×(S ₂ ~S ₂ +3)→D~D+7																	
S ₁ の使用範囲	<table border="0"> <tr> <td>コ0000~コ1574</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7574</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1774</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3774</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99774</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7774</td> <td>@E0000~@E7774</td> </tr> <tr> <td>774M1 000000~037774</td> <td>774M1 @000000~@037774</td> </tr> </table>				コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	774M1 000000~037774	774M1 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																	
コ2000~コ7574	@コ2000~@コ7574																	
b0000~b1774	@b0000~@b1774																	
b2000~b3774	@b2000~@b3774																	
09000~99774	@09000~@99774																	
E0000~E7774	@E0000~@E7774																	
774M1 000000~037774	774M1 @000000~@037774																	
S ₂ の使用範囲	<table border="0"> <tr> <td>コ0000~コ1574</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7574</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1774</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3774</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99774</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7774</td> <td>@E0000~@E7774</td> </tr> <tr> <td>774M1 000000~037774</td> <td>774M1 @000000~@037774</td> </tr> </table>				コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	774M1 000000~037774	774M1 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																	
コ2000~コ7574	@コ2000~@コ7574																	
b0000~b1774	@b0000~@b1774																	
b2000~b3774	@b2000~@b3774																	
09000~99774	@09000~@99774																	
E0000~E7774	@E0000~@E7774																	
774M1 000000~037774	774M1 @000000~@037774																	
Dの使用範囲	<table border="0"> <tr> <td>コ0000~コ1570</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7570</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1770</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3770</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99770</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7770</td> <td>@E0000~@E7774</td> </tr> <tr> <td>774M1 000000~037770</td> <td>774M1 @000000~@037774</td> </tr> </table>				コ0000~コ1570	@コ0000~@コ1574	コ2000~コ7570	@コ2000~@コ7574	b0000~b1770	@b0000~@b1774	b2000~b3770	@b2000~@b3774	09000~99770	@09000~@99774	E0000~E7770	@E0000~@E7774	774M1 000000~037770	774M1 @000000~@037774
コ0000~コ1570	@コ0000~@コ1574																	
コ2000~コ7570	@コ2000~@コ7574																	
b0000~b1770	@b0000~@b1774																	
b2000~b3770	@b2000~@b3774																	
09000~99770	@09000~@99774																	
E0000~E7770	@E0000~@E7774																	
774M1 000000~037770	774M1 @000000~@037774																	
演算条件	入力信号の立上り(OFF→ON)																	
演算後	S ₁ ~S ₁ +3の内容	不変																
	S ₂ ~S ₂ +3の内容	不変																
	D~D+7の内容	演算結果(バイナリ64ビット)																
フラグ	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354														
	0	0	0	0														

(解説)



命令	
STR	04000
F-215d	09000
	09100
	09200

入力条件04000がOFF→ONの変化時に、レジスタ09000~09003の内容(32ビットデータ)とレジスタ09100~09103の内容(32ビットデータ)をバイナリ乗算して結果をレジスタ09200~09207に格納します。



上記演算は17341734_(H)×22852285_(H)=0320FC45EF28F604_(H)を示します。

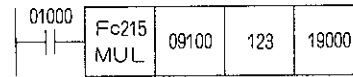
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S₁、S₂、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)

**Fc215
MUL**

**レジスタと定数のバイナリ乗算(8ビット×8ビット)
(MULtiply)**

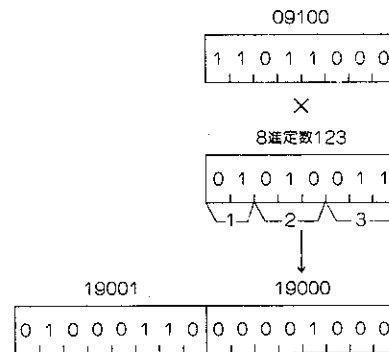
シンボル	Fc215 MUL S _i n D			
機能	レジスタSの内容と8進定数nをバイナリ乗算してレジスタD、D+1に格納する。			
演算内容	S _i × n → D、D+1			
S _i の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~09777 @09000~@09774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774			
nの使用範囲	000~377(8)			
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~09776 @09000~@09774 E0000~E7776 @E0000~@E7774 771#1 000000~037776 771#1 @000000~@037774			
演算条件	入力信号の立上り(OFF→ON)			
演算後	S _i の内容	不変		
	Dの内容	演算結果(下位)		
	D+1の内容	演算結果(上位)		
	フラグ	ゼロ 07357 0	キャリー 07356 0	エラー 07355 0

(解説)



命 令	
STR	01000
Fc215	09100
	123
	19000

入力条件01000がOFF→ONの変化時に、レジスタ09100の内容と8進定数123をバイナリ乗算して、結果をレジスタ19000、19001に格納します。



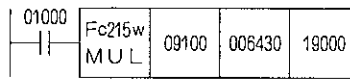
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

**Fc215w
MUL**

**レジスタと定数のバイナリ乗算(16ビット×16ビット)
(MULtiply)**

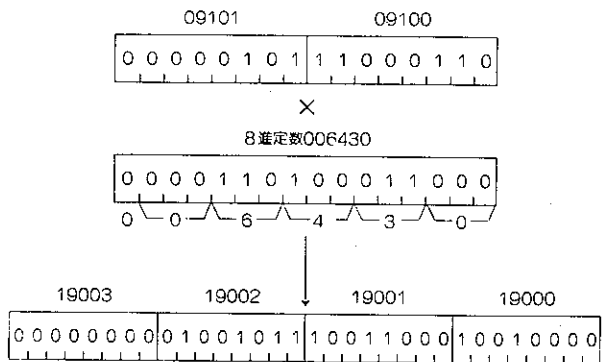
シンボル				
機能	レジスタS _i 、S _i +1の内容(16ビットデータ)と8進定数nをバイナリ乗算してレジスタD、D+1、D+2、D+3に格納する。			
演算内容	(S _i 、S _i +1)×n→D、D+1、D+2、D+3			
S _i の使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 77441 000000~037776 : 77441 @000000~@037774			
nの使用範囲	000000~177777 (8)			
Dの使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 77441 000000~037776 : 77441 @000000~@037774			
演算条件	入力信号の立上り(OFF→ON)			
演算後	S _i 、S _i +1の内容	不変		
	Dの内容	演算結果(下位)		
	D+1の内容	演算結果		
	D+2の内容	演算結果		
	D+3の内容	演算結果(上位)		
フラグ	ゼロ 07357 0	キャリー 07356 0	エラー 07355 0	ノンキャリー 07354 0

(解説)



命令	
STR	01000
Fc215w	09100 006430 19000

入力条件01000がOFF→ONの変化時に、レジスタ09100、09101の内容(16ビットデータ)と8進定数006430をバイナリ乗算して、結果をレジスタ19000、19001、19002、19003に格納します。



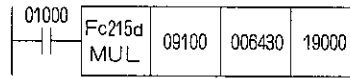
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S_i、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)

**Fc215d
MUL**

**レジスタと定数のバイナリ乗算(32ビット×16ビット)
(MULtiply)**

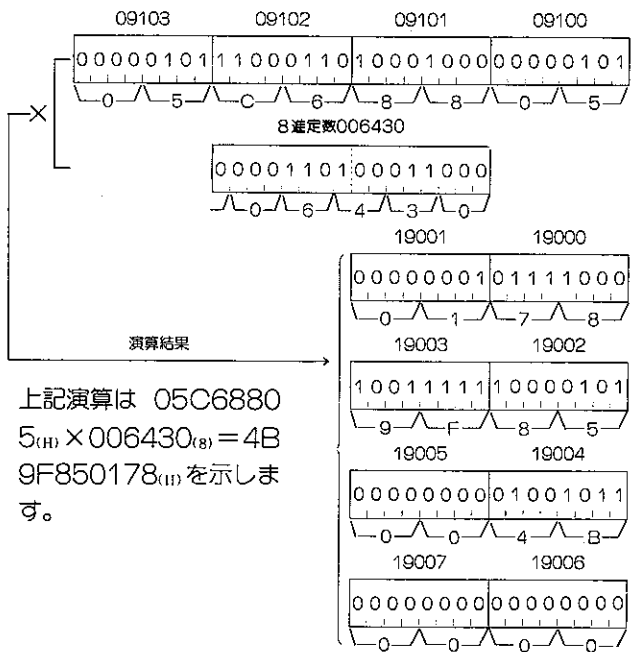
シンボル																		
機能	レジスタ $S_i \sim S_i+3$ の内容(32ビットデータ)と8進定数 n をバイナリ乗算してレジスタ $D \sim D+7$ に格納する。																	
演算内容	$(S_i \sim S_i+3) \times n \rightarrow D \sim D+7$																	
S_i の使用範囲	<table border="0"> <tr><td>コ0000~コ1574</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7574</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1774</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3774</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99774</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7774</td><td>@E0000~@E7774</td></tr> <tr><td>771#1 000000~037774</td><td>771#1 @000000~@037774</td></tr> </table>				コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	771#1 000000~037774	771#1 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																	
コ2000~コ7574	@コ2000~@コ7574																	
b0000~b1774	@b0000~@b1774																	
b2000~b3774	@b2000~@b3774																	
09000~99774	@09000~@99774																	
E0000~E7774	@E0000~@E7774																	
771#1 000000~037774	771#1 @000000~@037774																	
n の使用範囲	000000~177777 ₍₈₎																	
D の使用範囲	<table border="0"> <tr><td>コ0000~コ1570</td><td>@コ0000~@コ1574</td></tr> <tr><td>コ2000~コ7570</td><td>@コ2000~@コ7574</td></tr> <tr><td>b0000~b1770</td><td>@b0000~@b1774</td></tr> <tr><td>b2000~b3770</td><td>@b2000~@b3774</td></tr> <tr><td>09000~99770</td><td>@09000~@99774</td></tr> <tr><td>E0000~E7770</td><td>@E0000~@E7774</td></tr> <tr><td>771#1 000000~037770</td><td>771#1 @000000~@037774</td></tr> </table>				コ0000~コ1570	@コ0000~@コ1574	コ2000~コ7570	@コ2000~@コ7574	b0000~b1770	@b0000~@b1774	b2000~b3770	@b2000~@b3774	09000~99770	@09000~@99774	E0000~E7770	@E0000~@E7774	771#1 000000~037770	771#1 @000000~@037774
コ0000~コ1570	@コ0000~@コ1574																	
コ2000~コ7570	@コ2000~@コ7574																	
b0000~b1770	@b0000~@b1774																	
b2000~b3770	@b2000~@b3774																	
09000~99770	@09000~@99774																	
E0000~E7770	@E0000~@E7774																	
771#1 000000~037770	771#1 @000000~@037774																	
演算条件	入力信号の立上り(OFF→ON)																	
演算	$S_i \sim S_i+3$ の内容	不変																
	$D \sim D+7$ の内容	演算結果(バイナリ64ビット)																
後フラグ	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354														
	0	0	0	0														

(解説)



命令	
STR	01000
Fc215d	09100
	006430
	19000

入力条件01000がOFF→ONの変化時に、レジスタ09100~09103の内容(32ビットデータ)と8進定数006430をバイナリ乗算して、結果をレジスタ19000~19007に格納します。



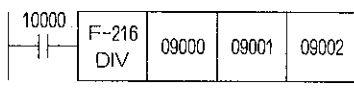
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S_i 、 D には必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)

F-216
DIV

レジスタ間のバイナリ除算(8ビット÷8ビット)
(DIVide)

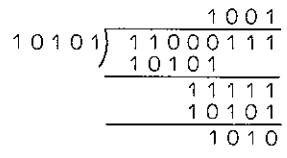
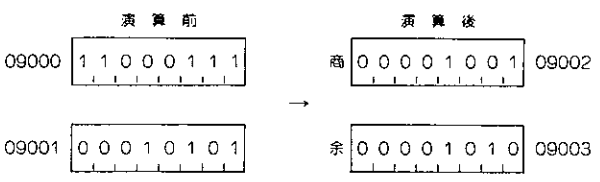
シンボル	<table border="1"> <tr> <td>F-216</td> <td>S₁</td> <td>S₂</td> <td>D</td> </tr> <tr> <td>DIV</td> <td></td> <td></td> <td></td> </tr> </table>				F-216	S ₁	S ₂	D	DIV			
F-216	S ₁	S ₂	D									
DIV												
機能	レジスタS ₁ の内容をレジスタS ₂ の内容でバイナリ除算し、レジスタDに商をレジスタD+1に余を格納する。											
演算内容	S ₁ ÷ S ₂ → D, D+1											
S ₁ の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 77#1 00000~03777 77#1 @00000~@03774											
S ₂ の使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 77#1 00000~03777 77#1 @00000~@03774											
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 77#1 00000~03776 77#1 @00000~@03774											
演算条件	入力信号の立上り(OFF→ON)											
演算後	S ₁ の内容	不変										
	S ₂ の内容	不変										
	Dの内容	演算結果の商	レジスタS ₂ の内容が000 _(N) のとき不変									
	D+1の内容	演算結果の余										
フラグ	レジスタS ₁ の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354							
	000 _(N)	0	0	1	0							
	上記以外	0	0	0	0							

(解説)



命令	
STR	10000
F-216	09000
	09001
	09002

入力条件10000がOFF→ONの変化時に、レジスタ09000の内容をレジスタ09001の内容でバイナリ除算し、商をレジスタ09002に余をレジスタ09003に格納します。



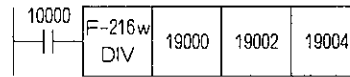
●コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

**F-216w
DIV**

**レジスタ間のバイナリ除算(15ビット÷15ビット)
(DIVide)**

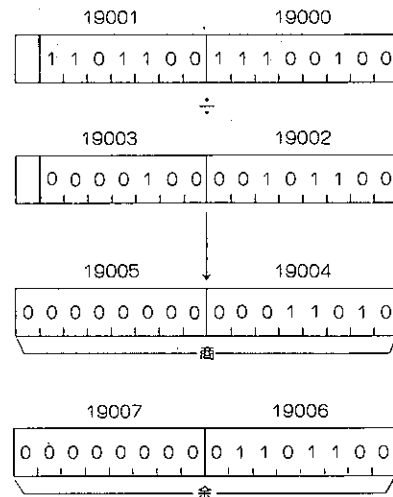
シンボル	— F-216w DIV S ₁ S ₂ D				
機能	レジスタS ₁ 、S ₁ +1の内容(15ビットデータ)をレジスタS ₂ 、S ₂ +1の内容(15ビットデータ)でバイナリ除算し、レジスタD、D+1に商を、レジスタD+2、D+3に余を格納する。				
演算内容	(S ₁ , S ₁ +1) ÷ (S ₂ , S ₂ +1) → D, D+1, D+2, D+3				
S ₁ の使用範囲	コ0000~コ1576 ; @コ0000~@コ1574 コ2000~コ7576 ; @コ2000~@コ7574 b0000~b1776 ; @b0000~@b1774 b2000~b3776 ; @b2000~@b3774 09000~99776 ; @09000~@99774 E0000~E7776 ; @E0000~@E7774 771#1 000000~037776 ; 771#1 @000000~@037774				
S ₂ の使用範囲	コ0000~コ1576 ; @コ0000~@コ1574 コ2000~コ7576 ; @コ2000~@コ7574 b0000~b1776 ; @b0000~@b1774 b2000~b3776 ; @b2000~@b3774 09000~99776 ; @09000~@99774 E0000~E7776 ; @E0000~@E7774 771#1 000000~037776 ; 771#1 @000000~@037774				
Dの使用範囲	コ0000~コ1574 ; @コ0000~@コ1574 コ2000~コ7574 ; @コ2000~@コ7574 b0000~b1774 ; @b0000~@b1774 b2000~b3774 ; @b2000~@b3774 09000~99774 ; @09000~@99774 E0000~E7774 ; @E0000~@E7774 771#1 000000~037774 ; 771#1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	S ₁ 、S ₁ +1の内容	不変			
	S ₂ 、S ₂ +1の内容	不変			
	Dの内容	演算結果の商(下位)	レジスタS ₂ 、S ₂ +1の内容が000000 ₍₈₎ のとき不変		
	D+1の内容	演算結果の商(上位)			
	D+2の内容	演算結果の余(下位)			
D+3の内容	演算結果の余(上位)				
フラグ	レジスタS ₂ 、S ₂ +1の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
	000000 ₍₈₎	0	0	1	0
	上記以外	0	0	0	0

(解説)



命令	
STR	10000
F-216w	19000
	19002
	19004

入力条件10000がOFF→ONの変化時に、レジスタ19000、19001の内容(15ビットデータ)をレジスタ19002、19003の内容(15ビットデータ)でバイナリ除算し、商をレジスタ19004、19005に余をレジスタ19006、19007に格納します。



レジスタ19001、19003のMSB(ビット7)は、無視します。

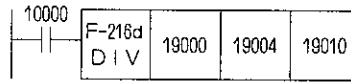
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S₁、S₂、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)

**F-216d
DIV**

**レジスタ間のバイナリ除算(31ビット÷31ビット)
(DIVide)**

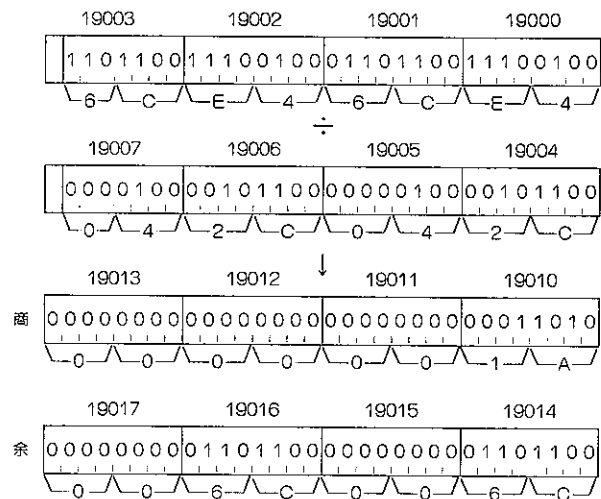
シンボル					
機能	レジスタS ₁ ~S ₁ +3の内容(31ビットデータ)をレジスタS ₂ ~S ₂ +3の内容(31ビットデータ)でバイナリ除算し、レジスタD~D+3に商を、レジスタD+4~D+7に余を格納する。				
演算内容	(S ₁ ~S ₁ +3)÷(S ₂ ~S ₂ +3)→D~D+7				
S ₁ の使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 771M1 000000~037774 : 771M1 @000000~@037774				
S ₂ の使用範囲	コ0000~コ1574 : @コ0000~@コ1574 コ2000~コ7574 : @コ2000~@コ7574 b0000~b1774 : @b0000~@b1774 b2000~b3774 : @b2000~@b3774 09000~99774 : @09000~@99774 E0000~E7774 : @E0000~@E7774 771M1 000000~037774 : 771M1 @000000~@037774				
Dの使用範囲	コ0000~コ1570 : @コ0000~@コ1574 コ2000~コ7570 : @コ2000~@コ7574 b0000~b1770 : @b0000~@b1774 b2000~b3770 : @b2000~@b3774 09000~99770 : @09000~@99774 E0000~E7770 : @E0000~@E7774 771M1 000000~037770 : 771M1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	S ₁ ~S ₁ +3の内容	不変			
	S ₂ ~S ₂ +3の内容	不変			
	D~D+3の内容	演算結果の商 (バイナリ31ビット)	レジスタS ₁ ~S ₁ +3の内容が00のとき 不変		
	D+4~D+7の内容	演算結果の余 (バイナリ31ビット)			
フラグ	レジスタS ₂ , S ₂ +1の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
	000000 _(H)	0	0	1	0
	上記以外			0	

(解説)



命令	
STR	10000
F-216d	19000
	19004
	19010

入力条件10000がOFF→ONの変化時に、レジスタ19000~19003の内容(31ビットデータ)をレジスタ19004~19007の内容(31ビットデータ)でバイナリ除算し、商をレジスタ19010~19013に余をレジスタ19014~19017に格納します。



レジスタ19003、19007のMSB(ビット7)は、無視します。

上記演算は
 $6CE46CE4_{(H)} \div 042C042C_{(H)} = 1A_{(H)}$
 の商と余り $6C006C_{(H)}$ を示します。

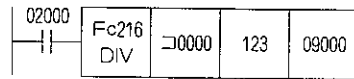
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S₁、S₂、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)

**Fc216
DIV**

**レジスタと定数のバイナリ除算(8ビット÷8ビット)
(DIVide)**

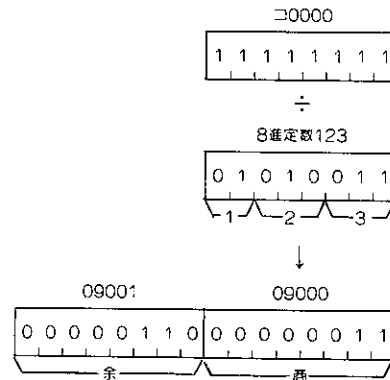
シンボル	<table border="1"> <tr> <td>Fc216 DIV</td> <td>S_i</td> <td>n</td> <td>D</td> </tr> </table>				Fc216 DIV	S _i	n	D										
Fc216 DIV	S _i	n	D															
機能	レジスタS _i の内容を8進定数nでバイナリ除算し、レジスタDに商をレジスタD+1に余を格納する。																	
演算内容	S _i ÷ n → D, D+1																	
S _i の使用範囲	<table border="0"> <tr> <td>コ0000~コ1577</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7577</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>ボ0000~ボ1777</td> <td>@ボ0000~@ボ1774</td> </tr> <tr> <td>ボ2000~ボ3777</td> <td>@ボ2000~@ボ3774</td> </tr> <tr> <td>オ9000~オ9977</td> <td>@オ9000~@オ9974</td> </tr> <tr> <td>エ0000~エ7777</td> <td>@エ0000~@エ7774</td> </tr> <tr> <td>7777#1 000000~037777</td> <td>7777#1 @000000~@037774</td> </tr> </table>				コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	ボ0000~ボ1777	@ボ0000~@ボ1774	ボ2000~ボ3777	@ボ2000~@ボ3774	オ9000~オ9977	@オ9000~@オ9974	エ0000~エ7777	@エ0000~@エ7774	7777#1 000000~037777	7777#1 @000000~@037774
コ0000~コ1577	@コ0000~@コ1574																	
コ2000~コ7577	@コ2000~@コ7574																	
ボ0000~ボ1777	@ボ0000~@ボ1774																	
ボ2000~ボ3777	@ボ2000~@ボ3774																	
オ9000~オ9977	@オ9000~@オ9974																	
エ0000~エ7777	@エ0000~@エ7774																	
7777#1 000000~037777	7777#1 @000000~@037774																	
nの使用範囲	000~377(8)																	
Dの使用範囲	<table border="0"> <tr> <td>コ0000~コ1576</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7576</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>ボ0000~ボ1776</td> <td>@ボ0000~@ボ1774</td> </tr> <tr> <td>ボ2000~ボ3776</td> <td>@ボ2000~@ボ3774</td> </tr> <tr> <td>オ9000~オ9976</td> <td>@オ9000~@オ9974</td> </tr> <tr> <td>エ0000~エ7776</td> <td>@エ0000~@エ7774</td> </tr> <tr> <td>7777#1 000000~037776</td> <td>7777#1 @000000~@037774</td> </tr> </table>				コ0000~コ1576	@コ0000~@コ1574	コ2000~コ7576	@コ2000~@コ7574	ボ0000~ボ1776	@ボ0000~@ボ1774	ボ2000~ボ3776	@ボ2000~@ボ3774	オ9000~オ9976	@オ9000~@オ9974	エ0000~エ7776	@エ0000~@エ7774	7777#1 000000~037776	7777#1 @000000~@037774
コ0000~コ1576	@コ0000~@コ1574																	
コ2000~コ7576	@コ2000~@コ7574																	
ボ0000~ボ1776	@ボ0000~@ボ1774																	
ボ2000~ボ3776	@ボ2000~@ボ3774																	
オ9000~オ9976	@オ9000~@オ9974																	
エ0000~エ7776	@エ0000~@エ7774																	
7777#1 000000~037776	7777#1 @000000~@037774																	
演算条件	入力信号の立上り(OFF→ON)																	
演算後	Sの内容	不変																
	Dの内容	演算結果の商	n=000のとき 不変															
	D+1の内容	演算結果の余																
フラグ	8進定数n	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354													
	000 _(n)	0	0	1	0													
	上記以外	0	0	0	0													

(解説)



命 令	
STR	02000
Fc216	コ0000
	123
	09000

入力条件02000がOFF→ONの変化時に、レジスタコ0000の内容を8進定数123でバイナリ除算し、商をレジスタ09000に、余をレジスタ09001に格納します。



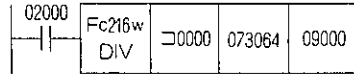
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)

Fc216w
DIV

レジスタと定数のバイナリ除算(15ビット÷15ビット)
(DIVide)

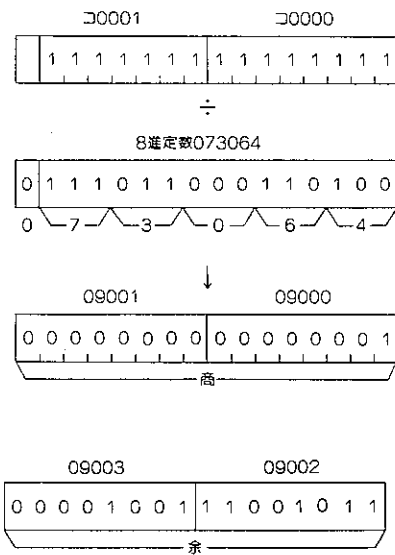
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">Fc216w DIV</td> <td style="padding: 2px;">S_i</td> <td style="padding: 2px;">n</td> <td style="padding: 2px;">D</td> </tr> </table>					Fc216w DIV	S _i	n	D										
Fc216w DIV	S _i	n	D																
機能	レジスタS _i 、S _i +1の内容(15ビットデータ)を8進定数nでバイナリ除算し、レジスタD、D+1に商をレジスタD+2、D+3に余を格納する。																		
演算内容	$(S_i, S_i+1) \div n \rightarrow D, D+1, D+2, D+3$																		
S _i の使用範囲	<table border="0" style="font-size: small;"> <tr> <td>コ0000~コ1574</td><td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7574</td><td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1774</td><td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3774</td><td>@b2000~@b3774</td> </tr> <tr> <td>09000~99774</td><td>@09000~@99774</td> </tr> <tr> <td>E0000~E7774</td><td>@E0000~@E7774</td> </tr> <tr> <td>77441 000000~037774</td><td>77441 @000000~@037774</td> </tr> </table>					コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	77441 000000~037774	77441 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																		
コ2000~コ7574	@コ2000~@コ7574																		
b0000~b1774	@b0000~@b1774																		
b2000~b3774	@b2000~@b3774																		
09000~99774	@09000~@99774																		
E0000~E7774	@E0000~@E7774																		
77441 000000~037774	77441 @000000~@037774																		
nの使用範囲	000000~077777(8)																		
Dの使用範囲	<table border="0" style="font-size: small;"> <tr> <td>コ0000~コ1574</td><td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7574</td><td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1774</td><td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3774</td><td>@b2000~@b3774</td> </tr> <tr> <td>09000~99774</td><td>@09000~@99774</td> </tr> <tr> <td>E0000~E7774</td><td>@E0000~@E7774</td> </tr> <tr> <td>77441 000000~037774</td><td>77441 @000000~@037774</td> </tr> </table>					コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	77441 000000~037774	77441 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																		
コ2000~コ7574	@コ2000~@コ7574																		
b0000~b1774	@b0000~@b1774																		
b2000~b3774	@b2000~@b3774																		
09000~99774	@09000~@99774																		
E0000~E7774	@E0000~@E7774																		
77441 000000~037774	77441 @000000~@037774																		
演算条件	入力信号の立上り(OFF→ON)																		
演算後	S _i 、S _i +1の内容	不変																	
	Dの内容	演算結果の商(下位)		n=000000 のとき不変															
	D+1の内容	演算結果の商(上位)																	
	D+2の内容	演算結果の余(上位)																	
	D+3の内容	演算結果の余(下位)																	
フラグ	8進定数n	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354														
	000000	0	0	1	0														
	上記以外			0															

(解説)



命 令	
STR	02000
Fc216w	00000
	073064
	09000

入力条件02000がOFF→ONの変化時に、レジスタコ0000、コ0001の内容(15ビットデータ)を8進定数073064でバイナリ除算し、商をレジスタ09000、09001に余をレジスタ09002、09003に格納します。



レジスタコ0001のMSB(ビット7)は無視します

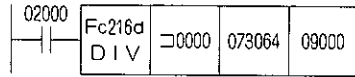
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)

**Fc216d
DIV**

**レジスタと定数のバイナリ除算(31ビット÷15ビット)
(DIVide)**

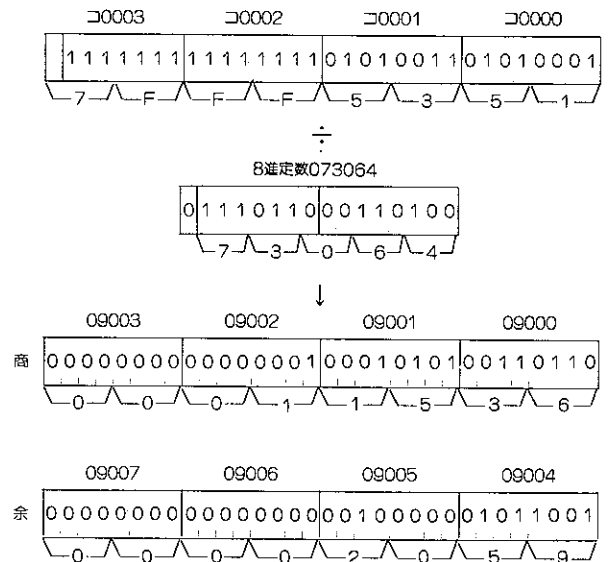
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">Fc216d DIV</td> <td style="text-align: center;">S_i</td> <td style="text-align: center;">n</td> <td style="text-align: center;">D</td> </tr> </table>					Fc216d DIV	S _i	n	D										
Fc216d DIV	S _i	n	D																
機能	レジスタS _i ~S _i +3の内容(31ビットデータ)を8進定数nでバイナリ除算し、レジスタD~D+3に商をレジスタD+4~D+7に余を格納する。																		
演算内容	(S _i ~S _i +3) ÷ n → D~D+7																		
S _i の使用範囲	<table border="0" style="width: 100%;"> <tr> <td>コ0000~コ1574</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7574</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1774</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3774</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99774</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7774</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771#1 000000~037774</td> <td>771#1 @000000~@037774</td> </tr> </table>					コ0000~コ1574	@コ0000~@コ1574	コ2000~コ7574	@コ2000~@コ7574	b0000~b1774	@b0000~@b1774	b2000~b3774	@b2000~@b3774	09000~99774	@09000~@99774	E0000~E7774	@E0000~@E7774	771#1 000000~037774	771#1 @000000~@037774
コ0000~コ1574	@コ0000~@コ1574																		
コ2000~コ7574	@コ2000~@コ7574																		
b0000~b1774	@b0000~@b1774																		
b2000~b3774	@b2000~@b3774																		
09000~99774	@09000~@99774																		
E0000~E7774	@E0000~@E7774																		
771#1 000000~037774	771#1 @000000~@037774																		
nの使用範囲	000000~077777 ₍₈₎																		
Dの使用範囲	<table border="0" style="width: 100%;"> <tr> <td>コ0000~コ1570</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7570</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1770</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3770</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99770</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7770</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771#1 000000~037770</td> <td>771#1 @000000~@037774</td> </tr> </table>					コ0000~コ1570	@コ0000~@コ1574	コ2000~コ7570	@コ2000~@コ7574	b0000~b1770	@b0000~@b1774	b2000~b3770	@b2000~@b3774	09000~99770	@09000~@99774	E0000~E7770	@E0000~@E7774	771#1 000000~037770	771#1 @000000~@037774
コ0000~コ1570	@コ0000~@コ1574																		
コ2000~コ7570	@コ2000~@コ7574																		
b0000~b1770	@b0000~@b1774																		
b2000~b3770	@b2000~@b3774																		
09000~99770	@09000~@99774																		
E0000~E7770	@E0000~@E7774																		
771#1 000000~037770	771#1 @000000~@037774																		
演算条件	入力信号の立上り(OFF→ON)																		
演算後	S _i ~S _i +3の内容	不変																	
	D~D+3の内容	演算結果の商 (バイナリ31ビット)	n=0のとき 不変																
	D+4~D+7の内容	演算結果の余 (バイナリ31ビット)																	
フラグ	8進定数n	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354														
	000000	0	0	1	0														
	上記以外	0	0	0	0														

(解説)



命 令	
STR	02000
Fc216d	00000
	073064
	09000

入力条件02000がOFF→ONの変化時に、レジスタコ0000~コ0003の内容(31ビットデータ)を8進定数073064でバイナリ除算し、商をレジスタ09000~09003に余をレジスタ09004~09007に格納します。



レジスタコ0003のMSB(ビット7)は無視します。上記演算は 7FFF5351_(H) ÷ 73064₍₈₎ = 11536_(H) の商と余り2059_(H)を示します。

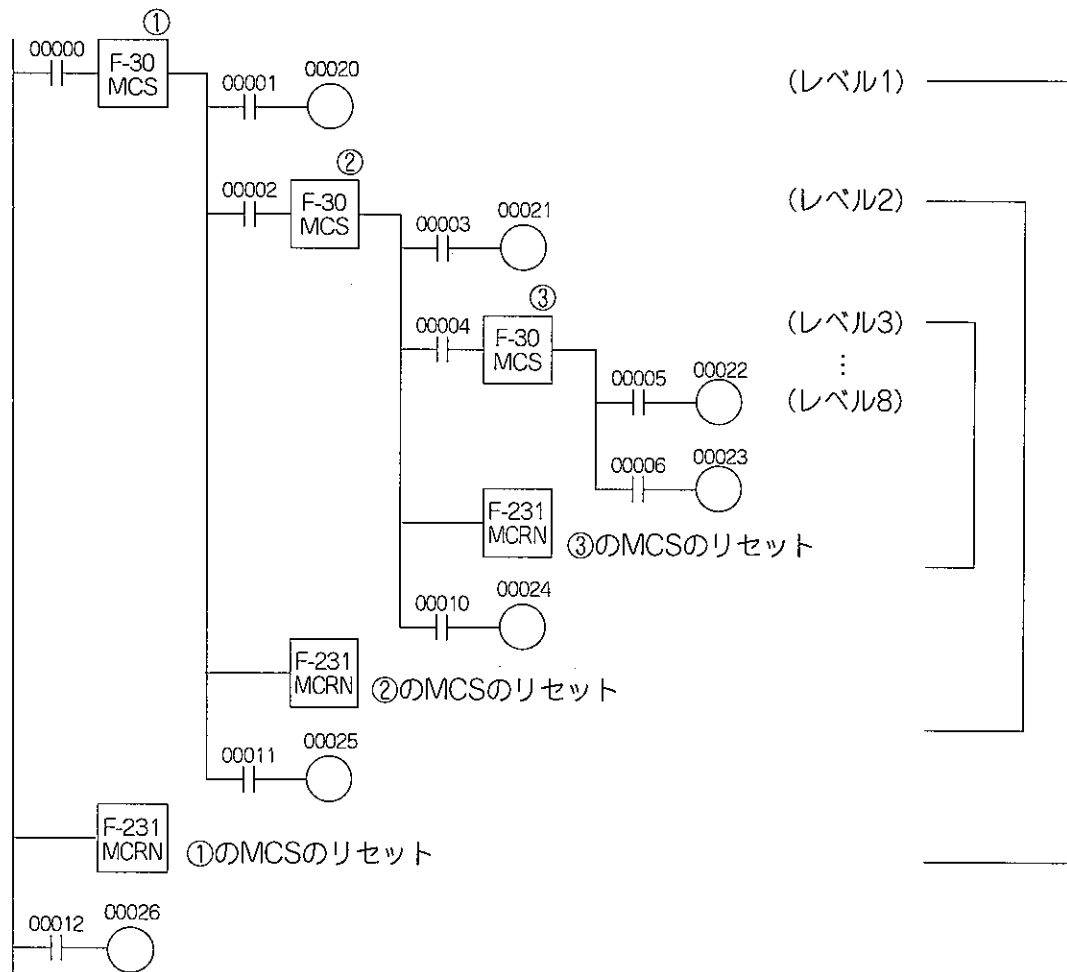
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- S、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)

**F-231
MCRN**

マスターコントロール ネスティングリセット
(Master Control Reset Nesting)

F-30 (MCS) と併用し、共通演算条件以後の回路が複数の出力に分岐している場合に使用します。
F-31 (MCR) はネスティングはできませんが、F-231 (MCRN) を使用するとレベル8までのネスティングが可能です。

(例) レベル3のネスティング



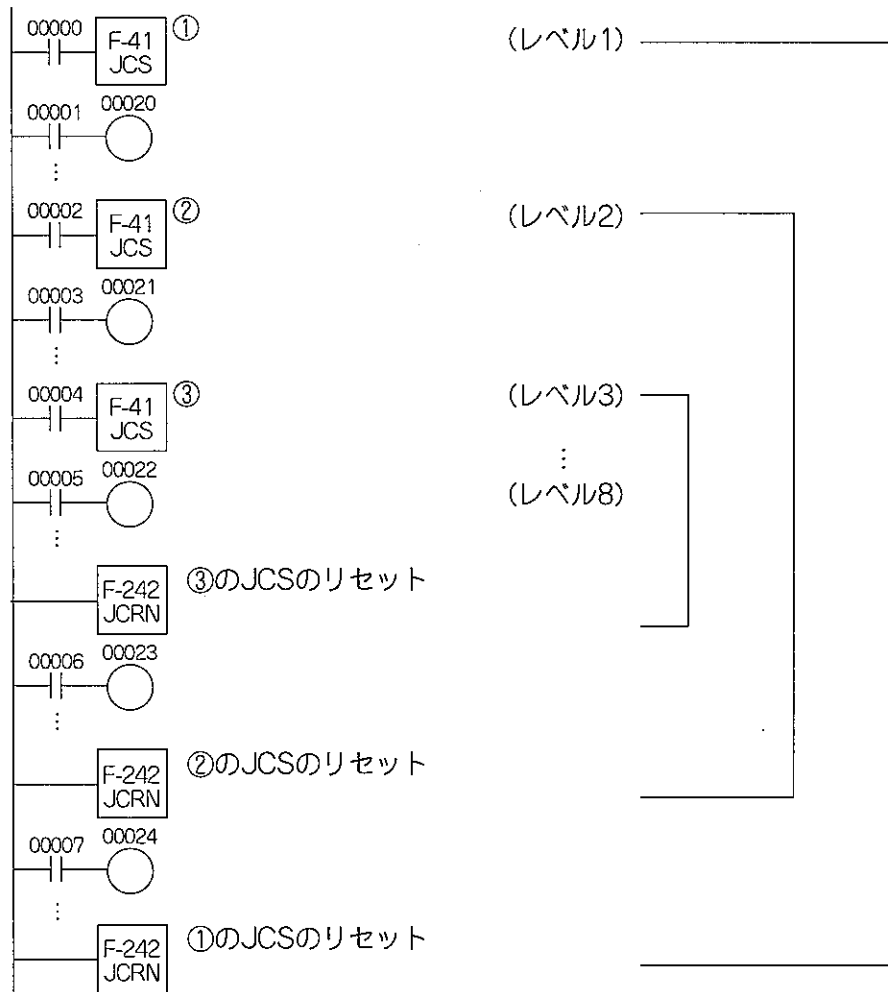
●F-30 (MCS)、F-31 (MCR) を合わせてご参照ください。(9・114ページ)

**F-242
JCRN**

ジャンプコントロール ネスティングリセット
(Jump Control Reset Nesting)

F-41 (JCS) の条件がOFFの時、F-242 (JCRN) までにあるEND命令を除くすべての命令は実行しません。
F-42 (JCR) は、ネスティングはできませんが、F-242 (JCRN) を使用するとレベル8までのネスティングが可能です。

(例) レベル3のネスティング



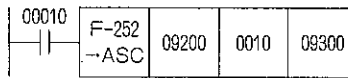
●F-41 (JCS)、F-42 (JCR) を合わせてご参照ください。(9・126ページ)

F-252
→ASC

HEX(16進)コード→ASCIIコード変換
(→ASCII)

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-252</td><td>S</td><td>n</td><td>D</td></tr><tr><td>→ASC</td><td></td><td></td><td></td></tr></table>			F-252	S	n	D	→ASC			
F-252	S	n	D								
→ASC											
機能	レジスタSを先頭としたnバイトの領域にある16進(HEX)コードをASCIIコードに変換しレジスタDを先頭に格納する。変換はSの下位4ビット側から変換する。										
演算内容	〈S、S+1……S+n-1〉→ASCII変換 →D、D+1……D+2n-1										
Sの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 ファイル000000~037777 ファイル@000000~@037774										
nの使用範囲	0000~1777(8) (0000にすると1024バイトになる)										
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 ファイル000000~037777 ファイル@000000~@037774										
演算条件	入力信号の立上り(OFF→ON)										
演算後	S~S+n-1の内容	不変									
	D~D+2n-1の内容	演算結果									
	フラグ	不変									

(解説)



命令	
STR	00010
F-252	09200
	0010
	09300

入力条件00010がOFF→ONの変化時に、レジスタ09200~09207までの010(8)バイト(10進数で8バイト)データをASCII変換し、レジスタ09300を先頭にデータ格納します。

	演算前		演算後	
09200	1 0	→	3 0	09300
			3 1	09301
09201	3 2	→	3 2	09302
			3 3	09303
09202	5 4	→	3 4	09304
			3 5	09305
09203	7 6	→	3 6	09306
			3 7	09307
09204	9 8	→	3 8	09310
			3 9	09311
09205	B A	→	4 1	09312
			4 2	09313
09206	D C	→	4 3	09314
			4 4	09315
09207	F E	→	4 5	09316
			4 6	09317

- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- n、Dで設定したレジスタ領域に注意してください。演算結果のレジスタ使用バイト数は16進コード領域の2倍になります。
- 演算結果が、ファイル番号0のファイルアドレス001600~001777と035600~035777に入らないようにしてください。入ると誤動作の原因になります。
- nの値を0000にすると1024バイトになります。

- 16進コードとASCIIコードはつぎのようになります。

16進コード	0	1	2	3	4	5	6	7
ASCIIコード	30	31	32	33	34	35	36	37

16進コード	8	9	A	B	C	D	E	F
ASCIIコード	38	39	41	42	43	44	45	46

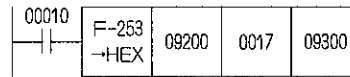
F-253
→HEX

ASCIIコード→HEX(16進)コード変換
(→HEX)

シンボル	— F-253 →HEX S n D				
機能	レジスタSを先頭としたnバイトの領域にあるASCIIコードをHEX(16進)コードに変換しレジスタDを先頭に格納する。変換データはレジスタDの下位4ビット側から格納する。16進コードに変換できないASCIIコードがあるとコードをレジスタD領域最終アドレスに格納し変換を中止する。				
演算内容	$\langle S, S+1 \dots S+n-1 \rangle \rightarrow \text{HEX変換}$ $\rightarrow D, D+1 \dots D+\frac{n}{2}-1$ 変換不能コード $\rightarrow D+\frac{n}{2}-1$ (n奇数時最終アドレス = $D+\frac{n}{2}$)				
Sの使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 771M1 000000~037777 : 771M1 @000000~@037774				
nの使用範囲	0000~1777(8) (0000にすると1024バイトになる)				
Dの使用範囲	コ0000~コ1577 : @コ0000~@コ1574 コ2000~コ7577 : @コ2000~@コ7574 b0000~b1777 : @b0000~@b1774 b2000~b3777 : @b2000~@b3774 09000~99777 : @09000~@99774 E0000~E7777 : @E0000~@E7774 771M1 000000~037777 : 771M1 @000000~@037774				
演算条件	入力信号の立上り(OFF→ON)				
演算後	S~S+n-1の内容	不変			
	D~D+ $\frac{n}{2}-1$ の内容	演算結果	変換不能コードが有りてコードをレジスタD領域最終アドレスに格納し変換を中止		
	D + $\frac{n}{2}$	正常時不変			
フラグ	変換動作	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
	変換できないコード有 上記以外	0	0	1 0	0

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)
- 演算結果が、ファイル番号0のファイルアドレス001600~001777と035600~035777以降に入らないようにしてください。入ると誤動作の原因になります。
- nの値を0000にすると1024バイトになります。
- nの値を奇数バイトにすると最後の上位4ビットデータは0になります。

【解説】



命 令	
STR	00010
F-253	09200
	0017
	09300

入力条件00010がOFF→ONの変化時にレジスタ09200~09216までの017(8)バイト(10進数で15/バイト)ASCIIデータを16進変換しレジスタ09300を先頭にデータ格納します。

	演算前	演算後	
09200	3 0	→	1 0 09300
09201	3 1		
09202	3 2	→	3 2 09301
09203	3 3		
09204	3 4	→	5 4 09302
09205	3 5		
09206	3 6	→	7 6 09303
09207	3 7		
09210	3 8	→	9 8 09304
09211	3 9		
09212	4 1	→	B A 09305
09213	4 2		
09214	4 3	→	D C 09306
09215	4 4		
09216	4 5	→	0 E 09307

変換不能なASCIIコードが存在すると最終レジスタ09307へそのASCIIコードを格納します。

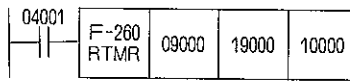
- 16進コードとASCIIコードはつぎのようになります。16進コードに変換できないASCIIコードがあるとその時点で変換を停止しエラーフラグ(07355)をONします。同時にD領域最終レジスタに変換不能コードを格納します。

16進	0	1	2	3	4	5	6	7
ASCII	30	31	32	33	34	35	36	37
16進	8	9	A	B	C	D	E	F
ASCII	38	39	41	42	43	44	45	46

シンボル	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">F-260 RTMR</td> <td style="text-align: center;">S</td> <td style="text-align: center;">D</td> <td style="text-align: center;">BIT</td> </tr> </table>					F-260 RTMR	S	D	BIT
F-260 RTMR	S	D	BIT						
機能	レジスタD、D+1の内容(タイマ現在値)は、レジスタS、S+1の内容(タイマ設定値)から0.1秒ごとに-1され、0になるとリレーBITをONし、入力信号がONの間保持する。								
演算内容	$(S, S+1) - \text{経過時間} \rightarrow (D, D+1)$ ↓ $(D, D+1) = 0$ になれば、BIT(ON)								
Sの使用範囲	コ0000~コ1576 コ2000~コ7576 b0000~b3776 09000~99776 E0000~E7776 774M1000000~037776		@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b3774 @09000~@99774 @E0000~@E7774 774M1@000000~@037774						
Dの使用範囲	コ0000~コ1576 コ2000~コ7576 b0000~b3776 09000~99776 E0000~E7776 774M1000000~037776		@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b3774 @09000~@99774 @E0000~@E7774 774M1@000000~@037774						
BITの使用範囲	00000~15777, 20000~75777								
演算条件	入力信号の立上り(OFF→ON)で計数を開始し、入力信号がONの間計数								
演算前	S、S+1の内容	タイマ設定値0000~9999 (BCD4桁、0~999.9秒)							
	D、D+1の内容	タイマ現在値 S、S+1の内容と同じ							
	BITの内容	OFF							
演算中	S、S+1の内容	不変							
	D、D+1の内容	演算結果のタイマ現在値 0000~9999(BCD4桁)							
	BITの内容	ON(タイマ現在値=0時)							
演算後	レジスタDの内容	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354				
	BCDコードの時	0	0	0	0				
	BCDコードでない時			1					

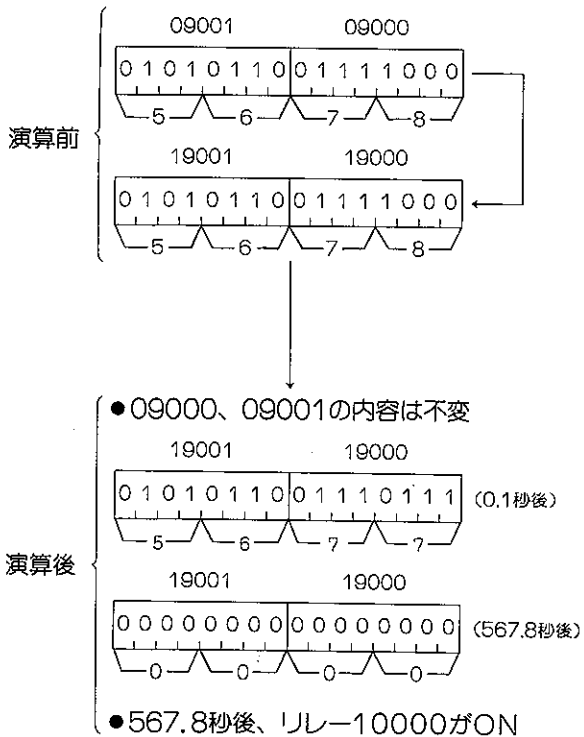
- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。
- プログラム入力後、演算条件がONで、かつレジスタDの内容が0のまま運転モードに変えると、出力リレー(BIT)がONとなりますのでご注意ください。

(解説)



命令	
STR	04001
F-260	09000
	19000
	10000

入力条件04001がOFF→ONの変化後、ONの間レジスタ19000、19001の内容(タイマ現在値)は0.1秒ごとにレジスタ09000、09001の内容(タイマ設定値)から-1され、0になるとリレー10000がONし、04001がONの間保持します。



上記の演算はタイマ設定値を5678(567.8秒)に設定した場合です。

- 停電保持モード(システムメモリ#201参照)でご使用の場合、レジスタDはキーブリレーエリア及びレジスタ(b0000以降あるいは09000以降)をご使用ください。
- 機能は減算式TMR命令(8・6ページ)と同様です。

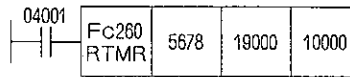
入力信号	タイマ現在値	リレーBIT
OFF	タイマ設定値	OFF
ON(現在値>0)	0.1秒ごとに-1される	OFF
ON(現在値=0)	0	ON

Fc260 RTMR

減算タイマ(定数、レジスタ指定)

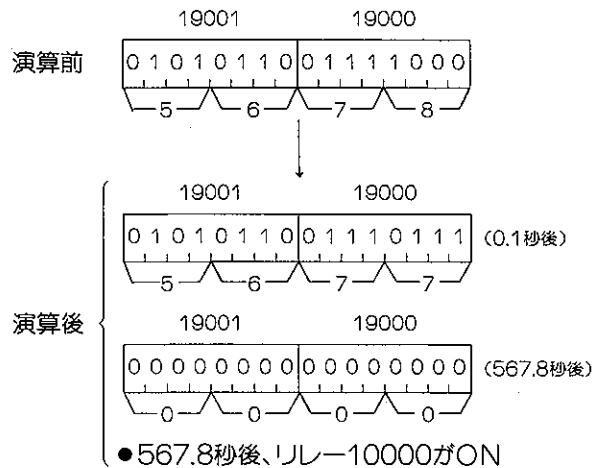
シンボル	— Fc260 RTMR n D BIT			
機能	レジスタD、D+1の内容(タイマ現在値)は、n(タイマ設定値)から0.1秒ごとに-1され、0になるとリレーBITをONし、入力信号がONの間保持する。			
演算内容	$n - \text{経過時間} \rightarrow (D, D+1)$ $(D, D+1) = 0$ になれば、BIT(ON)			
nの使用範囲	タイマ設定値0000~9999 (0~999.9秒)			
Dの使用範囲	コ0000~コ1576 : @コ0000~@コ1574 コ2000~コ7576 : @コ2000~@コ7574 b0000~b1776 : @b0000~@b1774 b2000~b3776 : @b2000~@b3774 09000~99776 : @09000~@99774 E0000~E7776 : @E0000~@E7774 77776 : @77776 : @00000~@03776 : @00000~@03774			
BITの使用範囲	00000~15777, 20000~75777			
演算条件	入力信号の立上り(OFF→ON)で計数を開始し、入力信号がONの間計数			
演算前	D、D+1の内容	タイマ現在値 nの値と同じ		
	BITの内容	OFF		
演算後	D、D+1の内容	演算結果のタイマ現在値 0000~9999(BCD4桁)		
	BITの内容	ON(タイマ現在値=0時)		
	フラグ	F-260と同じ(前ページ参照)		

(解説)



命令	
STR	04001
Fc260	5678 19000 10000

入力条件04001がOFF→ONの変化後、ONの間レジスタ19000、19001の内容(タイマ現在値)は0.1秒ごとにnの値5678(タイマ設定値)から-1され、0になるとリレー10000がONし、04001がONの間保持します。



上記の演算はタイマ設定値を5678(567.8秒)に設定した場合です。

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 機能は減算式TMR命令(8・6ページ)と同様です。

入力信号	タイマ現在値	リレーBIT
OFF	タイマ設定値	OFF
ON(現在値>0)	0.1秒ごとに-1される	OFF
ON(現在値=0)	0	ON

- プログラム入力後、演算条件がONで、かつレジスタDの内容が0のまま運転モードに変えると、出力リレー(BIT)がONとなりますのでご注意ください
- 停電保持モード(システムメモリ#201参照)でご使用の場合、レジスタDはキープリレーエリア及びレジスタ(b0000以降あるいは09000以降)をご使用ください。

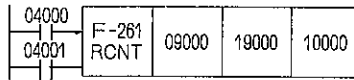
**F-261
RCNT**

減算カウンタ(設定値、レジスタ指定)

シンボル	① F-261 RCNT S D BIT				
機能	リセット入力②(OFF)の間レジスタD、D+1の内容(カウンタ現在値)は、レジスタS、S+1の内容(カウンタ設定値)から計数入力①がOFF→ONに変化することにより-1され、0になるとリレーBITをONして保持する。リセット入力②(ON)の時、カウンタ現在値 = カウンタ設定値およびリレーBIT(OFF)となる。				
演算内容	(S、S+1) - 計数入力回数 → (D、D+1) ↓ (D、D+1) = 0になれば、BIT(ON)				
Sの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 ファイル1 000000~037776 ファイル1 @000000~@037774				
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 ファイル1 000000~037776 ファイル1 @000000~@037774				
BITの使用範囲	00000~15777, 20000~75777				
演算条件	リセット入力②(OFF)の間、計数入力①(OFF→ON)				
演算前	S、S+1の内容	タイマ設定値 0000~9999(BCD 4桁)			
	D、D+1の内容	カウンタ現在値 S、S+1の内容と同じ			
	BITの内容	OFF			
演算中	S、S+1の内容	不変			
	D、D+1の内容	演算結果のカウンタ現在値 0000~9999(BCD 4桁)			
	BITの内容	ON(カウンタ現在値=0時)			
演算後	レジスタDの内容	ゼロ 07357	キャリー 07356	エラー 07355	
	フラグ	BCDコードの時	0	0	0
		BCDコードでない時	0	0	1

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 間接アドレスの使用方法に関しては、9・3ページ「間接アドレス指定」の項をご参照ください。

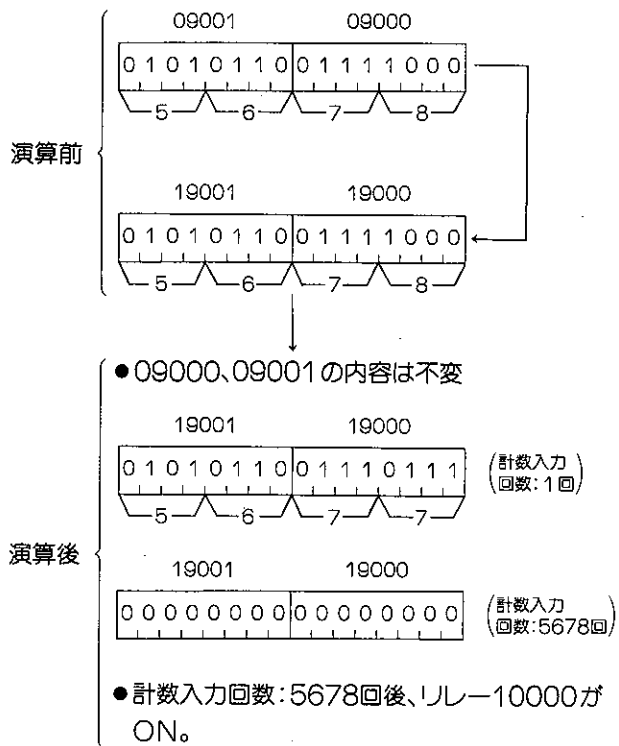
(解説)



命 令	
STR	04000
STR	04001
F-261	09000
	19000
	10000

リセット入力04001(OFF)の間レジスタ19000、19001の内容(カウンタ現在値)はレジスタ09000、09001の内容(カウンタ設定値)から計数入力04000がOFF→ONに変化することにより-1され、0になるとリレー10000をONして保持します。

リセット入力04001(ON)の時、(レジスタ19000、19001の内容)=(レジスタ09000、09001の内容)およびリレー10000(OFF)となります。



上記の演算はカウンタ設定値を5678回に設定した場合です。

注1 注2 Fc261と同じ(次ページ参照)

- 機能は減算式CNT命令(8・7ページ)と同様です。

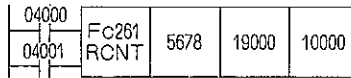
リセット入力②	カウンタ現在値	リレーBIT
ON	カウンタ設定値	OFF
OFF(現在値>0)	計数入力①がOFF→ONとなることにより-1	OFF
OFF(現在値=0)	0	ON

**Fc261
RCNT**

減算カウンタ(定数、レジスタ指定)

シンボル	① Fc261 RCNT n D BIT	
機能	リセット入力②(OFF)の間レジスタD、D+1の内容(カウンタ現在値)は、n(カウンタ設定値)から計数入力①がOFF→ONに変化することにより-1され、0になるとリレーBITをONして保持する。リセット入力②(ON)の時、カウンタ現在値=カウンタ設定値およびリレーBIT(OFF)となる。	
演算内容	n - 計数入力回数 → (D, D+1) ↓ (D, D+1) = 0になれば、BIT(ON)	
nの使用範囲	カウンタ設定値0000~9999 (0~9999回)	
Dの使用範囲	コ0000~コ1576 ; @コ0000~@コ1574 コ2000~コ7576 ; @コ2000~@コ7574 b0000~b1776 ; @b0000~@b1774 b2000~b3776 ; @b2000~@b3774 09000~99776 ; @09000~@99774 E0000~E7776 ; @E0000~@E7774 774M1 000000~037776 ; 774M1 @000000~@037774	
BITの使用範囲	00000~15777, 20000~75777	
演算条件	リセット入力②(OFF)の間、計数入力①(OFF→ON)	
演算前	D、D+1の内容	カウンタ現在値 nの値と同じ
	BITの内容	OFF
演算後	D、D+1の内容	演算結果のカウンタ現在値 0000~9999(BCD4桁)
	BITの内容	ON(カウンタ現在値=0時)
フラグ	F-261と同じ(前ページ参照)	

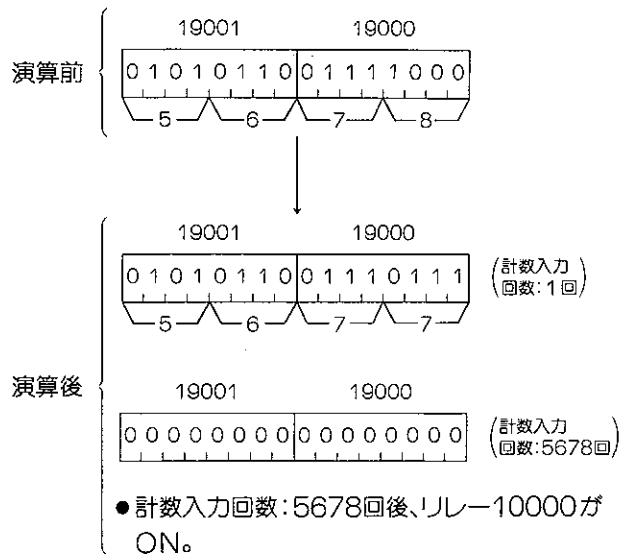
(解説)



命 令	
STR	04000
STR	04001
Fc261	5678
	19000
	10000

リセット入力04001(OFF)の間レジスタ19000、19001の内容(カウンタ現在値)は、nの値5678(カウンタ設定値)から計数入力04000がOFF→ONに変化することにより-1され、0になるとリレー10000をONして保持します。

リセット入力04001(ON)の時、(レジスタ19000、19001の内容)=(nの値5678)およびリレー10000(OFF)になります。



上記の演算はカウンタ設定値を5678回に設定した場合です。

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- 機能は減算式CNT命令(8・7ページ)と同様です。

リセット入力②	カウンタ現在値	リレーBIT
ON	カウンタ設定値	OFF
OFF(現在値>0)	計数入力①がOFF→ONとなるごとに-1	OFF
OFF(現在値=0)	0	ON

注1 プログラム入力後、リセット入力OFFで、かつレジスタDの値が0のまま運転モードにすると、出力リレー(BIT)がONとなりますのでご注意ください。

注2 レジスタDはキーブリレーエリア及びレジスタ(09000以降)をご使用ください。入出力リレーエリア又は補助リレーエリアのコXXXXを使用した場合、電源OFF→ON時に出力リレー(BIT)がONとなります。

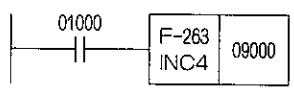
F-263 加算(+4)カウンタ(1バイトバイナリ)
INC4 (INCrement)

シンボル						
機能	レジスタDの内容(バイナリデータ)を加算(+4)カウントする。					
演算内容	$\langle D \rangle + 4 \rightarrow D$					
Dの使用範囲	コ0000~コ1577 @コ0000~@コ1574 コ2000~コ7577 @コ2000~@コ7574 b0000~b1777 @b0000~@b1774 b2000~b3777 @b2000~@b3774 09000~99777 @09000~@99774 E0000~E7777 @E0000~@E7774 771#1 000000~037777 771#1 @000000~@037774					
演算条件	入力信号の立上り(OFF→ON)					
演算後	Dの内容	演算結果(バイナリコード)				
	フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354
		374→000 (8)(注2)	1	1	0	0
		375(8)→001	0	1	0	0
		376(8)→002 377(8)→003	0	0	0	0
上記以外	0	0	0	1		

- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- Dの内容はバイナリコードです。10進表現では000~255、8進表現では000~377₍₈₎と見なすことができます。
- フラグの状態はそのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)

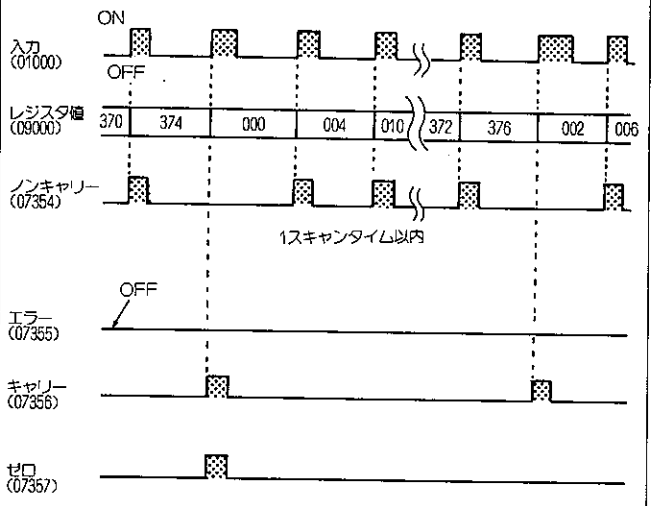
参考 下記のF命令は働きが類似しています。
 F-63、F-63w、F-163、F-163w、F-263、F-263w

(解説)



命 令	
STR	01000
F-263	09000

入力条件01000がOFF→ONの変化時、レジスタ09000の内容(バイナリデータ)を加算(+4)カウントします。

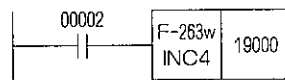


**F-263w
INC4**

**加算(+4)カウンタ(1ワードバイナリ)
(INCrement)**

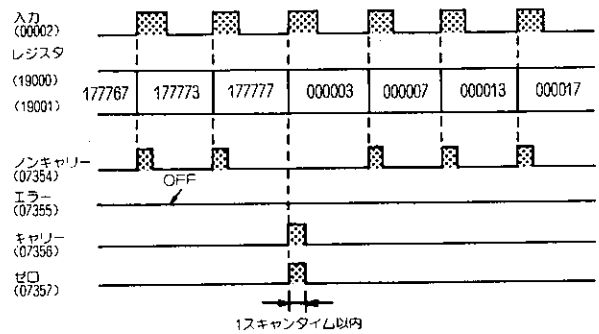
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-263w INC4</td><td style="padding: 2px;">D</td></tr></table>					F-263w INC4	D
F-263w INC4	D						
機能	レジスタD、D+1の内容(バイナリデータ)を加算(+4)カウントする。						
演算内容	$\langle D, D+1 \rangle + 4 \rightarrow D, D+1$						
Dの使用範囲	コ0000~コ1576: @コ0000~@コ1574 コ2000~コ7576: @コ2000~@コ7574 b0000~b1776: @b0000~@b1774 b2000~b3776: @b2000~@b3774 09000~99776: @09000~@99774 E0000~E7776: @E0000~@E7774 774M1 000000~037776: 774M1 @000000~@037774						
演算条件	入力信号の立上り(OFF→ON)						
演算後	Dの内容	演算結果(下位)					
	D+1の内容	演算結果(上位)					
フラグ		演算結果(8進)	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354	
		177774→000000	1	1	0	0	
		177775→000001					
		177776→000002	0	1	0	0	
		177777→000003					
	上記以外	0	0	0	1		

(解説)



命令	
STR	00002
F-263w	19000

入力条件00002がOFF→ONの変化時、レジスタ19000と19001の内容(バイナリデータ)を加算カウント(+4)します。



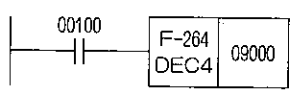
- コ0734~コ0737は特殊領域です。
(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。
(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。
(9・7ページ「データ処理命令とフラグ」参照)

参考 下記のF命令は動きが類似しています。
F-63、F-63w、F-163、F-163w、F-263、
F-263w

F-264 減算(-4)カウンタ(1バイトバイナリ)
DEC4 (DECrement)

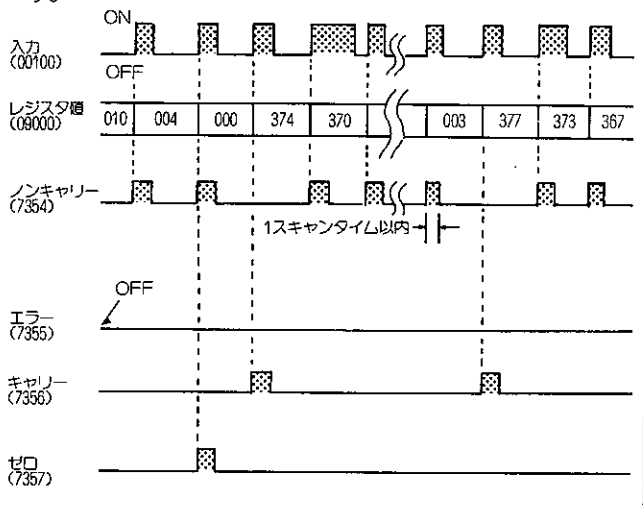
シンボル	<table border="1" style="display: inline-table;"> <tr> <td style="border: none;">F-264</td> <td style="border: none;">D</td> </tr> <tr> <td style="border: none;">DEC4</td> <td style="border: none;"></td> </tr> </table>					F-264	D	DEC4											
F-264	D																		
DEC4																			
機能	レジスタDの内容(バイナリデータ)を減算(-4)カウントする。																		
演算内容	$\langle D \rangle - 4 \rightarrow D$																		
Dの使用範囲	<table border="0"> <tr> <td>コ0000~コ1577</td> <td>@コ0000~@コ1574</td> </tr> <tr> <td>コ2000~コ7577</td> <td>@コ2000~@コ7574</td> </tr> <tr> <td>b0000~b1777</td> <td>@b0000~@b1774</td> </tr> <tr> <td>b2000~b3777</td> <td>@b2000~@b3774</td> </tr> <tr> <td>09000~99777</td> <td>@09000~@99774</td> </tr> <tr> <td>E0000~E7777</td> <td>@E0000~@E7774</td> </tr> <tr> <td>771M1 000000~037777</td> <td>771M1 @000000~@037774</td> </tr> </table>					コ0000~コ1577	@コ0000~@コ1574	コ2000~コ7577	@コ2000~@コ7574	b0000~b1777	@b0000~@b1774	b2000~b3777	@b2000~@b3774	09000~99777	@09000~@99774	E0000~E7777	@E0000~@E7774	771M1 000000~037777	771M1 @000000~@037774
コ0000~コ1577	@コ0000~@コ1574																		
コ2000~コ7577	@コ2000~@コ7574																		
b0000~b1777	@b0000~@b1774																		
b2000~b3777	@b2000~@b3774																		
09000~99777	@09000~@99774																		
E0000~E7777	@E0000~@E7774																		
771M1 000000~037777	771M1 @000000~@037774																		
演算条件	入力信号の立上り(OFF→ON)																		
演算後	Dの内容	演算結果(バイナリコード)																	
	フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354													
		004(8)→000(8)	1	0	0	1													
		003(8)→377																	
		002(8)→376	0	1	0	0													
001(8)→375																			
000(8)→374																			
上記以外	0	0	0	1															

(解説)



命令	
STR	00100
F-264	09000

入力条件00100がOFF→ONの変化時、レジスタ09000の内容(バイナリデータ)を減算(-4)カウントします。



- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)

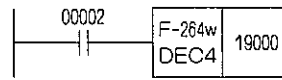
参考 下記のF命令は動きが類似しています。
 F-64、F-64w、F-164、F-164w、F-264、F-264w

F-264w
DEC4

減算(-4)カウンタ(1ワードバイナリ) (DECrement)

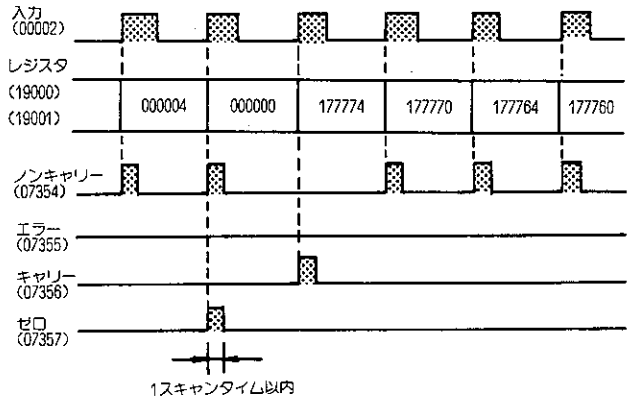
シンボル						
機能	レジスタD、D+1の内容(バイナリデータ)を減算(-4)カウントする。					
演算内容	$\langle D, D+1 \rangle - 4 \rightarrow D, D+1$					
Dの使用範囲	コ0000~コ1576 @コ0000~@コ1574 コ2000~コ7576 @コ2000~@コ7574 b0000~b1776 @b0000~@b1774 b2000~b3776 @b2000~@b3774 09000~99776 @09000~@99774 E0000~E7776 @E0000~@E7774 774#1 000000~037776 774#1 @000000~@037774					
演算条件	入力信号の立上り(OFF→ON)					
演算後	Dの内容	演算結果(下位)				
	D+1の内容	演算結果(上位)				
	フラグ	演算結果(8進)	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354
		000004→000000	1	0	0	1
		000003(e)→17777 000002(e)→17776 000001(e)→17775	0	1	0	0
上記以外		0	0	0	1	

(解説)



命令	
STR	00002
F-264w	19000

入力条件00002がOFF→ONの変化時、レジスタ19000と19001の内容(バイナリデータ)を減算(-4)カウントします。



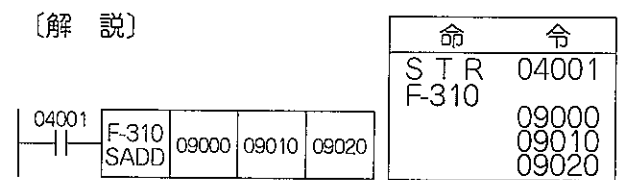
- コ0734~コ0737は特殊領域です。(2・5ページ「特殊リレー」参照)
- Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(9・7ページ「データ処理命令とフラグ」参照)

参考 下記のF命令は働きが類似しています。
F-64、F-64w、F-164、F-164w、F-264、F-264w

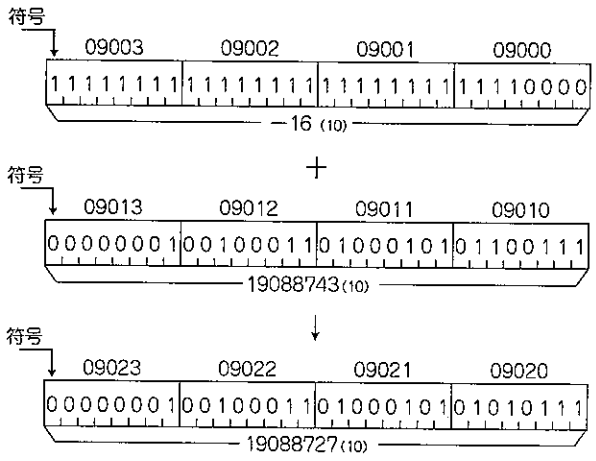
**F-310
SADD**

レジスタ間の符号付バイナリ加算 (31ビット+31ビット)
(Signed ADD)

シンボル	<table border="1"> <tr> <td>F-310 SADD</td> <td>S₁</td> <td>S₂</td> <td>D</td> </tr> </table>				F-310 SADD	S ₁	S ₂	D
F-310 SADD	S ₁	S ₂	D					
機能	レジスタS ₁ ~S ₁ +3の内容とS ₂ ~S ₂ +3の内容を符号付31ビット数として加算して、D~D+3に格納する。							
演算内容	(S ₁ ~S ₁ +3) + (S ₂ ~S ₂ +3) → (D~D+3)							
S ₁ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 774#1 000000~037774		@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 774#1 @000000~@037774					
S ₂ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 774#1 000000~037774		@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 774#1 @000000~@037774					
Dの使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 774#1 000000~037774		@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 774#1 @000000~@037774					
演算条件	入力信号の立上り (OFF→ON)							
演算後	S ₁ ~S ₁ +3の内容	不変						
	S ₂ ~S ₂ +3の内容	不変						
	D~D+3の内容	演算結果 (符号付バイナリ31ビット)						
	フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354		
	±0	1	0	0	1			
	オーバーフロー	0	1	1	0			
	アンダーフロー	0	0	1	1			
	上記以外	0	0	0	1			



入力条件04001がOFF→ONの変化時に、レジスタ09000~09003の内容とレジスタ09010~09013の内容を符号付バイナリ加算して、その結果をレジスタ09020~09023に格納します。



上記演算は、-16 + 19088743 = 19088727 を示します。

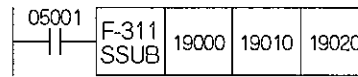
- 扱える数値の範囲は-2147483648~2147483647(10)です。(9・12ページ「符号付演算」参照)
- コ0734~コ0737は特殊領域です。(2・4ページ「特殊リレー」参照)
- S₁、S₂、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

**F-311
SSUB**

レジスタ間の符号付バイナリ減算 (31ビット-31ビット)
(Signed SUB tract)

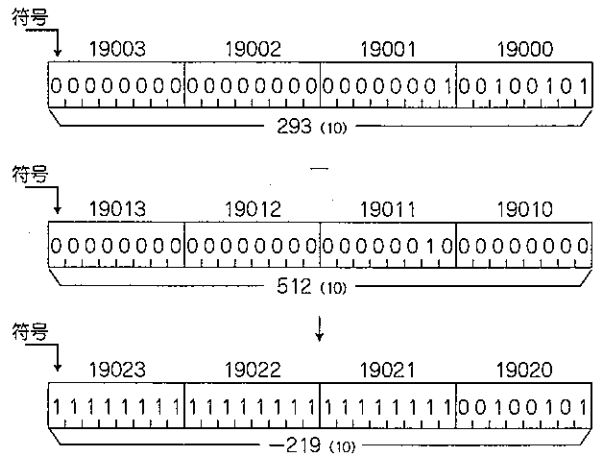
シンボル	— F-311 SSUB S ₁ S ₂ D				
機能	レジスタS ₁ ~S ₁ +3の内容とS ₂ ~S ₂ +3の内容を符号付31ビット数として減算して、D~D+3に格納する。				
演算内容	(S ₁ ~S ₁ +3) - (S ₂ ~S ₂ +3) → (D~D+3)				
S ₁ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774			
S ₂ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774			
Dの使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774			
演算条件	入力信号の立上り (OFF→ON)				
演算後	S ₁ ~S ₁ +3の内容	不変			
	S ₂ ~S ₂ +3の内容	不変			
	D~D+3の内容	演算結果 (符号付バイナリ31ビット)			
	フラグ	演算結果	ゼロ 07357	キャリー 07356	エラー 07355
	±0	1	0	0	1
	オーバーフロー	0	1	1	0
	アンダーフロー	0	0	1	1
	上記以外	0	0	0	1

(解説)



命令	
STR	05001
F-311	19000 19010 19020

入力条件05001がOFF→ONの変化時に、レジスタ19000~19003の内容とレジスタ19010~19013の内容を符号付バイナリ減算して、その結果をレジスタ19020~19023に格納します。



上記演算は、293-512=-219を示します。

- 扱える数値の範囲は-2147483648~2147483647(10)です。(9・12ページ「符号付演算」参照)
- コ0734~コ0737は特殊領域です。(2・4ページ「特殊リレー」参照)
- S₁、S₂、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

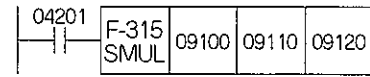
**F-315
SMUL**

レジスタ間の符号付バイナリ乗算 (31ビット×31ビット)
(Signed MUL tiply)

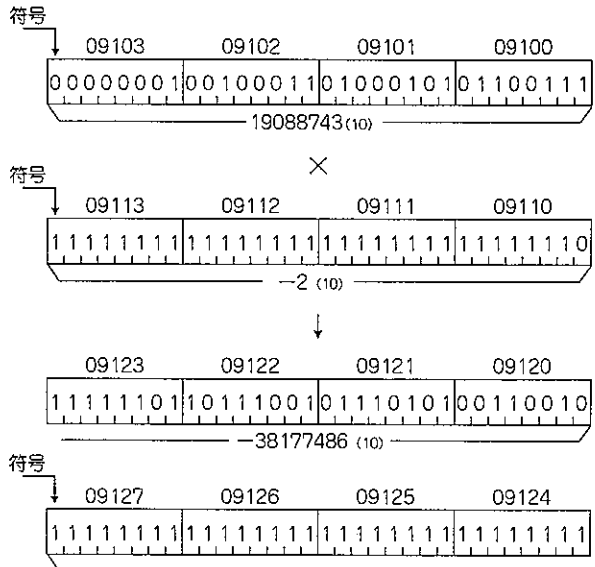
シンボル	<table border="1"> <tr> <td>F-315 SMUL</td> <td>S₁</td> <td>S₂</td> <td>D</td> </tr> </table>				F-315 SMUL	S ₁	S ₂	D
F-315 SMUL	S ₁	S ₂	D					
機能	レジスタS ₁ ~S ₁ +3の内容とS ₂ ~S ₂ +3の内容を符号付31ビット数として乗算して、D~D+7に格納する。							
演算内容	(S ₁ ~S ₁ +3) × (S ₂ ~S ₂ +3) → (D~D+7)							
S ₁ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 774#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 774#1 @000000~@037774						
S ₂ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 774#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 774#1 @000000~@037774						
Dの使用範囲	コ0000~コ1570 コ2000~コ7570 b0000~b1770 b2000~b3770 09000~99770 E0000~E7770 774#1 000000~037770	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 774#1 @000000~@037774						
演算条件	入力信号の立上り (OFF→ON)							
演算	S ₁ ~S ₁ +3の内容	不変						
	S ₂ ~S ₂ +3の内容	不変						
後	D~D+7の内容	演算結果 (符号付63ビット)						
	フラグ	ゼロ 07357	キャリー 07356	エラー 07355	ノンキャリー 07354			
		0	0	0	0			

(解説)

命令	
STR	04201
F-315	09100
	09110
	09120



入力条件04201がOFF→ONの変化時に、レジスタ09100~09103の内容とレジスタ09110~09113の内容を符号付バイナリ乗算して、その結果をレジスタ09120~09127に格納します。



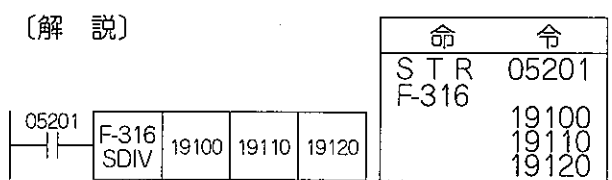
上記演算は、19088743×(-2)=-38177486を示します。

- 扱える数値の範囲は-2147483648~2147483647₍₁₀₎です。(9・12ページ「符号付演算」参照)
- コ0734~コ0737は特殊領域です。(2・4ページ「特殊リレー」参照)
- S₁、S₂、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

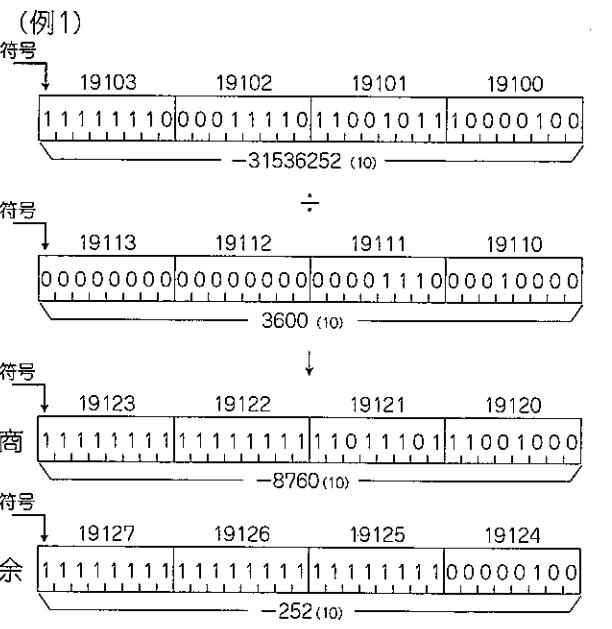
F-316
SDIV

レジスタ間の符号付バイナリ除算 (31ビット÷31ビット)
(Signed DIV ide)

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-316 SDIV</td><td>S₁</td><td>S₂</td><td>D</td></tr></table>				F-316 SDIV	S ₁	S ₂	D
F-316 SDIV	S ₁	S ₂	D					
機能	レジスタS ₁ ~S ₁ +3の内容をS ₂ ~S ₂ +3の内容で符号付31ビット数として除算して、レジスタD~D+3に商を、D+4~D+7に余を格納する。							
演算内容	(S ₁ ~S ₁ +3) ÷ (S ₂ ~S ₂ +3) → (D~D+7)							
S ₁ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774						
S ₂ の使用範囲	コ0000~コ1574 コ2000~コ7574 b0000~b1774 b2000~b3774 09000~99774 E0000~E7774 771#1 000000~037774	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774						
Dの使用範囲	コ0000~コ1570 コ2000~コ7570 b0000~b1770 b2000~b3770 09000~99770 E0000~E7770 771#1 000000~037770	@コ0000~@コ1574 @コ2000~@コ7574 @b0000~@b1774 @b2000~@b3774 @09000~@99774 @E0000~@E7774 771#1 @000000~@037774						
演算条件	入力信号の立上り (OFF→ON)							
演算	S ₁ ~S ₁ +3の内容	不変						
	S ₂ ~S ₂ +3の内容	不変						
	D~D+3の内容	演算結果の商 (符号付31ビット)						
	D+4~D+7の内容	演算結果の余 (符号付31ビット)						
後	レジスタS ₂ ~S ₂ +3の内容	ゼロ 07357	キャリー 07356	エラー 07355	ノキャリー 07354			
	フラグ	0	0	0	1	0		
	上記以外	0	0	0	0			



入力条件05201がOFF→ONの変化時に、レジスタ19100~19103の内容とレジスタ19110~19113の内容を符号付バイナリ除算して、その結果をレジスタ19120~19127に格納します。



上記演算は、 $-31536252 \div 3600 = -8760$ の商と余り -252 (H)を示します。

(例2) $31536252 \div (-3600) = -8760 \cdots 252$

(例3) $-31536252 \div (-3600) = 8760 \cdots -252$

- 扱える数値の範囲は-2147483648~2147483647(10)です。(9・12ページ「符号付演算」参照)
- コ0734~コ0737は特殊領域です。(2・4ページ「特殊リレー」参照)
- S₁、S₂、Dには必ず偶数アドレスを設定してください。(コ0011、19003等は禁止)
- フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(9・7ページ「データ処理命令とフラグ」参照)

改訂履歴

版、作成年月は表紙の右上に記載しております。

版	作成年月	改訂内容
初版	1995年11月	——
改訂1.1版	1996年8月	・説明追加による改訂 ・ユニット追加に伴う改訂 ・誤り修正
改訂2.0版	1997年8月	・JW-31CUH1/32CUH1/33CUH1/33CUH2/33CUH3への対応に伴う改訂
改訂2.1版	1998年11月	・説明改善 1・2、2・13、5・4、5・9、6・1、6・2、6・6、6・8～10
改訂2.2版	2000年9月	・説明改善(追記) 5・15、9・106～111、9・207、9・213

●商品に関するお問い合わせ先

シャープマニファクチャリングシステム(株)

首都圏営業部 〒162-8408 東京都新宿区市谷八幡町8番地 ☎(03)3235-7351
 中部営業部 〒454-0011 名古屋市中川区山王3丁目5番5号 ☎(052)332-2691
 豊田営業所 〒471-0833 豊田市山之手8丁目124番地 ☎(0565)29-0131
 近畿営業部 〒581-8581 大阪府八尾市跡部本町4丁目1番33号 ☎(0729)91-0682
 広島営業所 〒731-0113 広島市安佐南区西原2丁目13番地4号 ☎(082)875-8611

●修理・消耗品についてのお問い合わせ先

シャープドキュメントシステム(株)

札幌技術センター 〒063-0801 札幌市西区二十四軒1条7丁目3番17号 ☎(011)641-0751
 仙台技術センター 〒984-0002 仙台市若林区卸町東3丁目1番27号 ☎(022)288-9161
 宇都宮技術センター 〒320-0833 宇都宮市不動前4丁目2番41号 ☎(028)634-0256
 前橋技術センター 〒371-0855 前橋市周屋町1丁目3番7号 ☎(027)252-7311
 東京フィールド
 サポートセンター 〒114-0012 東京都北区田端新町2丁目2番12号 ☎(03)3810-9962
 横浜技術センター 〒235-0036 横浜市磯子区中原1丁目2番23号 ☎(045)753-9540
 静岡技術センター 〒422-8006 静岡市曲金6丁目8番44号 ☎(054)283-9497
 名古屋技術センター 〒454-0011 名古屋市中川区山王3丁目5番5号 ☎(052)332-2671
 金沢技術センター 〒921-8801 石川県石川郡野々市町字御経塚町1096の1 ☎(076)249-9033
 大阪フィールド
 サポートセンター 〒547-8510 大阪市平野区加美南3丁目7番19号 ☎(06)6794-9721
 岡山技術センター 〒701-0301 岡山県都窪郡早島町大字矢尾828 ☎(086)292-5830
 広島技術センター 〒731-0113 広島市安佐南区西原2丁目13番4号 ☎(082)874-6100
 高松技術センター 〒760-0065 高松市朝日町6丁目2番8号 ☎(087)823-4980
 松山技術センター 〒791-8036 松山市高岡町178の1 ☎(089)973-0121
 福岡技術センター 〒816-0081 福岡市博多区井相田2丁目12番1号 ☎(092)572-2617

・上記の所在地・電話番号などは変わることがあります。その節はご容赦願います。

シャープマニファクチャリングシステム株式会社

本社 〒581-8581 大阪府八尾市跡部本町4丁目1番33号

●インターネットホームページによるシャープ制御機器の情報サービス
<http://www.sharp.co.jp/sms/>

お客様へ……お買いあげ日、販売店名を記入されますと、修理などの依頼のときに便利です。

お買いあげ日	年	月	日
販売店名	電話 ()	局	番

TINSJ5286NCZZ
 00J 0.8 A①
 2000年9月作成