

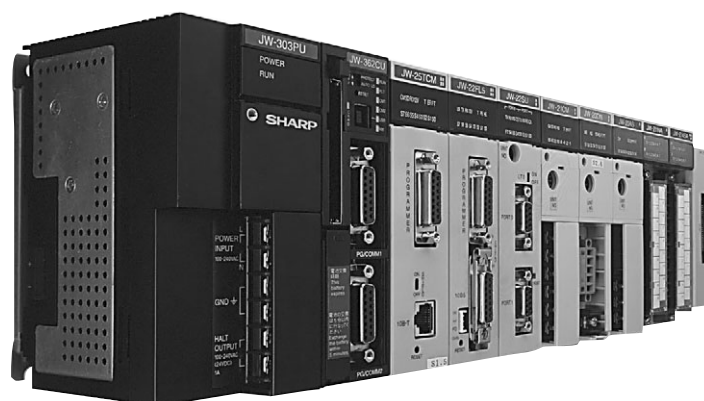
SHARP[®]

改訂1.3版

プログラマブルコントローラ

エヌ サテライト JW300

プログラミングマニュアル・ラダー命令編



このたびは、プログラマブルコントローラJW300をお買いあげいただき、まことにありがとうございます。
本書(プログラミングマニュアル・ラダー命令編)はJW300のデータメモリ・システムメモリ・命令語など、主にソフト的な要素について説明しています。

ご使用前に、本書およびJW300のシステムを構成するユニットのユーザーズマニュアルをよくお読みいただき、各ユニットの機能・使用方法などを十分理解したうえで、正しくご使用ください。

また、JW300のシステム構成等のハード的な要素についての説明は、「JW300ユーザーズマニュアル・ハード編」を参照願います。

本書の記載について

- ・本書は、JW300 コントロールユニット(JW-311CU ~ JW-362CU)のソフトバージョン S2 について記載しています。
- ・本書では、アドレス・設定値の数値を下記で表現しております。
8進数……(8) 16進数……(H) 10進数……(D) または、なし

ご注意

- ・当社制御機器(以下、当社製品)をご使用いただくにあたりましては、万一当社製品に故障・不具合などが発生した場合でも重大な事故に至らない用途であること、および故障・不具合発生時にはバックアップやフェールセーフ機能が機器外部でシステム的に実施されることをご使用の条件とさせていただきます。
- ・当社製品は、一般工業などへの用途を対象とした汎用品として設計・製作されています。したがって、各電力会社様の原子力発電所およびその他発電所向けなどの公共への影響が大きい用途などで、特別品質保証体制をご要求になる用途には、当社製品の適用を除外させていただきます。ただし、これらの用途であっても、用途を限定して特別な品質をご要求されないことをお客様に承認いただいた場合には、適用可能とさせていただきます。
また、航空、医療、鉄道、燃焼・燃料装置、有人搬送装置、娯楽機械、安全機械など人命や財産に大きな影響が予測され、安全面や制御システムに特に高信頼性が要求される用途へのご使用をご検討いただいている場合には、当社の営業部門へご相談いただき、必要な仕様書の取り交しなどをさせていただきます。

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、お気づきのことがありましたらお買いあげの販売店、あるいは当社までご連絡ください。
- ・本書の内容の一部または全部を無断で複製することを禁止しています。
- ・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。

第1章 概 要

第2章 データメモリ

第3章 プログラムメモリ、パラメータメモリ

第4章 システムメモリ

第5章 コントロールユニットの動作

第6章 機能説明(ブロック運転など)

第7章 命 令 語 一 覧

第8章 基本命令の説明

第9章 応用命令の説明

第10章 応用命令(F-00 ~ Fx14d)

第11章 応用命令(F-15 ~ F-49)

第12章 応用命令(F-50 ~ F-79d)

第13章 応用命令(F-80 ~ F-173d)

第14章 応用命令(F-174 ~ F-403)

目 次

第1章 概要	1・1
第2章 データメモリ	2・1～14
2-1 ファイルアドレス	2・1
2-2 メモリマップ	2・2
〔1〕バイトアドレス順	2・2
(1) リレー、TMR・CNT接点、TMR・CNT・MD現在値のバイトアドレス(ファイルアドレス)	2・2
(2) レジスタのバイトアドレス(ファイルアドレス): 全機種共通	2・3
(3) ファイルレジスタのバイトアドレス(ファイルアドレス)	2・4
〔2〕ファイルアドレス順	2・5
2-3 リレー領域	2・6
〔1〕特殊リレー	2・6
2-4 TMR・CNT・MDのデータ格納領域	2・8
2-5 レジスタ領域の予約領域	2・10
〔1〕時計機能で使用する領域	2・10
〔2〕異常履歴格納領域	2・12
(1) 各ユニットへのレジスタ割付	2・12
(2) 異常データの格納順序	2・13
(3) 異常データの内容	2・14
第3章 プログラムメモリ、パラメータメモリ、 シンボル・コメント専用メモリ	3・1～8
3-1 プログラムメモリ	3・1
〔1〕プログラムアドレス	3・1
〔2〕構造化プログラム	3・2
3-2 パラメータメモリ	3・3
〔1〕特殊I/Oユニット用パラメータ	3・3
〔2〕オプションユニット用パラメータ	3・4
3-3 シンボル・コメント専用メモリ	3・5
〔1〕使用上のご注意	3・5
〔2〕設定方法	3・6
〔3〕注意事項	3・8
第4章 システムメモリ	4・1～27
4-1 システムメモリー覧	4・1
4-2 システムメモリの内容	4・5
第5章 コントロールユニットの動作	5・1～6
5-1 運転サイクル	5・1
〔1〕動作フローチャート	5・1
5-2 自己診断	5・2
〔1〕自己診断内容	5・3
〔2〕停止出力	5・6
〔3〕特殊リレー	5・6

- [4] 異常コード 5・6
 - (1) 特殊レジスタ 5・6
 - (2) システムメモリ 5・6
 - (3) レジスタ 5・6
- [5] 異常時の出力ユニットのON/OFF状態 5・6

第6章 ブロック運転、ロギング、故障診断、PCカード、 セキュリティ、編集履歴、モード変更履歴……………6・1～35

- 6 - 1 ブロック運転機能 6・1
 - [1] ブロック状態設定 6・3
 - [2] ブロック起動リレー 6・4
 - [3] I/Oリフレッシュ 6・6
- 6 - 2 ロギング機能 6・7
 - [1] 応用命令F-403(LOG)のプログラミング 6・7
 - [2] システムメモリ 2100～ 2165の設定 6・8
 - [3] 使用例 6・10
 - (1) システムメモリ(2100～ 2155)の設定例 6・10
 - (2) 応用命令F-403(LOG)のプログラム例 6・12
- 6 - 3 故障診断機能 6・13
 - [1] 故障診断の原理 6・14
 - (1) 入出力サイクル 6・14
 - (2) 常時監視リレー 6・14
 - [2] 故障診断の異常内容 6・15
 - (1) 入出力サイクルの異常 6・15
 - (2) 常時監視リレーの異常 6・16
 - [3] 設定内容 6・17
 - [4] 入力リレーのフィルタ機能 6・17
- 6 - 4 PCカードの使用方法 6・18
 - [1] セーブ・ロードの対象ファイル 6・19
 - (1) CFカードの対応フォーマット 6・19
 - (2) ファイルの種類と拡張子 6・19
 - (3) ファイル名 6・19
 - (4) ディレクトリ 6・19
 - (5) 他機種ファイル 6・19
 - [2] 特殊リレー(CFカード関係) 6・20
 - [3] システムメモリの設定 6・20
 - [4] セーブ・ロードの操作 6・22
 - (1) システムメモリによる操作 6・23
 - ファイルセーブの操作 6・23
 - ファイルロードの操作 6・23
 - ファイル自動セーブの操作 6・23
 - (2) 特殊リレーによる操作(ファイルセーブ) 6・24
 - (3) 自動セーブ・ロードの操作 6・24
 - 電源投入時に、CFカードのファイルをJW300にロード(読出)する場合 6・24
 - CFカードを挿入時に、JW300のファイルをCFカードにセーブ(書込)する場合 6・24

6 - 5 セキュリティ機能	6・25
〔 1 〕 使用上のご注意	6・25
〔 2 〕 設定方法	6・26
〔 3 〕 注意事項	6・29
6 - 6 編集履歴機能	6・30
〔 1 〕 使用上のご注意	6・30
〔 2 〕 設定方法	6・30
〔 3 〕 注意事項	6・31
6 - 7 モード変更履歴機能	6・32
〔 1 〕 対象機種	6・32
〔 2 〕 設定方法	6・32
〔 3 〕 モード変更履歴格納データ	6・34
〔 4 〕 注意事項	6・35
第 7 章 命令語一覧	7・1 ~ 24
7 - 1 基本命令一覧	7・1
7 - 2 応用命令一覧	7・3
〔 1 〕 番号順	7・3
〔 2 〕 動作による分類	7・15
7 - 3 命令語の処理時間	7・20
〔 1 〕 基本命令の処理時間	7・20
〔 2 〕 応用命令の処理時間	7・21
第 8 章 基本命令の説明	8・1 ~ 31
8 - 1 基本命令の演算	8・1
(1) データメモリ	8・1
(2) 論理演算部	8・1
(3) アキュムレータ(ACC)	8・1
(4) スタックレジスタ(S1、S2、S3、S4、S5、S6、S7、S8)	8・1
(5) 内部記憶エリア(ACC1、ACC2、ACC3、・・・、ACC64)	8・1
8 - 2 各基本命令の説明	8・2
第 9 章 応用命令の説明	9・1 ~ 27
9 - 1 応用命令に関する留意事項	9・1
〔 1 〕 ソースとデスティネーション	9・1
S、Dの使用範囲について	9・3
(1) 使用範囲 A	9・3
(2) 使用範囲 B	9・4
(3) 使用範囲 C	9・5
(4) 使用範囲 E	9・6
(5) 使用範囲 F	9・7
(6) 使用範囲 G	9・8
(7) 使用範囲 H	9・9
(8) 使用範囲 J	9・10
(9) 使用範囲 K	9・11
〔 2 〕 間接アドレス指定	9・12
間接アドレスの「fileN、アドレスn」	9・13

- 9 - 2 インデックス修飾機能 9・17
 - [1] インデックス修飾のプログラム方法 9・17
 - (1) 通常修飾 9・17
 - (2) 自動修飾 9・17
 - [2] インデックス修飾の適用領域 9・18
 - [3] インデックス修飾のプログラム例 9・19
 - (1) 通常修飾の場合 9・19
 - (2) 自動修飾の場合 9・21
 - (3) TMR・CNT番号をインデックス修飾する例 9・22
 - (4) ラベル番号をインデックス修飾する例 9・22
- 9 - 3 倍長演算機能 9・23
 - (1) 倍長演算の対応命令 9・23
 - (2) 倍長演算時のプログラム 9・23
 - (3) 倍長演算時の内部処理 9・24
 - (4) 倍長演算に関する注意事項 9・24
- 9 - 4 符号付き演算機能 9・25
 - (1) 符号付き演算の対応命令 9・25
 - (2) 符号付き演算の数値範囲 9・25
- 9 - 5 データメモリのブロックと基準アドレス 9・26

第10章 応用命令(F-00 ~ Fx14d)	10・1 ~ 53
第11章 応用命令(F-15 ~ F-49)	11・1 ~ 48
第12章 応用命令(F-50 ~ F-79d)	12・1 ~ 50
第13章 応用命令(F-80 ~ F-173d)	13・1 ~ 55
第14章 応用命令(F-174 ~ F-403)	14・1 ~ 58

第 1 章 概 要

ニューサテライトJW300は、中・大規模制御用の高速・高機能プログラブルコントローラであり、JW30Hシリーズの上位互換機種です。

特長

- (1) 高速処理、大容量メモリ
 - ・ 基本命令 33ns～、応用命令 99ns～（トータル処理速度：当社従来機JW30H比 約20%高速化）
 - ・ プログラムメモリ 最大256K語(JW30H比 約 4 倍)
ファイルレジスタ 最大8Mバイト(JW30H比 約 4 倍)
- (2) メモリカード対応
 - ・ CFカードにプログラム、パラメータなどをバックアップできます。
 - ・ SRAMカードに拡張ファイルメモリとして、ロギングデータなどを格納できます。
- (3) USBポート装備
パソコンとの間でプログラム、データを高速転送できます。
- (4) コミュニケーションポートを 3 ポート装備
コントロールユニットに 2 ポート(JW-311CU/312CUは 1 ポート)、さらにI/Oバス拡張アダプタ(JW-32EA)に 1 ポートのコミュニケーションポートがあり、コントローラターミナルや画像センサカメラなどを容易に接続できます。
- (5) 構造化プログラム/ブロック運転
 - ・ プログラムは複数のブロックに分割して運転できますので、マシン単位での試運転などに有効です。
 - ・ 各ブロックはサブプログラム単位に分割してプログラムできますので、複数人での分業設計が行えるとともに、プログラムを部品化し、標準化・再利用を行えます。
- (6) 設備故障診断機能内蔵
リレー番号と監視時間の設定だけで設備の状態監視を行えますので、従来の異常検出用のラダープログラムを大幅に削減できます。
- (8) 豊富なラインナップ
 - ・ コントロールユニットは10機種をラインナップし、制御規模、予算に応じた選択を行えます。
 - ・ I/Oユニットや特殊I/Oユニットは、JW20H/30H用をすべて使用できます。また、オプションユニットはJW300対応品を使用できます。
 - ・ 構造化プログラム編集用ツールとして、Windows版のラダー設計支援ソフトJW-300SPがあります。
また、現場でのプログラム変更、モニタに便利なハンディプログラマJW-15PGもあります。
- (9) 各種オープンネットに対応
情報系はEthernet、制御系はFL-net、フィールド系はDeviceNet、センサレベルはAS-i に対応し、各階層間でシームレスな情報交換を行えます。

第 2 章 データメモリ

2 - 1 ファイルアドレス

コントロールユニット(JW-3**CU)のデータメモリは、機種によりデータ容量が異なります。

ファイルアドレス(容量)					
JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
00000000 ₍₈₎ } 00073777 ₍₈₎ (30K ₁₆ 1 ₁₆) [リレー 30720点 (3.75K ₁₆ 1 ₁₆) ・TMR/CNT接点 1024点 (0.25K ₁₆ 1 ₁₆) ・TMR/CNT/MD の現在値 (2K ₁₆ 1 ₁₆) ・レジスタ (24K ₁₆ 1 ₁₆)]	00000000 ₍₈₎ } 00105777 ₍₈₎ (35K ₁₆ 1 ₁₆) [リレー 53248点 (6.5K ₁₆ 1 ₁₆) ・TMR/CNT接点 2048点 (0.5K ₁₆ 1 ₁₆) ・TMR/CNT/MD の現在値 (4K ₁₆ 1 ₁₆) ・レジスタ (24K ₁₆ 1 ₁₆)]	00000000 ₍₈₎ } 00177777 ₍₈₎ (64K ₁₆ 1 ₁₆) [リレー 180224点 (22K ₁₆ 1 ₁₆) ・TMR/CNT接点 8192点 (2K ₁₆ 1 ₁₆) ・TMR/CNT/MD の現在値 (16K ₁₆ 1 ₁₆) ・レジスタ (24K ₁₆ 1 ₁₆)]	00200000 ₍₈₎ } 00277777 ₍₈₎ (32K ₁₆ 1 ₁₆) ファイルレジスタ	00200000 ₍₈₎ } 00577777 ₍₈₎ (128K ₁₆ 1 ₁₆)	00200000 ₍₈₎ } 02177777 ₍₈₎ (512K ₁₆ 1 ₁₆)
	00200000 ₍₈₎ 【00000000 ₍₈₎ 】 } 00277777 ₍₈₎ 【00077777 ₍₈₎ 】 (32K ₁₆ 1 ₁₆)	00200000 ₍₈₎ 【00000000 ₍₈₎ 】 } 00377777 ₍₈₎ 【00377777 ₍₈₎ 】 (128K ₁₆ 1 ₁₆)	00200000 ₍₈₎ 【00000000 ₍₈₎ 】 } 01777777 ₍₈₎ 【01777777 ₍₈₎ 】 (512K ₁₆ 1 ₁₆)	00200000 ₍₈₎ 【00000000 ₍₈₎ 】 } 10177777 ₍₈₎ 【07777777 ₍₈₎ 】 (2048K ₁₆ 1 ₁₆)	00200000 ₍₈₎ 【00000000 ₍₈₎ 】 } 40177777 ₍₈₎ 【37777777 ₍₈₎ 】 (8192K ₁₆ 1 ₁₆)

【 】内は、バイトアドレスです。

メモリマップ 2・2-5ページ参照

参 考

・ JW30Hの場合、ファイルレジスタはファイル1～3、10～2C_(H)があり、各64Kバイト(ファイル1のみ16K₁₆ 1₁₆ 独立しています。JW300では、最大8192Kバイト(JW-362CUのとき)の連続アドレスとなります。

なお、JW30HのプログラムをJW-300SR(ラダー設計支援ソフト)によりJW300にプログラム変換すると、「JW30Hのファイル1～3、10～2C_(H)」は「JW300のファイルレジスタ」に変換されます。

2 - 2 メモリマップ

データメモリのメモリマップを、バイトアドレス順(2・2・4ページ)とファイルアドレス順(2・5ページ)に説明します。

〔1〕バイトアドレス順

「リレー、TMR・CNT接点、TMR・CNT・MD現在値」と「ファイルレジスタ」のバイトアドレス(容量)は、コントロールユニットの機種(JW-3**CU)により異なります。(1)(3)

「レジスタ」のバイトアドレス(容量)は、コントロールユニットの全機種で共通です。(2)

(1) リレー、TMR・CNT接点、TMR・CNT・MD現在値のバイトアドレス(ファイルアドレス)

JW-311CU/312CUのとき

領域	バイトアドレス(8)(リレー番号(8))	ファイルアドレス(8)	容量(バイト)
リレー (30720点)	コ00000 ~ コ01577 (000000 ~ 015777)	00000000 ~ 00001577	3840 (3.75K)
	コ02000 ~ コ07577 (020000 ~ 075777)	00030000 ~ 00035577	
TMR・CNT接点 (1024点)	TまたはC 00000 ~ 00777	00001600 ~ 00001777	256 (0.25K)
	TまたはC 01000 ~ 01777	00035600 ~ 00035777	
TMR・CNT・MD 現在値	b00000 ~ b01777	00002000 ~ 00003777	2048 (2K)
	b02000 ~ b03777	00026000 ~ 00027777	

JW-321CU/322CUのとき

領域	バイトアドレス(8)(リレー番号(8))	ファイルアドレス(8)	容量(バイト)
リレー (53248点)	コ00000 ~ コ01577 (000000 ~ 015777)	00000000 ~ 00001577	6656 (6.5K)
	コ02000 ~ コ07577 (020000 ~ 075777)	00030000 ~ 00035577	
	コ10000 ~ コ15377 (100000 ~ 153777)	00074000 ~ 00101377	
TMR・CNT接点 (2048点)	TまたはC 00000 ~ 00777	00001600 ~ 00001777	512 (0.5K)
	TまたはC 01000 ~ 01777	00035600 ~ 00035777	
	TまたはC 02000 ~ 03777	00101400 ~ 00101777	
TMR・CNT・MD 現在値	b00000 ~ b01777	00002000 ~ 00003777	4096 (4K)
	b02000 ~ b03777	00026000 ~ 00027777	
	b04000 ~ b07777	00102000 ~ 00105777	

JW-331CU/332CU、JW-341CU/342CU、JW-352CU、JW-362CUのとき

領域	バイトアドレス(8)(リレー番号(8))	ファイルアドレス(8)	容量(バイト)
リレー (180224点)	コ00000 ~ コ01577 (000000 ~ 015777)	00000000 ~ 00001577	22528 (22K)
	コ02000 ~ コ07577 (020000 ~ 075777)	00030000 ~ 00035577	
	コ10000 ~ コ54377 (100000 ~ 543777)	00074000 ~ 00140377	
TMR・CNT接点 (8192点)	TまたはC 00000 ~ 00777	00001600 ~ 00001777	2048 (2K)
	TまたはC 01000 ~ 01777	00035600 ~ 00035777	
	TまたはC 02000 ~ 17777	00140400 ~ 00143777	
TMR・CNT・MD 現在値	b00000 ~ b01777	00002000 ~ 00003777	16384 (16K)
	b02000 ~ b03777	00026000 ~ 00027777	
	b04000 ~ b37777	00144000 ~ 00177777	

リレー領域内には、次の固定領域があります。

種類(リレー領域内)	バイトアドレス(8)	リレー番号(8)	ファイルアドレス(8)	容量
I/Oリンク[モード7、8] 用リレー	コ00100 ~ コ00177	001000 ~ 001777	00000100 ~ 00000177	512点(64バイト)
特殊リレー	コ00730 ~ コ00737	007300 ~ 007377	00000730 ~ 00000737	64点(8バイト)
オプションユニット用リレー	コ01000 ~ コ01477	010000 ~ 014777	00001000 ~ 00001477	2560点(320バイト)
オプションユニット用フラグ	コ01500 ~ コ01567	015000 ~ 015677	00001500 ~ 00001567	448点(56バイト)
I/Oリンク用フラグ	コ01570 ~ コ01577	015700 ~ 015777	00001570 ~ 00001577	64点(8バイト)
I/Oリンク[モード1~6] 用リレー	コ02000 ~ コ02377	020000 ~ 023777	00030000 ~ 00030377	2048点(256バイト)
特殊I/Oユニット [基本システム]用リレー-1	コ03000 ~ コ03777	030000 ~ 037777	00031000 ~ 00031777	4096点(512バイト)
特殊I/Oユニット [リモートI/O子局]用リレー	コ04000 ~ コ04177	040000 ~ 041777	00032000 ~ 00032177	1024点(128バイト)
特殊I/Oユニット [基本システム]用リレー-2	コ04200 ~ コ05177	042000 ~ 051777	00032200 ~ 00033177	4096点(512バイト)

(2) レジスタのバイトアドレス(ファイルアドレス): 全機種共通

「レジスタ」のバイトアドレス(容量)は、全機種(JW-3**CU)で共通です。

領 域	バイトアドレス(8)	ファイルアドレス(8)	容量(バイト)
レジスタ	009000 ~ 009777	00004000 ~ 00004777	512
	019000 ~ 019777	00005000 ~ 00005777	512
	029000 ~ 029777	00006000 ~ 00006777	512
	039000 ~ 039777	00007000 ~ 00007777	512
	049000 ~ 049777	00010000 ~ 00010777	512
	059000 ~ 059777	00011000 ~ 00011777	512
	069000 ~ 069777	00012000 ~ 00012777	512
	079000 ~ 079777	00013000 ~ 00013777	512
	089000 ~ 089777	00014000 ~ 00014777	512
	099000 ~ 099777	00015000 ~ 00015777	512
	E0000 ~ E0777	00016000 ~ 00016777	512
	E1000 ~ E1777	00017000 ~ 00017777	512
	E2000 ~ E2777	00020000 ~ 00020777	512
	E3000 ~ E3777	00021000 ~ 00021777	512
	E4000 ~ E4777	00022000 ~ 00022777	512
	E5000 ~ E5777	00023000 ~ 00023777	512
	E6000 ~ E6777	00024000 ~ 00024777	512
	E7000 ~ E7777	00025000 ~ 00025777	512
	109000 ~ 109777	00036000 ~ 00036777	512
	119000 ~ 119777	00037000 ~ 00037777	512
	129000 ~ 129777	00040000 ~ 00040777	512
	139000 ~ 139777	00041000 ~ 00041777	512
	149000 ~ 149777	00042000 ~ 00042777	512
	159000 ~ 159777	00043000 ~ 00043777	512
	169000 ~ 169777	00044000 ~ 00044777	512
	179000 ~ 179777	00045000 ~ 00045777	512
	189000 ~ 189777	00046000 ~ 00046777	512
	199000 ~ 199777	00047000 ~ 00047777	512
	209000 ~ 209777	00050000 ~ 00050777	512
	219000 ~ 219777	00051000 ~ 00051777	512
	229000 ~ 229777	00052000 ~ 00052777	512
	239000 ~ 239777	00053000 ~ 00053777	512
	249000 ~ 249777	00054000 ~ 00054777	512
	259000 ~ 259777	00055000 ~ 00055777	512
	269000 ~ 269777	00056000 ~ 00056777	512
	279000 ~ 279777	00057000 ~ 00057777	512
	289000 ~ 289777	00060000 ~ 00060777	512
	299000 ~ 299777	00061000 ~ 00061777	512
	309000 ~ 309777	00062000 ~ 00062777	512
	319000 ~ 319777	00063000 ~ 00063777	512
	329000 ~ 329777	00064000 ~ 00064777	512
	339000 ~ 339777	00065000 ~ 00065777	512
349000 ~ 349777	00066000 ~ 00066777	512	
359000 ~ 359777	00067000 ~ 00067777	512	
369000 ~ 369777	00070000 ~ 00070777	512	
379000 ~ 379777	00071000 ~ 00071777	512	
389000 ~ 389777	00072000 ~ 00072777	512	
Z000 ~ Z377	00073000 ~ 00073777	512	

24K

Zレジスタは2バイト単位です。

(3) ファイルレジスタのバイトアドレス(ファイルアドレス)

JW-311CU/312CUのとき

ファイルレジスタはありません。

JW-321CU/322CUのとき

領 域	バイトアドレス(8)	ファイルアドレス(8)	容量(バイト)
ファイルレジスタ	00000000 ~ 00077777	00200000 ~ 00277777	32K

JW-331CU/332CUのとき

領 域	バイトアドレス(8)	ファイルアドレス(8)	容量(バイト)
ファイルレジスタ	00000000 ~ 00377777	00200000 ~ 00577777	128K

JW-341CU/342CUのとき

領 域	バイトアドレス(8)	ファイルアドレス(8)	容量(バイト)
ファイルレジスタ	00000000 ~ 01777777	00200000 ~ 02177777	512K

JW-352CUのとき

領 域	バイトアドレス(8)	ファイルアドレス(8)	容量(バイト)
ファイルレジスタ	00000000 ~ 07777777	00200000 ~ 10177777	2048K

JW-362CUのとき

領 域	バイトアドレス(8)	ファイルアドレス(8)	容量(バイト)
ファイルレジスタ	00000000 ~ 37777777	00200000 ~ 40177777	8192K

〔 2 〕 ファイルアドレス順

データメモリは、ファイルアドレス順には次のとおりです。

ファイル アドレス ⁽⁸⁾	JW-311CU/312CU	JW-321CU/322CU	JW-331CU/332CU JW-341CU/342CU JW-352CU JW-362CU	バイト アドレス
00000000		リレー (7168点)		000000
00001577				001577
00001600		TMR・CNT 00000 ~ 00777の接点 (512点)		
00001777				b00000
00002000		TMR・CNT・MD 00000 ~ 00777の現在値 (1024バイト)		b01777
00003777				009000
00004000		レジスタ009000 ~ 099777 (5120バイト)		099777
00015777				E0000
00016000		レジスタE0000 ~ E7777 (4096バイト)		E7777
00025777				b02000
00026000		TMR・CNT 01000 ~ 01777の現在値 (1024バイト)		b03777
00027777				002000
00030000		リレー (23552点)		007577
00035577				
00035600		TMR・CNT 01000 ~ 01777の接点 (512点)		
00035777				109000
00036000		レジスタ109000 ~ 199777 (5120バイト)		199777
00047777				209000
00050000		レジスタ209000 ~ 299777 (5120バイト)		299777
00061777				309000
00062000		レジスタ309000 ~ 389777 (4608バイト)		389777
00072777				Z000
00073000		レジスタZ000 ~ Z377 (512バイト)		Z377
00073777				
		00074000 ~ 010000 リレー (22528点)	00074000 ~ 010000	
		00101377 ~ 015377		
		00101400 ~ 0101777 TMR・CNT02000 ~ 03777 の接点 (1024点)	00140377 ~ 0154377	
		00101777 ~ 0102000 TMR・CNT・MD 02000 ~ 03777 の現在値 (2048バイト)	00140400 ~ 0101777 TMR・CNT02000 ~ 17777 の接点 (7168点)	
		00105777 ~ 0105777	00143777 ~ 01044000 TMR・CNT・MD 02000 ~ 17777 の現在値 (14336バイト)	
			00177777 ~ 0103777	
		ファイルレジスタ	ファイルレジスタ	

ファイルレジスタについては、2・4ページを参照願います。

2 - 3 リレー領域

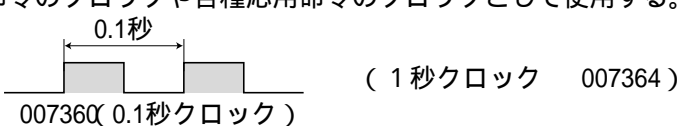
〔1〕特殊リレー

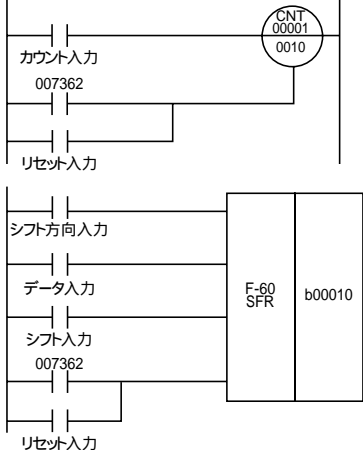
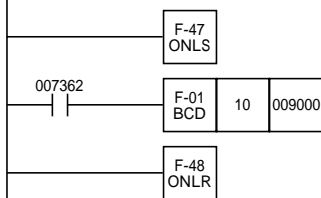
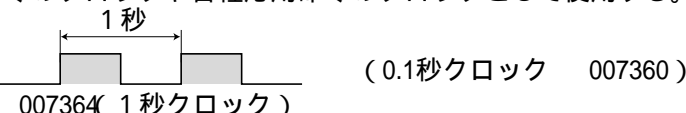
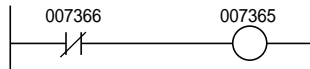
リレー番号007300～007377(コ00730～コ00737)の64点は、特殊リレー領域です。

- ・特殊リレー(007365を除く)は、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、応用命令のソースとして使用してください。

OUT命令、応用命令のデスティネーションとしては使用できません。2バイト以上のデータメモリを扱う命令、分配・抽出命令、一括転送命令では特に注意願います。

- ・予約領域(下表参照)は、ユーザープログラムでは使用しないでください。

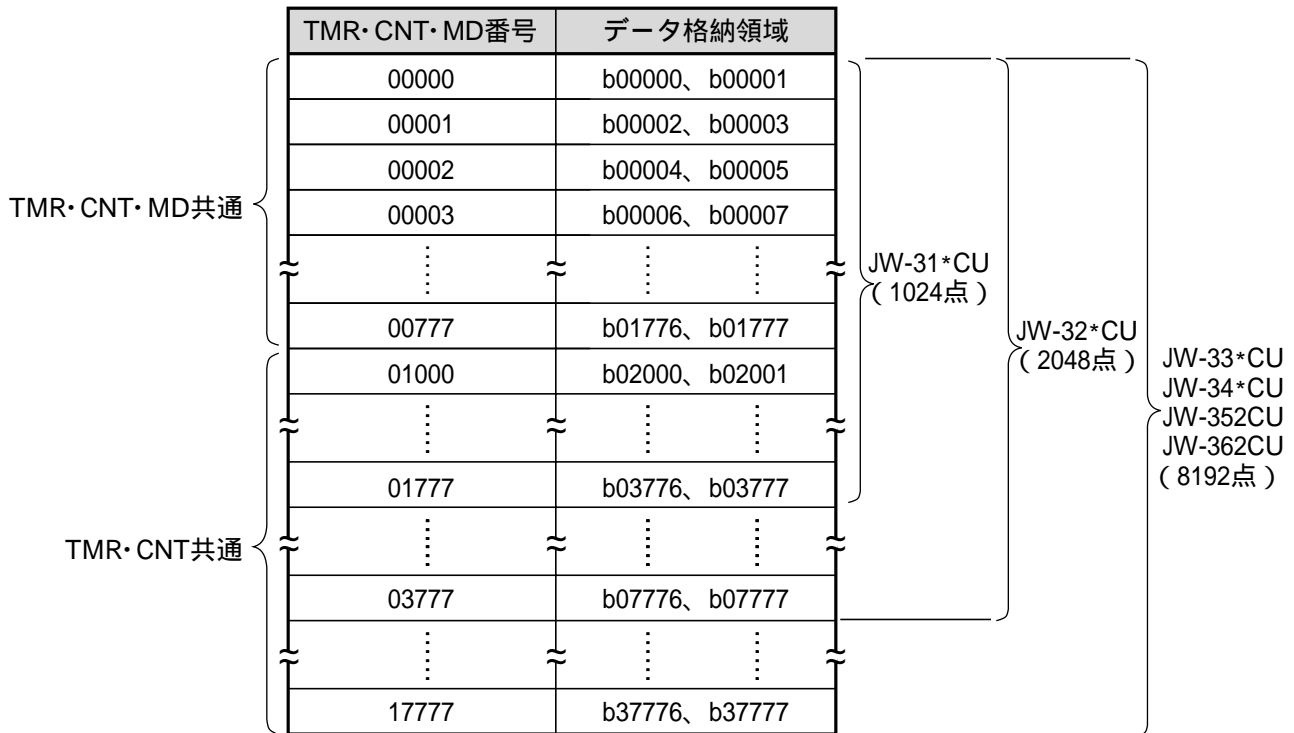
リレー番号	内 容	
007300	CFカード書込時のサムチェック異常発生 ・CFカードへプログラム保存する場合のみ、保存開始時にプログラムのサムチェックを計算し、計算したサムチェックがPLC内のサムチェックと異なるとONします。 ・サポートツールからのプログラム書込(全ブロック)、CFカードからのプログラムロード、CFカードへのプログラム保存が正常に行えたときにはOFFします。	
007301 ～007327	予約領域	
007330	MWフラグ ・プログラムメモリを変更するとONし、コントロールユニット(JW-3**CU)のMWランプが点滅する。	
007331	MWリセット ・MWフラグ(007330)のリセットとして使用する。 MWフラグがONのとき、MWリセット(007331)をOFF ONすると、MWフラグはOFFし、コントロールユニットのMWランプが消灯する。MWリセットも同時にOFFする。 MWフラグとMWランプを無効(OFF)にする場合は、ユーザープログラムにて、MWリセットリレー(007331)を常時ONにしてください。	
007332	CFカードへセーブ実行 ・立上り(OFF ON)で、システムメモリ#2221に種類を設定したファイルをセーブ(JW300 CFカード)します。 ファイル名は、システムメモリ#2230～#2236に設定します。 6・24ページ	
007333	CFカードビジーフラグ ・CFカードを操作中、ONします。 6・20ページ	
007334	CFカードエラーフラグ ・CFカードを操作時に異常が発生すると、ONします。 6・20ページ	
007335 ～007337	予約領域	
007340 ～007347	異常コードの格納 ・自己診断の結果、異常が発生すると、その異常コード(5・2ページ)が格納される。 ・特殊レジスタのバイトアドレス「コ00734」として扱う。 ・異常が回復すると、異常コードはクリアされる。	
007350 ～007353	予約領域	
007354	ノンキャリーフラグ	・フラグに影響を与える応用命令を実行時、演算内容に応じてセットされる。
007355	エラーフラグ	
007356	キャリーフラグ	
007357	ゼロフラグ	
007360	0.1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する。  (1秒クロック 007364)	
007361	予約領域	

リレー番号	内 容
007362	<p>イニシャライズパルス</p> <ul style="list-style-type: none"> ・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する。 ・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)に使用する。  <p>・イニシャライズパルスを上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る。</p> <p>【例】</p> 
007363	<p>ヒューズ切れ</p> <ul style="list-style-type: none"> ・JW-262S(64点出力ユニット)のヒューズ切れを検出時にONする。
007364	<p>1秒クロック</p> <ul style="list-style-type: none"> ・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する。  <p>007364(1秒クロック) (0.1秒クロック 007360)</p>
007365	<p>設定値変更スイッチ</p> <ul style="list-style-type: none"> ・サポートツールでキーリレー領域(初期状態:007000~015777,020000~075777)以外のリレーをセット・リセットするとき、プログラム上で007365をONする必要があります。 ・常時ONとするプログラム  <p>常時OFFの接点007366を使用する。</p>
007366	<p>常時OFFの接点</p> <ul style="list-style-type: none"> ・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する。
007367	予約領域
007370	メモリ異常
007371	CPU異常
007372	電池異常
007373	入出力異常
007374	オプション異常
007375	特殊I/Oユニット異常フラグ
007376	増設電源異常フラグ
007377	電源異常

自己診断結果
・自己診断の結果、異常内容に応じた接点がONする。
・詳細は「自己診断」参照。 5・2ページ

2 - 4 TMR・CNT・MDのデータ格納領域

b*****のレジスタは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1点あたり2バイトを使用します。TMR・CNT・MD番号とb*****の関係は次のとおりです。



b*****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		

分類1のデータフォーマット

	7	6	5	4	3	2	1	0	
MD	$(\times 10^1)$				$(\times 10^0)$				n
	8	4	2	1	8	4	2	1	
	1	入力情報			$(\times 10^2)$				n+1
		S ₁	S ₂	S ₃	8	4	2	1	

- ・数値はBCDで扱います。
- ・n、n+1はアドレス順を表します。

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	
TMR	$(\times 10^0)$				$(\times 10^{-1})$				n
	8	4	2	1	8	4	2	1	
	リセット	$(\times 10^2)$			$(\times 10^1)$				n+1
		4	2	1	8	4	2	1	
CNT	$(\times 10^1)$				$(\times 10^0)$				n
	8	4	2	1	8	4	2	1	
	リセット	$(\times 10^3)$			$(\times 10^2)$				n+1
		4	2	1	8	4	2	1	
DTMR (BCD)	$(\times 10^0)$				$(\times 10^{-1})$				n
	8	4	2	1	8	4	2	1	
		$(\times 10^2)$			$(\times 10^1)$				n+1
		4	2	1	8	4	2	1	
UTMR (BCD)	$(\times 10^0)$				$(\times 10^{-1})$				n
	8	4	2	1	8	4	2	1	
	リセット	$(\times 10^2)$			$(\times 10^1)$				n+1
		4	2	1	8	4	2	1	
DCNT (BCD)	$(\times 10^1)$				$(\times 10^0)$				n
	8	4	2	1	8	4	2	1	
	リセット	$(\times 10^3)$			$(\times 10^2)$				n+1
		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	$(\times 10^3)$			$(\times 10^2)$				n+1
		4	2	1	8	4	2	1	

- ・数値はBCDで扱います。
- ・n、n+1はアドレス順を表します。
- ・TMR、CNT、U、Dの判別はJW300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・CNTの動作中は1(ON)、非計測はリセット状態でも1(ON)となります。

2 - 5 レジスタ領域の予約領域

〔 1 〕 時計機能で使用する領域

JW300では、コントロールユニットの全機種(JW-3**CU)に時計機能があります。
 時計データの読出と時刻設定に、レジスタ 099770 ~ 099777 の 8 バイトを使用します。
 ただし、時計機能で上記レジスタを使用しないときは、システムメモリ #0223 に 01_(H)を設定してください。 4・11ページ参照

レジスタ	内 容
099770	秒：00 ~ 59 (BCD)
099771	分：00 ~ 59 (BCD)
099772	時：00 ~ 23 (BCD)
099773	日：01 ~ 31 (BCD)
099774	月：01 ~ 12 (BCD)
099775	年：00 ~ 99 (BCD)
099776	曜日：0 ~ 6 (BCD)
099777	コントロール

1. 月ごとの「日付」と「うるう年」の計算は、自動的に行います。
2. 「年」は、西暦の下 2 桁を表します。
 「うるう年」の自動判別は、4 年ごとの年数のみの判別です。
 (04 年、08 年、12 年は「うるう年」と判別)
3. 「曜日」は、時刻を設定時に合わせてください。
 日付が変わるごとに順次、0 ~ 6 に変化します。曜日は、年月日設定による計算は行いません。

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

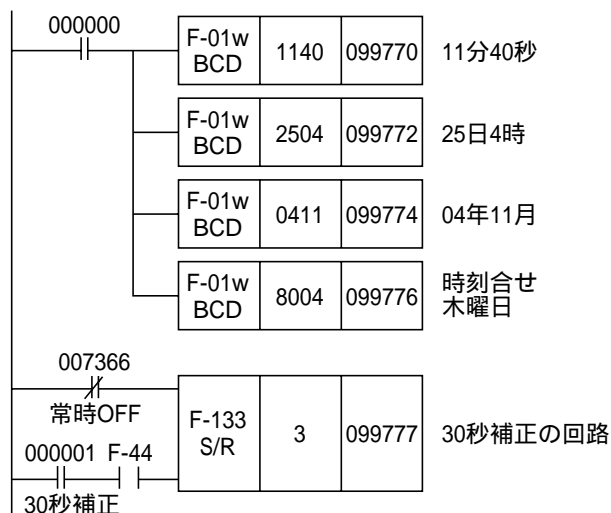
4. 「コントロール」は、レジスタ 099777 のビット(D0、D3、D7)で設定(ON / OFF)します。

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

- ・ビット D0 は、「時計の運転 / 停止」を設定します。
- ・ビット D3 は、「30 秒補正」に使用します。現在時刻の秒値により、補正(桁上げ)が変わります。
 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)でビット D3 をセット(ON)してください。
 補正が完了すると、自動的にリセット(OFF)されます。

- ・ビットD7は、ONで「時刻合せ」、OFFで「時刻モニタ」になります。
時刻合せは、I/O演算の終了後に行うため、下例の回路で行うと、時計を停止(ビットD0をON)しなくても使用できます。
ビットD7は、ONにすると、時刻合せ完了で自動的にリセット(OFF)されます。

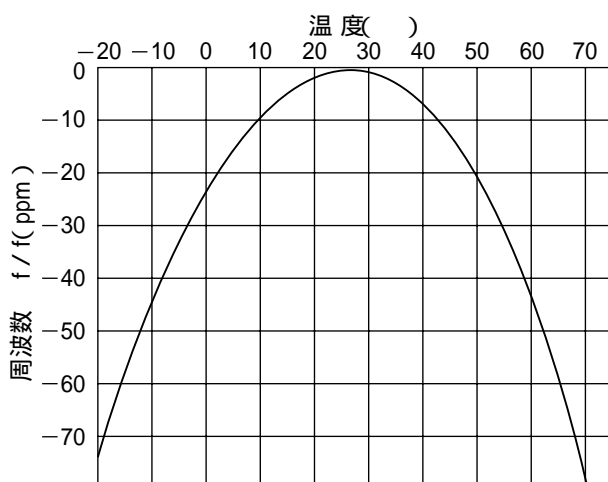
【例】2004年11月25日4時11分40秒木曜日にセットするとき



留意点

- ・レジスタ099777のビットD0、D7は、応用命令等で連続ONにしないでください。時計が正常に動作しなくなります。
- ・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(at 25)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 #0010 ~ #0017 も、レジスタ 099770 ~ 099777 と同じ機能です。 4・5ページ参照

〔 2 〕 異常履歴格納領域

レジスタ E5600 ~ E7777 には、オプションユニット(JW-21CM/22CM/21MN/25CM)とコントロールユニットの異常履歴が格納されます。 - システムメモリ #0213 = 02_H(初期設定)のとき #0213 = 00_Hに設定すると、E5600 ~ E7777 は汎用レジスタとして使用できます。

4・11ページ参照

(1) 各ユニットへのレジスタ割付

レジスタ E5600 ~ E7777(1152バイト)は、異常履歴格納領域として、1ユニットあたり 128 バイトを割り付けています。

レジスタ	ユニットNo.スイッチ の設定値	バイト数	内 容
E5600 } E5777	7	128	・オプションユニット(JW-21CM/22CM/21MN/25CM)の異常データ
E6000 } E6177	6	128	
E6200 } E6377	5	128	
E6400 } E6577	4	128	
E6600 } E6777	3	128	
E7000 } E7177	2	128	
E7200 } E7377	1	128	
E7400 } E7577	0	128	
E7600 } E7777	コントロール ユニット	128	・コントロールユニット、PGインターフェイス、 コミュニケーションポートの異常データ

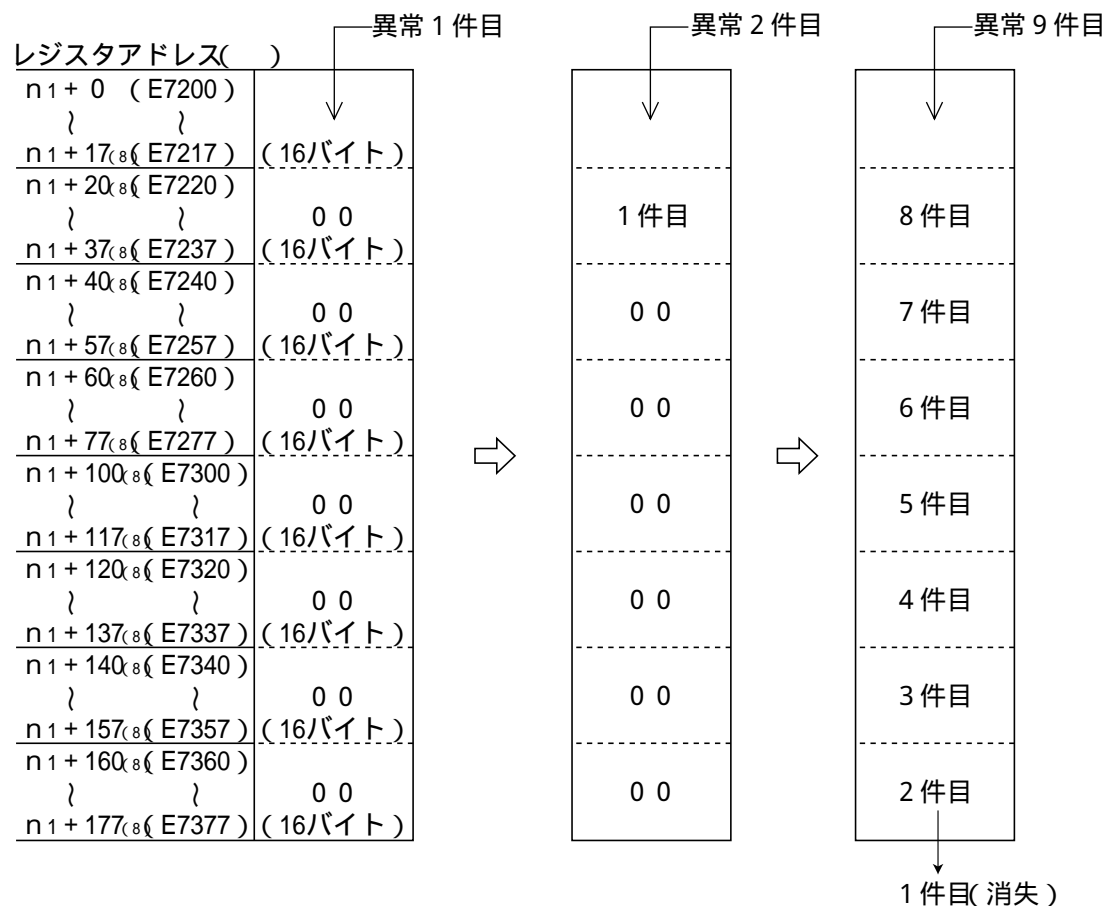
E5600 ~ E7577 は、オプションユニット(上記)のユニット No. スイッチ設定値(0 ~ 7)により分割しています。

(2) 異常データの格納順序

(1)で各ユニットに割り付けられた異常履歴格納領域(128バイト)は、16バイトづつ8分割され、異常データ(16バイト：次ページ)が発生順に8回まで格納されます。

異常が9回以上になると、最初に格納された異常データから順に消失します。

各ユニットの異常履歴格納領域(128バイト)



()内のアドレスは、オプションユニットでユニットNo.スイッチを1に設定時です。

(3) 異常データの内容

異常履歴格納領域に格納される異常データは、異常1件あたり16バイトで構成しています。

異常データ(16バイト)

アドレス(1)	内 容	
n2+0 (E7400)	秒	1回目の発生日時
n2+1 (E7401)	分	
n2+2 (E7402)	時	
n2+3 (E7403)	日	
n2+4 (E7404)	月	
n2+5 (E7405)	年	
n2+6 (E7406)	曜日	
n2+7 (E7407)	異常コード	2
n2+10 ₍₈₎ (E7410)	3	
n2+11 ₍₈₎ (E7411)	発生回数	000~377 ₍₈₎ 4
n2+12 ₍₈₎ (E7412)	秒	最後の発生日時
n2+13 ₍₈₎ (E7413)	分	
n2+14 ₍₈₎ (E7414)	時	
n2+15 ₍₈₎ (E7415)	日	
n2+16 ₍₈₎ (E7416)	月	
n2+17 ₍₈₎ (E7417)	年	

1 先頭アドレスがE7400のとき

2 コントロールユニット(JW-3**CU)のとき、「自己診断」の異常コード(5・2ページ)が格納されます。オプションユニットのとき、各ユニットの異常コードが格納されます。

3 入出力ユニットが異常のとき、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)を示します。オプションユニット、デバイスネットユニット、I/Oリンク親局ユニットが異常時は、00_(H)が格納されます。

4 同じ異常コードが発生すると、発生回数を+1して、377₍₈₎回まで加算されます。400₍₈₎回以上は377₍₈₎のままとなります。

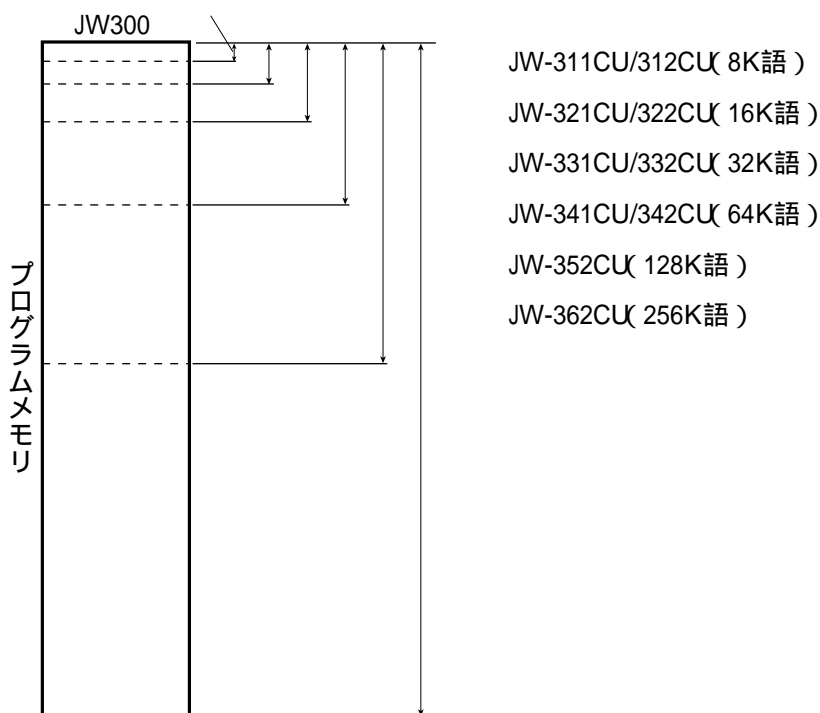
第 3 章 プログラムメモリ、パラメータメモリ、シンボル・コメント専用メモリ

3 - 1 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、JW300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。

〔 1 〕 プログラムアドレス

コントロールユニットの機種(JW-3**CU)により、プログラム容量が異なります。



・プログラムアドレスは次のとおりです。

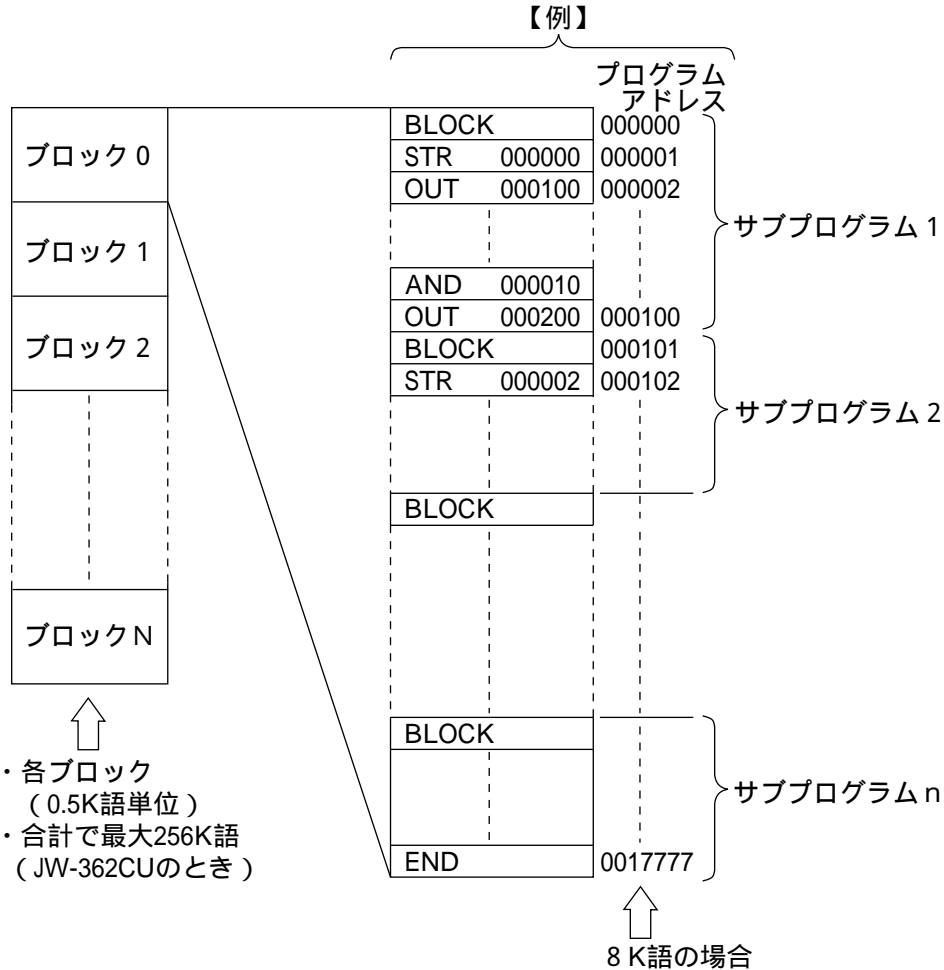
コントロールユニット	プログラムメモリ容量	プログラムアドレス		
		8進数	10進数	16進数
JW-311CU/312CU	8K語	000000 ~ 017777	000000 ~ 008191	00000 ~ 01FFF
JW-321CU/322CU	16K語	000000 ~ 037777	000000 ~ 016383	00000 ~ 03FFF
JW-331CU/332CU	32K語	000000 ~ 077777	000000 ~ 032767	00000 ~ 07FFF
JW-341CU/342CU	64K語	000000 ~ 177777	000000 ~ 065535	00000 ~ 0FFFF
JW-352CU	128K語	000000 ~ 377777	000000 ~ 131071	00000 ~ 1FFFF
JW-362CU	256K語	000000 ~ 777777	000000 ~ 262143	00000 ~ 3FFFF

・命令語には 1 語命令、2 語命令、3 語命令、4 語命令があります。

	命令語
1 語命令	STR、AND等
2 語命令	TMR、CNT等
3 語命令	F-00、F-01等
4 語命令	F-10、F-11等

〔 2 〕 構造化プログラム

JW300は、1台のコントロールユニットに複数のプログラム(以下、ブロック)を格納できます。
 各ブロックは、プログラム容量を0.5K語単位にJW-300SP(ラダー設計支援ソフト)で設定します。
 各ブロックの最終アドレスには、END命令(F-40)が書き込まれます。
 また、各ブロックは、JW-300SPで複数のサブプログラムに分割してプログラムできます。
 各サブプログラムの先頭アドレスには、識別命令(BLOCK)が書き込まれます。



設定可能なブロック数、サブプログラム数は、コントロールユニットの機種(JW-3**CU)により異なります。

	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
ブロック数 (最大)	16	32	64	128	256	512
サブプログラム数 (最大)	256	512	1024	2048	4096	8192

ブロックの動作については、「6 - 1 ブロック運転機能」を参照願います。

留 意 点

- ・ JW-15PG(ハンディプログラマ)は、ブロック、サブプログラムの分割を行えません。
 JW-300SPで分割したブロック - サブプログラムの編集のみ行えます。
- ・ ブロック単位にメモリクリアすると、ブロックの先頭アドレスにBLOCK命令、最終アドレスにEND命令、その他のアドレスにNOP命令が書き込まれます。

3 - 2 パラメータメモリ

JW300における特殊I/Oユニット、オプションユニットのパラメータ領域を説明します。

〔1〕特殊I/Oユニット用パラメータ

特殊I/Oユニットの下記機種については、動作条件をコントロールユニット(JW-3**CU)の特殊I/Oパラメータに、サポートツール(JW-15PG、JW-300SP)を使用して設定します。

JW-21HC、JW-22HC、JW-24AD、JW-22DA、JW-21PS、JW-21SU

- ・特殊I/Oパラメータ領域は、特殊I/OユニットのユニットNo. スイッチ(設定値0～7)により決定され、1ユニットあたり256バイトを使用します。
- ・特殊I/Oパラメータの内容は、特殊I/Oユニット(JW-21HC等)の各ユーザズマニュアルを参照願います。

ユニットNo. スイッチ の設定値	特殊I/Oパラメータのアドレス ⁽⁸⁾							
	ラック0		ラック1		ラック2		ラック3	
0	T-00	000～377	T-10	000～377	T-20	000～377	T-30	000～377
1	T-01	000～377	T-11	000～377	T-21	000～377	T-31	000～377
2	T-02	000～377	T-12	000～377	T-22	000～377	T-32	000～377
3	T-03	000～377	T-13	000～377	T-23	000～377	T-33	000～377
4	T-04	000～377	T-14	000～377	T-24	000～377	T-34	000～377
5	T-05	000～377	T-15	000～377	T-25	000～377	T-35	000～377
6	T-06	000～377	T-16	000～377	T-26	000～377	T-36	000～377
7	T-07	000～377	T-17	000～377	T-27	000～377	T-37	000～377

ユニットNo. スイッチ の設定値	特殊I/Oパラメータのアドレス ⁽⁸⁾									
	ラック4		ラック5		ラック6		ラック7		リモート/O子局	
0	T-40	000～377	T-50	000～377	T-60	000～377	T-70	000～377	T-80	000～377
1	T-41	000～377	T-51	000～377	T-61	000～377	T-71	000～377	T-81	000～377
2	T-42	000～377	T-52	000～377	T-62	000～377	T-72	000～377	T-82	000～377
3	T-43	000～377	T-53	000～377	T-63	000～377	T-73	000～377	T-83	000～377
4	T-44	000～377	T-54	000～377	T-64	000～377	T-74	000～377	T-84	000～377
5	T-45	000～377	T-55	000～377	T-65	000～377	T-75	000～377	T-85	000～377
6	T-46	000～377	T-56	000～377	T-66	000～377	T-76	000～377	T-86	000～377
7	T-47	000～377	T-57	000～377	T-67	000～377	T-77	000～377	T-87	000～377

参 考

- ・JW30Hの場合、ラック4～7には特殊I/Oユニットを実装できません。
また、特殊I/Oパラメータ領域は、1ユニットあたり128バイトを使用します。

〔 2 〕 オプションユニット用パラメータ

オプションユニットの下記機種については、動作条件をコントロールユニット(JW-3**CU)のオプションパラメータに、サポートツール(JW-15PG、JW-300SP)を使用して設定します。

JW-21CM、JW-22SU、JW-20FL5/20FLT、JW-22FL5/22FLT、
JW-255CM、JW-25TCM、JW-22CM、JW-21MN、JW-300CM

- ・オプションパラメータ領域は、オプションユニットのユニットNo.スイッチ(設定値0～7)により決定され、1ユニットあたり2Kバイトを使用します。
- ・オプションパラメータの内容は、オプションユニット(JW-21CM等)の各ユーザーズマニュアルを参照願います。

ユニットNo.スイッチ の設定値	オプションパラメータ のアドレス ⁽⁸⁾
0	00000 ~ 03777
1	00000 ~ 03777
2	00000 ~ 03777
3	00000 ~ 03777
4	00000 ~ 03777
5	00000 ~ 03777
6	00000 ~ 03777
7	00000 ~ 03777

参 考

- ・JW30Hの場合、JW-255CM、JW-25TCM、JW-22CM、JW-21MN、JW-300CM(Ethernetモード時)のパラメータは各ユニットに設定します。また、オプションパラメータ領域は、1ユニットあたり64バイトを使用します。

3 - 3 シンボル・コメント専用メモリ

シンボル・コメント専用メモリとは、シンボル・コメントを格納する専用のメモリエリアです。従来のようにファイルメモリを使用することなく、シンボル・コメントを格納できます。

〔 1 〕 使用上のご注意

(1) 対象機種について

シンボル・コメント専用メモリは、ソフトウェアバージョン S2.00 以上の JW300 コントロールユニット (JW-311CU ~ JW-362CU)にて使用できます。

(2) サポートツールについて

シンボル・コメント専用メモリを使用する場合、JW-300SP のソフトウェアバージョン 3.0 以上が必要となります。シンボル・コメント未対応の JW-300SP では使用できません。

(3) シンボル・コメント専用メモリサイズについて

各コントロールユニットのシンボル・コメント専用メモリサイズは、次のとおりです。

コントロールユニット	メモリサイズ
JW-311CU / JW-312CU	128Kバイト
JW-321CU / JW-322CU	256Kバイト
JW-331CU / JW-332CU	256Kバイト
JW-341CU / JW-342CU	256Kバイト
JW-352CU	512Kバイト
JW-362CU	616Kバイト

〔 2 〕 設定方法

シンボル・コメント専用メモリの設定は、システムメモリ(#0410 ~ #0414)で行います。
 転送条件を「シンボル・コメント専用メモリに転送する場合」で設定してください。

システムメモリ番号(8)	設定項目	内 容																																																																																																																																
# 0410 } # 0413	シンボル・コメントの転送条件の設定	<p>ラダー設計支援ソフトJW-300SPにて、リレー、タイマ・カウンタ、レジスタ、ファイルレジスタ、F-90(REM)命令のシンボル・コメントをJW300に転送(書込 / 読出)する条件、およびシンボル・コメントの種類を設定します。</p> <p>転送条件(#0410 ~ #0413)</p> <p>ファイルレジスタに転送する場合 転送する先頭アドレスとして、ファイルレジスタのファイルアドレスを#0410 ~ #0412に設定します。#0413には00_(H)を設定します。</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>#0411</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </table> <p>5桁目 4桁目 3桁目</p> </div> <div style="text-align: center;"> <p>#0410</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </table> <p>2桁目 1桁目</p> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="text-align: center;"> <p>#0413</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> <p>00_(H)</p> </div> <div style="text-align: center;"> <p>#0412</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </table> <p>8桁目 7桁目 6桁目</p> </div> </div> <p>【例】ファイルレジスタ17560000₍₈₎(ファイルアドレス17760000₍₈₎) が先頭アドレスのとき</p> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="text-align: center;"> <p>#0411</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> <p>6 0 0 0 0 0</p> </div> <div style="text-align: center;"> <p>#0410</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> <p>0 0 0 0</p> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="text-align: center;"> <p>#0413</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> </div> <div style="text-align: center;"> <p>#0412</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table> <p>1 7 7</p> </div> </div> <p>PCカードに転送する場合 転送する先頭アドレスとして、PCカードのアドレスを#0410 ~ #0413に設定します。#0413には80_(H)を設定します。</p> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="text-align: center;"> <p>#0411</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </table> <p>5桁目 4桁目 3桁目</p> </div> <div style="text-align: center;"> <p>#0410</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </table> <p>2桁目 1桁目</p> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="text-align: center;"> <p>#0413</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> <p>80_(H)</p> </div> <div style="text-align: center;"> <p>#0412</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </table> <p>8桁目 7桁目 6桁目</p> </div> </div> <p>シンボル・コメント専用メモリに転送する場合 #0410、#0411に00_(H)、#0412に01_(H)、#0413にはシンボル・コメントのJW-300SP形式により9A_(H)または9B_(H)を設定します。</p> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="text-align: center;"> <p>#0411</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> <p>00_(H)</p> </div> <div style="text-align: center;"> <p>#0410</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> <p>00_(H)</p> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="text-align: center;"> <p>#0413</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td> </td></tr> </table> <p>9A_(H)または9B_(H)</p> </div> <div style="text-align: center;"> <p>#0412</p> <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> </table> <p>01_(H)</p> </div> </div> <p>【シンボル・コメントのJW-300SP形式】 [0 : Ver3.0形式の場合 [1 : Ver3.1形式の場合</p>																	0	0	0	0	0	0	0	0									1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1																	1	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1		0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0																																																																																																																											
1	1	1	0	0	0	0	0																																																																																																																											
0	0	0	0	0	0	0	0																																																																																																																											
0	0	0	0	0	0	0	0																																																																																																																											
0	0	0	1	1	1	1	1																																																																																																																											
1	0	0	0	0	0	0	0																																																																																																																											
0	0	0	0	0	0	0	0																																																																																																																											
0	0	0	0	0	0	0	0																																																																																																																											
1	0	0	1	1	0	1																																																																																																																												
0	0	0	0	0	0	0	1																																																																																																																											
# 0414	シンボル・コメントの種類の設定	<p>シンボル・コメントの種類(#0414) #0414のビット(5 ~ 0)をON(1)して、種類を設定します。</p> <p style="text-align: center;">ビット 7 6 5 4 3 2 1 0</p> <p>#0414 <table border="1" style="border-collapse: collapse; width: 100px; height: 20px;"> <tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </table> (初期値 : 00_(H))</p> <p>サブコメント ———— シンボル(グローバル) ネットワークコメント ———— コメント(グローバル) コメント(ローカル) ———— シンボル(ローカル)</p>																																																																																																																																

JW-300SPのプロパティシートで設定する場合



内 容	
	<p>転送先メモリにシンボル・コメント専用メモリ(画面の「コメント専用メモリ」)を指定し、「V3.0形式」または「V3.1形式」を選択してください。</p> <ul style="list-style-type: none"> ・ JW-300SPのソフトバージョンVer3.0とVer3.1以上ではフォーマットが異なります。PLCのシンボル・コメント専用メモリにV3.0形式で書き込んだ場合はVer3.0以上で読み出し可能ですが、V3.1形式で書き込んだ場合はVer3.0では読み出し不可です。 ・ 転送先メモリを「コメント専用メモリ」に変更した場合の初期値はV3.1形式です。V3.1形式を選択時には次のメッセージが表示されます。 <div data-bbox="598 1144 946 1308" data-label="Image"> </div>
	転送するコメントの種類を選択します。
	転送対象となるシンボル・コメントのサイズが表示されます。

[3] 注意事項

(1) 転送先メモリについて

転送先メモリには複数を選択できません。(例えばコメント専用メモリとファイルメモリなど)

(2) PC カードについて

ファイルをセーブすると次の拡張子になります。

ファイルの種類	拡張子	内 容
シンボル・コメント	psc	シンボル・コメント専用メモリのデータがすべてセーブ・ロードされます。

システムメモリにファイルの種類を設定する場合、次のように設定します。

システムメモリ番号 ⁽⁸⁾	設定項目	内 容								
# 2221	ファイル(種類)の選択	<p>CFカードとセーブ・ロードするファイルの種類を設定します。</p> <p>ビット 7 6 5 4 3 2 1 0</p> <p>#2221 <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="width: 20px; text-align: center;">0</td> <td style="width: 20px; text-align: center;">0</td> <td style="width: 20px;"></td> <td style="width: 20px;"></td> <td style="width: 20px;"></td> <td style="width: 20px;"></td> <td style="width: 20px;"></td> <td style="width: 20px;"></td> </tr> </table> (初期値: 3F_H)</p> <p style="margin-left: 150px;"> </p> <ul style="list-style-type: none"> — ユーザープログラム(ブロック情報とブロック、サブプログラムのタイトルを含む) — システムメモリ — パラメータメモリ — データメモリ(ファイルレジスタを除く) — ファイルレジスタ — シンボル・コメントメモリ <p>・ JW-312CUのとき、ファイルレジスタを選択できません。</p>	0	0						
0	0									

第 4 章 システムメモリ

システムメモリは、JW300の各種機能を設定、およびJW300の異常内容等をモニタする場合に使用します。

4 - 1 システムメモリー覧

システムメモリは#0000～#2777(1.5Kバイト)のメモリで、電池によりバックアップしています。本項ではユーザーに開放のメモリ番号を説明します。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⁾	初期値 (H)	内 容	詳細 ページ	JW30H
#0010	時刻	秒	4・5	
#0011		分		
#0012		時		
#0013		日		
#0014		月		
#0015		年		
#0016		曜日		
#0017	00	コントロール		
#0030	00	下位	4・6	
#0031		上位		
#0032		下位		
#0033		上位		
#0034		下位		
#0035	上位			
#0040	—	システムプログラムのバージョンのモニタ (サブバージョン)	—	
#0041	—	システムプログラムのバージョンのモニタ	—	
#0046	00	異常を検知したI/Oのラック、スロットのモニタ (HEX)	4・7	
#0050	00	下位	4・8	
#0051		⋮		
#0052		⋮		
#0053		上位		
#0054	00	ユーザープログラム ソースサムチェックコード	—	
#0055				
#0056				
#0057				
#0060	00	ユーザープログラム ソースサムチェック異常 (異常 = 01 _(H) 、正常 = 00 _(H))	—	
#0064	—	ブートプログラムバージョンのモニタ (サブバージョン)	4・9	
#0065	—	ブートプログラムバージョンのモニタ		
#0066	—	ブートプログラムの機種コード		
#0067	—			
#0150	00			
#0152	00	異常デバイスネットユニット等のスイッチ設定値のモニタ		

JW30Hにて
 { : 同機能
 : 機能変更
 x : なし

システムメモリ 番号 ⁽⁸⁾	初期値 (H)	内 容	詳細 ページ	JW30H
# 0160	00	自己診断結果の異常コード	4・10	
# 0161				
# 0162				
# 0163				
# 0164				
# 0165				
# 0166				
# 0167				
# 0170	00	オプションユニットの異常コード	4・10	
# 0171				
# 0172				
# 0173				
# 0174				
# 0175				
# 0176				
# 0177				
# 0201	00	TMRのリセット条件設定	4・11	
# 0202		CNTのリセット条件設定		
# 0206	00	ヒューズ断検出時、運転継続 / 停止の設定	4・11	
# 0207	00	オプションユニットが異常時、運転継続 / 停止の設定		
# 0211	00	デバイスネットユニット等が異常時、運転継続 / 停止の設定		
# 0213	02	異常履歴格納領域の使用選択		
# 0222	00	PG/COMM 2 ポートの通信方式の選択	4・12	×
# 0223	00	時計機能の選択		
# 0225	00	1msタイマ機能の設定		
# 0226	00	コンスタントスキャン時間の設定	4・13	
# 0227	00	10msタイマ機能の設定		
# 0230	C0	下位 キープリレー領域の設定 (OCT)	4・14	
# 0231	01	上位		
# 0232	00	下位 出力保持アドレスの設定 (OCT)	4・15	
# 0233		上位		
# 0234	00	PG/COMM1ポートの設定	4・16	
# 0235	00			
# 0236	00	PG/COMM2ポートの設定		
# 0237	00			
# 0240	00	割込処理の設定	4・16	
# 0241				
# 0242				
# 0243				
# 0244				
# 0245				
# 0246	0A	瞬停検出時間延長の設定	4・18	
# 0247	00	I/Oアドレスの登録方法の設定		
# 0250	00	下位 キープリレー領域の設定 (拡張エリア)	4・13	
# 0251	30	上位		

システムメモリ 番号 ⁽⁸⁾	初期値 (H)	内 容	詳細 ページ	JW30H
# 0252	00	下位	4・14	
# 0253	30	出力保持アドレスの設定 (拡張エリア) 上位		
# 0255	00	ROM運転モードの設定	4・19	
# 0256	80	ROM格納領域の設定		
# 0263	00	モード変更履歴の実行設定	6・32	×
# 0266	00	EA-PGポートの設定	4・20	
# 0267	00			
# 0270	FF	システムメモリ # 0000 ~ # 2777のレジスタ転送		
# 0271	FF			
# 0272	00			
# 0277	不定	BCCチェックコード		
# 0300 } # 0377	各00	デバイスネットJW-20DN2の設定 (ユニットNo. 0)	4・23	
# 0410 } # 0414	-	シンボル・コメントの使用領域の設定 (ラダー設計支援ソフトJW-300SPを使用時)	4・21	
# 0440	00	JW100H互換の割付	4・22	×
# 0500 } # 0577	各00	デバイスネットJW-20DN2の設定 (ユニットNo. 1)	4・23	
# 1600	00	異常コード	4・25	×
# 1601	30	異常発生回数 異常コード別の発生回数 No. 1		
# 1602	00	異常コード		
# 1603	80	異常発生回数 異常コード別の発生回数 No. 2		
# 1604	各00	異常コード別の発生回数 No. 3		
# 1675		異常コード別の発生回数 No. 31		
# 1676		00		
# 1677	00	異常発生回数 異常コード別の発生回数 No. 32		

システムメモリ 番号 ₍₈₎	初期値 (H)	内 容	詳細 ページ	JW30H
# 2100	00	ロギングデータを格納する領域の先頭アドレス (ファイルアドレス) JW300のシステムプログラムのバージョンがVer1.20以降 の場合、# 2101の初期値は08 _(H) です。	4・26	×
# 2101				
# 2102				
# 2103				
# 2104	00	ロギング格納数		
# 2105	00	タイムスタンプのフォーマット		
# 2106	FF	ロギング指定レジスタ 1 のレジスタアドレス (ファイルアドレス)		
# 2107	FF			
# 2110	FF			
# 2111	FF			
# 2112	FF	ロギング指定レジスタ 2 のレジスタアドレス (ファイルアドレス)		
# 2113	FF			
# 2114	FF			
# 2115	FF			
# 2116	各FF	ロギング指定レジスタ 3 のレジスタアドレス		
# 2151		ロギング指定レジスタ 9 のレジスタアドレス		
# 2152	FF	ロギング指定レジスタ10のレジスタアドレス (ファイルアドレス)		
# 2153	FF			
# 2154	FF			
# 2155	FF			
# 2156	各00	次のロギングデータの格納カウンタ		
# 2161				
# 2162	各00	ロギングデータの格納回数		
# 2165				
# 2200				
# 2220	00	CFカードの設定		
# 2221	1F			
# 2222	各00			
# 2225				
# 2230	各00			
# 2236				
# 2237	00			
# 2260	800	モード変更履歴の格納設定	6・32	
# 2263				
# 2264	00			
# 2300	各00	デバイスネットJW-20DN2の設定 (ユニットNo. 2)	4・24	
# 2377				
# 2400	各00	デバイスネットJW-20DN2の設定 (ユニットNo. 3)		
# 2477				

4 - 2 システムメモリの内容

システムメモリー一覧(4・1-4ページ)の各項目について、詳細を説明します。

システムメモリ番号(8)	設定項目	内 容																										
#0010	時計機能	秒：00～59 (BCD)																										
#0011		分：00～59 (BCD)																										
#0012		時：00～23 (BCD)																										
#0013		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。																										
#0014		月：01～12 (BCD)																										
#0015		年：00～99 (BCD) 西暦の下2桁を示します。「うるう年」の自動判別は4年ごとの判別のみです。 ・04年、08年、12年は「うるう年」と判別します。																										
#0016		曜日：00～06 (BCD) 時刻を設定時に曜日を合わせてください。日付が変わるごとに順次、00～06に変化します。曜日は、年月日設定による計算は行いません。 <table border="1" style="margin-left: 20px;"> <tr> <td>曜 日</td> <td>日</td> <td>月</td> <td>火</td> <td>水</td> <td>木</td> <td>金</td> <td>土</td> </tr> <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> </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																					
#0017	コントロール：時刻合せ/時刻モニタ、30秒補正、時計停止/時計運転をビット(D7、D3、D0)のON/OFFで行います。 <table border="1" style="margin-left: 20px;"> <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 010₍₈₎を書き込むと、0～29秒は0秒となります。30～59秒は0秒となり、1分の桁上げとなります。 ・時刻合せ 001₍₈₎を書き込んで時計を停止して、#0010～#0016に時刻を入力し、200₍₈₎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。 (注)001₍₈₎を書き込むとシステムメモリ#0010～#0016、レジスタ099770～099776の表示は更新しませんが、内部での時計は動作しています。(見かけ上、時計停止) 	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#0223が00_(H)のとき、レジスタ099770～099777を使用して、JW300のプログラムでも時刻をセットできます。
- ・時計機能は、コントロールユニットの電池でバックアップしています。

#0046 異常を検知した I/O のラック、スロット番号のモニタ

JW300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。(異常内容によっては、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)

異常コード(BCD)

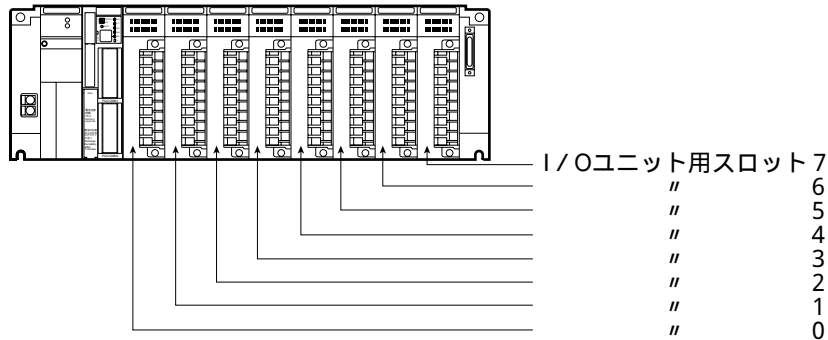
I/Oデータバスエラー.....	44
出力データチェックエラー.....	42
実装ユニットエラー.....	40
I/Oベースエラー.....	48
テーブル照合エラー.....	60
ユニットNo. スイッチ照合エラー.....	61
テーブル登録エラー.....	70
ユニットなしエラー.....	71
I/O点数オーバーエラー.....	72
ユニットNo. スイッチ設定エラー.....	73

ラック番号とスロット番号のビット配置

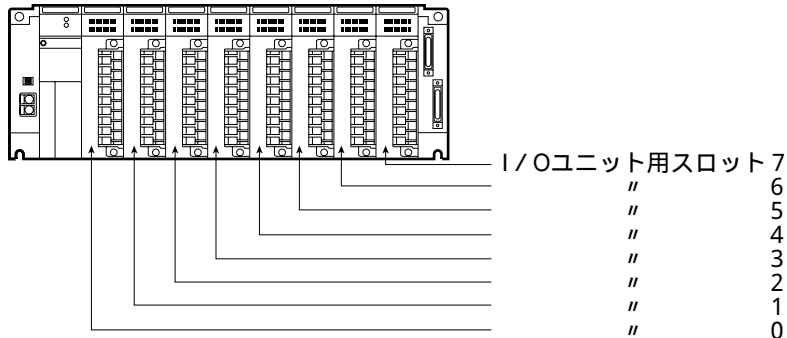
ビット	6	5	4	3	2	1	0	
#0046	0	4	2	1	0	4	2	1
	ラック番号 (0~7)				スロット番号 (0~7)			

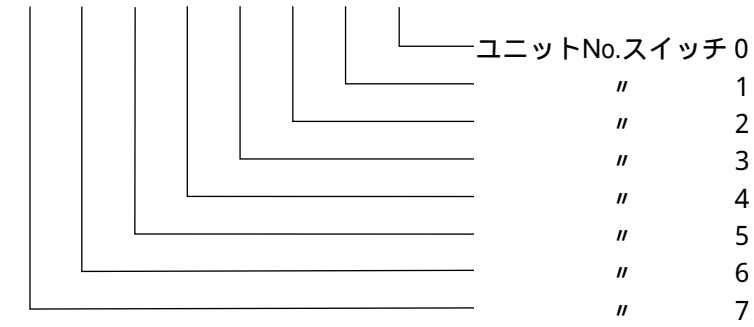
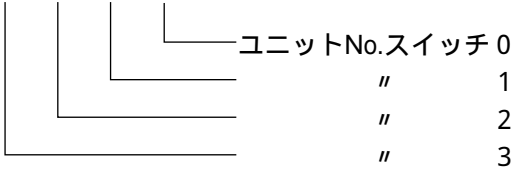
- ラック番号は基本ベースユニットが「0」、増設ベースユニットはラック番号スイッチの設定値「1~7」になります。
I/Oバス拡張アダプタを使用時は、増設ベースユニット(JW-34ZB/36ZB/38ZB)とI/Oバス拡張アダプタ(JW-32EA)のラック番号スイッチの組合せでラック番号が決まります。
- スロット番号は、基本/増設ベースユニットのI/Oユニット用スロットの番号「0~7」を格納します。

【例】 基本ベースユニット：JW-318KB



増設ベースユニット：JW-38ZB



<p># 0150</p>	<p>異常オプションユニットのスイッチ設定値のモニタ</p>	<p>自己診断の結果、システムメモリ#0160に異常コード53(オプション異常)を格納しているとき、#0150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ユニットNo.スイッチ設定値を確認できます。ユニットNo.スイッチ番号は、ビットパターンで示します。</p> <p>..... 正常(0 : OFF) 異常(1 : ON)</p> <p>ビット 7 6 5 4 3 2 1 0</p> <p>#0150 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/></p>  <p>【例】ユニットNo.スイッチ設定値2のオプションユニットが異常のとき</p> <p>#0150 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/></p> <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8台を実装できます。 ・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。 ・異常が解消してもクリアされませんので、クリアする必要があるときは、サポートツールで0(OFF)を書き込んでください。
<p># 0152</p>	<p>異常デバイスネットユニット、異常I/Oリンクユニットのスイッチ設定値のモニタ</p>	<p>自己診断の結果、システムメモリ#0160に異常コード53(オプション異常)を格納しているとき、#0152をモニタすると、基本ベースユニットに実装している「異常デバイスネットマスターユニット(JW-20DN2)または異常I/Oリンク親局ユニット(JW-23LMH)」のユニットNo.スイッチ設定値を確認できます。</p> <p>ユニットNo.スイッチ設定値は、ビットパターンで示します。</p> <p>..... 正常(0 : OFF) 異常(1 : ON)</p> <p>ビット 7 6 5 4 3 2 1</p> <p>#0152 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/></p>  <p>【例】ユニットNo.スイッチ設定値2の「JW-20DN2またはJW-23LMH」が異常のとき</p> <p>#0152 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/></p> <ul style="list-style-type: none"> ・JW-20DN2とJW-23LMHIは、基本ベースユニットに合計で最大4台を実装できます。 ・増設ベースユニットには、JW-20DN2とJW-23LMHを実装できません。 ・異常が解消してもクリアされませんので、クリアする必要があるときは、サポートツールで0(OFF)を書き込んでください。

#0160 ~ #0167	自己診断結果 の異常コード	<p>コントロール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"> ・#0160~#0167はシフトレジスタとして働き、8回の異常発生(異常コード)を記憶できます。 ・異常発生の時刻を含めた情報はE7600~E7777に格納します。 ・異常が解消してもクリアされませんので、クリアする必要があるときは、サポートツールで00(H)を書き込んでください。
#0170 ~ #0177	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、異常内容に応じた異常コードを格納します。</p> <ul style="list-style-type: none"> ・#0170~#0177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オプションユニット(JW-21CM等)の各ユーザーズマニュアルを参照願います。 ・異常発生の時刻を含めた情報は、E5600~E7577にオプションユニットのユニットNo.スイッチ(0~7)ごとに格納します。 ・すべてのオプションユニットが#0170~#0177を共用します。 ・異常が解消してもクリアされませんので、クリアする必要があるときは、サポートツールで00(H)を書き込んでください。
#0201	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>00(H)... 復電時リセット(初期設定) 01(H)... 停電時の状態記憶</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。
#0202	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>00(H)... ONでリセット(初期設定) 01(H)... 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-60 (F / B SFR) F-160(NSFR) F-60w(F / B SFR) Fc160(NSFR) F-60d(F / B SFR) F-261(RCNT) F-62 (U / D C) Fc261(RCNT) F-62w(U / D C) F-62d(U / D C)
#0206	ヒューズ断を検出 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、JW300の運転継続または停止を設定します。</p> <p>00(H)..... 運転継続(初期設定) 08(H)..... 停止</p>

# 0207	オプションユニットが異常時、 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、JW300の運転継続また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. スイッチ設定値(0~7)のビットをONすると運転継続になります。</p> <p>ビット 7 6 5 4 3 2 1 0 #0207 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 停止(0:OFF) 運転継続(1:ON)</p> <p>7 6 5 4 3 2 1 0 オプションユニット のユニットNo. スイッチ設定値 初期設定は00_H)で、全ビット停止状態)です。</p>												
# 0211	デバイスネット ユニット、I/O リンクユニット が異常時、 運転継続 / 停止 の設定	<p>デバイスネットマスターユニット(JW-20DN2)またはI/Oリンク親局ユニット(JW-23LMH)で、ユニット異常(異常コード53)を検出したとき、JW300の運転継続または停止を設定します。 設定は、JW-20DN2またはJW-23LMHのユニットNo. スイッチ設定値のビットをONすると運転継続になります。</p> <p>ビット 7 6 5 4 3 2 1 0 #0211 <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 停止(0:OFF) 運転継続(1:ON)</p> <p>3 2 1 0 JW-20DN2またはJW-23LMH のユニットNo. スイッチ設定値 初期設定は00_H)で、全ビット停止状態)です。</p>												
# 0213	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。 00_H)... 非使用 02_H)... レジスタE5600~E7777₈(初期設定)</p>												
# 0222	PG/COMM2 ポート の通信方式 の選択	<p>コミュニケーションポート2(PG/COMM2ポート)の通信方式を選択します。</p> <table border="1" data-bbox="635 1265 1460 1415"> <thead> <tr> <th>設定値</th> <th>通信方式</th> <th>使用例</th> </tr> </thead> <tbody> <tr> <td>00_H)</td> <td>RTS信号は、データ送信中OFF、</td> <td>RS-232CでRTS信号のフロー制御時</td> </tr> <tr> <td>02_H)</td> <td>データ送信中以外ONします。</td> <td></td> </tr> <tr> <td>08_H)</td> <td>RTS信号は、常にOFFです。</td> <td>RS-422A接続時</td> </tr> </tbody> </table> <p>・上記設定は、JW300コントロールユニットのソフトウェアバージョンがVer.1.17以上で可能です。 初期設定は00_H)です。</p>	設定値	通信方式	使用例	00 _H)	RTS信号は、データ送信中OFF、	RS-232CでRTS信号のフロー制御時	02 _H)	データ送信中以外ONします。		08 _H)	RTS信号は、常にOFFです。	RS-422A接続時
設定値	通信方式	使用例												
00 _H)	RTS信号は、データ送信中OFF、	RS-232CでRTS信号のフロー制御時												
02 _H)	データ送信中以外ONします。													
08 _H)	RTS信号は、常にOFFです。	RS-422A接続時												
# 0223	時計機能の選択	<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。「レジスタ使用」に設定すると、JW300のプログラムで時刻合せを行えます。 00_H)... レジスタ使用(初期設定) 01_H)... 非使用 レジスタは099770~099777を使用します。 2・10ページ参照</p>												

#0225 1msタイマ機能の選択

ビットのON / OFFで、TMR01770 ~ 01777の各領域を1ms/100msタイマに設定できます。

ビット	7	6	5	4	3	2	1	0
#0225	0	0	0	0	0	0	0	

初期値 : 00_(H)

TMR01770 ~ 01777
 (1(ON)...1msタイマ)
 (α(OFF)...100msタイマ)

・DTMR、UTMRでは100msタイマとしてのみ動作します。

#0226 コンスタントスキャン

JW300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。

#0226の設定値	JW300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

初期値は00(BCD)です。

- ・スキャンタイムの設定よりプログラム演算が長いときは、プログラム演算時間でJW300スキャンタイムが決まります。
- ・JW300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。

・設定時間n精度 = 設定値 ± 1 msです。

#0227 10msタイマ機能の選択

ビットのON / OFFで、TMR00000 ~ 00777の各領域を10ms/100msタイマに設定できます。

ビット	7	6	5	4	3	2	1	0
#0227								

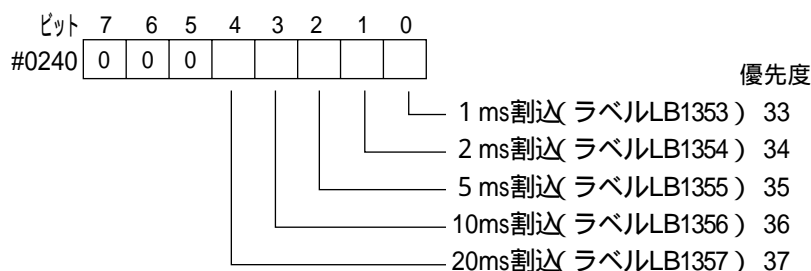
初期値 : 00_(H)

TMR00000 ~ 00077
 TMR00100 ~ 00177
 TMR00200 ~ 00277
 TMR00300 ~ 00377
 TMR00400 ~ 00477 (1(ON)...10msタイマ)
 TMR00500 ~ 00577 (α(OFF)...100msタイマ)
 TMR00600 ~ 00677
 TMR00700 ~ 00777

・DTMR、UTMRでは100msタイマとしてのみ動作します。

# 0230 # 0231	キープリレー領域の設定	<p>キープリレー領域を初期状態(下記)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。</p> <table border="1" data-bbox="651 280 1430 499"> <thead> <tr> <th></th> <th>設定範囲 (ワード、8進)</th> <th>バイトアドレス</th> <th>リレー番号</th> </tr> </thead> <tbody> <tr> <td>#0230、#0231</td> <td>00000000 ~ 00001577</td> <td>コ00000 ~ コ01577</td> <td>000000 ~ 015777</td> </tr> <tr> <td>#0250、#0251</td> <td>00030000 ~ 00035577</td> <td>コ02000 ~ コ07577</td> <td>020000 ~ 075777</td> </tr> </tbody> </table> <p>キープリレー領域は、設定したバイトアドレスから最終バイトアドレス(コ01577、コ07577)のリレーとなります。 コ10000 ~ コ54377(リレー番号100000 ~ 543777)は、常にキープリレー領域です。</p> <p>【例】コ00200以降(リレー番号002000 ~ 015777)をキープリレーに設定するとき</p> <div data-bbox="678 763 1430 913"> <p>#0231: 0 0 0 0 0 0 0 0 #0230: 1 0 0 0 0 0 0 0 [#0230 = 200₈] [#0231 = 000₈] ファイルアドレス00000200 (バイトアドレス : コ00200)</p> </div>		設定範囲 (ワード、8進)	バイトアドレス	リレー番号	#0230、#0231	00000000 ~ 00001577	コ00000 ~ コ01577	000000 ~ 015777	#0250、#0251	00030000 ~ 00035577	コ02000 ~ コ07577	020000 ~ 075777
	設定範囲 (ワード、8進)	バイトアドレス	リレー番号											
#0230、#0231	00000000 ~ 00001577	コ00000 ~ コ01577	000000 ~ 015777											
#0250、#0251	00030000 ~ 00035577	コ02000 ~ コ07577	020000 ~ 075777											
# 0250 # 0251	キープリレー領域の設定 (拡張リレーエリア)	<p>初期状態 # 0230、# 0231の初期値 コ00700以降(リレー番号007000 ~ 015777)がキープリレーに設定されてます。</p> <div data-bbox="678 1064 1430 1214"> <p>#0231: 0 0 0 0 0 0 0 1 #0230: 1 1 0 0 0 0 0 0 (初期値) [#0230 = 300₈] [#0231 = 001₈] ファイルアドレス00000700 (バイトアドレス : コ00700)</p> </div> <p># 0250、# 0251の初期値 コ02000以降(リレー番号020000 ~ 075777、100000 ~ 543777)がキープリレーに設定されてます。</p> <div data-bbox="678 1332 1430 1482"> <p>#0251: 0 0 1 1 0 0 0 0 #0250: 0 0 0 0 0 0 0 0 (初期値) [#0250 = 000₈] [#0251 = 060₈] ファイルアドレス00030000 (バイトアドレス : コ02000)</p> </div>												

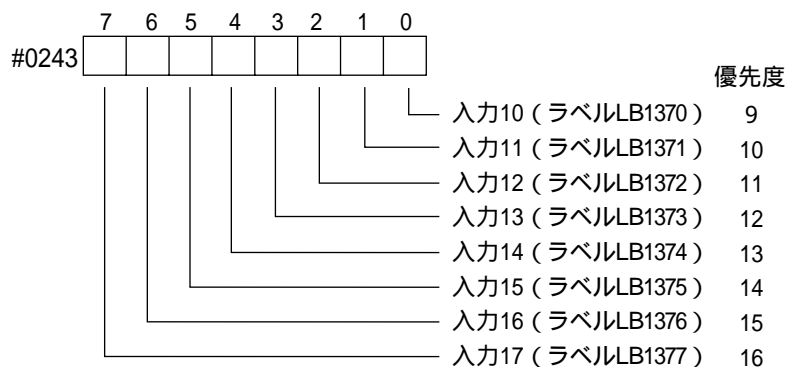
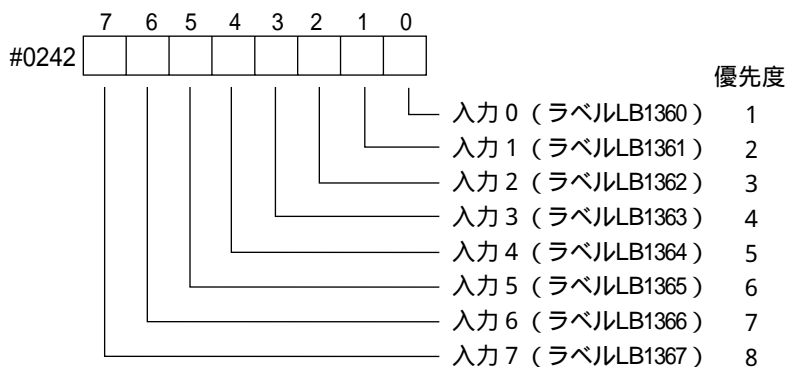
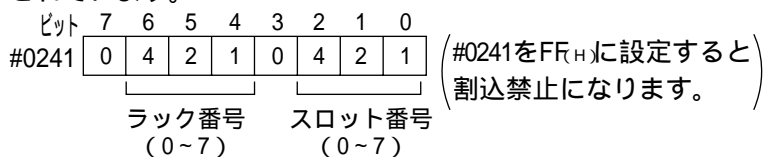
JW300の割込演算条件を設定します。タイマ割込と入力割込が37種類あります。割込は、JW300演算中はもちろん、I/O処理中にも実行します。各割込がかかると指定ラベルのサブルーチンをコールします。
 #0240は各ビットごとに割込時間が異なります。ラベルの番号は割込時のサブルーチンラベルです。5種類の時間を全て使用できます。



(1 : ON)は使用、 (0 : OFF)は非使用です。

#0241 ~ #0245は入力ユニットの入力割込条件を設定します。JW300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。

- ・#0241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8 / 16点ユニットのときは、#0244、#0245で設定の入力20 ~ 37は無効です。なお、64点入力ユニットは割込入力に使用できません。
- ・#0242 ~ #0245は、#0241で指定した入力信号32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。
- ・#0242 ~ #0245で指定する各ビットは、入力ユニットの32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。



は 0 ... 立下り(ON OFF)で割り込みます。
 は 1 ... 立上り(OFF ON)で割り込みます。

#0240、#0242、#0243の初期値は00_H、#241の初期値はFF_Hです。

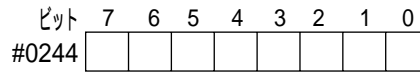
割込処理の設定
 ・#0240 TMR割込み設定
 ・#0241 入力割込みラック・スロット設定
 ・#0242 ~ #0245 入力割込み条件設定

0240
)
 # 0245

0240

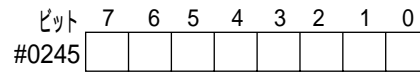
0245

割込処理の設定



優先度

- 入力20 (ラベルLB1400) 17
- 入力21 (ラベルLB1401) 18
- 入力22 (ラベルLB1402) 19
- 入力23 (ラベルLB1403) 20
- 入力24 (ラベルLB1404) 21
- 入力25 (ラベルLB1405) 22
- 入力26 (ラベルLB1406) 23
- 入力27 (ラベルLB1407) 24



優先度

- 入力30 (ラベルLB1410) 25
- 入力31 (ラベルLB1411) 26
- 入力32 (ラベルLB1412) 27
- 入力33 (ラベルLB1413) 28
- 入力34 (ラベルLB1414) 29
- 入力35 (ラベルLB1415) 30
- 入力36 (ラベルLB1416) 31
- 入力37 (ラベルLB1417) 32

は 0 ... 立下り(ON OFF)で割り込みます。
 は 1 ... 立上り(OFF ON)で割り込みます。

#0244、#0245の初期値は00_Hです。

留意点

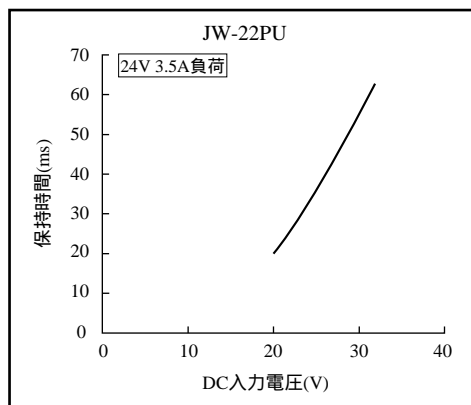
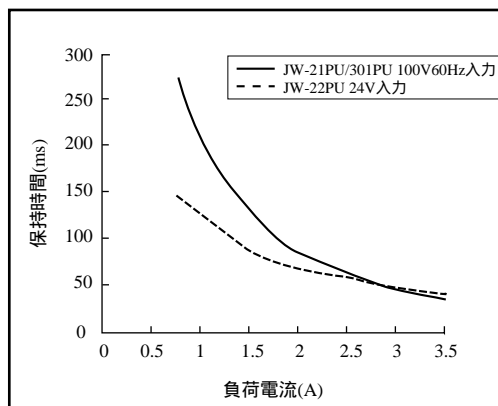
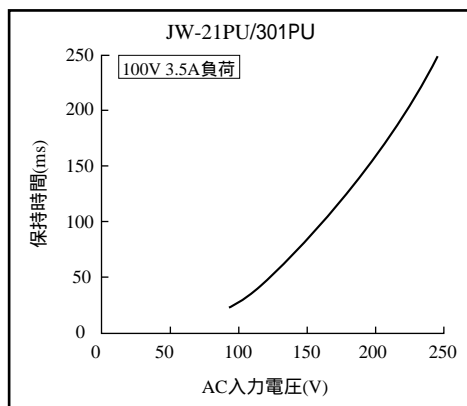
- ・ 割込はF-14(CALL)命令やF-14(CAL+)と同じ使い方をします。注意事項も同じです。
- ・ サブルーチンのラベルはF-4(END)以降に設けてプログラムを作ってください。
- ・ 割込プログラムは、1 ms以内の演算にしてください。1 ms以上だと割込が無視されるときがあります。
- ・ 入力割込のラック、スロット番号に出力ユニットや特殊I/Oユニットを実装しても、割込が働きますのでユニット実装に注意願います。
- ・ リモートI/O子局ユニット(JW-21RS)に実装した入力ユニットの入力割込は行いません。
- ・ タイマ割込、入力割込ではジャンプ先ラベル(F140で指定)が無いとき、割込は無視されます。
- ・ 割込が複数個で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。

入出力ユニット 特殊I/Oユニット	データ交換が終了後に、割込を実行します。
オプションユニット	データ交換中に、割込を実行します。

- ・ 割込を設定中にはブレーク機能は働きません。
- ・ 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。

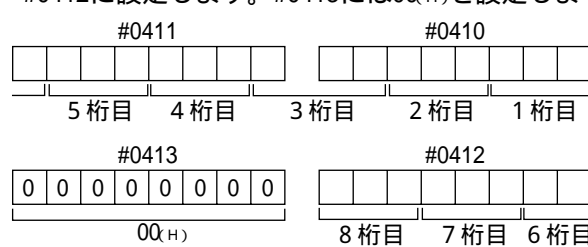
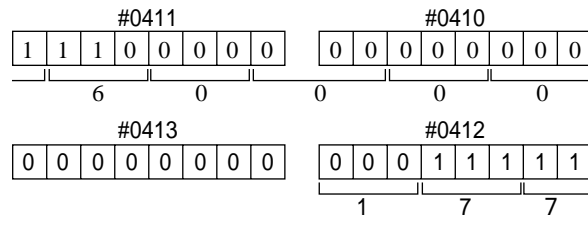

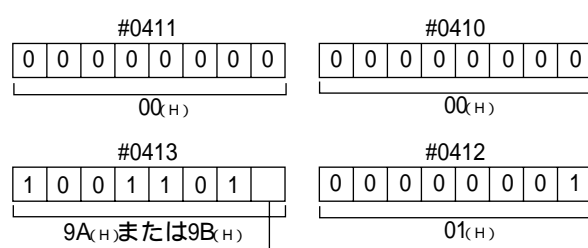
# 0246	瞬停検出時間 延長の設定	瞬停検出時間を0～255msまで可変するときを設定します。設定値は10進数で行います。 初期値は010 _(D) (10ms)です。
--------	-----------------	---

- ・システムメモリ#0246を設定するときには、電源入力電圧およ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、出力保持時間特性を参照して瞬停検出時間を延長してください。
- ・瞬停中、コントロールユニットはすべての動作を停止し、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および増設ベースユニットでの5V容量とのかね合いのため、十分注意して使用願います。



# 0247	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>00_(H) 電源ON時に自動I/O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₍₈₎から連続して割り付けられます。</p> <p>03_(H) 電源ON時の自動I/O登録を禁止 自動I/O登録後に03_(H)を書き込んで、自動I/O登録を禁止にして運転を行ってください。</p> <p>04_(H) ラック先頭アドレスを設定可 各ラック(1～7)に、先頭アドレスをサポートツール(JW-300SP)により設定できます。</p>
--------	-------------------------	---

# 0255	ROM運転 モードの設定	ROM運転モードを設定します。																																																																
		<table border="1"> <thead> <tr> <th colspan="2">設定値</th> <th rowspan="2">ROM RAM 転送 (電源ON時)</th> <th rowspan="2">電源ON 転送後の データメモリ</th> <th rowspan="2">電源ON 転送後の モード</th> <th rowspan="2">ツールによる ROM RAM 転送</th> </tr> <tr> <th>8進</th> <th>16進</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>00</td> <td>しない</td> <td>-</td> <td>-</td> <td>可</td> </tr> <tr> <td>021</td> <td>11</td> <td>する</td> <td>保持</td> <td>電源OFF時のモード</td> <td>可</td> </tr> <tr> <td>042</td> <td>22</td> <td>する</td> <td>クリア 1</td> <td>停止</td> <td>可</td> </tr> <tr> <td>104</td> <td>44</td> <td>する</td> <td>クリア 1</td> <td>運転</td> <td>可</td> </tr> </tbody> </table>	設定値		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	8進	16進	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 1	停止	可	104	44	する	クリア 1	運転	可	1 ROM内に格納されているデータは保持されます。																															
設定値		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																																																									
8進	16進																																																																	
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 1	停止	可																																																													
104	44	する	クリア 1	運転	可																																																													
		初期値は00 _(H) です。 詳細は、JW300ユーザーズマニュアル・ハード編の「ROM運転」を参照 願います。																																																																
# 0256	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。																																																																
		<table border="1"> <thead> <tr> <th colspan="2">設定値</th> <th colspan="4">ROM化される内容 [: ROM化される、× : ROM化されない]</th> </tr> <tr> <th>8進</th> <th>16進</th> <th>システムメモリ [#0200 ~ #2777]</th> <th>2 プログラム</th> <th>3 レジスタ</th> <th>4 ファイル レジスタ</th> <th>6 パラメータ</th> </tr> </thead> <tbody> <tr> <td>200</td> <td>80</td> <td></td> <td></td> <td>×</td> <td>×</td> <td></td> </tr> <tr> <td>201</td> <td>81</td> <td></td> <td></td> <td></td> <td>×</td> <td></td> </tr> <tr> <td>202</td> <td>82</td> <td></td> <td></td> <td>×</td> <td>4</td> <td></td> </tr> <tr> <td>203</td> <td>83</td> <td></td> <td>×</td> <td></td> <td>×</td> <td></td> </tr> <tr> <td>204</td> <td>84</td> <td></td> <td>×</td> <td>×</td> <td>5</td> <td></td> </tr> <tr> <td>205</td> <td>85</td> <td></td> <td></td> <td></td> <td>4</td> <td></td> </tr> <tr> <td>206</td> <td>86</td> <td></td> <td>×</td> <td></td> <td>5</td> <td></td> </tr> </tbody> </table>	設定値		ROM化される内容 [: ROM化される、× : ROM化されない]				8進	16進	システムメモリ [#0200 ~ #2777]	2 プログラム	3 レジスタ	4 ファイル レジスタ	6 パラメータ	200	80			×	×		201	81				×		202	82			×	4		203	83		×		×		204	84		×	×	5		205	85				4		206	86		×		5		初期値は80 _(H) です。 【注】シンボル・コメント専用メモリ(ソフトバージョンS2.00以上)はROM化 されません。	
設定値		ROM化される内容 [: ROM化される、× : ROM化されない]																																																																
8進	16進	システムメモリ [#0200 ~ #2777]	2 プログラム	3 レジスタ	4 ファイル レジスタ	6 パラメータ																																																												
200	80			×	×																																																													
201	81				×																																																													
202	82			×	4																																																													
203	83		×		×																																																													
204	84		×	×	5																																																													
205	85				4																																																													
206	86		×		5																																																													
		<p>2プログラム のROM化領域</p> <table border="1"> <thead> <tr> <th>機種</th> <th>容量</th> <th>ROM化領域</th> </tr> </thead> <tbody> <tr> <td>JW-311CU/312CU</td> <td>8 K語</td> <td>009000 ~ 099777</td> </tr> <tr> <td>JW-321CU/322CU</td> <td>16K語</td> <td>109000 ~ 199777</td> </tr> <tr> <td>JW-331CU/332CU</td> <td>32K語</td> <td>209000 ~ 299777</td> </tr> <tr> <td>JW-341CU/342CU</td> <td>64K語</td> <td>309000 ~ 389777</td> </tr> <tr> <td>JW-352CU</td> <td>128K語</td> <td>・E0000 ~ E7777は ROM化されません</td> </tr> <tr> <td>JW-362CU</td> <td>256K語</td> <td></td> </tr> </tbody> </table> <p>3 レジスタのROM化領域</p> <table border="1"> <thead> <tr> <th>機種</th> <th>ROM化領域</th> </tr> </thead> <tbody> <tr> <td>JW-311CU/312CU</td> <td>なし</td> </tr> <tr> <td>JW-321CU/322CU</td> <td>00000000 ~ 00077777₍₈₎ (32K_(H) 1_(t))</td> </tr> <tr> <td>JW-331CU/332CU</td> <td>00000000 ~ 00077777₍₈₎ (32K_(H) 1_(t))</td> </tr> <tr> <td>JW-341CU/342CU</td> <td>00000000 ~ 00777777₍₈₎ (256K_(H) 1_(t))</td> </tr> <tr> <td>JW-352CU</td> <td>00000000 ~ 00777777₍₈₎ (256K_(H) 1_(t))</td> </tr> <tr> <td>JW-362CU</td> <td>00000000 ~ 00777777₍₈₎ (256K_(H) 1_(t))</td> </tr> </tbody> </table> <p>4 ファイルレジスタ のROM化領域</p> <table border="1"> <thead> <tr> <th>機種</th> <th>ROM化領域</th> </tr> </thead> <tbody> <tr> <td>JW-311CU/312CU</td> <td>なし</td> </tr> <tr> <td>JW-321CU/322CU</td> <td>00000000 ~ 00077777₍₈₎ (32K_(H) 1_(t))</td> </tr> <tr> <td>JW-331CU/332CU</td> <td>00000000 ~ 00077777₍₈₎ (32K_(H) 1_(t))</td> </tr> <tr> <td>JW-341CU/342CU</td> <td>00000000 ~ 01777777₍₈₎ (512K_(H) 1_(t))</td> </tr> <tr> <td>JW-352CU</td> <td>00000000 ~ 01777777₍₈₎ (512K_(H) 1_(t))</td> </tr> <tr> <td>JW-362CU</td> <td>00000000 ~ 03777777₍₈₎ (1 M_(H) 1_(t))</td> </tr> </tbody> </table> <p>5 ファイルレジスタ のROM化領域</p> <p>6 特殊I/Oとオプションのパラメータ</p>					機種	容量	ROM化領域	JW-311CU/312CU	8 K語	009000 ~ 099777	JW-321CU/322CU	16K語	109000 ~ 199777	JW-331CU/332CU	32K語	209000 ~ 299777	JW-341CU/342CU	64K語	309000 ~ 389777	JW-352CU	128K語	・E0000 ~ E7777は ROM化されません	JW-362CU	256K語		機種	ROM化領域	JW-311CU/312CU	なし	JW-321CU/322CU	00000000 ~ 00077777 ₍₈₎ (32K _(H) 1 _(t))	JW-331CU/332CU	00000000 ~ 00077777 ₍₈₎ (32K _(H) 1 _(t))	JW-341CU/342CU	00000000 ~ 00777777 ₍₈₎ (256K _(H) 1 _(t))	JW-352CU	00000000 ~ 00777777 ₍₈₎ (256K _(H) 1 _(t))	JW-362CU	00000000 ~ 00777777 ₍₈₎ (256K _(H) 1 _(t))	機種	ROM化領域	JW-311CU/312CU	なし	JW-321CU/322CU	00000000 ~ 00077777 ₍₈₎ (32K _(H) 1 _(t))	JW-331CU/332CU	00000000 ~ 00077777 ₍₈₎ (32K _(H) 1 _(t))	JW-341CU/342CU	00000000 ~ 01777777 ₍₈₎ (512K _(H) 1 _(t))	JW-352CU	00000000 ~ 01777777 ₍₈₎ (512K _(H) 1 _(t))	JW-362CU	00000000 ~ 03777777 ₍₈₎ (1 M _(H) 1 _(t))											
機種	容量	ROM化領域																																																																
JW-311CU/312CU	8 K語	009000 ~ 099777																																																																
JW-321CU/322CU	16K語	109000 ~ 199777																																																																
JW-331CU/332CU	32K語	209000 ~ 299777																																																																
JW-341CU/342CU	64K語	309000 ~ 389777																																																																
JW-352CU	128K語	・E0000 ~ E7777は ROM化されません																																																																
JW-362CU	256K語																																																																	
機種	ROM化領域																																																																	
JW-311CU/312CU	なし																																																																	
JW-321CU/322CU	00000000 ~ 00077777 ₍₈₎ (32K _(H) 1 _(t))																																																																	
JW-331CU/332CU	00000000 ~ 00077777 ₍₈₎ (32K _(H) 1 _(t))																																																																	
JW-341CU/342CU	00000000 ~ 00777777 ₍₈₎ (256K _(H) 1 _(t))																																																																	
JW-352CU	00000000 ~ 00777777 ₍₈₎ (256K _(H) 1 _(t))																																																																	
JW-362CU	00000000 ~ 00777777 ₍₈₎ (256K _(H) 1 _(t))																																																																	
機種	ROM化領域																																																																	
JW-311CU/312CU	なし																																																																	
JW-321CU/322CU	00000000 ~ 00077777 ₍₈₎ (32K _(H) 1 _(t))																																																																	
JW-331CU/332CU	00000000 ~ 00077777 ₍₈₎ (32K _(H) 1 _(t))																																																																	
JW-341CU/342CU	00000000 ~ 01777777 ₍₈₎ (512K _(H) 1 _(t))																																																																	
JW-352CU	00000000 ~ 01777777 ₍₈₎ (512K _(H) 1 _(t))																																																																	
JW-362CU	00000000 ~ 03777777 ₍₈₎ (1 M _(H) 1 _(t))																																																																	

# 0410) # 0413	シンボル・コメントの使用領域(種類)の設定	<p>ラダー設計支援ソフトJW-300SPにて、リレー、タイマ・カウンタ、レジスタ、ファイルレジスタ、F-90(REM)命令のシンボル・コメントをJW300に転送(書込/読出)する条件、およびシンボル・コメントの種類を設定します。</p> <p>転送条件(#0410~#0413)</p> <p>ファイルレジスタに転送する場合 転送する先頭アドレスとして、ファイルレジスタのファイルアドレスを#0410~#0412に設定します。#0413には00(H)を設定します。</p> <div style="text-align: center;">  </div> <p>【例】ファイルレジスタ17560000x(ファイルアドレス17760000x)が先頭アドレスのとき</p> <div style="text-align: center;">  </div> <p>PCカードに転送する場合 転送する先頭アドレスとして、PCカードのアドレスを#0410~#0413に設定します。#0413には80(H)を設定します。</p> <div style="text-align: center;">  </div> <p>シンボル・コメント専用メモリに転送する場合 #0410、#0411に00(H)、#0412に01(H)、#0413にはシンボル・コメントのJW-300SP形式により9A(H)または9B(H)を設定します。</p> <div style="text-align: center;">  </div> <p>【シンボル・コメントのJW-300SP形式】 0: Ver3.0形式の場合 1: Ver3.1形式の場合</p>
	# 0414	シンボル・コメントの種類の設定

<p>#0440</p>	<p>JW100H互換 の割付</p>	<p>サポートツールJW-300SPにより、JW100Hモードにすると、#0440は40_Hに設定されます。 自動登録 / 任意登録は、#0247に設定されます。 4・18ページ参照 【JW100Hモードについて】</p> <ul style="list-style-type: none"> ・ラック0、スロット0の先頭アドレスは、必ず「C00000」に設定してください。 ・オプションユニット、空きスロットは「0点」として扱われます。 ・64点入出力ユニットは、特殊I/Oユニットではなく、64点I/Oとして扱われます。
--------------	-------------------------	---

デバイスネットの設定

<p># 0300) # 0377</p>	<p>デバイスネット ユニットの設定 (ユニットNo. 0)</p>	<p>デバイスネットマスターユニット(JW-20DN2)にて、ユニットNo.スイッチ設定を「0」で使用時に、下記を設定します。</p> <p>マスターモードの場合</p> <ul style="list-style-type: none"> ・#0300～#0303：入出力テーブルの先頭アドレス() ・#0304～#0307：診断テーブルの先頭アドレス() ・#0310～#0313：Explicitメッセージテーブル1の先頭アドレス() ・#0314～#0317：スキャンリストテーブルの先頭アドレス() ・#0321：スキャンリスト編集時のノード割付バイト数 ・#0324、#0325：ISDX(通信監視時間) ・#0326、#0327：EPRX(通信監視時間) ・#0330：JW300が停止時、スレーブへの出力状態 ・#0340～#0343：Explicitメッセージテーブル2の先頭アドレス() <p>スレーブモードの場合</p> <ul style="list-style-type: none"> ・#0360～#0363：入出力テーブルの先頭アドレス ・#0364～#0367：入出力バイト数 ・#0370：通信異常時のスレーブエリアの保持/クリア ・#0371、#0372：マスターへのレスポンス時間
<p># 0500) # 0577</p>	<p>デバイスネット ユニットの設定 (ユニットNo. 1)</p>	<p>デバイスネットマスターユニット(JW-20DN2)にて、ユニットNo.スイッチ設定を「1」で使用時に、下記を設定します。</p> <p>マスターモードの場合</p> <ul style="list-style-type: none"> ・#0500～#0503：入出力テーブルの先頭アドレス() ・#0504～#0507：診断テーブルの先頭アドレス() ・#0510～#0513：Explicitメッセージテーブル1の先頭アドレス() ・#0514～#0517：スキャンリストテーブルの先頭アドレス() ・#0521：スキャンリスト編集時のノード割付バイト数 ・#0524、#0525：ISDX(通信監視時間) ・#0526、#0527：EPRX(通信監視時間) ・#0530：JW300が停止時、スレーブへの出力状態 ・#0540～#0543：Explicitメッセージテーブル2の先頭アドレス() <p>スレーブモードの場合</p> <ul style="list-style-type: none"> ・#0560～#0563：入出力テーブルの先頭アドレス ・#0564～#0567：入出力バイト数 ・#0570：通信異常時のスレーブエリアの保持/クリア ・#0571、#0572：マスターへのレスポンス時間

任意割付のとき

・上記設定の詳細は、JW-20DN2ユーザズマニュアルを参照願います。

# 2300) # 2377	デバイスネット ユニットの設定 (ユニットNo. 2)	デバイスネットマスターユニット(JW-20DN2)にて、ユニットNo.スイッチ設定を「2」で使用時に、下記を設定します。 マスターモードの場合 ・#2300～#2303：入出力テーブルの先頭アドレス() ・#2304～#2307：診断テーブルの先頭アドレス() ・#2310～#2313：Explicitメッセージテーブル1の先頭アドレス() ・#2314～#2317：スキャンリストテーブルの先頭アドレス() ・#2321：スキャンリスト編集時のノード割付バイト数 ・#2324、#2325：ISD(通信監視時間) ・#2326、#2327：EPR(通信監視時間) ・#2330：JW300が停止時、スレーブへの出力状態 ・#2340～#2343：Explicitメッセージテーブル2の先頭アドレス() スレーブモードの場合 ・#2360～#2363：入出力テーブルの先頭アドレス() ・#2364～#2367：入出力バイト数 ・#2370：通信異常時のスレーブエリアの保持/クリア ・#2371、#2372：マスターへのレスポンス時間
# 2400) # 2477	デバイスネット ユニットの設定 (ユニットNo. 3)	デバイスネットマスターユニット(JW-20DN2)にて、ユニットNo.スイッチ設定を「3」で使用時に、下記を設定します。 マスターモードの場合 ・#2400～#2403：入出力テーブルの先頭アドレス() ・#2404～#2407：診断テーブルの先頭アドレス() ・#2410～#2413：Explicitメッセージテーブル1の先頭アドレス() ・#2414～#2417：スキャンリストテーブルの先頭アドレス() ・#2421：スキャンリスト編集時のノード割付バイト数 ・#2424、#2425：ISD(通信監視時間) ・#2426、#2427：EPR(通信監視時間) ・#2430：JW300が停止時、スレーブへの出力状態 ・#2440～#2443：Explicitメッセージテーブル2の先頭アドレス() スレーブモードの場合 ・#2460～#2463：入出力テーブルの先頭アドレス() ・#2464～#2467：入出力バイト数 ・#2470：通信異常時のスレーブエリアの保持/クリア ・#2471、#2472：マスターへのレスポンス時間

任意割付のとき

- ・上記設定の詳細は、JW-20DN2ユーザズマニュアルを参照願います。

異常コード別の発生回数

異常が発生した異常コード(最大32種類)別に、発生回数が格納されます。

# 1600、# 1601	異常コード別の発生回数 No. 1	#1600：異常コード _(H) #1601：異常発生回数(000～377 ₍₈₎)						
# 1602、# 1603	異常コード別の発生回数 No. 2	<table border="1"> <thead> <tr> <th>アドレス</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>n</td> <td>異常コード_(H)</td> </tr> <tr> <td>n+1</td> <td>異常発生回数(000～377₍₈₎)</td> </tr> </tbody> </table>	アドレス	内容	n	異常コード _(H)	n+1	異常発生回数(000～377 ₍₈₎)
アドレス	内容							
n	異常コード _(H)							
n+1	異常発生回数(000～377 ₍₈₎)							
# 1604、# 1605	” No. 3							
# 1606、# 1607	” No. 4							
# 1610、# 1611	” No. 5							
# 1612、# 1613	” No. 6							
# 1614、# 1615	” No. 7							
# 1616、# 1617	” No. 8							
# 1620、# 1621	” No. 9							
# 1622、# 1623	” No.10							
# 1624、# 1625	” No.11							
# 1626、# 1627	” No.12							
# 1630、# 1631	” No.13							
# 1632、# 1633	” No.14							
# 1634、# 1635	” No.15							
# 1636、# 1637	” No.16							
# 1640、# 1641	” No.17							
# 1642、# 1643	” No.18							
# 1644、# 1645	” No.19							
# 1646、# 1647	” No.20							
# 1650、# 1651	” No.21							
# 1652、# 1653	” No.22							
# 1654、# 1655	” No.23							
# 1656、# 1657	” No.24							
# 1660、# 1661	” No.25							
# 1662、# 1663	” No.26							
# 1664、# 1665	” No.27							
# 1666、# 1667	” No.28							
# 1670、# 1671	” No.29							
# 1672、# 1673	” No.30							
# 1674、# 1675	” No.31							
# 1676、# 1677	” No.32							

・ No.1～No.32には、異常が発生した順に異常コードが格納されます。

ロギングの設定

#2100 ~ #2165は、ロギング機能(F-403命令)を使用時に設定します。(設定内容の詳細 6・7ページ参照)

#2100 ~ #2103	ロギングデータ格納領域の先頭アドレス	<p>ロギングデータを格納する領域の先頭アドレスを、ファイルアドレスで 2100 ~ 2103に設定します。SRAMカードによる拡張メモリ(ファイルアドレス20000000000⁽⁶⁾ ~)にも設定可能です。 6・18ページ参照</p> <p>【例】レジスタ109000を設定時</p> <div style="text-align: center;"> <table border="0"> <tr> <td style="text-align: center;">#2101</td> <td style="text-align: center;">#2100</td> </tr> <tr> <td style="text-align: center;">0 0 1 1 1 1 0 0</td> <td style="text-align: center;">0 0 0 0 0 0 0 0</td> </tr> <tr> <td style="text-align: center;">└─ 3 ─┘ └─ 6 ─┘ └─ 0 ─┘</td> <td style="text-align: center;">└─ 0 ─┘ └─ 0 ─┘ └─ 0 ─┘</td> </tr> <tr> <td colspan="2" style="text-align: center;"> </td> </tr> <tr> <td style="text-align: center;">#2103</td> <td style="text-align: center;">#2102</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> <tr> <td style="text-align: center;">└─ 0 ─┘ └─ 0 ─┘ └─ 0 ─┘</td> <td style="text-align: center;">└─ 0 ─┘ └─ 0 ─┘ └─ 0 ─┘</td> </tr> </table> </div> <p>・レジスタ109000のファイルアドレスは0003600₈です。</p> <p>(注) JW300のシステムプログラムのバージョンがVer1.20以上の場合、ロギングデータを格納する先頭アドレスにレジスタ09000未満は設定できません。レジスタ09000未満を設定していた場合、ロギングデータは格納されません。</p>	#2101	#2100	0 0 1 1 1 1 0 0	0 0 0 0 0 0 0 0	└─ 3 ─┘ └─ 6 ─┘ └─ 0 ─┘	└─ 0 ─┘ └─ 0 ─┘ └─ 0 ─┘			#2103	#2102	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	└─ 0 ─┘ └─ 0 ─┘ └─ 0 ─┘	└─ 0 ─┘ └─ 0 ─┘ └─ 0 ─┘
#2101	#2100															
0 0 1 1 1 1 0 0	0 0 0 0 0 0 0 0															
└─ 3 ─┘ └─ 6 ─┘ └─ 0 ─┘	└─ 0 ─┘ └─ 0 ─┘ └─ 0 ─┘															
#2103	#2102															
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0															
└─ 0 ─┘ └─ 0 ─┘ └─ 0 ─┘	└─ 0 ─┘ └─ 0 ─┘ └─ 0 ─┘															
#2104	ロギング格納数	<p>ロギングデータの格納数(設定値×10回)を設定します。</p> <p>【例】設定値が05₁₆のとき、50×10=500回分のロギングデータを格納します。</p>														
#2105	タイムスタンプのフォーマット	<p>ロギングデータに付加するタイムスタンプについて、フォーマットを設定します。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">設定値_(H)</th> <th style="text-align: center;">フォーマット</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">00</td> <td>無し</td> </tr> <tr> <td style="text-align: center;">01</td> <td>年月日時分秒(6バイトをロギングデータに付加)</td> </tr> <tr> <td style="text-align: center;">02</td> <td>時分秒(3バイトをロギングデータに付加)</td> </tr> </tbody> </table>	設定値 _(H)	フォーマット	00	無し	01	年月日時分秒(6バイトをロギングデータに付加)	02	時分秒(3バイトをロギングデータに付加)						
設定値 _(H)	フォーマット															
00	無し															
01	年月日時分秒(6バイトをロギングデータに付加)															
02	時分秒(3バイトをロギングデータに付加)															
#2106 ~ #2111	ロギング指定レジスタ1	<p>ロギングを行うレジスタアドレスを、ファイルアドレスで設定します。設定方法は#2100 ~ #2103と同様です。</p>														
#2112 ~ #2115	ロギング指定レジスタ2	<p>ロギング指定レジスタ2 ~ 10を、ロギング指定レジスタ1(#2106 ~ #2111)と同様に設定します。</p>														
#2116 ~ #2121	ロギング指定レジスタ3															
#2122 ~ #2125	ロギング指定レジスタ4															
#2126 ~ #2131	ロギング指定レジスタ5															
#2132 ~ #2135	ロギング指定レジスタ6															
#2136 ~ #2141	ロギング指定レジスタ7															
#2142 ~ #2145	ロギング指定レジスタ8															
#2146 ~ #2151	ロギング指定レジスタ9															
#2152 ~ #2155	ロギング指定レジスタ10															
#2156 ~ #2161	<p>次のロギングデータを格納するカウンタとなります。</p> <p style="text-align: center;">0 ~ [ロギング格納数(2104) - 1]</p>															
#2162 ~ #2165	<p>ロギングデータを格納した回数となります。</p> <p style="text-align: center;">0 ~ FFFFFFFF_(H)</p>															

CFカードの設定

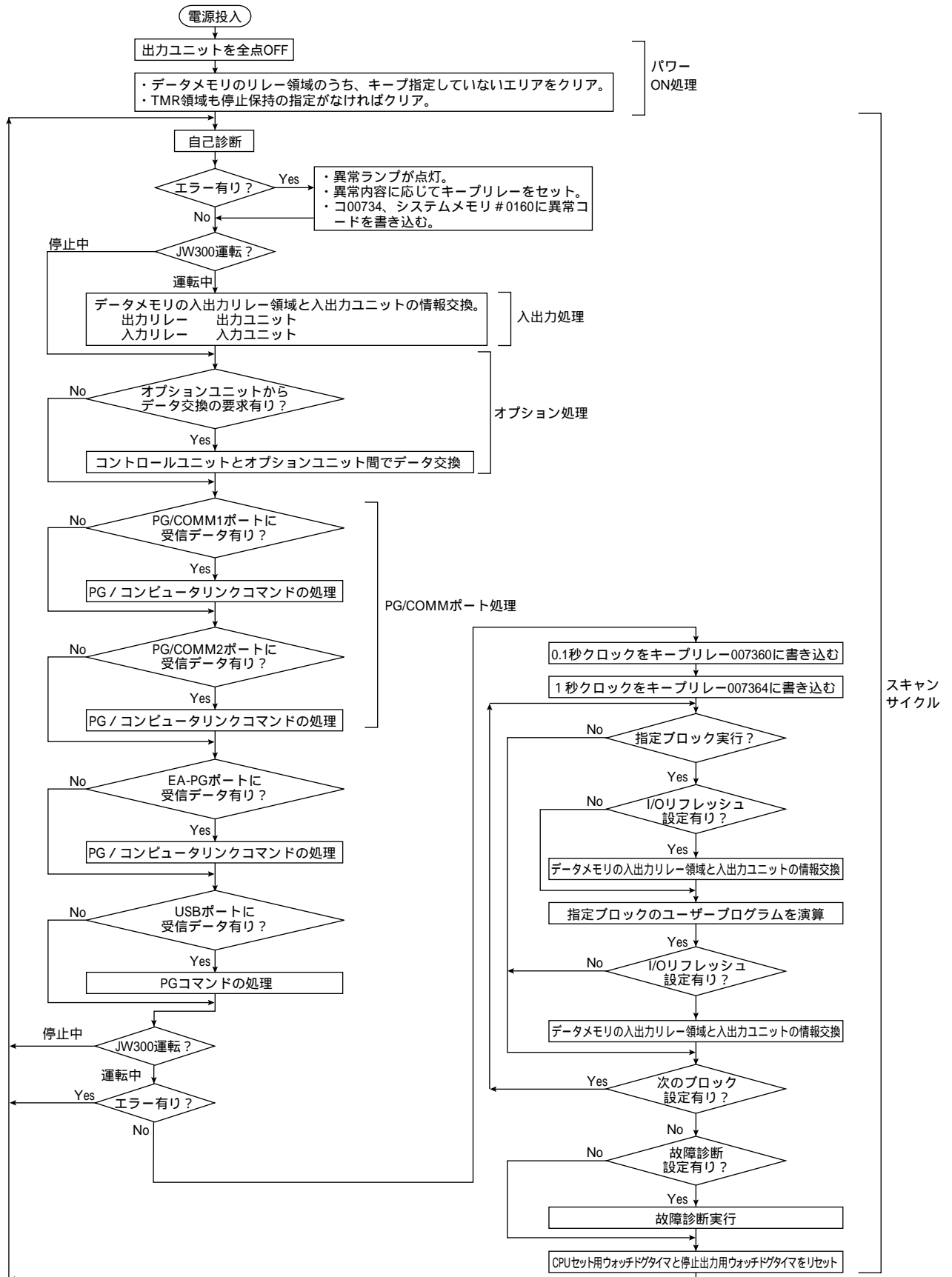
#2220 ~ #2225、#2230 ~ #2237は、コントロールユニットがJW-3*2CUで、CFカードを使用時に設定します。(設定内容の詳細 6・20ページ参照)

# 2220	CFカードへのセーブ(書込)の禁止・許可を設定します。
# 2221	CFカードとセーブ・ロードするファイルの種類を設定します。
# 2222	CFカードとJW-3*2CU間のセーブ・ロードを実行します。
# 2223 ~ # 2225	設定した時間毎に、自動でCFカードへのセーブを設定します。
# 2230 ~ # 2236	特殊リレー(007332)を使用時、またはCFカードを挿入時に、ファイルをセーブする場合のファイル名を設定します。
# 2237	CFカードの保存時にプログラムサムチェック異常が発生した場合のファイル保存を設定します。

第 5 章 コントロールユニットの動作

5-1 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 #0240 ~ #0245 でプログラムを演算できます。

5 - 2 自己診断

JW300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。

JW300ユーザーズマニュアル・ハード編の「トラブルシューティング」を併せて参照

項目	内容	JW300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			異常コード(BCD)					
				FLT (異常)	POWER (電源)	RUN (運転中)	特殊リレー 3	特殊レジスタ コ00734	システムメモリ #0160~#0167						
自 己 診 断	メモリ異常	停止		点 灯		消 灯	007370	20			24				
											23				
											25				
											26				
											28				
	CPU異常	運転		消 灯				—	00			31			
												39			
												32			
												35			
	入 出 力 異 常	リフレッ シュ時	停止	開	点 灯		点 灯					44			
												42			
												40			
												48			
												45			
		テーブル 照合時	停止			点 灯			消 灯	007373	60		60		
													61		
		テーブル 登録時	停止			点 灯					70		70		
													71		
													72		
	特殊I/O異常	運転	閉		消 灯		点 灯	点 灯	点 灯	007375	40	46			
47															
49															
オプション異常	ハードエラー	1	運転	閉	消 灯	点 灯	点 灯	007374	50						
		2	停止	開	点 灯	点 灯	消 灯								
	オプションコマンドエラー	1	運転	閉	消 灯	点 灯	点 灯					007374	50		
		2	停止	開	点 灯	点 灯	消 灯								
		3	運転	閉	消 灯	点 灯	点 灯								
システム保護エラー	1	運転	閉	消 灯	点 灯	点 灯	007374	50							
	2	停止	開	点 灯	点 灯	消 灯									
電源異常	停電/電圧低下	停止	開	消 灯	消 灯	消 灯	007377	10		13	4				
増設電源異常	停電/電圧低下	停止	開	消 灯	消 灯	消 灯	007376	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	007372	20		22					

停止出力 リレー出力、AC100/200V DC30V、1A、JW300運転中はON(閉)

1 システムメモリ#0206、#0207のヒューズ断時またはオプション異常時の設定

5・3-5ページ参照

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

2 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。

#0206を運転停止に設定した場合は特に注意してください。

3 特殊リレー07370~07377は自己診断を検出時にONする特殊リレーです。

異常時に、サポートツール/上位通信/データリンクにより特殊リレーをモニタすることで、異常内容を確認できます。(特殊リレーは異常時にONしますが、異常時はI/O処理も実行しませんので、出力から取り出せません)

4 電源異常は、正常状態でも電源ON時に格納されます。

(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、JW300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

自己診断結果の異常（③前ページ）について、各内容は以下のとおりです。

命令コードチェック

すべての命令の演算を実行時、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#0050～#0053に格納します。

システムメモリ設定チェック

システムメモリ#0200～#0256のサムチェックを行います。

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。

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し、電源投入時にサムチェックを実行してユーザープログラムの内容が変わっていないかを確認します。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)を行って復旧させる必要があります。

I/O登録テーブルチェック

I/O登録時に登録データのサムチェックコードを生成し、電源投入時にサムチェックを実行して登録データの内容が変わっていないかを確認します。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。

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(スキャンタイムが900ms以上)すると、この異常になります。(プログラムモードと同じ状態)

ウォッチドグタイマ2

CPUのウォッチドグタイマがタイムアップ(スキャンタイムが400ms以上、900ms以下)すると、停止出力が「開」となります。このとき演算等は継続されます。

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ことをチェックします。

ハードウェアチェック

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。

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#0046に異常ユニット位置を格納します。

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のときエラーになります。

・特殊I/Oユニットは本チェックを行いません。

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のときエラーになります。

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を確認します。開いているポートがあるとエラーになります。

ユニットバイト数チェック

16 / 32 / 64点の入出力ユニット処理時に、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とき、エラーになります。

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のとき本エラーになります。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時も本エラーになります。

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I/O / オプション / デバイスネット / I/Oリンクユニットの場合、ユニットNo.スイッチの設定も照合します。照合NGのとき本エラーになります。

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とき、本エラーになります。

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」として登録されたとき、本エラーになります。ユニットを全く実装していない場合の状態ではありません。

I/O点数オーバーエラー

I/Oテーブル登録時に、ユニットの実装数が多すぎて、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とき、本エラーになります。

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I/O / 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とき、本エラーになります。

②① 特殊I/Oハードエラー

特殊I/Oユニット自身の異常により、特殊I/Oユニット内蔵のCPUのウォッチドグタイマが働いたとき、本エラーになります。

②② 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I/O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のとき、本エラーになります。

②③ JW-262S(特殊I/O)のヒューズ断

JW-262Sのヒューズが切れたとき、本異常になります。

JW-262Sに外部電源が供給されていないときも、ヒューズ断異常となります。

システムメモリ#0206を「運転停止」に設定時は特に注意願います。

②④ オプションハードエラー

オプション / デバイスネット / I/Oリンク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とき、本エラーになります。

②⑤ 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とき、本エラーになります。

②⑥ 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とき、本エラーになります。

②⑦ オプションバージョンエラー

オプションユニットがJW300に未対応のとき、本エラーになります。

⑳ 電源異常

JW300は10ms以下の瞬時停電のとき、これに応答せず運転を続行します。

これ以上の停電のとき、CPUが停止して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。

㉑ 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、本異常となります。

本異常が発生時、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。

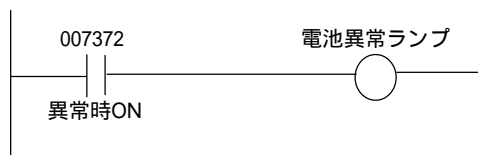
- ・増設電源異常が発生したとき、他の入出力異常が同時に発生し、異常履歴として増設電源異常より優先して格納することがあります。

㉒ 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をチェックします。

特殊リレー007372を使って、電池異常時にランプを点灯させたり、ブザーを鳴らせます。

JW300に電源が投入されている限り、電池異常状態でもJW300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。
システムの異常停止回路にJW300の停止出力を接続すると、JW300が異常時、システムを異常停止
できます。

(注)異常状態がウォッチドグタイム(300ms)以内に復旧すれば、停止出力は「開」になりません。
ウォッチドグタイム(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。

自己診断の結果、異常を検知してJW300が停止した場合、サポートツールにより特殊リレー(007370
~ 007377)を検索して、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JW300は運転を再開して、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。
- ・特殊リレーの中で007372(電池異常)、007374(オプション異常)だけが、JW300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サポートツールで読み出
してください。

なお、特殊リレーの内容はデータリンクで読み出せません。

- ・007377(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

自己診断の結果、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00734)に
異常コードが書き込まれます。

- ・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されます。

(2) システムメモリ

自己診断の結果、異常と判断した場合、システムメモリ(#0160 ~ #0167)にも異常コードが書き込
まれます。

#0160 ~ #0167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。

	#0167	#0166	#0165	#0164	#0163	#0162	#0161	#0160	
クリア状態	00	00	00	00	00	00	00	00	
	00	00	00	00	00	00	00	24	命令コード異常
	00	00	00	00	00	00	24	44	I/Oデータバス異常
24 (消失)	44	**	**	**	**	**	**	22	電池異常

- ・特殊レジスタには代表コードが書き込まれますが、システムメモリには異常内容をさらに分類
した個別コードが書き込まれます。
- ・システムメモリの異常コードは異常回復後もクリアされません。クリアするときは、サポート
ツールでシステムメモリ(#0160 ~ #0167)に00(H)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まれません。

(3) レジスタ

レジスタE5600 ~ E7777には、異常発生時刻を含んだ異常コード内容が格納されます。

(システムメモリ#0213に02(H)を設定時)

〔 5 〕 異常時の出力ユニットのON/OFF状態

自己診断の結果、JW300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#0232、
#0233(000000 ~ 015777)、#0252、#0253(020000 ~ 075777)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

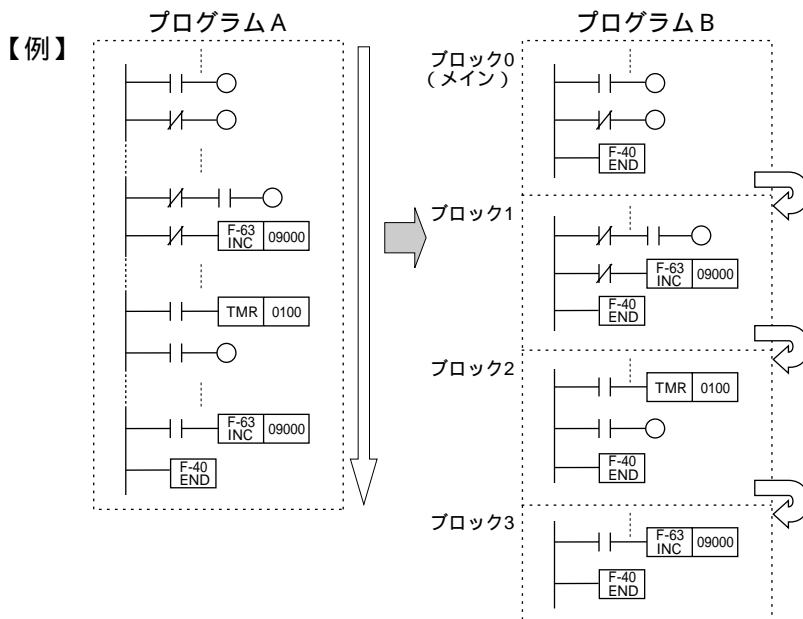
ただし、異常内容によっては出力保持アドレス以前の出力ユニットをOFFできない場合があります。
JW300が異常時にOFFする必要がある出力は、コントロールユニットの停止出力を直列に接続
してください。接続方法に関しては、JW300ユーザーズマニュアル・ハード編を参照願います。

第 6 章 ブロック運転、ロギング、故障診断、 PCカード、セキュリティ、編集履歴、モード変更履歴

6 - 1 ブロック運転機能

JW300には、ユーザープログラムを各処理単位等の処理ブロックに分割(最小0.5K語)してプログラミングできる「ブロック運転機能」があります。

ブロック運転機能を使用すると、プログラムをブロック単位で管理、およびブロック単位で独立したPLCのように扱うことができ、ユーザープログラムの管理が容易になります。また、スキャンタイムの短縮、デバッグ時、テスト運転時などに有効です。



- ・プログラムAを機能毎に、プログラムBのようにブロック分割して、管理を容易にします。また、ブロック毎に起動・待機を行って、テスト運転・デバッグ時などに有効活用します。
- ・通常は必要なブロックだけを常に実行させ、毎スキャンには必要が無いブロックを必要時のみ実行して、スキャンタイムの短縮などを図ります。

ブロック単位で可能な処理

処 理	内 容
起動 / 待機	サポートツール(JW-15PG、JW-300SP)、ブロック起動リレーにより、ブロック毎に起動、待機を操作(設定)できます。 待機状態のブロックは、プログラム演算を実行せず、停止モード状態となります。 (注)ブロックが起動から待機になったとき、ブロック内のコイルとタイマは、すべて保持されます。
I/Oリフレッシュ	ブロック毎に設定するI/Oリフレッシュ動作が可能です。
プログラム管理	RUN中のプログラム変更を含め、ブロック毎に独立したプログラムとして扱えます。

ブロック数、サブプログラム数

ブロック運転で使用可能なブロック数、サブプログラム数は、JW300コントロールユニットの機種により異なります。 構造化プログラム(3・2ページ)参照

	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
ブロック数 (最大)	16	32	64	128	256	512
サブプログラム数 (最大)	256	512	1024	2048	4096	8192

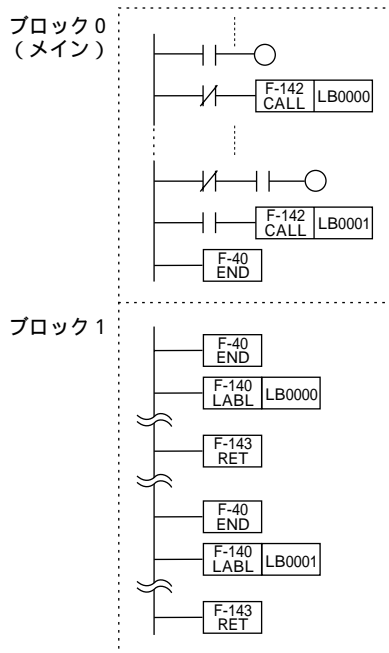
ブロック間のジャンプ

ブロック間でジャンプできます。

次の例では、ブロック0(メイン)を通常ユーザープログラムとして使用し、ブロック1をサブルーチン用のブロックとして使用することにより、ブロックの管理が容易になります。

この場合、ブロック1が待機の状態であっても、ジャンプ先の演算は実行します。

【例】

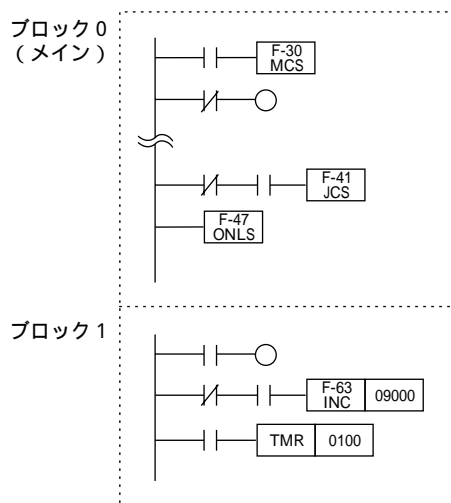


MCS命令、ONLS命令について

MCS(F-30)命令、ONLS(F-47)命令などはリセットしなくても、これらの演算は次のブロックに影響しません。

また、PUSH命令で記憶されたアキュムレータ(ACC)、スタックレジスタ(SR)の内容は、各ブロックの演算開始時にすべて初期化されます。

【例】



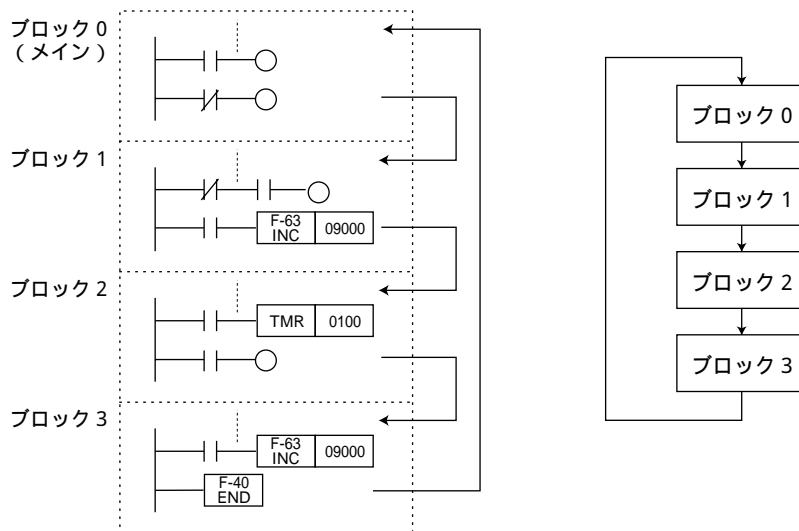
- ・ブロック0でMCS(F-30)命令、ONLS(F-47)命令などをリセットしなくても、ブロック1に影響しません。

〔 1 〕 ブロック状態設定

各ブロックの状態(運転 / 待機)を、サポートツール(JW-15PG、JW-300SP)により設定できます。
装置別のデバッグ等、部分デバッグを行う場合などに有効です。

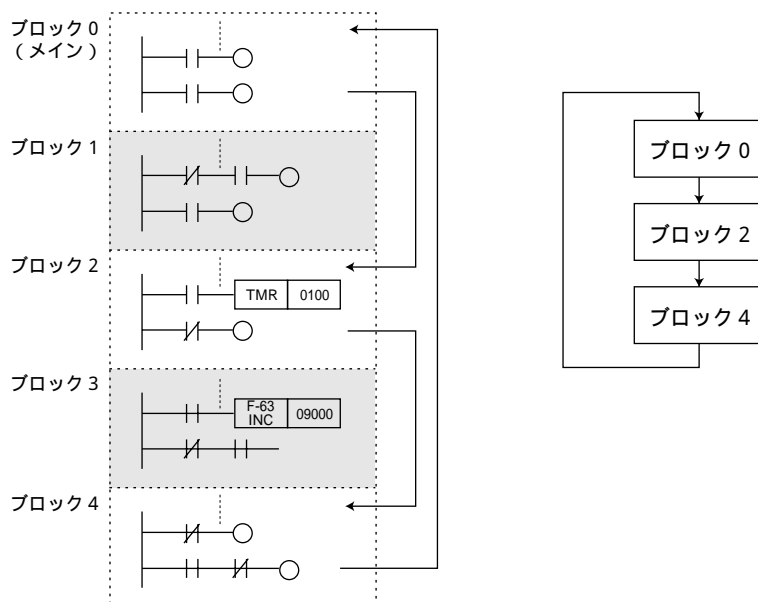
【例 1】

- ・ブロック数 = 4
- ・状態設定 = 全てのブロックを運転



【例 2】

- ・ブロック数 = 5
- ・状態設定
 - ブロック 0、2、4 = 運転
 - ブロック 1、3 = 待機

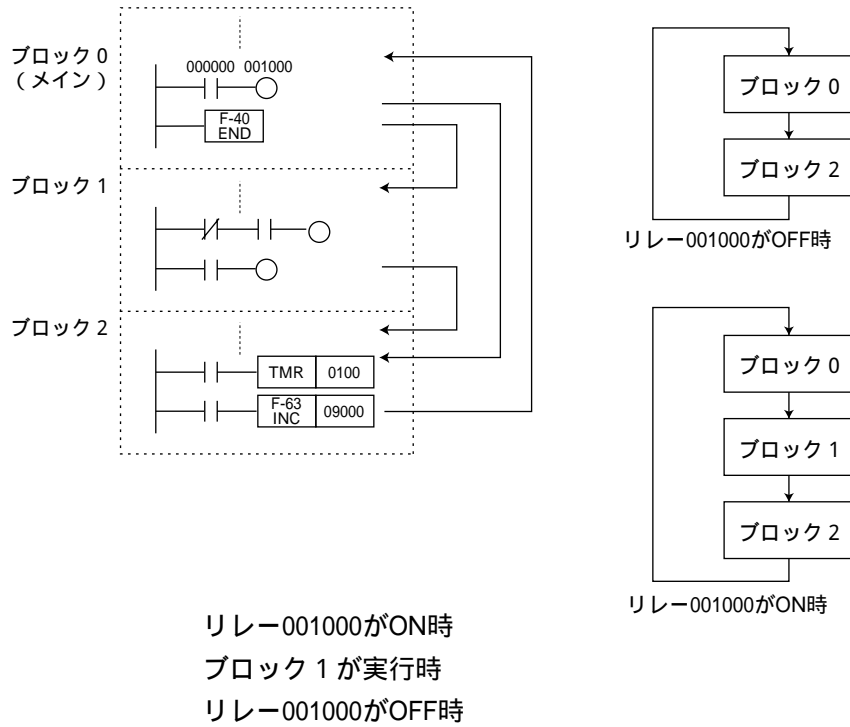


〔 2 〕 ブロック起動リレー

サポートツール(JW-15PG、JW-300SP)で各ブロック(1以降)にブロック起動リレーを設定すると、ブロックの運転(起動)待機(停止)をラダープログラムにより管理します。

【例 1】

- ・ブロック数 = 3
- ・ブロック起動リレーの設定
 - ブロック 1 = リレー 001000 が ON 時に動作
 - ブロック 2 = ブロック起動リレーなし



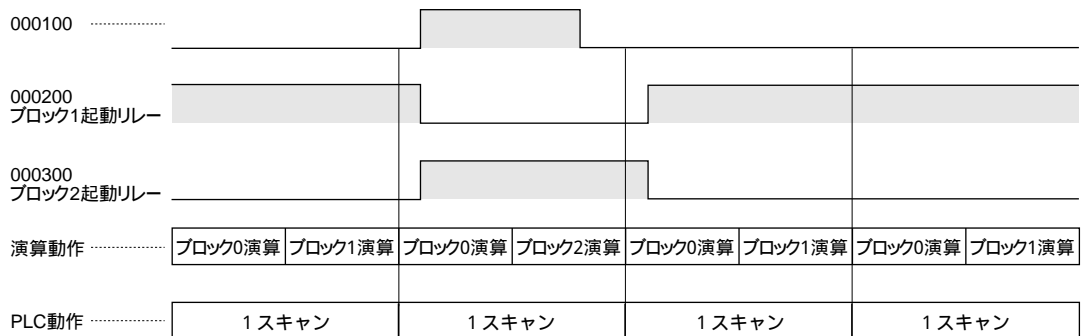
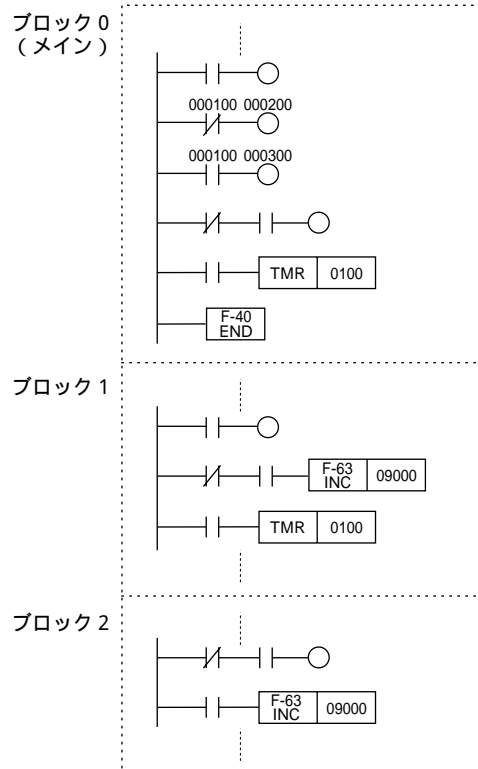
起動リレー 001000							
演算動作	ブロック0演算	ブロック2演算	ブロック0演算	ブロック1演算	ブロック2演算	ブロック0演算	ブロック2演算
PLC動作	1 スキャン		1 スキャン		1 スキャン		

【例 2】

ブロック起動リレーを利用すると、ブロックを分岐できます。

本例では、リレー000200がONのときブロック 1、リレー000300がONのときブロック 2 を実行(起動)します。

- ・ブロック数 = 3
- ・ブロック起動リレーの設定
 - ブロック 1 = リレー 000200 が ON 時に動作
 - ブロック 2 = リレー 000300 が ON 時に動作

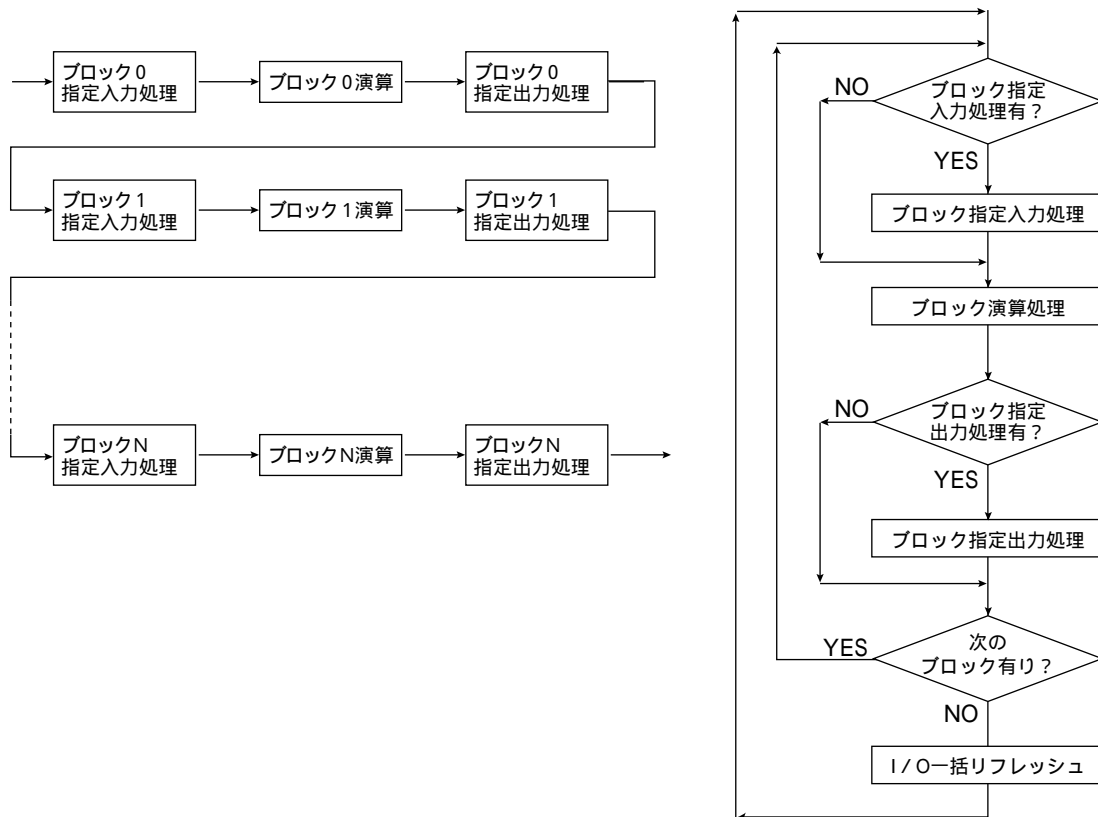


[3] I/Oリフレッシュ

各ブロックで指定のI/Oをリフレッシュ可能です。

ブロック毎にラック・スロット単位のI/O処理を設定でき、各I/Oを高速で処理可能です。

I/Oリフレッシュ処理



【注意事項】

- ・各ブロックでのI/Oリフレッシュが未設定の場合は、未設定ブロックのI/Oリフレッシュ処理は行いません。
- ・ブロックが待機状態の場合は、そのブロックで設定されたI/Oリフレッシュは行いません。
- ・入力割込み(システムメモリ#0240～#0245で設定)で使用している入力ユニットのアドレスは使用しないでください。
- ・スロット番号の上限は使用するベースユニットによって異なります。

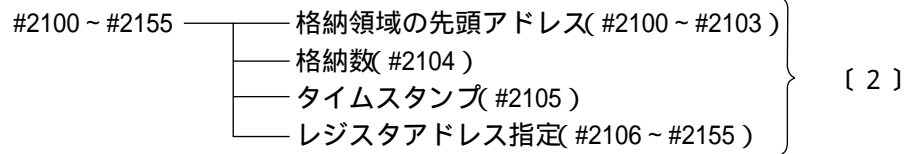
6 - 2 ログ機能

指定するレジスタをトリガ条件として、データメモリをロギング可能です。このロギング機能は、異常発生時の装置の解析、および定刻のデータ収集時に有効活用できます。

ロギング機能を使用するには、応用命令F-403(LOG)のプログラミング、およびシステムメモリ #2100 ~ #2155の設定が必要です。

応用命令F-403(LOG) —— ログ機能のトリガ [1]

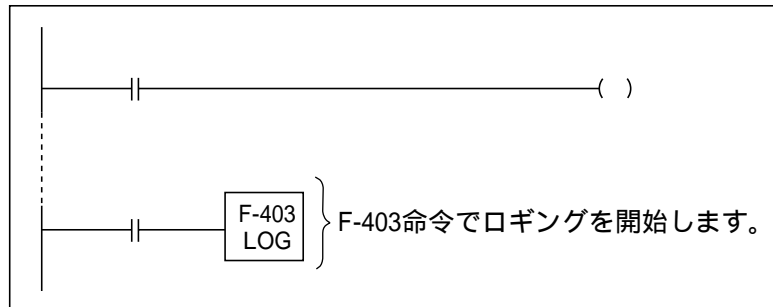
システムメモリ



[1] 応用命令F-403(LOG)のプログラミング

ロギング用応用命令F-403(LOG)のトリガにより、ロギングを開始します。

・ F-403(LOG)の演算条件は、入力信号の立上り(OFF → ON)です。



〔 2 〕 システムメモリ#2100～#2165の設定

ロギングデータについて、格納する領域の先頭アドレス(#2100～#2103)、格納数(#2104)、タイムスタンプ(#2105)、ロギングするレジスタアドレス(#2106～#2155)等を、各システムメモリに設定します。

システムメモリ番号(8)	設定項目	内 容																																
# 2100 ～ # 2103	ロギングデータ格納領域の先頭アドレス	<p>ロギングデータを格納する領域の先頭アドレスを、ファイルアドレスで#2100～#2103に設定します。</p> <p>【例1】レジスタ109000を設定時</p> <div style="text-align: center;"> <table style="margin: auto;"> <tr> <td style="text-align: center;">#2101</td> <td style="text-align: center;">#2100</td> </tr> <tr> <td style="text-align: center;">0 0 1 1 1 1 0 0</td> <td style="text-align: center;">0 0 0 0 0 0 0 0</td> </tr> <tr> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</td> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</td> </tr> <tr> <td style="text-align: center;">3 6 0</td> <td style="text-align: center;">0 0 0 0</td> </tr> </table> <table style="margin: auto;"> <tr> <td style="text-align: center;">#2103</td> <td style="text-align: center;">#2102</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> <tr> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</td> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</td> </tr> <tr> <td style="text-align: center;">0 0 0 0</td> <td style="text-align: center;">0 0 0 0 0 0</td> </tr> </table> </div> <p>・レジスタ109000のファイルアドレスは、00036000₍₈₎です。</p> <p>【例2】ファイルレジスタ00400000₍₈₎を設定時</p> <div style="text-align: center;"> <table style="margin: auto;"> <tr> <td style="text-align: center;">#2101</td> <td style="text-align: center;">#2100</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> <tr> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</td> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</td> </tr> <tr> <td style="text-align: center;">0 0 0 0</td> <td style="text-align: center;">0 0 0 0 0 0</td> </tr> </table> <table style="margin: auto;"> <tr> <td style="text-align: center;">#2103</td> <td style="text-align: center;">#2102</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 1 1</td> </tr> <tr> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</td> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</td> </tr> <tr> <td style="text-align: center;">0 0 0 0</td> <td style="text-align: center;">0 0 0 0 6</td> </tr> </table> </div> <p>・ファイルレジスタ00400000₍₈₎のファイルアドレスは、00600000₍₈₎です。</p>	#2101	#2100	0 0 1 1 1 1 0 0	0 0 0 0 0 0 0 0	└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘	3 6 0	0 0 0 0	#2103	#2102	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘	0 0 0 0	0 0 0 0 0 0	#2101	#2100	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘	0 0 0 0	0 0 0 0 0 0	#2103	#2102	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1 1	└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘	0 0 0 0	0 0 0 0 6
#2101	#2100																																	
0 0 1 1 1 1 0 0	0 0 0 0 0 0 0 0																																	
└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘																																	
3 6 0	0 0 0 0																																	
#2103	#2102																																	
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0																																	
└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘																																	
0 0 0 0	0 0 0 0 0 0																																	
#2101	#2100																																	
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0																																	
└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘																																	
0 0 0 0	0 0 0 0 0 0																																	
#2103	#2102																																	
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1 1																																	
└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘																																	
0 0 0 0	0 0 0 0 6																																	
# 2104	ロギング格納数	<p>ロギングデータの格納数(設定値 × 10回)を設定します。 設定範囲は、001～255_(D)です。</p> <p>【例】設定値が050_(D)のとき、50 × 10 = 500回分のロギングデータを格納します。</p>																																
# 2105	タイムスタンプのフォーマット	<p>ロギングデータに付加するタイムスタンプについて、フォーマットを設定します。</p> <table border="1" style="margin: auto;"> <thead> <tr> <th>設定値(H)</th> <th>フォーマット</th> <th>使用バイト数</th> </tr> </thead> <tbody> <tr> <td>00 (初期値)</td> <td>タイムスタンプ無し</td> <td>0</td> </tr> <tr> <td>01</td> <td>年月日時分秒</td> <td>6</td> </tr> <tr> <td>02</td> <td>時分秒</td> <td>3</td> </tr> </tbody> </table>	設定値(H)	フォーマット	使用バイト数	00 (初期値)	タイムスタンプ無し	0	01	年月日時分秒	6	02	時分秒	3																				
設定値(H)	フォーマット	使用バイト数																																
00 (初期値)	タイムスタンプ無し	0																																
01	年月日時分秒	6																																
02	時分秒	3																																

システムメモリ番号 ⁽⁸⁾	設定項目	内 容																						
# 2106 ~ # 2155	ロギング指定レジスタ	ロギングを行うレジスタアドレスを、ファイルアドレスで、最大10レジスタまで設定できます。																						
		<table border="1"> <thead> <tr> <th></th> <th>システムメモリ番号</th> </tr> </thead> <tbody> <tr> <td>ロギングレジスタ 1</td> <td>#2106 ~ #2111</td> </tr> <tr> <td>ロギングレジスタ 2</td> <td>#2112 ~ #2115</td> </tr> <tr> <td>ロギングレジスタ 3</td> <td>#2116 ~ #2121</td> </tr> <tr> <td>ロギングレジスタ 4</td> <td>#2122 ~ #2125</td> </tr> <tr> <td>ロギングレジスタ 5</td> <td>#2126 ~ #2131</td> </tr> <tr> <td>ロギングレジスタ 6</td> <td>#2132 ~ #2135</td> </tr> <tr> <td>ロギングレジスタ 7</td> <td>#2136 ~ #2141</td> </tr> <tr> <td>ロギングレジスタ 8</td> <td>#2142 ~ #2145</td> </tr> <tr> <td>ロギングレジスタ 9</td> <td>#2146 ~ #2151</td> </tr> <tr> <td>ロギングレジスタ 10</td> <td>#2152 ~ #2155</td> </tr> </tbody> </table>		システムメモリ番号	ロギングレジスタ 1	#2106 ~ #2111	ロギングレジスタ 2	#2112 ~ #2115	ロギングレジスタ 3	#2116 ~ #2121	ロギングレジスタ 4	#2122 ~ #2125	ロギングレジスタ 5	#2126 ~ #2131	ロギングレジスタ 6	#2132 ~ #2135	ロギングレジスタ 7	#2136 ~ #2141	ロギングレジスタ 8	#2142 ~ #2145	ロギングレジスタ 9	#2146 ~ #2151	ロギングレジスタ 10	#2152 ~ #2155
			システムメモリ番号																					
		ロギングレジスタ 1	#2106 ~ #2111																					
		ロギングレジスタ 2	#2112 ~ #2115																					
		ロギングレジスタ 3	#2116 ~ #2121																					
		ロギングレジスタ 4	#2122 ~ #2125																					
		ロギングレジスタ 5	#2126 ~ #2131																					
		ロギングレジスタ 6	#2132 ~ #2135																					
		ロギングレジスタ 7	#2136 ~ #2141																					
ロギングレジスタ 8	#2142 ~ #2145																							
ロギングレジスタ 9	#2146 ~ #2151																							
ロギングレジスタ 10	#2152 ~ #2155																							
<ul style="list-style-type: none"> ロギングレジスタを未設定のシステムメモリ番号は、設定値をFFFFFFFF_(H) (初期値)にしてください。 																								
<p>【例 1】ロギングレジスタ 1 に、レジスタ 009000 を設定時</p> <table style="margin-left: 40px;"> <tr> <td style="text-align: center;">#2107</td> <td style="text-align: center;">#2106</td> </tr> <tr> <td style="text-align: center;">0 0 0 0 1 0 0 0</td> <td style="text-align: center;">0 0 0 0 0 0 0 0</td> </tr> <tr> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</td> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</td> </tr> <tr> <td style="text-align: center;">0 4 0 0 0 0 0 0</td> <td style="text-align: center;">0 0 0 0 0 0 0 0</td> </tr> </table> <table style="margin-left: 40px;"> <tr> <td style="text-align: center;">#2111</td> <td style="text-align: center;">#2110</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> <tr> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</td> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</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>	#2107	#2106	0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0	└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘	0 4 0 0 0 0 0 0	0 0 0 0 0 0 0 0	#2111	#2110	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0								
#2107	#2106																							
0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0																							
└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘																							
0 4 0 0 0 0 0 0	0 0 0 0 0 0 0 0																							
#2111	#2110																							
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0																							
└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘																							
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0																							
<ul style="list-style-type: none"> レジスタ 009000 のファイルアドレスは、0000400₍₈₎ です。 																								
<p>【例 2】ロギングレジスタ 3 に、ファイルレジスタ 0100000₍₈₎ を設定時</p> <table style="margin-left: 40px;"> <tr> <td style="text-align: center;">#2117</td> <td style="text-align: center;">#2116</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> <tr> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</td> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</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> <table style="margin-left: 40px;"> <tr> <td style="text-align: center;">#2121</td> <td style="text-align: center;">#2120</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 1 0 1</td> </tr> <tr> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</td> <td style="text-align: center;">└─┬─┬─┬─┬─┬─┬─┬─┘</td> </tr> <tr> <td style="text-align: center;">0 0 0 0 0 0 0 0</td> <td style="text-align: center;">0 1 2</td> </tr> </table>	#2117	#2116	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	#2121	#2120	0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 1	└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘	0 0 0 0 0 0 0 0	0 1 2								
#2117	#2116																							
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0																							
└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘																							
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0																							
#2121	#2120																							
0 0 0 0 0 0 0 0	0 0 0 0 0 1 0 1																							
└─┬─┬─┬─┬─┬─┬─┬─┘	└─┬─┬─┬─┬─┬─┬─┬─┘																							
0 0 0 0 0 0 0 0	0 1 2																							
<ul style="list-style-type: none"> ファイルレジスタ 0100000₍₈₎ のファイルアドレスは、0120000₍₈₎ です。 																								

システムメモリ番号 ⁽⁸⁾	設定内容
# 2156 ~ # 2161	次のロギングデータを格納するカウンタとなります。 0 ~ [ロギング格納数(#2104) - 1]
# 2162 ~ # 2165	ロギングデータを格納した回数となります。 0 ~ FFFFFFFF _(H)

〔 3 〕 使用例

ロギング機能の使用例を説明します。

(1) システムメモリ(#2100 ~ #2155)の設定例

ロギング機能のシステムメモリ(#2100 ~ #2155)に次の設定を行った場合について、ロギングデータの格納例を示します。

項 目	システムメモリ番号	設定値	内 容
ロギングデータ 格納領域 の先頭アドレス	#2100 ~ #2103	00200000 ₍₈₎	ファイルレジスタ00000000 ₍₈₎ ファイルアドレス00200000 ₍₈₎
ロギング格納数	#2104	20 _(D)	200個 (20 _(D) × 10)
タイムスタンプ	#2105	01 _(H)	年月日時分秒
ロギングレジスタ 1	#2106 ~ #2111	00000000 ₍₈₎	コ00000 ファイルアドレス00000000 ₍₈₎
ロギングレジスタ 2	#2112 ~ #2115	00000001 ₍₈₎	コ00001 ファイルアドレス00000001 ₍₈₎
ロギングレジスタ 3	#2116 ~ #2121	00000002 ₍₈₎	コ00002 ファイルアドレス00000002 ₍₈₎
ロギングレジスタ 4	#2122 ~ #2125	00000003 ₍₈₎	コ00002 ファイルアドレス00000003 ₍₈₎
ロギングレジスタ 5	#2126 ~ #2131	00004000 ₍₈₎	009000 ファイルアドレス00004000 ₍₈₎
ロギングレジスタ 6	#2132 ~ #2135	00004001 ₍₈₎	009001 ファイルアドレス00004001 ₍₈₎
ロギングレジスタ 7	#2136 ~ #2141	00004002 ₍₈₎	009002 ファイルアドレス00004002 ₍₈₎
ロギングレジスタ 8	#2142 ~ #2145	00004003 ₍₈₎	009003 ファイルアドレス00004003 ₍₈₎
ロギングレジスタ 9	#2146 ~ #2151	00016000 ₍₈₎	E0000 ファイルアドレス00016000 ₍₈₎
ロギングレジスタ 10	#2152 ~ #2155	00016001 ₍₈₎	E0001 ファイルアドレス00016001 ₍₈₎

ロギングデータの格納例

ロギングデータは、前ページのシステムメモリ設定の場合、ファイルレジスタ00000000₍₈₎を先頭とする領域に格納されます。

ファイルレジスタ ₍₈₎	データ (例)	内 容
00000000	03	年
00000001	11	月
00000002	20	日
00000003	13	時
00000004	10	分
00000005	35	秒
00000006	02	ロギング指定レジスタ 1 (J00000のデータ)
00000007	05	ロギング指定レジスタ 2 (J00001のデータ)
00000008	0A	ロギング指定レジスタ 3 (J00002のデータ)
00000009	A5	ロギング指定レジスタ 4 (J00003のデータ)
0000000A	F1	ロギング指定レジスタ 5 (009000のデータ)
0000000B	E0	ロギング指定レジスタ 6 (009001のデータ)
0000000C	09	ロギング指定レジスタ 7 (009002のデータ)
0000000D	03	ロギング指定レジスタ 8 (009003のデータ)
0000000E	24	ロギング指定レジスタ 9 (E00000のデータ)
0000000F	57	ロギング指定レジスタ10 (E00001のデータ)
⋮		
00000C70	03	年
00000C71	11	月
00000C72	20	日
00000C73	18	時
00000C74	25	分
00000C75	14	秒
00000C76	02	ロギング指定レジスタ 1 (J00000のデータ)
00000C77	05	ロギング指定レジスタ 2 (J00001のデータ)
00000C78	0A	ロギング指定レジスタ 3 (J00002のデータ)
00000C79	A5	ロギング指定レジスタ 4 (J00003のデータ)
00000C7A	F1	ロギング指定レジスタ 5 (009000のデータ)
00000C7B	E0	ロギング指定レジスタ 6 (009001のデータ)
00000C7C	09	ロギング指定レジスタ 7 (009002のデータ)
00000C7D	03	ロギング指定レジスタ 8 (009003のデータ)
00000C7E	24	ロギング指定レジスタ 9 (E00000のデータ)
00000C7F	57	ロギング指定レジスタ10 (E00001のデータ)

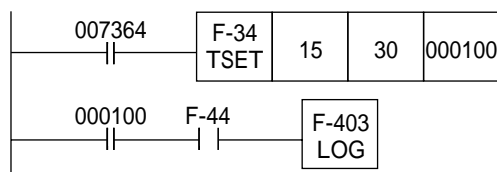
1回目のロギングデータ

200回目のロギングデータ

・201回目以降は、1回目のデータエリアから上書きされます。

(2) 応用命令F-403(LOG)のプログラム例

ロギング用応用命令F-403(LOG)を次のようにプログラミングした場合には、15時30分に指定レジスタのデータが1回分、ロギングデータとして格納されます。



留 意 点

- ・ロギング格納数(システムメモリ#2104)を000_D)に設定した場合でも、1回だけロギングを実行します。
- ・ロギング指定レジスタ(システムメモリ#2106～#2155)に設定範囲外のアドレスがある場合、そのアドレスのデータは格納されず、設定範囲内のアドレスのデータのみ格納されます。
- ・ロギング格納数を越えると、1回目のロギングデータから上書きされます。
さらに格納回数を越えると、繰り返し、1回目のロギングデータから上書きされます。
- ・PCカードで拡張RAMを使用する場合、ロギングデータ格納領域の最終アドレスは、ファイルアドレス2007777777₈)まで設定可能です。
- ・JW300運転中にロギング用システムメモリ(#2100～#2155)を変更した場合、応用命令F-403(LOG)を実行時の設定値が有効となります。

6 - 3 故障診断機能

各サイクルの実行時間の監視による、リレーの故障診断を行えます。従来のユーザープログラムによる監視を、この故障診断機能により監視できます。故障診断機能を使用すると、ユーザープログラムを削減でき、また、異常なリレーや状態をすばやく取得できますので、メンテナンス性の向上にも有効です。

故障診断の「実行 / 非実行」は、システムメモリ#2200により設定します。

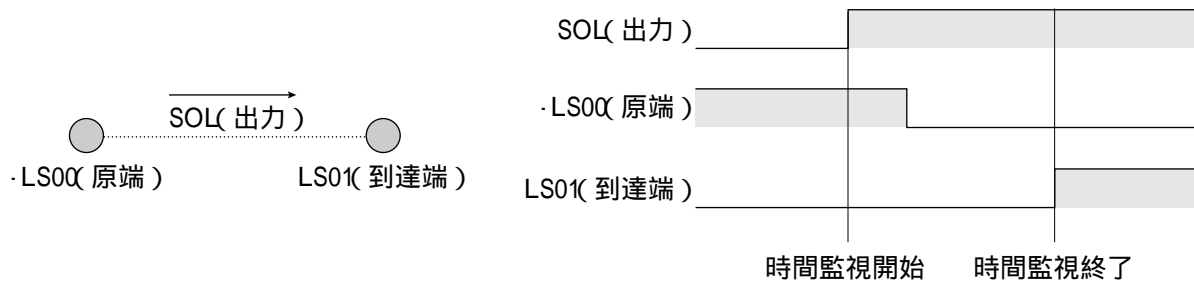
アドレス	データ	内容
#2200	00(H)	初期値
	01(H)	実行
	上記以外	非実行

各機種(コントロールユニット)の設定可能サイクル数は、次のとおりです。

	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
入出力サイクル数	128	256	512	1024	2048	4096
常時監視リレー数	128	256	512	1024	2048	4096
監視時間	0 ~ 32767ms					

〔 1 〕 故障診断の原理

（ 1 ） 入出力サイクル



原端 (LS00) から到達端 (LS01) へ移動する場合、出力 (SOL) がONになってから到達端がONになるまでの間、原端 (LS00) と到達端 (LS01) の状態を監視し、次の項目の異常を検出できます。

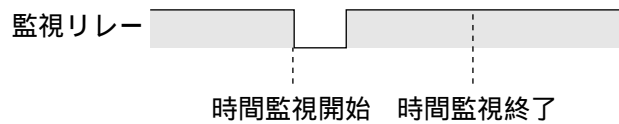
監視出力	マシン状態
到達端ONせず	原端がOFFで、到達端がOFFからONに変化しない場合
出力動作せず	原端がONからOFFに、到達端がOFFからONに変化しない場合
原端OFFせず	到達端がONで、原端がONからOFFに変化しない場合
原端ON	前回の出力の原端がONしている場合
到達端OFF	前回の出力到達端がOFFしている場合

（ 2 ） 常時監視リレー

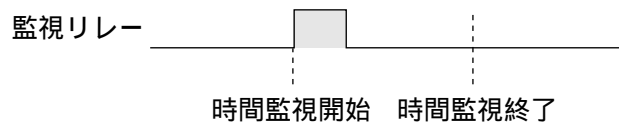
指定するリレーを、常時監視する機能です。

ONであることを監視されているリレーが設定時間内でOFFになったこと、および、OFFであることを監視されているリレーが設定時間内でONになったことを検出できます。

- ・ ONであることを監視



- ・ OFFであることを監視

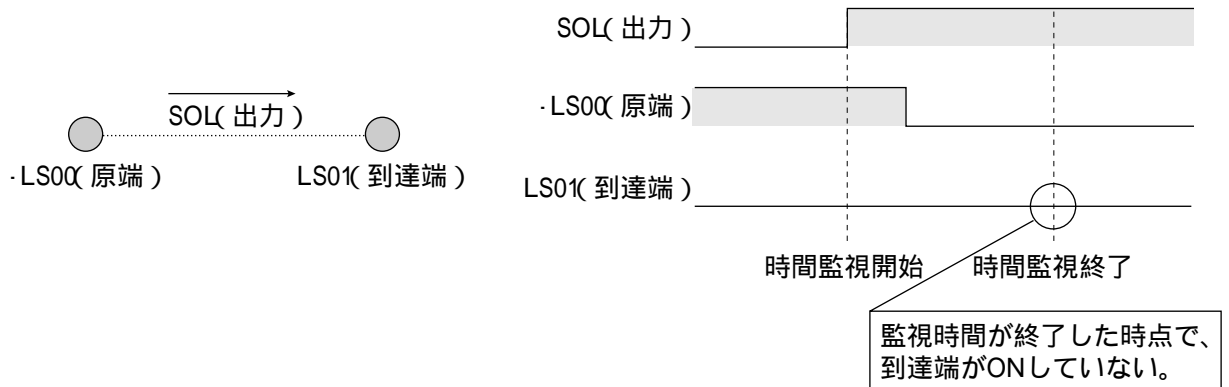


〔 2 〕 故障診断の異常内容

（ 1 ） 入出力サイクルの異常

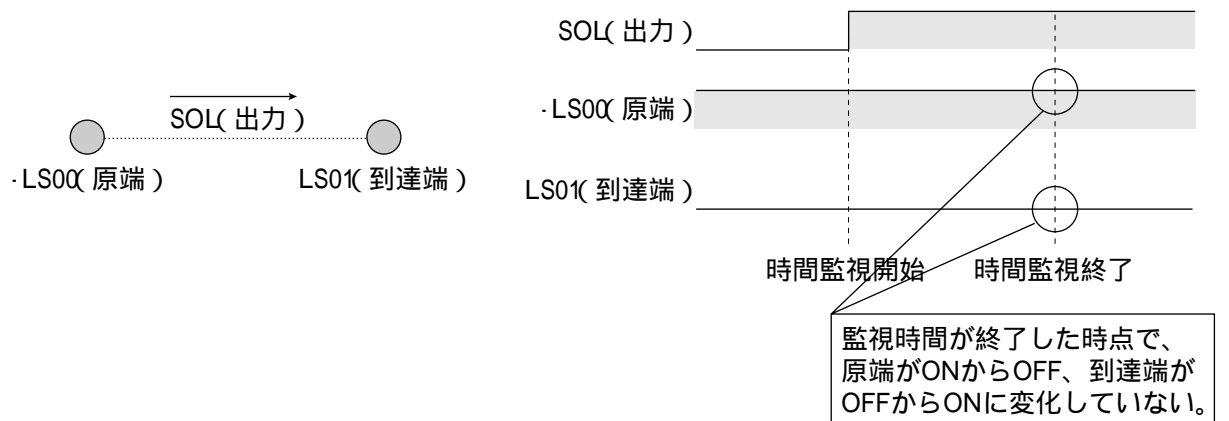
到達端 ON せず

原端が OFF で、到達端が OFF から ON に変化しない場合



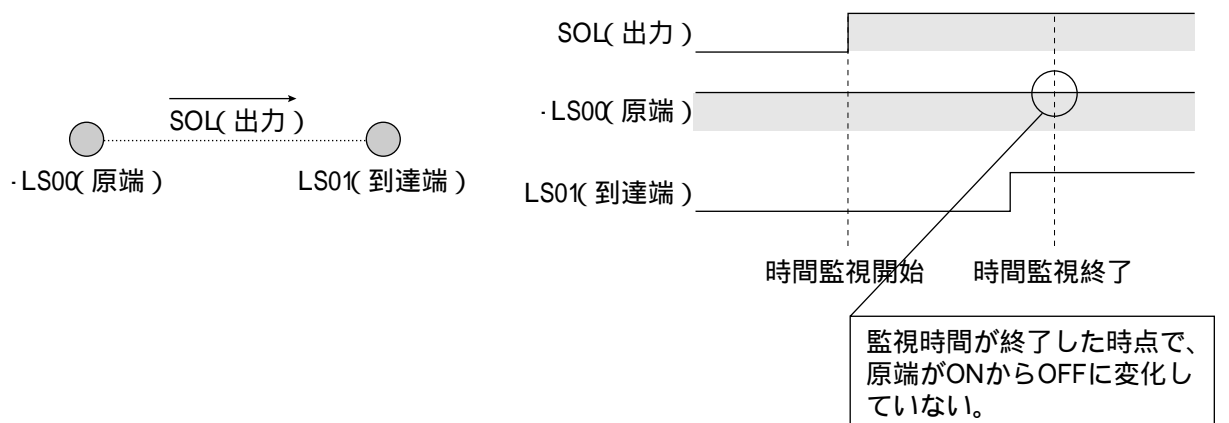
出力動作せず

原端が ON から OFF に、到達端が OFF から ON に変化しない場合



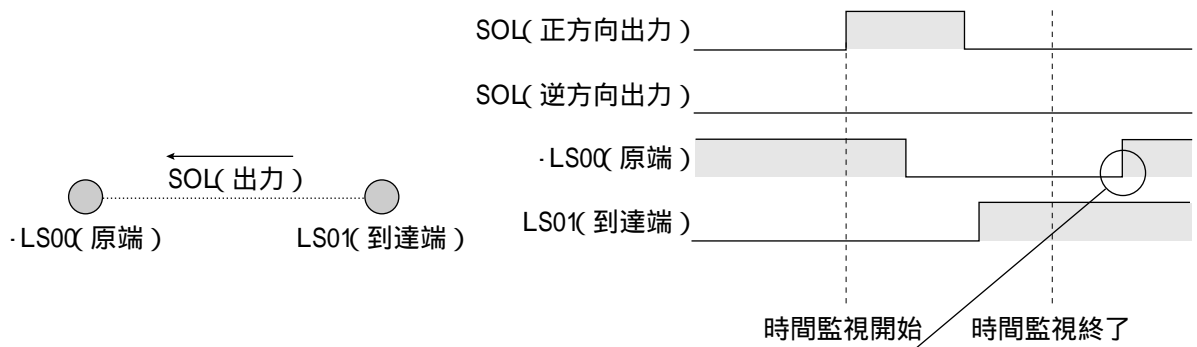
原端 OFF せず

到達端が ON で、原端が ON から OFF に変化しない場合



原端 ON

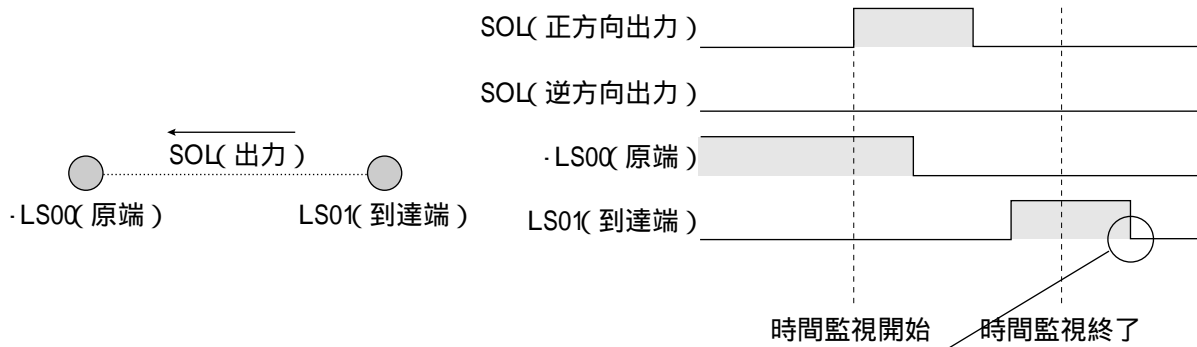
前回の出力の原端が ON している場合



逆方向出力がONしていないのに、原端がONした。

到達端 OFF

前回の出力到達端が OFF している場合

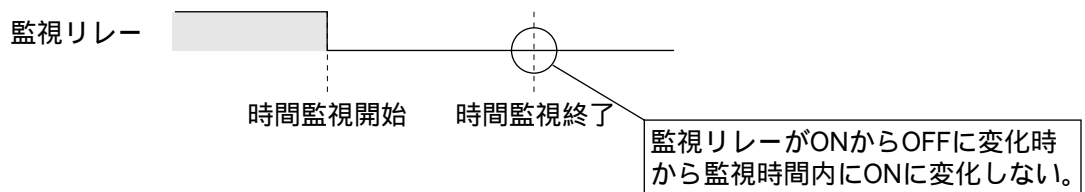


逆方向出力がONしていないのに、到達端がOFFした。

(2) 常時監視リレーの異常

ON 監視異常

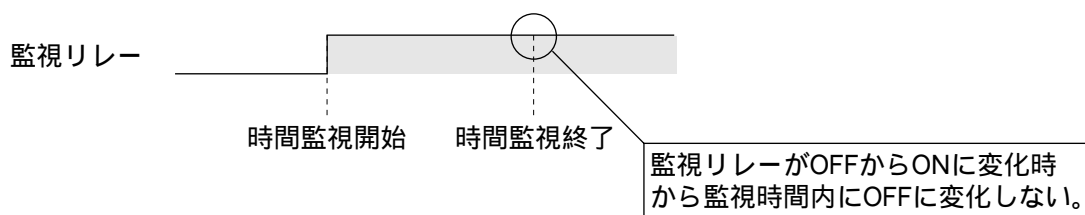
ON であることを監視されているリレーが OFF の場合



監視リレーがONからOFFに変化時から監視時間内にONに変化しない。

OFF 監視異常

OFF であることを監視されているリレーが ON の場合



監視リレーがOFFからONに変化時から監視時間内にOFFに変化しない。

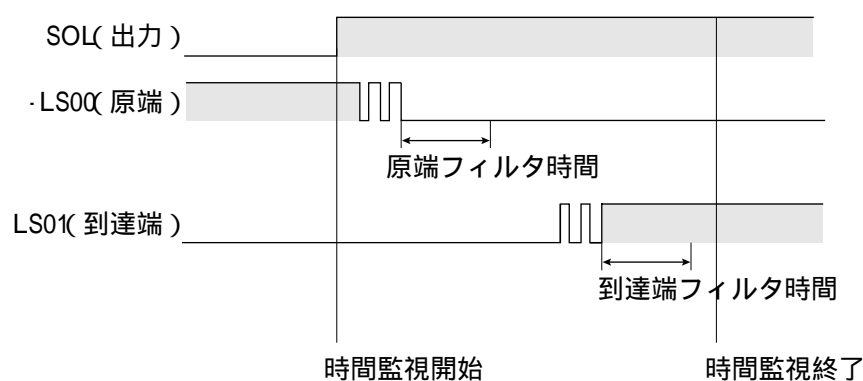
〔 3 〕 設定内容

故障診断機能の使用には、サポートツール(故障診断設定用ソフトJW-300SPD)により、次の項目を設定します。

設定項目	内容
診断許可リレー	故障診断機能の実行を開始するリレーです。 OFFのとき、故障診断は行いません。
診断結果リセットリレー	ONのとき、診断結果をすべてクリアします。
履歴の設定	異常が発生した履歴を、データまたはファイルメモリに格納します。
診断結果異常時出力リレー	故障診断を実行の結果、異常が検出されたとき、ONします。

〔 4 〕 入力リレーのフィルタ機能

入出力サイクル(原端、到達端)に設定するフィルタ時間とは、設定時間内にリレーが変化しても、変化していないとする時間です。本設定により、チャタリング対策が可能です。



留意点

- ・故障診断を行うと、全体のスキャンタイムが長くなります。
故障診断の実行時間は、次のとおりです。
 $24 \mu\text{s} + (3 \mu\text{s} \times N \text{ サイクル})$

6 - 4 PCカードの使用方法

JW300コントロールユニット(JW-3*2CU)はメモリカードインターフェイスを装備しています。
メモリカードの用途は次のとおりです。

ファイルのセーブ・ロード

ユーザープログラム、データ等を、CFカードにセーブ・ロードが可能です。また、現場にサポ
ートツールが無い場合も可能です。 6・19~24ページ参照

メモリの拡張

SRAMカードの使用により、データメモリ容量を拡張(最大16Mバイト 下記参照)でき、

- ・ユーザープログラムのシンボル・コメントデータ格納用として使用できます。
- ・大量の運転データをロギング(動作記録収集)して、トラブル解析や短タクトタイム検討な
どに利用できます。

JW300(JW-3*2CU)のメモリカードインターフェイス

項目	仕様
インターフェイス (コネクタ)	PCカード Type ・ 用コネクタ(68ピン)
電源電圧	3.3V / 5V電源
使用可能 メモリカード	CFカード(変換アダプタ要) 1 PCカードTYPE 変換アダプタを使用 【動作確認済カード】 (株)ハギワラシスコム製 16MB、32MB、64MB、128MB、256MB、512MB 1 SRAMカード(スモールサイズは変換アダプタ要) 【動作確認済カード】ロジテック株 製 1 MB、松下電器産業(株)製 2 MB 1
対象データ	ファイルのセーブ・ロード用として使用する場合 ・ユーザープログラム ・パラメータメモリ ・システムメモリ ・データメモリ メモリの拡張として使用する場合 ・データメモリ(シンボル・コメントデータ、ロギングデータ)

SRAMカードによるメモリの拡張

ファイル アドレス ⁽⁸⁾	JW-312CU	JW-322CU	JW-332CU	JW-342CU	JW-352CU	JW-362CU
00000000						
00073777						
00105777						
00200000						
00277777						
00577777						
02177777						
10177777						
40177777						
20000000000						
	2	2	2	2	2	2
20077777777 (最大)						

2 最大16MバイトのSRAMカードを使用できます。

〔 1 〕セーブ・ロードの対象ファイル

ユーザープログラム、データ等をセーブ・ロードする場合、CFカードを使用してください。

(1) CFカードの対応フォーマット

FAT12
FAT16
VFAT
FAT32

・VFAT、FAT32のときも、ロングファイル名は使用できません。

(2) ファイルの種類と拡張子

CFカードとのセーブ・ロードが可能なファイルは次表のとおりで、ファイルをセーブすると記載の拡張子になります。

ファイルの種類	拡張子	内 容
ユーザープログラム	ppg	ブロック情報とブロック、サブプログラムのタイトルとセットで使用してください。
ブロック情報	pbk	ブロックに関する情報で、ユーザープログラムをセーブ(保存)時に作成されます。ユーザープログラムとブロック、サブプログラムのタイトルとセットで使用してください。
ブロック、サブプログラムのタイトル	ptl	ブロック、サブプログラムのタイトル情報で、ユーザープログラムをセーブ(保存)時に作成されます。ユーザープログラムとブロック情報とセットで使用してください。
システムメモリ	psm	システムメモリのデータがセーブ・ロードされます。 ・セーブ時は#0000～#2777、ロード時は#00200～#2777の範囲です。
パラメータメモリ	ppa	パラメータメモリのデータすべてがセーブ・ロードされます。
データメモリ (ファイルレジスタを除く)	pdt	データメモリ(ファイルレジスタを除く)のデータすべてがセーブ・ロードされます。
ファイルレジスタ	pfl	ファイルレジスタのデータすべてがセーブ・ロードされます。

(3) ファイル名

半角(最大8文字)で設定してください。

ただし、次の文字は使用できません。

「¥」「/」「:」「,」「;」「*」「?」「"」「>」「<」「|」

(4) ディレクトリ

JW300では、ディレクトリ内のファイルを扱えません。使用するファイルは、必ずCFカードのルートディレクトリに入れてください。

(5) 他機種種のファイル

コントロールユニット(JW-3*2CU)の他機種種でセーブ(保存)したファイルは扱えません。サポートツール(JW-300SP)で変換すると、使用できるようになります。

〔 2 〕 特殊リレー(CFカード関係)

CFカードに関する特殊リレーは、次のとおりです。

リレー番号	内 容
007300	CFカード書込時のサムチェック異常発生 ・ CFカードへプログラム保存する場合のみ、保存開始時にプログラムのサムチェックを計算し、計算したサムチェックがPLC内のサムチェックと異なるとONします。 ・ サポートツールからのプログラム書込(全ブロック)、CFカードからのプログラムロード、CFカードへのプログラム保存が正常に行えたときにはOFFします。
007332	CFカードへセーブ実行 ・ 立上り(OFF ON)で、システムメモリ#2221に種類を設定したファイルをセーブ(JW300 CFカード)します。 ファイル名は、システムメモリ#2230～#2236に設定します。 6・24ページ
007333	CFカードビジーフラグ ・ CFカードを操作中、ONします。
007334	CFカードエラーフラグ ・ CFカードを操作時に異常が発生するとONします。ONのときは、CFカードへのセーブを行いません。 ・ CFカードを挿入して正常にマウントされるとOFFします。

〔 3 〕 システムメモリの設定

CFカードに関するシステムメモリの設定は、次のとおりです。

システムメモリ番号(8)	設定項目	内 容								
#2220	CFカードへのセーブ禁止設定	CFカードへのセーブ(書込)の禁止・許可を設定します。 ビット 7 6 5 4 3 2 1 0 #2220 <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td></td> </tr> </table> 初期値：00 _(H) 〓 セーブ (1(ON)...禁止) (0(OFF)...許可) 〓 自動セーブ (1(ON)...禁止) (0(OFF)...許可)	0	0	0	0	0	0		
0	0	0	0	0	0					
#2221	ファイル(種類)の選択	CFカードとセーブ・ロードするファイルの種類を設定します。 ビット 7 6 5 4 3 2 1 0 #2221 <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>0</td><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table> 初期値：3F _(H) ユーザープログラム(ブロック情報とブロック、サブプログラムのタイトルを含む) システムメモリ パラメータメモリ データメモリ(ファイルレジスタを除く) シンボル・コメントメモリ ファイルレジスタ ・ JW-312CUのとき、ファイルレジスタを選択できません。	0	0						
0	0									
#2222	CFカードとのセーブ・ロードの選択	2221で選択のファイルについて、セーブ(JW300 CFカードへの書込) ロード(CFカード JW300への読出)を選択して実行します。 <table border="1" style="display: inline-table; vertical-align: middle;"> <thead> <tr> <th>設定値_(H)</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>11</td> <td>セーブ</td> </tr> <tr> <td>22</td> <td>ロード</td> </tr> </tbody> </table> 初期値：00 _(H) ・ 2222で扱えるファイル名は「SHARP.***」のみです。 ・ ロードは、JW300が運転中のときは実行できません。 ・ CFカードにセーブする場合は、必ずフォーマット済みのカードを使用してください。 ・ セーブ・ロードを実行後、設定値は00 _(H) になります。(異常時も同様)	設定値 _(H)	内容	11	セーブ	22	ロード		
設定値 _(H)	内容									
11	セーブ									
22	ロード									

システムメモリ番号(8)	設定項目	内 容																																																
#2223 #2224 #2225	CFカードへの自動セーブ時間の設定	<p>設定した時間毎に自動でCFカードへセーブします。セーブするファイル(種類)は、#2221で設定します。</p> <p>#2223 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr></table> 初期値：00_(H)</p> <p style="margin-left: 40px;">実行単位(1~127日)</p> <p style="margin-left: 40px;">実行フラグ (1(ON)...実行 0(OFF)...未実行)</p> <p>#2224 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr></table> 初期値：00_(H)</p> <p style="margin-left: 40px;">時(0~23)</p> <p>#2225 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr></table> 初期値：00_(H)</p> <p style="margin-left: 40px;">分(0~59)</p> <ul style="list-style-type: none"> ・ファイル名は「AT年年月月日日.***」となります。 ・実行単位の設定が00_(H)のとき、128日となります。 ・#2224、#2225はBCDで設定してください。 	7	6	5	4	3	2	1	0									7	6	5	4	3	2	1	0									7	6	5	4	3	2	1	0								
7	6	5	4	3	2	1	0																																											
7	6	5	4	3	2	1	0																																											
7	6	5	4	3	2	1	0																																											
#2230 ~ #2236	セーブ時のファイル名	<ul style="list-style-type: none"> ・特殊リレー(007332)により、ファイルをセーブする場合 #2230等の設定値によって、セーブするファイル名が異なります。 6・24ページ ・CFカードを挿入時に、ファイルを自動セーブする場合 #2230~#2236に「英数字のASCIIコード」を設定します。セーブするファイル名は「@*****」となります。 6・24ページ <p style="text-align: right;">初期値：00_(H)</p>																																																
#2237	異常発生時のファイル保存の設定	<p>CFカードへプログラムを保存時にプログラムサムチェック異常が発生した場合について、ファイル保存の設定を行います。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">設定値(H)</th> <th>内 容</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>保存しない(初期値)</td> </tr> <tr> <td>01</td> <td>ファイル名「SHARPx26」で現在のメモリを保存する</td> </tr> <tr> <td>02</td> <td>ファイル名「AUTOLOAD」()で最新ファイルをコピーする</td> </tr> <tr> <td>03</td> <td>ファイル名「SHARPx26」で現在のメモリを保存し、ファイル名「AUTOLOAD」()で最新のファイルをコピーする</td> </tr> <tr> <td>上記以外</td> <td>保存しない</td> </tr> </tbody> </table> <ul style="list-style-type: none"> ・ファイルの種類はシステムメモリ #2221の設定となります。 ・複数回、発生した場合は上書きされます。 <p>ファイル名「AUTOLOAD」は、システムメモリ #2230に77_(H)(ファイル名更新型)を設定し、最新のファイル「xxxxxx00」が存在するときのみファイルを作成します。 (ファイル名「AUTOLOAD」は、最新のファイル「xxxxxx00」がコピーされます。)</p>	設定値(H)	内 容	00	保存しない(初期値)	01	ファイル名「SHARPx26」で現在のメモリを保存する	02	ファイル名「AUTOLOAD」()で最新ファイルをコピーする	03	ファイル名「SHARPx26」で現在のメモリを保存し、ファイル名「AUTOLOAD」()で最新のファイルをコピーする	上記以外	保存しない																																				
設定値(H)	内 容																																																	
00	保存しない(初期値)																																																	
01	ファイル名「SHARPx26」で現在のメモリを保存する																																																	
02	ファイル名「AUTOLOAD」()で最新ファイルをコピーする																																																	
03	ファイル名「SHARPx26」で現在のメモリを保存し、ファイル名「AUTOLOAD」()で最新のファイルをコピーする																																																	
上記以外	保存しない																																																	

[4] セーブ・ロードの操作

CFカードとのファイルのセーブ・ロード(書込・読出)は、次の各操作により行えます。

操作方法		ファイル名	詳細
システムメモリによる	ファイルセーブ	SHARP	次ページ
	ファイルロード	SHARP	
	ファイル自動保存	AT*****	
特殊リレーによるファイルセーブ		システムメモリ#2230 ~ #2236の設定名	6・24ページ
自動セーブ・ロード	電源投入時に自動ロード	AUTOLOAD	
	CFカードを挿入時に自動セーブ	システムメモリ#2230 ~ #2236の設定名	

留意点

JW300コントロールユニット(JW-3*2CU)が「停止中」にCFカードへセーブ(書込)する場合は、短い時間でセーブが終了します。JW-3*2CUが「運転中」にセーブする場合には、長い時間を要します。

操 作	コントロールユニット(JW-3*2CU)の状態	
	停止中	運転中
ファイルセーブ (JW300 CFカードへの書込)	可能 (約5秒以下)	可能 (セーブ時間に注意 下表)
ファイルロード (CFカード JW300への読出)	可能 (約1秒以下)	不可

CFカードへのセーブについて、注意事項は下記のとおりです。

- JW-3*2CUが運転中のCFカードへのセーブ時間は、JW-3*2CUのスキャン時間と機種(メモリ容量)で決定します。ファイルレジスタなど大容量のデータをセーブすると、数時間を要す場合があります。 下表 参照
CARDランプ(JW-3*2CUの前面LED)が点灯時は、絶対に「メモリカードの取り外し」と「JW300の電源断」を行わないでください。誤って行くとCFカード、JW300のデータが破壊される場合があります。
- JW-3*2CUが運転中のCFカードへのセーブは、JW300のスキャン時間への影響を少なくするために、複数スキャンで分割処理しています。よって、同じスキャンで同時にはCFカードへセーブできません。データメモリ等で刻々変化しているデータをCFカードにセーブする場合は注意してください。
- CFカードにアクセス中は、サポートツール等からデータ等を転送・編集しないでください。アクセス中に行うと、CFカードに誤ったデータをセーブする場合があります。

JW-3*2CU 運転中の「CFカードへのセーブ時間」

下表は実測例で、CFカードによる差もありますので、多少のバラツキがあります。

対象機種	JW-312CU		JW-322CU		JW-332CU		JW-342CU		JW-352CU		JW-362CU		
	10ms	30ms	10ms	30ms	10ms	30ms	10ms	30ms	10ms	30ms	10ms	30ms	
セーブ する ファイル	1	3	5	4	7	5	11	7	17	12	30	29	73
データメモリ	2	1	1	1	1	1	1	1	1	1	1	1	1
ファイルレジスタ	-	-	-	1	1	1	3	3	10	15	33	53	130
セーブの合計時間(秒)	4	6	6	9	7	15	11	28	28	64	83	204	

1 プログラムメモリ + システムメモリ + パラメータメモリ

3分24秒

2 ファイルレジスタ以外

1分13秒

セーブ時間は、スキャン時間が長くなると、ほぼ比例して長くなります。上記(10、30ms)以外のスキャン時間については、上記のセーブ時間から比例計算した値を目安にしてください。

(1) システムメモリによる操作

ファイルセーブの操作

システムメモリ設定によるファイルセーブ(JW300 CFカードへの書込)は、次の手順で行います。

【例】ユーザープログラムセーブの場合

1. JW300コントロールユニット(JW-3*2CU)のPCカードスロットに、フォーマット済みのCFカードを挿入する。
2. システムメモリ#2221に01_(H)を設定する。
3. システムメモリ#2222に11_(H)を設定する。
4. システムメモリ#2222が00_(H)になれば、セーブが完了する。

ファイル名「SHARP.ppg」,「SHARP.pbk」,「SHARP.ptl」のファイルが、CFカードに書き込まれています。

CFカードに同名のファイルが既に存在するときは、上書きされます。

ファイルロードの操作

システムメモリ設定によるファイルロード(CFカード JW300への読出)は、次の手順で行います。

【例】ユーザープログラムロードの場合

1. JW300コントロールユニット(JW-3*2CU)のPCカードスロットに、フォーマット済みのCFカードを挿入する。
2. JW300コントロールユニットを停止する。
3. システムメモリ#2221に01_(H)を設定する。
4. ファイル名「SHARP.ppg」,「SHARP.pbk」,「SHARP.ptl」のファイルが、CFカードに存在するのを確認する。
(プログラムメモリのとき、この3種類のファイルが存在しないとロードできません。)
5. システムメモリ#2222に22_(H)を設定する。
6. システムメモリ#2222が00_(H)になれば、ロードが完了する。

ファイル自動セーブの操作

指定した時間毎に自動で、CFカードへファイルをセーブ(書込)できます。時間はシステムメモリ#2223のD₇がONしてからの時間です。手順は次のとおりです。

【例】ユーザープログラムを7日毎の12時30分にセーブする場合

1. JW300コントロールユニット(JW-3*2CU)のPCカードスロットに、フォーマット済みのCFカードを挿入する。
2. JW300コントロールユニットの時計を確認する。
3. システムメモリ#2221に01_(H)を設定する。
4. システムメモリ#2224に12_(BCD), #2225に30_(BCD)を設定する。
5. システムメモリ#2223に87_(H)を設定する。
6. 5を操作の7日後に、ファイル名「AT040201.ppg」,「AT040201.pbk」,「AT040201.ptl」のファイルが、CFカードにセーブされます。(ファイル名は2004年2月1日のとき)

(2) 特殊リレーによる操作(ファイルセーブ)

- 特殊リレー(007332)の立上り(OFF ON)で、ファイルをセーブ(JW300 CFカード)します。
- ・セーブするファイルの種類は、システムメモリ#2221に設定します。 6・20ページ
 - ・ファイル名は、システムメモリ#2230～#2236に設定します。

システムメモリの設定値	セーブするファイル名																																										
#2230 = 00(H)	@SHARP																																										
#2230～#2236 = 「英数字」以外のASCIIコード																																											
#2230～#2236 = 「英数字」のASCIIコード	@***** (#2230～#2236に設定の英数字)																																										
#2230 = FF(H)	月日日時時分																																										
#2230 = 88(H)	***** (レジスタ099670～099677に設定の英数字) ・「英数字」以外のASCIIコードを設定時、@SHARPとなります。																																										
#2230 = 77(H)	#2231～#2236の設定と、書込カウンタ(00～04)がファイル名になります。 ・書込カウンタは、04 00の順に新しくなります。 【例】 <table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="3">セーブ前</th> <th colspan="3">セーブ後</th> </tr> <tr> <th>ファイル名</th> <th>更新時間</th> <th></th> <th>ファイル名</th> <th>更新時間</th> <th></th> </tr> </thead> <tbody> <tr> <td>*****00</td> <td>2003/11/11 09:43</td> <td></td> <td>*****00</td> <td>2003/12/12 14:56</td> <td></td> </tr> <tr> <td>*****01</td> <td>2003/10/10 07:10</td> <td>↙</td> <td>*****01</td> <td>2003/11/11 09:43</td> <td></td> </tr> <tr> <td>*****02</td> <td>2003/09/09 08:45</td> <td>↙</td> <td>*****02</td> <td>2003/10/10 07:10</td> <td></td> </tr> <tr> <td>*****03</td> <td>2003/08/08 13:00</td> <td>↙</td> <td>*****03</td> <td>2003/09/09 08:45</td> <td></td> </tr> <tr> <td>*****04</td> <td>2003/07/07 12:19</td> <td>↙</td> <td>*****04</td> <td>2003/08/08 13:00</td> <td></td> </tr> </tbody> </table> ・#2231～#2236に、「英数字」以外のASCIIコードを設定時、@SHARPとなります。	セーブ前			セーブ後			ファイル名	更新時間		ファイル名	更新時間		*****00	2003/11/11 09:43		*****00	2003/12/12 14:56		*****01	2003/10/10 07:10	↙	*****01	2003/11/11 09:43		*****02	2003/09/09 08:45	↙	*****02	2003/10/10 07:10		*****03	2003/08/08 13:00	↙	*****03	2003/09/09 08:45		*****04	2003/07/07 12:19	↙	*****04	2003/08/08 13:00	
セーブ前			セーブ後																																								
ファイル名	更新時間		ファイル名	更新時間																																							
*****00	2003/11/11 09:43		*****00	2003/12/12 14:56																																							
*****01	2003/10/10 07:10	↙	*****01	2003/11/11 09:43																																							
*****02	2003/09/09 08:45	↙	*****02	2003/10/10 07:10																																							
*****03	2003/08/08 13:00	↙	*****03	2003/09/09 08:45																																							
*****04	2003/07/07 12:19	↙	*****04	2003/08/08 13:00																																							

(3) 自動セーブ・ロードの操作

電源投入時に、CFカードのファイルをJW300にロード(読出)する場合

JW300コントロールユニット(JW-3*2CU)にCFカードを挿入している状態で電源を投入時に、CFカードにファイル名「AUTOLOAD」のファイルが存在し、JW-3*2CUのAUTO LDスイッチがONの場合、CFカードからJW300へロードされます。設定などは特に必要無く、CFカードに存在するファイル名「AUTOLOAD」のファイルすべてがロードされます。

留 意 点

・「ユーザープログラム」をロードするときは、必ず「ブロック情報」と「ブロック、サブプログラムのタイトル」のファイルが必要です。この3ファイルが存在しない場合は、ロードされません。

CFカードを挿入時に、JW300のファイルをCFカードにセーブ(書込)する場合

JW300コントロールユニット(JW-3*2CU)にCFカードを挿入時に、JW-3*2CUのAUTO LDスイッチがOFFの場合、JW300のファイルがCFカードにセーブされます。

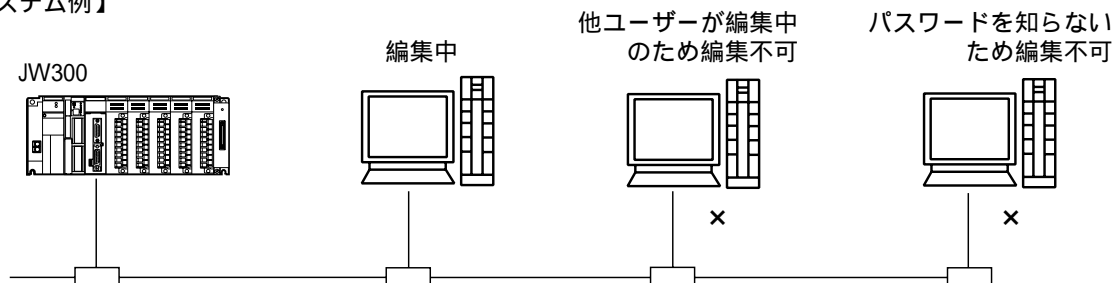
- ・セーブするファイルの種類はシステムメモリ#2221に設定します。00(H)を設定時は、自動セーブは実行されません。
- ・ファイル名はシステムメモリ#2230～#2236に、英数字のASCIIコードで設定し、「@*****」となります。
- ・CFカードに同名のファイルが既に存在するときは、上書きされます。

6 - 5 セキュリティ機能

ネットワークの普及により、複数の場所からプログラムアクセスが容易になっており、プログラムの書込、モード変更(運転/停止)が誰でも簡単に行えるようになりました。

セキュリティ機能では、パスワードを設定することにより、使用できるユーザーを制限します。また、プログラムの書込やモード変更時にログイン動作を行うことにより同時アクセスを防止します。

【システム例】



〔 1 〕 使用上のご注意

(1) 対象機種について

セキュリティ機能は、ソフトバージョン S2.00 以上の JW300 コントロールユニット(JW-311CU ~ JW-362CU)にて使用できます。

(2) サポートツールについて

セキュリティ機能を使用する場合、JW-300SP(ラダー設計支援ソフト)はソフトバージョン 3.0 以上が必要です。セキュリティ機能に未対応の JW-300SP および JW-15PG(ハンディプログラマ)では、セキュリティ機能を操作できません。

セキュリティ機能が有効になっている場合、未対応のサポートツールでは下記の機能を使用できません。(セキュリティ機能が無効の場合は通常どおりの操作が可能です。)

- ・プログラムの書込(設定により JW-15PG のみ可能)
- ・モードの変更(運転 / 停止)
- ・強制 I / O
- ・ブレーク機能

(3) コンピュータリンクについて

セキュリティ機能が有効になっている場合、下記のコンピュータリンクコマンドを受信するとエラーコード「07」を返信します。

FL-net、Ethernet のコンピュータリンク

- ・F8 (PLC の停止 / 停止解除)

コミュニケーションポート、JW-21CM のコンピュータリンク

- ・HLT(PLC の演算停止)
- ・RUN(PLC の運転再開)
- ・CTQ(YMR・CNT の設定値の変更)

〔 2 〕 設定方法

（ 1 ） 有効範囲

下記の操作についてセキュリティ機能が有効になります。

- ・プログラムの書込
- ・モード変更(運転 / 停止)
- ・強制 I / O 処理
- ・ブレーク機能

上記以外の操作に関しては、セキュリティ機能が有効であっても通常の操作が可能です。

（ 2 ） パスワードの種類

パスワードには3種類あり、それぞれ用途が異なります。

パスワード	内 容
管理者用	セキュリティ機能の有効時に設定し、同機能の無効設定時に必要となります。
ログイン用	セキュリティ機能の有効時に設定し、ログイン時に必要となります。
ユーザー用	ログイン時に設定し、ログオフ(強制ログオフ)時に必要となります。

（ 3 ） セキュリティレベル

セキュリティのレベルは、2段階に分けることができます。

セキュリティレベル	内 容
レベル1	すべての有効範囲に対して、セキュリティ機能が有効になります。
レベル2	セキュリティ機能が有効時でも、JW-15PGのプログラム書込のみ許可します。

（ 4 ） 設定手順

セキュリティ機能を設定する手順は以下のとおりです。

1. JW-300SP(バージョン3.0以上)を起動し、プロジェクトを開きます。(JW-300SP ヘルプを参照)
2. メインメニューでセキュリティ ON を選択します。

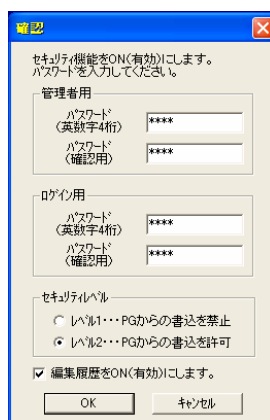
オンライン(O)

PLC操作(I)

セキュリティ機能

セキュリティON

3. 確認ダイアログでパスワード等を設定し、OK を押します。

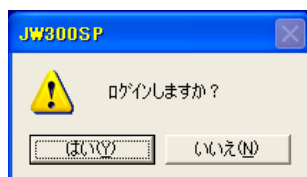


- ・パスワードは英数字4桁を入力してください。
大文字小文字は区別されます。

4. セキュリティ機能が有効になれば、メッセージウィンドウに次のメッセージが表示されます。



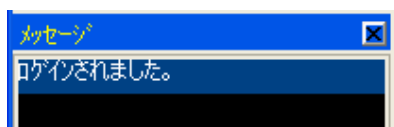
5. モニタ開始操作を行うと、ログイン確認メッセージが表示されます。
「はい」を選択・・・ログイン画面へ移行します。
「いいえ」を選択・・・読み取り専用モードでモニタを開始します。



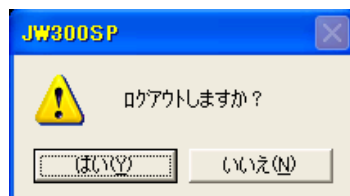
6. ログイン確認ダイアログで、ユーザー名やパスワード等を入力します。

- ・パスワードは英数字4桁を入力してください。
大文字小文字は区別されます。

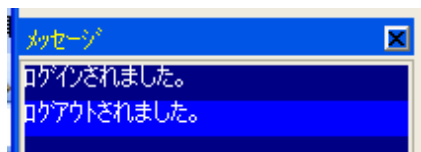
7. ログインすると、メッセージウィンドウに次のメッセージが表示されます。



8. モニタ停止や運転 / 停止の動作後に、ログアウト確認メッセージが表示されます。
「はい」を選択・・・ログアウトを実施します。
「いいえ」を選択・・・ログインを継続したまま、処理を終了します。



9. ログアウトすると、メッセージウィンドウに次のメッセージが表示されます。



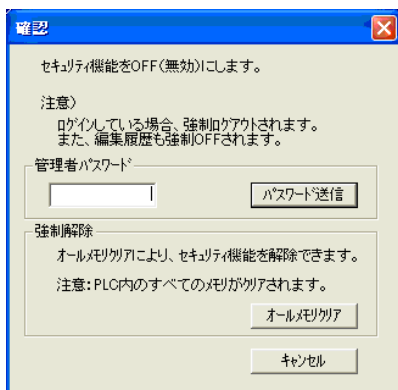
以下は、セキュリティ機能を無効に設定する手順です。

10. メインメニューでセキュリティ OFF を選択します。

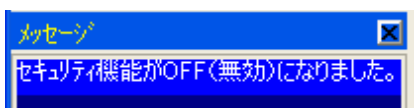
オンライン(O) PLC操作(I) **セキュリティ機能** **セキュリティOFF**

11. 確認ダイアログで管理者パスワードを入力し、[パスワード送信]ボタンを押します。

パスワードを忘れた場合は、セキュリティ機能を無効にできません。その場合、オールメモリクリアによりセキュリティ機能を解除できます。



12. セキュリティ機能が無効になると、メッセージウィンドウに次のメッセージが表示されます。



[3] 注意事項

(1) セキュリティ機能の無効について

セキュリティ機能を無効にした場合、ログインしている状態でも強制的にログオフされます。

(2) スキャンタイムの遅延について

セキュリティ機能の有効 / 無効、セキュリティレベル、編集履歴の有効 / 無効を設定時には、1スキャンのみスキャンタイムが大きく遅延します。(約 400ms)
よって、動作に影響がないときに、これらの設定を行ってください。

(3) 強制ログオフについて

ログインユーザーがユーザーパスワードを忘れたなどの場合、管理者パスワードでログオフすることができます。

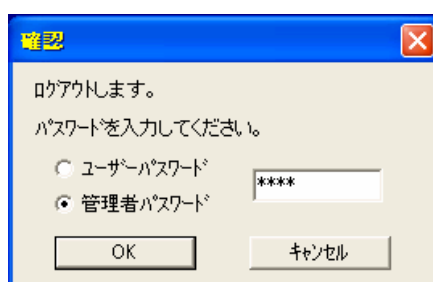
メインメニュー

オンライン(Q)

PLC操作(I)

セキュリティ機能

ログアウト



・ユーザーパスワードでも可

(4) ログイン状況確認について

ログイン状況確認は、次の操作で可能です。

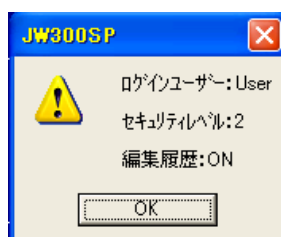
メインメニュー

オンライン(Q)

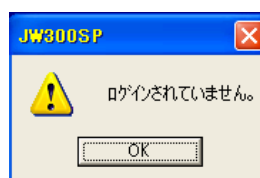
PLC操作(I)

セキュリティ機能

ログイン状況確認



ログインユーザー有りの場合



ログインユーザー無しの場合

(5) メモリクリアについて

パラメータのクリアが実行されると、セキュリティ機能のパスワード等の設定はクリアされて機能は無効になります。

(6) 電源断時について

PLCの電源断でログインされている状態であっても、ログインユーザーなしの状態になります。また、セキュリティ機能の有効 / 無効、セキュリティレベル、編集履歴機能の有効 / 無効、管理者用およびログイン用パスワードは、電源断前の状態を維持します。

6 - 6 編集履歴機能

編集履歴機能では、いつ・誰が・どこから・どのような目的でプログラム編集を行ったかの履歴を残すことにより、編集箇所が明確となりメンテナンス性も向上します。

〔 1 〕 使用上のご注意

(1) 対象機種について

編集履歴機能は、ソフトバージョン S2.00 以上の JW300 コントロールユニット(JW-311CU ~ JW-362CU)にて使用できます。

(2) サポートツールについて

編集履歴機能を使用する場合、JW-300SP のソフトバージョン 3.0 以上が必要となります。

(3) 編集履歴コメントエリアについて

編集履歴コメントは専用のエリアを使用します。各機種にて約 1000 個()の編集履歴コメントを格納できます。コメントの内容により異なる場合があります。

(4) セキュリティ機能について

セキュリティ機能設定が有効時のみ、編集履歴機能の有効 / 無効設定が可能となります。セキュリティ機能が無効時には、編集履歴機能の有効 / 無効を設定できません。

また、セキュリティ機能を無効にすると、同時に編集履歴機能も無効となります。

(5) 命令語について

編集履歴が追加されると、専用の命令語(COM REM)を使用するためにプログラムメモリを 1 語使用します。専用の命令語は NOP と同様の動作をするため、動作には影響いたしません。

〔 2 〕 設定方法

(1) 有効範囲

次の操作について、編集履歴機能が有効になります。

プログラムのオンライン編集(アドレス・設定値の変更、接点の変更を含む)

(2) 設定手順

編集履歴機能を設定する手順は下記のとおりです。

1. セキュリティ機能を有効にし、編集履歴を ON(有効)にします。

メインメニュー

「6 - 5 セキュリティ機能」参照

オンライン(O)

PLC操作(I)

セキュリティ機能

セキュリティON

2. モニタを実行し、ログイン操作を実施します。 「6 - 5 セキュリティ機能」参照

3. 回路編集を実行します。

4. 回路編集を終了時に編集履歴入力画面が表示されるので、コメントを入力します。

5. 編集履歴は次のように格納されます。

JW300SP

接続先: PLC本体
編集内容を書き込みます、よろしいですか?

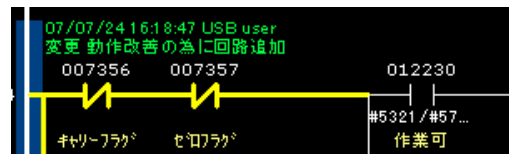
履歴内容

内容: 変更

担当者: user

コメント: 動作改善の為に回路追加

はい(Y) いいえ(N) キャンセル



(3) その他の機能

一覧表示

メインメニュー

オンライン(V)

編集履歴一覧

日付	フロット/プログラム	内容	接続元	担当者	コメント	
07/07/24	14:57:11	ラインファクト回路スタート	変更	USB	user	
07/07/24	15:39:50	ラインファクト回路スタート	変更	USB	user	
07/07/24	15:49:31	ラインファクト回路スタート	変更	USB	user	
07/07/24	16:18:47	ラインファクト出力回路	変更	USB	user	動作改善のために回路追加

履歴コメント編集

ラダー画面で右クリックメニュー

履歴コメント編集

編集(E) ... 履歴コメントの編集を行います。

履歴コメント編集

削除(D) ... 履歴コメントの削除を行います。

履歴コメント編集を行うと、JW-300SPの更新日付が変わります。

また、履歴コメントの削除を行うと、該当する命令語(COM REM)も削除されます。

[3] 注意事項

(1) メモリクリアについて

プログラムメモリをクリアすると、編集履歴コメントメモリもクリアされます。

(JW-300SP で操作時のみクリアされます。 JW-15PG で操作時はクリアされません。)

(2) 電源断時について

電源断を行っても編集履歴機能の有効/無効設定は、電源断前の状態を維持します。

編集履歴のコメントは電池でバックアップされているエリアに格納されていますので、電池電圧が低下するまで保持します。

(3) 時間について

編集履歴の時間は、PLCの時計の時間となります。

使用する前には必ず時計設定を行ってください。

6 - 7 モード変更履歴機能

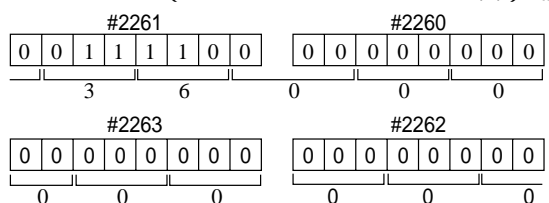
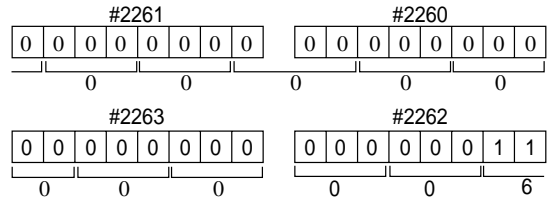
モード変更履歴とは、PLC 運転モードの変更履歴をデータメモリに格納する機能です。
本機能では、いつ・どこから PLC 運転モードが変更されたかを履歴に残すことにより、操作情報を明確にすることができます。

〔1〕対象機種

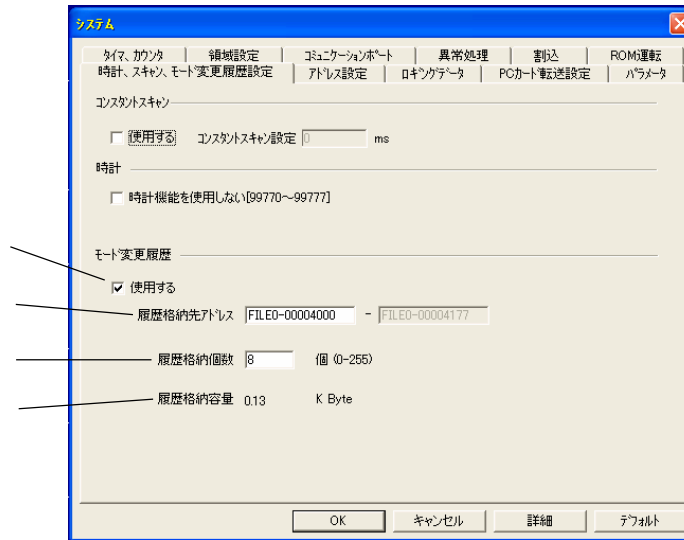
モード変更履歴機能は、ソフトバージョン S2.00 以上の JW300 コントロールユニット(JW-311CU ~ JW-362CU)にて使用できます。

〔2〕設定方法

モード変更履歴機能の設定は、システムメモリ(#0263、#2260 ~ #2264)で行います。
詳細は次のとおりです。

システムメモリ番号 ⁽⁸⁾	設定項目	内 容
# 0263	モード変更履歴の実行設定	モード変更履歴機能の実行/非実行を設定します。 53 _(H) ・・・実行 53 _(H) 以外・・・非実行 (初期値：00 _(H))
# 2260 } # 2263	モード変更履歴データ格納領域の先頭アドレス	モード変更履歴を格納する領域の先頭アドレスを、ファイルアドレスで #2260 ~ #2263 に設定します。 なお、先頭アドレスには「4 の倍数(009000、009004 等)」を設定してください。「4 の倍数」以外するとき #2260 のビット D0、D1 が 0 として指定されます。 例として、009001(ファイルアドレス 004001 ₍₈₎)のときは 009000 となります。 【例 1】レジスタ 109000 ₍₈₎ (ファイルアドレス 00036000 ₍₈₎)を設定時  【例 2】ファイルレジスタ 00400000 ₍₈₎ (ファイルアドレスは 00600000 ₍₈₎)を設定時  ・レジスタ 09000 未満のアドレスを指定した場合、データは格納されません。 ・#2260 ~ #2263 の初期値は 04000 ₍₈₎ です。
# 2264	モード変更履歴の格納数	モード変更履歴データの格納数を設定します。 設定範囲は、001 ~ 255 _(D) です。 00 _(H) のときは格納されません。 (初期値：00 _(H))

JW-300SPのプロパティシートで設定する場合



	内 容
	モード変更履歴を使用する場合は、ONに設定します。
	履歴格納先アドレスを設定します。File0 - 00004000以上を設定してください。
	履歴格納個数を設定します。
	履歴格納容量が表示されます。履歴格納個数に変更されると更新されます。

〔 3 〕 モード変更履歴格納データ

(1) 格納データフォーマット

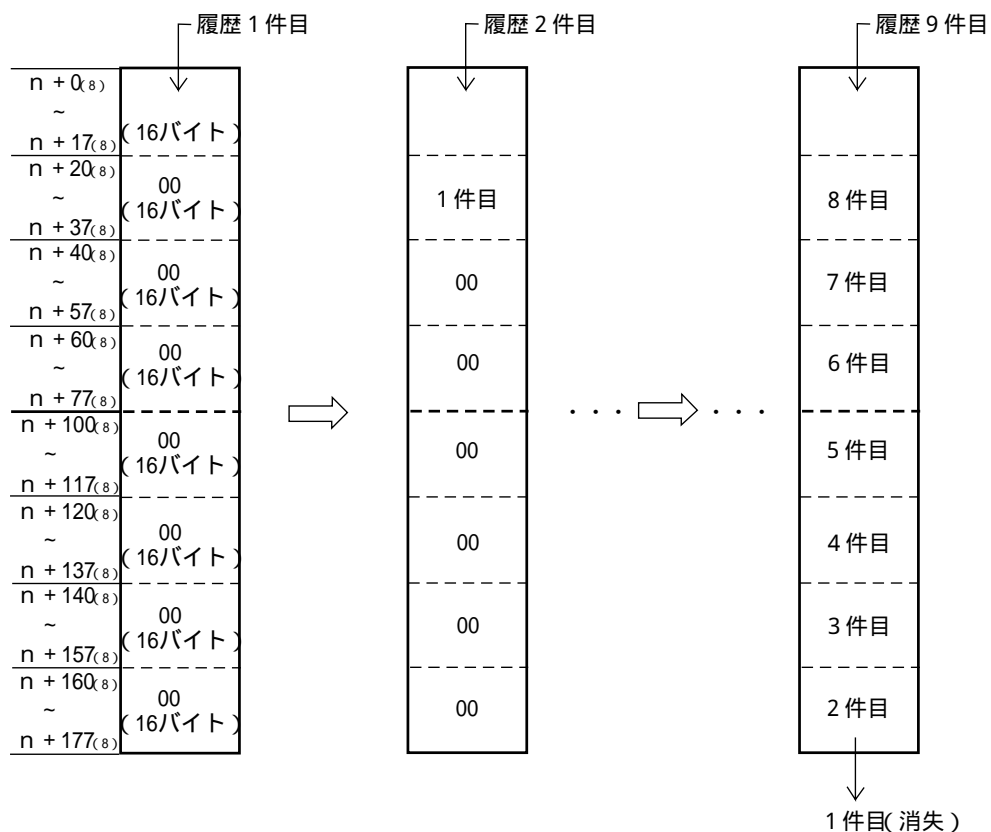
モード変更履歴格納データは、下記のフォーマットで格納されます。

アドレス	内 容			
n+0	秒(BCD)			
n+1	分(BCD)			
n+2	時(BCD)			
n+3	日(BCD)			
n+4	月(BCD)			
n+5	年(BCD)			
n+6	曜日(BCD)			
n+7	変更後のモード(0:モニタモード、2:運転モード、3:プログラムモード)			
n+10 ₍₈₎	接続元情報 (PLCからの 場合)	FF _H) 0:COM1、1:COM2、2:EA、3:USB	接続元情報	ラック番号(0~1)
n+11 ₍₈₎			(オプション からの場合)	ユニット番号(0~7)
n+12 ₍₈₎	予備(0を格納)			
n+13 ₍₈₎	予備(0を格納)			
n+14 ₍₈₎	接続元コード 接続元種類がイーサネット時のみに使用します。 イーサネット…IPアドレス その他…0を格納			
n+15 ₍₈₎				
n+16 ₍₈₎				
n+17 ₍₈₎				

(2) データ格納順序

モード変更履歴データ(16バイト/1件)は、発生順に格納数(システムメモリ#2264)まで格納されます。データが格納数以上になると、最初に格納されたデータから順に消失します。

【例】格納数を8に設定した場合



〔 4 〕 注意事項

（ 1 ） IP アドレスについて

JW-300SP のソフトバージョン 3.0 以上でイーサネット接続時のみ、接続元コードに IP アドレスが格納されます。それ以外の場合には 0 が格納されます。

また、イーサネットのコンピュータリンクモードでモードが変更された場合は、IP アドレスは格納されません。

（ 2 ） 格納エリアのクリアについて

格納エリアのクリア機能はありません。

クリアする場合は、ユーザープログラム等で行ってください。

（ 3 ） 時間について

モード変更履歴の時間は PLC の時間が格納されます。

使用する前には必ず PLC の時計設定を行ってください。

第 7 章 命令語一覽

7 - 1 基本命令一覽

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 007357	キャリー 007356	エラー 007355	リセット 007354	
STR		1	a 接点で論理を開始。中間結果の記憶						8・2
STR NOT		1	b 接点で論理を開始。中間結果の記憶						3
AND		1	論理積						4
AND NOT		1	論理積否定						4
OR		1	論理和						5
OR NOT		1	論理和否定						5
AND STR		1	中間結果との論理積						6
OR STR		1	中間結果との論理和						7
OUT		1	演算結果の出力						8
TMR		2	タイマ(減算式) スタート入力(ONで計数) 設定値 TMR番号(00000 ~ 17777) (0.1 ~ 799.9秒) 内部クロック0.1秒または (0.01 ~ 79.99秒) 0.01秒、0.001秒 (0.001 ~ 7.999秒)					スタート 入力 ON	9
DTMR (BCD)		2	タイマ(減算式) スタート入力(ONで計数) TMR番号(00000 ~ 17777) 設定値(0.1 ~ 799.9秒)					スタート 入力 ON	9
DTMR (BIN)		2	タイマ(減算式) スタート入力(ONで計数) TMR番号(00000 ~ 17777) 設定値(0.1 ~ 3276.7秒)					スタート 入力 ON	9
UTMR (BCD)		2	タイマ(加算式) スタート入力(ONで計数) TMR番号(00000 ~ 17777) 設定値(0.1 ~ 799.9秒)					スタート 入力 ON	9
UTMR (BIN)		2	タイマ(加算式) スタート入力(ONで計数) TMR番号(00000 ~ 17777) 設定値(0.1 ~ 3276.7秒)					スタート 入力 ON	9
CNT		2	カウンタ(減算式) 計数入力 CNT番号(00000 ~ 17777) リセット入力 設定値(1 ~ 7999)					計数入力 ↑	12
DCNT (BCD)		2	カウンタ(減算式) 計数入力 CNT番号(00000 ~ 17777) リセット入力 設定値(1 ~ 7999)					計数入力 ↑	12
DCNT (BIN)		2	カウンタ(減算式) 計数入力 CNT番号(00000 ~ 17777) リセット入力 設定値(1 ~ 32767)					計数入力 ↑	12
UCNT (BCD)		2	カウンタ(加算式) 計数入力 CNT番号(00000 ~ 17777) リセット入力 設定値(1 ~ 7999)					計数入力 ↑	12
UCNT (BIN)		2	カウンタ(加算式) 計数入力 CNT番号(00000 ~ 17777) リセット入力 設定値(1 ~ 32767)					計数入力 ↑	12
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	16

JW300には、下記の微分、出力命令、ビット操作に関連する追加命令(JW30Hに対する)があります。

命令語	シンボル	語数	機能	実行条件	フラグ				詳細ページ
					ゼロ 007357	キャリー 007356	エラー 007355	ノキャリー 007354	
STR POS		1	a 接点ロード立上り					8・17	
STR NEG		1	a 接点ロード立下り					18	
AND POS		1	a 接点論理積立上り					19	
AND NEG		1	a 接点論理積立下り					20	
OR POS		1	a 接点論理和立上り					21	
OR NEG		1	a 接点論理和立下り					22	
OUT POS		1	立上りコイル					23	
OUT NEG		1	立下りコイル					24	
OUT NOT		1	入力条件を反転出力					25	
SET		1	入力の立上り時にセット					26	
RST		1	入力の立上り時にリセット					27	
PUSH		1	アキュムレータ、スタックの内容を、内部の記憶エリアに退避					29	
POP		1	アキュムレータ、スタックの内容を、内部の記憶エリアから復帰						
MRD		1	アキュムレータ、スタックの内容を、内部の記憶エリアから一時的に読み出す						

7 - 2 応用命令一覧

〔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 007357	キャリー 007356	エラー 007355	リセット 007354					
F-00	— <table border="1"><tr><td>F-00</td><td>XFER</td><td>S</td><td>D</td></tr></table>	F-00	XFER	S	D	3	データレジスタ間の1バイト転送	↑					10・1
F-00	XFER	S	D										
F-00w	— <table border="1"><tr><td>F-00w</td><td>XFER</td><td>S</td><td>D</td></tr></table>	F-00w	XFER	S	D	3	データレジスタ間の1ワード転送	↑					1
F-00w	XFER	S	D										
F-00d	— <table border="1"><tr><td>F-00d</td><td>XFER</td><td>S</td><td>D</td></tr></table>	F-00d	XFER	S	D	3	データレジスタ間の2ワード転送	↑					2
F-00d	XFER	S	D										
F-01	— <table border="1"><tr><td>F-01</td><td>BCD</td><td>n</td><td>D</td></tr></table>	F-01	BCD	n	D	3	BCD定数(2桁)の転送	↑					3
F-01	BCD	n	D										
F-01w	— <table border="1"><tr><td>F-01w</td><td>BCD</td><td>n</td><td>D</td></tr></table>	F-01w	BCD	n	D	3	BCD定数(4桁)の転送	↑					3
F-01w	BCD	n	D										
F-01d	— <table border="1"><tr><td>F-01d</td><td>BCD</td><td>n</td><td>D</td></tr></table>	F-01d	BCD	n	D	3	BCD定数(8桁)の転送	↑					4
F-01d	BCD	n	D										
F-02	— <table border="1"><tr><td>F-02</td><td>XCHG</td><td>D₁</td><td>D₂</td></tr></table>	F-02	XCHG	D ₁	D ₂	3	レジスタ間(1バイト)のデータ交換	↑					5
F-02	XCHG	D ₁	D ₂										
F-02w	— <table border="1"><tr><td>F-02w</td><td>XCHG</td><td>D₁</td><td>D₂</td></tr></table>	F-02w	XCHG	D ₁	D ₂	3	レジスタ間(1バイト)のデータ交換	↑					5
F-02w	XCHG	D ₁	D ₂										
F-02d	— <table border="1"><tr><td>F-02d</td><td>XCHG</td><td>D₁</td><td>D₂</td></tr></table>	F-02d	XCHG	D ₁	D ₂	3	レジスタ間(2ワード)のデータ交換	↑					6
F-02d	XCHG	D ₁	D ₂										
F-03	— <table border="1"><tr><td>F-03</td><td>→BIN</td><td>S</td><td>D</td></tr></table>	F-03	→BIN	S	D	3	BCD(2桁) BIN(8ビット)変換	↑	0	0	↕	0	7
F-03	→BIN	S	D										
F-03w	— <table border="1"><tr><td>F-03w</td><td>→BIN</td><td>S</td><td>D</td></tr></table>	F-03w	→BIN	S	D	3	BCD(4桁) BIN(16ビット)変換	↑	0	0	↕	0	7
F-03w	→BIN	S	D										
F-03d	— <table border="1"><tr><td>F-03d</td><td>→BIN</td><td>S</td><td>D</td></tr></table>	F-03d	→BIN	S	D	3	BCD(8桁) BIN(32ビット)変換	↑	0	0	↕	0	8
F-03d	→BIN	S	D										
F-04	— <table border="1"><tr><td>F-04</td><td>→BCD</td><td>S</td><td>D</td></tr></table>	F-04	→BCD	S	D	3	BIN(8ビット) BCD(2桁)変換	↑					9
F-04	→BCD	S	D										
F-04w	— <table border="1"><tr><td>F-04w</td><td>→BCD</td><td>S</td><td>D</td></tr></table>	F-04w	→BCD	S	D	3	BIN(16ビット) BCD(6桁)変換	↑					9
F-04w	→BCD	S	D										
F-04d	— <table border="1"><tr><td>F-04d</td><td>→BCD</td><td>S</td><td>D</td></tr></table>	F-04d	→BCD	S	D	3	BIN(32ビット) BCD(12桁)変換	↑					10
F-04d	→BCD	S	D										
F-05	— <table border="1"><tr><td>F-05</td><td>DMPX</td><td>S</td><td>D</td></tr></table>	F-05	DMPX	S	D	3	1バイトデータの分配	↑					11
F-05	DMPX	S	D										
F-05w	— <table border="1"><tr><td>F-05w</td><td>DMPX</td><td>S</td><td>D</td></tr></table>	F-05w	DMPX	S	D	3	1ワードデータの分配	↑					12
F-05w	DMPX	S	D										
F-05d	— <table border="1"><tr><td>F-05d</td><td>DMPX</td><td>S</td><td>D</td></tr></table>	F-05d	DMPX	S	D	3	2ワードデータの分配	↑					13
F-05d	DMPX	S	D										
F-06	— <table border="1"><tr><td>F-06</td><td>MPX</td><td>S</td><td>D</td></tr></table>	F-06	MPX	S	D	3	1バイトデータの抽出	↑					14
F-06	MPX	S	D										
F-06w	— <table border="1"><tr><td>F-06w</td><td>MPX</td><td>S</td><td>D</td></tr></table>	F-06w	MPX	S	D	3	1ワードデータの抽出	↑					15
F-06w	MPX	S	D										
F-06d	— <table border="1"><tr><td>F-06d</td><td>MPX</td><td>S</td><td>D</td></tr></table>	F-06d	MPX	S	D	3	2ワードデータの抽出	↑					16
F-06d	MPX	S	D										
F-07	— <table border="1"><tr><td>F-07</td><td>DCML</td><td>n</td><td>D</td></tr></table>	F-07	DCML	n	D	3	10進定数(1バイト)の転送	↑					17
F-07	DCML	n	D										
F-07w	— <table border="1"><tr><td>F-07w</td><td>DCML</td><td>n</td><td>D</td></tr></table>	F-07w	DCML	n	D	3	10進定数(1ワード)の転送	↑					17
F-07w	DCML	n	D										
F-07d	— <table border="1"><tr><td>F-07d</td><td>DCML</td><td>n</td><td>D</td></tr></table>	F-07d	DCML	n	D	3	10進定数(2ワード)の転送	↑					18
F-07d	DCML	n	D										
F-08	— <table border="1"><tr><td>F-08</td><td>OCT</td><td>n</td><td>D</td></tr></table>	F-08	OCT	n	D	3	8進定数(1バイト)の転送	↑					19
F-08	OCT	n	D										
F-08w	— <table border="1"><tr><td>F-08w</td><td>OCT</td><td>n</td><td>D</td></tr></table>	F-08w	OCT	n	D	3	8進定数(1ワード)の転送	↑					19
F-08w	OCT	n	D										
F-08d	— <table border="1"><tr><td>F-08d</td><td>OCT</td><td>n</td><td>D</td></tr></table>	F-08d	OCT	n	D	3	8進定数(2ワード)の転送	↑					20
F-08d	OCT	n	D										
F-09	— <table border="1"><tr><td>F-09</td><td>INV</td><td>S</td><td>D</td></tr></table>	F-09	INV	S	D	3	8ビットデータの反転	↑					21
F-09	INV	S	D										
F-09w	— <table border="1"><tr><td>F-09w</td><td>INV</td><td>S</td><td>D</td></tr></table>	F-09w	INV	S	D	3	16ビットデータの反転	↑					21
F-09w	INV	S	D										
F-09d	— <table border="1"><tr><td>F-09d</td><td>INV</td><td>S</td><td>D</td></tr></table>	F-09d	INV	S	D	3	32ビットデータの反転	↑					22
F-09d	INV	S	D										

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 007357	キャリー 007356	エラー 007355	オーバー 007354	
F-10	$\overline{\text{F-10 ADD}} \quad S_1 \quad S_2 \quad D$	4	レジスタ間(BCD 2桁)の加算	↑	↑↓	↑↓	↑↓	↑↓	10・23
F-10w	$\overline{\text{F-10w ADD}} \quad S_1 \quad S_2 \quad D$	4	レジスタ間(BCD 4桁)の加算	↑	↑↓	↑↓	↑↓	↑↓	24
F-10d	$\overline{\text{F-10d ADD}} \quad S_1 \quad S_2 \quad D$	4	レジスタ間(BCD 8桁)の加算	↑	↑↓	↑↓	↑↓	↑↓	25
Fc10	$\overline{\text{Fc10 ADD}} \quad S_1 \quad n \quad D$	4	レジスタ(BCD 2桁)と定数(2桁)の加算	↑	↑↓	↑↓	↑↓	↑↓	26
Fc10w	$\overline{\text{Fc10w ADD}} \quad S_1 \quad n \quad D$	4	レジスタ(BCD 4桁)と定数(4桁)の加算	↑	↑↓	↑↓	↑↓	↑↓	27
Fc10d	$\overline{\text{Fc10d ADD}} \quad S_1 \quad n \quad D$	4	レジスタ(BCD 8桁)と定数(4桁)の加算	↑	↑↓	↑↓	↑↓	↑↓	28
F-11	$\overline{\text{F-11 SUB}} \quad S_1 \quad S_2 \quad D$	4	レジスタ間(BCD 2桁)の減算	↑	↑↓	↑↓	↑↓	↑↓	29
F-11w	$\overline{\text{F-11w SUB}} \quad S_1 \quad S_2 \quad D$	4	レジスタ間(BCD 4桁)の減算	↑	↑↓	↑↓	↑↓	↑↓	30
F-11d	$\overline{\text{F-11d SUB}} \quad S_1 \quad S_2 \quad D$	4	レジスタ間(BCD 8桁)の減算	↑	↑↓	↑↓	↑↓	↑↓	31
Fc11	$\overline{\text{Fc11 SUB}} \quad S_1 \quad n \quad D$	4	レジスタ(BCD 2桁)と定数(2桁)の減算	↑	↑↓	↑↓	↑↓	↑↓	32
Fc11w	$\overline{\text{Fc11w SUB}} \quad S_1 \quad n \quad D$	4	レジスタ(BCD 4桁)と定数(4桁)の減算	↑	↑↓	↑↓	↑↓	↑↓	33
Fc11d	$\overline{\text{Fc11d SUB}} \quad S_1 \quad n \quad D$	4	レジスタ(BCD 8桁)と定数(4桁)の減算	↑	↑↓	↑↓	↑↓	↑↓	34
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	↑↓	35
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	↑↓	36
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	↑↓	37
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	↑↓	38
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	↑↓	38
Fc12d	$\overline{\text{Fc12d CMP}} \quad S_1 \quad n$	3	レジスタと8進定数(2ワード)の比較	ON	↑↓	↑↓	0	↑↓	39
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	↑↓	40
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	↑↓	40
Fx12d	$\overline{\text{Fx12d CMP}} \quad S_1 \quad n$	3	レジスタと16進定数(2ワード)の比較	ON	↑↓	↑↓	0	↑↓	41
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	↑					42
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	↑					42
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	↑					43
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	↑					44
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	↑					44
Fc13d	$\overline{\text{Fc13d AND}} \quad n \quad D$	3	レジスタと8進定数(2ワード)の論理積	↑					45
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	↑					46
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	↑					46
Fx13d	$\overline{\text{Fx13d AND}} \quad n \quad D$	3	レジスタと16進定数(2ワード)の論理積	↑					47

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 007357	キャリー 007356	エラー 007355	ノキャリー 007354					
F-14	$\overline{F-14}$ <table border="1"><tr><td>OR</td><td>S</td><td>D</td></tr></table>	OR	S	D	3	レジスタ間(1バイト)の論理和	↑					10・48	
OR	S	D											
F-14w	$\overline{F-14w}$ <table border="1"><tr><td>OR</td><td>S</td><td>D</td></tr></table>	OR	S	D	3	レジスタ間(1ワード)の論理和	↑					48	
OR	S	D											
F-14d	$\overline{F-14d}$ <table border="1"><tr><td>OR</td><td>S</td><td>D</td></tr></table>	OR	S	D	3	レジスタ間(2ワード)の論理和	↑					49	
OR	S	D											
Fc14	$\overline{Fc14}$ <table border="1"><tr><td>OR</td><td>n</td><td>D</td></tr></table>	OR	n	D	3	レジスタと8進定数(1バイト)の論理和	↑					50	
OR	n	D											
Fc14w	$\overline{Fc14w}$ <table border="1"><tr><td>OR</td><td>n</td><td>D</td></tr></table>	OR	n	D	3	レジスタと8進定数(1ワード)の論理和	↑					50	
OR	n	D											
Fc14d	$\overline{Fc14d}$ <table border="1"><tr><td>OR</td><td>n</td><td>D</td></tr></table>	OR	n	D	3	レジスタと8進定数(2ワード)の論理和	↑					51	
OR	n	D											
Fx14	$\overline{Fx14}$ <table border="1"><tr><td>OR</td><td>n</td><td>D</td></tr></table>	OR	n	D	3	レジスタと16進定数(1バイト)の論理和	↑					52	
OR	n	D											
Fx14w	$\overline{Fx14w}$ <table border="1"><tr><td>OR</td><td>n</td><td>D</td></tr></table>	OR	n	D	3	レジスタと16進定数(1ワード)の論理和	↑					52	
OR	n	D											
Fx14d	$\overline{Fx14d}$ <table border="1"><tr><td>OR</td><td>n</td><td>D</td></tr></table>	OR	n	D	3	レジスタと16進定数(2ワード)の論理和	↑					53	
OR	n	D											
F-15	$\overline{F-15}$ <table border="1"><tr><td>MUL</td><td>S₁</td><td>S₂</td><td>D</td></tr></table>	MUL	S ₁	S ₂	D	4	レジスタ間(BCD 4桁)の乗算	↑	0	0	↕	0	11・1
MUL	S ₁	S ₂	D										
F-15d	$\overline{F-15d}$ <table border="1"><tr><td>MUL</td><td>S₁</td><td>S₂</td><td>D</td></tr></table>	MUL	S ₁	S ₂	D	4	レジスタ間(BCD 8桁)の乗算	↑	0	0	↕	0	2
MUL	S ₁	S ₂	D										
Fc15	$\overline{Fc15}$ <table border="1"><tr><td>MUL</td><td>S₁</td><td>n</td><td>D</td></tr></table>	MUL	S ₁	n	D	4	レジスタ間(BCD 4桁)とBCD定数(3桁)の乗算	↑	0	0	↕	0	3
MUL	S ₁	n	D										
Fc15d	$\overline{Fc15d}$ <table border="1"><tr><td>MUL</td><td>S₁</td><td>n</td><td>D</td></tr></table>	MUL	S ₁	n	D	4	レジスタ間(BCD 8桁)とBCD定数(4桁)の乗算	↑	0	0	↕	0	4
MUL	S ₁	n	D										
F-16	$\overline{F-16}$ <table border="1"><tr><td>DIV</td><td>S₁</td><td>S₂</td><td>D</td></tr></table>	DIV	S ₁	S ₂	D	4	レジスタ(BCD 4桁)とレジスタ(BCD 2桁)の除算	↑	0	0	↕	0	5
DIV	S ₁	S ₂	D										
F-16d	$\overline{F-16d}$ <table border="1"><tr><td>DIV</td><td>S₁</td><td>S₂</td><td>D</td></tr></table>	DIV	S ₁	S ₂	D	4	レジスタ(BCD 8桁)とレジスタ(BCD 8桁)の除算	↑	0	0	↕	0	6
DIV	S ₁	S ₂	D										
Fc16	$\overline{Fc16}$ <table border="1"><tr><td>DIV</td><td>S₁</td><td>n</td><td>D</td></tr></table>	DIV	S ₁	n	D	4	レジスタ(BCD 4桁)とBCD定数(2桁)の除算	↑	0	0	↕	0	7
DIV	S ₁	n	D										
Fc16d	$\overline{Fc16d}$ <table border="1"><tr><td>DIV</td><td>S₁</td><td>n</td><td>D</td></tr></table>	DIV	S ₁	n	D	4	レジスタ(BCD 8桁)とBCD定数(4桁)の除算	↑	0	0	↕	0	8
DIV	S ₁	n	D										
F-17	$\overline{F-17}$ <table border="1"><tr><td>XNR</td><td>S</td><td>D</td></tr></table>	XNR	S	D	3	レジスタ間(1バイト)の一致	↑					9	
XNR	S	D											
F-17w	$\overline{F-17w}$ <table border="1"><tr><td>XNR</td><td>S</td><td>D</td></tr></table>	XNR	S	D	3	レジスタ間(1ワード)の一致	↑					9	
XNR	S	D											
F-17d	$\overline{F-17d}$ <table border="1"><tr><td>XNR</td><td>S</td><td>D</td></tr></table>	XNR	S	D	3	レジスタ間(2ワード)の一致	↑					10	
XNR	S	D											
Fc17	$\overline{Fc17}$ <table border="1"><tr><td>XNR</td><td>n</td><td>D</td></tr></table>	XNR	n	D	3	レジスタと8進定数(1バイト)の一致	↑					11	
XNR	n	D											
Fc17w	$\overline{Fc17w}$ <table border="1"><tr><td>XNR</td><td>n</td><td>D</td></tr></table>	XNR	n	D	3	レジスタと8進定数(1ワード)の一致	↑					11	
XNR	n	D											
Fc17d	$\overline{Fc17d}$ <table border="1"><tr><td>XNR</td><td>n</td><td>D</td></tr></table>	XNR	n	D	3	レジスタと8進定数(2ワード)の一致	↑					12	
XNR	n	D											
Fx17	$\overline{Fx17}$ <table border="1"><tr><td>XNR</td><td>n</td><td>D</td></tr></table>	XNR	n	D	3	レジスタと16進定数(1バイト)の一致	↑					13	
XNR	n	D											
Fx17w	$\overline{Fx17w}$ <table border="1"><tr><td>XNR</td><td>n</td><td>D</td></tr></table>	XNR	n	D	3	レジスタと16進定数(1ワード)の一致	↑					13	
XNR	n	D											
Fx17d	$\overline{Fx17d}$ <table border="1"><tr><td>XNR</td><td>n</td><td>D</td></tr></table>	XNR	n	D	3	レジスタと16進定数(2ワード)の一致	↑					14	
XNR	n	D											
F-18	$\overline{F-18}$ <table border="1"><tr><td>XOR</td><td>S</td><td>D</td></tr></table>	XOR	S	D	3	レジスタ間(1バイト)の排他的論理和	↑					15	
XOR	S	D											
F-18w	$\overline{F-18w}$ <table border="1"><tr><td>XOR</td><td>S</td><td>D</td></tr></table>	XOR	S	D	3	レジスタ間(1ワード)の排他的論理和	↑					15	
XOR	S	D											
F-18d	$\overline{F-18d}$ <table border="1"><tr><td>XOR</td><td>S</td><td>D</td></tr></table>	XOR	S	D	3	レジスタ間(2ワード)の排他的論理和	↑					16	
XOR	S	D											
Fc18	$\overline{Fc18}$ <table border="1"><tr><td>XOR</td><td>n</td><td>D</td></tr></table>	XOR	n	D	3	レジスタと8進定数(1バイト)の排他的論理和	↑					17	
XOR	n	D											
Fc18w	$\overline{Fc18w}$ <table border="1"><tr><td>XOR</td><td>n</td><td>D</td></tr></table>	XOR	n	D	3	レジスタと8進定数(1ワード)の排他的論理和	↑					17	
XOR	n	D											
Fc18d	$\overline{Fc18d}$ <table border="1"><tr><td>XOR</td><td>n</td><td>D</td></tr></table>	XOR	n	D	3	レジスタと8進定数(2ワード)の排他的論理和	↑					18	
XOR	n	D											

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 007357	キャリー 007356	エラー 007355	オーバー 007354	
Fx18		3	レジスタと16進定数(1バイト)の排他的論理和	↑					11・19
Fx18w		3	レジスタと16進定数(1ワード)の排他的論理和	↑					19
Fx18d		3	レジスタと16進定数(2ワード)の排他的論理和	↑					20
F-20		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					21
F-21		3	レジスタ(BCD 8桁)の平方根	↑	0	0	↑↓	0	22
F-22		3	三角関数(SIN)の演算	↑	0	↑↓	↑↓	↑↓	22
F-23		3	三角関数(COS)の演算	↑	0	↑↓	↑↓	↑↓	23
F-24		3	三角関数(TAN)の演算	↑	0	↑↓	↑↓	↑↓	24
F-25		3	三角関数(SIN ⁻¹)の演算	↑	0	↑↓	↑↓	↑↓	25
F-26		3	三角関数(COS ⁻¹)の演算	↑	0	↑↓	↑↓	↑↓	26
F-27		3	三角関数(TAN ⁻¹)の演算	↑	0	↑↓	↑↓	↑↓	27
F-28		3	直交座標系(X,Y)データの極座標系(,)への変換	↑	0	0	↑↓	0	28
F-29		3	極座標系(,)データの直交座標系(X,Y)への変換	↑	0	0	↑↓	0	29
F-30		1	マスターコントロールのセット	ON					30
F-31		1	マスターコントロールのリセット						30
F-32		2	セットコイル	ON					33
F-33		2	リセットコイル	ON					34
F-34		4	時計の現在値との比較 (指定リレーのセット)	ON					36
F-35		4	時計の現在値との比較 (指定リレーのリセット)	ON					37
F-36		4	時計の加算	↑	↑↓	↑↓	↑↓	↑↓	38
F-37		4	時計の減算	↑	↑↓	↑↓	↑↓	↑↓	39
F-38		2	時計現在値の転送	↑					40
F-40		1	END命令						41
F-41		1	ジャンプコントロールのセット	OFF					42
F-42		1	ジャンプコントロールのリセット						42
F-43		1	ビット反転(ACCの内容を反転)						44
F-44		1	ON時微分接点	↑					45
F-45		1	OFF時微分接点	↓					46
F-47		1	レベル演算条件セット						47
F-48		1	レベル演算条件リセット						47
F-49		1	条件END	OFF					48

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 007357	キャリー 007356	エラー 007355	オーバー 007354	
F-50		3	4 16デコーダ	↑					12・1
F-51		3	16 4エンコーダ	↑					1
F-52		3	7 SEGデコーダ	↑					2
F-53		3	BCD(4桁) BIN(16ビット) 変換	↑	0	0	↑↓	0	3
F-54		3	BIN(16ビット) BCD(6桁) 変換	↑					3
F-55		3	上位 4 ビットと下位 4 ビットの交換	↑					4
F-56		3	1 バイトデータの10の補数	↑	0	0	↑↓	0	5
F-56w		3	1 ワードデータの10の補数	↑	0	0	↑↓	0	5
F-56d		3	2 ワードデータの10の補数	↑	0	0	↑↓	0	6
F-57		3	1 バイトデータの 2 の補数	↑					7
F-57w		3	1 ワードデータの 2 の補数	↑					7
F-57d		3	2 ワードデータの 2 の補数	↑					8
F-58		4	ONビット数の合計	↑					9
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	↑↓	10
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	↑↓	12
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	↑↓	13
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	↑↓	14
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	↑↓	15
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	↑↓	16
F-62		2	BCD 2 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 ↑	↑↓	↑↓	↑↓	↑↓	17
F-62w		2	BCD 4 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 ↑	↑↓	↑↓	↑↓	↑↓	18
F-62d		2	BCD 8 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 ↑	↑↓	↑↓	↑↓	↑↓	19
F-63		2	バイナリ加算カウンタ(1 バイト)	↑	↑↓	↑↓	0	↑↓	20
F-63w		2	バイナリ加算カウンタ(1 ワード)	↑	↑↓	↑↓	0	↑↓	20
F-63d		2	バイナリ加算カウンタ(2 ワード)	↑	↑↓	↑↓	0	↑↓	21
F-64		2	バイナリ減算カウンタ(1 バイト)	↑	↑↓	↑↓	0	↑↓	22
F-64w		2	バイナリ減算カウンタ(1 ワード)	↑	↑↓	↑↓	0	↑↓	22
F-64d		2	バイナリ減算カウンタ(2 ワード)	↑	↑↓	↑↓	0	↑↓	23

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 007357	キャリー 007356	エラー 007355	ノック 007354					
F-65	<table border="1"><tr><td>F-65 BCDI</td><td>D</td></tr></table>	F-65 BCDI	D	2	BCD加算カウンタ(1バイト)	↑	↑↓	↑↓	↑↓	↑↓	12・24		
F-65 BCDI	D												
F-65w	<table border="1"><tr><td>F-65w BCDI</td><td>D</td></tr></table>	F-65w BCDI	D	2	BCD加算カウンタ(1ワード)	↑	↑↓	↑↓	↑↓	↑↓	24		
F-65w BCDI	D												
F-65d	<table border="1"><tr><td>F-65d BCDI</td><td>D</td></tr></table>	F-65d BCDI	D	2	BCD加算カウンタ(2ワード)	↑	↑↓	↑↓	↑↓	↑↓	25		
F-65d BCDI	D												
F-66	<table border="1"><tr><td>F-66 BCDD</td><td>D</td></tr></table>	F-66 BCDD	D	2	BCD減算カウンタ(1バイト)	↑	↑↓	↑↓	↑↓	↑↓	26		
F-66 BCDD	D												
F-66w	<table border="1"><tr><td>F-66w BCDD</td><td>D</td></tr></table>	F-66w BCDD	D	2	BCD減算カウンタ(1ワード)	↑	↑↓	↑↓	↑↓	↑↓	26		
F-66w BCDD	D												
F-66d	<table border="1"><tr><td>F-66d BCDD</td><td>D</td></tr></table>	F-66d BCDD	D	2	BCD減算カウンタ(2ワード)	↑	↑↓	↑↓	↑↓	↑↓	27		
F-66d BCDD	D												
F-67	<table border="1"><tr><td>F-67 NSFH</td><td>n</td><td>D</td></tr></table>	F-67 NSFH	n	D	3	桁シフト(上位シフト)	↑					28	
F-67 NSFH	n	D											
F-68	<table border="1"><tr><td>F-68 NSFL</td><td>n</td><td>D</td></tr></table>	F-68 NSFL	n	D	3	桁シフト(下位シフト)	↑					28	
F-68 NSFL	n	D											
F-69	<table border="1"><tr><td>F-69 NXFR</td><td>S</td><td>D</td></tr></table>	F-69 NXFR	S	D	3	桁転送	↑					29	
F-69 NXFR	S	D											
F-70	<table border="1"><tr><td>F-70 FILE</td><td>n</td><td>S</td><td>D</td></tr></table>	F-70 FILE	n	S	D	4	nバイト一括転送	↑					30
F-70 FILE	n	S	D										
F-70w	<table border="1"><tr><td>F-70w FILE</td><td>n</td><td>S</td><td>D</td></tr></table>	F-70w FILE	n	S	D	4	nワード一括転送	↑					31
F-70w FILE	n	S	D										
F-70d	<table border="1"><tr><td>F-70d FILE</td><td>n</td><td>S</td><td>D</td></tr></table>	F-70d FILE	n	S	D	4	nダブルワード一括転送	↑					32
F-70d FILE	n	S	D										
F-71	<table border="1"><tr><td>F-71 CONS</td><td>n</td><td>D₁</td><td>D₂</td></tr></table>	F-71 CONS	n	D ₁	D ₂	4	8進定数(1バイト)一括転送	↑					33
F-71 CONS	n	D ₁	D ₂										
F-71w	<table border="1"><tr><td>F-71w CONS</td><td>n</td><td>D₁</td><td>D₂</td></tr></table>	F-71w CONS	n	D ₁	D ₂	4	8進定数(1ワード)一括転送	↑					34
F-71w CONS	n	D ₁	D ₂										
F-71d	<table border="1"><tr><td>F-71d CONS</td><td>n</td><td>D₁</td><td>D₂</td></tr></table>	F-71d CONS	n	D ₁	D ₂	4	8進定数(2ワード)一括転送	↑					34
F-71d CONS	n	D ₁	D ₂										
F-72	<table border="1"><tr><td>F-72 DMPX</td><td>n</td><td>S</td><td>D</td></tr></table>	F-72 DMPX	n	S	D	4	ファイルレジスタFILE 1へのnバイト分配	↑					35
F-72 DMPX	n	S	D										
F-72w	<table border="1"><tr><td>F-72w DMPX</td><td>n</td><td>S</td><td>D</td></tr></table>	F-72w DMPX	n	S	D	4	ファイルレジスタFILE 1へのnワード分配	↑					36
F-72w DMPX	n	S	D										
F-72d	<table border="1"><tr><td>F-72d DMPX</td><td>n</td><td>S</td><td>D</td></tr></table>	F-72d DMPX	n	S	D	4	ファイルレジスタFILE 1へのnダブルワード分配	↑					37
F-72d DMPX	n	S	D										
F-73	<table border="1"><tr><td>F-73 MPX</td><td>n</td><td>S</td><td>D</td></tr></table>	F-73 MPX	n	S	D	4	ファイルレジスタFILE 1からのnバイト抽出	↑					38
F-73 MPX	n	S	D										
F-73w	<table border="1"><tr><td>F-73w MPX</td><td>n</td><td>S</td><td>D</td></tr></table>	F-73w MPX	n	S	D	4	ファイルレジスタFILE 1からのnワード抽出	↑					39
F-73w MPX	n	S	D										
F-73d	<table border="1"><tr><td>F-73d MPX</td><td>n</td><td>S</td><td>D</td></tr></table>	F-73d MPX	n	S	D	4	ファイルレジスタFILE 1からのnダブルワード抽出	↑					40
F-73d MPX	n	S	D										
F-74	<table border="1"><tr><td>F-74 nXFR</td><td>n</td><td>S</td><td>D</td></tr></table>	F-74 nXFR	n	S	D	4	nバイト転送	↑					41
F-74 nXFR	n	S	D										
F-74w	<table border="1"><tr><td>F-74w nXFR</td><td>n</td><td>S</td><td>D</td></tr></table>	F-74w nXFR	n	S	D	4	nワード転送	↑					41
F-74w nXFR	n	S	D										
F-74d	<table border="1"><tr><td>F-74d nXFR</td><td>n</td><td>S</td><td>D</td></tr></table>	F-74d nXFR	n	S	D	4	nダブルワード転送	↑					42
F-74d nXFR	n	S	D										
F-76	<table border="1"><tr><td>F-76 FILR</td><td>S₁</td><td>S₂</td><td>D</td></tr></table>	F-76 FILR	S ₁	S ₂	D	4	nバイト一括転送	↑					43
F-76 FILR	S ₁	S ₂	D										
F-76w	<table border="1"><tr><td>F-76w FILR</td><td>S₁</td><td>S₂</td><td>D</td></tr></table>	F-76w FILR	S ₁	S ₂	D	4	nワード一括転送	↑					44
F-76w FILR	S ₁	S ₂	D										
F-76d	<table border="1"><tr><td>F-76d FILR</td><td>S₁</td><td>S₂</td><td>D</td></tr></table>	F-76d FILR	S ₁	S ₂	D	4	nダブルワード一括転送	↑					45
F-76d FILR	S ₁	S ₂	D										
F-77	<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	4	サムチェックコード生成	↑					46
F-77 CHKC	S ₁	S ₂	D										
F-78	<table border="1"><tr><td>F-78 CHK</td><td>S₁</td><td>S₂</td><td>S₃</td></tr></table>	F-78 CHK	S ₁	S ₂	S ₃	4	データのチェック	↑	0	0	↑↓	0	47
F-78 CHK	S ₁	S ₂	S ₃										
F-79	<table border="1"><tr><td>F-79 SORT</td><td>S₁</td><td>n₁</td><td>n₂</td></tr></table>	F-79 SORT	S ₁	n ₁	n ₂	4	1バイトデータの並べ替え	↑					48
F-79 SORT	S ₁	n ₁	n ₂										
F-79w	<table border="1"><tr><td>F-79w SORT</td><td>S₁</td><td>n₁</td><td>n₂</td></tr></table>	F-79w SORT	S ₁	n ₁	n ₂	4	1ワードデータの並べ替え	↑					49
F-79w SORT	S ₁	n ₁	n ₂										
F-79d	<table border="1"><tr><td>F-79d SORT</td><td>S₁</td><td>n₁</td><td>n₂</td></tr></table>	F-79d SORT	S ₁	n ₁	n ₂	4	レジスタ(2ワード)データの並べ替え	↑					50
F-79d SORT	S ₁	n ₁	n ₂										

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 007357	キャリー 007356	ゼロ 007355	オーバー 007354	
F-80	— F-80 IORF R,S	2	I/Oリフレッシュ	ON	0	↑	↑	↑	13・1
F-82	— F-82 IORF SW	2	特殊I/Oのリフレッシュ	ON	0	↑	↑	↑	2
F-85	— F-85 PRRD n ₁ SW,n ₂ D	4	特殊I/Oからの読出	↑	0	↑	↑	↑	3
F-86	— F-86 PRWR n ₁ D SW,n ₂	4	特殊I/Oへの書込	↑	0	↑	↑	↑	3
F-90	— F-90 REM n	2	リマーク n = 0000 ~ 3777						4
F-91	— F-91 BCD8 n ₁ n ₂ D	4	BCD定数(8桁)の転送	↑					5
F-97	— F-97 DML8 n ₁ n ₂ D	4	10進定数(8桁)の転送	↑					6
F-100	— F-100 ADRS S D	3	間接アドレスの設定	↑					7
F-101	— F-101 SEGM n FILE F D	4	間接アドレスの設定	↑					8
F-102	— F-102 MRD n FILE F D	4	直接指定アドレスのレジスタからの読出 (1バイト)	↑					9
F-102w	— F-102w MRD n FILE F D	4	直接指定アドレスのレジスタからの読出 (1ワード)	↑					9
F-102d	— F-102d MRD n FILE F D	4	直接指定アドレスのレジスタからの読出 (2ワード)	↑					10
F-103	— F-103 MWR S n FILE F	4	直接指定アドレスのレジスタへの書込 (1バイト)	↑					11
F-103w	— F-103w MWR S n FILE F	4	直接指定アドレスのレジスタへの書込 (1ワード)	↑					11
F-103d	— F-103d MWR S n FILE F	4	直接指定アドレスのレジスタへの書込 (2ワード)	↑					12
F-112	— F-112 NCMP S ₁ S ₂ S ₃	4	nバイト一括比較	ON	↑	↑	0	↑	13
F-112w	— F-112w NCMP S ₁ S ₂ S ₃	4	nワード一括比較	ON	↑	↑	0	↑	14
F-112d	— F-112d NCMP S ₁ S ₂ S ₃	4	nダブルワード一括比較	ON	↑	↑	0	↑	15
F-116	— F-116 DIV S ₁ S ₂ D	4	レジスタ(BCD 8桁)とレジスタ(BCD 8桁) の除算(小数部4桁)	↑	0	0	↑	0	16
F-130	— F-130 BIT → S ₁ S ₂	3	ビット抽出(間接指定)	ON	0	↑	0	0	17
F-131	— F-131 BIT → n S	3	ビット抽出(直接指定)	ON	0	↑	0	0	17
F-132	— 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					18
F-133	— 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					18
F-140	— F-140 LABL LBn	2	ラベルの設定 LB0000 ~ LB1377						19
F-141	— F-141 JMP LBn	2	ラベルへジャンプ	ON					20
F-142	— F-142 CALL LBn	2	ラベルをサブルーチンコール	↑					22
F-143	— F-143 RET	1	サブルーチンからのリターン						22
F-144	— F-144 FOR n	2	ループ回数の設定	↑					24
F-145	— F-145 NEXT	1	ループの終了						24
F-146	— F-146 FORR S	2	ループ回数のレジスタ設定	↑					26
F-147	— F-147 EXIT	1	ループの条件終了	OFF					27
F-148	— 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	↑					28
F-149	— F-149 RETC	1	サブルーチンからの条件リターン	OFF					29

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 007357	キャリー 007356	エラー 007355	オーバー 007354	
F-151		3	レジスタ設定ラベルへジャンプ	ON					13・30
F-153		3	BCD(8桁) BIN(32ビット)変換	↑	0	0	↑↓	0	31
F-154		3	BIN(32ビット) BCD(10桁)変換	↑					31
F-155		3	時(4桁)分,秒 秒(BCD 8桁)	↑	0	0	↑↓	0	32
F-156		3	秒(BCD 8桁) 時(4桁)分,秒(BCD)	↑	0	0	↑↓	0	33
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	↑↓	34
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	↑↓	35
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	↑↓	36
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	↑↓	38
F-161d		3	非同期シフトレジスタ(Nダブルワード)	シフト 入力ON	0	↑↓	0	↑↓	39
F-163		2	バイナリ加算(+ 2)カウンタ(1バイト)	↑	↑↓	↑↓	0	↑↓	40
F-163w		2	バイナリ加算(+ 2)カウンタ(1ワード)	↑	↑↓	↑↓	0	↑↓	40
F-163d		2	バイナリ加算(+ 2)カウンタ(2ワード)	↑	↑↓	↑↓	0	↑↓	41
F-164		2	バイナリ減算(- 2)カウンタ(1バイト)	↑	↑↓	↑↓	0	↑↓	42
F-164w		2	バイナリ減算(- 2)カウンタ(1ワード)	↑	↑↓	↑↓	0	↑↓	42
F-164d		2	バイナリ減算(- 2)カウンタ(2ワード)	↑	↑↓	↑↓	0	↑↓	43
F-170		4	データの挿入(1バイト)	↑	0	0	↑↓	0	44
F-170w		4	データの挿入(1ワード)	↑	0	0	↑↓	0	45
F-170d		4	データの挿入(2ワード)	↑	0	0	↑↓	0	46
F-171		4	データの削除(1バイト)	↑	0	0	↑↓	0	47
F-171w		4	データの削除(1ワード)	↑	0	0	↑↓	0	48
F-171d		4	データの削除(2ワード)	↑	0	0	↑↓	0	49
F-172		4	データの検索(1バイト)	↑	↑↓	↑↓	0	0	50
F-172w		4	データの検索(1ワード)	↑	↑↓	↑↓	0	0	51
F-172d		4	データの検索(2ワード)	↑	↑↓	↑↓	0	0	52
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	53
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	54
F-173d		4	データチェンジ(2ワード)	実行入力 ↑	↑↓	↑↓	0	0	55
F-174		3	レジスタ間(1バイト)データ交換	↑					14・1
F-175		3	上位4ビットと下位4ビットの交換	↑					1

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 007357	キャリー 007356	エラー 007355	オーバー 007354	
F-176	$\overline{\text{F-176}} \begin{array}{ c c c } \hline \text{DFRD} & \text{S} & \text{fileN} & \text{D} \\ \hline \end{array}$	4	直接指定アドレスのレジスタからの読出 (256バイト)	↑					14・2
F-177	$\overline{\text{F-177}} \begin{array}{ c c c } \hline \text{DFWR} & \text{S} & \text{D} & \text{fileN} \\ \hline \end{array}$	4	直接指定アドレスのレジスタへの書込 (256バイト)	↓					3
F-180	$\overline{\text{F-180}} \begin{array}{ c c c c } \hline \text{CP>} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 1バイトの比較 (>, リレー出力付)	ON	0	0	0	0	6
F-180w	$\overline{\text{F-180w}} \begin{array}{ c c c c } \hline \text{CP>} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 1ワードの比較 (>, リレー出力付)	ON	0	0	0	0	7
F-180d	$\overline{\text{F-180d}} \begin{array}{ c c c c } \hline \text{CP>} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 2ワードの比較 (>, リレー出力付)	ON	0	0	0	0	8
Fc180	$\overline{\text{Fc180}} \begin{array}{ c c c c } \hline \text{CP>} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	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	9
Fc180w	$\overline{\text{Fc180w}} \begin{array}{ c c c c } \hline \text{CP>} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	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	10
Fc180d	$\overline{\text{Fc180d}} \begin{array}{ c c c c } \hline \text{CP>} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	4	レジスタと定数 2ワードの比較 (>, リレー出力付)	ON	0	0	0	0	11
F-181	$\overline{\text{F-181}} \begin{array}{ c c c c } \hline \text{CP<} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 1バイトの比較 (<, リレー出力付)	ON	0	0	0	0	6
F-181w	$\overline{\text{F-181w}} \begin{array}{ c c c c } \hline \text{CP<} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 1ワードの比較 (<, リレー出力付)	ON	0	0	0	0	7
F-181d	$\overline{\text{F-181d}} \begin{array}{ c c c c } \hline \text{CP<} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 2ワードの比較 (<, リレー出力付)	ON	0	0	0	0	8
Fc181	$\overline{\text{Fc181}} \begin{array}{ c c c c } \hline \text{CP<} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	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	9
Fc181w	$\overline{\text{Fc181w}} \begin{array}{ c c c c } \hline \text{CP<} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	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	10
Fc181d	$\overline{\text{Fc181d}} \begin{array}{ c c c c } \hline \text{CP<} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	4	レジスタと定数 2ワードの比較 (<, リレー出力付)	ON	0	0	0	0	11
F-182	$\overline{\text{F-182}} \begin{array}{ c c c c } \hline \text{CP=} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 1バイトの比較 (=, リレー出力付)	ON	0	0	0	0	6
F-182w	$\overline{\text{F-182w}} \begin{array}{ c c c c } \hline \text{CP=} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 1ワードの比較 (=, リレー出力付)	ON	0	0	0	0	7
F-182d	$\overline{\text{F-182d}} \begin{array}{ c c c c } \hline \text{CP=} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 2ワードの比較 (=, リレー出力付)	ON	0	0	0	0	8
Fc182	$\overline{\text{Fc182}} \begin{array}{ c c c c } \hline \text{CP=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	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	9
Fc182w	$\overline{\text{Fc182w}} \begin{array}{ c c c c } \hline \text{CP=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	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	10
Fc182d	$\overline{\text{Fc182d}} \begin{array}{ c c c c } \hline \text{CP=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	4	レジスタと定数 2ワードの比較 (=, リレー出力付)	ON	0	0	0	0	11
F-183	$\overline{\text{F-183}} \begin{array}{ c c c c } \hline \text{CP>=} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 1バイトの比較 (>=, リレー出力付)	ON	0	0	0	0	6
F-183w	$\overline{\text{F-183w}} \begin{array}{ c c c c } \hline \text{CP>=} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 1ワードの比較 (>=, リレー出力付)	ON	0	0	0	0	7
F-183d	$\overline{\text{F-183d}} \begin{array}{ c c c c } \hline \text{CP>=} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 2ワードの比較 (>=, リレー出力付)	ON	0	0	0	0	8
Fc183	$\overline{\text{Fc183}} \begin{array}{ c c c c } \hline \text{CP>=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	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	9
Fc183w	$\overline{\text{Fc183w}} \begin{array}{ c c c c } \hline \text{CP>=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	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	10
Fc183d	$\overline{\text{Fc183d}} \begin{array}{ c c c c } \hline \text{CP>=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	4	レジスタと定数 2ワードの比較 (>=, リレー出力付)	ON	0	0	0	0	11
F-184	$\overline{\text{F-184}} \begin{array}{ c c c c } \hline \text{CP<=} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 1バイトの比較 (<=, リレー出力付)	ON	0	0	0	0	6
F-184w	$\overline{\text{F-184w}} \begin{array}{ c c c c } \hline \text{CP<=} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 1ワードの比較 (<=, リレー出力付)	ON	0	0	0	0	7
F-184d	$\overline{\text{F-184d}} \begin{array}{ c c c c } \hline \text{CP<=} & \text{S}_1 & \text{S}_2 & \text{BIT} \\ \hline \end{array}$	4	レジスタ間 2ワードの比較 (<=, リレー出力付)	ON	0	0	0	0	8
Fc184	$\overline{\text{Fc184}} \begin{array}{ c c c c } \hline \text{CP<=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	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	9
Fc184w	$\overline{\text{Fc184w}} \begin{array}{ c c c c } \hline \text{CP<=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	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	10
Fc184d	$\overline{\text{Fc184d}} \begin{array}{ c c c c } \hline \text{CP<=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	4	レジスタと定数 2ワードの比較 (<=, リレー出力付)	ON	0	0	0	0	11

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 007357	キャリー 007356	エラー 007355	オーバー 007354	
F-185	$\overline{\text{F-185 CP<>}}$ 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	14・6
F-185w	$\overline{\text{F-185w CP<>}}$ 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	7
F-185d	$\overline{\text{F-185d CP<>}}$ S ₁ S ₂ BIT	4	レジスタ間 2ワードの比較、リレー出力付)	ON	0	0	0	0	8
Fc185	$\overline{\text{Fc185 CP<>}}$ 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	9
Fc185w	$\overline{\text{Fc185w CP<>}}$ 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	10
Fc185d	$\overline{\text{Fc185d CP<>}}$ S n BIT	4	レジスタと定数 2ワードの比較、リレー出力付)	ON	0	0	0	0	11
F-202	$\overline{\text{F-202 OPCH UN,C, ST}}$ file N n	4	オープンチャンネル(局番 8 進定数設定)	ON					12
F-203	$\overline{\text{F-203 OPCH UN,C, ST}}$ file N n	4	オープンチャンネル(局番16進定数設定)	ON					12
F-204	$\overline{\text{F-204 SEND}}$ n S	3	送信命令	↑	↑↓	↑↓	↑↓	↑↓	13
F-205	$\overline{\text{F-205 RCV}}$ n D	3	受信命令	↑	↑↓	↑↓	↑↓	↑↓	14
F-206	$\overline{\text{F-206 EOP1 UN1,CH ST1 UN2}}$	4	オープンチャンネル 1(階層通信設定)	ON					15
F-207	$\overline{\text{F-207 EOP2 ST2 file N n}}$	4	オープンチャンネル 2(階層通信設定)	ON					15
F-210	$\overline{\text{F-210 ADD}}$ 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	↑↓	16
F-210w	$\overline{\text{F-210w ADD}}$ 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	↑↓	17
F-210d	$\overline{\text{F-210d ADD}}$ S ₁ S ₂ D	4	レジスタ間のバイナリ加算 (32ビット+32ビット)	↑	↑↓	↑↓	0	↑↓	18
Fc210	$\overline{\text{Fc210 ADD}}$ S ₁ n D	4	レジスタと定数のバイナリ加算 (8ビット+8ビット)	↑	↑↓	↑↓	0	↑↓	19
Fc210w	$\overline{\text{Fc210w ADD}}$ S ₁ n D	4	レジスタと定数のバイナリ加算 (16ビット+16ビット)	↑	↑↓	↑↓	0	↑↓	19
Fc210d	$\overline{\text{Fc210d ADD}}$ S ₁ n D	4	レジスタと定数のバイナリ加算 (32ビット+16ビット)	↑	↑↓	↑↓	0	↑↓	20
F-211	$\overline{\text{F-211 SUB}}$ S ₁ S ₂ D	4	レジスタ間のバイナリ減算 (8ビット-8ビット)	↑	↑↓	↑↓	0	↑↓	21
F-211w	$\overline{\text{F-211w SUB}}$ S ₁ S ₂ D	4	レジスタ間のバイナリ減算 (16ビット-16ビット)	↑	↑↓	↑↓	0	↑↓	22
F-211d	$\overline{\text{F-211d SUB}}$ S ₁ S ₂ D	4	レジスタ間のバイナリ減算 (32ビット-32ビット)	↑	↑↓	↑↓	0	↑↓	22
Fc211	$\overline{\text{Fc211 SUB}}$ S ₁ n D	4	レジスタと定数のバイナリ減算 (8ビット-8ビット)	↑	↑↓	↑↓	0	↑↓	23
Fc211w	$\overline{\text{Fc211w SUB}}$ S ₁ n D	4	レジスタと定数のバイナリ減算 (16ビット-16ビット)	↑	↑↓	↑↓	0	↑↓	23
Fc211d	$\overline{\text{Fc211d SUB}}$ S ₁ n D	4	レジスタと定数のバイナリ減算 (32ビット-16ビット)	↑	↑↓	↑↓	0	↑↓	24
F-212	$\overline{\text{F-212 WNDW}}$ S ₁ S ₂ S ₃	4	ウィンドウコンパレータ (1バイトレジスタ間)	ON	↑↓	↑↓	↑↓	↑↓	25
F-212w	$\overline{\text{F-212w WNDW}}$ S ₁ S ₂ S ₃	4	ウィンドウコンパレータ (1ワードレジスタ間)	ON	↑↓	↑↓	↑↓	↑↓	26
F-212d	$\overline{\text{F-212d WNDW}}$ S ₁ S ₂ S ₃	4	ウィンドウコンパレータ (2ワードレジスタ間)	ON	↑↓	↑↓	↑↓	↑↓	27
Fc212	$\overline{\text{Fc212 WNDW}}$ S ₁ n ₁ n ₂	4	ウィンドウコンパレータ (1バイト8進定数間)	ON	↑↓	↑↓	↑↓	↑↓	28
Fc212w	$\overline{\text{Fc212w WNDW}}$ S ₁ n ₁ n ₂	4	ウィンドウコンパレータ (1ワード8進定数間)	ON	↑↓	↑↓	↑↓	↑↓	28
Fc212d	$\overline{\text{Fc212d WNDW}}$ S ₁ n ₁ n ₂	4	ウィンドウコンパレータ (2ワード8進定数間)	ON	↑↓	↑↓	↑↓	↑↓	29
Fx212	$\overline{\text{Fx212 WNDW}}$ S ₁ n ₁ n ₂	4	ウィンドウコンパレータ (1バイト16進定数間)	ON	↑↓	↑↓	↑↓	↑↓	30
Fx212w	$\overline{\text{Fx212w WNDW}}$ S ₁ n ₁ n ₂	4	ウィンドウコンパレータ (1ワード16進定数間)	ON	↑↓	↑↓	↑↓	↑↓	30
Fx212d	$\overline{\text{Fx212d WNDW}}$ S ₁ n ₁ n ₂	4	ウィンドウコンパレータ (2ワード16進定数間)	ON	↑↓	↑↓	↑↓	↑↓	31

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 007357	キャリー 007356	エラー 007355	オーバー 007354	
F-215	$\overline{\text{F-215}} \begin{array}{ c c c } \hline \text{MUL} & S_1 & S_2 & D \\ \hline \end{array}$	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	14・32
F-215w	$\overline{\text{F-215w}} \begin{array}{ c c c } \hline \text{MUL} & S_1 & S_2 & D \\ \hline \end{array}$	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	32
F-215d	$\overline{\text{F-215d}} \begin{array}{ c c c } \hline \text{MUL} & S_1 & S_2 & D \\ \hline \end{array}$	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	33
Fc215	$\overline{\text{Fc215}} \begin{array}{ c c c } \hline \text{MUL} & S_1 & n & D \\ \hline \end{array}$	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	34
Fc215w	$\overline{\text{Fc215w}} \begin{array}{ c c c } \hline \text{MUL} & S_1 & n & D \\ \hline \end{array}$	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	34
Fc215d	$\overline{\text{Fc215d}} \begin{array}{ c c c } \hline \text{MUL} & S_1 & n & D \\ \hline \end{array}$	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	35
F-216	$\overline{\text{F-216}} \begin{array}{ c c c } \hline \text{DIV} & S_1 & S_2 & D \\ \hline \end{array}$	4	レジスタ間のバイナリ除算 (8ビット÷8ビット)	↑	0	0	↑↓	0	36
F-216w	$\overline{\text{F-216w}} \begin{array}{ c c c } \hline \text{DIV} & S_1 & S_2 & D \\ \hline \end{array}$	4	レジスタ間のバイナリ除算 (15ビット÷15ビット)	↑	0	0	↑↓	0	37
F-216d	$\overline{\text{F-216d}} \begin{array}{ c c c } \hline \text{DIV} & S_1 & S_2 & D \\ \hline \end{array}$	4	レジスタ間のバイナリ除算 (31ビット÷31ビット)	↑	0	0	↑↓	0	38
Fc216	$\overline{\text{Fc216}} \begin{array}{ c c c } \hline \text{DIV} & S_1 & n & D \\ \hline \end{array}$	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	39
Fc216w	$\overline{\text{Fc216w}} \begin{array}{ c c c } \hline \text{DIV} & S_1 & n & D \\ \hline \end{array}$	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	39
Fc216d	$\overline{\text{Fc216d}} \begin{array}{ c c c } \hline \text{DIV} & S_1 & n & D \\ \hline \end{array}$	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	40
F-231	$\overline{\text{F-231}} \begin{array}{ c } \hline \text{MCRN} \\ \hline \end{array}$	1	マスターコントロール ネスティング リセット						41
F-242	$\overline{\text{F-242}} \begin{array}{ c } \hline \text{JCRN} \\ \hline \end{array}$	1	ジャンプコントロール ネスティング リセット						42
F-252	$\overline{\text{F-252}} \begin{array}{ c c c } \hline \text{ASC} & S & n & D \\ \hline \end{array}$	4	HEX ASCII変換	↑					43
F-253	$\overline{\text{F-253}} \begin{array}{ c c c } \hline \text{HEX} & S & n & D \\ \hline \end{array}$	4	ASCII HEX変換	↑	0	0	↑↓	0	44
F-260	$\overline{\text{F-260}} \begin{array}{ c c c c } \hline \text{RTMR} & S & D & \text{BIT} \\ \hline \end{array}$	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	45
Fc260	$\overline{\text{Fc260}} \begin{array}{ c c c c } \hline \text{RTMR} & n & D & \text{BIT} \\ \hline \end{array}$	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	46
F-261	$\overline{\text{F-261}} \begin{array}{ c c c c } \hline \text{RCNT} & S & D & \text{BIT} \\ \hline \end{array}$	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	47
Fc261	$\overline{\text{Fc261}} \begin{array}{ c c c c } \hline \text{RCNT} & n & D & \text{BIT} \\ \hline \end{array}$	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	48
F-263	$\overline{\text{F-263}} \begin{array}{ c c } \hline \text{INC4} & D \\ \hline \end{array}$	2	バイナリ加算(+4)カウンタ(1バイト)	↑	↑↓	↑↓	0	↑↓	49
F-263w	$\overline{\text{F-263w}} \begin{array}{ c c } \hline \text{INC4} & D \\ \hline \end{array}$	2	バイナリ加算(+4)カウンタ(1ワード)	↑	↑↓	↑↓	0	↑↓	49
F-263d	$\overline{\text{F-263d}} \begin{array}{ c c } \hline \text{INC4} & D \\ \hline \end{array}$	2	バイナリ加算(+4)カウンタ(2ワード)	↑	↑↓	↑↓	0	↑↓	50
F-264	$\overline{\text{F-264}} \begin{array}{ c c } \hline \text{DEC4} & D \\ \hline \end{array}$	2	バイナリ減算(-4)カウンタ(1バイト)	↑	↑↓	↑↓	0	↑↓	51
F-264w	$\overline{\text{F-264w}} \begin{array}{ c c } \hline \text{DEC4} & D \\ \hline \end{array}$	2	バイナリ減算(-4)カウンタ(1ワード)	↑	↑↓	↑↓	0	↑↓	51
F-264d	$\overline{\text{F-264d}} \begin{array}{ c c } \hline \text{DEC4} & D \\ \hline \end{array}$	2	バイナリ減算(-4)カウンタ(2ワード)	↑	↑↓	↑↓	0	↑↓	52

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 007357	キャリー 007356	エラー 007355	オーバー 007354					
F-300	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-300 XFER</td><td style="padding: 2px;">S</td><td style="padding: 2px;">D</td></tr></table>	F-300 XFER	S	D	3	1バイトデータの転送	ON					14・53	
F-300 XFER	S	D											
F-300w	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-300w XFER</td><td style="padding: 2px;">S</td><td style="padding: 2px;">D</td></tr></table>	F-300w XFER	S	D	3	1ワードデータの転送	ON					53	
F-300w XFER	S	D											
F-300d	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-300d XFER</td><td style="padding: 2px;">S</td><td style="padding: 2px;">D</td></tr></table>	F-300d XFER	S	D	3	2ワードデータの転送	ON					54	
F-300d XFER	S	D											
F-310	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-310 SADD</td><td style="padding: 2px;">S₁</td><td style="padding: 2px;">S₂</td><td style="padding: 2px;">D</td></tr></table>	F-310 SADD	S ₁	S ₂	D	4	レジスタ間の符号付きバイナリ加算 (31ビット+31ビット)	↑	↕	↕	↕	↕	55
F-310 SADD	S ₁	S ₂	D										
F-311	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-311 SSUB</td><td style="padding: 2px;">S₁</td><td style="padding: 2px;">S₂</td><td style="padding: 2px;">D</td></tr></table>	F-311 SSUB	S ₁	S ₂	D	4	レジスタ間の符号付きバイナリ減算 (31ビット-31ビット)	↑	↕	↕	↕	↕	56
F-311 SSUB	S ₁	S ₂	D										
F-315	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-315 SMUL</td><td style="padding: 2px;">S₁</td><td style="padding: 2px;">S₂</td><td style="padding: 2px;">D</td></tr></table>	F-315 SMUL	S ₁	S ₂	D	4	レジスタ間の符号付きバイナリ乗算 (31ビット×31ビット)	↑	0	0	0	0	57
F-315 SMUL	S ₁	S ₂	D										
F-316	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-316 SDIV</td><td style="padding: 2px;">S₁</td><td style="padding: 2px;">S₂</td><td style="padding: 2px;">D</td></tr></table>	F-316 SDIV	S ₁	S ₂	D	4	レジスタ間の符号付きバイナリ除算 (31ビット÷31ビット)	↑	0	0	↕	0	58
F-316 SDIV	S ₁	S ₂	D										
F-403	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-403 LOG</td></tr></table>	F-403 LOG	1	ロギング命令	↑					58			
F-403 LOG													
NOP		1	無効命令										

〔 2 〕 動作による分類

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 10・1 F-300 14・53	
		1 ワード	F-00w 10・1 F-300w 14・53	
		2 ワード	F-00d 10・2 F-300d 14・54	
		n バイト	F-70 12・30	
		n ワード	F-70w 31	
		nダブルワード	F-70d 32	
		n バイト (間接指定)	F-76 43	
		n ワード (間接指定)	F-76w 44	
		nダブルワード (間接指定)	F-76d 45	
		n バイト (同一データ)	F-74 41	
		n ワード (同一データ)	F-74w 41	
		nダブルワード (同一データ)	F-74d 42	
		BCD定数の転送	2 桁	F-01 10・3
			4 桁	F-01w 3
	8 桁		F-01d 4 F-91 13・5	
	10進定数の転送	1 バイト	F-07 10・17	
		1 ワード	F-07w 17	
		2 ワード	F-07d 18	
		8 桁	F-97 13・6	
	8進定数の転送	1 バイト	F-08 10・19	
		1 ワード	F-08w 19	
		2 ワード	F-08d 20	
		n バイト	F-71 12・33	
		n ワード	F-71w 34	
	分 配	nダブルワード	F-71d 34	
		1 バイト	F-05 10・11	
		1 ワード	F-05w 12	
		2 ワード	F-05d 13	
	抽 出	n バイト (ファイル1)	F-72 12・35	
		n ワード (ファイル1)	F-72w 36	
		nダブルワード (ファイル1)	F-72d 37	
		1 バイト	F-06 10・14	
		1 ワード	F-06w 15	
桁 転 送	2 ワード	F-06d 16		
	n バイト (ファイル1)	F-73 12・38		
	n ワード (ファイル1)	F-73w 39		
	nダブルワード (ファイル1)	F-73d 40		
	4 ビット	F-69 29		
	1 バイト	F-102 13・9		
ファイルの読出	1 ワード	F-102w 9		
	2 ワード	F-102d 10		
	256バイト	F-176 14・2		
ファイルへの書込	1 バイト	F-103 13・11		
	1 ワード	F-103w 11		
	2 ワード	F-103d 12		
	256バイト	F-177 14・3		

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 F-10 10・23 4 桁 + 4 桁 F-10w 24 8 桁 + 8 桁 F-10d 25	
		レジスタと定数	2 桁 + 2 桁 Fc10 26 4 桁 + 4 桁 Fc10w 27 8 桁 + 4 桁 Fc10d 28	
		BCD減算	レジスタ間	2 桁 - 2 桁 F-11 29 4 桁 - 4 桁 F-11w 30 8 桁 - 8 桁 F-11d 31
			レジスタと定数	2 桁 - 2 桁 Fc11 32 4 桁 - 4 桁 Fc11w 33 8 桁 - 4 桁 Fc11d 34
	BCD乗算		レジスタ間	4 桁 × 4 桁 F-15 11・1 8 桁 × 8 桁 F-15d 2
			レジスタと定数	4 桁 × 3 桁 Fc15 3 8 桁 × 4 桁 Fc15d 4
		BCD除算	レジスタ間	4 桁 ÷ 2 桁 F-16 5 8 桁 ÷ 8 桁 F-16d 6 8 桁 ÷ 8 桁 小数部 4 桁 F-116 13・16
	レジスタと定数		4 桁 ÷ 2 桁 Fc16 11・7 8 桁 ÷ 4 桁 Fc16d 8	
	バイナリ加算		レジスタ間	8 ビット + 8 ビット F-210 14・16 16 ビット + 16 ビット F-210w 17 32 ビット + 32 ビット F-210d 18
		レジスタと定数	8 ビット + 8 ビット Fc210 19 16 ビット + 16 ビット Fc210w 19 32 ビット + 16 ビット Fc210d 20	
		レジスタ間 (符号付)	31 ビット + 31 ビット F-310 55	
		バイナリ減算	レジスタ間	8 ビット - 8 ビット F-211 21 16 ビット - 16 ビット F-211w 22 32 ビット - 32 ビット F-211d 22
			レジスタと定数	8 ビット - 8 ビット Fc211 23 16 ビット - 16 ビット Fc211w 23 32 ビット - 16 ビット Fc211d 24
	レジスタ間 (符号付)		31 ビット - 31 ビット F-311 56	
	バイナリ乗算		レジスタ間	8 ビット × 8 ビット F-215 32 16 ビット × 16 ビット F-215w 32 32 ビット × 32 ビット F-215d 33
		レジスタと定数	8 ビット × 8 ビット Fc215 34 16 ビット × 16 ビット Fc215w 34 32 ビット × 16 ビット Fc215d 35	
		レジスタ間 (符号付)	31 ビット × 31 ビット F-315 57	
		バイナリ除算	レジスタ間	8 ビット ÷ 8 ビット F-216 36 15 ビット ÷ 15 ビット F-216w 37 31 ビット ÷ 31 ビット F-216d 38
			レジスタと定数	8 ビット ÷ 8 ビット Fc216 39 15 ビット ÷ 15 ビット Fc216w 39 31 ビット ÷ 15 ビット Fc216d 40
	レジスタ間 (符号付)		31 ビット ÷ 31 ビット F-316 58	

分		類	命令語	参照ページ	
論 理 演 算 命 令	論理積	レジスタ間	8ビット	F-13	10・42
			16ビット	F-13w	42
			32ビット	F-13d	43
		レジスタと 8進定数	8ビット	Fc13	44
			16ビット	Fc13w	44
			32ビット	Fc13d	45
		レジスタと 16進定数	8ビット	Fx13	46
			16ビット	Fx13w	46
			32ビット	Fx13d	47
	論理和	レジスタ間	8ビット	F-14	48
			16ビット	F-14w	48
			32ビット	F-14d	49
		レジスタと 8進定数	8ビット	Fc14	50
			16ビット	Fc14w	50
			32ビット	Fc14d	51
		レジスタと 16進定数	8ビット	Fx14	52
			16ビット	Fx14w	52
			32ビット	Fx14d	53
	一致	レジスタ間	8ビット	F-17	11・9
			16ビット	F-17w	9
			32ビット	F-17d	10
		レジスタと 8進定数	8ビット	Fc17	11
			16ビット	Fc17w	11
			32ビット	Fc17d	12
		レジスタと 16進定数	8ビット	Fx17	13
			16ビット	Fx17w	13
			32ビット	Fx17d	14
	排他的 論理和	レジスタ間	8ビット	F-18	15
			16ビット	F-18w	15
			32ビット	F-18d	16
		レジスタと 8進定数	8ビット	Fc18	17
			16ビット	Fc18w	17
			32ビット	Fc18d	18
レジスタと 16進定数		8ビット	Fx18	19	
		16ビット	Fx18w	19	
		32ビット	Fx18d	20	
反 転	8ビット	F-09	10・21		
	16ビット	F-09w	21		
	32ビット	F-09d	22		

分		類	命令語	参照ページ		
比 較 命 令	比 較	レジスタ間	1バイト	F-12	10・35	
			1ワード	F-12w	36	
			2ワード	F-12d	37	
			nバイト	F-112	13・13	
			nワード	F-112w	14	
			nダブルワード	F-112d	15	
		レジスタと 8進定数	1バイト	Fc12	10・38	
			1ワード	Fc12w	38	
			2ワード	Fc12d	39	
		レジスタと 16進定数	1バイト	Fx12	40	
			1ワード	Fx12w	40	
			2ワード	Fx12d	41	
		>	レジスタ間	1バイト	F-180	14・6
				1ワード	F-180w	7
				2ワード	F-180d	8
	レジスタと 8進定数		1バイト	Fc180	9	
			1ワード	Fc180w	10	
			2ワード	Fc180d	11	
	<		レジスタ間	1バイト	F-181	6
				1ワード	F-181w	7
				2ワード	F-181d	8
			レジスタと 8進定数	1バイト	Fc181	9
				1ワード	Fc181w	10
				2ワード	Fc181d	11
	=	レジスタ間	1バイト	F-182	6	
			1ワード	F-182w	7	
			2ワード	F-182d	8	
		レジスタと 8進定数	1バイト	Fc182	9	
			1ワード	Fc182w	10	
			2ワード	Fc182d	11	
	(リレー出力付)	レジスタ間	1バイト	F-183	6	
			1ワード	F-183w	7	
			2ワード	F-183d	8	
1バイト			Fc183	9		
1ワード			Fc183w	10		
2ワード			Fc183d	11		
レジスタと 8進定数		1バイト	F-184	6		
		1ワード	F-184w	7		
		2ワード	F-184d	8		
		1バイト	Fc184	9		
		1ワード	Fc184w	10		
		2ワード	Fc184d	11		
<	レジスタ間	1バイト	F-185	6		
		1ワード	F-185w	7		
		2ワード	F-185d	8		
	レジスタと 8進定数	1バイト	Fc185	9		
		1ワード	Fc185w	10		
		2ワード	Fc185d	11		
ウィ ンド ウ コン パ レー タ	レジスタ間	1バイト	F-212	25		
		1ワード	F-212w	26		
		2ワード	F-212d	27		
	レジスタと 8進定数	1バイト	Fc212	28		
		1ワード	Fc212w	28		
		2ワード	Fc212d	29		
	レジスタと 16進定数	1バイト	Fx212	30		
		1ワード	Fx212w	30		
		2ワード	Fx212d	31		

分 類		命 令 語	参 照 ペ ージ	
変 換 命 令	BCD BIN変換	2桁 8ビット	F-03 10・7	
			F-03w 7	
		4桁 16ビット	F-53 12・3	
			F-03d 10・8	
			F-153 13・31	
	BIN BCD変換	8ビット 2桁	F-04 10・9	
			F-04w 9	
		16ビット 6桁	F-54 12・3	
			F-04d 10・10	
		F-154 13・31		
	HEX ASCII変換		F-252	14・43
	ASCII HEX変換		F-253	44
	時・分・秒 秒変換		F-155	13・32
	秒 時・分・秒変換		F-156	13・33
	4 16デコーダ		F-50	12・1
	16 4エンコーダ		F-51	1
	7 SEGデコーダ		F-52	2
	10の補数	2桁	F-56	5
		4桁	F-56w	5
		8桁	F-56d	6
2の補数	8ビット	F-57	7	
	16ビット	F-57w	7	
	32ビット	F-57d	8	
ONビット数の合計		F-58	9	
極座標変換		F-28	11・28	
直交座標変換		F-29	11・29	
交 換 命 令	データの交換	1バイト	F-02 10・5	
		1ワード	F-02w 5	
		2ワード	F-02d 6	
		nバイト	F-174 14・1	
	上位4ビットと 下位4ビットの 交換	1バイト	F-55	12・4
		nバイト	F-175	14・1

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 13・44	
		1ワード	F-170w 45	
		2ワード	F-170d 46	
	データの削除	1バイト	F-171 47	
		1ワード	F-171w 48	
		2ワード	F-171d 49	
	データの検索	1バイト	F-172 50	
		1ワード	F-172w 51	
		2ワード	F-172d 52	
	データの変更	1バイト	F-173 53	
		1ワード	F-173w 54	
		2ワード	F-173d 55	
	データの並び替え	1バイト	F-79 12・48	
		1ワード	F-79w 49	
		2ワード	F-79d 50	
	SIN関数		F-22	11・22
	COS関数		F-23	23
	TAN関数		F-24	24
ASIN関数		F-25	25	
ACOS関数		F-26	26	
ATAN関数		F-27	27	
ビット反転		F-43	44	
ON時微分		F-44	45	
OFF時微分		F-45	46	
セットコイル		F-32	33	
リセットコイル		F-33	34	
ビ ッ ト 処 理 命 令	ビット抽出	間接指定	F-130 13・17	
		直接指定	F-131 17	
	ビットセット /リセット	間接指定	F-132 18	
		直接指定	F-133 18	

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	12・17	
		4桁	F-62w	18	
		8桁	F-62d	19	
	BCD加算 カウンタ	2桁	F-65	24	
		4桁	F-65w	24	
		8桁	F-65d	25	
	BCD減算 カウンタ	2桁	F-66	26	
		4桁	F-66w	26	
		8桁	F-66d	27	
	バイナリ 加算カウ ンタ	+ 1	1バイト	F-63	20
			1ワード	F-63w	20
			2ワード	F-63d	21
		+ 2	1バイト	F-163	13・40
			1ワード	F-163w	40
			2ワード	F-163d	41
	+ 4	1バイト	F-263	14・49	
		1ワード	F-263w	49	
		2ワード	F-263d	50	
	バイナリ 減算カウ ンタ	- 1	1バイト	F-64	12・22
			1ワード	F-64w	22
2ワード			F-64d	23	
- 2		1バイト	F-164	13・42	
		1ワード	F-164w	42	
		2ワード	F-164d	43	
- 4	1バイト	F-264	14・51		
	1ワード	F-264w	51		
	2ワード	F-264d	52		
拡張 タイマ	減算タイマ (設定値、レジスタ指定)		F-260	45	
	減算タイマ (定数、レジスタ指定)		Fc260	46	
拡張 カウンタ	減算カウンタ (設定値、レジスタ指定)		F-261	47	
	減算カウンタ (定数、レジスタ指定)		Fc261	48	
シフト 命令	両方向シフト レジスタ	8ビット	F-60	12・10	
		16ビット	F-60w	12	
		32ビット	F-60d	13	
		nビット (レジスタ指定)	F-160	13・34	
		nビット (定数指定)	Fc160	35	
	非同期両方向 シフトレジスタ	1バイト	F-61	12・14	
		1ワード	F-61w	15	
		2ワード	F-61d	16	
		nバイト	F-161	13・36	
		nワード	F-161w	38	
	nダブルワード	F-161d	39		
桁シフト(上位シフト)		F-67	12・28		
桁シフト(下位シフト)		F-68	28		

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	11・30
	マスターコントロールのリセット		F-31	30
	マスターコントロール ネスティングリセット		F-231	14・41
	ジャンプコントロールのセット		F-41	11・42
	ジャンプコントロールのリセット		F-42	42
	ジャンプコントロール ネスティングリセット		F-242	14・42
	レベル演算条件のセット		F-47	11・47
	レベル演算条件のリセット		F-48	47
	エンド	無条件エンド	F-40	41
		条件エンド	F-49	48
分岐 命令	ラベル		F-140	13・19
	ジャンプ	直接指定	F-141	20
		間接指定	F-151	30
	サブルーチン コール	直接指定	F-142	22
		間接指定	F-148	28
	サブルーチン からのリターン	無条件リターン	F-143	22
		条件リターン	F-149	29
ループ回数 の設定	直接指定	F-144	24	
	間接指定	F-146	26	
ループの終了		F-145	24	
ループの強制終了		F-147	27	

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	11・36	
	時計現在値との比較 (指定リレーのリセット)	F-35	37	
	時間の加算	F-36	38	
	時間の減算	F-37	39	
	時計現在値の転送	F-38	40	
通 信 命 令	オープンチャンネル命令	F-202 F-203	14・12	
	オープンチャンネル1 (階層通信設定)	F-206	15	
	オープンチャンネル2 (階層通信設定)	F-207	15	
	送信命令	F-204	14・13	
	受信命令	F-205	14	
そ の 他 の 命 令	メンテナンスディスプレイ (MD)	F-20	11・21	
	平方根	F-21	22	
	データのサムチェック コードの生成	F-77	12・46	
	データのチェック	F-78	47	
	I/Oリフレッシュ	F-80	13・1	
	特殊I/O ユニット	リフレッシュ	F-82	2
		読出	F-85	3
		書込	F-86	3
	間接アドレスの設定	F-100	7	
		F-101	8	
リマーク (コメント識別用命令)	F-90 (REM)	4		
ロギング命令	F-403	14・58		

7 - 3 命令語の処理時間

基本命令、応用命令の各処理時間を列記します。

〔 1 〕 基本命令の処理時間

命令語	処理時間(μs)
STR	0.033
STR NOT	0.033
AND	0.033
AND NOT	0.033
OR	0.033
OR NOT	0.033
AND STR	0.033
OR STR	0.033
OUT	0.066
OUT NOT	0.066
SET	0.066
RST	0.066

命令語	処理時間(μs)
STR POS	2.19
STR NEG	2.19
AND POS	2.31
AND NEG	2.13
OR POS	2.25
OR NEG	2.07
OUT POS	2.44
OUT NEG	2.44
PUSH	1.12
POP	1.12
MRD	1.12

命令語	処理時間(μs)
TMR	2.44
DTMR(BCD)	2.64
DTMR(BIN)	2.64
UTMR(BCD)	2.64
UTMR(BIN)	2.64
CNT	2.99
DCNT(BCD)	3.37
DCNT(BIN)	3.37
UCNT(BCD)	3.37
UCNT(BIN)	3.37
MD	1.05
NOP	0.03

留意点

- ・基本命令、応用命令でインデックス修飾機能(9・17ページ)を使用すると、その命令語の処理時間が長くなります。

〔 2 〕 応用命令の処理時間

命令語	処理時間(μ s)	
	実行時	非実行時
F-00	1.12	0.39
F-00w	1.33	0.39
F-00d	1.76	0.39
F-01	0.94	0.39
F-01w	1.10	0.39
F-01d	1.42	0.39
F-02	1.37	0.39
F-02w	1.75	0.39
F-02d	2.59	0.39
F-03	1.95	0.39
F-03w	2.74	0.39
F-03d	4.51	0.39
F-04	1.70	0.39
F-04w	3.72	0.39
F-04d	15.74	0.39
F-05	1.40	0.39
F-05w	1.70	0.39
F-05d	2.19	0.39
F-06	1.40	0.39
F-06w	1.70	0.39
F-06d	2.19	0.39
F-07	0.97	0.39
F-07w	1.09	0.39
F-07d	1.40	0.39
F-08	0.97	0.39
F-08w	1.09	0.39
F-08d	1.40	0.39
F-09	1.22	0.39
F-09w	1.52	0.39
F-09d	2.19	0.39
F-10	3.05	0.39
F-10w	4.75	0.39
F-10d	10.80	0.39
Fc10	2.74	0.39
Fc10w	3.96	0.39
Fc10d	9.52	0.39
F-11	3.17	0.39
F-11w	4.57	0.39
F-11d	9.88	0.39
Fc11	2.86	0.39
Fc11w	3.96	0.39
Fc11d	8.60	0.39
F-12	1.58	0.67
F-12w	2.01	0.67
F-12d	2.68	0.67
Fc12	1.40	0.67
Fc12w	1.70	0.67
Fc12d	1.95	0.67
Fx12	1.40	0.67
Fx12w	1.70	0.67
Fx12d	1.95	0.67
F-13	1.38	0.39
F-13w	1.84	0.39
F-13d	2.74	0.39

命令語	処理時間(μ s)	
	実行時	非実行時
Fc13	1.20	0.39
Fc13w	1.51	0.39
Fc13d	2.12	0.39
Fx13	1.20	0.39
Fx13w	1.51	0.39
Fx13d	2.12	0.39
F-14	1.38	0.39
F-14w	1.84	0.39
F-14d	2.74	0.39
Fc14	1.20	0.39
Fc14w	1.51	0.39
Fc14d	2.12	0.39
Fx14	1.20	0.39
Fx14w	1.51	0.39
Fx14d	2.12	0.39
F-15	18.00	0.39
F-15d	69.33	0.39
Fc15	17.33	0.39
Fc15d	67.87	0.39
F-16	8.97	0.39
F-16d	37.10	0.39
Fc16	8.66	0.39
Fc16d	36.13	0.39
F-17	1.43	0.39
F-17w	1.90	0.39
F-17d	2.88	0.39
Fc17	1.23	0.39
Fc17w	1.57	0.39
Fc17d	2.24	0.39
Fx17	1.23	0.39
Fx17w	1.57	0.39
Fx17d	2.24	0.39
F-18	1.38	0.39
F-18w	1.84	0.39
F-18d	2.76	0.39
Fc18	1.20	0.39
Fc18w	1.51	0.39
Fc18d	2.12	0.39
Fx18	1.20	0.39
Fx18w	1.51	0.39
Fx18d	2.12	0.39
F-20	1.05	0.39
F-21	3.60	0.39
F-22	41.50	0.39
F-23	41.50	0.39
F-24	55.17	0.39
F-25	45.89	0.39
F-26	50.78	0.39
F-27	34.66	0.39
F-28	102.05	0.39
F-29	151.85	0.39
F-30	1.46	—
F-31	0.87	—
F-32	0.09	—

・「実行時」の欄 - - B：バイト数、W：ワード数、D：ダブルワード数

命令語	処理時間(μ s)	
	実行時	非実行時
F-33	0.09	—
F-34	1.03	0.39
F-35	1.03	0.39
F-36	6.83	0.39
F-37	7.14	0.39
F-38	1.37	0.39
F-40	0.93	—
F-41	1.46	—
F-42	0.91	—
F-43	0.45	—
F-44	0.57	—
F-45	0.64	—
F-47	0.33	—
F-48	0.33	—
F-49	1.09	—
F-50	1.26	0.39
F-51	1.53	0.39
F-52	1.44	0.39
F-53	2.86	0.39
F-54	3.72	0.39
F-55	1.34	0.39
F-56	2.44	0.39
F-56w	3.54	0.39
F-56d	8.17	0.39
F-57	1.22	0.39
F-57w	1.58	0.39
F-57d	2.25	0.39
F-58	1.28+0.85B (B=1 ~ 8)	0.39
F-60	2.68	0.95
F-60w	2.99	1.02
F-60d	3.66	1.10
F-61	1.22	0.39
F-61w	1.40	0.39
F-61d	1.77	0.39
F-62	2.44	0.95
F-62w	2.56	1.02
F-62d	2.56	1.10
F-63	1.40	0.39
F-63w	1.70	0.39
F-63d	2.31	0.39
F-64	1.40	0.39
F-64w	1.70	0.39
F-64d	2.31	0.39
F-65	1.89	0.39
F-65w	2.38	0.39
F-65d	3.60	0.39
F-66	2.13	0.39
F-66w	2.62	0.39
F-66d	3.60	0.39
F-67	1.95+0.61B (B=1 ~ 256)	0.39
F-68	1.95+0.61B (B=1 ~ 256)	0.39
F-69	1.31	0.39
F-70	2.31+0.30B (B=1 ~ 256)	0.39
F-70w	2.31+0.61W (W=1 ~ 256)	0.39
F-70d	2.44+1.16D (D=1 ~ 256)	0.39

命令語	処理時間(μ s)	
	実行時	非実行時
F-71	2.74+0.12B (B=1 ~ 1024)	0.39
F-71w	2.86+0.24W (W=1 ~ 512)	0.39
F-71d	2.86+0.54D (D=1 ~ 256)	0.39
F-72	2.50+0.48B (B=1 ~ 256)	0.39
F-72w	2.56+0.79W (W=1 ~ 256)	0.39
F-72d	2.86+1.52D (D=1 ~ 256)	0.39
F-73	2.25+0.48B (B=1 ~ 256)	0.39
F-73w	2.25+0.85W (W=1 ~ 256)	0.39
F-73d	2.31+1.52D (D=1 ~ 256)	0.39
F-74	2.31+0.12B (B=1 ~ 256)	0.39
F-74w	2.56+0.24W (W=1 ~ 256)	0.39
F-74d	2.74+0.54D (D=1 ~ 256)	0.39
F-76	2.56+0.30B (B=1 ~ 256)	0.39
F-76w	2.56+0.61W (W=1 ~ 256)	0.39
F-76d	2.56+1.22D (D=1 ~ 256)	0.39
F-77	1.40+0.36B (B=1 ~ 256)	0.39
F-78	1.77+0.36B (B=1 ~ 256)	0.39
F-79	10バイト…47.17 20バイト…115.08 50バイト…569.97 100バイト…2179.29 200バイト…8608.42 256バイト…13980.08	0.39
F-79w	10ワード…63.04 20ワード…164.08 50ワード…839.90 100ワード…3213.16 200ワード…11977.49 256ワード…20361.73	0.39
F-79d	10ダブルワード…72.75 20ダブルワード…189.35 50ダブルワード…969.26 100ダブルワード…3708.07 200ダブルワード…13822.37 256ダブルワード…23498.02	0.39
F-80	15.99	0.39
F-82	特殊I/Oの状態による	0.39
F-85	特殊I/Oの状態による	0.39
F-86	特殊I/Oの状態による	0.39
F-90	0.07	—
F-91	1.39	0.39
F-97	2.95	0.39
F-100	1.70	0.39
F-101	1.40	0.39
F-102	1.77	0.39
F-102w	2.07	0.39
F-102d	2.80	0.39
F-103	1.89	0.39
F-103w	2.25	0.39
F-103d	2.99	0.39
F-112	1.34+0.91B (B=1 ~ 256)	0.39
F-112w	1.34+1.83W (W=1 ~ 256)	0.39
F-112d	3.71+5.76D (D=1 ~ 256)	0.39
F-116	27.83	0.39

・「実行時」の欄 - - B : バイト数、W : ワード数、D : ダブルワード数

命令語	処理時間(μ s)	
	実行時	非実行時
F-130	1.46	0.39
F-131	1.22	0.39
F-132	1.49	0.39
F-133	1.24	0.39
F-140	0.54	—
F-141	1.22	0.29
F-142	2.50	0.39
F-143	1.52	—
F-144	2.19	0.39
F-145	2.19	0.52
F-146	2.31	0.89
F-147	0.73	0.56
F-148	2.99	0.39
F-149	0.85	0.49
F-151	1.70	0.79
F-153	1.53	0.39
F-154	15.74	0.39
F-155	18.24	0.39
F-156	9.52	0.39
F-160	20.69	0.39
Fc160	23.77	0.39
F-161	2.01+0.42B (B=1 ~ 256)	0.82
F-161w	2.19+0.67W (W=1 ~ 256)	0.82
F-161d	2.56+1.09D (D=1 ~ 256)	0.82
F-163	1.43	0.39
F-163w	1.83	0.39
F-163d	2.53	0.39
F-164	1.41	0.39
F-164w	1.80	0.39
F-164d	10.19	0.39
F-170	1.73+0.42B (B=1 ~ 256)	0.39
F-170w	1.89+0.73W (W=1 ~ 256)	0.39
F-170d	1.95+1.34D (D=1 ~ 256)	0.39
F-171	1.77+0.48B (B=1 ~ 256)	0.39
F-171w	1.87+0.85W (W=1 ~ 256)	0.39
F-171d	2.16+1.34D (D=1 ~ 256)	0.39
F-172	2.13+0.42B (B=1 ~ 256)	0.39
F-172w	2.38+0.61W (W=1 ~ 256)	0.39
F-172d	2.80+1.03D (D=1 ~ 256)	0.39
F-173	2.31+0.42B (B=1 ~ 256)	0.39
F-173w	2.74+0.61W (W=1 ~ 256)	0.39
F-173d	3.54+1.03D (D=1 ~ 256)	0.39
F-174	1.52+0.67B (B=1 ~ 1024)	0.39
F-175	1.34+0.54B (B=1 ~ 1024)	0.39
F-176	88.86	0.39
F-177	88.86	0.39
F-180	1.95	0.39
F-180w	2.31	0.39
F-180d	3.05	0.39
Fc180	1.77	0.39
Fc180w	1.95	0.39
Fc180d	2.31	0.39
F-181	1.95	0.39
F-181w	2.31	0.39
F-181d	3.05	0.39

シフト領域の全データが00_Hのとき

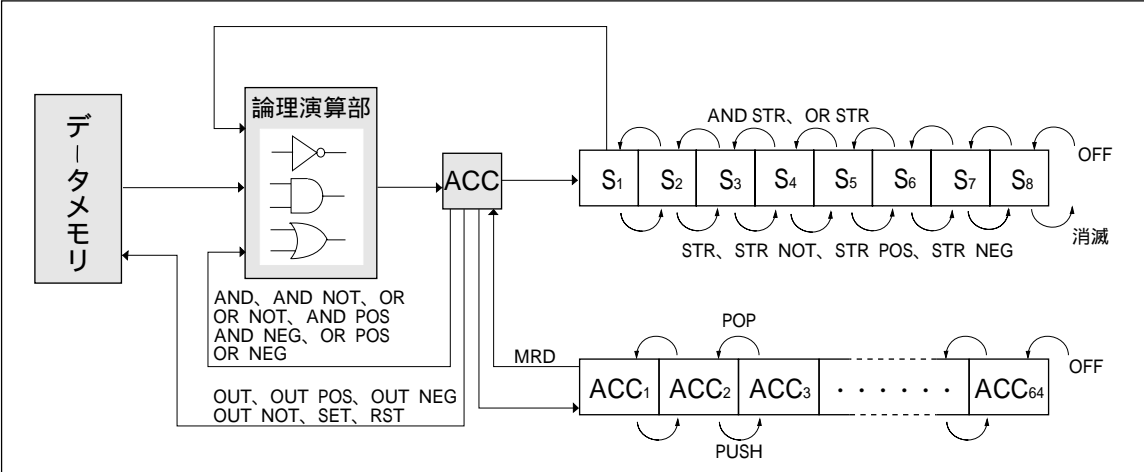
命令語	処理時間(μ s)	
	実行時	非実行時
Fc181	1.77	0.39
Fc181w	1.95	0.39
Fc181d	2.31	0.39
F-182	1.95	0.39
F-182w	2.31	0.39
F-182d	3.05	0.39
Fc182	1.77	0.39
Fc182w	1.95	0.39
Fc182d	2.31	0.39
F-183	1.95	0.39
F-183w	2.31	0.39
F-183d	3.05	0.39
Fc183	1.77	0.39
Fc183w	1.95	0.39
Fc183d	2.31	0.39
F-184	1.95	0.39
F-184w	2.31	0.39
F-184d	3.05	0.39
Fc184	1.77	0.39
Fc184w	1.95	0.39
Fc184d	2.31	0.39
F-185	1.95	0.39
F-185w	2.31	0.39
F-185d	3.05	0.39
Fc185	1.77	0.39
Fc185w	1.95	0.39
Fc185d	2.31	0.39
F-202	1.77	0.39
F-203	1.77	0.39
F-204	2.38	0.39
F-205	2.38	0.39
F-206	1.77	0.39
F-207	1.77	0.39
F-210	1.89	0.39
F-210w	2.44	0.39
F-210d	3.54	0.39
Fc210	1.70	0.39
Fc210w	2.07	0.39
Fc210d	2.80	0.39
F-211	1.95	0.39
F-211w	2.50	0.39
F-211d	3.54	0.39
Fc211	1.77	0.39
Fc211w	2.13	0.39
Fc211d	2.80	0.39
F-212	1.95	0.39
F-212w	2.50	0.39
F-212d	3.60	0.39
Fc212	1.58	0.39
Fc212w	1.77	0.39
Fc212d	2.13	0.39
Fx212	1.58	0.39
Fx212w	1.77	0.39
Fx212d	2.13	0.39
F-215	2.19	0.39
F-215w	2.86	0.39
F-215d	4.55	0.39

命令語	処理時間 (μ s)	
	実行時	非実行時
Fc215	1.95	0.39
Fc215w	2.50	0.39
Fc215d	4.00	0.39
F-216	2.62	0.39
F-216w	3.35	0.39
F-216d	4.69	0.39
Fc216	2.56	0.39
Fc216w	2.99	0.39
Fc216d	3.96	0.39
F-231	1.06	—
F-242	1.19	—
F-252	1.83	0.39
F-253	2.56	0.39
F-260	2.31	0.39
Fc260	2.25	0.39
F-261	3.29	0.39
Fc261	2.99	0.39
F-263	1.40	0.39
F-263w	1.77	0.39
F-263d	2.38	0.39
F-264	1.34	0.39
F-264w	1.70	0.39
F-264d	2.38	0.39
F-300	0.09	0.39
F-300w	0.09	0.39
F-300d	1.06	0.39
F-310	3.42	0.39
F-311	3.42	0.39
F-315	4.75	0.39
F-316	2.99	0.39
F-403	$29.78 + 0.49 \times \text{設定個数}$ (設定個数 = 1 ~ 10)	0.39

第 8 章 基本命令の説明

8 - 1 基本命令の演算

基本命令の演算には、データメモリ部と論理演算部、アキュムレータ(ACC)、およびスタックレジスタ(S1~S8)を使用します。

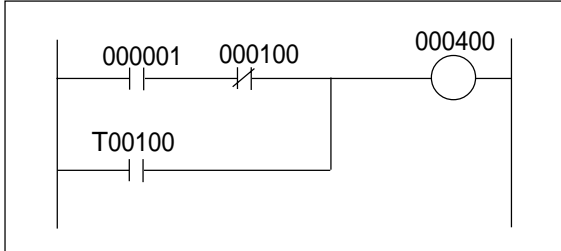


- (1) データメモリ
- ・リレー番号(入力リレー、出力リレー、補助リレー、タイマ・カウンタ接点)によって指定されるメモリで、ON / OFF情報を格納します。
 - ・基本命令では、次の命令でデータメモリのON/OFF情報を読み出し、
STR、STR NOT、AND、AND NOT、OR、OR NOT
STR POS、STR NEG、AND POS、AND NEG、OR POS、OR NEG
次の命令で書き込みます。
OUT、OUT POS、OUT NEG、OUT NOT、SET、RST、タイマ・カウンタ

[命令語記述例]

STR	000001
AND NOT	000100
OR	T00100
OUT	000400
命令	リレー番号

[ラダー図記述例]



- (2) 論理演算部
命令の内容に従い、論理演算を行います。
- (3) アキュムレータ(ACC)
・論理演算結果を格納する1ビットのレジスタです。
・次の命令によって変化します。
STR、STR NOT、AND、AND NOT、OR、OR NOT、AND STR、OR STR
STR POS、STR NEG、AND POS、AND NEG、OR POS、OR NEG、PUSH、POP
- (4) スタックレジスタ(S1、S2、S3、S4、S5、S6、S7、S8)
・直並列回路の演算や、複数の入力条件があるカウンタ命令、応用命令(F-60、F-60w、F-62、F-62w等)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。
・次の命令によって変化します。
STR、STR NOT、AND STR、OR STR、STR POS、STR NEG
- (5) 内部記憶エリア(ACC1、ACC2、ACC3、・・・、ACC64)
・アキュムレータ、スタックの内容を記憶するエリアです。
・PUSH、POPの命令によって変化します。

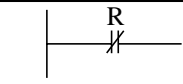
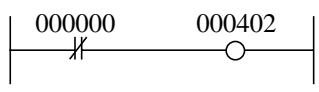
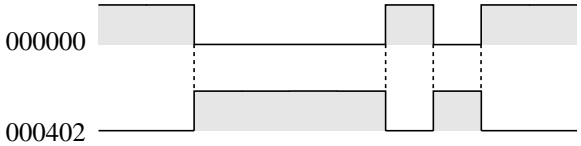

8 - 2 各基本命令の説明

STR (ストア)

シンボル		[使用例1]		<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>000000</td></tr> <tr><td>OUT</td><td>000400</td></tr> </table>	命 令		STR	000000	OUT	000400
命 令										
STR	000000									
OUT	000400									
機能	母線からの第1接点、または回路ブロックの第1接点がa接点の場合に使用する。									
演算内容	リレー番号Rのデータメモリの内容 (ON/OFF情報) をアキュムレータ (ACC) に格納する。 また、以前にACCに入っていたON/OFF情報はスタックレジスタの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する。 	入力リレー000000がONのとき、出力リレー000400がONします。 								
演算後	Rの内容	保持	[使用例2]							
	ACC	Rの内容								
	S1	演算前のACCの内容								
	S2	演算前のS1の内容								
	S3	演算前のS2の内容								
	S4	演算前のS3の内容								
	S5	演算前のS4の内容								
	S6	演算前のS5の内容								
	S7	演算前のS6の内容								
S8	演算前のS7の内容									
			STR 000000 ----- 母線(ブロックA)の第1接点 OR 000001 STR 000002 ----- ブロックBの第1接点 STR 000003 ----- ブロックCの第1接点 AND 000004 OR STR AND STR OUT 000401							

Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777	000000 ~ 015777			000000 ~ 015777	
	020000 ~ 075777	020000 ~ 075777			020000 ~ 075777	
	T00000 ~ T01777	100000 ~ 153777			100000 ~ 543777	
	C00000 ~ C01777	T00000 ~ T03777 C00000 ~ C03777			T00000 ~ T17777 C00000 ~ C17777	

STR NOT (ストア ノット)

シンボル			<p>[使用例1]</p> <div style="display: flex; align-items: center;">  <table border="1" style="margin-left: 20px; border-collapse: collapse;"> <tr><th colspan="2" style="text-align: center;">命 令</th></tr> <tr><td style="padding: 2px;">STR NOT</td><td style="padding: 2px;">000000</td></tr> <tr><td style="padding: 2px;">OUT</td><td style="padding: 2px;">000402</td></tr> </table> </div> <p>入力リレー000000がOFFのとき、出力リレー000402がONします。</p> 	命 令		STR NOT	000000	OUT	000402
命 令									
STR NOT	000000								
OUT	000402								
機能	母線からの第1接点、または回路ブロックの第1接点がb接点の場合に使用する。								
演算内容	<p>リレー番号Rのデータメモリの内容 (ON/OFF情報) を反転し、アキュムレータ (ACC) に格納する。 また、以前にACCに入っていたON/OFF情報はスタックレジスタの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する。</p> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">R</div> <div style="margin-right: 5px;">→</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">ACC</div> <div style="margin-right: 5px;">→</div> <div style="display: flex; gap: 2px;"> <div style="border: 1px solid black; padding: 2px;">S₁</div> <div style="border: 1px solid black; padding: 2px;">S₂</div> <div style="border: 1px solid black; padding: 2px;">S₃</div> <div style="border: 1px solid black; padding: 2px;">S₄</div> <div style="border: 1px solid black; padding: 2px;">S₅</div> <div style="border: 1px solid black; padding: 2px;">S₆</div> <div style="border: 1px solid black; padding: 2px;">S₇</div> <div style="border: 1px solid black; padding: 2px;">S₈</div> </div> <div style="margin-left: 10px; font-size: small;"> ↓ シフト </div> </div>								
演算後	Rの内容	保持							
	ACC	Rの内容を反転した値							
	S ₁	演算前のACCの内容							
	S ₂	演算前のS ₁ の内容							
	S ₃	演算前のS ₂ の内容							
	S ₄	演算前のS ₃ の内容							
	S ₅	演算前のS ₄ の内容							
	S ₆	演算前のS ₅ の内容							
	S ₇	演算前のS ₆ の内容							
	S ₈	演算前のS ₇ の内容							
		<p>[使用例2]</p>  <p>STR NOT 000000 母線(ブロックA)の第1接点 OR 000001 STR NOT 000002 ブロックBの第1接点 STR NOT 000003 ブロックCの第1接点 AND 000004 OR STR AND STR OUT 000403</p>							

Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777 020000 ~ 075777 T00000 ~ T01777 C00000 ~ C01777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777 T00000 ~ T03777 C00000 ~ C03777	000000 ~ 015777 020000 ~ 075777 100000 ~ 543777 T00000 ~ T17777 C00000 ~ C17777			

AND (アンド)

シンボル		[使用例]	<table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>000000</td> </tr> <tr> <td>AND</td> <td>000001</td> </tr> <tr> <td>OUT</td> <td>000404</td> </tr> </tbody> </table>	命 令		STR	000000	AND	000001	OUT	000404
命 令											
STR	000000										
AND	000001										
OUT	000404										
機能	直列接点がa接点の場合に使用する。										
演算内容	リレー番号Rのデータメモリの内容 (ON/OFF情報)とアキュムレータ (ACC)の内容をAND演算してその結果をACCに格納する。 	入力リレー000000と000001が共にONのとき、出力リレー000404がONします。 									
演算後	Rの内容	保持									
	ACC	Rの内容と演算前のACCの内容をAND演算した値									
	S ₁ ~ S ₈	保持									

Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777 020000 ~ 075777 T00000 ~ T01777 C00000 ~ C01777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777 T00000 ~ T03777 C00000 ~ C03777		000000 ~ 015777 020000 ~ 075777 100000 ~ 543777 T00000 ~ T17777 C00000 ~ C17777		

AND NOT (アンド ノット)

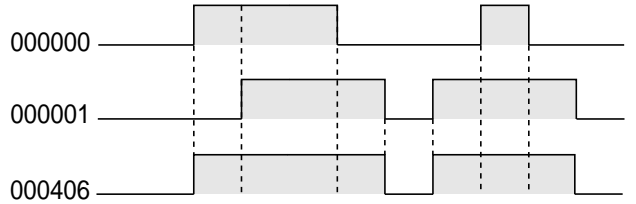
シンボル		[使用例]	<table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>000000</td> </tr> <tr> <td>AND NOT</td> <td>000001</td> </tr> <tr> <td>OUT</td> <td>000405</td> </tr> </tbody> </table>	命 令		STR	000000	AND NOT	000001	OUT	000405
命 令											
STR	000000										
AND NOT	000001										
OUT	000405										
機能	直列接点がb接点の場合に使用する。										
演算内容	リレー番号Rのデータメモリの内容 (ON/OFF情報を反転後、アキュムレータ (ACC)の内容とAND演算してその結果をACCに格納する。 	入力リレー000000がON、000001がOFFのとき、出力リレー000405がONします。 									
演算後	Rの内容	保持									
	ACC	Rの内容を反転した内容と演算前のACCの内容をAND演算した値									
	S ₁ ~ S ₈	保持									

Rの使用範囲	上記 (AND) と同じ
--------	--------------

OR (オア)

シンボル		[使用例] <table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>000000</td> </tr> <tr> <td>OR</td> <td>000001</td> </tr> <tr> <td>OUT</td> <td>000406</td> </tr> </tbody> </table>	命 令		STR	000000	OR	000001	OUT	000406
命 令										
STR	000000									
OR	000001									
OUT	000406									
機能	並列接点がa接点の場合に使用する。									
演算内容	リレー番号Rのデータメモリの内容 (ON/OFF情報)とアキュムレータ (ACC)の内容をOR演算してその結果をACCに格納する。 									
演算後	Rの内容	保持								
	ACC	Rの内容と演算前のACCの内容をOR演算した値								
	S ₁ ~ S ₈	保持								

入力リレー000000または000001がONのとき、出力リレー000406がONします。

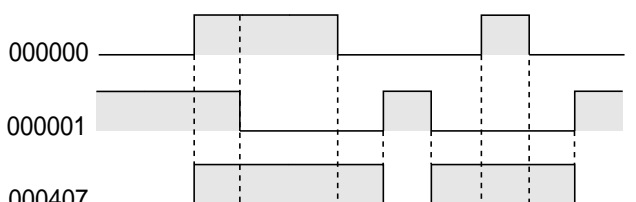


Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777 020000 ~ 075777 T00000 ~ T01777 C00000 ~ C01777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777 T00000 ~ T03777 C00000 ~ C03777		000000 ~ 015777 020000 ~ 075777 100000 ~ 543777 T00000 ~ T17777 C00000 ~ C17777		

OR NOT (オアノット)

シンボル		[使用例] <table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>000000</td> </tr> <tr> <td>OR NOT</td> <td>000001</td> </tr> <tr> <td>OUT</td> <td>000407</td> </tr> </tbody> </table>	命 令		STR	000000	OR NOT	000001	OUT	000407
命 令										
STR	000000									
OR NOT	000001									
OUT	000407									
機能	並列接点がb接点の場合に使用する。									
演算内容	リレー番号Rのデータメモリの内容 (ON/OFF情報)を反転後、アキュムレータ (ACC)の内容とOR演算してその結果をACCに格納する。 									
演算後	Rの内容	保持								
	ACC	Rの内容を反転した内容と演算前のACCの内容をOR演算した値								
	S ₁ ~ S ₈	保持								

入力リレー000000がON、または000001がOFFのとき、出力リレー000407がONします。

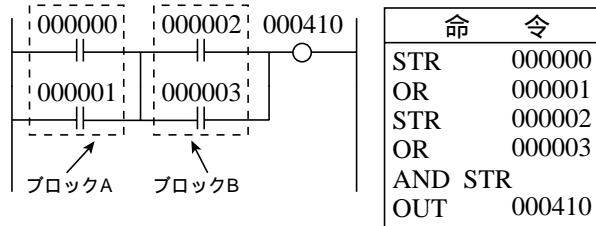


Rの使用範囲	上記 OR)と同じ
--------	------------

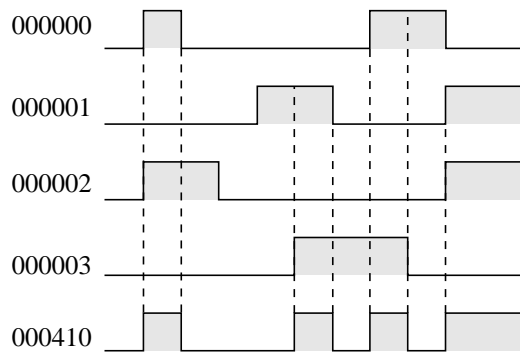
AND STR (アンドストア)

機能	回路ブロックと回路ブロックを直列接続する場合に使用する。	
演算内容	スタックレジスタS ₁ と、アキュムレータ(ACC)の内容(ON/OFF情報)をAND演算し、結果をACCに格納する。 また、以前にS ₂ に入っていたON/OFF情報はS ₁ に、S ₃ の情報はS ₂ に、以後S ₄ S ₃ 、S ₅ S ₄ 、S ₆ S ₅ 、S ₇ S ₆ 、S ₈ S ₇ とシフトし、S ₈ にはOFFの情報が入る。	
演算後	ACC	演算前のS ₁ の内容とACCの内容をAND演算した値
	S ₁	演算前のS ₂ の内容
	S ₂	演算前のS ₃ の内容
	S ₃	演算前のS ₄ の内容
	S ₄	演算前のS ₅ の内容
	S ₅	演算前のS ₆ の内容
	S ₆	演算前のS ₇ の内容
	S ₇	演算前のS ₈ の内容
	S ₈	OFF (0)

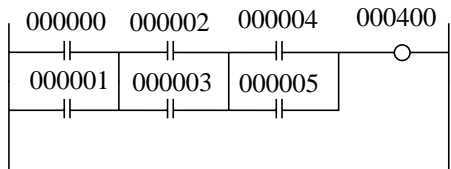
[使用例]



ブロックAとブロックBを直列接続します。
入力リレー000000または000001がONで、かつ000002または000003がONのとき、出力リレー000410がONします。

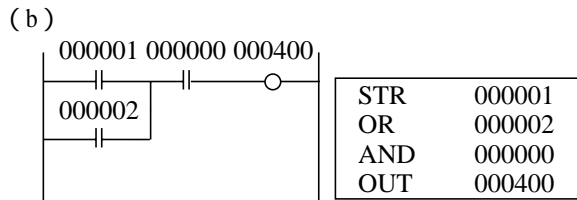
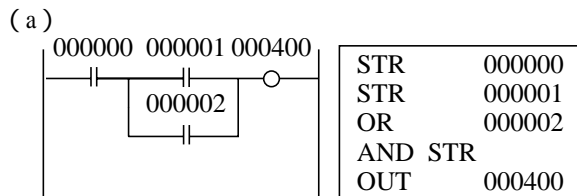


(注1) 次のラダー図をプログラムする方法として (a) (b) の2種類の方法があります。



(a)	(b)
STR 000000	STR 000000
OR 000001	OR 000001
STR 000002	STR 000002
OR 000003	OR 000003
AND STR	STR 000004
STR 000004	OR 000005
OR 000005	AND STR
AND STR	AND STR
OUT 000400	OUT 000400

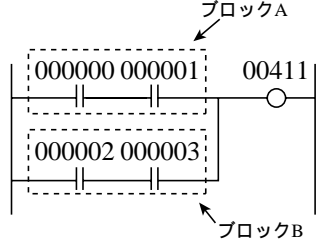
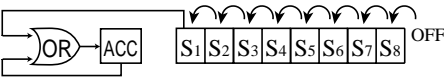
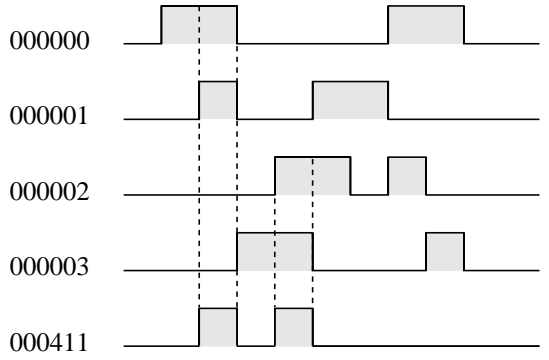
(注2) 次の (a) (b) の動作内容は同じです。



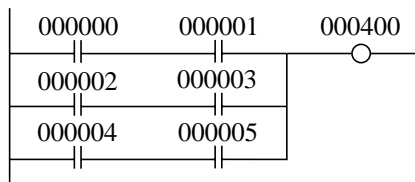
プログラムステップ数は (b) の方が1ステップ少なくなります。

(a)と(b)は、同じ演算結果が得られますが、(a)はスタックをS₁のみ、(b)はS₁とS₂を使用します。JW300のスタックは8個ですので(b)のようなプログラムを行うと最大9ブロックの接続しかできません。

OR STR (オアストア)

機能	回路ブロックと回路ブロックを並列接続する場合に使用する。	[使用例] 														
演算内容	<p>スタックレジスタS₁とアキュムレータ (ACC)の内容(ON/OFF情報)をOR演算し、結果をACCに格納する。また、以前にS₂に入っていたON/OFF情報はS₁に、S₃の情報はS₂に、以後S₄ S₃、S₅ S₄、S₆ S₅、S₇ S₆、S₈ S₇とシフトし、S₈にはOFFの情報が入る。</p> 		<table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>000000</td> </tr> <tr> <td>AND</td> <td>000001</td> </tr> <tr> <td>STR</td> <td>000002</td> </tr> <tr> <td>AND</td> <td>000003</td> </tr> <tr> <td>OR STR</td> <td></td> </tr> <tr> <td>OUT</td> <td>000411</td> </tr> </tbody> </table> <p>ブロックAとブロックBを並列接続します。入力リレー000000かつ000001、または000002かつ000003がONのとき、出力リレー000411がONします。</p> 	命 令		STR	000000	AND	000001	STR	000002	AND	000003	OR STR		OUT
命 令																
STR	000000															
AND	000001															
STR	000002															
AND	000003															
OR STR																
OUT	000411															
演算後	ACC	演算前のS ₁ の内容とACCの内容をOR演算した値														
	S ₁	演算前のS ₂ の内容														
	S ₂	演算前のS ₃ の内容														
	S ₃	演算前のS ₄ の内容														
	S ₄	演算前のS ₅ の内容														
	S ₅	演算前のS ₆ の内容														
	S ₆	演算前のS ₇ の内容														
	S ₇	演算前のS ₈ の内容														
S ₈	OFF (0)															

(注1) 次のラダー図をプログラムする方法として (a)、(b)の2種類の方法があります。



(a)

STR	000000
AND	000001
STR	000002
AND	000003
OR STR	
STR	000004
AND	000005
OR STR	
OUT	000400

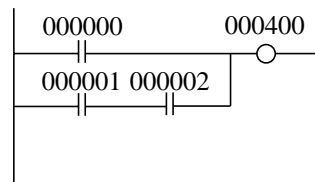
(b)

STR	000000
AND	000001
STR	000002
AND	000003
STR	000004
AND	000005
OR STR	
OR STR	
OUT	000400

(a)と(b)は、同じ演算結果が得られますが、(a)はスタックをS₁のみ、(b)はS₁とS₂を使用します。JW300のスタックは8個ですので(b)のようなプログラムを行うと最大9ブロックの接続しかできません。

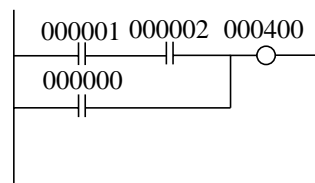
(注2) 次の(a)、(b)の動作内容は同じです。

(a)



STR	000000
STR	000001
AND	000002
OR STR	
OUT	000400

(b)



STR	000001
AND	000002
OR	000000
OUT	000400

プログラムステップ数は(b)の方が1ステップ少なくなります。

OUT (アウト)

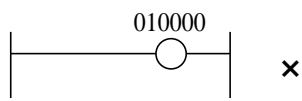
シンボル		<p>[使用例]</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>000000</td> </tr> <tr> <td>OUT</td> <td>000412</td> </tr> <tr> <td>OUT</td> <td>000413</td> </tr> <tr> <td>OUT</td> <td>000414</td> </tr> </tbody> </table> <p>入力リレー000000がONのとき、出力リレー000412、000413、000414をONします。 (OUT命令ではACCの内容は変化しないためOUTの連続使用が可能です。)</p>	命 令		STR	000000	OUT	000412	OUT	000413	OUT	000414
命 令												
STR	000000											
OUT	000412											
OUT	000413											
OUT	000414											
機 能	演算結果の出力に使用する。											
演 算 内 容	<p>アキュムレータ(ACC)の内容をリレー番号Rのデータメモリに書き込む。</p>											
演 算 後	Rの内容	ACCの内容										
	ACC	保持										
	S ₁ ~ S ₈	保持										

Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777 020000 ~ 075777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777	000000 ~ 015777 020000 ~ 075777 100000 ~ 543777			

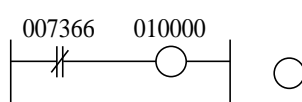
(注1) 特殊リレー領域(007300~007377)は、出力として使用できません。

(注4) 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。

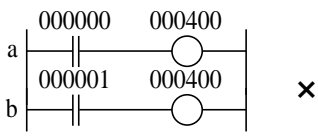
(注2) OUT命令は直接母線から始めることはできません。



常時ONしたいリレーは常時OFF接点(007366)を使用してください。



(注3) OUT命令は同一リレー番号を2重使用すると、サポートツールJW-15PG等のプログラムチェックでエラー表示します。



STR	000000
OUT	000412
AND	000001
OUT	000413
AND	000002
OUT	000414

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て減算式、加算式、および計数値をBCDまたはバイナリで取り扱う5種類があります。

また、設定値は固定値に加えて、レジスタ指定での設定も可能です。レジスタへの書込命令と組み合わせて使用すると、設定値の変更が容易になります。

(1) TMR命令の種類

名称	演算方法	計数値	TMR番号 1	設定範囲 2
TMR	減算式	BCD	00000 ~ 17777	0000 ~ 7999 レジスタ指定 3
DTMR (BCD)			00000 ~ 17777	0000 ~ 7999 レジスタ指定 3
DTMR (BIN)		バイナリ	00000 ~ 17777	00000 ~ 32767 レジスタ指定 4
UTMR (BCD)	加算式	BCD	00000 ~ 17777	0000 ~ 7999 レジスタ指定 3
UTMR (BIN)		バイナリ	00000 ~ 17777	00000 ~ 32767 レジスタ指定 4

1 TMR番号は、コントロールユニットの機種により異なります。

JW-311CU	JW-321CU	JW-331CU	JW-341CU	JW-352CU	JW-362CU
JW-312CU	JW-322CU	JW-332CU	JW-342CU		
00000 ~ 01777	00000 ~ 03777	00000 ~ 17777			

なお、TMR番号はCNT、MDと共有しています。

2 レジスタ指定では、データメモリのすべてのバイトアドレスを設定できます。

(間接アドレス@は設定不可)

なお、バイトアドレスは2バイトを使用し、必ず偶数アドレスを設定してください。

3 TMR、DTMR(BCD)、UTMR(BCD)のとき、レジスタには0000 ~ 7999(BCD)を設定してください。

4 DTMR(BIN)、UTMR(BIN)のとき、レジスタには00000 ~ 32767(バイナリ)を設定してください。

(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の間はこの状態を保持します。

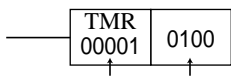
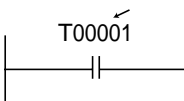
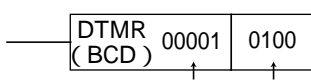
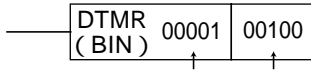
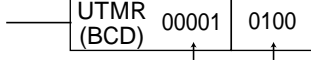
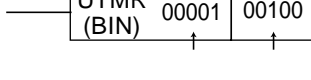
スタート入力	設定値	TMR接点
OFF	設定値	OFF
ON(現在値 > 0)	0.1秒ごとに - 1	OFF
ON(現在値 = 0)	0	ON

(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の間はこの状態を保持します。

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

スタート入力
ONでスタート

TMR番号

00000 ~ 17777⁽⁸⁾... { 00000 ~ 00777 : TMR・CNT・MD共有
01000 ~ 17777 : TMR・CNT共有

設定値

0.1秒(100ms)単位

0.01秒(10ms)単位

0.001秒(1ms)単位

TMR00000 ~ 00777 はシステムメモリ #0227 により 10ms タイマに、TMR01770 ~ 01777 は #0225 により 1ms タイマに設定できます。

ただし、DTMR、UTMR は 100ms としてのみ動作します。

精度

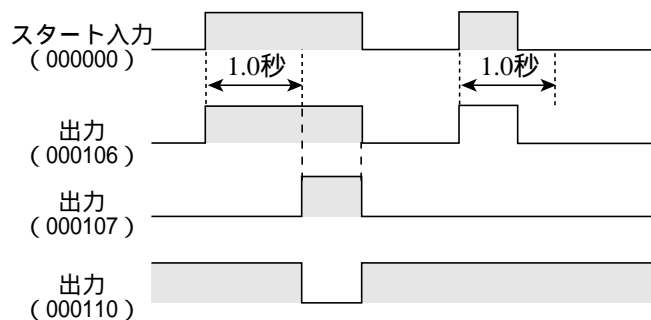
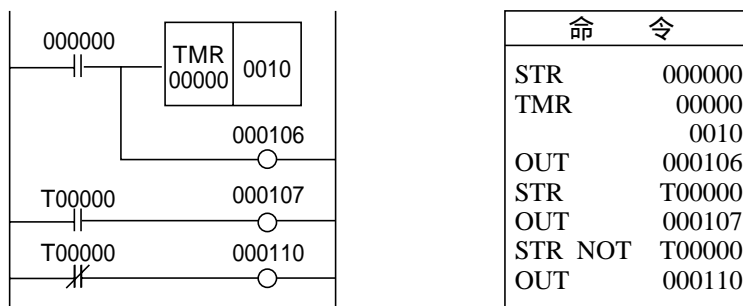
100ms タイマのとき、設定値⁺⁰_{-0.1} + スキャンタイム (秒)

10ms タイマのとき、設定値⁺⁰_{-0.01} + スキャンタイム (秒)

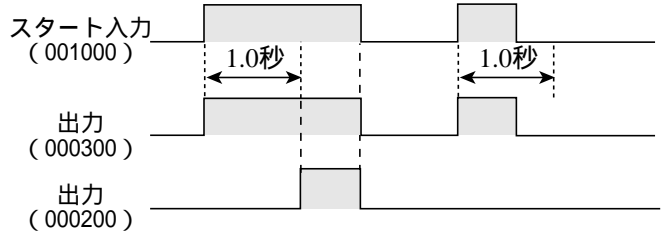
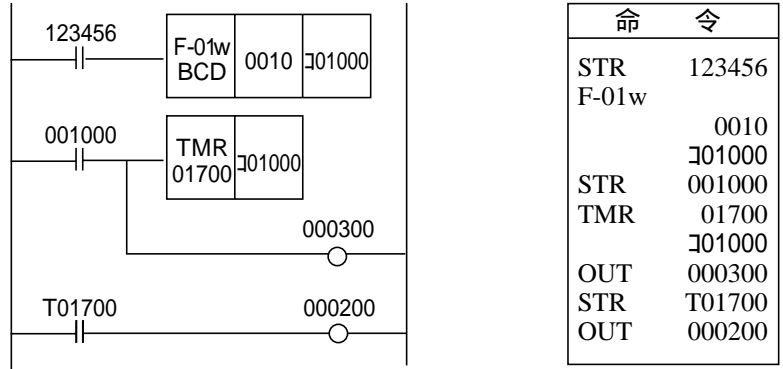
1ms タイマのとき、設定値⁺⁰_{-0.001} + スキャンタイム (秒)

(5) 使用例

例 1

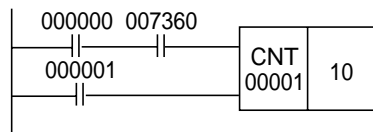


例 2

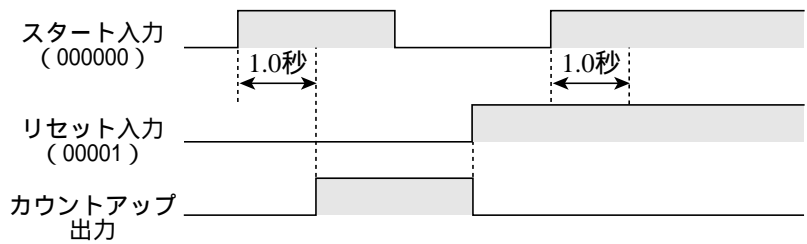


留 意 点

- ・ TMR番号は、CNT・MDと共通使用です。CNT・MDに使用した番号は、TMRには使用しないでください。また、同一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、サポートツールのプログラムチェックでエラー表示します。
- ・ TMR接点はTMR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、b*****に格納されます。 2・8ページ参照
- ・ JW300の電源投入時、タイマはリセットします。よって、タイマのスタート入力がON状態で、JW300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(#0201)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。
- ・ 接点00736Q(0.1秒クロック)とCNT命令を利用して停電記憶のタイマ、および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。



命 令	
STR	000000
AND	007360
STR	000001
CNT	00001
	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、および計数回数をBCD値またはバイナリ値で取り扱う5種類があります。

また、設定値は固定値に加えて、レジスタ指定での設定も可能です。レジスタへの書込命令と組み合わせて使用すると、設定値の変更が容易になります。

(1) CNT命令の種類

名称	演算方法	計数値	CNT番号 1	設定範囲 2
CNT	減算式	BCD	00000 ~ 17777	0000 ~ 7999 レジスタ指定 3
DCNT (BCD)			00000 ~ 17777	0000 ~ 7999 レジスタ指定 3
DCNT (BIN)		バイナリ	00000 ~ 17777	00000 ~ 32767 レジスタ指定 4
UCNT (BCD)	加算式	BCD	00000 ~ 17777	0000 ~ 7999 レジスタ指定 3
UCNT (BIN)		バイナリ	00000 ~ 17777	00000 ~ 32767 レジスタ指定 4

1 CNT番号は、コントロールユニットの機種により異なります。

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
00000 ~ 01777	00000 ~ 03777	00000 ~ 17777			

なお、CNT番号はTMR、MDと共有しています。

2 レジスタ指定では、データメモリのすべてのバイトアドレスを設定できます。

(間接アドレス@は設定不可)

なお、バイトアドレスは2バイトを使用し、必ず偶数アドレスを設定してください。

3 CNT、DCNT(BCD)、UCNT(BCD)のとき、レジスタには0000 ~ 7999(BCD)を設定してください。

4 DCNT(BIN)、UCNT(BIN)のとき、レジスタには00000 ~ 32767(バイナリ)を設定してください。

(2) 減算式CNT命令

- リセット入力ONの間、計数入力が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。
- リセット入力がOFFの間、計数入力がOFF ONに変化するごとに現在値は - 1 し、現在値が0になるとCNT接点はONし、リセット入力がOFFの間はこの状態を保持します。

リセット入力	設定値	CNT接点
ON	設定値	OFF
OFF(現在値 > 0)	・計数入力がOFF ON となるごとに - 1	OFF
OFF(現在値 = 0)	0	ON

(3) 加算式CNT命令

リセット入力ONの間、計数入力が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。

リセット入力がOFFの間、計数入力が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、リセット入力がOFFの間はこの状態を保持します。

リセット入力	設定値	CNT接点
ON	0	OFF
OFF(現在値 < 0)	・ 計数入力がOFF ON となるごとに +1	OFF
OFF(現在値 = 0)	設定値	ON

(4) シンボルマーク

名称	CNTのシンボル	CNT接点のシンボル
CNT		
DCNT (BCD)		
DCNT (BIN)		
UCNT (BCD)		
UCNT (BIN)		

計数入力

OFF ONを検知

リセット入力

ONでリセット

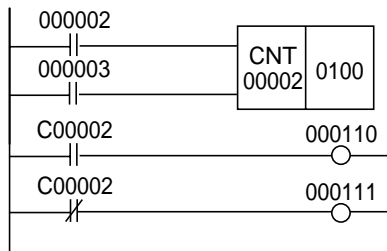
CNT番号

00000 ~ 17777⁽⁸⁾... { 00000 ~ 00777 : TMR・CNT・MD共有
01000 ~ 17777 : TMR・CNT共有

設定値

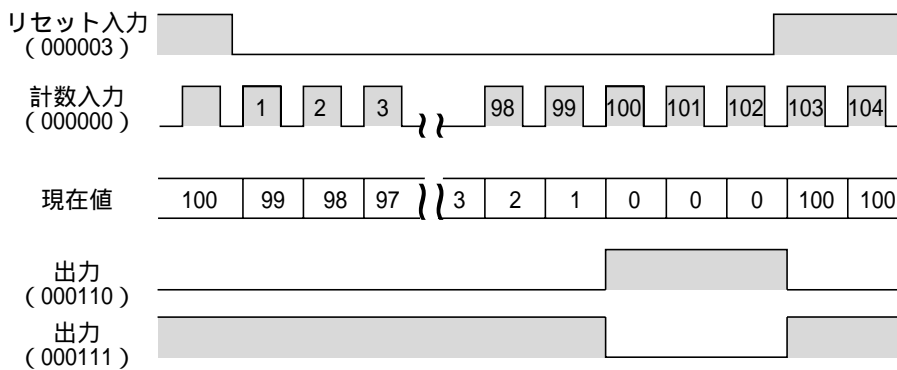
(5) 使用例

例 1



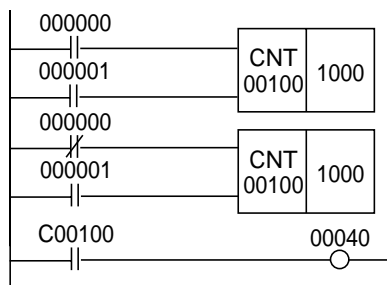
命 令	
STR	000002
STR	000003
CNT	00002
	0100
STR	C00002
OUT	00110
STR NOT	C00002
OUT	00111

計数入力がOFF ONへ変化したときに減算するカウンタです。



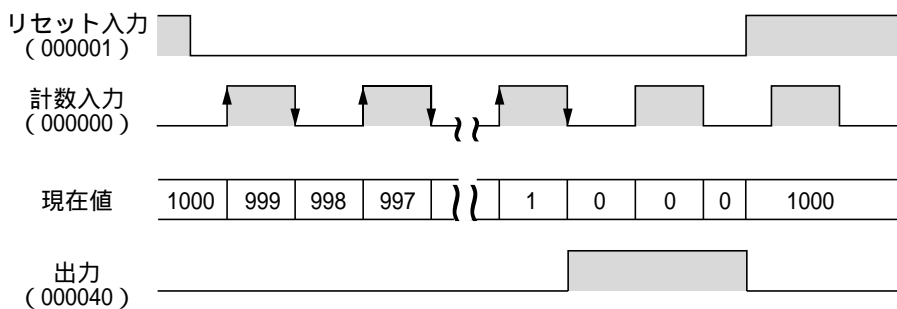
例 2

計数入力の立上り、立下りで計数するカウンタ

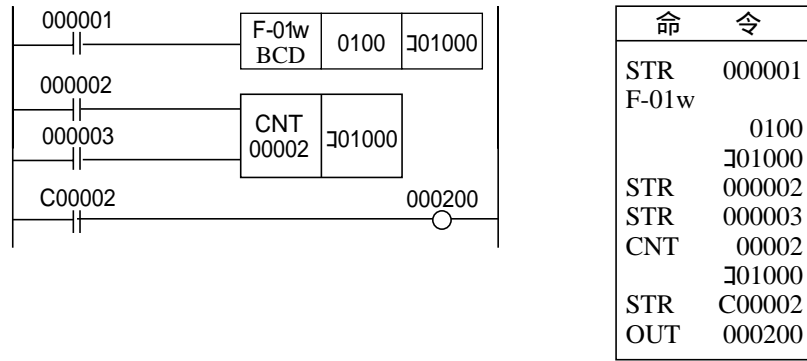


命 令	
STR	000000
STR	000001
CNT	00100
	1000
STR NOT	000000
STR	000001
CNT	00100
	1000
STR	C00100
OUT	000040

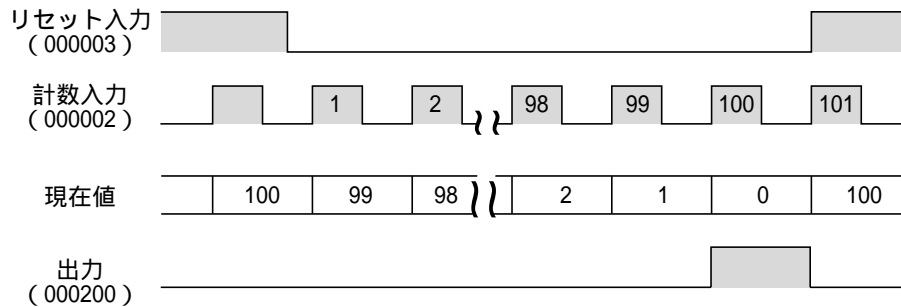
計数入力がOFF ONに変化したとき、ON OFFに変化したときのいずれの場合も減算するカウンタです。



例 3



計数入力がOFF ONへ変化したときに減算するカウンタです。

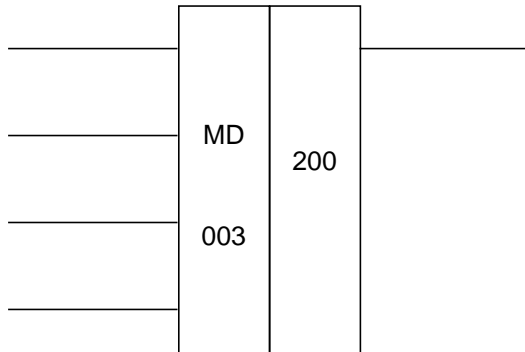


留 意 点

- ・ 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、サポートツールのプログラムチェックでエラー表示します。また、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。
- ・ 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、計数を開始するときはリセット入力を一旦ONした後、再びOFFにするか、サポートツールにより、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、b*****に格納されます。 2・8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に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、電源投入時に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(0202)にリセット条件を設定することにより「OFFでリセット」も可能です。

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。

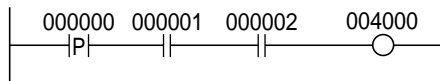


入力情報	のMDデータと共に外部に出力する接点情報で、リレー、TMR・CNTの接点を使用できます。
出力指示条件	で指定したMD番号のデータメモリまたはリレー領域に、 、 の接点情報および のMDデータを出力するかを指示する入力で、リレー、TMR・CNTの接点を使用できます。ONのとき出力します。OFFになっても接点情報、MDデータは変化しません。
MD拡張出力	MD命令を同一出力指示条件で連続して使用するとき、それぞれのMD命令に の条件をプログラムする必要はありません。
MD番号	MD命令は、 、 の接点情報、 のMDデータの各情報を格納するデータメモリ領域として、TMR・CNTの現在値格納領域 (b00000 ~ b01777) またはリレー領域 (r00000 ~ r01577) 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。 (注) TMR・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 r***** でプログラムします。例として、r00000 をプログラムすると、r00000、r00001 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、 、 の接点情報と、 のMDデータを外部出力(表示)できます。
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。

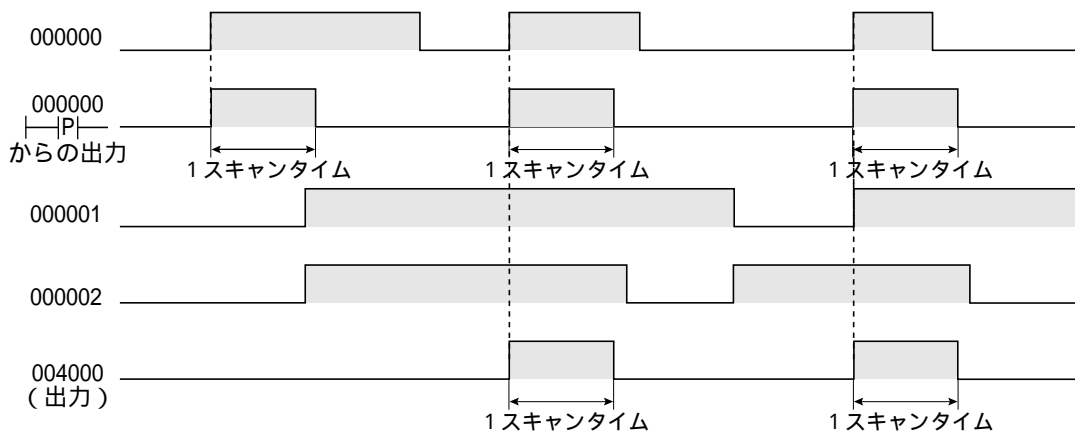
STR POS a 接点ロード立上り

シンボル						
機能	<p>STR POS命令で指定するリレー番号RがOFF ONの変化時に、1 スキャンタイムの間ONします。</p> <ul style="list-style-type: none"> ・本命令は、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。 ・本命令は、電源投入時と運転開始直後の1 スキャンの間は演算しません。 					
Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777 020000 ~ 075777 T00000 ~ T01777 C00000 ~ C01777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777 T00000 ~ T03777 C00000 ~ C03777		000000 ~ 015777 020000 ~ 075777 100000 ~ 543777 T00000 ~ T17777 C00000 ~ C17777		

[使用例]



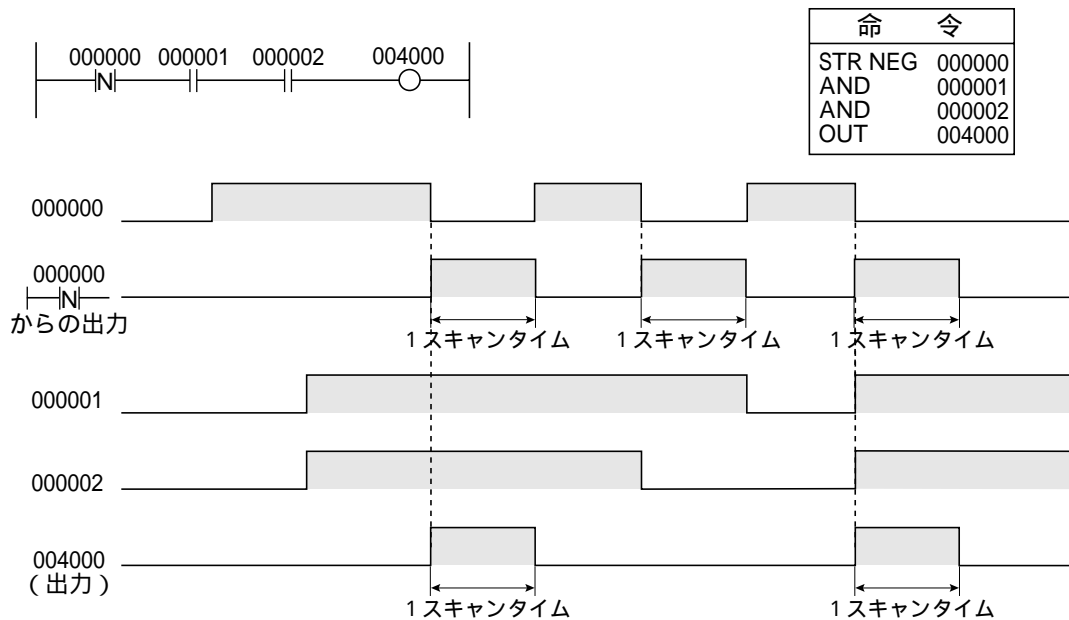
命 令	
STR POS	000000
AND	000001
AND	000002
OUT	004000



STR NEG a 接点ロード立下り

シンボル						
機能	<p>STR NEG命令で指定するリレー番号RがON OFFの変化時に、1 スキャンタイムの間ONします。</p> <ul style="list-style-type: none"> ・本命令は、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。 ・本命令は、電源投入時と運転開始直後の1 スキャンの間は演算しません。 					
Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777 020000 ~ 075777 T00000 ~ T01777 C00000 ~ C01777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777 T00000 ~ T03777 C00000 ~ C03777			000000 ~ 015777 020000 ~ 075777 100000 ~ 543777 T00000 ~ T17777 C00000 ~ C17777	

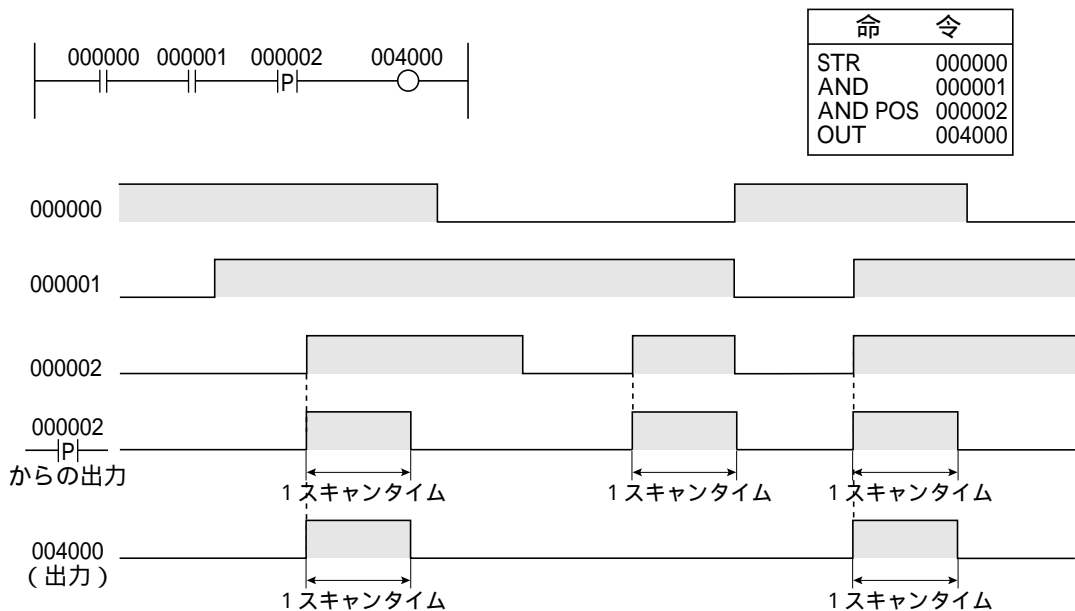
[使用例]



AND POS a 接点論理積立上り

シンボル						
機能	<p>AND POS命令で指定するリレー番号RがOFF ONの変化時に、1 スキャンタイムのパルスを発生し、ACCの内容とAND演算して、その結果をACCに格納します。</p> <ul style="list-style-type: none"> ・本命令は、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。 ・本命令は、電源投入時と運転開始直後の1 スキャンの間は演算しません。 					
Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777 020000 ~ 075777 T00000 ~ T01777 C00000 ~ C01777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777 T00000 ~ T03777 C00000 ~ C03777		000000 ~ 015777 020000 ~ 075777 100000 ~ 543777 T00000 ~ T17777 C00000 ~ C17777		

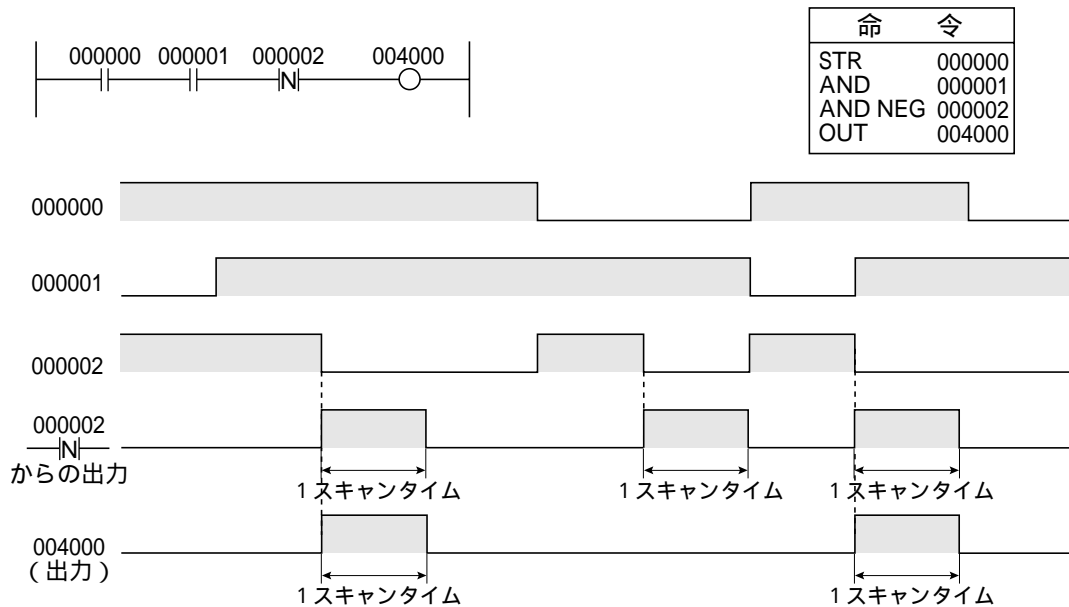
[使用例]



AND NEG a 接点論理積立下り

シンボル						
機能	<p>AND NEG命令で指定するリレー番号RがON OFFの変化時に、1 スキャンタイムのパルスが発生し、ACCの内容とAND演算して、その結果をACCに格納します。</p> <ul style="list-style-type: none"> ・本命令は、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。 ・本命令は、電源投入時と運転開始直後の1 スキャンの間は演算しません。 					
Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777 020000 ~ 075777 T00000 ~ T01777 C00000 ~ C01777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777 T00000 ~ T03777 C00000 ~ C03777			000000 ~ 015777 020000 ~ 075777 100000 ~ 543777 T00000 ~ T17777 C00000 ~ C17777	

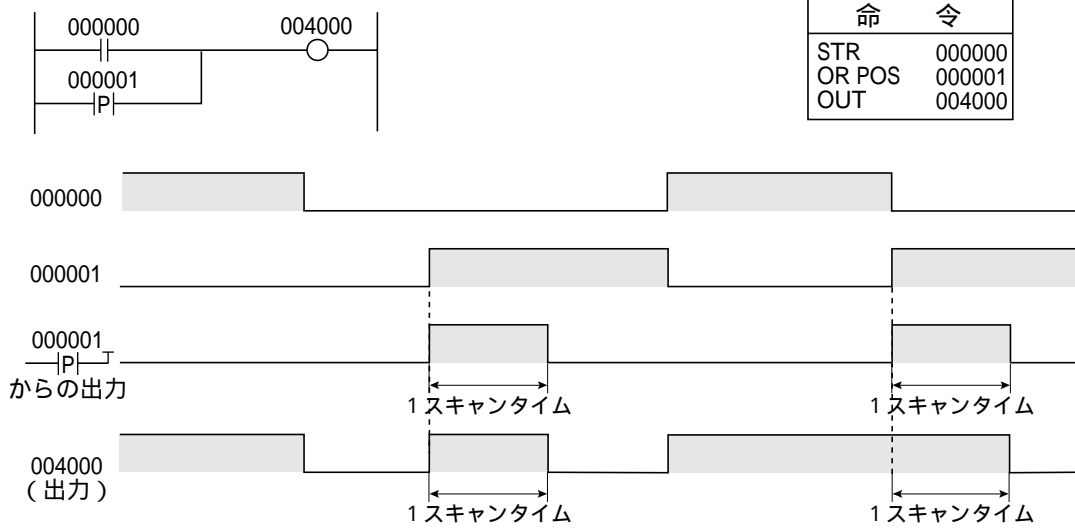
[使用例]



OR POS a 接点論理和立上り

シンボル						
機能	<p>OR POS命令で指定するリレー番号RがOFF ONの変化時に、1 スキャンタイムのパルスを発生し、ACCの内容とOR演算して、その結果をACCに格納します。</p> <ul style="list-style-type: none"> ・本命令は、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。 ・本命令は、電源投入時と運転開始直後の1 スキャンの間は演算しません。 					
Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777 020000 ~ 075777 T00000 ~ T01777 C00000 ~ C01777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777 T00000 ~ T03777 C00000 ~ C03777		000000 ~ 015777 020000 ~ 075777 100000 ~ 543777 T00000 ~ T17777 C00000 ~ C17777		

[使用例]

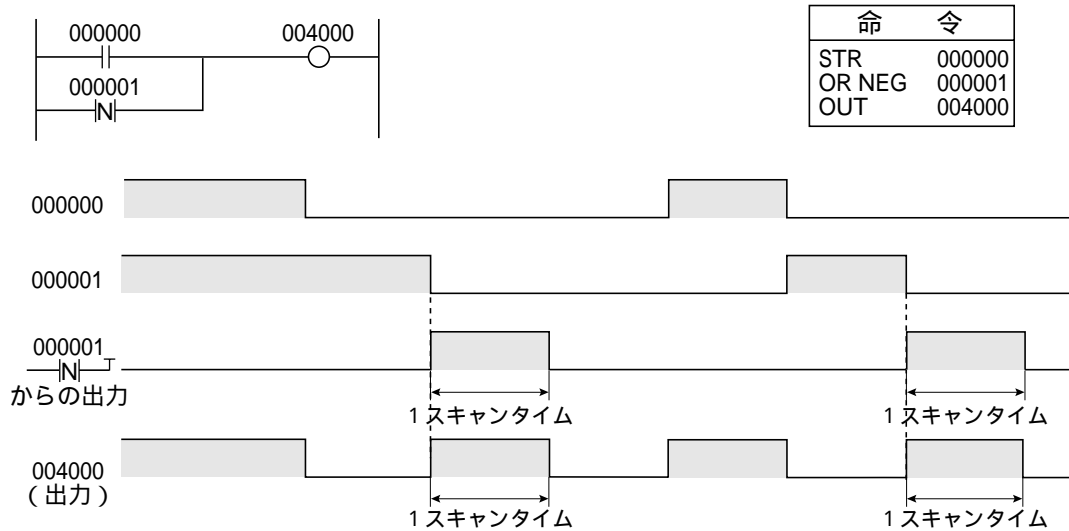


OR NEG

a 接点論理和立下り

シンボル						
機能	<p>OR NEG命令で指定するリレー番号RがON OFFの変化時に、1 スキャンタイムのパルスが発生し、ACCの内容とOR演算して、その結果をACCに格納します。</p> <ul style="list-style-type: none"> ・本命令は、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。 ・本命令は、電源投入時と運転開始直後の1 スキャンの間は演算しません。 					
Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777 020000 ~ 075777 T00000 ~ T01777 C00000 ~ C01777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777 T00000 ~ T03777 C00000 ~ C03777			000000 ~ 015777 020000 ~ 075777 100000 ~ 543777 T00000 ~ T17777 C00000 ~ C17777	

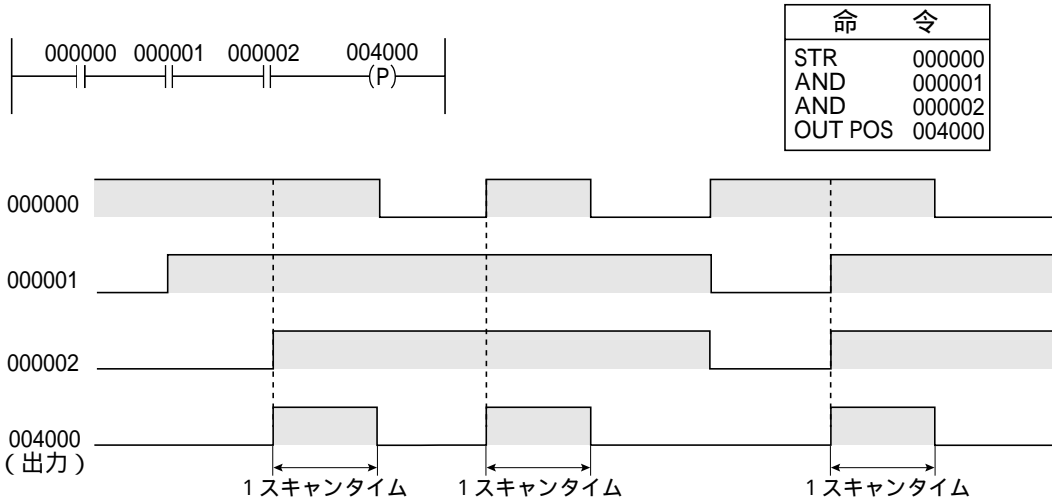
[使用例]



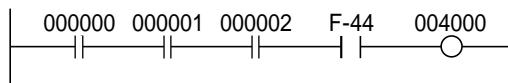
OUT POS 立上りコイル

シンボル						
機能	<p>OUT POS命令の直前のACC(アキュムレータ)の状態がOFF ONの変化時に、1スキャンタイムのパルスが発生し、ACCの内容を指定したデータメモリに転送します。次の命令と同じ演算となります。</p> <p style="text-align: center;"> </p> <ul style="list-style-type: none"> ・本命令は、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。 ・本命令は、電源投入時と運転開始直後の1スキャンの間は演算しません。 					
Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777 020000 ~ 075777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777		000000 ~ 015777 020000 ~ 075777 100000 ~ 543777		

[使用例]



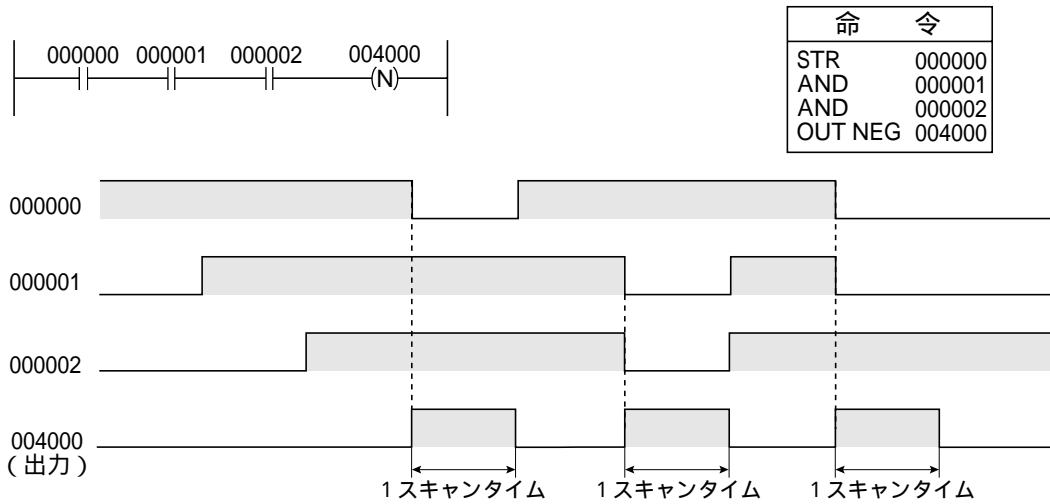
・上例のとき、下記と同じ演算になります。



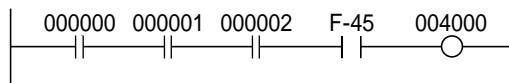
OUT NEG 立下りコイル

シンボル						
機能	<p>OUT NEG命令の直前のACC(アキュムレータ)の状態がON OFFの変化時に、1スキャンタイムのパルスを発生し、ACCの内容を指定したデータメモリに転送します。次の命令と同じ演算となります。</p> <ul style="list-style-type: none"> ・本命令は、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。 ・本命令は、電源投入時と運転開始直後の1スキャンの間は演算しません。 					
Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777 020000 ~ 075777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777			000000 ~ 015777 020000 ~ 075777 100000 ~ 543777	

[使用例]



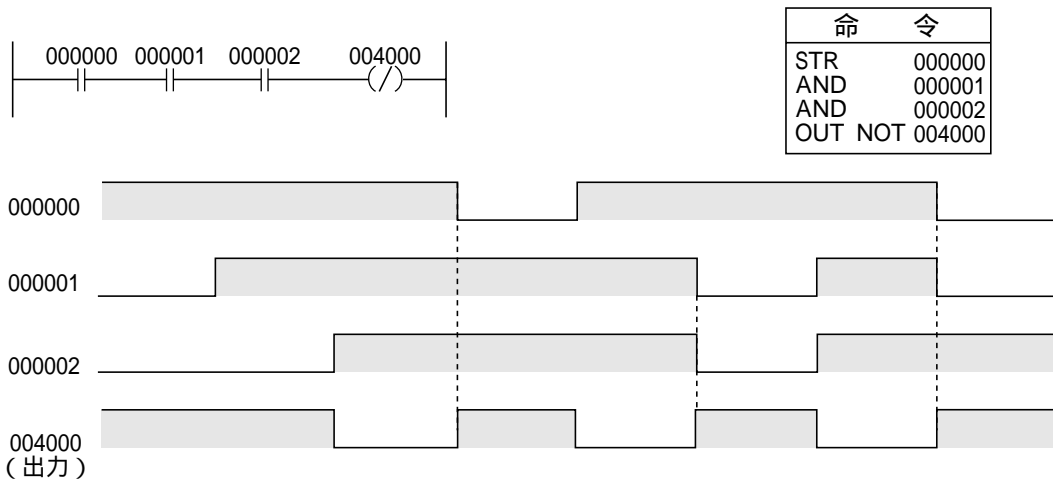
・上例のとき、下記と同じ演算になります。



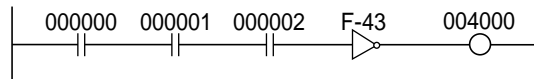
OUT NOT 入力条件を反転出力

シンボル						
機能	ACCの内容を反転して、指定したデータメモリに出力します。 次の命令と同じ演算となります。 					
Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777 020000 ~ 075777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777		000000 ~ 015777 020000 ~ 075777 100000 ~ 543777		

[使用例]



・ 上例のとき、下記と同じ演算になります。



SET

入力の立上り時にセット

シンボル												
機能	セット入力ONすると、指定するOUTをONして保持する。											
Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU						
	000000 ~ 015777 020000 ~ 075777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777		000000 ~ 015777 020000 ~ 075777 100000 ~ 543777								
[使用例]												
				<table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>000000</td> </tr> <tr> <td>SET</td> <td>004000</td> </tr> </tbody> </table>			命 令		STR	000000	SET	004000
命 令												
STR	000000											
SET	004000											
<p>セット入力000000がOFF ONの変化時に、OUT004000がONになります。 ONになったOUT004000は、セット入力000000がOFFになってもONのまま保持します。 セット入力000000がOFFのときは、OUT004000の状態は変化しません。</p> <p>上例のとき、下記と同じ演算となります。</p>												

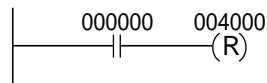
- ・本命令がF-30(MCS)命令の中にあるとき、本命令でONしたOUTは、F-30命令がOFFしてもONを保持します。
- ・本命令を使用すると、1個のOUTを複数の回路上で制御できます。
- ・本命令で指定するOUTがキープ指定領域内のときは、復電後も停電前の状態を保持します。また、指定するOUTがキープ指定領域外のときは、復電時にリセットします。
- ・本命令で指定するOUTがJW300停止時に出力保持する領域内のときは、JW300停止時に停止前の状態を保持します。また、指定するOUTがJW300停止時に出力保持する領域外のときは、JW300停止時にリセットされます。システムメモリ#0232、#0233、#0252、#0253参照
- ・本命令は、RST命令(次ページ)とペアで使用してください。
- ・MCS(F-30)とMCR(F-31)の間にあるSET命令とRST命令は、MCS(F-30)の演算条件がOFFのとき動作しません。

RST

入力の立上り時にリセット

シンボル						
機能	セット入力ONすると、指定するOUTをOFFして保持する。 次の命令と同じ演算となります。 <div style="text-align: center;"> </div>					
リレー番号Rの使用範囲	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
	000000 ~ 015777 020000 ~ 075777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777		000000 ~ 015777 020000 ~ 075777 100000 ~ 543777		

[使用例]

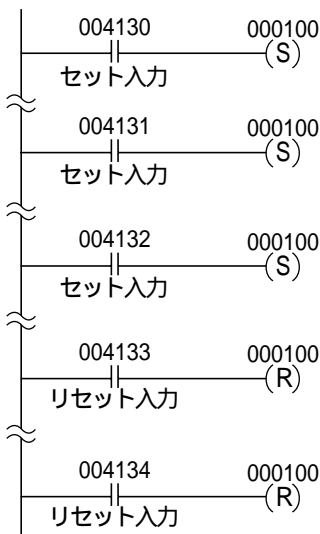


命 令	
STR	000000
RST	004000

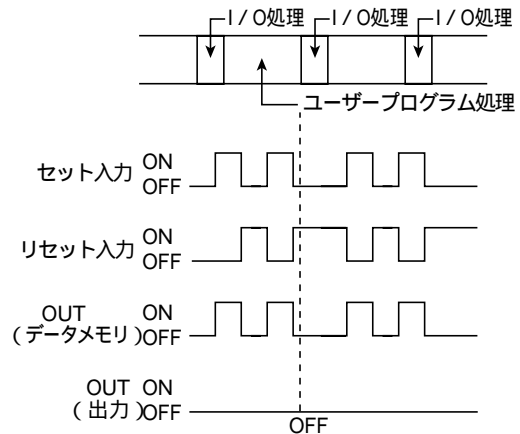


リセット入力000000がOFF ONの変化時に、OUT004000がOFFになります。
 OFFになったOUT004000は、リセット入力000000がOFFになってもOFFのまま保持します。
 リセット入力000000がOFFのときは、OUT004000の状態は変化しません。

- ・本命令で指定するOUTがキープ指定領域内のとき、復電後も停電前の状態を保持します。また、指定するOUTがキープ指定領域外のとき、復電時にリセットします。
- ・本命令で指定するOUTがJW300停止時に出力保持する領域内のとき、停止前の状態を保持します。また、指定するOUTがJW300停止時に出力保持する領域外のとき、停止時にリセットします。
 システムメモリ#0232、#0233、#0252、#0253参照
- ・本命令は、SET命令(前ページ)とペアで使用してください。
- ・SET命令と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。



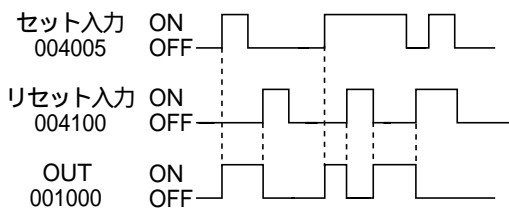
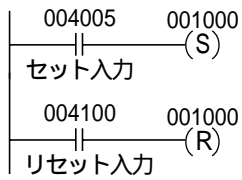
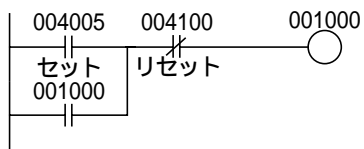
- ・セット入力とリセット入力が1スキャン内で複数回ON/OFFする場合、OUTとして使用しているデータメモリは1スキャン周期内でON/OFFを繰り返します。ただし、出力ユニットの出力端子はI/O処理直前のOUTの結果(ONまたはOFF)を出力します。



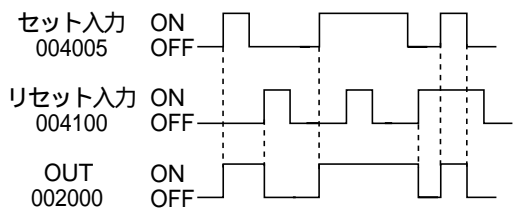
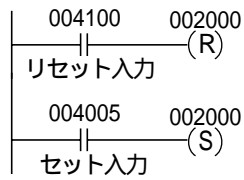
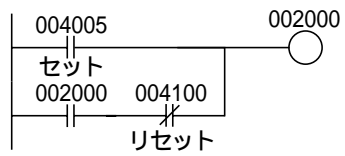
、 、 ではモニタ不可
 また、ユーザープログラム処理中にデータメモリが複数回ON/OFFしても、I/O処理直前の結果のみモニタできます。

・SET命令とRST命令をペアで使用すると、自己保持回路等を簡略化できます。

リセット優先自己保持回路



セット優先自己保持回路



PUSH

POP

MRD

命令語	機 能
PUSH	直前のアキュムレータ(ACC) スタックの内容を内部の記憶エリアに退避します。
POP	PUSH命令で記憶したアキュムレータ(ACC) スタックの内容を内部の記憶エリアから復帰します。演算後、PUSH命令で記憶した内容をクリアします。
MRD	PUSH命令で記憶したアキュムレータ(ACC) スタックの内容を内部の記憶エリアから一時的に読み出します。

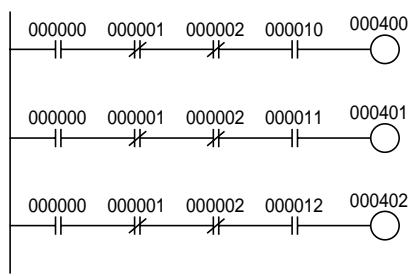
- ・ 1回路でのPUSH命令は64以内にしてください。
- ・ PUSH、POP命令の使用数は、必ず同一にしてください。
なお、PUSH命令を使用した後に、POP命令を使用してください。
- ・ MRD命令は、直前に記憶した記憶エリアの内容しか読み出せません。

使用例

8・30-31ページ

使用例 1

・ PUSH、POP、MRDを使用しない場合

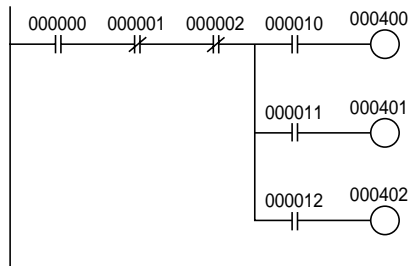


```

STR 000000
AND NOT 000001
AND NOT 000002
AND 000010
OUT 000400
STR 000000
AND NOT 000001
AND NOT 000002
AND 000011
OUT 000401
STR 000000
AND NOT 000001
AND NOT 000002
AND 000012
OUT 000402
    
```



・ PUSH、POP、MRDを使用した場合



```

STR 000000
AND NOT 000001
AND NOT 000002
PUSH
AND 000010
OUT 000400
MRD
AND 000011
OUT 000401
POP
AND 000012
OUT 000402
    
```

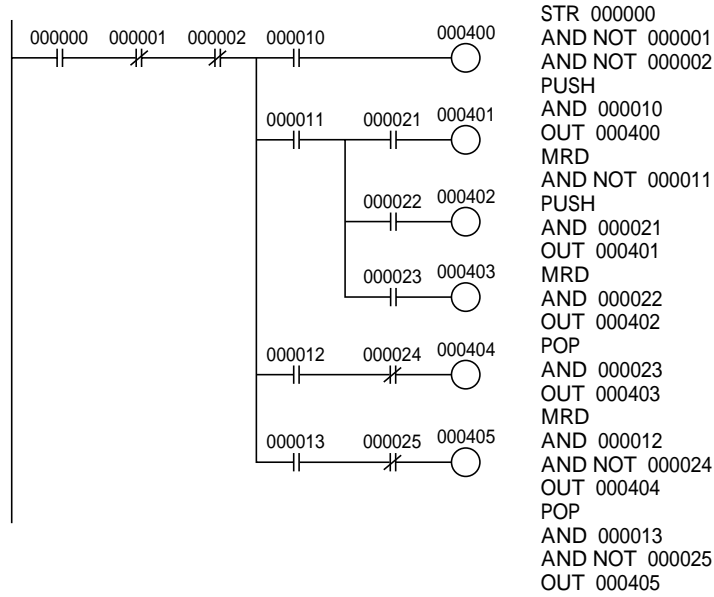
「PUSH」
ACC、スタックの内容を
退避する。

「MRD」
退避したACC、スタック
の内容を読み出す。

「POP」
退避したACC、スタック
の内容を元に戻す。

命令	アキュムレータ ACC	内部記憶エリア
		ACC ₁
STR 000000		
AND NOT 000001		
AND NOT 000002		
PUSH		
AND 000010		
OUT 000400		
MRD		
AND 000011		
OUT 000401		
POP		
AND 000012		
OUT 000402		

使用例 2



命令	アキュムレータ ACC	内部エリア	
		ACC ₁	ACC ₂
STR 000000	000000		
AND NOT 000001	000000 000001		
AND NOT 000002	000000 000001 000002		
PUSH	000000 000001 000002	000000 000001 000002	
AND 000010	000000 000001 000002 000010	000000 000001 000002	
OUT 000400	000000 000001 000002 000010	000000 000001 000002	
MRD	000000 000001 000002	000000 000001 000002	
AND 000011	000000 000001 000002 000011	000000 000001 000002	
PUSH	000000 000001 000002 000011	000000 000001 000002 000011	000000 000001 000002
AND 000021	000000 000001 000002 000011 00021	000000 000001 000002 000011	000000 000001 000002
OUT 000401	000000 000001 000002 000011 00021	000000 000001 000002 000011	000000 000001 000002
MRD	000000 000001 000002 000011	000000 000001 000002 000011	000000 000001 000002
AND 000022	000000 000001 000002 000011 00022	000000 000001 000002 000011	000000 000001 000002
OUT 000402	000000 000001 000002 000011 00022	000000 000001 000002 000011	000000 000001 000002
POP	000000 000001 000002 000011	000000 000001 000002	
AND 000023	000000 000001 000002 000011 00023	000000 000001 000002	
OUT 000403	000000 000001 000002 000011 00023	000000 000001 000002	
MRD	000000 000001 000002	000000 000001 000002	
AND 000012	000000 000001 000002 000012	000000 000001 000002	
AND NOT 000024	000000 000001 000002 000012 00024	000000 000001 000002	
OUT 000404	000000 000001 000002 000012 00024	000000 000001 000002	
POP	000000 000001 000002		
AND 000013	000000 000001 000002 000013		
AND NOT 000025	000000 000001 000002 000013 00025		
OUT 000405	000000 000001 000002 000013 00025		

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 〕 ソースとデスティネーション

転送命令や算術演算命令などの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)、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。

【例】

		S	D
—	F-00 XFER	001010	009200

・ 001010(S)の内容を009200(D)に転送します。

		S ₁	S ₂	D
—	F-10 ADD	001001	001002	019100

・ 001001(S₁)の内容と001002(S₂)の内容を加算し、結果を019100(D)に格納します。

		S	D
—	F-00w XFER	009000	019000

・ 009000(S)、009001(S+1)の内容を019000(D)、019001(D+1)に転送します。

ワード処理命令(w の付く命令)と2ワード処理命令(d の付く命令)は、必ずソースと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(S)、デスティネーション(D)が2バイト以上のデータメモリで演算する命令(F-70等)について、各領域を越えて設定した場合は、次のように続きのファイルアドレスの領域となります。

2・5ページ参照

S、D	S+1、D+1
001577	TMR・CNTの接点領域 (ファイルアドレス00001600)
007577	TMR・CNTの接点領域 (ファイルアドレス00035600)
b01777	009000
b03777	002000
b37777	109000
009777	019000
019777	029000
⋮	⋮
099777	E0000

S、D	S+1、D+1
E0777	E1000
⋮	⋮
E7777	b02000
109777	119000
119777	129000
⋮	⋮
199777	209000
209777	219777
⋮	⋮
299777	309000
309777	319777
⋮	⋮
389777	Z000

(注) S+1、D+1 等が下記の領域に入らないように設定してください。

1. TMR・CNT の接点領域 上記

ファイルアドレス 00001600 ~ 00001777⁽⁸⁾、00035600 ~ 00035777⁽⁸⁾、00101400 ~ 00101777⁽⁸⁾
(JW-32*CU のとき)、00140400 ~ 00143777⁽⁸⁾(JW-33*CU ~ 362CU のとき)

2. 各機種(JW-3**CU)のデータメモリの領域外

【例】 JW-31*CU のとき、ファイルアドレス 00074000⁽⁸⁾以降

3. JW-33*CU ~ 362CU のとき、ファイルアドレス00177777⁽⁸⁾から00200000⁽⁸⁾(ファイルレジスタ : FILE 1)へ入る設定は禁止です。

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。

【例】

	S	D
F-03 BIN	009110	009110

・ 009110(S)の内容(BCD 2桁)をバイナリコードに変換し、009110(D)に格納します。

特殊リレー領域の300730 ~ 300737は、CPUが書き込む領域で、デスティネーションに指定しないでください。

S、Dの使用範囲について

応用命令の説明(第10～14章)にて、ソース(S)とデスティネーション(D)の使用範囲は「A～C、E～H、J、K」で記載しています。各々の範囲は以下を参照願います。

(1) 使用範囲 A

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
100000 ~ 101577 102000 ~ 107577					
100000 ~ 115377	110000 ~ 154377				
b00000 ~ b01777 b02000 ~ b03777					
b04000 ~ b07777	b04000 ~ b37777				
009000 ~ 099777 E00000 ~ E07777 109000 ~ 199777 209000 ~ 299777 309000 ~ 389777 Z000 ~ Z377					
ファイルレジスタ (FILE 1)	00000000 ~ 00077777	00000000 ~ 00377777	00000000 ~ 01777777	00000000 ~ 07777777	00000000 ~ 37777777

Z000 ~ Z377は、ワード命令(F□w)にのみ使用可能です。

間接アドレス指定のとき

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@100000 ~ @101574 @102000 ~ @107574					
@100000 ~ @115374	@110000 ~ @154374				
@b00000 ~ @b01774 @b02000 ~ @b03774					
@b04000 ~ @b07774	@b04000 ~ b37774				
@009000 ~ @099774 @E00000 ~ @E07774 @109000 ~ @199774 @209000 ~ @299774 @309000 ~ @389774					
ファイルレジスタ (FILE 1)	@00000000 ~ @00077774	@00000000 ~ @00377774	@00000000 ~ @01777774	@00000000 ~ @07777774	@00000000 ~ @37777774

【適用される命令語】

F-0(S、D)、F-01(D)、F-02(D1、D2)、F-03(S、D)、F-04(S、D)など

(2) 使用範囲 B

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
100000 ~ 101576 102000 ~ 107576					
/	110000 ~ 115376	110000 ~ 1154376			
b00000 ~ b01776 b02000 ~ b03776					
/	b04000 ~ b07776	b04000 ~ b37776			
009000 ~ 099776 E00000 ~ E07776 109000 ~ 199776 209000 ~ 299776 309000 ~ 389776 Z000 ~ Z377 _____					
ファイルリスト (FILE 1)	00000000 ~ 00077776	00000000 ~ 00377776	00000000 ~ 01777776	00000000 ~ 07777776	00000000 ~ 37777776

Z000 ~ Z377は、ワード命令(F□w)にのみ使用可能です。

間接アドレス指定のとき

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@100000 ~ @101574 @102000 ~ @107574					
/	@110000 ~ @115374	@110000 ~ @1154374			
@b00000 ~ @b01774 @b02000 ~ @b03774					
/	@b04000 ~ @b07774	@b04000 ~ b37774			
@009000 ~ @099774 @E00000 ~ @E07774 @109000 ~ @199774 @209000 ~ @299774 @309000 ~ @389774					
ファイルリスト (FILE 1)	@00000000 ~ @00077774	@00000000 ~ @00377774	@00000000 ~ @01777774	@00000000 ~ @07777774	@00000000 ~ @37777774

【適用される命令語】

F-00w(S, D) F-01w(D) F-02w(D1, D2) F-03w(S, D) F-04w(S) など

(3) 使用範囲 C

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
100000 ~ 101574 102000 ~ 107574					
110000 ~ 115374	110000 ~ 154374				
b00000 ~ b01774 b02000 ~ b03774					
b04000 ~ b07774	b04000 ~ b37774				
009000 ~ 099774 E00000 ~ E07774 109000 ~ 199774 209000 ~ 299774 309000 ~ 389774 Z000 ~ Z376 _____					
ファイルリスト (FILE 1)	00000000 ~ 00077774	00000000 ~ 00377774	00000000 ~ 01777774	00000000 ~ 07777774	00000000 ~ 37777774

Z000 ~ Z376は、ワード命令(F□w)にのみ
使用可能です。

間接アドレス指定のとき

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@100000 ~ @101574 @102000 ~ @107574					
@110000 ~ @115374	@110000 ~ @154374				
@b00000 ~ @b01774 @b02000 ~ @b03774					
@b04000 ~ @b07774	@b04000 ~ b37774				
@009000 ~ @099774 @E00000 ~ @E07774 @109000 ~ @199774 @209000 ~ @299774 @309000 ~ @389774					
ファイルリスト (FILE 1)	@00000000 ~ @00077774	@00000000 ~ @00377774	@00000000 ~ @01777774	@00000000 ~ @07777774	@00000000 ~ @37777774

【適用される命令語】

F-00α(S, D)、F-02α(D1, D2)、F-05w(S)、F-06w(D)、F-09α(S, D)など

(4) 使用範囲 E

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
100000 ~ 101575 102000 ~ 107575					
110000 ~ 115375	110000 ~ 1154375				
b00000 ~ b01775 b02000 ~ b03775					
b04000 ~ b07775	b04000 ~ b37775				
009000 ~ 099775 E00000 ~ E07775 109000 ~ 199775 209000 ~ 299775 309000 ~ 389775 Z000 ~ Z376					
ファイルリスト (FILE 1)	00000000 ~ 00077775	00000000 ~ 00377775	00000000 ~ 01777775	00000000 ~ 07777775	00000000 ~ 37777775

Z000 ~ Z376は、ワード命令(F□w)にのみ
使用可能です。

間接アドレス指定のとき

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@100000 ~ @101574 @102000 ~ @107574					
@110000 ~ @115374	@110000 ~ @1154374				
@b00000 ~ @b01774 @b02000 ~ @b03774					
@b04000 ~ @b07774	@b04000 ~ b37774				
@009000 ~ @099774 @E00000 ~ @E07774 @109000 ~ @199774 @209000 ~ @299774 @309000 ~ @389774					
ファイルリスト (FILE 1)	@00000000 ~ @00077774	@00000000 ~ @00377774	@00000000 ~ @01777774	@00000000 ~ @07777774	@00000000 ~ @37777774

【適用される命令語】

F-04w(D) F-16(D) Fc16(D) F-22(S) F-23(S) F-24(S)など

(5) 使用範囲 F

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
100000、100400、101000、101400 102000、102400、103000、・・・、107000、107400					
/	110000、 110400、 ・ ・ 115000	110000、110400、111000、・・・、153400、154000			
b00000、b00400、b01000、b01400 b02000、b02400、b03000、b03400					
/	b04000、 b04400、 ・ ・ b07400	b04000、b04400、b05000、・・・、b37000、b37400			
009000、009400、・・・、099000、099400 E00000、E00400、・・・、E07000、E07400 109000、109400、・・・、199000、199400 209000、209400、・・・、299000、299400 309000、309400、・・・、389000、389400 Z000、Z200					
ファイルリスト (FILE 1)	00000000、 00000400、 00001000、 ・ ・ 00077400	00000000、 00000400、 00001000、 ・ ・ 00377400	00000000、 00000400、 00001000、 ・ ・ 01777400	00000000、 00000400、 00001000、 ・ ・ 07777400	00000000、 00000400、 00001000、 ・ ・ 37777400

Z000、Z200は、ワード命令(F□w)にのみ使用可能です。

間接アドレス指定のとき

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@100000、@100400、@101000、@101400 @102000、@102400、@103000、・・・、@107000、@107400					
/	@110000、 @110400、 ・ ・ @115000	@110000、@110400、@111000、 ・ ・ @153400、@154000			
@b00000、@b00400、@b01000、@b01400 @b02000、@b02400、@b03000、@b03400					
/	@b04000、 @b04400、 ・ ・ @b07400	@b04000、@b04400、@b05000、 ・ ・ @b37000、@b37400			
@009000、@009400、・・・、@099000、@099400 @E00000、@E00400、・・・、@E07000、@E07400 @109000、@109400、・・・、@199000、@199400 @209000、@209400、・・・、@299000、@299400 @309000、@309400、・・・、@389000、@389400					
ファイルリスト (FILE 1)	@00000000、 @00000400、 @00001000、 ・ ・ @00077400	@00000000、 @00000400、 @00001000、 ・ ・ @00377400	@00000000、 @00000400、 @00001000、 ・ ・ @01777400	@00000000、 @00000400、 @00001000、 ・ ・ @07777400	@00000000、 @00000400、 @00001000、 ・ ・ @37777400

【適用される命令語】

F-05(D)、F-05w(D)、F-06(S)、F-06w(S)

(6) 使用範囲 G

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
100000 ~ 101570 102000 ~ 107570					
110000 ~ 115370	110000 ~ 1154370				
b00000 ~ b01770 b02000 ~ b03770					
b04000 ~ b07770	b04000 ~ b37770				
009000 ~ 099770 E00000 ~ E07770 109000 ~ 199770 209000 ~ 299770 309000 ~ 389770					
ファイルジスタ (FILE 1)	00000000 ~ 00077770	00000000 ~ 00377770	00000000 ~ 01777770	00000000 ~ 07777770	00000000 ~ 37777770

間接アドレス指定のとき

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@100000 ~ @101574 @102000 ~ @107574					
@110000 ~ @115374	@110000 ~ @1154374				
@b00000 ~ @b01774 @b02000 ~ @b03774					
@b04000 ~ @b07774	@b04000 ~ b37774				
@009000 ~ @099774 @E00000 ~ @E07774 @109000 ~ @199774 @209000 ~ @299774 @309000 ~ @389774					
ファイルジスタ (FILE 1)	@00000000 ~ @00077774	@00000000 ~ @00377774	@00000000 ~ @01777774	@00000000 ~ @07777774	@00000000 ~ @37777774

【適用される命令語】

Fc15α D) F-16α D) Fc16α D) F-2α S、D など

(7) 使用範囲 H

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
100000 ~ 101572 102000 ~ 107572					
10000 ~ 115372	110000 ~ 154372				
b00000 ~ b01772 b02000 ~ b03772					
b04000 ~ b07772	b04000 ~ b37772				
009000 ~ 099772 E00000 ~ E07772 109000 ~ 199772 209000 ~ 299772 309000 ~ 389772					
ファイルリスト (FILE 1)	00000000 ~ 00077772	00000000 ~ 00377772	00000000 ~ 01777772	00000000 ~ 07777772	00000000 ~ 37777772

間接アドレス指定のとき

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@100000 ~ @101574 @102000 ~ @107574					
@10000 ~ @115374	@110000 ~ @154374				
@b00000 ~ @b01774 @b02000 ~ @b03774					
@b04000 ~ @b07774	@b04000 ~ b37774				
@009000 ~ @099774 @E00000 ~ @E07774 @109000 ~ @199774 @209000 ~ @299774 @309000 ~ @389774					
ファイルリスト (FILE 1)	@00000000 ~ @00077774	@00000000 ~ @00377774	@00000000 ~ @01777774	@00000000 ~ @07777774	@00000000 ~ @37777774

【適用される命令語】

F-05(S) \ F-11(D) \ F-15(D)

(8) 使用範囲 J

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
100000 ~ 101200 102000 ~ 107200					
110000 ~ 115000	110000 ~ 1154000				
b00000 ~ b01400 b02000 ~ b03400					
b04000 ~ b07400	b04000 ~ b37400				
009000 ~ 099400 E00000 ~ E07400 109000 ~ 199400 209000 ~ 299400 309000 ~ 389400					
ファイルリスト (FILE 1)	00000000 ~ 00077400	00000000 ~ 00377400	00000000 ~ 01777400	00000000 ~ 07777400	00000000 ~ 37777400

間接アドレス指定のとき

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
@100000 ~ @101574 @102000 ~ @107574					
@110000 ~ @115374	@110000 ~ @1154374				
@b00000 ~ @b01774 @b02000 ~ @b03774					
@b04000 ~ @b07774	@b04000 ~ b37774				
@009000 ~ @099774 @E00000 ~ @E07774 @109000 ~ @199774 @209000 ~ @299774 @309000 ~ @389774					
ファイルリスト (FILE 1)	@00000000 ~ @00077774	@00000000 ~ @00377774	@00000000 ~ @01777774	@00000000 ~ @07777774	@00000000 ~ @37777774

【適用される命令語】

F-17(D)

(9) 使用範囲 K

JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU
000000 ~ 015777 020000 ~ 075777	000000 ~ 015777 020000 ~ 075777 100000 ~ 153777			000000 ~ 015777 020000 ~ 075777 100000 ~ 543777	

【適用される命令語】

F-32(OUT) F-33(OUT) F-34(BIT) F-35(BIT)

〔 2 〕 間接アドレス指定

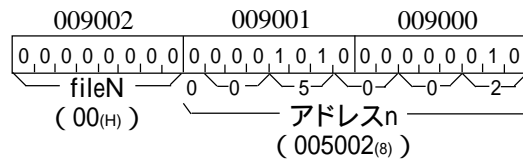
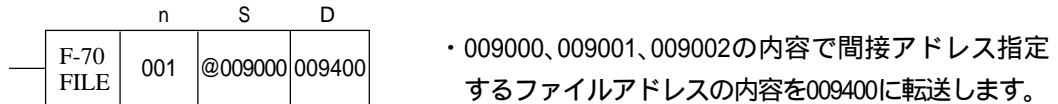
ソース、デスティネーションを間接アドレスで指定すると、指定したバイトアドレス自身が演算を実行するのではなく、そのバイトアドレスを先頭とする3バイトの内容で指定する「fileN、アドレスn」に対応するファイルアドレスが演算を実行します。

間接アドレス指定の「fileN、アドレスn」については、次ページ以降を参照願います。

fileN : 00 ~ 80_(H)、アドレスn : 000000 ~ 177777₍₈₎

間接アドレスを指定するには、バイトアドレスの前に@ (アットマーク) を付加します。

【例】



上例では、file00_(H)のアドレス005002₍₈₎はファイルアドレス00005002₍₈₎(レジスタ019002)で、結果的に@009000は019002を示します。

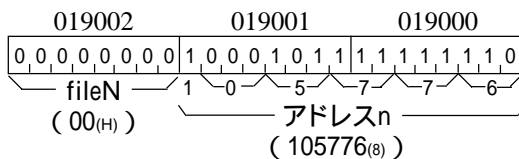
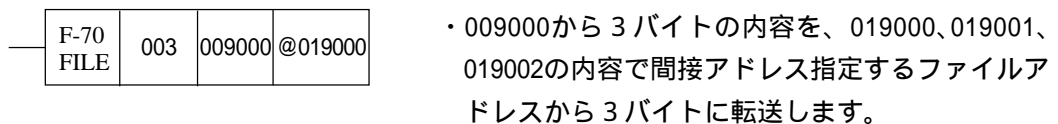


間接アドレス指定には必ず偶数アドレスを設定してください。奇数アドレスを設定すると自動的にアドレスを - 1した偶数アドレスと同じ動作となります。

(@009003と設定すると、@009002と見なす)

間接アドレス指定するアドレスは、各機種(JW-3*CU)のデータメモリ範囲内に設定してください。範囲外の場合、演算は実行されません。

【JW-32*CUの禁止例】



上例では、file00_(H)のアドレス105776₍₈₎はファイルアドレス00105776₍₈₎です。演算後、ファイルアドレス00105776 ~ 00106000₍₈₎の3バイトにデータが転送されますが、00106000₍₈₎は設定範囲外(禁止領域)です。

TMR・CNTの接点領域(ファイルアドレス00001600 ~ 00001777₍₈₎等)内を間接アドレスに指定しないでください。

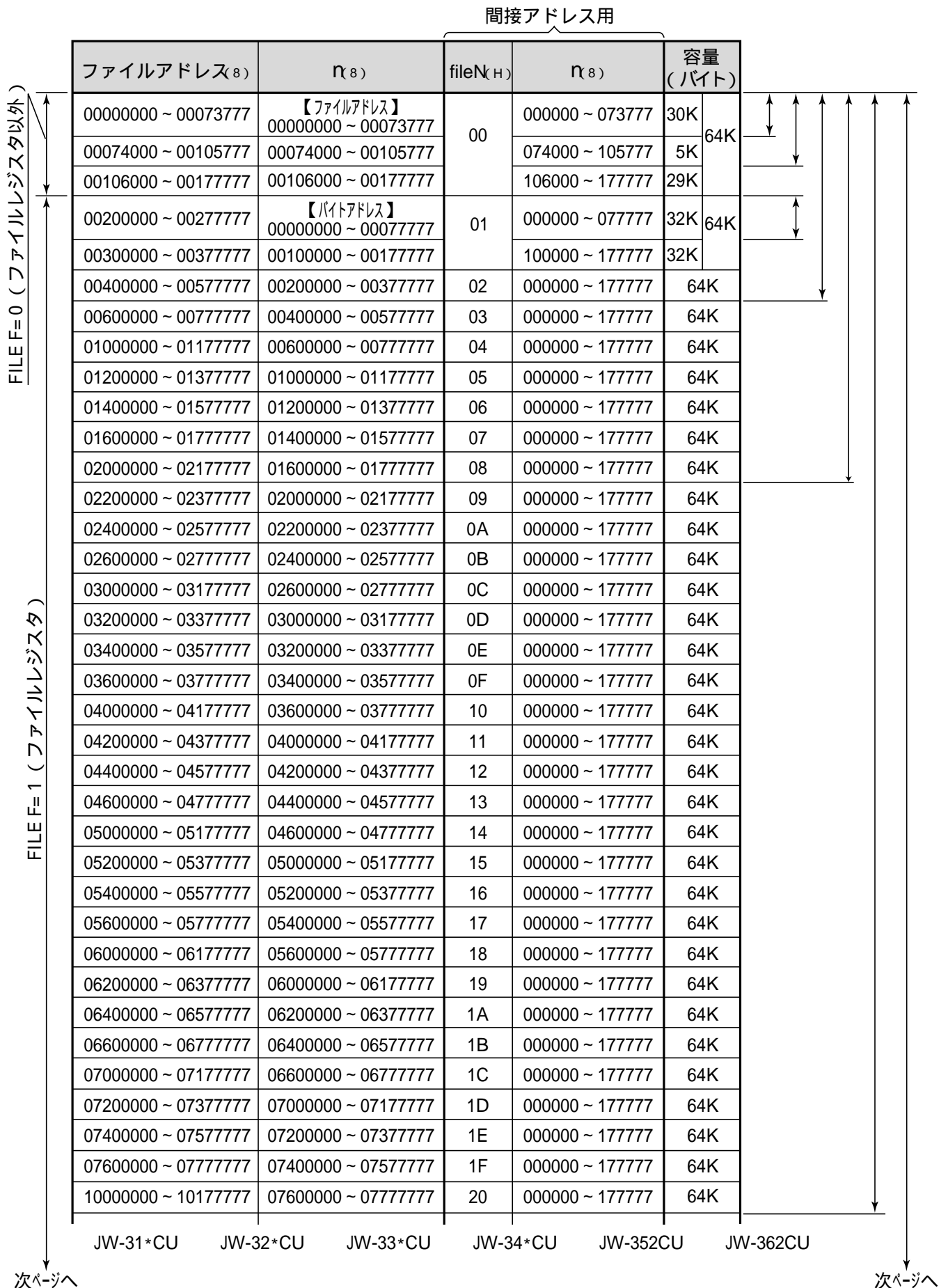
ワード処理命令、ダブルワード処理命令では、間接指定するファイルアドレスには偶数アドレスを設定してください。奇数アドレスを設定すると、自動的に - 1した偶数アドレスと同じ演算となります。

JW-33*CU、JW-34*CU、JW-352CU、JW-362CUの場合、file00_(H)の最終アドレス177777₍₈₎を越えると、演算は実行されません。file01_(H)以降は、次のfile番号の先頭アドレスへ連続し、演算が実行されます。

間接アドレスの「fileN、アドレス n」

間接アドレス指定、応用命令 (F-100、F-101) で使用する「fileN、アドレス n」は、「ファイルアドレス、FILE F、アドレス n」と以下の関係になります。

【例】ファイルアドレス 03100000₍₈₎は、fileN = 0C_(H)のアドレス n = 100000₍₈₎となります。



間接アドレス用

前ページより

前ページより

FILE F=1 (ファイルレジスタ)

ファイルアドレス ⁸⁾	$\text{N}^{(8)}$ 【バイトアドレス】	fileN _H	$\text{N}^{(8)}$	容量 (バイト)
10200000 ~ 10377777	10000000 ~ 10177777	21	000000 ~ 177777	64K
10400000 ~ 10577777	10200000 ~ 10377777	22	000000 ~ 177777	64K
10600000 ~ 10777777	10400000 ~ 10577777	23	000000 ~ 177777	64K
11000000 ~ 11177777	10600000 ~ 10777777	24	000000 ~ 177777	64K
11200000 ~ 11377777	11000000 ~ 11177777	25	000000 ~ 177777	64K
11400000 ~ 11577777	11200000 ~ 11377777	26	000000 ~ 177777	64K
11600000 ~ 11777777	11400000 ~ 11577777	27	000000 ~ 177777	64K
12000000 ~ 12177777	11600000 ~ 11777777	28	000000 ~ 177777	64K
12200000 ~ 12377777	12000000 ~ 12177777	29	000000 ~ 177777	64K
12400000 ~ 12577777	12200000 ~ 12377777	2A	000000 ~ 177777	64K
12600000 ~ 12777777	12400000 ~ 12577777	2B	000000 ~ 177777	64K
13000000 ~ 13177777	12600000 ~ 12777777	2C	000000 ~ 177777	64K
13200000 ~ 13377777	13000000 ~ 13177777	2D	000000 ~ 177777	64K
13400000 ~ 13577777	13200000 ~ 13377777	2E	000000 ~ 177777	64K
13600000 ~ 13777777	13400000 ~ 13577777	2F	000000 ~ 177777	64K
14000000 ~ 14177777	13600000 ~ 13777777	30	000000 ~ 177777	64K
14200000 ~ 14377777	14000000 ~ 14177777	31	000000 ~ 177777	64K
14400000 ~ 14577777	14200000 ~ 14377777	32	000000 ~ 177777	64K
14600000 ~ 14777777	14400000 ~ 14577777	33	000000 ~ 177777	64K
15000000 ~ 15177777	14600000 ~ 14777777	34	000000 ~ 177777	64K
15200000 ~ 15377777	15000000 ~ 15177777	35	000000 ~ 177777	64K
15400000 ~ 15577777	15200000 ~ 15377777	36	000000 ~ 177777	64K
15600000 ~ 15777777	15400000 ~ 15577777	37	000000 ~ 177777	64K
16000000 ~ 16177777	15600000 ~ 15777777	38	000000 ~ 177777	64K
16200000 ~ 16377777	16000000 ~ 16177777	39	000000 ~ 177777	64K
16400000 ~ 16577777	16200000 ~ 16377777	3A	000000 ~ 177777	64K
16600000 ~ 16777777	16400000 ~ 16577777	3B	000000 ~ 177777	64K
17000000 ~ 17177777	16600000 ~ 16777777	3C	000000 ~ 177777	64K
17200000 ~ 17377777	17000000 ~ 17177777	3D	000000 ~ 177777	64K
17400000 ~ 17577777	17200000 ~ 17377777	3E	000000 ~ 177777	64K
17600000 ~ 17777777	17400000 ~ 17577777	3F	000000 ~ 177777	64K
20000000 ~ 20177777	17600000 ~ 17777777	40	000000 ~ 177777	64K
20200000 ~ 20377777	20000000 ~ 20177777	41	000000 ~ 177777	64K
20400000 ~ 20577777	20200000 ~ 20377777	42	000000 ~ 177777	64K
20600000 ~ 20777777	20400000 ~ 20577777	43	000000 ~ 177777	64K
21000000 ~ 21177777	20600000 ~ 20777777	44	000000 ~ 177777	64K
21200000 ~ 21377777	21000000 ~ 21177777	45	000000 ~ 177777	64K
21400000 ~ 21577777	21200000 ~ 21377777	46	000000 ~ 177777	64K
21600000 ~ 21777777	21400000 ~ 21577777	47	000000 ~ 177777	64K
22000000 ~ 22177777	21600000 ~ 21777777	48	000000 ~ 177777	64K
22200000 ~ 22377777	22000000 ~ 22177777	49	000000 ~ 177777	64K
22400000 ~ 22577777	22200000 ~ 22377777	4A	000000 ~ 177777	64K

次ページへ

次ページへ

間接アドレス用

前ページより	ファイルアドレス ⁽⁸⁾	$n^{(8)}$ 【バイトアドレス】	fileN ^(H)	$n^{(8)}$	容量 (バイト)	前ページより
	22600000 ~ 22777777	22400000 ~ 22577777	4B	000000 ~ 177777	64K	
	23000000 ~ 23177777	22600000 ~ 22777777	4C	000000 ~ 177777	64K	
	23200000 ~ 23377777	23000000 ~ 23177777	4D	000000 ~ 177777	64K	
	23400000 ~ 23577777	23200000 ~ 23377777	4E	000000 ~ 177777	64K	
	23600000 ~ 23777777	23400000 ~ 23577777	4F	000000 ~ 177777	64K	
	24000000 ~ 24177777	23600000 ~ 23777777	50	000000 ~ 177777	64K	
	24200000 ~ 24377777	24000000 ~ 24177777	51	000000 ~ 177777	64K	
	24400000 ~ 24577777	24200000 ~ 24377777	52	000000 ~ 177777	64K	
	24600000 ~ 24777777	24400000 ~ 24577777	53	000000 ~ 177777	64K	
	25000000 ~ 25177777	24600000 ~ 24777777	54	000000 ~ 177777	64K	
	25200000 ~ 25377777	25000000 ~ 25177777	55	000000 ~ 177777	64K	
	25400000 ~ 25577777	25200000 ~ 25377777	56	000000 ~ 177777	64K	
	25600000 ~ 25777777	25400000 ~ 25577777	57	000000 ~ 177777	64K	
	26000000 ~ 26177777	25600000 ~ 25777777	58	000000 ~ 177777	64K	
	26200000 ~ 26377777	26000000 ~ 26177777	59	000000 ~ 177777	64K	
	26400000 ~ 26577777	26200000 ~ 26377777	5A	000000 ~ 177777	64K	
	26600000 ~ 26777777	26400000 ~ 26577777	5B	000000 ~ 177777	64K	
	27000000 ~ 27177777	26600000 ~ 26777777	5C	000000 ~ 177777	64K	
	27200000 ~ 27377777	27000000 ~ 27177777	5D	000000 ~ 177777	64K	
	27400000 ~ 27577777	27200000 ~ 27377777	5E	000000 ~ 177777	64K	
	27600000 ~ 27777777	27400000 ~ 27577777	5F	000000 ~ 177777	64K	
	30000000 ~ 30177777	27600000 ~ 27777777	60	000000 ~ 177777	64K	
	30200000 ~ 30377777	30000000 ~ 30177777	61	000000 ~ 177777	64K	
	30400000 ~ 30577777	30200000 ~ 30377777	62	000000 ~ 177777	64K	
	30600000 ~ 30777777	30400000 ~ 30577777	63	000000 ~ 177777	64K	
	31000000 ~ 31177777	30600000 ~ 30777777	64	000000 ~ 177777	64K	
	31200000 ~ 31377777	31000000 ~ 31177777	65	000000 ~ 177777	64K	
	31400000 ~ 31577777	31200000 ~ 31377777	66	000000 ~ 177777	64K	
	31600000 ~ 31777777	31400000 ~ 31577777	67	000000 ~ 177777	64K	
	32000000 ~ 32177777	31600000 ~ 31777777	68	000000 ~ 177777	64K	
	32200000 ~ 32377777	32000000 ~ 32177777	69	000000 ~ 177777	64K	
	32400000 ~ 32577777	32200000 ~ 32377777	6A	000000 ~ 177777	64K	
	32600000 ~ 32777777	32400000 ~ 32577777	6B	000000 ~ 177777	64K	
	33000000 ~ 33177777	32600000 ~ 32777777	6C	000000 ~ 177777	64K	
	33200000 ~ 33377777	33000000 ~ 33177777	6D	000000 ~ 177777	64K	
	33400000 ~ 33577777	33200000 ~ 33377777	6E	000000 ~ 177777	64K	
	33600000 ~ 33777777	33400000 ~ 33577777	6F	000000 ~ 177777	64K	
	34000000 ~ 34177777	33600000 ~ 33777777	70	000000 ~ 177777	64K	
	34200000 ~ 34377777	34000000 ~ 34177777	71	000000 ~ 177777	64K	
	34400000 ~ 34577777	34200000 ~ 34377777	72	000000 ~ 177777	64K	
	34600000 ~ 34777777	34400000 ~ 34577777	73	000000 ~ 177777	64K	
	35000000 ~ 35177777	34600000 ~ 34777777	74	000000 ~ 177777	64K	

FILE F=1 (ファイルレジスタ)

次ページへ

次ページへ

前ページより	間接アドレス用				前ページより
	ファイルアドレス ⁸⁾	n_8 【バイトアドレス】	fileN _H	n_8	
FILE F=1 (ファイルレジスタ)	35200000 ~ 35377777	35000000 ~ 35177777	75	000000 ~ 177777	64K
	35400000 ~ 35577777	35200000 ~ 35377777	76	000000 ~ 177777	64K
	35600000 ~ 35777777	35400000 ~ 35577777	77	000000 ~ 177777	64K
	36000000 ~ 36177777	35600000 ~ 35777777	78	000000 ~ 177777	64K
	36200000 ~ 36377777	36000000 ~ 36177777	79	000000 ~ 177777	64K
	36400000 ~ 36577777	36200000 ~ 36377777	7A	000000 ~ 177777	64K
	36600000 ~ 36777777	36400000 ~ 36577777	7B	000000 ~ 177777	64K
	37000000 ~ 37177777	36600000 ~ 36777777	7C	000000 ~ 177777	64K
	37200000 ~ 37377777	37000000 ~ 37177777	7D	000000 ~ 177777	64K
	37400000 ~ 37577777	37200000 ~ 37377777	7E	000000 ~ 177777	64K
	37600000 ~ 37777777	37400000 ~ 37577777	7F	000000 ~ 177777	64K
	40000000 ~ 40177777	37600000 ~ 37777777	80	000000 ~ 177777	64K

9 - 2 インデックス修飾機能

インデックスレジスタZ000 ~ Z377(以下、Z_xxx)を、基本命令・応用命令で直接指定するリレー・レジスタ等(次ページの適用領域)にインデックス修飾すると、Z_xxxの内容を加減算したアドレスで演算が実行されます。

〔1〕インデックス修飾のプログラム方法

Z_xxxによるインデックス修飾のプログラム方法には、「通常修飾」と「自動修飾」があります。

(1) 通常修飾

「リレー・レジスタ等、Z_xxx」をプログラムすると、その命令を実行時、リレー・レジスタ等のアドレスにZ_xxxの内容を加減算したアドレスで演算されます。

加減算は、Z_xxxの内容の最上位(符号)ビットで決まり、0のとき「加算」、1のとき「減算」となります。

Z_xxxの内容

Z_xxxは1レジスタで2バイトあり、Z_xxxの内容には符号付き2進数(-77777 ~ +77777₍₈₎)を設定します。最上位ビットが符号(+/-)になります。

Z_xxxの内容		
符号付き2進数	8進数	10進数
01111111 11111111	→ +77777	→ +32767
01111111 11111110	→ +77776	→ +32766
⋮		
00000000 11111111	→ +377	→ +255
⋮		
00000000 00000010	→ +2	→ +2
00000000 00000001	→ +1	→ +1
00000000 00000000	→ 0	→ 0
11111111 11111111	→ -1	→ -1
11111111 11111110	→ -2	→ -2
⋮		
11111111 00000000	→ -400	→ -256
⋮		
10000000 00000001	→ -77776	→ -32766
10000000 00000000	→ -77777	→ -32767

負の数値は、「2の補数」をとります。

↑ 符号ビット(1 = -, 0 = +)

・「2の補数」とは、2進数で表したデータのすべてのビットを反転(0 1、1 0)させ、1を加算した値です。

【例】11111111 00000000のとき

11111111 00000000

ビット反転

00000000 11111111

1を加算

00000001 00000000

8進数に変換(符号: -)

-400₍₈₎

(2) 自動修飾

「リレー・レジスタ等、Z_xxx + yy」または「リレー・レジスタ等、Z_xxx - yy」をプログラムすると、「通常修飾」と同じ演算が実行された後、Z_xxxの内容(上記)に指定値yy(0 ~ 255)が加減算されます。

- ・基本命令にyyを付加した場合、アキュムレータ(ACC)の内容に関わらず、加減算されます。
- ・応用命令にyyを付加した場合、その命令が実行されたときのみ加減算されます。

〔 2 〕 インデックス修飾の適用領域

Z_xxxは、次表のデータメモリ、TMR・CNT番号、ラベル番号の領域でインデックス修飾が可能です。
 なお、コントロールユニットの機種(JW-3**CU)により、適用領域は異なります。

領 域		JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU	
データメモリ	リレー ¹	00000 ~ 01577 (00000 ~ 015777)						} 2
		02000 ~ 07577 (02000 ~ 075777)						
		/		01000 ~ 015377 (100000 ~ 153777)	01000 ~ 054377 (100000 ~ 543777)			
	TMR・CNT 接点	T / C 00000 ~ 00777						
		T / C 01000 ~ 01777						
		/		T / C 02000 ~ 03777	T / C 02000 ~ 17777			
	TMR・CNT ¹ ・MD現在値	b00000 ~ b01777						
		b02000 ~ b03777						
		/		b04000 ~ b07777	b04000 ~ b37777			
	レジスタ ¹	009000 ~ E07777						
109000 ~ 389777								
TMR・CNT番号	00000 ~ 00777							
	01000 ~ 01777							
	/		02000 ~ 03777	02000 ~ 17777				
ラベル番号	LB0000 ~ LB1777							

1 リレー、レジスタは、間接アドレスにもインデックス修飾を行えます。

2 ()内はリレー番号です。

(注)インデックス修飾は、上表の各領域(~)内で加減算が行われます。

- ・Z_xxxの内容を加減算したアドレス(番号)が各領域の範囲外るとき、その演算は実行されません。
- ・「自動修飾」の場合、指定値 yy の加減算が繰り返えされて、アドレス(番号)が各領域の範囲外になると、演算は範囲外になる前のアドレス(番号)で実行されます。

留 意 点

- ・特殊リレー、インデックスレジスタ Z_xxx にはインデックス修飾しないでください。
- ・倍長演算機能を有する応用命令にて、レジスタにインデックス修飾する場合、加減算後のアドレスが必ず偶数アドレスとなるようにプログラムしてください。
- ・応用命令の定数には、インデックス修飾できません。
- ・AND STR、PUSH 等の設定アドレスが無い基本命令には、インデックス修飾できません。
- ・インデックス修飾したラベルで、ジャンプするラベルが存在しない場合、演算は実行されません。
- ・インデックス修飾した命令語の処理時間は長くなります。(命令語の処理時間 7・20 ページ)

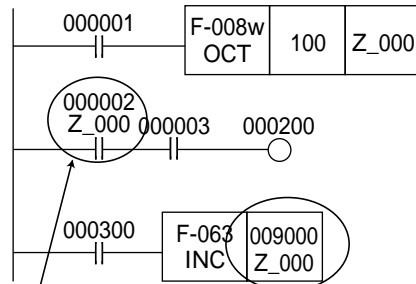
〔 3 〕 インデックス修飾のプログラム例

(1) 通常修飾の場合

「リレー等, Z_xxx」をプログラムすると、その命令を実行時、リレー等のアドレスにZ_xxxの内容を加減算したアドレスで演算されます。

加減算は、Z_xxxの内容の最上位(符号)ビットで決まり、0のとき「加算」、1のとき「減算」となります。

加算(通常修飾)の例



STR	000001
F-008w	100
	Z_000
STR	000002, Z_000
AND	000003
OUT	000200
STR	000300
F-063	009000, Z_000

リレー番号 000002にZ_000(+100₈)を加算したSTR 000102を実行します。

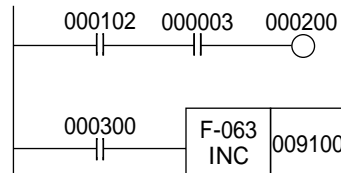
レジスタ009000にZ_000(+100₈)を加算したレジスタ009100に演算内容を格納します。

Z_000の内容

符号付き 2 進数	8 進数
00000000 01000000	→ +100

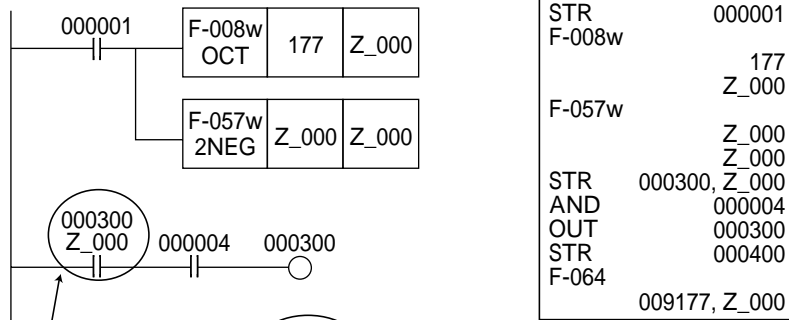
↑ 符号ビット(0 = +)

・ 上例のとき、下記と同じ演算になります。



STR	000002
AND	000003
OUT	000200
STR	000300
F-063	009100

減算(通常修飾)の例



```

STR 000001
F-008w 177 Z_000
F-057w Z_000 Z_000
STR 000300, Z_000
AND 000004 000004
OUT 000300 000300
STR 000400
F-064 009177, Z_000
    
```

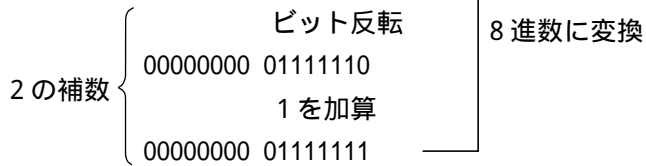
リレー番号000300にZ_000(-177₈)を減算したSTR 000101を実行します。

レジスタ009177にZ_000(-177₈)を減算したレジスタ009000に演算内容を格納します。

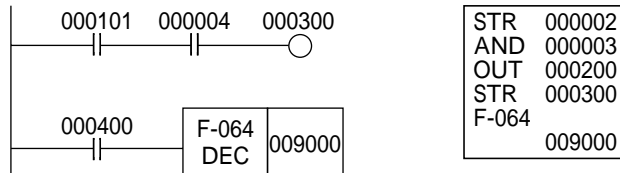
Z_000の内容

符号付き2進数	8進数
11111111 10000001	- 177

↑ 符号ビット(1 = -)



・上例のとき、下記と同じ演算になります。



```

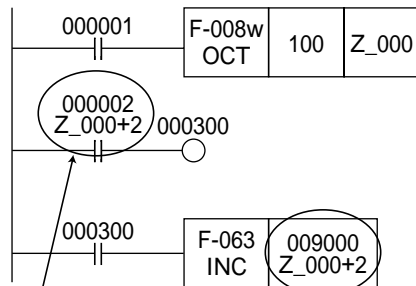
STR 000002
AND 000003
OUT 000200
STR 000300
F-064 009000
    
```

(2) 自動修飾の場合

「リレー等, Z_xxx + yy」または「リレー等, Z_xxx - yy」をプログラムすると、通常修飾と同様に演算が実行後、Z_xxxの内容に指定値yyが加減算されます。

- ・基本命令では、演算の実行 / 非実行に関わらず、yyが加減算されます。
- ・応用命令では、その命令が実行されたときのみ、yyが加減算されます。

加算(自動)の例



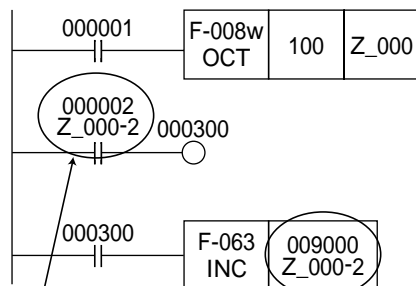
```

STR      000001
F-008w
          100
          Z_000
STR      000002, Z_000+2
AND NOT  000003
STR      000300
F-063
          009000, Z_000+2
    
```

リレー番号000002にZ_000(+100₈)を加算したSTR 000102を実行します。命令が実行後、Z_000の内容に2が加算されて+102₈となります。

レジスタ009000にZ_000(+102₈)を加算したレジスタ09102に演算内容を格納します。命令が実行後、Z_000の内容に2が加算されて+104₈となります。

減算(自動)の例



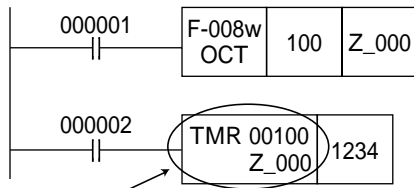
```

STR      000001
F-008w
          100
          Z_000
STR      000002, Z_000-2
AND NOT  000003
STR      000300
F-063
          009000, Z_000-2
    
```

リレー番号000002にZ_000(+100₈)を加算したSTR 000102を実行します。命令が実行後、Z_000の内容に2が減算されて+076₈となります。

レジスタ009076にZ_000(+076₈)を加算したレジスタ009076に演算内容を格納します。命令が実行後、Z_000の内容に2が減算されて+074₈となります。

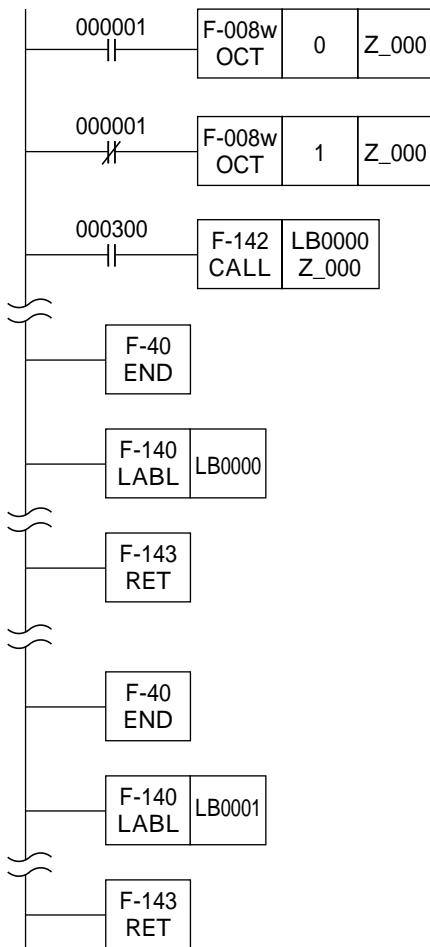
(3) TMR・CNT番号をインデックス修飾する例



STR	000001
F-008w	100
	Z_000
STR	000002
TMR	00100, Z_000
	1234

TMR番号00100にZ_000(+100₈)を加算したTMR番号00200を実行します。

(4) ラベル番号をインデックス修飾する例



STR	000001
F-008w	000
	Z_000
STR NOT	000001
F-008w	001
	Z_000
STR	000300
F-142	LB0000, Z_000
.....	
F-40	
F-140	LB0000
.....	
F-143	
.....	
F-40	
F-140	LB0000
.....	
F-143	

- ・リレー000001がONのときラベル番号LB0000のサブルーチンに移り、リレー 000001がOFFのときラベル番号LB0001のサブルーチンに移ります。

9 - 3 倍長演算機能

(1) 倍長演算の対応命令

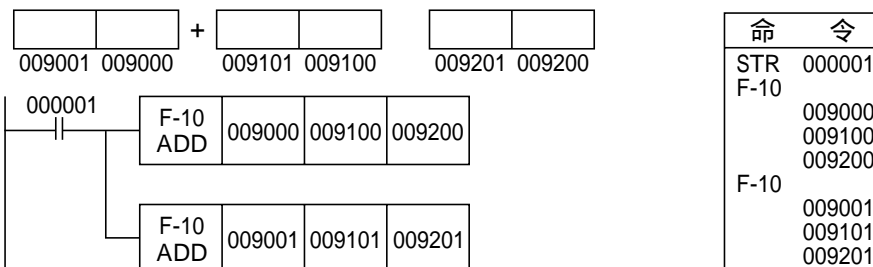
次の応用命令(21種類)には、2バイト以上(バイト命令)、4バイト以上(ワード命令)、8バイト以上(ダブルワード命令)のデータ演算を可能とする倍長演算の機能が有ります。

- F-10、F-10w、F-10d - - - レジスタ間の加算
- Fc10、Fc10w、Fc10d - - - レジスタとBCD定数の加算
- F-11、F-11w、F-11d - - - レジスタ間の減算
- Fc11、Fc11w、Fc11d - - - レジスタとBCD定数の減算
- F-12、F-12w、F-12d - - - レジスタ間の比較
- Fc12、Fc12w、Fc12d - - - レジスタと8進定数の比較
- Fx12、Fx12w、Fx12d - - - レジスタと16進定数の比較

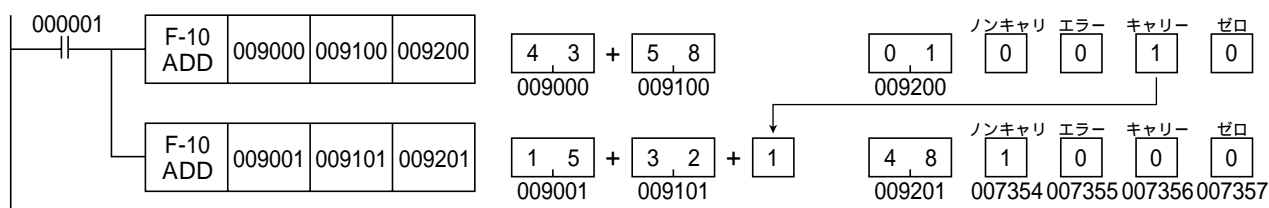
(2) 倍長演算時のプログラム

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機能です。倍長演算時のプログラムは、下記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作成します。

【例1】

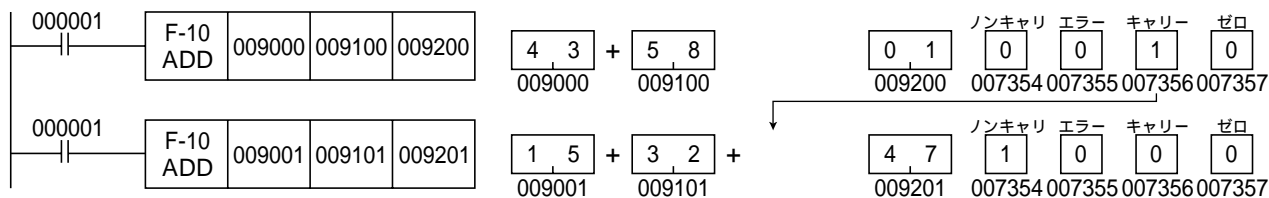


1 5 4 3 + 3 2 5 8 の場合
009001 009000 009101 009100

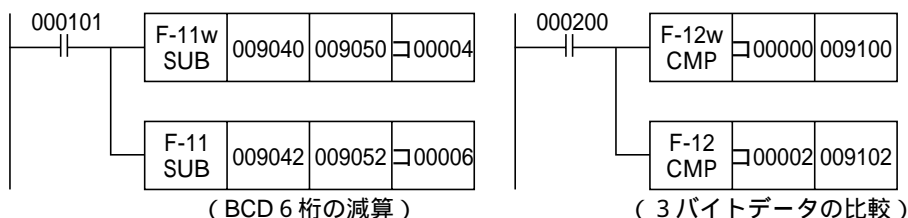


上の桁の演算時に下の桁のキャリーフラグを加算します。

【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

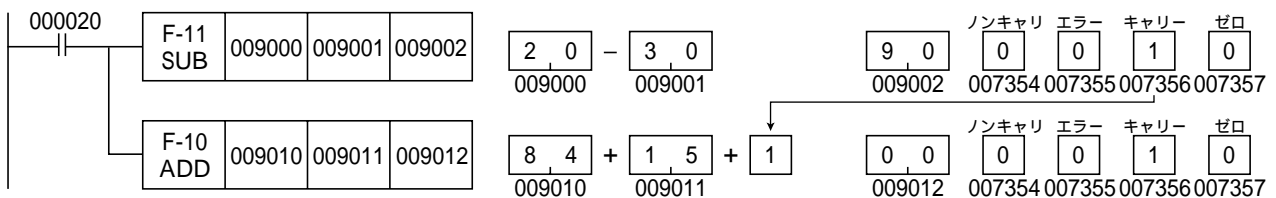
演算実行条件の以後、最初にプログラムしている倍長演算対応命令(21種類：前ページ)は、それ以前のフラグ状態を含めずに演算します。

共通の演算実行条件で、次に倍長演算対応命令のいずれかの命令があると、下記のように演算します。

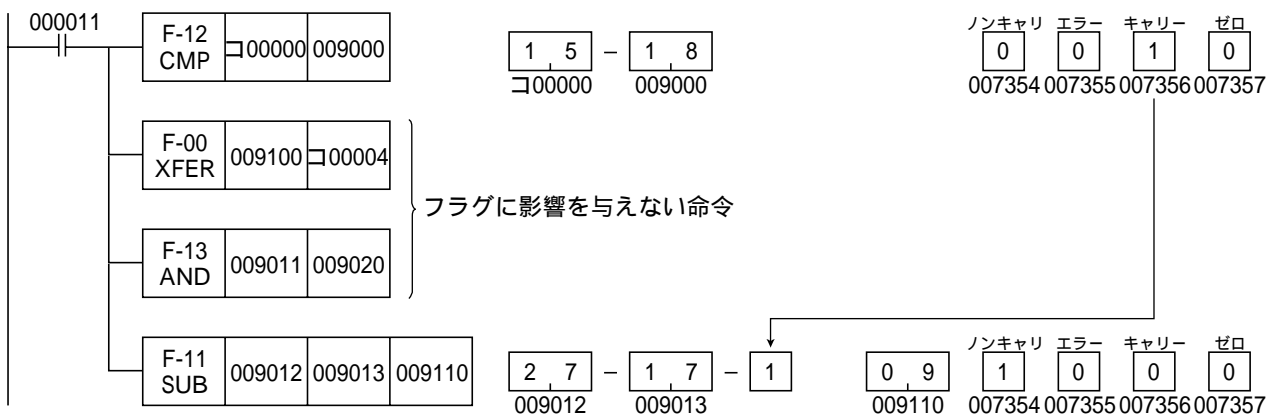
1. 直前のキャリーフラグの状態を含めて、演算を実行します。
 - ・ F-10、F-10w、F-10d、Fc10、Fc10w、Fc10d - - 直前のキャリーフラグの状態を加算
 - ・ F-11、F-11w、F-11d、Fc11、Fc11w、Fc11d - - 直前のキャリーフラグの状態を減算
 - ・ F-12、F-12w、F-12d、Fc12、Fc12w、Fc12d、Fx12、Fx12w、Fx12d - - 直前のキャリーフラグの状態を減算
2. ゼロフラグは、「直前のゼロフラグの状態」と「当該命令の演算によるゼロフラグの状態」をANDし、両方が1のときにセットされます。

(4) 倍長演算に関する注意事項

倍長演算対応命令(21種類：前ページ)は、共通の演算条件でプログラムすると、異種命令間でもフラグを含んだ演算を行います。

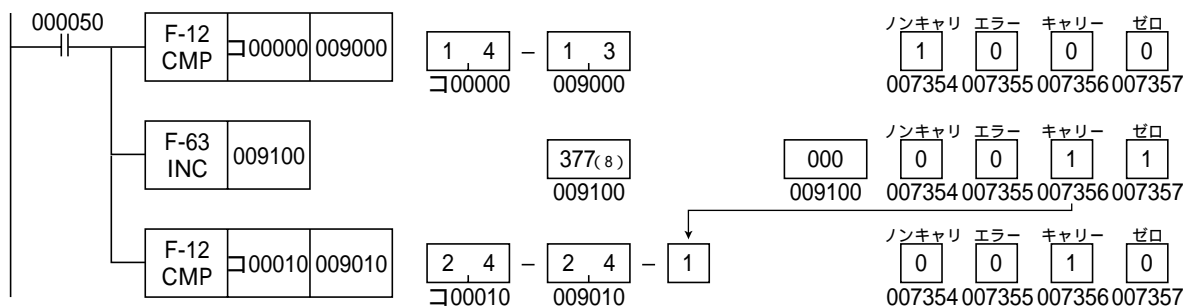


倍長演算対応命令の間に、フラグに影響を与えない命令があっても、倍長演算として実行します。



・ 多数の命令が間に入る場合、特にご注意ください。

倍長演算対応命令の間に、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。



F-10(w,d)、Fc10(w,d)、F-11(w,d)、Fc11(w,d)で、BCDコード以外を使用すると、エラーフラグがONし、それ以後の倍長演算は実行しません。

9 - 4 符号付き演算機能

(1) 符号付き演算の対応命令

次の応用命令(4種類)には、符号付き演算の機能があります。

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

(2) 符号付き演算の数値範囲

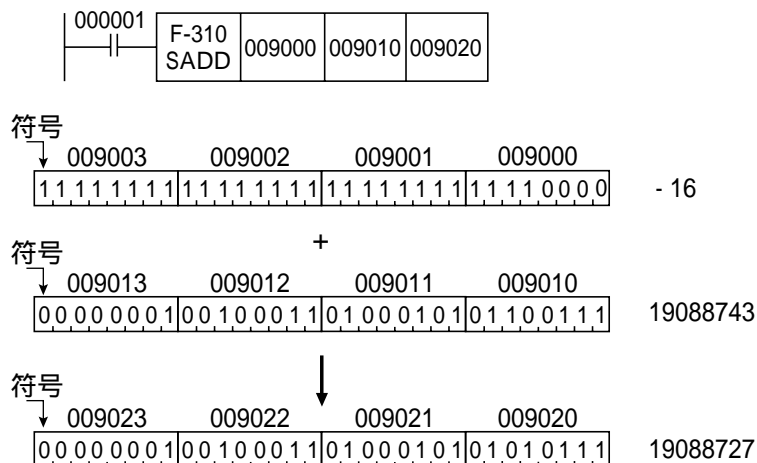
符号付き演算で扱える数値の範囲は、2147483647 ~ - 2147483648 です。

最上位ビットが符号(+/-)になり、負の数値は「2の補数」で表現します。

符号付き2進数(2の補数)	10進数
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

↑ 符号ビット(1 = -, 0 = +)

【例】31ビット+31ビット



上記の演算は、- 16 + 19088743 = 19088727 を示します。

9 - 5 データメモリのブロックと基準アドレス

次の応用命令(12種類)では、基準アドレスを使用します。

- ・F-05、F-05w、F-05d(分配) ・F-06、F-06w、F-06d(抽出)
- ・F-72、F-72w、F-72d(ファイルレジスタへの分配)
- ・F-73、F-73w、F-73d(ファイルレジスタからの抽出)

基準アドレスとは、データメモリ(2・1・5ページ)を256バイト(1ブロック)で分割したときの、各ブロックの先頭アドレスを示します。

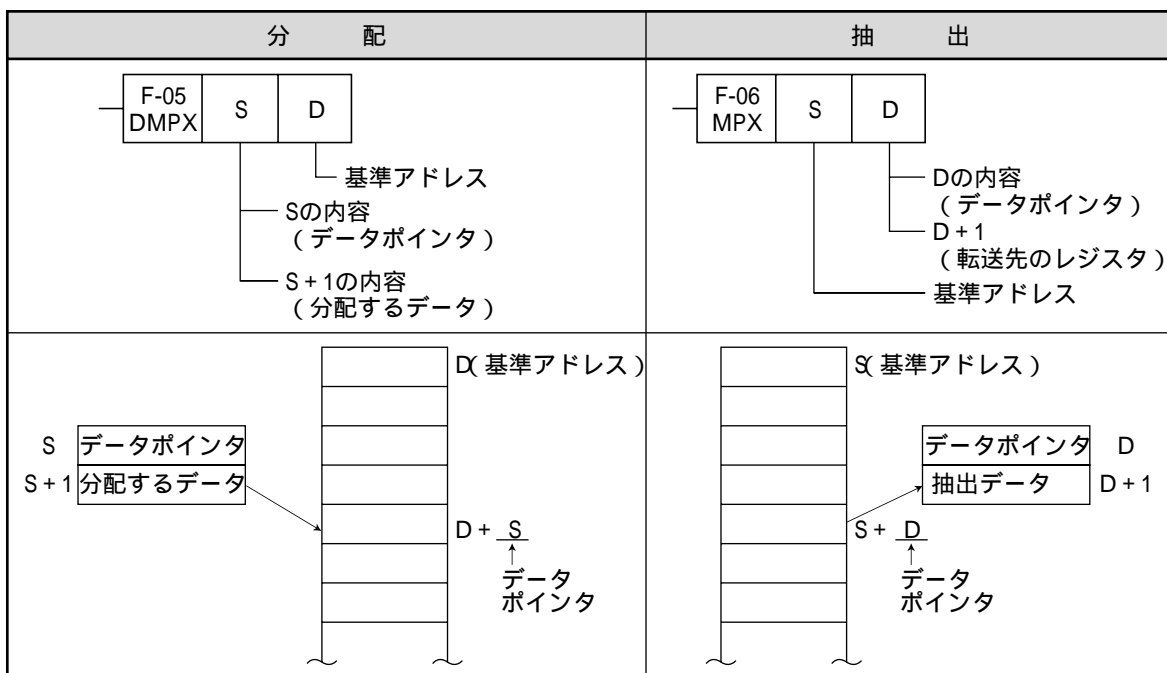
ブロック	基準アドレス	範囲	ブロック	基準アドレス	範囲
リレー	コ00000	コ00000 ~ コ00377	レジスタ	009000	009000 ~ 009377
	コ00400	コ00400 ~ コ00777		009400	009400 ~ 009777
	コ01000	コ01000 ~ コ01377		019000	019000 ~ 019377
	コ01400 1	コ01400 ~ コ01577		⋮	⋮
	コ02000	コ02000 ~ コ02377		099400	099400 ~ 099777
	コ02400	コ02400 ~ コ02777		E0000	E0000 ~ E0377
	⋮	⋮		E0400	E0400 ~ E0777
	コ07000	コ07000 ~ コ07377		E1000	E1000 ~ E1377
	コ07400 2	コ07400 ~ コ07577		⋮	⋮
	コ10000	コ10000 ~ コ10377		E7400	E7400 ~ E7777
	コ10400	コ10400 ~ コ10777		109000	109000 ~ 109377
	⋮	⋮		109400	109400 ~ 109777
	コ15000	コ15000 ~ コ15377		119000	119000 ~ 119377
	コ15400	コ15400 ~ コ15777		⋮	⋮
	⋮	⋮		209000	209000 ~ 209377
	コ54000	コ54000 ~ コ54377		⋮	⋮
TMR ・CNT ・MD 現在値	b00000	b00000 ~ b00377	309000	309000 ~ 309377	
	b00400	b00400 ~ b00777	⋮	⋮	
	b01000	b01000 ~ b01377	⋮	⋮	
	⋮	⋮	389400	389400 ~ 389777	
	b03400	b03400 ~ b03777	Z000 3	Z000 ~ Z177	
	b04000	b04000 ~ b04377	Z200 3	Z200 ~ Z377	
	⋮	⋮	ファイル レジスタ	00000000	00000000 ~ 00000377
	⋮	⋮		00000400	00000400 ~ 00000777
	b07400	b07400 ~ b07777		00001000	00001000 ~ 00001377
	b10000	b10000 ~ b10377		⋮	⋮
⋮	⋮	00077400		00077400 ~ 00077777	
⋮	⋮	00100000		00100000 ~ 00100377	
b37400	b37400 ~ b37777	⋮		⋮	
⋮	⋮	00377400		00377400 ~ 00377777	
⋮	⋮	00400000		00400000 ~ 00400377	
⋮	⋮	⋮		⋮	
⋮	⋮	01777400	01777400 ~ 01777777		
⋮	⋮	02000000	02000000 ~ 02000377		
⋮	⋮	⋮	⋮		
⋮	⋮	07777400	07777400 ~ 07777777		
⋮	⋮	10000000	10000000 ~ 10000377		
⋮	⋮	⋮	⋮		
⋮	⋮	37777400	37777400 ~ 37777777		

JW-31*CU
JW-32*CU
JW-33*CU
JW-34*CU
JW-352CU
JW-362CU

- 1 コ01400 ~ コ01577のブロックは、128バイトです。
- 2 コ07400 ~ コ07577のブロックは、128バイトです。
- 3 Z000、Z200はワード命令(F-05w、F-06w、F-72w、F-73w)にのみ使用可能です。

基準アドレスを使用する応用命令(前ページの12種類)は、レジスタ間でデータを転送する命令で、「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します。

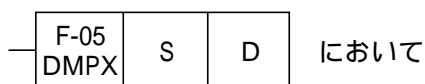
F-05、F-06 の場合



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($\text{C}00000$ 、 \dots 、 $\text{b}00000$ 、 \dots 、 $\text{0}09000$ 等)を設定します。ブロックの先頭アドレス以外のアドレスも設定できますが、JW300 の演算では、そのアドレスを含む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】

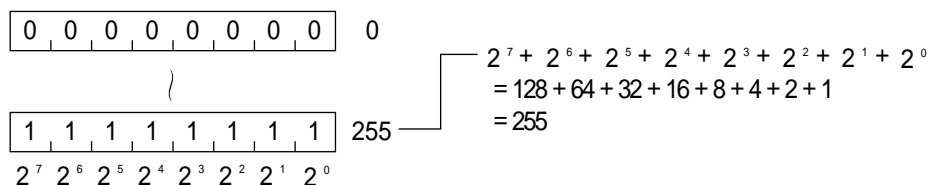


Dの設定	演算上の基準アドレス
$\text{C}00200$	$\text{C}00000$
$\text{b}00110$	$\text{b}00000$
009005	009000

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、 $0 \sim 255$ ($0 \sim 377_{(8)}$) の値になります。



「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・抽出元に設定できます。

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・抽出元のアドレスを変化させられます。

基準アドレス-----009000

データポインタの内容 ---

1	0	1	1	0	0	1	1
---	---	---	---	---	---	---	---

 (10進数では179)

$\underbrace{\hspace{2em}}_2 \quad \underbrace{\hspace{2em}}_6 \quad \underbrace{\hspace{2em}}_3$
 $009000 + 263_{(8)} = 009263$ が分配先(抽出元)

第 10 章 応用命令 (F-00 ~ Fx14d)

F-00 XFER

1 バイトデータの転送

シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">F-00</td> <td style="padding: 2px;">S</td> <td style="padding: 2px;">D</td> </tr> <tr> <td style="text-align: center;">XFER</td> <td></td> <td></td> </tr> </table>	F-00	S	D	XFER				[使用例]		
F-00	S	D									
XFER											
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。		<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">004004</td> <td style="padding: 2px;">F-00</td> <td style="padding: 2px;">009000</td> <td style="padding: 2px;">J00001</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">XFER</td> <td></td> <td></td> </tr> </table>	004004	F-00	009000	J00001		XFER		
004004	F-00	009000	J00001								
	XFER										
演算内容	S → D		<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">STR</td> <td style="padding: 2px;">004004</td> </tr> <tr> <td style="padding: 2px;">F-00</td> <td style="padding: 2px;">009000</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">J00001</td> </tr> </table>	STR	004004	F-00	009000		J00001		
STR	004004										
F-00	009000										
	J00001										
S	使用範囲 A		入力条件004004がOFF ONの変化時に、レジスタ009000の内容を、レジスタJ00001に転送します。 <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">009000</td> <td style="text-align: center;">→</td> <td style="text-align: center;">J00001</td> </tr> <tr> <td style="text-align: center;">0 1 1 1 0 1 0 1</td> <td></td> <td style="text-align: center;">0 1 1 1 0 1 0 1</td> </tr> </table>	009000	→	J00001	0 1 1 1 0 1 0 1		0 1 1 1 0 1 0 1		
009000	→	J00001									
0 1 1 1 0 1 0 1		0 1 1 1 0 1 0 1									
D	使用範囲 A										
演算条件	入力信号の立上り(OFF ON)										
演算後の内容	S	不変									
	D	レジスタSの内容									
	フラグ	不変									

(類似命令) F-00w、F-00d、F-70、F-70w、F-70d、F-74、F-74w、F-74d、F-76、F-76w、F-76d

F-00w XFER

1 ワードデータの転送

シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">F-00w</td> <td style="padding: 2px;">S</td> <td style="padding: 2px;">D</td> </tr> <tr> <td style="text-align: center;">XFER</td> <td></td> <td></td> </tr> </table>	F-00w	S	D	XFER				[使用例]				
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。		<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">004000</td> <td style="padding: 2px;">F-00w</td> <td style="padding: 2px;">009000</td> <td style="padding: 2px;">J00000</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">XFER</td> <td></td> <td></td> </tr> </table>	004000	F-00w	009000	J00000		XFER				
004000	F-00w	009000	J00000										
	XFER												
演算内容	S、S+1 → D、D+1		<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">STR</td> <td style="padding: 2px;">004000</td> </tr> <tr> <td style="padding: 2px;">F-00w</td> <td style="padding: 2px;">009000</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">J00000</td> </tr> </table>	STR	004000	F-00w	009000		J00000				
STR	004000												
F-00w	009000												
	J00000												
S	使用範囲 B		入力条件004000がOFF ONの変化時に、レジスタ009000、009001の内容(1ワードデータ)を、レジスタJ00000、J00001に転送します。 <table style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">009001</td> <td style="text-align: center;">009000</td> </tr> <tr> <td style="text-align: center;">0 1 1 0 1 0 1 0</td> <td style="text-align: center;">1 0 1 0 1 1 1 0 0</td> </tr> <tr> <td colspan="2" style="text-align: center;">↓</td> </tr> <tr> <td style="text-align: center;">J00001</td> <td style="text-align: center;">J00000</td> </tr> <tr> <td style="text-align: center;">0 1 1 0 1 0 1 0</td> <td style="text-align: center;">1 0 1 0 1 1 1 0 0</td> </tr> </table>	009001	009000	0 1 1 0 1 0 1 0	1 0 1 0 1 1 1 0 0	↓		J00001	J00000	0 1 1 0 1 0 1 0	1 0 1 0 1 1 1 0 0
009001	009000												
0 1 1 0 1 0 1 0	1 0 1 0 1 1 1 0 0												
↓													
J00001	J00000												
0 1 1 0 1 0 1 0	1 0 1 0 1 1 1 0 0												
D	使用範囲 B												
演算条件	入力信号の立上り(OFF ON)												
演算後の内容	S、S+1	不変											
	D	レジスタSの内容											
	D+1	レジスタS+1の内容											
	フラグ	不変											

S、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

(類似命令) F-00、F-00d、F-70、F-70w、F-70d、F-74、F-74w、F-74d、F-76、F-76w、F-76d

**F-00d
XFER**

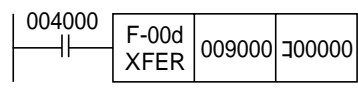
2ワードデータの転送

シンボル	<table border="1"> <tr> <td>F-00d XFER</td> <td>S</td> <td>D</td> </tr> </table>	F-00d XFER	S	D
F-00d XFER	S	D		
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			
S	使用範囲C			
D	使用範囲C			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S~S+3	不変		
	D~D+3	レジスタS~S+3の内容		
	フラグ	不変		

S、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

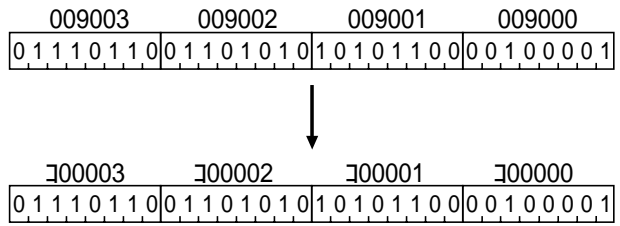
(類似命令)F-00、F-00w、F-70、F-70w、F-70d、F-74、F-74w、F-74d、F-76、F-76w、F-76d

[使用例]



命 令	
STR	004000
F-00d	009000 100000

入力条件004000がOFF ONの変化時に、レジスタ009000~009003の内容(2ワードデータ)を、レジスタ100000~100003に転送します。



F-01 BCD

BCD定数(2桁)の転送

シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">F-01 BCD</td> <td style="padding: 2px;">n</td> <td style="padding: 2px;">D</td> </tr> </table>	F-01 BCD	n	D	<p>[使用例]</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="margin-right: 10px;"> <table border="1" style="border-collapse: collapse;"> <tr><td style="padding: 2px;">004004</td></tr> </table> </div> <div style="margin-right: 10px;"> <table border="1" style="border-collapse: collapse;"> <tr><td style="padding: 2px;">F-01 BCD</td></tr> <tr><td style="padding: 2px;">15</td></tr> <tr><td style="padding: 2px;">009100</td></tr> </table> </div> </div> <table border="1" style="border-collapse: collapse; margin-left: auto;"> <thead> <tr><th colspan="2" style="text-align: center;">命 令</th></tr> </thead> <tbody> <tr><td style="padding: 2px;">STR</td><td style="padding: 2px;">004001</td></tr> <tr><td style="padding: 2px;">F-01</td><td style="padding: 2px;">15</td></tr> <tr><td></td><td style="padding: 2px;">009100</td></tr> </tbody> </table> <p style="font-size: small;">入力条件004004がOFF ONの変化時に、BCD定数15をレジスタ009100に転送します。</p> <div style="text-align: center; margin-top: 10px;"> <table border="1" style="border-collapse: collapse; margin: 0 auto;"> <tr><td style="padding: 2px;">009100</td></tr> <tr><td style="padding: 2px;">0 0 0 1 0 1 0 1</td></tr> <tr><td style="padding: 2px;">└───┬───┬───┬───┬───┬───┬───┬───┘</td></tr> <tr><td style="padding: 2px;">1 5</td></tr> </table> <p style="margin-top: 5px;">転送後</p> </div>	004004	F-01 BCD	15	009100	命 令		STR	004001	F-01	15		009100	009100	0 0 0 1 0 1 0 1	└───┬───┬───┬───┬───┬───┬───┬───┘	1 5
F-01 BCD	n	D																			
004004																					
F-01 BCD																					
15																					
009100																					
命 令																					
STR	004001																				
F-01	15																				
	009100																				
009100																					
0 0 0 1 0 1 0 1																					
└───┬───┬───┬───┬───┬───┬───┬───┘																					
1 5																					
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	使用範囲 00 ~ 99																				
D	使用範囲 A																				
演算条件	入力信号の立上り(OFF ON)																				
演算内容 後	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td style="padding: 2px;">D</td><td style="padding: 2px;">n</td></tr> <tr><td style="padding: 2px;">フラグ</td><td style="padding: 2px;">不変</td></tr> </table>	D	n	フラグ	不変																
D	n																				
フラグ	不変																				

(類似命令) F-01w、 F-01d、 F-91

F-01w BCD

BCD定数(4桁)の転送

シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">F-01w BCD</td> <td style="padding: 2px;">n</td> <td style="padding: 2px;">D</td> </tr> </table>	F-01w BCD	n	D	<p>[使用例]</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="margin-right: 10px;"> <table border="1" style="border-collapse: collapse;"> <tr><td style="padding: 2px;">004001</td></tr> </table> </div> <div style="margin-right: 10px;"> <table border="1" style="border-collapse: collapse;"> <tr><td style="padding: 2px;">F-01w BCD</td></tr> <tr><td style="padding: 2px;">1984</td></tr> <tr><td style="padding: 2px;">019100</td></tr> </table> </div> </div> <table border="1" style="border-collapse: collapse; margin-left: auto;"> <thead> <tr><th colspan="2" style="text-align: center;">命 令</th></tr> </thead> <tbody> <tr><td style="padding: 2px;">STR</td><td style="padding: 2px;">004001</td></tr> <tr><td style="padding: 2px;">F-01w</td><td style="padding: 2px;">1984</td></tr> <tr><td></td><td style="padding: 2px;">019100</td></tr> </tbody> </table> <p style="font-size: small;">入力条件004001がOFF ONの変化時に、BCD定数1984を、レジスタ019100、019101に転送します。</p> <div style="text-align: center; margin-top: 10px;"> <table border="1" style="border-collapse: collapse; margin: 0 auto;"> <tr><td style="padding: 2px;">019101</td><td style="padding: 2px;">019100</td></tr> <tr><td style="padding: 2px;">0 0 0 1 1 0 0 1</td><td style="padding: 2px;">1 1 0 0 0 0 1 0 0</td></tr> <tr><td style="padding: 2px;">└───┬───┬───┬───┬───┬───┬───┬───┘</td><td style="padding: 2px;">└───┬───┬───┬───┬───┬───┬───┬───┘</td></tr> <tr><td style="padding: 2px;">1 9</td><td style="padding: 2px;">8 4</td></tr> </table> <p style="margin-top: 5px;">転送後</p> </div>	004001	F-01w BCD	1984	019100	命 令		STR	004001	F-01w	1984		019100	019101	019100	0 0 0 1 1 0 0 1	1 1 0 0 0 0 1 0 0	└───┬───┬───┬───┬───┬───┬───┬───┘	└───┬───┬───┬───┬───┬───┬───┬───┘	1 9	8 4
F-01w BCD	n	D																							
004001																									
F-01w BCD																									
1984																									
019100																									
命 令																									
STR	004001																								
F-01w	1984																								
	019100																								
019101	019100																								
0 0 0 1 1 0 0 1	1 1 0 0 0 0 1 0 0																								
└───┬───┬───┬───┬───┬───┬───┬───┘	└───┬───┬───┬───┬───┬───┬───┬───┘																								
1 9	8 4																								
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	使用範囲 0000 ~ 9999																								
D	使用範囲 B ・ Dには必ず偶数アドレスを設定してください。(019003等は禁止)																								
演算条件	入力信号の立上り(OFF ON)																								
演算内容 後	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td style="padding: 2px;">D、D+1</td><td style="padding: 2px;">n</td></tr> <tr><td style="padding: 2px;">フラグ</td><td style="padding: 2px;">不変</td></tr> </table>	D、D+1	n	フラグ	不変																				
D、D+1	n																								
フラグ	不変																								

(類似命令) F-01、 F-01d、 F-91

F-01d BCD	BCD定数(8桁)の転送
--------------	----------------

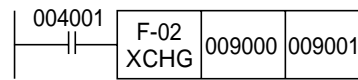
シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">F-01d BCD</td> <td style="padding: 2px 5px;">n</td> <td style="padding: 2px 5px;">D</td> </tr> </table>	F-01d BCD	n	D	<p>[使用例]</p> <table border="1" style="display: inline-table; border-collapse: collapse; margin-right: 10px;"> <tr> <td style="padding: 2px 5px;">004001</td> <td style="padding: 2px 5px;">F-01d BCD</td> <td style="padding: 2px 5px;">19842563</td> <td style="padding: 2px 5px;">029100</td> </tr> </table> <table border="1" style="display: inline-table; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">命 令</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px 5px;">STR</td> <td style="padding: 2px 5px;">004001</td> </tr> <tr> <td style="padding: 2px 5px;">F-01d</td> <td style="padding: 2px 5px;">19842563 029100</td> </tr> </tbody> </table>	004001	F-01d BCD	19842563	029100	命 令		STR	004001	F-01d	19842563 029100											
F-01d BCD	n	D																								
004001	F-01d BCD	19842563	029100																							
命 令																										
STR	004001																									
F-01d	19842563 029100																									
機 能	8桁のBCD定数 nを、レジスタD ~ D+3に転送する。	<p>入力条件004001がOFF ONの変化時に、BCD定数19842563を、レジスタ029100 ~ 029103に転送します。</p> <p>転送後</p> <table style="margin-left: 20px;"> <tr> <td style="text-align: center;">029103</td> <td style="text-align: center;">029102</td> <td style="text-align: center;">029101</td> <td style="text-align: center;">029100</td> </tr> <tr> <td style="text-align: center;">00011001</td> <td style="text-align: center;">10000100</td> <td style="text-align: center;">00100101</td> <td style="text-align: center;">01100011</td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">9</td> <td style="text-align: center;">8</td> <td style="text-align: center;">4</td> </tr> <tr> <td></td> <td></td> <td style="text-align: center;">2</td> <td style="text-align: center;">5</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="text-align: center;">6</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="text-align: center;">3</td> </tr> </table>	029103	029102	029101	029100	00011001	10000100	00100101	01100011	1	9	8	4			2	5				6				3
029103	029102	029101	029100																							
00011001	10000100	00100101	01100011																							
1	9	8	4																							
		2	5																							
			6																							
			3																							
演算内容	n → D ~ D+3																									
n	使用範囲 00000000 ~ 99999999																									
D	使用範囲 C ・ Dには必ず偶数アドレスを設定してください。(019003等は禁止)																									
演算条件	入力信号の立上り(OFF ON)																									
演算内容	D ~ D+3 n																									
演算後	フラグ 不変																									

(類似命令) F-01、F-01w、F-91

F-02 XCHG 1 バイトデータの交換 (eXCHanGe)

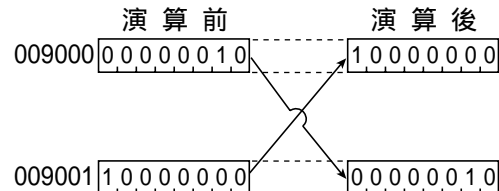
シンボル		
機能	レジスタD1の内容とレジスタD2の内容を交換する。	
演算内容	D1 ↔ D2	
D1	使用範囲 A	
D2	使用範囲 A	
演算条件	入力信号の立上り(OFF ON)	
演算後の内容	D1	レジスタD2の内容
	D2	レジスタD1の内容
	フラグ	不変

[使用例]



命 令	
STR	004001
F-02	009000
	009001

入力条件004001がOFF ONの変化時に、レジスタ009000の内容とレジスタ009001の内容を交換します。

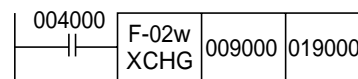


(類似命令) F-02w、F-02d、F-174

F-02w XCHG 1 ワードデータの交換 (eXCHanGe)

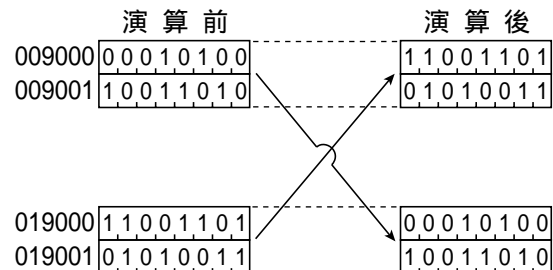
シンボル		
機能	レジスタD1、D1+1の内容(1ワードデータ)と、レジスタD2、D2+1の内容(1ワードデータ)を交換する。	
演算内容	D1、D1+1 ↔ D2、D2+1	
D1	使用範囲 B ・D1には必ず偶数アドレスを設定してください。	
D2	使用範囲 B ・D2には必ず偶数アドレスを設定してください。	
演算条件	入力信号の立上り(OFF ON)	
演算後の内容	D1	レジスタD2の内容
	D1+1	レジスタD2+1の内容
	D2	レジスタD1の内容
	D2+1	レジスタD1+1の内容
	フラグ	不変

[使用例]



命 令	
STR	004000
F-02w	009000
	019000

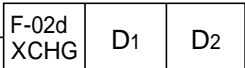
入力条件004000がOFF ONの変化時に、レジスタ009000、009001の内容(1ワードデータ)と、レジスタ019000、019001の内容(1ワードデータ)を交換します。



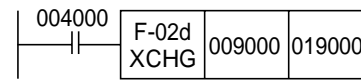
(類似命令) F-02、F-02d、F-174

**F-02d
XCHG**

**2ワードデータの交換
(eXCHanGe)**

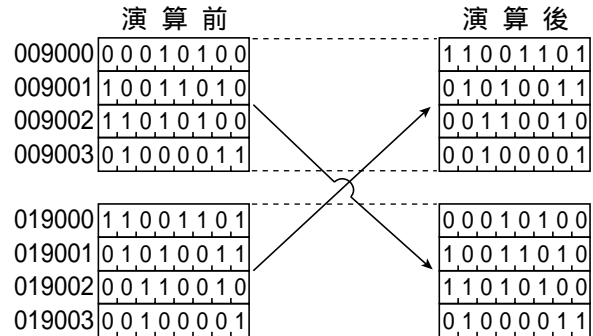
シンボル		
機能	レジスタD1~D1+3の内容(2ワードデータ)と、レジスタD2~D2+3の内容(2ワードデータ)を交換する。	
演算内容	D1~D1+3 ↔ D2~D2+3	
D1	使用範囲C ・D1には必ず偶数アドレスを設定してください。	
D2	使用範囲C ・D2には必ず偶数アドレスを設定してください。	
演算条件	入力信号の立上り(OFF ON)	
演算後の内容	D1~D1+3	レジスタD2~D2+3の内容
	D2~D2+3	レジスタD1~D1+3の内容
	フラグ	不変

[使用例]



命 令	
STR	004000
F-02d	009000
	019000

入力条件004000がOFF ONの変化時に、レジスタ009000~009003の内容(2ワードデータ)と、レジスタ019000~019003の内容(2ワードデータ)を交換します。



(類似命令) F-02、F-02w、F-174

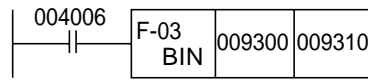
**F-03
BIN**

BCD(2桁) BIN(8ビット)変換

シンボル	— F-03 BIN S D				
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	使用範囲 A				
D	使用範囲 A				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S	不変			
	D	演算結果 ・レジスタSの内容がBCDコード以外 のとき不変			
フラグ	レジスタSの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	BCDコード	0	0	0	0
	BCDコード以外			1	0

(類似命令) F-03、F-03w、F-53、F-153

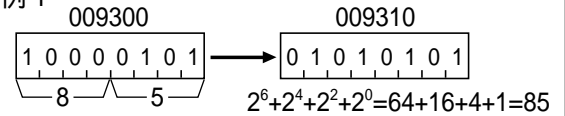
[使用例]



命 令	
STR	004006
F-03	009300 009310

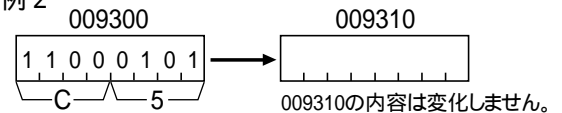
入力条件004006がOFF ONの変化時に、レジスタ009300の8ビットデータをBCDコードと見なし、Binary(2進数)のコードに変換して、レジスタ009310に転送します。
レジスタ009300の内容は不変です。009300の内容がBCDコード以外るとき009310の内容は変化せず、エラーフラグがON(1)します。

例 1



ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
0	0	0	0

例 2



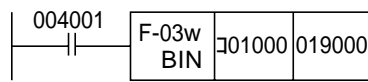
ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
0	0	1	0

**F-03w
BIN**

BCD(4桁) BIN(16ビット)変換

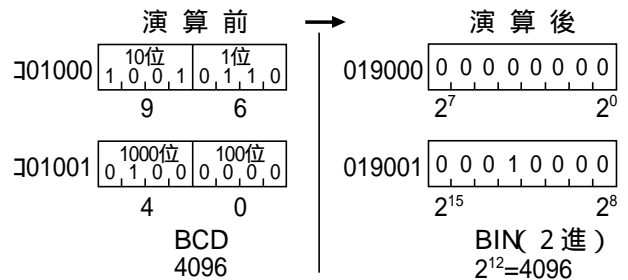
シンボル	— F-03w BIN S D				
機能	レジスタS、S+1(2バイト)のBCD 4桁データを、2進数に変換し、レジスタD、D+1(2バイト)に格納する。				
演算内容	S、S+1 → D、D+1				
S	使用範囲 B				
D	使用範囲 B				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S、S+1	不変			
	D	演算結果 (0 ~ 255)	・レジスタS、S+1の内容 がBCDコード以外の とき不変		
D+1	演算結果 (256 ~ 9999)				
フラグ	レジスタS、S+1の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	BCDコード	0	0	0	0
	BCDコード以外			1	0

[使用例]



命 令	
STR	004001
F-03w	010100 019000

入力条件004001がOFF ONの変化時に、レジスタ010100、010101のBCD 4桁データを、2進数に変換し、レジスタ019000、019001(2バイト)に格納します。



(注) F-53でプログラムを作成すると、モニタ時にF-03wで表示します。

(類似命令) F-03、F-03d、F-53、F-153

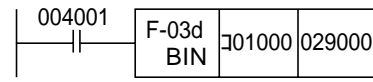
**F-03d
BIN**

BCD(8桁) BIN(32ビット)変換

シンボル	F-03d BIN				S	D
機能	レジスタS ~ S+3(4バイト)のBCD 8桁データを、2進数に変換し、レジスタD ~ D+3(4バイト)に格納する。					
演算内容	S ~ S+3 → D ~ D+3					
S	使用範囲 C					
D	使用範囲 C					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	S ~ S+3	不変				
	D	0 ~ 255		演算結果 ・レジスタS ~ S+3の内容がBCDコード以外るとき不変。		
	D+1	256 ~ 65280				
	D+2	65536 ~ 16711680				
	D+3	16777216 ~ 99999999				
フラグ	レジスタS ~ S+3の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	
	BCDコード	0	0	0	0	
	BCDコード以外	0	0	1	0	

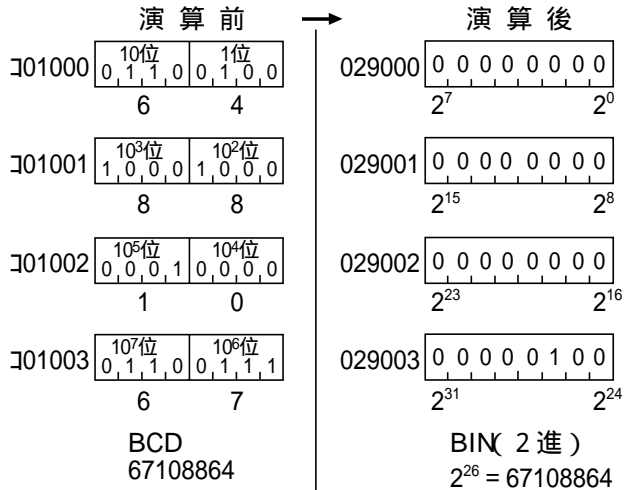
(類似命令) F-03、F-03w、F-53、F-153

[使用例]



命 令	
STR	004001
F-03d	J01000 J029000

入力条件004001がOFF ONの変化時に、レジスタJ01000 ~ J01003のBCD 8桁データを、2進数に変換し、レジスタ029000 ~ 029003(4バイト)に格納します。



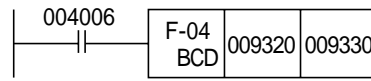
F-04 BCD

BIN(8ビット) BCD(2桁)変換

シンボル	—	F-04 BCD	S	D	
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	使用範囲A				
D	使用範囲A				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S	不変			
	D	演算結果			
	フラグ	不変			

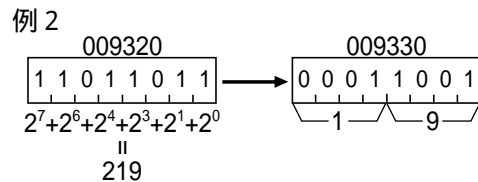
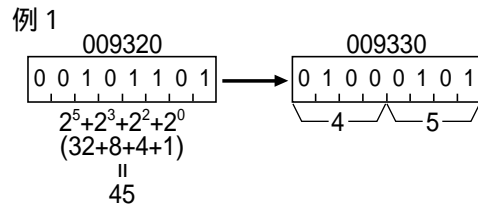
(類似命令) F-04w、F-04d、F-54、F-154

[使用例]



命 令	
STR	004006
F-04	009320
	009330

入力条件004006がOFF ONの変化時に、レジスタ009320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009330に転送します。なお、レジスタ009320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。

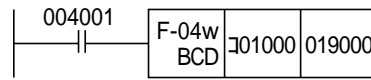


F-04w BCD

BIN(16ビット) BCD(6桁)変換

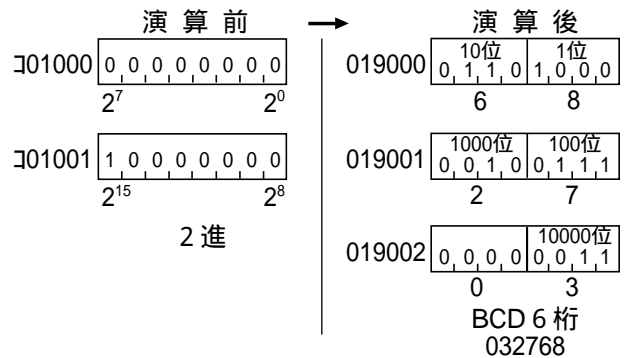
シンボル	—	F-04w BCD	S	D	
機能	レジスタS、S+1の2バイト(2進データ)を、BCD 6桁に変換し、レジスタD、D+1、D+2の3バイトに格納する。				
演算内容	S、S+1 → D、D+1、D+2				
S	使用範囲B				
D	使用範囲E				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S、S+1	不変			
	D	演算結果(1と10の位)			
	D+1	演算結果(100と1000の位)			
	D+2	演算結果(10000の位)			
	フラグ	不変			

[使用例]



命 令	
STR	004001
F-04w	001000
	019000
	019000

入力条件004001がOFF ONの変化時に、レジスタ001000と001001の2バイト(2進データ)をBCD 6桁に変換し、レジスタ019000~019002(3バイト)に格納します。



(注) F-54でプログラムを作成すると、モニタ時にF-04wで表示します。

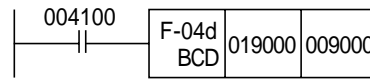
(類似命令) F-04、F-04d、F-54、F-154

F-04d
BCD

BIN(32ビット) BCD(10桁)変換

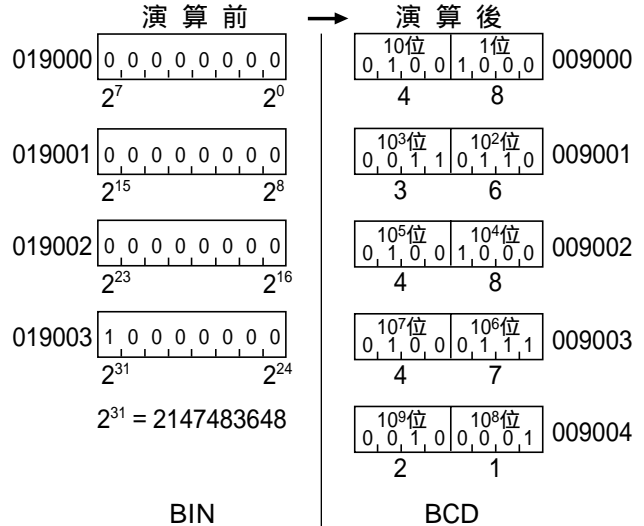
シンボル		
機能	レジスタS~S+3(4バイト : 32ビット) のバイナリデータをBCDコードに変換し、レジスタD~D+4(5バイト)に格納する。	
演算内容	S~S+3 → D~D+4	
S	使用範囲C	
D	使用範囲H	
演算条件	入力信号の立上り(OFF ON)	
演算後の内容	S~S+3	不変
	D	演算結果(1と10の位)
	D+1	演算結果(10 ² と10 ³ の位)
	D+2	演算結果(10 ⁴ と10 ⁵ の位)
	D+3	演算結果(10 ⁶ と10 ⁷ の位)
	D+4	演算結果(10 ⁸ と10 ⁹ の位)
フラグ	不変	

[使用例]



命 令	
STR	004100
F-04d	019000
	009000

入力条件004100がOFF ONの変化時に、レジスタ019000~019003(32ビット)のバイナリデータをBCDコードに変換して、レジスタ009000~009004に格納します。



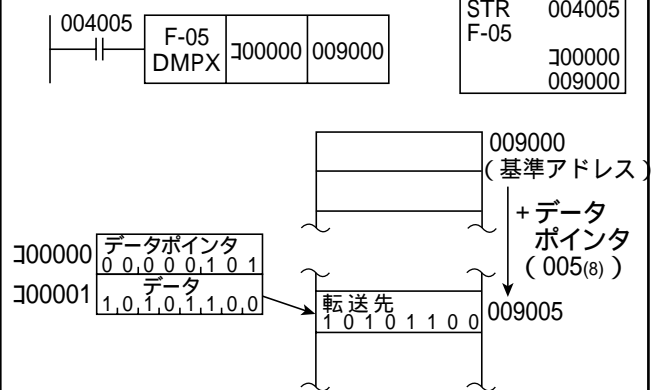
S, Dには必ず偶数アドレスを設定してください。
(類似命令) F-04、F-04w、F-54、F-154

F-05 DMPX 1バイトデータの分配 (DeMultiPleXer)

シンボル	F-05 DMPX S D	
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(データポイント)だけ変位したレジスタに転送する。	
演算内容	$S+1 \rightarrow D+ S$ 	
S	使用範囲 B	
D	使用範囲 F	
演算条件	入力信号の立上り(OFF ON)	
演算後の内容	S, S+1	不変
	D	不変
	D+ S	S+1のレジスタ内容
	フラグ	不変

(類似命令) F-05w, F-05d, F-73, F-73w

[使用例]



入力条件004005がOFF ONの変化時に、000000+1 (000001)のデータを、基準アドレス009000からデータポイント(000000)の内容(005₍₈₎)だけ変位したアドレス(009005)に転送します。
 ・データポイントには000~377₍₈₎を設定できます。よって、本例では基準アドレスが009000で、データポイントの設定値により、アドレス009000~009377にデータを分配できます。

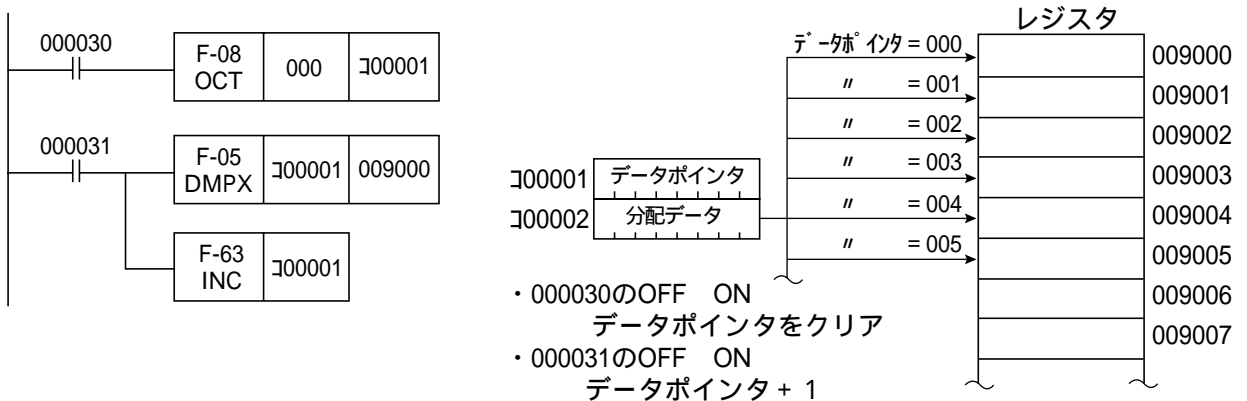
・D(基準アドレス)には、ブロック(データメモリ)の先頭アドレス以外も設定できますが、演算ではそのアドレスを含むブロックの先頭アドレスを基準アドレスとして処理します。

【例】

Dの設定	演算上の基準アドレス
000050	000000
b00210	b00000
009105	009000
00033210	00033000

データメモリのブロックと基準アドレス(9・26ページ)参照

【参考】 データポイントを変化させ、分配先を移動させるプログラム例を示します。



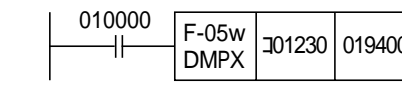
000030をOFF ONすると、000001に000₍₈₎を転送します。(データポイント000₍₈₎)
 000031をOFF ONすると、000002の内容を009000+000₍₈₎ = 009000に転送します。
 000001の内容は、F-63により + 1 され001₍₈₎となります。
 000031を再びOFF ONすると、000002の内容を009000+001₍₈₎ = 009001に転送します。
 000001の内容は、F-63により + 1 され002₍₈₎となります。
 以後、同様に009377までのレジスタに、000002の内容を分配します。

**F-05w
DMPX**

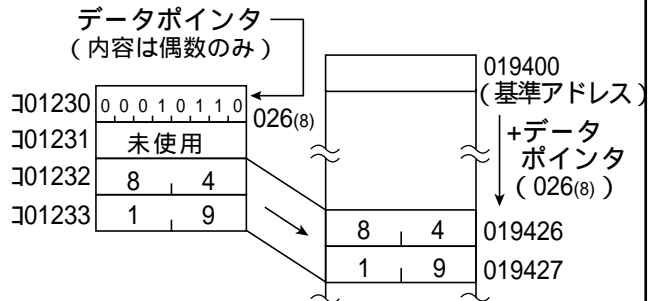
1 ワードデータの分配
(DeMultiPleXer)

シンボル	<table border="1"><tr><td>F-05w</td><td>S</td><td>D</td></tr></table>	F-05w	S	D
F-05w	S	D		
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 (データポインタ)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	$S+2, S+3 \rightarrow D+S, D+S+1$ <table border="1"> <tr><td>データポインタ</td></tr> <tr><td>基準アドレス</td></tr> </table>	データポインタ	基準アドレス	
データポインタ				
基準アドレス				
S	使用範囲 C ・Sには必ず偶数アドレスを設定してください。(00011等は禁止) ・レジスタSに設定する内容は、ワードアドレス(000 ~ 376 ⁽⁸⁾ の偶数)です。			
D	使用範囲 F			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S、S+1、S+2、S+3	不変		
	D	不変		
	D+S	S+2のレジスタ内容		
	D+S+1	S+3のレジスタ内容		
	フラグ	不変		

[使用例]



命 令	
STR	010000
F-05w	01230
	019400



入力条件010000がOFF ONの変化時に、下記の転送を行います。

- ・ 01230+2、01230+3 (01232、01233)のデータを、基準アドレス(019400)からデータポインタ(01230)の内容(026⁽⁸⁾)だけ変位したアドレス(019426)からの2バイトに転送します。

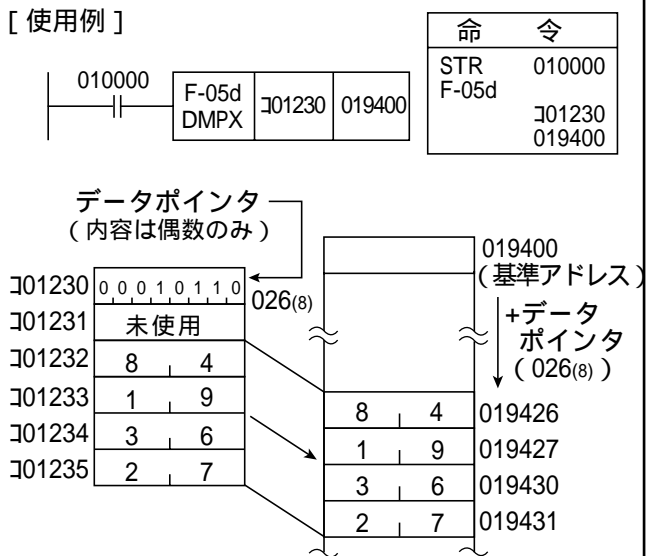
(類似命令) F-05、F-05d、F-72、F-72w、F-72d

- ・ 基準アドレス、データポインタ
データメモリのブロックと基準アドレス(9・26ページ)参照

**F-05d
DMPX**

2ワードデータの分配
(DeMultiPleXer)

シンボル	<table border="1"> <tr> <td>F-05d</td> <td>S</td> <td>D</td> </tr> <tr> <td>DMPX</td> <td></td> <td></td> </tr> </table>		F-05d	S	D	DMPX			[使用例]	<table border="1"> <tr> <th colspan="2">命 令</th> </tr> <tr> <td>STR</td> <td>010000</td> </tr> <tr> <td>F-05d</td> <td></td> </tr> <tr> <td></td> <td>01230</td> </tr> <tr> <td></td> <td>019400</td> </tr> </table>	命 令		STR	010000	F-05d			01230		019400
F-05d	S	D																		
DMPX																				
命 令																				
STR	010000																			
F-05d																				
	01230																			
	019400																			
機能	レジスタS+2 ~ S+5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からの4バイトに転送する。																			
演算内容	$S+2 \sim S+5 \rightarrow D+ S \sim D+ S +3$ 																			
S	使用範囲 H ・ Sには必ず偶数アドレスを設定してください。(00011等は禁止) ・ レジスタSに設定する内容は、ワードアドレス(000 ~ 376 ⁽⁸⁾ の偶数)です。																			
D	使用範囲 F																			
演算条件	入力信号の立上り(OFF ON)																			
演算後の内容	S ~ S+5	不変																		
	D	不変																		
	D+ S	S+2のレジスタ内容																		
	D+ S +1	S+3のレジスタ内容																		
	D+ S +2	S+4のレジスタ内容																		
	D+ S +3	S+5のレジスタ内容																		
フラグ	不変																			



入力条件010000がOFF ONの変化時に、下記の転送を行います。

- ・ 01230+2 ~ 01230+5(01232 ~ 01235)のデータを、基準アドレス(019400)からデータポインタ(01230)の内容(026⁽⁸⁾)だけ変位したアドレス(019426)からの4バイトに転送します。

(類似命令) F-05、F-05w、F-72、F-72w、F-72d

- ・ 基準アドレス、データポインタ
データメモリのブロックと基準アドレス(9・26ページ)参照

F-06 MPX 1バイトデータの抽出
(MultiPleXer)

シンボル	F-06 MPX S D			[使用例]	<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004005</td></tr> <tr><td>F-06</td><td>009000</td></tr> <tr><td></td><td>300000</td></tr> </table>	命 令		STR	004005	F-06	009000		300000
命 令													
STR	004005												
F-06	009000												
	300000												
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	$S + D \rightarrow D+1$ ↳ データポインタ ↳ 基準アドレス												
S	使用範囲 F			入力条件04005がOFF ONの変化時に、基準アドレス009000からデータポインタ(300000)の内容(100 ₍₈₎)だけ変位したアドレス(009100)の内容を、300000+1(300001)に転送します。 ・データポインタには000~377 ₍₈₎ を設定できます。 よって、本例ではデータポインタの設定値により、アドレス009000~009377からデータを抽出できます。									
D	使用範囲 B												
演算条件	入力信号の立上り(OFF ON)												
演算後の内容	S	不変											
	D	不変(データポインタ)											
	D+1	S+ D のレジスタ内容											
	フラグ	不変											

(類似命令) F-06w、F-06d、F-73、F-73w

・S(基準アドレス)には、ブロック(データメモリ)の先頭アドレス以外も設定できますが、演算ではそのアドレスを含む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】

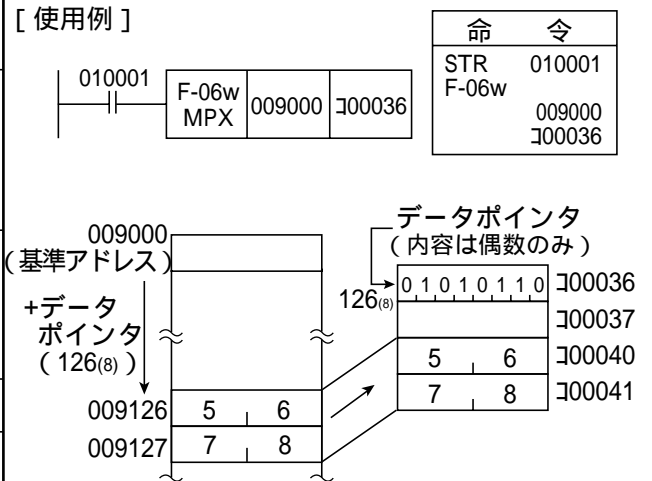
Sの設定	演算上の基準アドレス
300051	300000
b00106	b00000
009023	009000
00031257	00031000

データメモリのブロックと基準アドレス(9・26ページ)参照

**F-06w
MPX**

1 ワードデータの抽出
(MultiPleXer)

シンボル		[使用例]	
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 → D+2、D+3 		
S	使用範囲 F		
D	使用範囲 C ・ Dには必ず偶数アドレスを設定してください。(300011等は禁止) ・ レジスタDに設定する内容は、ワードアドレス(000 ~ 376 ⁽⁸⁾ の偶数)です。		
演算条件	入力信号の立上り(OFF ON)		
演算後の内容	S	不変	
	D、D+1	不変	
	D+2	S+ D のレジスタ内容	
	D+3	S+ D +1のレジスタ内容	
	フラグ	不変	



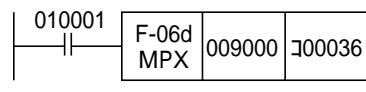
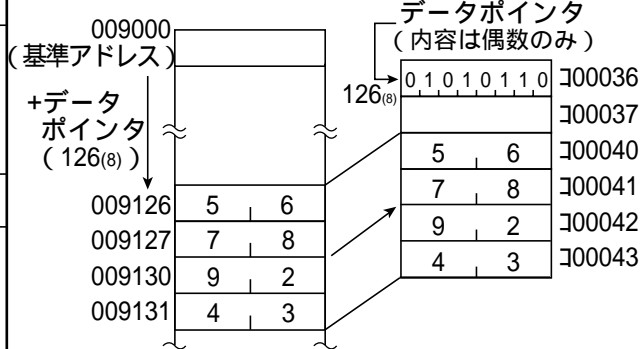
入力条件010001がOFF ONの変化時に、下記の転送を行います。

- ・ 基準アドレス(009000)からデータポインタ (300036)の内容(126⁽⁸⁾)だけ変位したアドレスから2バイト(009126、009127)の内容を、300040 (300036+2)、300041(300036+3)に転送します。

(類似命令) F-06、F-06w、F-73、F-73w、F-73d

- ・ 基準アドレス、データポインタ
データメモリのブロックと基準アドレス(9・26ページ)参照

F-06d MPX 2ワードデータの抽出
(MultiPleXer)

シンボル	<table border="1"><tr><td>F-06d</td><td>S</td><td>D</td></tr></table>	F-06d	S	D	[使用例]	<table border="1"><tr><td>命 令</td><td></td></tr><tr><td>STR</td><td>010001</td></tr><tr><td>F-06d</td><td>009000</td></tr><tr><td></td><td>00036</td></tr></table>	命 令		STR	010001	F-06d	009000		00036
F-06d	S	D												
命 令														
STR	010001													
F-06d	009000													
	00036													
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4バイトの内容を、レジスタD+2~D+5に転送する。													
演算内容	$S + D \sim S + D + 3 \rightarrow D + 2 \sim D + 5$ データポインタ 基準アドレス													
S	使用範囲 F													
D	使用範囲 H ・Dには必ず偶数アドレスを設定してください。(00011等は禁止) ・レジスタDに設定する内容は、ワードアドレス(000~376 ⁽⁸⁾ の偶数)です。													
演算条件	入力信号の立上り(OFF ON)													
演算後の内容	S	不変	入力条件010001がOFF ONの変化時に、下記の転送を行います。 ・基準アドレス(009000)からデータポインタ(00036)の内容(126 ⁽⁸⁾)だけ変位したアドレスから4バイト(009126~009131)の内容を、00040(00036+2)~00043(00036+5)に転送します。											
	D、D+1	不変												
	D+2	S+ D のレジスタ内容												
	D+3	S+ D +1のレジスタ内容												
	D+4	S+ D +2のレジスタ内容												
	D+5	S+ D +3のレジスタ内容												
フラグ	不変													

(類似命令) F-06、F-06w、F-73、F-73w、F-73d

- ・基準アドレス、データポインタ
データメモリのブロックと基準アドレス(9・26^{ページ})参照

F-07 DCML 10進定数(1バイト)の転送
(DeCiMaL)

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-07 DCML</td><td>n</td><td>D</td></tr></table>		F-07 DCML	n	D	<p>[使用例]</p> <table border="1" style="margin-left: 20px;"> <tr><td>004004</td><td>F-07 DCML</td><td>015</td><td>009100</td></tr> </table> <table border="1" style="margin-left: 20px; margin-top: 10px;"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004004</td></tr> <tr><td>F-07</td><td>015</td></tr> <tr><td></td><td>009100</td></tr> </table> <p>入力条件004004がOFF ONの変化時に、レジスタ009100に10進定数15を転送します。 レジスタ009100は転送後、バイナリコードで次の数値になります。</p> <table border="1" style="margin-left: 20px; margin-top: 10px;"> <tr><td>009100</td><td>0 0 0 0 1 1 1 1</td></tr> <tr><td></td><td>$2^3+2^2+2^1+2^0=15$</td></tr> </table>	004004	F-07 DCML	015	009100	命 令		STR	004004	F-07	015		009100	009100	0 0 0 0 1 1 1 1		$2^3+2^2+2^1+2^0=15$
F-07 DCML	n	D																				
004004	F-07 DCML	015	009100																			
命 令																						
STR	004004																					
F-07	015																					
	009100																					
009100	0 0 0 0 1 1 1 1																					
	$2^3+2^2+2^1+2^0=15$																					
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	使用範囲 000 ~ 255																					
D	使用範囲 A																					
演算条件	入力信号の立上り(OFF ON)																					
演算後の内容	D	n (000 ~ 255)																				
演算後の内容	フラグ	不変																				

(類似命令) F-07w、F-07d、F-97

F-07w DCML 10進定数(1ワード)の転送
(DeCiMaL)

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-07w DCML</td><td>n</td><td>D</td></tr></table>		F-07w DCML	n	D	<p>[使用例]</p> <table border="1" style="margin-left: 20px;"> <tr><td>004001</td><td>F-07w DCML</td><td>22659</td><td>019100</td></tr> </table> <table border="1" style="margin-left: 20px; margin-top: 10px;"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004001</td></tr> <tr><td>F-07w</td><td>22659</td></tr> <tr><td></td><td>019100</td></tr> </table> <p>入力条件004001がOFF ONの変化時に、10進定数22659を、レジスタ019100、019101に転送します。</p> <table border="1" style="margin-left: 20px; margin-top: 10px;"> <tr><td>019101</td><td>019100</td></tr> <tr><td>転送後</td><td>0 1 0 1 1 0 0 0 1 0 0 0 0 0 1 1</td></tr> <tr><td></td><td>$2^{14}+2^{13}+2^{11}+2^7+2^1+2^0=22659$</td></tr> </table>	004001	F-07w DCML	22659	019100	命 令		STR	004001	F-07w	22659		019100	019101	019100	転送後	0 1 0 1 1 0 0 0 1 0 0 0 0 0 1 1		$2^{14}+2^{13}+2^{11}+2^7+2^1+2^0=22659$
F-07w DCML	n	D																						
004001	F-07w DCML	22659	019100																					
命 令																								
STR	004001																							
F-07w	22659																							
	019100																							
019101	019100																							
転送後	0 1 0 1 1 0 0 0 1 0 0 0 0 0 1 1																							
	$2^{14}+2^{13}+2^{11}+2^7+2^1+2^0=22659$																							
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	使用範囲 00000 ~ 65535																							
D	使用範囲 B ・Dには必ず偶数アドレスを設定してください。(019003等は禁止)																							
演算条件	入力信号の立上り(OFF ON)																							
演算後の内容	D、D+1	n																						
演算後の内容	フラグ	不変																						

(類似命令) F-07、F-07d、F-97

**F-07d
DCML**

**10進定数(2ワード)の転送
(DeCiMaL)**

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-07d DCML</td> <td style="padding: 2px;">n</td> <td style="padding: 2px;">D</td> </tr> </table>		F-07d DCML	n	D	<p>[使用例]</p> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="margin-right: 10px;"> <p>004001</p> </div> <table border="1" style="border-collapse: collapse;"> <tr> <td style="padding: 2px;">F-07d DCML</td> <td style="padding: 2px;">2563074179</td> <td style="padding: 2px;">019100</td> </tr> </table> </div> <table border="1" style="border-collapse: collapse; margin-bottom: 10px;"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">STR</td> <td style="padding: 2px;">004001</td> </tr> <tr> <td style="padding: 2px;">F-07d</td> <td style="padding: 2px;">2563074179</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">019100</td> </tr> </tbody> </table> <p>入力条件004001がOFF ONの変化時に、10進定数2563074179を、レジスタ019100～019103に転送します。 019100～019103は転送後、バイナリコードで次の数値になります。</p> <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 2px 10px;">019103</td> <td style="padding: 2px 10px;">019102</td> <td style="padding: 2px 10px;">019101</td> <td style="padding: 2px 10px;">019100</td> </tr> <tr> <td style="padding: 2px 10px;">10011000</td> <td style="padding: 2px 10px;">11000101</td> <td style="padding: 2px 10px;">01101000</td> <td style="padding: 2px 10px;">10000011</td> </tr> <tr> <td colspan="4" style="text-align: center; padding: 2px 10px;"> $2^{31}+2^{28}+2^{27}+2^{23}+2^{22}+2^{18}+2^{16}+2^{14}+2^{13}+2^{11}+2^7+2^1+2^0$ =2563074179 </td> </tr> </table>	F-07d DCML	2563074179	019100	命 令		STR	004001	F-07d	2563074179		019100	019103	019102	019101	019100	10011000	11000101	01101000	10000011	$2^{31}+2^{28}+2^{27}+2^{23}+2^{22}+2^{18}+2^{16}+2^{14}+2^{13}+2^{11}+2^7+2^1+2^0$ =2563074179			
F-07d DCML	n	D																											
F-07d DCML	2563074179	019100																											
命 令																													
STR	004001																												
F-07d	2563074179																												
	019100																												
019103	019102	019101	019100																										
10011000	11000101	01101000	10000011																										
$2^{31}+2^{28}+2^{27}+2^{23}+2^{22}+2^{18}+2^{16}+2^{14}+2^{13}+2^{11}+2^7+2^1+2^0$ =2563074179																													
機能	10進定数 n をレジスタ D ~ D+3 に転送する。																												
演算内容	n → D ~ D+3																												
n	使用範囲 0000000000 ~ 4294967295																												
D	使用範囲 C ・ D には必ず偶数アドレスを設定してください。(019003等は禁止)																												
演算条件	入力信号の立上り(OFF ON)																												
演算後の内容	D ~ D+3	n																											
	フラグ	不変																											

(類似命令) F-07、F-07w、F-97

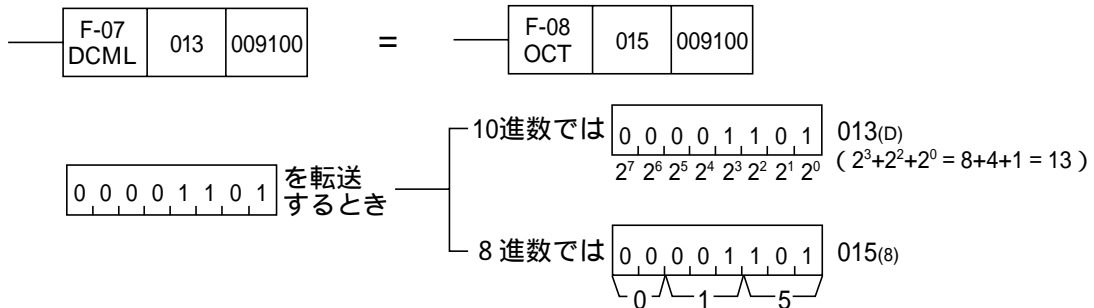
F-08 OCT

8進定数(1バイト)の転送 (OCTal)

シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">F-08 OCT</td> <td style="padding: 2px;">n</td> <td style="padding: 2px;">D</td> </tr> </table>	F-08 OCT	n	D	<p>[使用例]</p> <table border="1" style="display: inline-table; border-collapse: collapse; margin-right: 20px;"> <tr> <td style="padding: 2px;">004004</td> <td style="padding: 2px;">F-08 OCT</td> <td style="padding: 2px;">015</td> <td style="padding: 2px;">009100</td> </tr> </table> <table border="1" style="display: inline-table; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">命 令</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">STR</td> <td style="padding: 2px;">004004</td> </tr> <tr> <td style="padding: 2px;">F-08</td> <td style="padding: 2px;">015</td> </tr> <tr> <td></td> <td style="padding: 2px;">009100</td> </tr> </tbody> </table> <p>入力条件004004がOFF ONの変化時に、レジスタ009100に8進定数015を転送します。レジスタ009100は転送後、次の数値になります。</p> <div style="text-align: center;"> <table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">009100</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">1</td> <td style="padding: 2px;">1</td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">1</td> </tr> <tr> <td></td> <td style="padding: 2px;">0</td> <td style="padding: 2px;">1</td> <td style="padding: 2px;">5</td> <td colspan="5"></td> </tr> </table> </div>	004004	F-08 OCT	015	009100	命 令		STR	004004	F-08	015		009100	009100	0	0	0	0	1	1	0	1		0	1	5					
F-08 OCT	n	D																																	
004004	F-08 OCT	015	009100																																
命 令																																			
STR	004004																																		
F-08	015																																		
	009100																																		
009100	0	0	0	0	1	1	0	1																											
	0	1	5																																
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	使用範囲 000 ~ 377 ₍₈₎																																		
D	使用範囲 A																																		
演算条件	入力信号の立上り(OFF ON)																																		
演算後の内容	D	n(000 ~ 377 ₍₈₎)																																	
	フラグ	不変																																	

(類似命令) F-08w、F-08d、F-71、F-71w

参考 F-07(10進定数の転送)とF-08(8進定数の転送)は、プログラムで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共にバイナリコードとなります。



F-08は、F-05(分配)、F-06(抽出)等のデータポインタ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)を直感的に把握できます。

F-08w OCT

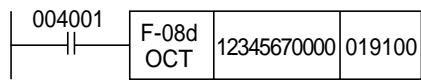
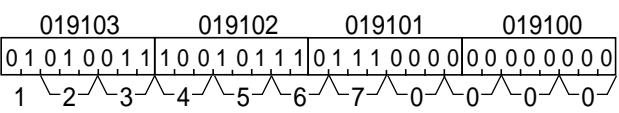
8進定数(1ワード)の転送 (OCTal)

シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">F-08w OCT</td> <td style="padding: 2px;">n</td> <td style="padding: 2px;">D</td> </tr> </table>	F-08w OCT	n	D	<p>[使用例]</p> <table border="1" style="display: inline-table; border-collapse: collapse; margin-right: 20px;"> <tr> <td style="padding: 2px;">004001</td> <td style="padding: 2px;">F-08w OCT</td> <td style="padding: 2px;">123456</td> <td style="padding: 2px;">019100</td> </tr> </table> <table border="1" style="display: inline-table; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">命 令</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">STR</td> <td style="padding: 2px;">004001</td> </tr> <tr> <td style="padding: 2px;">F-08w</td> <td style="padding: 2px;">123456</td> </tr> <tr> <td></td> <td style="padding: 2px;">019100</td> </tr> </tbody> </table> <p>入力条件004001がOFF ONの変化時に、8進定数123456を、レジスタ019100、019101に転送します。</p> <div style="text-align: center;"> <table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">019101</td> <td style="padding: 2px;">019100</td> </tr> <tr> <td style="padding: 2px;">1</td><td style="padding: 2px;">0</td><td style="padding: 2px;">1</td><td style="padding: 2px;">0</td><td style="padding: 2px;">0</td><td style="padding: 2px;">1</td><td style="padding: 2px;">1</td><td style="padding: 2px;">1</td> </tr> <tr> <td style="padding: 2px;">1</td><td style="padding: 2px;">2</td><td style="padding: 2px;">3</td><td style="padding: 2px;">4</td><td style="padding: 2px;">5</td><td style="padding: 2px;">6</td><td colspan="2"></td> </tr> </table> </div>	004001	F-08w OCT	123456	019100	命 令		STR	004001	F-08w	123456		019100	019101	019100	1	0	1	0	0	1	1	1	1	2	3	4	5	6		
F-08w OCT	n	D																																	
004001	F-08w OCT	123456	019100																																
命 令																																			
STR	004001																																		
F-08w	123456																																		
	019100																																		
019101	019100																																		
1	0	1	0	0	1	1	1																												
1	2	3	4	5	6																														
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	使用範囲 000000 ~ 177777 ₍₈₎																																		
D	使用範囲 B ・Dには必ず偶数アドレスを設定してください。(019003等は禁止)																																		
演算条件	入力信号の立上り(OFF ON)																																		
演算後の内容	D、D+1	n																																	
	フラグ	不変																																	

(類似命令) F-08、F-08d、F-71、F-71w

F-08d
OCT

8進定数(2ワード)の転送
(OCTal)

シンボル	$\overline{\text{F-08d OCT}} \quad n \quad D$	[使用例] 	<table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>004001</td> </tr> <tr> <td>F-08d</td> <td>12345670000</td> </tr> <tr> <td></td> <td>019100</td> </tr> </tbody> </table>	命 令		STR	004001	F-08d	12345670000		019100
命 令											
STR	004001										
F-08d	12345670000										
	019100										
機能	8進定数nをレジスタD~D+3に転送する。	<p>入力条件004001がOFF ONの変化時に、8進定数12345670000を、レジスタ019100~019103に転送します。</p> <p>転送後</p> 									
演算内容	$n \rightarrow D \sim D+3$										
n	使用範囲 00000000000 ~ 37777777777 ₍₈₎										
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。(019003等は禁止)										
演算条件	入力信号の立上り(OFF ON)										
演算後の内容	D~D+3	n									
	フラグ	不変									

(類似命令) 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 INV</td><td>S</td><td>D</td></tr></table>			F-09 INV	S	D	[使用例]	<table border="1" style="float: right;"> <thead> <tr><th colspan="2">命 令</th></tr> </thead> <tbody> <tr><td>STR</td><td>004002</td></tr> <tr><td>F-09</td><td>009000</td></tr> <tr><td></td><td>009003</td></tr> </tbody> </table>	命 令		STR	004002	F-09	009000		009003
F-09 INV	S	D														
命 令																
STR	004002															
F-09	009000															
	009003															
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			<p>入力条件004002がOFF ONの変化時に、レジスタ009000の8ビットの内容を反転させ、その内容をレジスタ009003に格納します。</p>												
S	使用範囲 A															
D	使用範囲 A			<p style="text-align: center;"> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>009000</td></tr> <tr><td>0 1 0 1 0 0 1 1</td></tr> </table> → <table border="1" style="display: inline-table;"> <tr><td>009003</td></tr> <tr><td>1 0 1 0 1 1 0 0</td></tr> </table> </p>	009000	0 1 0 1 0 0 1 1	009003	1 0 1 0 1 1 0 0								
009000																
0 1 0 1 0 0 1 1																
009003																
1 0 1 0 1 1 0 0																
演算条件	入力信号の立上り(OFF ON)			<p>なお、レジスタ009000の内容は不変です。</p>												
演算後の内容	S	不変														
	D	レジスタSの内容の反転データ														
	フラグ	不変														

F-09w INV 16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-09w INV</td><td>S</td><td>D</td></tr></table>			F-09w INV	S	D	[使用例]	<table border="1" style="float: right;"> <thead> <tr><th colspan="2">命 令</th></tr> </thead> <tbody> <tr><td>STR</td><td>004000</td></tr> <tr><td>F-09w</td><td>009000</td></tr> <tr><td></td><td>100000</td></tr> </tbody> </table>	命 令		STR	004000	F-09w	009000		100000
F-09w INV	S	D														
命 令																
STR	004000															
F-09w	009000															
	100000															
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	S、S+1 → D、D+1			<p>入力条件004000がOFF ONの変化時に、レジスタ009000、009001の16ビットの内容を反転させ、その内容をレジスタ100000、100001に格納します。</p>												
S	使用範囲 B															
D	使用範囲 B			<p style="text-align: center;"> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>009001</td><td>009000</td></tr> <tr><td>0 1 1 0 1 0 1 0</td><td>1 0 1 0 1 1 0 0</td></tr> </table> ↓ <table border="1" style="display: inline-table;"> <tr><td>100001</td><td>100000</td></tr> <tr><td>1 0 0 1 0 1 0 1</td><td>0 1 0 1 0 0 1 1</td></tr> </table> </p>	009001	009000	0 1 1 0 1 0 1 0	1 0 1 0 1 1 0 0	100001	100000	1 0 0 1 0 1 0 1	0 1 0 1 0 0 1 1				
009001	009000															
0 1 1 0 1 0 1 0	1 0 1 0 1 1 0 0															
100001	100000															
1 0 0 1 0 1 0 1	0 1 0 1 0 0 1 1															
演算条件	入力信号の立上り(OFF ON)			<p>なお、レジスタ009000、009001の内容は不変です。</p>												
演算後の内容	S、S+1	不変														
	D	レジスタSの内容の反転データ														
	D+1	レジスタS+1の内容の反転データ														
フラグ	不変															

S、Dには必ず偶数アドレスを設定してください。(019003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;"> <tr> <td style="text-align: center;">F-09d INV</td> <td style="text-align: center;">S</td> <td style="text-align: center;">D</td> </tr> </table>	F-09d INV	S	D
F-09d INV	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	使用範囲 C			
D	使用範囲 C			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S~S+3	不変		
	D~D+3	レジスタS~S+3の内容の反転データ		
	フラグ	不変		

S、Dには必ず偶数アドレスを設定してください。
(019003等は禁止)

[使用例]

004000	F-09d INV	009000	300000
--------	--------------	--------	--------

命 令	
STR	004000
F-09d	009000 300000

入力条件004000がOFF ONの変化時に、レジスタ009000~009003の32ビットの内容を反転させ、その内容をレジスタ300000~300003に格納します。

009003	009002	009001	009000
11100110	01101010	10101100	00100001
↓			
300003	300002	300001	300000
00011001	10010101	01010011	111011110

なお、レジスタ009000~009003の内容は不変です。

F-10 ADD レジスタ間(BCD 2桁)の加算 (ADD)

シンボル	— F-10 ADD S1 S2 D					
機能	レジスタS1の内容とレジスタS2の内容を加算(BCD 2桁加算)して、レジスタDに格納する。					
演算内容	S1 + S2 → D					
S1	使用範囲 A					
S2	使用範囲 A					
D	使用範囲 A					
演算条件	入力信号の立上り(OFF → ON)					
演算後の内容	S1	不変				
	S2	不変				
	D	演算結果(下位2桁) ・レジスタS1、S2の内容がBCDコード以外 のとき不変。(演算を実行しない)				
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		0	1	0	0	1
1~99		0	0	0	1	
100		1	1	0	0	
101以上	0	1	0	0		
S1、S2の内容がBCDコード以外	0	0	1	0		

[使用例]

004000	F-10 ADD	009000	009010	009020
--------	----------	--------	--------	--------

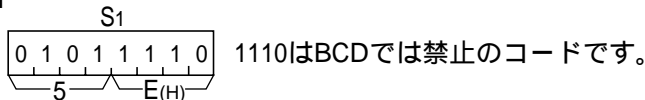
命 令	
STR	004000
F-10	009000
	009010
	009020

入力条件004000がOFF → ONの変化時に、レジスタ009000の内容とレジスタ009010の内容を加算して、レジスタ009020に格納します。
なお、レジスタ009000、009010の内容は不変です。

演算結果とフラグの推移

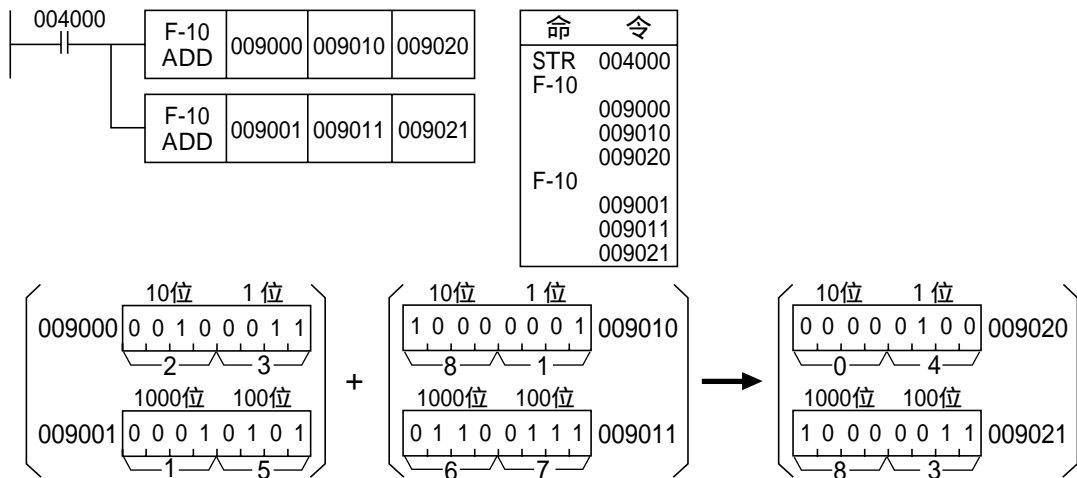
S1、S2の内容がBCDコード以外の数値の場合、エラーフラグ(007355)をONし、加算は行いません。

【例】



【参考】 BCDで3桁以上を加算する場合、F-10命令を続けて設定します。

この場合、2つ目以降のF-10命令では、キャリーフラグ(007356)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(007356)の内容を加算しません。

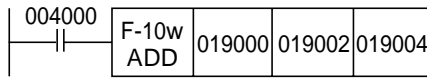


- ・上記の演算は、1523 + 6781 = 8304を示します。
- ・下の桁から順次プログラムすると、桁上げの情報が上位桁に入ります。 倍長演算(9・23ページ)参照

F-10w ADD レジスタ間(BCD 4桁)の加算 (ADD)

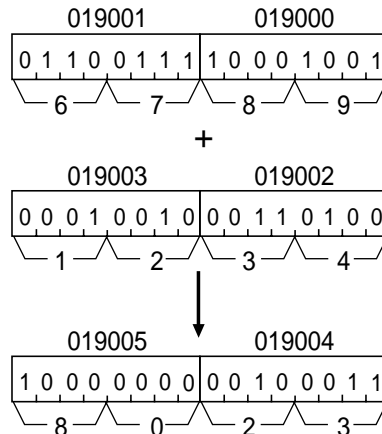
シンボル	F-10w ADD S1 S2 D				
機能	レジスタS1、S1+1の内容と、レジスタS2、S2+1の内容を加算(BCD 4桁加算)して、レジスタD、D+1に格納する。				
演算内容	(S1、S1+1) + (S2、S2+1) → D、D+1				
S1	使用範囲 B				
S2	使用範囲 B				
D	使用範囲 B				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S1、S1+1 不変				
	S2、S2+1 不変				
	D 演算結果(下位2桁)	レジスタS1、S1+1、S2、S2+1の内容がBCDコード以外のとき不変。			
	D+1 演算結果(上位2桁)				
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355
	0	1	0	0	1
	1~9999	0	0	0	1
	10000	1	1	0	0
	10001以上	0	1	0	0
	BCDコード以外	0	0	1	0

[使用例]



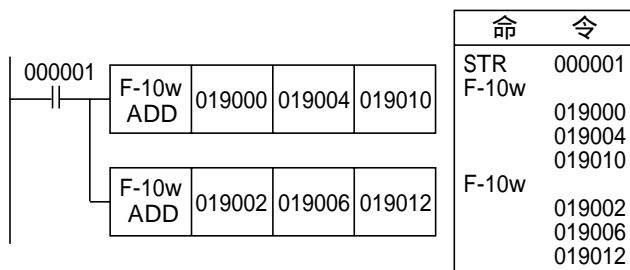
命 令	
STR	004000
F-10w	019000
	019002
	019004

入力条件004000がOFF ONの変化時に、レジスタ019000、019001の内容 BCD 4桁 とレジスタ019002、019003の内容(BCD 4桁)を加算して、レジスタ019004、019005に格納します。

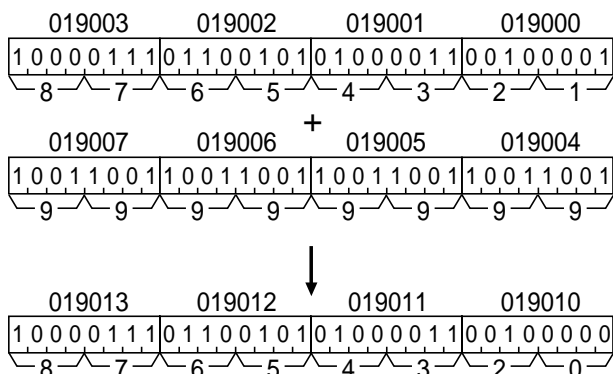


S1、S2、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

参考 F-10w命令もF-10命令と同様に、倍長演算(9・23ビット)が可能です。よって、BCD 8桁以上を加算する場合、F-10w命令を続けて設定します。



命 令	
STR	000001
F-10w	019000
	019004
	019010
F-10w	019002
	019006
	019012

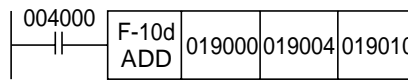


**F-10d
ADD**

**レジスタ間(BCD8桁)の加算
(ADD)**

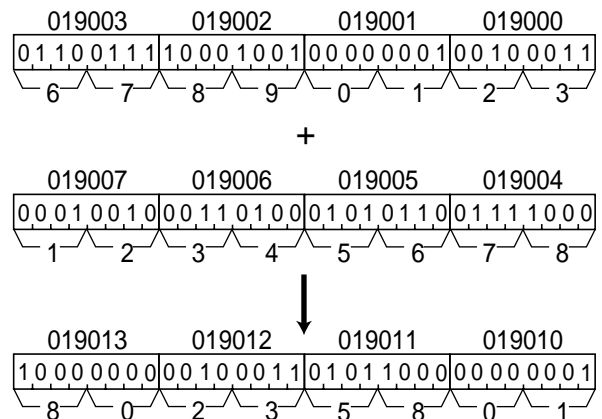
シンボル	$\overline{\text{F-10d ADD}} \text{ S1 S2 D}$					
機能	レジスタS1～S1+3の内容とレジスタS2～S2+3の内容を加算(BCD8桁加算)して、レジスタD～D+3に格納する。					
演算内容	$(S1 \sim S1+3) + (S2 \sim S2+3) \rightarrow D \sim D+3$					
S1	使用範囲C					
S2	使用範囲C					
D	使用範囲C					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	S1～S1+3	不変				
	S2～S2+3	不変				
	D～D+3	演算結果・S1～S1+3、S2～S2+3がBCD(BCD8桁)コード以外するとき不変。				
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		0	1	0	0	1
		1～99999999	0	0	0	1
100000000		1	1	0	0	
100000001以上	0	1	0	0		
BCDコード以外	0	0	1	0		

[使用例]



命 令	
STR	004000
F-10d	019000 019004 019010

入力条件004000がOFF ONの変化時に、レジスタ019000～019003の内容(BCD8桁)とレジスタ019004～019007の内容(BCD8桁)を加算して、レジスタ019010～019013に格納します。



S1、S2、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

- ・ S1～S1+3、S2～S2+3の内容がBCDコード以外
のとき、エラーフラグ(007355)がONし、演算は
実行しません。(D～D+3の内容は不変)

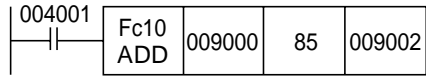
[参考] F-10d命令は、F-10w命令と同様に倍長演算(9・23ページ)が可能です。よって、BCD16桁以上を加算する場合、F-10d命令を続けて設定できます。

**Fc10
ADD**

**レジスタ(BCD 2桁)と定数(2桁)の加算
(ADD)**

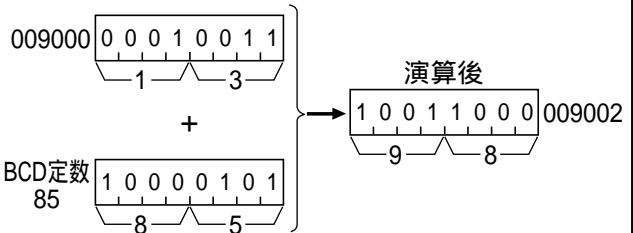
シンボル	$\overline{\text{Fc10}} \text{ ADD } S1 \quad n \quad D$					
機能	レジスタS1の内容と2桁のBCD定数nを加算して、レジスタDに格納する。					
演算内容	$S1 + n \rightarrow D$					
S1	使用範囲 A					
n	使用範囲 00 ~ 99					
D	使用範囲 A					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	S1	不変				
	D	演算結果(下位2桁) ・レジスタS1の内容がBCDコード以外 のとき不変。(演算を実行しない)				
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	パンキャリー 007354
		0	1	0	0	1
		1~99	0	0	0	1
		100	1	1	0	0
101以上		0	1	0	0	
S1の内容がBCDコード以外	0	0	1	0		

[使用例]



命 令	
STR	004001
Fc10	009000
	85
	009002

入力条件004001がOFF ONの変化時に、レジスタ009000の内容とBCD定数85を加算して、レジスタ009002に格納します。
タイミング関係はF-10と同様です。

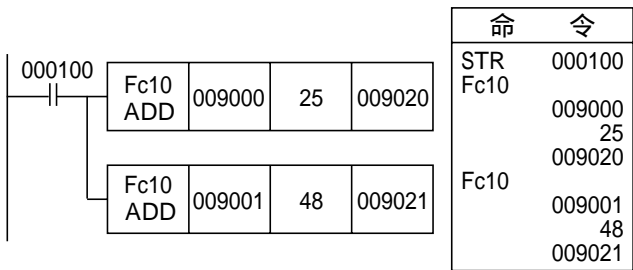


S1の内容がBCDコード以外の数値の場合、エラーフラグ(007355)がONし、加算は行いません。

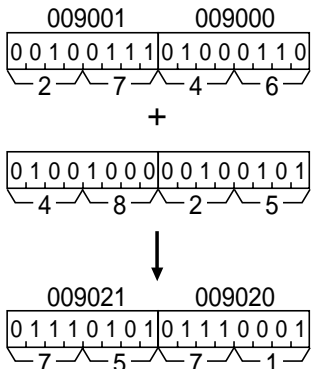
【例】



【参考】 F-10と同様にBCD 3桁以上の加算が可能です。 倍長演算(9・23ページ)参照



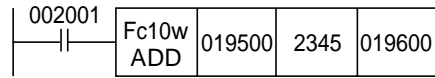
命 令	
STR	000100
Fc10	009000
	25
	009020
Fc10	009001
	48
	009021



Fc10w ADD レジスタ(BCD 4桁)と定数(4桁)の加算

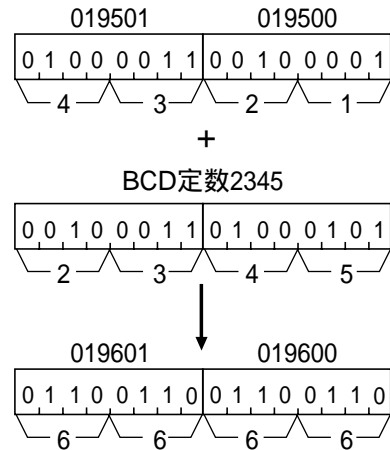
シンボル	Fc10w ADD S1 n D					
機能	レジスタS1、S1+1の内容(BCD 4桁)と、4桁のBCD定数nを加算して、レジスタD、D+1に格納する。					
演算内容	(S1、S1+1) + n → D、D+1					
S1	使用範囲 B					
n	使用範囲 0000 ~ 9999					
D	使用範囲 B					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	S1、S1+1	不変				
	D	演算結果(下位2桁)	・ S1、S1+1の内容がBCDコード以外 のとき不変。 (演算を実行しない)			
	D+1	演算結果(上位2桁)				
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		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	0	0	1	0		

[使用例]



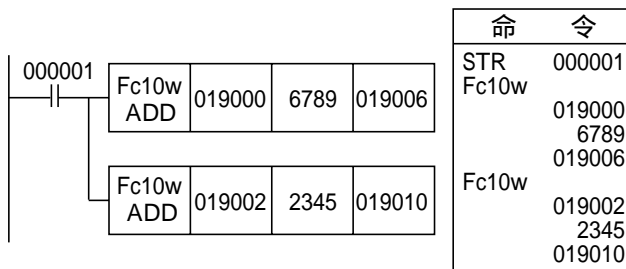
命令	
STR	002001
Fc10w	019500
	2345
	019600

入力条件002001がOFF ONの変化時に、レジスタ019500、019501の内容 BCD 4桁とBCD定数2345を加算して、レジスタ019600、019601に格納します。

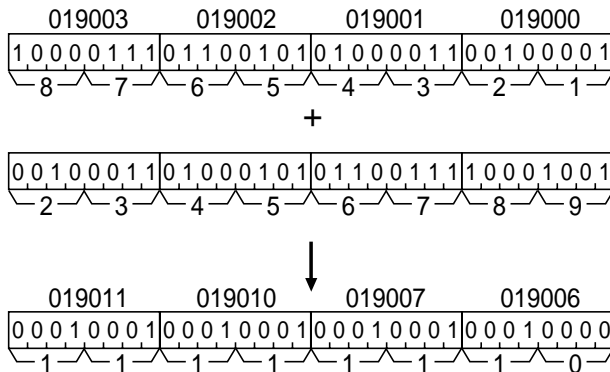


S1、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

参考 Fc10w命令もFc10命令と同様に、倍長演算(9・23ページ)が可能です。よって、BCD 8桁以上を加算する場合、Fc10w命令を続けて設定します。



命令	
STR	000001
Fc10w	019000
	6789
	019006
Fc10w	019002
	2345
	019010

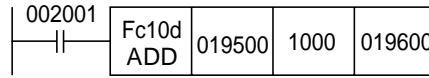


Fc10d ADD

レジスタ(BCD 8桁)と定数(4桁)の加算 (ADD)

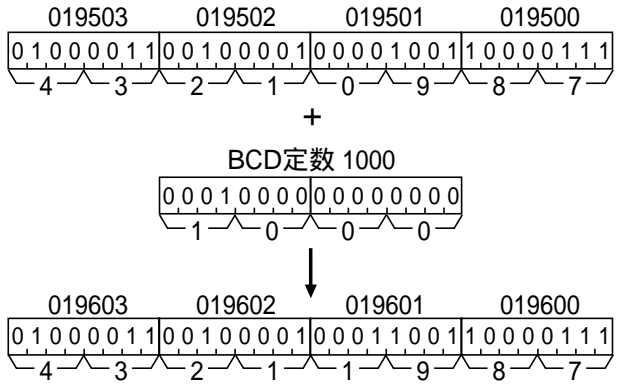
シンボル	Fc10d ADD S1 n D					
機能	レジスタS1 ~ S1+3の内容(BCD 8桁)と、4桁のBCD定数nを加算して、レジスタD ~ D+3に格納する。					
演算内容	(S1 ~ S1+3) + n → D ~ D+3					
S1	使用範囲 C					
n	使用範囲 0000 ~ 9999					
D	使用範囲 C					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	S1 ~ S1+3	不変				
	D ~ D+3	演算結果(BCD 8桁) ・ S1 ~ S1+3の内容がBCDコード以外 のとき不変。(演算を実行しない)				
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		0	1	0	0	1
		1 ~ 99999999	0	0	0	1
100000000		1	1	0	0	
100000001以上	0	1	0	0		
BCD以外	0	0	1	0		

[使用例]



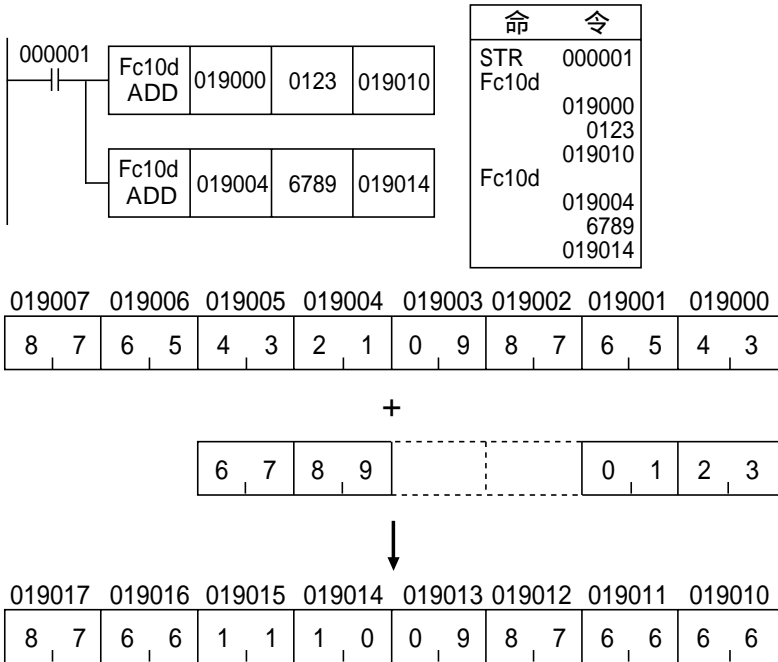
命令	
STR	002001
Fc10d	019500
	1000
	019600

入力条件002001がOFF ONの変化時に、レジスタ019500 ~ 019503の内容(BCD 8桁)とBCD定数1000を加算して、レジスタ019600 ~ 019603に格納します。



S1、Dには必ず偶数アドレスを設定してください。
(019003等は禁止)

参考 Fc10d命令もFc10命令と同様に、倍長演算(9・23ページ)が可能です。
よって、BCD16桁以上を加算する場合、Fc10d命令を続けて設定します。

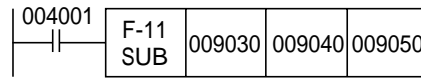


**F-11
SUB**

**レジスタ間(BCD 2桁)の減算
(SUBtract)**

シンボル	— F-11 SUB S1 S2 D					
機能	レジスタS1の内容からレジスタS2の内容を減算(BCD 2桁減算)して、レジスタDに格納する。					
演算内容	S1 - S2 → D					
S1	使用範囲 A					
S2	使用範囲 A					
D	使用範囲 A					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	S1	不変				
	S2	不変				
	D	演算結果 ・レジスタS1、S2の内容がBCDコード以外 のとき不変。(演算を実行しない)				
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		0	1	0	0	1
1~99		0	0	0	1	
負の数値		0	1	0	0	
S1、S2の内容がBCDコード以外	0	0	1	0		

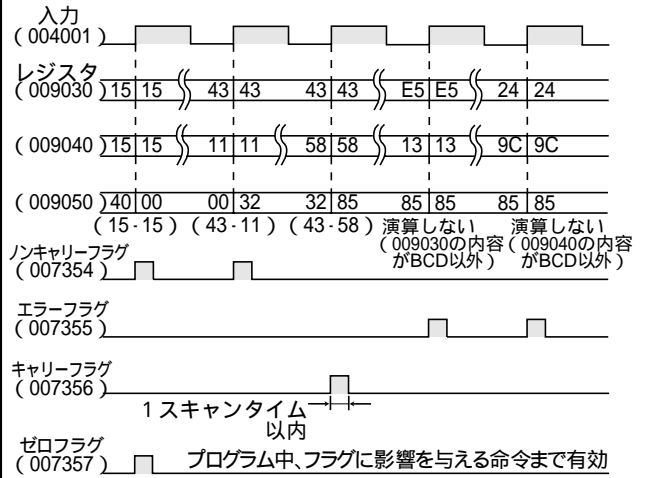
[使用例]



命 令	
STR	004001
F-11	009030
	009040
	009050

入力条件004001がOFF ONの変化時に、レジスタ009030の内容から、レジスタ009040の内容を減算して、レジスタ009050に格納します。
なお、レジスタ009030、009040の内容は不変です。

演算結果とフラグの推移

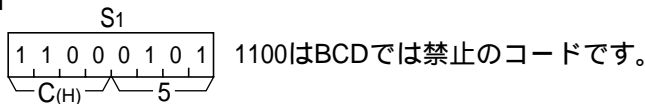


(S1の内容) < (S2の内容) を演算すると、答は100の補数となります。

【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。)

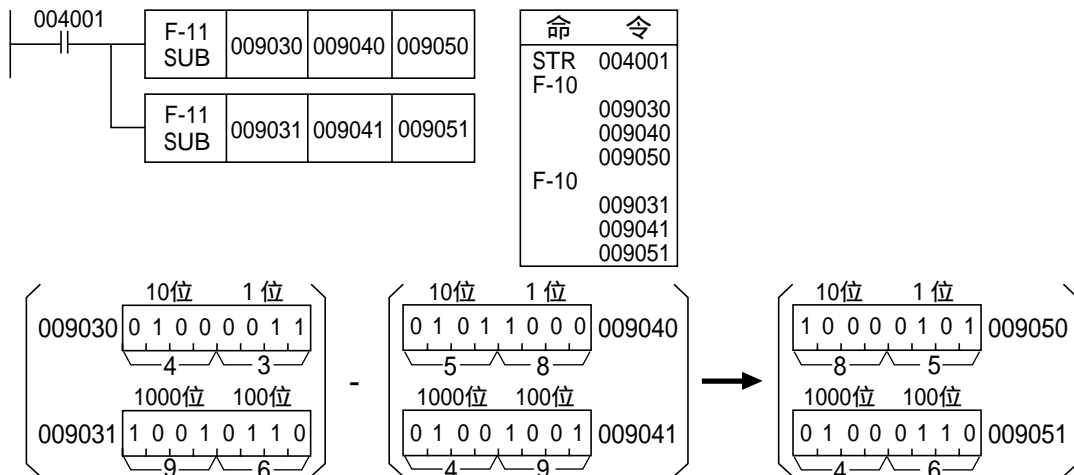
S1、S2の内容がBCDコード以外の数値の場合、エラーフラグ(007355)をONし、減算は行いません。

【例】 (Dの内容は不変です。)



【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。

この場合、2つ目以降のF-11命令では、キャリーフラグ(007356)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(007356)の内容を減算しません。



・上記の演算は、9643 - 4958 = 4685を示します。

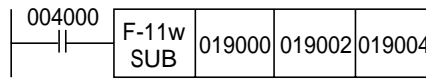
・下の桁から順次プログラムすると、桁下げの情報が上位桁に入ります。倍長演算(9・23ページ)参照

**F-11w
SUB**

**レジスタ間(BCD 4桁)の減算
(SUBtract)**

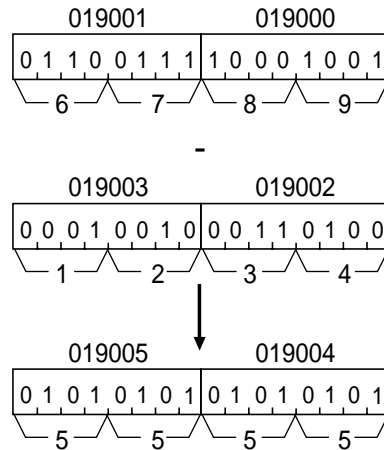
シンボル	F-11w SUB S1 S2 D				
機能	レジスタS1、S1+1の内容からレジスタS2、S2+1の内容を減算(BCD 4桁減算)して、レジスタD、D+1に格納する。				
演算内容	(S1、S1+1) - (S2、S2+1) → D、D+1				
S1	使用範囲 B				
S2	使用範囲 B				
D	使用範囲 B				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S1、S1+1	不変			
	S2、S2+1	不変			
	D	演算結果(下位2桁)			
	D+1	演算結果(上位2桁)			
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355
	0	1	0	0	1
	1~9999	0	0	0	1
	負の数値	0	1	0	0
	BCDコード以外	0	0	1	0

[使用例]



命令	
STR	004000
F-11w	019000
	019002
	019004

入力条件004000がOFF ONの変化時に、レジスタ019000、019001の内容(BCD 4桁)からレジスタ019002、019003の内容(BCD 4桁)を減算して、レジスタ019004、019005に格納します。

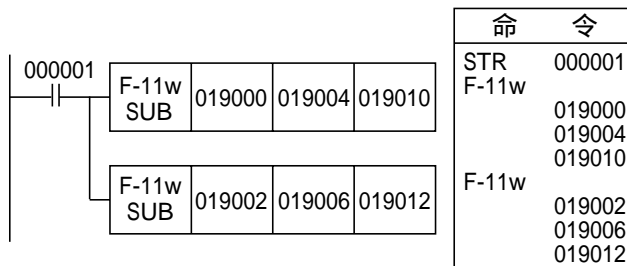


S1、S2、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

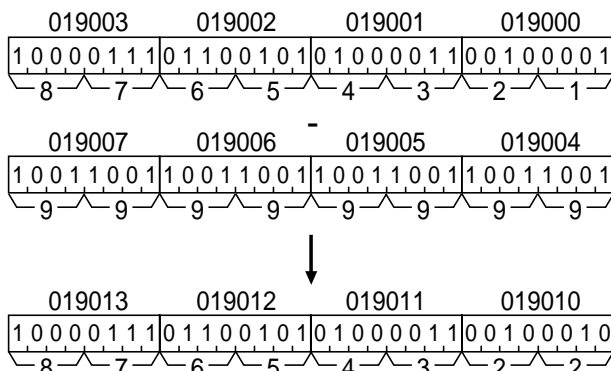
(S1、S1+1の内容) < (S2、S2+1の内容) を演算すると、答えは10000の補数となります。

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

【参考】F-11w命令もF-11命令と同様に、倍長演算(9・23ページ)が可能です。
よって、BCD16桁以上を減算する場合、F-11w命令を続けて設定します。



命令	
STR	000001
F-11w	019000
	019004
	019010
F-11w	019002
	019006
	019012

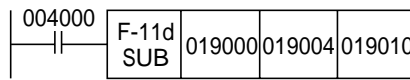


**F-11d
SUB**

**レジスタ間(BCD8桁)の減算
(SUBtract)**

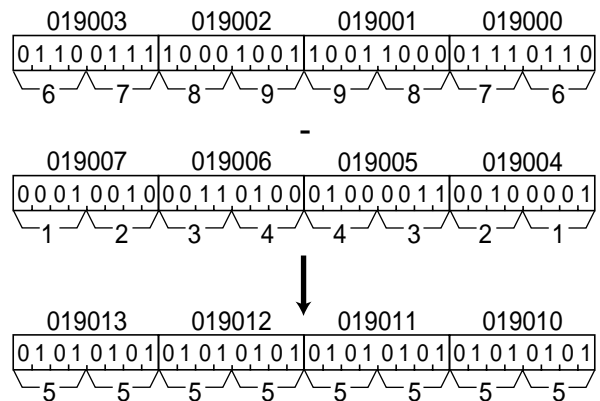
シンボル	$\overline{\text{F-11d SUB}} \text{ S1 S2 D}$	
機能	レジスタS1~S1+3の内容からレジスタS2~S2+3の内容を減算(BCD8桁減算)して、レジスタD~D+3に格納する。	
演算内容	$(S1 \sim S1+3) - (S2 \sim S2+3) \rightarrow D \sim D+3$	
S1	使用範囲C	
S2	使用範囲C	
D	使用範囲C	
演算条件	入力信号の立上り(OFF ON)	
演算後の内容	S1~S1+3	不変
	S2~S2+3	不変
	D~D+3	演算結果(BCD8桁) ・レジスタS1~S1+3、S2~S2+3の内容がBCDコード以外するとき不変。 (演算を実行しない)
	フラグ	演算結果 ゼロ キャリー エラー ノンキャリー 007357 007356 007355 007354
		0 1 0 0 1
	1~99999999 0 0 0 1	
	負の数値 0 1 0 0	
	BCDコード以外 0 0 1 0	

[使用例]



命 令	
STR	004000
F-11d	019000
	019004
	019010

入力条件004000がOFF ONの変化時に、レジスタ019000~019003の内容(BCD8桁)からレジスタ019004~019007の内容(BCD8桁)を減算して、レジスタ019010~019013に格納します。



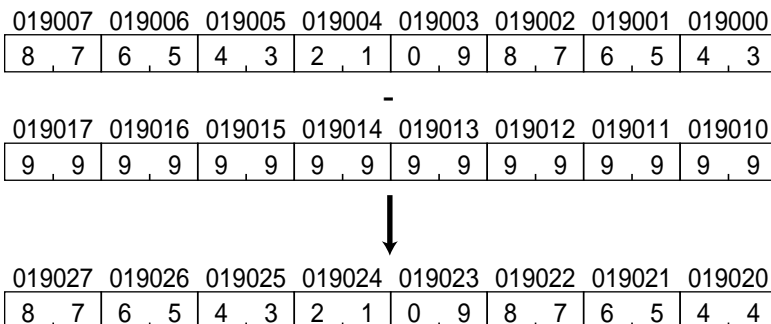
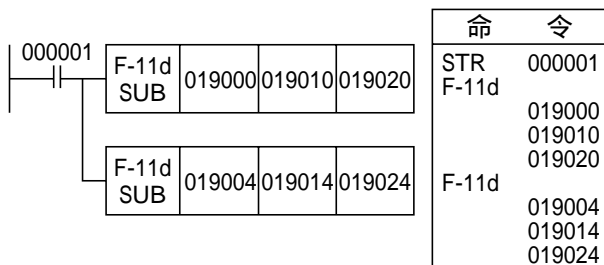
S1, S2, Dには必ず偶数アドレスを設定してください。(019003等は禁止)

(S1~S1+3の内容) < (S2~S2+3の内容) を演算すると、答は100000000の補数となります。

【例】25780000 - 78900000 = -53120000は、53120000の100000000の補数46880000が答となります。
(125780000 - 78900000 = 46880000と考えてください。)

S1~S1+3、S2~S2+3の内容がBCDコード以外するとき、エラーフラグ(007355)がONし、演算は実行しません。
(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(9・23ページ)が可能です。
よって、BCD16桁以上を減算する場合、F-11d命令を続けて設定します。

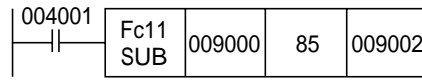


**Fc11
SUB**

**レジスタ(BCD 2桁)と定数(2桁)の減算
(SUBtract)**

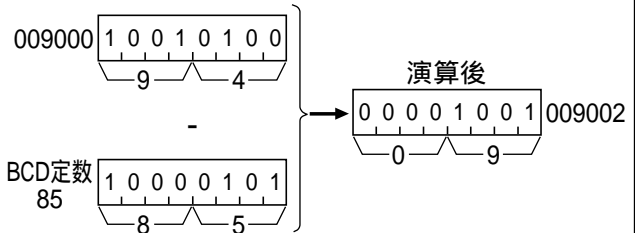
シンボル	$\text{Fc11 SUB } S1 \quad n \quad D$	
機能	レジスタS1の内容から2桁のBCD定数nを減算して、レジスタDに格納する。	
演算内容	$S1 - n \rightarrow D$	
S1	使用範囲 A	
n	使用範囲 00 ~ 99	
D	使用範囲 A	
演算条件	入力信号の立上り(OFF ON)	
演算後の内容	S1	不変
	D	演算結果 ・レジスタS1の内容がBCDコード以外 のとき不変(演算を実行しない)
	フラグ	演算結果
		演算結果 ゼロ キャリー エラー ノンキャリー 007357 007356 007355 007354
		0 1 0 0 1
	1 ~ 99 0 0 0 1	
	負の数値 0 1 0 0	
	S1の内容がBCDコード以外 0 0 1 0	

[使用例]



命 令	
STR	004001
Fc11	009000
	85
	009002

入力条件004001がOFF ONの変化時に、レジスタ009000の内容からBCD定数85を加算して、レジスタ009002に格納します。
タイミング関係はF-11と同様です。

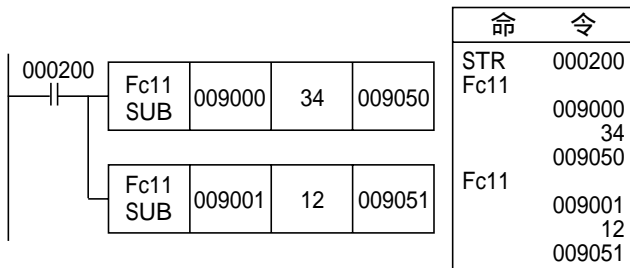


(S1の内容) < nを演算すると、答は100の補数となります。

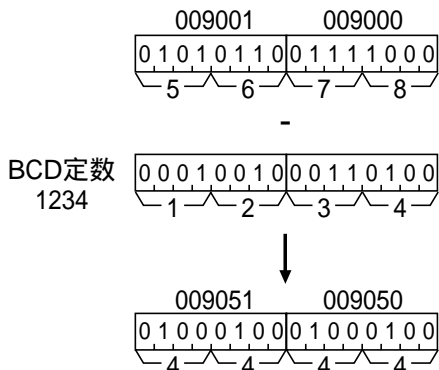
【例】23 - 85 = -62は、62の100の補数38が答となります。(123 - 85 = 38と考えてください。)
S1の内容がBCDコード以外の数値の場合、エラーフラグ(007355)がONし、減算は行いません。

【例】 $\overbrace{1001}^9 \overbrace{1010}^{A_{CH}}$ 1010はBCDでは禁止のコードです。(Dの内容は不変です)

【参考】F-11命令と同様にBCD 3桁以上の減算が可能です。 倍長演算(9・23ページ)参照



命 令	
STR	000200
Fc11	009000
	34
	009050
Fc11	009001
	12
	009051

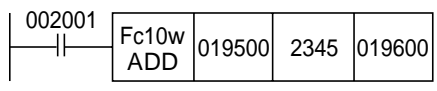


Fc11w SUB レジスタ(BCD 4桁)と定数(4桁)の減算

(SUBtract)

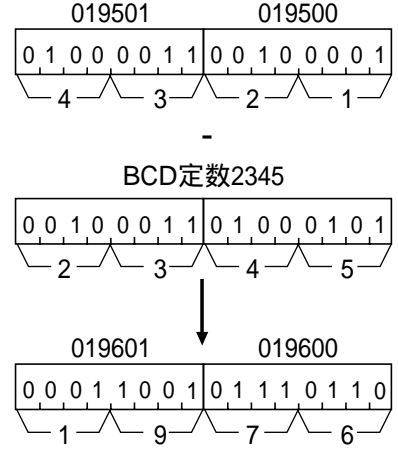
シンボル	Fc11w SUB S1 n D					
機能	レジスタS1, S1+1の内容(BCD 4桁)から4桁のBCD定数nを減算して、レジスタD, D+1に格納する。					
演算内容	(S1, S1+1) - n → D, D+1					
S1	使用範囲 B					
n	使用範囲 0000 ~ 9999					
D	使用範囲 B					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	S1, S1+1	不変				
	D	演算結果(下位2桁)・S1, S1+1の内容がBCDコード以外のとき不変				
	D+1	演算結果(上位2桁)				
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		0	1	0	0	1
1~9999		0	0	0	1	
負の数値		0	1	0	0	
BCD以外	0	0	1	0		

[使用例]



命令	
STR	002001
Fc11w	019500
	2345
	019600

入力条件002001がOFF ONの変化時に、レジスタ019500、019501の内容 BCD 4桁 からBCD定数2345を減算して、レジスタ019600、019601に格納します。

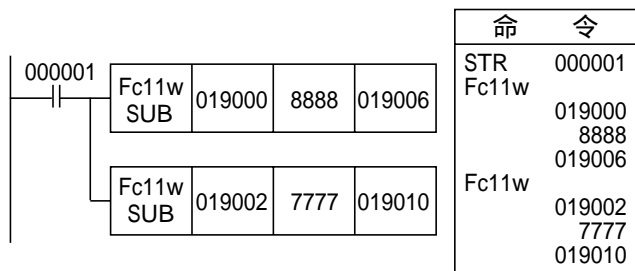


S1, Dには必ず偶数アドレスを設定してください。(019003等は禁止)

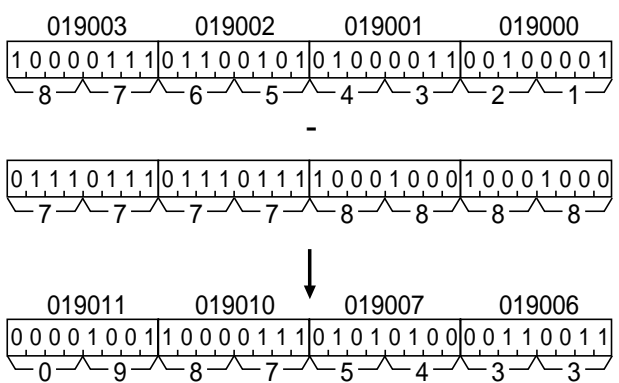
(S1, S1+1の内容) < nを演算すると、答は10000の補数となります。

【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(9・23ページ)が可能です。
よって、BCD 9桁以上を減算する場合、Fc11w命令を続けて設定します。



命令	
STR	000001
Fc11w	019000
	8888
	019006
Fc11w	019002
	7777
	019010

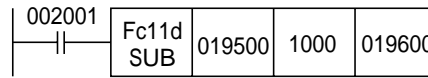


Fc11d SUB

レジスタ(BCD 8桁)と定数(4桁)の減算 (SUBtract)

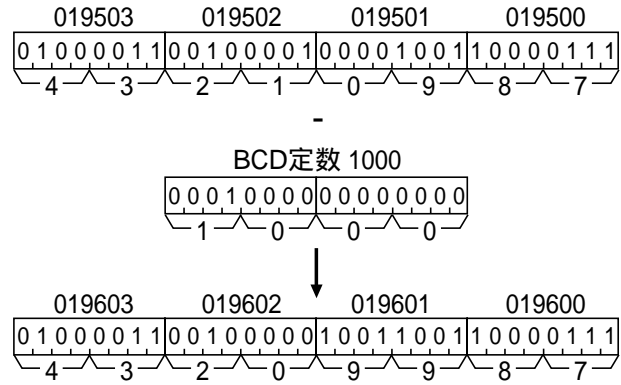
シンボル	Fc11d SUB S1 n D					
機能	レジスタS1~S1+3の内容(BCD 8桁)から4桁のBCD定数nを減算して、レジスタD~D+3に格納する。					
演算内容	(S1~S1+3) - n → D~D+3					
S1	使用範囲 C					
n	使用範囲 0000 ~ 9999					
D	使用範囲 C					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	S1~S1+3	不変				
	D~D+3	演算結果(BCD 8桁) ・ S1~S1+3の内容がBCDコード以外 のとき不変。(演算を実行しない)				
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		0	1	0	0	1
		1~99999999	0	0	0	1
	負の数値	0	1	0	0	
	BCDコード以外	0	0	1	0	

[使用例]



命 令	
STR	002001
Fc11d	019500
	1000
	019600

入力条件002001がOFF ONの変化時に、レジスタ019500~019503の内容(BCD 8桁)からBCD定数1000を減算して、レジスタ019600~019603に格納します。

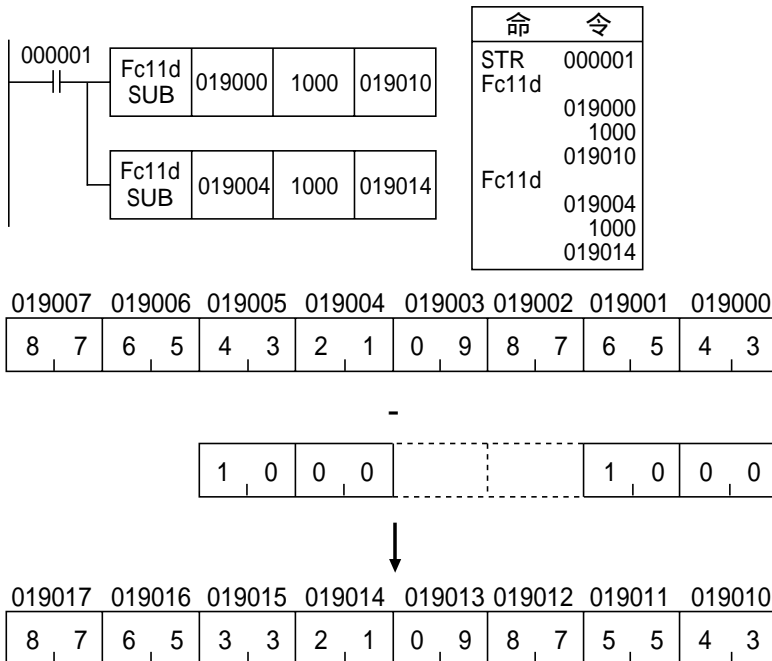


S1, Dには必ず偶数アドレスを設定してください。(019003等は禁止)

(S1~S1+3の内容) < nを演算すると、答は100000000の補数となります。

【例】4568 - 7890 = -3322は、3322の100000000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(9・23ページ)が可能です。
よって、BCD16桁以上を減算する場合、Fc11d命令を続けて設定します。



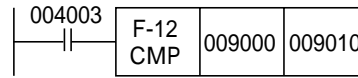
F-12 CMP レジスタ間(1 バイト)の比較 (CoMPare)

シンボル	F-12 CMP S1 S2					
機能	レジスタS1の内容とレジスタS2の内容を大小比較する。					
演算内容	S1 < = > S2 → フラグ					
S1	使用範囲 A					
S2	使用範囲 A					
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)					
演算後の内容	S1	不変				
	S2	不変				
	フラグ	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		S1 > S2	0	0	0	1
S1 = S2		1	0	0	1	
S1 < S2	0	1	0	0		

エラーフラグ(007355)は常に 0 です。

(類似命令) F-12w, F-12d, Fc12, Fc12w, Fc12d, F-112, F-112w, F-112d

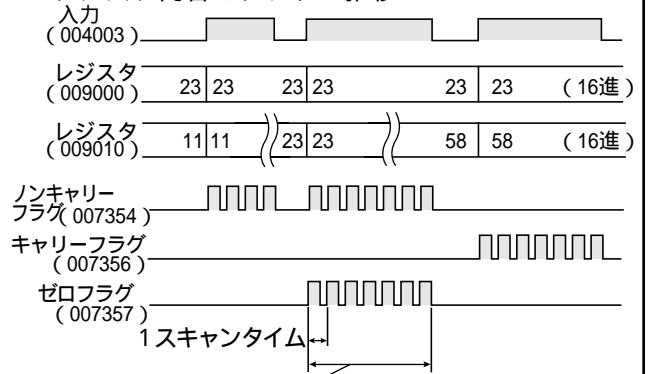
[使用例]



命 令	
STR	004003
F-12	009000 009010

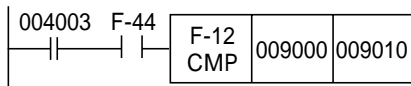
入力条件04003がONのとき、レジスタ009000とレジスタ009010の内容を大小比較して、その結果をノンキャリーフラグ、キャリーフラグ、ゼロフラグに設定します。
このとき、レジスタ009000とレジスタ009010の内容は不変です。

レジスタ内容とフラグの推移



入力条件がONの間、1スキャン毎に比較し、ゼロフラグがONします。

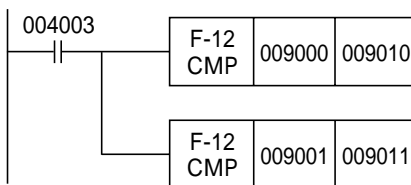
[参考] 入力条件のOFF ONの変化時のみ大小比較する場合は、入力条件に微分命令を組み合わせてください。



命 令	
STR	004003
F-44	F-44
F-12	009000 009010

[参考] 2バイト以上のデータを大小比較する場合は、加算・減算(F-10・F-11)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(007356)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(007356)の内容は比較対象から除外します。



命 令	
STR	004003
F-12	009000 009010
F-12	009001 009011

下の桁から順次プログラムすると、桁下げの情報が上位桁に入ってきます。 倍長演算(9・23ページ)参照

**F-12w
CMP**

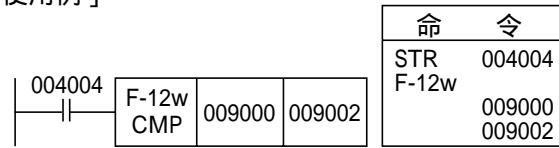
**レジスタ間(1ワード)の比較
(CoMPare)**

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-12w CMP</td><td>S1</td><td>S2</td></tr></table>					F-12w CMP	S1	S2
F-12w CMP	S1	S2						
機能	レジスタS1、S1+1の内容(1ワードデータ)と、レジスタS2、S2+1の内容(1ワードデータ)を大小比較する。							
演算内容	S1、S1+1 < = > S2、S2+1 → フラグ							
S1	使用範囲 B							
S2	使用範囲 B							
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)							
演算後の内容	S1、S1+1	不変						
	S2、S2+1	不変						
	フラグ	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354		
		S1、S1+1 > S2、S2+1	0	0	0	1		
S1、S1+1 = S2、S2+1		1	0	0	1			
	S1、S1+1 < S2、S2+1	0	1	0	0			

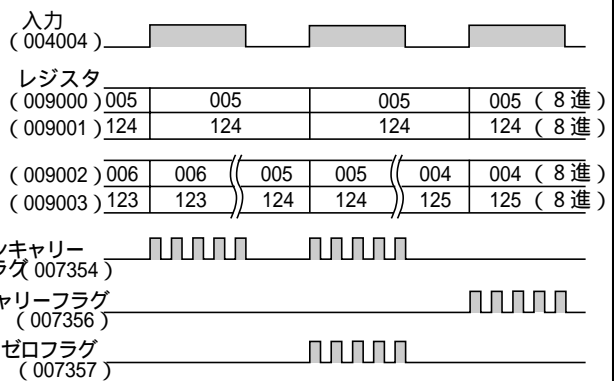
S1、S2には必ず偶数アドレスを設定してください。

(類似命令) F-12、F-12d、Fc12、Fc12w、Fc12d、
F-112、F-112w、F-112d

[使用例]



入力条件004004がONのとき、レジスタ009000、009001の内容(1ワードデータ)とレジスタ009002、009003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、キャリーフラグ、ゼロフラグに設定します。このときレジスタ009000 ~ 009003の内容は不変です。



参考 F-12w命令もF-12命令と同様に、倍長演算(9・23ページ)が可能です。

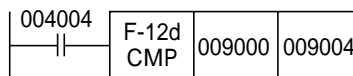
F-12d CMP レジスタ間(2ワード)の比較 (CoMPare)

シンボル	F-12d CMP S1 S2				
機能	レジスタS1~S1+3の内容(2ワードデータ)と、レジスタS2~S2+3の内容(2ワードデータ)を大小比較する。				
演算内容	S1~S1+3 <=> S2~S2+3 → フラグ				
S1	使用範囲 C				
S2	使用範囲 C				
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)				
演算後の内容	S1~S1+3	不変			
	S2~S2+3	不変			
	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	S1~S1+3 > S2~S2+3	0	0	0	1
	S1~S1+3 = S2~S2+3	1	0	0	1
S1~S1+3 < S2~S2+3	0	1	0	0	

S1, S2には必ず偶数アドレスを設定してください。

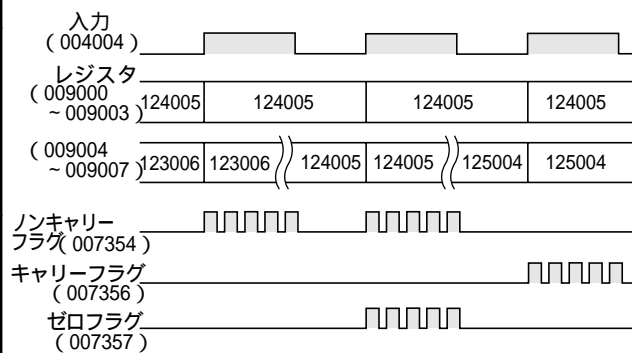
(類似命令) F-12, F-12w, Fc12, Fc12w, Fc12d, F-112, F-112w, F-112d

[使用例]

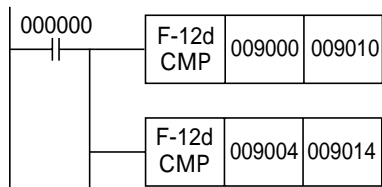


命令	
STR	004004
F-12d	009000
	009004

入力条件004004がONのとき、レジスタ009000~009003の内容(2ワードデータ)とレジスタ009004~009007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、キャリーフラグ、ゼロフラグに設定します。このとき、レジスタ009000~009007の内容は不変です。



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を大小比較できます。倍長演算(9・23ページ)参照



命令	
STR	000000
F-12d	009000
	009010
F-12d	009004
	009014

Fc12 CMP

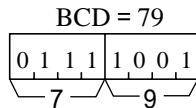
レジスタと8進定数(1バイト)の比較 (CoMPare)

シンボル	— <table border="1" style="display: inline-table;"><tr><td>Fc12 CMP</td><td>S1</td><td>n</td></tr></table>				Fc12 CMP	S1	n
Fc12 CMP	S1	n					
機能	レジスタS1の内容と8進定数nを大小比較する。						
演算内容	S1 < = > n → フラグ						
S1	使用範囲 A						
n	使用範囲 000 ~ 377 ⁽⁸⁾						
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)						
演算後の内容	S1	不変					
	フラグ	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	
		S1 > n	0	0	0	1	
		S1 = n	1	0	0	1	
S1 < n	0	1	0	0			

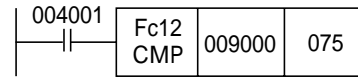
エラーフラグ(007355)は常に0です。

(類似命令) F-12、F-12w、F-12d、Fc12w、Fc12d、F-112、F-112w、F-112d

参考 Fc12はプログラムの書込に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。

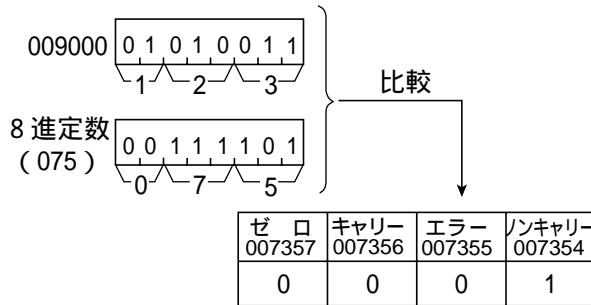


[使用例]



命 令	
STR	004001
Fc12	009000 075

入力条件004001がONのとき、レジスタ009000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、キャリーフラグ、ゼロフラグに設定します。このとき、レジスタ009000の内容は不変です。



Fc12w CMP

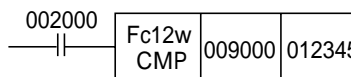
レジスタと8進定数(1ワード)の比較 (COMParE)

シンボル	— <table border="1" style="display: inline-table;"><tr><td>Fc12w CMP</td><td>S1</td><td>n</td></tr></table>				Fc12w CMP	S1	n
Fc12w CMP	S1	n					
機能	レジスタS1、S1+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	S1、S1+1 < = > n → フラグ						
S1	使用範囲 B ・ S1には必ず偶数アドレスを設定してください。(300011等は禁止)						
n	使用範囲 000000 ~ 177777 ⁽⁸⁾						
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)						
演算後の内容	S1、S1+1	不変					
	フラグ	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	
		S1、S1+1 > n	0	0	0	1	
		S1、S1+1 = n	1	0	0	1	
S1、S1+1 < n	0	1	0	0			

(類似命令) F-12、F-12w、F-12d、Fc12、Fc12d、F-112、F-112w

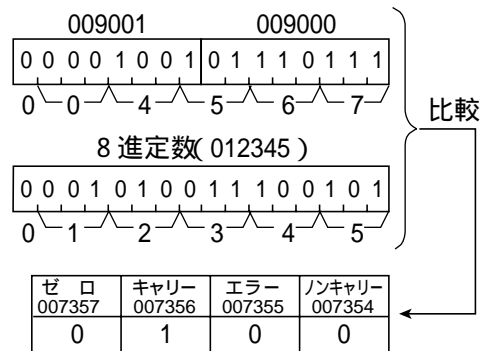
参考 Fc12、Fc12w命令もF-12命令と同様に、倍長演算(9・23ビット)が可能です。

[使用例]



命 令	
STR	002000
Fc12w	009000 012345

入力条件002000がONのとき、レジスタ009000、009001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、キャリーフラグ、ゼロフラグに設定します。このとき、レジスタ009000、009001の内容は不変です。タイミング関係はF-12wと同様です。

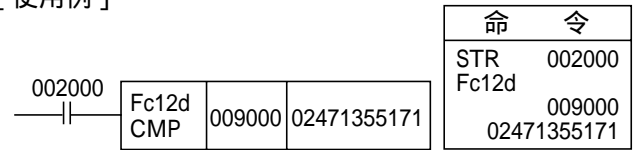


Fc12d CMP レジスタと8進定数(2ワード)の比較 (COMPare)

シンボル	$\overline{\text{Fc12d CMP}} \quad S_1 \quad n$				
機能	レジスタ $S_1 \sim S_{1+3}$ の内容(2ワードデータ)と8進定数 n を大小比較する。				
演算内容	$S_1 \sim S_{1+3} < = > n \rightarrow$ フラグ				
S_1	使用範囲 C ・ S_1 には必ず偶数アドレスを設定してください。(J00011等は禁止)				
n	使用範囲 000000000000 ~ 377777777777 ⁽⁸⁾				
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)				
演算後の内容	$S_1 \sim S_{1+3}$ 不変				
	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	$S_1 \sim S_{1+3} > n$	0	0	0	1
	$S_1 \sim S_{1+3} = n$	1	0	0	1
$S_1 \sim S_{1+3} < n$	0	1	0	0	

(類似命令) F-12、F-12w、F-12d、Fc12、Fc12w、F-112、F-112w

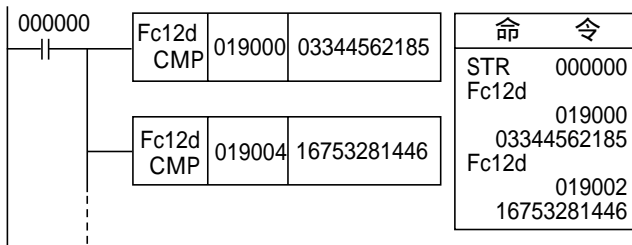
[使用例]



入力条件002000がONのとき、レジスタ009000 ~ 009003の内容(2ワードデータ)と8進定数02471355171を大小比較して、その結果をノンキャリーフラグ、キャリーフラグ、ゼロフラグに設定します。このとき、レジスタ009000 ~ 009003の内容は不変です。タイミング関係はF-12dと同様です。



参考 Fc12d命令を連続して使用すると、8バイト以上のデータを大小比較できます。
倍長演算(9・23ページ)参照



Fx12 CMP

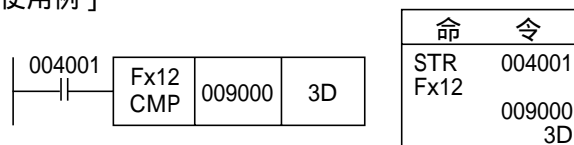
レジスタと16進定数(1バイト)の比較 (CoMPare)

シンボル	— <table border="1" style="display: inline-table;"><tr><td>Fx12 CMP</td><td>S₁</td><td>n</td></tr></table>				Fx12 CMP	S ₁	n
Fx12 CMP	S ₁	n					
機能	レジスタS ₁ の内容と16進定数nを大小比較する。						
演算内容	S ₁ < = > n → フラグ						
S ₁	使用範囲 A						
n	使用範囲 00 ~ FF						
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)						
演算後の内容	S ₁	不変					
	フラグ	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	
		S ₁ > n	0	0	0	1	
		S ₁ = n	1	0	0	1	
S ₁ < n	0	1	0	0			

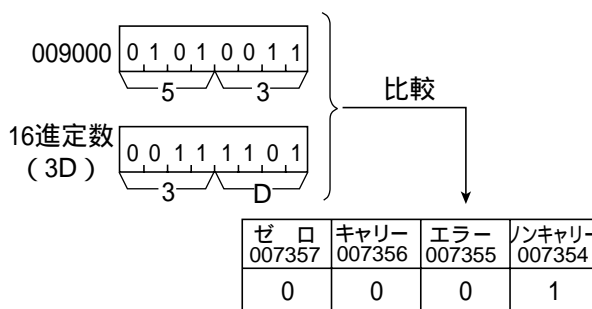
エラーフラグ(007355)は常に 0 です。

(類似命令) F-12、F-12w、F-12d、Fc12w、Fc12d、
Fx12、Fx12w、Fx12d、F-112、F-112w、
F-112d

[使用例]



入力条件004001がONのとき、レジスタ009000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、キャリーフラグ、ゼロフラグに設定します。
このとき、レジスタ009000の内容は不変です。



参考 Fx12命令もF-12命令と同様に、倍長演算(9・23ページ)が可能です。

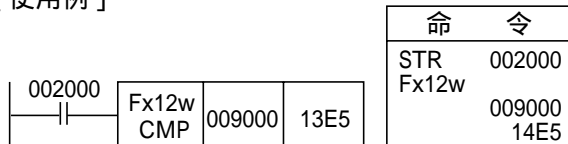
Fx12w CMP

レジスタと16進定数(1ワード)の比較 (CoMPare)

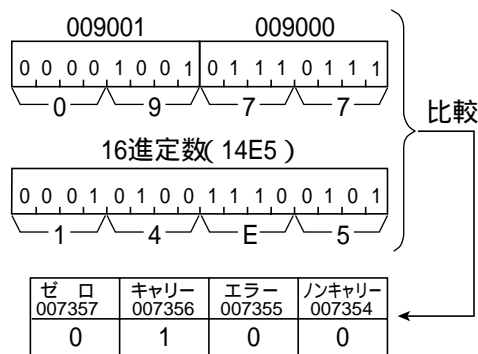
シンボル	— <table border="1" style="display: inline-table;"><tr><td>Fx12w CMP</td><td>S₁</td><td>n</td></tr></table>				Fx12w CMP	S ₁	n
Fx12w CMP	S ₁	n					
機能	レジスタS ₁ 、S ₁ +1の内容(1ワードデータ)と16進定数nを大小比較する。						
演算内容	S ₁ 、S ₁ +1 < = > n → フラグ						
S ₁	使用範囲 B ・ S ₁ には必ず偶数アドレスを設定してください。(300011等は禁止)						
n	使用範囲 0000 ~ FFFF(H)						
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)						
演算後の内容	S ₁ 、S ₁ +1	不変					
	フラグ	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	
		S ₁ 、S ₁ +1 > n	0	0	0	1	
		S ₁ 、S ₁ +1 = n	1	0	0	1	
S ₁ 、S ₁ +1 < n	0	1	0	0			

(類似命令) F-12、F-12w、F-12d、Fc12、Fc12w、
Fc12d、Fx12、Fx12d、F-112、F-112w

[使用例]



入力条件002000がONのとき、レジスタ009000、009001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、キャリーフラグ、ゼロフラグに設定します。このとき、レジスタ009000、009001の内容は不変です。
タイミング関係はF-12wと同様です。



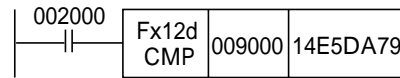
参考 Fx12w命令もF-12命令と同様に、倍長演算(9・23ページ)が可能です。

Fx12d CMP レジスタと16進定数(2ワード)の比較 (COMPare)

シンボル	$\overline{\text{Fx12d CMP}} \quad S_1 \quad n$
機能	レジスタ $S_1 \sim S_{1+3}$ の内容(2ワードデータ)と16進定数 n を大小比較する。
演算内容	$S_1 \sim S_{1+3} < = > n \rightarrow$ フラグ
S_1	使用範囲 C ・ S_1 には必ず偶数アドレスを設定してください。(000011等は禁止)
n	使用範囲 00000000 ~ FFFFFFFF(H)
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)
演算後の内容	$S_1 \sim S_{1+3}$ 不変
	フラグ
	レジスタの内容
	ゼロ 007357 キャリー 007356 エラー 007355 ノンキャリー 007354
	$S_1 \sim S_{1+3} > n$ 0 0 0 1
	$S_1 \sim S_{1+3} = n$ 1 0 0 1
	$S_1 \sim S_{1+3} < n$ 0 1 0 0

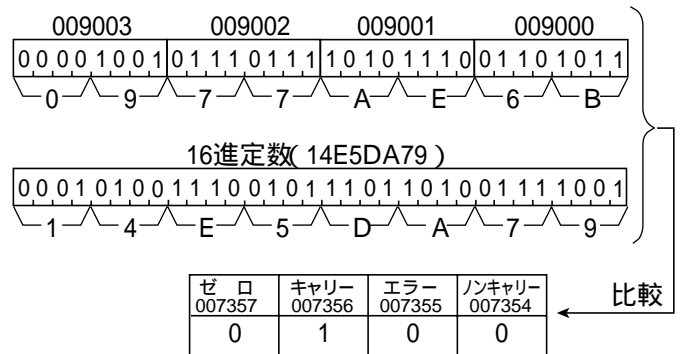
(類似命令) F-12、F-12w、F-12d、Fc12、Fc12w、Fc12d、Fx12、Fx12w、F-112、F-112w

[使用例]

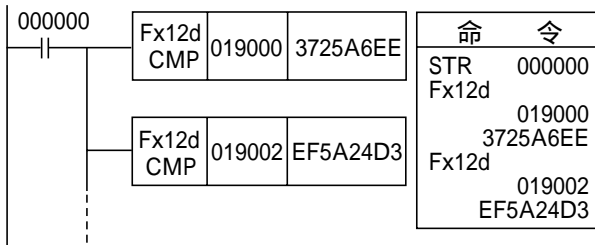


命 令	
STR	002000
Fx12d	009000
	14E5DA79

入力条件002000がONのとき、レジスタ009000～009003の内容(2ワードデータ)と16進定数14E5DA79を大小比較して、その結果をノンキャリーフラグ、キャリーフラグ、ゼロフラグに設定します。このとき、レジスタ009000～009003の内容は不変です。タイミング関係はF-12dと同様です。



参考 Fx12d命令を連続して使用すると、8バイト以上のデータを大小比較できます。倍長演算(9・23ページ)参照



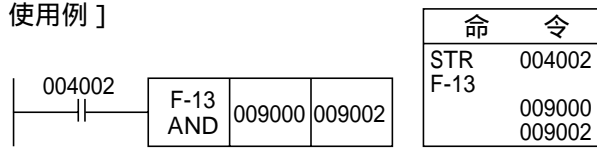
F-13 AND レジスタ間(1 バイト)の論理積 (AND)

シンボル	F-13 AND 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)の論理積をとり、レジスタDに格納する。
演算内容	S D → D
S	使用範囲 A
D	使用範囲 A
演算条件	入力信号の立上り(OFF ON)
演算後の内容	S 不変
	D 演算結果
	フラグ 不変

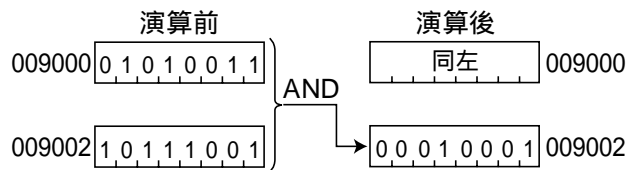
論理積 AND の真理値表

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

[使用例]



入力条件004002がOFF ONの変化時に、レジスタ009000の内容(8ビット)とレジスタ009002の内容(8ビット)の論理積(AND)をとり、レジスタ009002に格納します。
 なお、レジスタ009000の内容は不変です。

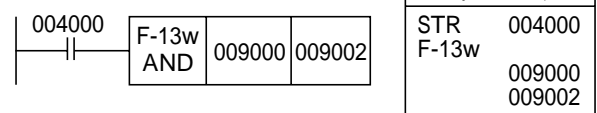


F-13w AND レジスタ間(1 ワード)の論理積 (AND)

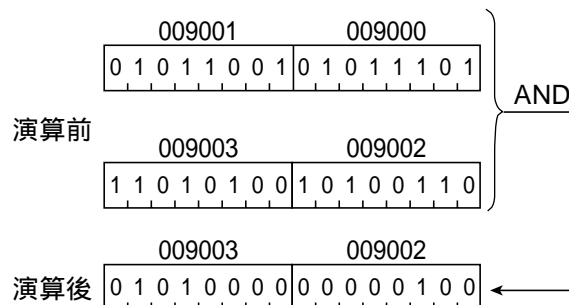
シンボル	F-13w AND S D
機能	レジスタS、S+1の内容(16ビットデータ)と、レジスタD、D+1の内容(16ビットデータ)の論理積をとり、レジスタD、D+1に格納する。
演算内容	S、S+1 D、D+1 → D、D+1
S	使用範囲 B
D	使用範囲 B
演算条件	入力信号の立上り(OFF ON)
演算後の内容	S、S+1 不変
	D、D+1 演算結果
	フラグ 不変

S、Dには必ず偶数アドレスを設定してください。
 (019003等は禁止)

[使用例]



入力条件004000がOFF ONの変化時に、レジスタ009000、009001の内容(16ビットデータ)と、レジスタ009002、009003の内容(16ビットデータ)の論理積をとり、レジスタ009002、009003に格納します。



なお、レジスタ009000、009001の内容は不変です。

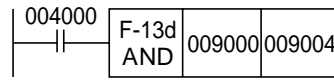
**F-13d
AND**

**レジスタ間(2ワード)の論理積
(AND)**

シンボル	
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	使用範囲 C
D	使用範囲 C
演算条件	入力信号の立上り(OFF ON)
演算後の内容	S ~ S+3 不変
	D ~ D+3 演算結果
	フラグ 不変

S、Dには必ず偶数アドレスを設定してください。
(019003等は禁止)

[使用例]



命 令	
STR	004000
F-13d	009000 009004

入力条件004000がOFF → ONの変化時に、レジスタ009000 ~ 009003の内容(32ビットデータ)と、レジスタ009004 ~ 009007の内容(32ビットデータ)の論理積をとり、レジスタ009004 ~ 009007に格納します。

演算前

009003	009002	009001	009000
01011001	01011101	01011001	01011101

AND

009007	009006	009005	009004
11010100	10100110	11010100	10100110

演算後

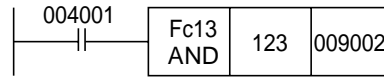
009007	009006	009005	009004
01010000	00000100	01010000	00000100

なお、レジスタ009000 ~ 009003の内容は不変です。

Fc13 AND レジスタと8進定数(1バイト)の論理積 (AND)

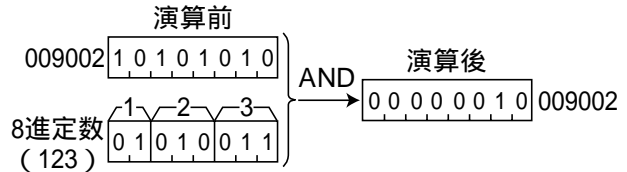
シンボル	$\overline{\text{Fc13 AND}} \quad n \quad D$
機能	8進定数nとレジスタDの内容の論理積をとり、レジスタDに格納する。
演算内容	$n \quad D \rightarrow D$
n	使用範囲 000 ~ 377
D	使用範囲 A
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D 演算結果
	フラグ 不変

[使用例]



命 令	
STR	004001
Fc13	123
	009002

入力条件004001がOFF ONの変化時に、8進定数123とレジスタ009002の内容の論理積をとり、レジスタ009002に格納します。



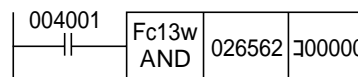
論理積 (AND) の真理値表

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)

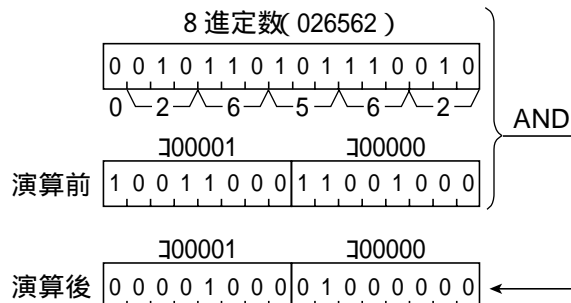
シンボル	$\overline{\text{Fc13w AND}} \quad n \quad D$
機能	8進定数nとレジスタD、D+1の内容(16ビットデータ)の論理積をとり、レジスタD、D+1に格納する。
演算内容	$n \quad D, D+1 \rightarrow D, D+1$
n	使用範囲 000000 ~ 177777(8)
D	使用範囲 B ・Dには必ず偶数アドレスを設定してください。(300011等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D、D+1 演算結果
	フラグ 不変

[使用例]



命 令	
STR	004001
Fc13w	026562
	300000

入力条件004001がOFF ONの変化時に、8進定数026562とレジスタ300000、300001の内容(16ビットデータ)の論理積をとり、レジスタ300000、300001に格納します。

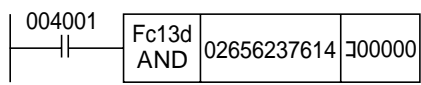


**Fc13d
AND**

**レジスタと8進定数(2ワード)の論理積
(AND)**

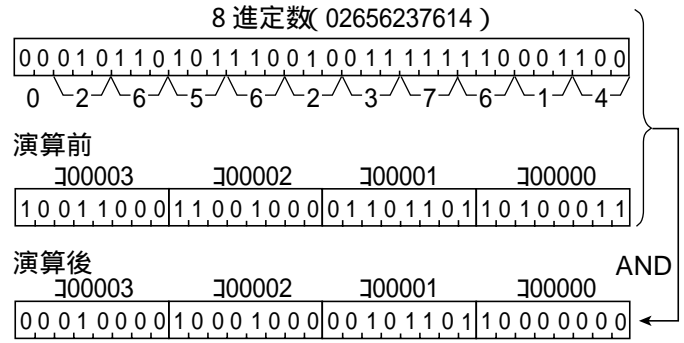
シンボル	$\overline{\text{Fc13d AND}} \quad n \quad D$
機能	8進定数nとレジスタD~D+3の内容(32ビットデータ)の論理積をとり、レジスタD~D+3に格納する。
演算内容	$n \quad D \sim D+3 \rightarrow D \sim D+3$
n	使用範囲 00000000000 ~ 37777777777(8)
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。(300011等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D~D+3 演算結果
	フラグ 不変

[使用例]



命 令	
STR	004001
Fc13d	02656237614 300000

入力条件004001がOFF ONの変化時に、8進定数02656237614とレジスタ300000~300003の内容(32ビットデータ)の論理積をとり、レジスタ300000~300003に格納します。



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 D → D
n	使用範囲 00 ~ FF
D	使用範囲 A
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D 演算結果
	フラグ 不変

[使用例]

命 令	
STR	004001
Fx13	53
	009002

入力条件004001がOFF ONの変化時に、16進定数53とレジスタ009002の内容の論理積をとり、レジスタ009002に格納します。

演算前

009002

16進定数 (53)

AND

演算後 009002

論理積 (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	
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
n	使用範囲 0000 ~ FFFF(H)
D	使用範囲 B ・ D には必ず偶数アドレスを設定してください。(300011等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D、D+1 演算結果
	フラグ 不変

[使用例]

命 令	
STR	004001
Fx13w	2D72
	300000

入力条件004001がOFF ONの変化時に、16進定数2D72とレジスタ300000、300001の内容(16ビットデータ)の論理積をとり、レジスタ300000、300001に格納します。

16進定数(2D72)

演算前

AND

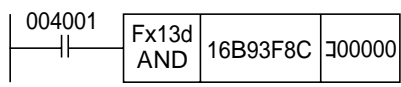
演算後

**Fx13d
AND**

**レジスタと16進定数(2ワード)の論理積
(AND)**

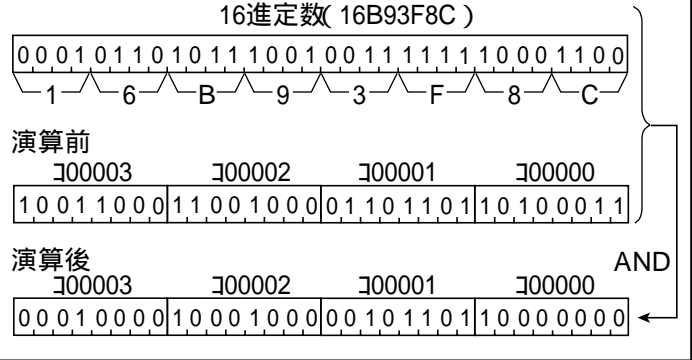
シンボル	$\overline{\text{Fx13d AND}} \quad n \quad D$
機能	16進定数nとレジスタD~D+3の内容(32ビットデータ)の論理積をとり、レジスタD~D+3に格納する。
演算内容	$n \quad D \sim D+3 \rightarrow D \sim D+3$
n	使用範囲 00000000 ~ FFFFFFFF(H)
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。(300011等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D~D+3 演算結果
フラグ	不変

[使用例]



命 令	
STR	004001
Fx13d	16B93F8C
	300000

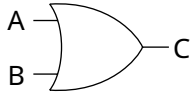
入力条件004001がOFF ONの変化時に、16進定数16B93F8Cとレジスタ300000 ~ 300003の内容(32ビットデータ)の論理積をとり、レジスタ300000 ~ 300003に格納します。



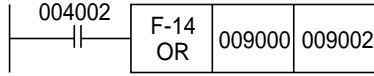
F-14 OR レジスタ間(1 バイト)の論理和 (OR)

シンボル	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	S D → D		
S	使用範囲A		
D	使用範囲A		
演算条件	入力信号の立上り(OFF ON)		
演算後の内容	S	不変	
	D	演算結果	
	フラグ	不変	

論理和(OR)の真理値表

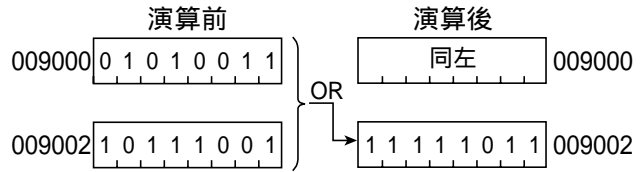
シンボル	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	1

[使用例]



命 令	
STR	004002
F-14	009000 009002

入力条件004002がOFF ONの変化時に、レジスタ009000の内容(8ビット)とレジスタ009002の内容(8ビット)の論理和(OR)をとり、レジスタ009002に格納します。
 なお、レジスタ009000の内容は不変です。

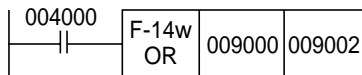


F-14w OR レジスタ間(1 ワード)の論理和 (OR)

シンボル	F-14w OR S D		
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 D、D+1 → D、D+1		
S	使用範囲B		
D	使用範囲B		
演算条件	入力信号の立上り(OFF ON)		
演算後の内容	S、S+1	不変	
	D、D+1	演算結果	
	フラグ	不変	

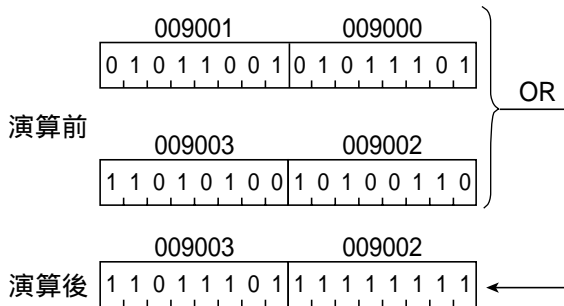
S、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

[使用例]



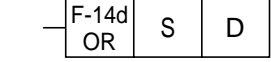
命 令	
STR	004000
F-14w	009000 009002

入力条件004000がOFF ONの変化時に、レジスタ009000、009001の内容(16ビットデータ)と、レジスタ009002、009003の内容(16ビットデータ)の論理和をとり、レジスタ009002、009003に格納します。



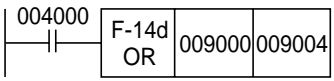
なお、レジスタ009000、009001の内容は不変です。

F-14d
OR レジスタ間(2ワード)の論理和
(OR)

シンボル		
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+3 D ~ D+3 → D ~ D+3	
S	使用範囲 C	
D	使用範囲 C	
演算条件	入力信号の立上り(OFF ON)	
演算後の内容	S ~ S+3	不変
	D ~ D+3	演算結果
	フラグ	不変

S、Dには必ず偶数アドレスを設定してください。
(019003等は禁止)

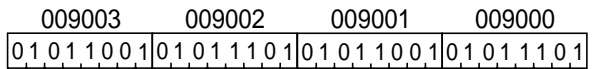
[使用例]



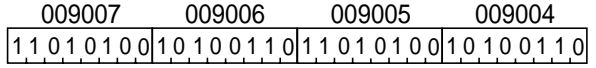
命 令	
STR	004000
F-14d	009000 009004

入力条件004000がOFF ONの変化時に、レジスタ009000 ~ 009003の内容(32ビットデータ)と、レジスタ009004 ~ 009007の内容(32ビットデータ)の論理和をとり、レジスタ009004 ~ 009007に格納します。

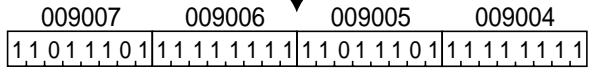
演算前



OR



演算後

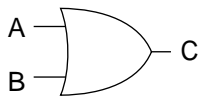


なお、レジスタ009000 ~ 009003の内容は不変です。

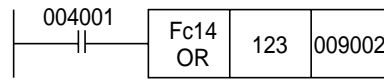
Fc14 OR レジスタと8進定数(1バイト)の論理和 (OR)

シンボル	$\overline{\text{Fc14 OR}} \quad n \quad D$
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。
演算内容	$n \quad D \rightarrow D$
n	使用範囲 000 ~ 377
D	使用範囲 A
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D 演算結果
	フラグ 不変

論理和 (OR) の真理値表

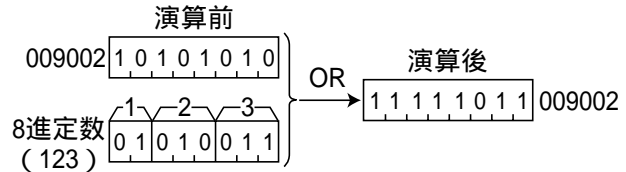
シンボル	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	1

[使用例]



命 令	
STR	004001
Fc14	123
	009002

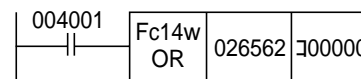
入力条件004001がOFF ONの変化時に、8進定数123とレジスタ009002の内容の論理和 (OR) をとり、レジスタ009002に格納します。



Fc14w OR レジスタと8進定数(1ワード)の論理和 (OR)

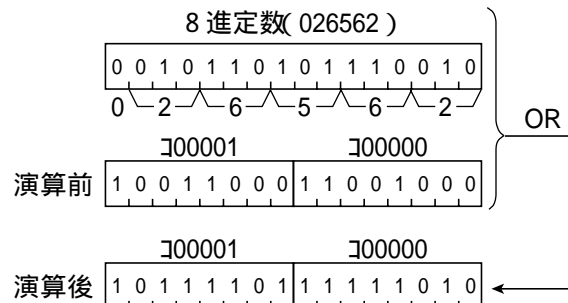
シンボル	$\overline{\text{Fc14w OR}} \quad n \quad D$
機能	8進定数nとレジスタD、D+1の内容(16ビットデータ)の論理和をとり、レジスタD、D+1に格納する。
演算内容	$n \quad D, D+1 \rightarrow D, D+1$
n	使用範囲 000000 ~ 177777(8)
D	使用範囲 B ・Dには必ず偶数アドレスを設定してください。(300011等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D、D+1 演算結果
	フラグ 不変

[使用例]



命 令	
STR	004001
Fc14w	026562
	300000

入力条件004001がOFF ONの変化時に、8進定数026562とレジスタ300000、300001の内容(16ビットデータ)の論理和をとり、レジスタ300000、300001に格納します。

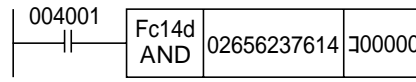


**Fc14d
OR**

**レジスタと8進定数(2ワード)の論理和
(OR)**

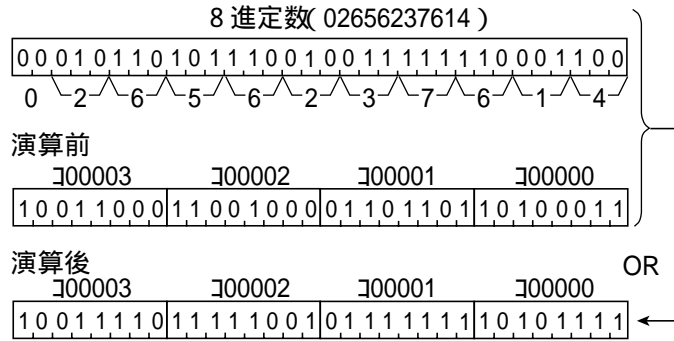
シンボル	$\overline{\text{Fc14d OR}} \quad n \quad D$
機能	8進定数nとレジスタD~D+3の内容(32ビットデータ)の論理和をとり、レジスタD~D+3に格納する。
演算内容	$n \quad D \sim D+3 \rightarrow D \sim D+3$
n	使用範囲 000000000000 ~ 37777777777(8)
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。(300011等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D~D+3 演算結果
	フラグ 不変

[使用例]



命 令	
STR	004001
Fc14d	02656237614
	300000

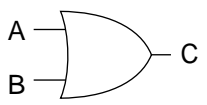
入力条件004001がOFF ONの変化時に、8進定数02656237614とレジスタ300000~300003の内容(32ビットデータ)の論理和をとり、レジスタ300000~300003に格納します。



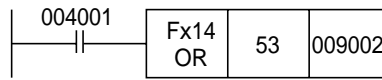
Fx14 OR レジスタと16進定数(1バイト)の論理和 (OR)

シンボル	$\overline{\text{Fx14 OR}} \quad n \quad D$
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	$n \quad D \rightarrow D$
n	使用範囲 00 ~ FF
D	使用範囲 A
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D 演算結果
	フラグ 不変

論理和 (OR) の真理値表

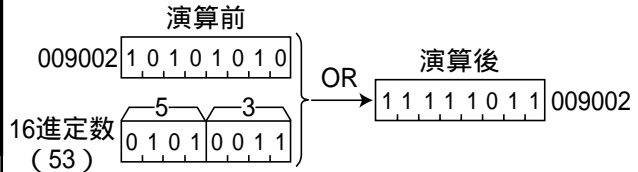
シンボル	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	1

[使用例]



命 令	
STR	004001
Fc14	53
	009002

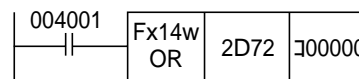
入力条件004001がOFF ONの変化時に、16進定数53とレジスタ009002の内容の論理和 (OR) をとり、レジスタ009002に格納します。



Fx14w OR レジスタと16進定数(1ワード)の論理和 (OR)

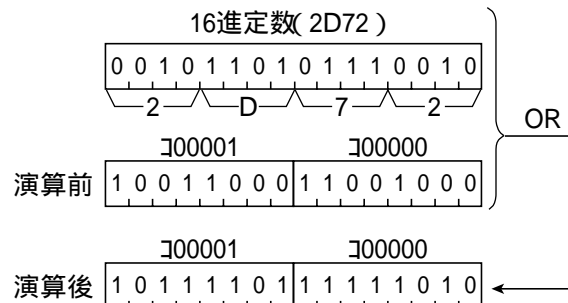
シンボル	$\overline{\text{Fx14w OR}} \quad n \quad D$
機能	16進定数 n とレジスタ D、D+1 の内容(16ビットデータ)の論理和をとり、レジスタ D、D+1 に格納する。
演算内容	$n \quad D, D+1 \rightarrow D, D+1$
n	使用範囲 0000 ~ FFFF(H)
D	使用範囲 B ・ D には必ず偶数アドレスを設定してください。(30001等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D、D+1 演算結果
	フラグ 不変

[使用例]



命 令	
STR	004001
Fx14w	2D72
	300000

入力条件004001がOFF ONの変化時に、16進定数2D72とレジスタ300000、300001の内容(16ビットデータ)の論理和をとり、レジスタ300000、300001に格納します。

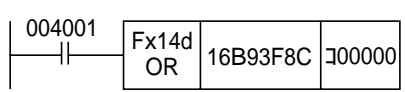


**Fx14d
OR**

**レジスタと16進定数(2ワード)の論理和
(OR)**

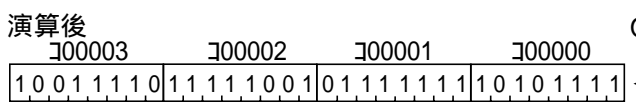
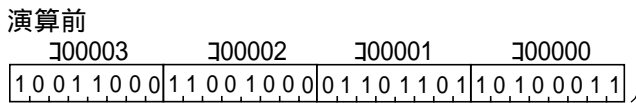
シンボル	$\text{Fx14d OR } n \text{ D}$
機能	16進定数nとレジスタD~D+3の内容(32ビットデータ)の論理和をとり、レジスタD~D+3に格納する。
演算内容	$n \text{ D} \sim \text{D}+3 \rightarrow \text{D} \sim \text{D}+3$
n	使用範囲 00000000 ~ FFFFFFFF(H)
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。(300011等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算内容	D~D+3 演算結果
演算後	フラグ 不変

[使用例]



命 令	
STR	004001
Fx14d	16B93F8C 300000

入力条件004001がOFF ONの変化時に、16進定数16B93F8Cとレジスタ300000 ~ 300003の内容(32ビットデータ)の論理和をとり、レジスタ300000 ~ 300003に格納します。

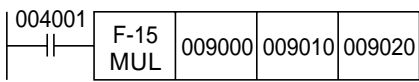


第 11 章 応用命令 (F-15 ~ F-49)

F-15 MUL レジスタ間(BCD 4桁)の乗算 (MULTiPLY)

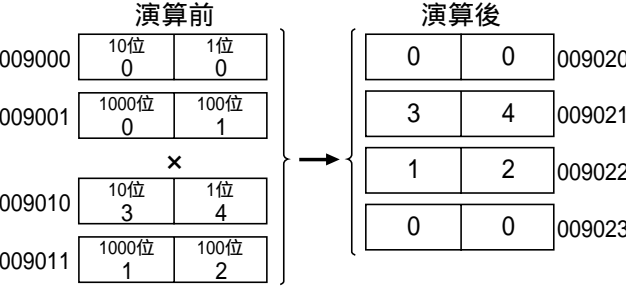
シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">F-15 MUL</td> <td style="padding: 2px;">S1</td> <td style="padding: 2px;">S2</td> <td style="padding: 2px;">D</td> </tr> </table>					F-15 MUL	S1	S2	D
F-15 MUL	S1	S2	D						
機能	レジスタS1、S1+1の内容(BCD 4桁)と、レジスタS2、S2+1内容(BCD 4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	$(S1, S1+1) \times (S2, S2+1) \rightarrow D, D+1, D+2, D+3$								
S1	使用範囲 B								
S2	使用範囲 B								
D	使用範囲 C								
演算条件	入力信号の立上り(OFF ON)								
演算後の内容	S1, S1+1	不変							
	S2, S2+1	不変							
	D	演算結果 (1と10の位)	・レジスタS1、S1+1、S2、S2+1の内容がBCDコード以外の場合、不変。 (演算を実行しない)						
	D+1	演算結果 (100と1000の位)							
	D+2	演算結果 (10000と100000の位)							
D+3	演算結果 (1000000と10000000の位)								
フラグ	レジスタS1、S1+1、S2、S2+1の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354				
	BCDコード	0	0	0	0				
	BCDコード以外	0	0	1	0				

[使用例]



命 令	
STR	004001
F-15	009000
	009010
	009020

入力条件004001がOFF ONの変化時に、レジスタ009000、009001のBCD 4桁と、レジスタ009010、009011のBCD 4桁を乗算し、レジスタ009020からの4バイトに格納します。
 下記の演算は、 $100 \times 1234 = 123400$ を示します。

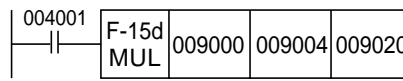


**F-15d
MUL**

**レジスタ間(BCD 8桁)の乗算
(MULTiPLY)**

シンボル	F-15d MUL S1 S2 D																														
機能	レジスタS1～S1+3の内容(BCD 8桁)と、レジスタS2～S2+3の内容(BCD 8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	$(S1 \sim S1+3) \times (S2 \sim S2+3) \rightarrow D \sim D+7$																														
S1	使用範囲 C																														
S2	使用範囲 C																														
D	使用範囲 G																														
演算条件	入力信号の立上り(OFF ON)																														
演算後の内容	S1～S1+3	不変																													
	S2～S2+3	不変																													
演算後の内容	D～D+7	<p>演算結果</p> <table border="1"> <thead> <tr> <th></th> <th>MSB</th> <th>LSB</th> </tr> </thead> <tbody> <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>10¹³</td><td>10¹²</td></tr> <tr><td>D+7</td><td>10¹⁵</td><td>10¹⁴</td></tr> </tbody> </table> <p>・レジスタS1～S1+3、S2～S2+3の内容がBCDコード以外するとき不変。 (演算を実行しない)</p>				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
		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S1～S1+3、S2～S2+3の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354																										
	BCDコード	0	0	0	0																										
	BCDコード以外	0	0	1	0																										

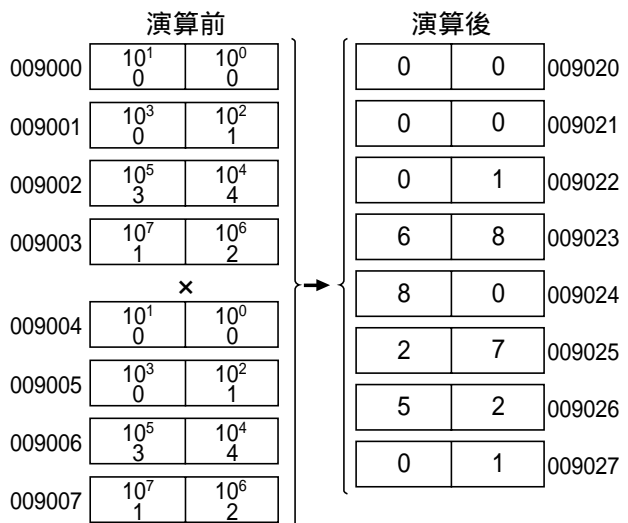
[使用例]



命 令	
STR	004001
F-15d	009000
	009004
	009020

入力条件004001がOFF ONの変化時に、レジスタ009000～009003のBCD 8桁と、レジスタ009004～009007のBCD 8桁を乗算して、レジスタ009020からの8バイトに格納します。

下記の演算は、
 $12340100 \times 12340100 = 152278068010000$
 を示します。



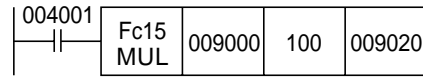
S1、S2、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

**Fc15
MUL**

**レジスタ(BCD 4桁)とBCD定数(3桁)の乗算
(MULTIPLY)**

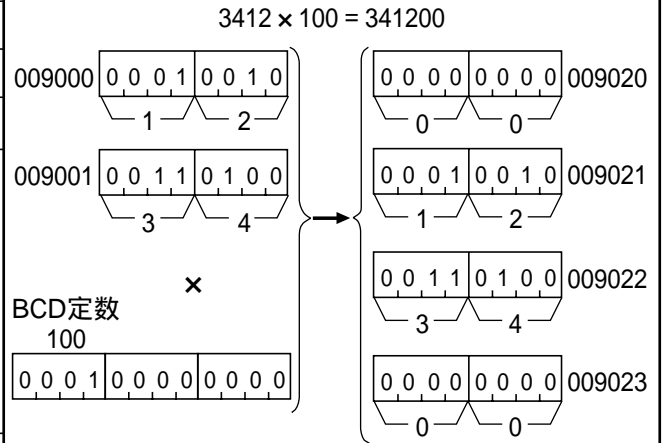
シンボル	— Fc15 MUL S1 n D				
機能	レジスタS1、S1+1の内容(BCD 4桁)と、3桁のBCD定数nを乗算して、レジスタDからの4バイトに格納する。				
演算内容	$(S1, S1+1) \times n \rightarrow D, D+1, D+2, D+3$				
S1	使用範囲 B				
n	使用範囲 000 ~ 999				
D	使用範囲 C				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S1, S1+1	不変			
	D	演算結果 (1と10の位)	・レジスタS1、S1+1の内容がBCDコード以外るとき不変。(演算を実行しない)		
	D+1	演算結果 (100と1000の位)			
	D+2	演算結果 (10000と100000の位)			
	D+3	演算結果 (1000000と10000000の位)			
フラグ	レジスタS1、S1+1の内容	ゼロ 007357	キャリー 007356	エラー 007355	ハシキャリー 007354
	BCDコード	0	0	0	0
	BCDコード以外	0	0	1	0

[使用例]



命 令	
STR	004001
Fc15	009000
	100
	009020

入力条件004001がOFF ONの変化時に、レジスタ009000、009001のBCD 4桁とBCD定数100(3桁)を乗算して、レジスタ009020からの4バイトに格納します。

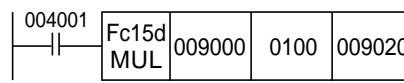


**Fc15d
MUL**

**レジスタ間(BCD 8桁)とBCD定数(4桁)の乗算
(MULTIPLY)**

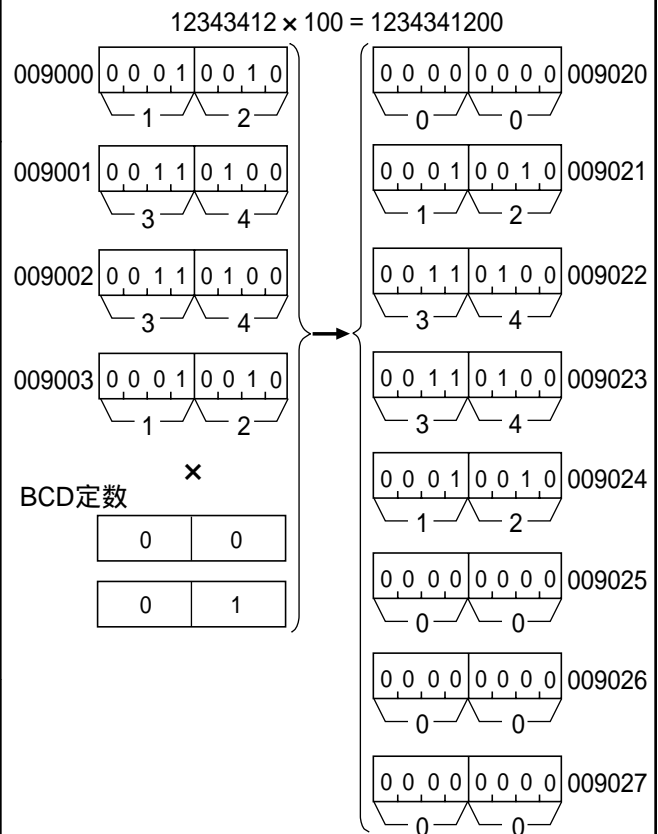
シンボル	Fc15d MUL S1 n D																										
機能	レジスタS1~S1+3の内容(BCD 8桁)と4桁のBCD定数nを乗算して、レジスタDからの8バイトに格納する。																										
演算内容	(S1~S1+3) × n → D~D+5																										
S1	使用範囲 C																										
n	使用範囲 0000 ~ 9999																										
D	使用範囲 G																										
演算条件	入力信号の立上り(OFF ON)																										
演算後の内容	S1~S1+3	不変																									
	D~D+5	<p>演算結果</p> <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> <p>・レジスタS1~S1+3の内容がBCDコード以外のとき不変。 (演算を実行しない)</p>			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
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 S1~S1+3の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354																						
	BCDコード	0	0	0	0																						
	BCDコード以外	0	0	1	0																						

[使用例]



命 令	
STR	004001
Fc15d	009000
	0100
	009020

入力条件004001がOFF ONの変化時に、レジスタ009000~009003のBCD 8桁とBCD定数0100(4桁)を乗算して、レジスタ009020からの8バイトに格納します。

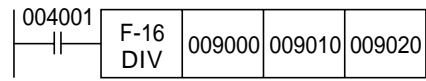


**F-16
DIV**

**レジスタ(BCD 4桁)とレジスタ(BCD 2桁)の除算
(DIVide)**

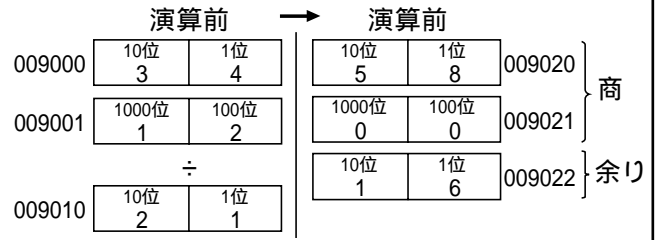
シンボル	— F-16 DIV S1 S2 D			
機能	レジスタS1、S1+1の内容(BCD 4桁)を、レジスタS2の内容(BCD 2桁)で除算し、レジスタDからの2バイトに商、3バイト目に余を格納する。			
演算内容	$(S1, S1+1) \div S2 \rightarrow D, D+1, D+2$			
S1	使用範囲 B			
S2	使用範囲 A			
D	使用範囲 E			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S1, S1+1	不変		
	S2	不変		
	D	演算結果の商 (1と10の位)	レジスタS1、S1+1、S2の内容がBCDコード以外 のとき、S2の内容が00(H)のとき、不変。 (演算を実行しない)	
	D+1	演算結果の商 (100と1000の位)		
フラグ	D+2	演算結果の余		
	レジスタS1、S1+1、S2の内容	ゼロ 007357	キャリー 007356	エラー 007355
	BCDコード	0	0	0
	・BCDコード以外 ・S2の内容が00(H)	0	0	1

[使用例]



命 令	
STR	004001
F-16	009000
	009010
	009020

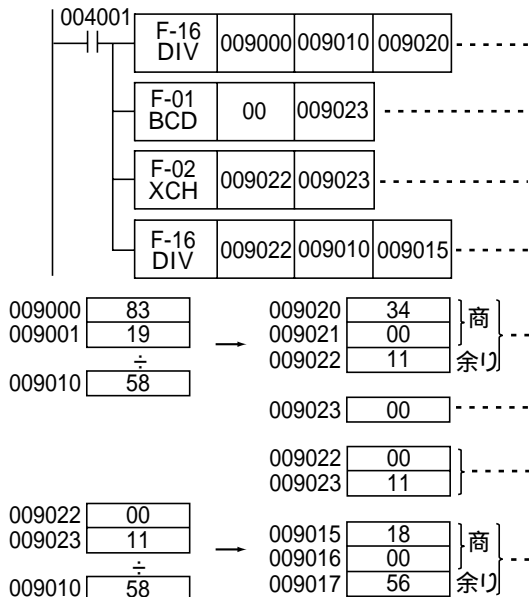
入力条件004001がOFF ONの変化時に、レジスタ009000、009001のBCD 4桁を、レジスタ009010のBCD 2桁で除算し、レジスタ009020からの2バイトに商、3バイト目に余りを格納します。
下記の演算は、 $1234 \div 21 = 58$ 余り16を示します。



・分子 < 分母 (S1 < S2, S1+1 = 0) のとき、演算結果の商 (D, D+1の内容) は0、余り (D+2の内容) は分子 (S1の内容) となります。例として $20 \div 30$ を実行すると、答えは0余り20です。

【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



命 令	
STR	004001
F-16	009000
	009010
	009020
F-01	00
	009023
F-02	009022
	009023
F-16	009022
	009010
	009015

入力条件004001がOFF ONのとき、レジスタ009000、009001の内容をレジスタ009010の内容で除算し、結果は009020、009021に商、009022に余りを格納する。

009023にデータ00(H)を入れる。

009022と009023の内容を交換し、余りを千、百の位に変換します。

のデータを再度009010の内容で除算し、009015、009016に商、009017に余りを格納する。

009015に格納したデータが小数点以下の2桁になる。

**F-16d
DIV**

**レジスタ間(BCD 8桁)とレジスタ(BCD 8桁)の除算
(DIVide)**

シンボル	— F-16d DIV S1 S2 D				[使用例]	<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004001</td></tr> <tr><td>F-16d</td><td>009000</td></tr> <tr><td></td><td>009010</td></tr> <tr><td></td><td>009020</td></tr> </table>	命 令		STR	004001	F-16d	009000		009010		009020
命 令																
STR	004001															
F-16d	009000															
	009010															
	009020															
機能	レジスタS1～S1+3の内容(BCD 8桁)をレジスタS2～S2+3の内容(BCD 8桁)で除算し、レジスタD～D+3に商、D+4～D+7に余を格納する。															
演算内容	$(S1 \sim S1+3) \div (S2 \sim S2+3) \rightarrow D \sim D+7$															
S1	使用範囲 C															
S2	使用範囲 C															
D	使用範囲 G															
演算条件	入力信号の立上り(OFF ON)															
演算後の内容	S1-S1+3	不変														
	S2-S2+3	不変														
	D～D+3	演算結果の商(BCD 8桁)	・レジスタS1～S1+3、S2～S2+3の内容がBCDコード以外するとき、およびS2～S2+3の内容が0のとき、不変。(演算を実行しない)													
	D+4～D+7	演算結果の余(BCD 8桁)														
	フラグ	レジスタS1-S1+3、S2-S2+3の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354										

004001	F-16d DIV	009000	009010	009020
--------	-----------	--------	--------	--------

入力条件004001がOFF ONの変化時に、レジスタ009000～009003のBCD 8桁を、レジスタ009010～009013のBCD 8桁で除算し、レジスタ009020～009023に商、009024～009027に余りを格納します。下記の演算は、 $12345678 \div 4321 = 2857$ 余り581を示します。

	演算前		→	演算後		
009000	10^1 7	10^0 8		10^1 5	10^0 7	009020
009001	10^3 5	10^2 6		10^3 2	10^2 8	009021
009002	10^5 3	10^4 4		10^5 0	10^4 0	009020
009003	10^7 1	10^6 2		10^7 0	10^6 0	009023
	÷					
009010	10^1 2	10^0 1		10^1 8	10^0 1	009024
009011	10^3 4	10^2 3		10^3 0	10^2 5	009025
009012	10^5 0	10^4 0		10^5 0	10^4 0	009026
009013	10^7 0	10^6 0		10^7 0	10^6 0	009027

S1、S2、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

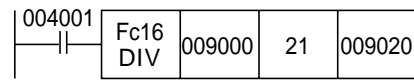
- 分子 < 分母 (S1～S1+3 < S2～S2+3) のとき、演算結果の商(D～D+3の内容)は0、余り(D+4～D+7の内容)は分子(S1～S1+3の内容)となります。例として20÷30を実行すると、答は0余り20です。

**Fc16
DIV**

**レジスタ(BCD 4桁)とBCD定数(2桁)の除算
(DIVide)**

シンボル	— Fc16 DIV S ₁ n D				
機能	レジスタS ₁ 、S ₁ +1の内容(BCD 4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、3バイト目に余を格納する。				
演算内容	(S ₁ 、S ₁ +1) ÷ n → D、D+1、D+2				
S ₁	使用範囲 B				
n	使用範囲 00 ~ 99				
D	使用範囲 E				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S ₁ 、S ₁ +1	不変			
	D	演算結果の商 (1と10の位)	・レジスタS ₁ 、S ₁ +1の内容がBCDコード以外 のとき、およびnが 00のとき、不変。 (演算を実行しない)		
	D+1	演算結果の商 (100と1000の位)			
	D+2	演算結果の余			
フラグ	レジスタS ₁ 、 S ₁ +1、nの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	BCDコード			0	
	・BCDコード 以外 ・nが00	0	0	1	0

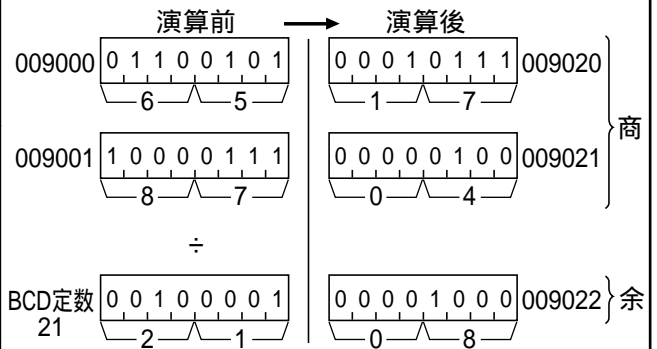
[使用例]



命 令	
STR	004001
Fc16	009000
	21
	009020

入力条件004001がOFF ONの変化時に、レジスタ009000、009001のBCD 4桁を、BCD定数21で除算し、レジスタ009020から2バイトに商、3バイト目に余りを格納します。

8765 ÷ 21 = 417余り8



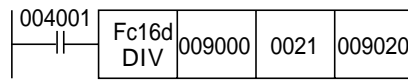
・分子 < 分母 (S₁ < n、S₁+1 = 0) のとき、演算結果の商(D、D+1の内容)は0、余り(D+2の内容)は分子(S₁の内容)となります。例として20 ÷ 30を実行すると、答は0余り20です。

**Fc16d
DIV**

**レジスタ(BCD 8桁)とBCD定数(4桁)の除算
(DIVide)**

シンボル	— Fc16d DIV S1 n D				
機能	レジスタS1～S1+3の内容(BCD 8桁を、4桁のBCD定数nで除算し、レジスタD～D+3に商、D+4～D+7に余を格納する。				
演算内容	$(S1 \sim S1+3) \div n \rightarrow D \sim D+7$				
S1	使用範囲 C				
n	使用範囲 0000～9999				
D	使用範囲 G				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S1～S1+3	不変			
	D～D+3	演算結果の商(BCD 8桁)	・レジスタS1～S1+3の内容がBCDコード以外 のとき、およびnの内容が0000のとき、不変。 (演算を実行しない)		
	D+4～D+7	演算結果の余(BCD 8桁)			
	フラグ	レジスタS1～S1+3、nの内容	ゼロ 007357	キャリー 007356	エラー 007355
	BCDコード			0	
	・BCDコード以外 ・nの内容が0000	0	0	1	0

[使用例]



命 令	
STR	004001
Fc16d	009000
	0021
	009020

入力条件004001がOFF ONの変化時に、レジスタ009000～009003のBCD 8桁を、BCD定数0021で除算し、レジスタ009020～009023に商、009024～009027に余りを格納します。

演算前		演算後	
009000	$\begin{matrix} 10^1 & 10^0 \\ 0 & 0 \end{matrix}$	$\begin{matrix} 10^1 & 10^0 \\ 0 & 9 \end{matrix}$	009020
009001	$\begin{matrix} 10^3 & 10^2 \\ 0 & 0 \end{matrix}$	$\begin{matrix} 10^3 & 10^2 \\ 3 & 8 \end{matrix}$	009021
009002	$\begin{matrix} 10^5 & 10^4 \\ 6 & 5 \end{matrix}$	$\begin{matrix} 10^5 & 10^4 \\ 1 & 7 \end{matrix}$	009022
009003	$\begin{matrix} 10^7 & 10^6 \\ 8 & 7 \end{matrix}$	$\begin{matrix} 10^7 & 10^4 \\ 0 & 4 \end{matrix}$	009023
÷			
BCD 定数	$\begin{matrix} 2 & 1 \end{matrix}$	$\begin{matrix} 10^1 & 10^0 \\ 1 & 1 \end{matrix}$	009024
	$\begin{matrix} 0 & 0 \end{matrix}$	$\begin{matrix} 10^3 & 10^2 \\ 0 & 0 \end{matrix}$	009025
		$\begin{matrix} 10^5 & 10^4 \\ 0 & 0 \end{matrix}$	009026
		$\begin{matrix} 10^7 & 10^6 \\ 0 & 0 \end{matrix}$	009027

上記の演算は、87650000 ÷ 21 = 4173809余り11を示します。

S1、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

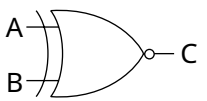
・分子 < 分母 (S1～S1+3 < n) のとき、演算結果の商(D～D+3の内容)は0、余り(D+4～D+7の内容)は分子(S1～S1+3の内容)となります。例として20 ÷ 30を実行すると、答は0余り20です。

F-17 XNR

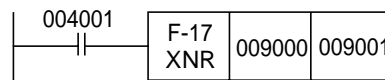
レジスタ間(1 バイト)の一致 (eXclusive NoR)

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	使用範囲 A	
D	使用範囲 A	
演算条件	入力信号の立上り(OFF ON)	
演算後の内容	S	不変
	D	演算結果
	フラグ	不変

否定排他的論理和(exclusive NOR)の真理値表

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

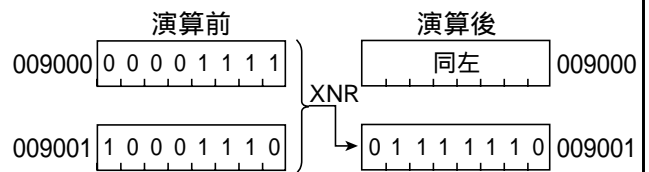
[使用例]



命 令	
STR	004001
F-17	009000 009001

入力条件004001がOFF ONの変化時に、レジスタ009000の内容とレジスタ009001の内容の否定排他的論理和(exclusive NOR)をとり、レジスタ009001に格納します。

なお、レジスタ009000の内容は不変です。



・009000と009001で一致したビット(0と0、1と1)は1、不一致のビット(0と1)は0になります。

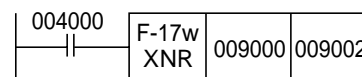
F-17w XNR

レジスタ間(1 ワード)の一致 (eXclusive NoR)

シンボル	F-17w XNR S D	
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	使用範囲 B	
D	使用範囲 B	
演算条件	入力信号の立上り(OFF ON)	
演算後の内容	S、S+1	不変
	D、D+1	演算結果
	フラグ	不変

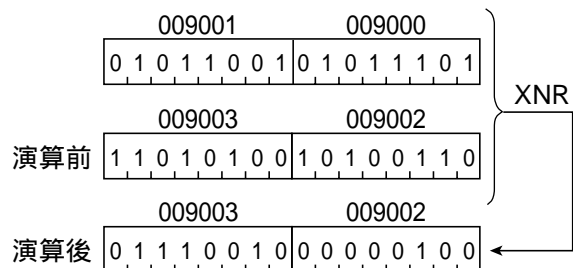
S、Dには必ず偶数アドレスを設定してください。
(019003等は禁止)

[使用例]



命 令	
STR	004000
F-17w	009000 009002

入力条件004000がOFF ONの変化時に、レジスタ009000、009001の内容(16ビットデータ)と、レジスタ009002、009003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009002、009003に格納します。なお、レジスタ009000、009001の内容は不変です。



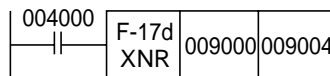
**F-17d
XNR**

**レジスタ間(2ワード)の一致
(eXclusive NoR)**

シンボル	$\overline{\text{F-17d XNR}}$ 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	$\overline{S \sim S+3} \oplus \overline{D \sim D+3} \rightarrow D \sim D+3$
S	使用範囲 C
D	使用範囲 C
演算条件	入力信号の立上り(OFF ON)
演算後の内容	S ~ S+3 不変
	D ~ D+3 演算結果
	フラグ 不変

S、Dには必ず偶数アドレスを設定してください。
(019003等は禁止)

[使用例]



命 令	
STR	004000
F-17d	009000 009004

入力条件004000がOFF ONの変化時に、レジスタ009000 ~ 009003の内容(32ビットデータ)と、レジスタ009004 ~ 009007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009004 ~ 009007に格納します。

演算前

009003	009002	009001	009000
01011001	01011101	01011001	01011101

XNR

009007	009006	009005	009004
11010100	10100110	11010100	10100110

演算後

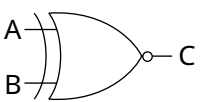
009007	009006	009005	009004
01110010	00000100	01110010	00000100

なお、レジスタ009000 ~ 009003の内容は不変です。

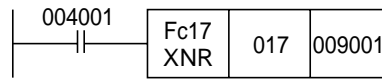
Fc17 XNR レジスタと 8 進定数(1 バイト)の一致 (XNR)

シンボル	$\overline{\text{Fc17 XNR}} \quad n \quad D$
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、レジスタ D に格納する。
演算内容	$\overline{n \oplus D} \rightarrow D$
n	使用範囲 000 ~ 377
D	使用範囲 A
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D 演算結果
	フラグ 不変

否定排他的論理和(exclusive NOR)の真理値表

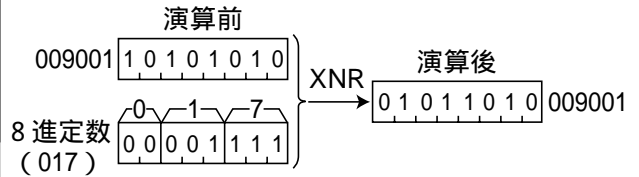
シンボル	A	B	C
	0	0	1
	1	0	0
	0	1	0
	1	1	1

[使用例]



命 令	
STR	004001
Fc17	017
	009001

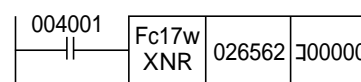
入力条件004001がOFF ONの変化時に、8 進定数017とレジスタ009001の内容の否定排他的論理和(exclusive NOR)をと、レジスタ009001に格納します。



Fc17w XNR レジスタと 8 進定数(1 ワード)の一致 (eXclusive NoR)

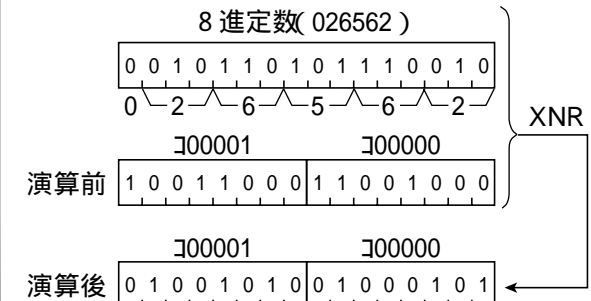
シンボル	$\overline{\text{Fc17w XNR}} \quad n \quad D$
機能	8 進定数 n とレジスタ D、D+1 の内容(16 ビットデータ)の否定排他的論理和をと、レジスタ D、D+1 に格納する。
演算内容	$\overline{n \oplus D, D+1} \rightarrow D, D+1$
n	使用範囲 000000 ~ 177777(8)
D	使用範囲 B ・ D には必ず偶数アドレスを設定してください。(300011等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D、D+1 演算結果
	フラグ 不変

[使用例]



命 令	
STR	004001
Fc17w	026562
	300000

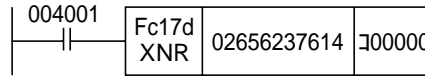
入力条件004001がOFF ONの変化時に、8 進定数026562とレジスタ300000、300001の内容(16 ビットデータ)の否定排他的論理和をと、レジスタ300000、300001に格納します。



Fc17d XNR レジスタと8進定数(2ワード)の一致
(eXclusive NoR)

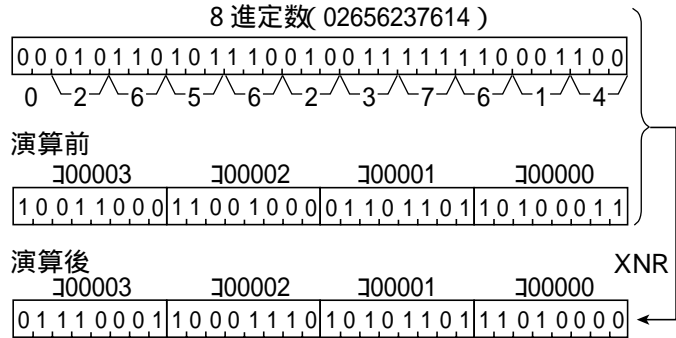
シンボル	$\overline{\text{Fc17d XNR}}$ n D
機能	8進定数nとレジスタD~D+3の内容(32ビットデータ)の否定排他的論理和をとり、レジスタD~D+3に格納する。
演算内容	$\overline{n \oplus D \sim D+3} \rightarrow D \sim D+3$
n	使用範囲 000000000000 ~ 17777777777(8)
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。(300011等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算内容	D~D+3 演算結果
演算後	フラグ 不変

[使用例]



命 令	
STR	004001
Fc17d	02656237614 300000

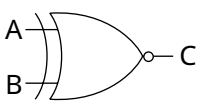
入力条件004001がOFF ONの変化時に、8進定数02656237614とレジスタ300000~300003の内容(32ビットデータ)の否定排他的論理和をとり、レジスタ300000~300003に格納します。



Fx17 XNR レジスタと16進定数(1バイト)の一致
(eXclusive NoR)

シンボル	$\overline{\text{Fx17 XNR}}$ n D
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、レジスタ D に格納する。
演算内容	$\overline{n \oplus D} \rightarrow D$
n	使用範囲 00 ~ FF
D	使用範囲 A
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D 演算結果
	フラグ 不変

否定排他的論理和(exclusive NOR)の真理値表

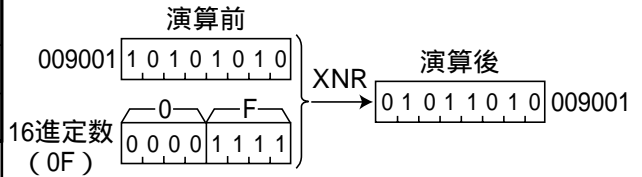
シンボル	A	B	C
	0	0	1
	1	0	0
	0	1	0
	1	1	1

[使用例]



命 令	
STR	004001
Fx17	0F
	009001

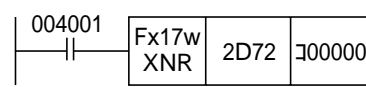
入力条件004001がOFF ONの変化時に、16進定数0Fとレジスタ009001の内容の否定排他的論理和(exclusive NOR)をと、レジスタ009001に格納します。



Fx17w XNR レジスタと16進定数(1ワード)の一致
(eXclusive NoR)

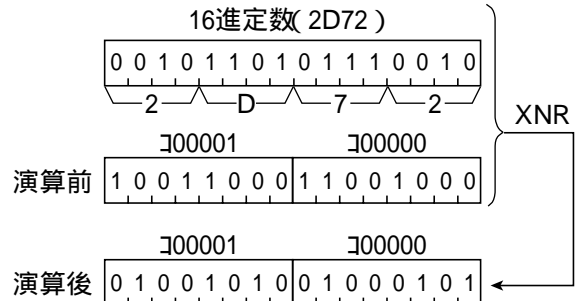
シンボル	$\overline{\text{Fx17w XNR}}$ n D
機能	16進定数 n とレジスタ D、D+1 の内容(16ビットデータ)の否定排他的論理和をと、レジスタ D、D+1 に格納する。
演算内容	$\overline{n \oplus D, D+1} \rightarrow D, D+1$
n	使用範囲 0000 ~ FFFF(H)
D	使用範囲 B ・ D には必ず偶数アドレスを設定してください。(300011等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D、D+1 演算結果
	フラグ 不変

[使用例]



命 令	
STR	004001
Fx17w	2D72
	300000

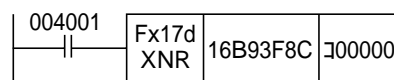
入力条件004001がOFF ONの変化時に、16進定数2D72とレジスタ300000、300001の内容(16ビットデータ)の否定排他的論理和をと、レジスタ300000、300001に格納します。



Fx17d XNR レジスタと16進定数(2ワード)の一致
(eXclusive NoR)

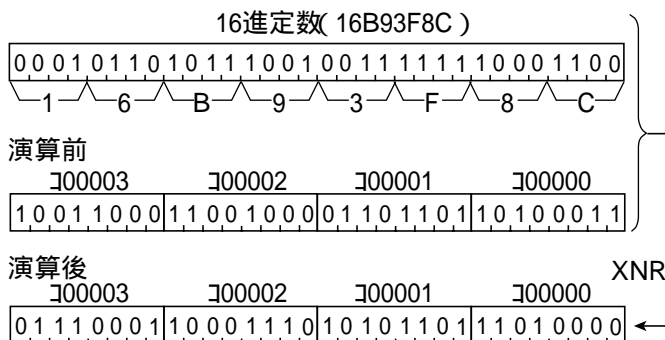
シンボル	$\overline{\text{Fx17d XNR}}$ n D
機能	16進定数nとレジスタD~D+3の内容(32ビットデータ)の否定排他的論理和をとり、レジスタD~D+3に格納する。
演算内容	$n \oplus D \sim D+3 \rightarrow D \sim D+3$
n	使用範囲 00000000 ~ FFFFFFFF(H)
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。(300011等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算内容 後	D~D+3 演算結果
	フラグ 不変

[使用例]



命 令	
STR	004001
Fx17d	16B93F8C
	300000

入力条件004001がOFF ONの変化時に、16進定数16B93F8Cとレジスタ300000 ~ 300003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300000 ~ 300003に格納します。

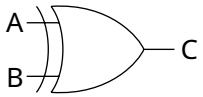


F-18 XOR

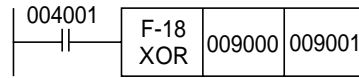
レジスタ間(1 バイト)の排他的論理和 (eXclusive OR)

シンボル	F-18 XOR S D		
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	使用範囲 A		
D	使用範囲 A		
演算条件	入力信号の立上り(OFF ON)		
演算後の内容	S	不変	
	D	演算結果	
	フラグ	不変	

排他的論理和(Exclusive OR)の真理値表

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

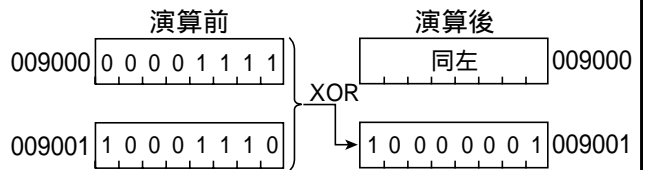
[使用例]



命 令	
STR	004001
F-18	009000 009001

入力条件004001がOFF ONの変化時に、レジスタ009000の内容とレジスタ009001の内容の排他的論理和(exclusive OR)をとり、レジスタ009001に格納します。

なお、レジスタ009000の内容は不変です。



・009000と009001で不一致のビット(0と1)は1、一致のビット(0と0、1と1)は0になります。

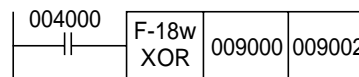
F-18w XOR

レジスタ間(1 ワード)の排他的論理和 (eXclusive OR)

シンボル	F-18w XOR S D		
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	$S, S+1 \oplus D, D+1 \rightarrow D, D+1$		
S	使用範囲 B		
D	使用範囲 B		
演算条件	入力信号の立上り(OFF ON)		
演算後の内容	S、S+1	不変	
	D、D+1	演算結果	
	フラグ	不変	

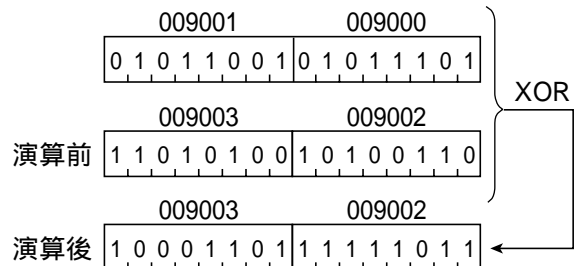
S、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

[使用例]



命 令	
STR	004000
F-18w	009000 009001 009002 009003

入力条件004000がOFF ONの変化時に、レジスタ009000、009001の内容(16ビットデータ)と、レジスタ009002、009003の内容(16ビットデータ)の排他的論理和をとり、レジスタ009002、009003に格納します。



なお、レジスタ009000、009001の内容は不変です。

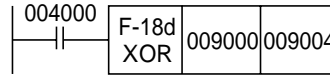
**F-18d
XOR**

**レジスタ間(2ワード)の排他的論理和
(eXclusive OR)**

シンボル	$\overline{\text{F-18d XOR}} \text{ S D}$
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	$S \sim S+3 \oplus D \sim D+3 \rightarrow D \sim D+3$
S	使用範囲 C
D	使用範囲 C
演算条件	入力信号の立上り(OFF ON)
演算後の内容	S ~ S+3 不変
	D ~ D+3 演算結果
	フラグ 不変

S、Dには必ず偶数アドレスを設定してください。
(019003等は禁止)

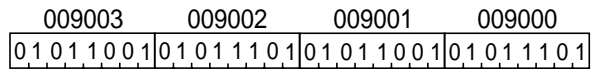
[使用例]



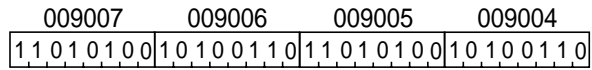
命 令	
STR	004000
F-18d	009000 009004

入力条件004000がOFF ONの変化時に、レジスタ009000~009003の内容(32ビットデータ)と、レジスタ009004~009007の内容(32ビットデータ)の排他的論理和をとり、レジスタ009004~009007に格納します。

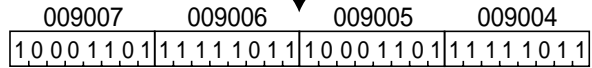
演算前



XOR



演算後



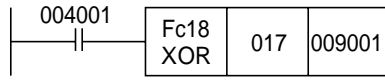
なお、レジスタ009000~009003の内容は不変です。

Fc18 XOR

レジスタと8進定数(1バイト)の排他的論理和
(eXclusive OR)

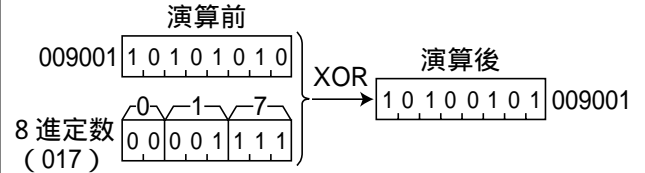
シンボル	$\overline{\text{Fc18 XOR}} \quad n \quad D$
機能	8進定数nとレジスタDの内容の排他的論理和をとり、レジスタDに格納する。
演算内容	$n \oplus D \rightarrow D$
n	使用範囲 000 ~ 377
D	使用範囲 A
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D 演算結果
	フラグ 不変

[使用例]



命 令	
STR	004001
Fc18	017
	009001

入力条件004001がOFF ONの変化時に、8進定数017とレジスタ009001の内容の排他的論理和(exclusive OR)をとり、レジスタ009001に格納します。



排他的論理和(Exclusive OR)の真理値表

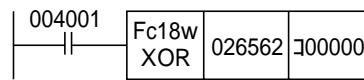
シンボル	A	B	C
	0	0	0
	1	0	1
	0	1	1
	1	1	0

Fc18w XOR

レジスタと8進定数(1ワード)の排他的論理和
(eXclusive OR)

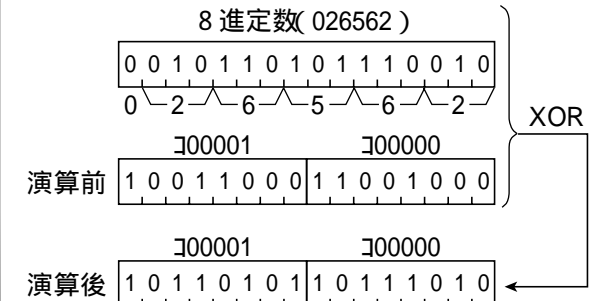
シンボル	$\overline{\text{Fc18w XOR}} \quad n \quad D$
機能	8進定数nとレジスタD、D+1の内容(16ビットデータ)の排他的論理和をとり、レジスタD、D+1に格納する。
演算内容	$n \oplus D, D+1 \rightarrow D, D+1$
n	使用範囲 000000 ~ 177777(8)
D	使用範囲 B ・Dには必ず偶数アドレスを設定してください。(300011等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D、D+1 演算結果
	フラグ 不変

[使用例]



命 令	
STR	004001
Fc18w	026562
	300000

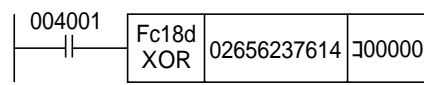
入力条件004001がOFF ONの変化時に、8進定数026562とレジスタ300000、300001の内容(16ビットデータ)の排他的論理和をとり、レジスタ300000、300001に格納します。



Fc18d XOR レジスタと8進定数(2ワード)の排他的論理和
(eXclusive OR)

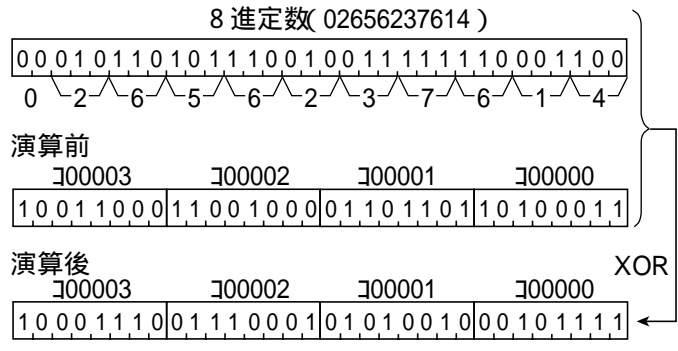
シンボル	$\overline{\text{Fc18d XOR}}$ n D
機能	8進定数nとレジスタD~D+3の内容(32ビットデータ)の排他的論理和をとり、レジスタD~D+3に格納する。
演算内容	$n \oplus D \sim D+3 \rightarrow D \sim D+3$
n	使用範囲 00000000000 ~ 3777777777(8)
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。(300011等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算内容 の 後	D~D+3 演算結果
	フラグ 不変

[使用例]



命 令	
STR	004001
Fc18d	02656237614 300000

入力条件004001がOFF ONの変化時に、8進定数02656237614とレジスタ300000~300003の内容(32ビットデータ)の排他的論理和をとり、レジスタ300000~300003に格納します。



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	$\text{Fx18 XOR } n \text{ D}$	[使用例]	<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004001</td></tr> <tr><td>Fx18</td><td>0F</td></tr> <tr><td></td><td>009001</td></tr> </table>	命 令		STR	004001	Fx18	0F		009001
命 令											
STR	004001										
Fx18	0F										
	009001										
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。		<p>入力条件004001がOFF ONの変化時に、16進定数0Fとレジスタ009001の内容の排他的論理和(exclusive OR)をとり、レジスタ009001に格納します。</p>								
演算内容	$n \oplus D \rightarrow D$	<p>演算前</p> <p>009001 $\begin{matrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{matrix}$</p> <p>16進定数 (0F) $\begin{matrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{matrix}$</p> <p>XOR</p> <p>演算後 $\begin{matrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{matrix}$ 009001</p>									
n	使用範囲 00 ~ FF										
D	使用範囲 A										
演算条件	入力信号の立上り(OFF ON)										
演算後の内容	D 演算結果										
フラグ	不変										

排他的論理和(Exclusive OR)の真理値表

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	$\text{Fx18w XOR } n \text{ D}$	[使用例]	<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004001</td></tr> <tr><td>Fx18w</td><td>2D72</td></tr> <tr><td></td><td>300000</td></tr> </table>	命 令		STR	004001	Fx18w	2D72		300000
命 令											
STR	004001										
Fx18w	2D72										
	300000										
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。		<p>入力条件004001がOFF ONの変化時に、16進定数2D72とレジスタ300000、300001の内容(16ビットデータ)の排他的論理和をとり、レジスタ300000、300001に格納します。</p>								
演算内容	$n \oplus D, D+1 \rightarrow D, D+1$	<p>16進定数(2D72)</p> <p>$\begin{matrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{matrix}$</p> <p>2 ————— D ————— 7 ————— 2</p> <p>演算前</p> <p>300001 $\begin{matrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{matrix}$</p> <p>300000 $\begin{matrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{matrix}$</p> <p>XOR</p> <p>演算後</p> <p>300001 $\begin{matrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{matrix}$</p>									
n	使用範囲 0000 ~ FFFF(H)										
D	使用範囲 B ・ D には必ず偶数アドレスを設定してください。(300011等は禁止)										
演算条件	入力信号の立上り(OFF ON)										
演算後の内容	D、D+1 演算結果										
フラグ	不変										

**Fx18d
XOR**

**レジスタと16進定数(2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+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	$n \oplus D \sim D+3 \rightarrow D \sim D+3$
n	使用範囲 00000000 ~ FFFFFFFF(H)
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。(000011等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算内容	D~D+3 演算結果
	フラグ 不変

[使用例]

命 令	
STR	004001
Fx18d	16B93F8C 000000

入力条件004001がOFF ONの変化時に、16進定数16B93F8Cとレジスタ000000 ~ 000003の内容(32ビットデータ)の排他的論理和をとり、レジスタ000000 ~ 000003に格納します。

16進定数(16B93F8C)

00010110101110010011111110001100

1 6 B 9 3 F 8 C

演算前

000003	000002	000001	000000
10011000	11001000	01101101	10100011

演算後 XOR

000003	000002	000001	000000
10001110	01110001	01010010	00101111

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。
MD(メンテナンスディスプレイ)参照

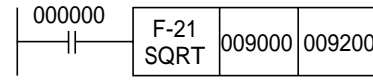
F-21 SQRT

レジスタBCD(8桁)の平方根 (Square Root)

シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 30px;">F-21 SQRT</td> <td style="width: 30px;">S</td> <td style="width: 30px;">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(BCD 8桁)の平方根を求め、結果をレジスタD、D+1に格納する。小数点以下は切り捨てる。															
演算内容	$\sqrt{(S \sim S+3)} \rightarrow D, D+1$															
S	使用範囲 C															
D	使用範囲 B															
演算条件	入力信号の立上り(OFF ON)															
演算後の内容	S~S+3	不変														
	D、D+1	演算結果 ・レジスタS~S+3の内容がBCDコード以外のとき不変。(演算は実行しない)														
	フラグ	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td>レジスタ S~S+3の内容</td> <td>ゼロ 007357</td> <td>キャリー 007356</td> <td>エラー 007355</td> <td>ノンキャリー 007354</td> </tr> <tr> <td>BCDコード</td> <td>0</td> <td>0</td> <td>0</td> <td>0</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>	レジスタ S~S+3の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	BCDコード	0	0	0	0	BCDコード以外	0	0	1
レジスタ S~S+3の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354												
BCDコード	0	0	0	0												
BCDコード以外	0	0	1	0												

S、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

[使用例]



命 令	
STR	000000
F-21	009000 009200

入力条件000000がOFF ONの変化時に、レジスタ009000~009003のBCD 8桁の平方根を求め、レジスタ009200、009201に格納します。



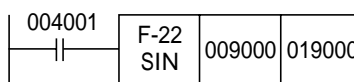
・演算結果の小数点以下は切り捨てます。

F-22 SIN

三角関数(SIN)の演算

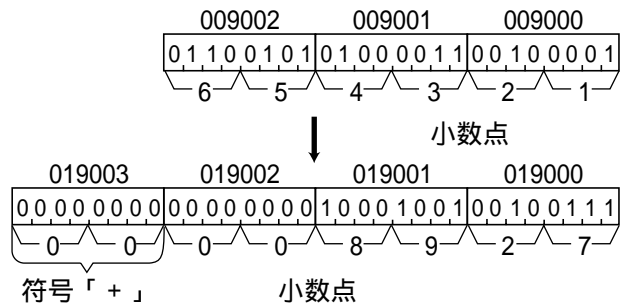
シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 30px;">F-22 SIN</td> <td style="width: 30px;">S</td> <td style="width: 30px;">D</td> </tr> </table>	F-22 SIN	S	D	
F-22 SIN	S	D			
機能	レジスタS~S+2の内容(BCD 6桁)の正弦(SIN)を求め、レジスタD~D+3にBCD 8桁で格納する。				
演算内容	$\text{SIN}(S \sim S+2) \rightarrow D \sim D+3$				
S	使用範囲 E				
D	使用範囲 C				
演算条件	入力信号の立上り(OFF ON)				
演算前の内容	S	角度の小数部(BCD 2桁)			
	S+1、S+2	角度の整数部(BCD 4桁)			
		・角度の範囲 0 ~ 9999.99 °			
演算後の内容	S~S+2	不変			
	D、D+1	演算結果の小数部(BCD 4桁)			
	D+2	演算結果の整数部(BCD 2桁)			
	D+3	演算結果の符号 [00(H) : 正(+) (80(H) : 負(-)] (BCD 2桁)			
フ ラ グ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	正(+)	0	0	0	1
	負(-)	0	1	0	0
	S~S+2の内容がBCD以外	0	0	1	0

[使用例]



命 令	
STR	004001
F-22	009000 019000

入力条件004001がOFF ONの変化時に、レジスタ009000~009003(BCD 6桁データ)の正弦(SIN)を求め、演算結果をレジスタ019000~019003に格納します。下記の演算は、 $\text{SIN } 6543.21^\circ = 0.8927$ を示します。



D~D+3(演算後の内容)

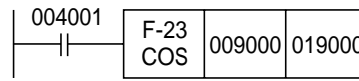
- ・演算結果の範囲は、- 1.0000 ~ 1.0000で、小数第 5 位を四捨五入する。
- ・S~S+2の内容がBCDコード以外のとき不変。

F-23
COS

三角関数(COS)の演算

シンボル	F-23 COS					S	D
機能	レジスタS~S+2の内容(BCD 6桁)の余弦(COS)を求め、レジスタD~D+3にBCD 8桁で格納する。						
演算内容	COS(S~S+2) → D~D+3						
S	使用範囲E						
D	使用範囲C						
演算条件	入力信号の立上り(OFF ON)						
演算前 の内容	S	角度の小数部(BCD 2桁)			・角度の範囲 0~9999.99°		
	S+1, S+2	角度の整数部(BCD 4桁)					
演算後 の内容	S~S+2	不変					
	D、D+1	演算結果の小数部(BCD 4桁)					
	D+2	演算結果の整数部(BCD 2桁)					
	D+3	演算結果の符号 [00(H): 正(+) 80(H): 負(-)] (BCD 2桁)					
フ ラ グ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354		
	正(+)	0	0	0	1		
	負(-)	0	1	0	0		
	S~S+2の内容 がBCD以外	0	0	1	0		

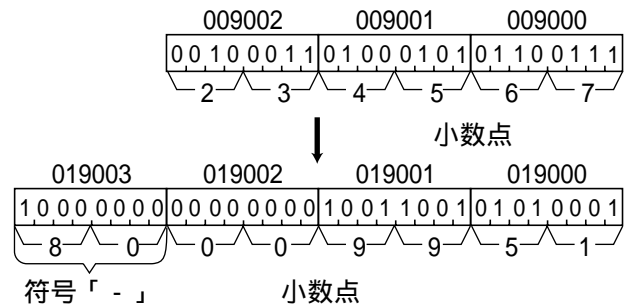
[使用例]



命 令	
STR	004001
F-23	009000 019000

入力条件004001がOFF ONの変化時に、レジスタ009000~009002(BCD 6桁データ)の余弦(COS)を求め、演算結果をレジスタ019000~019003に格納します。

COS 2345.67° - 0.9951の演算



D~D+3 演算後の内容)

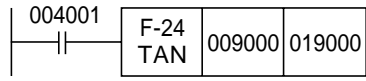
- ・演算結果は、範囲が - 1.0000 ~ 1.0000で、小数第5位を四捨五入する。
- ・S~S+2の内容がBCDコード以外るとき不変。

**F-24
TAN**

三角関数(TAN)の演算

シンボル	<table border="1"> <tr> <td>F-24 TAN</td> <td>S</td> <td>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(BCD 6桁)の正接(TAN)を求め、レジスタD~D+3にBCD 8桁で格納する。							
演算内容	TAN(S~S+2) → D~D+3							
S	使用範囲 E							
D	使用範囲 C							
演算条件	入力信号の立上り(OFF ON)							
演算前内容	S	角度の小数部(BCD 2桁)			・角度の範囲 0 ~ 9999.99°			
	S+1, S+2	角度の整数部(BCD 4桁)						
演算後内容	S~S+2	不変						
	D, D+1	演算結果の小数部(BCD 4桁)						
	D+2	演算結果の整数部(BCD 2桁)						
	D+3	演算結果の符号 (BCD 2桁) [00(H): 正(+)] [80(H): 負(-)]						
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354		
正(+)		0	0	0	1			
負(-)		0	1	0	0			
+100以上、 または-100以下 S~S+2の内容 がBCD以外		0	0	1	0			

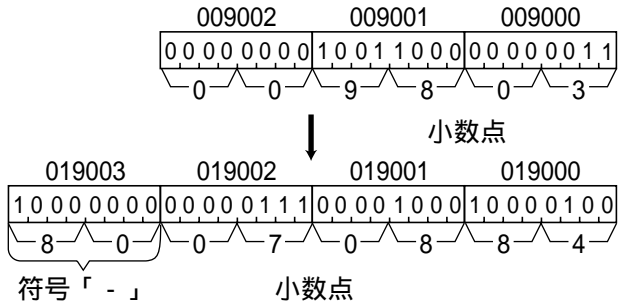
[使用例]



命 令	
STR	004001
F-24	009000 019000

入力条件004001がOFF ONの変化時に、レジスタ009000~009002(BCD 6桁データ)の正接(TAN)を求め、演算結果をレジスタ019000~019003に格納します。

TAN 98.03° - 7.0884の演算



D~D+3(演算後の内容)

- ・演算結果は、範囲が - 99.9999 ~ 99.9999 で、小数第5位を四捨五入する。
- ・S~S+2の内容がBCD以外るとき不変。

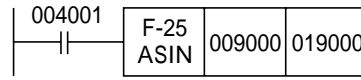
演算結果が100.0000以上または - 100.0000以下になるSの内容の場合、エラーフラグをONして演算しません。

F-25
ASIN

三角関数(SIN⁻¹)の演算

シンボル	F-25 ASIN S D				
機能	レジスタS~S+3の内容(BCD 8桁)の逆正弦(SIN ⁻¹)を求め、レジスタD~D+3にBCD 8桁で格納する。				
演算内容	SIN ⁻¹ (S~S+3) → D~D+3				
S	使用範囲C				
D	使用範囲C				
演算条件	入力信号の立上り(OFF ON)				
演算前の内容	S, S+1	SIN ⁻¹ データの小数部 (BCD 4桁)		・SIN ⁻¹ データの範囲 - 1.0000 ~ 1.0000	
	S+2	SIN ⁻¹ データの整数部 (BCD 2桁)			
	S+3	SIN ⁻¹ データの符号 (BCD 2桁) [00(H): 正(+)] [80(H): 負(-)]			
演算後の内容	S~S+3	不変			
	D	演算結果の小数部(BCD 2桁)			
	D+1, D+2	演算結果の整数部(BCD 4桁)			
	D+3	演算結果の符号 [00(H): 正(+)] [80(H): 負(-)] (BCD 2桁)			
フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	正(+)	0	0	0	1
	負(-)	0	1	0	0
	S~S+3の内容がBCD以外	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	0	0	1	0

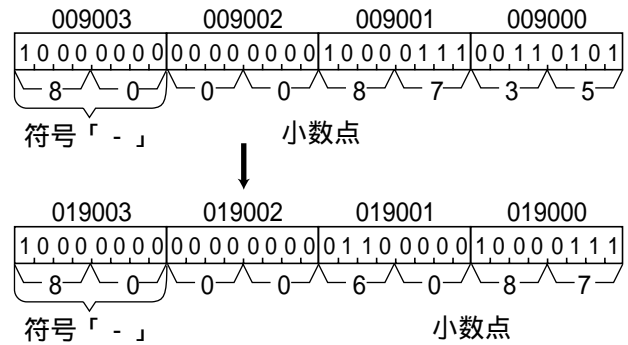
[使用例]



命 令	
STR	004001
F-25	009000 019000

入力条件004001がOFF ONの変化時に、レジスタ009000~009003(BCD 8桁データ)の逆正弦(SIN⁻¹)を求め、演算結果をレジスタ019000~019003に格納します。

SIN⁻¹(- 0.8735) - 60.87° の演算



D~D+3(演算後の内容)

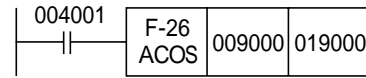
- ・演算結果は、範囲が - 90.00° ~ 90.00° で、小数第3位を四捨五入する。
- ・S~S+3の内容がBCD以外るとき不変。

F-26
ACOS

三角関数(COS⁻¹)の演算

シンボル	F-26 ACOS S D				
機能	レジスタS~S+3の内容(BCD 8桁)の逆余弦(COS ⁻¹)を求め、レジスタD~D+3にBCD 8桁で格納する。				
演算内容	COS ⁻¹ (S~S+3) → D~D+3				
S	使用範囲 C				
D	使用範囲 C				
演算条件	入力信号の立上り(OFF ON)				
演算前内容	S, S+1	COS ⁻¹ データの小数部 (BCD 4桁)		・COS ⁻¹ データの範囲 - 1.0000 ~ 1.0000	
	S+2	COS ⁻¹ データの整数部 (BCD 2桁)			
	S+3	COS ⁻¹ データの符号 (BCD 2桁) 00(H): 正(+) 80(H): 負(-)			
演算後内容	S~S+3	不変			
	D	演算結果の小数部(BCD 2桁)			
	D+1, D+2	演算結果の整数部(BCD 4桁)			
	D+3	演算結果の符号 00(H): 正(+) 80(H): 負(-)			
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355
	正(+)	0	0	0	1
	負(-)	0	1	0	0
	S~S+3の内容がBCD以外	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	0	0	1	0

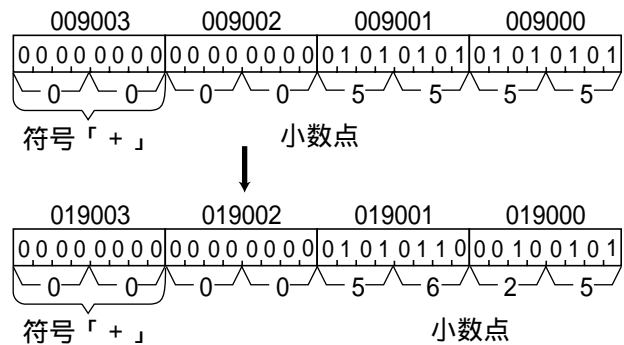
[使用例]



命令	
STR	004001
F-26	009000
	019000

入力条件004001がOFF ONの変化時に、レジスタ009000~009003(BCD 8桁データ)の逆余弦(COS⁻¹)を求め、演算結果をレジスタ019000~019003に格納します。

COS⁻¹(0.5555) 56.25° の演算



D~D+3(演算後の内容)

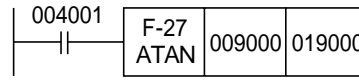
- ・演算結果は、範囲が0~180.00°で、小数第3位を四捨五入する。
- ・S~S+3の内容がBCD以外るとき不変。

**F-27
ATAN**

三角関数(TAN⁻¹)の演算

シンボル	F-27 ATAN S D				
機能	レジスタS~S+3の内容(BCD 8桁)の逆正接(TAN ⁻¹)を求め、レジスタD~D+3にBCD 8桁で格納する。				
演算内容	TAN ⁻¹ (S~S+3) → D~D+3				
S	使用範囲 C				
D	使用範囲 C				
演算条件	入力信号の立上り(OFF ON)				
演算の前内容	S, S+1	TAN ⁻¹ データの小数部 (BCD 4桁)		・TAN ⁻¹ データの範囲 - 98.9999 ~ 98.9999	
	S+2	TAN ⁻¹ データの整数部 (BCD 2桁)			
	S+3	TAN ⁻¹ データの符号 (BCD 2桁) [00(H): 正(+)] [80(H): 負(-)]			
演算後の内容	S~S+3	不変			
	D	演算結果の小数部 (BCD 2桁)			
	D+1, D+2	演算結果の整数部 (BCD 4桁)			
	D+3	演算結果の符号 [00(H): 正(+)] [80(H): 負(-)] (BCD 2桁)			
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355
	正(+)	0	0	0	1
	負(-)	0	1	0	0
	S~S+3の内容がBCD以外	0	0	1	0
	S~S+3の内容が+99.0000以上、または-99.0000以下	0	0	1	0

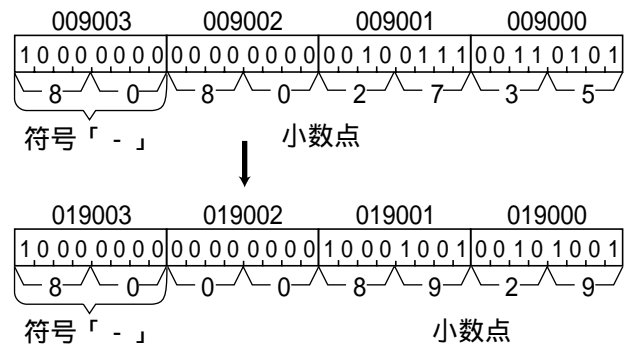
[使用例]



命 令	
STR	004001
F-27	009000 019000

入力条件004001がOFF ONの変化時に、レジスタ009000~009003(BCD 8桁データ)の逆正接(TAN⁻¹)を求め、演算結果をレジスタ019000~019003に格納します。

TAN⁻¹(- 80.2735) - 89.29° の演算



D~D+3(演算後の内容)

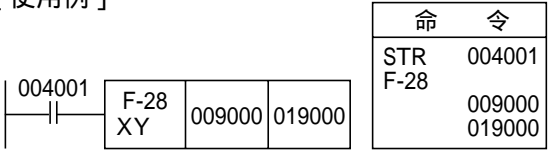
- ・演算結果は、範囲が - 89.42 ~ 89.42で、小数第3位を四捨五入する。
- ・S~S+3の内容がBCD以外のとき不変。

**F-28
XY**

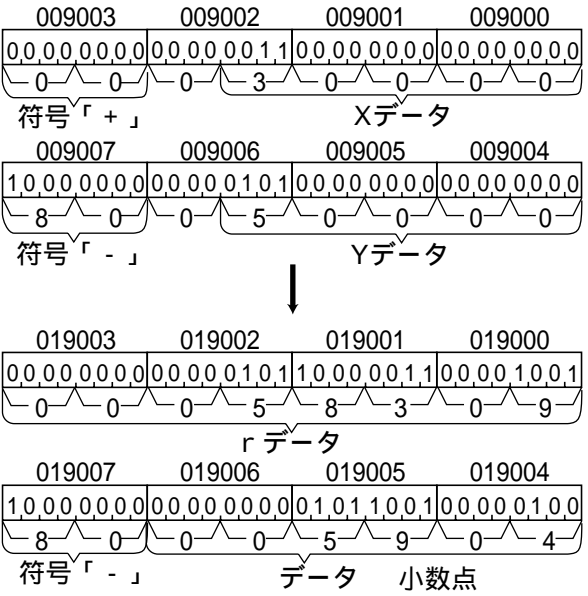
直交座標系(X、Y)データの極座標系(r、)変換

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="text-align: center;">F-28 XY</td><td style="text-align: center;">S</td><td style="text-align: center;">D</td></tr></table>				F-28 XY	S	D
F-28 XY	S	D					
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、)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) (D+4~D+7)						
S	使用範囲 G						
D	使用範囲 G						
演算条件	入力信号の立上り(OFF ON)						
演算前の内容	S~S+2	Xデータの整数部 (BCD 5桁)	・ Xデータの範囲 - 99999 ~ 99999				
	S+3	Xデータの符号 [00(H) : 正(+) 80(H) : 負(-)]					
	S+4 ~S+6	Yデータの整数部 (BCD 3桁)	・ Yデータの範囲 - 99999 ~ 99999				
	S+7	Yデータの符号 [00(H) : 正(+) 80(H) : 負(-)]					
演算後の内容	S~S+7	不変					
	D~D+3	rデータの整数部 (BCD 8桁)	・ X、YデータがBCD コード以外るとき不変				
	D+4	データの小数部 (BCD 2桁)	・ データの範囲 - 179.99 ~ 180.00 °				
	D+5、 D+6	データの整数部 (BCD 3桁)	・ X、YデータがBCD コード以外るとき不変				
	D+7	データの符号 [00(H) : 正(+) 80(H) : 負(-)]					
フラグ	レジスタSの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354		
	X、Yデータが BCDコード	0	0	0	0		
	X、Yデータが BCDコード以外	0	0	1	0		

[使用例]



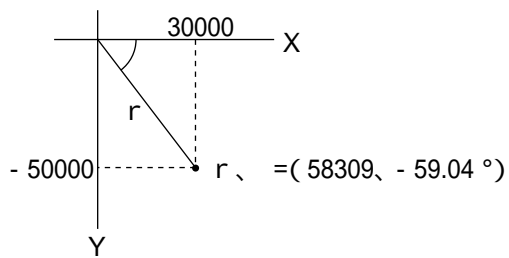
入力条件004001がOFF ONの変化時に、レジスタ009000~009007の直交座標(X、Y)データを極座標(r、)データに変換し、レジスタ019000~019007に格納します。



上記は次の演算を示します。

$$X(30000) Y(- 50000)$$

$$\rightarrow r(58309) (- 59.04^\circ)$$



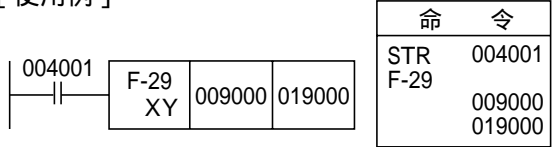
F-29
XY

極座標(r、)データの直交座標 X、Y 変換

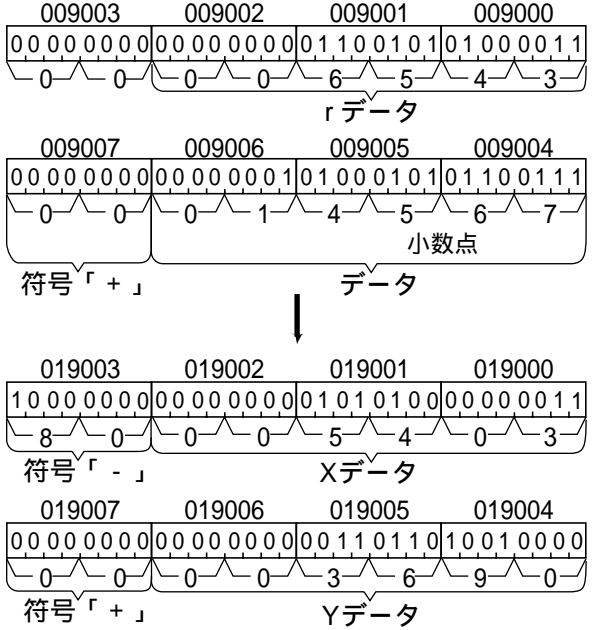
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="text-align: center;">F-29 XY</td><td style="text-align: center;">S</td><td style="text-align: center;">D</td></tr></table>					F-29 XY	S	D
F-29 XY	S	D						
機能	レジスタS~S+3とレジスタS+4~S+7の極座標(r、)を直交座標 X、Y)に変換し、レジスタD~D+3とレジスタD+4~D+7に格納する。							
演算内容	$((S \sim S+3) (S+4 \sim S+7) \rightarrow X(D \sim D+3) Y(D+4 \sim D+7)$							
S	使用範囲 G							
D	使用範囲 G							
演算条件	入力信号の立上り(OFF ON)							
演算前の内容	S~S+3	rデータの整数部 (BCD 8桁)		・ rデータの範囲 0~999999				
	S+4	データの小数部 (BCD 2桁)		・ データの範囲 -179.99~180.00				
	S+5、S+6	データの整数部 (BCD 3桁)						
	S+7	データの符号 (BCD 2桁) [00(H): 正(+)] [80(H): 負(-)]						
演算後の内容	S~S+7	不変						
	D~D+2	Xデータの整数部 (BCD 5桁)		・ r、データがBCDコード以外るとき不変				
	D+3	Xデータの符号 (BCD 2桁) [00(H): 正(+)] [80(H): 負(-)]						
	D+4~D+6	Yデータの整数部 (BCD 5桁)						
	D+7	Yデータの符号 (BCD 2桁) [00(H): 正(+)] [80(H): 負(-)]						
フラグ	レジスタSの内容	ゼロ	キャリー	エラー	ノンキャリー			
		007357	007356	007355	007354			
	r、データがBCD	0	0	0	0			
	r、データがBCD以外	0	0	1	0			

・ S+4~S+7のデータが -180.00以下、または 180.01以上のときは演算しません。

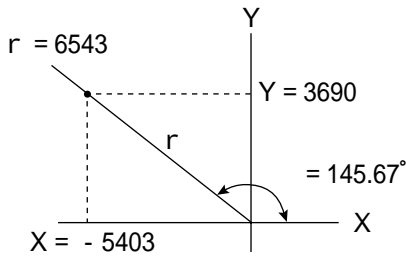
[使用例]



入力条件004001がOFF ONの変化時に、レジスタ009000~009007の極座標(r、)データを直交座標(r、)データに変換し、レジスタ019000~019007に格納します。



上記は次の演算を示します。
r(6543) (145.67°)
→ X(-5403) Y(3690)



**F-30
MCS**

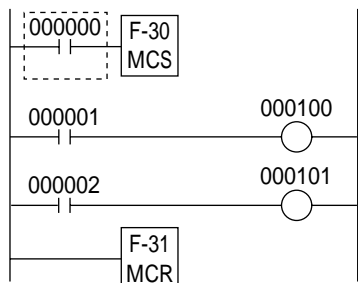
**マスターコントロールセット
(Master Control Set)**

**F-31
MCR**

**マスターコントロールリセット
(Master Control Reset)**

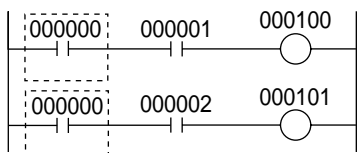
MCS、MCRは、共通演算条件以後の回路が複数の出力に分岐している場合に使用します。

・MCS、MCRを使用した場合

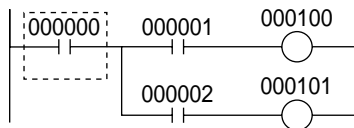


MCS	STR	000000
	F-30	
	STR	000001
	OUT	000100
	STR	000002
	OUT	000101
MCR	F-31	

・MCS、MCRを使わない場合



・リレー盤の場合



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。

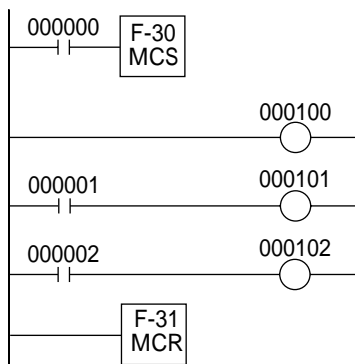
F-31(MCR)は、このANDする範囲の終了を意味します。

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。

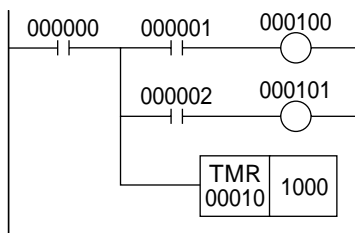
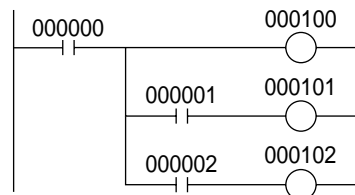
F-30(MCS)で派生した母線に、直接OUT、TMR、CNTの各命令および応用命令は接続できません。

・MCS、MCRで禁止のプログラム

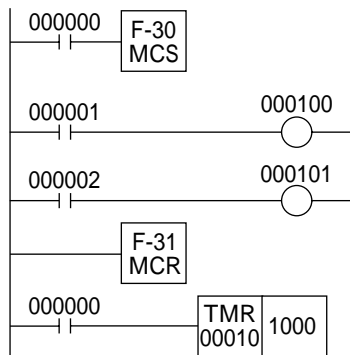
(参考)リレー盤の場合



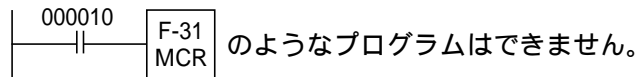
命 令	
STR	000000
F-30	
OUT	000100
STR	000001
OUT	000101
STR	000002
OUT	000102
F-31	



右記のようにプログラムしてください。

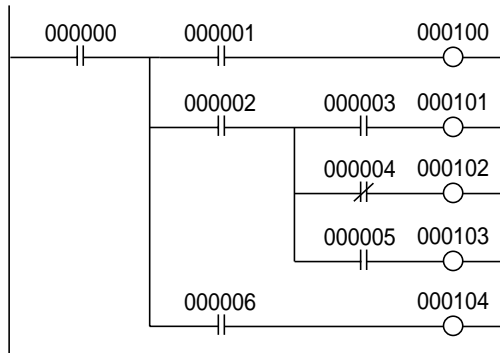


F-31(MCR)は無条件命令です。



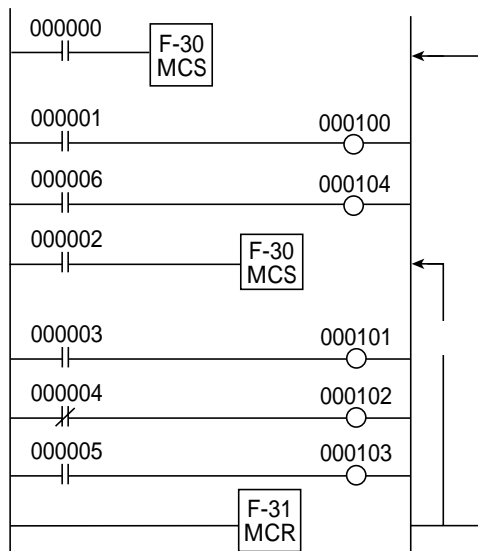
MCS、MCRの間にさらにMCSを使用できます。

・リレー盤の場合



・左記のリレー盤のラダー図は、MCS、MSRを用いて下記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の入換えを要する場合があります。(印部)

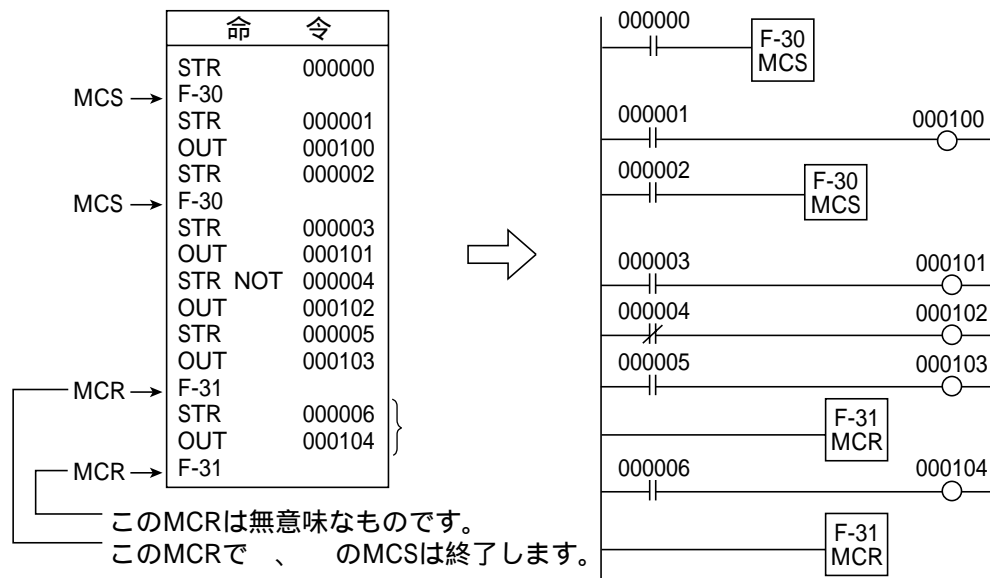
・MCS、MCRを使用した場合



命 令	
STR	000000
F-30	
STR	000001
OUT	000100
STR	000006
OUT	000104
STR	000002
F-30	
STR	000003
OUT	000101
STR NOT	000004
OUT	000102
STR	000005
OUT	000103
F-31	

・F-31(MCR)は、それ以前のF-30(MCS)...左記の場合、...の終了を示します。

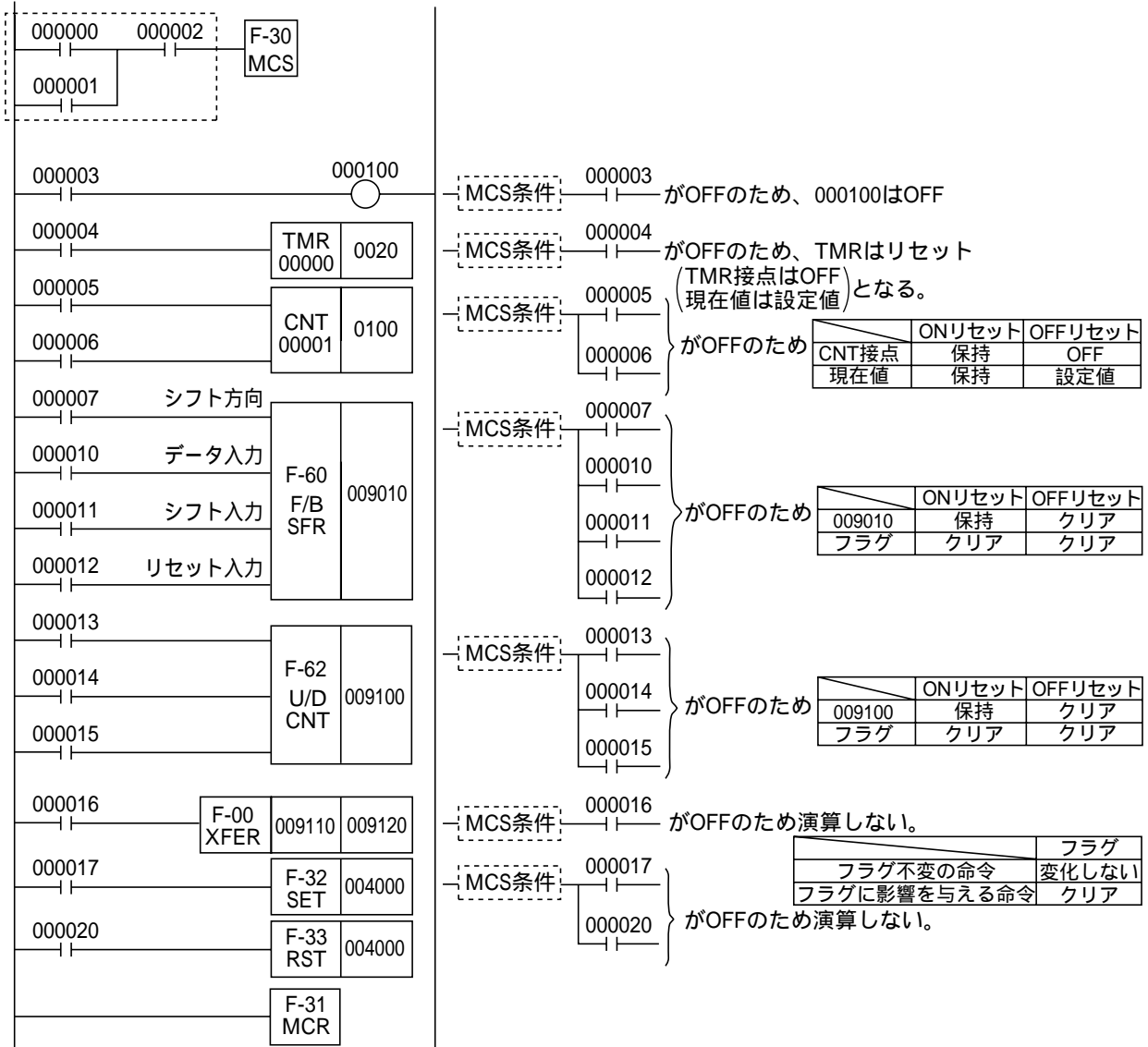
次のようにプログラムすると、初期の回路にはなりません。



このMCRは無意味なものです。
このMCRで、 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。このようなネ스팅を行う場合は、F-23(MCRN)「マスターコントロールネ스팅リセット」命令をF-31(MCR)の代わりに使用してください。

MCSの条件(点線内)がOFFのとき、MCSとMCRの間にある命令は次のように処理します。



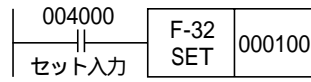
CNT、F-60、F-62の各命令は、システムメモリ#0202でリセット条件を「ONリセット」または「OFFリセット」に設定できます。「OFFリセット」に設定時、MCSによりリセットします。

**F-32
SET**

セットコイル

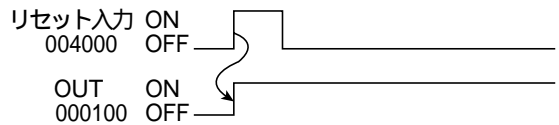
シンボル		
機能	セット入力ONすると、F-32で指定するOUTをONする。	
演算内容	F-32で指定するOUTをON	
OUT	使用範囲K	
演算条件	セット入力ONのとき (OFF ONの変化時に限定されない)	
演算内容 後	OUT	ON
	フラグ	不変

[使用例]



命 令	
STR	004000
F-32	000100

セット入力004000がONのとき、OUT000100がONします。
ONしたOUT000100は、セット入力004000がOFFしてもONを保持します。
セット入力004000がOFFのとき、OUT000100の状態は変化しません。



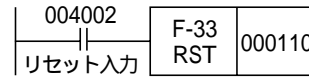
- ・本命令がF-30(MCS)命令の中にあるとき、F-32でONしたOUTは、F-30命令がOFFしてもONを保持します。
- ・本命令を使用すると、1個のOUTを複数の回路上で制御できます。
- ・本命令で指定するOUTがキープ指定領域内のときは、復電後も停電前の状態を保持します。また、指定するOUTがキープ指定領域外のときは、復電時にリセットします。
- ・本命令で指定するOUTがJW300停止時に出力保持する領域内のときは、JW300停止時に停止前の状態を保持します。また、指定するOUTがJW300停止時に出力保持する領域外のときは、JW300停止時にリセットされます。システムメモリ#0232、#0233、#0252、#0253参照
- ・本命令は、F-33(RST)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

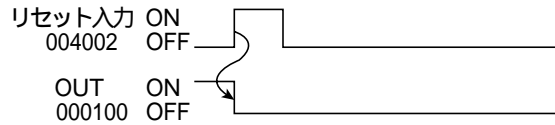
シンボル	— F-33 RST OUT
機能	リセット入力ONすると、F-33で指定するOUTをOFFする。
演算内容	F-33で指定するOUTをOFF
OUT	使用範囲K
演算条件	リセット入力ONのとき (OFF ONの変化時に限定されない)
演算内容	OUT OFF
後内容	フラグ 不変

[使用例]



命 令	
STR	004002
F-33	000110

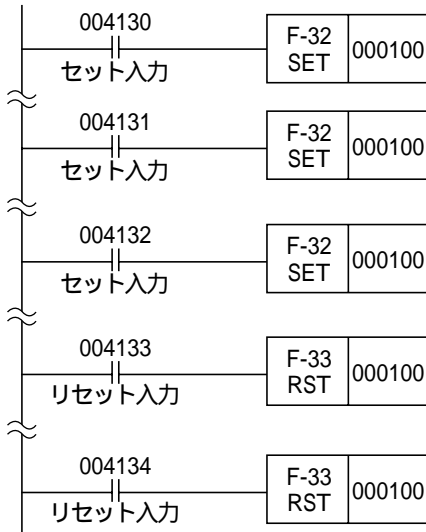
リセット入力004002がONのとき、OUT000110がOFFします。
OFFしたOUT000110は、リセット入力004002がOFFしてもOFFを保持します。
リセット入力004002がOFFのとき、OUT000110の状態は変化しません。



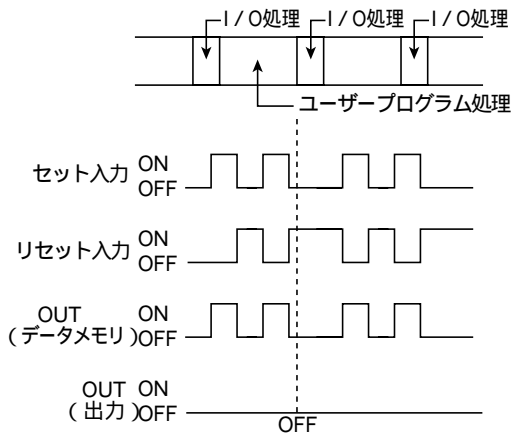
- ・本命令で指定するOUTがキープ指定領域内のとき、復電後も停電前の状態を保持します。また、指定するOUTがキープ指定領域外のとき、復電時にリセットします。
- ・本命令で指定するOUTがJW300停止時に出力保持する領域内のとき、停止前の状態を保持します。また、指定するOUTがJW300停止時に出力保持する領域外のとき、停止時にリセットします。

システムメモリ#0232、#0233、#0252、#0253参照

- ・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。



- ・セット入力とリセット入力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を繰り返します。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またはOFF)を出力します。

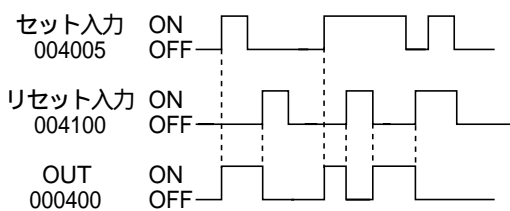
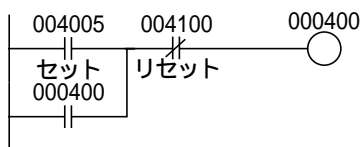


、 、 ではモニタ不可

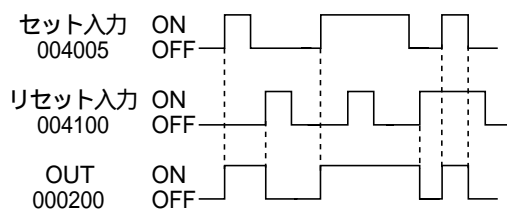
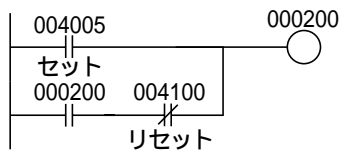
また、ユーザープログラム処理中にデータメモリが複数回ON/OFFしても、I/O処理直前の結果のみモニタできます。

・ F-3ㄨ SET 命令とF-3ㄨ RST 命令をペアで使用すると、自己保持回路等を簡略化できます。

リセット優先自己保持回路



セット優先自己保持回路

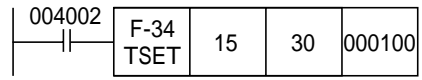


**F-34
TSET**

時計の現在値との比較(指定リレーのセット)

シンボル	F-34 TSET			n1	n2	BIT	[使用例]	<table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>004002</td> </tr> <tr> <td>F-34</td> <td>15 30 000100</td> </tr> </tbody> </table>	命 令		STR	004002	F-34	15 30 000100
命 令														
STR	004002													
F-34	15 30 000100													
機 能	定数n1(時)、n2(分)と時計の現在値を比較し、一致すると指定したBIT(リレー)をセット(ON)する。													
演算内容	n1, n2 < = > 時計の現在値を比較し、比較結果が一致でリレーをON。													
n1	使用範囲 00 ~ 23 (10進)													
n2	使用範囲 00 ~ 59 (10進)													
BIT	使用範囲 K													
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)													
演算後の内容	n1	不変												
	n2	不変												
	フラグ	不変												

[使用例]



入力条件004002がONのとき、設定した時計時刻と時計の現在値が一致すると、リレー000100がONします。ONしたリレー000100は、入力条件004002がOFFしてもONのまま保持します。時計の現在値と設定時刻が一致しないときは、リレー000100の状態は変化しません。

- ・設定時刻 ≥ 時計の現在値のとき、指定リレー不変
- ・設定時刻 = 時計の現在値のとき、指定リレーON

F-34(TSET)命令がF-30(MCS)命令の中にあるとき、ONしたリレーは、F-30命令がOFFしても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)命令で指定するリレーは、キープ指定領域内のとき復電後も停電前の状態を保持し、指定領域以外のとき復電時にリセットします。

F-34(TSET)命令で指定するリレーは、JW300が停止時に出力保持を行う領域内のとき停止前の状態を保持し、領域以外のとき停止時にリセットします。

F-34(TSET)命令は、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)

シンボル	F-35 TRST n1 n2 BIT				[使用例]	<table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>004003</td> </tr> <tr> <td>F-35</td> <td>09 15 000110</td> </tr> </tbody> </table>	命 令		STR	004003	F-35	09 15 000110
命 令												
STR	004003											
F-35	09 15 000110											
機能	定数n1(時)、n2(分)と時計の現在値を比較し、一致すると指定したBIT(リレー)をリセット(OFF)する。											
演算内容	n1, n2 < = > 時計の現在値を比較し、比較結果が一致でリレーをリセット											
n1	使用範囲 00 ~ 23 (10進)											
n2	使用範囲 00 ~ 59 (10進)											
BIT	使用範囲 K											
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)											
演算後の内容	n1	不変										
	n2	不変										
	フラグ	不変										

入力条件004003がONのとき、設定した時計時刻と時計の現在値が一致すると、リレー000110がOFFします。OFFしたリレー000110は、入力条件004003がOFFしてもOFFのまま保持します。時計の現在値と設定時刻が一致しないときは、リレー000110の状態は変化しません。

- ・ 設定時刻 ≥ 時計の現在値のとき、指定リレー不変
- ・ 設定時刻 = 時計の現在値のとき、指定リレーOFF

F-35(TRST)命令で指定するリレーは、キープ指定領域内のとき復電後も停電前の状態を保持し、指定領域以外のとき復電時にリセットします。

F-35(TRST)命令で指定するリレーは、JW300が停止時に出力保持を行う領域内のとき停止前の状態を保持し、領域以外のとき停止時にリセットします。

F-35(TRST)命令は、F-34(TSET)命令とペアで使用してください。

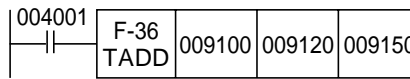
**F-36
TADD**

時計の加算

シンボル	F-36 TADD	S1	S2	D		
機能	レジスタS1~S1+2の内容とレジスタS2~S2+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	$(S1 \sim S1+2) + (S2 \sim S2+2) \rightarrow D \sim D+2$					
S1	使用範囲 E					
S2	使用範囲 E					
D	使用範囲 E					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	S1-S1+2	不変				
	S2-S2+2	不変				
	D~D+2	演算結果				
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		0	1	0	0	1
		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		

・24時間を越えて明日の時間になると、キャリーフラグ(007356)がONします。

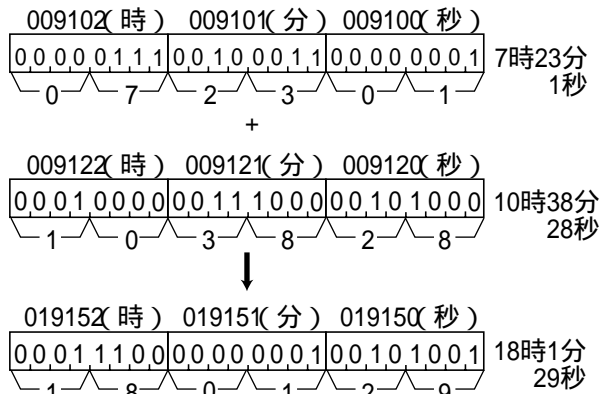
[使用例]



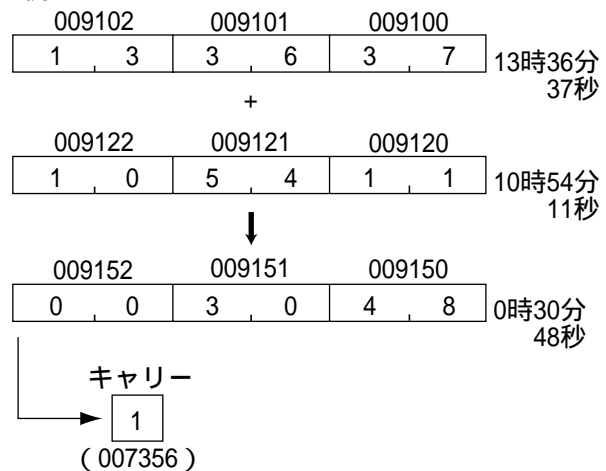
命 令	
STR	004001
F-36	009100 009120 009150

入力条件004001がOFF ONの変化時に、レジスタ009100~009102の内容(BCD 6桁)とレジスタ009120~009122の内容(BCD 6桁)を加算して、レジスタ009150~009152に格納します。

例 1



例 2



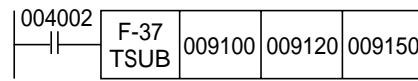
**F-37
TSUB**

時計の減算

シンボル	F-37 TSUB	S1	S2	D		
機能	レジスタS1~S1+2の内容からレジスタS2~S2+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	$(S1 \sim S1+2) - (S2 \sim S2+2) \rightarrow D \sim D+2$					
S1	使用範囲 E					
S2	使用範囲 E					
D	使用範囲 E					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	S1-S1+2	不変				
	S2-S2+2	不変				
	D~D+2	演算結果				
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		0	1	0	0	1
000001 ~235959		0	0	0	1	
負の数値		0	1	0	0	
時刻以外	0	0	1	0		

・ 0時間を越えて昨日の時間になると、キャリーフラグ(007356)がONします。

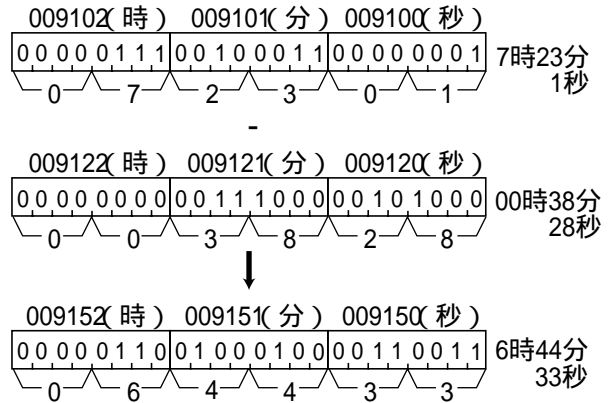
[使用例]



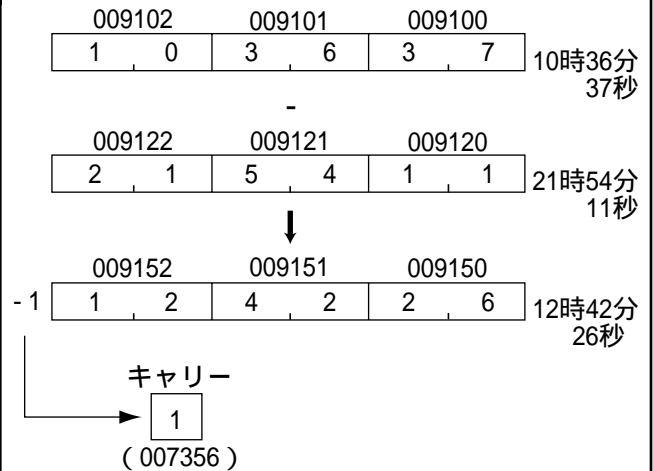
命 令	
STR	004002
F-37	009100 009120 009150

入力条件004002がOFF ONの変化時に、レジスタ009100~009102の内容(BCD 6桁)とレジスタ009120~009122の内容(BCD 6桁)を減算して、レジスタ009150~009152に格納します。

例 1



例 2

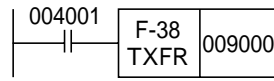


**F-38
TXFR**

時計現在値の転送

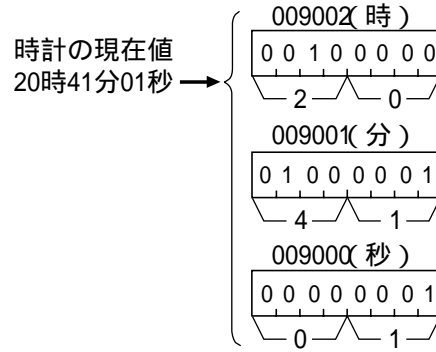
シンボル	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">F-38</td> <td style="text-align: center;">n</td> <td style="text-align: center;">D</td> </tr> <tr> <td style="text-align: center;">TXFR</td> <td></td> <td></td> </tr> </table>	F-38	n	D	TXFR		
F-38	n	D					
TXFR							
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	使用範囲 E						
演算条件	入力信号の立上り(OFF ON)						
演算後の内容	D~D+2	時計データ					
	フラグ	不変					

[使用例]



命 令	
STR	004001
F-38	009000

入力条件004001がOFF ONの変化時に、時計の現在値をレジスタ009000、009001、009002に転送します。



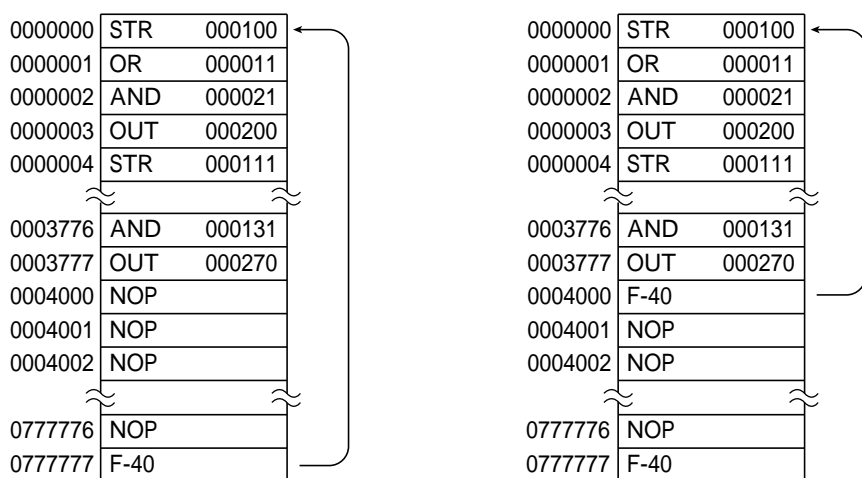
**F-40
END**

**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命令はプログラム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まれます。よって、下記 (1)(2)の場合を除き、特に書き込む必要はありません。

(1) スキャンタイムを速くする場合

- ・スキャンタイムは、ユーザープログラム処理時間と入出力処理時間などの合計です。ユーザープログラム処理時間は、各ブロックのプログラムアドレス0000000から END命令までの全命令の処理時間の合計です。
- ・プログラムメモリのクリアで自動的に書き込まれるEND命令の位置は、コントロールユニットが JW-362CU のとき、メインブロックの0777777(256K語目)となります。
- ・設計完了したラダー図をサポートツールで書き込んだとき、最終アドレスが例としてメインブロックの0003777(2048語目)とすると、0004000 ~ 0777776にNOP命令、0777777にEND命令が存在し、このNOPの処理時間を空費します。
少しでも演算時間を短縮するには、メインブロックの0004000にF-40を書き込んで、以下のNOP命令を処理せずに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(0777777)のみ (b) 0004000にF-40(END)を書込

複数ブロックを使用している場合は、各ブロックで同様にF-40を書き込んでください。

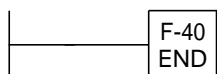
(2) 試運転でプログラムを部分的に実行させる場合

シーケンス動作の区切毎にF-40を挿入することで、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、複数個のF-40が存在する場合があります。

この場合、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)命令は無条件命令です。



F-40は優先度が一番高い命令ですが、「F-141(JMP)とF-140(LABEL)間」または「F-140(CALL)とF-140(LABEL)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。

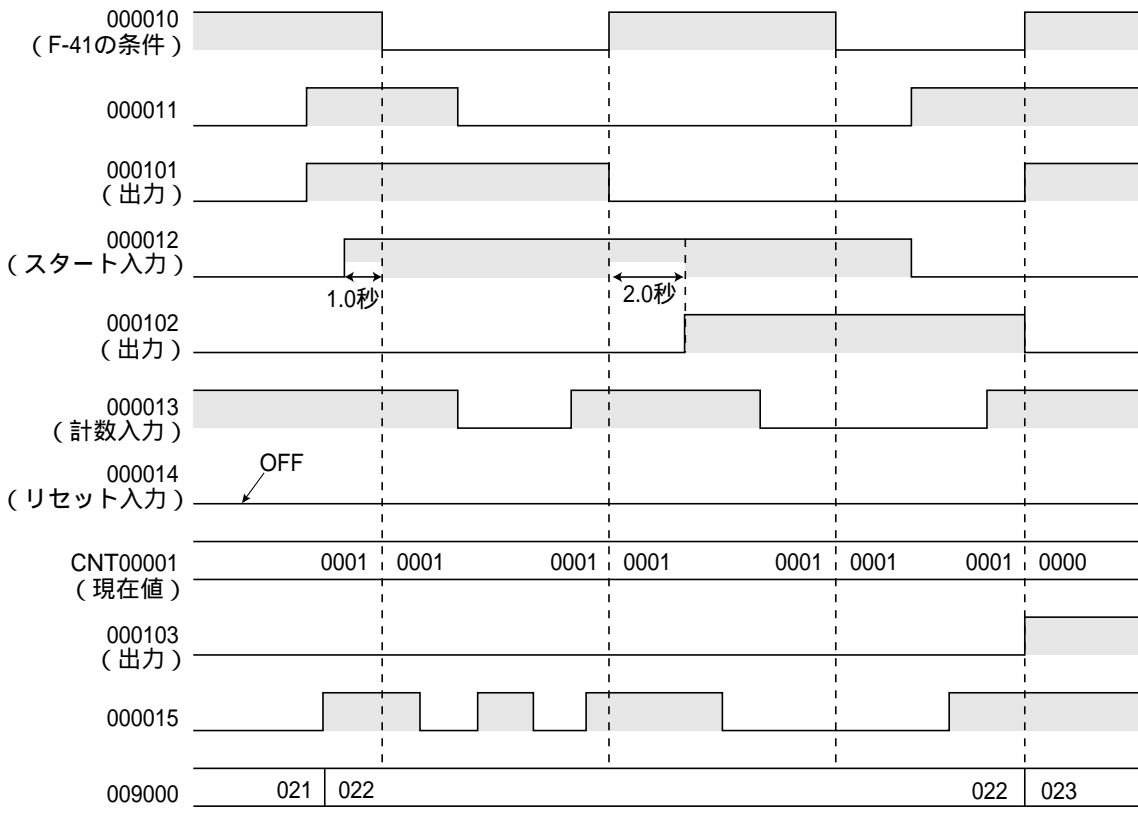
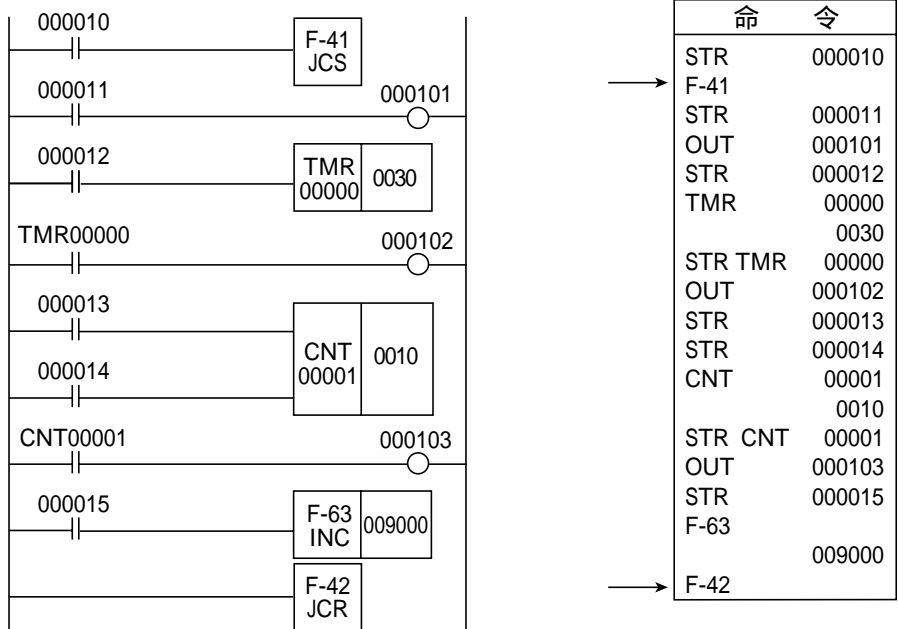
**F-41
JCS**

**ジャンプコントロールセット
(Jump Control Set)**

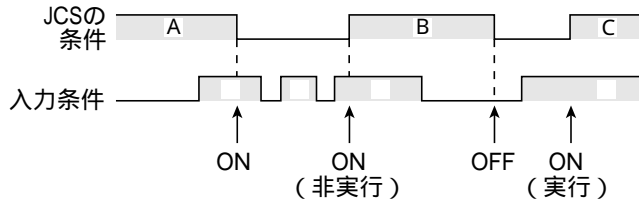
**F-42
JCR**

**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のとき、F-42(JCR)までにある全ての命令(END命令を除く)を実行しません。よってOUT命令、TMR・CNT・MD命令、応用命令等、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時の状態を保持します。



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願います。



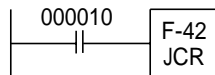
- ・ の立上りでは、JCSの条件ONのため、演算します。
- ・ の立上りでは、JCSの条件OFFのため、演算しません。
- ・ の立上りでは、JCSの条件OFFのため、演算しません。
- ・ がONの間にJCSの条件がONとなりますが、AのJCSの条件がON OFFとなる時の入力条件がON、BのJCSの条件がOFF ONとなる時の入力条件もONのため、入力条件がOFF ONに変化したとは見なさず演算しません。
- ・ の立上りでは、JCSの条件がOFFのため、演算しません。
- ・ がONの間にJCSの条件がONとなります。BのJCSの条件がON OFFとなる時の入力条件がOFF、CのJCSの条件がOFF ONとなる時の入力条件はONと変化しているため、CのJCS条件がOFF ONに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、ユーザプログラムの演算は終了し、次のスキャンサイクル(または、次のブロック演算)に移ります

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。

上記のようなネスティングを行う場合は、F-24(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)に換えて使用してください。

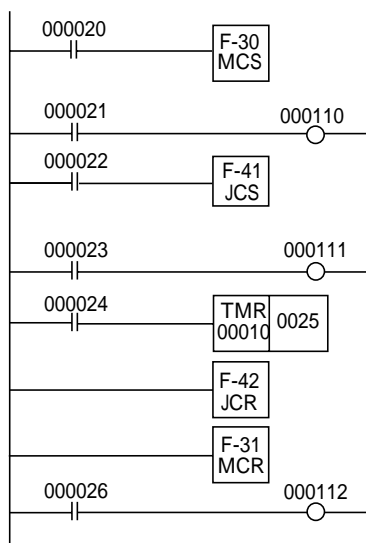
F-42(JCR)は無条件命令です。



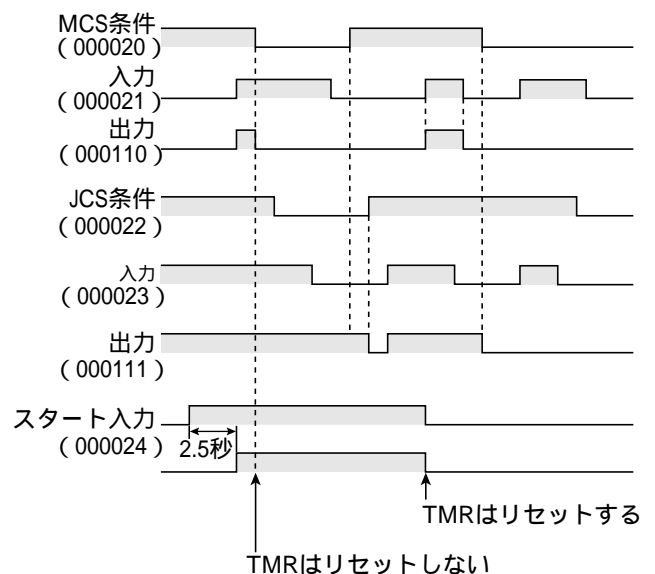
のようにはプログラムできません。

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-30(MCS)とF-31(MCR)の間に、F-41(JCS)、F-42(JCR)を入れた構造でプログラムできます。ただし、MCSの条件がOFFすると、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。



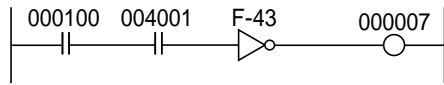
命 令	
STR	000020
F-30	
STR	000021
OUT	000110
STR	000022
F-41	
STR	000023
OUT	000111
STR	000024
TMR	00010
	0025
F-42	
F-31	
STR	000026
OUT	000112



**F-43
CPL**

**ビット反転
(ComPLement)**

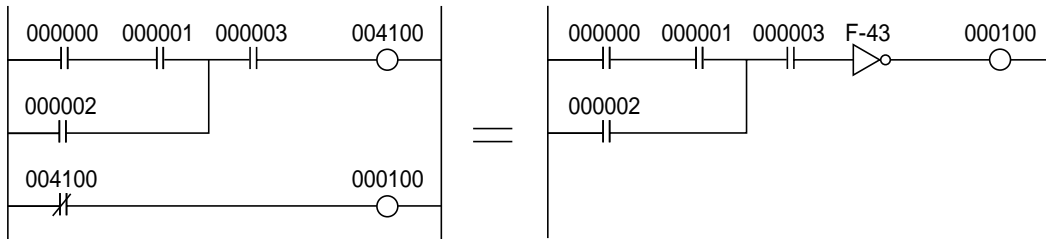
F-43は直前のACC(アキュムレータ)の内容を反転する命令です。



命 令	
STR	000100
AND	004001
F-43	
OUT	000007

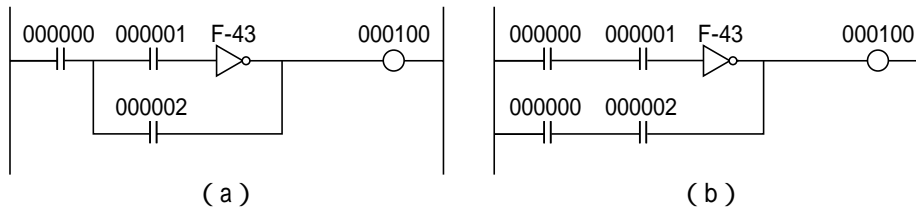


- ・ STR命令からF-43命令までの演算結果を反転し、出力リレー000007に出力します。
- ・ F-43を使用すると補助接点を使うことなく、反転出力が得られます。



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願います。

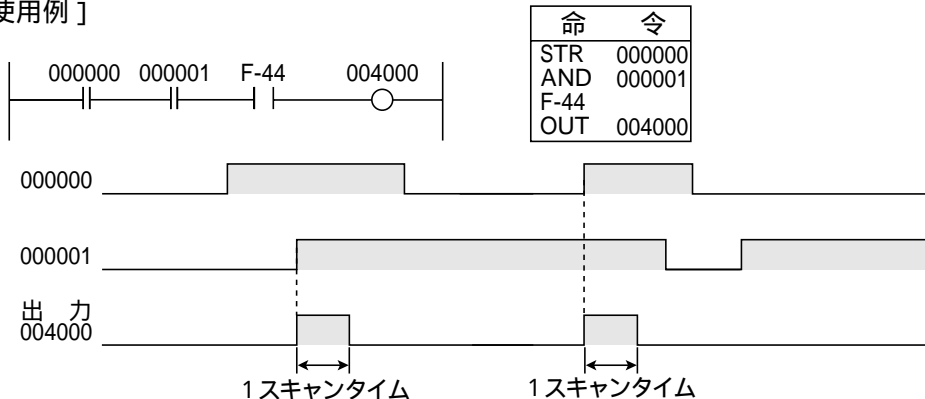


F-44

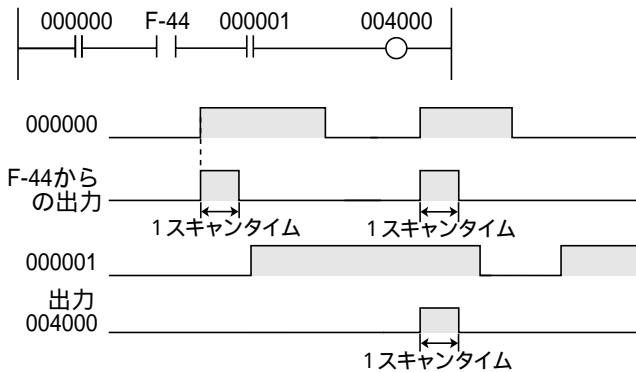
ON時微分

F-44命令の直前のACC(アキュムレータ)の状態がOFF ONの変化時に、1スキャンタイムのパルスを発生します。

[使用例]

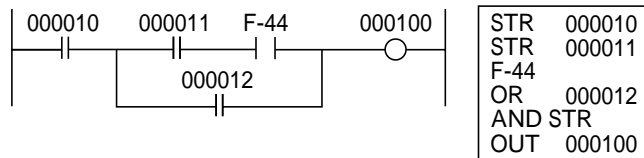


上記のラダー図でF-44のプログラム順序を変えると、結果が変わりますので注意願います。(F-45も同様)

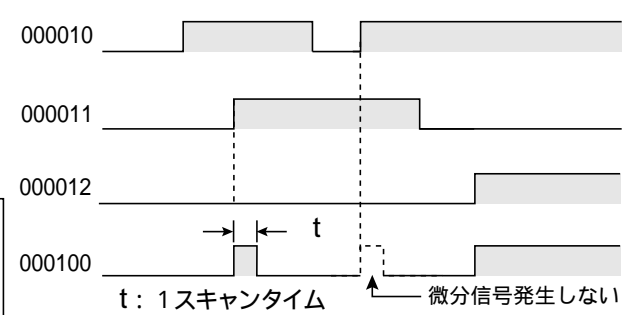


F-44命令の条件は、1接点でも複数の接点でも構いません。

下例の場合、AND STR命令で000010とのANDを演算するため、000011がONのとき000010がOFF ONしても微分信号は発生しません。



	アキュムレータ ACC	スタックレジスタ S1
STR 000010	000010	
STR 000011	000011	000010
F-44	000011 F-44	000010
OR 000012	000011 F-44 000012	000010
AND STR	000010 000011 F-44 000012	
OUT 000100	000010 000011 F-44 000012	



000011がOFF ONに変化した
スキャンサイクルのみACCがON

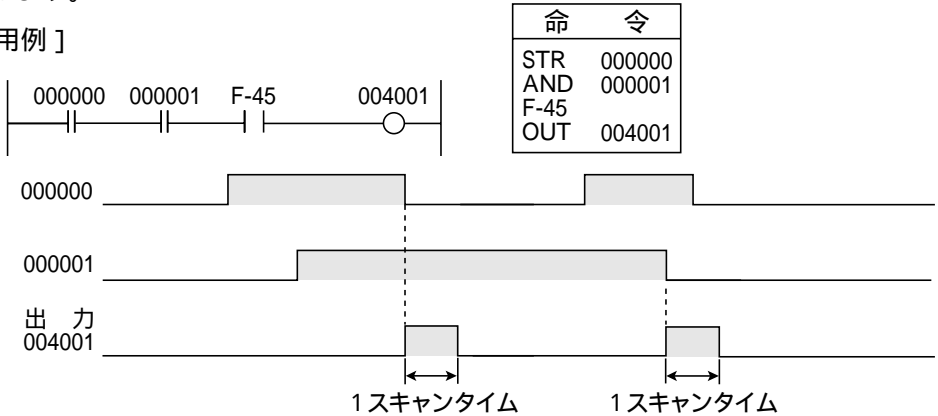
F-44命令は、F-47(レベル演算条件セット)とF-48(レベル演算条件リセット)の間に入れても、1スキャンしか演算しません。

F-45

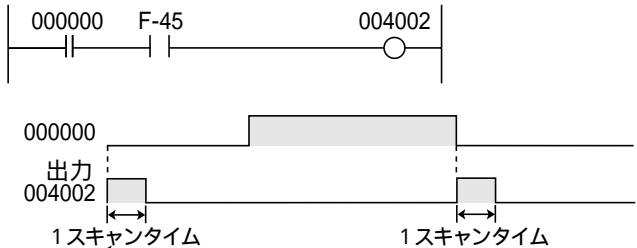
OFF時微分

F-45命令の直前のACC(アキュムレータ)の状態がON OFFの変化時に、1スキャンタイムのパルスを発生します。

[使用例]

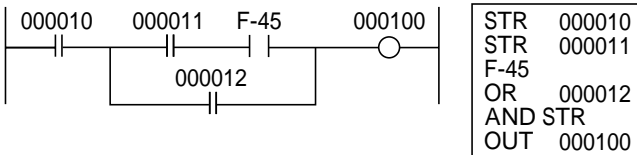


微分命令(F-44、F-45)は、プログラム中に何回でも使用できます。



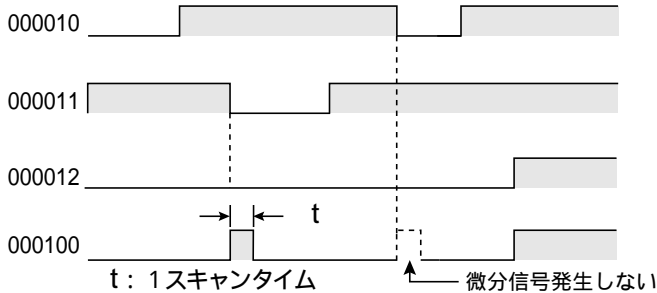
プログラム書込直後の運転開始時に、入力(000000)がOFF状態の場合、出力(004002)はONします。

下例の場合、AND STR命令で000010とのANDを演算するため、000011がONのとき000010がOFF ONしても微分信号は発生しません。



	アキュムレータ ACC	スタックレジスタ S1
STR 000010	000010	
STR 000011	000011	000010
F-45	000011 F-45	000010
OR 000012	000011 F-45 000012	000010
AND STR	000010 000011 F-45 000012	
OUT 000100	000010 000011 F-45 000012	

000011がOFF ONに変化した
スキャンサイクルのみACCがON



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。

**F-47
ONLS**

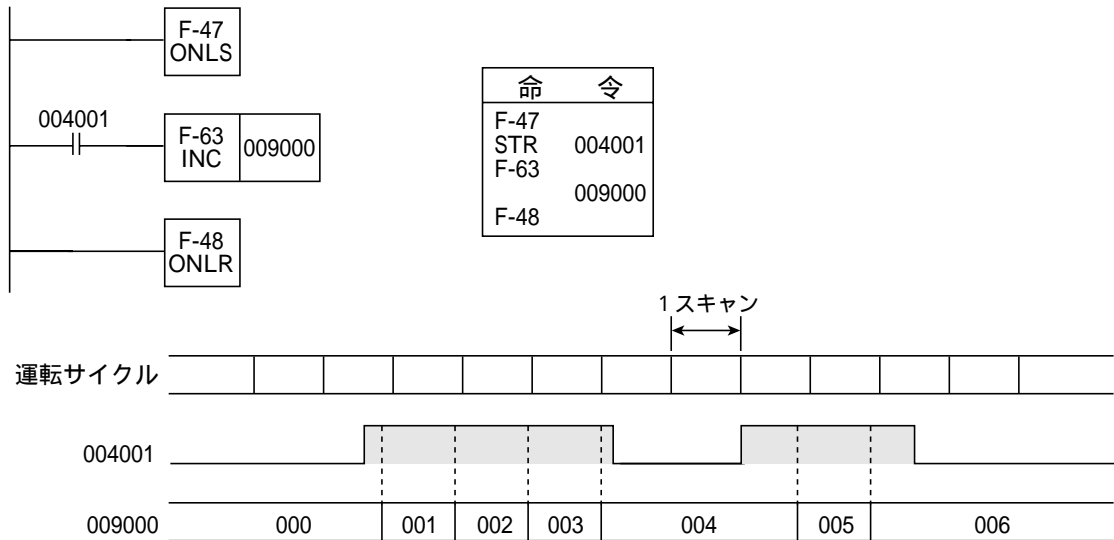
レベル演算条件セット
(ON Level Set)

**F-48
ONLR**

レベル演算条件リセット
(ON Level Reset)

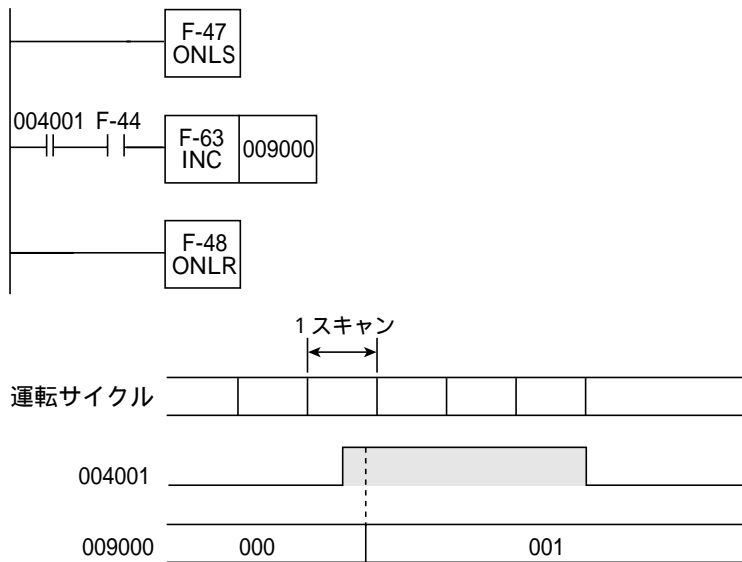
F-47(ONLS)とF-48(ONLR)の間の命令の立上がり演算条件を、レベル演算条件(ONで演算)に設定します。

【使用例】



F-47(ONLS)とF-48(ONLR)の間に、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、004001の立上り時の 1 スキャンだけ演算します。

【F-44の例】

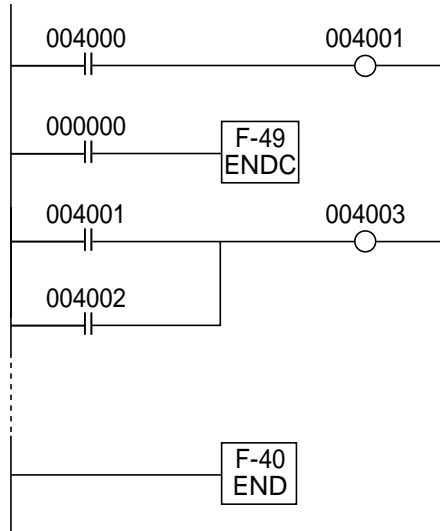


**F-49
ENDC**

条件エンド

F-49の条件がOFFのとき、シーケンス演算を終了します。

[使用例]



アドレス	命 令
.....
0000010	STR 004000
0000011	OUT 004001
0000012	STR 000000
0000013	F-49
0000014	STR 004001
0000015	OR 004002
0000016	OUT 004003
.....
0036777	F-40

- 入力条件000000がONのとき、F-40命令(アドレス0036777)までの命令を実行します。
- 入力条件000000がOFFのとき、アドレス0000014以降の命令を実行しません。

第 12 章 応用命令 (F-50 ~ F-79d)

F-50 4 16 デコーダ

シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="text-align: center;">F-50</td> <td style="width: 10px;"></td> <td style="text-align: center;">S</td> <td style="width: 10px;"></td> <td style="text-align: center;">D</td> </tr> <tr> <td style="text-align: center;">4</td> <td style="text-align: center;">16</td> <td></td> <td></td> <td></td> </tr> </table>	F-50		S		D	4	16			
F-50		S		D							
4	16										
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	使用範囲 A ・レジスタSの上位4ビットは演算上、無視します。										
D	使用範囲 B										
演算条件	入力信号の立上り(OFF ON)										
演算後の内容	S	不変									
	D	演算結果(0 ~ 7)									
	D+1	演算結果(8 ~ 15)									
	フラグ	不変									

[使用例]

命 令	
STR	004006
F-50	000000 009350

入力条件004006がOFF ONの変化時に、レジスタ000000の下位4ビットのデータをデコードし、レジスタ009350、009351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。

F-51 16 4 エンコーダ

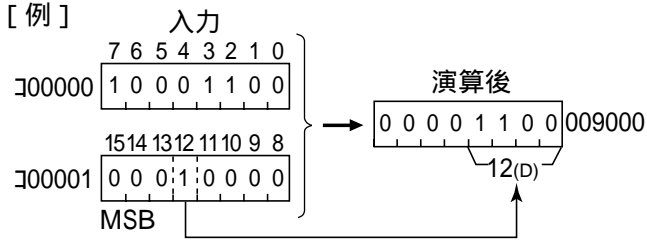
シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="text-align: center;">F-51</td> <td style="width: 10px;"></td> <td style="text-align: center;">S</td> <td style="width: 10px;"></td> <td style="text-align: center;">D</td> </tr> <tr> <td style="text-align: center;">16</td> <td style="text-align: center;">4</td> <td></td> <td></td> <td></td> </tr> </table>	F-51		S		D	16	4			
F-51		S		D							
16	4										
機能	レジスタS、S+1(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	使用範囲 B										
D	使用範囲 A										
演算条件	入力信号の立上り(OFF ON)										
演算後の内容	S、S+1	不変									
	D	演算結果									
	フラグ	不変									

[使用例]

命 令	
STR	004001
F-51	000000 009000

入力条件004001がOFF ONの変化時に、レジスタ000000、000001(2バイト)のデータをエンコードし、レジスタ009000に格納します。

- ・演算後、D(例では009000)の上位4ビットは常に0です。
- ・エンコーダの入力は、MSB側が優先されます。

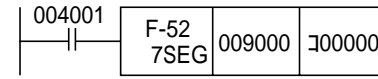


**F-52
7SEG**

7 SEGデコーダ

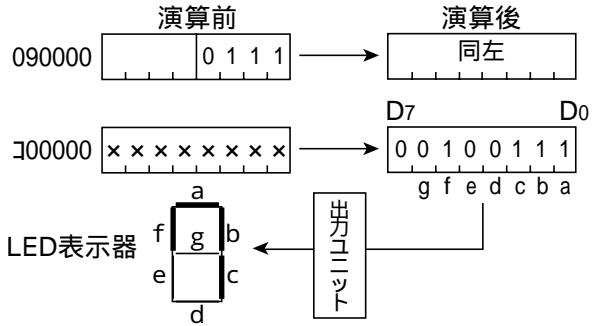
シンボル	F-52 7SEG	S	D
機能	レジスタSの下位4ビットのデータを、7セグメントの表示データにデコードする。		
演算内容	S → D		
S	使用範囲 A		
D	使用範囲 A ・レジスタDへの出力データD0～D6は、7セグメント表示器のa～gに対応しています。D7の出力は常に「0」です。		
演算条件	入力信号の立上り(OFF ON)		
演算後の内容	S	不変	
	D	演算結果 「7セグメントデコーダ表」参照	
フラグ	不変		

[使用例]



命 令	
STR	004001
F-52	009000 300000

入力条件004001がOFF ONの変化時に、レジスタ009000の内容(下位4ビット)を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」を参照願います。



7セグメントデコーダ表

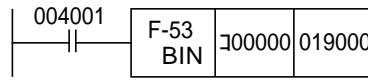
入力データ	出力データ	表示出力
	g f e d c b a	
0 0 0 0 0 0 0 0	0 0 1 1 1 1 1 1	0
0 0 0 0 0 0 0 1	0 0 0 0 0 1 1 0	1
0 0 0 0 0 0 1 0	0 1 0 1 1 0 1 1	2
0 0 0 0 0 0 1 1	0 1 0 0 1 1 1 1	3
0 0 0 0 0 1 0 0	0 1 1 0 0 1 1 0	4
0 0 0 0 0 1 0 1	0 1 1 0 1 1 0 1	5
0 0 0 0 0 1 1 0	0 1 1 1 1 1 0 1	6
0 0 0 0 0 1 1 1	0 0 1 0 0 1 1 1	7
0 0 0 0 1 0 0 0	0 1 1 1 1 1 1 1	8
0 0 0 0 1 0 0 1	0 1 1 0 1 1 1 1	9
0 0 0 0 1 0 1 0	0 1 1 1 0 1 1 1	A
0 0 0 0 1 0 1 1	0 1 1 1 1 1 0 0	b
0 0 0 0 1 1 0 0	0 0 1 1 1 0 0 1	c
0 0 0 0 1 1 0 1	0 1 0 1 1 1 1 0	d
0 0 0 0 1 1 1 0	0 1 1 1 1 0 0 1	E
0 0 0 0 1 1 1 1	0 1 1 1 0 0 0 1	F

**F-53
BIN**

BCD(4 桁) BIN(16ビット)変換

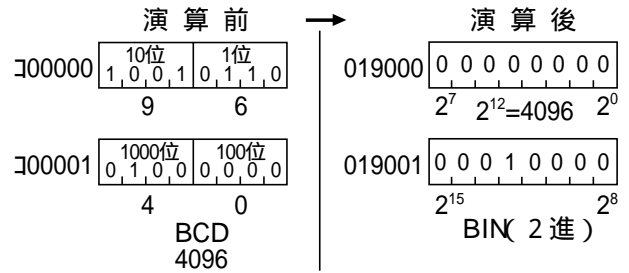
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-53 BIN</td><td>S</td><td>D</td></tr></table>					F-53 BIN	S	D
F-53 BIN	S	D						
機能	レジスタS、S+1(2 バイト)のBCD 4 桁データを 2 進数に変換し、レジスタD、D+1(2 バイト)に格納する。							
演算内容	S、S+1 → D、D+1							
S	使用範囲 B							
D	使用範囲 B							
演算条件	入力信号の立上り(OFF ON)							
演算後の内容	S、S+1	不変						
	D	演算結果 (0 ~ 255)	・レジスタS、S+1の内容がBCDコード以外のとき不変					
	D+1	演算結果 (256 ~ 9999)						
	フラグ	レジスタS、S+1の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354		
BCDコード		0	0	0	0			
	BCDコード以外	0	0	1	0			

[使用例]



命 令	
STR	004001
F-53	000000 019000

入力条件004001がOFF ONの変化時に、レジスタ001000、001001のBCD 4 桁データを、2 進数に変換し、レジスタ019000、019001(2 バイト)に格納します。



(注) F-53でプログラムを作成すると、モニタ時にF-03wで表示します。

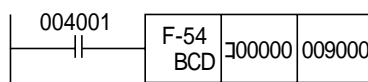
(類似命令) F-03、F-03w、F-03d、F-153

**F-54
BCD**

BIN(16ビット) BCD(6 桁)変換

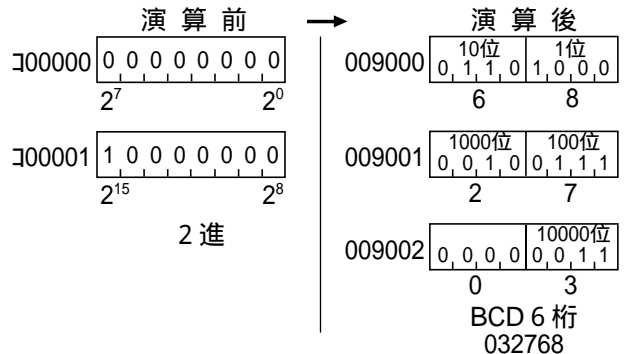
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-54 BCD</td><td>S</td><td>D</td></tr></table>					F-54 BCD	S	D
F-54 BCD	S	D						
機能	レジスタS、S+1の 2 バイト(2 進データ)を、BCD 6 桁に変換し、レジスタD、D+1、D+2の 3 バイトに格納する。							
演算内容	S、S+1 → D、D+1、D+2							
S	使用範囲 B							
D	使用範囲 E							
演算条件	入力信号の立上り(OFF ON)							
演算後の内容	S、S+1	不変						
	D	演算結果(1と10の位)						
	D+1	演算結果(100と1000の位)						
	D+2	演算結果(10000の位)						
フラグ	不変							

[使用例]



命 令	
STR	004001
F-54	000000 009000 009000

入力条件004001がOFF ONの変化時に、レジスタ000000と000001の 2 バイト(2 進データ)をBCD 6 桁に変換し、レジスタ009000 ~ 009002(3 バイト)に格納します。



(注) F-54でプログラムを作成すると、モニタ時にF-04wで表示します。

(類似命令) F-04、F-04w、F-04d、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	<table border="1"> <tr> <td>F-55 SWAP</td> <td>S</td> <td>D</td> </tr> </table>		F-55 SWAP	S	D	<p>[使用例]</p> <table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>004001</td> </tr> <tr> <td>F-55</td> <td>009000 009001</td> </tr> </tbody> </table> <p>入力条件004001がOFF ONの変化時に、レジスタ009000の上下4ビットずつを交換し、レジスタ009001に格納します。 なお、レジスタ009000の内容は不変です。</p> <p style="text-align: center;">演算前 → 演算後</p> <table border="1"> <tr> <td>009000</td> <td>0 0 0 0 1 1 1 1</td> <td>0 0 0 0 1 1 1 1</td> </tr> <tr> <td>009001</td> <td>x x x x x x x x</td> <td>1 1 1 1 0 0 0 0</td> </tr> </table>	命 令		STR	004001	F-55	009000 009001	009000	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1	009001	x x x x x x x x	1 1 1 1 0 0 0 0
F-55 SWAP	S	D																
命 令																		
STR	004001																	
F-55	009000 009001																	
009000	0 0 0 0 1 1 1 1	0 0 0 0 1 1 1 1																
009001	x x x x x x x x	1 1 1 1 0 0 0 0																
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	使用範囲 A																	
D	使用範囲 A																	
演算条件	入力信号の立上り(OFF ON)																	
演算後の内容	S	不変																
	D	演算結果																
	フラグ	不変																

参考 F-55命令は次のようなときに有効です。

F-52命令(7 SEG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。多桁を表示するとき、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。

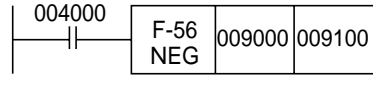
(類似命令) F-175

**F-56
NEG**

1 バイトデータの10の補数

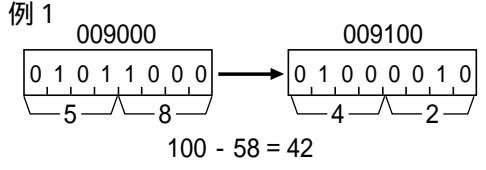
シンボル	— F-56 NEG S D				
機能	レジスタSの内容(1バイトデータ)を2桁のBCDコードと見なし、その値の10の補数を取り、レジスタDに格納する。				
演算内容	100 - S → D				
S	使用範囲 A				
D	使用範囲 A				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S	不変			
	D	演算結果 ・レジスタSの内容がBCDコード以外 のとき不変			
フラグ	レジスタSの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	BCDコード	0	0	0	0
	BCDコード以外			1	

[使用例]

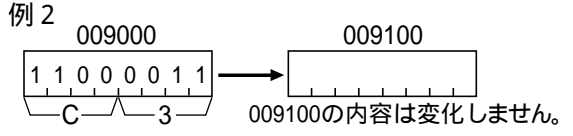


命 令	
STR	004000
F-56	009000
	009100

入力条件004000がOFF ONの変化時に、レジスタ009000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009100に格納します。009000の内容がBCDコード以外するとき、009100の内容は変化せず、エラーフラグがON(1)します。



ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
0	0	0	0



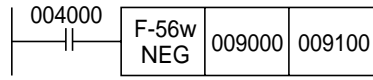
ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
0	0	1	0

**F-56w
NEG**

1 ワードデータの10の補数

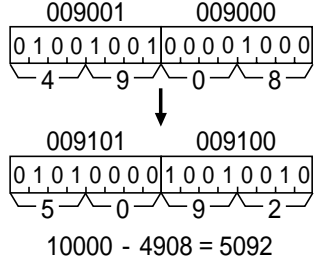
シンボル	— F-56w NEG S D				
機能	レジスタS、S+1の内容(1ワードデータ)を4桁のBCDコードと見なし、その値の10の補数を取り、レジスタD、D+1に格納する。				
演算内容	10000 - (S, S+1) → D, D+1				
S	使用範囲 B				
D	使用範囲 B				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S, S+1	不変			
	D	演算結果(下位)	・レジスタS、S+1の内容がBCDコード以外 のとき不変。		
D+1	演算結果(上位)				
フラグ	レジスタS、S+1の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	BCDコード	0	0	0	0
	BCDコード以外			1	

[使用例]



命 令	
STR	004000
F-56w	009000
	009100

入力条件004000がOFF ONの変化時に、レジスタ009000、009001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009100、009101に格納します。



ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
0	0	0	0

S, Dには必ず偶数アドレスを設定してください。(009003等は禁止)

**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	[使用例]											
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。				<table border="1"> <tr> <th colspan="2">命 令</th> </tr> <tr> <td>STR</td> <td>004000</td> </tr> <tr> <td>F-56d</td> <td>009000</td> </tr> <tr> <td></td> <td>009100</td> </tr> </table>		命 令		STR	004000	F-56d	009000		009100					
命 令																			
STR	004000																		
F-56d	009000																		
	009100																		
演算内容	10000000 - (S ~ S+3) → D ~ D+3				<p>入力条件004000がOFF ONの変化時に、レジスタ009000 ~ 009003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009100 ~ 009103に格納します。</p>														
S	使用範囲 B				<table border="1"> <tr> <td>009003</td> <td>009002</td> <td>009001</td> <td>009000</td> </tr> <tr> <td>0 1 0 0 1 0 0 1</td> <td>0 0 0 0 1 0 0 0</td> <td>0 1 0 0 1 0 0 1</td> <td>0 0 0 0 1 0 0 0</td> </tr> <tr> <td>4</td> <td>9</td> <td>0</td> <td>8</td> </tr> </table>		009003	009002	009001	009000	0 1 0 0 1 0 0 1	0 0 0 0 1 0 0 0	0 1 0 0 1 0 0 1	0 0 0 0 1 0 0 0	4	9	0	8	
009003	009002	009001	009000																
0 1 0 0 1 0 0 1	0 0 0 0 1 0 0 0	0 1 0 0 1 0 0 1	0 0 0 0 1 0 0 0																
4	9	0	8																
D	使用範囲 B				<table border="1"> <tr> <td>009103</td> <td>009102</td> <td>009101</td> <td>009100</td> </tr> <tr> <td>0 1 0 1 0 0 0 0</td> <td>1 0 0 1 0 0 0 1</td> <td>0 1 0 1 0 0 0 0</td> <td>1 0 0 1 0 0 1 0</td> </tr> <tr> <td>5</td> <td>0</td> <td>9</td> <td>1</td> </tr> </table>		009103	009102	009101	009100	0 1 0 1 0 0 0 0	1 0 0 1 0 0 0 1	0 1 0 1 0 0 0 0	1 0 0 1 0 0 1 0	5	0	9	1	
009103	009102	009101	009100																
0 1 0 1 0 0 0 0	1 0 0 1 0 0 0 1	0 1 0 1 0 0 0 0	1 0 0 1 0 0 1 0																
5	0	9	1																
演算条件	入力信号の立上り(OFF ON)				<p>10000000 - 49084908 = 50915092</p>														
演算後の内容	S ~ S+3	不変				<table border="1"> <tr> <td>ゼロ</td> <td>キャリー</td> <td>エラー</td> <td>ノンキャリー</td> </tr> <tr> <td>007357</td> <td>007356</td> <td>007355</td> <td>007354</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>		ゼロ	キャリー	エラー	ノンキャリー	007357	007356	007355	007354	0	0	0	0
	ゼロ	キャリー	エラー	ノンキャリー															
	007357	007356	007355	007354															
0	0	0	0																
D ~ D+3	演算結果 ・レジスタS ~ S+3の内容がBCDコード以外るとき不変。				<table border="1"> <tr> <td>ゼロ</td> <td>キャリー</td> <td>エラー</td> <td>ノンキャリー</td> </tr> <tr> <td>007357</td> <td>007356</td> <td>007355</td> <td>007354</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>		ゼロ	キャリー	エラー	ノンキャリー	007357	007356	007355	007354	0	0	0	0	
ゼロ	キャリー	エラー	ノンキャリー																
007357	007356	007355	007354																
0	0	0	0																
フラグ	レジスタS ~ S+3の内容	ゼロ	キャリー	エラー	ノンキャリー														
	BCDコード	0	0	0	0														
	BCDコード以外			1															

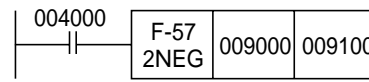
S, Dには必ず偶数アドレスを設定してください。(009003等は禁止)

**F-57
2NEG**

1バイトデータの2の補数

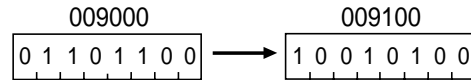
シンボル	<table border="1"><tr><td>F-57</td><td>S</td><td>D</td></tr><tr><td>2NEG</td><td></td><td></td></tr></table>	F-57	S	D	2NEG		
F-57	S	D					
2NEG							
機能	レジスタSの内容(8ビットデータ)の2の補数を取り、レジスタDに格納する。						
演算内容	0 - S → D						
S	使用範囲A						
D	使用範囲A						
演算条件	入力信号の立上り(OFF ON)						
演算後の内容	S	不変					
	D	演算結果					
	フラグ	不変					

[使用例]



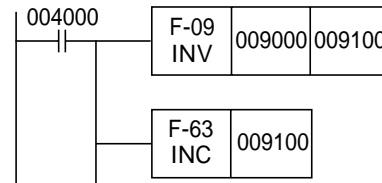
命 令	
STR	004000
F-57	009000
	009100

入力条件004000がOFF ONの変化時に、レジスタ009000の内容(8ビットデータ)の2の補数を取り、レジスタ009100に格納します。



- 2の補数の作り方
- すべてのビットを反転する。
(0なら1、1なら0とする)
 - 反転した数に+1する。

・上例の場合、下記命令と同じ動作となります。

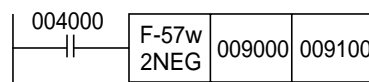


**F-57w
2NEG**

1ワードデータの2の補数

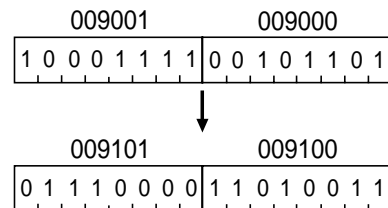
シンボル	<table border="1"><tr><td>F-57w</td><td>S</td><td>D</td></tr><tr><td>2NEG</td><td></td><td></td></tr></table>	F-57w	S	D	2NEG		
F-57w	S	D					
2NEG							
機能	レジスタS,S+1の内容(1ワードデータ)の2の補数を取り、レジスタD、D+1に格納する。						
演算内容	0 -(S,S+1) → D、D+1						
S	使用範囲B						
D	使用範囲B						
演算条件	入力信号の立上り(OFF ON)						
演算後の内容	S,S+1	不変					
	D	演算結果(下位)					
	D+1	演算結果(上位)					
	フラグ	不変					

[使用例]

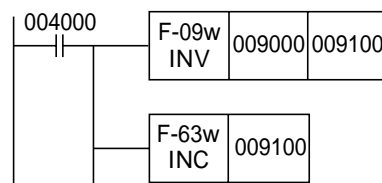


命 令	
STR	004000
F-57w	009000
	009100

入力条件004000がOFF ONの変化時に、レジスタ009000、009001の内容(16ビットデータ)の2の補数を取り、レジスタ009100、009101に格納します。



・上例の場合、下記命令と同じ動作となります。

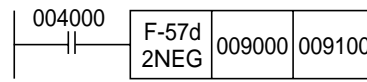


**F-57d
2NEG**

2ワードデータの2の補数

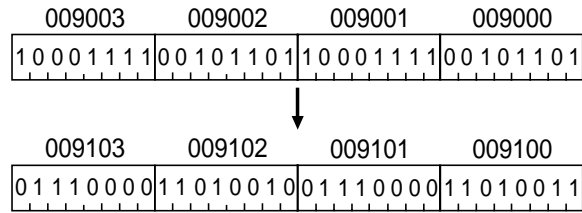
シンボル	F-57d 2NEG	S	D
機能	レジスタS~S+3の内容(2ワードデータ)の2の補数を取り、レジスタD~D+3に格納する。		
演算内容	$0 - (S \sim S+3) \rightarrow D \sim D+3$		
S	使用範囲 C		
D	使用範囲 C		
演算条件	入力信号の立上り(OFF ON)		
演算後の内容	S~S+3	不変	
	D~D+3	演算結果(D:下位、D+3:上位)	
	フラグ	不変	

[使用例]



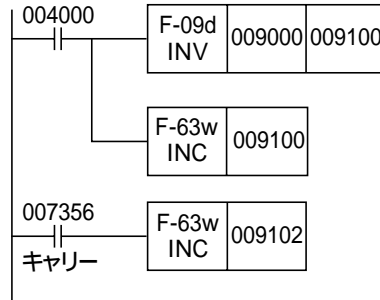
命 令	
STR	004000
F-57d	009000 009100

入力条件004000がOFF ONの変化時に、レジスタ009000~009003の内容(32ビットデータ)の2の補数を取り、レジスタ009100~009103に格納します。



S,Dには必ず偶数アドレスを設定してください。

・上例の場合、下記命令と同じ動作となります。

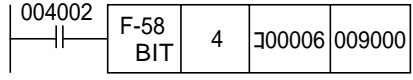


**F-58
BIT**

ONビット数の合計

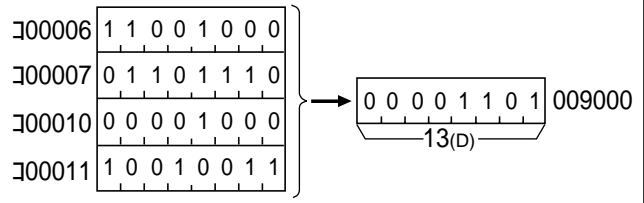
シンボル	F-58 BIT n S D		
機能	レジスタSを先頭とするnバイトのレジスタ中のONビット数を、レジスタDに格納する。		
演算内容	ONビット数 → D		
n	使用範囲 0 ~ 7 (0のとき8バイト)		
S	使用範囲 A		
D	使用範囲 A		
演算条件	入力信号の立上り(OFF ON)		
演算後の内容	S, S+1, ... S+n-1	不変	
	D	演算結果	
	フラグ	不変	

[使用例]



命 令	
STR	004002
F-58	4
	J00006
	009000

入力条件004002がOFF ONの変化時に、レジスタJ00006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009000に格納します。



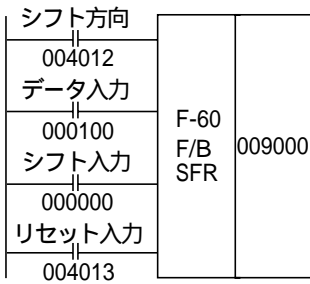
・ J00006 ~ J00011の32ビット中、13ビットがON。

**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		シフト方向指示入力 データ入力 シフト入力 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 に従って上位ビット、または下位ビットへシフトする。																
演算内容	レジスタD シフト方向指示入力 がONのとき <table border="1" style="display: inline-table; margin-left: 20px;"> <tr><td>キャリア</td><td>007356</td></tr> </table> MSB <table border="1" style="display: inline-table; width: 100px; height: 15px;"></table> LSB データ入力 シフト方向指示入力 がOFFのとき データ入力 MSB <table border="1" style="display: inline-table; width: 100px; height: 15px;"></table> LSB <table border="1" style="display: inline-table; margin-left: 20px;"> <tr><td>キャリア</td><td>007356</td></tr> </table>		キャリア	007356	キャリア	007356											
キャリア	007356																
キャリア	007356																
D	使用範囲 A																
演算条件	リセット入力 がOFFのとき、シフト入力 の立上り(OFF ON)でシフト																
演算後の内容	D	・リセット入力 がOFFのとき、演算結果 ・リセット入力 がONのとき、全ビットOFF															
	フラグ	<table border="1" style="width: 100%; text-align: center;"> <tr> <th>リセット入力</th> <th>ゼロ 007357</th> <th>キャリア 007356</th> <th>エラー 007355</th> <th>ノンキャリア 007354</th> </tr> <tr> <td>OFF</td> <td>0または1</td> <td>0または1</td> <td>0</td> <td>1または0</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	ゼロ 007357	キャリア 007356	エラー 007355	ノンキャリア 007354	OFF	0または1	0または1	0	1または0	ON	0	0		0
		リセット入力	ゼロ 007357	キャリア 007356	エラー 007355	ノンキャリア 007354											
OFF	0または1	0または1	0	1または0													
ON	0	0		0													

[使用例]



命 令	
STR	004012
STR	000100
STR	000000
STR	004013
F-60	009000

シフト入力000000がOFF ONの変化時に、シフト方向指示入力004012の状態により、次のようにシフトします。

- ・004012がONのとき

キャリア	007356	009000	000100
	<input type="checkbox"/>	MSB <table border="1" style="display: inline-table; width: 100px; height: 15px;"></table> LSB	<input type="checkbox"/>
- ・004012がOFFのとき

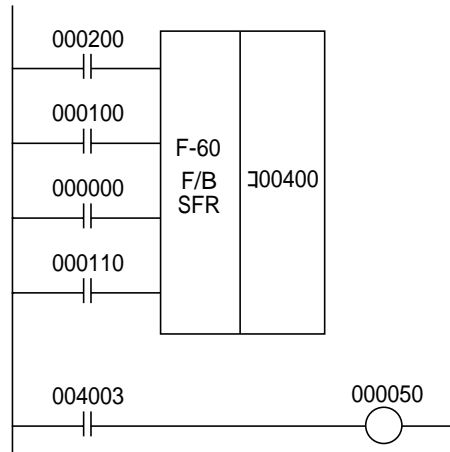
キャリア	000100	007356
	<input type="checkbox"/>	MSB <table border="1" style="display: inline-table; width: 100px; height: 15px;"></table> LSB

入力条件	009000(演算前)								009000(演算後)								ゼロ 007357	キャリア 007356	ノンキャリア 007354	
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0				
004012																				
000100																				
000000 \uparrow																				
004013																				
004012																				
000100																				
000000 \uparrow																				
004013																				
004012																				
000100																				
000000 \uparrow																				
004013																				
004012																				
000100																				
000000 \uparrow																				
004013																				
004013																				

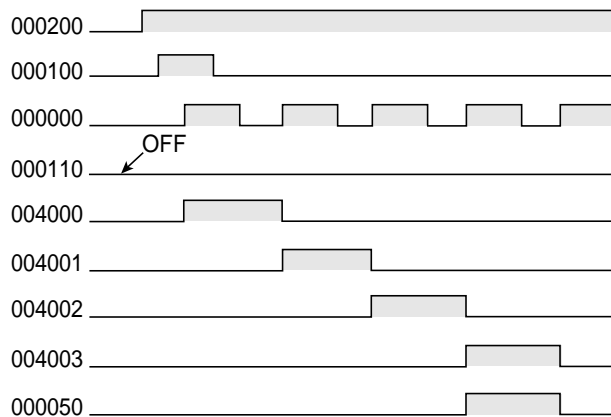
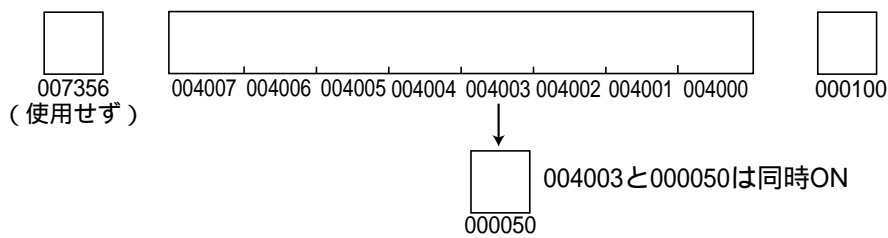
・エラーフラグ(007355)は常にOFFです。 : OFF : ON

・リセット入力 は、システムメモリ(#0202)にリセット条件を設定すると、「OFFでリセット」も可能です。
(類似命令) F-60w、F-60d、F-160、Fc160

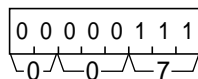
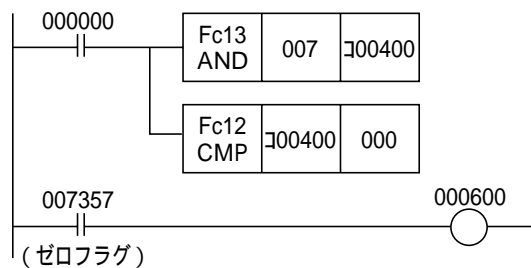
【参考】 Dにxxxxxの領域を使用すると、nビット(n < 8)のシフトレジスタを構成できます。



000200がONの場合



- 004004 ~ 004007にもデータをシフトします。
- ゼロフラグは004000 ~ 004007が全て0のとき1となります。004000 ~ 004002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。



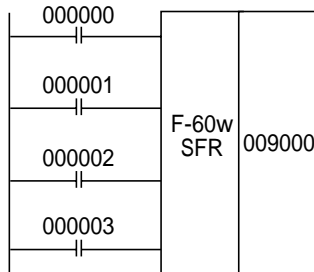
とANDすることで、004003 ~ 004007をマスク(すべてを0にする)しています。

**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															
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 に従って、上位ビットまた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 がONのとき</p> <table border="1"> <tr> <td>キャリアー 007356</td> <td>MSB</td> <td>レジスタD+1</td> <td>レジスタD</td> <td>LSB</td> <td>データ 入力</td> </tr> </table> <p>シフト方向指示入力がOFFのとき</p> <table border="1"> <tr> <td>データ 入力</td> <td>MSB</td> <td>レジスタD+1</td> <td>レジスタD</td> <td>LSB</td> <td>キャリアー 007356</td> </tr> </table>	キャリアー 007356	MSB	レジスタD+1	レジスタD	LSB	データ 入力	データ 入力	MSB	レジスタD+1	レジスタD	LSB	キャリアー 007356		
キャリアー 007356	MSB	レジスタD+1	レジスタD	LSB	データ 入力										
データ 入力	MSB	レジスタD+1	レジスタD	LSB	キャリアー 007356										
D	<p>使用範囲 B</p> <ul style="list-style-type: none"> ・Dには必ず偶数アドレスを設定してください。(019003等は禁止) 														
演算条件	リセット入力 がOFFのとき、シフト入力 の立上り(OFF ON)でシフト														
演算後の内容	D、D+1	<ul style="list-style-type: none"> ・リセット入力 がOFFのとき、演算結果 ・リセット入力 がONのとき、全ビットOFF 													
	フラグ	<table border="1"> <thead> <tr> <th>リセット入力</th> <th>ゼロ 007357</th> <th>キャリアー 007356</th> <th>エラー 007355</th> <th>ノンキャリアー 007354</th> </tr> </thead> <tbody> <tr> <td>OFF</td> <td>0または1</td> <td>0または1</td> <td rowspan="2">0</td> <td>1または0</td> </tr> <tr> <td>ON</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	リセット入力	ゼロ 007357	キャリアー 007356	エラー 007355	ノンキャリアー 007354	OFF	0または1	0または1	0	1または0	ON	0	0
リセット入力	ゼロ 007357	キャリアー 007356	エラー 007355	ノンキャリアー 007354											
OFF	0または1	0または1	0	1または0											
ON	0	0		0											

[使用例]

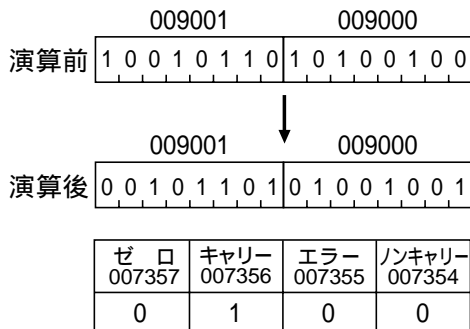


命 令	
STR	000000
STR	000001
STR	000002
STR	000003
F-60w	009000

入力条件

- 000000()ON ----- MSB方向へシフト
- 000001()ON ----- データ入力ON
- 000002()OFF ON ----- シフト指示
- 000003()OFF ----- リセット機能なし

演算結果は次のとおりです。



・リセット入力 は、システムメモリ(#0202)の設定で「OFFでリセット」も可能です。

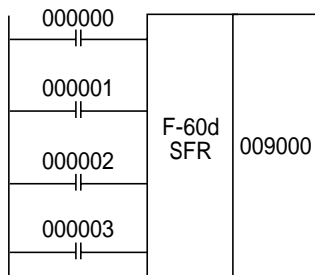
(類似命令) F-60、F-60d、F-160、Fc160

**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		シフト方向指示入力 データ入力 シフト入力 リセット入力																		
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 に従って、上位ビットまた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 がONのとき</p> <p>シフト方向指示入力 がOFFのとき</p>																			
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。(019003等は禁止)																			
演算条件	リセット入力 がOFFのとき、シフト入力 の立上り(OFF ON)でシフト																			
演算後の内容	<p>D~D+3</p> <ul style="list-style-type: none"> ・リセット入力 がOFFのとき、演算結果 ・リセット入力 がONのとき、全ビットOFF 																			
フラグ	<table border="1"> <tr> <td>リセット入力</td> <td>ゼロ 007357</td> <td>キャリー 007356</td> <td>エラー 007355</td> <td>ノンキャリー 007354</td> <td></td> </tr> <tr> <td>OFF</td> <td>0または1</td> <td>0または1</td> <td>0</td> <td>1または0</td> <td>———</td> </tr> <tr> <td>ON</td> <td>0</td> <td>0</td> <td></td> <td>0</td> <td></td> </tr> </table>	リセット入力	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354		OFF	0または1	0または1	0	1または0	———	ON	0	0		0		
リセット入力	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354																
OFF	0または1	0または1	0	1または0	———															
ON	0	0		0																

[使用例]



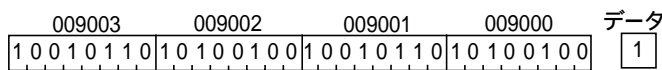
命 令	
STR	000000
STR	000001
STR	000002
STR	000003
F-60d	009000

入力条件

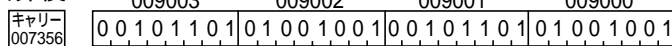
- 000000()ON MSB方向へシフト
- 000001()ON データ入力ON
- 000002()OFF ON ---- シフト指示
- 000003()OFF リセット機能なし

演算結果は次のとおりです。

演算前



演算後



ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
0	1	0	0

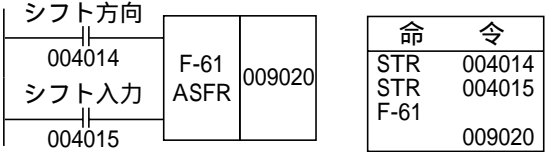
・リセット入力 は、システムメモリ(#0202)の設定で「OFFでリセット」も可能です。

(類似命令) F-60、F-60w、F-160、Fc160

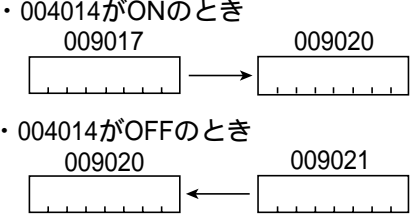
F-61 ASFR 非同期両方向シフトレジスタ(1バイト)
(Asynchronous ShiFt Register)

シンボル			シフト方向指示入力 シフト入力				
機能	シフト方向指示入力 に従って、レジスタD・1(ON)またはレジスタD+1(OFF)の1バイトデータを、レジスタDにシフトする。						
演算内容	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 ・演算を実行すると、シフトしたレジスタ(D-1またはD+1)の内容はクリアします。 ・レジスタDの内容が0以外のとき、演算は実行しません。						
D	使用範囲 B						
演算条件	レジスタDの内容が0のとき、シフト入力 がONでシフト (OFF ONの変化時に限定されない)						
レ ジ の ス 内 タ 容		ON		OFF		ON / OFF	
		演算前	演算後	演算前	演算後	演算前	演算後
	D-1	D1	0	D1	同左	D1	同左
	D	0	D1	0	D2	0以外	同左
	D+1	D2	同左	D2	0	D2	同左
フ ラ ゲ	ノンキャリア 007354	1 (D1=0) 0 (D1=0)		1 (D2=0) 0 (D2=0)		1	
	エラー 007355	0		0		0	
	キャリア 007356	0 (D1=0) 1 (D1=0)		0 (D2=0) 1 (D2=0)		0	
	ゼロ 007357	0		0		0	

[使用例]



シフト入力004015がONの間、シフト方向指示入力004014の状態により、次のように1バイト単位のデータがシフトします。



- ・レジスタ009020の内容が0以外のとき、シフト(演算)しません。
- ・シフトしたレジスタ(009017または009021)の内容はクリアします。


入力条件	演算前								演算後								ゼロ 007357	キャリア 007356	ノンキャリア 007354	
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0				
004014 004015	009017 009020 009021																			
004014 004015	009017 009020 009021																			
004014 004015	009017 009020 009021																			
004014 004015	009017 009020 009021																			

・エラーフラグ(007355)は常にOFFです。 : OFF、 : ON

D-1またはD+1から0以外のデータがシフトした場合のみ、キャリアフラグ(007356)がONします。

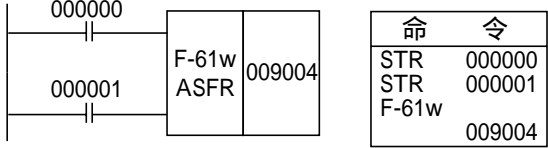
(類似命令) F-61w、 F-61d、 F-161、 F-161w、 F-161d

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	 シフト方向指示入力 シフト入力
機能	シフト方向指示入力 に従って、レジスタD-2、D-1(ON)またはレジスタD+2、D+3(OFF)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	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 ・演算を実行すると、シフトしたレジスタ(D-2、D-1またはD+2、D+3)の内容はクリアします。 ・レジスタD、D+1の内容が0以外のとき、演算は実行しません。
D	使用範囲C (Dには必ず偶数アドレスを設定してください。)
演算条件	レジスタD、D+1の内容が0のとき、シフト入力 がONでシフト (OFF ONの変化時に限定されない)

		ON		OFF		ON / OFF	
		演算前	演算後	演算前	演算後	演算前	演算後
レジ の ス 内 容	D-2、D-1	D1	0	D1	同左	D1	同左
	D、D+1	0	D1	0	D2	0以外	同左
	D+2、D+3	D2	同左	D2	0	D2	同左
フ ラ グ	ハンキャリー 007354	1 (D1 = 0) 0 (D1 0)		1 (D2 = 0) 0 (D2 0)		1	
	エラー 007355	0		0		0	
	キャリー 007356	0 (D1 = 0) 1 (D1 0)		0 (D2 = 0) 1 (D2 0)		0	
	ゼロ 007357	0		0		0	

[使用例]



000000()ON.....009002、009003よりシフト
 000001()ON.....シフト指示
 009004、009005の内容0000(H)

入力条件が上記のとき、演算結果は次のとおりです。

	演算前					演算後				
009000	1	2	3	4	↙	1	2	3	4	
009001						009001				
009002						009002	0	0	0	0
009003	5	6	7	8		009003				
009004	0	0	0	0		009004	5	6	7	8
009005						009005				
009006						009006	9	8	7	6
009007	9	8	7	6		009007				
009010						009010	5	4	3	2
009011	5	4	3	2		009011				

キャリーフラグ(007356)のみONします。

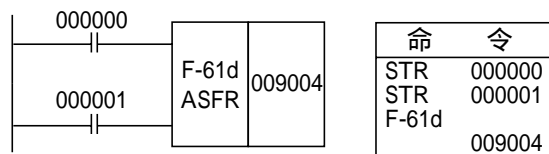
(類似命令) F-61、F-61d、F-161、F-161w、F-161d

**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			シフト方向指示入力 シフト入力				
機能	シフト方向指示入力に従って、レジスタD-4~D-1(ON)またはレジスタD+4~D+7(OFF)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	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				
	・演算を実行すると、シフトしたレジスタ(D-4~D-1またはD+4~D+7)の内容はクリアします。 ・レジスタD~D+3の内容が0以外のとき、演算は実行しません。						
D	使用範囲C(Dには必ず偶数アドレスを設定してください。)						
演算条件	レジスタD~D+3の内容が0のとき、シフト入力 がONでシフト(OFF ONの変化時に限定されない)						
		ON		OFF		ON / OFF	
		演算前	演算後	演算前	演算後	演算前	演算後
レジの ス内 タ容	D-4~D-1	D1	0	D1	同左	D1	同左
	D~D+3	0	D1	0	D2	0以外	同左
	D+4~D+7	D2	同左	D2	0	D2	同左
フラ グ	ノンキャリア 007354	1 (D1=0) 0 (D1 0)		1 (D2=0) 0 (D2 0)		1	
	エラー 007355	0		0		0	
	キャリア 007356	0 (D1=0) 1 (D1 0)		0 (D2=0) 1 (D2 0)		0	
	ゼロ 007357	0		0		0	

[使用例]



000000()ON.....009000~009003よりシフト
 000001()ON.....シフト指示
 009004~009007の内容00000000(H)

入力条件が上記のとき、演算結果は次のとおりです。

	演算前				演算後				
009000	1	2	3	4	0	0	0	0	009000
009001									009001
009002									009002
009003	5	6	7	8	0	0	0	0	009003
009004					1	2	3	4	009004
009005									009005
009006					5	6	7	8	009006
009007									009007
009010					9	8	7	6	009010
009011									009011
009012					5	4	3	2	009012
009013									009013

キャリアフラグ(007356)のみONします。

(類似命令)F-61、F-61w、F-161、F-161w、F-161d

**F-62
U/DC**

**BCD 2桁のアップ・ダウンカウンタ
(Up/Down Counter)**

シンボル			アップダウン指示入力 カウント入力 リセット入力			
機能	アップダウン指示入力に従って、レジスタDの内容(BCD 2桁)を、加算(ON)または減算(OFF)する。					
演算内容	・アップダウン指示入力 ONのとき $D + 1 \rightarrow D$ ・アップダウン指示入力 OFFのとき $D - 1 \rightarrow D$					
D	使用範囲 A					
演算条件	リセット入力 がOFFのとき、カウント入力の立上り(OFF ON)					
演算後の内容 フラグ	D	・リセット入力 がOFFのとき 演算結果(BCDコード) ・リセット入力 がONのとき 全ビットOFF				
	アップダウン指示入力	演算結果	ゼロフラグ (007357)	キャリーフラグ (007356)	エラーフラグ (007355)	ノンキャリーフラグ (007354)
	ON	99 + 1 00	1	1	0	0
		00 ~ 98 + 1 01 ~ 99	0	0	0	1
	OFF	BCD以外の数値	0	0	1	0
		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 がON	0	0	0	0	0	

[使用例]

リセット入力0004011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004010がONのとき加算、OFFのとき減算カウンタとして動作します。レジスタ009000の内容がBCDコード以外るとき、エラーフラグがONし、カウント動作は実行しません。(下例では7F)

アップダウン指示 (004010) UP DOWN

カウント入力 (000000)

リセット入力 (004011)

レジスタ (009000) 97 98 98 99 99 00 00 01 01 00 00 99)) 7F 7F 00 00

ノンキャリーフラグ(007354)

エラーフラグ(007355)

キャリーフラグ(007356)

ゼロフラグ(007357)

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

・ Dの内容がBCDコード以外るとき、エラーフラグ(007355)がONし、演算を実行しません。
(Dの内容は不変)

・ リセット入力 は、システムメモリ(#0202)の設定により「OFF」でリセットも可能です。

(類似命令) F-62w、F-62d、F-65、F-65w、F-65d、F-66、F-66w

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		アップダウン 指示入力 カウント入力 リセット入力	[使用例]										
機能	アップダウン指示入力 に従って、レジスタD、D+1の内容(BCD 4桁)を、加算(ON)または減算(OFF)する。												
演算内容	<ul style="list-style-type: none"> ・アップダウン指示入力 ONのとき D、D+1 +1 → D、D+1 ・アップダウン指示入力 OFFのとき D、D+1 -1 → D、D+1 		<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>000001</td></tr> <tr><td>STR</td><td>000002</td></tr> <tr><td>STR</td><td>000003</td></tr> <tr><td>F-62w</td><td>019000</td></tr> </table>	命 令		STR	000001	STR	000002	STR	000003	F-62w	019000
命 令													
STR	000001												
STR	000002												
STR	000003												
F-62w	019000												
D	使用範囲B ・Dには必ず偶数アドレスを設定してください。		リセット入力000003がOFFで計数可能となります。 (ONリセットに設定時)										
演算条件	リセット入力 がOFFのとき、カウント入力の立上り(OFF ON)		アップダウン指示入力000001がONのとき加算、OFFのとき減算カウンタとして動作します。レジスタ019000、019001の内容がBCDコード以外の場合、エラーフラグがONし、カウント動作は実行しません。										
演算後の内容 フラグ	D	演算結果(下2桁)	・リセット入力 ONのとき、全ビットOFF										
	D+1	演算結果(上2桁)											
	ON	アップダウン指示入力	演算結果	ゼロフラグ(007357)	キャリーフラグ(007356)	エラーフラグ(007355)	ノンキャリーフラグ(007354)						
		0000 ~ 9998 +1	9999 + 1	1	1	0	0						
		BCDコード以外の数値	0000 ~ 9998 + 1	0	0	0	1						
		OFF	0000 - 1	0000 - 1	0	1	0	0					
			0001 - 1	0001 - 1	1	0	0	1					
	リセット入力 がON	0002 ~ 9999 - 1	0002 ~ 9999 - 1	0	0	0	1						
		BCDコード以外の数値	BCDコード以外の数値	0	0	1	0						
	レジスタ (019000)		97 98 99 00 01 00 99 99 00 00										
レジスタ (019001)		99 99 99 00 00 00 00 AC A0 00 00											
アップダウン指示 (000001)		UP		DOWN									
カウント入力 (000002)													
リセット入力 (000003)													
ノンキャリーフラグ(007351)													
エラーフラグ(007355)													
キャリーフラグ(007356)													
ゼロフラグ(007357)													
		1 スキャンタイム以内											

・リセット入力 は、システムメモリ (#0202)の設定により「OFF」でリセットも可能です。

(類似命令) F-62、F-62d、F-65、F-65w、F-65d、F-66、F-66w

F-62d
U/DC

BCD 8桁のアップ・ダウンカウンタ (Up/Down Counter)

シンボル			アップダウン 指示入力 カウント入力 リセット入力			
機能	アップダウン指示入力 に従って、レジスタD~D+3の内容(BCD 8桁)を、加算(ON)または減算(OFF)する。					
演算内容	・アップダウン指示入力 ONのとき $D \sim D+3 + 1 \rightarrow D \sim D+3$ ・アップダウン指示入力 OFFのとき $D \sim D+3 - 1 \rightarrow D \sim D+3$					
D	使用範囲C ・Dには必ず偶数アドレスを設定してください。					
演算条件	リセット入力 がOFFのとき、カウント入力の立上り(OFF ON)					
演算後の内容 フラグ	D~D+3	演算結果(8桁) ・リセット入力 ONのとき、全ビットOFF				
	アップダウン指示入力	演算結果	ゼロフラグ (007357)	キャリーフラグ (007356)	エラーフラグ (007355)	ノンキャリーフラグ (007354)
	ON	99999999 + 1	1	1	0	0
		00000000 ~ 99999998 + 1	0	0	0	1
	OFF	BCDコード以外の数値	0	0	1	0
		00000000 - 1	0	1	0	0
	OFF	00000001 - 1	1	0	0	1
00000002 ~ 99999999 - 1		0	0	0	1	
リセット入力 がON		0	0	0	0	

[使用例]

命 令	
STR	000001
STR	000002
STR	000003
F-62d	019000

リセット入力000003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000001がONのとき加算、OFFのとき減算カウンタとして動作します。レジスタ019000~019003の内容がBCDコード以外のとき、エラーフラグがONし、カウント動作は実行しません。

アップダウン指示 (000001) UP DOWN

カウント入力 (000002)

リセット入力 (000003)

レジスタ (019000~019003) 9999 9999 9999 0 1 0 (9999 999A 0 0

ノンキャリーフラグ(007354)

エラーフラグ(007355)

キャリーフラグ(007356)

ゼロフラグ(007357)

1スキャンタイム以内

・D~D+3の内容がBCDコード以外のとき、エラーフラグ(007355)がONし、演算を実行しません。
(D~D+3の内容は不変)

・リセット入力 は、システムメモリ(#0202)の設定により「OFF」でリセットも可能です。

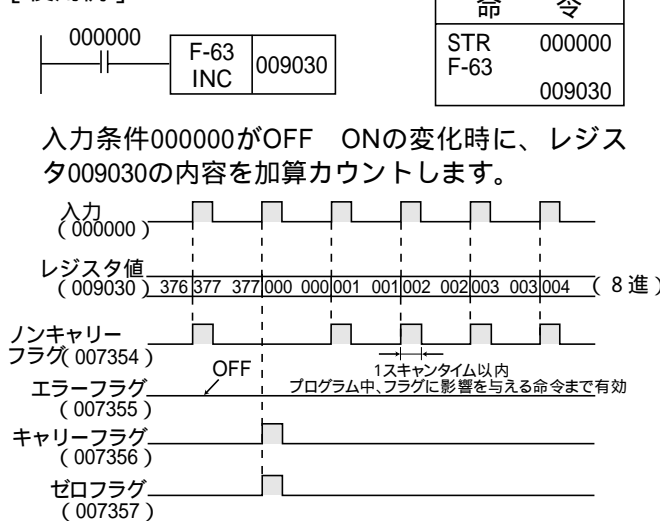
(類似命令) F-62、F-62w、F-65、F-65w、F-65d、F-66、F-66w

**F-63
INC** **加算カウンタ(1 バイト)**
(INCRement)

シンボル	F-63 INC		D			
機能	レジスタDの内容(バイナリデータ)を加算カウントする。					
演算内容	D +1 → D					
D	使用範囲 A					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果(バイナリコード)				
	フラグ	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		377 000のとき	1	1	0	0
		上記以外	0	0	0	1

(類似命令) F-63w、F-63d、F-163、F-163w、
F-163d、F-263、F-263w、F-263d

[使用例]



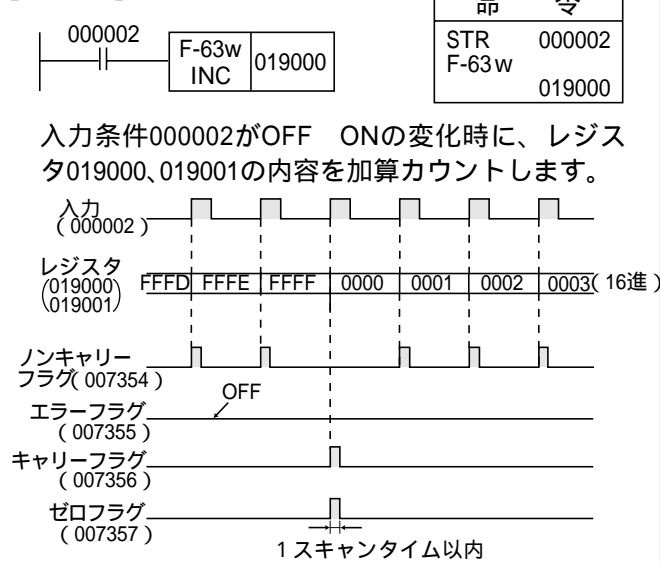
・Dの内容はバイナリコードです。10進表現では000~255、8進表現では000~377(8)と見なせます。

**F-63w
INC** **加算カウンタ(1 ワード)**
(INCRement)

シンボル	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	D、D+1 +1 → D、D+1				
D	使用範囲 B ・Dには必ず偶数アドレスを設定してください。(019003等は禁止)				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	D	演算結果(下位)			
	D+1	演算結果(上位)			
フラグ	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	177777 000000 のとき	1	1	0	0
	上記以外	0	0	0	1

(類似命令) F-63、F-63d、F-163、F-163w、F-163d、F-263、F-263w、F-263d

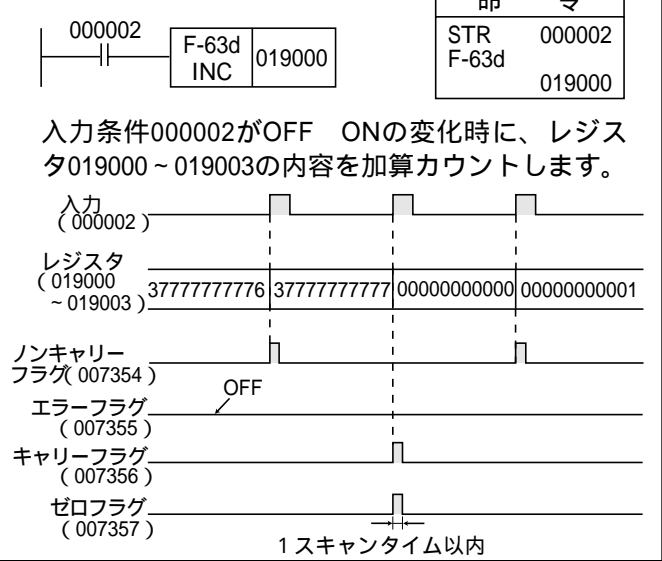
[使用例]



**F-63d
INC** 加算カウンタ(2ワード)
(INCRement)

シンボル	F-63d INC		D		
機能	レジスタD ~ D+3の内容(バイナリデータ)を加算カウントする。				
演算内容	D ~ D+3 +1 → D ~ D+3				
D	使用範囲 C ・ Dには必ず偶数アドレスを設定してください。(019003等は禁止)				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	演算結果(D : 下位、D+3 : 上位)				
フラグ	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	3777777777 00000000000のとき	1	1	0	0
	上記以外	0	0	0	1

[使用例]



(類似命令) F-63、F-63w、F-163、F-163w、F-163d、F-263、F-263w、F-263d

**F-64
DEC** 減算カウンタ(1 バイト)
(DECrement)

シンボル	F-64 DEC		D			
機能	レジスタDの内容(バイナリデータ)を減算カウントする。					
演算内容	D - 1 → D					
D	使用範囲 A					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果(バイナリコード)				
	フラグ	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		001 000のとき	1	0	0	1
		000 377のとき	0	1	0	0
上記以外	0	0	0	1		

[使用例]

命 令	
STR	000000
F-64	009000

入力条件000000がOFF ONの変化時に、レジスタ009000の内容を減算カウントします。

入力 (000000)

レジスタ値 (009000) 001 000 000 377 377 376 376 375 375 374 373 (8進)

ノンキャリーフラグ (007354) OFF

エラーフラグ (007355) OFF

キャリーフラグ (007356)

ゼロフラグ (007357)

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

(類似命令) F-64w、F-64d、F-164、F-164w、F-164d、F-264、F-264w、F-264d

・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。

**F-64w
DEC** 減算カウンタ(1 ワード)
(DECrement)

シンボル	F-64w DEC		D			
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	D、D+1 - 1 → D、D+1					
D	使用範囲 B ・Dには必ず偶数アドレスを設定してください。(019003等は禁止)					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果(下位)				
	D+1	演算結果(上位)				
	フラグ	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		000001 000000のとき	1	0	0	1
000000 177777のとき		0	1	0	0	
上記以外	0	0	0	1		

[使用例]

命 令	
STR	000002
F-64w	019000

入力条件000002がOFF ONの変化時に、レジスタ019000、019001の内容を減算カウントします。

入力 (000002)

レジスタ (019000 (019001)) 0002 0001 0000 FFFF FFFE FFFD FFFC 16進)

ノンキャリーフラグ (007354) OFF

エラーフラグ (007355) OFF

キャリーフラグ (007356)

ゼロフラグ (007357)

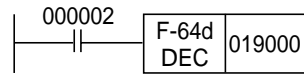
1スキャンタイム以内

(類似命令) F-64、F-64d、F-164、F-164w、F-164d、F-264、F-264w、F-264d

F-64d DEC	減算カウンタ(2ワード) (DECrement)
--------------	---------------------------------

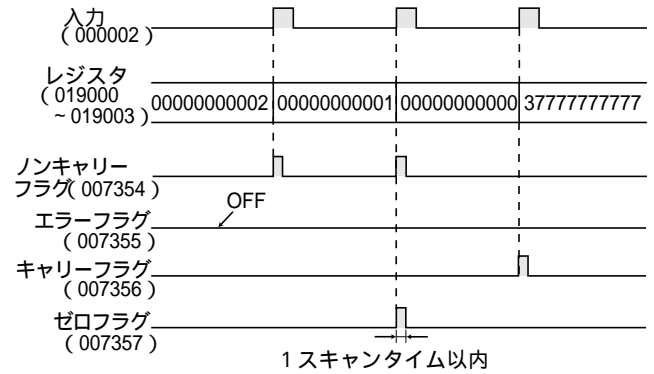
シンボル	F-64d DEC		D			
機能	レジスタD ~ D+3の内容(バイナリデータ)を減算カウントする。					
演算内容	D ~ D+3 -1 → D ~ D+3					
D	使用範囲 C ・ Dには必ず偶数アドレスを設定してください。(019003等は禁止)					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D ~ D+3	演算結果(D : 下位、 D+3 : 上位)				
		演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	0000000001 00000000000のとき	1	0	0	1	
	0000000000 3777777777のとき	0	1	0	0	
	上記以外	0	0	0	1	

[使用例]



命 令	
STR	000002
F-64d	019000

入力条件000002がOFF ONの変化時に、レジスタ 019000 ~ 019003の内容を減算カウントします。

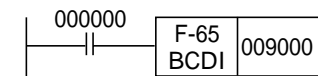


(類似命令) F-64、F-64w、F-164、F-164w、F-164d、F-264、F-264w、F-264d

F-65 BCDI **BCD加算カウンタ(1バイト)**
(BCD Increment)

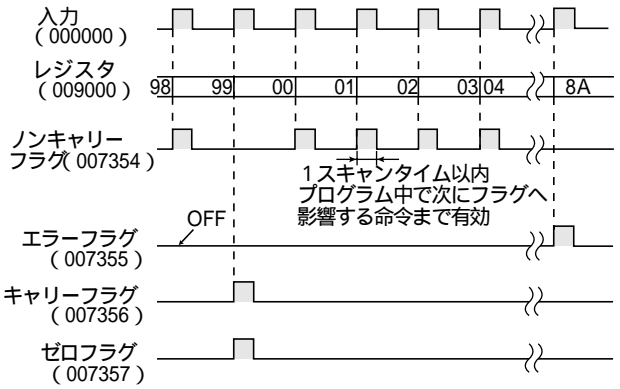
シンボル	F-65 BCDI D					
機能	レジスタDの内容(BCDデータ)を加算カウントする。					
演算内容	D +1 → D					
D	使用範囲 A					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果(BCDコード) ・レジスタDの内容がBCDコード以外 のとき不変				
	フラグ	演算結果(BCD)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		99 00のとき	1	1	0	0
		上記以外	0	0	0	1
BCDコード以外	0	0	1	0		

[使用例]



命 令	
STR	000000
F-65	009000

入力条件000000がOFF ONの変化時に、レジスタ009000の内容を加算カウント(+ 1)します。



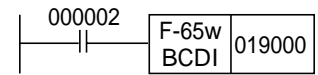
・ Dの内容がBCDコード以外
のとき、エラーフラグ(007355)がONし、演算は実行しません。
(Dの内容は不変)

(類似命令) F-62、F-62w、F-62d、F-65w、F-65d

F-65w BCDI **BCD加算カウンタ(1ワード)**
(BCD Increment)

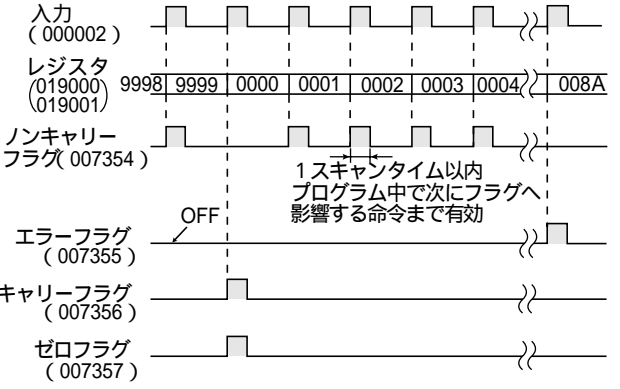
シンボル	F-65w BCDI D					
機能	レジスタD、D+1の内容(BCDデータ)を加算カウントする。					
演算内容	D、D+1 +1 → D、D+1					
D	使用範囲 B ・ Dには必ず偶数アドレスを設定してください。(019003等は禁止)					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果(下位)	・レジスタD、D+1の内容がBCDコード以外 のとき不変			
	D+1	演算結果(上位)				
	フラグ	演算結果(BCD)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		9999 0000 のとき	1	1	0	0
上記以外		0	0	0	1	
BCDコード以外	0	0	1	0		

[使用例]



命 令	
STR	000002
F-65w	019000

入力条件000002がOFF ONの変化時に、レジスタ019000、019001の内容を加算カウントします。



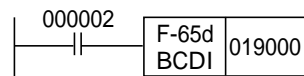
・ D、D+1の内容がBCDコード以外
のとき、エラーフラグ(007355)がONし、演算は実行しません。
(D、D+1の内容は不変)

(類似命令) F-62、F-62w、F-62d、F-65、F-65d

F-65d BCD加算カウンタ(2ワード)
BCDI (BCD Increment)

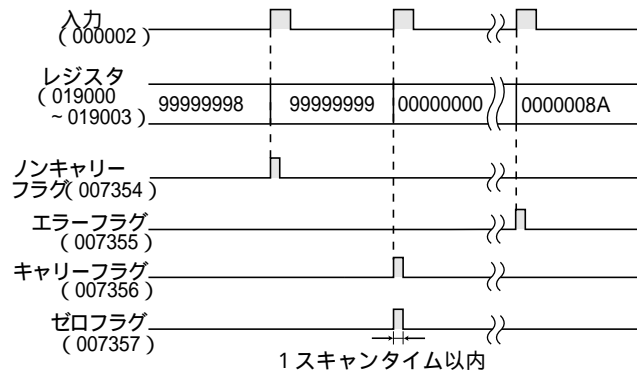
シンボル	F-65d BCDI		D	
機能	レジスタD~D+3の内容(BCDデータ)を加算カウントする。			
演算内容	D~D+3 +1 → D~D+3			
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。(019003等は禁止)			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	D~D+3	演算結果(D : 下位、 D+3 : 上位)		
	演算結果(BCD)	ゼロ 007357	キャリー 007356	エラー 007355
	99999999 00000000のとき	1	1	0
	上記以外	0	0	1
BCDコード以外	0	0	1	0

[使用例]



命 令	
STR	000002
F-65d	019000

入力条件000002がOFF ONの変化時に、レジスタ019000~019003の内容を加算カウントします。



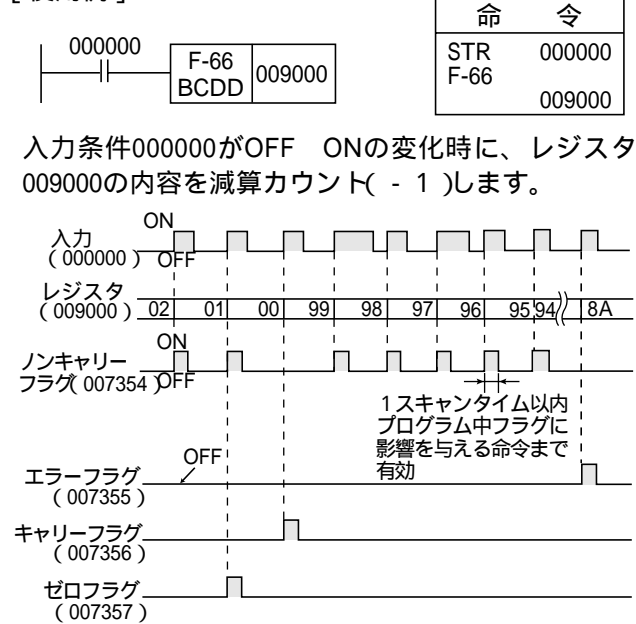
D~D+3の内容がBCDコード以外るとき、エラーフラグ(007355)がONし、演算は実行しません。
 (D~D+3の内容は不変)

(類似命令) 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	D - 1 → D					
D	使用範囲 A					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果(BCDコード) ・レジスタDの内容がBCDコード以外のとき不変				
	フラグ	演算結果(BCD)	ゼロ (007357)	キャリー (007356)	エラー (007355)	ノンキャリー (007354)
		01 00のとき	1	0	0	1
		00 99のとき	0	1	0	0
		上記以外	0	0	0	1
BCDコード以外	0	0	1	0		

[使用例]



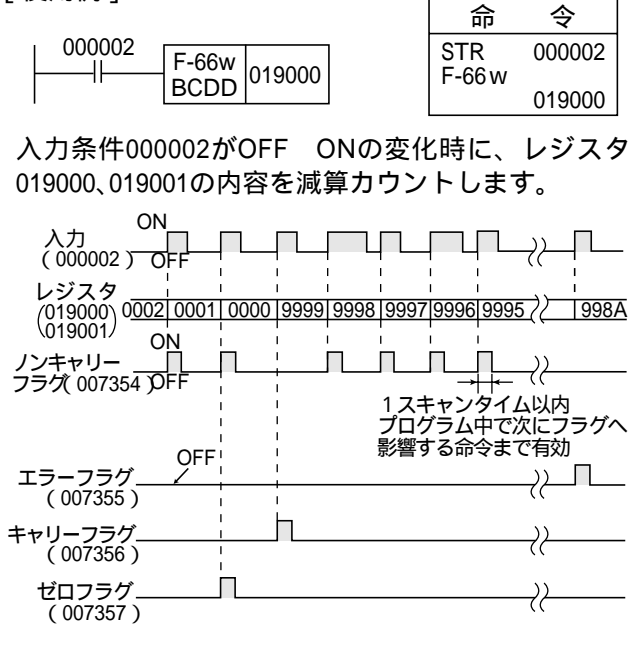
・ Dの内容がBCDコード以外るとき、エラーフラグ(007355)がONし、演算は実行しません。
(Dの内容は不変)

(類似命令) F-62、F-62w、F-62d、F-66w、F-66d

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	D、D+1 - 1 → D、D+1					
D	使用範囲 B ・ Dには必ず偶数アドレスを設定してください。(019003等は禁止)					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果(下位)	・ レジスタD、D+1の内容がBCDコード以外るとき不変			
	D+1	演算結果(上位)				
	フラグ	演算結果(BCD)	ゼロ (007357)	キャリー (007356)	エラー (007355)	ノンキャリー (007354)
		0001 0000のとき	1	0	0	1
		0000 9999のとき	0	1	0	0
上記以外		0	0	0	1	
BCDコード以外	0	0	1	0		

[使用例]



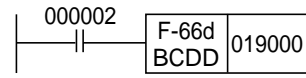
・ D、D+1の内容がBCDコード以外るとき、エラーフラグ(007355)がONし、演算は実行しません。
(D、D+1の内容は不変)

(類似命令) F-62、F-62w、F-62d、F-66、F-66d

F-66d BCD減算カウンタ(2ワード)
BCDD (BCD Decrement)

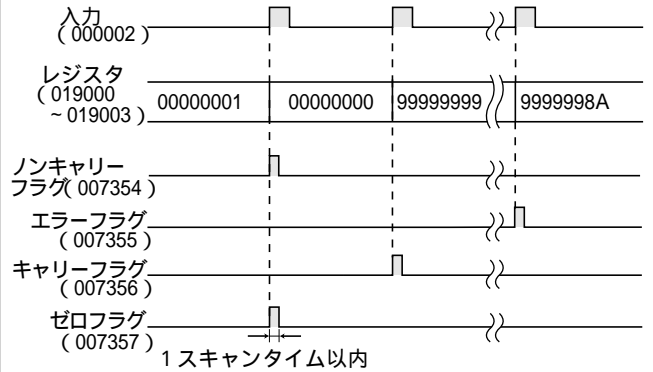
シンボル	F-66d BCDD		D		
機能	レジスタD~D+3の内容(BCDデータ)を減算カウントする。				
演算内容	D~D+3 -1 → D~D+3				
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。(019003等は禁止)				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	D~D+3	演算結果(D : 下位、 D+3 : 上位)			
	フラグ	演算結果(BCD)	ゼロ 007357	キャリー 007356	エラー 007355
		00000001 00000000のとき	1	0	0
		00000000 99999999のとき	0	1	0
		上記以外	0	0	0
BCDコード以外	0	0	1		

[使用例]



命 令	
STR	000002
F-66d	019000

入力条件000002がOFF ONの変化時に、レジスタ019000~019003の内容を減算カウントします。



D~D+3の内容がBCDコード以外るとき、エラーフラグ(007355)がONし、演算は実行しません。
 (D~D+3の内容は不変)

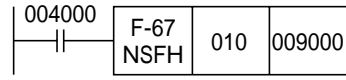
(類似命令) F-62、F-62w、F-62d、F-66、F-66w

**F-67
NSFH**

桁シフト(上位)

シンボル	— 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、上位へ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 ・演算後、先頭レジスタDの下位4ビットに0を格納します。また、最終レジスタ(nバイト目)の上位4ビットのデータは、消去(クリア)します。
n	使用範囲 000 ~ 377 ₍₈₎ (000 ₍₈₎ のとき256バイト)
D	使用範囲 A
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D 演算結果(シフト結果)
	フラグ 不変

[使用例]



命 令	
STR	004000
F-67	010
	009000

入力条件004000がOFF ONの変化時に、レジスタ009000 ~ 009007の8(010₍₈₎)バイトのデータを、4ビットづつ上位へシフトします。

		演算前		→	演算後	
8バイト	009000	2	1	→	1	0
	009001	4	3	→	3	2
	009002	6	5	→	5	4
	009003	8	7	→	7	6
	009004	10	9	→	9	8
	009005	12	11	→	11	10
	009006	14	13	→	13	12
	009007	16	15	→	15	14
						16
						消去

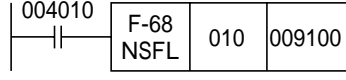
↑ 4ビットづつ上位へシフト

**F-68
NSFL**

桁シフト(下位)

シンボル	— 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、下位へ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を上位へ4ビットづつシフト ・演算後、レジスタD+n-1の上位4ビットに0を格納します。また、レジスタDの下位4ビットのデータは消去(クリア)します。
n	使用範囲 000 ~ 377 ₍₈₎ (000 ₍₈₎ のとき256バイト)
D	使用範囲 A
演算条件	入力信号の立上り(OFF ON)
演算後の内容	D 演算結果(シフト結果)
	フラグ 不変

[使用例]



命 令	
STR	004010
F-68	010
	009100

入力条件004010がOFF ONの変化時に、レジスタ009100 ~ 009107の8(010₍₈₎)バイトのデータを、4ビットづつ下位へシフトします。

		演算前		→	演算後	
8バイト	009100	15	16	→	14	15
	009101	13	14	→	12	13
	009102	11	12	→	10	11
	009103	9	10	→	8	9
	009104	7	8	→	6	7
	009105	5	6	→	4	5
	009106	3	4	→	2	3
	009107	1	2	→	0	1
						16 消去

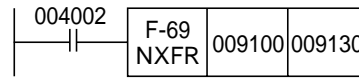
↑ 4ビットづつ下位へシフト

F-69
NXFR

桁転送

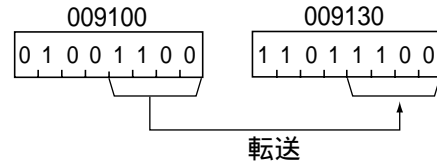
シンボル	F-69 NXFR		S	D
機能	レジスタSの下位4ビットを、レジスタDの下位4ビットに転送する。			
演算内容	Sの下位4ビット→Dの下位4ビット			
S	使用範囲A			
D	使用範囲A			
演算条件	入力信号の立上り			
演算後の内容	S	不変		
	D	レジスタSの下位4ビットの内容 (上位は不変)		
フラグ	不変			

[使用例]



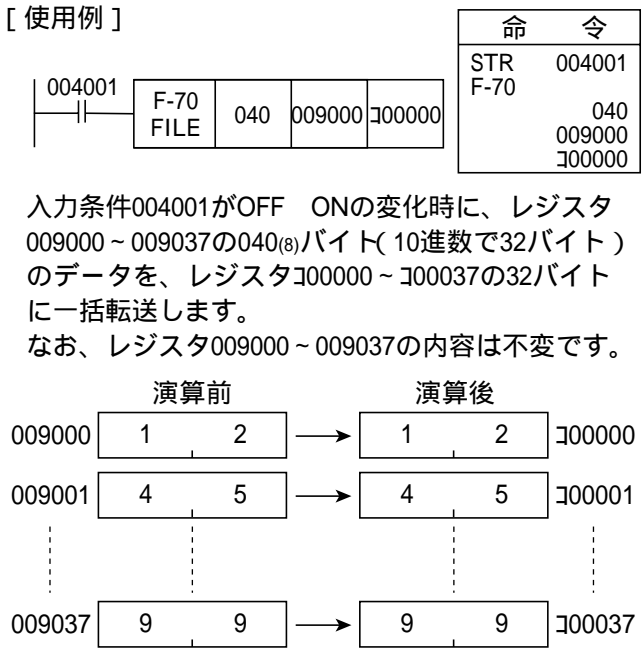
命 令	
STR	004002
F-69	009100 009130

入力条件004002がOFF ONの変化時に、レジスタ009100の下位4ビットの内容を、レジスタ009130の下位4ビットに格納します。

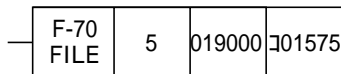


F-70 FILE **nバイト一括転送 (FILE)**

シンボル		[使用例]	
機能	レジスタSからS+n-1のデータ(nバイト)を、レジスタDからD+n-1のnバイトに一括転送する。		
演算内容	S, S+1, ……、S+n-2, S+n-1 → D, D+1, ……、D+n-2, D+n-1		
n	使用範囲 000 ~ 377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256バイト)		
S	使用範囲 A		
D	使用範囲 A		
演算条件	入力信号の立上り(OFF ON)		
演算後の内容	S, ……、S+n-1	不変	
	D	レジスタSの内容	
	D+1	レジスタS+1の内容	
	⋮	⋮	
	D+n-2 D+n-1	レジスタS+n-2の内容 レジスタS+n-1の内容	
フ ラ グ	不変		

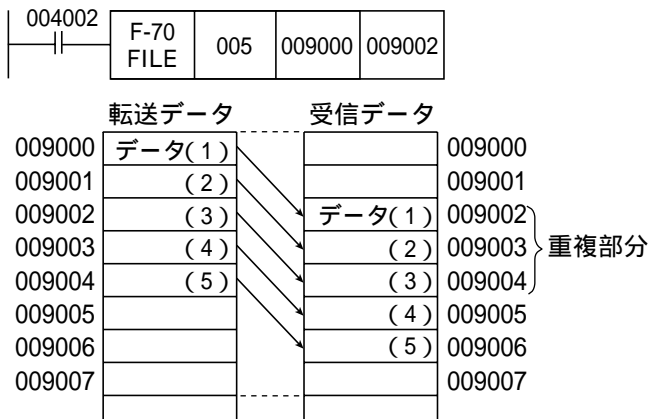


ファイル番号0のタイマ・カウンタの接点領域(ファイルアドレス00001600 ~ 00001777等)には転送しないでください。



上記のプログラムでは、00001600、00001601に019003、019004の内容が転送されてしまいます。

転送元、転送先が重複するようなn、S、Dの設定も可能です。

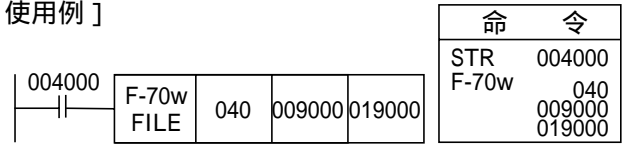


(類似命令) F-00、F-00w、F-00d、F-70w、F-70d、F-74、F-74w、F-74d、F-76、F-76w、F-76d

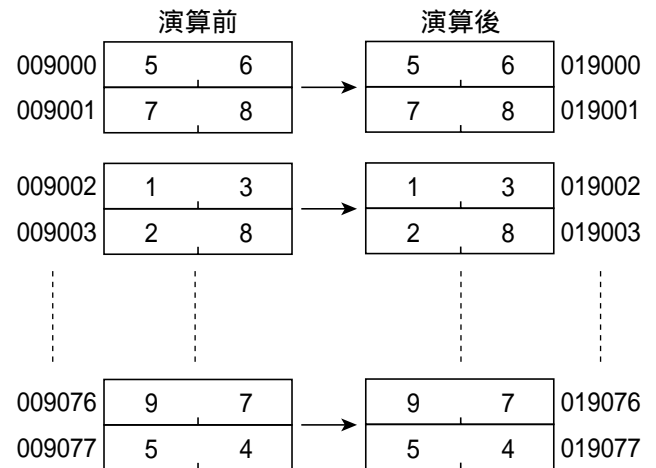
F-70w nワード一括転送
FILE (FILE)

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-70w</td> <td style="padding: 2px;">n</td> <td style="padding: 2px;">S</td> <td style="padding: 2px;">D</td> </tr> </table>	F-70w	n	S	D
F-70w	n	S	D		
機能	レジスタSからS+2n-1のデータ(nワード)を、レジスタDからD+2n-1のnワードに一括転送する。				
演算内容	S、S+1、……、S+2n-2、S+2n-1 →D、D+1、……、D+2n-2、D+2n-1				
n	使用範囲 000 ~ 377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256ワード)				
S	使用範囲 B ・Sには必ず偶数アドレスを設定してください。				
D	使用範囲 B ・Dには必ず偶数アドレスを設定してください。				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S、……、S+2n-1 不変				
	D レジスタSの内容				
	D+1 レジスタS+1の内容				
	⋮				
D+2n-2 レジスタS+2n-2の内容					
D+2n-1 レジスタS+2n-1の内容					
フラグ	不変				

[使用例]



入力条件004000がOFF ONの変化時に、レジスタ009000 ~ 009077の040⁽⁸⁾ワード(10進数で32ワード)のデータを、レジスタ019000 ~ 019077の32ワードに一括転送します。
 なお、レジスタ009000 ~ 009077の内容は不変です。



(類似命令) F-00、F-00w、F-00d、F-70、F-70d、F-74、F-74w、F-74d、F-76、F-76w、F-76d

F-70d nダブルワード一括転送
FILE (FILE)

シンボル	— F-70d FILE n S D			[使用例]	命 令
機 能	レジスタSからS+4n-1のデータ(nダブルワード)を、レジスタDからD+4n-1のnダブルワードに一括転送する。			004000 F-70d FILE 040 009000 019000	STR 004000 F-70d 040 009000 019000
演算内容	S ~ S+3、……、S+4n-4 ~ S+4n-1 → D ~ D+3、……、D+4n-4 ~ D+4n-1			入力条件004000がOFF ONの変化時に、レジスタ009000 ~ 009177の040 ⁽⁸⁾ ダブルワード(10進数で32ダブルワード)のデータを、レジスタ019000 ~ 019177の32ダブルワードに一括転送します。なお、レジスタ009000 ~ 009177の内容は不変です。	
n	使用範囲 000 ~ 377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256ダブルワード)			演算前	
S	使用範囲 C ・ Sには必ず偶数アドレスを設定してください。			演算後	
D	使用範囲 C ・ Dには必ず偶数アドレスを設定してください。			009000 5 6 019000 009001 7 8 019001 009002 1 3 019002 009003 2 8 019003 009004 A B 019004 009005 C D 019005 009006 E F 019006 009007 1 2 019007 ⋮ ⋮ ⋮ 009174 5 6 019174 009175 8 7 019175 009176 9 7 019176 009177 5 4 019177	
演算条件	入力信号の立上り(OFF ON)			→	
演算後の内容	S, ..., S+4n-1	不変			
	D ~ D+3	レジスタS ~ S+3の内容			
	D+4 ~ D+7 ⋮	レジスタS+4 ~ S+7の内容 ⋮			
	D+4n-4 ~ D+4n-1	レジスタS+4n-4の内容 ~ S+4n-1の内容			
フ ラ グ	不変				

(類似命令) F-00、F-00w、F-00d、F-70、F-70w、F-74、F-74w、F-74d、F-76、F-76w、F-76d

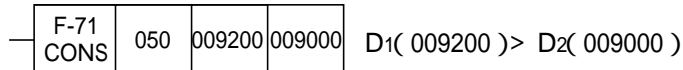
F-71 CONS 8進定数(1バイト)一括転送 (CONStant)

シンボル	— F-71 CONS n D1 D2			[使用例]																				
機能	レジスタD1からレジスタD2に、8進定数nを一括転送する。																							
演算内容	n → D1、……、D2			<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004001</td></tr> <tr><td>F-71</td><td>000</td></tr> <tr><td></td><td>009000</td></tr> <tr><td></td><td>009037</td></tr> </table>	命 令		STR	004001	F-71	000		009000		009037										
命 令																								
STR	004001																							
F-71	000																							
	009000																							
	009037																							
n	使用範囲 000 ~ 377 ⁽⁸⁾																							
D1	使用範囲 A																							
D2	使用範囲 A																							
演算条件	入力信号の立上り(OFF ON)			<p>入力条件004001がOFF ONの変化時に、レジスタ009000から009037に8進定数000を一括転送します。</p> <p>演算前</p> <table border="1"> <tr><td>009000</td><td>x x x x x x x x</td><td>→</td><td>0 0 0 0 0 0 0 0</td><td>009000</td></tr> <tr><td>009001</td><td>x x x x x x x x</td><td>→</td><td>0 0 0 0 0 0 0 0</td><td>009001</td></tr> <tr><td>⋮</td><td>⋮</td><td></td><td>⋮</td><td></td></tr> <tr><td>009037</td><td>x x x x x x x x</td><td>→</td><td>0 0 0 0 0 0 0 0</td><td>009037</td></tr> </table> <p>演算後</p>	009000	x x x x x x x x	→	0 0 0 0 0 0 0 0	009000	009001	x x x x x x x x	→	0 0 0 0 0 0 0 0	009001	⋮	⋮		⋮		009037	x x x x x x x x	→	0 0 0 0 0 0 0 0	009037
009000	x x x x x x x x	→	0 0 0 0 0 0 0 0		009000																			
009001	x x x x x x x x	→	0 0 0 0 0 0 0 0	009001																				
⋮	⋮		⋮																					
009037	x x x x x x x x	→	0 0 0 0 0 0 0 0	009037																				
演算後の内容	D1 D1+1 ⋮ D2-1 D2	定数 n																						
フラグ	不変																							

次表の ~ の領域をまたがるようなD1、D2を設定すると演算を実行しません。

領域	JW-311CU JW-312CU	JW-321CU JW-322CU	JW-331CU JW-332CU	JW-341CU JW-342CU	JW-352CU	JW-362CU	
リレー	300000 ~ 301577						
	302000 ~ 307577						
	/		310000 ~ 315377	310000 ~ 354377			
b00000 ~ b01777							
TMR・CNT ・MD現在値	b02000 ~ b03777						
	/		b04000 ~ b07777	b04000 ~ b37777			
			009000 ~ E07777				
レジスタ	109000 ~ Z377						
ファイル1	/		00000000 ~ 00077777	00000000 ~ 00377777	00000000 ~ 01777777	00000000 ~ 07777777	00000000 ~ 37777777

D1 > D2となるアドレスを設定すると、演算しません。



(類似命令) F-08、F-08w、F-08d、F-71w、F-71d

**F-71w
CONS**

**8進定数(1ワード)一括転送
(CONStant)**

シンボル	<table border="1"><tr><td>F-71w CONS</td><td>n</td><td>D1</td><td>D2</td></tr></table>	F-71w CONS	n	D1	D2	[使用例]	<table border="1"><tr><th colspan="2">命 令</th></tr><tr><td>STR</td><td>004000</td></tr><tr><td>F-71w</td><td>012345 009000 009036</td></tr></table>	命 令		STR	004000	F-71w	012345 009000 009036
F-71w CONS	n	D1	D2										
命 令													
STR	004000												
F-71w	012345 009000 009036												
機能	8進定数nを、レジスタD1、D1+1からレジスタD2、D2+1に一括転送する。												
演算内容	$n \rightarrow (D1, D1+1), \dots, (D2, D2+1)$												
n	使用範囲 000000 ~ 177777(8)	入力条件004000がOFF ONの変化時に、8進定数											
D1	使用範囲B ・D1には必ず偶数アドレスを設定してください。	012345を、レジスタ009000、009001から009036、009037											
D2	使用範囲B ・D2には必ず偶数アドレスを設定してください。	に一括転送します。											
演算条件	入力信号の立上り(OFF ON)												
演算後の内容	D1、D1+1 D1+2、D1+3 ⋮ D2-2、D2-1 D2、D2+1	演算後											
	フラグ	不変											

(類似命令) F-08、F-08w、F-08d、F-71、F-71d

**F-71d
CONS**

**8進定数(2ワード)一括転送
(CONStant)**

シンボル	<table border="1"><tr><td>F-71d CONS</td><td>n</td><td>D1</td><td>D2</td></tr></table>	F-71d CONS	n	D1	D2	[使用例]	<table border="1"><tr><th colspan="2">命 令</th></tr><tr><td>STR</td><td>004000</td></tr><tr><td>F-71d</td><td>01234567000 009000 009040</td></tr></table>	命 令		STR	004000	F-71d	01234567000 009000 009040
F-71d CONS	n	D1	D2										
命 令													
STR	004000												
F-71d	01234567000 009000 009040												
機能	8進定数nを、レジスタD1~D1+3からレジスタD2~D2+3に一括転送する。												
演算内容	$n \rightarrow (D1 \sim D1+3), \dots, (D2 \sim D2+3)$	入力条件004000がOFF ONの変化時に、8進定数012345を、											
n	使用範囲 00000000000 ~ 37777777777(8)	レジスタ009000、009001から009036、009037に一括転送し											
D1	使用範囲C ・D1には必ず偶数アドレスを設定してください。	ます。											
D2	使用範囲C ・D2には必ず偶数アドレスを設定してください。												
演算条件	入力信号の立上り(OFF ON)												
演算後の内容	D1~D1+3 D1+4~D1+7 ⋮ D2-4~D2-1 D2~D2+3	演算後											
	フラグ	不変											

(類似命令) F-08、F-08w、F-08d、F-71、F-71w

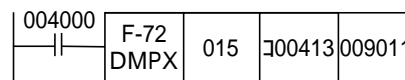
**F-72
DMPX**

**ファイルレジスタ(FILE 1)へのnバイト分配
(DeMultiPleXer)**

シンボル	F-72 DMPX	n	S	D
機能	レジスタSを含む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バイトの内容を、レジスタD~D+2の内容でアドレス指定するファイルレジスタ(ファイル番号1)に転送する。			
演算内容	$X + S, \dots, X + S + n - 1$ $\rightarrow D \sim D + 2, \dots, D \sim D + 2 + n - 1$ X : Sを含むデータメモリブロックの先頭アドレス(基準アドレス) S : データポインタ			
n	使用範囲 000 ~ 377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256バイト)			
S	使用範囲 A			
D	使用範囲 E			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	ファイル0のレジスタ	不変		
	D ~ D+2	レジスタX+ S の内容		
	D ~ D+2 + 1	レジスタX+ S + 1の内容		
	⋮	⋮		
D ~ D+2 + n - 1	レジスタX+ S + n - 1の内容			
フラグ	不変			

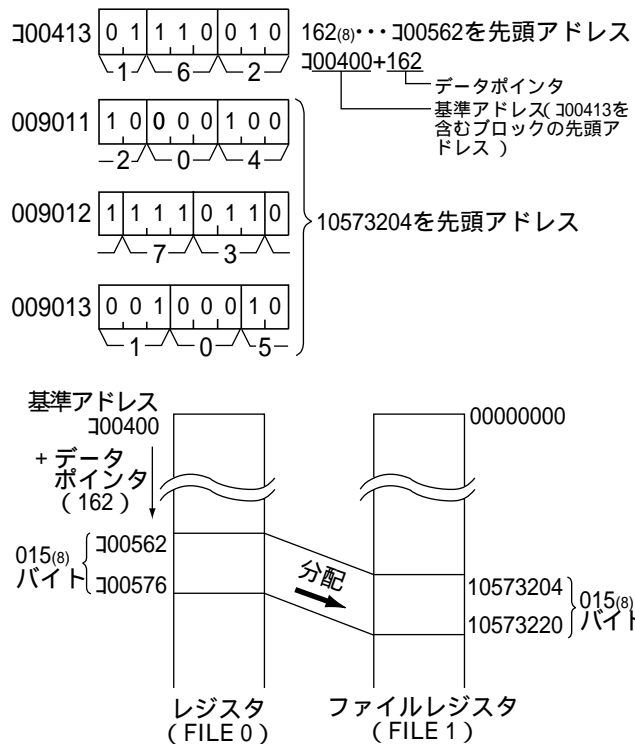
(類似命令) F-05、F-05w、F-05d、F-72w、F-72d

[使用例]



命 令	
STR	004000
F-72	015
	00413
	009011

入力条件004000がOFF ONの変化時に、00400(レジスタ00413を含む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00413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⁾バイトの内容を、レジスタ009011、009012の内容でアドレス指定するファイルレジスタ(FILE 1)を先頭とする015⁽⁸⁾バイトに転送します。



レジスタ側の基準アドレスは000000、004000、010000、.....、019000、019400、.....で、Sを含むデータメモリのブロックの先頭アドレスとなります。

データメモリのブロックと基準アドレス(9・26ページ)参照

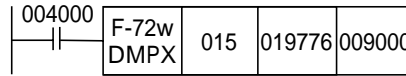
**F-72w
DMPX**

**ファイルレジスタ(FILE 1)へのnワード分配
(DeMultiPleXer)**

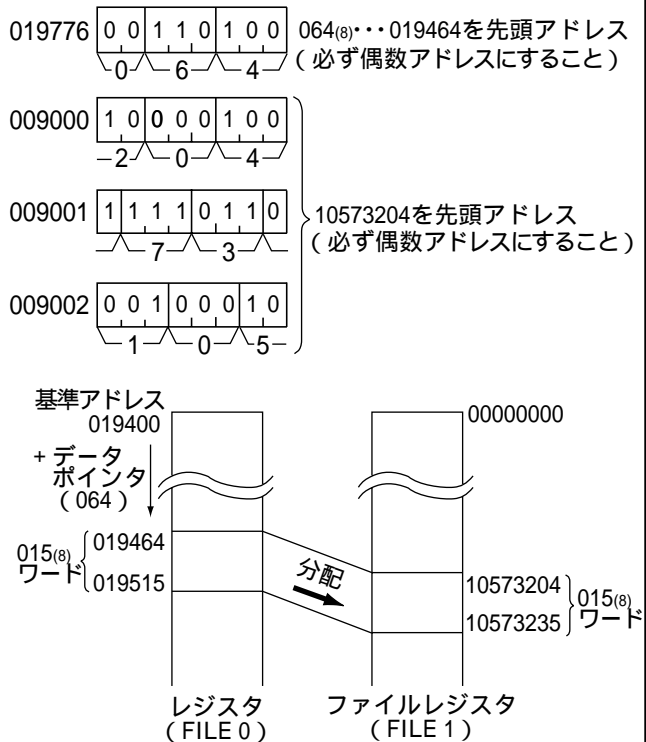
シンボル	F-72w DMPX	n	S	D
機能	レジスタSを含むデータメモリブロックの先頭アドレス(基準アドレス)から、Sの内容(データポインタ)だけ変位したレジスタを先頭とするnワードの内容を、レジスタD ~ D+2の内容でアドレス指定するファイルレジスタ(ファイル番号 1)に転送する。			
演算内容	$X + S, \dots, X + S + 2n - 1$ $\rightarrow D \sim D + 2, \dots, D \sim D + 2 + 2n - 1$ X : Sを含むデータメモリブロックの先頭アドレス(基準アドレス) S : データポインタ			
n	使用範囲 000 ~ 377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256ワード)			
S	使用範囲 B ・ Sには必ず偶数アドレスを設定してください。			
D	使用範囲 C ・ Dには必ず偶数アドレスを設定してください。			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	ファイル0のレジスタ	不変		
	D ~ D+2	レジスタX+ S の内容		
	D ~ D+2	レジスタX+ S +1の内容		
	⋮	⋮		
	D ~ D+2 +2n-1	レジスタX+ S +2n-1の内容		
フラグ	不変			

[使用例]

命 令	
STR	004000
F-72w	015
	019776
	009000

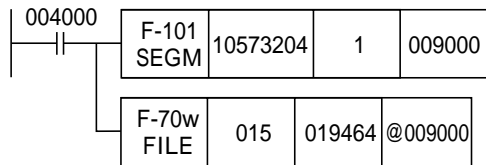


入力条件004000がOFF ONの変化時に、019400 (レジスタ019776を含むデータメモリブロックの先頭アドレス...基準アドレス)から、レジスタ019776の内容(データポインタ)だけ変位したレジスタを先頭とする015⁽⁸⁾ワードの内容を、レジスタ009000 ~ 009002の内容でアドレス指定するファイルレジスタ(FILE 1)を先頭とする015⁽⁸⁾ワードに転送します。



(類似命令) F-05、F-05w、F-05d、F-72、F-72d

参考 上記 使用例]のとき、次の命令と同じ動作となります。



レジスタ側の基準アドレスは000000、000400、001000、.....、019000、019400、.....で、Sを含むデータメモリのブロックの先頭アドレスとなります。

データメモリのブロックと基準アドレス(9・26ページ)参照

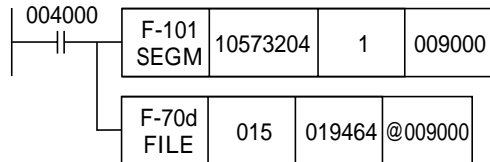
**F-72d
DMPX**

**ファイルレジスタ(FILE 1)へのnダブルワード分配
(DeMultiPleXer)**

シンボル	<table border="1"><tr><td>F-72d</td><td>n</td><td>S</td><td>D</td></tr></table>	F-72d	n	S	D
F-72d	n	S	D		
機能	レジスタSを含む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ダブルワードの内容を、レジスタD~D+2の内容でアドレス指定するファイルレジスタ(ファイル番号1)に転送する。				
演算内容	$X + S, \dots, X + S + 4n - 1$ → $D \sim D + 2, \dots, D \sim D + 2 + 4n - 1$ X : Sを含むデータメモリブロックの先頭アドレス(基準アドレス) S : データポインタ				
n	使用範囲 000 ~ 377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256ダブルワード)				
S	使用範囲 B ・Sには必ず偶数アドレスを設定してください。				
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	ファイル0のレジスタ 不変				
	D~D+2 レジスタX+ S の内容				
	D~D+2+1 レジスタX+ S +1の内容				
	D~D+2+4n-1 レジスタX+ S +4n-1の内容				
フラグ	不変				

(類似命令) F-05、F-05w、F-05d、F-72、F-72w

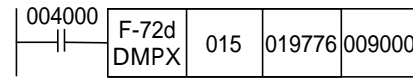
参考 上記使用例]のとき、次の命令と同じ動作となります。



レジスタ側の基準アドレスは000000、000400、001000、.....、019000、019400、.....で、Sを含むデータメモリブロックの先頭アドレスとなります。

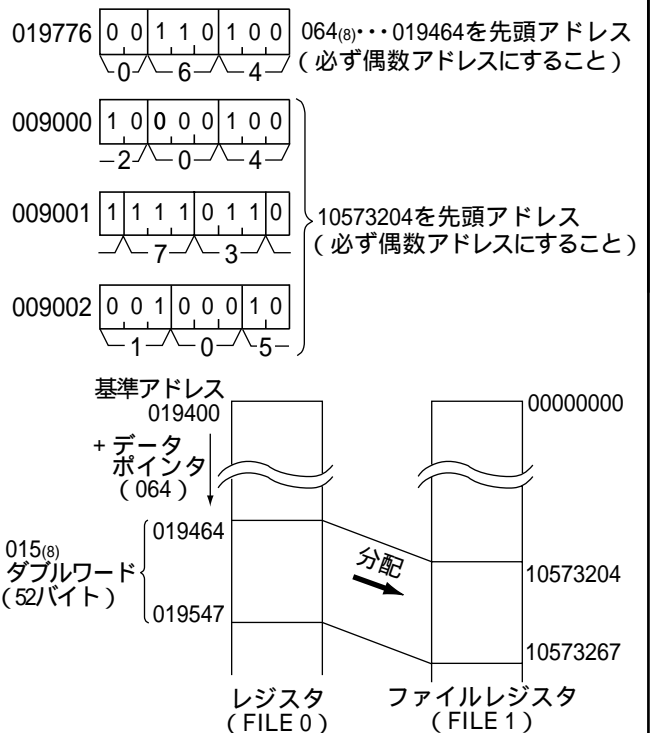
データメモリブロックと基準アドレス(9・26ページ)参照

[使用例]



命 令	
STR	004000
F-72d	015
	019776
	009000

入力条件004000がOFF ONの変化時に、019400 (レジスタ019776を含む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019776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⁾ダブルワードの内容を、レジスタ009000~009002の内容でアドレス指定するファイルレジスタ(FILE 1)を先頭とする015⁽⁸⁾ダブルワードに転送します。



**F-73
MPX**

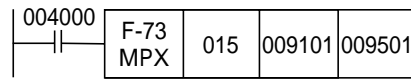
**ファイルレジスタ(FILE 1)からのnバイト抽出
(MultiPleXer)**

シンボル	F-73 MPX n S D	
機能	レジスタS~S+2の内容でアドレス指定するファイルレジスタ(ファイル番号1)を先頭とするnバイトの内容を、レジスタDを含む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バイトに転送する。	
演算内容	S~S+2、.....、S~S+2+n-1 → X+D、.....、X+D+n-1 X: Dを含むデータメモリブロックの先頭アドレス(基準アドレス) D: データポインタ	
n	使用範囲 000 ~ 377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256バイト)	
S	使用範囲 E	
D	使用範囲 A	
演算条件	入力信号の立上り(OFF ON)	
演算後の内容	ファイル1のレジスタ	不変
	X+D	ファイル1のS~S+2の内容
	X+D+1	ファイル1のS~S+2+1の内容

X+D+n-1	ファイル1のS~S+2+n-1の内容	
フラグ	不変	

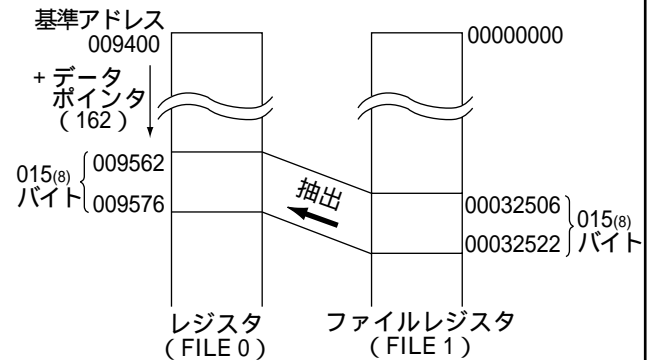
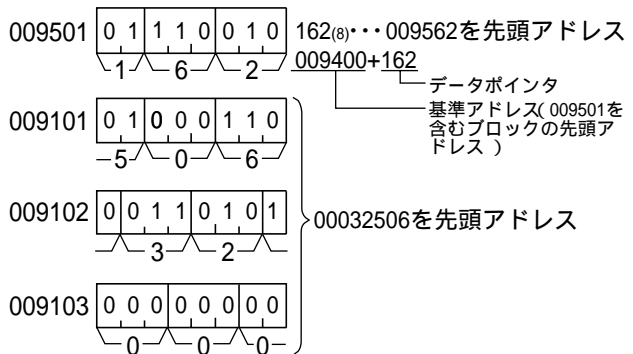
(類似命令) F-06、F-06w、F-06d、F-73w、F-73d

[使用例]



命 令	
STR	004000
F-73	015
	009101
	009501

入力条件004000がOFF ONの変化時に、レジスタ009101~009103の内容でアドレス指定するファイルレジスタ(FILE 1)を先頭とする015⁽⁸⁾バイトの内容を、009400(009501を含む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、009501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⁾バイトに転送します。



レジスタ側の基準アドレスは000000、000400、001000、.....、019000、019400、.....で、Sを含むデータメモリのブロックの先頭アドレスとなります。
データメモリのブロックと基準アドレス(9・26ページ)参照

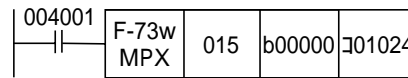
**F-73w
MPX**

**ファイルレジスタ(FILE 1)からの nワード抽出
(MultiPleXer)**

シンボル	F-73w MPX	n	S	D
機能	レジスタS~S+2の内容でアドレス指定するファイルレジスタ(ファイル番号1)を先頭とするnワードの内容を、レジスタDを含む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(データポイント)だけ変位したレジスタを先頭とするnワードに転送する。			
演算内容	S~S+2、.....、S~S+2+2n-1 → X+D、.....、X+D+2n-1 X: Dを含むデータメモリブロックの先頭アドレス(基準アドレス) D: データポイント			
n	使用範囲 000~377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256ワード)			
S	使用範囲 C ・Sには必ず偶数アドレスを設定してください。			
D	使用範囲 B ・Dには必ず偶数アドレスを設定してください。			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	ファイル1のレジスタ	不変		
	X+D	ファイル1の S~S+2 の内容		
	X+D+1	ファイル1の S~S+2 +1の内容		
	⋮	⋮		
	X+D+2n-1	ファイル1の S~S+2 +2n-1の内容		
フラグ	不変			

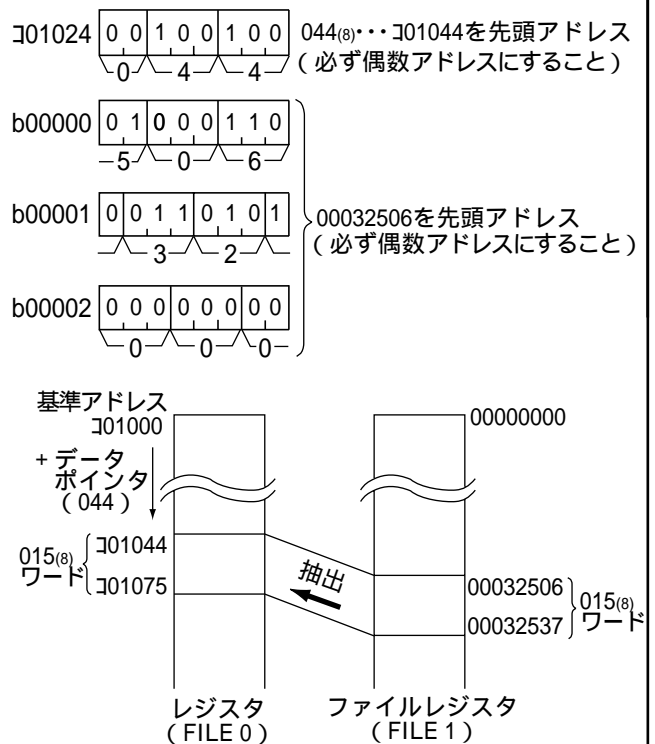
(類似命令) F-06、F-06w、F-06d、F-73、F-73d

[使用例]



命 令	
STR	004001
F-73w	015
	b00000
	J01024

入力条件004001がOFF ONの変化時に、レジスタb00000~b00002の内容でアドレス指定するファイルレジスタ(FILE 1)を先頭とする015⁽⁸⁾ワードの内容を、J01000(レジスタJ01024を含む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J01024の内容(データポイント)だけ変位したレジスタを先頭とする015⁽⁸⁾ワードに転送する。



レジスタ側の基準アドレスはJ00000、J00400、J01000、.....、019000、019400、.....で、Sを含むデータメモリのブロックの先頭アドレスとなります。

データメモリのブロックと基準アドレス(9・26ページ)参照

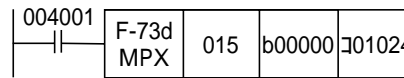
**F-73d
MPX**

**ファイルレジスタ(FILE 1)からのnダブルワード抽出
(MultiPleXer)**

シンボル	F-73d MPX n S D
機能	レジスタS~S+2の内容でアドレス指定するファイルレジスタ(ファイル番号1)を先頭とするnダブルワードの内容を、レジスタDを含む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ダブルワードに転送する。
演算内容	S~S+2、.....、S~S+2+4n-1 → X+D、.....、X+D+4n-1 X: Dを含むデータメモリブロックの先頭アドレス(基準アドレス) D: データポインタ
n	使用範囲 000~377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256ダブルワード)
S	使用範囲 C ・Sには必ず偶数アドレスを設定してください。
D	使用範囲 B ・Dには必ず偶数アドレスを設定してください。
演算条件	入力信号の立上り(OFF ON)
演算後の内容	ファイル1のレジスタ 不変 X+D ファイル1のS~S+2の内容 X+D+1 ファイル1のS~S+2+1の内容 ⋮ ⋮ X+D+4n-1 ファイル1のS~S+2+4n-1の内容 フラグ 不変

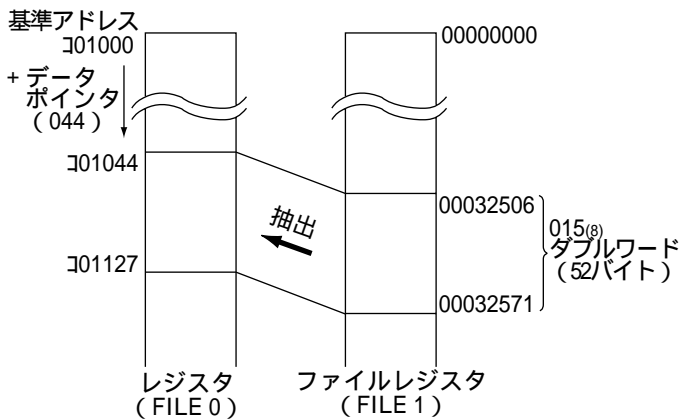
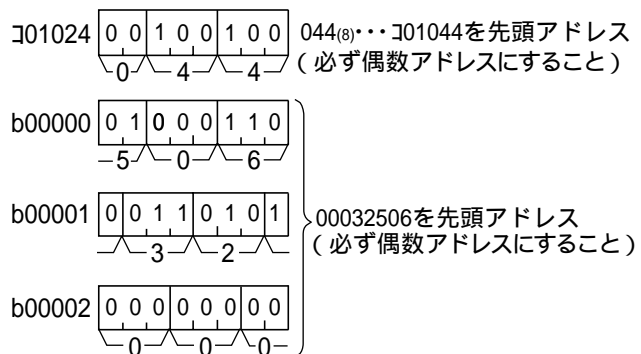
(類似命令) F-06、F-06w、F-06d、F-73、F-73w

[使用例]



命 令	
STR	004001
F-73d	015
	b00000
	01024

入力条件004001がOFF ONの変化時に、レジスタb00000~b00002の内容でアドレス指定するファイルレジスタ(FILE 1)を先頭とする015⁽⁸⁾ダブルワードの内容を、010100(レジスタ01024を含む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01024の内容(データポイント)だけ変位したレジスタを先頭とする015⁽⁸⁾ダブルワードに転送する。



レジスタ側の基準アドレスは00000、00400、01000、.....、01900、01940、.....で、Sを含むデータメモリのブロックの先頭アドレスとなります。

データメモリのブロックと基準アドレス(9・26ページ)参照

F-74 nXFR

nバイト転送

シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">F-74 nXFR</td> <td style="padding: 2px;">n</td> <td style="padding: 2px;">S</td> <td style="padding: 2px;">D</td> </tr> </table>	F-74 nXFR	n	S	D	<p>[使用例]</p> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="margin-right: 20px;"> <table border="1" style="border-collapse: collapse;"> <tr> <td style="padding: 2px;">004000</td> <td style="padding: 2px;">F-74 nXFR</td> <td style="padding: 2px;">010</td> <td style="padding: 2px;">009013</td> <td style="padding: 2px;">019416</td> </tr> </table> </div> <div> <table border="1" style="border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">命 令</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">STR</td> <td style="padding: 2px;">004000</td> </tr> <tr> <td style="padding: 2px;">F-74</td> <td style="padding: 2px;">010</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">009013</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">019416</td> </tr> </tbody> </table> </div> </div> <p>入力条件004000がOFF ONの変化時に、レジスタ019416を先頭とする010₍₈₎バイトのレジスタに、レジスタ009013の内容を転送します。</p> <div style="margin-top: 20px;"> <table style="border-collapse: collapse;"> <tr> <td style="padding: 2px;">009013</td> <td style="padding: 2px;">1 0 0 1 0 0 1 0</td> <td style="padding: 2px;">→</td> <td style="padding: 2px;">1 0 0 1 0 0 1 0</td> <td style="padding: 2px;">019416</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="padding: 2px;">1 0 0 1 0 0 1 0</td> <td style="padding: 2px;">019417</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="padding: 2px;">≈</td> <td style="padding: 2px;">≈</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="padding: 2px;">1 0 0 1 0 0 1 0</td> <td style="padding: 2px;">019424</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="padding: 2px;">1 0 0 1 0 0 1 0</td> <td style="padding: 2px;">019425</td> </tr> </table> <p style="text-align: center; margin-top: 10px;">010₍₈₎バイト</p> </div>	004000	F-74 nXFR	010	009013	019416	命 令		STR	004000	F-74	010		009013		019416	009013	1 0 0 1 0 0 1 0	→	1 0 0 1 0 0 1 0	019416				1 0 0 1 0 0 1 0	019417				≈	≈				1 0 0 1 0 0 1 0	019424				1 0 0 1 0 0 1 0	019425
F-74 nXFR	n	S	D																																											
004000	F-74 nXFR	010	009013	019416																																										
命 令																																														
STR	004000																																													
F-74	010																																													
	009013																																													
	019416																																													
009013	1 0 0 1 0 0 1 0	→	1 0 0 1 0 0 1 0	019416																																										
			1 0 0 1 0 0 1 0	019417																																										
			≈	≈																																										
			1 0 0 1 0 0 1 0	019424																																										
			1 0 0 1 0 0 1 0	019425																																										
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+1, ……、D+n-1																																												
n		使用範囲 000 ~ 377 ₍₈₎ (000のとき256バイト)																																												
S		使用範囲 A																																												
D		使用範囲 A																																												
演算条件		入力信号の立上り(OFF ON)																																												
演算後の内容	S	不変																																												
	D	レジスタSの内容																																												
	D+1																																													
	⋮																																													
D+2n-1																																														
フラグ		不変																																												

(類似命令) F-00、F-00w、F-00d、F-70、F-70w、F-70d、F-74w、F-74d、F-76、F-76w、F-76d

F-74w nXFR

nワード転送 (nXFR)

シンボル	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="padding: 2px;">F-74w nXFR</td> <td style="padding: 2px;">n</td> <td style="padding: 2px;">S</td> <td style="padding: 2px;">D</td> </tr> </table>	F-74w nXFR	n	S	D	<p>[使用例]</p> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="margin-right: 20px;"> <table border="1" style="border-collapse: collapse;"> <tr> <td style="padding: 2px;">004000</td> <td style="padding: 2px;">F-74w nXFR</td> <td style="padding: 2px;">010</td> <td style="padding: 2px;">009014</td> <td style="padding: 2px;">019416</td> </tr> </table> </div> <div> <table border="1" style="border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">命 令</th> </tr> </thead> <tbody> <tr> <td style="padding: 2px;">STR</td> <td style="padding: 2px;">004000</td> </tr> <tr> <td style="padding: 2px;">F-74w</td> <td style="padding: 2px;">010</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">009014</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">019416</td> </tr> </tbody> </table> </div> </div> <p>入力条件004000がOFF ONの変化時に、レジスタ019416、019417を先頭とする010₍₈₎ワードのレジスタに、レジスタ009014、009015の内容を転送します。</p> <div style="margin-top: 20px;"> <table style="border-collapse: collapse;"> <tr> <td style="padding: 2px;">009015</td> <td style="padding: 2px;">009014</td> <td style="padding: 2px;">→</td> <td style="padding: 2px;">019417</td> <td style="padding: 2px;">019416</td> </tr> <tr> <td style="padding: 2px;">6</td> <td style="padding: 2px;">7</td> <td style="padding: 2px;">8</td> <td style="padding: 2px;">9</td> <td style="padding: 2px;">6</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td style="padding: 2px;">7</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td style="padding: 2px;">8</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td style="padding: 2px;">9</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td style="padding: 2px;">≈</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td style="padding: 2px;">≈</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td style="padding: 2px;">6</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td style="padding: 2px;">7</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td style="padding: 2px;">8</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td style="padding: 2px;">9</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td style="padding: 2px;">019435</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td style="padding: 2px;">019434</td> </tr> </table> <p style="text-align: center; margin-top: 10px;">010₍₈₎ワード (16バイト)</p> </div>	004000	F-74w nXFR	010	009014	019416	命 令		STR	004000	F-74w	010		009014		019416	009015	009014	→	019417	019416	6	7	8	9	6					7					8					9					≈					≈					6					7					8					9					019435					019434
F-74w nXFR	n	S	D																																																																																			
004000	F-74w nXFR	010	009014	019416																																																																																		
命 令																																																																																						
STR	004000																																																																																					
F-74w	010																																																																																					
	009014																																																																																					
	019416																																																																																					
009015	009014	→	019417	019416																																																																																		
6	7	8	9	6																																																																																		
				7																																																																																		
				8																																																																																		
				9																																																																																		
				≈																																																																																		
				≈																																																																																		
				6																																																																																		
				7																																																																																		
				8																																																																																		
				9																																																																																		
				019435																																																																																		
				019434																																																																																		
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		使用範囲 000 ~ 377 ₍₈₎ (000のとき256ワード)																																																																																				
S		使用範囲 B																																																																																				
D		使用範囲 B																																																																																				
演算条件		入力信号の立上り(OFF ON)																																																																																				
演算後の内容	S, S+1	不変																																																																																				
	D	レジスタSの内容																																																																																				
	D+1	レジスタS+1の内容																																																																																				
	⋮	⋮																																																																																				
	D+2n-2	レジスタSの内容																																																																																				
D+2n-1	レジスタS+1の内容																																																																																					
フラグ		不変																																																																																				

S, Dには必ず偶数アドレスを設定してください。

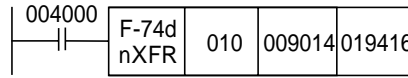
(類似命令) F-00、F-00w、F-00d、F-70、F-70w、F-70d、F-74、F-74d、F-76、F-76w、F-76d

**F-74d
nXFR**

**nダブルワード転送
(nXFR)**

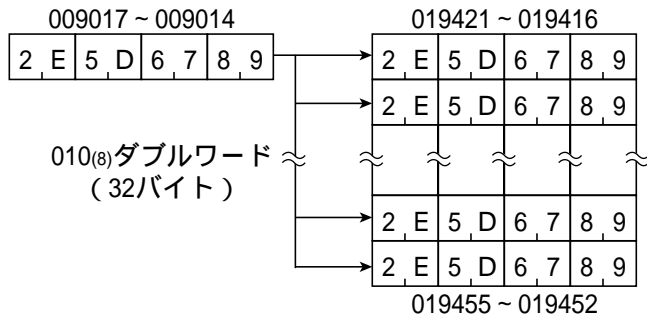
シンボル	<table border="1"><tr><td>F-74d</td><td>n</td><td>S</td><td>D</td></tr></table>	F-74d	n	S	D
F-74d	n	S	D		
機能	レジスタD ~ D+3を先頭とするnダブルワードのレジスタに、レジスタS ~ S+3の内容を転送する。				
演算内容	S ~ S+3 → D ~ D+3, ……、D+4n-4 ~ D+4n-1				
n	使用範囲 000 ~ 377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256ダブルワード)				
S	使用範囲 C				
D	使用範囲 C				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S ~ S+3 不変				
	D ~ D+3 レジスタS ~ S+3の内容				
	⋮				
	D+4n-3 ~ D+4n-1 レジスタS ~ S+3の内容				
フラグ	不変				

[使用例]



命 令	
STR	004000
F-74d	010 009014 019416

入力条件004000がOFF ONの変化時に、レジスタ019416 ~ 019421を先頭とする010⁽⁸⁾ダブルワードのレジスタに、レジスタ009014 ~ 009017の内容を転送します。



S、Dには必ず偶数アドレスを設定してください。

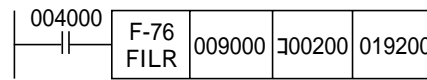
(類似命令) F-00、F-00w、F-00d、F-70、F-70w、F-70d、F-74、F-74w、F-76、F-76w、F-76d

**F-76
FILR**

**nバイト一括転送
(FILR)**

シンボル	F-76 FILR S1 S2 D		
機能	レジスタS2からS2+(S1)-1のデータを、レジスタDからD+(S1)-1に一括転送する。 ・(S1)はレジスタS1の内容で指定するバイト数。		
演算内容	S2, S2+1, ……、S2+(S1)-1 → D, D+1, ……、D+(S1)-1		
S1	使用範囲 A ・レジスタS1に設定する内容は000 ~ 377 ₍₈₎ です。(000 ₍₈₎ のとき256バイト)		
S2	使用範囲 A		
D	使用範囲 A		
演算条件	入力信号の立上り(OFF ON)		
演算後の内容	S1	不変	
	S2, ……、S2+(S1)-1	不変	
	D	レジスタS2の内容	
	D+1	レジスタS2+1の内容	
	⋮	⋮	
D+(S1)-2	レジスタS2+(S1)-2の内容		
D+(S1)-1	レジスタS2+(S1)-1の内容		
フラグ	不変		

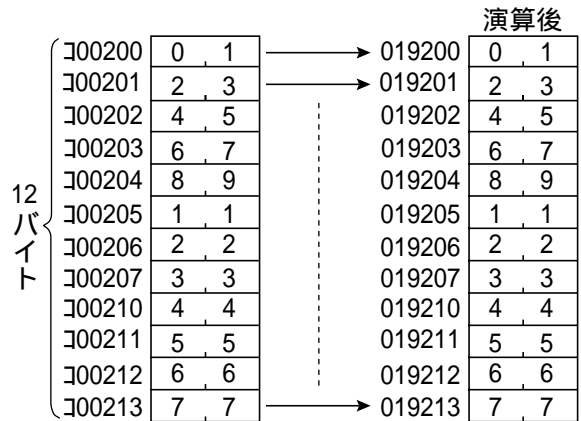
[使用例]



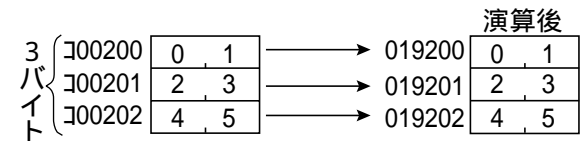
命 令	
STR	004000
F-76	009000 00200 019200

入力条件004000がOFF ONの変化時に、レジスタ00200から、レジスタ009000の内容のバイト分を、レジスタ019200から始まる領域に一括転送する。

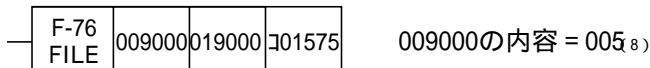
009000の内容が012_(D)(014₍₈₎)バイトのとき



009000の内容が003_(D)(003₍₈₎)バイトのとき

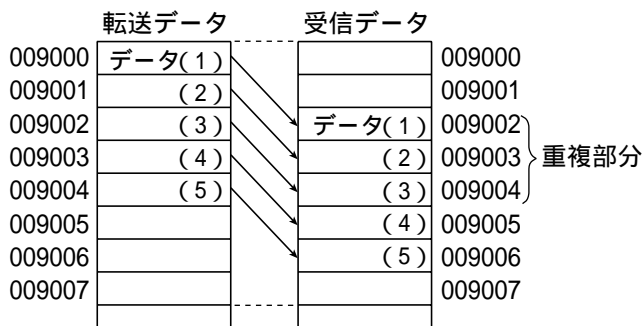
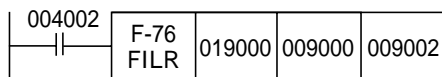


タイマ・カウンタの接点領域(ファイルアドレス00001600 ~ 00001777₍₈₎等)には転送しないでください。



上記のプログラムでは、00001600、00001601に019003、019004の内容が転送されてしまいます。

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。

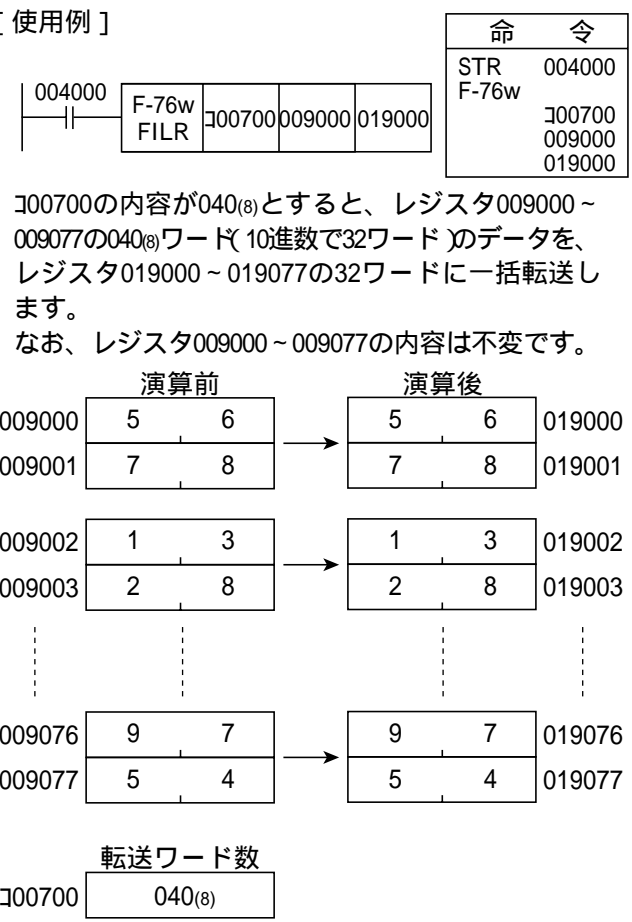


(類似命令) F-00、F-00w、F-00d、F-70、F-70w、F-70d、F-74、F-74w、F-74d、F-76w、F-76d

**F-76w
FILR**

**nワード一括転送
(FILR)**

シンボル	<table border="1"> <tr> <td>F-76w FILR</td> <td>S1</td> <td>S2</td> <td>D</td> </tr> </table>				F-76w FILR	S1	S2	D	[使用例]	<table border="1"> <tr> <th colspan="2">命 令</th> </tr> <tr> <td>STR</td> <td>004000</td> </tr> <tr> <td>F-76w</td> <td>00700</td> </tr> <tr> <td></td> <td>009000</td> </tr> <tr> <td></td> <td>019000</td> </tr> </table>	命 令		STR	004000	F-76w	00700		009000		019000
F-76w FILR	S1	S2	D																	
命 令																				
STR	004000																			
F-76w	00700																			
	009000																			
	019000																			
機 能	レジスタS2からS2+ $\alpha(S1)-1$ のデータ(ワード数: S1の内容)を、レジスタDからD+ $\alpha(S1)-1$ に一括転送する。																			
演算内容	S2, S2+1, ……、S2+ $\alpha(S1)-1$ → D, D+1, ……、D+ $\alpha(S1)-1$																			
S1	使用範囲 A ・レジスタS1に設定する内容は000 ~ 377 ₍₈₎ です。(000 ₍₈₎ のとき256ワード)																			
S2	使用範囲 B ・S2には必ず偶数アドレスを設定してください。																			
D	使用範囲 B ・Dには必ず偶数アドレスを設定してください。																			
演算条件	入力信号の立上り(OFF ON)																			
演算後の内容	S1	不変																		
	S2, …, S2 + $\alpha(S1) - 1$	不変																		
	D	レジスタS2の内容																		
	D+1	レジスタS2+1の内容																		
	⋮	⋮																		
	D+ $\alpha(S1)-2$ D+ $\alpha(S1)-1$	レジスタS2+ $\alpha(S1)-2$ の内容 レジスタS2+ $\alpha(S1)-1$ の内容																		
フ ラ グ	不変																			



(類似命令) F-00、F-00w、F-00d、F-70、F-70w、F-70d、F-74、F-74w、F-74d、F-76、F-76d

F-76d nダブルワード一括転送
FILR (FILR)

シンボル	<table border="1"> <tr> <td>F-76d FILR</td> <td>S1</td> <td>S2</td> <td>D</td> </tr> </table>			F-76d FILR	S1	S2	D	[使用例]	<table border="1"> <tr> <th colspan="2">命 令</th> </tr> <tr> <td>STR</td> <td>004000</td> </tr> <tr> <td>F-76d</td> <td>100700</td> </tr> <tr> <td></td> <td>009000</td> </tr> <tr> <td></td> <td>019000</td> </tr> </table>	命 令		STR	004000	F-76d	100700		009000		019000																									
F-76d FILR	S1	S2	D																																									
命 令																																												
STR	004000																																											
F-76d	100700																																											
	009000																																											
	019000																																											
機 能	レジスタS2からS2+4(S1)-1のデータを、レジスタDからD+4(S1)-1に一括転送する。 ・(S1)はレジスタS1の内容で指定するダブルワード数。				<p>100700の内容が040⁽⁸⁾とすると、レジスタ009000～009177の040⁽⁸⁾ダブルワード(128バイト)のデータを、レジスタ019000～019177(128バイト)に一括転送します。</p> <p>なお、レジスタ009000～009177の内容は不変です。</p>																																							
演算内容	$S2 \sim S2+3, \dots, S2+4(S1)-4 \sim S2+4(S1)-1$ $\rightarrow D \sim D+3, \dots, D+4(S1)-4 \sim D+4(S1)-1$			<p>演算前</p> <table border="1"> <tr><td>009000</td><td>5</td><td>6</td><td></td><td></td></tr> <tr><td>009001</td><td>7</td><td>8</td><td></td><td></td></tr> <tr><td>009002</td><td>1</td><td>3</td><td></td><td></td></tr> <tr><td>009003</td><td>2</td><td>8</td><td></td><td></td></tr> </table> <p>演算後</p> <table border="1"> <tr><td>009000</td><td>5</td><td>6</td><td></td><td>019000</td></tr> <tr><td>009001</td><td>7</td><td>8</td><td></td><td>019001</td></tr> <tr><td>009002</td><td>1</td><td>3</td><td></td><td>019002</td></tr> <tr><td>009003</td><td>2</td><td>8</td><td></td><td>019003</td></tr> </table>	009000	5	6			009001	7	8			009002	1	3			009003	2	8			009000	5	6		019000	009001	7	8		019001	009002	1	3		019002	009003	2	8		019003
009000	5	6																																										
009001	7	8																																										
009002	1	3																																										
009003	2	8																																										
009000	5	6		019000																																								
009001	7	8		019001																																								
009002	1	3		019002																																								
009003	2	8		019003																																								
S1	使用範囲 A ・レジスタS1に設定する内容は000～377 ⁽⁸⁾ です。 (000 ⁽⁸⁾ のとき256ダブルワード)			<table border="1"> <tr><td>009004</td><td>A</td><td>B</td><td></td><td>019004</td></tr> <tr><td>009005</td><td>C</td><td>D</td><td></td><td>019005</td></tr> <tr><td>009006</td><td>E</td><td>F</td><td></td><td>019006</td></tr> <tr><td>009007</td><td>1</td><td>2</td><td></td><td>019007</td></tr> </table>	009004	A	B		019004	009005	C	D		019005	009006	E	F		019006	009007	1	2		019007																				
009004	A	B		019004																																								
009005	C	D		019005																																								
009006	E	F		019006																																								
009007	1	2		019007																																								
S2	使用範囲 C ・S2には必ず偶数アドレスを設定してください。			<table border="1"> <tr><td>009004</td><td>A</td><td>B</td><td></td><td>019004</td></tr> <tr><td>009005</td><td>C</td><td>D</td><td></td><td>019005</td></tr> <tr><td>009006</td><td>E</td><td>F</td><td></td><td>019006</td></tr> <tr><td>009007</td><td>1</td><td>2</td><td></td><td>019007</td></tr> </table>	009004	A	B		019004	009005	C	D		019005	009006	E	F		019006	009007	1	2		019007																				
009004	A	B		019004																																								
009005	C	D		019005																																								
009006	E	F		019006																																								
009007	1	2		019007																																								
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。			<table border="1"> <tr><td>009174</td><td>5</td><td>6</td><td></td><td>019174</td></tr> <tr><td>009175</td><td>8</td><td>7</td><td></td><td>019175</td></tr> <tr><td>009176</td><td>9</td><td>7</td><td></td><td>019176</td></tr> <tr><td>009177</td><td>5</td><td>4</td><td></td><td>019177</td></tr> </table>	009174	5	6		019174	009175	8	7		019175	009176	9	7		019176	009177	5	4		019177																				
009174	5	6		019174																																								
009175	8	7		019175																																								
009176	9	7		019176																																								
009177	5	4		019177																																								
演算条件	入力信号の立上り(OFF ON)																																											
演算後の内容	S1	不変																																										
	S2, ..., S2+4(S1)-1	不変																																										
	D ~ D+3	レジスタS2 ~ S2+3の内容																																										
	D+4 ~ D+7	レジスタS2+4 ~ S2+7の内容																																										
	⋮	⋮																																										
D+4n-4 ~ D+4n-1	レジスタS2+4(S1)-4 ~ S2+4(S1)-1の内容																																											
フ ラ グ	不変			<p>転送ダブルワード数</p> <table border="1"> <tr> <td>100700</td> <td>040⁽⁸⁾</td> </tr> </table>	100700	040 ⁽⁸⁾																																						
100700	040 ⁽⁸⁾																																											

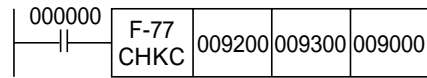
(類似命令) F-00、F-00w、F-00d、F-70、F-70w、F-70d、F-74、F-74w、F-74d、F-76、F-76w

**F-77
CHKC**

データサムチェックコードの生成
(CHeck Code)

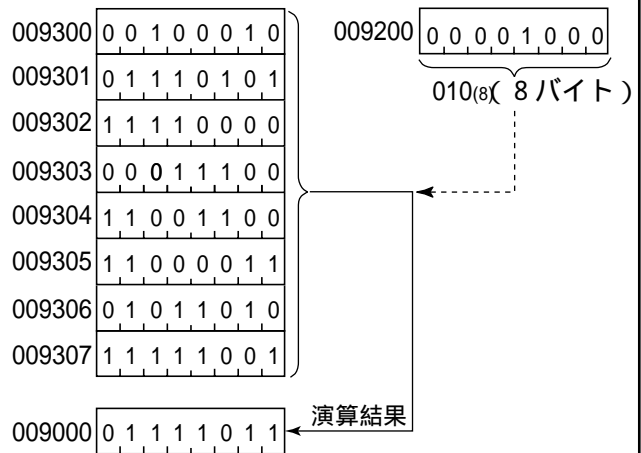
シンボル	$\overline{\text{F-77}} \text{CHCK} \quad \text{S1} \quad \text{S2} \quad \text{D}$	
機能	レジスタS2からS2+(S1)-1まで(バイト数:レジスタS1の内容)のサムチェックコードを作成し、レジスタDに格納する。	
演算内容	$0 \cdot (\text{S2} \sim \text{S2}+(\text{S1})-1) \rightarrow \text{D}$	
S1	使用範囲 A ・レジスタS1に設定する内容は、000 ~ 377 ₍₈₎ です。(000のとき256バイト)	
S2	使用範囲 A	
D	使用範囲 A	
演算条件	入力信号の立上り(OFF ON)	
演算後の内容	S1	不変
	S2 ~ S2+n-1	不変
	D	演算結果
	フラグ	不変

[使用例]



命 令	
STR	000000
F-77	009200 009300 009000

入力条件000000がOFF ONの変化時に、レジスタ009300から「009200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009000に格納します。
レジスタ009200の値が8バイトのとき、レジスタ009300~009307のサムチェックコードを生成し、レジスタ009000へ格納します。



サムチェックコードの求め方

```

    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))
  
```

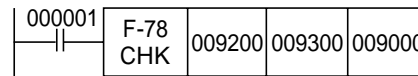
よって、サムチェック値は7B(H)です。

**F-78
CHK**

**データのチェック
(CHekK)**

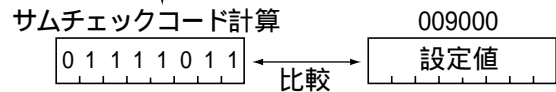
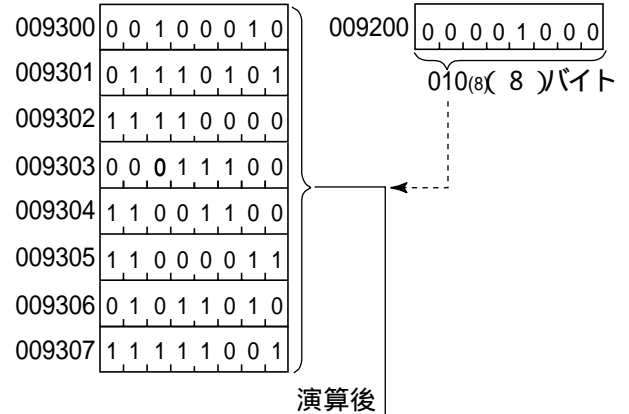
シンボル	— F-78 CHK S1 S2 S3				
機能	レジスタS2からS2+(S1)-1までのサムチェックコードを作成し、F-77で作成済のサムチェックコードを格納しているレジスタS3と比較し、フラグを変化させる。 (S1)はレジスタS1の内容で指定するバイト数。				
演算内容	$[0 - (S2 \sim S2+(S1)-1)] \xrightarrow{\text{比較}} S3$ → フラグ				
S1	使用範囲 A ・レジスタS1に設定する内容は、000 ~ 377 ₍₈₎ です。(000 ₍₈₎ のとき256バイト)				
S2	使用範囲 A				
S3	使用範囲 A				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S1	不変			
	S2	不変			
	S3	不変			
フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	エラー無し サムチェックエラー	0 0	0 0	0 1	0 0

[使用例]



命 令	
STR	000001
F-78	009200
	009300
	009000

入力条件000001がOFF ONの変化時に、レジスタ009300から指定バイト(009200の内容)分のサムチェックを計算し、F-77で作成済のサムチェックコードを格納しているレジスタ009000と比較します。



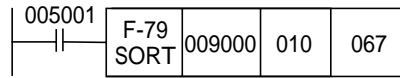
比較の結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
同じ	0	0	0	0
異なる	0	0	1	0

**F-79
SORT**

レジスタ(1バイト)データの並べ替え

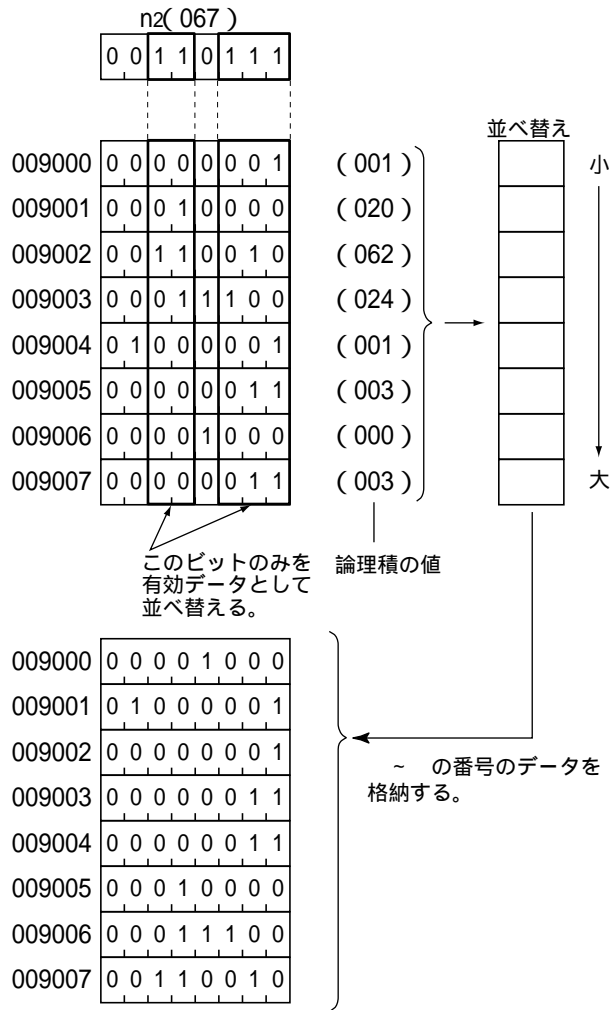
シンボル	— F-79 SORT S n1 n2			
機能	レジスタSからS+n1-1までの内容(1バイト毎)を、小さい方から順に並べ替える。 n2は比較データのマスク値です。			
演算内容	S < S+1 < …… < S+n1-2 < S+n1-1			
S	使用範囲 A			
n1	使用範囲 000 ~ 377 ⁽⁸⁾ (000のとき256バイト)			
n2	使用範囲 000 ~ 377 ⁽⁸⁾ ・ n2が000、377のときは全ビットが有効となります。			
演算条件	入力信号の立上り(OFF ON)			
演算内容 後	S~S+n1-1	演算結果(1バイト毎に、小さい方からの順)		
	フラグ	不変		

[使用例]



命 令	
STR	005001
F-79	009000 010 067

入力条件005001がOFF ONの変化時に、レジスタ009000~009007の8バイト(n1)を、n2の067(8)で有効ビットを論理積(AND)し、その結果で大小比較して、小さい順にレジスタ009000から並べます。



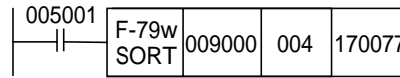
データそのものは変化しません。

**F-79w
SORT**

レジスタ(1ワード)データの並べ替え

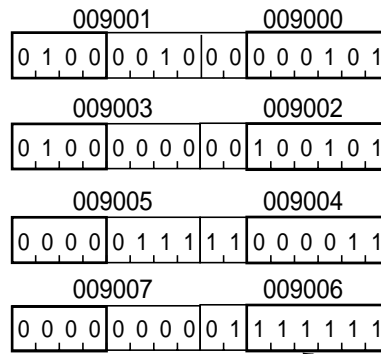
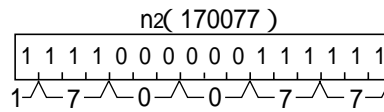
シンボル	$\overline{\text{F-79w SORT}}$ S n1 n2
機能	レジスタS、S+1からS+2n1-2、S+2n1-1までの内容(1ワード毎)を、小さい方から順に並べ替える。 n2は比較データのマスク値です。
演算内容	S, S+1 < < S+2n1-2, S+2n1-1
S	使用範囲 B ・Sには必ず偶数アドレスを設定してください。(019003等は禁止)
n1	使用範囲 000 ~ 377 ⁽⁸⁾ (000のとき256ワード)
n2	使用範囲 000000 ~ 177777 ⁽⁸⁾ ・n2が000000、177777のときは全ビットが有効となります。
演算条件	入力信号の立上り(OFF ON)
演算後の内容	S ~ S+2n1-1 演算結果(1ワード毎に、小さい方からの順) フラグ 不変

[使用例]

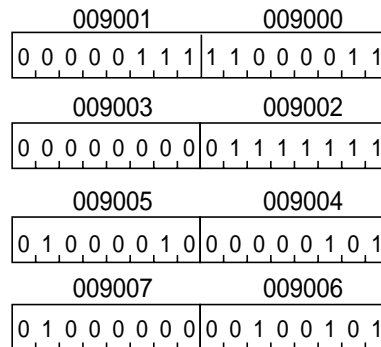


命令	
STR	005001
F-79w	009000
	004
	170077

入力条件005001がOFF ONの変化時に、レジスタ009000 ~ 009007の4ワード(n1)を、n2の170077(8)で有効ビットを論理積(AND)し、その結果で大小比較して、小さい順にレジスタ009000、009001から並べます。



このビットのみを有効データとして並べ替える。



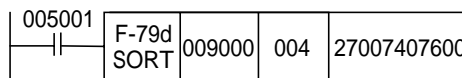
データそのものは変化しません。

F-79d
SORT

レジスタ(2ワード)データの並べ替え

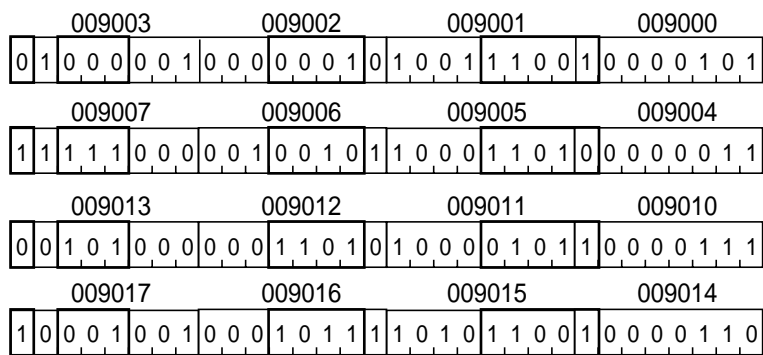
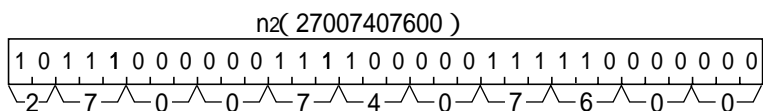
シンボル	F-79d SORT	S	n1	n2
機能	レジスタS~S+3からS+4n1-4~S+4n1-1までの内容(2ワード毎)を、小さい方から順に並べ替える。 n2は比較データのマスク値です。			
演算内容	S~S+3 <.....< S+4n1-4~S+4n1-1			
S	使用範囲C ・Sには必ず偶数アドレスを設定してください。(019003等は禁止)			
n1	使用範囲 000~377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256ダブルワード)			
n2	使用範囲 000000000000~3777777777 ⁽⁸⁾ ・n2が000000000000、3777777777 ⁽⁸⁾ のときは全ビットが有効となります。			
演算条件	入力信号の立上り(OFF ON)			
演算内容 後内容	S~S+4n1-1	演算結果(2ワード毎に、小さい方からの順)		
	フラグ	不変		

[使用例]

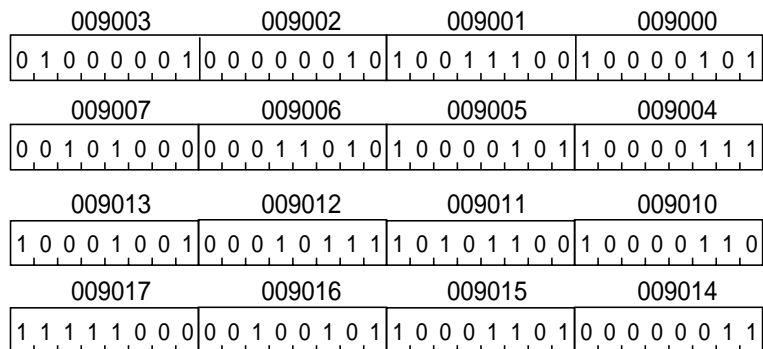


命 令	
STR	005001
F-79d	009000
	004
	27007407600

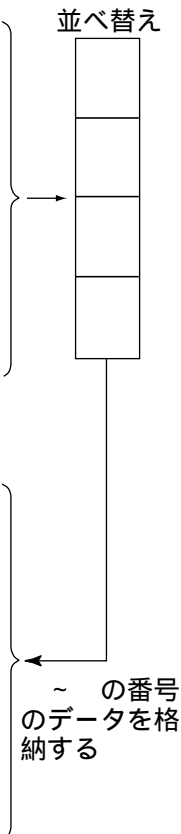
入力条件005001がOFF ONの変化時に、レジスタ009000~009017の4ダブルワード(n1)を、n2の27007407600⁽⁸⁾で有効ビットを論理積(AND)し、その結果で大小比較して、小さい順にレジスタ009000~009003から並べます。



このビットのみを有効データとして並べ替える



データそのものは変化しません。



第 13 章 応用命令 (F-80 ~ F-173d)

**F-80
IORF**

I / Oリフレッシュ
(I/O ReFresh)

シンボル	F-80 IORF RACK・SLOT					
機能	RACK(ラック)番号とSLOT(スロット)番号で指定する入出力ユニットのすべてのデータと、PLCデータメモリ間でデータを交換する。					
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					
RACK	使用範囲 0 ~ 7					
SLOT	使用範囲 0 ~ 7					
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)					
演算後の内容	入力ユニット	データメモリの更新		・バス異常のとき、データメモリまた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 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		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		

[使用例]

004000	
└──┘	

F-80	RACK・SLOT
IORF	0 2

命 令	
STR	04000
F-80	
R	0
S	2

入力条件004000がONのときラック番号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を交換します。

- ・本命令は、プログラム演算中に何回でも使用できます。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は、プログラマ(JW-15PG)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 プログラマの取扱説明書を参照)
- ・入力割込用(システムメモリ#0240~#0243で設定)で使っ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		[使用例]									
機能	SW(ラック番号とユニットNo.スイッチで決まる番号)で指定する特殊 I / O ユニットと、PLCのデータメモリ間でデータメモリ(16バイト)と制御リレーの交換を行う。				<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004000</td></tr> <tr><td>F-82</td><td>SW23</td></tr> </table>		命 令		STR	004000	F-82	SW23
命 令												
STR	004000											
F-82	SW23											
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		<p>入力条件004000がONのとき、ラック番号 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 PLCのデータをリフレッシュします。</p>									
SW	設定範囲 00 ~ 77 ⁽⁸⁾ (上位桁：ラック番号(0 ~ 7) 下位桁：ユニットNo.スイッチ(0 ~ 7))		<p>SW23で指定する特殊ユニットのデータ I03460 ~ I03477</p> <p>128点(16バイト) ← 特殊ユニット用リレー</p>									
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)											
演算後の内容 フラグ	特殊入力ユニット	データメモリの更新	バス異常または特殊 I / O 異常のとき、データメモリおよ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										
	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、またはバス異常	007357	007356	007355	007354							
	転送完了	0	0	1	0							
転送データなし、または特殊 I / O ユニットなし	0	0	0	0	1							
非実行時	0	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。
 本命令は、特殊 I / O ユニットと特殊 I / O ユニット用リレーのデータ交換用で、転送データバイト数は16バイトです。
 本命令でデータをリフレッシュする特殊 I / O ユニットは、通常の I / O サイクルでもデータをリフレッシュします。
 特殊 I / O ユニットのデータ変換が完了していないときに本命令を実行すると、キャリーフラグ(007354)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、エラーフラグが ON します。
 非実行時とは入力条件 OFF のときです。すべてのフラグが 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 ユニットはリフレッシュできません。

**F-85
PRRD**

特殊I/Oからの読出し

**F-86
PRWD**

特殊I/Oへの書込み

シンボル	F-85 PRRD				F-86 PRWD				
	n1	SW, n2	D		n1	D	SW, n2		
機能	SW(ラック番号とユニットNo.スイッチで決まる番号)で指定する特殊I/Oユニットの特殊I/O専用F命令領域(n2)のバイト数(n1)を、レジスタDを先頭とするD+n1-1に読み出す。				レジスタDを先頭とするn1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)で指定する特殊I/Oユニットの特殊I/O専用F命令領域(n2)へ転送する。				
n1	使用範囲 000 ~ 377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256バイト)				使用範囲 000 ~ 377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256バイト)				
SW	使用範囲 00 ~ 77 ⁽⁸⁾ (上位桁:ラック番号(0~7) 下位桁:ユニットNo.スイッチ(0~7))				使用範囲 00 ~ 77 ⁽⁸⁾ (上位桁:ラック番号(0~7) 下位桁:ユニットNo.スイッチ(0~7))				
n2	特殊I/O専用F命令領域 0 ~ 3				特殊I/O専用F命令領域 0 ~ 3				
D	使用範囲 A				使用範囲 A				
演算条件	入力信号の立上り(OFF ON)				入力信号の立上り(OFF ON)				
演算後の内容	n1	不変				不変			
	SW	不変				不変			
	n2	不変				不変			
	D ~ D+n1-1	n2の内容				不変			
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	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				
転送完了したとき		0	1	0	0				
上記以外	0	0	0	0					

F-85、F-86は、特殊I/Oユニット(JW-21SU、JW-21PS、JW-21DU/22DU)に限り使用する命令です。本命令を必要とするユニット以外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。

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	<table border="1"> <tr> <td>F-90 REM</td> <td>n</td> </tr> </table>	F-90 REM	n	<p>[解説]</p> <ul style="list-style-type: none"> ・ F-90用のシンボル・コメントは、JW-300SP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) ・ ラダー印字のとき、シンボル・コメント内容を印字し、F-90内容は印字しません。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。 ・ 命令語印字のとき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。
F-90 REM	n			
機能	ラダー設計支援ソフト(JW-300SP)でラダー・命令語を印字のとき、行コメントを印字する。			
演算内容	演算しない(NOPと同じ)			
n	使用範囲 0000 ~ 3777 ⁽⁸⁾			
演算後の内容	フラグ等データメモリは不変			

[使用例]

<p>ラダープログラミング</p>	<p>シンボル・コメント設定</p> <table border="1"> <thead> <tr> <th>アドレス</th> <th>シンボル</th> <th>コメント</th> </tr> </thead> <tbody> <tr> <td>F-90 0010</td> <td>No.10</td> <td>異常処理監視部</td> </tr> <tr> <td>F-90 0011</td> <td>@</td> <td></td> </tr> </tbody> </table> <p>↑ ラダー印字のとき、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	@									



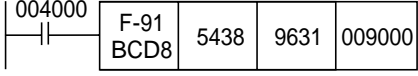
<p>ラダー印字</p>	<p>命令語印字</p> <pre> STR 000000 OUT 000400 F-90[REM] 0010 No.10 : 異常処理監視部 STR 000100 OR 000101 AND 000102 OUT 000401 </pre>
---------------------	---

**F-91
BCD8**

BCD定数(8桁)の転送

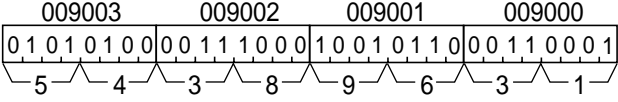
シンボル	$\overline{\text{F-91 BCD8}}$ n1 n2 D														
機能	8桁のBCD定数n1、n2(n1は上位4桁、n2は下位4桁)を、レジスタD～D+3に転送する。														
演算内容	n1 → D+3、D+2 n2 → D+1、D														
n1	使用範囲 0000～9999 (FFFF(H)まで可)														
n2	使用範囲 0000～9999 (FFFF(H)まで可)														
D	使用範囲 C														
演算条件	入力信号の立上り(OFF ON)														
演算後の内容	演算結果 <table border="1"> <tr> <td>D</td> <td>10⁷</td> <td>10⁶</td> <td rowspan="2">} n2</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> <td rowspan="2">} n1</td> </tr> <tr> <td>D+3</td> <td>10⁷</td> <td>10⁶</td> </tr> </table>	D	10 ⁷	10 ⁶	} n2	D+1	10 ³	10 ²	D+2	10 ⁵	10 ⁴	} n1	D+3	10 ⁷	10 ⁶
	D	10 ⁷	10 ⁶	} n2											
D+1	10 ³	10 ²													
D+2	10 ⁵	10 ⁴	} n1												
D+3	10 ⁷	10 ⁶													
フラグ	不変														

[使用例]



命 令	
STR	004000
F-91	5438 9631 009000

入力条件004000がOFF ONの変化時に、BCD定数54389631をレジスタ009000～009003に転送します。転送後、BCDコードで次の数値となります。



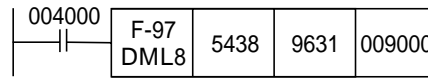
(類似命令) F-01、F-01w、F-01d

**F-97
DML8**

10進定数(8桁)の転送

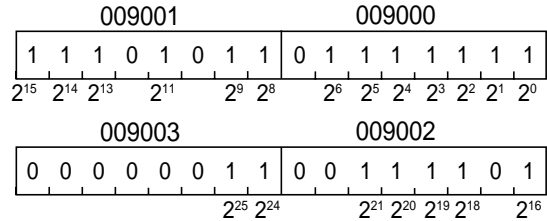
シンボル	F-97 DML8	n1	n2	D
機能	8桁の10進定数n1、n2(n1 × 10000 + n2) を、レジスタD ~ D+3に転送する。			
演算内容	n1 × 10000 + n2 → D ~ D+3			
n1	使用範囲 0000 ~ 9999			
n2	使用範囲 0000 ~ 9999			
D	使用範囲 C			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	D ~ D+3	演算結果 (10進定数 : 00000000 ~ 99999999)		
	フラグ	不変		

[使用例]



命 令	
STR	004000
F-97	5438 9631 009000

入力条件004000がOFF ONの変化時に、10進定数54389631をレジスタ009000 ~ 009003に転送します。転送後、バイナリコードで次の数値となります。



$$1+2+4+8+16+32+64+256+512+2048+8192+16384+32768+65536+262144+524288+1048576+2097152+16777216+33554432 = 54389631$$

バイナリコードの各ビットの重みは下記のとおりです。

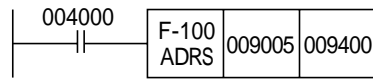
	7	6	5	4	3	2	1	0
D	128 (2 ⁷)	64 (2 ⁶)	32 (2 ⁵)	16 (2 ⁴)	8 (2 ³)	4 (2 ²)	2 (2 ¹)	1 (2 ⁰)
D+1	32768 (2 ¹⁵)	16384 (2 ¹⁴)	8192 (2 ¹³)	4096 (2 ¹²)	2048 (2 ¹¹)	1024 (2 ¹⁰)	512 (2 ⁹)	256 (2 ⁸)
D+2	8388608 (2 ²³)	4194304 (2 ²²)	2097152 (2 ²¹)	1048576 (2 ²⁰)	524288 (2 ¹⁹)	262144 (2 ¹⁸)	131072 (2 ¹⁷)	65536 (2 ¹⁶)
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間接アドレス用のfileNとアドレスnに変換して、レジスタD ~ D+2に設定する。		
演算内容	S → D ~ D+2 ・演算前のレジスタSの内容は、演算に関与しません。		
S	使用範囲 A		
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。		
演算条件	入力信号の立上り(OFF ON)		
演算後の内容	S	不変	
	D	レジスタSのアドレスr(下位)	
	D+1	" (下位)	
	D+2	レジスタSのfileN	
	フラグ	不変	

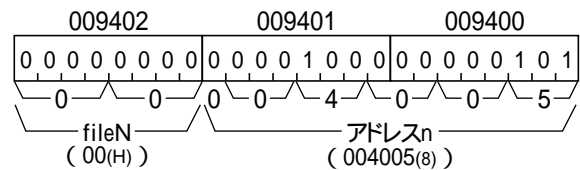
[使用例]



命 令	
STR	004000
F-100	009005 009400

入力条件004000がOFF ONの変化時に、レジスタ009005(ファイルアドレス00004005₈)を、間接アドレス用のfileNとアドレスnに変換して、レジスタ009400 ~ 009402に設定します。

演算後のレジスタ



・レジスタ009005は、file0_(H)のアドレス004005₍₈₎になります。

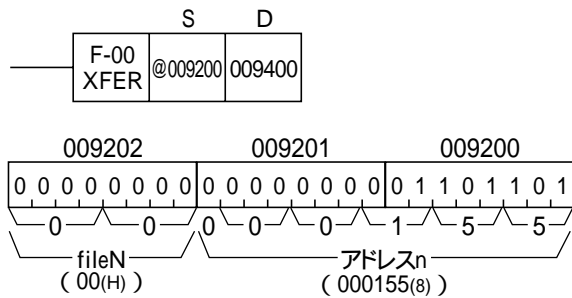
設定したfileNとアドレスnは間接アドレス指定時の直接アドレスとなります。(@009400)

fileNとアドレスnについては、「間接アドレス指定」を参照願います。 9・13ページ

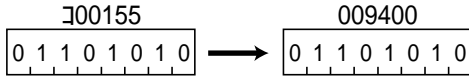
参考 間接アドレスとは

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です。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】009200 ~ 009202の内容で指定するレジスタの内容を009400に転送します。

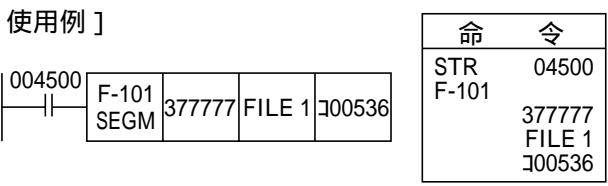


上例では、file0_(H)のアドレス000155₍₈₎は00155で、結果的に@009200は00155を示します。



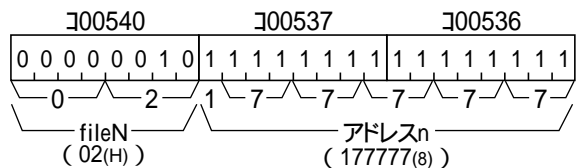
F-101 SEGM 間接アドレスの設定

シンボル	F-101 SEGM n FILE F D				[使用例]	<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>04500</td></tr> <tr><td>F-101</td><td>377777</td></tr> <tr><td></td><td>FILE 1</td></tr> <tr><td></td><td>∩00536</td></tr> </table>	命 令		STR	04500	F-101	377777		FILE 1		∩00536
命 令																
STR	04500															
F-101	377777															
	FILE 1															
	∩00536															
機能	ファイル番号Fとファイルアドレスnを、間接アドレス指定に使用するfileNとアドレスnに変換して、fileNをレジスタD+2、アドレスnをレジスタD、D+1に設定する。															
演算内容	アドレスn(間接アドレス用) → D、D+1 fileN(間接アドレス用) → D+2															
n	使用範囲 00000000 ~ 37777777 ⁽⁸⁾ (ファイルレジスタのとき、バイトアドレス)															
F	0 (ファイルレジスタを除くデータメモリのとき) 1 (ファイルレジスタのとき)															
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。															
演算条件	入力信号の立上り(OFF ON)															
演算後の内容	D	n (下位)														
	D+1	n (上位)														
	D+2	N														
	フラグ	不変														



入力条件004500がOFF ONの変化時に、ファイルレジスタ377777⁽⁸⁾(ファイル番号1)を、間接アドレス用のfileNとアドレスnに変換して、レジスタ∩00536 ~ ∩00540に設定します。

演算後のレジスタ

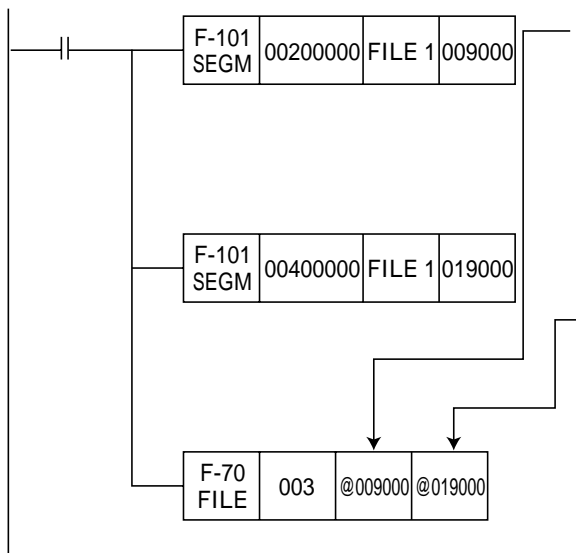


・ファイルレジスタ377777⁽⁸⁾は、FILE 1のバイトアドレス00377777⁽⁸⁾で、file02_Hのアドレス177777⁽⁸⁾になります。

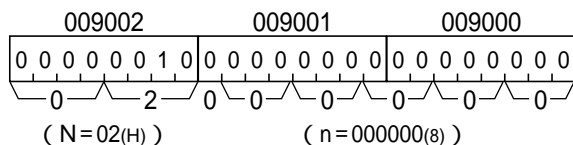
設定したファイルアドレスは、間接アドレス指定時の直接アドレスとなります。(@∩00536)

fileNとアドレスnについては、「間接アドレス指定」を参照願います。 9・13ページ

【例】ファイルレジスタ00200000⁽⁸⁾から3バイトを、ファイルレジスタ00400000⁽⁸⁾から3バイトに転送するプログラム

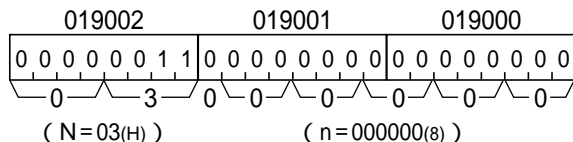


・間接アドレスの設定(F-70のソース側)



ファイルレジスタ00200000⁽⁸⁾は、ファイルアドレスが00400000⁽⁸⁾で、file02_Hのアドレス000000⁽⁸⁾です。

・間接アドレスの設定(F-70のデスティネーション側)



ファイルレジスタ00400000⁽⁸⁾は、ファイルアドレスが00600000⁽⁸⁾は、file03_Hのアドレス000000⁽⁸⁾です。

・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)

シンボル	— F-102 MRD n FILE F D				[使用例]	<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr> <td>STR</td> <td>005000</td> </tr> <tr> <td>F-102</td> <td>00000536</td> </tr> <tr> <td></td> <td>FILE 1</td> </tr> <tr> <td></td> <td>019003</td> </tr> </table>	命 令		STR	005000	F-102	00000536		FILE 1		019003
命 令																
STR	005000															
F-102	00000536															
	FILE 1															
	019003															
機能	ファイル番号Fのファイルアドレス n の内容を、レジスタDに転送する。															
演算内容	ファイル番号Fの n → D				入力条件005000がOFF ONの変化時に、ファイルレジスタ00000536 ⁽⁸⁾ の内容をレジスタ019003に転送します。											
n	使用範囲 00000000 ~ 37777777 ⁽⁸⁾ (ファイルレジスタのとき、バイトアドレス)															
F	0 (ファイルレジスタを除くデータメモリのとき) 1 (ファイルレジスタのとき)				ファイルレジスタ 00000536 019003 											
D	使用範囲 A				Fとnの関係 <table border="1"> <thead> <tr> <th>データメモリ</th> <th>F</th> <th>n の使用範囲</th> </tr> </thead> <tbody> <tr> <td>ファイルレジスタ以外</td> <td>0</td> <td>00000000 ~ 00177777⁽⁸⁾ (ファイルアドレス 2・2,3ページ)</td> </tr> <tr> <td>ファイルレジスタ</td> <td>1</td> <td>00000000 ~ 37777777⁽⁸⁾ (バイトアドレス 2・4ページ)</td> </tr> </tbody> </table>	データメモリ	F	n の使用範囲	ファイルレジスタ以外	0	00000000 ~ 00177777 ⁽⁸⁾ (ファイルアドレス 2・2,3ページ)	ファイルレジスタ	1	00000000 ~ 37777777 ⁽⁸⁾ (バイトアドレス 2・4ページ)		
データメモリ	F	n の使用範囲														
ファイルレジスタ以外	0	00000000 ~ 00177777 ⁽⁸⁾ (ファイルアドレス 2・2,3ページ)														
ファイルレジスタ	1	00000000 ~ 37777777 ⁽⁸⁾ (バイトアドレス 2・4ページ)														
演算条件	入力信号の立上り(OFF ON)				・コントロールユニットの機種(JW-3**CU)により、使用範囲が異なります。											
演算後の内容	n	不変														
	D	レジスタ n の内容														
	フラグ	不変														

(類似命令) 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)

シンボル	— F-102w MRD n FILE F D				[使用例]	<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr> <td>STR</td> <td>005000</td> </tr> <tr> <td>F-102w</td> <td>00000536</td> </tr> <tr> <td></td> <td>FILE 1</td> </tr> <tr> <td></td> <td>019004</td> </tr> </table>	命 令		STR	005000	F-102w	00000536		FILE 1		019004
命 令																
STR	005000															
F-102w	00000536															
	FILE 1															
	019004															
機能	ファイル番号Fのファイルアドレスn、n+1のレジスタ内容を、レジスタD、D+1に転送する。															
演算内容	ファイル番号Fのn、n+1 → D、D+1				入力条件005000がOFF ONの変化時に、ファイルレジスタ00000536、00000537 ⁽⁸⁾ の内容(1 ワードデータ)を、レジスタ019004、019005に転送します。											
n	使用範囲 00000000 ~ 37777776 ⁽⁸⁾ (ファイルレジスタのとき、バイトアドレス) ・nには必ず偶数アドレスを設定してください。(00000003等は禁止)															
F	0 (ファイルレジスタを除くデータメモリのとき) 1 (ファイルレジスタのとき)				ファイルレジスタ 00000536 01101011 → 01101011 019004 00000537 11110010 → 11110010 019005											
D	使用範囲 B ・Dには必ず偶数アドレスを設定してください。(019003等は禁止)				Fとnの関係 <table border="1"> <thead> <tr> <th>データメモリ</th> <th>F</th> <th>n の使用範囲</th> </tr> </thead> <tbody> <tr> <td>ファイルレジスタ以外</td> <td>0</td> <td>00000000 ~ 00177776⁽⁸⁾ (ファイルアドレス 2・2,3ページ)</td> </tr> <tr> <td>ファイルレジスタ</td> <td>1</td> <td>00000000 ~ 37777776⁽⁸⁾ (バイトアドレス 2・4ページ)</td> </tr> </tbody> </table>	データメモリ	F	n の使用範囲	ファイルレジスタ以外	0	00000000 ~ 00177776 ⁽⁸⁾ (ファイルアドレス 2・2,3ページ)	ファイルレジスタ	1	00000000 ~ 37777776 ⁽⁸⁾ (バイトアドレス 2・4ページ)		
データメモリ	F	n の使用範囲														
ファイルレジスタ以外	0	00000000 ~ 00177776 ⁽⁸⁾ (ファイルアドレス 2・2,3ページ)														
ファイルレジスタ	1	00000000 ~ 37777776 ⁽⁸⁾ (バイトアドレス 2・4ページ)														
演算条件	入力信号の立上り(OFF ON)				・コントロールユニットの機種(JW-3**CU)により、使用範囲が異なります。											
演算後の内容	n+1	不変														
	D	レジスタnの内容														
	D+1	レジスタn+1の内容														
フラグ	不変															

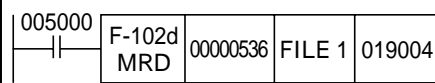
(類似命令) F-102、F-102d、F-176

**F-102d
MRD**

直接指定アドレスのレジスタからの読出(2ワード)

シンボル	$\overline{\text{F-102d MRD}}$ n FILE F D
機能	ファイル番号Fのファイルアドレスn~n+3のレジスタ内容を、レジスタD~D+3に転送する。
演算内容	ファイル番号Fのn~n+3 → D~D+3
n	使用範囲 00000000 ~ 37777774 ⁽⁸⁾ (ファイルレジスタのとき、バイトアドレス) ・nには必ず偶数アドレスを設定してください。(00000003等は禁止)
F	0 (ファイルレジスタを除くデータメモリのとき) 1 (ファイルレジスタのとき)
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。(019003等は禁止)
演算条件	入力信号の立上り(OFF ON)
演算後の内容	n+1 不変
	D レジスタnの内容
	D+1 レジスタn+1の内容
	D+2 レジスタn+2の内容
	D+3 レジスタn+3の内容
フラグ	不変

[使用例]



命 令	
STR	005000
F-102d	00000536 FILE 1 019004

入力条件005000がOFF ONの変化時に、ファイルレジスタ00000536 ~ 00000541⁽⁸⁾の内容(2ワードデータ)を、レジスタ019004 ~ 019007に転送します。

ファイルレジスタ

00000536	0 1 1 0 1 0 1 1	→	0 1 1 0 1 0 1 1	019004
00000537	1 1 1 1 0 0 1 0	→	1 1 1 1 0 0 1 0	019005
00000540	1 0 0 1 1 1 0 0	→	1 0 0 1 1 1 0 0	019006
00000541	0 0 1 1 0 1 1 0	→	0 0 1 1 0 1 1 0	019007

Fとnの関係

データメモリ	F	n の使用範囲
ファイルレジスタ以外	0	00000000 ~ 00177774 ⁽⁸⁾ (ファイルアドレス 2・2,3ページ)
ファイルレジスタ	1	00000000 ~ 37777774 ⁽⁸⁾ (バイトアドレス 2・4ページ)

・コントロールユニットの機種(JW-3**CU)により、使用範囲が異なります。

(類似命令) F-102、F-102w、F-176

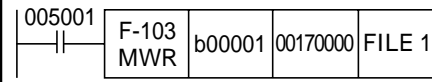
**F-103
MWR**

直接指定アドレスのレジスタへの書込(1 バイト)

シンボル	— F-103 MWR S n FILE F			
機能	レジスタSの内容を、ファイル番号F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 → ファイル番号Fの n			
S	使用範囲 A			
n	使用範囲 00000000 ~ 37777777 ⁽⁸⁾ (ファイルレジスタのとき、バイトアドレス)			
F	0 (ファイルレジスタを除くデータメモリのとき) 1 (ファイルレジスタのとき)			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S	不変		
	n	レジスタSの内容		
	フラグ	不変		

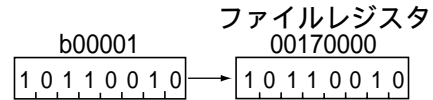
(類似命令) F-103w、F-103d、F-177

[使用例]



命 令	
STR	005001
F-103	b00001
	00170000
	FILE 1

入力条件005001がOFF ONの変化時に、レジスタ b00001の内容を、ファイルレジスタ00170000⁽⁸⁾に転送します。



Fとnの関係

データメモリ	F	n の使用範囲
ファイルレジスタ以外	0	00000000 ~ 00177777 ⁽⁸⁾ (ファイルアドレス 2・2,3ページ)
ファイルレジスタ	1	00000000 ~ 37777777 ⁽⁸⁾ (バイトアドレス 2・4ページ)

・コントロールユニットの機種(JW-3**CU)により、使用範囲が異なります。

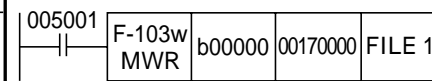
**F-103w
MWR**

直接指定アドレスのレジスタへの書込(1 ワード)

シンボル	— F-103w MWR S n FILE F			
機能	レジスタS、S+1の内容を、ファイル番号F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 → ファイル番号Fの n、n+1			
S	使用範囲 B ・Sには必ず偶数アドレスを設定してください。(019003等は禁止)			
n	使用範囲 00000000 ~ 37777776 ⁽⁸⁾ (ファイルレジスタのとき、バイトアドレス) ・nには必ず偶数アドレスを設定してください。(00000003等は禁止)			
F	0 (ファイルレジスタを除くデータメモリのとき) 1 (ファイルレジスタのとき)			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S、S+1	不変		
	n	レジスタSの内容		
	n+1	レジスタS+1の内容		
	フラグ	不変		

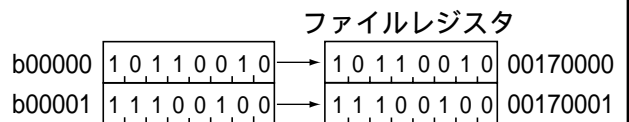
(類似命令) F-103、F-103d、F-177

[使用例]



命 令	
STR	005001
F-103w	b00000
	00170000
	FILE 1

入力条件005001がOFF ONの変化時に、レジスタ b00000、b00001の内容(1 ワードデータ)を、ファイルレジスタ00170000、00170001⁽⁸⁾に転送します。



Fとnの関係

データメモリ	F	n の使用範囲
ファイルレジスタ以外	0	00000000 ~ 00177776 ⁽⁸⁾ (ファイルアドレス 2・2,3ページ)
ファイルレジスタ	1	00000000 ~ 37777776 ⁽⁸⁾ (バイトアドレス 2・4ページ)

・コントロールユニットの機種(JW-3**CU)により、使用範囲が異なります。

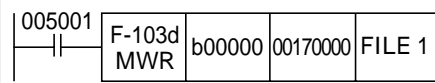
**F-103d
MWR**

直接指定アドレスのレジスタへの書込(2ワード)

シンボル	$\overline{\text{F-103d MWR}}$ S n FILE F
機能	レジスタS ~ S+3の内容を、ファイル番号Fのファイルアドレスn ~ n+3のレジスタに転送する。
演算内容	S ~ S+3 → ファイル番号Fの n ~ n+3
S	使用範囲 C ・Sには必ず偶数アドレスを設定してください。(019003等は禁止)
n	使用範囲 00000000 ~ 37777774 ⁽⁸⁾ (ファイルレジスタのとき、バイトアドレス) ・nには必ず偶数アドレスを設定してください。(00000003等は禁止)
F	0 (ファイルレジスタを除くデータメモリのとき) 1 (ファイルレジスタのとき)
演算条件	入力信号の立上り(OFF ON)
演算後の内容	S ~ S+3 不変
	n レジスタSの内容
	n+1 レジスタS+1の内容
	n+2 レジスタS+2の内容
	n+3 レジスタS+3の内容
	フラグ 不変

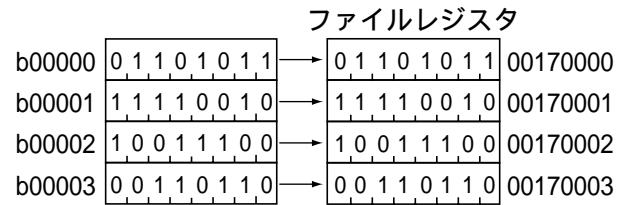
(類似命令) F-103、F-103w、F-177

[使用例]



命 令	
STR	005001
F-103d	b00000
	00170000
	FILE 1

入力条件005001がOFF ONの変化時に、レジスタ b00000 ~ b00003の内容(2ワードデータ)を、ファイルレジスタ00170000 ~ 00170003⁽⁸⁾に転送します。



Fとnの関係

データメモリ	F	n の使用範囲
ファイルレジスタ以外	0	00000000 ~ 00177774 ⁽⁸⁾ (ファイルアドレス 2・2、3ページ)
ファイルレジスタ	1	00000000 ~ 37777774 ⁽⁸⁾ (バイトアドレス 2・4ページ)

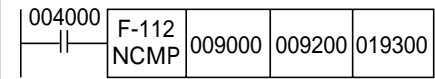
・コントロールユニットの機種(JW-3**CU)により、使用範囲が異なります。

**F-112
NCMP**

n バイト一括比較(1 バイトレジスタ間)

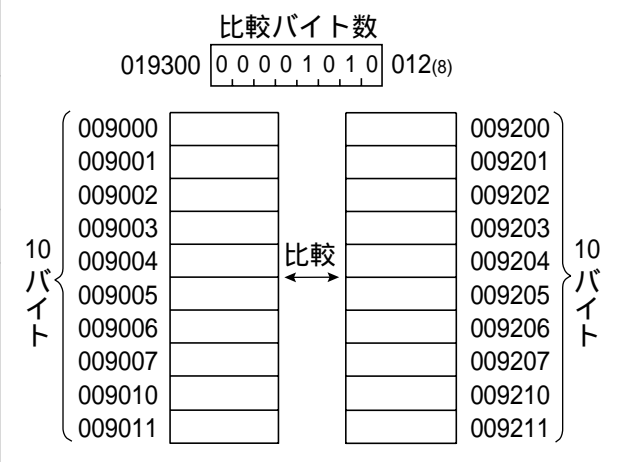
シンボル	F-112 NCMP S1 S2 S3				
機能	レジスタS1から指定バイト数(レジスタS3の内容)のデータと、レジスタS2から指定バイト数(レジスタS3の内容)のデータを大小比較する。				
演算内容	比較結果 → フラグ				
S1	使用範囲 A				
S2	使用範囲 A				
S3	使用範囲 A ・レジスタS3に設定する内容は000 ~ 377 ⁽⁸⁾ です。(000 ⁽⁸⁾ のとき256バイト)				
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)				
演算後の内容	S1 ~ S1+(S3)-1	不変			
	S2 ~ S2+(S3)-1	不変			
	S3	不変			
	フラグ	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355
	S1 ~ S1+(S3)-1 > S2 ~ S2+(S3)-1	0	0	0	1
	S1 ~ S1+(S3)-1 = S2 ~ S2+(S3)-1	1	0	0	1
	S1 ~ S1+(S3)-1 < S2 ~ S2+(S3)-1	0	1	0	0

[使用例]



命 令	
STR	004000
F-112	009000 009200 019300

レジスタ019300の内容が012⁽⁸⁾(10バイト)とすると、入力条件004000がONのときに、レジスタ009000 ~ 009011の内容と009200 ~ 009211の内容を大小比較して、その結果をノンキャリーフラグ、キャリーフラグ、ゼロフラグに設定します。



- ・009000 ~ 009011 > 009200 ~ 009211のとき
 ノンキャリーON
- ・009000 ~ 009011 = 009200 ~ 009211のとき
 ゼロON、ノンキャリーON
- ・009000 ~ 009011 < 009200 ~ 009211のとき
 キャリーON

(注) 本命令は倍長演算できません。

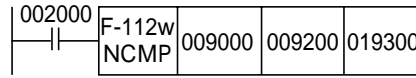
(類似命令) F-12、F-12w、F-12d、F-112w、F-112d、Fc12、Fc12w、Fc12d

**F-112w
NCMP**

nワード一括比較

シンボル	F-112w NCMP				S1	S2	S3
機能	レジスタS1から指定ワード数(レジスタS3の内容)のデータと、レジスタS2から指定ワード数(レジスタS3の内容)のデータを大小比較する。						
演算内容	比較結果 → フラグ						
S1	使用範囲 B ・ S1には必ず偶数アドレスを設定してください。						
S2	使用範囲 B ・ S2には必ず偶数アドレスを設定してください。						
S3	使用範囲 A ・ レジスタS3に設定する内容は000 ~ 377 ₍₈₎ です。(000 ₍₈₎ のとき256ワード)						
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)						
演算後の内容	S1 ~ S1+2(S3)-1	不変					
	S2 ~ S2+2(S3)-1	不変					
	S3	不変					
	フラグ	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	
	S1 ~ S1+2(S3)-1 > S2 ~ S2+2(S3)-1	0	0	0	1		
	S1 ~ S1+2(S3)-1 = S2 ~ S2+2(S3)-1	1	0	0	1		
	S1 ~ S1+2(S3)-1 < S2 ~ S2+2(S3)-1	0	1	0	0		

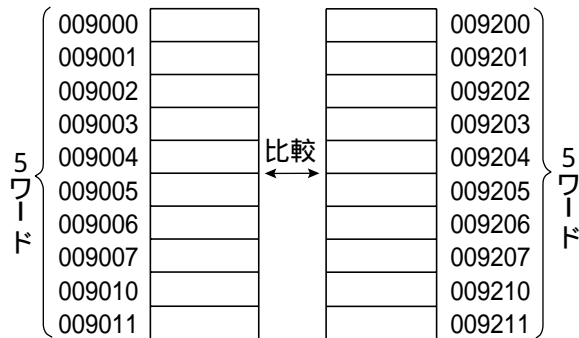
[使用例]



命 令	
STR	002000
F-112w	009000
	009200
	019300

レジスタ019300の内容が005₍₈₎(5ワード)とすると、入力条件002000がONのときに、レジスタ009000 ~ 009011の内容と009200 ~ 009211の内容を大小比較して、その結果をノンキャリーフラグ、キャリーフラグ、ゼロフラグに設定します。

019300 0 0 0 0 1 0 1 005₍₈₎



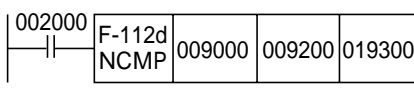
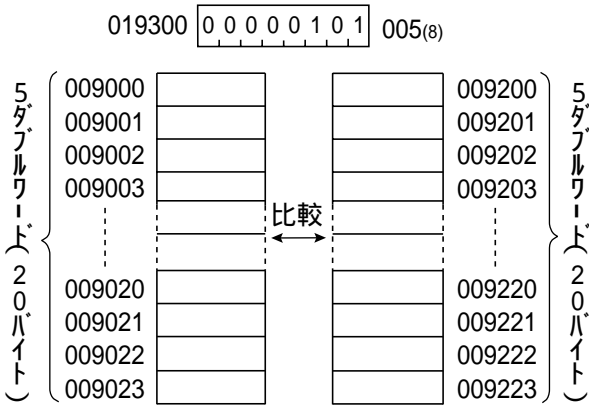
- ・ 009000 ~ 009011 > 009200 ~ 009211のとき
 ノンキャリーON
- ・ 009000 ~ 009011 = 009200 ~ 009211のとき
 ゼロON、ノンキャリーON
- ・ 009000 ~ 009011 < 009200 ~ 009211のとき
 キャリーON

(注) 本命令は倍長演算できません。

(類似命令) F-12、F-12w、F-12d、F-112、F-112d、Fc12、Fc12w、Fc12d

F-112d
NCMP

nダブルワード一括比較

シンボル	F-112d NCMP S1 S2 S3				[使用例]																								
機能	レジスタS1から指定ダブルワード数(レジスタS3の内容)のデータと、レジスタS2から指定ダブルワード数(レジスタS3の内容)のデータを大小比較する。				 <table border="1" style="float: right;"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>002000</td></tr> <tr><td>F-112d</td><td>009000</td></tr> <tr><td></td><td>009200</td></tr> <tr><td></td><td>019300</td></tr> </table>					命 令		STR	002000	F-112d	009000		009200		019300										
命 令																													
STR	002000																												
F-112d	009000																												
	009200																												
	019300																												
演算内容	比較結果 → フラグ				レジスタ019300の内容が005 ⁽⁸⁾ (5ダブルワード)とすると、入力条件002000がONのときに、レジスタ009000～009023の内容と009200～009223の内容を大小比較して、その結果をノンキャリーフラグ、キャリーフラグ、ゼロフラグに設定します。																								
S1	使用範囲 C ・ S1には必ず偶数アドレスを設定してください。																												
S2	使用範囲 C ・ S2には必ず偶数アドレスを設定してください。																												
S3	使用範囲 A ・ レジスタS3に設定する内容は000～377 ⁽⁸⁾ です。(000 ⁽⁸⁾ のとき256ダブルワード)																												
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)				<ul style="list-style-type: none"> ・ 009000～009023 > 009200～009223のとき ノンキャリーON ・ 009000～009023 = 009200～009223のとき ゼロON、ノンキャリーON ・ 009000～009023 < 009200～009223のとき キャリーON 																								
演算後の内容	S1～S1+4(S3)-1	不変			<table border="1"> <thead> <tr> <th>レジスタの内容</th> <th>ゼロ 007357</th> <th>キャリー 007356</th> <th>エラー 007355</th> <th>ノンキャリー 007354</th> </tr> </thead> <tbody> <tr> <td>S1～S1+4(S3)-1 > S2～S2+4(S3)-1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>S1～S1+4(S3)-1 = S2～S2+4(S3)-1</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>S1～S1+4(S3)-1 < S2～S2+4(S3)-1</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> </tbody> </table>					レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	S1～S1+4(S3)-1 > S2～S2+4(S3)-1	0	0	0	1	S1～S1+4(S3)-1 = S2～S2+4(S3)-1	1	0	0	1	S1～S1+4(S3)-1 < S2～S2+4(S3)-1	0	1	0	0
	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355						ノンキャリー 007354																			
	S1～S1+4(S3)-1 > S2～S2+4(S3)-1	0	0	0						1																			
	S1～S1+4(S3)-1 = S2～S2+4(S3)-1	1	0	0						1																			
	S1～S1+4(S3)-1 < S2～S2+4(S3)-1	0	1	0						0																			
S2～S2+4(S3)-1	不変																												
S3	不変																												
フラグ																													

(注) 本命令は倍長演算できません。

(類似命令) F-12、F-12w、F-12d、F-112、F-112w、Fc12、Fc12w、Fc12d

**F-116
DIV**

レジスタ(BCD 8桁)とレジスタ(BCD 8桁)の除算
(DIVide) (商は整数部 8桁、小数部 4桁)

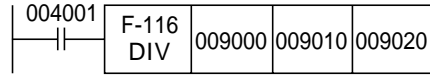
シンボル	— F-116 DIV S1 S2 D			
機能	レジスタS1～S1+3の内容(BCD 8桁)を、レジスタS2～S2+3の内容(BCD 8桁)で除算し、レジスタDからの6バイトに小数4桁と整数8桁の商を格納する。			
演算内容	$(S1 \sim S1+3) \div (S2 \sim S2+3) \rightarrow D \sim D+5$			
S1	使用範囲 C			
S2	使用範囲 C			
D	使用範囲 H			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S1～S1+3	不変		
	S2～S2+3	不変		
	D～D+1	演算結果(小数部4桁)	・レジスタS1～S1+3、S2～S2+3の内容がBCDコード以外するとき、またはS2～S2+3の内容が0のとき、不変。(演算を実行しない)	
	D+2～D+5	演算結果(整数部8桁)		
	フラグ	レジスタS1～S1+3、S2～S2+3の内容 BCDコード ・BCDコード以外 ・S2～S2+3の内容が0	ゼロ 007357	キャリー 007356

S1, S2, Dには必ず偶数アドレスを設定してください。

- ・分子 < 分母 (S1～S1+3 < S2～S2+3) のとき、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桁以下を切り捨てた値になります。

(類似命令) F-16、F-16d、Fc16、Fc16d

[使用例]



命 令	
STR	004001
F-116	009000
	009010
	009020

入力条件004001がOFF ONの変化時に、レジスタ009000～009003のBCD 8桁をレジスタ009010～009013のBCD 8桁で除算し、レジスタ009020、009021に小数4桁、009022～009025に整数8桁を格納します。

	演算前	演算後		
009000	$\begin{matrix} 10^1 & 10^0 \\ 3 & 2 \end{matrix}$	$\begin{matrix} 10^{-3} & 10^{-4} \\ 5 & 0 \end{matrix}$	009020 } 小数部	
009001	$\begin{matrix} 10^3 & 10^2 \\ 5 & 4 \end{matrix}$	$\begin{matrix} 10^{-1} & 10^{-2} \\ 4 & 0 \end{matrix}$		009021 }
009002	$\begin{matrix} 10^5 & 10^4 \\ 7 & 6 \end{matrix}$			
009003	$\begin{matrix} 10^7 & 10^6 \\ 9 & 8 \end{matrix}$	$\begin{matrix} 10^1 & 10^0 \\ 9 & 4 \end{matrix}$	009022 } 整数部	
	÷	$\begin{matrix} 10^3 & 10^2 \\ 7 & 3 \end{matrix}$		009023 }
009010	$\begin{matrix} 10^1 & 10^0 \\ 7 & 8 \end{matrix}$	$\begin{matrix} 10^5 & 10^4 \\ 0 & 1 \end{matrix}$		009024 }
009011	$\begin{matrix} 10^3 & 10^2 \\ 5 & 6 \end{matrix}$	$\begin{matrix} 10^7 & 10^6 \\ 0 & 0 \end{matrix}$		009025 }
009012	$\begin{matrix} 10^5 & 10^4 \\ 0 & 0 \end{matrix}$			
009013	$\begin{matrix} 10^7 & 10^6 \\ 0 & 0 \end{matrix}$			

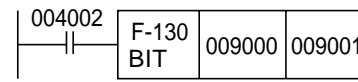
上記の演算は $98765432 \div 5678 = 17394.4050$ を示しています。

F-130 BIT

ビット抽出(間接指定)

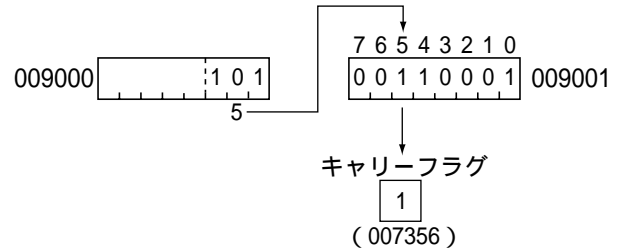
シンボル	— F-130 BIT S1 S2					
機能	レジスタS1の内容で指定するレジスタS2のビット内容を、キャリーフラグ(007356)に転送する。					
演算内容	S2のビット S1 → キャリーフラグ					
S1	使用範囲 A					
S2	使用範囲 A					
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)					
演算後の内容	S1	不変				
	S2	不変				
	フラグ	指定ビットの状態	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		0 (OFF)	0	0	0	0
		1 (ON)	0	1	0	0

[使用例]



命 令	
STR	004002
F-130	009000
	009001

入力条件004002がONのとき、レジスタ009000の下位3ビットで指定するレジスタ009001のビット内容を、キャリーフラグ(007356)に転送します。

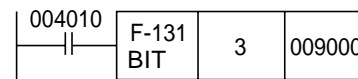


F-131 BIT

ビット抽出(直接指定)

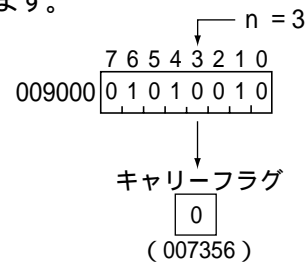
シンボル	— F-131 BIT n S					
機能	レジスタSのビットnの内容を、キャリーフラグに転送する。					
演算内容	Sのビットn → キャリーフラグ					
n	使用範囲 0 ~ 7					
S	使用範囲 A					
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)					
演算後の内容	S	不変				
	フラグ	指定ビットの状態	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		0 (OFF)	0	0	0	0
		1 (ON)	0	1	0	0

[使用例]



命 令	
STR	004010
F-131	3
	009000

入力条件004010がONのとき、レジスタ009000のビット3の内容を、キャリーフラグ(007356)に転送します。



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>セット/リセット指示入力 入力条件</p>
機能	レジスタSの内容(下位3ビット)で指定するレジスタDのビットを、セット/リセット指示入力に従って、セットまたはリセットする。
演算内容	の状態 → Dのビット S
S	使用範囲 A
D	使用範囲 A
演算条件	入力条件 がONのとき (OFF → ONの変化時に限定されない)
演算後の内容	S 不変
	D 指定ビットのみ変化
フラグ	不変

[使用例]

命令

STR	004000
STR	004001
F-132	019000
F-132	000010

入力条件 004001 がONのとき、レジスタ 019000 の下位3ビットで指定するレジスタ 000010 のビットを、004000 がON時にセット、OFF時にリセットします。

019000

000010

004000 が ON →

004000 が OFF →

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>セット/リセット指示入力 入力条件</p>
機能	レジスタDのビットnをセット/リセット指示入力に従って、セットまたはリセットする。
演算内容	の状態 → Dのビット n
n	使用範囲 0 ~ 7
D	使用範囲 A
演算条件	入力条件 がONのとき (OFF → ONの変化時に限定されない)
演算後の内容	D 指定ビットのみ変化
	フラグ 不変

[使用例]

命令

STR	004002
STR	004003
F-133	7
F-133	b01000

入力条件 004003 がONのとき、レジスタ b01000 のビット7を、004002 がON時にセット、OFF時にリセットします。

b01000

004002 が ON →

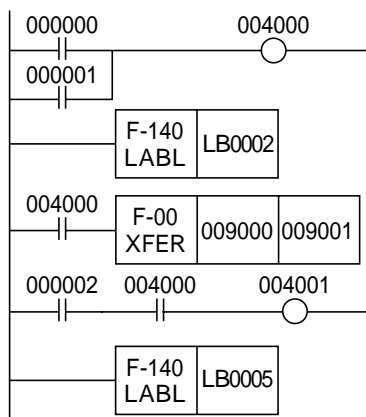
004002 が OFF →

F-140
LABL **ラベルの設定**
 (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	使用範囲 0000 ~ 1777 ⁽⁸⁾

を示す。

[使用例]



アドレス	命 令
0000000	STR 000000
0000001	OR 000001
0000002	OUT 004000
0000003	F-140
0000004	LB0002
0000005	STR 004000
0000006	F-00
0000007	009000
0000010	009001
0000011	STR 000002
0000012	AND 004000
0000013	OUT 004001
0000014	F-140
0000015	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。よって、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 ~ LB1777)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。使用方法はシステムメモリ#0240、サブルーチンコールのリターン命令(F-143)の説明を参照願います。
- ・ ラベル番号(LB1360 ~ LB1417)は入力割込用に使用します。使用方法はシステムメモリ#0241 ~ #0245、サブルーチンコールのリターン命令(F-143)の説明を参照願います。

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LBr(F-140)のプログラムアドレスへ移す。																											
n	使用範囲 0000 ~ 1777 ⁽⁸⁾																											
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)																											
[使用例]																												
<table border="1"> <thead> <tr> <th>アドレス</th> <th>命 令</th> </tr> </thead> <tbody> <tr> <td>0000005</td> <td>F-140</td> </tr> <tr> <td>0000006</td> <td>LB0005</td> </tr> <tr> <td>⋮</td> <td>⋮</td> </tr> <tr> <td>0000777</td> <td>STR 000000</td> </tr> <tr> <td>0001000</td> <td>F-141</td> </tr> <tr> <td>0001001</td> <td>LB0200</td> </tr> <tr> <td>⋮</td> <td>⋮</td> </tr> <tr> <td>0002002</td> <td>STR 000001</td> </tr> <tr> <td>0002003</td> <td>F-141</td> </tr> <tr> <td>0002004</td> <td>LB0005</td> </tr> <tr> <td>0002005</td> <td>F-140</td> </tr> <tr> <td>0002006</td> <td>LB0200</td> </tr> </tbody> </table>			アドレス	命 令	0000005	F-140	0000006	LB0005	⋮	⋮	0000777	STR 000000	0001000	F-141	0001001	LB0200	⋮	⋮	0002002	STR 000001	0002003	F-141	0002004	LB0005	0002005	F-140	0002006	LB0200
アドレス	命 令																											
0000005	F-140																											
0000006	LB0005																											
⋮	⋮																											
0000777	STR 000000																											
0001000	F-141																											
0001001	LB0200																											
⋮	⋮																											
0002002	STR 000001																											
0002003	F-141																											
0002004	LB0005																											
0002005	F-140																											
0002006	LB0200																											
<ul style="list-style-type: none"> • 入力条件000000がONのとき、LB0200のプログラムアドレス0002005へジャンプし、アドレス0002007以降の命令へ移ります。 • 入力条件000001がONのとき、LB0005のプログラムアドレス0000005へジャンプし、アドレス000007以降の命令へ移ります。 																												

F-141命令を実行後、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を短縮できます。

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。ラベル番号(LB1353 ~ LB1417)は、タイマ割込(#0240設定)と入力割込(#0241 ~ #0245設定)で使用します。

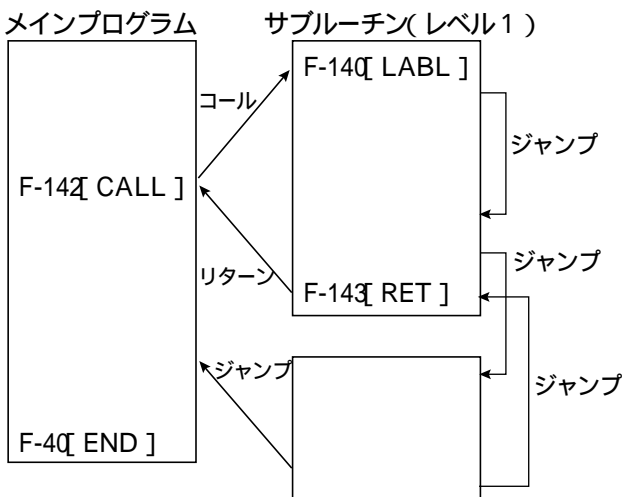
F-142(CALL) / F-143(RET)の多重使用 (ネスティング) に使用する場合

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。

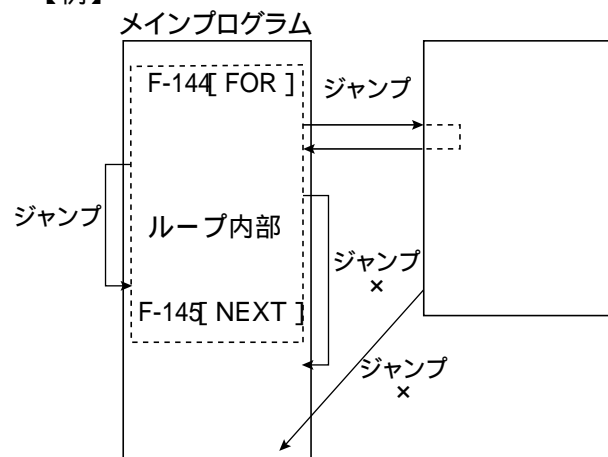
F-144(FOR) / F-145(NEXT)の多重使用 (ネスティング) に使用する場合

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】



【例】

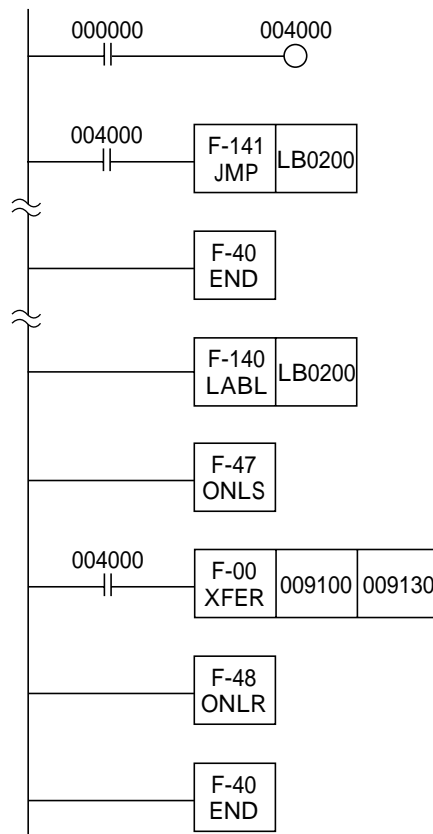
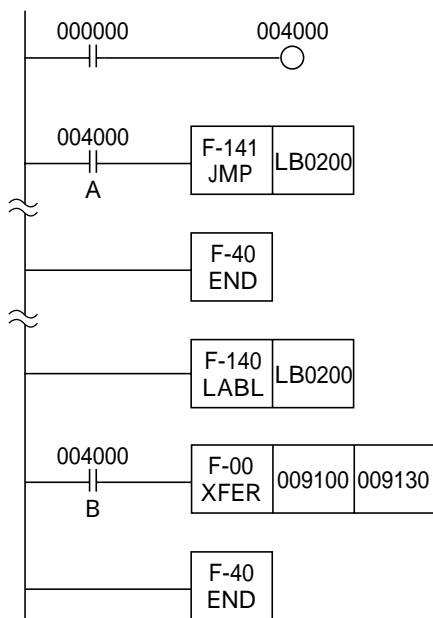


F-140(LABL)、F-141(JMP)命令で下記のプログラム例 1、例 2)を組んだとき、プログラム例 1 ではF-00(XFER)命令は目的の動作を行いません。
 動作は、プログラム例 1 とプログラム例 2 とともに接点004000(Aの接点)がONするとF-141を実行し、F-141で指定するジャンプ先までジャンプします。ジャンプ後、次の接点004000(Bの接点)がONのときF-00を実行するプログラム例です。

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004000(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004000(Bの接点)がONになっているためF-00を実行します。ところが、接点004000がONになった 2 回目のサイクルでは接点004000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は立上りと認識しないためです。
- ・プログラム例 2 では、接点004000(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時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と比較し、その結果、立上りと認識したときに実行する命令です。

ジャンプ先は同じブロック内にしてください。
 (他のブロックへのジャンプは行わないでください。)

F-142
CALL

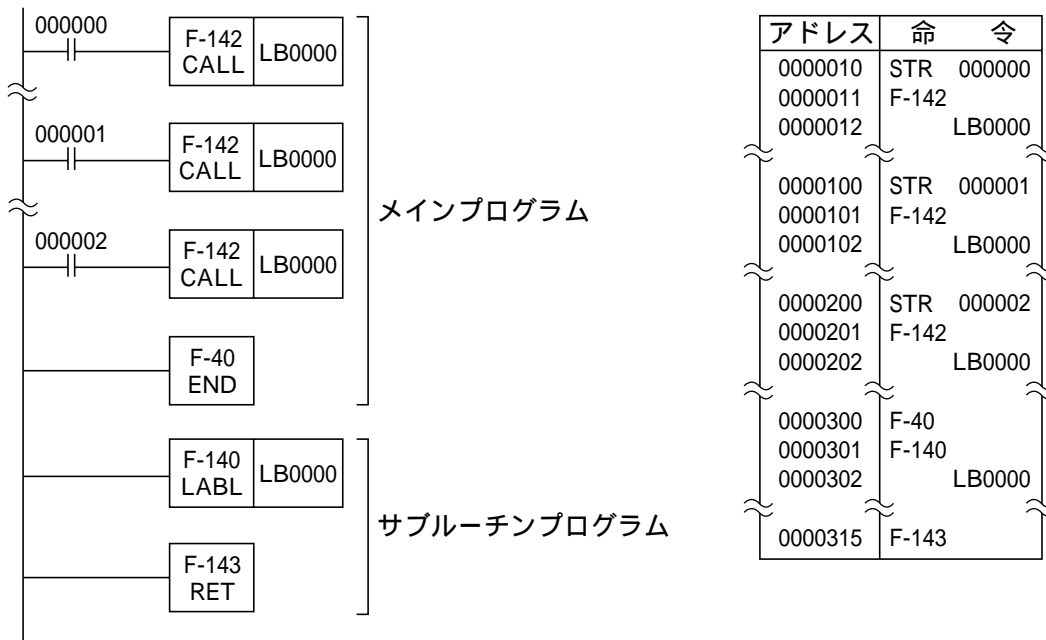
ラベルをサブルーチンコール
(CALL)

F-143
RET

サブルーチンからのリターン
(Return)

シンボル	$\overline{\text{F-142 CALL}} \text{LBn}$ 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	使用範囲 0000 ~ 1777(8)
演算条件	入力信号の立上り(OFF ON)

[使用例]



- ・ F-142(CALL)、F-143(RET)命令は、プログラム中で何度も実行する部分がある場合、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。
- ・ 上例では、入力条件000000がOFF ONの変化時に、メインプログラムからアドレス0000303 ~ 0000314のサブルーチンプログラムに移り、F-143命令でメインプログラムのアドレス0000013以降の命令を実行します。

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。

次の各命令は、サブルーチンプログラムに入れられません。

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)

サブルーチンコールでは、F-148(CAL+)やF-149(RETC)も使用できません。

F-143(RET)命令はタイマ割込(#0240の設定)や入力割込(#0241 ~ #0245の設定)でサブルーチンプログラムを作成時にも使用します。

タイマ割込のラベル番号はLB1353 ~ LB1357を使用します。

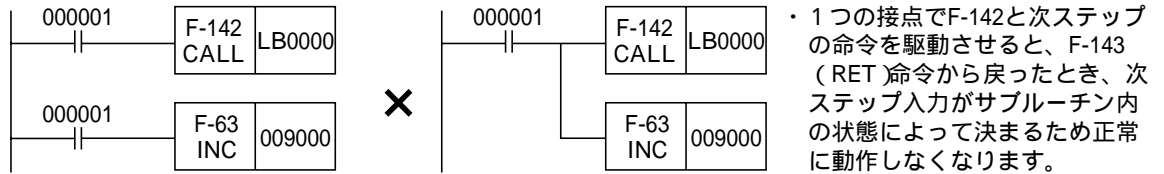
入力割込のラベル番号はLB1360 ~ LB1417を使用します。

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。

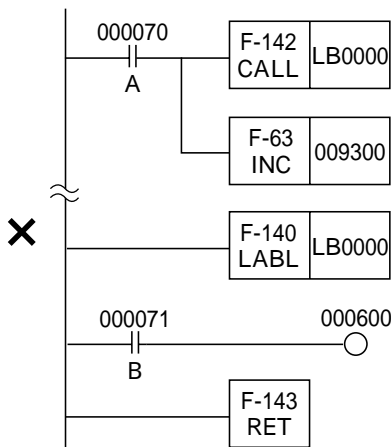
F-143はF-142へ命令の実行を移行するのではなく、F-142の次ステップへ命令の実行を移行します。

F-14〔CALL〕命令の次ステップは接点入力で始めてください。

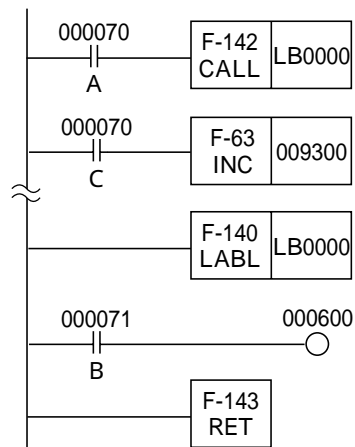


F-14〔CALL〕、F-143〔RET〕命令を下記プログラム(例1 / 例2)で作成すると、両方とも動作は同じですが、プログラム例1ではF-63〔JNC〕命令が接点000070の条件で動作しません。動作は、例1 / 例2ともに接点000070〔Aの接点〕がONすると、F-142を実行した後、F-142で指定しているサブルーチンへ命令の実行が移動します。F-143でF-142の次ステップのF-63へ戻り、F-63を実行する例です。

プログラム例1



プログラム例2

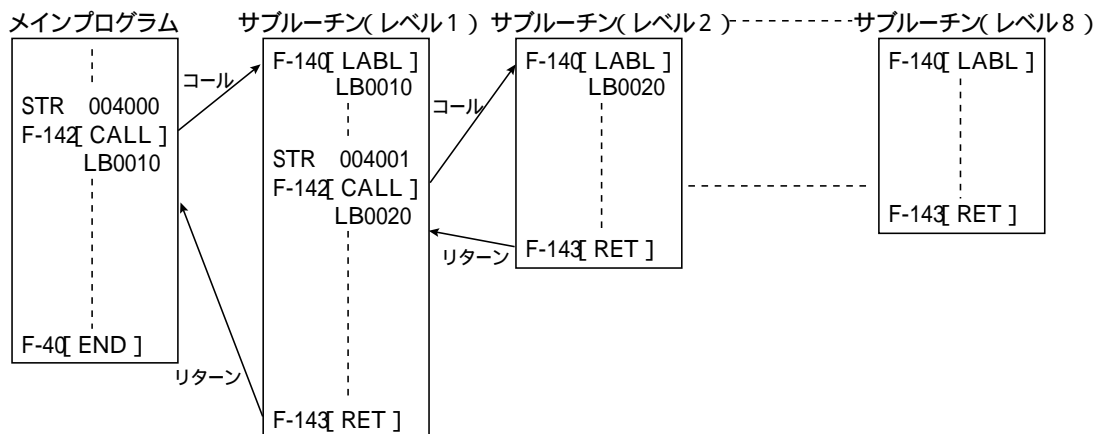


- ・プログラム例1では、接点000070〔Aの接点〕がONするとF-142を実行し、F-142で指定しているサブルーチンへ移行し、F-143まで命令を実行した後、F-142の次ステップ(F-63)へ命令の実行が移行します。このときF-63は接点000070〔Bの接点〕の条件で実行します。これはF-143でF-142の次ステップへ戻ったとき、F-63の実行条件はF-63の1つ前のACC〔アキュムレータ〕に書き込まれた状態(例1はサブルーチン内の最後にACCに書き込まれた状態)で実行するため、F-63はプログラムどおりに動作しません。
- ・プログラム例2では、F-143でF-142の次ステップへ戻っても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実行するのではなく、戻った後の接点000070〔Cの接点〕の条件で実行するためプログラムどおりに動作します。

以上よりF-142の次ステップは、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。

F-14〔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】



(類似命令) F-142、F-148、F-143、F-149

F-144
FOR

ループ回数の設定
(FOR)

F-145
NEXT

ループの終了
(NEXT)

シンボル	— F-144 FOR n F-145 NEXT
機能	F-144(FOR)とF-145(NEXT)間のプログラムを n 回繰り返す。
n	使用範囲 000 ~ 377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256回繰り返す)
演算条件	入力信号の立上り(OFF ON)
[使用例]	
<p>入力条件000001がOFF ONの変化時に、F-63(INC) 命令を 4 回実行します。</p>	

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。

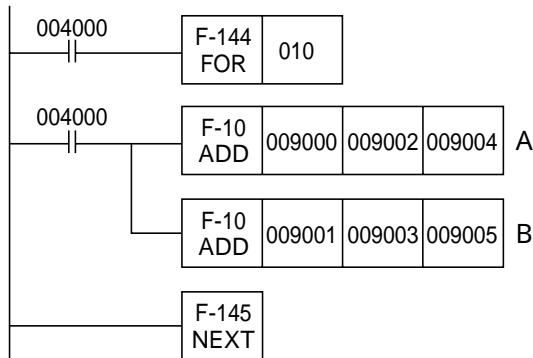
非実行時、F-144(FOR)とF-145(NEXT)の間のデータメモリの内容は変化しません。

次の各命令は、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)

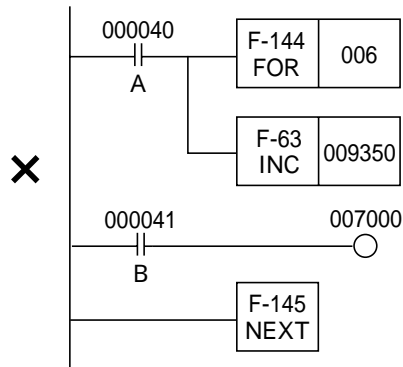
F-144(FOR)/F-145(NEXT)の間で実行する命令数は極力少くしてください。演算時間を考慮し、設計してください。

F-10命令は倍長演算が可能ですが、下記プログラムでBの命令の演算フラグはAの命令に影響しません。

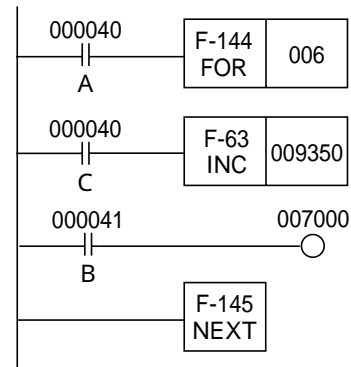


F-144(FOR)、F-145(NEXT)命令を下記プログラム例1 / 例2)で作成すると、両方とも動作は同じですが、プログラム例1ではF-63(INC)命令が動作しません。動作は、例1 / 例2ともに接点000040(Aの接点)がONすると、F-144の次ステップからF-145の間のプログラムを、F-144で指定する回数だけ繰り返し実行します。

プログラム例1



プログラム例2



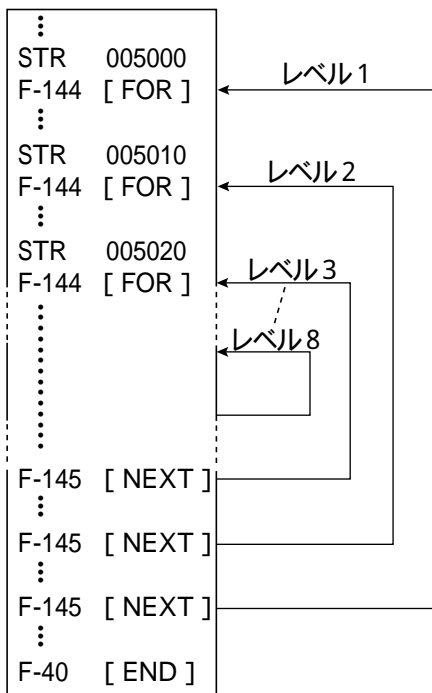
- ・プログラム例1では、接点000040(Aの接点)がONするとF-144とF-63を実行した後、F-144の次ステップからF-145の間のプログラムをF-144で指定する回数だけ繰り返し実行します。このとき、F-63は接点000041(Bの接点)の条件で実行します。これはF-145でF-144の1つ前のステップに戻ったとき、F-63の実行条件はF-63の1つ前のACC(アキュムレータ)に書き込まれた状態(この例では、F-145の1つ前にACCに書き込まれた状態)で実行するため、プログラムどおりに動作しません。
- ・プログラム例2では、F-145でF-144の1つ前のステップへ戻っても、F-63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000040(Cの接点)の条件で実行するため、プログラムどおりに動作します。

以上よりF-144の次ステップは、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るときはF-146(FORR)命令を、また途中でループ演算を抜け出すときはF-147(EXIT)命令を使用してください。

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】



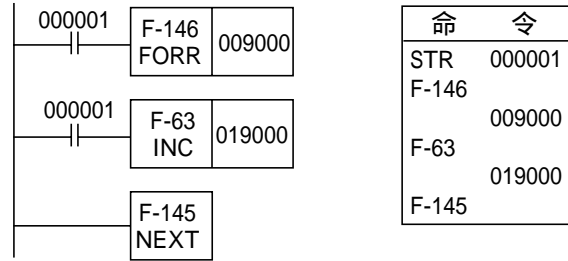
**F-146
FORR**

**ループ回数のレジスタ設定
(FORR)**

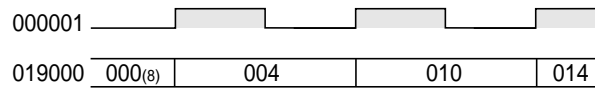
シンボル	F-146 FORR S
機能	F-146(FORR)とF-145(NEXT)間のプログラムを、Sの内容の回数くり返す。
S	使用範囲A ・レジスタSに設定する内容は000~377 ⁽⁸⁾ です。(000 ⁽⁸⁾ のとき256回)
演算条件	入力信号の立上り(OFF ON)
演算内容	S 不変
	フラグ 不変

(類似命令) F-144

[使用例]



入力条件000001がOFF ONの変化時に、F-63(INC)命令をレジスタ009000の内容の回数実行します。下記はレジスタ009000の内容が4の設定例です。



F-146(FORR)とF-145(NEXT)は、必ず一対で使用してください。

F-146(FORR)命令を実行時、F-146(FORR)とF-145(NEXT)間の命令は、レベル演算条件(ONで実行)となります。

非実行時、F-146(FORR)とF-145(NEXT)間のデータメモリの内容は変化しません。

ループ回数の途中で演算を終了させるときは、F-147(EXIT)を使用してください。

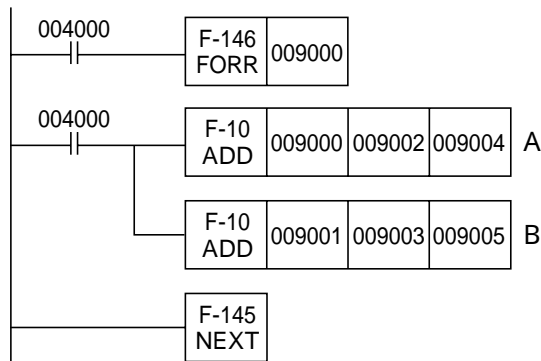
次の各命令を、F-146(FORR)とF-145(NEXT)間に入れることはできません。

TMR、CNT、F-30(MCS)、F-31(MCR)、F-4(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(ネスティング)は最大8レベルまで可能です。使用方法はF-144(FOR)と同じです。

F-146(FORR)と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。


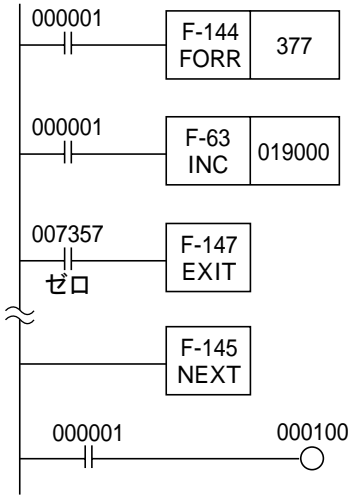
F-10命令は倍長演算が可能ですが、下記プログラムでBの命令の演算フラグはAの命令に影響を与えません。



その他の注意は、F-144(FOR)を参照願います。

**F-147
EXIT**

ループの条件終了

シンボル																									
機能	F-144(FOR)または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のとき (ON OFFの変化時に限定されない)																								
[使用例]																									
 <table border="1" data-bbox="630 548 790 940"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>000001</td> </tr> <tr> <td>F-144</td> <td></td> </tr> <tr> <td></td> <td>377</td> </tr> <tr> <td>F-63</td> <td></td> </tr> <tr> <td></td> <td>019000</td> </tr> <tr> <td>STR</td> <td>007357</td> </tr> <tr> <td>F-147</td> <td></td> </tr> <tr> <td></td> <td>⋮</td> </tr> <tr> <td>F-145</td> <td></td> </tr> <tr> <td>STR</td> <td>000001</td> </tr> <tr> <td>OUT</td> <td>000100</td> </tr> </tbody> </table>		命 令		STR	000001	F-144			377	F-63			019000	STR	007357	F-147			⋮	F-145		STR	000001	OUT	000100
命 令																									
STR	000001																								
F-144																									
	377																								
F-63																									
	019000																								
STR	007357																								
F-147																									
	⋮																								
F-145																									
STR	000001																								
OUT	000100																								
<ul style="list-style-type: none"> ・入力条件007357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の次命令より実行します。 ・入力条件007357がONのときは、F-147(EXIT)は何の影響もしません。 																									

F-147は、F-144(FOR)とF-145(NEXT)間、またはF-146(FORR)とF-145(NEXT)間で必ず使用してください。それ以外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。

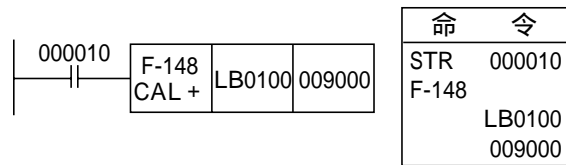
**F-148
CAL +**

**レジスタ設定ラベルをサブルーチンコール
(CAL +)**

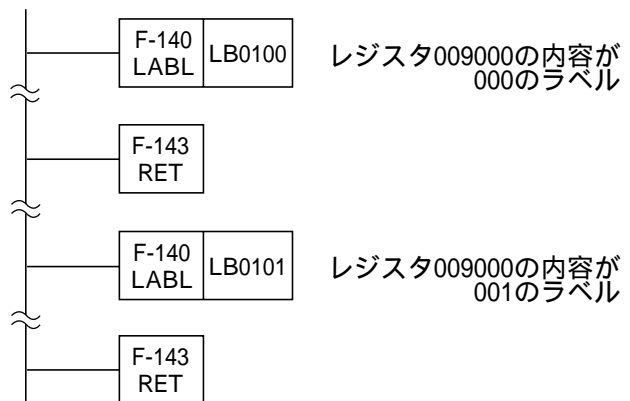
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-148 CAL +</td> <td style="padding: 2px;">LBn</td> <td style="padding: 2px;">S</td> </tr> </table>			F-148 CAL +	LBn	S
F-148 CAL +	LBn	S				
機能	プログラムの実行をLBnとレジスタSの内容を加算した値のラベル(F-140)のサブルーチンに移し、F-143またはF-149命令で戻る。					
LBn	使用範囲 0000 ~ 1370 ⁽⁸⁾ (最下位桁は0とみなす)					
S	使用範囲 A					
演算条件	入力信号の立上り(OFF ON)					
演算後内容	S	不変				
	フラグ	不変				

(類似命令) F-142

[使用例]



入力条件000010がOFF ONの変化時に、LB0100とレジスタ009000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。



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」とみなします。(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。

次の各命令は、サブルーチンプログラムに入れられません。

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)

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。

F-148(CAL +)とF-143(RET)または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。

その他の注意は、F-142(CALL)を参照願います。

**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-14X CALL)またはF-14X 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)
<p>[使用例]</p> <p>命令</p> <pre> F-140 LB0100 ⋮ STR 000002 F-149 ⋮ F-143 </pre> <ul style="list-style-type: none"> ・入力条件000002がOFFすると、F-14X RETC)とF-14X 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000002がONのときは、F-14X RETC)は何も影響しません。 	

F-149は、F-14X LABL)とF-14X RET)間で必ず使用してください。それ以外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、使用できます。

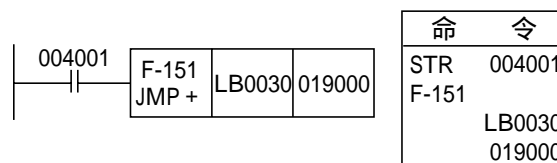
**F-151
JMP +**

**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	使用範囲 0000 ~ 1370 (最下位桁は 0 とみなす)				
S	使用範囲 A				
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)				
演算後内容	S	不変			
	フラグ	不変			

(類似命令) F-151

[使用例]



入力条件004001がONとき、LB0030とレジスタ019000の内容を加算した値のラベル(F-140)のプログラムへ移ります。



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命令を実行後、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。

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。

その他の注意はF-141(JMP)を参照願います。

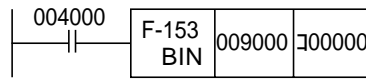
F-153 BCD(8桁) BIN(32ビット)変換

シンボル	— F-153 BIN S D				
機能	レジスタS~S+3(4バイト)のBCD 8桁データを、バイナリコードに変換し、レジスタD~D+3(4バイト)に格納する。				
演算内容	S ~ S+3 → D ~ D+3				
S	使用範囲 C				
D	使用範囲 C				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S ~ S+3	不変			
	D	0 ~ 255	演算結果 ・レジスタS~S+3の内容がBCDコード以外 のとき 不変。		
	D+1	256 ~ 65280			
	D+2	65536 ~ 16711680			
	D+3	16777216 ~ 99999999			
フラグ	レジスタS~S+3の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	BCDコード	0	0	0	0
	BCDコード以外	0	0	1	0

S、Dには必ず偶数アドレスを設定してください。

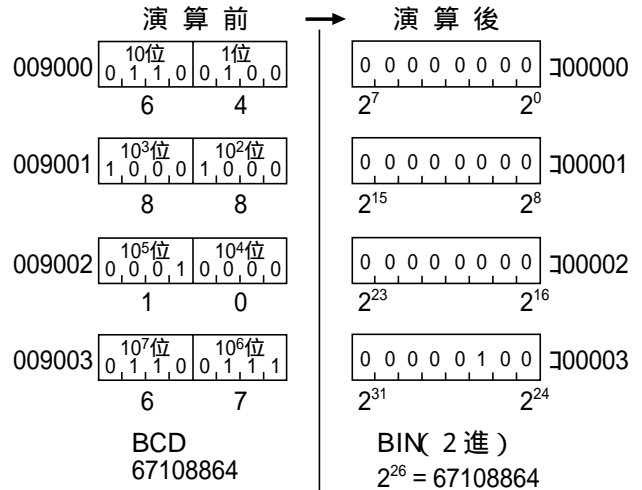
(類似命令) F-03、F-03w、F-03d、F-53

[使用例]



命 令	
STR	004000
F-153	009000
	009003
	000000
	000003

入力条件004000がOFF ONの変化時に、レジスタ009000~009003のBCD 8桁データを、バイナリコードに変換して、レジスタ000000~000003に格納します。



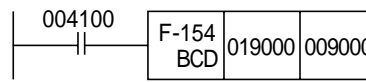
F-154 BIN(32ビット) BCD(10桁)変換

シンボル	— F-154 BCD S D			
機能	レジスタS~S+3(4バイト : 32ビット)のバイナリデータをBCDコードに変換し、レジスタD~D+4(5バイト)に格納する。			
演算内容	S ~ S+3 → D ~ D+4			
S	使用範囲 C			
D	使用範囲 H			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S ~ S+3	不変		
	D	演算結果(1と10の位)		
	D+1	演算結果(10 ² と10 ³ の位)		
	D+2	演算結果(10 ⁴ と10 ⁵ の位)		
	D+3	演算結果(10 ⁶ と10 ⁷ の位)		
	D+4	演算結果(10 ⁸ と10 ⁹ の位)		
フラグ	不変			

S、Dには必ず偶数アドレスを設定してください。

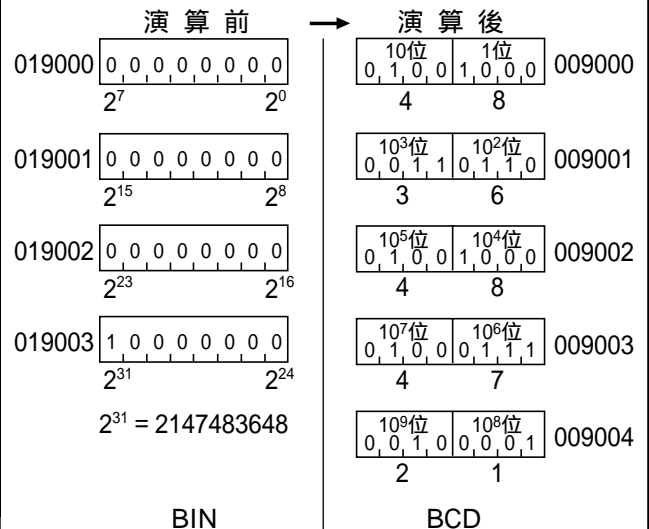
(類似命令) F-04、F-04w、F-04d、F-54、F-154 13 • 31

[使用例]



命 令	
STR	004100
F-154	019000
	009000
	009004

入力条件004100がOFF ONの変化時に、レジスタ019000~019003(32ビット)のバイナリデータをBCDコードに変換して、レジスタ009000~009004に格納します。



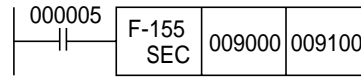
**F-155
SEC**

時(BCD 4桁)分(BCD 2桁)秒(BCD 2桁)→秒(BCD 8桁)変換

シンボル	— F-155 SEC S D				
機能	レジスタ α (秒) S+1(分) S+2(時・下2桁) S+3(時・上2桁)の時間データ(4バイト)を秒データに変換し、レジスタD~D+3(4バイト)に格納する。				
演算内容	α (秒) S+1(分) S+2とS+3(時) → D~D+3(秒) 最大値 9999時59分59秒 → 35999999秒				
S	使用範囲 C				
D	使用範囲 C				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S~S+3	不変			
	D~D+3	演算結果(秒換算) ・レジスタS~S+3の内容がBCDコード以外のとき、または分・秒が60以上のとき、不変。 (演算を実行しない)			
	フラグ	レジスタS~S+3の内容	ゼロ 007357	キャリー 007356	エラー 007355
	BCDコード			0	
	・BCDコード以外 ・分・秒データが 60以上	0	0	1	0

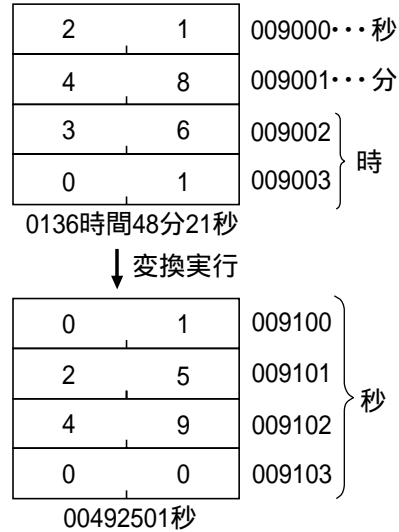
S、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

[使用例]



命 令	
STR	000005
F-155	009000
	009100

入力条件000005がOFF ONの変化時に、レジスタ009000~009003に格納の時・分・秒データを秒データに変換し、レジスタ009100~009103に格納します。(データはBCDコード)



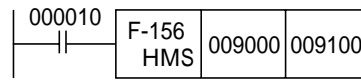
**F-156
HMS**

秒(BCD 8桁) → 時(BCD 4桁)分(BCD 2桁)秒(BCD 2桁)変換

シンボル	— F-156 HMS S D				
機能	レジスタS~S+3に格納している秒データ(BCD 8桁)を、時(BCD 4桁)分(BCD 2桁)秒(2桁)に変換して、レジスタD~D+3に格納する。				
演算内容	S~S+3 → D(秒) D+1(分) D+2とD+3(時) 最大値 35999999 → 9999時59分59秒				
S	使用範囲 C				
D	使用範囲 C				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S~S+3	不変			
	D~D+3	演算結果(時、分、秒) ・S~S+3がBCDコード以外るとき、または最大値を越えるとき、D~D+3は不変。 (演算を実行しない)			
	フラグ	レジスタS~S+3の内容	ゼロ 007357	キャリー 007356	エラー 007355
	BCDコード			0	0
	・BCDコード以外 ・最大値を越える	0	0	1	0

S、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

[使用例]



命 令	
STR	000010
F-156	009000
	009100

入力条件000010がOFF ONの変化時に、レジスタ009001~009003に格納している秒データを、時・分・秒データに変換し、レジスタ009100~009103に格納します。(データはBCDコード)

7	8	009000	} 秒
5	6	009001	
3	4	009002	
1	2	009003	

12345678秒

↓ 変換実行

1	8	009100	} 時
2	1	009101	
2	9	009102	
3	4	009103	

3429時間21分18秒

**F-160
NSFR**

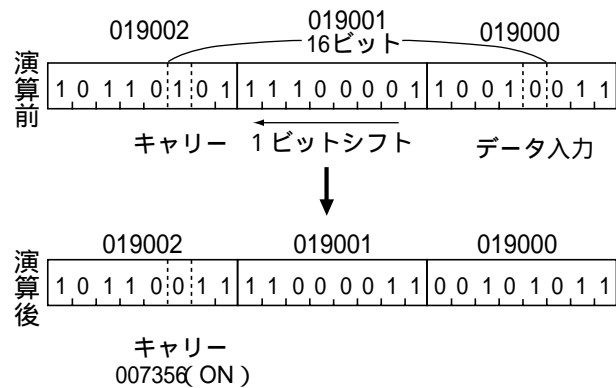
**nビットシフトレジスタ
(N bit ShiFt Register)**

シンボル		シフト方向 指示入力 データ入力 シフト入力 リセット入力			
機能	レジスタDの指定ビット(レジスタS2の内容)から、シフト方向指示入力に従って、指定ビット領域(レジスタS1の内容)で、上位または下位へ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 がONのとき <ul style="list-style-type: none"> シフト方向指示入力 がOFFのとき 				
S1	使用範囲 A ・レジスタS1に設定する内容は000 ~ 255(D)です。(000(D)のとき256ビット)				
S2	使用範囲 A ・レジスタS2に設定する内容は000 ~ 007(D)です。				
D	使用範囲 A				
演算条件	リセット入力 がOFFのとき、シフト入力の立上り(OFF ON)でシフト				
演算後の内容	<ul style="list-style-type: none"> リセット入力 がOFF時、演算結果 リセット入力 がON時、領域内のビットOFF 				
フラグ	リセット入力	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	OFF	0	0または1	0	0
	ON	0	0	0	0

[使用例]

000000					F-160 NSFR	009000	009001	019000	命 令
000001				STR		000000			
000002				STR		000001			
000003				STR		000002			
				STR	000003				
				F-160	009000				
					009001				
					019000				

- ・レジスタ009000の内容が016(D)ビット
 - ・レジスタ009001の内容が003(D)ビット
 - ・方向指示入力000000がON
 - ・データ入力000001がON
 - ・リセット入力000003がOFF
- のとき、シフト入力000002がOFF ONに変化時の動作は次のとおりです。



S1、S2、Dで設定するシフト領域が、タイマ・カウンタの接点領域(ファイルアドレス 00001600 ~ 00001777₈等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。

(類似命令) F-60、F-60w、F-60d、Fc160

Fc160 NSFR nビットシフトレジスタ (N bit ShiFt Register)

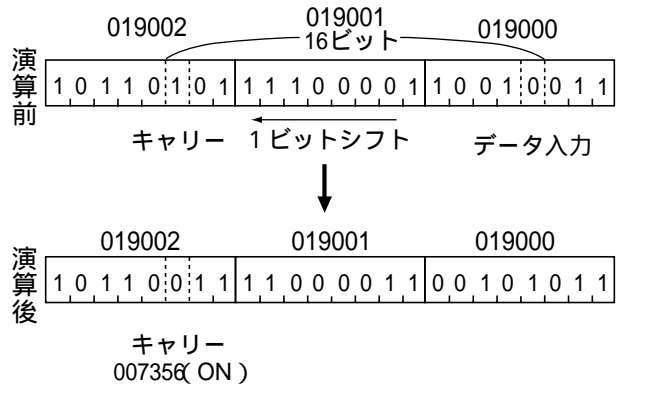
シンボル																
機能	レジスタDのビットn2から、シフト方向指示入力に従って、n1のビット領域で、上位または下位へ1ビットシフトする。															
演算内容	<ul style="list-style-type: none"> シフト方向指示入力 がONのとき <ul style="list-style-type: none"> シフト方向指示入力 がOFFのとき 															
n1	使用範囲000 ~ 255(D) (000(D)のとき256ビット)															
n2	使用範囲000 ~ 007(D)															
D	使用範囲A															
演算条件	リセット入力 がOFFのとき、シフト入力の立上り(OFF ON)でシフト															
演算後の内容	指令領域	<ul style="list-style-type: none"> リセット入力 がOFF時、演算結果 リセット入力 がON時、領域内のビットOFF 														
	フラグ	<table border="1"> <tr> <td>リセット入力</td> <td>ゼロ 007357</td> <td>キャリー 007356</td> <td>エラー 007355</td> <td>ノンキャリー 007354</td> </tr> <tr> <td>OFF</td> <td>0</td> <td>0または1</td> <td>0</td> <td>0</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	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	OFF	0	0または1	0	0	ON	0	0	0
リセット入力	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354												
OFF	0	0または1	0	0												
ON	0	0	0	0												

[使用例]

000000	Fc160 NSFR	016	003	019000	命 令
000001					STR 000000
000002					STR 000001
000003					STR 000002

STR	000000
STR	000001
STR	000002
STR	000003
Fc160	016
	003
	019000

- シフト領域が016(D)ビット
 - データ入力がビット003(D)
 - 方向指示入力000000がON
 - データ入力000001がON
 - リセット入力000003がOFF
- のとき、シフト入力000002がOFF ONに変化時の動作は次のとおりです。



Dで設定するシフト領域が、タイマ・カウンタの接点領域(ファイルアドレス00001600 ~ 00001777(8)等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。

(類似命令) F-60、F-60w、F-60d、F-160

**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																
機能	<p>「レジスタDからD+(n-1)」または「レジスタD-(n-1)からD」の領域内で、データが00(H)のレジスタ(1バイト)に、直前または直後のレジスタ(1バイト)のデータをシフトする。</p>															
演算内容	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>シフト方向指示入力 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>シフト方向指示入力 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> <ul style="list-style-type: none"> ・演算が実行されると、シフトしたレジスタ(1バイト)の内容はクリアされます。 ・レジスタ領域内にデータ00(H)がないとき、演算は実行されません。 ・Dとnで設定するシフト領域は、ファイルアドレス0000000Q₈以前、およびタイマ・カウンタの接点領域 00001600 ~ 0000177₈等)内、ファイルレジスタ0000000Q₈以前、レジスタ、ファイルレジスタの最終アドレス以降でないようにしてください。 															
D	使用範囲 A															
n	使用範囲 000 ~ 377 ₈ バイト (000 ₈ のとき256バイト)															
演算条件	シフト入力 がONでシフト (OFF ONの変化時に限定されない) ・シフト入力 がONの間、毎スキャンサイクルで演算を実行します。															
	ON	OFF														
	演算前	演算後														
データが0の直前のレジスタの内容	DATA1	0														
データが0のレジスタの内容	0	DATA1														
データが0の直後のレジスタの内容	DATA2	同左														
フ(演算後)	D+(n-1)=0のとき 1 D+(n-1) 0のとき 0	D-(n-1)=0のとき 1 D-(n-1) 0のとき 0														
ノンキャリア 007354	1															
エラー 007355	0															
キャリア 007356	D+(n-1)=0のとき 0 D+(n-1) 0のとき 1	D-(n-1)=0のとき 0 D-(n-1) 0のとき 1														
ゼロ 007357	0															
[使用例]	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th colspan="2">命 令</th> </tr> <tr> <td>STR</td> <td>000000</td> </tr> <tr> <td>STR</td> <td>000001</td> </tr> <tr> <td>F-161</td> <td>009000</td> </tr> <tr> <td></td> <td>006</td> </tr> </table> <p>00000()ON.....シフト指示 00001()ON.....009000 ~ 009005をシフト</p>		命 令		STR	000000	STR	000001	F-161	009000		006				
命 令																
STR	000000															
STR	000001															
F-161	009000															
	006															
1. 009003、009004のデータが00(H)で、入力条件が上記のとき、演算結果は次のとおりです。	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th>演算前</th> <th>演算後</th> </tr> <tr> <td>009000</td> <td>3 4</td> </tr> <tr> <td>009001</td> <td>1 2</td> </tr> <tr> <td>009002</td> <td>7 8</td> </tr> <tr> <td>009003</td> <td>0 0</td> </tr> <tr> <td>009004</td> <td>0 0</td> </tr> <tr> <td>009005</td> <td>5 6</td> </tr> </table> <p>ノンキャリアフラグ(007354)のみONします。</p>		演算前	演算後	009000	3 4	009001	1 2	009002	7 8	009003	0 0	009004	0 0	009005	5 6
演算前	演算後															
009000	3 4															
009001	1 2															
009002	7 8															
009003	0 0															
009004	0 0															
009005	5 6															
2. 009001 ~ 009005のデータが00(H)のとき、009000のデータは1演算で009005にシフトします。	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <th>演算前</th> <th>演算後</th> </tr> <tr> <td>009000</td> <td>3 4</td> </tr> <tr> <td>009001</td> <td>0 0</td> </tr> <tr> <td>009002</td> <td>0 0</td> </tr> <tr> <td>009003</td> <td>0 0</td> </tr> <tr> <td>009004</td> <td>0 0</td> </tr> <tr> <td>009005</td> <td>0 0</td> </tr> </table> <p>キャリアフラグ(007356)のみONします。</p>		演算前	演算後	009000	3 4	009001	0 0	009002	0 0	009003	0 0	009004	0 0	009005	0 0
演算前	演算後															
009000	3 4															
009001	0 0															
009002	0 0															
009003	0 0															
009004	0 0															
009005	0 0															

キャリアフラグ(007356)は、「D+(n-1)」または「D-(n-1)」に、0以外のデータがシフトしたときのみ、ONします。

(類似命令) F-61、F-61w、F-61d、F-161w、F-161d

1.の例で3バイトすべてをシフトするには、3スキャンが必要です。

009000	3	4	0	0
009001	1	2	0	0
009002	7	8	3	4
009003	0	0	1	2
009004	0	0	7	8
009005	5	6	5	6

シフトするレジスタの直前データが00(H)のとき、1スキャンで複数データがシフトします。

009000	3	4	0	0
009001	0	0	3	4
009002	5	6	0	0
009003	0	0	0	0
009004	0	0	5	6
009005	9	0	9	0

**F-161w
NASR**

非同期両方向シフトレジスタ(Nワード)
(N word Asynchronous Shift Register)

シンボル			シフト方向指示入力 シフト入力																																																																																																					
機能	<p>「レジスタDからD+2(n-1)+1」または「レジスタD-2(n-1)からD+1」の領域内で、データが0000(H)のレジスタ(1ワード)に、直前または直後のレジスタ(1ワード)のデータをシフトする。</p>																																																																																																							
演算内容	<p>シフト方向指示入力 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>シフト方向指示入力 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-3</td><td>D-4</td></tr> <tr><td>D-1</td><td>D-2</td></tr> <tr><td>D+1</td><td>D</td></tr> </table> <p style="text-align: center;">↑ シフト</p> <ul style="list-style-type: none"> ・演算が実行されると、シフトしたレジスタ(1ワード)の内容はクリアされます。 ・レジスタ領域内にデータ0000(H)がないとき、演算は実行されません。 ・Dとnで設定するシフト領域は、ファイルアドレス00000000Q₈以前、およびタイマ・カウンタの接点領域 00001600 ~ 00001777Q₈等内、ファイルレジスタ00000000Q₈以前、レジスタ、ファイルレジスタの最終アドレス以降でないようにしてください。 			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-3	D-4	D-1	D-2	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-3	D-4																																																																																																							
D-1	D-2																																																																																																							
D+1	D																																																																																																							
D	使用範囲 B																																																																																																							
n	使用範囲 000 ~ 377(8)ワード (000(8)のとき256ワード)																																																																																																							
演算条件	シフト入力 がONでシフト (OFF ONの変化時に限定されない) ・シフト入力 がONの間、毎スキャンサイクルで演算を実行します。																																																																																																							
データが0の直前のレジスタの内容 データが0のレジスタの内容 データが0の直後のレジスタの内容	ON		OFF		ON / OFF																																																																																																			
	演算前	演算後	演算前	演算後	演算前	演算後																																																																																																		
	DATA1	0	DATA1	同左	DATA1	同左																																																																																																		
	0	DATA1	0	DATA2	0以外	同左																																																																																																		
DATA2	同左	DATA2	0	DATA2	同左																																																																																																			
フラグ(演算後)	ノンキャリアー 007354 D+2(n-1)+1, D+2(n-1) = 0のとき 1 D+2(n-1)+1, D+2(n-1) 0のとき 0		D-2(n-1)+1, D-2(n-1) = 0のとき 1 D-2(n-1)+1, D-2(n-1) 0のとき 0		1																																																																																																			
エラー 007355	0		0		0																																																																																																			
キャリアー 007356	D+2(n-1)+1, D+2(n-1) = 0のとき 0 D+2(n-1)+1, D+2(n-1) 0のとき 1		D-2(n-1)+1, D-2(n-1) = 0のとき 0 D-2(n-1)+1, D-2(n-1) 0のとき 1		0																																																																																																			
ゼロ 007357	0		0		0																																																																																																			
<p>[使用例]</p> <div style="display: flex; align-items: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;"> 000000 000001 </div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;"> F-161w NASR </div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;"> 009000 </div> <div style="border: 1px solid black; padding: 5px;"> 005 </div> </div> <table border="1" style="margin-left: 20px;"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>000000</td></tr> <tr><td>STR</td><td>000001</td></tr> <tr><td>F-161w</td><td>009000</td></tr> <tr><td></td><td>005</td></tr> </table> <p>000000()ON.....009000 ~ 009011をシフト 000001()ON.....シフト指示</p> <p>1. 009004, 009005のデータが0000(H)で、入力条件が上記のとき、演算結果は次のとおりです。</p> <table border="1" style="margin-left: 20px;"> <tr><th colspan="2">演算前</th><th colspan="2">演算後</th></tr> <tr><td>009000</td><td>1 2 3 4</td><td>1 2 3 4</td><td></td></tr> <tr><td>009001</td><td>1 2 3 4</td><td>1 2 3 4</td><td></td></tr> <tr><td>009002</td><td>5 6 7 8</td><td>0 0 0 0</td><td></td></tr> <tr><td>009003</td><td>5 6 7 8</td><td>0 0 0 0</td><td></td></tr> <tr><td>009004</td><td>0 0 0 0</td><td>5 6 7 8</td><td></td></tr> <tr><td>009005</td><td>0 0 0 0</td><td>5 6 7 8</td><td></td></tr> <tr><td>009006</td><td>9 8 7 6</td><td>9 8 7 6</td><td></td></tr> <tr><td>009007</td><td>9 8 7 6</td><td>9 8 7 6</td><td></td></tr> <tr><td>009010</td><td>5 4 3 2</td><td>5 4 3 2</td><td></td></tr> <tr><td>009011</td><td>5 4 3 2</td><td>5 4 3 2</td><td></td></tr> </table> <p>ノンキャリアーフラグ(007354)のみONします。</p> <p>2. 009002 ~ 009011のデータが0000(H)のとき、009000、009001のデータは1演算で009010、009011にシフトします。</p> <table border="1" style="margin-left: 20px;"> <tr><th colspan="2">演算前</th><th colspan="2">演算後</th></tr> <tr><td>009000</td><td>1 2 3 4</td><td>0 0 0 0</td><td></td></tr> <tr><td>009001</td><td>1 2 3 4</td><td>0 0 0 0</td><td></td></tr> <tr><td>009002</td><td>0 0 0 0</td><td>0 0 0 0</td><td></td></tr> <tr><td>009003</td><td>0 0 0 0</td><td>0 0 0 0</td><td></td></tr> <tr><td>009004</td><td>0 0 0 0</td><td>0 0 0 0</td><td></td></tr> <tr><td>009005</td><td>0 0 0 0</td><td>0 0 0 0</td><td></td></tr> <tr><td>009006</td><td>0 0 0 0</td><td>0 0 0 0</td><td></td></tr> <tr><td>009007</td><td>0 0 0 0</td><td>0 0 0 0</td><td></td></tr> <tr><td>009010</td><td>0 0 0 0</td><td>1 2 3 4</td><td></td></tr> <tr><td>009011</td><td>0 0 0 0</td><td>1 2 3 4</td><td></td></tr> </table> <p>キャリアーフラグ(007356)のみONします。</p>							命 令		STR	000000	STR	000001	F-161w	009000		005	演算前		演算後		009000	1 2 3 4	1 2 3 4		009001	1 2 3 4	1 2 3 4		009002	5 6 7 8	0 0 0 0		009003	5 6 7 8	0 0 0 0		009004	0 0 0 0	5 6 7 8		009005	0 0 0 0	5 6 7 8		009006	9 8 7 6	9 8 7 6		009007	9 8 7 6	9 8 7 6		009010	5 4 3 2	5 4 3 2		009011	5 4 3 2	5 4 3 2		演算前		演算後		009000	1 2 3 4	0 0 0 0		009001	1 2 3 4	0 0 0 0		009002	0 0 0 0	0 0 0 0		009003	0 0 0 0	0 0 0 0		009004	0 0 0 0	0 0 0 0		009005	0 0 0 0	0 0 0 0		009006	0 0 0 0	0 0 0 0		009007	0 0 0 0	0 0 0 0		009010	0 0 0 0	1 2 3 4		009011	0 0 0 0	1 2 3 4	
命 令																																																																																																								
STR	000000																																																																																																							
STR	000001																																																																																																							
F-161w	009000																																																																																																							
	005																																																																																																							
演算前		演算後																																																																																																						
009000	1 2 3 4	1 2 3 4																																																																																																						
009001	1 2 3 4	1 2 3 4																																																																																																						
009002	5 6 7 8	0 0 0 0																																																																																																						
009003	5 6 7 8	0 0 0 0																																																																																																						
009004	0 0 0 0	5 6 7 8																																																																																																						
009005	0 0 0 0	5 6 7 8																																																																																																						
009006	9 8 7 6	9 8 7 6																																																																																																						
009007	9 8 7 6	9 8 7 6																																																																																																						
009010	5 4 3 2	5 4 3 2																																																																																																						
009011	5 4 3 2	5 4 3 2																																																																																																						
演算前		演算後																																																																																																						
009000	1 2 3 4	0 0 0 0																																																																																																						
009001	1 2 3 4	0 0 0 0																																																																																																						
009002	0 0 0 0	0 0 0 0																																																																																																						
009003	0 0 0 0	0 0 0 0																																																																																																						
009004	0 0 0 0	0 0 0 0																																																																																																						
009005	0 0 0 0	0 0 0 0																																																																																																						
009006	0 0 0 0	0 0 0 0																																																																																																						
009007	0 0 0 0	0 0 0 0																																																																																																						
009010	0 0 0 0	1 2 3 4																																																																																																						
009011	0 0 0 0	1 2 3 4																																																																																																						

キャリアーフラグ(007356)は、「D+2(n-1)+1, D+2(n-1)」または「D-2(n-1)+1, D-2(n-1)」に、0以外のデータがシフトしたときのみ、ONします。

**F-161d
NASR**

非同期両方向シフトレジスタ(Nダブルワード)

シンボル			シフト方向指示入力 シフト入力																																																																									
機能	<p>「レジスタDからD+4(n-1)+3」または「レジスタD-4(n-1)からD+3」の領域内で、データが0のレジスタ(2ワード)に、直前または直後のレジスタ(2ワード)のデータをシフトする。</p>																																																																											
演算内容	<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>シフト方向指示入力 がONのとき、 レジスタDからレジスタD+4n方向へ シフトします。</p> </div> <div style="text-align: center;"> <p>シフト方向指示入力 がOFFのとき、 レジスタDからレジスタD-4n方向へ シフトします。</p> </div> </div> <ul style="list-style-type: none"> ・演算が実行されると、シフトしたレジスタ(2ワード)の内容はクリアされます。 ・レジスタ領域内にデータ0(2ワード)がないとき、演算は実行されません。 ・Dとnで設定するシフト領域は、ファイルアドレス00000000₈以前、およびタイマ・カウンタの接点領域 00001600 ~ 00001777₈等内、ファイルレジスタ00000000₈以前、レジスタ、ファイルレジスタの最終アドレス以降でないようにしてください。 																																																																											
D	使用範囲 C																																																																											
n	使用範囲 000 ~ 377 ₈ ダブルワード (000 ₈ のとき256ダブルワード)																																																																											
演算条件	シフト入力 がONでシフト (OFF ONの変化時に限定されない) ・シフト入力 がONの間、毎スキャンサイクルで演算を実行します。																																																																											
	ON		OFF																																																																									
	演算前	演算後	ON / OFF																																																																									
データが0の直前のレジスタの内容	DATA1	0	DATA1																																																																									
データが0のレジスタの内容	0	DATA1	同左																																																																									
データが0の直後のレジスタの内容	DATA2	同左	DATA2																																																																									
フラグ(演算後)	D+4(n-1) ~ D+4(n-1)+3 = 0のとき 1 D+4(n-1) ~ D+4(n-1)+3 = 0のとき 0		D-4(n-1) ~ D-4(n-1)+3 = 0のとき 1 D-4(n-1) ~ D-4(n-1)+3 = 0のとき 0																																																																									
エラー	0		0																																																																									
キャリー	D+4(n-1) ~ D+4(n-1)+3 = 0のとき 0 D+4(n-1) ~ D+4(n-1)+3 = 0のとき 1		D-4(n-1) ~ D-4(n-1)+3 = 0のとき 0 D-4(n-1) ~ D-4(n-1)+3 = 0のとき 1																																																																									
ゼロ	0		0																																																																									
[使用例]	<div style="display: flex; align-items: center;"> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>000000</td> </tr> <tr> <td>STR</td> <td>000001</td> </tr> <tr> <td>F-161d</td> <td>009000</td> </tr> <tr> <td></td> <td>005</td> </tr> </tbody> </table> <div style="margin-left: 20px;"> <p>000000()ON..... 009000 ~ 009020をシフト</p> <p>000001()ON..... シフト指示</p> </div> </div> <p>009010 ~ 009017のデータが0で、入力条件が上記のとき、演算結果は次のとおりです。</p> <table border="1" style="width: 100%; text-align: center;"> <thead> <tr> <th></th> <th colspan="4">演算前</th> <th colspan="4">演算後</th> </tr> <tr> <th></th> <th>1</th><th>2</th><th>3</th><th>4</th> <th>1</th><th>2</th><th>3</th><th>4</th> </tr> </thead> <tbody> <tr> <td>009000 ~ 009003</td> <td>1</td><td>2</td><td>3</td><td>4</td> <td>1</td><td>2</td><td>3</td><td>4</td> </tr> <tr> <td>009004 ~ 009007</td> <td>9</td><td>A</td><td>B</td><td>C</td> <td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td>009010 ~ 009013</td> <td>0</td><td>0</td><td>0</td><td>0</td> <td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td>009014 ~ 009017</td> <td>0</td><td>0</td><td>0</td><td>0</td> <td>9</td><td>A</td><td>B</td><td>C</td> </tr> <tr> <td>009020 ~ 009023</td> <td>7</td><td>6</td><td>5</td><td>4</td> <td>7</td><td>6</td><td>5</td><td>4</td> </tr> </tbody> </table> <p>ノンキャリーフラグ(007354)のみONします。</p>			命 令		STR	000000	STR	000001	F-161d	009000		005		演算前				演算後					1	2	3	4	1	2	3	4	009000 ~ 009003	1	2	3	4	1	2	3	4	009004 ~ 009007	9	A	B	C	0	0	0	0	009010 ~ 009013	0	0	0	0	0	0	0	0	009014 ~ 009017	0	0	0	0	9	A	B	C	009020 ~ 009023	7	6	5	4	7	6	5	4
命 令																																																																												
STR	000000																																																																											
STR	000001																																																																											
F-161d	009000																																																																											
	005																																																																											
	演算前				演算後																																																																							
	1	2	3	4	1	2	3	4																																																																				
009000 ~ 009003	1	2	3	4	1	2	3	4																																																																				
009004 ~ 009007	9	A	B	C	0	0	0	0																																																																				
009010 ~ 009013	0	0	0	0	0	0	0	0																																																																				
009014 ~ 009017	0	0	0	0	9	A	B	C																																																																				
009020 ~ 009023	7	6	5	4	7	6	5	4																																																																				

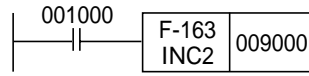
キャリーフラグ(007356)は、「D+4(n-1) ~ D+4(n-1)+3」または「D-4(n-1) ~ D-4(n-1)+3」に、0以外のデータがシフトしたときのみ、ONします。
 (類似命令) F-61、F-61w、F-61d、F-161、F-161w

**F-163
INC2** 加算(+2)カウンタ(1バイト)

シンボル	F-163 INC2	D				
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	D +2 → D					
D	使用範囲 A					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果				
	フラグ	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		376 000 のとき	1	1	0	0
		377 001 のとき	0	1	0	0
上記以外	0	0	0	1		

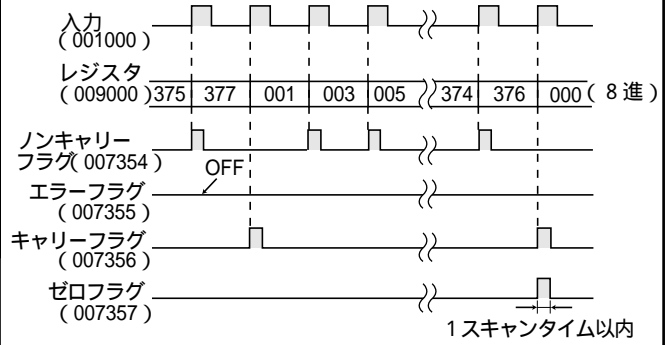
(類似命令) F-63、F-63w、F-63d、F-163w、
F-163d、F-263、F-263w、F-263d

[使用例]



命 令	
STR	001000
F-163	009000

入力条件001000がOFF ONの変化時に、レジスタ009000の内容(バイナリデータ)を+2します。

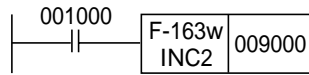


**F-163w
INC2** 加算(+2)カウンタ(1ワード)

シンボル	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	D、D+1 +2 → D、D+1					
D	使用範囲 B ・Dには必ず偶数アドレスを設定してください。(019003等は禁止)					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果(下位)				
	D+1	演算結果(上位)				
	フラグ	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		177776 000000 のとき	1	1	0	0
177777 000001 のとき		0	1	0	0	
上記以外	0	0	0	1		

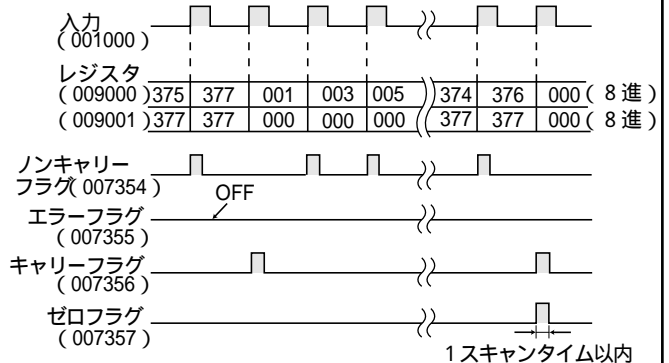
(類似命令) F-63、F-63w、F-63d、F-163、F-163d、F-263、F-263w、F-263d

[使用例]



命 令	
STR	001000
F-163w	009000

入力条件001000がOFF ONの変化時、レジスタ009000、009001の内容(16ビットのバイナリデータ)を+2します。

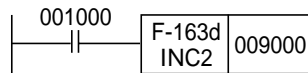


F-163d
INC2

加算(+ 2)カウンタ(2ワード)

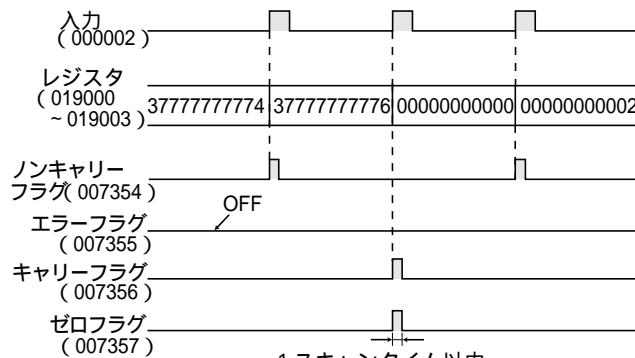
シンボル	F-163d INC2 D				
機能	レジスタD ~ D+3の内容(バイナリデータ)を加算(+ 2)カウントする。				
演算内容	D ~ D+3 +2 → D ~ D+3				
D	使用範囲C ・Dには必ず偶数アドレスを設定してください。(019003等は禁止)				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	D ~ D+3	演算結果(D : 下位、 D+3 : 上位)			
	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	377777776 0000000000のとき	1	1	0	0
	377777777 0000000001のとき	0	1	0	0
	上記以外	0	0	0	1

[使用例]



命 令	
STR	001000
F-163d	009000

入力条件001000がOFF ONの変化時、レジスタ009000 ~ 009003の内容(32ビットのバイナリデータ)を + 2 します。



1 スキャンタイム以内

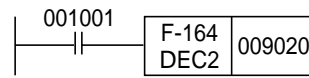
(類似命令) F-63、F-63w、F-63d、F-163、F-163w、F-263、F-263w、F-263d

**F-164
DEC2** 減算(- 2)カウンタ(1 バイト)

シンボル	F-164 DEC2	D				
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	使用範囲 A					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果				
	フラグ	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		002 000 のとき	1	0	0	1
		001 377 000 376 のとき	0	1	0	0
上記以外	0	0	0	1		

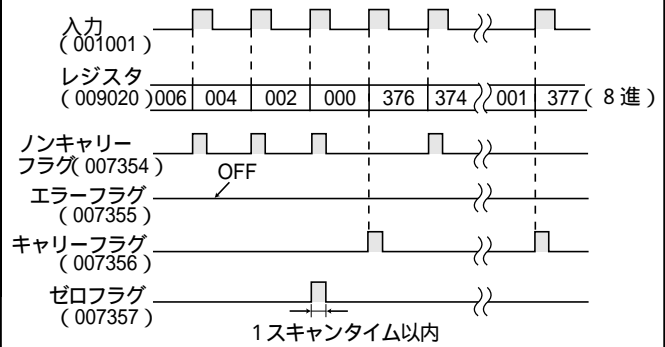
(類似命令) F-64、F-64w、F-64d、F-164w、
F-164d、F-264、F-264w、F-264d

[使用例]



命 令	
STR	001001
F-164	009020

入力条件001001がOFF ONの変化時に、レジスタ009020の内容(バイナリデータ)を - 2 します。

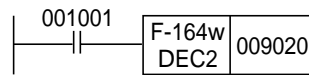


**F-164w
DEC2** 減算(- 2)カウンタ(1 ワード)

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	D、D+1 - 2 → D、D+1					
D	使用範囲 B ・ Dには必ず偶数アドレスを設定してください。(019003等は禁止)					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果(下位)				
	D+1	演算結果(上位)				
	フラグ	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		000002 000000 のとき	1	0	0	1
000001 177777 000000 177776 のとき		0	1	0	0	
上記以外	0	0	0	1		

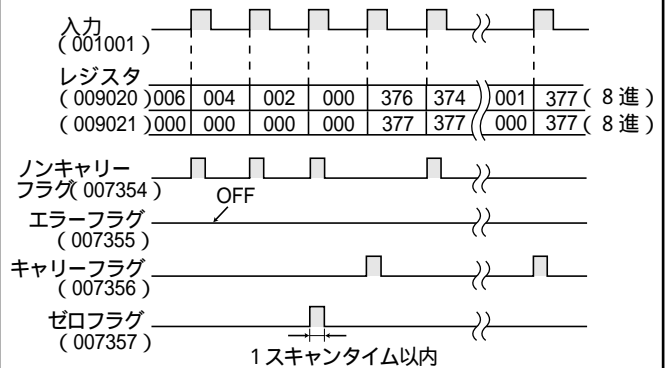
(類似命令) F-64、F-64w、F-64d、F-164、F-164d、F-264、F-264w、F-264d

[使用例]



命 令	
STR	001001
F-164w	009020

入力条件001001がOFF ONの変化時、レジスタ009020、009021の内容(16ビットのバイナリデータ)を - 2 します。



F-164d 減算(- 2)カウンタ(2ワード)
DEC2

シンボル	F-164d DEC2		D		
機能	レジスタD ~ D+3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 ~ D+3 - 2 → D ~ D+3				
D	使用範囲 C ・ Dには必ず偶数アドレスを設定してください。(019003等は禁止)				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容 フラグ	D ~ D+3	演算結果(D : 下位、 D+3 : 上位)			
	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	0000000002 0000000000のとき	1	0	0	1
	0000000001 3777777777、 0000000000 3777777776のとき	0	1	0	0
	上記以外	0	0	0	1

[使用例]

命 令	
STR	001001
F-164d	009020

001001 F-164d
DEC2 009020

入力条件001001がOFF ONの変化時、レジスタ009020 ~ 009023の内容(32ビットのバイナリデータ)を - 2 します。

入力 (000002)

レジスタ (009020 ~ 009023) 0000000004 0000000002 0000000000 3777777776

ノンキャリーフラグ (007354) OFF

エラーフラグ (007355)

キャリーフラグ (007356)

ゼロフラグ (007357)

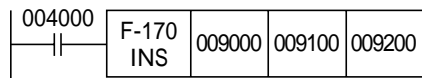
1 スキャンタイム以内

(類似命令) F-64、 F-64w、 F-64d、 F-164、 F-164w、 F-264、 F-264w、 F-264d

F-170 INS **データの挿入(1バイト)**
(INsert)

シンボル	— F-170 INS S D1 D2				
機能	レジスタD1を先頭アドレスとし、D2の内容をバイト数とするシフトレジスタにおいて、D1+(D2+1)のアドレスにレジスタSの内容を挿入する。				
演算内容	<ul style="list-style-type: none"> ・ D1、D2で設定するシフトレジスタ領域が、タイマ・カウンタの接点領域(ファイルアドレス00001600 ~ 00001777₍₈₎等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。 ・ 終了アドレスの内容が00_(H)でないとき、またはD2 D2+1の内容のとき、エラーフラグ(007355)がONして演算しません。 ・ 終了アドレスの内容が00_(H)でないときは、中間データが00_(H)であっても演算しません。中間の00_(H)はデータとして処理します。 				
S	使用範囲 A				
D1	使用範囲 A				
D2	使用範囲 B ・ レジスタD2に設定する内容は000 ~ 377 ₍₈₎ です。(000 ₍₈₎ のとき256バイト) ・ レジスタD2+1に設定する内容は000 ~ 377 ₍₈₎ です。(000 ₍₈₎ のとき先頭アドレス)				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S	不変			
	D2、D2+1	不変			
	D1 ~ D1+(D2)-1	1バイトシフトしてSの内容を挿入 ・ 終了アドレスの内容が00 _(H) 以外のとき、またはD2 D2+1の内容のとき不変			
	フラグ	終了アドレス等の内容	ゼロ 007357	キャリー 007356	エラー 007355
	終了アドレス内容が00 _(H)			0	
	終了アドレス内容が00 _(H) 以外 またはD2 D2+1の内容	0	0	1	0

[使用例]



命 令	
STR	004000
F-170	009000
	009100
	009200

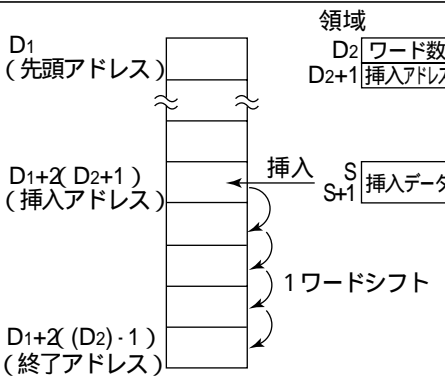
- ・ 入力条件004000がOFF ONの変化時に演算します。
- ・ レジスタ009000の内容(55_(H))が挿入データで、先頭アドレスはレジスタ009100です。
- ・ 挿入アドレスは、レジスタ009100から005₍₈₎バイト(009201の内容)を加算したアドレスです。
- ・ 終了アドレスは、009100から010_(D)バイト(009200の内容)の位置です。バイト数には先頭アドレスも含まれます。
- ・ 終了アドレスの内容が00_(H)のとき、挿入アドレスからデータを1バイトシフトし、009000の内容(55_(H))を挿入アドレスに格納します。

009000	55 _(H)	挿入データ
009200	010 _(D)	—009100から10バイト(009100含む)
009201	005 ₍₈₎	—009100から 5 バイト目に挿入

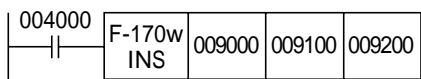
		演算前	→	演算後	
10 バ イ ト	先頭アドレス	009100	12	12	009100
		009101	34	34	009101
		009102	00	00	009102
		009103	78	78	009103
		009104	90	90	009104
	挿入アドレス	009105	09	55	009105
		009106	87	09	009106
		009107	65	87	009107
		009110	00	65	009110
	終了アドレス	009111	00	00	009111
			1バイトシフト		

**F-170w
INS**

**データの挿入(1ワード)
(INSert)**

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F-170w INS</td><td>S</td><td>D1</td><td>D2</td></tr></table>					F-170w INS	S	D1	D2
F-170w INS	S	D1	D2						
機能	レジスタD1を先頭アドレスとし、D2の内容をワード数とするシフトレジスタにおいて、 $D1+\Delta D_{2+1}$ 、 $D1+\Delta D_{2+1}+1$ のアドレスにレジスタS、S+1の内容を挿入する。								
演算内容	 <ul style="list-style-type: none"> ・ D1、D2で設定するシフトレジスタ領域が、タイマ・カウンタの接点領域(ファイルアドレス00001600 ~ 00001777₍₈₎等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。 ・ 終了アドレスの内容が0000_(H)以外の場合、またはD2、D2+1の内容の場合、エラーフラグ(007355)がONして演算しません。 ・ 終了アドレスの内容が0000_(H)以外の場合、中間データが0000_(H)であっても演算しません。中間の0000_(H)はデータとして処理します。 								
S	使用範囲 B ・ Sには必ず偶数アドレスを設定してください。								
D1	使用範囲 B ・ D1には必ず偶数アドレスを設定してください。								
D2	使用範囲 B ・ レジスタD2に設定する内容は000 ~ 377 ₍₈₎ です。(000 ₍₈₎ のとき256ワード) ・ レジスタD2+1に設定する内容は000 ~ 377 ₍₈₎ です。(000 ₍₈₎ のとき先頭アドレス)								
演算条件	入力信号の立上り(OFF ON)								
演算後の内容	S、S+1	不変							
	D2、D2+1	不変							
	D1 ~ $D1+\Delta D_{2+1}-1$	1ワードシフトしてS、S+1の内容を挿入 ・ 終了アドレスの内容が0000 _(H) 以外の場合、またはD2、D2+1の内容のとき不変							
	フラグ	終了アドレス等の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354			
	終了アドレス内容が0000 _(H)	0	0	0	0				
	終了アドレス内容が0000 _(H) 以外 またはD2、D2+1の内容	0	0	1	0				

[使用例]



命 令	
STR	004000
F-170w	009000
	009100
	009200

- ・ 入力条件004000がOFF ONの変化時に演算します。
- ・ レジスタ009000、009001の内容(0155)が挿入データで、先頭アドレスはレジスタ009100です。
- ・ 挿入アドレスは、レジスタ009100から002₍₈₎ワード(009201の内容)を加算したアドレスです。
- ・ 終了アドレスは、009100から006₍₈₎ワード(009200の内容)の位置です。ワード数には先頭アドレスも含まれます。
- ・ 終了アドレスの内容が0000_(H)のとき、挿入アドレスからデータを1ワードシフトし、009000、009001の内容(0155)を挿入アドレスに格納します。

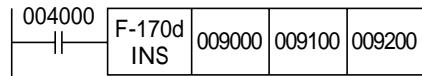
009000	55(下位)) 挿入データ
009001	01(上位)	
009200	006 ₍₈₎	— 009100から 6 ワード(009100含む)
009201	002 ₍₈₎	— 009100から 2 ワード目に挿入

		演算前	→	演算後
6 ワード	先頭アドレス	(009100	12	12
		(009101	34	34
		009102	00	00
	挿入アドレス	009103	78	78
		(009104	90	55
		(009105	09	01
		009106	87	90
		009107	65	09
		009110	00	87
	終了アドレス	009111	10	65
		(009112	00	00
		(009113	00	10
		1ワードシフト		

F-170d INS **データの挿入(2ワード)**
(INsErT)

シンボル	— F-170d INS S D1 D2				
機能	レジスタD1を先頭アドレスとし、D2の内容をダブルワード数とするシフトレジスタにおいて、D1+4(D2+1)~D1+4(D2+1)+3のアドレスにレジスタS~S+3の内容を挿入する。				
演算内容	<p>領域 D2(ダブルワード数) D2+1(挿入アドレス)</p> <p>挿入 S~S+3(挿入データ)</p> <p>2ワードシフト</p> <p>D1(先頭アドレス)</p> <p>D1+4(D2+1)(挿入アドレス)</p> <p>D1+4(D2-1)(終了アドレス)</p> <ul style="list-style-type: none"> ・ D1、D2で設定するシフトレジスタ領域が、タイマ・カウンタの接点領域(ファイルアドレス00001600~00001777(8)等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。 ・ 終了アドレスの内容が00000000_H以外のとき、またはD2 D2+1の内容のとき、エラーフラグ(007355)がONして演算しません。 ・ 終了アドレスの内容が0以外のときは、中間データが0であっても演算しません。中間の0はデータとして処理します。 				
S	使用範囲 C ・ Sには必ず偶数アドレスを設定してください。				
D1	使用範囲 C ・ D1には必ず偶数アドレスを設定してください。				
D2	使用範囲 B ・ レジスタD2に設定する内容は000~377(8)です。(000(8)のとき256ダブルワード) ・ レジスタD2+1に設定する内容は000~377(8)です。(000(8)のとき先頭アドレス)				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S~S+3	不変			
	D2、D2+1	不変			
	D1~D1+4(D2)-1	1ワードシフトしてS~S+3の内容を挿入 ・ 終了アドレスの内容が0以外のとき、またはD2 D2+1の内容のとき不変			
	フラグ	終了アドレス等の内容	ゼロ 007357	キャリー 007356	エラー 007355
	終了アドレス内容が0	0	0	0	0
	終了アドレス内容が0以外 またはD2 D2+1の内容	0	0	1	0

[使用例]



命 令	
STR	004000
F-170d	009000
	009100
	009200

- ・ 入力条件004000がOFF ONの変化時に演算します。
- ・ レジスタ009000~009003の内容(46320155_H)が挿入データで、先頭アドレスはレジスタ009100です。
- ・ 挿入アドレスは、レジスタ009100から002(8)ダブルワード(009201の内容)を加算したアドレスです。
- ・ 終了アドレスは、009100から006(8)ダブルワード(009200の内容)の位置です。ワード数には先頭アドレスも含まれます。
- ・ 終了アドレスの内容が0のとき、挿入アドレスからデータを2ワードシフトし、009000~009003の内容を挿入アドレスに格納します。

009000	55(下位)	} 挿入データ
009001	01	
009002	32	
009003	46(上位)	
009200	006(8)	— 009100から6ダブルワード
009201	002(8)	— 009100から2ダブルワード目に挿入

		演算前	→	演算後
6 ダ ブ ル ワ ー ド	先頭アドレス	009100	12	12
		009101	34	34
		009102	00	00
		009103	78	78
		009104	23	23
		009105	34	34
		009106	45	45
	挿入アドレス	009110	52	55
		009111	76	01
		009112	38	32
		009113	14	46
		009114	37	52
		009115	52	76
		009116	18	38
終了アドレス	009117	45	14	
	009120	27	37	
	009121	68	52	
	009122	51	18	
	009123	39	45	
	009124	00	27	
	009125	00	68	
009126	00	51		
009127	00	39		

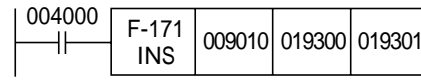
2ワードシフト

**F-171
DEL**

データの削除 (1バイト)
(DELeTe)

シンボル	— F-171 DEL S1 S2 S3				
機能	レジスタS1を先頭アドレスとし、S2の内容をバイト数とするシフトレジスタにおいて、S1+(S3)のアドレスデータを削除する。それ以後のデータは全てシフトする。				
演算内容	<p>領域</p> <p>S1 (先頭アドレス)</p> <p>S2 バイト数</p> <p>S3 削除アドレス</p> <p>S1+(S3) (削除アドレス) → データ削除</p> <p>1バイトシフト</p> <p>S1+(S2)-1 (終了アドレス)</p> <ul style="list-style-type: none"> ・ S2, S3で設定するシフトレジスタ領域が、タイマ・カウンタの接点領域(ファイルアドレス00001600 ~ 00001777₍₈₎)等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。 ・ レジスタ内容がS2 S3のとき、エラーフラグ(007355)がONして演算しません。 ・ 演算完了で、終了アドレスデータは00_(H)となります。 				
S1	使用範囲 A				
S2	使用範囲 A ・ レジスタS2に設定する内容は000 ~ 377 ₍₈₎ です。(000 ₍₈₎ のとき256バイト)				
S3	使用範囲 A ・ レジスタS3に設定する内容は000 ~ 377 ₍₈₎ です。(000 ₍₈₎ のとき先頭アドレス)				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S2	不変			
	S3	不変			
	S1 ~ S1+(S2)-1	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 ・ S2 S3の内容のとき不変			
	フラグ	S2, S3の内容	ゼロ 007357	キャリー 007356	エラー 007355
	S2 > S3	0	0	0	0
	S2 S3	0	0	1	0

[使用例]



命 令	
STR	004000
F-171	009010
	019300
	019301

- ・ 入力条件004000がOFF ONの変化時に演算します。
- ・ 先頭アドレスはレジスタ009010です。
- ・ 削除アドレスは、レジスタ009010から005₍₈₎バイト(019301の内容)を加算したアドレスです。
- ・ 終了アドレスは、009010から010_(D)バイト(019300の内容)の位置です。バイト数には先頭アドレスも含めます。
- ・ 演算を実行すると、削除アドレスのデータを消し、削除アドレス以後から終了アドレスまでのデータを1バイトシフトします。終了アドレスのデータは00_(H)となります。

019300 010_(D) — 009010から10バイト(009010含む)
 019301 005₍₈₎ — 009010から5バイト目を削除

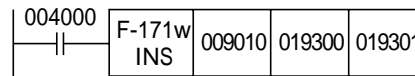
		演算前	→	演算後	
10 バ イ ト	先頭アドレス	009010	23	23	009010
		009011	34	34	009011
		009012	45	45	009012
		009013	67	67	009013
		009014	89	89	009014
	削除アドレス	009015	11	22	009015
		009016	22	33	009016
		009017	33	44	009017
		009020	44	55	009020
	終了アドレス	009021	55	00	009021

1バイトシフト

F-171w DEL データの削除 (1ワード)
(DElete)

シンボル	F-171w DEL S1 S2 S3				
機能	レジスタS1を先頭アドレスとし、S2の内容をワード数とするシフトレジスタにおいて、 $S1+2(S3)$ 、 $S1+2(S3)+1$ のアドレスデータを削除する。それ以後のデータは全てシフトする。				
演算内容	<p>領域 S2 ワード数 S3 削除アドレス</p> <p>→ データ削除 1ワードシフト</p> <ul style="list-style-type: none"> S2, S3で設定するシフトレジスタ領域が、タイマ・カウンタの接点領域(ファイルアドレス00001600 ~ 00001777₍₈₎等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。 レジスタ内容がS2 S3のとき、エラーフラグ(007355)がONして演算しません。 演算完了で、終了アドレスデータは0000_(H)となります。 				
S1	使用範囲 C ・ S1には必ず偶数アドレスを設定してください。				
S2	使用範囲 A ・ レジスタS2に設定する内容は000 ~ 377 ₍₈₎ です。(000 ₍₈₎ のとき256ワード)				
S3	使用範囲 A ・ レジスタS3に設定する内容は000 ~ 377 ₍₈₎ です。(000 ₍₈₎ のとき先頭アドレス)				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S2	不変			
	S3	不変			
	S1 ~ $S1+2(S2)-1$	削除アドレスのデータを消去し、1ワードのデータシフト ・ S2 S3の内容のとき不変			
	フラグ	S2, S3の内容	ゼロ 007357	キャリー 007356	エラー 007355
	S2 > S3	0	0	0	0
	S2 S3	0	0	1	0

[使用例]



命 令	
STR	004000
F-171w	009010 019300 019301

- ・ 入力条件004000がOFF ONの変化時に演算します。
- ・ 先頭アドレスはレジスタ009010です。
- ・ 削除アドレスは、レジスタ009010から002₍₈₎ワード(019301の内容)を加算したアドレスです。
- ・ 終了アドレスは、009010から005₍₈₎ワード(019300の内容)の位置です。ワード数には先頭アドレスも含めます。
- ・ 演算を実行すると、削除アドレスのデータを消し、削除アドレス以後から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0000_(H)となります。

019300 005₍₈₎ — 009010から5ワード(009010含む)
019301 002₍₈₎ — 009010から2ワード目を削除

		演算前	→	演算後
5 ワ ー ド	先頭アドレス	(009010)	23	23
		(009011)	34	34
		009012	45	45
		009013	67	67
		(009014)	89	22
	削除アドレス	(009015)	11	33
		009016	22	44
		009017	33	55
	終了アドレス	(009020)	44	00
		(009021)	55	00
		1ワードシフト		

F-171d DEL データの削除 (2ワード)
(DElete)

シンボル	— F-171d DEL S1 S2 S3				
機能	レジスタS1を先頭アドレスとし、S2の内容をダブルワード数とするシフトレジスタにおいて、S1+4(S3)~S1+4(S3)+3のアドレスデータを削除する。それ以後のデータは全てシフトする。				
演算内容	<p>領域 S2 <u>ダブルワード数</u> S3 <u>削除アドレス</u></p> <p>→ データ削除 1ワードシフト</p> <ul style="list-style-type: none"> ・ S2, S3で設定するシフトレジスタ領域が、タイム・カウンタの接点領域(ファイルアドレス00001600~00001777(8)等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。 ・ レジスタ内容がS2 S3のとき、エラーフラグ(007355)がONして演算しません。 ・ 演算完了で、終了アドレスデータは00000000(H)となります。 				
S1	使用範囲 C ・ S1には必ず偶数アドレスを設定してください。				
S2	使用範囲 A ・ レジスタS2に設定する内容は000~377(8)です。(000(8)のとき256ダブルワード)				
S3	使用範囲 A ・ レジスタS3に設定する内容は000~377(8)です。(000(8)のとき先頭アドレス)				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S2	不変			
	S3	不変			
	S1~S1+4(S2)·1	削除アドレスのデータを消去し、2ワードのデータシフト ・ S2 S3の内容のとき、不変			
	フラグ	S2, S3の内容	ゼロ 007357	キャリー 007356	エラー 007355
	S2 > S3	0	0	0	0
	S2 S3	0	0	1	0

[使用例]

004000	F-171d INS	009010	019300	019301	命令
					STR 004000
					F-171d 009010
					019300
					019301

- ・ 入力条件004000がOFF ONの変化時に演算します。
- ・ 先頭アドレスはレジスタ009010です。
- ・ 削除アドレスは、レジスタ009010から002(8)ダブルワード(019301の内容)を加算したアドレスです。
- ・ 終了アドレスは、009010から005(8)ダブルワード(019300の内容)の位置です。ダブルワード数には先頭アドレスも含まれます。
- ・ 演算を実行すると、削除アドレスのデータを消し、削除アドレス以後から終了アドレスまでのデータを2ワードシフトします。終了アドレスのダブルワードデータは0となります。

019300 005(8) — 009010から5ダブルワード
019301 002(8) — 009010から2ダブルワード目を削除

		演算前	→	演算後
5 ダブル ワード	先頭アドレス	009010	12	12
		009011	34	34
		009012	00	00
		009013	78	78
		009014	52	52
	削除アドレス	009015	76	76
		009016	38	38
		009017	14	14
		009020	52	37
		009021	76	52
終了アドレス	009022	38	18	
	009023	14	45	
	009024	37	27	
	009025	52	68	
	009026	18	51	
	009027	45	39	
	009030	27	00	
	009031	68	00	
	009032	51	00	
	009033	39	00	

2ワードシフト

**F-172
SRCH**

**データの検索(1 バイト)
(SeaRCH)**

シンボル	F-172 SRCH	S	D1	D2	
機能	レジスタD1を先頭アドレスとし、バイト数(D2の内容)で指示する領域内で検索する。レジスタSの内容が検索データです。 データを検索して最初の一致アドレス(D1を基準)が何バイト目であるかをD2+2へ格納する。また、一致した個数をD2+1へ格納する。				
演算内容	<p> $D1$ (先頭アドレス) $D1+1$ $D1+2$ \approx \approx $D1+(D2)-1$ (終了アドレス) </p> <ul style="list-style-type: none"> D1、D2で設定するレジスタ領域が、タイマ・カウンタの接点領域(ファイルアドレス00001600~00001777₍₈₎等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。 				
S	使用範囲 A				
D1	使用範囲 A				
D2	使用範囲 E ・レジスタD2に設定する内容は000~377 ₍₈₎ です。(000 ₍₈₎ のとき256バイト) ・レジスタD2+1の内容は000~377 ₍₈₎ です。000 ₍₈₎ のとき、一致個数が0個または256個です。キャリーフラグ(007356)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を基準として何バイト目であるかの値が格納されます。(000 ₍₈₎ のとき先頭アドレス)				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S	不変			
	$D1 \sim D1+(D2)-1$	不変			
	D2	不変			
	D2+1、D2+2	演算結果 ・一致個数が0個または256個のとき、D2+1は000 ₍₈₎ となる。			
フラグ	検索データ	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	有	0	1	0	0
無	1	0			

[使用例]

004000	F-172 SRCH	009000	100500	019000
--------	---------------	--------	--------	--------

命 令	
STR	004000
F-172	009000
	100500
	019000

- 入力条件004000がOFF ONの変化時に演算します。
- 009000の内容(55_(H))が検索データです。
- 先頭アドレスはレジスタ100500です。
- 終了アドレスは、100500から012₍₈₎バイト(レジスタ019000の内容)の位置です。バイト数には先頭アドレスも含めます。
- レジスタ019001に検索個数を格納します。
- レジスタ019002に、最初の一致アドレスが何バイト目であるか(D1を基準)を格納します。

009000 (55_(H)) 検索データ

019000	012 ₍₈₎	バイト数
019001	004 ₍₈₎	一致個数
019002	002 ₍₈₎	バイト目

10
バ
イ
ト
 先頭アドレス 100500 11
 100501 22
 100502 55 ← 最初の一致
アドレス
 100503 33
 100504 44
 100505 55 ←
 100506 66
 100507 55 ←
 100510 77
 終了アドレス 100511 55 ←

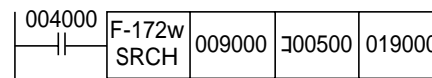
- 演算後、キャリーフラグ(007356)がONします。

**F-172w
SRCH**

データの検索(1ワード)
(SeaRCH)

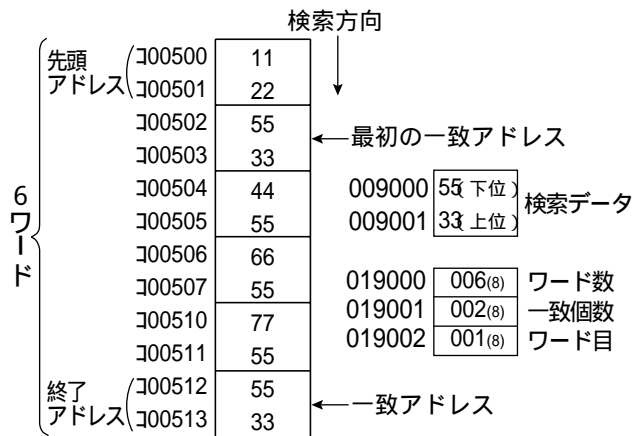
シンボル	F-172w SRCH S D1 D2					
機能	レジスタD1を先頭アドレスとし、ワード数(D2の内容)で指示する領域内で検索する。レジスタS、S+1の内容が検索データです。 データを検索して最初の一一致アドレス(D1を基準)が何ワード目であるかをD2+2へ格納する。また、一致した個数をD2+1へ格納する。					
演算内容	<p>・D1、D2で設定するレジスタ領域が、タイマ・カウンタの接点領域(ファイルアドレス00001600~00001777₍₈₎)等内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。</p>					
S	使用範囲B ・Sには必ず偶数アドレスを設定してください。					
D1	使用範囲B ・D1には必ず偶数アドレスを設定してください。					
D2	使用範囲E ・レジスタD2に設定する内容は000~377 ₍₈₎ です。(000 ₍₈₎ のとき256ワード) ・レジスタD2+1の内容は000~377 ₍₈₎ です。000 ₍₈₎ のとき、一致個数が0個または256個です。キャリーフラグ(007356)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を基準として何ワード目であるかの値が格納されます。(000 ₍₈₎ のとき先頭アドレス)					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	S、S+1	不変				
	D1~D1+2(D2)-1	不変				
	D2	不変				
	D2+1、D2+2	演算結果 ・一致個数が0個または256個のとき、D2+1は000 ₍₈₎ となる。				
フラグ	検索データ	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	
		有	0	1	0	0
		無	1	0	0	0

[使用例]



命令	
STR	004000
F-172w	009000
	000500
	019000

- ・入力条件004000がOFF ONの変化時に演算します。
- ・009000、009001の内容(3355_(H))が検索データです。
- ・先頭アドレスはレジスタ000500です。
- ・終了アドレスは、000500から006₍₈₎ワード(レジスタ019000の内容)の位置です。ワード数には先頭アドレスも含まれます。
- ・レジスタ019001に検索個数を格納します。
- ・レジスタ019002に、最初の一一致アドレスが何ワード目であるか(D1を基準)を格納します。

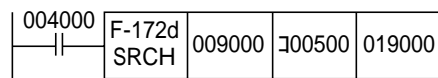


- ・演算後、キャリーフラグ(007356)がONします。

F-172d SRCH データの検索(2ワード)
(SeaRCH)

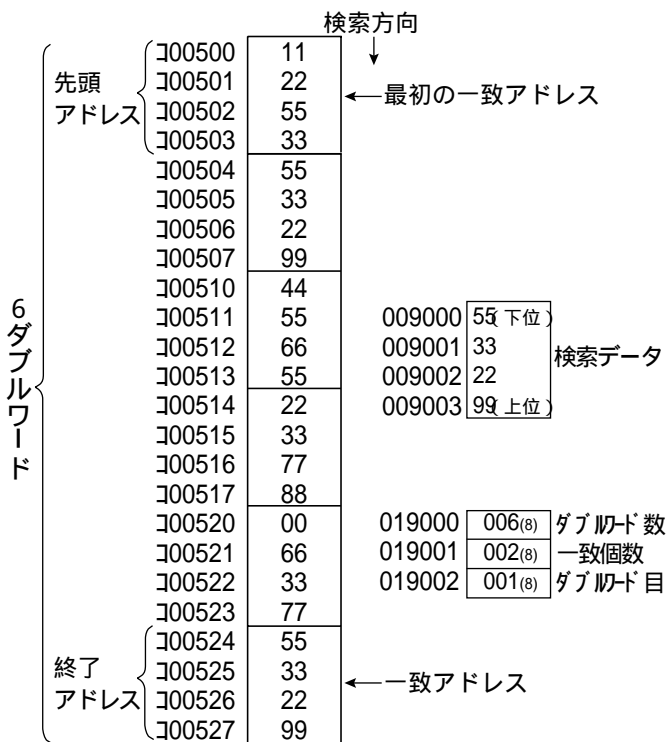
シンボル	— F-172d SRCH S D1 D2			
機能	レジスタD1を先頭アドレスとし、ダブルワード数(D2の内容)で指示する領域内で検索する。レジスタS~S+3の内容が検索データです。 データを検索して最初の一致アドレス(D1を基準)が何ダブルワード目であるかをD2+2へ格納する。また、一致した個数をD2+1へ格納する。			
演算内容	<p>・ D1、D2で設定するレジスタ領域が、タイマ・カウンタの接点領域(ファイルアドレス00001600 ~ 00001777(8)等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。</p>			
S	使用範囲 C ・ Sには必ず偶数アドレスを設定してください。			
D1	使用範囲 C ・ D1には必ず偶数アドレスを設定してください。			
D2	使用範囲 E ・ レジスタD2に設定する内容は000 ~ 377(8)です。(000(8)のとき256ダブルワード) ・ レジスタD2+1の内容は000 ~ 377(8)です。000(8)のとき、一致個数が0個または256個です。キャリーフラグ(007356)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を基準として何ダブルワード目であるかの値が格納されます。(000(8)のとき先頭アドレス)			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S ~ S+3	不変		
	D1 ~ D1+4(D2) - 1	不変		
	D2	不変		
	D2+1、D2+2	演算結果 ・ 一致個数が0個または256個のとき、D2+1は000(8)となる。		
フラグ	検索データ	ゼロ 007357	キャリー 007356	エラー 007355
	有	0	1	0
	無	1	0	0

[使用例]



命 令	
STR	004000
F-172d	009000
	100500
	019000

- ・ 入力条件004000がOFF ONの変化時に演算します。
- ・ 009000 ~ 009003の内容(99223355(H))が検索データです。
- ・ 先頭アドレスはレジスタ100500です。
- ・ 終了アドレスは、100500から006(8)ダブルワード(レジスタ019000の内容)の位置です。ダブルワード数には先頭アドレスも含まれます。
- ・ レジスタ019001に検索個数を格納します。
- ・ レジスタ019002に、最初の一致アドレスが何ダブルワード目であるか(D1を基準)を格納します。



- ・ 演算後、キャリーフラグ(007356)がONします。

**F-173
CHNG**

**データチェンジ(1バイト)
(CHaNG)**

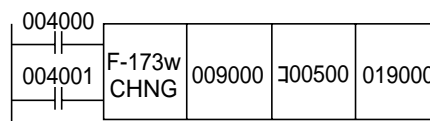
シンボル	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="text-align: center;">F-173 CHNG</td> <td style="text-align: center;">S</td> <td style="text-align: center;">D1</td> <td style="text-align: center;">D2</td> </tr> </table> モード指定 実行入力	F-173 CHNG	S	D1	D2	[使用例] <table border="1" style="float: right; margin-top: 10px;"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004000</td></tr> <tr><td>STR</td><td>004001</td></tr> <tr><td>F-173</td><td></td></tr> <tr><td></td><td>009000</td></tr> <tr><td></td><td>000500</td></tr> <tr><td></td><td>019000</td></tr> </table>	命 令		STR	004000	STR	004001	F-173			009000		000500		019000																																																																				
F-173 CHNG	S	D1	D2																																																																																					
命 令																																																																																								
STR	004000																																																																																							
STR	004001																																																																																							
F-173																																																																																								
	009000																																																																																							
	000500																																																																																							
	019000																																																																																							
機能	<p>レジスタD1を先頭アドレスとし、バイト数(D2の内容)で指示する領域内で検索する。レジスタSの内容が検索データです。</p> <p>データを検索して最初の一致アドレスが何バイト目(D1を基準)であるかをD2+2へ格納する。また、一致した個数をD2+1へ格納する。</p> <p>書き換えるデータはレジスタS+1の内容で、モード指定 入力の条件で書き換える。</p> <ul style="list-style-type: none"> モード指定 入力OFFのとき 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 入力ONのとき すべての一致アドレスのデータを書き換える。 	<ul style="list-style-type: none"> 実行入力条件004001がOFF ONの変化時に演算します。 009000の内容(55(H))が検索データです。 先頭アドレスはレジスタ000500です。 終了アドレスは、000500から010(D)バイト(レジスタ019000の内容)の位置です。バイト数には先頭アドレスも含まれます。 レジスタ019001に検索個数を格納します。 レジスタ019002に、最初の一致アドレスが何バイト目であるか(D1を基準)を格納します。 書換データは、レジスタ009001の内容(64(H))です。 下記の例は004000がONの動作です。 																																																																																						
演算内容	<p>・ D1、D2で設定するレジスタ領域が、タイマ・カウンタの接点領域(ファイルアドレス00001600 ~ 00001777(8)等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。</p>	<table border="1" style="margin-bottom: 10px;"> <tr> <td>009000</td> <td>55</td> <td>検索データ</td> <td>019000</td> <td>010(D)</td> <td>バイト数</td> </tr> <tr> <td>009001</td> <td>64</td> <td>書換データ</td> <td>019001</td> <td>004(8)</td> <td>個数</td> </tr> <tr> <td></td> <td></td> <td></td> <td>019002</td> <td>002(8)</td> <td>バイト目</td> </tr> </table> <table border="1" style="margin-bottom: 10px;"> <thead> <tr> <th colspan="2"></th> <th>演算前</th> <th>→</th> <th>演算後</th> <th colspan="2"></th> </tr> </thead> <tbody> <tr> <td rowspan="10" style="vertical-align: middle;">10 バ イ ト</td> <td>先頭アドレス</td> <td>000500</td> <td>11</td> <td>→</td> <td>11</td> <td>000500</td> </tr> <tr> <td></td> <td>000501</td> <td>22</td> <td>→</td> <td>22</td> <td>000501</td> </tr> <tr> <td>一致アドレス (最初)</td> <td>000502</td> <td>55</td> <td>→</td> <td>64</td> <td>000502</td> </tr> <tr> <td></td> <td>000503</td> <td>33</td> <td>→</td> <td>33</td> <td>000503</td> </tr> <tr> <td></td> <td>000504</td> <td>44</td> <td>→</td> <td>44</td> <td>000504</td> </tr> <tr> <td></td> <td>000505</td> <td>55</td> <td>→</td> <td>64</td> <td>000505</td> </tr> <tr> <td></td> <td>000506</td> <td>66</td> <td>→</td> <td>66</td> <td>000506</td> </tr> <tr> <td></td> <td>000507</td> <td>55</td> <td>→</td> <td>64</td> <td>000507</td> </tr> <tr> <td></td> <td>000510</td> <td>77</td> <td>→</td> <td>77</td> <td>000510</td> </tr> <tr> <td>終了アドレス</td> <td>000511</td> <td>55</td> <td>→</td> <td>64</td> <td>000511</td> </tr> </tbody> </table> <p>・ 演算後、キャリーフラグ(007356)がONする。</p>	009000	55	検索データ	019000	010(D)	バイト数	009001	64	書換データ	019001	004(8)	個数				019002	002(8)	バイト目			演算前	→	演算後			10 バ イ ト	先頭アドレス	000500	11	→	11	000500		000501	22	→	22	000501	一致アドレス (最初)	000502	55	→	64	000502		000503	33	→	33	000503		000504	44	→	44	000504		000505	55	→	64	000505		000506	66	→	66	000506		000507	55	→	64	000507		000510	77	→	77	000510	終了アドレス	000511	55	→	64	000511
009000	55	検索データ	019000	010(D)	バイト数																																																																																			
009001	64	書換データ	019001	004(8)	個数																																																																																			
			019002	002(8)	バイト目																																																																																			
		演算前	→	演算後																																																																																				
10 バ イ ト	先頭アドレス	000500	11	→	11	000500																																																																																		
		000501	22	→	22	000501																																																																																		
	一致アドレス (最初)	000502	55	→	64	000502																																																																																		
		000503	33	→	33	000503																																																																																		
		000504	44	→	44	000504																																																																																		
		000505	55	→	64	000505																																																																																		
		000506	66	→	66	000506																																																																																		
		000507	55	→	64	000507																																																																																		
		000510	77	→	77	000510																																																																																		
	終了アドレス	000511	55	→	64	000511																																																																																		
S	使用範囲 B																																																																																							
D1	使用範囲 A																																																																																							
D2	使用範囲 E <ul style="list-style-type: none"> レジスタD2に設定する内容は000 ~ 377(8)です。(000(8)のとき256バイト) レジスタD2+1の内容は000 ~ 377(8)です。000(8)のとき、一致個数が0個または256個です。キャリーフラグ(007356)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を基準として何バイト目であるかの値が格納されます。(000(8)のとき先頭アドレス) 	<table border="1" style="width: 100%;"> <tr> <td rowspan="3" style="writing-mode: vertical-rl; text-orientation: upright;">演算後の内容</td> <td>S、S+1</td> <td colspan="4">不変</td> </tr> <tr> <td>D2</td> <td colspan="4">不変</td> </tr> <tr> <td>D2+1 D2+2</td> <td colspan="4">演算結果 ・ 一致個数が0個または256個のとき、D2+1は000(8)となる。</td> </tr> <tr> <td rowspan="2">フラグ</td> <td>演算結果</td> <td>ゼロ 007357</td> <td>キャリー 007356</td> <td>エラー 007355</td> <td>ノンキャリー 007354</td> </tr> <tr> <td></td> <td>有</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td></td> <td>無</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>	演算後の内容	S、S+1	不変				D2	不変				D2+1 D2+2	演算結果 ・ 一致個数が0個または256個のとき、D2+1は000(8)となる。				フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354		有	0	1	0		無	1	0	0	0																																																					
演算後の内容	S、S+1	不変																																																																																						
	D2	不変																																																																																						
	D2+1 D2+2	演算結果 ・ 一致個数が0個または256個のとき、D2+1は000(8)となる。																																																																																						
フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354																																																																																			
		有	0	1	0																																																																																			
	無	1	0	0	0																																																																																			
演算条件	入力信号(実行入力)の立上り(OFF ON)																																																																																							

**F-173w
CHNG**

**データチェンジ(1ワード)
(CHaNG)**

シンボル	<table border="1"> <tr> <td>F-173w CHNG</td> <td>S</td> <td>D1</td> <td>D2</td> </tr> </table> モード指定 実行入力	F-173w CHNG	S	D1	D2
F-173w CHNG	S	D1	D2		
機能	<p>レジスタD1を先頭アドレスとし、ワード数(D2の内容)で指示する領域内で検索する。レジスタS、S+1の内容が検索データです。</p> <p>データを検索して最初の一致アドレスが何ワード目(D1を基準)であるかをD2+2へ格納する。また、一致した個数をD2+1へ格納する。</p> <p>書き換えるデータはレジスタS+2、S+3の内容で、モード指定 入力の条件で書き換える。</p> <ul style="list-style-type: none"> モード指定 入力OFFのとき 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 入力ONのとき すべての一致アドレスのデータを書き換える。 				
演算内容	<p> $D1+2(D2-1)$ (終了アドレス) </p> <ul style="list-style-type: none"> D1、D2で設定するレジスタ領域が、タイマ・カウンタの接点領域(ファイルアドレス00001600~00001777₍₈₎等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。 				
S	使用範囲 C ・Sには必ず偶数アドレスを設定してください。				
D1	使用範囲 B ・D1には必ず偶数アドレスを設定してください。				
D2	使用範囲 E ・レジスタD2に設定する内容は000~377 ₍₈₎ です。(000 ₍₈₎ のとき256ワード) ・レジスタD2+1の内容は000~377 ₍₈₎ です。000 ₍₈₎ のとき、一致個数が0個または256個です。キャリーフラグ(007356)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を基準として何ワード目であるかの値が格納されます。(000 ₍₈₎ のとき先頭アドレス)				
演算条件	入力信号(実行入力)の立上り(OFF ON)				

[使用例]



命 令	
STR	004000
STR	004001
F-173w	009000
	00500
	019000

- ・実行入力条件004001がOFF ONの変化時に演算します。
- ・009000、009001の内容(3355_(H))が検索データです。
- ・先頭アドレスはレジスタ00500です。
- ・終了アドレスは、00500から005₍₈₎ワード(レジスタ019000の内容)の位置です。ワード数には先頭アドレスも含まれます。
- ・レジスタ019001に検索個数を格納します。
- ・レジスタ019002に、最初の一致アドレスが何ワード目であるか(D1を基準)を格納します。
- ・書換データは、レジスタ009002、009003の内容(1964_(H))です。
- ・下記の例は004000がONの動作です。

009000	55 下位	検索データ	019000	005 ₍₈₎	ワード数
009001	33 上位		019001	001 ₍₈₎	個数
009002	64 下位	書換データ	019002	001 ₍₈₎	ワード目
009003	19 上位				

		演算前	→	演算後
5 ワ ド	先頭 アドレス	(00500 00501)		11 22
	最初の一致 アドレス	(00502 00503)	書き換え	55 33
		00504 00505 00506 00507		44 55 66 55
	終了 アドレス	(00510 00511)		77 55

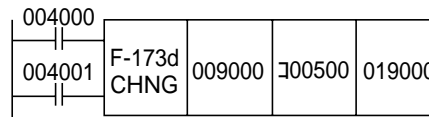
演算後の内容	S ~ S+3	不変				
	D2	不変				
	D2+1 D2+2	演算結果 ・一致個数が0個または256個のとき、D2+1は000 ₍₈₎ となる。				
	フ ラ グ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		有	0	1	0	0
		無	1	0		

**F-173d
CHNG**

**データチェンジ(2ワード)
(CHaNG)**

シンボル	<table border="1"> <tr> <td>F-173d CHNG</td> <td>S</td> <td>D1</td> <td>D2</td> </tr> </table> モード指定 実行入力	F-173d CHNG	S	D1	D2
F-173d CHNG	S	D1	D2		
機能	<p>レジスタD1を先頭アドレスとし、ダブルワード数(D2の内容)で指示する領域内で検索する。レジスタS~S+3の内容が検索データです。</p> <p>データを検索して最初の一致アドレスが何ダブルワード目(D1を基準)であるかをD2+2へ格納する。また、一致した個数をD2+1へ格納する。</p> <p>書き換えるデータはレジスタS+4~S+7の内容で、モード指定 入力ONの条件で書き換える。</p> <ul style="list-style-type: none"> モード指定 入力OFFのとき 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 入力ONのとき すべての一致アドレスのデータを書き換える。 				
演算内容	<p>D1 (先頭アドレス) D1+4 D1+10(8) D2 (ダブルワード数) D2+1 一致個数 D2+2 一致アドレス D1+4 (D2-1) (終了アドレス)</p> <ul style="list-style-type: none"> D1、D2で設定するレジスタ領域が、タイマ・カウンタの接点領域(ファイルアドレス00001600~00001777(8)等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。 				
S	使用範囲G ・Sには必ず偶数アドレスを設定してください。				
D1	使用範囲C ・D1には必ず偶数アドレスを設定してください。				
D2	使用範囲E ・レジスタD2に設定する内容は000~377(8)です。(000(8)のとき256ダブルワード) ・レジスタD2+1の内容は000~377(8)です。000(8)のとき、一致個数が0個または256個です。キャリーフラグ(007356)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を基準として何ダブルワード目であるかの値が格納されます。(000(8)のとき先頭アドレス)				
演算条件	入力信号(実行入力)の立上り(OFF ON)				

[使用例]



命 令	
STR	004000
STR	004001
F-173d	
	009000
	000500
	019000

- ・実行入力条件004001がOFF ONの変化時に演算します。
- ・009000~009003の内容(99223355(H))が検索データです。
- ・先頭アドレスはレジスタ000500です。
- ・終了アドレスは、000500から004(8)ダブルワード(レジスタ019000の内容)の位置です。ダブルワード数には先頭アドレスも含めます。
- ・レジスタ019001に検索個数を格納します。
- ・レジスタ019002に、最初の一致アドレスが何ダブルワード目であるか(D1を基準)を格納します。
- ・書換データは、レジスタ009004~009007の内容(19643385(H))です。
- ・下記の例は004000がONの動作です。

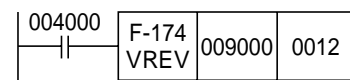
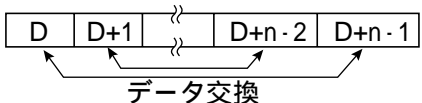
009000	55(下位)	019000	004(8)	ダブルワード数
009001	33	019001	001(8)	個数
009002	22	019002	001(8)	ダブルワード目
009003	99(上位)			
009004	85(下位)			
009005	33			
009006	64			
009007	19(上位)			

		演算前	→	演算後
4 ダ ブ ル ワ ー ド	先頭 アドレス	000500	11	11
		000501	22	22
		000502	55	55
		000503	33	33
	最初の一 致 ア ド レ ス	000504	55	85
		000505	33	33
		000506	22	64
		000507	99	19
	終了 ア ド レ ス	000510	44	44
		000511	55	55
		000512	66	66
		000513	55	55
		000514	22	22
		000515	33	33
		000516	77	77
		000517	88	88

演算後の 内 容	S~S+7	不変				
	D2	不変				
	D2+1 D2+2	演算結果 ・一致個数が0個または256個のとき、 D2+1は000(8)となる。				
	フ ラ グ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	有	0	1	0	0	
	無	1	0			

第 14 章 応用命令 (F-174 ~ F-403)

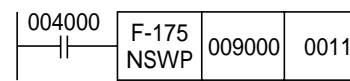
F-174 VREV レジスタ間 (1 バイト) データ交換 (Vertical REVerse)

シンボル	— F-174 VREV D n	[使用例]	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004000</td></tr> <tr><td>F-174</td><td>009000</td></tr> <tr><td></td><td>0012</td></tr> </table>	命 令		STR	004000	F-174	009000		0012																																			
命 令																																														
STR	004000																																													
F-174	009000																																													
	0012																																													
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、各レジスタの内容(順序)について、「最下位アドレス 最上位アドレス」から「最上位アドレス 最下位アドレス」へ入れ換える。	 <p>入力条件004000がOFF ONの変化時に、レジスタ009000 ~ 009011の10(0012⁽⁸⁾)バイトの内容(1バイトデータ)を、「009000 009011」から「009011 009000」へ入れ換えます。</p>																																												
演算内容		<table style="margin-left: auto; margin-right: auto;"> <tr><th colspan="2">演算前</th><th colspan="2">演算後</th></tr> <tr><td>009000</td><td>10</td><td>23</td><td></td></tr> <tr><td>009001</td><td>32</td><td>01</td><td></td></tr> <tr><td>009002</td><td>54</td><td>15</td><td></td></tr> <tr><td>009003</td><td>76</td><td>26</td><td></td></tr> <tr><td>009004</td><td>98</td><td>37</td><td></td></tr> <tr><td>009005</td><td>37</td><td>98</td><td></td></tr> <tr><td>009006</td><td>26</td><td>76</td><td></td></tr> <tr><td>009007</td><td>15</td><td>54</td><td></td></tr> <tr><td>009010</td><td>01</td><td>32</td><td></td></tr> <tr><td>009011</td><td>23</td><td>10</td><td></td></tr> </table>	演算前		演算後		009000	10	23		009001	32	01		009002	54	15		009003	76	26		009004	98	37		009005	37	98		009006	26	76		009007	15	54		009010	01	32		009011	23	10	
演算前		演算後																																												
009000	10	23																																												
009001	32	01																																												
009002	54	15																																												
009003	76	26																																												
009004	98	37																																												
009005	37	98																																												
009006	26	76																																												
009007	15	54																																												
009010	01	32																																												
009011	23	10																																												
D	使用範囲 A																																													
n	使用範囲 0000 ~ 1777 ⁽⁸⁾ (0000 ⁽⁸⁾ のとき1024バイト)																																													
演算条件	入力信号の立上り(OFF ON)																																													
演算後の内容	D ~ D+n-1	演算結果																																												
	フ ラ グ	不変																																												

D、nで設定するレジスタ領域が、タイマ・カウンタの接点領域(ファイルアドレス00001600 ~ 00001777⁽⁸⁾等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。

(類似命令) F-02、F-02w、F-02d

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	— F-175 NSWP D n	[使用例]	<table border="1" style="margin-left: auto; margin-right: auto;"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004000</td></tr> <tr><td>F-175</td><td>009000</td></tr> <tr><td></td><td>0011</td></tr> </table>	命 令		STR	004000	F-175	009000		0011																															
命 令																																										
STR	004000																																									
F-175	009000																																									
	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、各レジスタのデータ上位4ビットと下位4ビットの内容を交換する。	 <p>入力条件004000がOFF ONの変化時に、レジスタ009000 ~ 009010の9(0011⁽⁸⁾)バイトの内容(1バイトデータ)をニブル交換します。</p>																																								
演算内容	D ~ D+n-1のニブル交換	<table style="margin-left: auto; margin-right: auto;"> <tr><th colspan="2">演算前</th><th colspan="2">演算後</th></tr> <tr><td>009000</td><td>9 : 1</td><td>1 : 9</td><td></td></tr> <tr><td>009001</td><td>8 : 2</td><td>2 : 8</td><td></td></tr> <tr><td>009002</td><td>7 : 3</td><td>3 : 7</td><td></td></tr> <tr><td>009003</td><td>6 : 4</td><td>4 : 6</td><td></td></tr> <tr><td>009004</td><td>0 : 5</td><td>5 : 0</td><td></td></tr> <tr><td>009005</td><td>4 : 6</td><td>6 : 4</td><td></td></tr> <tr><td>009006</td><td>3 : 7</td><td>7 : 3</td><td></td></tr> <tr><td>009007</td><td>2 : 8</td><td>8 : 2</td><td></td></tr> <tr><td>009010</td><td>1 : 9</td><td>9 : 1</td><td></td></tr> </table>	演算前		演算後		009000	9 : 1	1 : 9		009001	8 : 2	2 : 8		009002	7 : 3	3 : 7		009003	6 : 4	4 : 6		009004	0 : 5	5 : 0		009005	4 : 6	6 : 4		009006	3 : 7	7 : 3		009007	2 : 8	8 : 2		009010	1 : 9	9 : 1	
演算前		演算後																																								
009000	9 : 1	1 : 9																																								
009001	8 : 2	2 : 8																																								
009002	7 : 3	3 : 7																																								
009003	6 : 4	4 : 6																																								
009004	0 : 5	5 : 0																																								
009005	4 : 6	6 : 4																																								
009006	3 : 7	7 : 3																																								
009007	2 : 8	8 : 2																																								
009010	1 : 9	9 : 1																																								
D	使用範囲 A																																									
n	使用範囲 0000 ~ 1777 ⁽⁸⁾ (0000 ⁽⁸⁾ のとき1024バイト)																																									
演算条件	入力信号の立上り(OFF ON)																																									
演算後の内容	D ~ D+n-1	演算結果																																								
	フ ラ グ	不変																																								

D、nで設定するレジスタ領域が、タイマ・カウンタの接点領域(ファイルアドレス00001600 ~ 00001777⁽⁸⁾等)内、およびレジスタ、ファイルレジスタの最終アドレス以降に入らないようにしてください。

(類似命令) F-55

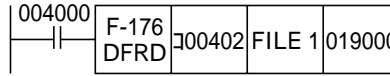
**F-176
DFRD**

**指定アドレスのレジスタからの続出(256バイト)
(Direct File Read)**

シンボル	F-176 DFRD	S	FILE F	D
機能	ファイル番号F内にて、レジスタS、S+1の内容で指定するブロック番号のデータ(256バイト)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	ファイル番号F、S、S+1(ブロック番号) → Dへ256バイト転送			
S	使用範囲B ・Sには必ず偶数アドレスを設定してください。			
F	0(ファイルレジスタを除くデータメモリのとき) 1(ファイルレジスタのとき)			
D	使用範囲J ・Dには必ず偶数アドレスを設定してください。			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S~S+255	不変		
	D~D+255	演算結果		
	フラグ	不変		

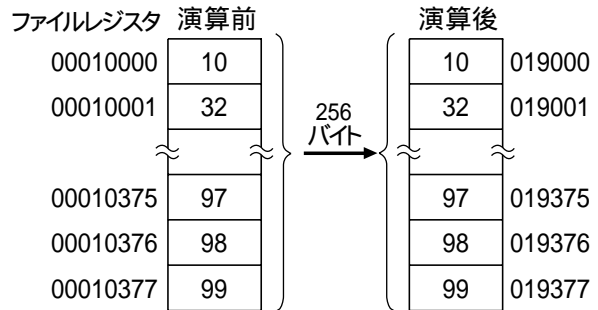
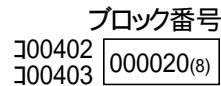
(類似命令) F-102、F-102w、F-102d

[使用例]



命 令	
STR	004000
F-176	00402
	FILE 1
	019000

入力条件004000がOFF ONの変化時に、ファイル番号1(ファイルレジスタ)のブロック番号000020⁽⁸⁾(レジスタ00402、00403の内容)から、レジスタ019000を先頭とする256バイトにデータをブロック転送します。



・ファイル番号1(ファイルレジスタ)のブロック番号000020⁽⁸⁾の先頭アドレスは、00010000⁽⁸⁾です。

S、S+1の内容は、ファイル番号0(ファイルアドレス)、ファイル番号1(バイトアドレス)を、それぞれ256バイト単位で区切ったブロック番号(000000~077777⁽⁸⁾)を使用します。

ブロック番号とデータメモリ(14・4~5ページ)参照

本命令では256バイトのデータをブロック転送します。

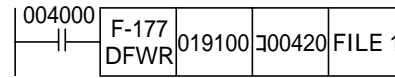
Dで設定する先頭アドレスに注意願います。そのアドレス以後256バイトが転送領域になります。

F-177 DFWR 指定アドレスのレジスタへ書込(256バイト)
(Direct File WRite)

シンボル	— F-177 DFWR S D FILE F			
機能	レジスタSを先頭アドレスとする256バイトのデータを、ファイル番号F内にて、レジスタD、D+1の内容で指定するブロック番号の領域へ転送する。			
演算内容	S → ファイル番号FのD、D+1 (ブロック番号)へ256バイト転送			
S	使用範囲B ・Sには必ず偶数アドレスを設定してください。			
D	使用範囲B			
F	0 (ファイルレジスタを除くデータメモリのとき) 1 (ファイルレジスタのとき)			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S~S+255	不変		
	D~D+255	演算結果		
	フラグ	不変		

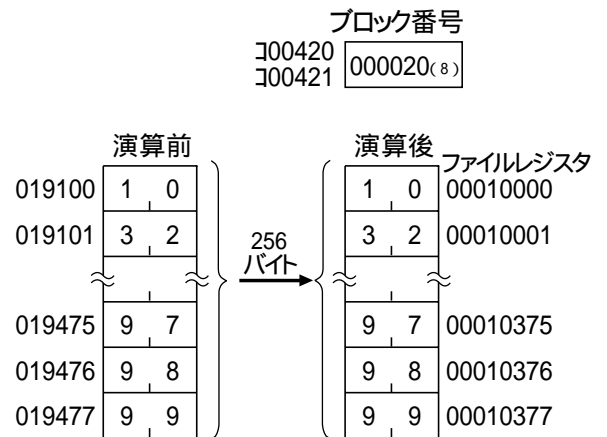
(類似命令) F-103、F-103w、F-103d

[使用例]



命 令	
STR	004000
F-177	019100
	000420
	FILE 1

入力条件004000がOFF ONの変化時に、レジスタ019100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(ファイルレジスタ)のブロック番号000020₍₈₎(レジスタ000420、000421の内容)で指定する領域へブロック転送します。



・ファイル番号1(ファイルレジスタ)のブロック番号000020₍₈₎の先頭アドレスは、00010000₍₈₎です。

D、D+1の内容は、ファイル番号0(ファイルアドレス)、ファイル番号1(バイトアドレス)を、それぞれ256バイト単位で区切ったブロック番号(000000~077777₍₈₎)を使用します。

ブロック番号とデータメモリ(14・4~5ページ)参照

本命令では256バイトのデータをブロック転送します。

Dのブロック番号設定では、ファイル番号0のタイマ・カウンタの接点領域を含むブロック番号₍₈₎3、73、203(JW-321CU/322CU)、301~307(JW-331CU~362CU)を使用しないでください。

ブロック番号とデータメモリ(先頭アドレス)

応用命令 F-176/177(14・2～3ページ)で設定するブロック番号とデータメモリ(先頭アドレス)の関係は以下のとおりです。

ファイル番号 0 のとき(FILEF = 0 に設定時)

データメモリ(ファイルレジスタを除く)の先頭アドレスは、ファイルアドレスで記載しています。

ブロック番号(8)	先頭ファイルアドレス(8)	ブロック番号(8)	先頭ファイルアドレス(8)	ブロック番号(8)	先頭ファイルアドレス(8)	ブロック番号(8)	先頭ファイルアドレス(8)
000000	00000000	000100	00040000	000200	00100000	000300	00140000
000001	00000400	000101	00040400	000201	00100400	* 000301	00140400
000002	00001000	000102	00041000	000202	00101000	* 000302	00141000
* 000003	00001400	000103	00041400	* 000203	00101400	* 000303	00141400
000004	00002000	000104	00042000	000204	00102000	* 000304	00142000
000005	00002400	000105	00042400	000205	00102400	* 000305	00142400
000006	00003000	000106	00043000	000206	00103000	* 000306	00143000
000007	00003400	000107	00043400	000207	00103400	* 000307	00143400
000010	00004000	000110	00044000	000210	00104000	000310	00144000
000011	00004400	000111	00044400	000211	00104400	000311	00144400
000012	00005000	000112	00045000	000212	00105000	000312	00145000
000013	00005400	000113	00045400	000213	00105400	000313	00145400
000014	00006000	000114	00046000	000214	00106000	000314	00146000
000015	00006400	000115	00046400	000215	00106400	000315	00146400
000016	00007000	000116	00047000	000216	00107000	000316	00147000
000017	00007400	000117	00047400	000217	00107400	000317	00147400
000020	00010000	000120	00050000	000220	00110000	000320	00150000
000021	00010400	000121	00050400	000221	00110400	000321	00150400
000022	00011000	000122	00051000	000222	00111000	000322	00151000
000023	00011400	000123	00051400	000223	00111400	000323	00151400
000024	00012000	000124	00052000	000224	00112000	000324	00152000
000025	00012400	000125	00052400	000225	00112400	000325	00152400
000026	00013000	000126	00053000	000226	00113000	000326	00153000
000027	00013400	000127	00053400	000227	00113400	000327	00153400
000030	00014000	000130	00054000	000230	00114000	000330	00154000
000031	00014400	000131	00054400	000231	00114400	000331	00154400
000032	00015000	000132	00055000	000232	00115000	000332	00155000
000033	00015400	000133	00055400	000233	00115400	000333	00155400
000034	00016000	000134	00056000	000234	00116000	000334	00156000
000035	00016400	000135	00056400	000235	00116400	000335	00156400
000036	00017000	000136	00057000	000236	00117000	000336	00157000
000037	00017400	000137	00057400	000237	00117400	000337	00157400
000040	00020000	000140	00060000	000240	00120000	000340	00160000
000041	00020400	000141	00060400	000241	00120400	000341	00160400
000042	00021000	000142	00061000	000242	00121000	000342	00161000
000043	00021400	000143	00061400	000243	00121400	000343	00161400
000044	00022000	000144	00062000	000244	00122000	000344	00162000
000045	00022400	000145	00062400	000245	00122400	000345	00162400
000046	00023000	000146	00063000	000246	00123000	000346	00163000
000047	00023400	000147	00063400	000247	00123400	000347	00163400
000050	00024000	000150	00064000	000250	00124000	000350	00164000
000051	00024400	000151	00064400	000251	00124400	000351	00164400
000052	00025000	000152	00065000	000252	00125000	000352	00165000
000053	00025400	000153	00065400	000253	00125400	000353	00165400
000054	00026000	000154	00066000	000254	00126000	000354	00166000
000055	00026400	000155	00066400	000255	00126400	000355	00166400
000056	00027000	000156	00067000	000256	00127000	000356	00167000
000057	00027400	000157	00067400	000257	00127400	000357	00167400
000060	00030000	000160	00070000	000260	00130000	000360	00170000
000061	00030400	000161	00070400	000261	00130400	000361	00170400
000062	00031000	000162	00071000	000262	00131000	000362	00171000
000063	00031400	000163	00071400	000263	00131400	000363	00171400
000064	00032000	000164	00072000	000264	00132000	000364	00172000
000065	00032400	000165	00072400	000265	00132400	000365	00172400
000066	00033000	000166	00073000	000266	00133000	000366	00173000
000067	00033400	000167	00073400	000267	00133400	000367	00173400
000070	00034000	000170	00074000	000270	00134000	000370	00174000
000071	00034400	000171	00074400	000271	00134400	000371	00174400
000072	00035000	000172	00075000	000272	00135000	000372	00175000
* 000073	00035400	000173	00075400	000273	00135400	000373	00175400
000074	00036000	000174	00076000	000274	00136000	000374	00176000
000075	00036400	000175	00076400	000275	00136400	000375	00176400
000076	00037000	000176	00077000	000276	00137000	000376	00177000
000077	00037400	000177	00077400	000277	00137400	000377	00177400

- ・ブロック番号(8)の設定範囲 - - - JW-331CU/332CU/341CU/342CU/352CU/362CU : 000000 ~ 000377、
JW-311CU/312CU : 000000 ~ 000167(上記 1)、 JW-321CU/322CU : 000000 ~ 000213(上記 2)
- ・使用禁止のブロック番号(8)(上記 *) - - - 000003、000073、000203、000301 ~ 000307

ファイル番号 1 のとき(FILEF = 1 に設定時)

ファイルレジスタの先頭アドレスは、 バイトアドレスで記載しています。 2・1ページ参照

ブロック番号 ₍₈₎	先頭バイトアドレス ₍₈₎
000000	00000000
⋮	⋮
000177	00077400
⋮	⋮
000777	00377400
⋮	⋮
003777	01777400
⋮	⋮
017777	07777400
⋮	⋮
077777	37777400

・ブロック番号₍₈₎の設定範囲

JW-321CU/322CU : 000000 ~ 000177

JW-331CU/332CU : 000000 ~ 000777

JW-341CU/342CU : 000000 ~ 003777

JW-352CU : 000000 ~ 017777

JW-362CU : 000000 ~ 077777

・先頭バイトアドレスの計算式

ブロック番号 × 256(400₍₈₎) = 先頭バイトアドレス

レジスタ間(1バイト)の比較(リレー出力付き)

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> S1 S2 BIT				— F-183 CP>= S1 S2 BIT																		
	— F-181 CP< S1 S2 BIT				— F-184 CP<= S1 S2 BIT																		
	— F-182 CP= S1 S2 BIT				— F-185 CP<> S1 S2 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	S1 > S2 → BIT ON			F-183	S1 S2 → BIT ON																	
	F-181	S1 < S2 → BIT ON			F-184	S1 S2 → BIT ON																	
	F-182	S1 = S2 → BIT ON			F-185	S1 S2 → BIT ON																	
S1	使用範囲 A				S2	使用範囲 A																	
BIT	使用範囲 K				演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)																	
演算後の内容	S1、S2	不変																					
	BIT	F-180	S1 > S2のとき、ON S1 S2のとき、OFF			F-183	S1 S2のとき、ON S1 < S2のとき、OFF																
		F-181	S1 < S2のとき、ON S1 S2のとき、OFF			F-184	S1 S2のとき、ON S1 > S2のとき、OFF																
		F-182	S1 = S2のとき、ON S1 S2のとき、OFF			F-185	S1 S2のとき、ON S1 = S2のとき、OFF																
フラグ	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	—————																		
[使用例]		<table border="1"> <tr> <td>004001</td> <td>F-180 CP></td> <td>009000</td> <td>009010</td> <td>000200</td> </tr> </table>			004001	F-180 CP>	009000	009010	000200	<table border="1"> <tr> <td>命 令</td> <td>004001</td> </tr> <tr> <td>STR</td> <td>009000</td> </tr> <tr> <td>F-180</td> <td>009010</td> </tr> <tr> <td></td> <td>000200</td> </tr> </table>		命 令	004001	STR	009000	F-180	009010		000200	入力条件004001がONのときに、レジスタ009000とレジスタ009010の内容を比較して、(009000)>(009010)のときにリレー000200がONします。 入力条件004001がOFFのとき、および(009000)<(009010)のときには、リレー000200はOFFします。			
004001	F-180 CP>	009000	009010	000200																			
命 令	004001																						
STR	009000																						
F-180	009010																						
	000200																						

レジスタ間(1ワード)の比較(リレー出力付き)

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 CP></td><td>S1</td><td>S2</td><td>BIT</td></tr> <tr><td>F-181w CP<</td><td>S1</td><td>S2</td><td>BIT</td></tr> <tr><td>F-182w CP=</td><td>S1</td><td>S2</td><td>BIT</td></tr> </table>		F-180w CP>	S1	S2	BIT	F-181w CP<	S1	S2	BIT	F-182w CP=	S1	S2	BIT	<table border="1"> <tr><td>F-183w CP>=</td><td>S1</td><td>S2</td><td>BIT</td></tr> <tr><td>F-184w CP<=</td><td>S1</td><td>S2</td><td>BIT</td></tr> <tr><td>F-185w CP<></td><td>S1</td><td>S2</td><td>BIT</td></tr> </table>		F-183w CP>=	S1	S2	BIT	F-184w CP<=	S1	S2	BIT	F-185w CP<>	S1	S2	BIT
F-180w CP>	S1	S2	BIT																									
F-181w CP<	S1	S2	BIT																									
F-182w CP=	S1	S2	BIT																									
F-183w CP>=	S1	S2	BIT																									
F-184w CP<=	S1	S2	BIT																									
F-185w CP<>	S1	S2	BIT																									
機能	レジスタS1, S1+1の内容(1ワードデータ)と、 S2, S2+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	S1, S1+1 > S2, S2+1 → BIT ON		F-183w	S1, S1+1 S2, S2+1 → BIT ON																							
	F-181w	S1, S1+1 < S2, S2+1 → BIT ON		F-184w	S1, S1+1 S2, S2+1 → BIT ON																							
	F-182w	S1, S1+1 = S2, S2+1 → BIT ON		F-185w	S1, S1+1 S2, S2+1 → BIT ON																							
S1	使用範囲 B			S2	使用範囲 B																							
BIT	使用範囲 K			演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)																							
演算後の内容	S1, S1+1 S2, S2+1	不変																										
	BIT	F-180w	S1, S1+1 > S2, S2+1のとき、 ON S1, S1+1 S2, S2+1のとき、 OFF		F-183w	S1, S1+1 S2, S2+1のとき、 ON S1, S1+1 < S2, S2+1のとき、 OFF																						
		F-181w	S1, S1+1 < S2, S2+1のとき、 ON S1, S1+1 S2, S2+1のとき、 OFF		F-184w	S1, S1+1 S2, S2+1のとき、 ON S1, S1+1 > S2, S2+1のとき、 OFF																						
		F-182w	S1, S1+1 = S2, S2+1のとき、 ON S1, S1+1 S2, S2+1のとき、 OFF		F-185w	S1, S1+1 S2, S2+1のとき、 ON S1, S1+1 = S2, S2+1のとき、 OFF																						
フラグ	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	_____																							
[使用例]		<table border="1"> <tr><td>004201</td><td>F-180w CP></td><td>009100</td><td>009110</td><td>000100</td></tr> </table>		004201	F-180w CP>	009100	009110	000100	<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004201</td></tr> <tr><td>F-180w</td><td>009100</td></tr> <tr><td></td><td>009110</td></tr> <tr><td></td><td>000100</td></tr> </table>		命 令		STR	004201	F-180w	009100		009110		000100	入力条件004201がONのときに、レジスタ009100、009101の内容とレジスタ009110、009111の内容を比較して、(009100, 009101) > (009110, 009111) のときにリレー000100がONします。 入力条件004201がOFFのとき、および(009100, 009101) (009011, 009111) のときには、リレー000100はOFFになります。							
004201	F-180w CP>	009100	009110	000100																								
命 令																												
STR	004201																											
F-180w	009100																											
	009110																											
	000100																											

S1, S2には必ず偶数アドレスを設定してください。(019003等は禁止)

レジスタ間(2ワード)の比較(リレー出力付き)

F-180d CP >	(ComPare >)	F-183d CP > =	(ComPare > =)
F-181d CP <	(ComPare <)	F-184d CP < =	(ComPare < =)
F-182d CP =	(ComPare =)	F-185d CP < >	(ComPare < >)

シンボル	<table border="1"> <tr> <td>F-180d CP></td> <td>S1</td> <td>S2</td> <td>BIT</td> </tr> <tr> <td>F-181d CP<</td> <td>S1</td> <td>S2</td> <td>BIT</td> </tr> <tr> <td>F-182d CP=</td> <td>S1</td> <td>S2</td> <td>BIT</td> </tr> </table>				F-180d CP>	S1	S2	BIT	F-181d CP<	S1	S2	BIT	F-182d CP=	S1	S2	BIT	<table border="1"> <tr> <td>F-183d CP>=</td> <td>S1</td> <td>S2</td> <td>BIT</td> </tr> <tr> <td>F-184d CP<=</td> <td>S1</td> <td>S2</td> <td>BIT</td> </tr> <tr> <td>F-185d CP<></td> <td>S1</td> <td>S2</td> <td>BIT</td> </tr> </table>				F-183d CP>=	S1	S2	BIT	F-184d CP<=	S1	S2	BIT	F-185d CP<>	S1	S2	BIT
F-180d CP>	S1	S2	BIT																													
F-181d CP<	S1	S2	BIT																													
F-182d CP=	S1	S2	BIT																													
F-183d CP>=	S1	S2	BIT																													
F-184d CP<=	S1	S2	BIT																													
F-185d CP<>	S1	S2	BIT																													
機能	レジスタS1~S1+3の内容(2ワードデータ)と、S2~S2+3の内容(2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-180d	S1~S1+3 > S2~S2+3 → BIT ON			F-183d	S1~S1+3 S2~S2+3 → BIT ON																										
	F-181d	S1~S1+3 < S2~S2+3 → BIT ON			F-184d	S1~S1+3 S2~S2+3 → BIT ON																										
	F-182d	S1~S1+3 = S2~S2+3 → BIT ON			F-185d	S1~S1+3 S2~S2+3 → BIT ON																										
S1	使用範囲 C				S2	使用範囲 C																										
BIT	使用範囲 K				演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)																										
演算後の内容	S1~S1+3 S2~S2+3	不変																														
	BIT	F-180d	S1~S1+3 > S2~S2+3のとき、ON S1~S1+3 S2~S2+3のとき、OFF			F-183d	S1~S1+3 S2~S2+3のとき、ON S1~S1+3 < S2~S2+3のとき、OFF																									
		F-181d	S1~S1+3 < S2~S2+3のとき、ON S1~S1+3 S2~S2+3のとき、OFF			F-184d	S1~S1+3 S2~S2+3のとき、ON S1~S1+3 > S2~S2+3のとき、OFF																									
		F-182d	S1~S1+3 = S2~S2+3のとき、ON S1~S1+3 S2~S2+3のとき、OFF			F-185d	S1~S1+3 S2~S2+3のとき、ON S1~S1+3 = S2~S2+3のとき、OFF																									
フラグ	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354																												
[使用例]		<table border="1"> <tr> <td>004201</td> <td>F-180d CP></td> <td>009100</td> <td>009110</td> <td>000100</td> </tr> </table>		004201	F-180d CP>	009100	009110	000100	<table border="1"> <tr> <td colspan="2">命 令</td> </tr> <tr> <td>STR</td> <td>004201</td> </tr> <tr> <td>F-180d</td> <td>009100</td> </tr> <tr> <td></td> <td>009110</td> </tr> <tr> <td></td> <td>000100</td> </tr> </table>		命 令		STR	004201	F-180d	009100		009110		000100	入力条件004201がONのときに、レジスタ009100~009103の内容とレジスタ009110~009113の内容を比較して、(009100~009103)>(009110~009113)のときにリレー000100がONします。 入力条件004201がOFFのとき、および(009100~009103)(009110~009113)のときには、リレー000100はOFFになります。											
004201	F-180d CP>	009100	009110	000100																												
命 令																																
STR	004201																															
F-180d	009100																															
	009110																															
	000100																															

S1, S2には必ず偶数アドレスを設定してください。(019003等は禁止)

レジスタと定数(1 バイト)の比較 (リレー出力付き)

Fc180 CP >	(ComPare >)	Fc183 CP > =	(ComPare > =)
Fc181 CP <	(ComPare <)	Fc184 CP < =	(ComPare < =)
Fc182 CP =	(ComPare =)	Fc185 CP < >	(ComPare < >)

シンボル	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px;">Fc180 CP></td> <td style="border: 1px solid black; padding: 2px;">S</td> <td style="border: 1px solid black; padding: 2px;">n</td> <td style="border: 1px solid black; padding: 2px;">BIT</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">Fc181 CP<</td> <td style="border: 1px solid black; padding: 2px;">S</td> <td style="border: 1px solid black; padding: 2px;">n</td> <td style="border: 1px solid black; padding: 2px;">BIT</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">Fc182 CP=</td> <td style="border: 1px solid black; padding: 2px;">S</td> <td style="border: 1px solid black; padding: 2px;">n</td> <td style="border: 1px solid black; padding: 2px;">BIT</td> </tr> </table>		Fc180 CP>	S	n	BIT	Fc181 CP<	S	n	BIT	Fc182 CP=	S	n	BIT	<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px;">Fc183 CP>=</td> <td style="border: 1px solid black; padding: 2px;">S</td> <td style="border: 1px solid black; padding: 2px;">n</td> <td style="border: 1px solid black; padding: 2px;">BIT</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">Fc184 CP<=</td> <td style="border: 1px solid black; padding: 2px;">S</td> <td style="border: 1px solid black; padding: 2px;">n</td> <td style="border: 1px solid black; padding: 2px;">BIT</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">Fc185 CP<></td> <td style="border: 1px solid black; padding: 2px;">S</td> <td style="border: 1px solid black; padding: 2px;">n</td> <td style="border: 1px solid black; padding: 2px;">BIT</td> </tr> </table>		Fc183 CP>=	S	n	BIT	Fc184 CP<=	S	n	BIT	Fc185 CP<>	S	n	BIT
Fc180 CP>	S	n	BIT																									
Fc181 CP<	S	n	BIT																									
Fc182 CP=	S	n	BIT																									
Fc183 CP>=	S	n	BIT																									
Fc184 CP<=	S	n	BIT																									
Fc185 CP<>	S	n	BIT																									
機能	レジスタSの内容と定数n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	使用範囲 A		n	使用範囲 000 ~ 377 ⁽⁸⁾																								
BIT	使用範囲 K		演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)																								
演算後の内容	S	不変																										
	BIT	Fc180	S > nのとき、ON S nのとき、OFF	Fc183	S nのとき、ON S < nのとき、OFF																							
		Fc181	S < nのとき、ON S nのとき、OFF	Fc184	S nのとき、ON S > nのとき、OFF																							
		Fc182	S = nのとき、ON S nのとき、OFF	Fc185	S nのとき、ON S = nのとき、OFF																							
フラグ	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354																								
	0	0	0	0																								
<p>[使用例]</p> <table style="border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px;">005001</td> <td style="border: 1px solid black; padding: 2px;">Fc180 CP></td> <td style="border: 1px solid black; padding: 2px;">019000</td> <td style="border: 1px solid black; padding: 2px;">012</td> <td style="border: 1px solid black; padding: 2px;">000300</td> </tr> </table>		005001	Fc180 CP>	019000	012	000300	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <th colspan="2" style="text-align: center;">命 令</th> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">STR</td> <td style="border: 1px solid black; padding: 2px;">005001</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">Fc180</td> <td style="border: 1px solid black; padding: 2px;">019000 012 000300</td> </tr> </table>		命 令		STR	005001	Fc180	019000 012 000300	<p>入力条件005001がONのときに、レジスタ019000の内容と8進定数012を比較して、(019000) > 012⁽⁸⁾のときにリレー000300がONします。</p> <p>入力条件005001がOFFのとき、および(019000) 012⁽⁸⁾のときには、リレー000300はOFFします。</p>													
005001	Fc180 CP>	019000	012	000300																								
命 令																												
STR	005001																											
Fc180	019000 012 000300																											

レジスタと定数(1ワード)の比較(リレー出力付き)

Fc180w CP >	(ComPare >)	Fc183w CP > =	(ComPare > =)
Fc181w CP <	(ComPare <)	Fc184w CP < =	(ComPare < =)
Fc182w CP =	(ComPare =)	Fc185w CP < >	(ComPare < >)

シンボル	— <table border="1"><tr><td>Fc180w CP></td><td>S</td><td>n</td><td>BIT</td></tr></table>				Fc180w CP>	S	n	BIT	— <table border="1"><tr><td>Fc183w CP>=</td><td>S</td><td>n</td><td>BIT</td></tr></table>				Fc183w CP>=	S	n	BIT							
	Fc180w CP>	S	n	BIT																			
	Fc183w CP>=	S	n	BIT																			
— <table border="1"><tr><td>Fc181w CP<</td><td>S</td><td>n</td><td>BIT</td></tr></table>				Fc181w CP<	S	n	BIT	— <table border="1"><tr><td>Fc184w CP<=</td><td>S</td><td>n</td><td>BIT</td></tr></table>				Fc184w CP<=	S	n	BIT								
Fc181w CP<	S	n	BIT																				
Fc184w CP<=	S	n	BIT																				
— <table border="1"><tr><td>Fc182w CP=</td><td>S</td><td>n</td><td>BIT</td></tr></table>				Fc182w CP=	S	n	BIT	— <table border="1"><tr><td>Fc185w CP<></td><td>S</td><td>n</td><td>BIT</td></tr></table>				Fc185w 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 → BIT ON			Fc183w	S、S+1 n → BIT ON																	
	Fc181w	S、S+1 < n → BIT ON			Fc184w	S、S+1 n → BIT ON																	
	Fc182w	S、S+1 = n → BIT ON			Fc185w	S、S+1 n → BIT ON																	
S	使用範囲 B				n	使用範囲 000000 ~ 177777 ₍₈₎																	
BIT	使用範囲 K				演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)																	
演算後の内容	S、S+1	不変																					
	BIT	Fc180w	S、S+1 > nのとき、ON S、S+1 nのとき、OFF			Fc183w	S、S+1 nのとき、ON S、S+1 < nのとき、OFF																
		Fc181w	S、S+1 < nのとき、ON S、S+1 nのとき、OFF			Fc184w	S、S+1 nのとき、ON S、S+1 > nのとき、OFF																
		Fc182w	S、S+1 = nのとき、ON S、S+1 nのとき、OFF			Fc185w	S、S+1 nのとき、ON S、S+1 = nのとき、OFF																
フラグ	ゼロ 007357	キャリー 007356	エラー 007355	ハジキャリ 007354	—————																		
[使用例]		<table border="1"> <tr> <td>005201</td> <td>Fc180w CP></td> <td>019100</td> <td>012345</td> <td>000150</td> </tr> </table>			005201	Fc180w CP>	019100	012345	000150	<table border="1"> <tr> <th colspan="2">命 令</th> </tr> <tr> <td>STR</td> <td>005201</td> </tr> <tr> <td>Fc180w</td> <td>019100 012345 000150</td> </tr> </table>				命 令		STR	005201	Fc180w	019100 012345 000150	入力条件005201がONのときに、レジスタ019100、019101の内容と8進定数012345を比較して、(019100、019101) > 012345のときにリレー000150がONします。 入力条件005201がOFFのとき、および(019100、019101) 012345のときには、リレー000150はOFFになります。			
005201	Fc180w CP>	019100	012345	000150																			
命 令																							
STR	005201																						
Fc180w	019100 012345 000150																						

Sには必ず偶数アドレスを設定してください。(019003等は禁止)

レジスタと定数(2ワード)の比較(リレー出力付き)

Fc180d CP >	(ComPare >)	Fc183d CP > =	(ComPare > =)
Fc181d CP <	(ComPare <)	Fc184d CP < =	(ComPare < =)
Fc182d CP =	(ComPare =)	Fc185d CP < >	(ComPare < >)

シンボル	— <table border="1"><tr><td>Fc180d CP></td><td>S</td><td>n</td><td>BIT</td></tr></table>		Fc180d CP>	S	n	BIT	— <table border="1"><tr><td>Fc183d CP>=</td><td>S</td><td>n</td><td>BIT</td></tr></table>		Fc183d CP>=	S	n	BIT		
	Fc180d CP>	S	n	BIT										
	Fc183d CP>=	S	n	BIT										
— <table border="1"><tr><td>Fc181d CP<</td><td>S</td><td>n</td><td>BIT</td></tr></table>		Fc181d CP<	S	n	BIT	— <table border="1"><tr><td>Fc184d CP<=</td><td>S</td><td>n</td><td>BIT</td></tr></table>		Fc184d CP<=	S	n	BIT			
Fc181d CP<	S	n	BIT											
Fc184d CP<=	S	n	BIT											
— <table border="1"><tr><td>Fc182d CP=</td><td>S</td><td>n</td><td>BIT</td></tr></table>		Fc182d CP=	S	n	BIT	— <table border="1"><tr><td>Fc185d CP<></td><td>S</td><td>n</td><td>BIT</td></tr></table>		Fc185d CP<>	S	n	BIT			
Fc182d CP=	S	n	BIT											
Fc185d CP<>	S	n	BIT											
機能	レジスタS~S+3の内容(1ワードデータ)と、定数nを大小比較し、比較結果が成立時にBITをONする。													
演算内容	Fc180d	S ~ S+3 > n → BIT ON	Fc183d	S ~ S+3 n → BIT ON										
	Fc181d	S ~ S+3 < n → BIT ON	Fc184d	S ~ S+3 n → BIT ON										
	Fc182d	S ~ S+3 = n → BIT ON	Fc185d	S ~ S+3 n → BIT ON										
S	使用範囲 C		n	使用範囲 00000000000 ~ 37777777777 ₍₈₎										
BIT	使用範囲 K		演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)										
演算後の内容	S, S+1	不変												
	BIT	Fc180d	S ~ S+3 > nのとき、ON S ~ S+3 nのとき、OFF	Fc183d	S ~ S+3 nのとき、ON S ~ S+3 < nのとき、OFF									
		Fc181d	S ~ S+3 < nのとき、ON S ~ S+3 nのとき、OFF	Fc184d	S ~ S+3 nのとき、ON S ~ S+3 > nのとき、OFF									
		Fc182d	S ~ S+3 = nのとき、ON S ~ S+3 nのとき、OFF	Fc185d	S ~ S+3 nのとき、ON S ~ S+3 = nのとき、OFF									
フラグ	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354										
	0	0	0	0										
[使用例]		<table border="1"> <tr> <td>005201</td> <td>Fc180d CP></td> <td>019100</td> <td>01234567777</td> <td>000150</td> </tr> </table> <table border="1"> <tr> <td>命 令</td> <td>STR 005201</td> <td>FC180d 019100</td> <td>FC180d 01234567777</td> <td>FC180d 000150</td> </tr> </table> <p>入力条件005201がONのときに、レジスタ019100~019103の内容と8進定数01234567777を比較して、(019100 ~ 019103) > 01234567777のときにリレー000150がONします。 入力条件005201がOFFのとき、および(019100 ~ 019103) 01234567777のときには、リレー000150はOFFになります。</p>			005201	Fc180d CP>	019100	01234567777	000150	命 令	STR 005201	FC180d 019100	FC180d 01234567777	FC180d 000150
005201	Fc180d CP>	019100	01234567777	000150										
命 令	STR 005201	FC180d 019100	FC180d 01234567777	FC180d 000150										

Sには必ず偶数アドレスを設定してください。(019003等は禁止)

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	—	F-202 OPCH	UN CH·ST	FILE F	n	—	F-203 OPCH	UN CH·ST	FILE F	n
機能	サテライトネット(JW-22CM、JW-20CM)を使用したPLC間データ交換用の相手局指定命令です。F-204(SEND)またはF-205(RCV)命令と組み合わせて使用します。									
UN	使用範囲 0 ~ 7 : JW-22CMのユニット番号									
CH	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	使用範囲 000 ~ 377 ⁽⁸⁾ 、00 ~ FF ^(H) : 通信相手局の局番 F-202は8進数、F-203は16進数を使用します。									
F	通信相手局のファイル番号 0 (ファイルレジスタを除くデータメモリのとき) 1 (ファイルレジスタのとき)									
n	通信相手局のファイルアドレス(データ先頭アドレス) 使用範囲 00000000 ~ 37777777 ⁽⁸⁾ (ファイルレジスタのとき、バイトアドレス)									
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)									
フラグ	不変									

・ Fとnの関係は、次のとおりです。

データメモリ	F	n の使用範囲
ファイルレジスタ以外	0	00000000 ~ 00177777 ⁽⁸⁾ (ファイルアドレス 2・2、3ページ)
ファイルレジスタ	1	00000000 ~ 37777777 ⁽⁸⁾ (バイトアドレス 2・4ページ)

コントロールユニットの機種(JW-3**CU)により、使用範囲が異なります。

・ 通信相手局がJW30H、JW50H/70H/100Hの場合、相手(JW30H等)の「file N、n」を、JW300の「FILE F、n」に変換してください。「JW-22CMユーザズマニュアル」参照

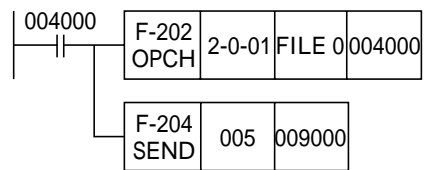
CH0、CH1、CH2、CH3と分けると、PLCプログラム中で4回同じユニット番号のJW-22CMを使用できます。本命令は必ず、F-204(SEND)命令またはF-205(RCV)命令と併用してください。

F-204 SEND

送信命令

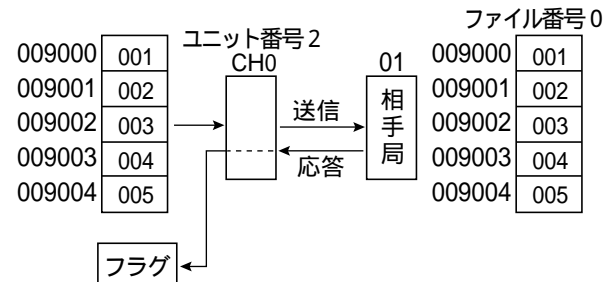
シンボル	— F-204 SEND n S					
機能	サテライトネット間データ通信の送信 (先頭アドレスと送信バイト数を指定)					
演算内容	S ~ S+n-1 → 指定局					
n	使用範囲 000 ~ 377 ⁽⁸⁾ (000 ⁽⁸⁾ のとき256バイト)					
S	使用範囲 A					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	S	不変				
	フラグ	通信内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		ポートからの 応答なし	0	0	1	0
		通信渋滞	0	0	0	1
		通信中 (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
相手局が 書込禁止	1	1	1	0		

[使用例]



命 令	
STR	004000
F-202	2-0-01 FILE 0 004000
F-204	005 009000

入力条件004000がOFF ONの変化時に、レジスタ009000~009004の内容(5バイトデータ)を、ユニット番号2のCH0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0000400(レジスタ009000)に送信します。



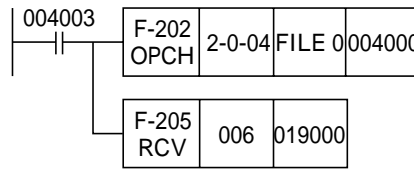
- ・ 本命令の使用方法は、ネットワークユニットJW-22CMのユーザーズマニュアルを参照願います。
- ・ 1階層通信の場合、必ずF-202またはF-203を併用してください。2階層通信の場合、必ずF-206とF-207を併用してください。

**F-205
RCV**

受信命令

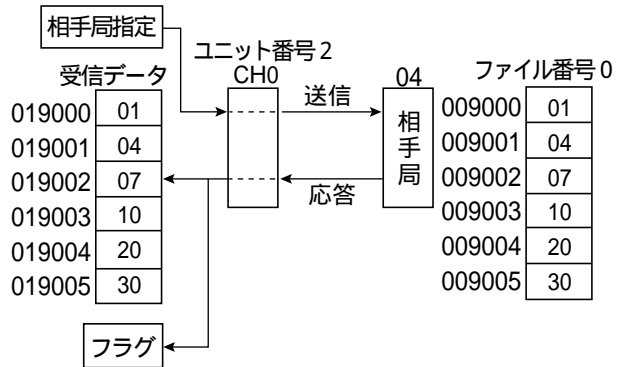
シンボル	— F-205 RCV n D				
機能	サテライトネット間データ通信の受信 (先頭アドレスと受信バイト数を指定)				
演算内容	指定局 → D ~ D+n-1				
n	使用範囲 000 ~ 377 ₍₈₎ (000 ₍₈₎ のとき256バイト)				
D	使用範囲 A				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	D	演算結果			
	通信内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	ポートからの 応答なし	0	0	1	0
	通信渋滞	0	0	0	1
	通信中 (相手からの 応答待ち)	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	

[使用例]



命 令	
STR	004003
F-202	2-0-04
FILE 0	004000
F-205	006
	019000

入力条件004003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004000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019000 ~ 019005に格納します。



- ・ 本命令の使用方法は、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願います。
- ・ 1階層通信の場合、必ずF-202またはF-203を併用してください。2階層通信の場合、必ずF-206とF-207を併用してください。

F-206
EOP1

オープンチャンネル 1(階層通信設定)

F-207
EOP2

オープンチャンネル 2(階層通信設定)

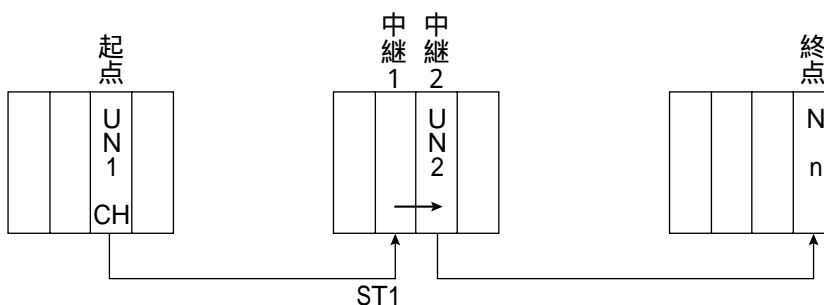
シンボル	—	F-206 EOP1	UN1,CH	ST1	UN2	—	F-207 EOP2	ST2	FILE F	n
機能	サテライトネット(JW-22CM、JW-20CM)を使用したPLC間データ交換用の相手局指定命令です。F-206(EOP1)はF-207(EOP2)、F-204(SEND)またはF-205(RCV)命令と組み合わせて使用します。									
UN1	使用範囲 0 ~ 7 : SEND、RECEIVE命令の起点となるユニットのユニット番号									
UN2	使用範囲 0 ~ 7 : SEND、RECEIVEの中継局 2 となるユニットのユニット番号 (注)中継局 2 がJW50H/70H/100Hのとき、スロット番号となる。									
CH	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	使用範囲 00 ~ 77 ⁽⁸⁾ : SEND、RECEIVE命令の中継局 1 の局番									
ST2	使用範囲 00 ~ 77 ⁽⁸⁾ : SEND、RECEIVE命令の終点局の局番									
F	SEND、RECEIVE命令の終点局でのファイル番号 0 (ファイルレジスタを除くデータメモリのとき) 1 (ファイルレジスタのとき)									
n	SEND、RECEIVE命令の終点局でのファイルアドレス 使用範囲 00000000 ~ 37777777 ⁽⁸⁾ (ファイルレジスタのとき、バイトアドレス)									
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)									
フラグ	不変									

・ Fとnの関係は、次のとおりです。

データメモリ	F	nの使用範囲
ファイルレジスタ以外	0	00000000 ~ 00177777 ⁽⁸⁾ (ファイルアドレス 2・2、3ページ)
ファイルレジスタ	1	00000000 ~ 37777777 ⁽⁸⁾ (バイトアドレス 2・4ページ)

コントロールユニットの機種(JW-3**CU)により、使用範囲が異なります。

・ 通信相手局がJW30H、JW50H/70H/100Hの場合、相手(JW30H等)の「file N、n」を、JW300の「FILE F、n」に変換してください。「JW-22CMユーザーズマニュアル」参照



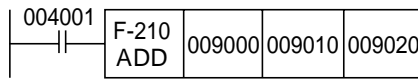
F-206、F-207、F-204の3命令、またはF-206、F-207、F-205の3命令は必ず併用してください。

**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 S1 S2 D				
機能	レジスタS1の内容とレジスタS2の内容をバイナリ加算して、結果をレジスタDに格納する。				
演算内容	S1+S2 → D				
S1	使用範囲 A				
S2	使用範囲 A				
D	使用範囲 A				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容 フラグ	S1	不変			
	S2	不変			
	D	演算結果			
	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	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	

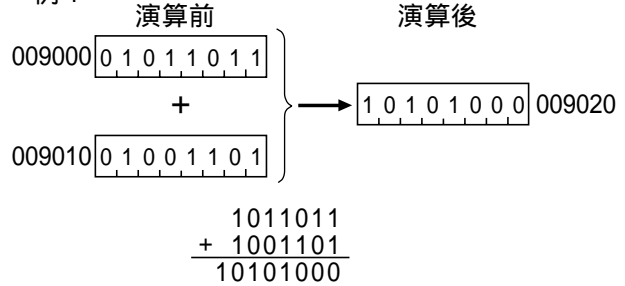
[使用例]



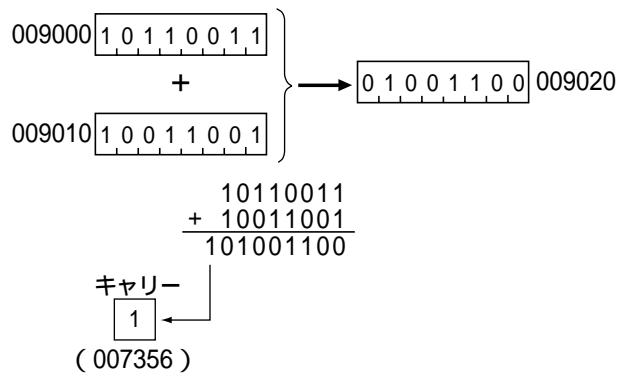
命 令	
STR	004001
F-210	009000
	009010
	009020

入力条件004001がOFF ONの変化時に、レジスタ009000の内容とレジスタ009010の内容をバイナリ加算して、結果をレジスタ009020に格納します。

例 1

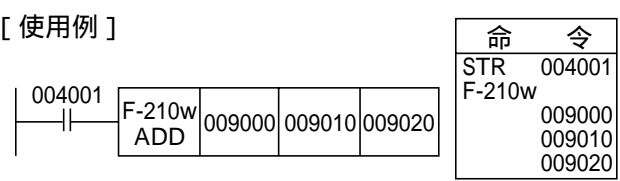


例 2

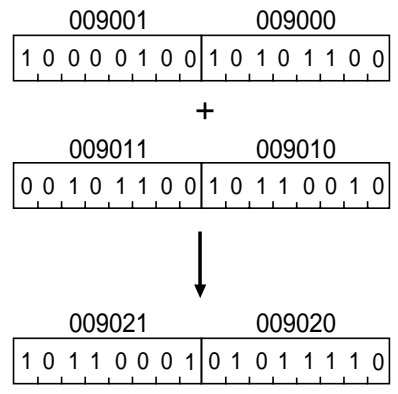


F-210w ADD レジスタ間のバイナリ加算(16ビット + 16ビット)
(ADD)

シンボル	<table border="1"> <tr> <td>F-210w ADD</td> <td>S1</td> <td>S2</td> <td>D</td> </tr> </table>				F-210w ADD	S1	S2	D	[使用例]		
F-210w ADD	S1	S2	D								
機能	レジスタS1、S1+1の内容と、レジスタS2、S2+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	(S1、S1+1) + (S2、S2+1) → D、D+1				<table border="1"> <tr> <th colspan="2">命 令</th> </tr> <tr> <td>STR</td> <td>004001</td> </tr> <tr> <td>F-210w</td> <td>009000 009010 009020</td> </tr> </table>	命 令		STR	004001	F-210w	009000 009010 009020
命 令											
STR	004001										
F-210w	009000 009010 009020										
S1	使用範囲 B										
S2	使用範囲 B										
D	使用範囲 B										
演算条件	入力信号の立上り(OFF ON)										
演算後の内容	S1、S1+1	不変									
	S2、S2+1	不変									
	D	演算結果(下位)									
	D+1	演算結果(上位)									
	フラグ	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノキャリー 007354					
	0	1	0	0	1						
	000001 ~ 177777	0	0	0	1						
	200000	1	1	0	0						
	200001 以上	0	1	0	0						



入力条件004001がOFF ONの変化時に、レジスタ009000、009001の内容とレジスタ009010、009011の内容をバイナリ加算して、レジスタ009020、009021に格納します。



S1、S2、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

F-210d レジスタ間のバイナリ加算(32ビット + 32ビット)
ADD (ADD)

シンボル	— F-210d ADD S1 S2 D					
機能	レジスタS1～S1+3の内容とレジスタS2～S2+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	(S1～S1+3) + (S2～S2+3) → D～D+3					
S1	使用範囲 C					
S2	使用範囲 C					
D	使用範囲 C					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	S1～S1+3	不変				
	S2～S2+3	不変				
	D～D+3	演算結果(バイナリ32ビット)				
		演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	フラグ	0	1	0	0	1
	0～3777777777	0	0	0	1	
	4000000000	1	1	0	0	
	4000000001以上	0	1	0	0	

[使用例]

命 令	
STR	004001
F-210d	009000 009010 009020

入力条件004001がOFF ONの変化時に、レジスタ009000～009003の内容とレジスタ009010～009013の内容をバイナリ加算して、レジスタ009020～009023に格納します。

```

009003  009002  009001  009000
10,00,01,00 | 10,10,11,00 | 10,00,01,00 | 10,10,11,00
+
009013  009012  009011  009010
00,10,11,00 | 10,11,00,10 | 00,10,11,00 | 10,11,00,10
↓
009023  009022  009021  009020
10,11,00,01 | 01,01,11,10 | 10,11,00,01 | 01,01,11,10

```

S1、S2、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

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" style="display: inline-table; vertical-align: middle;"><tr><td>Fc210 ADD</td><td>S1</td><td>n</td><td>D</td></tr></table>					Fc210 ADD	S1	n	D
Fc210 ADD	S1	n	D						
機能	レジスタS1の内容と8進定数 nをバイナリ加算して、レジスタDに格納する。								
演算内容	S1 + n → D								
S1	使用範囲 A								
n	使用範囲 000 ~ 377(8)								
D	使用範囲 A								
演算条件	入力信号の立上り(OFF ON)								
演算後の内容	S1	不変							
	D	演算結果							
	フラグ	演算結果(8進)	ゼロ	キャリー	エラー	ノンキャリー			
			007357	007356	007355	007354			
		0	1	0	0	1			
001 ~ 377		0	0	0	1				
400	1	1	0	0					
401以上	0	1	0	0					

[使用例]

004000	Fc210 ADD	009000	377	009020
--------	-----------	--------	-----	--------

命 令	
STR	004000
Fc210	009000
	377
	009020

入力条件004000がOFF ONの変化時に、レジスタ009000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009020に格納します。

演算前

009000	1 0 1 0 1 0 1 0
+	
8進定数 377	1 1 1 1 1 1 1 1

演算後

009020	1 0 1 0 1 0 0 1
--------	-----------------

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>Fc210w ADD</td><td>S1</td><td>n</td><td>D</td></tr></table>					Fc210w ADD	S1	n	D
Fc210w ADD	S1	n	D						
機能	レジスタS1、S1+1の内容と、8進定数 nを加算して、レジスタD、D+1に格納する。								
演算内容	(S1, S1+1) + n → D, D+1								
S1	使用範囲 B								
n	使用範囲 000000 ~ 177777(8)								
D	使用範囲 B								
演算条件	入力信号の立上り(OFF ON)								
演算後の内容	S1, S1+1	不変							
	D	演算結果(下位)							
	D+1	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			
			007357	007356	007355	007354			
0		1	0	0	1				
000001 ~ 177777		0	0	0	1				
200000	1	1	0	0					
200001以上	0	1	0	0					

[使用例]

004000	Fc210w ADD	009000	123321	009020
--------	------------	--------	--------	--------

命 令	
STR	004000
Fc210w	009000
	123321
	009020

入力条件004000がOFF ONの変化時に、レジスタ009000、009001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009020、009021に格納します。

009001 009000

1 1 1 0 1 0 0 1	1 1 0 0 0 1 1 0 1
+	
8進定数123321	
1 0 1 0 0 1 1 0	1 1 0 1 0 0 0 1
1 2 3 3 2 1	

↓

009021 009020

1 0 0 1 0 0 0 0	0 1 0 1 1 1 1 0
-----------------	-----------------

S1、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

**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	Fc210d ADD S1 n D			
機能	レジスタS1～S1+3の内容と、8進定数nを加算して、レジスタD～D+3に格納する。			
演算内容	$(S1 \sim S1+3) + n \rightarrow D \sim D+3$			
S1	使用範囲 C			
n	使用範囲 000000 ~ 177777 ⁽⁸⁾			
D	使用範囲 C			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S1～S1+3	不変		
	D～D+3	演算結果(バイナリ32ビット)		
	演算結果(8進)	ゼロ	キャリー	エラー
	0	1	0	0
	1～3777777777	0	0	0
フラグ	40000000000	1	1	0
	40000000001以上	0	1	0

[使用例]

004000	Fc210d ADD	009000	123321	009020
--------	---------------	--------	--------	--------

命 令	
STR	004000
Fc210d	009000
	123321
	009020

入力条件004000がOFF ONの変化時に、レジスタ009000～009003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009020～009023に格納します。

009003	009002	009001	009000
11101001	10001101	01101001	10001101
+			
8進定数123321			
1010011011010001			
1 2 3 3 2 1			
↓			
009023	009022	009021	009020
11101001	10001110	00010000	01011110

S1, Dには必ず偶数アドレスを設定してください。(019003等は禁止)

**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	— F-211 SUB S1 S2 D				
機能	レジスタS1の内容からレジスタS2の内容をバイナリ減算して、結果をレジスタDに格納する。				
演算内容	S1 - S2 → D				
S1	使用範囲 A				
S2	使用範囲 A				
D	使用範囲 A				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S1	不変			
	S2	不変			
	D	演算結果			
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355
	0	1	0	0	1
	001 ~ 377(8)	0	0	0	1
	負の数値	0	1	0	0

[使用例]

命 令	
STR	001000
F-211	019000
	019001
	019002

001000
└──┬──┘
F-211 SUB 019000 019001 019002

入力条件001000がOFF ONの変化時に、レジスタ019000の内容からレジスタ019001の内容をバイナリ減算して、結果をレジスタ019002に格納します。

例 1

演算前

019000 1 0 1 1 0 1 0 1
-
019001 0 1 0 1 0 1 1 1

演算後

0 1 0 1 1 1 1 0 019002

10110101
- 1010111
1011110

例 2

019000 0 0 1 0 1 1 0 1
-
019001 1 0 1 1 0 1 0 0

0 1 1 1 1 0 0 1 019002

101101
- 10110100
101111001

キャリー
1
(007356)

演算結果が負の数値{(S1の内容) < (S2の内容)}のとき、2の補数で表現されます。
2の補数とは、2進数で表したデータのすべてのビットを反転(0 1, 1 0)させ、1を加算した値です。
・上記「 例 2 」の場合、45(00101101) - 18(10110100) = 135により、135(10000111)に2の補数をとると、121(01111001)となります。

135 10000111
ビット反転
01111000
1を加算
01111001 121

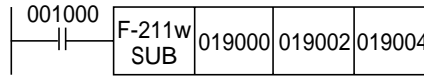
・ 2の補数を求める命令には F-57、F-57w、F-57d があります。

**F-211w
SUB**

レジスタ間のバイナリ減算(16ビット - 16ビット)
(SUBtract)

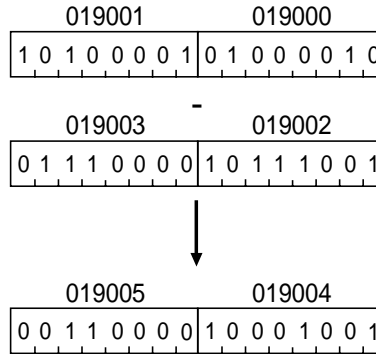
シンボル	F-211w SUB S1 S2 D				
機能	レジスタS1、S1+1の内容からレジスタS2、S2+1の内容をバイナリ減算して、レジスタD、D+1に格納する。				
演算内容	$(S1, S1+1) - (S2, S2+1) \rightarrow D, D+1$				
S1	使用範囲 B	1			
S2	使用範囲 B	1			
D	使用範囲 B	1			
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S1、S1+1	不変			
	S2、S2+1	不変			
	D	演算結果(下位) } 2			
	D+1	演算結果(上位) }			
フラグ	演算結果(8進)	ゼロ	キャリー	エラー	ノンキャリー
		007357	007356	007355	007354
	0	1	0	0	1
	1~177777	0	0	0	1
	負の数値	0	1	0	0

[使用例]



命令	
STR	001000
F-211w	019000 019002 019004

入力条件001000がOFF ONの変化時に、レジスタ019000、019001の内容からレジスタ019002、019003の内容をバイナリ減算して、結果をレジスタ019004、019005に格納します。



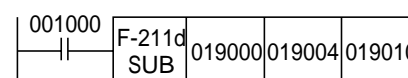
- 1 S1、S2、Dには必ず偶数アドレスを設定してください。(019003等は禁止)
- 2 演算結果が負の数値{(S1、S1+1の内容) < (S2、S2+1の内容)}のとき、2の補数で表現されます。 14・21ページ参照

**F-211d
SUB**

レジスタ間のバイナリ減算(32ビット - 32ビット)
(SUBtract)

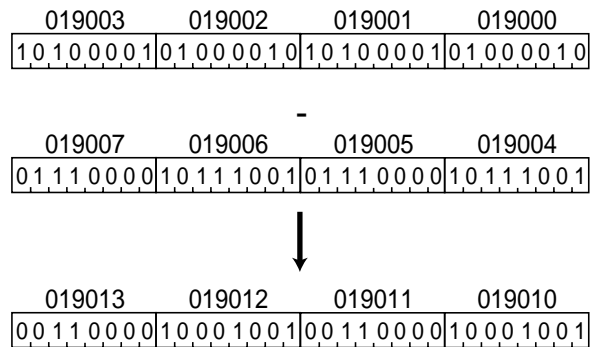
シンボル	F-211d SUB S1 S2 D				
機能	レジスタS1~S1+3の内容からレジスタS2~S2+3の内容をバイナリ減算して、レジスタD~D+3に格納する。				
演算内容	$(S1 \sim S1+3) - (S2 \sim S2+3) \rightarrow D \sim D+3$				
S1	使用範囲 C	1			
S2	使用範囲 C	1			
D	使用範囲 C	1			
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S1~S1+3	不変			
	S2~S2+3	不変			
	D~D+3	演算結果(バイナリ32ビット) } 2			
フラグ	演算結果(8進)	ゼロ	キャリー	エラー	ノンキャリー
		007357	007356	007355	007354
	0	1	0	0	1
	1~3777777777	0	0	0	1
	負の数値	0	1	0	0

[使用例]



命令	
STR	001000
F-211d	019000 019004 019010

入力条件001000がOFF ONの変化時に、レジスタ019000~019003の内容からレジスタ019004~019007の内容をバイナリ減算して、結果をレジスタ019010~019013に格納します。



- 1 S1、S2、Dには必ず偶数アドレスを設定してください。(019003等は禁止)
- 2 演算結果が負の数値{(S1~S1+3の内容) < (S2~S2+3の内容)}のとき、2の補数で表現されます。 14・21ページ参照

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="text-align: center;">Fc211 SUB</td> <td style="text-align: center;">S1</td> <td style="text-align: center;">n</td> <td style="text-align: center;">D</td> </tr> </table>	Fc211 SUB	S1	n	D		<p>[使用例]</p> <table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="text-align: center;">004000</td> <td style="text-align: center;">Fc211 SUB</td> <td style="text-align: center;">019000</td> <td style="text-align: center;">123</td> <td style="text-align: center;">009000</td> </tr> </table> <table border="1" style="display: inline-table; border-collapse: collapse; margin-left: 20px;"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004000</td></tr> <tr><td>Fc211</td><td>019000</td></tr> <tr><td></td><td>123</td></tr> <tr><td></td><td>009000</td></tr> </table> <p>入力条件004000がOFF ONの変化時に、レジスタ019000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009000に格納します。</p> <div style="text-align: center;"> <p>演算前</p> <table style="margin: 0 auto;"> <tr> <td style="border: 1px solid black; padding: 2px;">019000</td> <td style="border: 1px solid black; padding: 2px;">1 0 1 0 0 0 1 0</td> </tr> <tr><td colspan="2" style="text-align: center;">-</td></tr> <tr> <td style="border: 1px solid black; padding: 2px;">8進定数 123</td> <td style="border: 1px solid black; padding: 2px;">0 1 0 1 0 0 1 1</td> </tr> <tr> <td></td> <td style="text-align: center;">1 2 3</td> </tr> </table> <p>→</p> <table style="margin: 0 auto;"> <tr> <td style="border: 1px solid black; padding: 2px;">009000</td> <td style="border: 1px solid black; padding: 2px;">0 1 0 0 1 1 1 1</td> </tr> </table> <p>演算後</p> </div>	004000	Fc211 SUB	019000	123	009000	命 令		STR	004000	Fc211	019000		123		009000	019000	1 0 1 0 0 0 1 0	-		8進定数 123	0 1 0 1 0 0 1 1		1 2 3	009000	0 1 0 0 1 1 1 1
Fc211 SUB	S1	n	D																													
004000	Fc211 SUB	019000	123	009000																												
命 令																																
STR	004000																															
Fc211	019000																															
	123																															
	009000																															
019000	1 0 1 0 0 0 1 0																															
-																																
8進定数 123	0 1 0 1 0 0 1 1																															
	1 2 3																															
009000	0 1 0 0 1 1 1 1																															
機能	レジスタS1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	S1 - n → D																															
S1	使用範囲 A																															
n	使用範囲 000 ~ 377 ⁽⁸⁾																															
D	使用範囲 A																															
演算条件	入力信号の立上り(OFF ON)																															
演算後の内容	S1	不変																														
	D	演算結果																														
	フ ラ グ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354																										
		0	1	0	0	1																										
1 ~ 377 ⁽⁸⁾		0	0	0	1																											
	負の数値	0	1	0	0																											

演算結果が負の数値{(S1の内容) < n }のとき、2の補数で表現されます。 14・21ページ参照

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	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="text-align: center;">Fc211w SUB</td> <td style="text-align: center;">S1</td> <td style="text-align: center;">n</td> <td style="text-align: center;">D</td> </tr> </table>	Fc211w SUB	S1	n	D		<p>[使用例]</p> <table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="text-align: center;">004000</td> <td style="text-align: center;">Fc211w SUB</td> <td style="text-align: center;">019000</td> <td style="text-align: center;">123456</td> <td style="text-align: center;">009000</td> </tr> </table> <table border="1" style="display: inline-table; border-collapse: collapse; margin-left: 20px;"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004000</td></tr> <tr><td>Fc211w</td><td>019000</td></tr> <tr><td></td><td>123456</td></tr> <tr><td></td><td>009000</td></tr> </table> <p>入力条件004000がOFF ONの変化時に、レジスタ019000、019001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009000、009001に格納します。</p> <div style="text-align: center;"> <p>019001 019000</p> <table style="margin: 0 auto;"> <tr> <td style="border: 1px solid black; padding: 2px;">0 0 1 0 1 1 1 0 0</td> <td style="border: 1px solid black; padding: 2px;">1 1 0 1 1 1 1 0 0</td> </tr> <tr><td colspan="2" style="text-align: center;">-</td></tr> <tr> <td style="border: 1px solid black; padding: 2px;">8進定数 123456</td> <td style="border: 1px solid black; padding: 2px;">1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 1 0</td> </tr> <tr> <td></td> <td style="text-align: center;">1 2 3 4 5 6</td> </tr> </table> <p>↓</p> <table style="margin: 0 auto;"> <tr> <td style="border: 1px solid black; padding: 2px;">009001</td> <td style="border: 1px solid black; padding: 2px;">009000</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">1 0 0 0 0 1 0 1</td> <td style="border: 1px solid black; padding: 2px;">1 0 1 0 1 1 1 1 0</td> </tr> </table> </div>	004000	Fc211w SUB	019000	123456	009000	命 令		STR	004000	Fc211w	019000		123456		009000	0 0 1 0 1 1 1 0 0	1 1 0 1 1 1 1 0 0	-		8進定数 123456	1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 1 0		1 2 3 4 5 6	009001	009000	1 0 0 0 0 1 0 1	1 0 1 0 1 1 1 1 0
Fc211w SUB	S1	n	D																															
004000	Fc211w SUB	019000	123456	009000																														
命 令																																		
STR	004000																																	
Fc211w	019000																																	
	123456																																	
	009000																																	
0 0 1 0 1 1 1 0 0	1 1 0 1 1 1 1 0 0																																	
-																																		
8進定数 123456	1 0 1 0 0 1 1 1 0 0 1 0 1 1 1 1 0																																	
	1 2 3 4 5 6																																	
009001	009000																																	
1 0 0 0 0 1 0 1	1 0 1 0 1 1 1 1 0																																	
機能	レジスタS1、S1+1の内容から8進定数nを減算して、レジスタD、D+1に格納する。																																	
演算内容	(S1、S1+1) - n → D、D+1																																	
S1	使用範囲 B 1																																	
n	使用範囲 000000 ~ 177777 ⁽⁸⁾																																	
D	使用範囲 B 1																																	
演算条件	入力信号の立上り(OFF ON)																																	
演算後の内容	S1、S1+1	不変																																
	D	演算結果(下位) } 2																																
	D+1	演算結果(上位) }																																
	フ ラ グ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354																												
0		1	0	0	1																													
1 ~ 177777 ⁽⁸⁾		0	0	0	1																													
	負の数値	0	1	0	0																													

1 S1、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

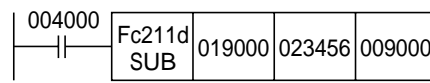
2 演算結果が負の数値{(S1、S1+1の内容) < n }のとき、2の補数で表現されます。 14・21ページ参照

**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)

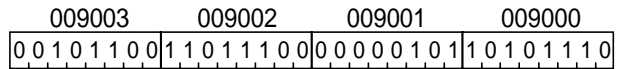
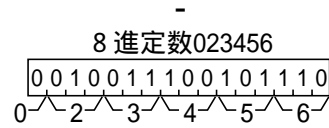
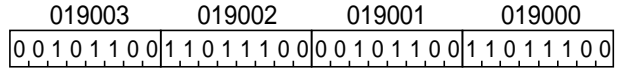
シンボル	Fc211d SUB S1 n D				
機能	レジスタS1 ~ S1+3の内容から8進定数nを減算して、レジスタD ~ D+3に格納する。				
演算内容	(S1 ~ S1+3) - n → D ~ D+3				
S1	使用範囲 C 1				
n	使用範囲 000000 ~ 177777(8)				
D	使用範囲 C 1				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S1 ~ S1+3	不変			
	D ~ D+3	演算結果(バイナリ32ビット) 2			
	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	0	1	0	0	1
1 ~ 3777777777(8)	0	0	0	1	
負の数値	0	1	0	0	

[使用例]



命 令	
STR	004000
Fc211d	019000 023456 009000

入力条件004000がOFF ONの変化時に、レジスタ019000 ~ 019003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009000 ~ 009003に格納します。



1 S1, Dには必ず偶数アドレスを設定してください。(019003等は禁止)

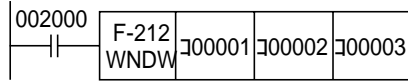
2 演算結果が負の数値{(S1 ~ S1+3の内容) < n}のとき、2の補数で表現されます。 14・21ページ参照

**F-212
WNDW**

ウィンドウコンパレータ(1バイトレジスタ間)

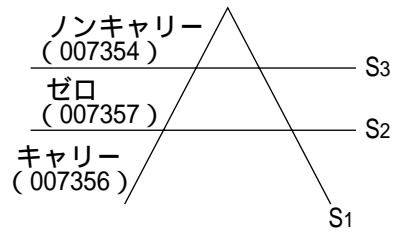
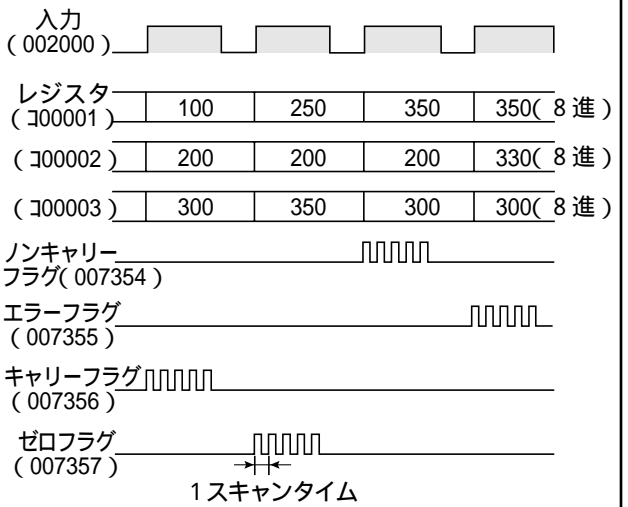
シンボル	F-212 WNDW S1 S2 S3				
機能	レジスタS1とレジスタS2、レジスタS3の内容を比較し、比較結果をフラグに格納する。				
演算内容	比較結果 → フラグ				
S1	使用範囲 A				
S2	使用範囲 A				
S3	使用範囲 A				
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)				
演算後の内容 フラグ	S1	不変			
	S2	不変			
	S3	不変			
	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	S1 < S2	0	1	0	0
S2 < S1	1	0	0	0	
S3 < S1	0	0	0	1	
S3 < S2	0	0	1	0	

[使用例]



命 令	
STR	002000
F-212	000001 000002 000003

入力条件002000がONのとき、レジスタ000001の内容が000001 < 000002、000002 > 000001、000003、000003 < 000001のどの範囲かを演算し、その結果をキャリーフラグ、ゼロフラグ、ノンキャリーフラグに設定します。
000002 > 000003の場合のみ演算し、000003 < 000002の場合は演算を中止し、エラーフラグをONします。



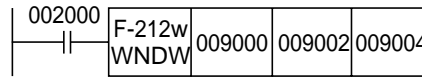
**F-212w
WNDW**

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	S1	S2	S3		
機能	レジスタS1、S1+1とレジスタS2、S2+1、レジスタS3、S3+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					
S1	使用範囲 B					
S2	使用範囲 B					
S3	使用範囲 B					
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)					
演算後の内容	S1、S1+1	不変				
	S2、S2+1	不変				
	S3、S3+1	不変				
	フラグ	レジスタの内容	ゼロ 007357	キャリア 007356	エラー 007355	ノンキャリア 007354
	S1、S1+1 < S2、S2+1	0	1	0	0	
S2、S2+1 < S1、S1+1 S3、S3+1	1	0	0	0		
S3、S3+1 < S1、S1+1	0	0	0	1		
S3、S3+1 < S2、S2+1	0	0	1	0		

S1、S2、S3には必ず偶数アドレスを設定してください。(009003等は禁止)

[使用例]



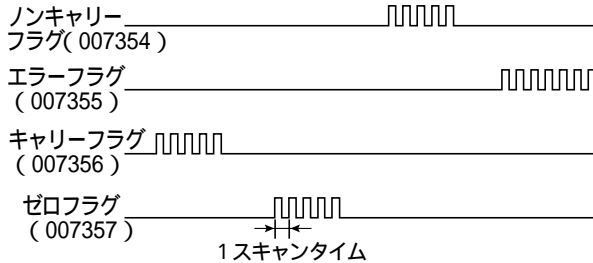
命 令	
STR	002000
F-212w	009000
	009002
	009004

入力条件002000がONのとき、レジスタ009000、009001(1ワード)の内容が(009000、009001) <(009002、009003) (009002、009003) (009000、009001) (009004、009005) <(009004、009005) <(009000、009001)のどの範囲かを演算し、その結果をキャリアフラグ、ゼロフラグ、ノンキャリアフラグに設定します。
(009002、009003) (009004、009005)の場合のみ演算し、(009004、009005) <(009002、009003)の場合は演算を中止してエラーフラグをONします。

入力 (002000)

レジスタ (009000) (009001) (009002) (009003) (009004) (009005)

000	100	100	100
100	200	300	200
000	000	000	100
200	200	200	200
000	000	000	000
300	300	300	200



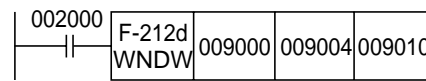
**F-212d
WNDW**

ウィンドウコンパレータ(2ワードレジスタ間)

シンボル	<table border="1"> <tr> <td>F-212d WNDW</td> <td>S1</td> <td>S2</td> <td>S3</td> </tr> </table>			F-212d WNDW	S1	S2	S3
F-212d WNDW	S1	S2	S3				
機能	レジスタS1～S1+3の内容と、レジスタS2～S2+3、S3～S3+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						
S1	使用範囲 C						
S2	使用範囲 C						
S3	使用範囲 C						
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)						
演算後の内容	S1～S1+3	不変					
	S2～S2+3	不変					
	S3～S3+3	不変					
	フラグ	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	
		S1～S1+3 < S2～S2+3	0	1	0	0	
	S2～S2+3 < S1～S1+3	1	0	0	0		
	S3～S3+3 < S1～S1+3	0	0	0	1		
	S3～S3+3 < S2～S2+3	0	0	1	0		

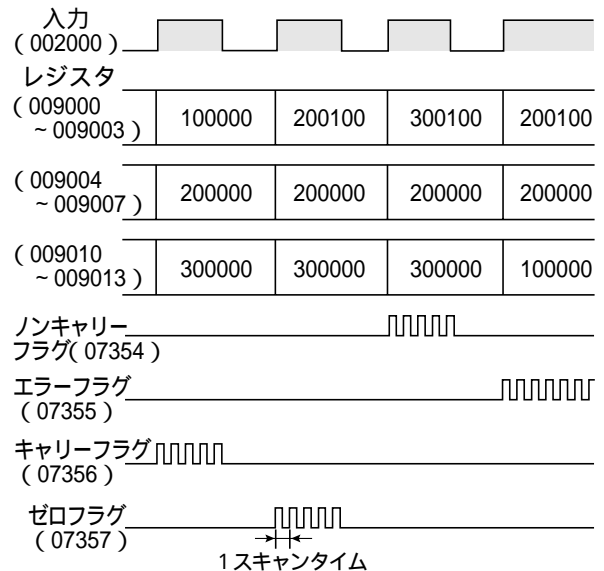
S1、S2、S3には必ず偶数アドレスを設定してください。(009003等は禁止)

[使用例]



命 令	
STR	002000
F-212d	009000
	009004
	009010

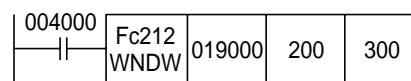
入力条件002000がONのとき、レジスタ009000～009003(2ワード)の内容が(009000～009003) < (009004～009007) (009004～009007) (009000～009003) (009010～009013) (009010～009013) < (009000～009003) のどの範囲かを演算し、その結果をキャリーフラグに設定します。
(009004～009007) (009010～009013) の場合のみ演算し、(009010～009013) < (009004～009007) の場合は演算を中止してエラーフラグをONします。



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	Fc212 WNDW S1 n1 n2					
機能	レジスタS1の内容と8進定数n1, n2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					
S1	使用範囲 A					
n1	使用範囲 000 ~ 377 ⁽⁸⁾					
n2	使用範囲 000 ~ 377 ⁽⁸⁾					
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)					
演算後の内容	S1	不変				
	フラグ	レジスタの内容	ゼロ (007357)	キャリー (007356)	エラー (007355)	ノンキャリー (007354)
		S1 < n1	0	1	0	0
		n1 S1 n2	1	0	0	0
		n2 < S1	0	0	0	1
n2 < n1	0	0	1	0		

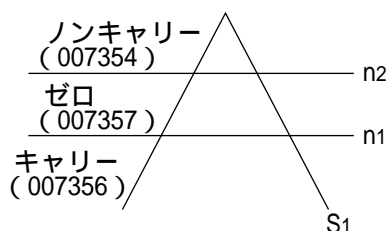
[使用例]



命 令	
STR	004000
Fc212	019000 200 300

入力条件004000がONのとき、レジスタ019000の内容が(019000) < 200⁽⁸⁾、200⁽⁸⁾ (019000) 300⁽⁸⁾、300⁽⁸⁾ < (019000) のどの範囲かを演算し、その結果をキャリーフラグ、ゼロフラグ、ノンキャリーフラグに設定します。

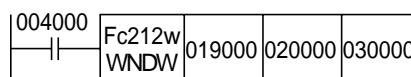
019000の内容 (8進)	ゼロ (007357)	キャリー (007356)	エラー (007355)	ノンキャリー (007354)
150	0	1	0	0
250	1	0	0	0
350	0	0	0	1



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	Fc212w WNDW S1 n1 n2					
機能	レジスタS1, S1+1の内容(1 ワードデータ)と 8 進定数n1, n2を比較して結果をフラグに設定する。					
演算内容	比較結果 → フラグ					
S1	使用範囲 B					
n1	使用範囲 000000 ~ 177777 ⁽⁸⁾					
n2	使用範囲 000000 ~ 177777 ⁽⁸⁾					
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)					
演算後の内容	S1, S1+1	不変				
	フラグ	レジスタの内容	ゼロ (007357)	キャリー (007356)	エラー (007355)	ノンキャリー (007354)
		S1, S1+1 < n1	0	1	0	0
		n1 S1, S1+1 n2	1	0	0	0
		n2 < S1, S1+1	0	0	0	1
		n2 < n1	0	0	1	0

[使用例]



命 令	
STR	004000
Fc212w	019000 020000 030000

入力条件004000がONのとき、レジスタ019000、019001(1 ワード)の内容が(019000, 019001) < 020000、020000 (019000, 019001) 030000、030000 < (019000, 019001) のどの範囲かを演算し、その結果をキャリーフラグ、ゼロフラグ、ノンキャリーフラグに設定します。

019000, 019001の内容	ゼロ (007357)	キャリー (007356)	エラー (007355)	ノンキャリー (007354)
015000	0	1	0	0
025000	1	0	0	0
035000	0	0	0	1

**Fc212d
WNDW**

ウィンドウコンパレータ(2ワード 8進定数間)

シンボル	— Fc212d WNDW				S1	n1	n2	[使用例]	<table border="1"> <tr> <td colspan="5">命 令</td> </tr> <tr> <td>STR</td> <td colspan="4">004000</td> </tr> <tr> <td>Fc212d</td> <td colspan="4">019000</td> </tr> <tr> <td></td> <td colspan="4">02000000000</td> </tr> <tr> <td></td> <td colspan="4">03000000000</td> </tr> </table>					命 令					STR	004000				Fc212d	019000					02000000000					03000000000			
命 令																																						
STR	004000																																					
Fc212d	019000																																					
	02000000000																																					
	03000000000																																					
機能	レジスタS1～S1+3の内容(2ワードデータ)と 8進定数n1、n2を比較して結果をフラグに設定する。							<table border="1"> <tr> <td>004000</td> <td> </td> <td>Fc212d WNDW</td> <td>019000</td> <td>02000000000</td> <td>03000000000</td> </tr> </table>	004000		Fc212d WNDW	019000	02000000000	03000000000																								
004000		Fc212d WNDW	019000	02000000000	03000000000																																	
演算内容	比較結果 → フラグ																																					
S1	使用範囲 C																																					
n1	使用範囲 00000000000 ~ 37777777777 ⁽⁸⁾																																					
n2	使用範囲 00000000000 ~ 37777777777 ⁽⁸⁾																																					
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)																																					
演算後の内容	S1～S1+3	不変																																				
	フ ラ グ	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354																																
		S1～S1+3 < n1	0	1	0	0																																
		n1 S1～S1+3 n2	1	0	0	0																																
		n2 < S1～S1+3	0	0	0	1																																
n2 < n1	0	0	1	0																																		

入力条件004000がONのとき、レジスタ019000～019003 (2ワード)の内容が(019000～019003)<02000000000、02000000000 (019000～019003) 03000000000、03000000000 <(019000～019003)のどの範囲かを演算し、その結果をキャリーフラグ、ゼロフラグ、ノンキャリーフラグに設定します。

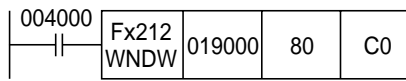
019000～019003 の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
01500000000	0	1	0	0
02500000000	1	0	0	0
03500000000	0	0	0	1

**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 S1 n1 n2					
機能	レジスタS1の内容と16進定数n1、n2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					
S1	使用範囲 A					
n1	使用範囲 00 ~ FF(H)					
n2	使用範囲 00 ~ FF(H)					
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)					
演算後の内容	S1	不変				
	フラグ	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		S1 < n1	0	1	0	0
		n1 S1 n2	1	0	0	0
		n2 < S1	0	0	0	1
n2 < n1	0	0	1	0		

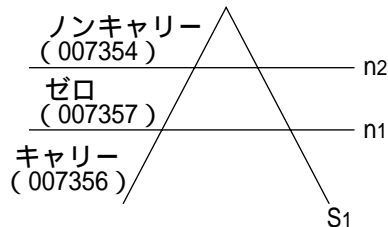
[使用例]



命 令	
STR	004000
Fx212	019000
	80
	C0

入力条件004000がONのとき、レジスタ019000の内容が(019000) < 80(H)、80(H) (019000) C0(H)、C0(H) < (019000) のどの範囲かを演算し、その結果をキャリーフラグ、ゼロフラグ、ノンキャリーフラグに設定します。

019000の内容 (16進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
70	0	1	0	0
90	1	0	0	0
D0	0	0	0	1

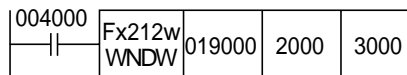


**Fx212w
WNDW**

ウィンドウコンパレータ(1ワード16進定数間)

シンボル	— Fx212w WNDW S1 n1 n2					
機能	レジスタS1、S1+1の内容(1ワードデータ)と16進定数n1、n2を比較して結果をフラグに設定する。					
演算内容	比較結果 → フラグ					
S1	使用範囲 B					
n1	使用範囲 0000 ~ FFFF(H)					
n2	使用範囲 0000 ~ FFFF(H)					
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)					
演算後の内容	S1、S1+1	不変				
	フラグ	レジスタの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		S1、S1+1 < n1	0	1	0	0
		n1 S1、S1+1 n2	1	0	0	0
		n2 < S1、S1+1	0	0	0	1
n2 < n1	0	0	1	0		

[使用例]



命 令	
STR	004000
Fx212w	019000
	2000
	3000

入力条件004000がONのとき、レジスタ019000、019001(1ワード)の内容が(019000、019001) < 2000、2000 (019000、019001) 3000、3000 < (019000、019001) のどの範囲かを演算し、その結果をキャリーフラグ、ゼロフラグ、ノンキャリーフラグに設定します。

019000、019001 の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
1500	0	1	0	0
2500	1	0	0	0
3500	0	0	0	1

**Fx212d
WNDW**

ウィンドウコンパレータ(2ワード16進定数間)

シンボル	— Fx212d WNDW S1 n1 n2				[使用例]																								
機能	レジスタS1～S1+3の内容(2ワードデータ)と16進定数n1、n2を比較して結果をフラグに設定する。																												
演算内容	比較結果 → フラグ				<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004000</td></tr> <tr><td>Fx212d</td><td>019000 20000000 30000000</td></tr> </table>					命 令		STR	004000	Fx212d	019000 20000000 30000000														
命 令																													
STR	004000																												
Fx212d	019000 20000000 30000000																												
S1	使用範囲 C				入力条件004000がONのとき、レジスタ019000～019003 (2ワード)の内容が(019000～019003)< 20000000、20000000 (019000～019003) 30000000、30000000 < (019000～019003)のどの範囲かを演算し、その結果をキャリーフラグ、ゼロフラグ、ノンキャリーフラグに設定します。																								
n1	使用範囲 00000000～FFFFFFFF(H)																												
n2	使用範囲 00000000～FFFFFFFF(H)																												
演算条件	入力信号がONのとき (OFF ONの変化時に限定されない)				<table border="1"> <thead> <tr> <th>019000、019001 の内容</th> <th>ゼ ロ 007357</th> <th>キャリー 007356</th> <th>エラー 007355</th> <th>ノンキャリー 007354</th> </tr> </thead> <tbody> <tr> <td>15000000</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>25000000</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>35000000</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> </tbody> </table>					019000、019001 の内容	ゼ ロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354	15000000	0	1	0	0	25000000	1	0	0	0	35000000	0	0	0	1
019000、019001 の内容	ゼ ロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354																									
15000000	0	1	0	0																									
25000000	1	0	0	0																									
35000000	0	0	0	1																									
演算後の内容	S1～S1+3	不変																											
	フ ラ グ	レジスタの内容	ゼ ロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354																							
		S1～S1+3 < n1	0	1	0	0																							
		n1 S1～S1+3 n2	1	0	0	0																							
		n2 < S1～S1+3	0	0	0	1																							
n2 < n1	0	0	1	0																									

F-215 MUL レジスタ間のバイナリ乗算(8ビット×8ビット)
(MULtipl y)

シンボル				
機能	レジスタS1の内容とレジスタS2の内容をバイナリ乗算し、結果をレジスタD、D+1に格納する。			
演算内容	S1 × S2 → D、D+1			
S1	使用範囲 A			
S2	使用範囲 A			
D	使用範囲 B			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S1	不変		
	S2	不変		
	D	演算結果(下位)		
	D+1	演算結果(上位)		
	フラグ	ゼロ 007357	キャリー 007356	エラー 007355
	0	0	0	0

[使用例]

入力条件004000がOFF ONの変化時に、レジスタ009000の内容とレジスタ009100の内容をバイナリ乗算して、結果をレジスタ009200、009201に格納します。

```

      009000
    00101100
      ×
      009001
    00011011
    -----
009201 009200
0000010010100100 ← 演算結果
  
```

```

      101100
      × 11011
      -----
      101100
      101100
      101100
      101100
      -----
     10010100100
  
```

F-215w MUL レジスタ間のバイナリ乗算(16ビット×16ビット)
(MULtipl y)

シンボル				
機能	レジスタS1、S1+1の内容とレジスタS2、S2+1の内容をバイナリ乗算し、結果をレジスタD ~ D+3に格納する。			
演算内容	(S1、S1+1) × (S2、S2+1) → D ~ D+3			
S1	使用範囲 B			
S2	使用範囲 B			
D	使用範囲 C			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S1、S1+1	不変		
	S2、S2+1	不変		
	D	演算結果(下位)		
	D+1	演算結果		
	D+2	演算結果		
	D+3	演算結果(上位)		
フラグ	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	0	0	0	0

[使用例]

入力条件004000がOFF ONの変化時に、レジスタ009000、009001の内容(16ビットデータ)とレジスタ009100、009101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009200 ~ 009203に格納します。

```

      009001 009000
    00010111 00110100
      ×
      009101 009100
    00100010 10000101
      ↓
009203 009202 009201 009200
00000011 00100000 11110110 00000100
  
```

S1、S2、Dには必ず偶数アドレスを設定してください。

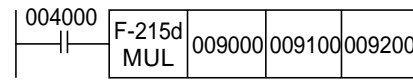
**F-215d
MUL**

**レジスタ間のバイナリ乗算(32ビット×32ビット)
(MULTIPLY)**

シンボル	F-215d MUL S1 S2 D			
機能	レジスタS1～S1+3の内容とレジスタS2～S2+3の内容をバイナリ乗算して、結果をレジスタD～D+7に格納する。			
演算内容	(S1～S1+3)×(S2～S2+3)→D～D+7			
S1	使用範囲C			
S2	使用範囲C			
D	使用範囲G			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S1～S1+3	不変		
	S2～S2+3	不変		
	D～D+7	演算結果(バイナリ64ビット)		
	フラグ	ゼロ 007357	キャリー 007356	エラー 007355
	0	0	0	0

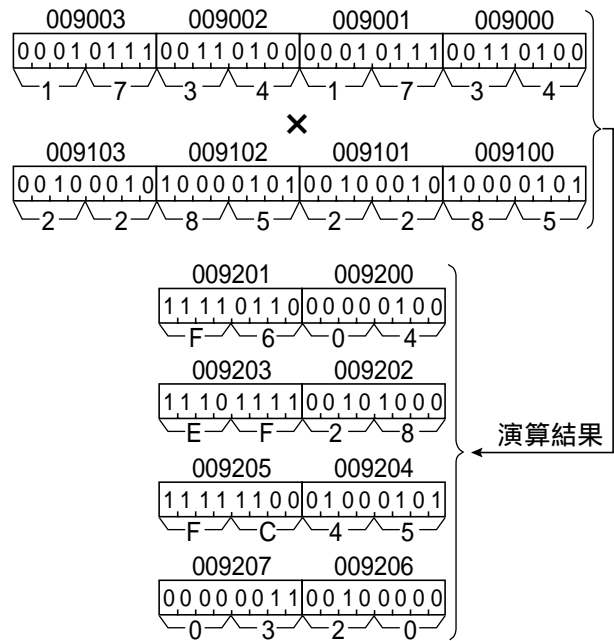
S1, S2, Dには必ず偶数アドレスを設定してください。(019003等は禁止)

[使用例]



命令	
STR	004000
F-215d	009000
	009100
	009200

入力条件004000がOFF ONの変化時に、レジスタ009000～009003の内容(32ビットデータ)とレジスタ009100～009103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009200～009207に格納します。

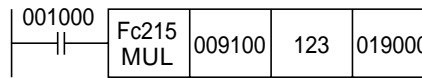


上記演算は17341734_(H) × 22852285_(H)
= 0320FC45EF28F604_(H)を示します。

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)

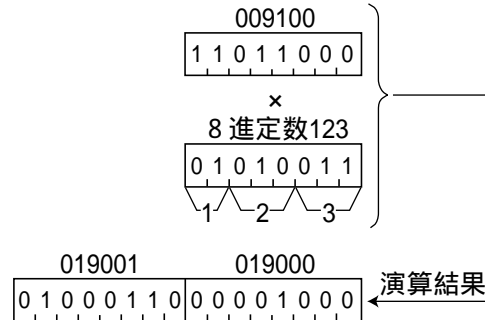
シンボル	Fc215 MUL S1 n D			
機能	レジスタS1の内容と8進定数nをバイナリ乗算し、結果をレジスタD、D+1に格納する。			
演算内容	S1 × n → D、D+1			
S1	使用範囲 A			
n	使用範囲 000 ~ 377(8)			
D	使用範囲 B			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S1	不変		
	D	演算結果(下位)		
	D+1	演算結果(上位)		
フラグ	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	0	0	0	0

[使用例]



命 令	
STR	001000
Fc215	009100 123 019000

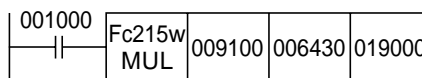
入力条件001000がOFF ONの変化時に、レジスタ009100の内容と8進定数123をバイナリ乗算して、結果をレジスタ019000、019001に格納します。



Fc215w MUL レジスタ間と定数のバイナリ乗算(16ビット× 16ビット)
(MULtiply)

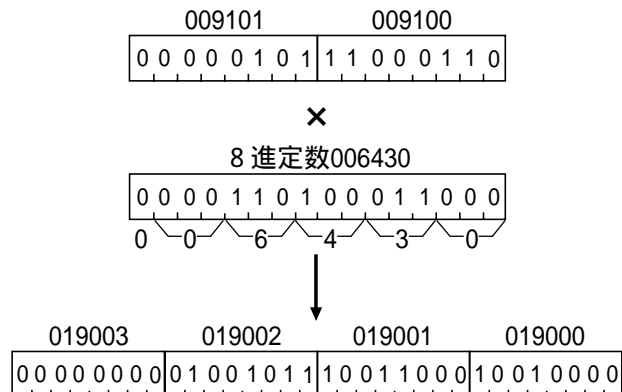
シンボル	Fc215w MUL S1 n D			
機能	レジスタS1、S1+1の内容(16ビットデータ)と8進定数nをバイナリ乗算し、結果をレジスタD ~ D+3に格納する。			
演算内容	(S1、S1+1) × n → D ~ D+3			
S1	使用範囲 B			
n	使用範囲 000000 ~ 177777(8)			
D	使用範囲 C			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S1、S1+1	不変		
	D	演算結果(下位)		
	D+1	演算結果		
	D+2	演算結果		
フラグ	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	0	0	0	0

[使用例]



命 令	
STR	001000
Fc215w	009100 006430 019000

入力条件001000がOFF ONの変化時に、レジスタ009100、009101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019000 ~ 019003に格納します。



S1、Dには必ず偶数アドレスを設定してください。

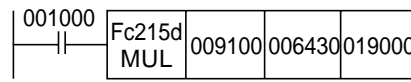
**Fc215d
MUL**

**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	Fc215d MUL S1 n D			
機能	レジスタS1~S1+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、結果をレジスタD~D+7に格納する。			
演算内容	$(S1 \sim S1+3) \times n \rightarrow D \sim D+7$			
S1	使用範囲 C			
n	使用範囲 000000 ~ 177777(8)			
D	使用範囲 G			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S1 ~ S1+3	不変		
	D ~ D+7	演算結果(バイナリ64ビット)		
フラグ	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	0	0	0	0

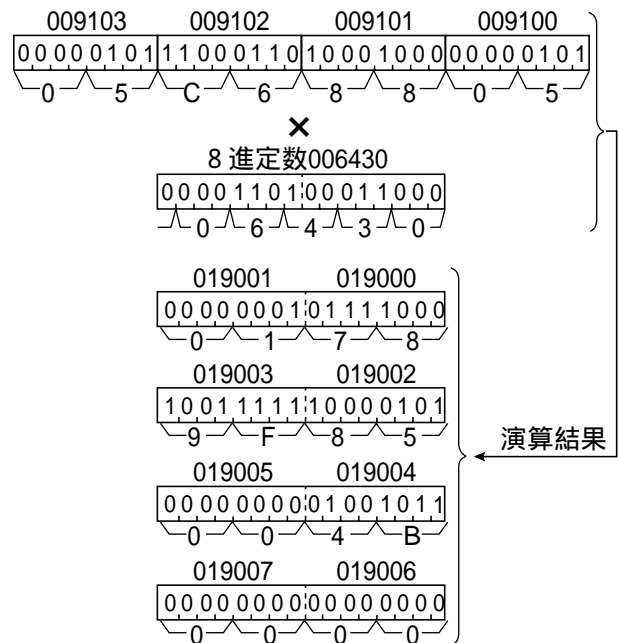
S1、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

[使用例]



命 令	
STR	001000
Fc215d	009100 006430 019000

入力条件001000がOFF ONの変化時に、レジスタ009100~009103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019000~019007に格納します。



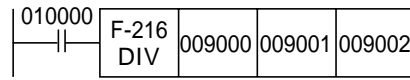
上記演算は05C68805(H) × 006430(8)
= 4B9F850178(H)を示します。

**F-216
DIV**

レジスタ間のバイナリ除算(8 ビット ÷ 8 ビット)
(DIVide)

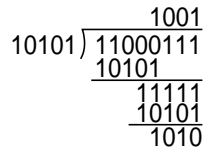
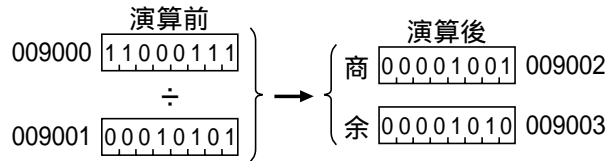
シンボル	— F-216 DIV S1 S2 D				
機能	レジスタS1の内容をレジスタS2の内容でバイナリ除算し、レジスタDに商、レジスタD+1に余を格納する。				
演算内容	S1 ÷ S2 → D, D+1				
S1	使用範囲 A				
S2	使用範囲 A				
D	使用範囲 B				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S1	不変			
	S2	不変			
	D	演算結果の商	・レジスタS2の内容が000(8)のとき不変。		
	D+1	演算結果の余			
フラグ	レジスタS2の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	000(8)	0	0	1	0
	上記以外			0	

[使用例]



命 令	
STR	010000
F-216	009000 009001 009002

入力条件010000がOFF ONの変化時に、レジスタ009000の内容をレジスタ009001の内容でバイナリ除算し、商をレジスタ009002、余をレジスタ009003に格納します。



**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					
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、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	使用範囲B				
S ₂	使用範囲B				
D	使用範囲C				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S ₁ 、S ₁ +1	不変			
	S ₂ 、S ₂ +1	不変			
	D	演算結果の商(下位)			
	D+1	" (上位)			
	D+2	演算結果の余(下位)			
フラグ	レジスタ S ₂ 、S ₂ +1の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	000000(8) 上記以外	0	0	1 0	0

[使用例]

命 令	
STR	010000
F-216w	019000 019002 019004

入力条件010000がOFF ONの変化時に、レジスタ019000、019001の内容(15ビットデータ)をレジスタ019002、019003の内容(15ビットデータ)でバイナリ除算し、商をレジスタ019004、019005、余をレジスタ019006、019007に格納します。

・レジスタ019001、019003のMSB(ビット7)は無視します。

S₁、S₂、Dには必ず偶数アドレスを設定してください。

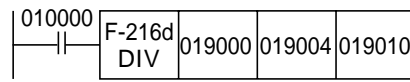
**F-216d
DIV**

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	— F-216d DIV S1 S2 D				
機能	レジスタS1～S1+3の内容(31ビットデータ)をレジスタS2～S2+3の内容(31ビットデータ)で除算し、レジスタD～D+3に商、D+4～D+7に余を格納する。				
演算内容	$(S1 \sim S1+3) \div (S2 \sim S2+3) \rightarrow D \sim D+7$				
S1	使用範囲 C				
S2	使用範囲 C				
D	使用範囲 G				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S1-S1+3	不変			
	S2-S2+3	不変			
D～D+3	演算結果の商(バイナリ31ビット)		レジスタS2～S2+3の内容が0のとき、不変。(演算を実行しない)		
	D+4～D+7				演算結果の余(バイナリ31ビット)
フラグ	レジスタ S2-S2+3の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	00000000(H)	0	0	1	0
	上記以外	0	0	0	0

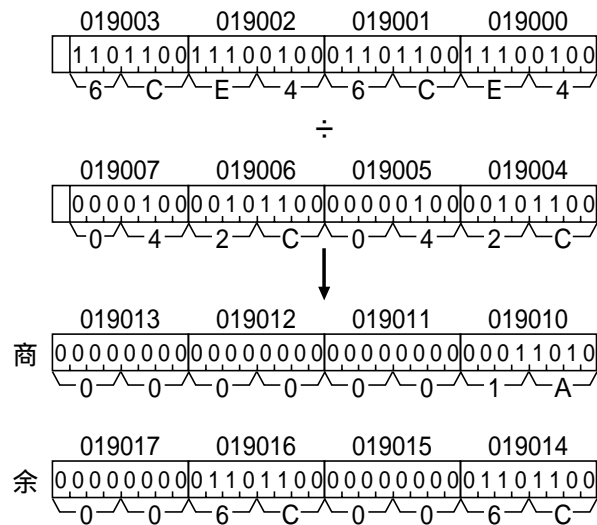
S1、S2、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

[使用例]



命 令	
STR	010000
F-216d	019000
	019004
	019010

入力条件010000がOFF ONの変化時に、レジスタ019000～019003の内容(31ビットデータ)をレジスタ019004～019007の内容(31ビットデータ)でバイナリ除算し、商をレジスタ019010～019013、余をレジスタ019014～019017に格納します。
下記演算は、6CE46CE4(H) ÷ 042C042C(H) = 1A(H) 余り6C006C(H)を示します。

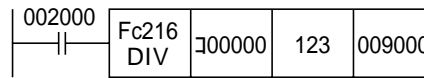


・レジスタ019003、019007のMSB(ビット7)は、無視します。

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)

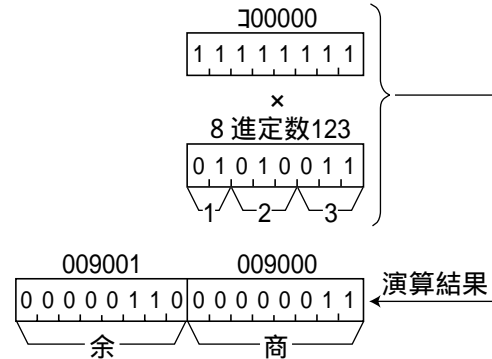
シンボル	— Fc216 DIV S1 n D				
機能	レジスタS1の内容を8進定数nでバイナリ除算し、レジスタDに商、レジスタD+1に余を格納する。				
演算内容	S1 ÷ n → D, D+1				
S1	使用範囲 A				
n	使用範囲 000 ~ 377 ₍₈₎				
D	使用範囲 B				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S	不変			
	D	演算結果の商			
	D+1	演算結果の余			
フラグ	8進定数 n	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	000 ₍₈₎	0	0	1	0
	上記以外	0	0	0	0

[使用例]



命 令	
STR	02000
Fc216	
	300000
	123
	009000

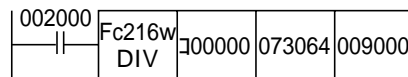
入力条件002000がOFF ONの変化時に、レジスタ300000の内容を8進定数123でバイナリ除算し、商をレジスタ009000、余をレジスタ009001に格納します。



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)

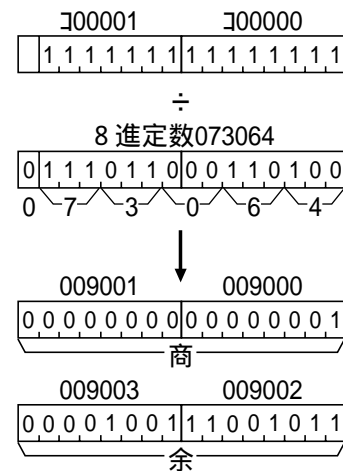
シンボル	— Fc216w DIV S1 n D				
機能	レジスタS1, S1+1の内容(15ビットデータ)を8進定数nでバイナリ除算し、レジスタD, D+1に商、レジスタD+2, D+3に余を格納する。				
演算内容	(S1, S1+1) ÷ n → D, D+1, D+2, D+3				
S1	使用範囲 B				
n	使用範囲 000000 ~ 077777 ₍₈₎				
D	使用範囲 C				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S1, S1+1	不変			
	D	演算結果の商(下位)			
	D+1	" (上位)			
	D+2	演算結果の余(下位)			
	D+3	" (上位)			
フラグ	8進定数 n	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	000000 ₍₈₎	0	0	1	0
	上記以外	0	0	0	0

[使用例]



命 令	
STR	002000
Fc216w	
	300000
	073064
	009000
	009001
	009002
	009003

入力条件002000がOFF ONの変化時に、レジスタ300000、009001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009000、009001、余をレジスタ009002、009003に格納します。



・レジスタ300001のMSB(ビット 7)は無視します。

S1, Dには必ず偶数アドレスを設定してください。

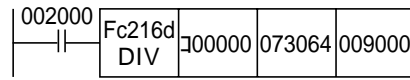
**Fc216d
DIV**

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	— Fc216d DIV S ₁ n D			
機能	レジスタS ₁ ~ S ₁ +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、レジスタD+4 ~ D+7に余を格納する。			
演算内容	(S ₁ ~ S ₁ +3) ÷ n → D ~ D+7			
S ₁	使用範囲 C			
n	使用範囲 000000 ~ 177777 ⁽⁸⁾			
D	使用範囲 G			
演算条件	入力信号の立上り(OFF ON)			
演算後の内容	S ₁ - S ₁ +3	不変		
	D ~ D+3	演算結果の商 (バイナリ31ビット)		
	D+4 ~ D+7	演算結果の余 (バイナリ31ビット)		
フラグ	8進定数n	ゼロ 007357	キャリー 007356	エラー 007355
	上記以外	0	0	1
				ノンキャリー 007354

S₁, Dには必ず偶数アドレスを設定してください。

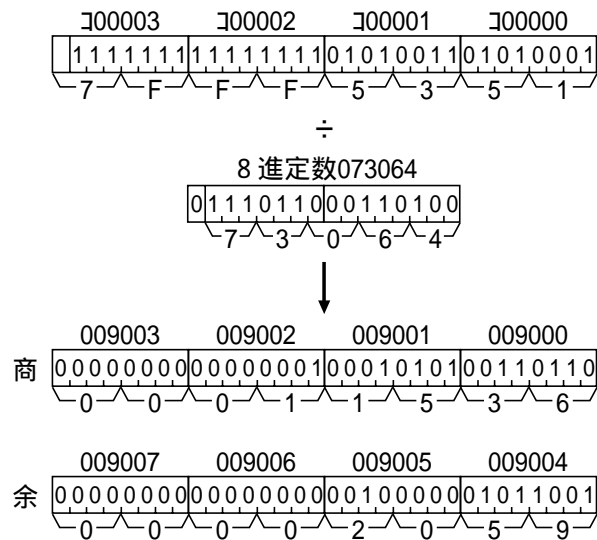
[使用例]



命 令	
STR	02000
Fc216d	000000 073064 009000

入力条件002000がOFF ONの変化時に、レジスタ000000 ~ 000003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009000 ~ 009003、余をレジスタ009004 ~ 009007に格納します。

下記演算は、7FFF5351^(H) ÷ 73064⁽⁸⁾ = 11536^(H)の商と余り2059^(H)を示します。



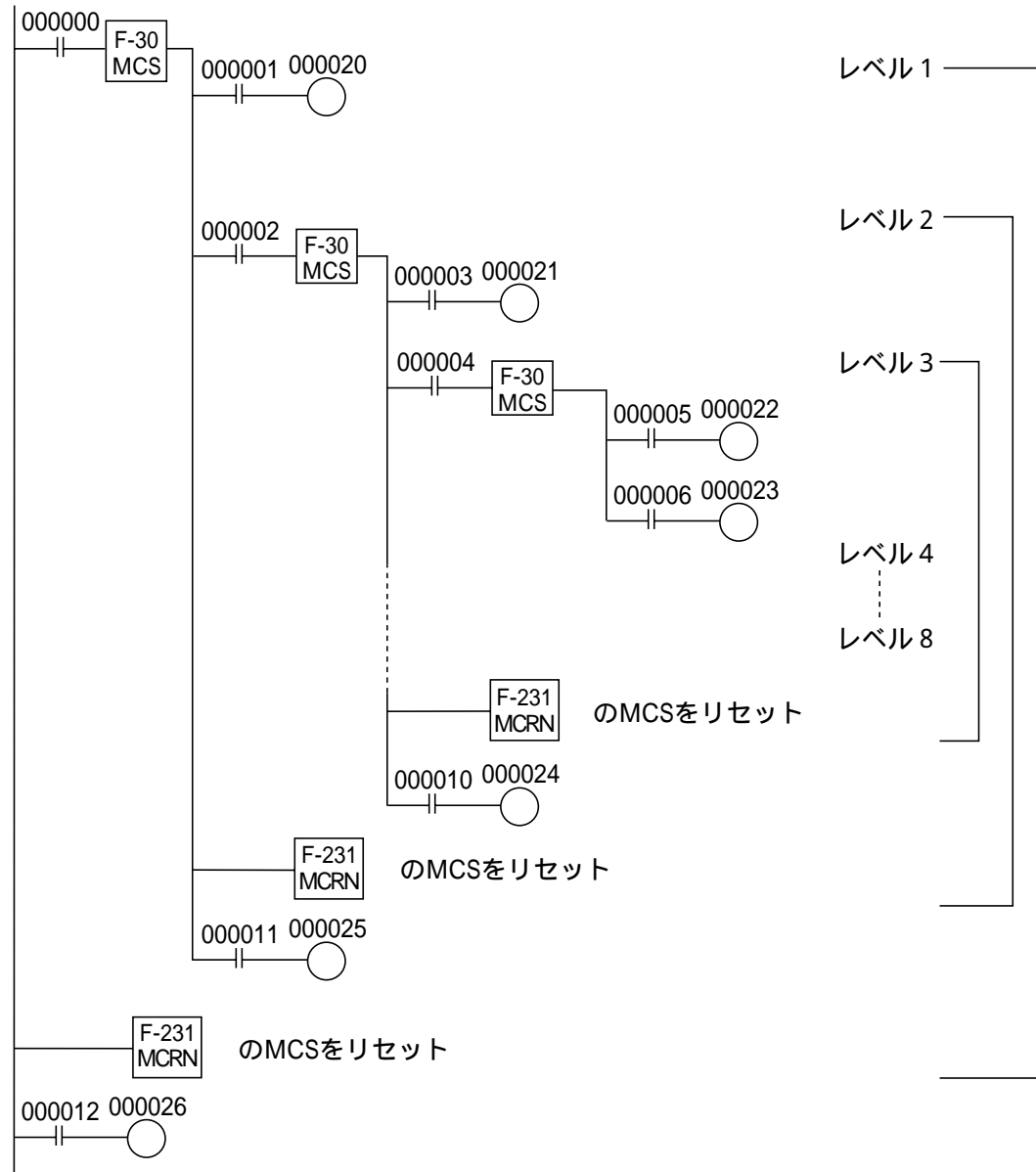
・レジスタ000003のMSB(ビット7)は無視します。

**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)はネスティングできませんが、F-231(MCRN)を使用するとレベル8までネスティングできます。

[使用例]



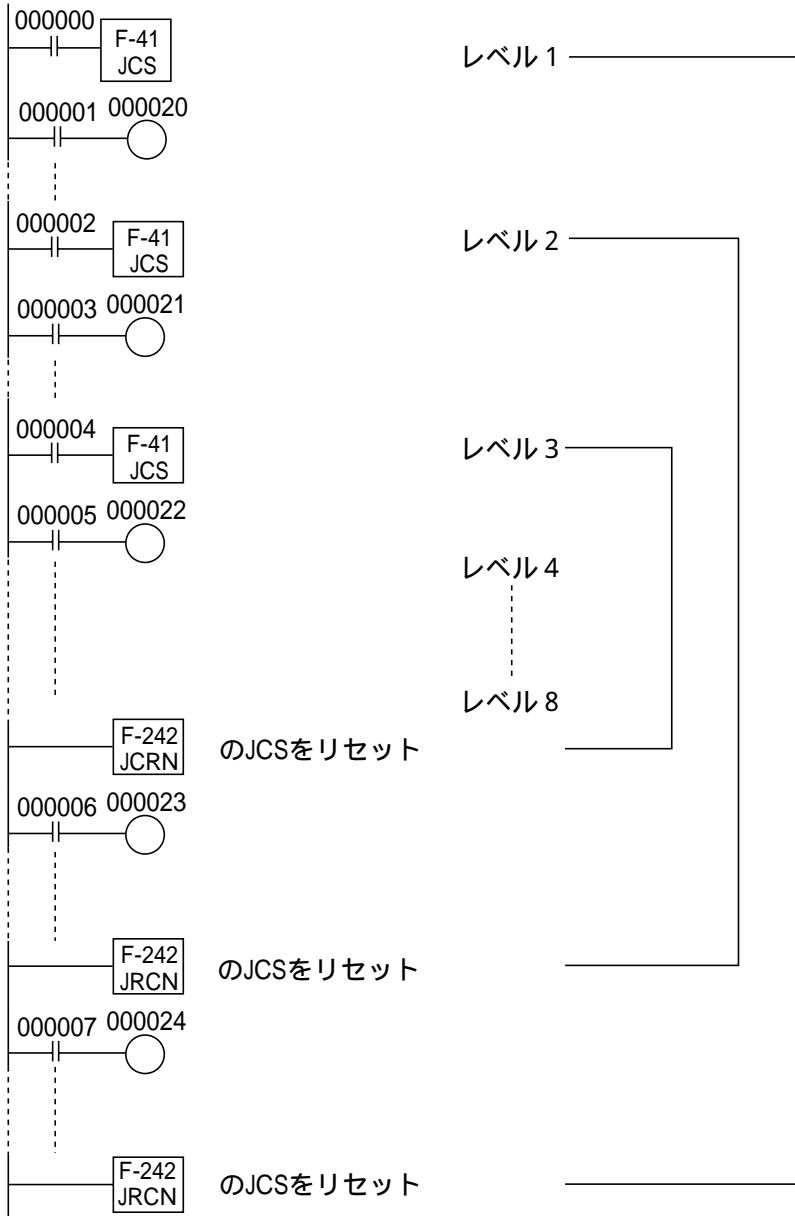
・ F-30(MCS)、 F-31(MCR) の説明も参照願います。

**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のとき、F-242(JCRN)までにあるEND命令を除くすべての命令を実行しません。
F-42(JCR)はネスティングできませんが、F-242(JCRN)を使用するとレベル8までネスティングできます。

[使用例]



・ F-41(JCS)、F-42(JCR)の説明も参照願います。

F-252 ASC **HEX(16進)コード ASCIIコード変換**
(ASCII)

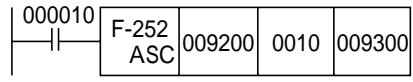
シンボル	$\overline{F-252\ ASC}$ S n D
機能	レジスタSを先頭とするnバイトの領域にて、16進(HEX)コードをASCIIコードに変換し、レジスタDを先頭に格納する。変換はSの下位4ビット側から変換する。
演算内容	S, S+1, ……、S+n-1 → ASCII変換 → D, D+1, ……、D+2n-1
S	使用範囲 A
n	使用範囲 0000 ~ 1777 ⁽⁸⁾ 1 (0000 ⁽⁸⁾ のとき1024バイト)
D	使用範囲 A 1
演算条件	入力信号の立上り(OFF ON)
演算後の内容	S ~ S+n-1 不変
	D ~ D+n-1 演算結果 2
	フラグ 不変

1 n、Dで設定する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。
2 演算結果が、ファイル番号0のタイマ・カウンタの接点領域(ファイルアドレス00001600 ~ 00001777⁽⁸⁾等)に入らないようにしてください。

16進コードとASCIIコードの関係

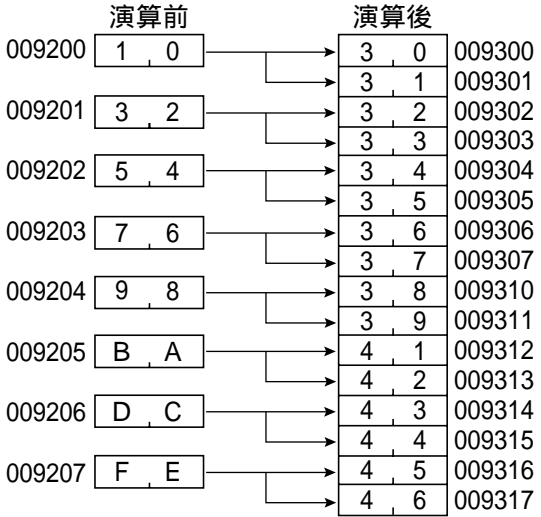
16進コード	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ASCIIコード	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46

[使用例]



命 令	
STR	000010
F-252	009200
	0010
	009300

入力条件000010がOFF ONの変化時に、レジスタ009200 ~ 009207の8(0010⁽⁸⁾)バイトのデータをASCII変換し、レジスタ009300を先頭に格納します。

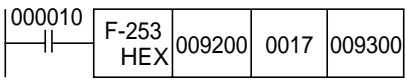


**F-253
HEX**

**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を先頭とする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領域の最終アドレスに格納して、変換を中止する。				
演算内容	S, S+1, ……、S+n-1 → HEX変換 → D, D+1, ……、D+ $\frac{n}{2}-1$ 変換不能コード → D+ $\frac{n}{2}-1$ (nが奇数のとき最終アドレス=D+ $\frac{n-1}{2}$)				
S	使用範囲 A				
n	使用範囲 0000 ~ 1777 ₍₈₎ (0000 ₍₈₎ のとき1024バイト)				
D	使用範囲 A				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S ~ S+n-1	不変			
	D ~ D+ $\frac{n}{2}-1$	演算結果	・変換不能コードがあると、コードをレジスタD領域の最終アドレスに格納して変換を中止。		
	D+ $\frac{n}{2}$	正常時、不変			
フラグ	変換動作	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	変換不能コード有り	0	0	1	0
	上記以外			0	

[使用例]



命令	
STR	000010
F-253	009200 0017 009300

入力条件000010がOFF ONの変化時に、レジスタ009200~009216の017₍₈₎(15)バイトにあるASCIIデータを、16進変換してレジスタ009300を先頭に格納します。

	演算前	演算後	
009200	3, 0	1, 0	009300
009201	3, 1		
009202	3, 2	3, 2	009301
009203	3, 3		
009204	3, 4	5, 4	009302
009205	3, 5		
009206	3, 6	7, 6	009303
009207	3, 7		
009210	3, 8	9, 8	009304
009211	3, 9		
009212	4, 1	B, A	009305
009213	4, 2		
009214	4, 3	D, C	009306
009215	4, 4		
009216	4, 5	0, E	009307

・変換不能なASCIIコードが存在すると、そのASCIIコードを最終レジスタ009307に格納します。

演算結果が、ファイル番号0のタイマ・カウンタの接点領域(ファイルアドレス00001600 ~ 00001777₍₈₎等)に入らないようにしてください。

nの値を奇数バイト数にすると、最終アドレスの上位4ビットデータは0になります。

16進コードとASCIIコードの関係

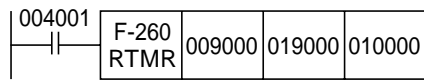
16進コード	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ASCIIコード	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46

**F-260
RTMR**

減算タイマ(設定値、レジスタ指定)

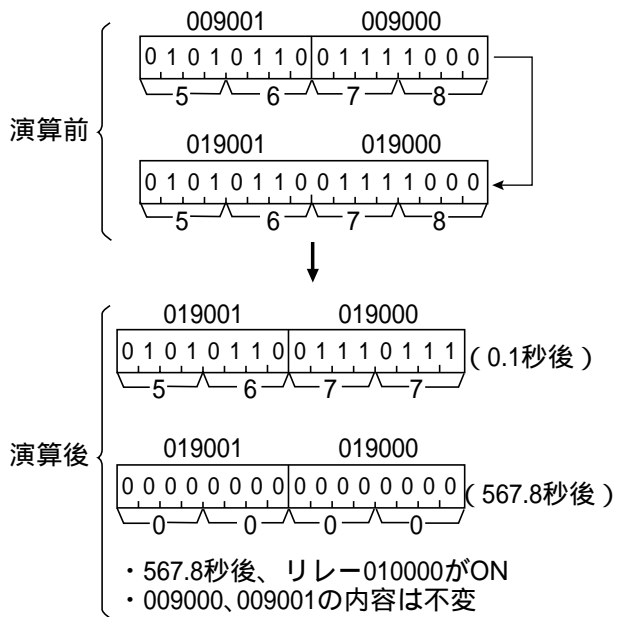
シンボル	F-260 RTMR S D BIT				
機能	レジスタD、D+1の内容(タイマ現在値)は、レジスタS、S+1の内容(タイマ設定値)から0.1秒ごとに-1され、0になるとリレー-BITをONし、入力信号がONの間は保持する。				
演算内容	$(S, S+1) - \text{経過時間} \rightarrow (D, D+1)$ $(D, D+1) = 0$ になるとBIT(ON)				
S	使用範囲 B				
D	使用範囲 B				
BIT	使用範囲 K				
演算条件	入力信号の立上り(OFF ON)で計数を開始し、入力信号がONの間は計数				
演算前の内容	S、S+1	タイマ設定値0000 ~ 9999 (BCD 4桁、0 ~ 999.9秒)			
	D、D+1	タイマ現在値 (S、S+1の内容と同じ)			
	BIT	OFF			
演算後の内容	S、S+1	不変			
	D、D+1	演算結果のタイマ現在値 (0000 ~ 9999、BCD 4桁)			
	BIT	ON(タイマ現在値 = 0 のとき)			
フラグ	レジスタDの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	BCDコード	0	0	0	0
	BCDコード以外	0	0	1	0

[使用例]



命令	
STR	004001
F-260	009000
	019000
	010000

入力条件004001がOFF ONの変化後、ONの間、レジスタ019000、019001の内容(タイマ現在値)は0.1秒ごとにレジスタ009000、009001の内容(タイマ設定値)から-1されます。0になるとリレー010000がONし、004001がONの間は保持します。



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#0201参照)で使用的場合、レジスタDはキープリレー領域およびレジスタ(b00000以降、009000以降)を使用してください。
- ・機能は減算式TMR命令と同様です。

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

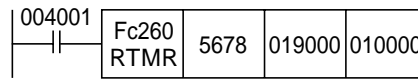
(注)プログラムを入力後、「演算条件がON」かつ「レジスタD、D+1の内容が0」で運転モードに変えると、出力リレー(BIT)がONします。

**Fc260
RTMR**

減算タイマ(定数、レジスタ指定)

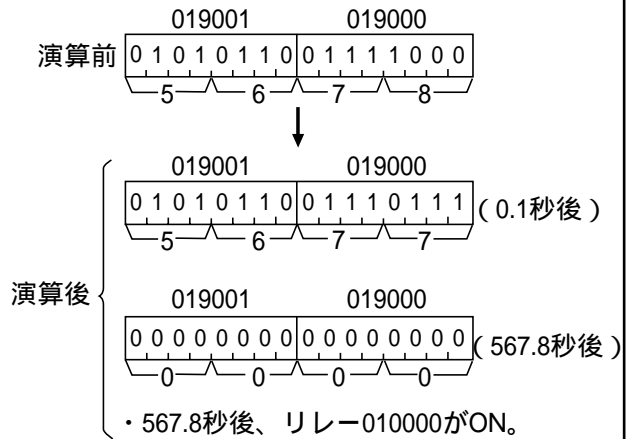
シンボル	Fc260 RTMR n D BIT				
機能	レジスタD、D+1の内容(タイマ現在値)は、n(タイマ設定値)から0.1秒ごとに-1され、0になるとリレー-BITをONし、入力信号がONの間は保持する。				
演算内容	$n - \text{経過時間} \rightarrow (D, D+1)$ $(D, D+1) = 0 \text{ になると BIT (ON)}$				
n	タイマ設定値 0000 ~ 9999 (0 ~ 999.9秒)				
D	使用範囲 B				
BIT	使用範囲 K				
演算条件	入力信号の立上り(OFF ON)で計数を開始し、入力信号がONの間は計数				
演算前の内容	D、D+1	タイマ現在値 (nの値と同じ)			
	BIT	OFF			
演算後の内容	D、D+1	演算結果のタイマ現在値 (0000 ~ 9999、BCD 4桁)			
	BIT	ON (タイマ現在値 = 0 のとき)			
フラグ	レジスタDの内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	BCDコード	0	0	0	0
	BCDコード以外	0	0	1	0

[使用例]



命 令	
STR	004001
Fc260	5678
	019000
	010000

入力条件004001がOFF ONの変化後、ONの間、レジスタ019000、019001の内容(タイマ現在値)が0.1秒ごとに5678(タイマ設定値567.8秒)から-1されます。0になるとリレー010000がONし、004001がONの間は保持します。



- ・ 停電保持モード(システムメモリ#0201参照)で使用的場合、レジスタDはキープリレー領域およびレジスタ(b00000以降、009000以降)を使用してください。
- ・ 機能は減算式TMR命令と同様です。

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

(注)プログラムを入力後、「演算条件がON」かつ「レジスタD、D+1の内容が0」で運転モードに変えると、出力リレー(BIT)がONします。

**F-261
RCNT**

減算カウンタ(設定値、レジスタ指定)

シンボル	F-261 RCNT S D BIT				
機能	リセット入力 (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 がOFF ONに変化するごとに - 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 (ON)のとき、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	$(S, S+1) - \text{計数入力回数} \rightarrow (D, D+1)$ $(D, D+1) = 0$ になればBIT(OFF)				
S	使用範囲 B				
D	使用範囲 B				
BIT	使用範囲 K				
演算条件	リセット入力 (OFF)の間、計数入力 (OFF ON)				
演算前の内容	S、S+1	カウンタ設定値 0000 ~ 9999 (BCD 4桁)			
	D、D+1	カウンタ現在値 (S、S+1の内容と同じ)			
	BIT	OFF			
演算後の内容	S、S+1	不変			
	D、D+1	演算結果のカウンタ現在値 (0000 ~ 9999、BCD 4桁)			
	BIT	ON (カウンタ現在値 = 0のとき)			
フラグ	レジスタDの内容	ゼロ 007357	キャリー 007356	エラー 007355	ハンキャリー 007354
	BCDコード	0	0	0	0
	BCDコード以外	0	0	1	0

[使用例]

リセット入力004001(OFF)の間、レジスタ019000、019001の内容(カウンタ現在値)はレジスタ009000、009001の内容(カウンタ設定値)から計数入力004000がOFF ONするごとに - 1され、0になるとリレー010000をONして保持します。リセット入力004001(ON)のとき、(レジスタ019000、019001の内容)=(レジスタ009000、009001の内容)およびリレー010000(OFF)となります。

カウンタ設定値が5678回の場合は次のとおりです。

演算前

009001 009000
0 1 0 1 0 1 1 0 0 1 1 1 1 0 0 0
5 6 7 8

019001 019000
0 1 0 1 0 1 1 0 0 1 1 1 1 0 0 0
5 6 7 8

↓

019001 019000
0 1 0 1 0 1 1 0 0 1 1 1 0 1 1 1 (計数入力回数 : 1回)

019001 019000
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 (計数入力回数 : 5678回)

- ・ 計数入力回数が5678回後、リレー010000がON。
- ・ 009000、009001の内容は不変。

命 令	
STR	004000
STR	004001
F-261	009000
	019000
	010000

機能は減算式CNT命令と同様です。

リセット入力	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 がOFF ONするごとに - 1	OFF
OFF(現在値 = 0)	0	ON

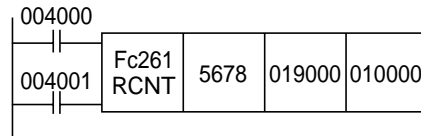
- (注1) プログラムを入力後、「リセット入力がOFF」かつ「レジスタD、D+1の内容が0」で運転モードに変えると、出力リレー(BIT)がONします。
- (注2) レジスタDには、キーリレー領域およびレジスタ(009000以降)を使用してください。入出力リレー領域または補助リレー領域のCxxxxxを使用すると、電源OFF ON時に出力リレー(BIT)がONします。

**Fc261
RCNT**

減算カウンタ(定数、レジスタ指定)

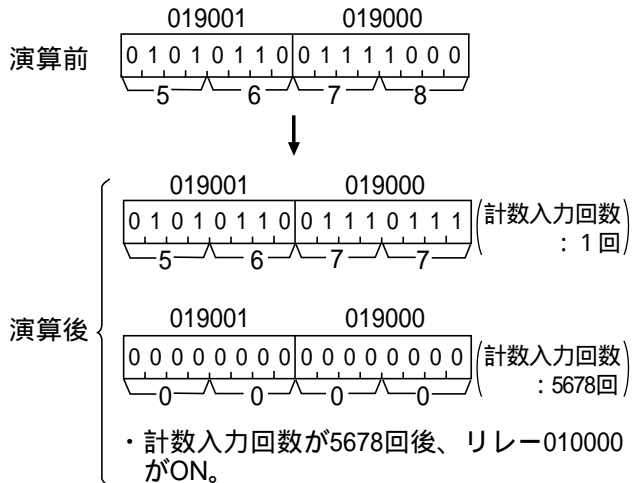
シンボル				
機能	リセット入力 (OFF)の間、レジスタ D、D+1の内容(カウンタ現在値)は、n (カウンタ設定値)から計数入力 が OFF ONに変化するごとに - 1され、0 になるとリレー-BITをONして保持する。リセット入力 (ON)のとき、カウンタ現在値 = カウンタ設定値、およびリレー-BIT(OFF)となる。			
演算内容	$n - \text{計数入力回数} \rightarrow (D, D+1)$ $(D, D+1) = 0 \text{ になれば BIT (ON)}$			
n	カウンタ設定値 0000 ~ 9999 (0 ~ 9999回)			
D	使用範囲 B			
BIT	使用範囲 K			
演算条件	リセット入力 (OFF)の間、計数入力 (OFF ON)			
演算前の内容	D、D+1	カウンタ現在値 (nと同じ)		
	BIT	OFF		
演算後の内容	D、D+1	演算結果のカウンタ現在値 (0000 ~ 9999、BCD 4桁)		
	BIT	ON (カウンタ現在値 = 0 のとき)		
フラグ	レジスタDの内容	ゼロ 007357	キャリー 007356	エラー 007355
	BCDコード	0	0	0
	BCDコード以外	0	0	1

[使用例]



命 令	
STR	004000
STR	004001
Fc261	5678
	019000
	010000

リセット入力004001(OFF)の間、レジスタ019000、019001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004000がOFF ONするごとに - 1され、0 になるとリレー-010000をONして保持します。
 リセット入力004001(ON)のとき、(レジスタ009000、009001の内容)=(nの値5678)、およびリレー-010000(OFF)となります。
 カウンタ設定値が5678回の場合は次のとおりです。



機能は減算式CNT命令と同様です。

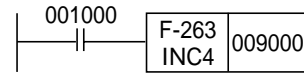
リセット入力	カウンタ現在値	リレー-BIT
ON	カウンタ設定値	OFF
OFF(現在値 > 0)	計数入力 がOFF ONするごとに - 1	OFF
OFF(現在値 = 0)	0	ON

- (注1)プログラムを入力後、「リセット入力がOFF」かつ「レジスタD、D+1の内容が0」で運転モードに変えると、出力リレー(BIT)がONします。
- (注2)レジスタDには、キーリレー領域およびレジスタ(009000以降)を使用してください。入出力リレー領域または補助リレー領域のxxxxxを使用すると、電源OFF ON時に出力リレー(BIT)がONします。

**F-263
INC4** 加算(+4)カウンタ(1バイトバイナリ)
(INCRement)

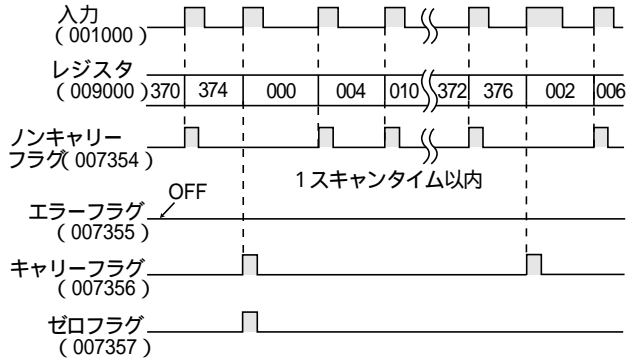
シンボル	F-263 INC4		D			
機能	レジスタDの内容(バイナリデータ)を加算(+4)カウントする。					
演算内容	D +4 → D					
D	使用範囲A					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果(バイナリコード)				
	フラグ	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		374 000 のとき	1	1	0	0
		375 001 376 002 377 003 のとき	0	1	0	0
		上記以外	0	0	0	1

[使用例]



命 令	
STR	001000
F-263	009000

入力条件001000がOFF ONの変化時に、レジスタ009000の内容(バイナリデータ)を加算(+4)カウントします。

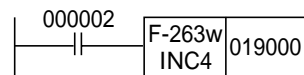


(類似命令) F-63、F-63w、F-63d、F-163、F-163w、F-163d、F-263d、F-263w

**F-263w
INC4** 加算(+4)カウンタ(1ワードバイナリ)
(INCRement)

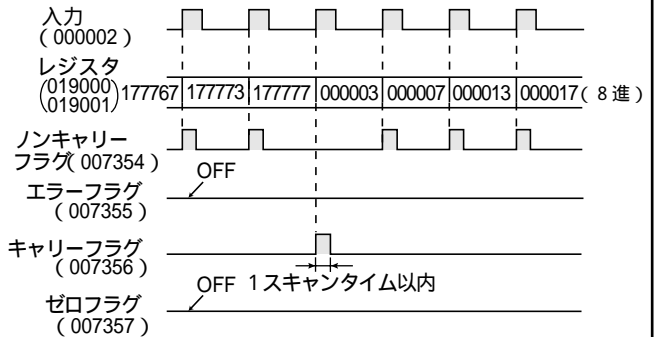
シンボル	F-263w INC4		D			
機能	レジスタD、D+1の内容(バイナリデータ)を加算(+4)カウントする。					
演算内容	D、D+1 +4 → D、D+1					
D	使用範囲B ・Dには必ず偶数アドレスを設定してください。(019003等は禁止)					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果(下位)				
	D+1	演算結果(上位)				
	フラグ	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		177774 000000 のとき	1	1	0	0
		177775 000001 177776 000002 177777 000003 のとき	0	1	0	0
上記以外		0	0	0	1	

[使用例]



命 令	
STR	000002
F-263w	019000

入力条件000002がOFF ONの変化時に、レジスタ019000、019001の内容(バイナリデータ)を加算(+4)します。



(類似命令) F-63、F-63w、F-63d、F-163、F-163w、F-163d、F-263、F-263d

F-263d 加算(+ 4)カウンタ(2ワードバイナリ)
INC4 (INCRement)

シンボル	F-263d INC4		D			
機能	レジスタD ~ D+3の内容(バイナリデータ)を加算(+ 4)カウントする。					
演算内容	D ~ D+3 +4 → D ~ D+3					
D	使用範囲 C ・Dには必ず偶数アドレスを設定してください。(019003等は禁止)					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D ~ D+3	演算結果(D : 下位 ~ D+3 : 上位)				
		演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	フラグ	3777777774 0000000000のとき	1	1	0	0
		3777777775 0000000001、 3777777776 0000000002、 3777777777 0000000003のとき	0	1	0	0
		上記以外	0	0	0	1

[使用例]

命 令	
STR	000002
F-263d	019000

入力条件000002がOFF ONの変化時に、レジスタ019000 ~ 019003の内容(バイナリデータ)を加算カウント(+ 4)します。

1 スキャンタイム以内

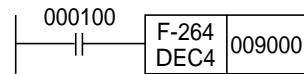
(類似命令) F-63、F-63w、F-63d、F-163、F-163w、F-163d、F-263、F-263w

F-264 減算(- 4)カウンタ(1バイトバイナリ)
DEC4 (DECrement)

シンボル	F-264 DEC4	D				
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	D - 4 → D					
D	使用範囲 A					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果(バイナリコード)				
	フラグ	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		004 000のとき	1	0	0	1
		003 377 002 376 001 375 000 374のとき	0	1	0	0
		上記以外	0	0	0	1

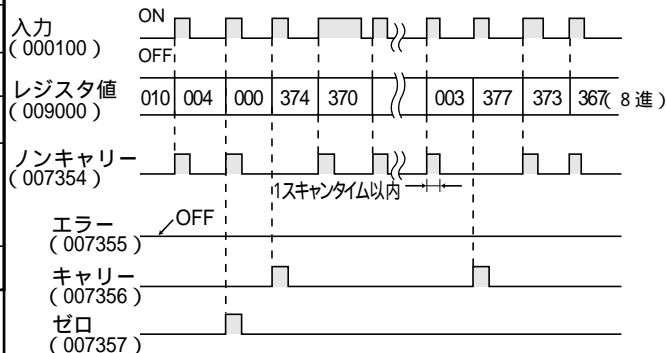
(類似命令) F-64、F-64w、F-64d、F-164、F-164w、F-164d、F-264w、F-264d

[使用例]



命 令	
STR	000100
F-264	009000

入力条件000100がOFF ONの変化時に、レジスタ009000の内容(バイナリデータ)を減算(- 4)カウントします。

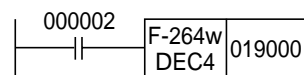


F-264w 減算(- 4)カウンタ(1ワードバイナリ)
DEC4 (DECrement)

シンボル	F-264w DEC4	D				
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	D、D+1 - 4 → D、D+1					
D	使用範囲 B ・ Dには必ず偶数アドレスを設定してください。(019003等は禁止)					
演算条件	入力信号の立上り(OFF ON)					
演算後の内容	D	演算結果(下位)				
	D+1	演算結果(上位)				
	フラグ	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		000004 000000 のとき	1	0	0	1
		000003 177777 000002 177776 000001 177775 000000 177774 のとき	0	1	0	0
上記以外		0	0	0	1	

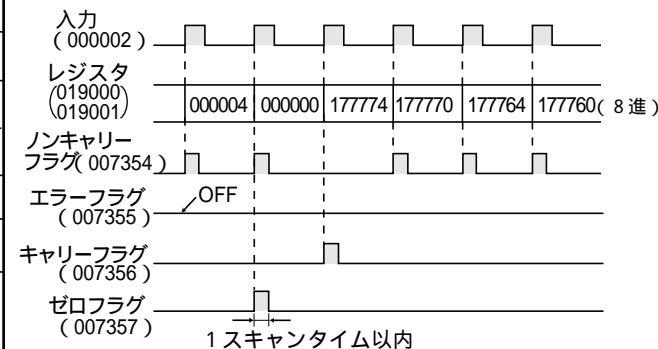
(類似命令) F-64、F-64w、F-64d、F-164、F-164w、F-164d、F-264、F-264d

[使用例]



命 令	
STR	000002
F-264w	019000

入力条件000002がOFF ONの変化時に、レジスタ019000、019001の内容(バイナリデータ)を減算(- 4)カウントします。



F-264d 減算(- 4)カウンタ(2ワードバイナリ)
DEC4 (DECrement)

シンボル	F-264d DEC4		D		
機能	レジスタD ~ D+3の内容(バイナリデータ)を、減算(- 4)カウントする。				
演算内容	D ~ D+3 - 4 → D ~ D+3				
D	使用範囲 C ・ Dには必ず偶数アドレスを設定してください。(019003等は禁止)				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	D ~ D+3	演算結果(D : 下位 ~ D+3 : 上位)			
	演算結果(8進)	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	0000000004 0000000000のとき	1	0	0	1
	0000000003 3777777777、 0000000002 3777777776、 0000000001 3777777775、 0000000000 3777777774のとき	0	1	0	0
	上記以外	0	0	0	1

[使用例]

命 令	
STR	000002
F-264d	019000

入力条件000002がOFF ONの変化時に、レジスタ019000 ~ 019003の内容(バイナリデータ)を減算(- 4)カウントします。

1 スキャンタイム以内

(類似命令) F-64、F-64w、F-64d、F-164、F-164w、F-164d、F-264、F-264w

**F-300
XFER**

1 バイトデータの転送

シンボル			[使用例]	<table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>004004</td> </tr> <tr> <td>F-300</td> <td>009000 009000 000001</td> </tr> </tbody> </table>	命 令		STR	004004	F-300	009000 009000 000001
命 令										
STR	004004									
F-300	009000 009000 000001									
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		<p>入力条件004004がONのとき、レジスタ009000の内容を、レジスタ000001に転送します。</p>							
S	使用範囲 A									
D	使用範囲 A									
演 算 条 件	入力信号がONのとき									
演算後の内容	S	不変								
	D	レジスタSの内容								
	フラグ	不変								

ファイルレジスタ、間接指定、インデックス修飾は使用不可です。
 (類似命令) F-00、F-00w、F-00d、F-70、F-70w、F-70d、F-74、F-74w、F-74d、F-76、F-76w、F-76d

**F-300w
XFER**

1 ワードデータの転送

シンボル			[使用例]	<table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>004000</td> </tr> <tr> <td>F-300w</td> <td>009000 009001 000000 000001</td> </tr> </tbody> </table>	命 令		STR	004000	F-300w	009000 009001 000000 000001
命 令										
STR	004000									
F-300w	009000 009001 000000 000001									
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		<p>入力条件004000がONのとき、レジスタ009000、009001の内容(1 ワードデータ)を、レジスタ000000、000001に転送します。</p>							
S	使用範囲 B									
D	使用範囲 B									
演 算 条 件	入力信号がONのとき									
演算後の内容	S、S+1	不変								
	D	レジスタSの内容								
	D+1	レジスタS+1の内容								
	フラグ	不変								

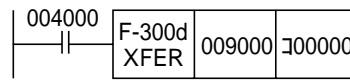
ファイルレジスタ、間接指定、インデックス修飾は使用不可です。
 また、S、Dには必ず偶数アドレスを設定してください。(019003等は禁止)
 (類似命令) F-00、F-00w、F-00d、F-70、F-70w、F-70d、F-74、F-74w、F-74d、F-76、F-76w、F-76d

**F-300d
XFER**

2ワードデータの転送

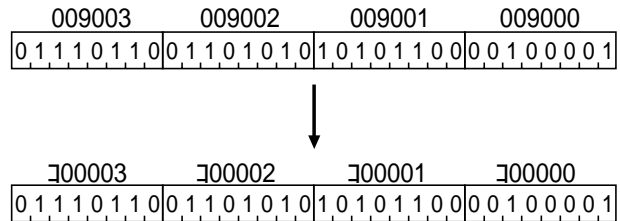
シンボル	
機能	レジスタS~S+3の内容(2ワードデータ)を、レジスタD~D+3に転送する。
演算内容	S~S+3 → D~D+3
S	使用範囲 C
D	使用範囲 C
演算条件	入力信号がONのとき
演算後の内容	S~S+3 不変
	D~D+3 レジスタS~S+3の内容
	フラグ 不変

[使用例]



命 令	
STR	004000
F-300d	009000 100000

入力条件004000がONのとき、レジスタ009000~009003の内容(2ワードデータ)を、レジスタ100000~100003に転送します。



ファイルレジスタ、間接指定、インデックス修飾は使用不可です。

また、S、Dには必ず4バイト単位の偶数アドレスを設定してください。(009000、009004、009010等)
(類似命令) F-00、F-00w、F-00d、F-70、F-70w、F-70d、F-74、F-74w、F-74d、F-76、F-76w、F-76d

**F-310
SADD**

**レジスタ間の符号付きバイナリ加算(31ビット+31ビット)
(Signed ADD)**

シンボル	— F-310 SADD S1 S2 D				[使用例]	<table border="1"> <tr><th colspan="2">命 令</th></tr> <tr><td>STR</td><td>004001</td></tr> <tr><td>F-310</td><td>009000</td></tr> <tr><td></td><td>009010</td></tr> <tr><td></td><td>009020</td></tr> </table>	命 令		STR	004001	F-310	009000		009010		009020																									
命 令																																									
STR	004001																																								
F-310	009000																																								
	009010																																								
	009020																																								
機能	レジスタS1～S1+3の内容とS2～S2+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。				<p>004001</p> <table border="1"> <tr><td>F-310 SADD</td><td>009000</td><td>009010</td><td>009020</td></tr> </table>	F-310 SADD	009000	009010	009020																																
F-310 SADD	009000	009010	009020																																						
演算内容	(S1 ~ S1+3) + (S2 ~ S2+3) → (D ~ D+3)				<p>入力条件004001がOFF ONの変化時に、レジスタ009000～009003の内容とレジスタ009010～009013の内容を符号付バイナリ加算して、その結果をレジスタ009020～009023に格納します。</p> <p>下記演算は、 - 16 + 19088743 = 19088727を示します。</p>																																				
S1	使用範囲 C																																								
S2	使用範囲 C																																								
D	使用範囲 C																																								
演算条件	入力信号の立上り(OFF ON)				<p>符号</p> <p>↓</p> <table border="1"> <tr><td>009003</td><td>009002</td><td>009001</td><td>009000</td></tr> <tr><td>11111111</td><td>11111111</td><td>11111111</td><td>11110000</td></tr> <tr><td colspan="4">- 16(D)</td></tr> </table> <p>+</p> <p>符号</p> <p>↓</p> <table border="1"> <tr><td>009013</td><td>009012</td><td>009011</td><td>009010</td></tr> <tr><td>00000001</td><td>00100011</td><td>01000101</td><td>01100111</td></tr> <tr><td colspan="4">19088743(D)</td></tr> </table> <p>↓</p> <p>符号</p> <p>↓</p> <table border="1"> <tr><td>09023</td><td>09022</td><td>09021</td><td>09020</td></tr> <tr><td>00000001</td><td>00100011</td><td>01000101</td><td>01010111</td></tr> <tr><td colspan="4">19088727(D)</td></tr> </table>	009003	009002	009001	009000	11111111	11111111	11111111	11110000	- 16(D)				009013	009012	009011	009010	00000001	00100011	01000101	01100111	19088743(D)				09023	09022	09021	09020	00000001	00100011	01000101	01010111	19088727(D)			
009003	009002	009001	009000																																						
11111111	11111111	11111111	11110000																																						
- 16(D)																																									
009013	009012	009011	009010																																						
00000001	00100011	01000101	01100111																																						
19088743(D)																																									
09023	09022	09021	09020																																						
00000001	00100011	01000101	01010111																																						
19088727(D)																																									
演算後の内容	S1～S1+3	不変																																							
	S2～S2+3	不変																																							
	D～D+3	演算結果(符号付バイナリ31ビット)																																							
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354																																			
		±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																																					

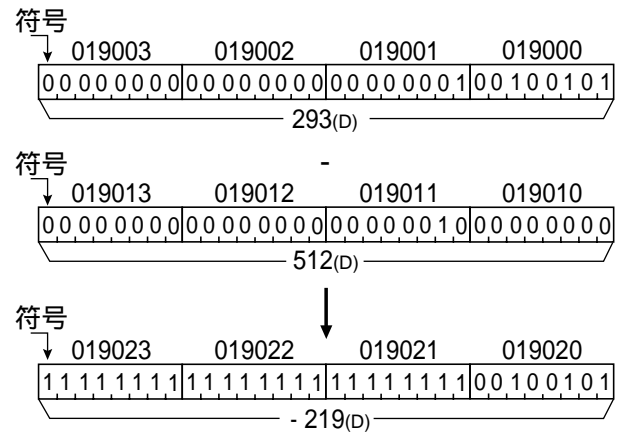
S1、S2、Dには必ず偶数アドレスを設定してください。

・扱える数値の範囲は、 - 2147483648 ~ 2147483647(D)です。 符号付き演算(9・25ページ)参照

F-311 SSUB レジスタ間の符号付きバイナリ減算(31ビット - 31ビット)
(Signed SUB tract)

シンボル	<table border="1"> <tr> <td>F-311 SSUB</td> <td>S1</td> <td>S2</td> <td>D</td> </tr> </table>				F-311 SSUB	S1	S2	D	<p>[使用例]</p> <table border="1"> <tr> <th colspan="2">命 令</th> </tr> <tr> <td>STR</td> <td>005001</td> </tr> <tr> <td>F-311</td> <td>019000</td> </tr> <tr> <td></td> <td>019010</td> </tr> <tr> <td></td> <td>019020</td> </tr> </table>	命 令		STR	005001	F-311	019000		019010		019020
F-311 SSUB	S1	S2	D																
命 令																			
STR	005001																		
F-311	019000																		
	019010																		
	019020																		
機能	レジスタS1 ~ S1+3の内容とS2 ~ S2+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	$(S1 \sim S1+3) - (S2 \sim S2+3) \rightarrow (D \sim D+3)$																		
S1	使用範囲 C																		
S2	使用範囲 C																		
D	使用範囲 C																		
演算条件	入力信号の立上り(OFF ON)																		
演算後の内容	S1 ~ S1+3	不変																	
	S2 ~ S2+3	不変																	
	D ~ D+3	演算結果 (符号付バイナリ31ビット)																	
	フラグ	演算結果	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354													
		±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005001がOFF ONの変化時に、レジスタ019000 ~ 019003の内容とレジスタ019010 ~ 019013の内容を符号付バイナリ減算して、その結果をレジスタ019020 ~ 019023に格納します。
下記演算は、293 - 512 = - 219を示します。



S1、S2、Dには必ず偶数アドレスを設定してください。

・扱える数値の範囲は、- 2147483648 ~ 2147483647(D)です。 符号付き演算(9・25ページ)参照

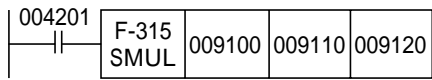
**F-315
SMUL**

レジスタ間の符号付きバイナリ乗算(31ビット×31ビット)
(Signed MUL tiply)

シンボル	— F-315 SMUL S1 S2 D				
機能	レジスタS1～S1+3の内容とS2～S2+3の内容を、符号付31ビット数として乗算し、D～D+7に格納する。				
演算内容	$(S1 \sim S1+3) \times (S2 \sim S2+3) \rightarrow (D \sim D+7)$				
S1	使用範囲 C				
S2	使用範囲 C				
D	使用範囲 G				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S1～S1+3	不変			
	S2～S2+3	不変			
	D～D+7	演算結果(符号付63ビット)			
	フラグ	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
		0	0	0	0

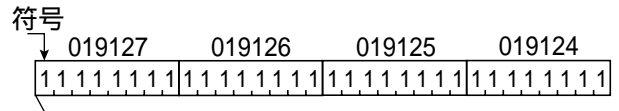
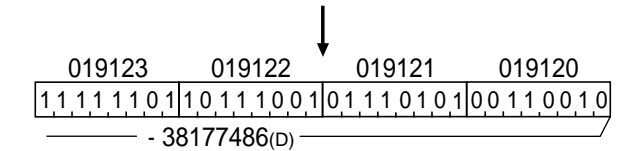
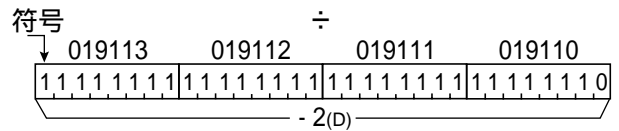
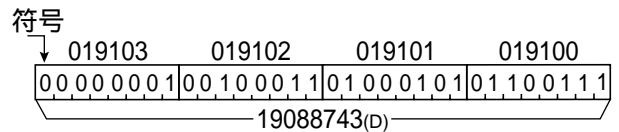
S1, S2, Dには必ず偶数アドレスを設定してください。
 ・扱える数値の範囲は、-2147483648～2147483647(D)です。 符号付き演算(9・25ページ)参照

[使用例]



命 令	
STR	004201
F-315	009100
	009110
	009120

入力条件004201がOFF ONの変化時に、レジスタ009100～009103の内容とレジスタ009110～009113の内容を符号付バイナリ乗算して、その結果をレジスタ009120～009127に格納します。
 下記演算は、 $19088743 \times (-2) = -38177486$ を示します。



**F-316
SDIV**

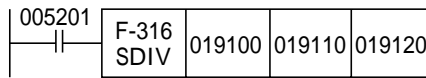
レジスタ間の符号付きバイナリ除算(31ビット÷31ビット)
(Signed DIV ide)

シンボル	F-316 SDIV S1 S2 D				
機能	レジスタS1～S1+3の内容をレジスタS2～S2+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、D+4～D+7に余を格納する。				
演算内容	(S1 ~ S1+3) ÷ (S2 ~ S2+3) → (D ~ D+7)				
S1	使用範囲 C				
S2	使用範囲 C				
D	使用範囲 G				
演算条件	入力信号の立上り(OFF ON)				
演算後の内容	S1～S1+3	不変			
	S2～S2+3	不変			
	D～D+3	演算結果の商 (符号付31ビット)			
	D+4～D+7	演算結果の余 (符号付31ビット)			
フラグ	レジスタS2～S2+3の内容	ゼロ 007357	キャリー 007356	エラー 007355	ノンキャリー 007354
	0	0	0	1	0
	上記以外	0	0	0	0

S1、S2、Dには必ず偶数アドレスを設定してください。(019003等は禁止)

・扱える数値の範囲は、-2147483648～2147483647(D)です。 符号付き演算(9・25ページ)参照

[使用例]

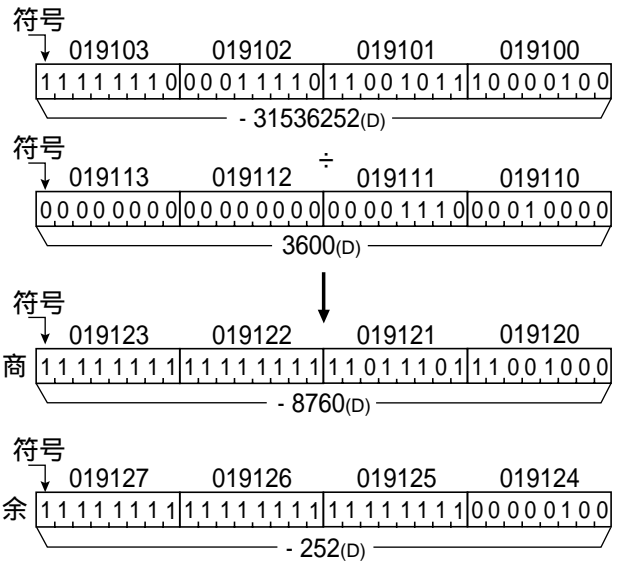


命 令	
STR	005201
F-316	019100
	019110
	019120

入力条件005201がOFF ONの変化時に、レジスタ019100～019103の内容をレジスタ019110～019113の内容で符号付バイナリ除算して、その結果をレジスタ019120～019127に格納します。

例 1

下記演算は、 $-31536252 \div 3600 = -8760$ 余り -252 を示します。



例 2 $31536252 \div (-3600) = -8760$ 余り 252

例 3 $-31536252 \div (-3600) = 8760$ 余り -252

**F-403
LOG**

ロギング命令

F-403命令は、ロギング機能で使用します。
ロギング機能(6・7ページ)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	2003年7月	—————
改訂1.1版	2004年2月	<ul style="list-style-type: none"> ・CFカードに関し、特殊リレー007332～007334の説明を追記 ・故障診断に「入力リレーのフィルタ機能」を追記
改訂1.2版	2005年2月	<ul style="list-style-type: none"> ・レジスタ領域の予約領域、倍長演算機能、符号付き演算機能、データメモリのブロックと基準アドレスの説明を追記 ・システムメモリ#0222、#0247の説明を変更、#0440を追記 ・オプションバージョンエラーを追記 ・CFカードの留意内容に説明を追記 ・TMR、CNTの設定範囲に説明を追記 ・間接アドレス「fileN、アドレスn」の説明を改善 ・F-202、F-203、F-206、F-207の説明を改善
改訂1.3版	2010年8月	<ul style="list-style-type: none"> ・セキュリティ機能、編集履歴機能、モード変更履歴機能を追記 6・25～35ページ ・シンボル・コメント専用メモリを追記 3・5ページ ・システムメモリ#0041、#0042、#0263、#0450、#0451、#2260～#2264、#2237の説明を追記 第4章 ・CPU異常に「ウォッチドグタイマ2」を追記 5・3ページ ・特殊リレー007300を追記 2・6、6・20ページ ・CFカードへのセーブ時間の向上を反映 6・22ページ ・「ブロック番号とデータメモリ(先頭アドレス)」を追記 14・4～5ページ

商品に関するお問い合わせ先 / ユーザーズマニュアルの依頼先

シャープマニファクチャリングシステム(株)

東日本営業部 〒162-8408 東京都新宿区市谷八幡町8番地 ☎(03)3267-0466
 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
 西日本営業部 〒581-8581 大阪府八尾市跡部本町4丁目1番33号 ☎(072)991-0682
 西日本営業部 〒812-0881 福岡市博多区井相田2丁目12番1号 ☎(092)582-6861
 (福岡駐在)

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 前橋市問屋町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)3810-9963
 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 〒424-0067 静岡県静岡市清水鳥坂1170 ☎(0543)44-5621
 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 〒812-0881 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

TINSJ5397NCZZ
 10H 0.1 A
 2010年8月作成