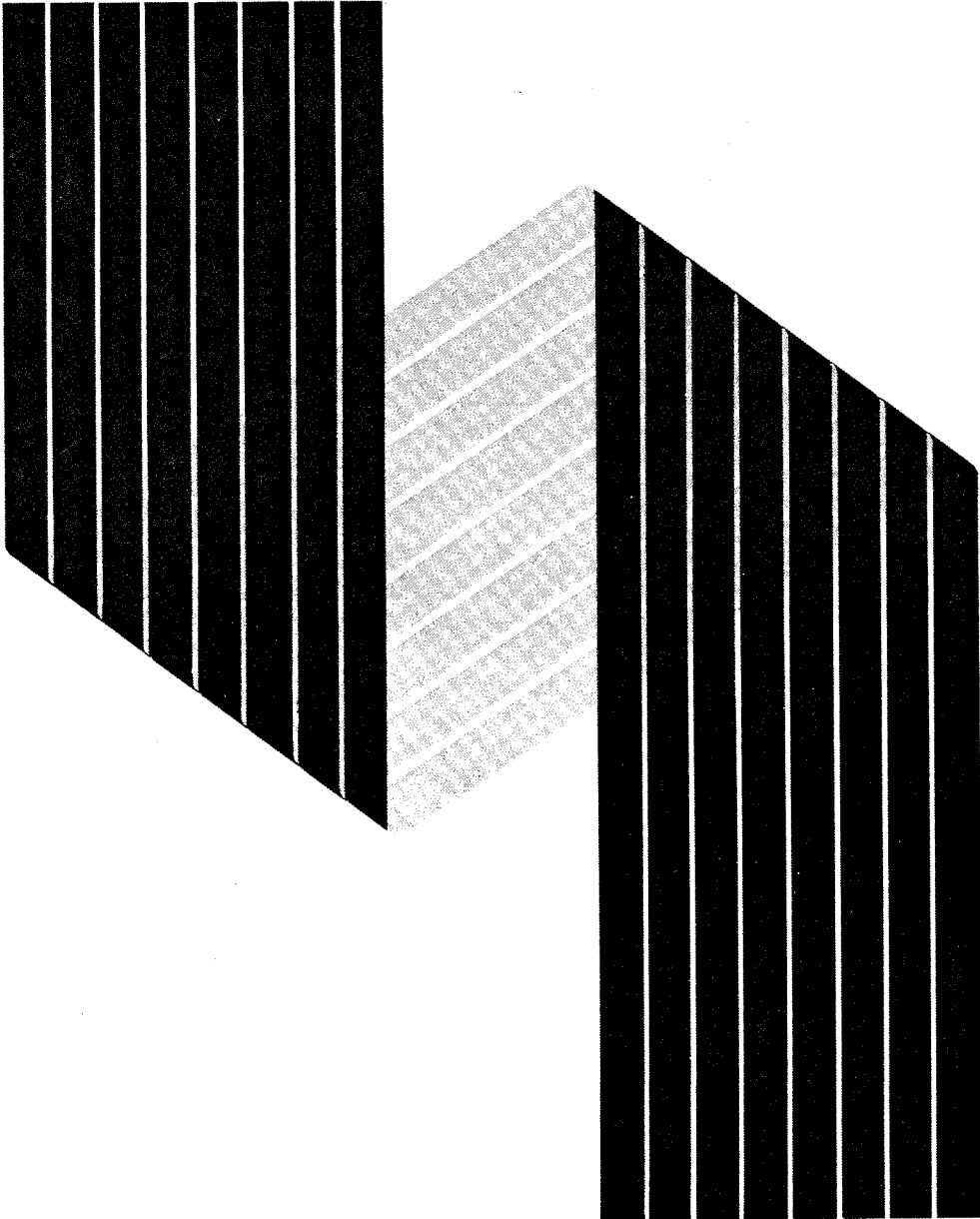




プログラミング編

ラダー図



HITACHI

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。
なお、不明な場合は、弊社担当営業にお問合わせください。

1987年 1月	(第1版)	SP-3-001	(廃版)
1987年12月	(第2版)	SP-3-101	(廃版)
1991年 9月	(第3版)	SP-3-201	

- このマニュアルの一部、又は全部を無断で転写したり複製することは、固くお断りいたします。
- このマニュアルの内容を、改良のため予告なしに変更することがあります。

はじめに

本マニュアルはラダープログラムを作成する時の命令語について説明したものです。

命令語には大きく分けてラダー命令と演算ファンクション命令があります。

ラダー命令はリレー回路の動作を行うものです。

演算ファンクションは加減乗除等の演算処理を行うものです。

目 次

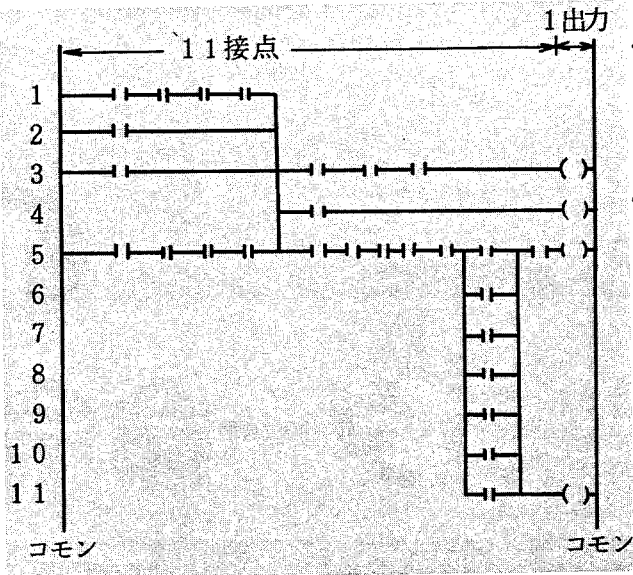
1	ラダー命令	1
1. 1	ラダー回路の大きさ	2
1. 2	右下り回路と動作順序	3
1. 3	ラダー回路とステップ	5
1. 4	停復電時とSTOPからRUN時の状態	5
1. 5	ラダー命令一覧	6
2	演算ファンクション	25
2. 1	機能概要	26
2. 2	機能仕様	28
2. 3	パラメータ入力	30
2. 4	演算ファンクション一覧表, 入力	31
3	処理時間	151
3. 1	2 α の処理時間	152
3. 2	4 α の処理時間	157
3. 2. 1	スキャンタイム	157
3. 2. 2	スキャンタイム計算例	157
3. 3	平均スキャンタイムの表示	159

1

ラダー命令

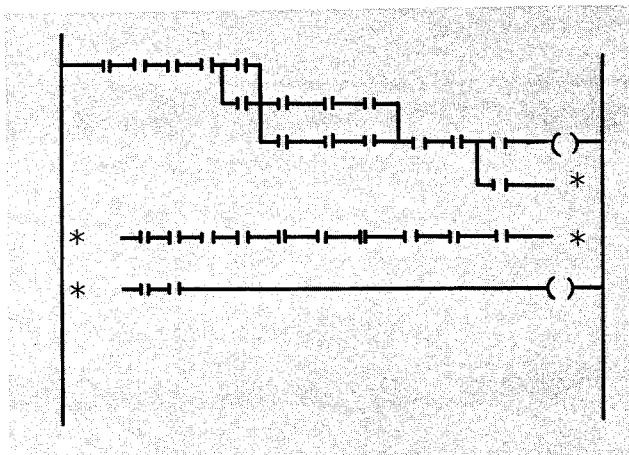
1 ラダー命令

1.1 ラダー回路の大きさ



- ラダー回路は左側のコモン線から右側のコモン線に接続するコイルで終了します。
この1つにまとまったラダー回路を1ブロックと呼びます。
- 最大回路は次となります。
横 11接点+1出力コイル
縦 11行

● 11接点以上をAND接続する時



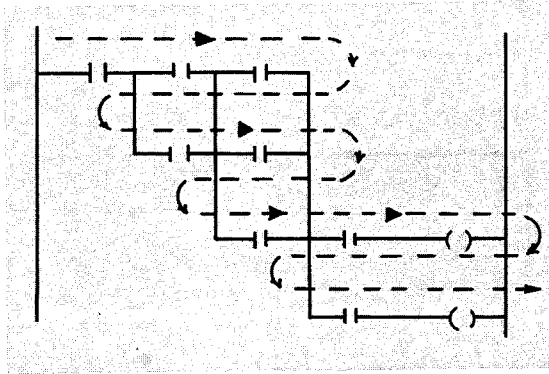
- AND接続が多い場合には図の様に“折り返し回路”となりAND接続11接点以上の回路が作成できます。

【制限】

- *印の前に分岐が残ってはいけません。
- *以後は分岐回路の作成はできません。

1.2 右下り回路と動作順序

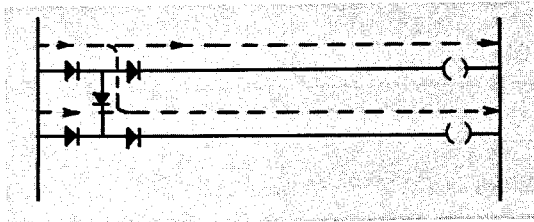
【右下り回路の例】



- ラダー回路は左から右、上から下への回路構成となります。

【動作順序の考え方】

ラダー回路においてダイオードを入れたイメージ



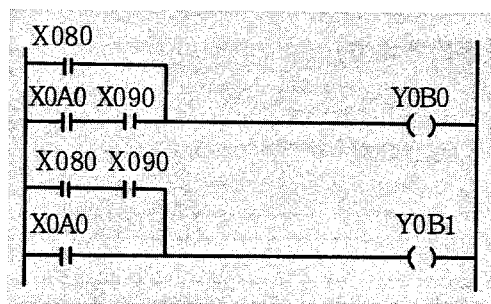
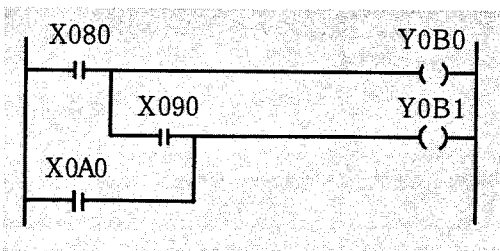
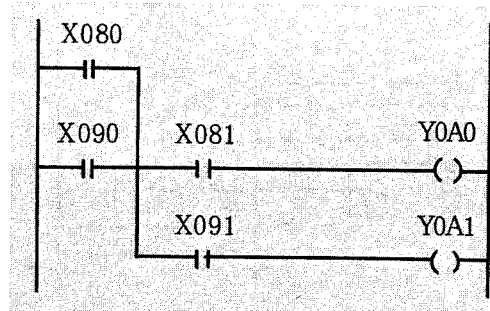
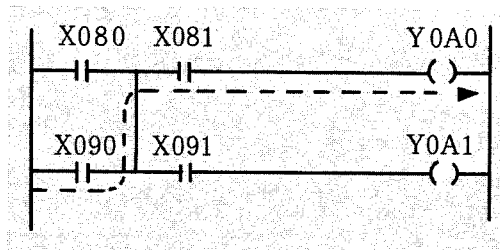
- ラダー回路の動作は、左から右、上から下への順に動作します。考え方としては、接点と下へ下がる分岐に、ダイオードを入れたイメージです。

1 ラダー命令

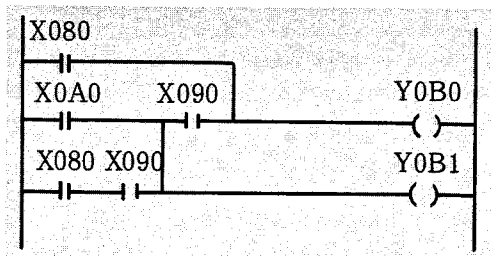
〔右下りラダー回路の動作〕

下図の左欄のリレー回路例の動作を行いたい時は、右欄の右下りラダー回路を作成してください。

〔リレー回路の例〕



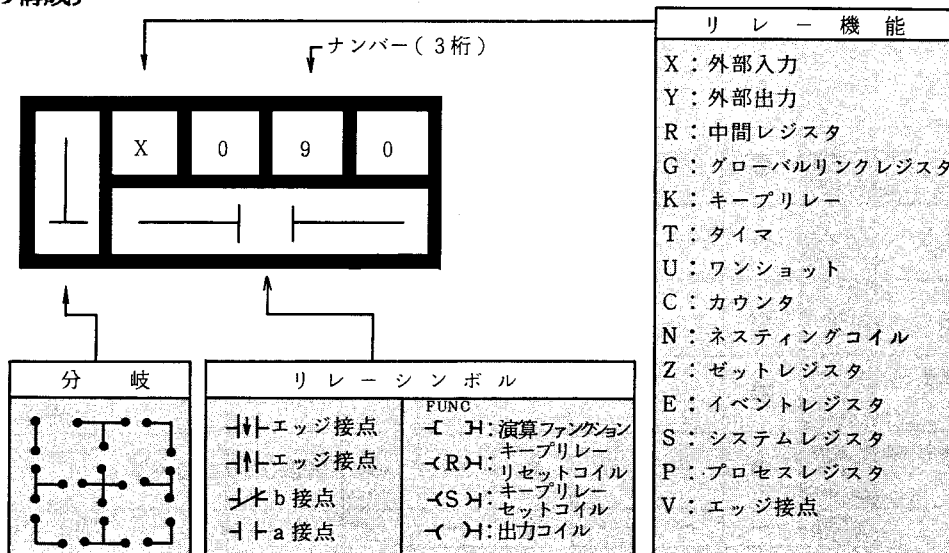
又は



1.3 ラダー回路とステップ

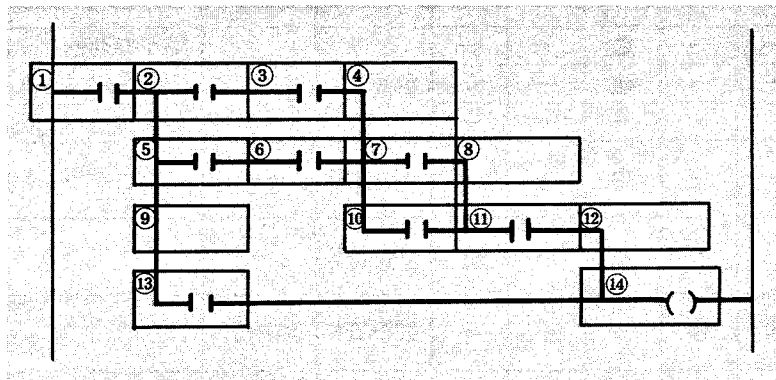
ラダー回路1ステップは次の構成になります。

〔1ステップの構成〕



● 分岐の \neg , \neg , \neg は1ステップとなります。

〔ラダー回路と命令語〕



□ : 1ステップ

・左図では14ステップ
となります。
またプログラムは○で示
した順番で格納, 実行さ
れます。

1.4 停復電時と、STOPからRUN時の状態

リレー機能	ON/OFF状態		計数值/数値	
	停復電時	STOP→RUN時	停復電時	STOP→RUN時
T,U	OFF	ホールド	0クリア	ホールド
C	ホールド	ホールド	ホールド	ホールド
K	ホールド	ホールド		
X,Y,R,G,N,Z,E,P,V	OFF	ホールド		
FW,DW			ホールド	ホールド

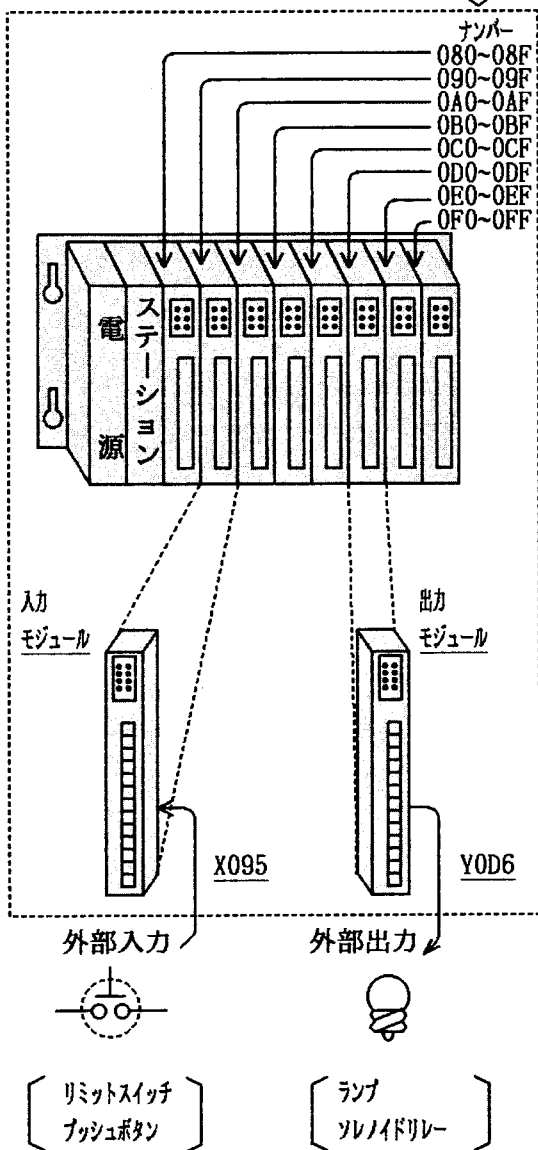
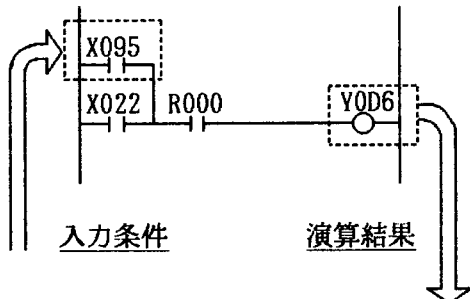
1.5 ラダー命令一覧

	名称	ソリ	↑	↓	↑↓	○	(S)	(R)	ナンバ		備考	
									2α	4α		
入出力	外部入力	X	●	-	-	-	-	-	000~7FF	000~1FF		
	外部出力	Y	●	-	-	●	-	-	000~7FF	000~1FF		
内	内部レジスタ	R	●	-	-	●	-	-	000~7FF	000~3FF		
	キープリレー	K	●	-	-	-	●	●	000~1FF	000~0FF		
	オンディレイ タイマ	T	●	-	-	●	-	-	000~1FF	000~0FF	000~00F(10msタイマに 変更可。)	
	ワンショット タイマ	U	●	-	-	●	-	-	000~07F	000~03F		
部	アップ・ダウン カウンタ	CU	-	-	-	●	-	-	00~3F	00~3F	CU: アップカウンタ CD: ダウンカウンタ CR: リセットコイル CO: 接点	
		CD	-	-	-	●	-	-				
		CR	-	-	-	●	-	-				
		CO	●	-	-	-	-	-				
補	グローバル リンクレジスタ	G	●	-	-	●	-	-	000~FFF	000~FFF		
助	ネスティング コイル	NM	-	-	-	●	-	-	01~FF	—	NM: マスタリセット) ナンバ入力 NZ: ソンコントロール) ごと選択 NO: 接点 (起動状態)	
		NZ	-	-	-	●	-	-				
		NO	●	-	-	-	-	-				
機 能	プロセス レジスタ	P	●	-	-	●	-	-	001~080	—	Cプログラム起動用	
	イベント レジスタ	E	●	-	-	●	-	-	000~0FF	—		
	エッジ接点	V	-	●	●	-	-	-	000~7FF	000~3FF		
	ゼットレジスタ	Z	●	-	-	●	-	-	000~01F 0FE, 0FF 200	000~00F 0FE, 0FF 200	0FE; トレーススタート信号 0FF; トレースストップ信号 200; H-7338方式 節起動信号	
	システム レジスタ	S	●	-	-	-	-	-	-	000~005	演算ファンクションフラグ	
										010~01F	ラダープログラム編成レジスタ	
										100~15F	ラダープログラム編成カウンタ	
300~47F										リモートI/O状態表示レジスタ		
500~6FF										580~5BF 6C0~6FF	オプションモジュール状態表示レジスタ	
BA0~BBF										I/O設定パターン		
BFO~BFF	CPUステータスレジスタ											
ファンクション データレジスタ	DW	-	-	-	-	-	-	000~FFF	000~7FF	定数データエリア		
ファンクション ワークレジスタ	FW	-	-	-	-	-	-	000~BFF	000~BFF	ワークエリア		

●: 使用できるシンボル

X, Y 外部入出力

	2α	4α
ナンバーの範囲	000~7FF	000~1FF



PCsに接続された外部入出力モジュール経由で信号の入出力を行います。

X…入力モジュール経由で外部の入力信号を取り込みます。

Y…ラダープログラムの演算結果を出力モジュールから外部へ出力します。

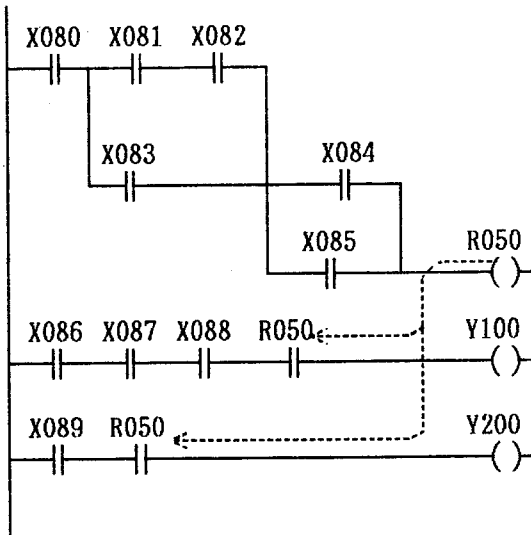
R 内部レジスタ

	2α	4α
ナンバーの範囲	000~7FF	000~3FF

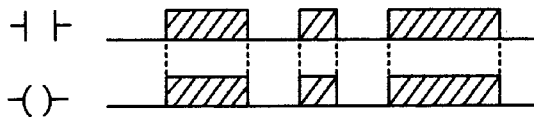
ラダー命令の演算結果を中継するための内部レジスタです。

コイル (-()) が ON と同時に接点 (| |) が ON し、コイル (-()) が OFF すると同時に接点 (| |) も OFF します。

回路例



タイムチャート

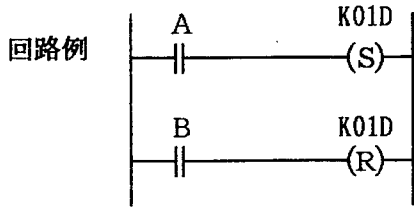


K キープリレー

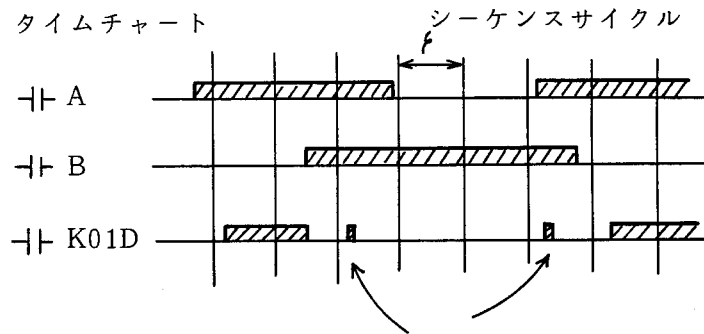
	2α	4α
ナンバの範囲	000~1FF	000~0FF
セットリングパルス幅	最小1スキャンタイム	
セット,リセット同時入力時	プログラムの下方にある方が優先されます。	

キープリレーは、セットコイル (-S-) がONすると、リセットコイル (-R-) がONするまで接点 (ノーマル開) がONし、接点の状態は停電保持されます。また、セットコイルとリセットコイルが同時にONした場合、プログラムの下方にある方が優先されます。

リセット優先回路

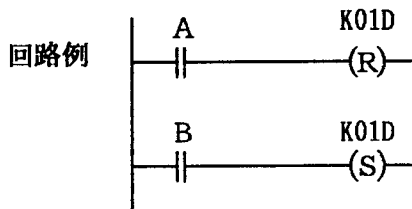


タイムチャート

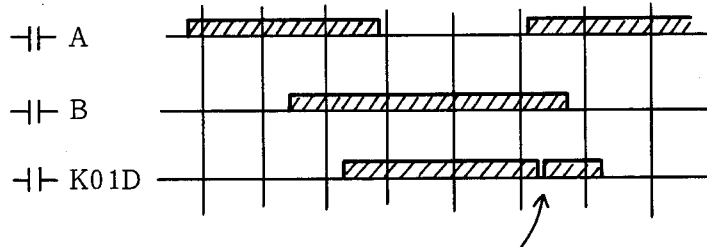


- -S- が実行され -R- までの間、ONする。

セット優先回路



タイムチャート



- -R- が実行され -S- が実行されるまでの間、OFFとなる。

オンディレイタイマの仕様

100msタイマ	2α	4α
ナンバの範囲	000~1FF	000~0FF
設定値	0.1~999.9秒	
誤差	最小100ms+1スキャンタイム	
セットリングパルス幅	最小100ms	

オンディレイタイマのコイル (-()-) がONしてから、設定値の時間だけ遅れて、接点 (-(|)) がONします。

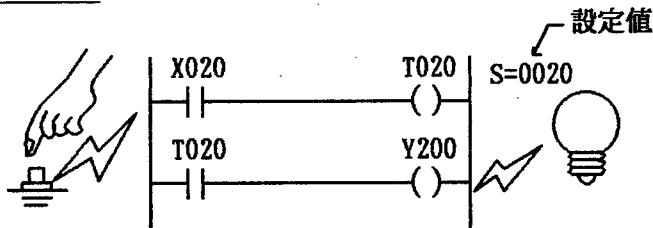
設定値の範囲は0.1~999.9秒で0.1秒単位に設定することができます。

なお、先頭から16点 (T000~T00F) はエディション設定により10msタイマとして使用することができます。

10msタイマ(エディション設定による。)

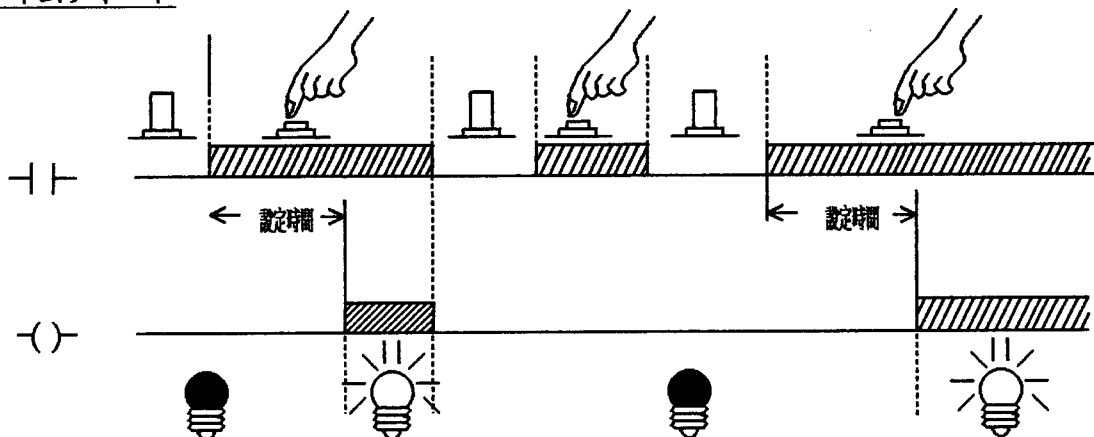
	2α	4α
ナンバの範囲	000~00F	
設定値	0.01~99.99秒	
誤差	最小10ms+1スキャンタイム	
セットリングパルス幅	最小10ms	

(a) 回路例



図に示した回路の場合押しボタン (X020) を押してから、設定時間後 (この場合2秒後) ランプ (Y200) が点灯し、手を離すと同時にランプが消灯します。

(b) タイムチャート



- ・ コイル (-()-) の時間幅が設定時間により短いと、接点 (-(|)) はONしません。

☆ 2αの計数値は、65535までカウントします。オーバーフロー時は、0からカウントを始めます。

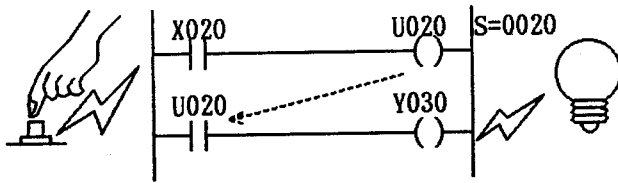
ワンショット

	2α	4α
ナンバの範囲	000~07F	000~03F
設定値	0.1~999.9秒	
誤差	最小100ms+1スキャンタイム	
セットリングパルス幅	最小100ms	

ワンショットのコイル (-(-)-) がONしてから設定値の時間だけ接点 (-(+)-) がONします。

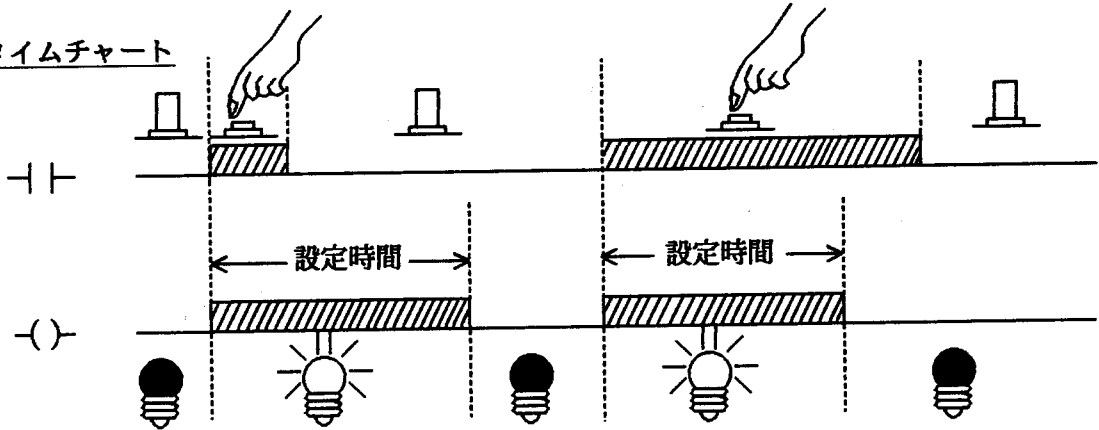
設定値の範囲は0.1~999.9秒で、0.1秒単位に設定することができます。

(a) 回路例



- 図に示した回路の場合押しボタンを押すと、ランプ (Y030) が設定時間 (この場合2秒) だけ点灯します。

(b) タイムチャート



☆ 2αの計算値は、65535までカウントします。
オーバーフロー時は、0からカウントを始めます。

C アップダウンカウンタ

アップ・ダウンカウンタの仕様

		2α		4α	
ナンバの 範囲	-()	CU	00	00	
		CD	~3F	~3F	
		CR			
		CO			
設定値	1~9999カウント				
セットリング パルス幅	最小1スキャンタイム				
セット、リセット 同時入力時	リセット優先				
停電時	不揮発				

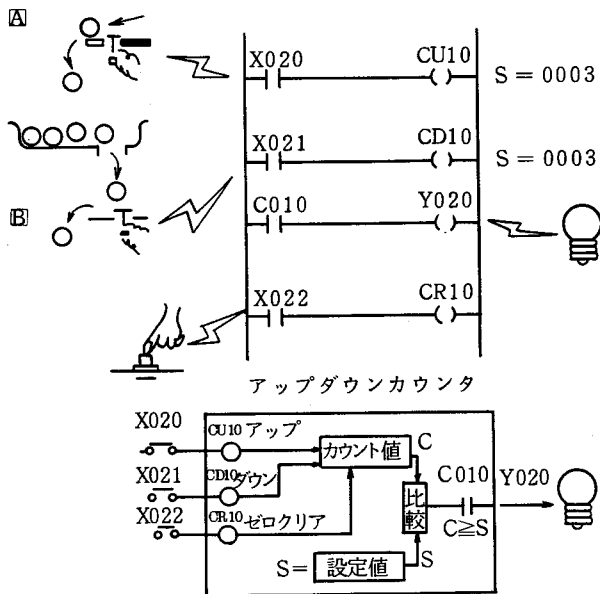
アップカウンタ (CUΔΔ) とダウンカウンタ (CDΔΔ) から構成され、カウント値はアップカウンタのコイル (-) 立上り時カウントアップ (+1), ダウンカウンタのコイル立上り時にカウントダウン (-1) します。

カウンタ接点 (| |) はカウント値が設定値を越えた時にONします。

リセットコイルはカウント値をゼロクリアし、接点がOFFします。

- CU: アップカウンタ
- CD: ダウンカウンタ
- CR: リセット
- CO: カウンタ接点

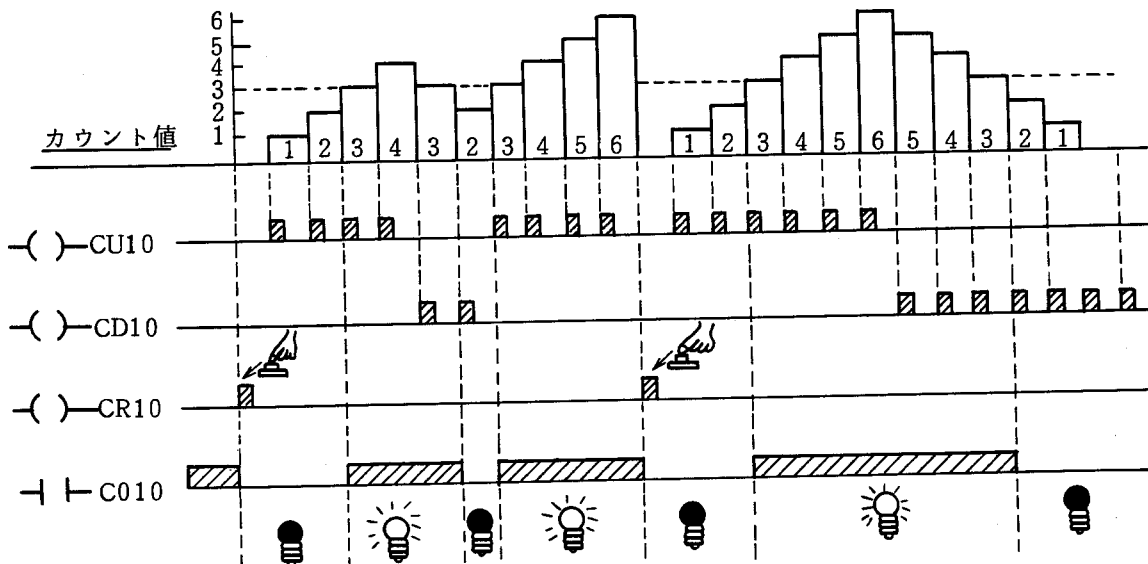
(a) 回路例



- 図の回路は、上方からバスケットに入ったボールの数をスイッチA (X020) でカウントし、下方へ落ちたボールの数をスイッチB (X021) でカウントし、現在バスケットに入っているボールの数を計数します。
- バスケットに入っているボールの数が3個以上になるとランプ (Y020) が点灯します。

また、プッシュボタン (X022) を押すとカウント値がゼロクリアされ、ランプが消灯します。

(b) タイムチャート



G グローバル リンクレジスタ

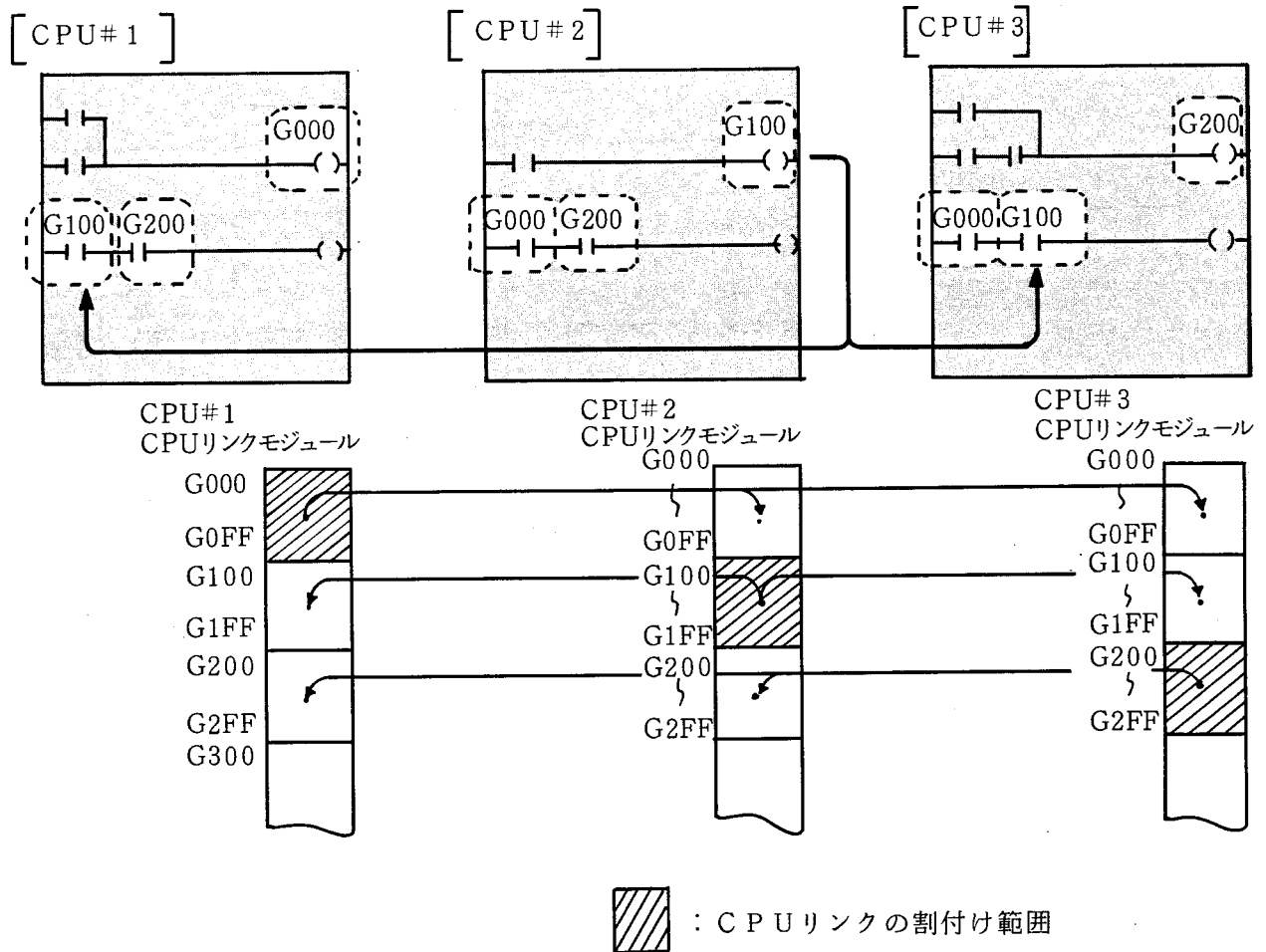
	2α	4α
ナンバの範囲	000~FFF	000~7FF

CPU間リンクモジュール (オプション) 実装時に CPU間のインタロック情報等を交換するためのレジスタです。

動作はコイル (-(-)-) のON (またはOFF) で、他CPUの同一ナンバの接点 (-|+) がON (またはOFF) します。

- ・ 【CPU間リンクの割付は、CPU間リンクマニュアルを参照ください。】

- ・ 回路例 (CPU間リンクカード使用時)



N ネスティングコイル

	2α	4α
ナンバの範囲	001~OFF	——
ネスティング Max. Level	4レベル	——

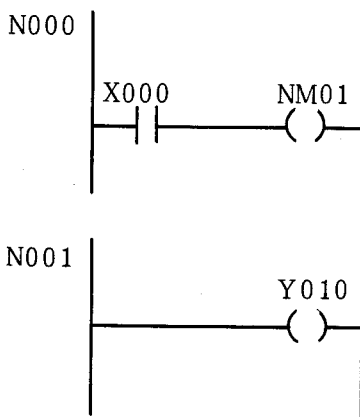
シーケンスプログラムを制御対象プラント毎に分割し、モジュール化するための機能を持ちます。

ネスティングコイルにはNコイルの立下り (ON→OFF) 時に使用されているコイルをOFFする“マスタコントロール”のものと、前回の状態を保持する“ゾーンコントロール”の2種より選択します。

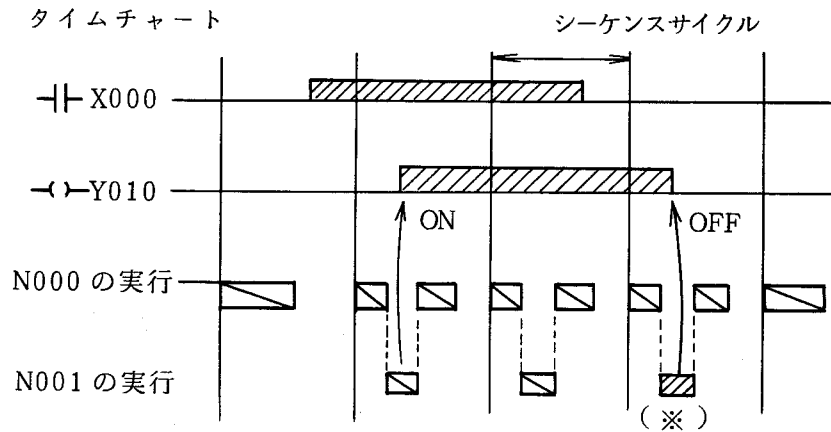
また、Nは最大4レベルまでのネスティングが可能です。

マスタコントロール (NM)

回路例



タイムチャート

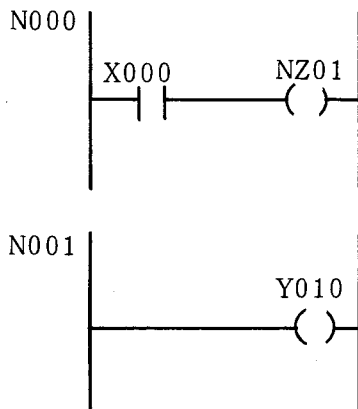


※ NM01の立下り (ON→OFF) の時、NM01で使用されているコイル (-(-)-) をすべてOFFします。

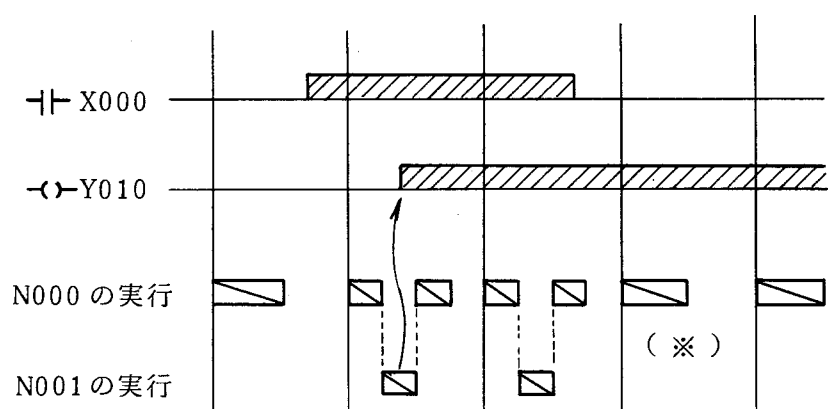
- ・ K (キープリレー) 及び \uparrow , \downarrow エッジ接点を使用した回路の場合、OFFされない場合があります。

ゾーンコントロール (NZ)

回路例




タイムチャート



※ NZ01が立下がってもマスタリセット付と異なり、N001で使用されているコイルは前回の状態を保持しています。

P プロセスレジスタ

	2α	4α
ナンバの範囲	001~080	—
起動方法	レベル起動 	—

ユーザが、C言語やアセンブラ言語で作成したコンピュータ言語によるプログラム（以下タスクと呼ぶ）をラダープログラムにより起動するためのレジスタです。

本コイルをONすることによりナンバに対応したタスクが起動（Queue）されます。

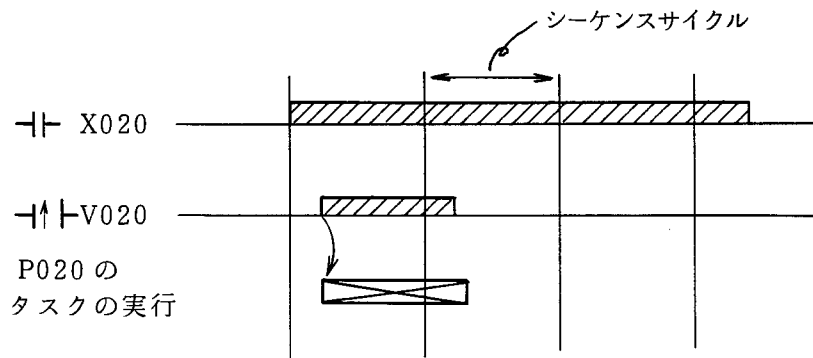
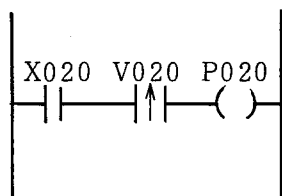
プロセスレジスタの割付け

区分	ナンバ	名称	内容
ユーザ作成	P001	イニシャルタスク	CPUをGR（リセット）した時に必ず起動されるタスクです。システムのイニシャルを行うプログラムを割付けてください。
	P002	ユーザ	ユーザが作成したプログラムを割付けてください。
	P07F	タスク	
システム専用	P080	システムタスク	CPMS等でデバッガサポート用のタスク等を割付けます。ユーザの使用は避けてください。

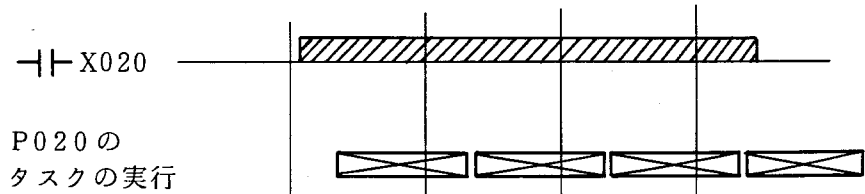
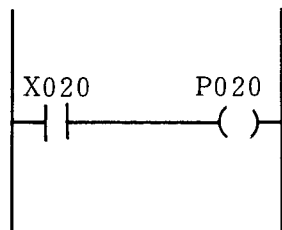
回路例

タイムチャート

■ON時、1回実行



■ON時、毎回実行

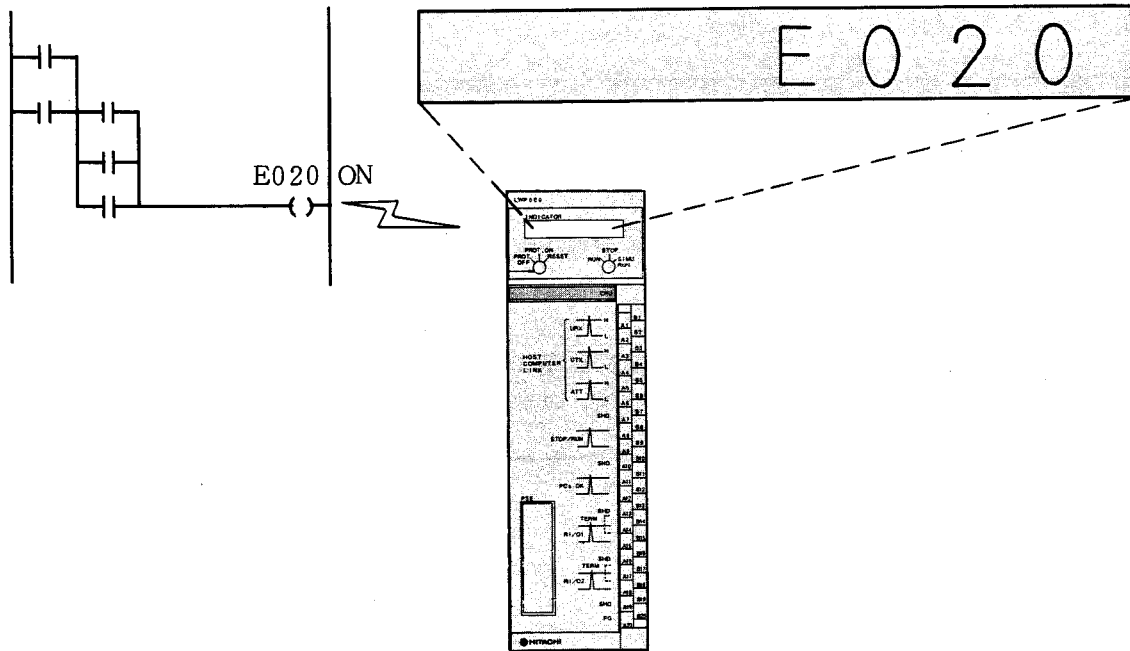


E イベントレジスタ

	2 α	4 α
ナンバー 範囲	000~OFF	—

CPUの8桁のLEDにユーザーのエラー等のイベント情報を表示するためのレジスタです。
本レジスタのコイルがONしている場合、そのナンバーがLEDに表示されます。

回路例



V エッジ接点

エッジ接点の仕様

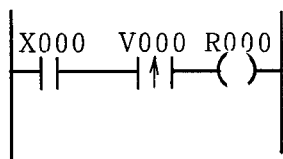
	2α	4α
ナンバの範囲	000~7FF	000~3FF

エッジ接点には立上りエッジ検出接点 (↑) と立下りエッジ検出接点 (↓) があり、それぞれのエッジを検出した1スキャンだけONします。

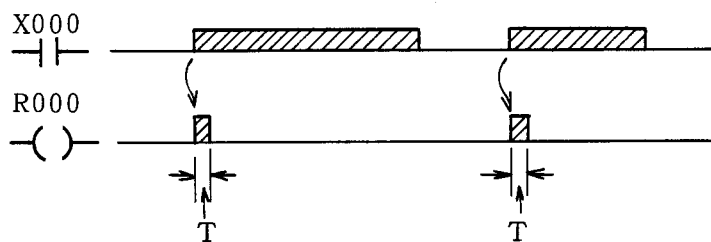
なお、同一ナンバを立上りエッジ接点及び立下りエッジ接点として使用することはできません。

■ 立上りエッジ接点

回路例



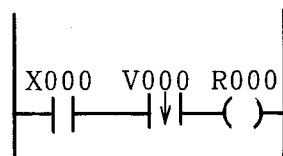
タイムチャート



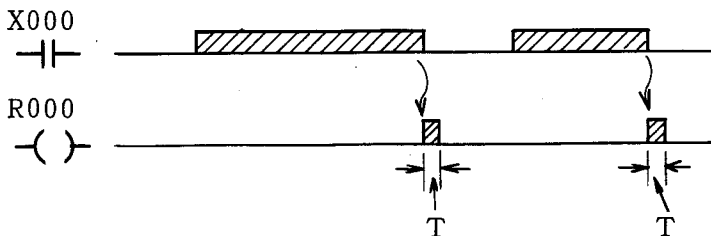
T : 1 スキャンタイム

■ 立下りエッジ接点

回路例



タイムチャート



T : 1 スキャンタイム

Z セットレジスタ

ゼットレジスタの割付け

	2α	4α	内 容
ナン バの 範囲	000 ~01F	000 ~00F	シーケンスサイクル単位 でトレースするI/O情報を セットする。
	0FE		ロジックトレース機能の スタート信号です。
	0FF		ロジックトレース機能の ストップ信号です。

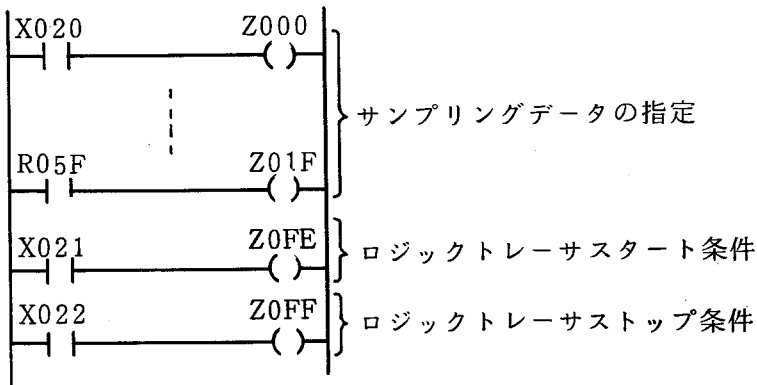
Z200	上位計算機へ、H-7338方 式の割込を発生します。
------	-------------------------------

ゼットレジスタには、ロジックトレース機能及びH-7338方式による上位割込用のレジスタが配置され、それぞれ左の表の様になります。

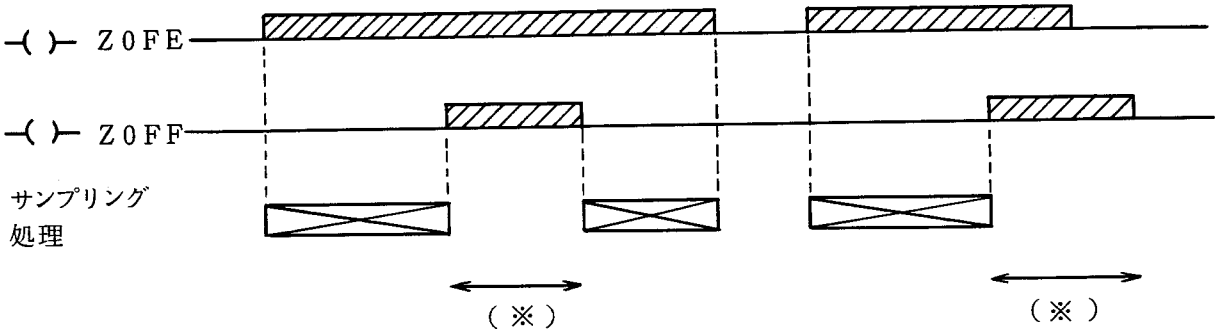
ロジックトレースの詳細については“PSEαオペレーションマニュアル”を参照してください。

ロジックトレースとZコイル

回路例



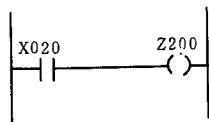
タイムチャート



上位割込レジスタ

■ 2αの場合

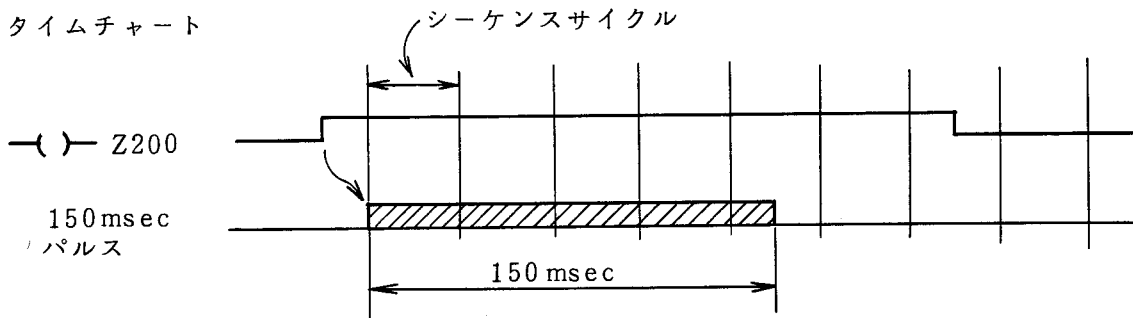
回路例



上位への割込Z200の立上り検出時に150msecの割込を
 発します。尚本処理はシーケンスサイクルと同期して行
 われます。150msecの割込を送信中に、再度Z200の立上
 りを検出した場合は再度150ms分のパルスを出力します。

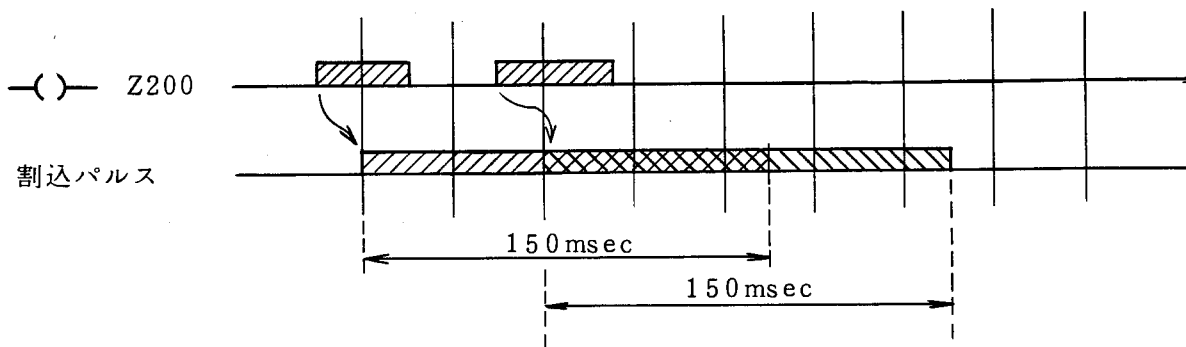
セッティングパルス幅：最小1シーケンスサイクル

タイムチャート



[割込パルスがのびる場合]

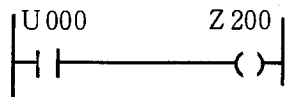
- 上記回路で、Z200のON⇔OFFが150msec以内で2回以上ON→OFFした場合、2重割込が発生しパルス幅が長くなります。



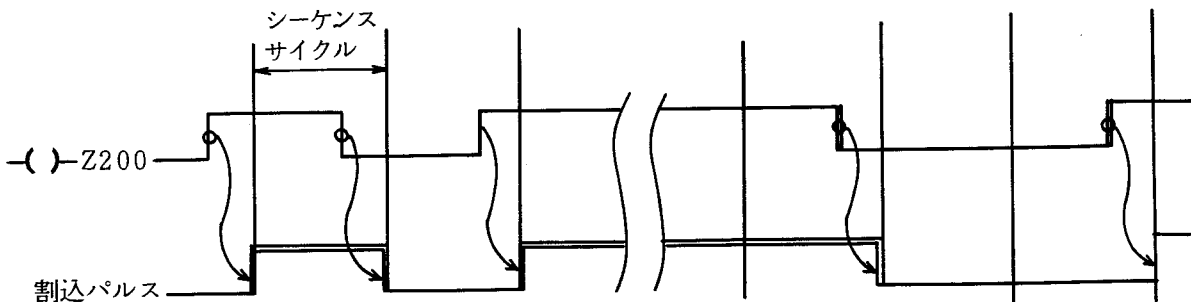
この様な事を防ぐには、割込を受け付ける上位計算機との間でインタロックを取ってください。

■ 4αの場合

回路例



- 上位への割込Z200の立上り検出時より、立下り検
 出時までの時間、割込信号を出力します。尚本処理はシ
 ーケンスサイクルと同期して行われます。
 (通常ワンショットの出力を使用します。)



S システムレジスタ

システムの動作状態等を反映した読込専用のレジスタです。

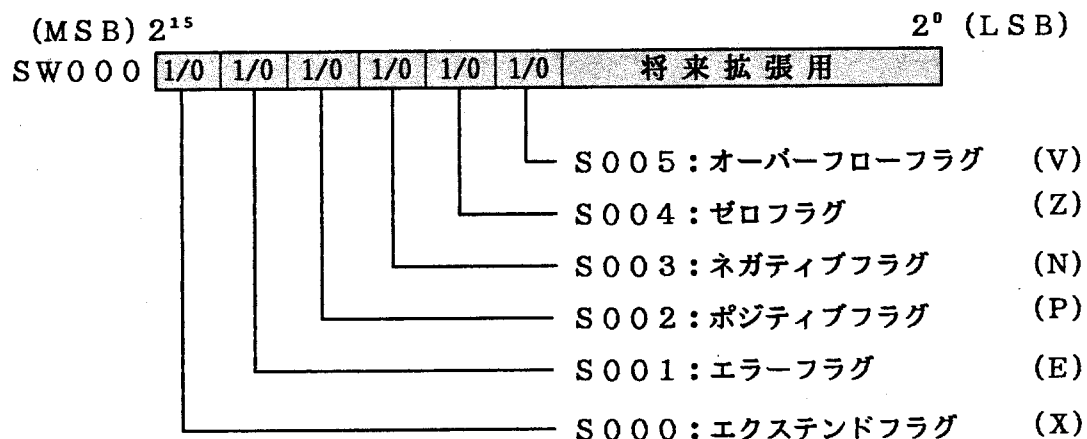
システムレジスタ一覧表

No	レジスタ No	概 要
1	S000~S00F	演算ファンクションフラグレジスタ
2	S010~S01F	ラダープログラム制御レジスタ
3	S100~S15F	ラダープログラム制御カウンタ
4	S300~S47F	リモートI/O状態レジスタ
5	S500~S6FF	オプションモジュール状態レジスタ
6	SBA0~SBBF	I/O固定パターン
7	SBF0~SBFF	CPUステータスレジスタ

- 以上のレジスタ以外は将来拡張用となっています。

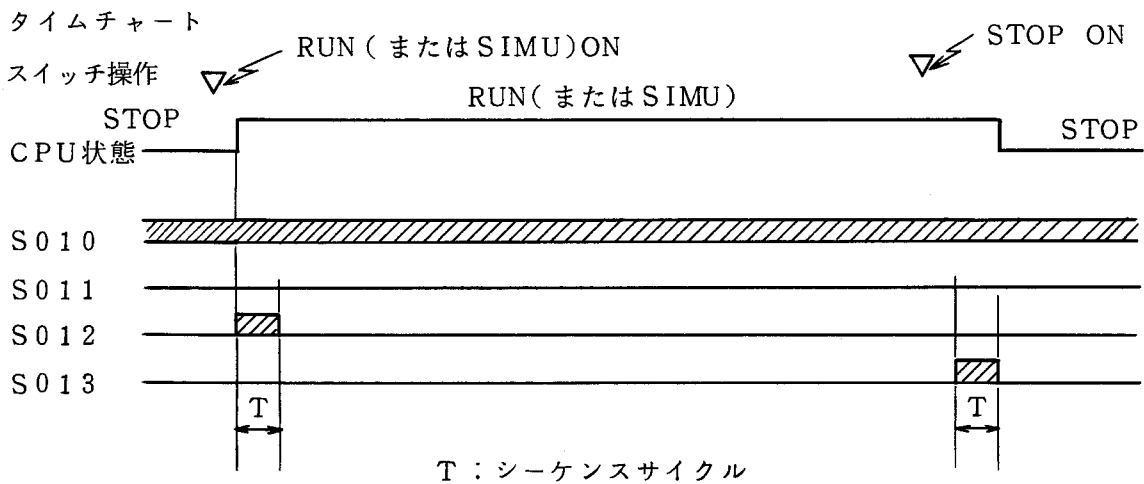
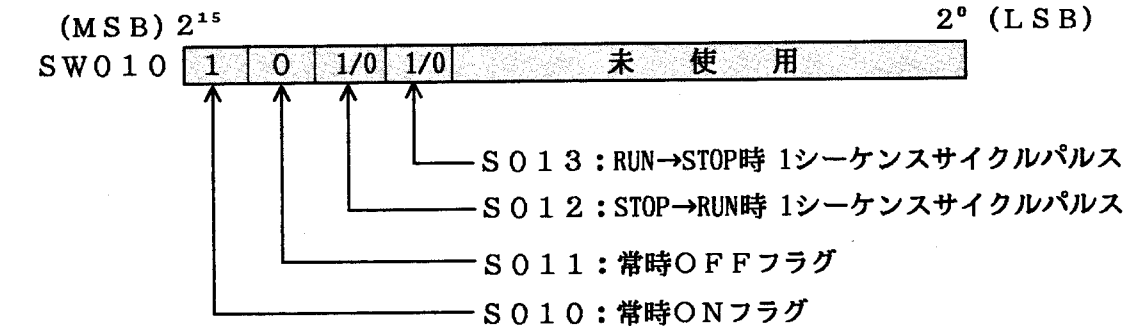
(1) 演算ファンクションフラグレジスタ

システム演算命令実行後のフラグの状態を示したレジスタです。各番号は次の様に対応します。



(2) ラダープログラム制御レジスタ

ラダープログラムの作成を、より容易にするための情報が格納されるレジスタです。



(3) ラダープログラム制御カウンタ

シーケンス制御で使用するこのできる積算カウンタです。

(MSB) 2^{15}		2^0 (LSB)
SW100	10ms 積算カウンタ	} CPU復電時及びリセット時にゼロクリアされます。
SW110	100ms 積算カウンタ	
SW120	1秒単位積算カウンタ	
SW140	シーケンスサイクル積算カウンタ	...CPUのSTOP→RUN時にゼロクリアされます。
SW150	リモートI/Oサイクル積算カウンタ	...リモートI/OがSTOP→RUNへ変化した時ゼロクリアされます。

- 全カウンタとも、オーバーフロー時は“ゼロ”から再カウントします。
- カウンタの精度はOSの割込みによって処理しているため、±10%程度の誤差が生じます。

S システムレジスタ

(4) リモートI/O状態レジスタ

リモートI/Oの登録状態、タイムアウト及びFUSE断ステーション情報を示すレジスタです。

レジスタの割付け

S300	登録 ステーション	・ 現在登録されているステーションに対応したレジスタに“1”がセットされます。
S380		・ 登録ステーション中タイムアウトエラーを発生しているステーションに対応したレジスタに“1”がセットされます。
S400		・ 登録ステーション中FUSE断を発生しているステーションに対応したレジスタに“1”がセットされます。
S47F	FUSE断 ステーション	

各ステーションとビットの対応

No	XまたはYのナンバ	登録ステーション	タイムアウトステーション	FUSE断ステーション
0	000~00F	S300	S380	S400
1	010~01F	S301	S381	S401
2	020~02F	S302	S382	S402
3	030~03F	S303	S383	S403
4	040~04F	S304	S384	S404
124	7C0~7CF	S37C	S3FC	S47C
125	7D0~7DF	S37D	S3FD	S47D
126	7E0~7EF	S37E	S3FE	S47E
127	7F0~7FF	S37F	S3FF	S47F

●4αのCPUベース内の出力モジュールがFUSE断を起した時、S400に“1”がセットされます。

●2αの場合、上記フラグはLED表示タイミング(2秒に1回)でセットされます。

(5) オプションモジュール状態レジスタ

CPU間リンク、外部機器リンク等のエラー情報が格納されるレジスタです。各モジュール毎の割付けは以下の様になります。なお、本レジスタへのデータセットは各モジュールのサポート用サブOSにて行うため、詳細なビット構成は拡張モジュール購入時に付属されているマニュアルを参照ください。

拡張モジュールレジスタの割付

S500	システム予約	
S580		… CPU間リンクモジュール情報
S5C0		… PSEリンクモジュール情報 (2αのみ使用)
S600		… 外部機器リンクモジュール情報 (2αのみ使用)
S640		… 高速リモートI/Oモジュール情報 (2αのみ使用)
S680	システム予約	
S6C0		… PC _s I/Oリンクモジュール情報 (4αのみ使用)

- 上記レジスタはCPU GR (復電時) で“ゼロ”クリアされます。
(2αはリセットでも同様となります。)

(6) I/O固定パターン (2αのみ)

OSがI/Oメモリのイニシャルチェックに使用する固定パターンを格納したエリアです。本エリアのパターンが破壊された場合、CPUリセット時に全I/Oメモリ（演算命令ワークエリアも含む）を全て“ゼロ”クリアします。

I/O固定パターン

	(MSB) 2 ¹⁵																	2 ⁰ (LSB)
SWBA0	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0		(=H55AA)
SWBB0	1	1	1	0	1	1	1	0	1	0	0	1	1	0	0	1		(=HEE99)

(7) CPUステータスレジスタ

現在のCPU状態を示すレジスタです。レジスタの各ビットは下記構成となっており、本エリアをPSE等で読出しすることにより、CPUの状態を知ることができます。

CPUステータスのビット構成

(MSB)	2 ¹⁵	2 ¹⁴	2 ¹³	2 ¹²	2 ¹¹	2 ¹⁰	2 ⁹	2 ⁸	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰ (LSB)
SWBF0	1/0	1/0	1/0	1/0	1/0	*	*	*	1/0	1/0	1/0	1/0	*	1/0	1/0	1/0
ビットNo	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

*は将来拡張用

ビットNo	リレー	各ビット内容	
		"1" ON	"0" OFF
0	SBF0	"STOP" 中	"RUN" 中
1	SBF1	シミュレーション中	通常RUN中
2	SBF2	FORCE モード実行中	通常モード実行中
3	SBF3	プロテクト SW "ON" 状態	プロテクト SW "OFF" 状態
4	SBF4	リモートI/O 動作中	リモートI/O 停止中
5	SBF5	未使用	
6	SBF6		
7	SBF7		
8	SBF8		
9	SBF9	タイムアウトエラーステーション有	無
10	SBFA	FUSE断ステーション有	無
11	SBFB	オプションモジュールエラー有	無
12	SBFC	将来拡張用	
13	Sbfd	GR時のみゼロクリア	
14	SBFE	CPUダウン中表示	CPU正常運転中
15	SbFF	IPL用プログラム実行中	CPU OS実行中

* 2αのみ

2

演算ファンクション

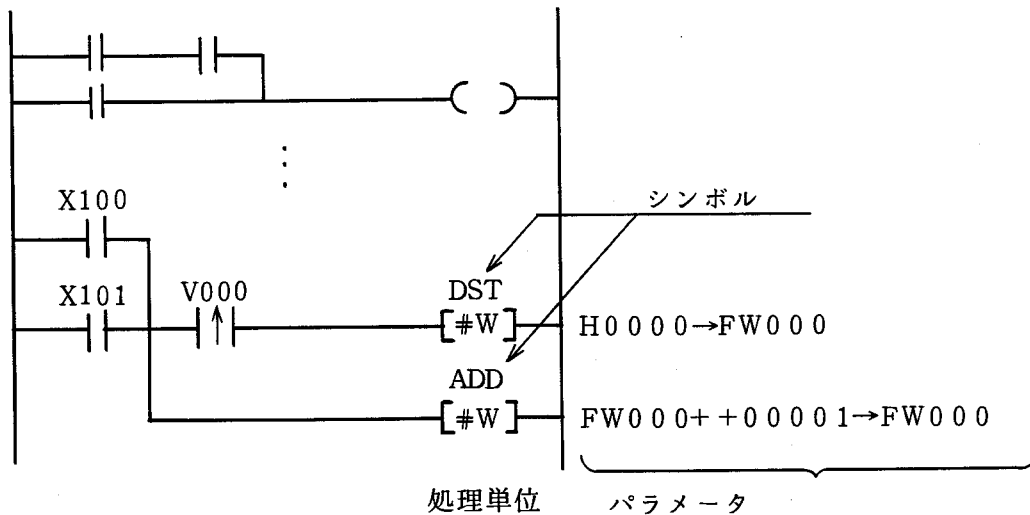
2 演算ファンクション

2.1 機能概要

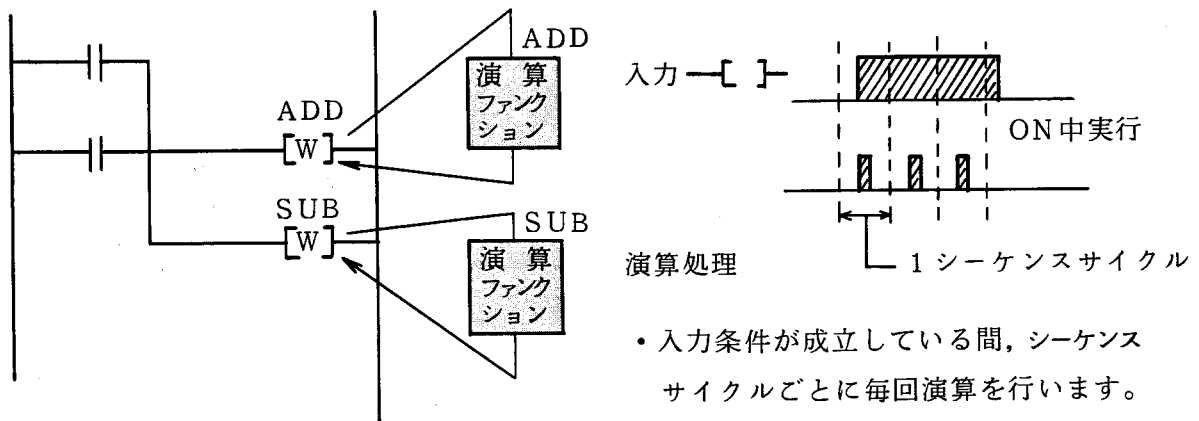
簡単な算術演算等を行いたい場合は、演算ファンクションを使用すれば簡単にプログラムすることができます。

〔1〕 演算ファンクションの動作

〔回路例〕



〔動作〕



(1) パラメータ

演算ファンクションは処理内容に対応してシンボルを割付けてあります。それぞれのシンボルに対してパラメータがあります。

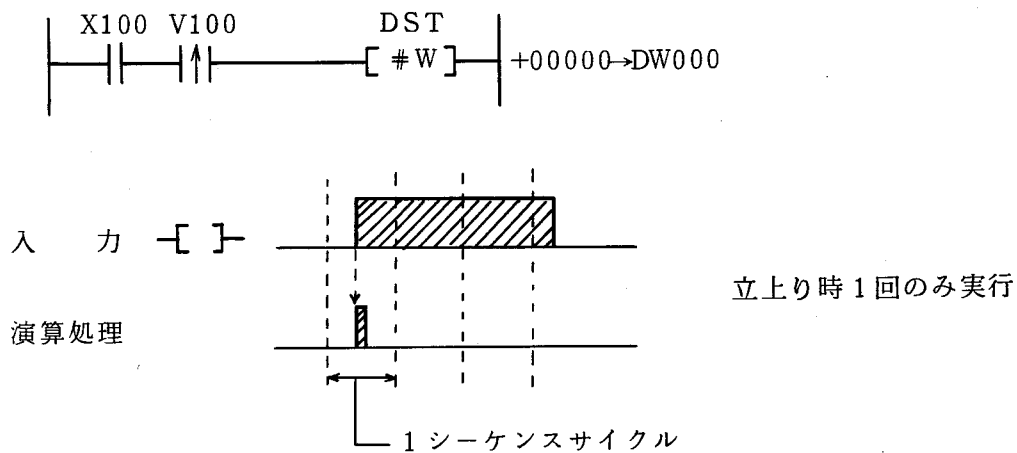
演算ファンクションで使用する定数データレジスタとワークレジスタは次のようになります。

名 称	シンボル	ナ ン バ		使 用 目 的
		2 a	4 a	
ファンクション データレジスタ	DW	000~FFF	000~7FF	定数データエリア
ファンクション ワークレジスタ	FW	000~BFF		ワークエリア

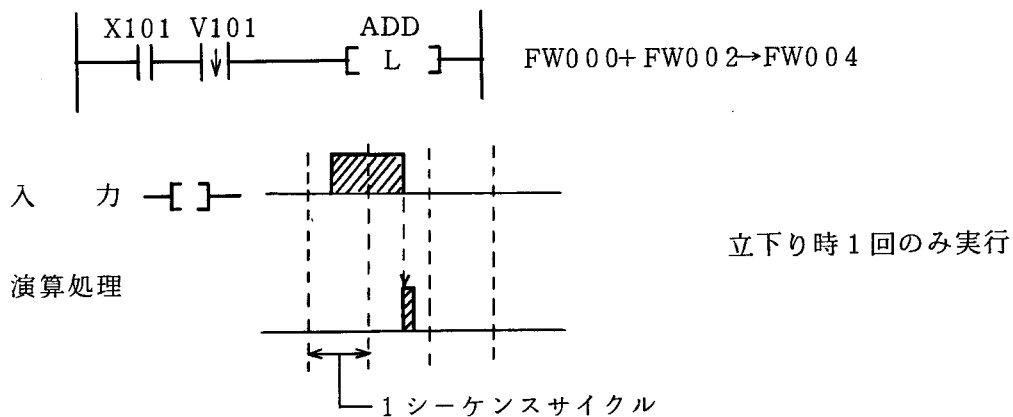
(2) 動 作

演算ファンクションはコイルの励磁信号のON中毎回起動されます。励磁信号の立上り時又は立下り時1回のみ起動させたい場合は、立上り接点又は立下り接点と組み合わせます。

(例1) 立上り接点との組み合わせ

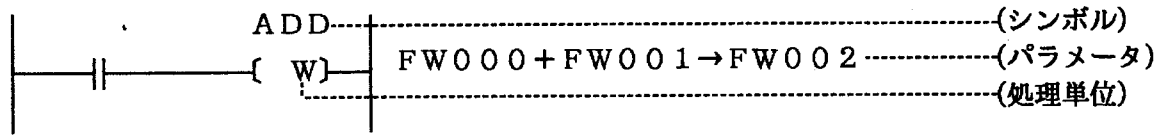


(例2) 立下り接点との組み合わせ



2.2 機能仕様

〔1〕 シンボル



- ・シンボル : 演算ファンクションの機能名称を示します。
- ・パラメータ : 演算の対称となるアドレスや定数データを示します。
- ・処理単位 : 演算の処理単位を示します。

(

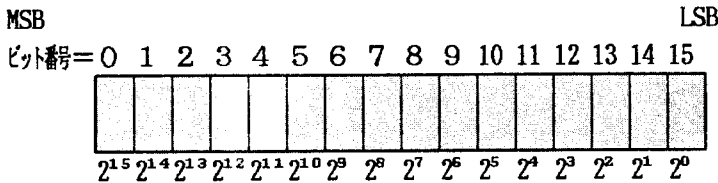
W : ワード
L : ロングワード
#W : イミディエートワードデータ
#L : イミディエートロングデータ

)

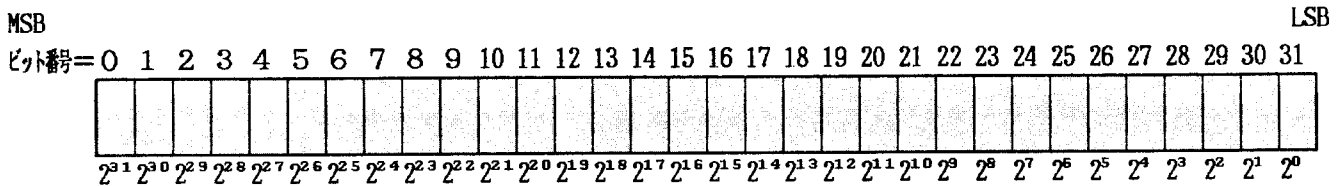
イミディエートデータ：
演算ファンクション命令のパラメータに
定数を指定するものです。

〔2〕 データフォーマット

■ ワード



■ ロングワード



演算ファンクションで使用されるデータは、符号付き16ビット単精度整数又は符号付き32ビット倍精度整数です。16ビットで表わされるデータをワードデータと呼び、32ビットで表わされるデータをロングワードデータと呼びます。各ビットは上記の様にビット番号が付けられています。

ビット番号 '0' のビットは符号ビットと呼ばれ、このビットが0の時=正、1の時=負の数を表わします。従って演算ファンクションで取扱うデータの範囲は次のようになります。

ワード:	-32768	≤	ワードデータ	≤	+32767	(10進数)
	(H8000)				(H7FFF)	(16進数: Hで表わします。)
ロングワード:	-2147483648	≤	ロングワードデータ	≤	+2147483647	(10進数)
	(H80000000)				(H7FFFFFFF)	(16進数: Hで表わします。)

[3] フラグの設定

演算ファンクションは、演算結果に従い各種フラグを設定します。以下にフラグの種類とフラグが設定されるエリア及びフラグが設定される条件を示します。

X:イクステンド	S000	X
E:エラー	S001	E
P:ポジティブ	S002	P
N:ネガティブ	S003	N
Z:ゼロ	S004	Z
V:オーバーフロー	S005	V
F.U.:将来拡張用	S006	F.U.
	S007	F.U.

No	種別	フラグ						特殊な条件			
		X	E	P	N	Z	V	ワード時	ロングワード時		
1	ADD	-	-	-	-	-	↑	V: 結果<-32768又は32767<結果の時1, それ以外0	V: 結果<-2147483648又は2147483647 <結果の時1, それ以外0		
	SUB	-	-	-	-	-	↑				
	INC	-	-	-	-	-	↑				
	DEC	-	-	-	-	-	↑				
	MUL	-	-	-	-	-	↑				
2	DIV	-	↑	-	-	-	↑	E:除数=0の時1, それ以外0 V:商=32768の時1, それ以外0	E:除数=0の時1, それ以外0 V:商=2147483648の時1, それ以外0		
	MOD	-	↑	-	-	-	↑				
3	SCL	-	↑	-	-	-	↑	E:除数=0の時1, それ以外0 V:繰<-32768又は32767<繰の時1, それ以外0			
4	TST	-	-	↑	↑	↑	-	P:データ>0の時1, それ以外0 N:データ<0の時1, それ以外0 Z:データ=0の時1, それ以外0			
		-	↑	-	-	-	↑			E:データ<0の時1, それ以外0 V:データ>9999の時1, それ以外0	E:データ<0の時1, それ以外0 V:データ>99999999の時1, それ以外0
		-	↑	-	-	-	-				
7	APB	-	↑	-	-	-	-	E:H30~H39, H41~H46以外のデータ 検出時1, それ以外0			
	AUB	-	↑	-	-	-	-				
8	DTS	-	-	-	-	-	↑		V:データ<-32768又は32767<データの時1, それ以外0		
9	ABS	-	-	-	-	-	↑	V:データ=-32768の時1, それ以外0	V:データ=-2147483648の時1, それ以外0		
	NEG	-	-	-	-	-	↑				
10	ECD	-	↑	-	-	-	-	E:データ=0の時1, それ以外0			
11	ASL	-	-	-	-	-	↑	V:シフト操作中に符号ビットが1回でも変化すれば1, それ以外0			
12	LIM	-	↑	-	-	-	-	E:上限値<下限値の時1, それ以外0			
13	BND	-	↑	-	-	-	↑	E:上限値<下限値の時1, それ以外0 V:繰<-32768又は32767<繰の時1, それ以外0	E:上限値<下限値の時1, それ以外0 V:繰<-2147483648又は2147483647 <結果の時1, それ以外0		
	ZON	-	↑	-	-	-	↑				
14	上記以外	-	-	-	-	-	-	すべて保持。			

- : 演算実行直前の値を保持。

↑ : 特殊な条件を参照

2.3 パラメータ入力

(1) ニーモニック入力の場合

設定可能エリア	入力例	備考
I/Oエリア (ビット)	X000	ビット指定
I/Oエリア (ワード)	YW000	Wはワードを示します。
ファンクションワークレジスタエリア	FW025	ワークエリア
ファンクションデータレジスタエリア	DW050	定数データエリア
T, U, C設定値エリア	TS003	Sは設定値を示します。
T, U, C計数值エリア	UC007	Cは計数值を示します。
高速I/O (ワード) エリア	IW000	

- I/Oエリア: X, Y, R, K, T, U, C, G, N, P, E, V, Z, S
- ナンバーは3桁入力で行います。
- I/Oエリアで複数のエリアを持つもの (T, U, Cのセットコイルと接点のエリア、Nのセットコイルと接点エリア等) を設定した場合は、接点エリアをアクセスします。
例: TW010 → T010 ~ T01Fの接点エリアを示します。

(2) データ入力の場合

(a) 10進数入力

(i) 直接数値を入力 (正の10進数)

(ii) +/-を入力後数値入力

(iii) 入力桁数は

・ワードの場合 MAX5桁

・ロングワードの場合 MAX9桁

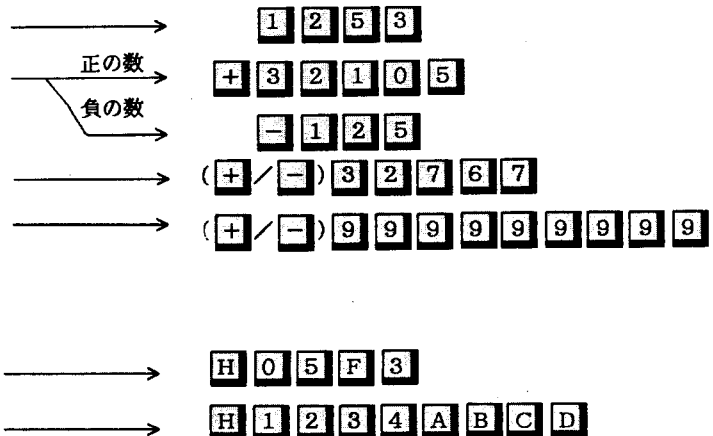
(b) 16進数入力

(i) 'H' を入力後数値入力

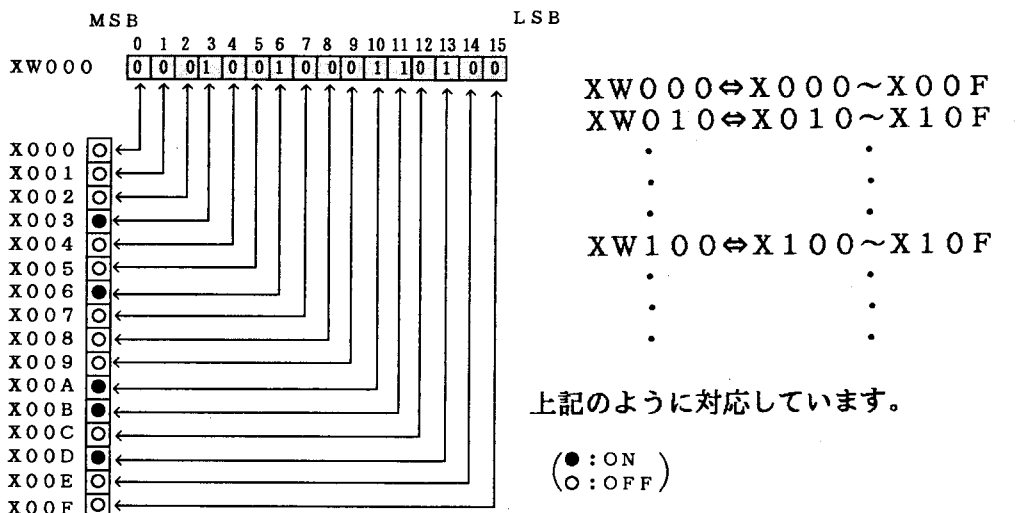
(ii) 入力桁数は

・ワードの場合 MAX4桁

・ロングワードの場合 MAX8桁



(3) I/Oエリアのビットエリアとワードエリアについて



2.4 演算ファンクション

大分類	分類	シンボル	処理内容	処理単位	フラグ						ステップ	時間(μs)		ページ	
					X	E	P	N	Z	V		2α	4α		
算術演算命令	加算	ADD	$(S) + (D) \rightarrow (R)$	ワード							↕	4	0.06	0.6	38
		ADD#	$(S) + n \rightarrow (R)$	ワード	-	-	-	-	-	-	↕	4	0.06	0.5	
		ADD!	$(S)L + (D)L \rightarrow (R)L$	ロングワード							↕	4	0.06	0.8	
	減算	SUB	$(S) - (D) \rightarrow (R)$	ワード							↕	4	0.06	0.6	40
		SUB#	$(S) - n \rightarrow (R)$	ワード	-	-	-	-	-	-	↕	4	0.06	0.5	
		SUB!	$(S)L - (D)L \rightarrow (R)L$	ロングワード							↕	4	0.06	0.8	
	+1	INC	$(S) + 1 \rightarrow (S)$	ワード							↕	2	0.04	0.5	42
		INC!	$(S)L + 1 \rightarrow (S)L$	ロングワード							↕	2	0.04	0.6	
	-1	DEC	$(S) - 1 \rightarrow (R)$	ワード							↕	2	0.04	0.5	44
		DEC!	$(S)L - 1 \rightarrow (S)L$	ロングワード							↕	2	0.04	0.6	
	乗算	MUL	$(S) \times (D) \rightarrow (R)$	ワード							↕	4	0.08	0.9	46
		MUL#	$(S) \times n \rightarrow (R)$	ワード	-	-	-	-	-	-	↕	4	0.08	0.8	
		MUL!	$(S)L \times (D)L \rightarrow (R)L$	ロングワード							↕	4	0.13	2.3	
	除算	DIV	$(S) \div (D) \rightarrow (R)$ (商)	ワード		↕					↕	4	0.09	1.0	48
		DIV#	$(S) \div n \rightarrow (R)$ (商)	ワード	-	↕	-	-	-	-	↕	4	0.09	0.9	
		DIV!	$(S)L \div (D)L \rightarrow (R)L$ (商)	ロングワード							↕	4	0.38	3.0	
	剰余	MOD	$(S) \div (D) \rightarrow (R)$ (剰余)	ワード		↕					↕	4	0.11	1.0	50
		MOD#	$(S) \div n \rightarrow (R)$ (剰余)	ワード	-	↕	-	-	-	-	↕	4	0.11	0.9	
MOD!		$(S)L \div (D)L \rightarrow (R)L$ (剰余)	ロングワード							↕	4	0.46	3.0		
スラッシュ	SCL	$(S) \times (D1) \div (D2) \rightarrow (R)$	ワード		↕					↕	5	0.12	3.1	52	
	SCL#	$(S) \times n1 \div n2 \rightarrow (R)$	ワード	-	↕	-	-	-	-	↕	5	0.11	2.9		
論理演算命令	論理AND	AND	$(S) \wedge (D) \rightarrow (R)$	ワード								4	0.05	0.6	54
		AND#	$(S) \wedge n \rightarrow (R)$	ワード	-	-	-	-	-	-		4	0.05	0.5	
		AND!	$(S)L \wedge (D)L \rightarrow (R)L$	ロングワード								4	0.05	0.8	
	論理OR	OR	$(S) \vee (D) \rightarrow (R)$	ワード								4	0.05	0.6	56
		OR#	$(S) \vee n \rightarrow (R)$	ワード	-	-	-	-	-	-		4	0.05	0.5	
		OR!	$(S)L \vee (D)L \rightarrow (R)L$	ロングワード								4	0.05	0.8	
排他的論理AND	EOR	$(S) \oplus (D) \rightarrow (R)$	ワード								4	0.05	0.6	58	
	EOR#	$(S) \oplus n \rightarrow (R)$	ワード	-	-	-	-	-	-		4	0.05	0.5		
	EOR!	$(S)L \oplus (D)L \rightarrow (R)L$	ロングワード								4	0.05	0.8		
否定	NOT	$\overline{(S)} \rightarrow (R)$	ワード								3	0.04	0.4	60	
	NOT!	$\overline{(S)L} \rightarrow (R)L$	ロングワード	-	-	-	-	-	-		3	0.04	0.6		
比較演算命令	=	EQU	$(S) = (D)$ の時 1→(R) $(S) \neq (D)$ の時 0→(R)	ワード								4	0.05	0.6	62
		EQU#	$(S) = n$ の時 1→(R) $(S) \neq n$ の時 0→(R)	ワード	-	-	-	-	-	-		4	0.05	0.5	
		EQU!	$(S)L = (D)L$ の時 1→(R) $(S)L \neq (D)L$ の時 0→(R)	ロングワード								4	0.06	0.7	
≠	NEQ	NEQ	$(S) = (D)$ の時 0→(R) $(S) \neq (D)$ の時 1→(R)	ワード								4	0.05	0.6	64
		NEQ#	$(S) = n$ の時 0→(R) $(S) \neq n$ の時 1→(R)	ワード	-	-	-	-	-	-		4	0.05	0.5	
		NEQ!	$(S)L = (D)L$ の時 0→(R) $(S)L \neq (D)L$ の時 1→(R)	ロングワード								4	0.06	0.7	

2 演算ファンクション

大分類	分類	シンボル	処理内容	処理単位	フラグ						ステップ 数	遅延時間(mS)		ページ
					X	E	P	N	Z	V		2α	4α	
比較 演算 命令	>	GT	(S) > (D)の時 1→(R) (S) ≤ (D)の時 0→(R)	ワード							4	0.05	0.06	66
		GT#	(S) > nの時 1→(R) (S) ≤ nの時 0→(R)	ワード	-	-	-	-	-	-	4	0.05	0.5	
		GT!	(S)L > (D)Lの時 1→(R) (S)L ≤ (D)Lの時 0→(R)	ロングワード							4	0.06	0.7	
	≥	GE	(S) ≥ (D)の時 1→(R) (S) < (D)の時 0→(R)	ワード							4	0.05	0.6	68
		GE#	(S) ≥ nの時 1→(R) (S) < nの時 0→(R)	ワード	-	-	-	-	-	-	4	0.05	0.5	
		GE!	(S)L ≥ (D)Lの時 1→(R) (S)L < (D)Lの時 0→(R)	ロングワード							4	0.06	0.7	
	<	LT	(S) < (D)の時 1→(R) (S) ≥ (D)の時 0→(R)	ワード							4	0.05	0.6	70
		LT#	(S) < nの時 1→(R) (S) ≥ nの時 0→(R)	ワード	-	-	-	-	-	-	4	0.05	0.5	
		LT!	(S)L < (D)Lの時 1→(R) (S)L ≥ (D)Lの時 0→(R)	ロングワード							4	0.06	0.7	
	≤	LE	(S) ≤ (D)の時 1→(R) (S) > (D)の時 0→(R)	ワード							4	0.05	0.6	72
		LE#	(S) ≤ nの時 1→(R) (S) > nの時 0→(R)	ワード	-	-	-	-	-	-	4	0.05	0.5	
		LE!	(S)L ≤ (D)Lの時 1→(R) (S)L > (D)Lの時 0→(R)	ロングワード							4	0.06	0.7	
テスト	TST	(S) をテストし、P, N, Zフラグをセットします。	ワード			↑	↑	↑		2	0.04	0.4	74	
	TST!	(S)L をテストし、P, N, Zフラグをセットします。	ロングワード			↑	↑	↑		2	0.04	0.5		
データ 転送 命令	転送	MOV	(S) → (D)	ワード							3	0.04	0.4	76
		MOV!	(S)L → (D)L	ロングワード							3	0.04	0.6	
	一括 転送	MOM	(S) → (D)	ワード							4	0.06+	0.3+	78
		MOM!	(S) または (S)L → (D) または (D)L	ロングワード							4	0.06+	0.3+	
	イン クリ メント	MSI	i(S) → (D)	ワード							4	0.05+	0.6+	80
		MDI	(S) → i(D)	ワード							4	0.05+	0.6+	
	同 チ タ キ 磁 錠	INI	(S) → (D)	ワード							4	0.05+	0.3+	84
		INI#	(S) または n → (D) または (D)L	ワード	-	-	-	-	-	-	4	0.05+	0.3+	
		INI!	(S) または (S)L → (D)L	ロングワード							4	0.06+	0.4+	
	交換	EXC	(S) ↔ (D)	ワード							3	0.04	0.7	86
EXC!		(S)L ↔ (D)L	ロングワード							3	0.04	1.0		
FIFO 書込	PSH	FIFO (S) → テーブル	ワード							3	0.06	1.2	88	

大分類	分類	シンボル	処理内容	処理単位	フラグ						実行 回数	遅延時間(μs)		ページ
					X	E	P	N	Z	V		2α	4α	
データ 転送 命令	FIFO 読出	POP	FIFO テーブル  → (D)	ワード	-	-	-	-	-	-	3	0.07+ 0.006n (nはポイント)	1.3+ 0.2n	90
	データ セット	DST#	n → (D)	ワード	-	-	-	-	-	-	3	0.04	0.3	92
		DST#!	nL → (D)L	ロングワード	-	-	-	-	-	-	4	0.05	0.4	
	アドレス セット	AST	S(アドレスデータ) → (D)L	ロングワード	-	-	-	-	-	-	3	0.04	0.4	94
	サーチ	SCH	S  D	ワード	-	-	-	-	-	-	5	0.07+ 0.004m	0.5+ 0.1m	96
SCH!		一致ナンバー n → (R)	ロングワード	-	-	-	-	-	-	5	0.08+ 0.004m	0.5+ 0.2m		
データ 変換 命令	BIN ↓ BCD	BTD	BIN → BCD (S) → (R)	ワード	-	↑	-	-	-	↓	3	0.26	1.6	98
		BTD!	BIN → BCD (S)L → (R)L	ロングワード	-	↑	-	-	-	↓	3	0.51	9.8	
	BCD ↓ BIN	DTB	BCD → BIN (S) → (R)	ワード	-	↑	-	-	-	-	3	0.21	1.3	100
		DTB!	BCD → BIN (S)L → (R)L	ロングワード	-	↑	-	-	-	-	3	0.37	7.1	
	BIN ↓ 7SEG	SEG	BIN → 7seg (S) → {R, R+1}	ワード	-	-	-	-	-	-	3	0.11	0.6	102
		SEG#	BIN → 7seg n → {R, R+1}	ワード	-	-	-	-	-	-	3	0.11	0.5	
		SEG!	BIN → 7seg (S)L → {R(L), R+2(L)}	ロングワード	-	-	-	-	-	-	3	0.21	0.9	
	BIN ↓ ASCII	ASP	BIN → ASCII (パックモード) (S) → {R, R+1}	ワード	-	-	-	-	-	-	3	0.13	0.6	104
		ASU	BIN → ASCII (アンパックモード) (S) → {R, R+1, R+2, R+3}	ワード	-	-	-	-	-	-	3	0.14	0.8	106
	ASCII ↓ BIN	APB	ASCII → BIN (パックモード) {S, S+1} → (R)	ワード	-	↑	-	-	-	-	3	0.13	0.6	108
		AUB	ASCII → BIN (アンパックモード) {S, S+1, S+2, S+3} → (R)	ワード	-	↑	-	-	-	-	3	0.13	0.8	110
	SINGLE ↓ DOUBLE	STD	(S) → (R)L	ワード	-	-	-	-	-	-	3	0.04	0.5	112
	DOUBLE ↓ SINGLE	DTS	(S)L → (R)	ロングワード	-	-	-	-	-	↑	3	0.05	0.6	114
	絶対値	ABS	(S) → (R)	ワード	-	-	-	-	-	↑	3	0.05	0.5	116
ABS!		(S)L → (R)L	ロングワード	-	-	-	-	-	↑	3	0.05	0.6		
+/-	NEG	-(S) → (R)	ワード	-	-	-	-	-	↑	3	0.05	0.5	118	
	NEG!	-(S)L → (R)L	ロングワード	-	-	-	-	-	↑	3	0.05	0.6		

2 演算ファンクション

大分類	分類	シンボル	処理内容	処理単位	フラグ						ステップ	処理時間 (μs)		ページ
					X	E	P	N	Z	V		2α	4α	
データ変換命令	デコード	DCD	(S) →	ワード							3	0.05	0.4	120
		DCD!	(S) →	ロングワード							3	0.05	0.5	
	エンコード	ECD	(S) → (x-x:任意)	ワード		↑					3	0.11	0.6	122
		ECD!	(S) → (x-x:任意)	ロングワード		↑					3	0.11	0.7	
シフト命令	論理右シフト	LSR		ワード							4	0.06	0.6	124
		LSR#		ワード							4	0.06	0.5	
		LSR!		ロングワード							4	0.06	1.0	
		LSR#!		ロングワード							4	0.06	0.9	
	論理左シフト	LSL		ワード							4	0.06	0.6	126
		LSL#		ワード							4	0.06	0.5	
		LSL!		ロングワード							4	0.06	1.0	
		LSL#!		ロングワード							4	0.06	0.9	

大分類	分類	シンボル	処理内容	処理単位	フラグ						ステップ	遅延時間 (mS)		ページ
					X	E	P	N	Z	V		2α	4α	
シフト命令	算術右シフト	ASR		ワード							4	0.06	0.6	128
		ASR#		ワード							4	0.06	0.5	
		ASR!		ロングワード							4	0.06	1.0	
		ASR#!		ロングワード							4	0.06	0.9	
	算術左シフト	ASL		ワード							4	0.07	0.8	130
		ASL#		ワード							4	0.07	0.7	
		ASL!		ロングワード							4	0.07	1.3	
		ASL#!		ロングワード							4	0.07	1.2	
ローテイト命令	右回転	ROR		ワード							4	0.06	0.6	132
		ROR#		ワード							4	0.06	0.5	
		ROR!		ロングワード							4	0.06	1.1	
		ROR#!		ロングワード							4	0.06	1.0	

2 演算ファンクション

大分類	分類	シンボル	処理内容	処理単位	フラグ						ステップ	処理時間(mS)		ページ
					X	E	P	N	Z	V		2α	4α	
ロー テ ィ ト 命 令	左 転	ROL	(S) 左回転 (R) 0 15-(D) 15 	ワード							4	0.06	0.6	134
		ROL#	(S) 左回転 (R) 0 15-n 15 	ワード							4	0.06	0.5	
		ROL!	左回転 0 (S)L 31-(D)L 31 	ロングワード							4	0.06	1.1	
		ROL#!	左回転 0 (S)L 31-nL 31 	ロングワード							4	0.06	1.0	
関 数 処 理 命 令	L I M I T E R	LIM	(D1) < (S) の時 (D1) → (R) (D2) ≤ (S) ≤ (D1) の時 (S) → (R) (S) < (D2) の時 (D2) → (R)	ワード							5	0.07	0.7	136
		LIM#	n1 < (S) の時 n1 → (R) n2 ≤ (S) ≤ n1 の時 (S) → (R) (S) < n2 の時 n2 → (R)	ワード	-	↕	-	-	-	-	5	0.07	0.5	
		LIM!	(D1)L < (S)L の時 (D1)L → (R)L (D2)L ≤ (S)L ≤ (D1)L の時 (S)L → (R)L (S)L < (D2)L の時 (D2)L → (R)L	ロングワード							5	0.08	1.0	
D E A D B A N D	B N D	BND	(D1) < (S) の時 (S) - (D1) → (R) (D2) ≤ (S) ≤ (D1) の時 0 → (R) (S) < (D2) の時 (S) - (D2) → (R)	ワード							5	0.08	0.7	138
		BND#	n1 < (S) の時 (S) - n1 → (R) n2 ≤ (S) ≤ n1 の時 0 → (R) (S) < n2 の時 (S) - n2 → (R)	ワード	-	↕	-	-	-	↕	5	0.08	0.5	
		BND!	(D1)L < (S)L の時 (S)L - (D1)L → (R)L (D2)L ≤ (S)L ≤ (D1)L の時 0 → (R)L (S)L < (D2)L の時 (S)L - (D2)L → (R)L	ロングワード							5	0.08	1.0	
D E A D Z O N E	Z O N	ZON	(S) > 0 の時 (S) + (D1) → (R) (S) = 0 の時 0 → (R) (S) < 0 の時 (S) + (D2) → (R)	ワード							5	0.08	0.7	140
		ZON#	(S) > 0 の時 (S) + n1 → (R) (S) = 0 の時 0 → (R) (S) < 0 の時 (S) + n2 → (R)	ワード	-	↕	-	-	-	↕	5	0.08	0.5	
		ZON!	(S)L > 0 の時 (S)L + (D1)L → (R)L (S)L = 0 の時 0 → (R)L (S)L < 0 の時 (S)L + (D2)L → (R)L	ロングワード							5	0.08	1.0	
ル ー ト	R O T	ROT	(S) ≥ 0 の時 √(S) → (R) (S) < 0 の時 0 → (R)	ワード							3	0.17	0.8	142
		ROT!	(S)L ≥ 0 の時 √(S)L → (R)L (S)L < 0 の時 0 → (R)L	ロングワード							3	0.20	2.2	

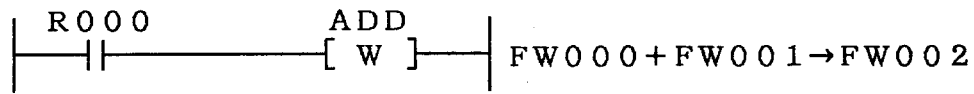
大分類	分類	シンボル	処理内容	処理単位	フラグ						ステップ数	遅延時間(μS)		ページ
					X	E	P	N	Z	V		2α	4α	
関数処理命令	MAX	MAX	(S) ≥ (D)の時 (S) → (R) (S) < (D)の時 (D) → (R)	ワード							4	0.05	0.6	144
		MAX#	(S) ≥ nの時 (S) → (R) (S) < nの時 n → (R)	ワード	-	-	-	-	-	-	4	0.05	0.5	
		MAX!	(S)L ≥ (D)Lの時(S)L → (R)L (S)L < (D)Lの時(D)L → (R)L	ロングワード							4	0.05	0.8	
	MIN	MIN	(S) ≤ (D)の時 (S) → (R) (S) > (D)の時 (D) → (R)	ワード							4	0.05	0.6	146
		MIN#	(S) ≤ nの時 (S) → (R) (S) > nの時 n → (R)	ワード	-	-	-	-	-	-	4	0.05	0.5	
		MIN!	(S)L ≤ (D)Lの時(S)L → (R)L (S)L > (D)Lの時(D)L → (R)L	ロングワード							4	0.05	0.8	
特殊命令	クリア	XCLR	Xのエリアをクリアします。	-	-	-	-	-	-	-	1	0.77	2.3	148
		YCLR	Yのエリアをクリアします。	-	-	-	-	-	-	-	1	0.77	2.3	
		GCLR	Gのエリアをクリアします。	-	-	-	-	-	-	-	1	0.77	2.3	
		RCLR	Rのエリアをクリアします。	-	-	-	-	-	-	-	1	0.77	2.3	
		KCLR	Kのエリアをクリアします。	-	-	-	-	-	-	-	1	0.77	2.3	
		TCLR	Tのエリア及び計数値エリアをクリアします。	-	-	-	-	-	-	-	1	2.21	7.0	
		UCLR	Uのエリア及び計数値エリアをクリアします。	-	-	-	-	-	-	-	1	1.50	4.0	
		CCLR	Cのエリア及び計数値エリアをクリアします。	-	-	-	-	-	-	-	1	1.50	4.0	
		VCLR	Vのエリアをクリアします。	-	-	-	-	-	-	-	1	0.77	2.3	
		ECLR	Eのエリアをクリアします。	-	-	-	-	-	-	-	1	0.77	2.3	
		FCLR	ファンクションフラグをクリアします。	-	0	0	0	0	0	0	1	0.007	0.4	

ADD 加算 = ADD

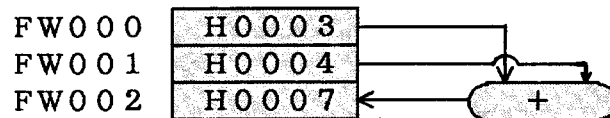
算術演算

機能	ソースの内容とデスティネーションの内容又はイミディエートデータの加算処理を行い、結果をリザルトへ格納します。									
シンボル	ADD	ADD#	ADD!	フ ラ グ						
				X	E	P	N	Z	V	
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	↑↓	
シンボル 及び パラメータ	ADD	[ADD W]			S + D → R					
	ADD#	[ADD #W]			S + n → R					
	ADD!	[ADD L]			S + D → R					
	S : ソース格納アドレス D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス									
処理内容	ADD	(S) + (D) → (R)								
	ADD#	(S) + n → (R)								
	ADD!	(S) L + (D) L → (R) L								
シンボル 入力手順	ADD	1	[FUNC] シフト + 設定				パラメータ入力			
		2	[FUNC] A D D 設定							
	ADD#	1	[FUNC] シフト + #							
		2	[FUNC] A D D #							
	ADD!	1	[FUNC] シフト + !							
		2	[FUNC] A D D !							
フラグの 設定	V : (ワード時) (R) < -32768又は32767 < (R) の時1、それ以外0 (ロングワード時) (R) < -2147483648又は2147483647 < (R) の時1、 それ以外0 他 : 保持									
注意事項	(1) オーバーフロー発生時、リザルトには下記フルスケール値がセットされます。									
		ワード	ロングワード							
	正のオーバーフロー時	H7FFF	H7FFFFFFF							
	負のオーバーフロー時	H8000	H80000000							

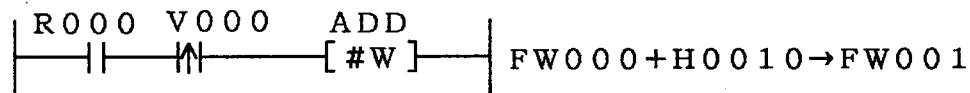
■ ADD



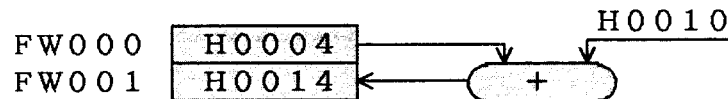
入力条件R000がONの時、FW000の内容とFW001の内容を加算し、その結果をFW002へ格納します。



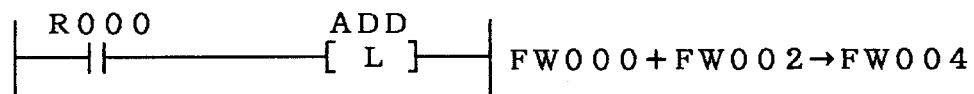
■ ADD #



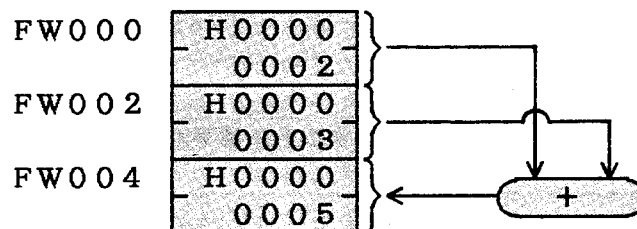
入力条件R000がOFF→ONの変化時、1回のみFW000の内容とイミディエートデータH0010を加算し、その結果をFW001へ格納します。



■ ADD !



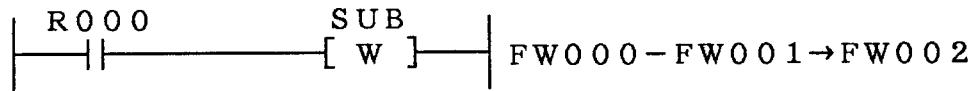
入力条件R000がONの時、FW000の内容とFW002の内容を加算し、その結果をFW004へ格納します。



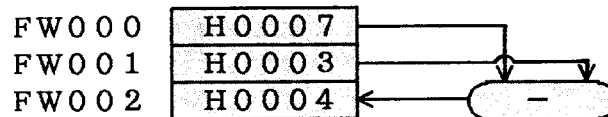
SUB 減算 = SUBTRACT

機能	ソースの内容とデスティネーションの内容又はイミディエートデータの減算処理を行い、結果をリザルトへ格納します。									
シンボル	SUB	SUB#	SUB!	フ ラ グ						
				X	E	P	N	Z	V	
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	↑↓	
シンボル及びパラメータ	SUB	_____ [SUB W] _____			S - D → R					
	SUB#	_____ [SUB #W] _____			S - n → R					
	SUB!	_____ [SUB L] _____			S - D → R					
	S : ソース格納アドレス D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス									
処理内容	SUB	(S) - (D) → (R)								
	SUB#	(S) - n → (R)								
	SUB!	(S) L - (D) L → (R) L								
シンボル入力手順	SUB	1	FUNC [] シフト - 設定				パラメータ入力			
		2	FUNC [] S U B 設定							
	SUB#	1	FUNC [] シフト - #							
		2	FUNC [] S U B #							
	SUB!	1	FUNC [] シフト - !							
		2	FUNC [] S U B !							
フラグの設定	V : (ワード時) (R) < -32768又は32767 < (R) の時1、それ以外0 (ロングワード時) (R) < -2147483648又は2147483647 < (R) の時1、それ以外0 他 : 保持									
注意事項	(1) オーバーフロー発生時、リザルトには下記フルスケール値がセットされます。									
		ワード	ロングワード							
	正のオーバーフロー時	H7FFF	H7FFFFFFF							
	負のオーバーフロー時	H8000	H80000000							

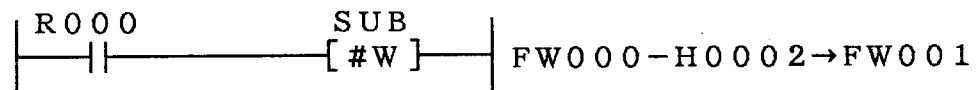
■ SUB



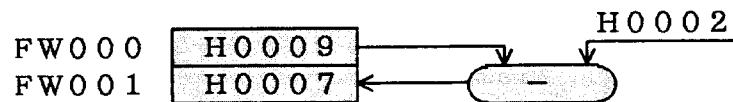
入力条件R000がONの時、FW000の内容からFW001の内容を減算し、その結果をFW002へ格納します。



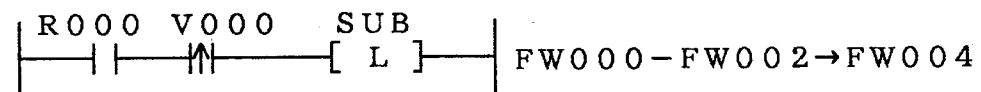
■ SUB



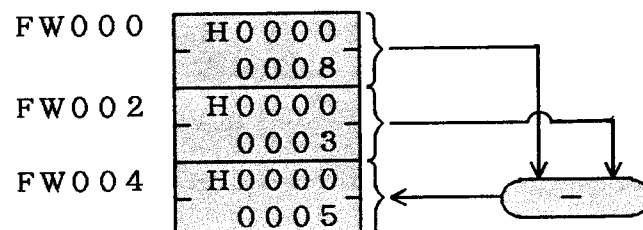
入力条件R000がONの時、FW000の内容からイミディエートデータH0002を減算し、その結果をFW001へ格納します。



■ SUB !



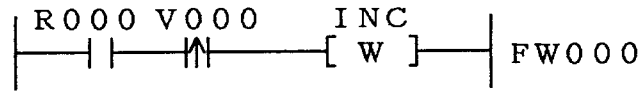
入力条件R000がOFF→ONの変化時1回のみ、FW000の内容からFW002の内容を減算し、その結果をFW004へ格納します。



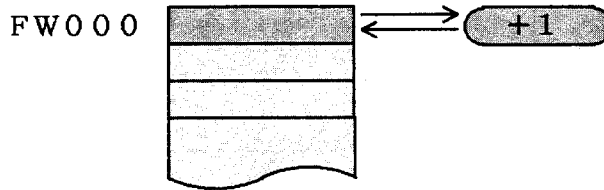
INC +1: INCREMENT

機能	ソースの内容に1を加算します。											
シンボル	INC	INC!	/	フ ラ グ								
				X	E	P	N	Z	V			
処理単位	ワード	ロングワード	/	-	-	-	-	-	↑ ↓			
シンボル 及び パラメータ	INC	[INC W]			S							
	INC!	[INC L]			S							
S: ソース格納アドレス												
処理内容	INC	(S) + 1 → (S)										
	INC!	(S) L + 1 → (S) L										
シンボル 入力手順	INC					<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>						
	INC!											
フラグの 設定	V: (ワード時) 結果=-32768の時1、それ以外0 (ロングワード時) 結果=-2147483648の時1、それ以外0 他: 保持											
注意事項	(1) オーバーフロー発生時、ソースには下記フルスケール値がセットされます。											
	<table border="1" style="margin: auto;"> <tr> <td>ワード</td> <td>ロングワード</td> </tr> <tr> <td>H7FFF</td> <td>H7FFFFFFF</td> </tr> </table>		ワード	ロングワード	H7FFF	H7FFFFFFF						
ワード	ロングワード											
H7FFF	H7FFFFFFF											

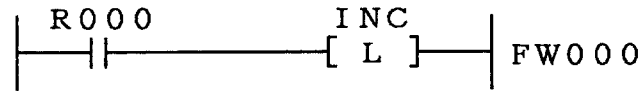
■ INC



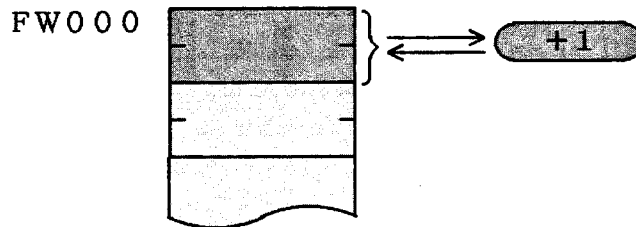
入力条件R000がOFF→ONの変化時、1回のみFW000の内容に1を加算します。



■ INC!



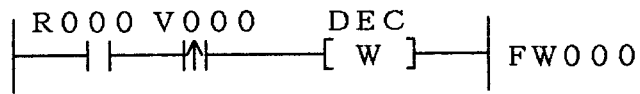
入力条件R000がONの時、FW000の内容に1を加算します。



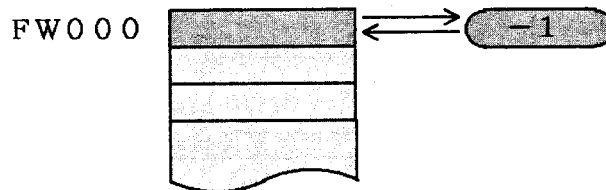
DEC -1: DECREMENT

機能	ソースの内容に1を減算します。													
シンボル	DEC	DEC!	/	フ ラ グ										
				X	E	P	N	Z	V					
処理単位	ワード	ロングワード	/	-	-	-	-	-	↕					
シンボル 及び パラメータ	DEC													
	DEC!													
	S: ソース格納アドレス													
処理内容	DEC	(S) - 1 → (S)												
	DEC!	(S) L - 1 → (S) L												
シンボル 入力手順	DEC					<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>								
	DEC!													
フラグの 設定	V: (ワード時) 結果=32767の時1、それ以外0 (ロングワード時) 結果=2147483647の時1、それ以外0 他: 保持													
注意事項	(1) オーバーフロー発生時、ソースには下記フルスケール値がセットされます。													
	<table border="1" style="margin: auto;"> <tr> <th>ワード</th> <th>ロングワード</th> </tr> <tr> <td>H8000</td> <td>H80000000</td> </tr> </table>		ワード	ロングワード	H8000	H80000000								
ワード	ロングワード													
H8000	H80000000													

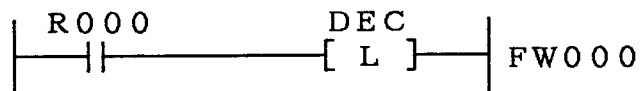
■ DEC



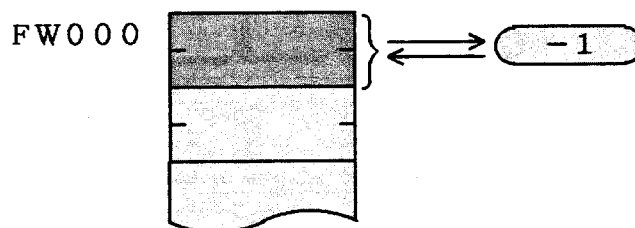
入力条件R000がOFF→ONの変化時、1回のみFW000の内容から1を減算します。



■ DEC!



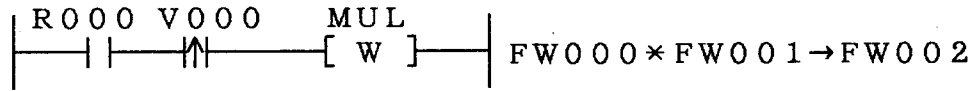
入力条件R000がONの時、FW000の内容から1を減算します。



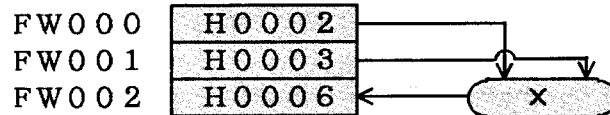
MUL 乗算: MULTIPLY

機能	ソースの内容とデスティネーションの内容又はイミディエートデータの乗算処理を行い、結果をリザルトへ格納します。									
シンボル	MUL	MUL#	MUL!	フ ラ グ						
				X	E	P	N	Z	V	
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	↑↓	
シンボル 及び パラメータ	MUL		[MUL W]	S * D → R						
	MUL#		[MUL #W]	S * n → R						
	MUL!		[MUL L]	S * D → R						
	S: ソース格納アドレス D: デスティネーション格納アドレス n: イミディエートデータ R: リザルト (演算結果) を格納するアドレス									
処理内容	MUL	(S) × (D) → (R)								
	MUL#	(S) × n → (R)								
	MUL!	(S) L × (D) L → (R) L								
シンボル 入力手順	MUL	1	FUNC [H] シフト × 設定				パラメータ入力			
		2	FUNC [H] M U L 設定							
	MUL#	1	FUNC [H] シフト × #							
		2	FUNC [H] M U L #							
	MUL!	1	FUNC [H] シフト × !							
		2	FUNC [H] M U L !							
フラグの 設定	V: (ワード時) (R) < -32768又は32767 < (R) の時1、それ以外0 (ロングワード時) (R) < -2147483648又は2147483647 < (R) の時1、 それ以外0 他: 保持									
注意事項	(1) オーバーフロー発生時、リザルトには下記フルスケール値がセットされます。									
				ワード	ロングワード					
				正のオーバーフロー時	H7FFF	H7FFFFFFF				
				負のオーバーフロー時	H8000	H80000000				

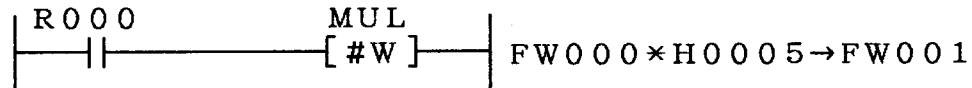
■ MUL



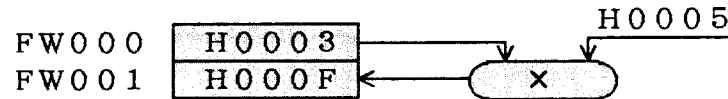
入力条件R000がOFF→ONの変化時1回のみ、FW000の内容とFW001の内容を乗算し、その結果をFW002へ格納します。



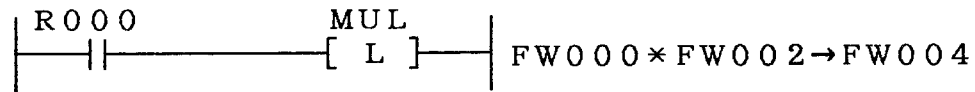
■ MUL #



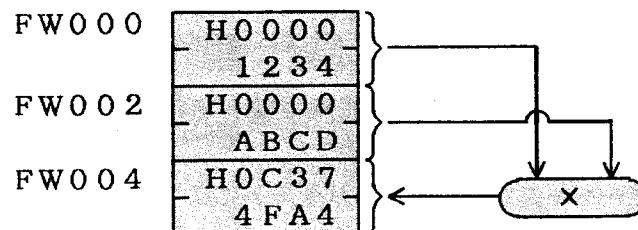
入力条件R000がONの時、FW000の内容とイミディエートデータH0005を乗算し、その結果をFW001へ格納します。



■ MUL !



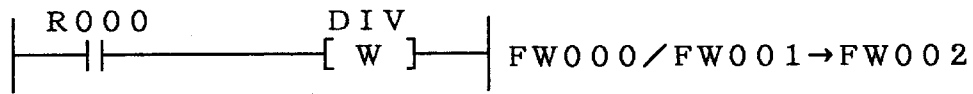
入力条件R000がONの時、FW000の内容とFW002の内容を乗算し、その結果をFW004へ格納します。



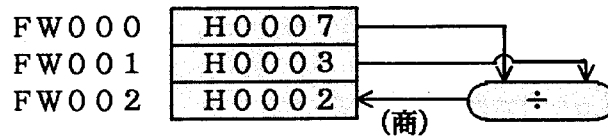
DIV 除算=DIVIDE

機能	ソースの内容とデスティネーションの内容又はイミディエートデータの除算処理を行い、商（整数部のみ）をリザルトへ格納します。												
シンボル	DIV	DIV#	DIV!	フ ラ グ									
				X	E	P	N	Z	V				
処理単位	ワード	ワード	ロングワード	-	↑↓	-	-	-	↑↓				
シンボル及びパラメータ	DIV	_____ [DIV W] _____			S / D → R								
	DIV#	_____ [DIV #W] _____			S / n → R								
	DIV!	_____ [DIV L] _____			S / D → R								
	S : ソース格納アドレス D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス												
処理内容	DIV	(S) ÷ (D) → (R)											
	DIV#	(S) ÷ n → (R)											
	DIV!	(S) L ÷ (D) L → (R) L											
シンボル入力手順	DIV	1	[FUNC] シフト ÷ 設定				[パラメータ入力]						
		2	[FUNC] D I V 設定										
	DIV#	1	[FUNC] シフト ÷ #										
		2	[FUNC] D I V #										
	DIV!	1	[FUNC] シフト ÷ !										
		2	[FUNC] D I V !										
フラグの設定	E : (D) = 0 又は n = 0 の時 1、それ以外 0 V : (ワード時) (R) = 32768 の時 1、それ以外 0 (ロングワード時) (R) = 2147483648 の時 1、それ以外 0 他 : 保持												
注意事項	(1) 0除算時、エラーフラグ (E) が ON し (オーバーフローフラグ (V) OFF)、リザルトは変化しません。 (2) オーバーフロー発生時、リザルトには下記フルスケール値がセットされます。												
	<table border="1" style="margin: auto;"> <tr> <td>ワード</td> <td>ロングワード</td> </tr> <tr> <td>H7FFF</td> <td>H7FFFFFFF</td> </tr> </table>		ワード	ロングワード	H7FFF	H7FFFFFFF							
ワード	ロングワード												
H7FFF	H7FFFFFFF												

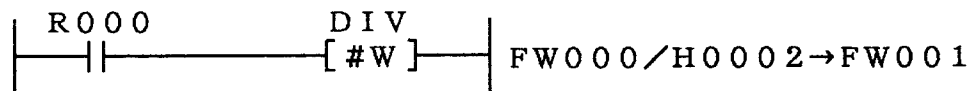
■ DIV



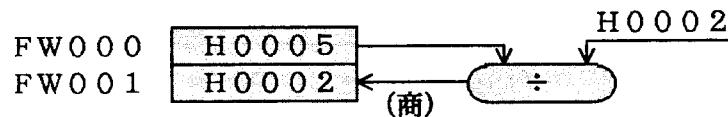
入力条件R000がONの時、FW000の内容をFW001の内容で除算し、その結果(商)をFW002へ格納します。



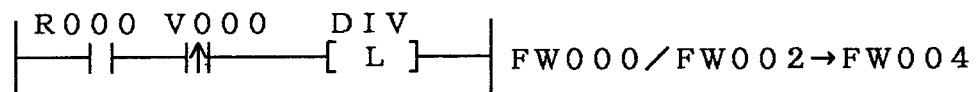
■ DIV #



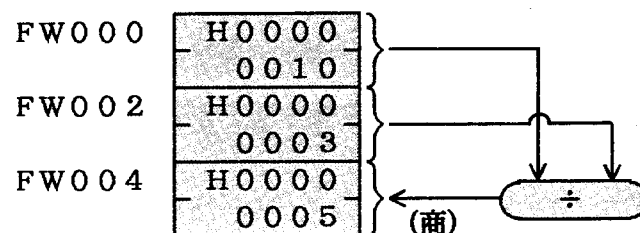
入力条件R000がONの時、FW000の内容とイミディエートデータH0002を除算し、その結果(商)をFW001へ格納します。



■ DIV !



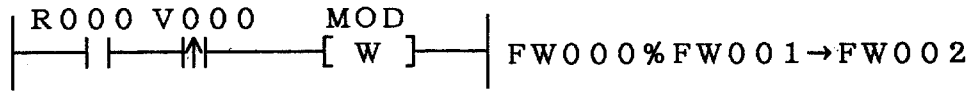
入力条件R000がOFF→ONの変化時1回のみ、FW000の内容をFW002の内容で除算し、その結果(商)をFW004へ格納します。



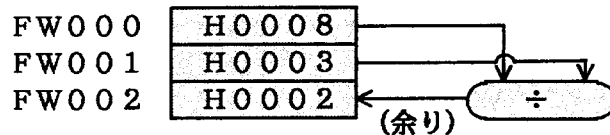
MOD 剰余 = MOD

機能	ソースの内容とデスティネーションの内容又はイミディエートデータの除算処理を行い、剰余をリザルトへ格納します。										
シンボル	MOD	MOD#	MOD!	フ ラ グ							
処理単位	ワード	ワード	ロングワード	X	E	P	N	Z	V		
シンボル及びパラメータ	MOD		$\left[\begin{array}{c} \text{MOD} \\ \text{W} \end{array} \right]$		$\underline{S} \% \underline{D} \rightarrow \underline{R}$						
	MOD#		$\left[\begin{array}{c} \text{MOD} \\ \#W \end{array} \right]$		$\underline{S} \% \underline{n} \rightarrow \underline{R}$						
	MOD!		$\left[\begin{array}{c} \text{MOD} \\ L \end{array} \right]$		$\underline{S} \% \underline{D} \rightarrow \underline{R}$						
処理内容	MOD	(S) % (D) → (R)									
	MOD#	(S) % n → (R)									
	MOD!	(S) L % (D) L → (R) L									
シンボル入力手順	MOD	1	FUNC	[]	シフト	MOD	設定	パラメータ入力			
		2	FUNC	[]	M	O	D				設定
	MOD#	1	FUNC	[]	シフト	MOD	#				
		2	FUNC	[]	M	O	D				#
	MOD!	1	FUNC	[]	シフト	MOD	!				
		2	FUNC	[]	M	O	D				!
フラグの設定	E : (D) = 0 又は n = 0 の時 1、それ以外 0 V : (ワード時) 商 = 32768 の時 1、それ以外 0 (ロングワード時) 商 = 2147483648 の時 1、それ以外 0 他 : 保持										
注意事項	(1) 0除算時、エラーフラグ (E) がONし (オーバーフローフラグ (V) OFF)、リザルトは変化しません。 (2) オーバーフロー発生時、リザルトには0が格納されます。										

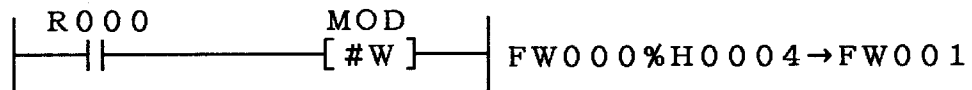
■ MOD



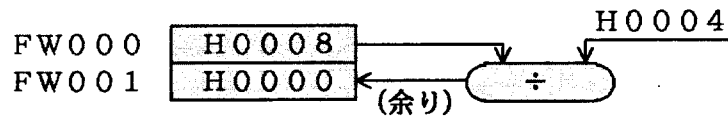
入力条件R000がOFF→ONの変化時1回のみ、FW000の内容をFW001の内容で除算し、剰余をFW002へ格納します。



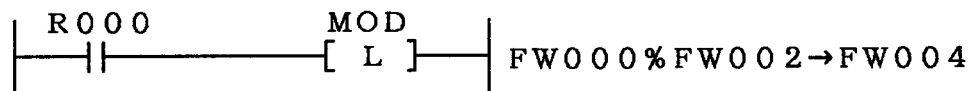
■ MOD #



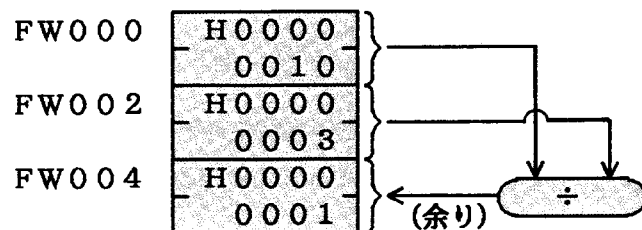
入力条件R000がONの時、FW000の内容をイミディエートデータH0004で除算し、剰余をFW001へ格納します。



■ MOD !



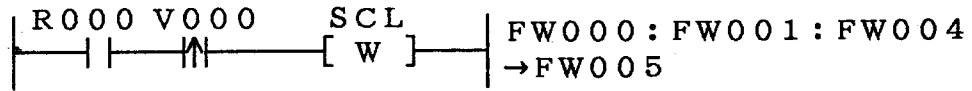
入力条件R000がONの時、FW000の内容をFW002の内容で除算し、剰余をFW004へ格納します。



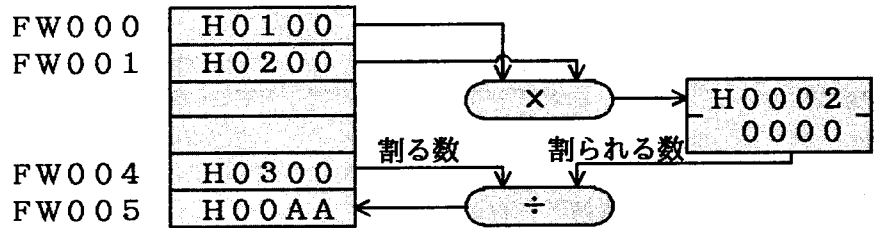
SCL スケール変換 = SCALE CHANGE

機能	ソースの内容のスケール変換（デスティネーションD1/D2又はイミディエートデータn1/n2との乗算）を行い、結果をリザルトへ格納します。							
シンボル	SCL	SCL#	フ ラ グ					
			X	E	P	N	Z	V
処理単位	ワード	ワード	-	↑ ↓	-	-	-	↑ ↓
シンボル 及び パラメータ	SCL	SCL [W]			S : D1 : D2 → R			
	SCL#	SCL [#W]			S : n1 : n2 → R			
処理内容	SCL	(S) × (D1) ÷ (D2) → (R)						
	SCL#	(S) × n1 ÷ n2 → (R)						
シンボル 入力手順	SCL	1	FUNC [H] シフト SCL 設定					パラメータ入力
		2	FUNC [H] S C L 設定					
	SCL#	1	FUNC [H] シフト SCL !					
		2	FUNC [H] S C L !					
フラグの 設定	E : (D2) = 0又はn2 = 0の時1、それ以外0 V : (R) < -32768又は32767 < (R)の時1、それ以外0 他 : 保持							
注意事項	(1) 0除算時、エラーフラグ (E) がONし (オーバーフローフラグ (V) OFF)、リザルトは変化しません。 (2) オーバーフロー発生時、リザルトには下記フルスケール値がセットされます。							
	正のオーバーフロー時		H7FFF					
	負のオーバーフロー時		H8000					

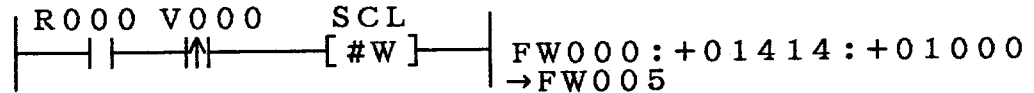
■ SCL



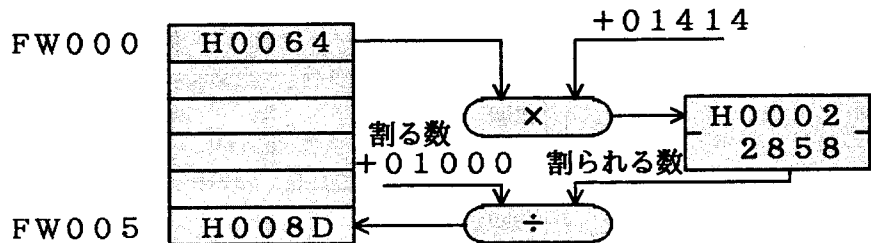
入力条件R000がON時FW000の内容のスケール変換を行い、その結果をFW005へ格納します。



■ SCL #



入力条件R000がOFF→ONの変化時1回のみ、FW000の内容のスケール変換を行い、その結果をFW005へ格納します。

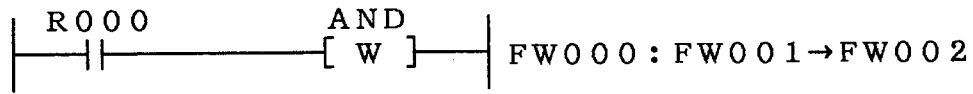


AND 論理積 : AND

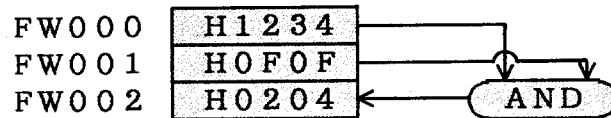
論理演算

機能	ソースの内容とデスティネーションの内容又はイミディエートデータの論理積処理を行い、結果をリザルトへ格納します。									
シンボル	AND	AND#	AND!	フ ラ グ						
				X	E	P	N	Z	V	
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	-	
シンボル 及び パラメータ	AND		$\left[\begin{array}{c} \text{AND} \\ \text{W} \end{array} \right]$	$\underline{S} : \underline{D} \rightarrow \underline{R}$						
	AND#		$\left[\begin{array}{c} \text{AND} \\ \#W \end{array} \right]$	$\underline{S} : \underline{n} \rightarrow \underline{R}$						
	AND!		$\left[\begin{array}{c} \text{AND} \\ L \end{array} \right]$	$\underline{S} : \underline{D} \rightarrow \underline{R}$						
	S : ソース格納アドレス D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス									
処理内容	AND	$(S) \wedge (D) \rightarrow (R)$								
	AND#	$(S) \wedge n \rightarrow (R)$								
	AND!	$(S) L \wedge (D) L \rightarrow (R) L$								
シンボル 入力手順	AND	1	FUNC [H] シフト AND 設定				パラメータ入力			
		2	FUNC [H] A N D 設定							
	AND#	1	FUNC [H] シフト AND #							
		2	FUNC [H] A N D #							
	AND!	1	FUNC [H] シフト AND !							
		2	FUNC [H] A N D !							
注意事項	(1) フラグはすべて保持。									

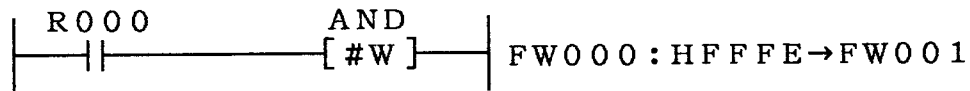
■ AND



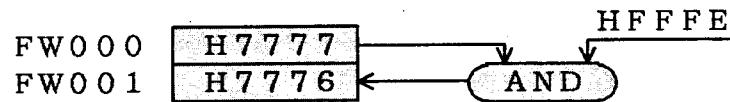
入力条件R000がONの時、FW000の内容とFW001の内容の論理積を取り、その結果をFW002へ格納します。



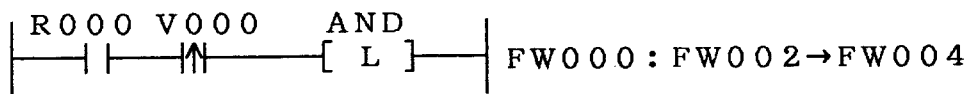
■ AND



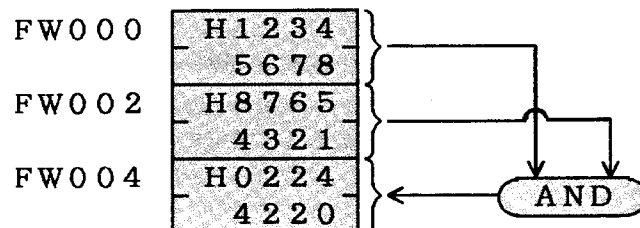
入力条件R000がONの時、FW000の内容とイミディエートデータHFFFEの論理積を取り、その結果をFW001へ格納します。









■ AND !

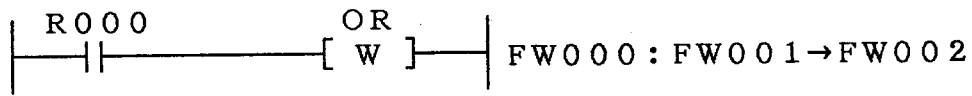


入力条件R000がOFF→ONの変化時1回のみ、FW000の内容とFW002の内容の論理積を取り、その結果をFW004へ格納します。

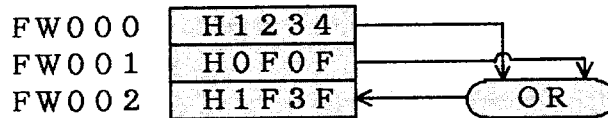


機能	ソースの内容とデスティネーションの内容又はイミディエートデータの論理和処理を行い、結果をリザルトへ格納します。									
シンボル	OR	OR#	OR!	フ ラ グ						
				X	E	P	N	Z	V	
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	-	
シンボル 及び パラメータ	OR		[OR W]	S : D → R						
	OR#		[OR #W]	S : n → R						
	OR!		[OR L]	S : D → R						
S : ソース格納アドレス D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス										
処理内容	OR	(S) V (D) → (R)								
	OR#	(S) V n → (R)								
	OR!	(S) L V(D) L → (R) L								
シンボル 入力手順	OR	1	 シフト OR 設定				<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>			
		2	 O R 設定							
	OR#	1	 シフト OR #							
		2	 O R #							
	OR!	1	 シフト OR !							
		2	 O R !							
注意事項	(1) フラグはすべて保持。									

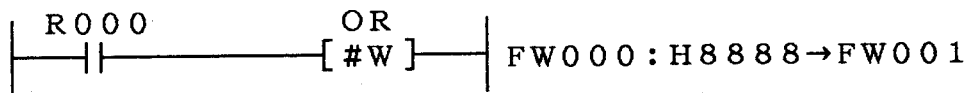
■ OR



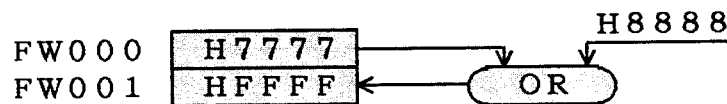
入力条件R000がONの時、FW000の内容とFW001の内容の論理和を取り、その結果をFW002へ格納します。



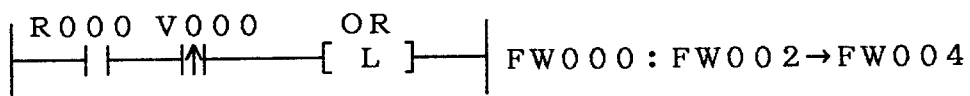
■ OR #



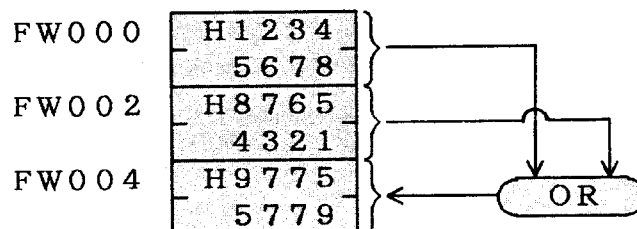
入力条件R000がONの時、FW000の内容とイミディエートデータH8888の論理和を取り、その結果をFW001へ格納します。



■ OR !



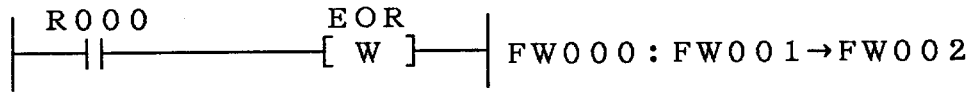
入力条件R000がOFF->ONの変化時1回のみ、FW000の内容とFW002の内容の論理和を取り、その結果をFW004へ格納します。



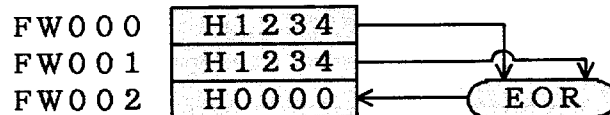
E O R 排他的論理和 : EXCLUSIVE OR

機能	ソースの内容とデスティネーションの内容又はイミディエートデータの排他的論理和处理を行い、結果をリザルトへ格納します。									
シンボル	EOR	EOR#	EOR!	フ ラ グ						
				X	E	P	N	Z	V	
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	-	
シンボル 及び パラメータ	EOR		[EOR W]	S : D → R						
	EOR#		[EOR #W]	S : n → R						
	EOR!		[EOR L]	S : D → R						
	S : ソース格納アドレス D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス									
処理内容	EOR	(S) ⊕ (D) → (R)								
	EOR#	(S) ⊕ n → (R)								
	EOR!	(S) L ⊕ (D) L → (R) L								
シンボル 入力手順	EOR	1	FUNC [H] シフト EOR 設定				パラメータ入力			
		2	FUNC [H] E O R 設定							
	EOR#	1	FUNC [H] シフト EOR #							
		2	FUNC [H] E O R #							
	EOR!	1	FUNC [H] シフト EOR !							
		2	FUNC [H] E O R !							
注意事項	(1) フラグはすべて保持。									

■ EOR



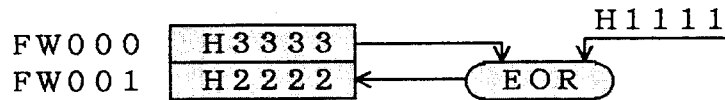
入力条件R000がONの時、FW000の内容とFW001の内容の排他的論理和を取り、その結果をFW002へ格納します。



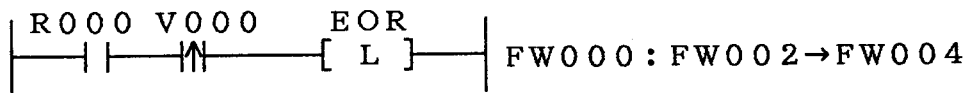
■ EOR



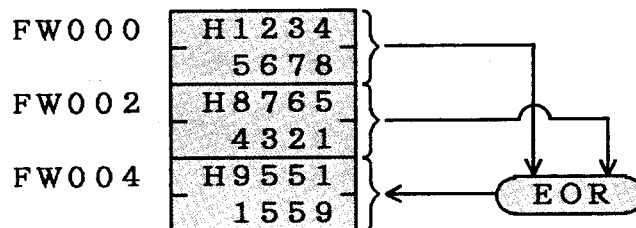
入力条件R000がONの時、FW000の内容とイミディエートデータH1111の排他的論理和を取り、その結果をFW001へ格納します。



■ EOR !



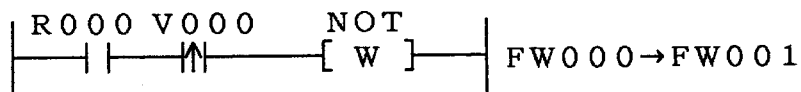
入力条件R000がOFF→ONの変化時1回のみ、FW000の内容とFW002の内容の排他的論理和を取り、その結果をFW004へ格納します。



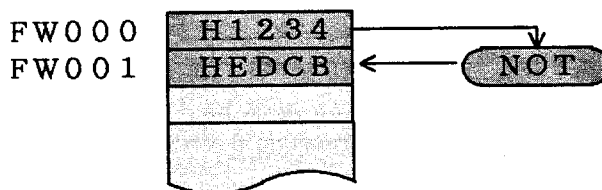
NOT 否定:NOT

機能	ソースの内容の否定 (ビット反転) を、リザルトへ格納します。										
シンボル	NOT	NOT!	/	フ ラ グ							
				X	E	P	N	Z	V		
処理単位	ワード	ロングワード	/	-	-	-	-	-	-		
シンボル 及び パラメータ	NOT										
	NOT!										
	S: ソース格納アドレス R: リザルト (演算結果) を格納するアドレス										
処理内容	NOT	$\overline{(S)} \rightarrow (R)$									
	NOT!	$\overline{(S) L} \rightarrow (R) L$									
シンボル 入力手順	NOT					<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>					
	NOT!										
注意事項	(1) フラグはすべて保持。										

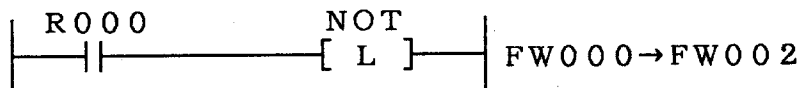
■ NOT



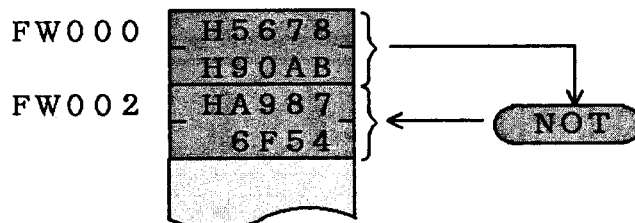
入力条件R000がOFF→ONの変化時、1回のみFW000の内容の否定を取り、FW001へ格納します。



■ NOT !



入力条件R000がONの時、FW000の内容の否定を取り、FW002へ格納します。

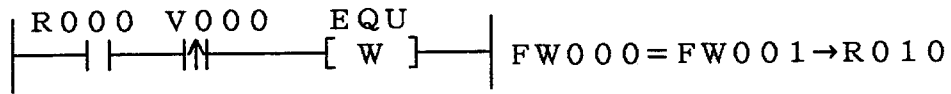


EQU =: EQUAL

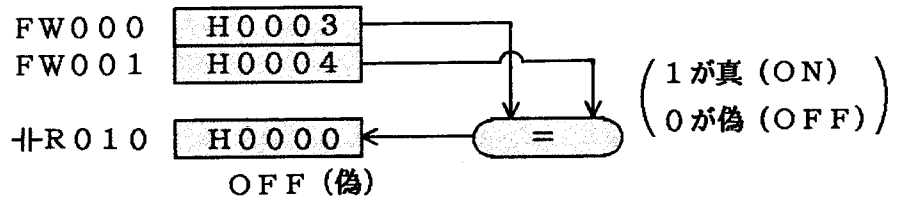
比較演算

機能	ソースの内容とデスティネーションの内容又はイミディエートデータの真偽判定を行い、等しければ1、等しくなければ0をリザルトへセットします。								
シンボル	EQU	EQU#	EQU!	フ ラ グ					
				X	E	P	N	Z	V
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	-
シンボル 及び パラメータ	EQU		[EQU W]	S = D → R					
	EQU#		[EQU #W]	S = n → R					
	EQU!		[EQU L]	S = D → R*					
	S : ソース格納アドレス D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス			*ワード指定					
処理内容	EQU	(S) = (D) の時 1 → (R) (S) ≠ (D) の時 0 → (R)							
	EQU#	(S) = n の時 1 → (R) (S) ≠ n の時 0 → (R)							
	EQU!	(S) L = (D) L の時 1 → (R) (S) L ≠ (D) L の時 0 → (R)							
シンボル 入力手順	EQU	1	FUNC [] シフト = 設定						<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;">パラメータ入力</div>
		2	FUNC [] E Q U 設定						
	EQU#	1	FUNC [] シフト = #						
		2	FUNC [] E Q U #						
	EQU!	1	FUNC [] シフト = !						
		2	FUNC [] E Q U !						
注意事項	(1) フラグはすべて保持。								

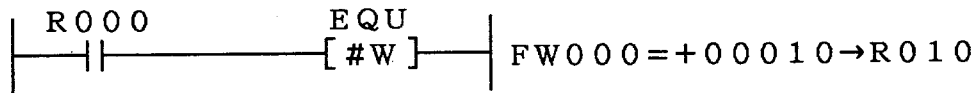
■ EQU



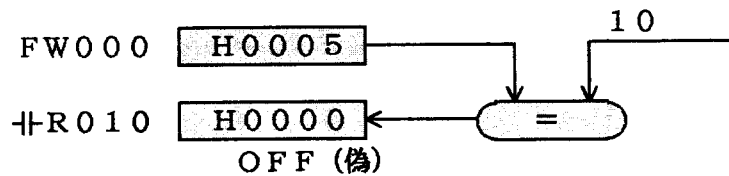
入力条件R000がOFF→ONの変化時、1回のみFW000の内容とFW001の内容の真偽判定を行い、その結果をR010へセットします。



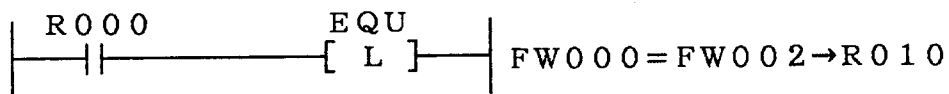
■ EQU #



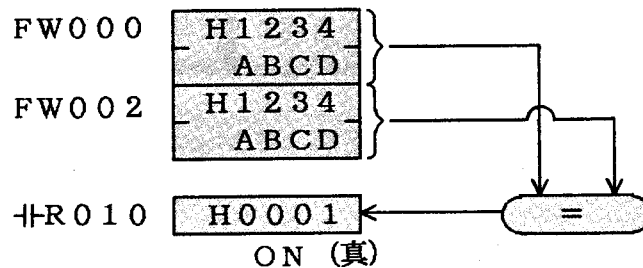
入力条件R000がONの時、FW000の内容とイミディエートデータ10の真偽判定を行い、その結果をR010へセットします。



■ EQU !



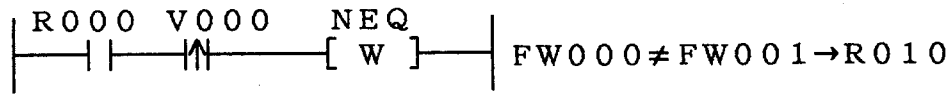
入力条件R000がONの時、FW000の内容とFW002の内容の真偽判定を行い、その結果をR010へセットします。



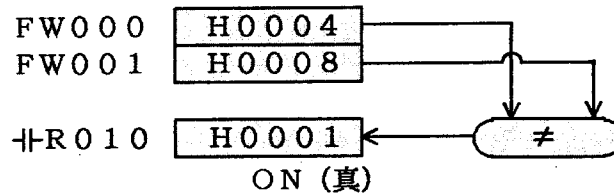
NEQ \neq : NOT EQUAL

機能		ソースの内容とデスティネーションの内容又はイミディエートデータの真偽判定を行い、等しければ0、等しくなければ1をリザルトへセットします。							
シンボル	NEQ	NEQ#	NEQ!	フ ラ グ					
				X	E	P	N	Z	V
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	-
シンボル 及び パラメータ	NEQ		$\left[\begin{array}{c} \text{NEQ} \\ \text{W} \end{array} \right]$	$\underline{S} \neq \underline{D} \rightarrow \underline{R}$					
	NEQ#		$\left[\begin{array}{c} \text{NEQ} \\ \#W \end{array} \right]$	$\underline{S} \neq \underline{n} \rightarrow \underline{R}$					
	NEQ!		$\left[\begin{array}{c} \text{NEQ} \\ \text{L} \end{array} \right]$	$\underline{S} \neq \underline{D} \rightarrow \underline{R}^*$					
	S: ソース格納アドレス D: デスティネーション格納アドレス n: イミディエートデータ R: リザルト (演算結果) を格納するアドレス			*ワード指定					
処理内容	NEQ	(S) = (D) の時 0 → (R) (S) \neq (D) の時 1 → (R)							
NEQ#	(S) = n の時 0 → (R) (S) \neq n の時 1 → (R)								
NEQ!	(S) L = (D) L の時 0 → (R) (S) L \neq (D) L の時 1 → (R)								
シンボル 入力手順	NEQ	1	FUNC [H] シフト \neq 設定					パラメータ入力	
		2	FUNC [H] N E Q 設定						
	NEQ#	1	FUNC [H] シフト \neq #						
		2	FUNC [H] N E Q #						
	NEQ!	1	FUNC [H] シフト \neq !						
		2	FUNC [H] N E Q !						
注意事項	(1) フラグはすべて保持。								

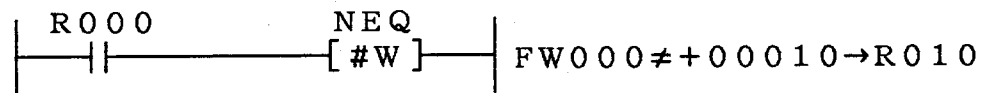
■ NEQ



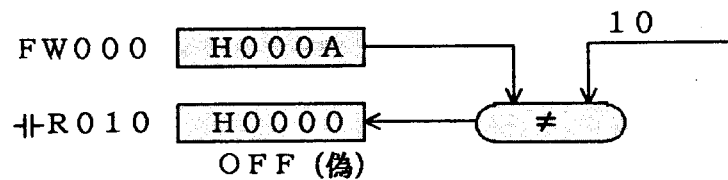
入力条件R000がOFF→ONの変化時、1回のみFW000の内容とFW001の内容の真偽判定を行い、その結果をR010へセットします。



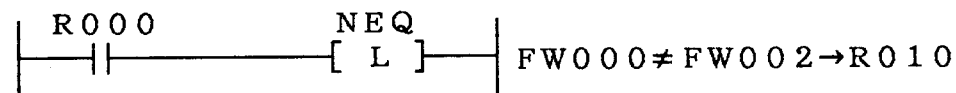
■ NEQ #



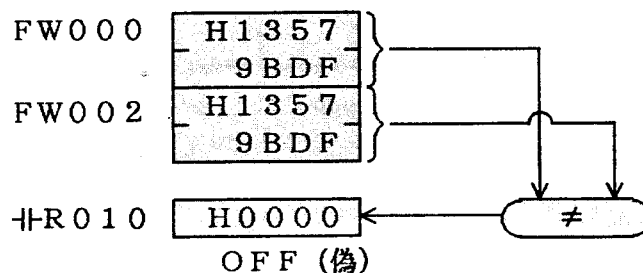
入力条件R000がONの時、FW000の内容とイミディエートデータ10の真偽判定を行い、その結果をR010へセットします。



■ NEQ !



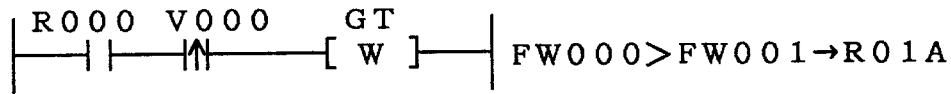
入力条件R000がONの時、FW000の内容とFW002の内容の真偽判定を行い、その結果をR010へセットします。



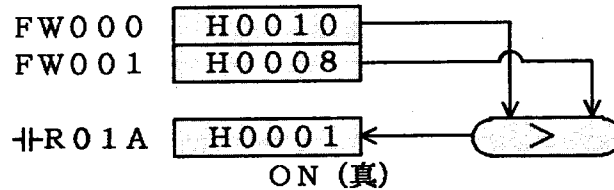
GT >: GREATER THAN

機能	ソースの内容とデスティネーションの内容又はイミディエートデータの大小比較を行い、結果をリザルトへ格納します。									
シンボル	GT	GT#	GT!	フ ラ グ						
				X	E	P	N	Z	V	
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	-	
シンボル及びパラメータ	GT		[GT W]	S > D → R						
	GT#		[GT #W]	S > n → R						
	GT!		[GT L]	S > D → R*						
S : ソース格納アドレス *ワード指定 D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス										
処理内容	GT	(S) > (D) の時 1 → (R) (S) ≤ (D) の時 0 → (R)								
	GT#	(S) > n の時 1 → (R) (S) ≤ n の時 0 → (R)								
	GT!	(S) L > (D) L の時 1 → (R) (S) L ≤ (D) L の時 0 → (R)								
シンボル入力手順	GT	1	FUNC [] シフト > 設定				<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>			
		2	FUNC [] G T 設定							
	GT#	1	FUNC [] シフト > #							
		2	FUNC [] G T #							
	GT!	1	FUNC [] シフト > !							
		2	FUNC [] G T !							
注意事項	(1) フラグはすべて保持。									

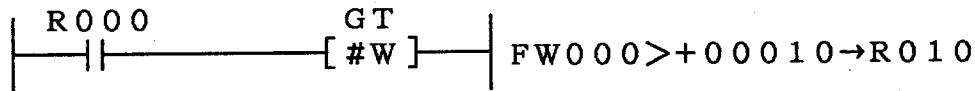
■ GT



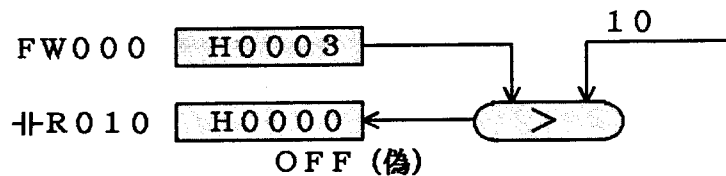
入力条件R000がOFF→ONの変化時、1回のみFW000の内容とFW001の内容の大小比較を行い、その結果をR01Aへセットします。



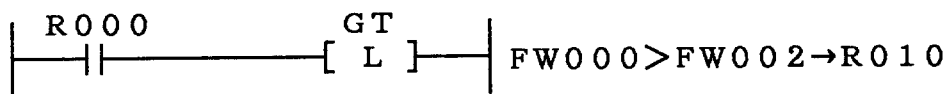
■ GT



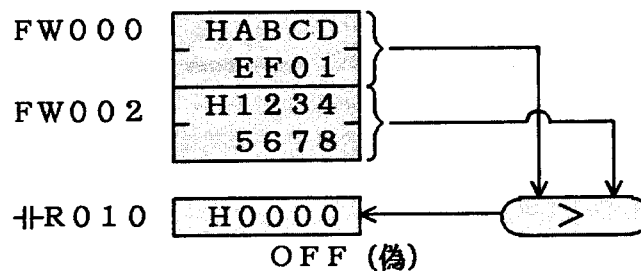
入力条件R000がONの時、FW000の内容とイミディエートデータ10の大小比較を行い、その結果をR010へセットします。



























■ GT !



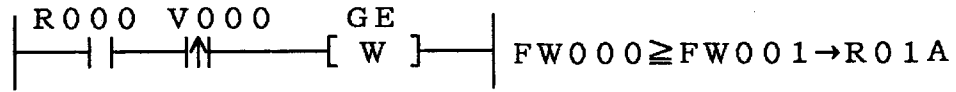
入力条件R000がONの時、FW000の内容とFW002の内容の大小比較を行い、その結果をR010へセットします。



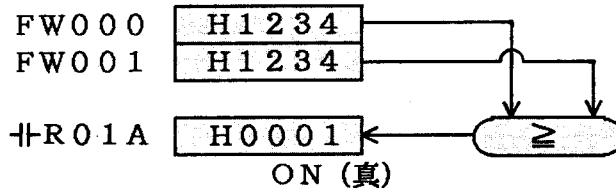
GE ≥: GREATER or EQUAL

機能	ソースの内容とデスティネーションの内容又はイミディエートデータの大小比較を行い、結果をリザルトへ格納します。										
シンボル	GE	GE #	GE !	フ ラ グ							
				X	E	P	N	Z	V		
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	-		
シンボル 及び パラメータ	GE		[GE W]	$S \geq D \rightarrow R$							
	GE #		[GE #W]	$S \geq n \rightarrow R$							
	GE !		[GE L]	$S \geq D \rightarrow R^*$							
				S : ソース格納アドレス *ワード指定 D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス							
処理内容	GE	(S) \geq (D) の時 1 \rightarrow (R) (S) $<$ (D) の時 0 \rightarrow (R)									
	GE #	(S) \geq n の時 1 \rightarrow (R) (S) $<$ n の時 0 \rightarrow (R)									
	GE !	(S) L \geq (D) L の時 1 \rightarrow (R) (S) L $<$ (D) L の時 0 \rightarrow (R)									
シンボル 入力手順	GE	1	   			<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>					
		2	   								
	GE #	1	   								
		2	   								
	GE !	1	   								
		2	   								
注意事項	(1) フラグはすべて保持。										

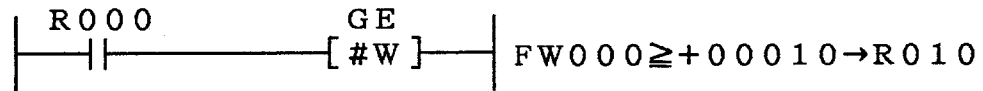
■ GE



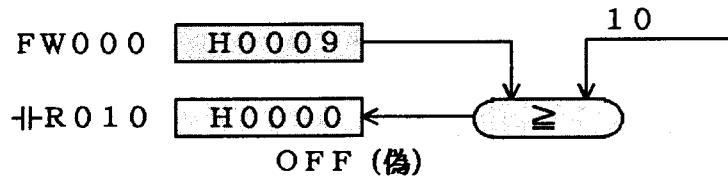
入力条件R000がOFF→ONの変化時、1回のみFW000の内容とFW001の内容の大小比較を行い、その結果をR01Aへセットします。



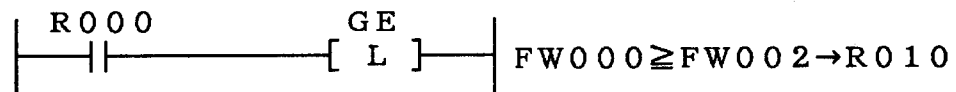
■ GE #



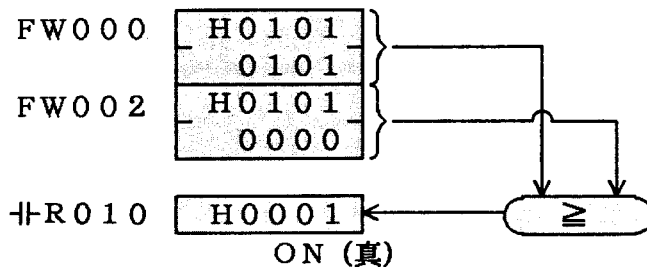
入力条件R000がONの時、FW000の内容とイミディエートデータ10の大小比較を行い、その結果をR010へセットします。



■ GE !



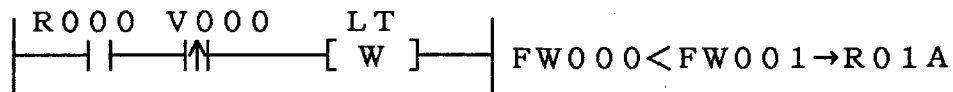
入力条件R000がONの時、FW000の内容とFW002の内容の大小比較を行い、その結果をR010へセットします。



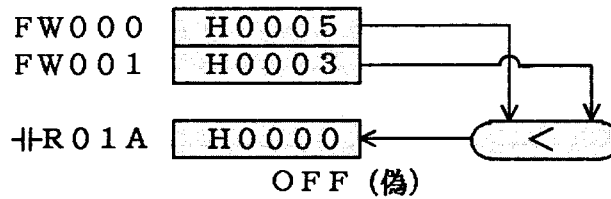
LT <: LESS THAN

機能		ソースの内容とデスティネーションの内容又はイミディエートデータの大小比較を行い、結果をリザルトへ格納します。								
シンボル	処理単位	LT	LT#	LT!	フ ラ グ					
					X	E	P	N	Z	V
	ワード	ワード	ワード	ロングワード	-	-	-	-	-	-
シンボル 及び パラメータ	LT									
	LT#									
	LT!									
		S : ソース格納アドレス *ワード指定 D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス								
処理内容	LT	(S) < (D) の時 1 → (R) (S) ≥ (D) の時 0 → (R)								
	LT#	(S) < n の時 1 → (R) (S) ≥ n の時 0 → (R)								
	LT!	(S) L < (D) L の時 1 → (R) (S) L ≥ (D) L の時 0 → (R)								
シンボル 入力手順	LT	1				<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>				
		2								
	LT#	1								
		2								
	LT!	1								
		2								
注意事項	(1) フラグはすべて保持。									

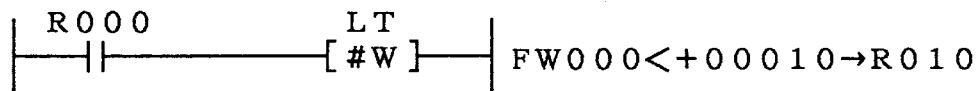
■ LT



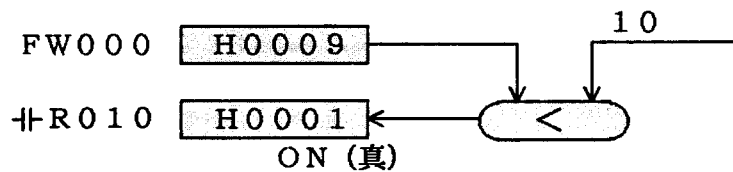
入力条件R000がOFF→ONの変化時、1回のみFW000の内容とFW001の内容の大小比較を行い、その結果をR01Aへセットします。



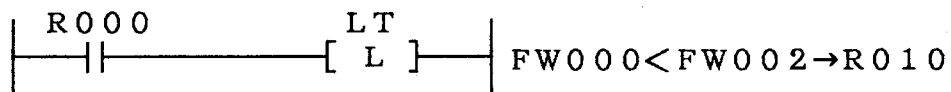
■ LT #



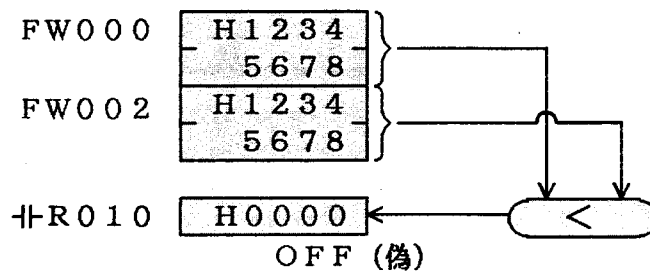
入力条件R000がONの時、FW000の内容とイミディエートデータ10の大小比較を行い、その結果をR010へセットします。



























■ LT !



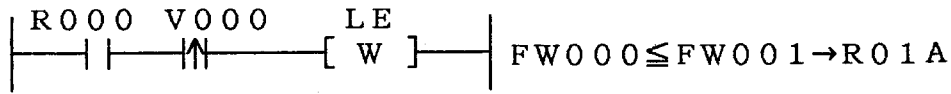
入力条件R000がONの時、FW000の内容とFW002の内容の大小比較を行い、その結果をR010へセットします。



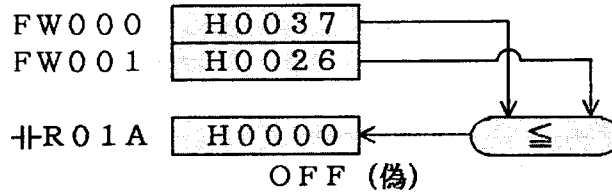
LE ≤: LESS or EQUAL

機能		ソースの内容とデスティネーションの内容又はイミディエートデータの大小比較を行い、結果をリザルトへ格納します。							
シンボル	LE	LE #	LE !	フ ラ グ					
				X	E	P	N	Z	V
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	-
シンボル 及び パラメータ	LE		[LE W]	$S \leq D \rightarrow R$					
	LE #		[LE #W]	$S \leq n \rightarrow R$					
	LE !		[LE L]	$S \leq D \rightarrow R^*$					
		S : ソース格納アドレス *ワード指定 D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス							
処理内容	LE	(S) \leq (D) の時 1 \rightarrow (R) (S) $>$ (D) の時 0 \rightarrow (R)							
	LE #	(S) \leq n の時 1 \rightarrow (R) (S) $>$ n の時 0 \rightarrow (R)							
	LE !	(S) L \leq (D) L の時 1 \rightarrow (R) (S) L $>$ (D) L の時 0 \rightarrow (R)							
シンボル 入力手順	LE	1	   			<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>			
		2	   						
	LE #	1	   						
		2	   						
	LE !	1	   						
		2	   						
注意事項	(1) フラグはすべて保持。								

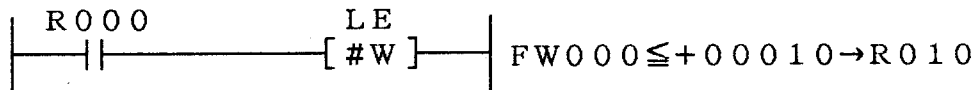
■ LE



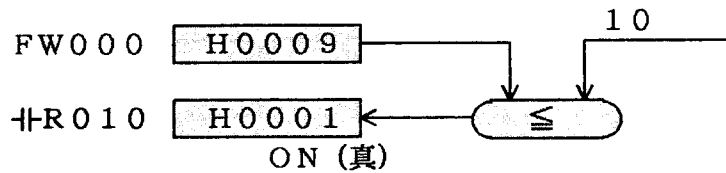
入力条件R000がOFF→ONの変化時、1回のみFW000の内容とFW001の内容の大小比較を行い、その結果をR01Aへセットします。



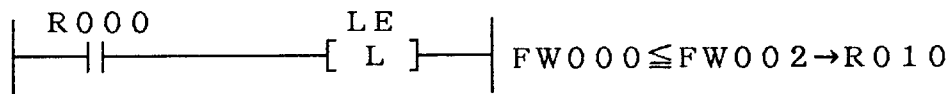
■ LE #



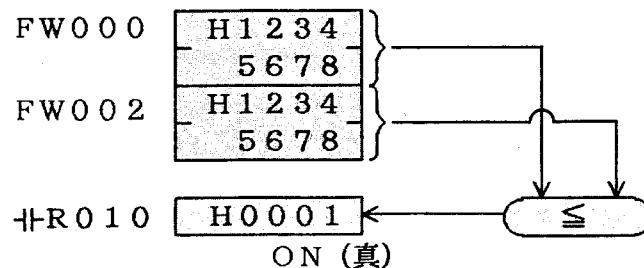
入力条件R000がONの時、FW000の内容とイミディエートデータ10の大小比較を行い、その結果をR010へセットします。



■ LE !



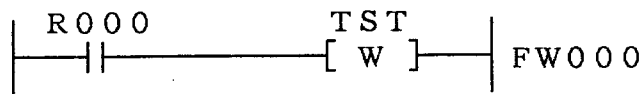
入力条件R000がONの時、FW000の内容とFW002の内容の大小比較を行い、その結果をR010へセットします。



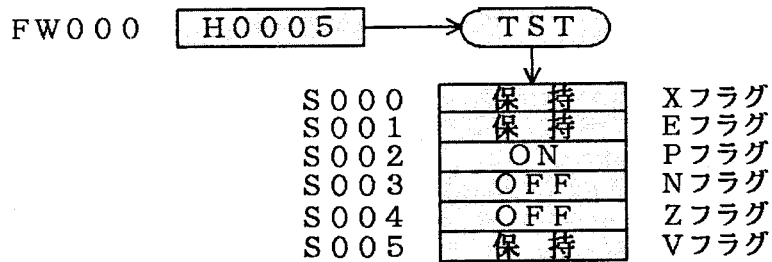
TST テスト:TEST

機能	ソースの内容の極性をテストし、ポジティブ (P) , ネガティブ (N) , ゼロ (Z) の各フラグをセットします。								
シンボル	TST	TST!		フ ラ グ					
				X	E	P	N	Z	V
処理単位	ワード	ロングワード		-	-	↑↓	↑↓	↑↓	-
シンボル 及び パラメータ	TST								
	TST!								
	S : ソース格納アドレス								
処理内容	TST	(S) > 0 : P ON (N, Z OFF) (S) = 0 : Z ON (P, N OFF) (S) < 0 : N ON (P, Z OFF)							
	TST!	(S) L > 0 : P ON (N, Z OFF) (S) L = 0 : Z ON (P, N OFF) (S) L < 0 : N ON (P, Z OFF)							
シンボル 入力手順	TST	1					<div style="border: 1px solid black; padding: 5px; text-align: center;">パラメータ入力</div>		
		2							
	TST!	1							
		2							
注意事項	(1) P, N, Z以外のフラグはすべて保持。								

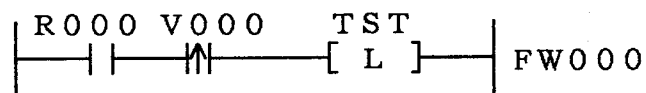
■ T S T



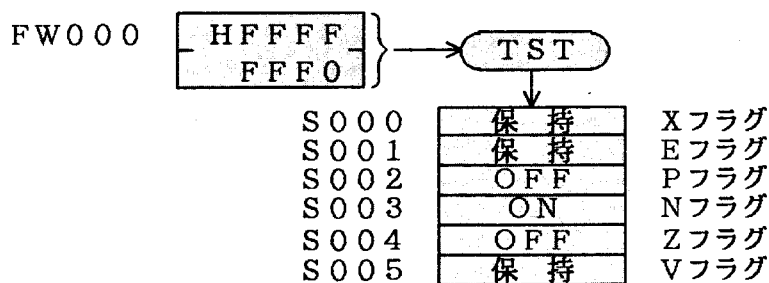
入力条件R000がONの時、FW000の内容の極性をテストし、フラグをセットします。



■ T S T !



入力条件R000がOFF→ONの変化時、FW000の内容の極性をテストし、フラグをセットします。

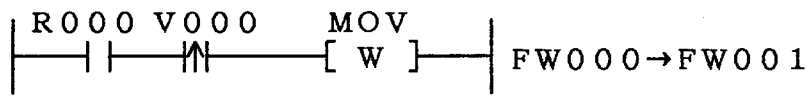


MOV 転送 = MOVE

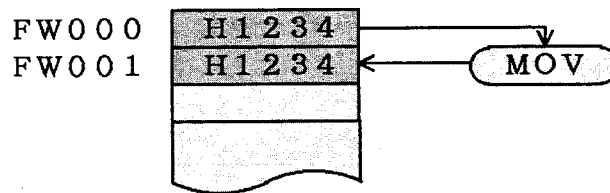
機能	ソースの内容をデスティネーションへ転送します。										
シンボル	MOV	MOV!	/	フ ラ グ							
				X	E	P	N	Z	V		
処理単位	ワード	ロングワード	/	-	-	-	-	-	-		
シンボル 及び パラメータ	MOV			[MOV W]	S → D						
	MOV!		[MOV L]	S → D							
	S : ソース格納アドレス D : デスティネーション格納アドレス										
処理内容	MOV	(S) → (D)									
	MOV!	(S) L → (D) L									
シンボル 入力手順	MOV	1	FUNC [H]	シフト	MOV	設定	パラメータ入力				
		2	FUNC [H]	M	O	V					設定
	MOV!	1	FUNC [H]	シフト	MOV	!					
		2	FUNC [H]	M	O	V					!
注意事項	(1) フラグはすべて保持。										

データ転送

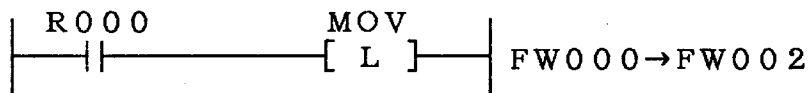
■ MOV



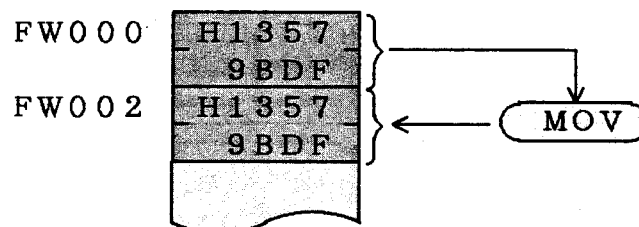
入力条件R000がOFF→ONの変化時、1回のみFW000の内容をFW001へ転送します。



■ MOV !



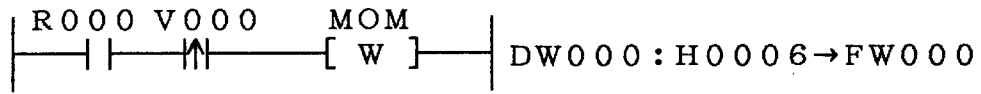
入力条件R000がONの時、FW000の内容をFW002へ転送します。



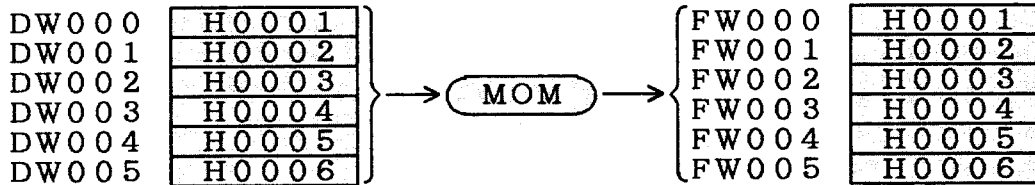
MOM 一括転送: MOVE MULTI

機能	ソースからnステップをデスティネーションからnステップに転送します。								
シンボル	MOM	MOM!	/	フ ラ グ					
				X	E	P	N	Z	V
処理単位	ワード	ロングワード	/	-	-	-	-	-	-
シンボル 及び パラメータ	MOM								
	MOM!								
S : ソース格納アドレス n : イミディエートデータ D : デスティネーション格納アドレス									
処理内容	MOM			MOM!					
シンボル 入力手順	MOM	1					パラメータ入力		
		2							
	MOM!	1							
		2							
注意事項	(1) フラグはすべて保持。 (2) ステップ数 $n \leq 0$ 又は $256 < n$ の時、未処理。								

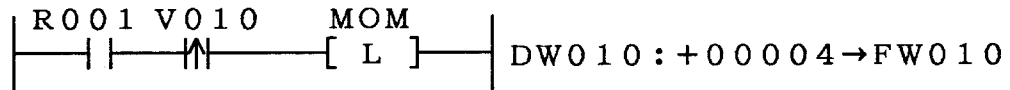
■ MOM



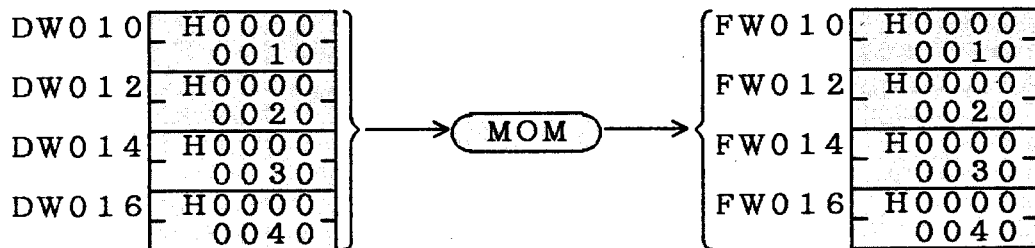
入力条件R000がOFF→ONの変化時、1回のみDW000から6ステップをFW000から6ステップに転送します。



■ MOM!



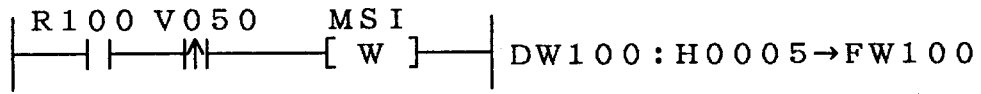
入力条件R001がOFF→ONの変化時、1回のみDW010から4ステップをFW010から4ステップに転送します。



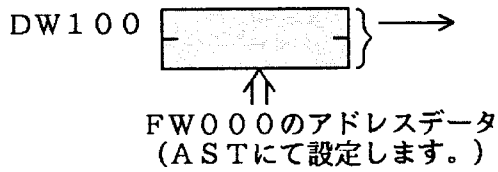
MSI ソースインダイレクト転送: MOVE MULTI SOURCE INDIRECT

機能	ソース (インダイレクトモード) から n ステップをデスティネーションから n ステップへ転送します。									
シンボル	MSI			フ ラ グ						
処理単位	ワード			X	E	P	N	Z	V	
シンボル 及び パラメータ	MSI			[MSI W]		S*: $n \rightarrow D$				
	S: ソース格納アドレス						* ロングワード指定			
	n: イミディエートデータ									
	D: デスティネーション格納アドレス									
処理内容	MSI									
シンボル 入力方法	MSI	1	FUNC []	シフト	MSI	設定	パラメータ入力			
		2	FUNC []	M	S	I				設定
注意事項	(1) フラグはすべて保持。 (2) ステップ数 $n \leq 0$ 又は $256 < n$ の時、未処理。									

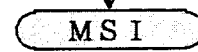
■ MSI



入力条件R100がOFF→ONの変化時、1回のみDW100に格納されたアドレスデータから5ステップをFW100から5ステップに転送します。

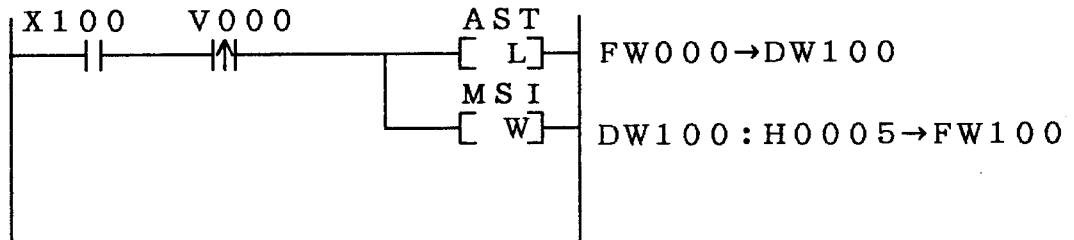


FW000	H0100
FW001	H00FF
FW002	H00FE
FW003	H00FD
FW004	H00FC



FW100	H0100
FW101	H00FF
FW102	H00FE
FW103	H00FD
FW104	H00FC

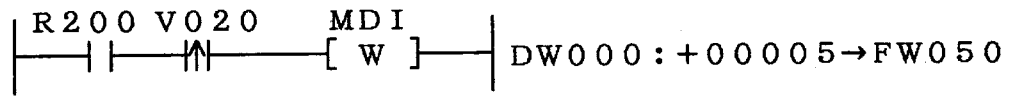
★ ASTとの組み合わせ回路



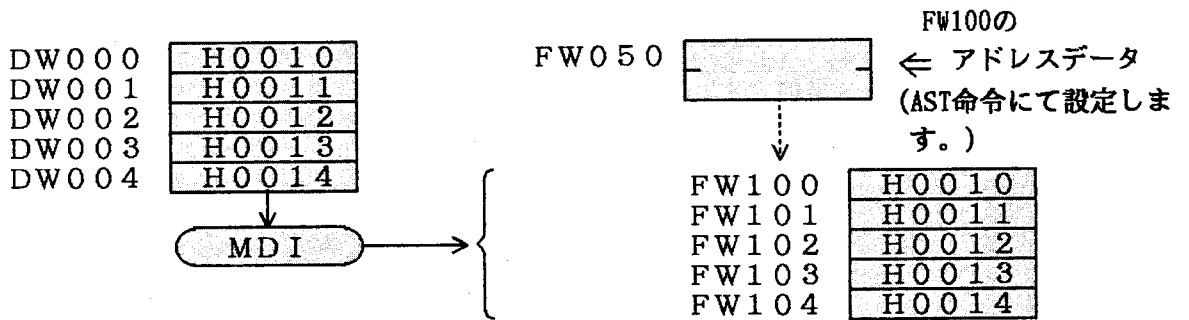
MDI | デスティネーションインダイレクト転送 = MOVE MULTI DESTINATION INDIRECT

機能	ソースからnステップをデスティネーション (インダイレクトモード) からnステップへ転送します。											
シンボル	MDI			フ ラ グ								
処理単位	ワード			X	E	P	N	Z	V			
シンボル 及び パラメータ	MDI			MDI W			S : n → D*					
	S : ソース格納アドレス						* ロングワード指定					
	n : イミディエートデータ											
	D : デスティネーション格納アドレス											
処理内容	MDI											
	S	S+1	S+2	⋮	S+(n-2)	S+(n-1)	(D)	(D)+1	(D)+2	⋮	(D)+(n-2)	(D)+(n-1)
	}						← { D (D) }					
シンボル 入力方法	MDI	1	FUNC [H]	シフト	MDI	設定	パラメータ入力					
		2	FUNC [H]	M	D	I					設定	
注意事項	(1) フラグはすべて保持。											
	(2) ステップ数 $n \leq 0$ 又は $256 < n$ の時、未処理。											

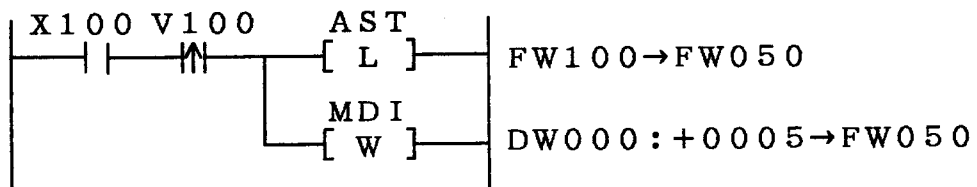
■ MDI



入力条件R200がOFF→ONの変化時、1回のみDW000から5ステップをFW050に格納されたアドレスデータから5ステップへ転送します。



★ ASTとの組み合わせ回路

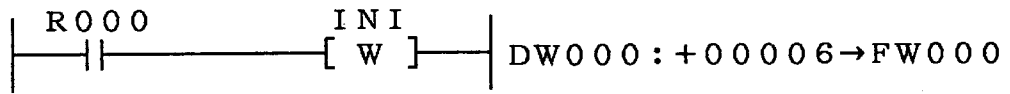


INI

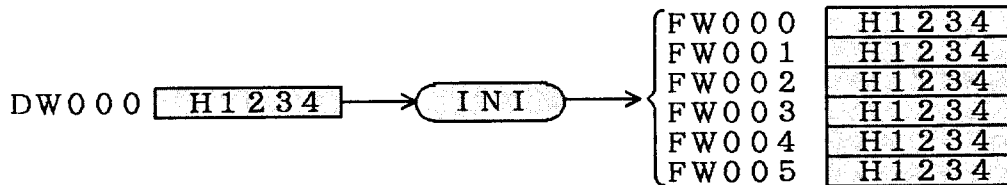
同一データ一括転送: INITIAL

機能	ソースの内容又はイミディエートデータ n_0 をデスティネーションから n ステップに転送します。									
シンボル	INI	INI#	INI!	フ ラ グ						
				X	E	P	N	Z	V	
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	-	
シンボル及びパラメータ	INI	[INI W]		$S : n \rightarrow D$						
	INI#	[INI #W]		$n_0 : n \rightarrow D$						
	INI!	[INI L]		$S : n \rightarrow D$						
<p>S: ソース格納アドレス n_0, n: イミディエートデータ D: デスティネーション格納アドレス</p>										
処理内容	INI			INI#			INI!			
シンボル入力手順	INI	1	FUNC [] シフト INI 設定				パラメータ入力			
		2	FUNC [] I N I 設定							
	INI#	1	FUNC [] シフト INI #							
		2	FUNC [] I N I #							
	INI!	1	FUNC [] シフト INI !							
		2	FUNC [] I N I !							
注意事項	<p>(1) フラグはすべて保持。 (2) ステップ数 $n \leq 0$ 又は $256 < n$ の時、未処理。</p>									

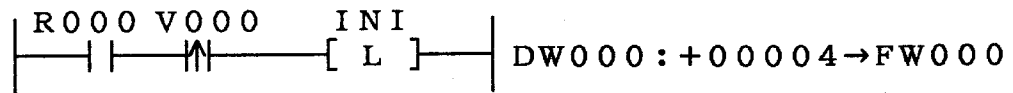
■ I N I



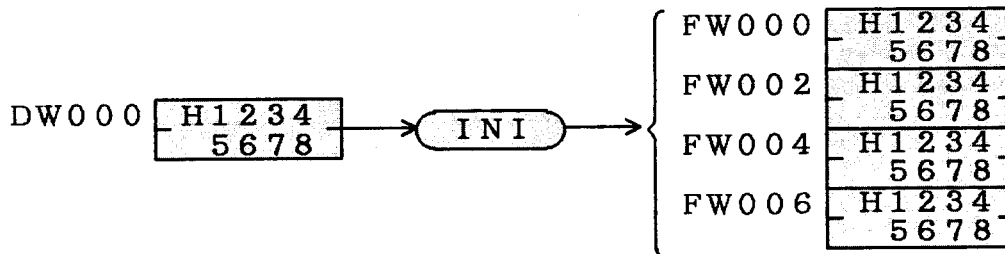
入力条件R000がONの時、DW000の内容をFW000から6ステップに転送します。



■ I N I !



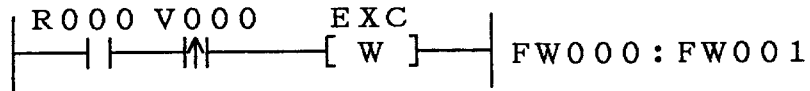
入力条件R000がOFF→ONの変化時、DW000の内容をFW000から4ステップに転送します。



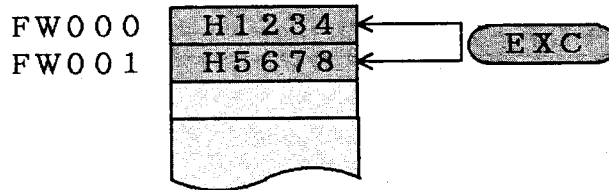
EXC 交換: EXCHANGE

機能	ソースの内容とデスティネーションの内容の交換を行います。									
シンボル	EXC	EXC!	/	フ ラ グ						
				X	E	P	N	Z	V	
処理単位	ワード	ロングワード	/	-	-	-	-	-	-	
シンボル 及び パラメータ	EXC				S : D					
	EXC!				S : D					
	S : ソース格納アドレス D : デスティネーション格納アドレス									
処理内容	EXC	(S) ↔ (D)								
	EXC!	(S) L ↔ (D) L								
シンボル 入力手順	EXC	1					<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>			
		2								
	EXC!	1								
		2								
注意事項	(1) フラグはすべて保持。									

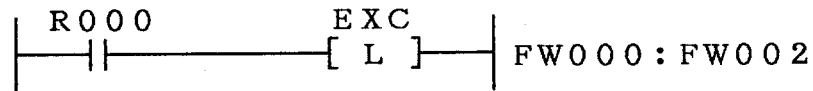
■ EXC



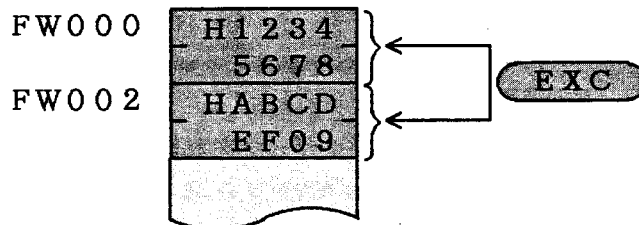
入力条件R000がOFF→ONの変化時、1回のみFW000の内容とFW001の内容の交換を行います。



■ EXC!



入力条件R000がONの時、FW000の内容とFW002の内容の交換を行います。

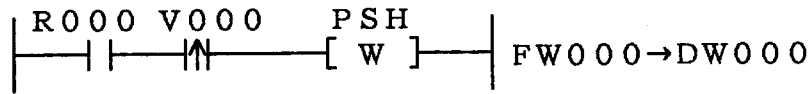


★ EXC!では、アドレスが重複しないようにしてください。

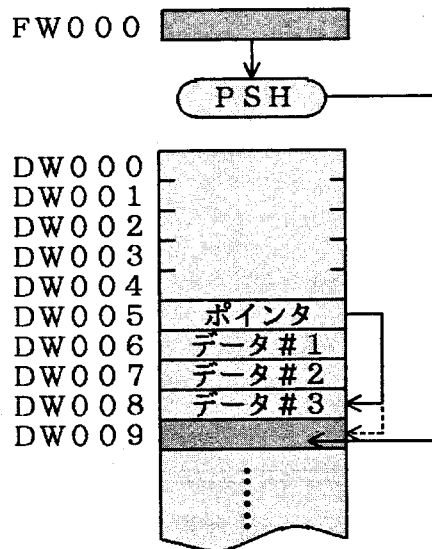
PSH FIFO書込: FIFO PUSH

機能	ソースの内容をFIFOテーブルへプッシュします。																													
シンボル	PSH	フ ラ グ				X	E	P	N	Z	V																			
処理単位	ワード					-	-	-	-	-	-	-																		
シンボル 及び パラメータ	PSH																													
パラメータ	S : ソース格納アドレス TB : FIFOテーブル先頭アドレス																													
処理内容	データの動作						FIFOデータテーブルの構成 <table border="1" style="display: inline-table; vertical-align: top;"> <tr><td>FIFOトップ+ 0</td><td>n (データサイズ)</td></tr> <tr><td>+ 1</td><td>ZEROフラグ</td></tr> <tr><td>+ 2</td><td>アドレス</td></tr> <tr><td>+ 3</td><td>FULLフラグ</td></tr> <tr><td>+ 4</td><td>アドレス</td></tr> <tr><td>+ 5</td><td>ポインタ</td></tr> <tr><td>+ 6</td><td>データ# 1</td></tr> <tr><td rowspan="2">}</td><td rowspan="2">データサイズで指定したデータ格納エリア</td></tr> <tr><td>データ# n</td></tr> <tr><td>n+5</td><td></td></tr> </table>					FIFOトップ+ 0	n (データサイズ)	+ 1	ZEROフラグ	+ 2	アドレス	+ 3	FULLフラグ	+ 4	アドレス	+ 5	ポインタ	+ 6	データ# 1	}	データサイズで指定したデータ格納エリア	データ# n	n+5	
FIFOトップ+ 0	n (データサイズ)																													
+ 1	ZEROフラグ																													
+ 2	アドレス																													
+ 3	FULLフラグ																													
+ 4	アドレス																													
+ 5	ポインタ																													
+ 6	データ# 1																													
}	データサイズで指定したデータ格納エリア																													
		データ# n																												
n+5																														
シンボル 入力方法	PSH	1	FUNC [H]	シフト	PSH	設定	パラメータ入力																							
		2	FUNC [H]	P	S	H						設定																		
注意事項	(1) ZEROフラグはOFFされます。 (2) プッシュ処理前にポインタがnであれば、FULLフラグをONし、プッシュ未処理とします。又、プッシュ処理後ポインタをインクリメントしてnになった場合もFULLフラグON。それ以外の場合、FULLフラグOFF。 (3) データサイズ $n \leq 0$ 又は $256 < n$ の場合、未処理。 (4) ポインタ < 0 又は $n < \text{ポインタ}$ の場合、未処理。 (5) フラグ (X, E, P, N, Z, V) はすべて保持。																													

■ PSH

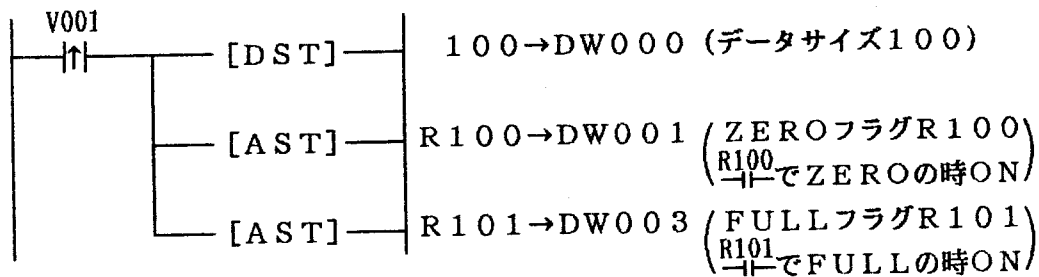


入力条件R000がOFF→ONの変化時、1回のみFW000の内容をDW000よりのFIFOデータテーブルへセットします。



- データサイズnはデータセットDST等で行います。
- ZEROフラグ, FULLフラグのアドレスはアドレスセットASTで行います。

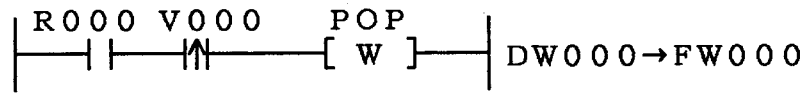
<例>



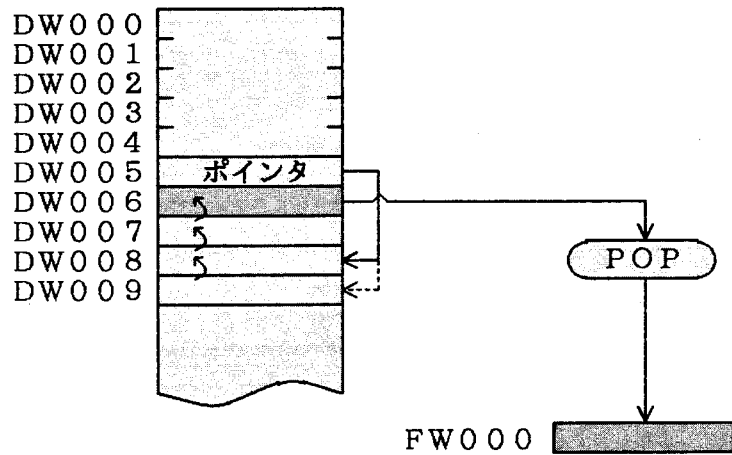
POP FIFO読出:FIFO POP

機能	FIFOテーブルのポップ処理を行い、ポップデータをデスティネーションへ格納します。																								
シンボル	POP	フ ラ グ				X	E	P	N	Z	V														
処理単位	ワード					-	-	-	-	-	-	-													
シンボル及びパラメータ	POP																								
処理内容	<p>データの動作</p> <p>FIFOテーブルの構成</p> <table border="1"> <tr><td>FIFOトップ+ 0</td><td>n(データサイズ)</td></tr> <tr><td>+ 1</td><td>ZEROフラグ</td></tr> <tr><td>+ 2</td><td>アドレス</td></tr> <tr><td>+ 3</td><td>FULLフラグ</td></tr> <tr><td>+ 4</td><td>アドレス</td></tr> <tr><td>+ 5</td><td>ポインタ</td></tr> <tr><td>+ 6</td><td>データ# 1</td></tr> <tr><td rowspan="2">}</td><td rowspan="2">データサイズで指定したデータ格納エリア</td></tr> <tr><td>データ# n</td></tr> <tr><td>n+5</td><td></td></tr> </table>						FIFOトップ+ 0	n(データサイズ)	+ 1	ZEROフラグ	+ 2	アドレス	+ 3	FULLフラグ	+ 4	アドレス	+ 5	ポインタ	+ 6	データ# 1	}	データサイズで指定したデータ格納エリア	データ# n	n+5	
FIFOトップ+ 0	n(データサイズ)																								
+ 1	ZEROフラグ																								
+ 2	アドレス																								
+ 3	FULLフラグ																								
+ 4	アドレス																								
+ 5	ポインタ																								
+ 6	データ# 1																								
}	データサイズで指定したデータ格納エリア																								
		データ# n																							
n+5																									
シンボル入力方法	POP	<p>1 FUNC [H] シフト POP 設定</p> <p>2 FUNC [H] P O P 設定</p>				パラメータ入力																			
注意事項	<p>(1) FULLフラグはOFFされます。</p> <p>(2) ポップ処理前にポインタが0であれば、ZEROフラグをONし、ポップ未処理とします。又、ポップ処理後ポインタをデクリメントして0になった場合もZEROフラグON。それ以外の場合、ZEROフラグOFF。</p> <p>(3) データサイズ$n \leq 0$又は$256 < n$の場合、未処理。</p> <p>(4) ポインタ< 0又は$n < \text{ポインタ}$の場合、未処理。</p> <p>(5) フラグ(X, E, P, N, Z, V)はすべて保持。</p>																								





■ POP



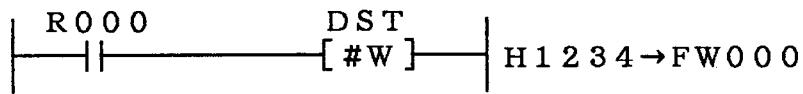
入力条件R000がOFF→ONの変化時、1回のみDW000よりのFIFOデータテーブルからデータを取込み、FW000へ格納します。



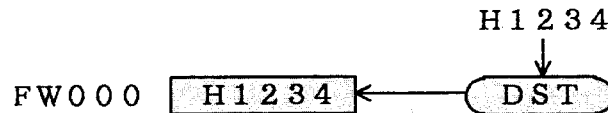
DST データセット: DATA SET

機能	イミディエートデータをデスティネーションへ格納します。									
シンボル	DST#	DST#!	/	フ ラ グ						
処理単位	ワード	ロングワード		X	E	P	N	Z	V	
シンボル 及び パラメータ	DST# [DST #W]			n → D					
	DST#! [DST #L]			n → D					
	n: イミディエートデータ D: デスティネーション格納アドレス									
処理内容	DST#	n → (D)								
	DST#!	nL → (D) L								
シンボル 入力手順	DST#	1					<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;">パラメータ入力</div>			
		2								
	DST#!	1								
		2								
注意事項	(1) フラグはすべて保持。									

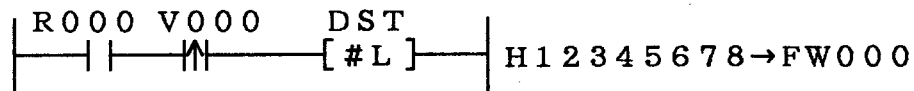
■ DST



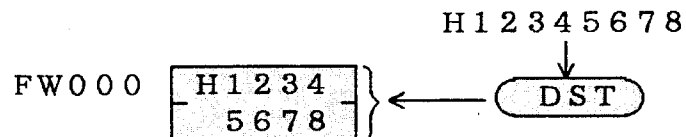
入力条件R000がONの時、イミディエートデータH1234をFW000へ格納します。



■ DST # !



入力条件R000がOFF→ONの変化時、1回のみイミディエートデータH12345678をFW000へ格納します。

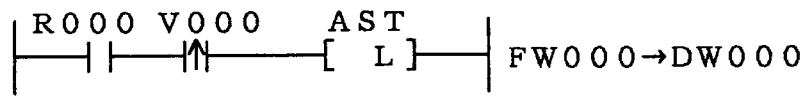


- イミディエートデータの10進入力の場合、データ範囲は
 $-999999999 \leq n \leq +999999999$
 となります。

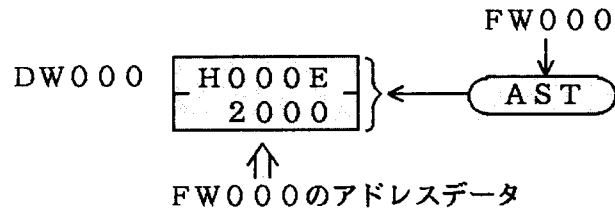
AST アドレスセット: ADDRESS SET

機能	ソースのアドレスデータをデスティネーションへ格納します。								
シンボル	AST			フ ラ グ					
				X	E	P	N	Z	V
処理単位	ロングワード			-	-	-	-	-	-
シンボル 及び パラメータ	AST								
		S: ソース格納アドレス D: デスティネーション格納アドレス							
処理内容	AST	S (アドレスデータ) → (D) L							
	AST	1					<div style="border: 1px solid black; padding: 5px; text-align: center;">パラメータ入力</div>		
		2							
注意事項	(1) フラグはすべて保持。								

■ AST



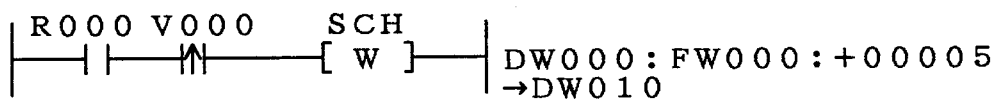
入力条件R000がOFF→ONの変化時、1回のみFW000のアドレスデータをDW000へ格納します。



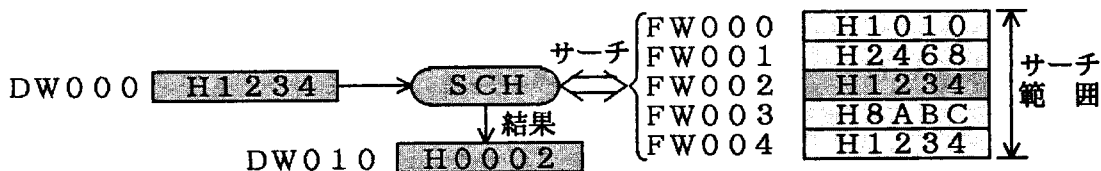
SCH サーチ:SEARCH

機能	ソースの内容がデスティネーションから指定エリア（サーチステップ数m）内に存在するかどうか探し、デスティネーションから探し出すまでのステップ数をリザルトに格納します。									
シンボル	SCH	SCH!	フ ラ グ	X	E	P	N	Z	V	
処理単位	ワード	ロングワード		-	-	-	-	-	-	
シンボル 及び パラメータ	SCH				$\underline{S} : \underline{D} : \underline{m}$ $\rightarrow \underline{R}$					
	SCH!				$\underline{S} : \underline{D} : \underline{m}$ $\rightarrow \underline{R}^*$					
<p>S: ソース格納アドレス * ワード指定 D: デスティネーション格納アドレス m: イミディエートデータ R: リザルト (演算結果) 格納するアドレス</p>										
処理内容	SCH				SCH!					
<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> </div> <div style="text-align: center;"> </div> </div>										
シンボル 入力手順	SCH	1				<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>				
		2								
	SCH!	1								
		2								
注意事項	<p>(1) フラグはすべて保持。 (2) 一致データは、サーチ範囲の先頭から最初に見つけたもののみ有効。 (3) サーチ範囲に一致データが無い時、リザルトには-1 (H F F F F) がセットされます。 (4) サーチステップ数 $m \leq 0$ 又は $256 < m$ の時、未処理。</p>									

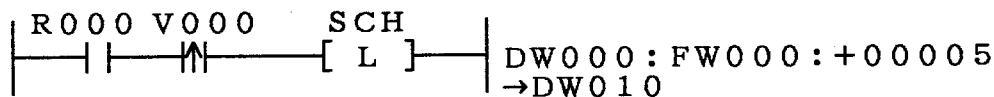
■ SCH



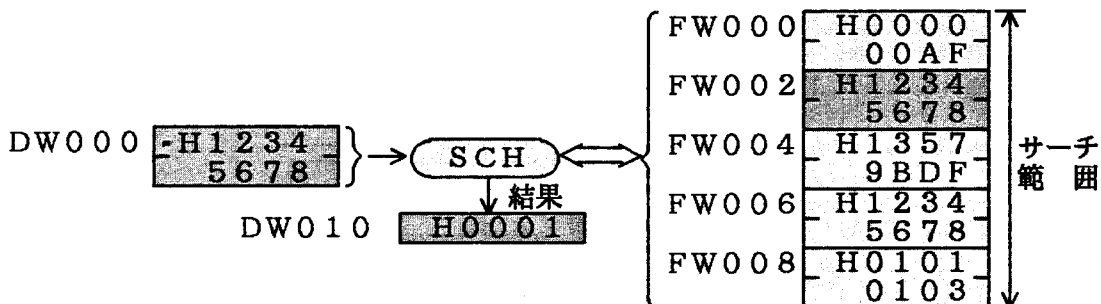
入力条件R000がOFF→ONの変化時、1回のみDW000の内容と同一データがFW000から5ステップ内に存在するかどうかが探し、その結果をDW010へ格納します。



■ SCH!



入力条件R000がOFF→ONの変化時、1回のみDW000の内容と同一データがFW000から5ステップ内に存在するかどうかが探し、その結果をDW010へ格納します。

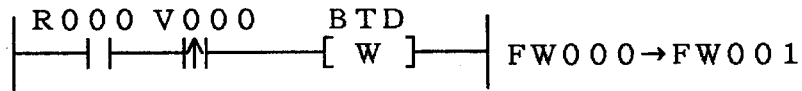


B T D BINARY→BCD変換

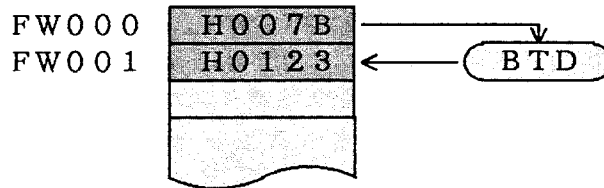
データ変換

機能	ソースの内容をBIN (バイナリー) からBCDへ換換し、結果をリザルトへ格納します。BCD: 2進10進数 (Binary Coded Decimal)													
シンボル	BTD	BTD!	/	フ ラ グ										
				X	E	P	N	Z	V					
処理単位	ワード	ロングワード	/	-	↕	-	-	-	-	↕				
シンボル 及び パラメータ	BTD									<u>S</u> → <u>R</u>				
	BTD!									<u>S</u> → <u>R</u>				
	S: ソース格納アドレス R: リザルト (演算結果) を格納するアドレス													
処理内容	BTD	(S) <u>B I N → B C D</u> (R)												
	BTD!	(S) L <u>B I N → B C D</u> (R) L												
シンボル 入力手順	BTD	1					<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;">パラメータ入力</div>							
		2												
	BTD!	1												
		2												
フラグの 設定	E: (S) < 0の時1、それ以外0 V: (ワード時) ; (S) > 9999の時1、それ以外0 (ロングワード時) ; (S) > 99999999の時1、それ以外0 他: 保持													
注意事項	(1) (S) < 0の時エラーフラグ (E) がON (オーバーフローフラグ (V) OFF) し、未処理。(リザルトは変化しません。) (2) オーバーフロー発生時、リザルトには下記フルスケール値がセットされます。													
	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td style="padding: 2px;">ワード</td><td style="padding: 2px;">ロングワード</td></tr> <tr><td style="padding: 2px;">H9999</td><td style="padding: 2px;">H99999999</td></tr> </table>		ワード	ロングワード	H9999	H99999999								
ワード	ロングワード													
H9999	H99999999													

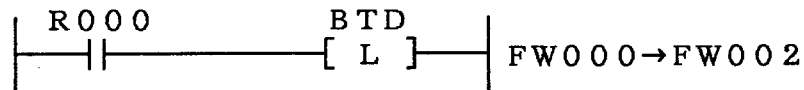
■ B T D



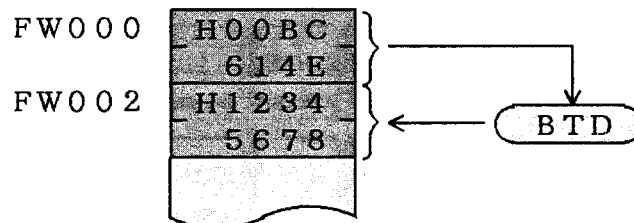
入力条件R000がOFF→ONの変化時、1回のみFW000の内容をBINからBCDへ変換し、その結果をFW001へ格納します。



■ B T D !



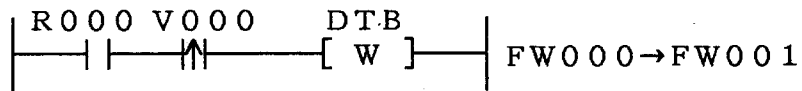
入力条件R000がONの時、FW000の内容をBINからBCDへ変換し、その結果をFW002へ格納します。



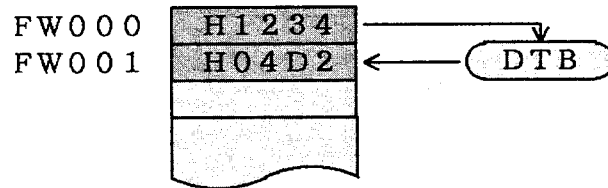
DTB BCD→BINARY変換

機能	ソースの内容をBCDからBIN (バイナリー) へ変換し、結果をリザルトへ格納します。										
シンボル	DTB	DTB!	フ ラ グ			X	E	P	N	Z	V
処理単位	ワード	ロングワード				-	↕	-	-	-	-
シンボル 及び パラメータ	DTB										
	DTB!										
S : ソース格納アドレス R : リザルト (演算結果) を格納するアドレス											
処理内容	DTB	(S) <u>BCD→BIN</u> (R)									
	DTB!	(S) L <u>BCD→BIN</u> (R) L									
シンボル 入力手順	DTB	1									
		2									
	DTB!	1									
		2									
<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>											
フラグの 設定	E : ソースデータ内のあるディジット (4 bit) がHA~HFの場合1、それ以外0 他 : 保持										
注意事項	(1) 上記エラーフラグ (E) がONした場合、未処理。(リザルトは変化しません。)										
	★ ディジットについて 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 ディジット ディジット ディジット ディジット										

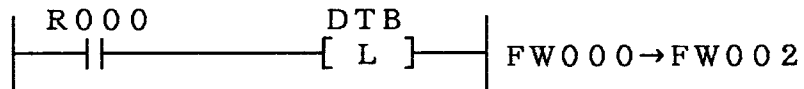
■ DTB



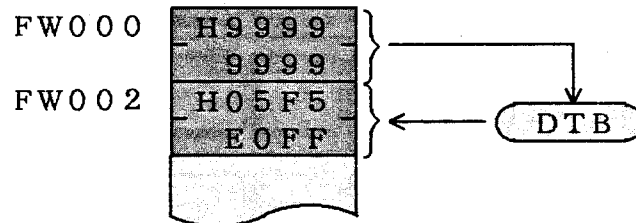
入力条件R000がOFF→ONの変化時、1回のみFW000の内容をBCDからBINへ変換し、その結果をFW001へ格納します。



■ DTB !



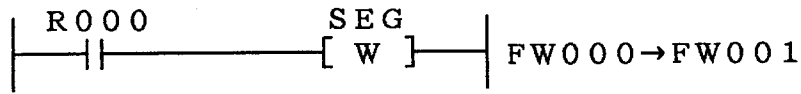
入力条件R000がONの時、FW000の内容をBCDからBINへ変換し、その結果をFW002へ格納します。



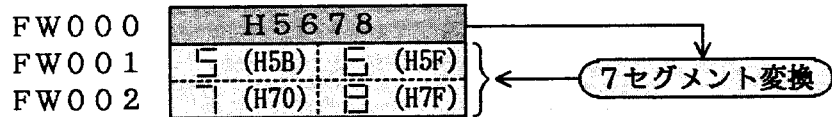
SEG BINARY → セグメント変換

機能	ソースの内容又はイミディエートデータをBIN (バイナリー) から7セグメントデータへ変換し、リザルトへ格納します。																																																											
シンボル	SEG	SEG#	SEG!	フ ラ グ																																																								
				X	E	P	N	Z	V																																																			
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	-																																																			
シンボル及びパラメータ	SEG		[SEG W]	S → R																																																								
	SEG#		[SEG #W]	n → R																																																								
	SEG!		[SEG L]	S → R																																																								
	S : ソース格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス																																																											
処理内容	SEG	(S)	BIN → 7セグメント → (R, R+1)																																																									
	SEG#	n	BIN → 7セグメント → (R, R+1)																																																									
	SEG!	(S) L	BIN → 7セグメント → ((R) L, (R+2) L)																																																									
シンボル入力手順	SEG	1	FUNC [H]	シフト	SEG	設定	パラメータ入力																																																					
		2	FUNC [H]	S	E	G		設定																																																				
	SEG#	1	FUNC [H]	シフト	SEG	#																																																						
		2	FUNC [H]	S	E	G		#																																																				
	SEG!	1	FUNC [H]	シフト	SEG	!																																																						
		2	FUNC [H]	S	E	G		!																																																				
セグメントデータ構成	(MSB) 0 1 2 3 4 5 6 7 ⇒																																																											
(1) 該当ビットがONの時表示有 (2) 7セグメント対応表																																																												
<table border="1" style="width: 100%; text-align: center;"> <tr> <th>No</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> <th>8</th> <th>9</th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> </tr> <tr> <th>表示</th> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> <td>8</td> <td>9</td> <td>A</td> <td>B</td> <td>C</td> <td>D</td> <td>E</td> <td>F</td> </tr> <tr> <th>アドレス</th> <td>H7E</td> <td>H30</td> <td>H6D</td> <td>H79</td> <td>H33</td> <td>H5B</td> <td>H5F</td> <td>H70</td> <td>H7F</td> <td>H7B</td> <td>H77</td> <td>H1F</td> <td>H4E</td> <td>H3D</td> <td>H4F</td> <td>H47</td> </tr> </table>										No	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	表示	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	アドレス	H7E	H30	H6D	H79	H33	H5B	H5F	H70	H7F	H7B	H77	H1F	H4E	H3D	H4F	H47
No	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																																												
表示	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																																												
アドレス	H7E	H30	H6D	H79	H33	H5B	H5F	H70	H7F	H7B	H77	H1F	H4E	H3D	H4F	H47																																												
注意事項	(1) フラグはすべて保持。																																																											

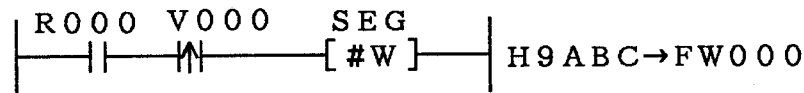
■ SEG



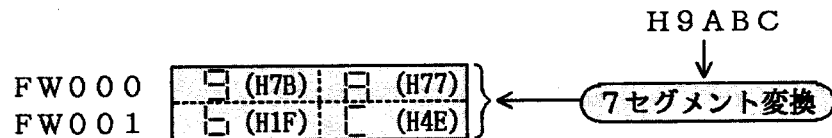
入力条件R000がONの時、FW000の内容をバイナリーデータから4文字の7セグメントデータに変換し、FW001へ格納します。



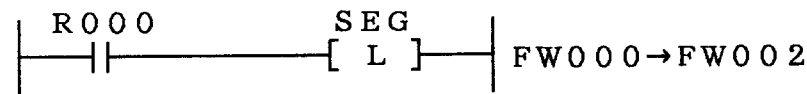
■ SEG #



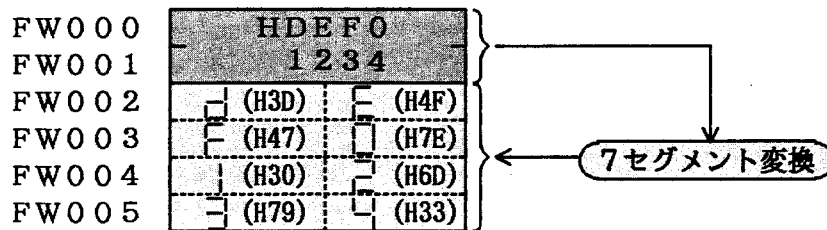
入力条件R000がOFF→ONの変化時、1回のみバイナリーデータH9ABCを4文字の7セグメントデータに変換し、FW000へ格納します。



■ SEG !



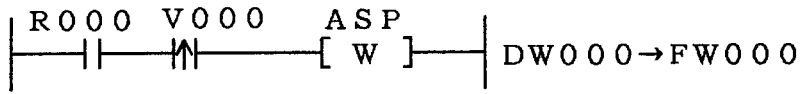
入力条件R000がONの時、FW000の内容をバイナリーデータから8文字の7セグメントデータに変換し、FW002へ格納します。



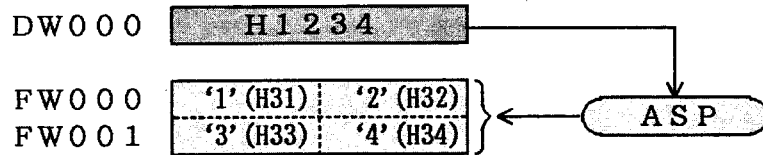
ASP BINARY→ASCII変換バックモード

機能	ソースの内容をバイナリーデータからASCIIデータへ変換し、バックモードで、結果をリザルトへ格納します。ASCII:American Standard Code for Information Interchange																																											
シンボル	ASP	/		フ ラ グ																																								
				X	E	P	N	Z	V																																			
処理単位	ワード	/		-	-	-	-	-	-																																			
シンボル及びパラメータ	ASP	[ASP W]			S → R																																							
	S: ソース格納アドレス R: リザルト (演算結果) を格納するアドレス																																											
処理内容	ASP	(S) BIN→ASCII バックモード → (R, R+1)																																										
シンボル入力方法	ASP	1	FUNC []	シフト	ASP	設定	パラメータ入力																																					
		2	FUNC []	A	S	P					設定																																	
ASCII対応表	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td>バイナリー</td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td> </tr> <tr> <td>ASCII</td> <td>H30</td><td>H31</td><td>H32</td><td>H33</td><td>H34</td><td>H35</td><td>H36</td><td>H37</td><td>H38</td><td>H39</td> </tr> </table> <table border="1" style="width: 60%; margin: auto; border-collapse: collapse; text-align: center;"> <tr> <td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td> </tr> <tr> <td>H41</td><td>H42</td><td>H43</td><td>H44</td><td>H45</td><td>H46</td> </tr> </table>										バイナリー	0	1	2	3	4	5	6	7	8	9	ASCII	H30	H31	H32	H33	H34	H35	H36	H37	H38	H39	A	B	C	D	E	F	H41	H42	H43	H44	H45	H46
バイナリー	0	1	2	3	4	5	6	7	8	9																																		
ASCII	H30	H31	H32	H33	H34	H35	H36	H37	H38	H39																																		
A	B	C	D	E	F																																							
H41	H42	H43	H44	H45	H46																																							
注意事項	<p>(1) フラグはすべて保持。</p> <p>(2) 結果は2バイトずつR (上位), R+1 (下位) へ格納されます。</p>																																											


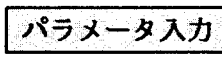
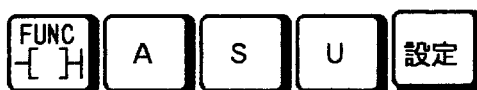
■ ASP



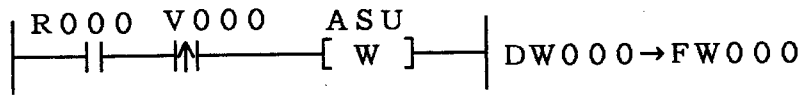
入力条件R000がOFF→ONの変化時、1回のみDW000の内容をバイナリーデータからASCIIデータへ変換して、FW000へ格納します。



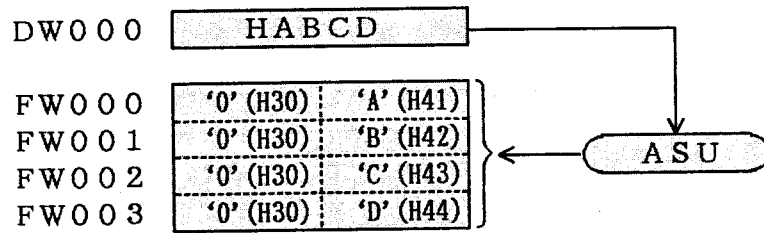
ASU BINARY→ASCII変換アンパックモード

機能	ソースの内容をバイナリーデータからASCIIデータへ変換し、アンパックモードで、結果をリザルトへ格納します。																																							
シンボル	ASU	フ ラ グ																																						
		X	E	P	N	Z V																																		
処理単位	ワード	-	-	-	-	-																																		
シンボル 及び パラメータ	ASU	ASU [W]				S → R																																		
	S : ソース格納アドレス R : リザルト (演算結果) を格納するアドレス																																							
処理内容	ASU	(S) BIN→ASCII アンパックモード → (R, R+1, R+2, R+3)																																						
シンボル 入力方法	ASU	1																																						
		2																																						
ASCII 対応表	<table border="1" style="width: 100%; text-align: center;"> <tr> <td>バイナリー</td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td> </tr> <tr> <td>ASCII</td> <td>H30</td><td>H31</td><td>H32</td><td>H33</td><td>H34</td><td>H35</td><td>H36</td><td>H37</td><td>H38</td><td>H39</td> </tr> </table> <table border="1" style="width: 50%; text-align: center; margin-left: auto; margin-right: auto;"> <tr> <td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td> </tr> <tr> <td>H41</td><td>H42</td><td>H43</td><td>H44</td><td>H45</td><td>H46</td> </tr> </table>						バイナリー	0	1	2	3	4	5	6	7	8	9	ASCII	H30	H31	H32	H33	H34	H35	H36	H37	H38	H39	A	B	C	D	E	F	H41	H42	H43	H44	H45	H46
バイナリー	0	1	2	3	4	5	6	7	8	9																														
ASCII	H30	H31	H32	H33	H34	H35	H36	H37	H38	H39																														
A	B	C	D	E	F																																			
H41	H42	H43	H44	H45	H46																																			
注意事項	<p>(1) フラグはすべて保持。</p> <p>(2) 結果は1バイトずつ上位データからR, R+1, R+2, R+3の下位バイトへ格納します。R~R+3の上位バイトはASCIIの'0' (H30) をセットします。</p>																																							

■ ASU



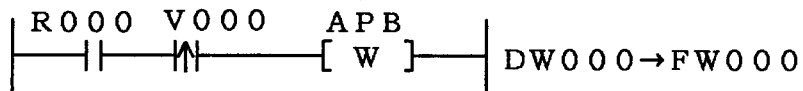
入力条件R000がOFF→ONの変化時、1回のみDW000の内容をバイナリーデータからASCIIデータへ変換して、アンパックモードでFW000へ格納します。



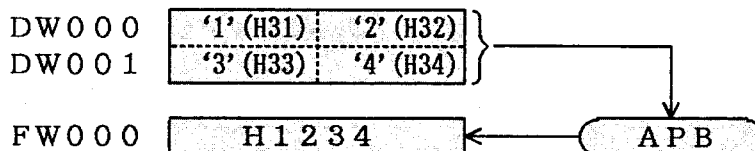
APB ASCII→BINARY変換バックモード

機能	ソースの内容をASCIIデータ（バックモード）からバイナリーデータへ変換し、結果をリザルトへ格納します。																																											
シンボル	APB			フ ラ グ																																								
処理単位	ワード			X	E	P	N	Z	V																																			
				-	↑ ↓	-	-	-	-	-																																		
シンボル 及び パラメータ	APB									S → R																																		
処理内容	APU	(S, S+1) $\xrightarrow[\text{バックモード}]{\text{ASCII} \rightarrow \text{BIN}}$ (R)																																										
シンボル 入力方法	APB	1																																										
		2																																										
ASCII 対応表	<table border="1"> <tr> <td>バイナリー</td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td> </tr> <tr> <td>ASCII</td> <td>H30</td><td>H31</td><td>H32</td><td>H33</td><td>H34</td><td>H35</td><td>H36</td><td>H37</td><td>H38</td><td>H39</td> </tr> </table> <table border="1"> <tr> <td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td> </tr> <tr> <td>H41</td><td>H42</td><td>H43</td><td>H44</td><td>H45</td><td>H46</td> </tr> </table>										バイナリー	0	1	2	3	4	5	6	7	8	9	ASCII	H30	H31	H32	H33	H34	H35	H36	H37	H38	H39	A	B	C	D	E	F	H41	H42	H43	H44	H45	H46
バイナリー	0	1	2	3	4	5	6	7	8	9																																		
ASCII	H30	H31	H32	H33	H34	H35	H36	H37	H38	H39																																		
A	B	C	D	E	F																																							
H41	H42	H43	H44	H45	H46																																							
フラグの 設定	E: H30~H39, H41~H46以外のデータ検出時1、それ以外0 他: 保持																																											
注意事項	(1) エラーフラグ (E) がONした時、未処理。(リザルトは変化しません。)																																											

■ APB



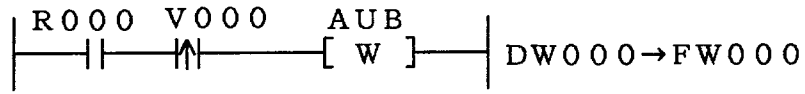
入力条件R000がOFF→ONの変化時、1回のみDW000の内容をASCIIデータ（パックモード）からバイナリーデータへ変換し、その結果をリザルトへ格納します。



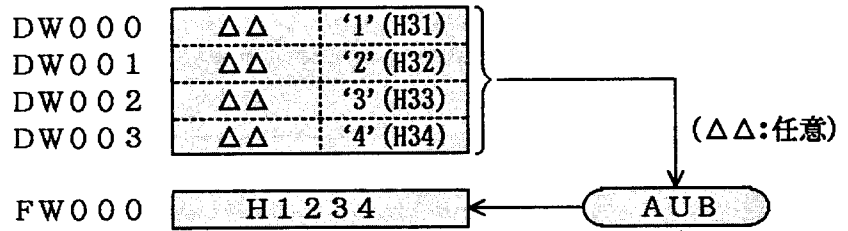
AUB ASCII→BINARY変換アンパックモード

機能	ソースの内容をASCIIデータ（アンパックモード）からバイナリーデータへ変換し、結果をリザルトへ格納します。																																											
シンボル	AUB			フ ラ グ																																								
				X	E	P	N	Z	V																																			
処理単位	ワード			-	↑ ↓	-	-	-	-																																			
シンボル 及び パラメータ	AUB																																											
処理内容	AUB	(S, S+1, S+2, S+3) $\xrightarrow[\text{アンパックモード}]{\text{ASCII} \rightarrow \text{BIN}}$ (R)																																										
シンボル 入力方法	AUB	1																																										
		2																																										
ASCII 対応表	<table border="1" style="width: 100%; text-align: center;"> <tr> <td>バイナリー</td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td> </tr> <tr> <td>ASCII</td> <td>H30</td><td>H31</td><td>H32</td><td>H33</td><td>H34</td><td>H35</td><td>H36</td><td>H37</td><td>H38</td><td>H39</td> </tr> </table> <table border="1" style="width: 60%; margin: auto; text-align: center;"> <tr> <td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td> </tr> <tr> <td>H41</td><td>H42</td><td>H43</td><td>H44</td><td>H45</td><td>H46</td> </tr> </table>										バイナリー	0	1	2	3	4	5	6	7	8	9	ASCII	H30	H31	H32	H33	H34	H35	H36	H37	H38	H39	A	B	C	D	E	F	H41	H42	H43	H44	H45	H46
バイナリー	0	1	2	3	4	5	6	7	8	9																																		
ASCII	H30	H31	H32	H33	H34	H35	H36	H37	H38	H39																																		
A	B	C	D	E	F																																							
H41	H42	H43	H44	H45	H46																																							
フラグの 設定	E: H30~H39, H41~H46以外のデータ検出時1、それ以外0 他: 保持																																											
注意事項	(1) エラーフラグ (E) がONした時、未処理。(リザルトは変化しません。) (2) S~S+3の上位バイトは任意。																																											

■ AUB



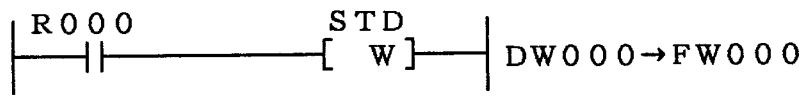
入力条件R000がOFF→ONの変化時、1回のみDW000の内容をASCIIデータ（アンパックモード）からバイナリーデータへ変換して、その結果をFW000へ格納します。



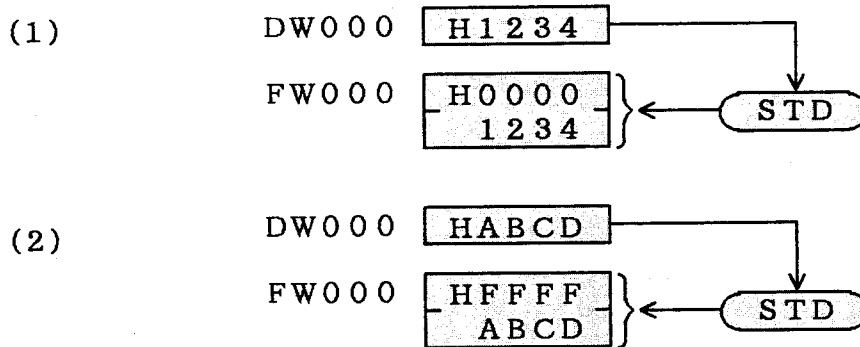
STD SINGLE→DOUBLE 変換

機能	ソースの内容をワードデータからロングワードデータに符号拡張してリザルトへ格納します。									
シンボル	STD	/		フ ラ グ						
処理単位	ワード	/		X	E	P	N	Z	V	
シンボル 及び パラメータ	STD	[STD W]			S → R*					
処理内容	STD	(S) → (R) L								
シンボル 入力方法	STD	1	FUNC []	シフト	STD	設定	パラメータ入力			
		2	FUNC []	S	T	D				
注意事項	(1) フラグはすべて保持。									

■ STD



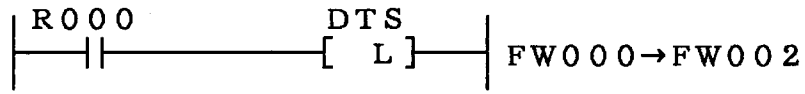
入力条件R000がONの時、DW000の内容をワードデータからロングワードデータに符号拡張して、その結果をFW000へ格納します。



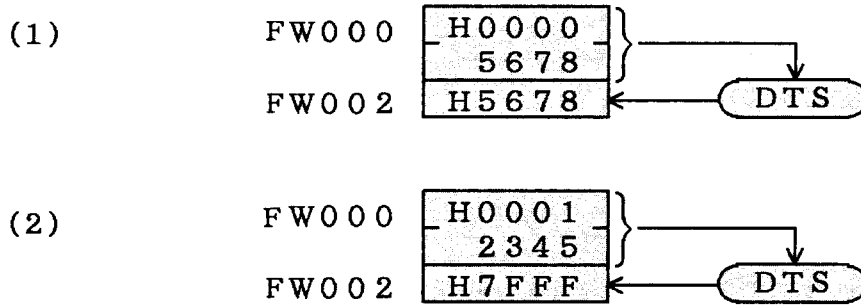
D T S DOUBLE→SINGLE 変換

機能	ソースの内容のロングワードデータをワードデータとしてリザルトへ格納します。									
シンボル	D T S			フ ラ グ						
処理単位	ロングワード			X	E	P	N	Z	V	
				-	-	-	-	-	↑ ↓	
シンボル 及び パラメータ	D T S				S → R*			* ワード指定		
処理内容	D T S	(S) L → (R)								
シンボル 入力方法	D T S	1					<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>			
		2								
フラグの 設定	V: (S) < -32768又は32767 < (S) の時1、それ以外0 他: 保持									
注意事項	(1) オーバーフロー発生時、リザルトには下記フルスケール値がセットされます。									
	(S) L > 32767の時				H7FFF					
	(S) L < -32768の時				H8000					

■ D T S



入力条件R000がONの時、FW000の内容のロングワードデータをワードデータとしてFW002へ格納します。

















この時VフラグがONします。

Vフラグ

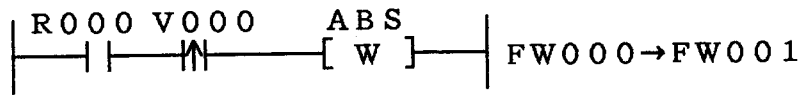
ON

 S005

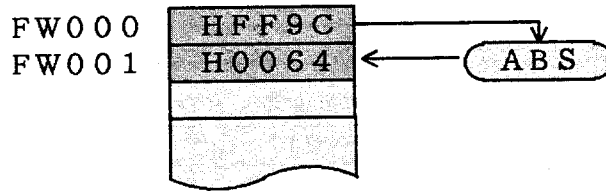
ABS 絶対値: ABSOLUTE VALUE

機能	ソースの内容の絶対値をリザルトへ格納します。											
シンボル	ABS	ABS!	/	フ ラ グ								
				X	E	P	N	Z	V			
処理単位	ワード	ロングワード		-	-	-	-	-	↑↓			
シンボル 及び パラメータ	ABS	[ABS W]			S → R							
	ABS!	[ABS L]			S → R							
	S: ソース格納アドレス R: リザルト (演算結果) を格納するアドレス											
処理内容	ABS	(S) → (R)										
	ABS!	(S) L → (R) L										
シンボル 入力手順	ABS	 シフト  			<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>							
		 A   										
	ABS!	 シフト  										
		 A   										
フラグの 設定	V: (ワード時) (S) = -32768の時1、それ以外0 (ロングワード時) (S) = -2147483648の時1、それ以外0 他: 保持											
注意事項	(1) オーバーフロー発生時、リザルトには下記フルスケール値がセットされます。											
	<table border="1" style="margin: auto;"> <tr> <td>ワード</td> <td>ロングワード</td> </tr> <tr> <td>H7FFF</td> <td>H7FFFFFFF</td> </tr> </table>		ワード	ロングワード	H7FFF	H7FFFFFFF						
ワード	ロングワード											
H7FFF	H7FFFFFFF											

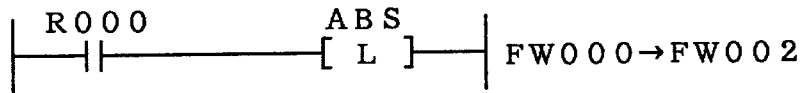
■ ABS



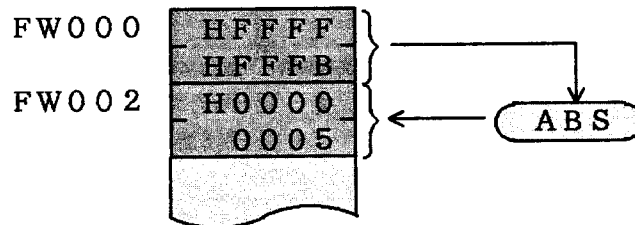
入力条件R000がOFF→ONの変化時、1回のみFW000の内容の絶対値を取り、FW001へ格納します。







■ ABS !



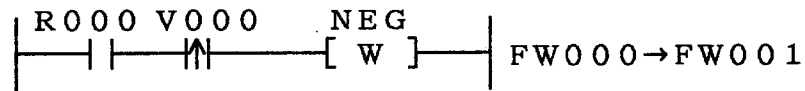
入力条件R000がONの時、FW000の内容の絶対値を取り、FW002へ格納します。



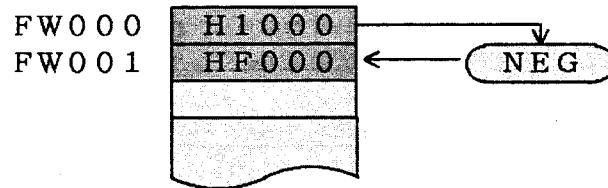
NEG 符号変換: SIGN CHANGE

機能	ソースの内容の符号変換処理を行い、リザルトへ格納します。													
シンボル	NEG	NEG!	/	フ ラ グ										
				X	E	P	N	Z	V					
処理単位	ワード	ロングワード	/	-	-	-	-	-	↑ ↓					
シンボル 及び パラメータ	NEG				[NEG W]			S → R						
	NEG!			[NEG L]			S → R							
	S: ソース格納アドレス R: リザルト (演算結果) を格納するアドレス													
処理内容	NEG	- (S) → (R)												
	NEG!	- (S) L → (R) L												
シンボル 入力手順	NEG	 シフト + / - 設定				<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>								
	NEG	 N E G 設定												
	NEG!	 シフト + / - !												
	NEG!	 N E G !												
フラグの 設定	V: (ワード時) (S) = -32768の時1、それ以外0 (ロングワード時) (S) = -2147483648の時1、それ以外0 他: 保持													
注意事項	(1) オーバーフロー発生時、リザルトには下記フルスケール値がセットされます。													
	<table border="1" style="margin: auto;"> <tr> <th>ワード</th> <th>ロングワード</th> </tr> <tr> <td>H7FFF</td> <td>H7FFFFFFF</td> </tr> </table>		ワード	ロングワード	H7FFF	H7FFFFFFF								
ワード	ロングワード													
H7FFF	H7FFFFFFF													

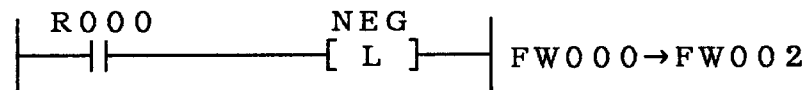
■ NEG



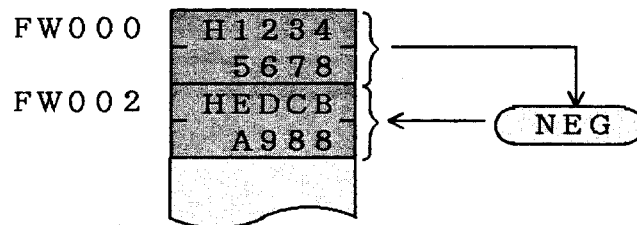
入力条件R000がOFF→ONの変化時、1回のみFW000の内容の符号変換を行い、その結果をリザルトへ格納します。



■ NEG!



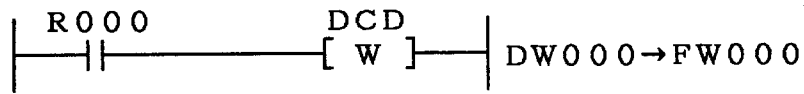
入力条件R000がONの時、FW000の内容の符号変換を行い、その結果をリザルトへ格納します。



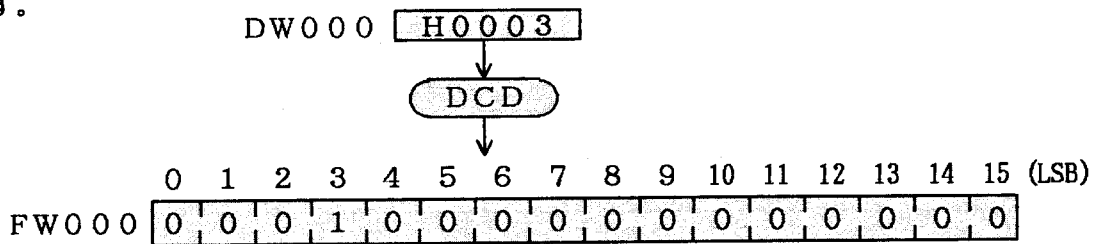
DCD デコード: DECODE

機 能	ソースの内容をデコードして、結果をリザルトへ格納します。								
シンボル	DCD	DCD!	/	フ ラ グ					
				X	E	P	N	Z	V
処理単位	ワード	ロングワード	/	-	-	-	-	-	-
シンボル 及び パラメータ	DCD								
	DCD!								
	S: ソース格納アドレス * ワード指定								
	R: リザルト (演算結果) を格納するアドレス								
処理内容	DCD								
	DCD!								
シンボル 入力手順	DCD					<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>			
	DCD!								
注意事項	<p>(1) フラグはすべて保持。</p> <p>(2) ソースデータはワード時下位4bit、ロングワード時下位5bitのみ有効。</p>								

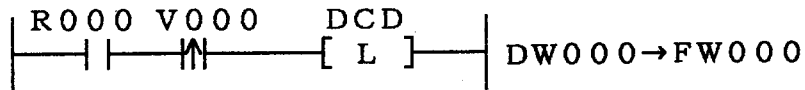
■ DCD



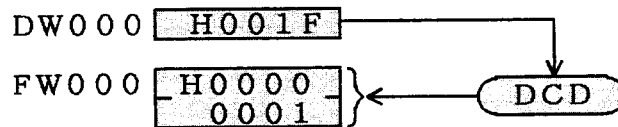
入力条件R000がONの時、DW000の内容をデコードして、その結果をFW000へ格納します。





■ DCD !



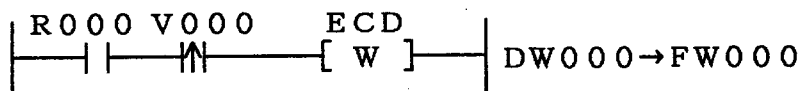
入力条件R000がOFF→ONの変化時一回のみ、DW000の内容をデコードしてその結果をFW000へ格納します。



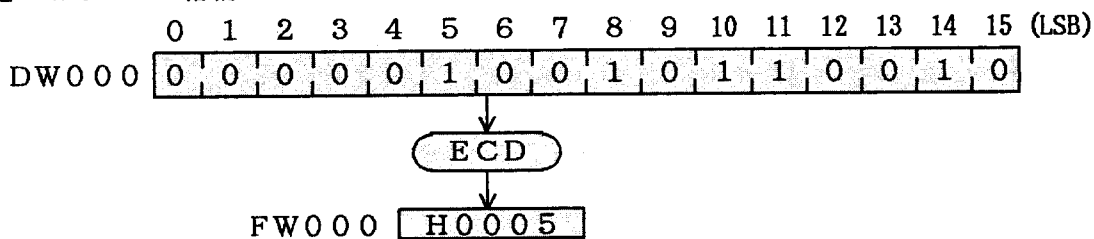
ECD エンコード: ENCODE

機能	ソースの内容をエンコードして、結果をリザルトへ格納します。								
シンボル	ECD	ECD!		フ ラ グ					
				X	E	P	N	Z	V
処理単位	ワード	ロングワード		-	↕	-	-	-	-
シンボル 及び パラメータ	ECD			$\left[\begin{array}{c} \text{ECD} \\ \text{W} \end{array} \right]$	$\underline{S} \rightarrow \underline{R}$				
	ECD!		$\left[\begin{array}{c} \text{ECD} \\ \text{L} \end{array} \right]$	$\underline{S} \rightarrow \underline{R}^*$					
	S: ソース格納アドレス								
	R: リザルト (演算結果) を格納するアドレス								
処理内容 ($\Delta \sim \Delta$: 任意)	ECD	$S \quad \begin{array}{c} 0 \quad n \quad 15(\text{LSB}) \\ \boxed{0 \sim 0} \quad \boxed{1} \quad \boxed{\Delta \sim \Delta} \end{array} \rightarrow R \quad \begin{array}{c} 12 \quad 15 \\ \boxed{n} \end{array}$							
	ECD!	$S L = \begin{array}{c} 0 \quad n \quad 31(\text{LSB}) \\ \boxed{0 \sim 0} \quad \boxed{1} \quad \boxed{\Delta \sim \Delta} \end{array} \rightarrow R \quad \begin{array}{c} 11 \quad 15 \\ \boxed{n} \end{array}$							
シンボル 入力手順	ECD				<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;">パラメータ入力</div>				
	ECD!								
フラグの 設定	E: (S) = 0の時1、それ以外0 他: 保持								
注意事項	(1) (S) = 0の時、未処理 (リザルトは変化しません)。 (2) エンコードするビットはソースのMSBから最初に検出された '1' についてのみ有効。								

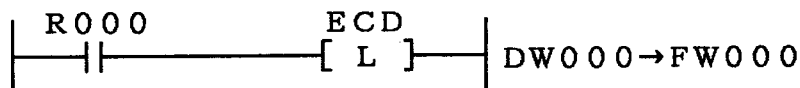
■ ECD



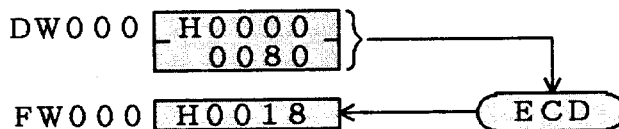
入力条件R000がOFF→ONの変化時、1回のみDW000の内容をエンコードして、その結果をFW000へ格納します。



■ ECD !

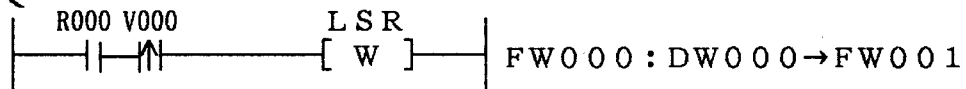


入力条件R000がONの時、DW000の内容をエンコードして、その結果をFW000へ格納します。

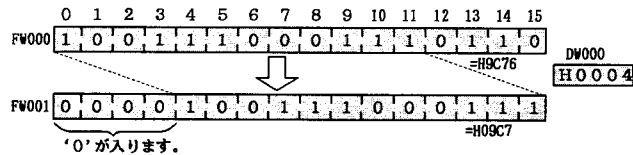


シンボル 入力手順	LSR#!	1	FUNC []	シフト	LSR	シフト	#	パラメータ入力
		2	FUNC []	L	S	R	シフト	
注意事項	(1) フラグはすべて保持。 (2) シフトビット数はワード時下位4ビット、ロングワード時下位5ビットのみ有効。							

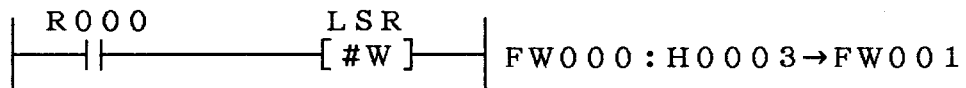
■ L S R



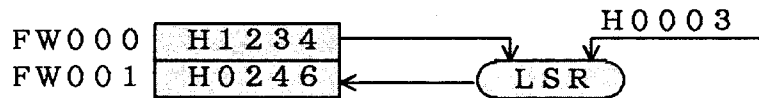
入力条件R000がOFF→ONの変化時、1回のみFW000の内容をDW000で指定された数だけ、右へシフトし、その結果をFW001へ格納します。



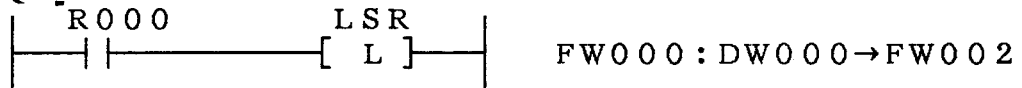
■ L S R #



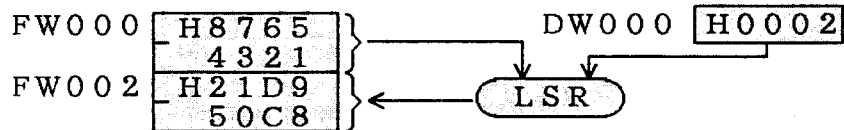
入力条件R000がONの時、FW000の内容を3ビット右へシフトし、その結果をFW001へ格納します。



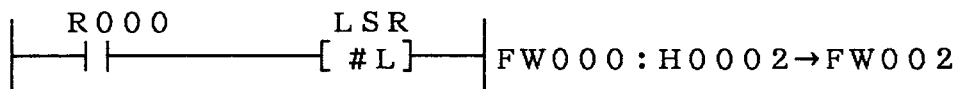
■ L S R !



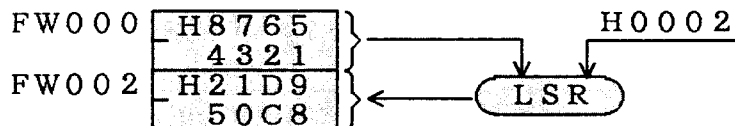
入力条件R000がONの時、FW000の内容をDW000で指定された数だけ、右へシフトし、その結果をFW002へ格納します。



■ L S R # !



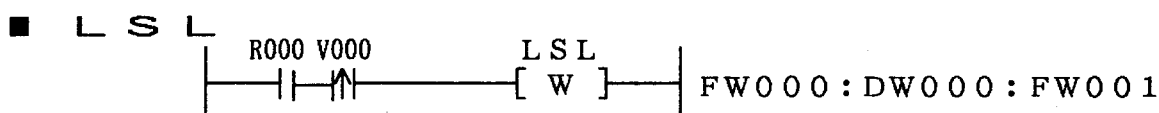
入力条件R000がONの時、FW000の内容を2ビット右へシフトし、その結果をFW002へ格納します。



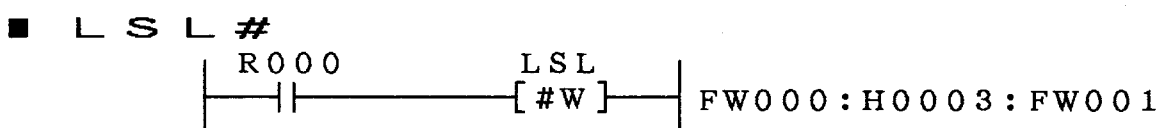
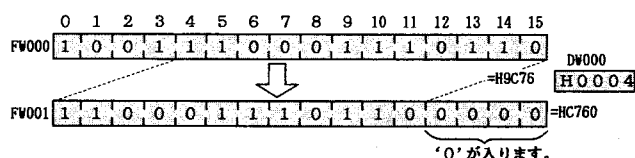
LSL 論理左シフト: LOGICAL SHIFT LEFT

機能		ソースの内容をデスティネーションの内容又はイミディエートデータの指定ビット数、左へシフトし、結果をリザルトへ格納する。									
シンボル	LSL	LSL#	LSL!	LSL#!	フ ラ グ						
					X	E	P	N	Z	V	
処理単位	ワード	ワード	ロングワード	ロングワード	-	-	-	-	-	-	
シンボル 及び パラメータ	LSL			S : D → R							
	LSL#			S : n → R							
	LSL!			S : D* → R							
	LSL#!			S : n → R							
S : ソース格納アドレス D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス					* ワード指定						
処理内容	LSL										
LSL#											
LSL!											
LSL#!											
シンボル 入力手順	LSL	1	FUNC [] シフト L S L 設定				パラメータ入力				
		2	FUNC [] L S L 設定								
	LSL#	1	FUNC [] シフト L S L #								
		2	FUNC [] L S L #								
	LSL!	1	FUNC [] シフト L S L !								
		2	FUNC [] L S L !								

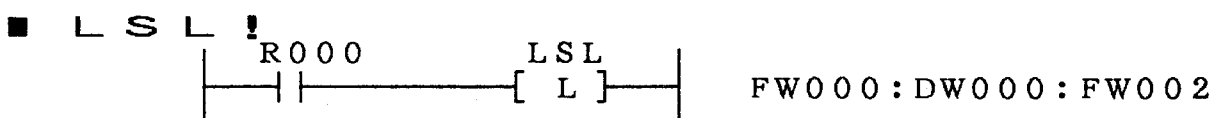
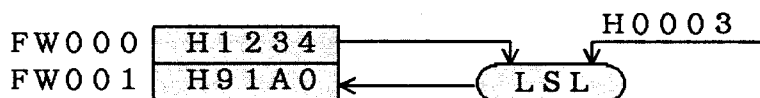
シンボル 入力手順	LSL#!	1		パラメータ入力
		2		
注意事項	(1) フラグはすべて保持。 (2) シフトビット数はワード時下位4ビット、ロングワード時下位5ビットのみ有効。			



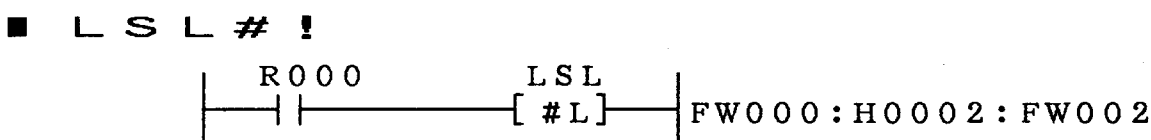
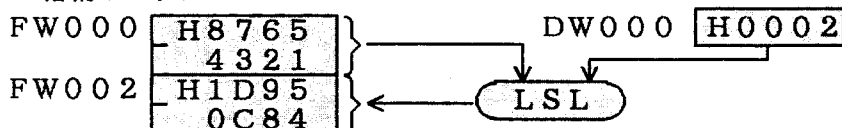
入力条件R000がOFF→ONの変化時、1回のみFW000の内容をDW000で指定された数だけ、左へシフトし、その結果をFW001へ格納します。



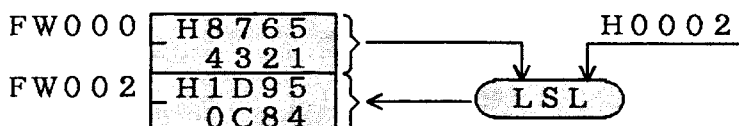
入力条件R000がONの時、FW000の内容を3ビット左へシフトし、その結果をFW001へ格納します。



入力条件R000がONの時、FW000の内容をDW000で指定された数だけ、左へシフトし、その結果をFW002へ格納します。



入力条件R000がONの時、FW000の内容を2ビット左へシフトし、その結果をFW002へ格納します。

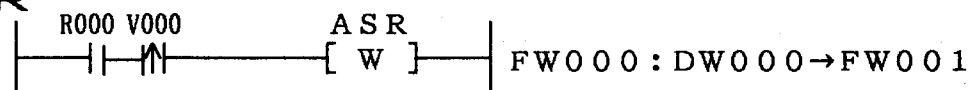


ASR 算術右シフト: ARITHMETIC SHIFT RIGHT

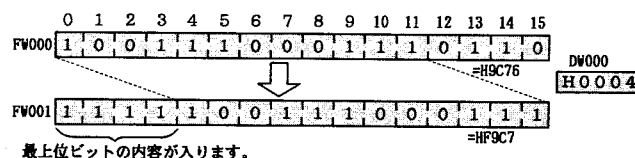
機能	ソースの内容をデスティネーションの内容又はイミディエートデータの指定ビット数、右へシフト (符号ビット保持) し、結果をリザルトへ格納する。									
シンボル	ASR	ASR#	ASR!	ASR#!	フ ラ グ					
					X	E	P	N	Z	V
処理単位	ワード	ワード	ロングワード	ロングワード	-	-	-	-	-	-
シンボル 及び パラメータ	ASR			$S : D \rightarrow R$						
	ASR#			$S : n \rightarrow R$						
	ASR!			$S : D^* \rightarrow R$						
	ASR#!			$S : n \rightarrow R$						
S : ソース格納アドレス D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス * ワード指定										
処理内容	ASR									
	ASR#									
	ASR!									
	ASR#!									
シンボル 入力手順	ASR	1		<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>						
		2								
	ASR#	1								
		2								
	ASR!	1								
		2								

シンボル 入力手順	ASR#!	1	FUNC []	シフト	ASR	シフト	#	パラメータ入力
		2	FUNC []	A	S	R	シフト	
注意事項	(1) フラグはすべて保持。 (2) シフトビット数はワード時下位4ビット、ロングワード時下位5ビットのみ有効。							

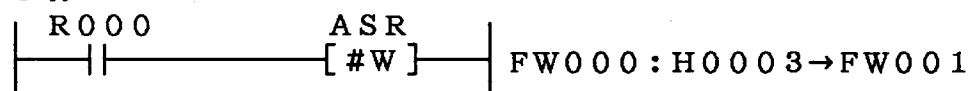
■ ASR



入力条件R000がOFF→ONの変化時、1回のみFW000の内容をDW000で指定された数だけ、右ヘシフトし、その結果をFW001へ格納します。



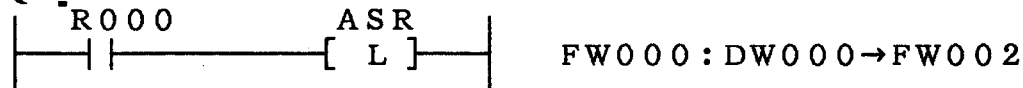
■ ASR #



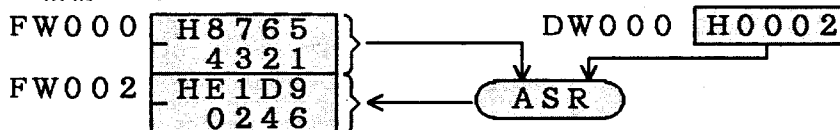
入力条件R000がONの時、FW000の内容を3ビット右ヘシフトし、その結果をFW001へ格納します。



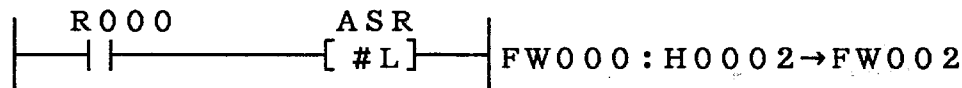
■ ASR !



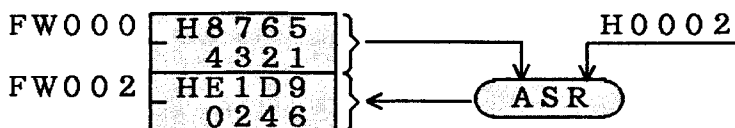
入力条件R000がONの時、FW000の内容をDW000で指定された数だけ、右ヘシフトし、その結果をFW002へ格納します。



■ ASR # !



入力条件R000がONの時、FW000の内容を2ビット右ヘシフトし、その結果をFW002へ格納します。

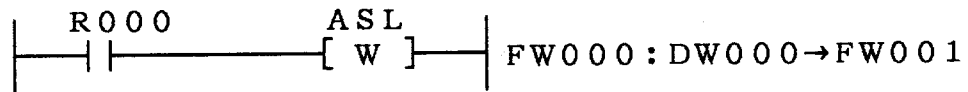


ASL 算術左シフト: ARITHMETIC SHIFT LEFT

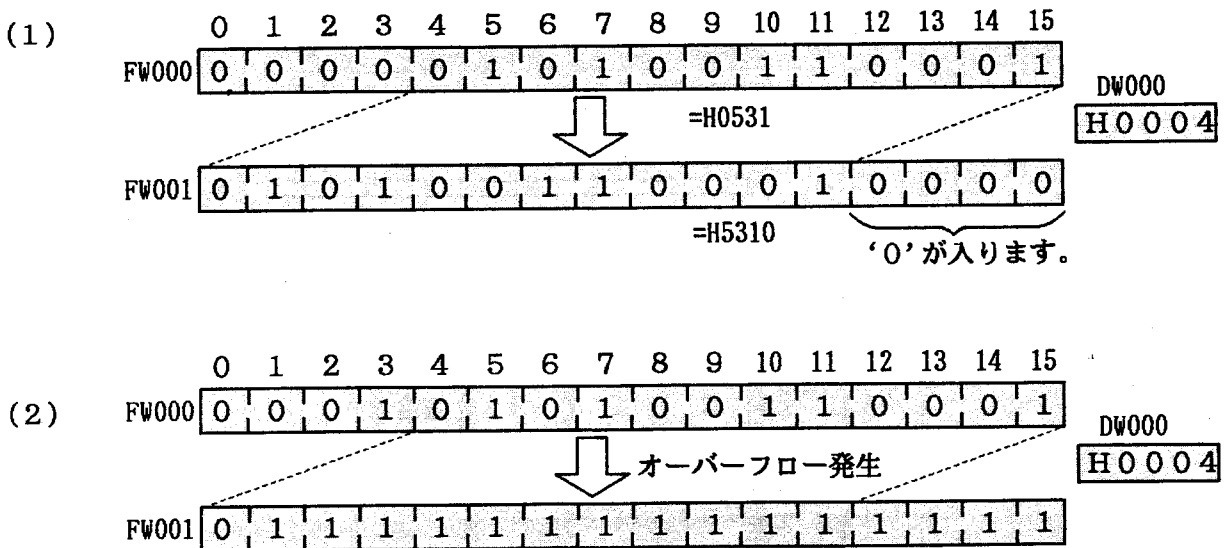
機能		ソースの内容をデスティネーションの内容又はイミディエートデータの指定ビット数、左へシフトし、結果をリザルトへ格納する。										
シンボル	ASL	ASL #	ASL !	ASL #!	フ ラ グ							
					X	E	P	N	Z	V		
処理単位	ワード	ワード	ロングワード	ロングワード	-	-	-	-	-	↑↓		
シンボル 及び パラメータ	ASL		ASL [W]		S : D : R							
	ASL #		ASL [#W]		S : n : R							
	ASL !		ASL [L]		S : D* : R							
	ASL #!		ASR [#L]		S : n : R							
S : ソース格納アドレス D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス					* ワード指定							
処理内容	ASL	(S)										
	ASL #	(S)										
	ASL !	(S)L										
	ASL #!	(S)L										
シンボル 入力手順	ASL	1	FUNC [] シフト ASL 設定				パラメータ入力					
		2	FUNC [] A S L 設定									
	ASL #	1	FUNC [] シフト ASL #									
		2	FUNC [] A S L #									
	ASL !	1	FUNC [] シフト ASL !									
		2	FUNC [] A S L !									

シンボル 入力手順	ASL#!	1	FUNC [H]	シフト	ASL	シフト	#	パラメータ入力
		2	FUNC [H]	A	S	L	シフト	
フラグの 設定	V: シフト操作中に符号ビットが1回でも変化すれば1、それ以外0 他: 保持							
注意事項	(1) オーバーフロー発生時、リザルトには下記フルスケール値がセットされます。							
			ワード	ロングワード				
		(S)>0の時	H7FFF	H7FFFFFFF				
		(S)<0の時	H8000	H80000000				

■ ASL



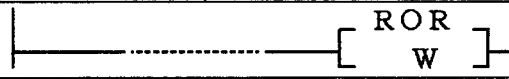
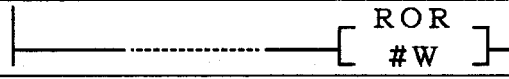
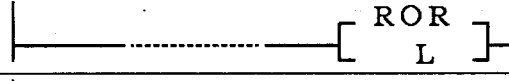
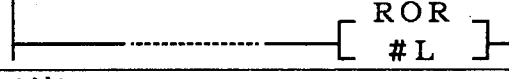
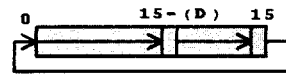
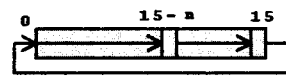
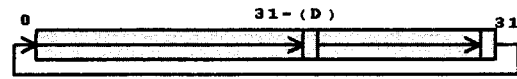







入力条件R000がONの時、FW000の内容をDW000で指定された数だけ、左へシフトし、その結果をFW001へ格納します。



この時、VフラグがONします。

Vフラグ ON S005

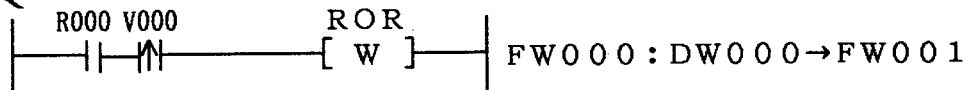
ROR 右回転: ROTATE RIGHT

機能	ソースの内容をデスティネーションの内容又はイミディエートデータの指定ビット数、右へ回転し、結果をリザルトへ格納する。									
シンボル	ROR	ROR#	ROR!	ROR#!	フ ラ グ					
					X	E	P	N	Z	V
処理単位	ワード	ワード	ロングワード	ロングワード	-	-	-	-	-	-
シンボル 及び パラメータ	ROR			$S : D \rightarrow R$						
	ROR#			$S : n \rightarrow R$						
	ROR!			$S : D^* \rightarrow R$						
	ROR#!			$S : n \rightarrow R$						
S: ソース格納アドレス D: デスティネーション格納アドレス n: イミディエートデータ R: リザルト (演算結果) を格納するアドレス					* ワード指定					
処理内容	ROR			(S) $\xrightarrow{\text{右回転}}$ (R)						
ROR#			(S) $\xrightarrow{\text{右回転}}$ (R)							
ROR!			(S) L $\xrightarrow{\text{右回転}}$ (R) L							
ROR#!			(S) L $\xrightarrow{\text{右回転}}$ (R) L							
シンボル 入力手順	ROR	1		パラメータ入力						
		2								
	ROR#	1								
		2								
	ROR!	1								
		2								

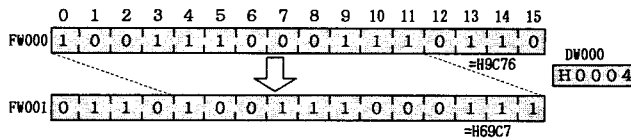
ローレイト

シンボル 入力手順	ROR#!	1	FUNC [H] シフト ROR シフト #	パラメータ入力
		2	FUNC [H] R O R シフト #	
注意事項	(1) フラグはすべて保持。 (2) 回転ビット数はワード時下位4ビット、ロングワード時下位5ビットのみ有効。			

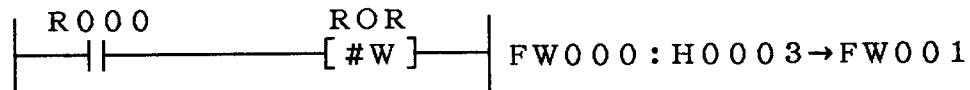
■ ROR



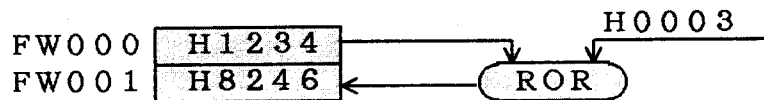
入力条件R000がOFF→ONの変化時、1回のみFW000の内容をDW000で指定された数だけ、右へ回転し、その結果をFW001へ格納します。



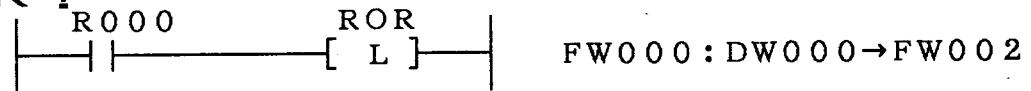
■ ROR #



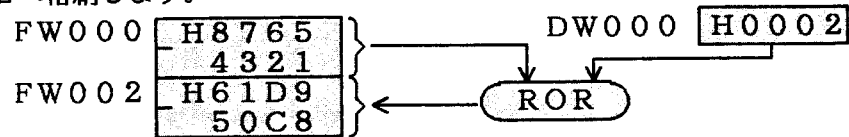
入力条件R000がONの時、FW000の内容を3ビット右へ回転し、その結果をFW001へ格納します。



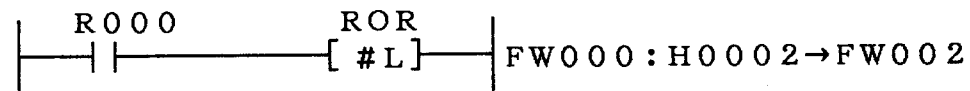
■ ROR !



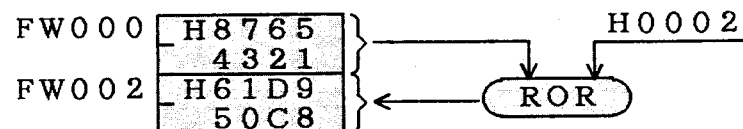
入力条件R000がONの時、FW000の内容をDW000で指定された数だけ、右へ回転し、その結果をFW002へ格納します。



■ ROR # !



入力条件R000がONの時、FW000の内容を2ビット右へ回転し、その結果をFW002へ格納します。

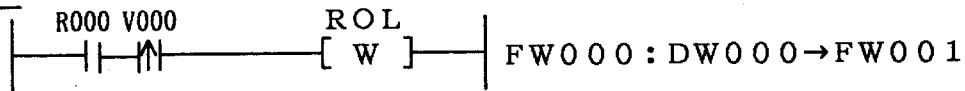


ROL 左回転: ROTATE LEFT

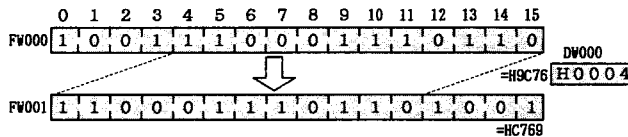
機能	ソースの内容をデスティネーションの内容又はイミディエートデータの指定ビット数、左へ回転し、結果をリザルトへ格納する。									
シンボル	ROL	ROL #	ROL !	ROL #!	フ ラ グ					
					X	E	P	N	Z	V
処理単位	ワード	ワード	ロングワード	ロングワード	-	-	-	-	-	-
シンボル 及び パラメータ	ROL				$S : D \rightarrow R$					
	ROL #				$S : n \rightarrow R$					
	ROL !				$S : D^* \rightarrow R$					
	ROL #!				$S : n \rightarrow R$					
S : ソース格納アドレス D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス * ワード指定										
処理内容	ROL									
ROL #										
ROL !										
ROL #!										
シンボル 入力手順	ROL	1				<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>				
		2								
	ROL #	1								
		2								
ROL !	1									
	2									

シンボル 入力手順	ROL#!	1	FUNC [H]	シフト	ROL	シフト	#	パラメータ入力
		2	FUNC [H]	R	O	L	シフト	
注意事項	(1) フラグはすべて保持。 (2) 回転ビット数はワード時下位4ビット、ロングワード時下位5ビットのみ有効。							

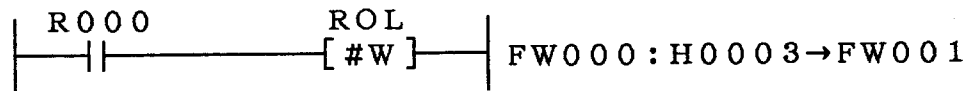
■ ROL



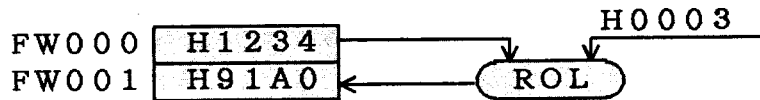
入力条件R000がOFF→ONの変化時、1回のみFW000の内容をDW000で指定された数だけ、左へ回転し、その結果をFW001へ格納します。



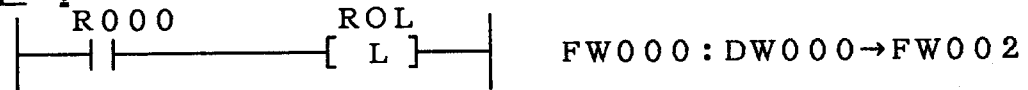
■ ROL #



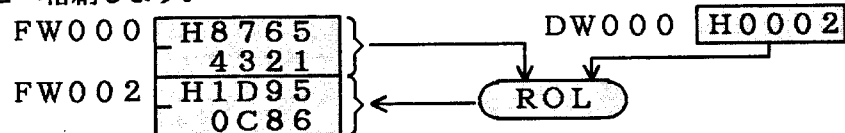
入力条件R000がONの時、FW000の内容を3ビット左へ回転し、その結果をFW001へ格納します。



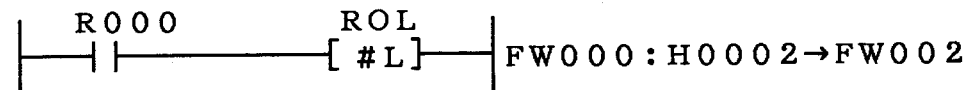
■ ROL !



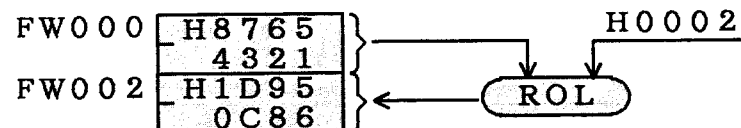
入力条件R000がONの時、FW000の内容をDW000で指定された数だけ、左へ回転し、その結果をFW002へ格納します。



■ ROL # !



入力条件R000がONの時、FW000の内容を2ビット左へ回転し、その結果をFW002へ格納します。

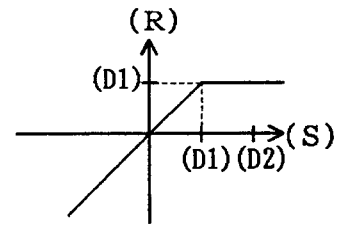


L I M リミッター=LIMITER

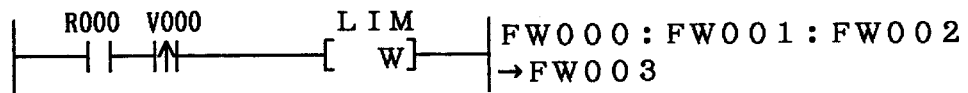
機能	ソースの内容を境界値 (デスティネーションD1, D2の内容 又はイミディエートデータn1, n2)と比較し、リザルトへ格 納します。(右図参照)									
	シンボル	LIM	LIM#	LIM!	フ ラ グ					
処理単位	ワード	ワード	ロングワード	X	E	P	N	Z	V	
シンボル 及び パラメータ	LIM				S : D1 : D2 → R					
	LIM#				S : n1 : n2 → R					
	LIM!				S : D1 : D2 → R					
	S : ソース格納アドレス D1, D2 : デスティネーション格納アドレス n1, n2 : イミディエートデータ R : リザルト (演算結果) を格納するアドレス									
処理内容	LIM	(1) (D1) < (S)の時 (2) (D2) ≤ (S) ≤ (D1)の時 (3) (S) < (D2)の時			(D1) → (R) (S) → (R) (D2) → (R)					
	LIM#	(1) n1 < (S)の時 (2) n2 ≤ (S) ≤ n1の時 (3) (S) < n2の時			n1 → (R) (S) → (R) n2 → (R)					
	LIM!	(1) (D1)L < (S)Lの時 (2) (D2)L ≤ (S)L ≤ (D1)Lの時 (3) (S)L < (D2)Lの時			(D1)L → (R)L (S)L → (R)L (D2)L → (R)L					
シンボル 入力手順	LIM	1				<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;">パラメータ入力</div>				
		2								
	LIM#	1								
		2								
	LIM!	1								
		2								
フラグの 設定	E : (D1) < (D2)又はn1 < n2の時1、それ以外0 他 : 保持									

関数処理

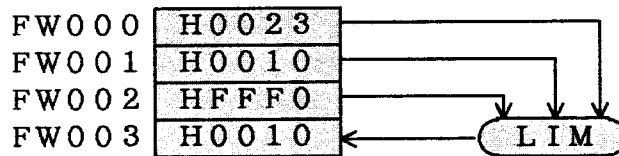
注意事項 (1) エラーフラグON時、(D2)又はn2との比較は行いません。
(右図参照)



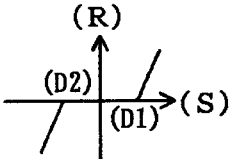

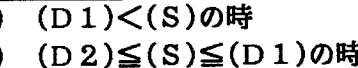
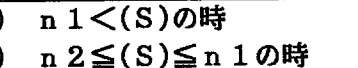






■ LIM



入力条件R000がOFF→ONの変化時、1回のみFW000の内容をFW001及びFW002の内容と比較して、結果をFW003へ格納します。



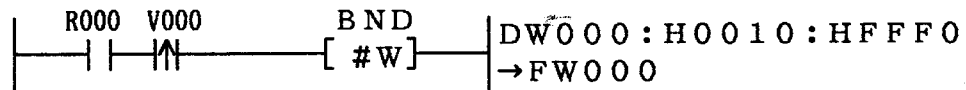
BND デッドバンド:DEAD BAND

機能	ソースの内容を境界値 (デスティネーションD1, D2の内容 又はイミディエートデータn1, n2)と比較し、境界範囲内を データ0 (不感帯)としてリザルトへ格納します。(右図参照)									
シンボル	BND	BND#	BND!	フ ラ グ						
処理単位	ワード	ワード	ロングワード	X	E	P	N	Z	V	
シンボル 及び パラメータ	BND			$S : D1 : D2$ $\rightarrow R$						
BND#			$S : n1 : n2$ $\rightarrow R$							
BND!			$S : D1 : D2$ $\rightarrow R$							
S : ソース格納アドレス D1, D2 : デスティネーション格納アドレス n1, n2 : イミディエートデータ R : リザルト (演算結果) を格納するアドレス										
処理内容	BND	(1) $(D1) < (S)$ の時 (2) $(D2) \leq (S) \leq (D1)$ の時 (3) $(S) < (D2)$ の時			$(S) - (D1) \rightarrow (R)$ $0 \rightarrow (R)$ $(S) - (D2) \rightarrow (R)$					
BND#	(1) $n1 < (S)$ の時 (2) $n2 \leq (S) \leq n1$ の時 (3) $(S) < n2$ の時			$(S) - n1 \rightarrow (R)$ $0 \rightarrow (R)$ $(S) - n2 \rightarrow (R)$						
BND!	(1) $(D1)L < (S)L$ の時 (2) $(D2)L \leq (S)L \leq (D1)L$ の時 (3) $(S)L < (D2)L$ の時			$(S)L - (D1)L \rightarrow (R)L$ $0 \rightarrow (R)L$ $(S)L - (D2)L \rightarrow (R)L$						
シンボル 入力手順	BND	1				<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;">パラメータ入力</div>				
BND	2									
BND#	1									
BND#	2									
BND!	1									
BND!	2									

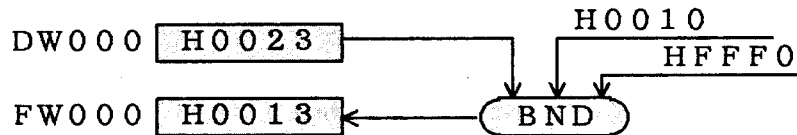
フラグの 設 定	E : (D1) < (D2) 又は $n_1 < n_2$ の時 1、それ以外 0 V : (ワード時) (R) < -32768 又は $32767 < (R)$ の時 1、それ以外 0 (ロングワード時) (R) < -2147483648 又は $2147483647 < (R)$ の時 1、それ以外 0 他 : 保持
注意事項	(1) (D1) < (D2) 又は $n_1 < n_2$ の時、エラーフラグ(E)がON (オーバーフローフラグ(V) OFF) し、未処理。(リザルトは変化しません。) (2) オーバーフロー発生時、リザルトには下記フルスケール値がセットされます。

	ワード	ロングワード
正のオーバーフロー時	H7FFF	H7FFFFFFF
負のオーバーフロー時	H8000	H80000000

■ BND #



入力条件R000がOFF→ONの変化時、1回のみFW000の内容をイミディエトデータH0010及びHFFF0と比較して、結果をFW000へ格納します。

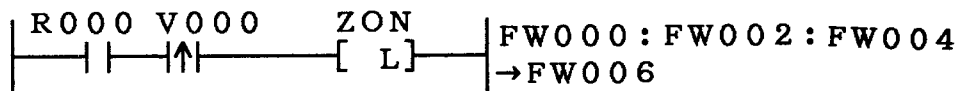


ZON デッドゾーン=DEAD ZONE

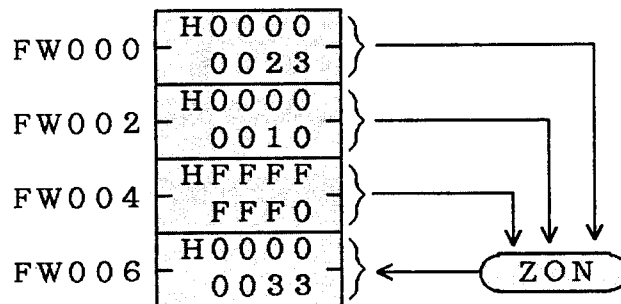
機能	ソースの内容を正負によりそれぞれバイアス (デスティネーションD1, D2の内容又はイミディエートデータn1, n2) を加算し、結果をリザルトへ格納します。(右図参照)									
シンボル	ZON	ZON#	ZON!	フ ラ グ						
処理単位	ワード	ワード	ロングワード	X	E	P	N	Z	V	
シンボル 及び パラメータ	ZON			S : D1 : D2 → R						
	ZON#			S : n1 : n2 → R						
	ZON!			S : D1 : D2 → R						
	S : ソース格納アドレス D : デスティネーション格納アドレス n : イミディエートデータ R : リザルト (演算結果) を格納するアドレス									
処理内容	ZON	(1) (S) >0の時 (S) + (D1) → (R) (2) (S) =0の時 0 → (R) (3) (S) <0の時 (S) + (D2) → (R)								
	ZON#	(1) (S) >0の時 (S) + n1 → (R) (2) (S) =0の時 0 → (R) (3) (S) <0の時 (S) + n2 → (R)								
	ZON!	(1) (S) L >0の時 (S) L + (D1) L → (R) L (2) (S) L =0の時 0 → (R) L (3) (S) L <0の時 (S) L + (D2) L → (R) L								
シンボル 入力手順	ZON	1					<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;">パラメータ入力</div>			
		2								
	ZON#	1								
		2								
	ZON!	1								
		2								

フラグの設定	<p>E: (D1) < (D2) 又は n1 < n2 の時 1, それ以外 0</p> <p>V: (ワード時) (R) < -32768 又は 32767 < (R) の時 1, それ以外 0 (ロングワード時) (R) < -2147483648 又は 2147483647 < (R) の時 1, それ以外 0</p> <p>他: 保持</p>									
注意事項	<p>(1) (D1) < (D2) 又は n1 < n2 時, エラーフラグ (E) が ON (オーバーフローフラグ (V) OFF) し, 未処理。(リザルトは変化しません。)</p> <p>(2) オーバーフロー発生時, リザルトには下記フルスケール値がセットされます。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th></th> <th>ワード</th> <th>ロングワード</th> </tr> </thead> <tbody> <tr> <td>正のオーバーフロー時</td> <td>H7FFF</td> <td>H7FFFFFFF</td> </tr> <tr> <td>負のオーバーフロー時</td> <td>H8000</td> <td>H/8000000</td> </tr> </tbody> </table>		ワード	ロングワード	正のオーバーフロー時	H7FFF	H7FFFFFFF	負のオーバーフロー時	H8000	H/8000000
	ワード	ロングワード								
正のオーバーフロー時	H7FFF	H7FFFFFFF								
負のオーバーフロー時	H8000	H/8000000								

■ ZON !



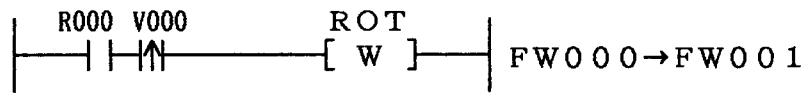
入力条件R000がOFF→ONの変化時、1回のみFW000の内容の正負によりFW002又はFW004の内容を加算し、その結果をFW006へ格納します。



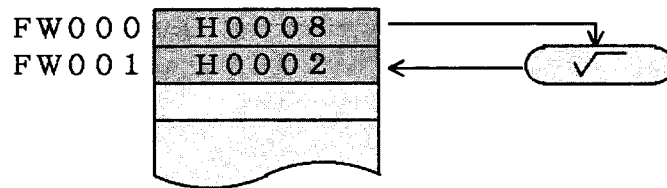
ROT 平方根: ROOT

機能	ソースの内容の平方根（整数部のみ）をリザルトへ格納します。									
シンボル	ROT	ROT!	/	フ ラ グ						
				X	E	P	N	Z	V	
処理単位	ワード	ロングワード	/	-	-	-	-	-	-	
シンボル 及び パラメータ	ROT									
	ROT!									
	S : ソース格納アドレス R : リザルト（演算結果）を格納するアドレス									
処理内容	ROT	(S) ≥ 0 の時 $\sqrt{(S)} \rightarrow (R)$ (S) < 0 の時 $0 \rightarrow (R)$								
	ROT!	(S) L ≥ 0 の時 $\sqrt{(S) L} \rightarrow (R) L$ (S) L < 0 の時 $0 \rightarrow (R) L$								
シンボル 入力手順	ROT	1					パラメータ入力			
		2								
	ROT!	1								
		2								
注意事項	(1) フラグはすべて保持。 (2) ソースの内容が負の場合、リザルトには0がセットされます。									

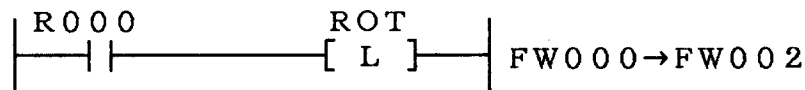
■ ROT



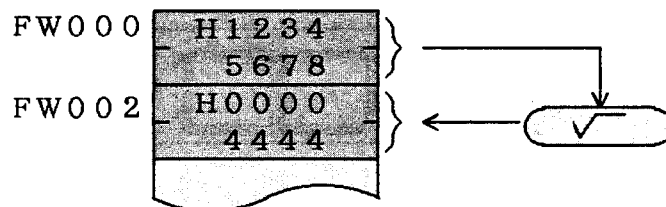
入力条件R000がOFF→ONの変化時、1回のみFW000の内容の平方根を取り、その結果をFW001へ格納します。



■ ROT !



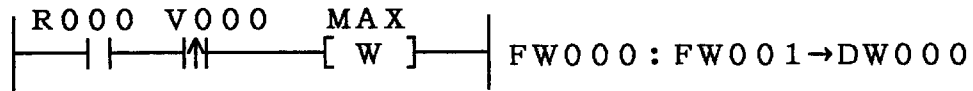
入力条件R000がONの時、FW000の内容の平方根を取り、その結果をFW002へ格納します。



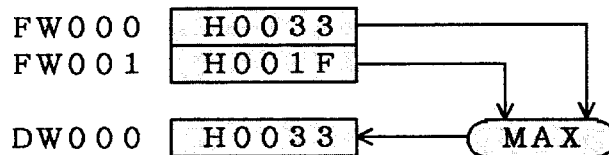
MAX 最大=MAXIMUM

機能	ソースの内容とデスティネーションの内容又はイミディエートデータの大小比較を行い、大きい値をリザルトへ格納します。									
シンボル	MAX	MAX#	MAX!	フ ラ グ						
				X	E	P	N	Z	V	
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	-	-
シンボル 及び パラメータ	MAX		$\left[\begin{array}{c} \text{MAX} \\ \text{W} \end{array} \right]$	$\underline{S} : \underline{D} \rightarrow \underline{R}$						
	MAX#		$\left[\begin{array}{c} \text{MAX} \\ \# \text{W} \end{array} \right]$	$\underline{S} : \underline{n} \rightarrow \underline{R}$						
	MAX!		$\left[\begin{array}{c} \text{MAX} \\ \text{L} \end{array} \right]$	$\underline{S} : \underline{D} \rightarrow \underline{R}$						
	S: ソース格納アドレス D: デスティネーション格納アドレス n: イミディエートデータ R: リザルト (演算結果) を格納するアドレス									
処理内容	MAX	(S) \geq (D) の時 (S) \rightarrow (R) (S) $<$ (D) の時 (D) \rightarrow (R)								
	MAX#	(S) \geq n の時 (S) \rightarrow (R) (S) $<$ n の時 n \rightarrow (R)								
	MAX!	(S) L \geq (D) L の時 (S) L \rightarrow (R) L (S) L $<$ (D) L の時 (D) L \rightarrow (R) L								
シンボル 入力手順	MAX	1	FUNC [] シフト MAX 設定							パラメータ入力
		2	FUNC [] M A X 設定							
	MAX#	1	FUNC [] シフト MAX #							
		2	FUNC [] M A X #							
	MAX!	1	FUNC [] シフト MAX !							
		2	FUNC [] M A X !							
注意事項	(1) フラグはすべて保持。									

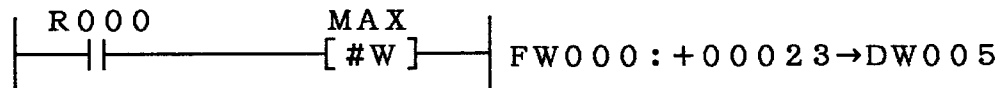
■ MAX



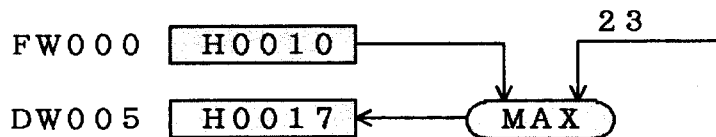
入力条件R000がOFF→ONの変化時、1回のみFW000の内容とFW001の内容の大小比較を行い、大きい値をDW000へ格納します。



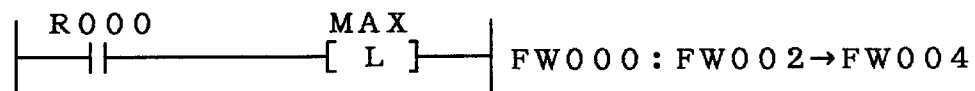
■ MAX



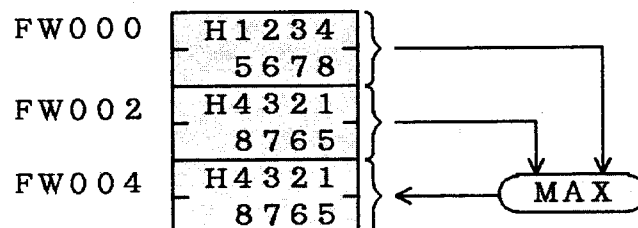
入力条件R000がONの時、FW000の内容とイミディエートデータ23の大小比較を行い、大きい値をDW005へ格納します。



■ MAX !



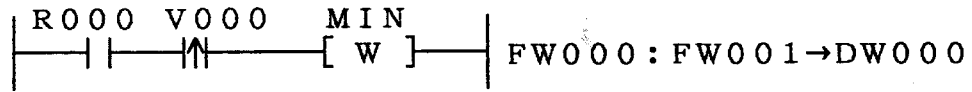
入力条件R000がONの時、FW000の内容とFW002の内容の大小比較を行い、大きい値をFW004へ格納します。



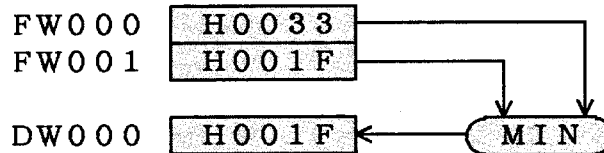
MIN 最小=MINIMUM

機能	ソースの内容とデスティネーションの内容又はイミディエートデータの大小比較を行い、小さい値をリザルトへ格納します。									
シンボル	MIN	MIN#	MIN!	フ ラ グ						
				X	E	P	N	Z	V	
処理単位	ワード	ワード	ロングワード	-	-	-	-	-	-	
シンボル 及び パラメータ	MIN		$\left[\begin{array}{c} \text{MIN} \\ \text{W} \end{array} \right]$	$\underline{S} : \underline{D} \rightarrow \underline{R}$						
	MIN#		$\left[\begin{array}{c} \text{MIN} \\ \# \text{W} \end{array} \right]$	$\underline{S} : \underline{n} \rightarrow \underline{R}$						
	MIN!		$\left[\begin{array}{c} \text{MIN} \\ \text{L} \end{array} \right]$	$\underline{S} : \underline{D} \rightarrow \underline{R}$						
S: ソース格納アドレス D: デスティネーション格納アドレス n: イミディエートデータ R: リザルト (演算結果) を格納するアドレス										
処理内容	MIN	$(S) \leq (D)$ の時 $(S) \rightarrow (R)$ $(S) > (D)$ の時 $(D) \rightarrow (R)$								
	MIN#	$(S) \leq n$ の時 $(S) \rightarrow (R)$ $(S) > n$ の時 $n \rightarrow (R)$								
	MIN!	$(S) L \leq (D) L$ の時 $(S) L \rightarrow (R) L$ $(S) L > (D) L$ の時 $(D) L \rightarrow (R) L$								
シンボル 入力手順	MIN	1								<div style="border: 1px solid black; padding: 5px; display: inline-block;">パラメータ入力</div>
		2								
	MIN#	1								
		2								
	MIN!	1								
		2								
注意事項	(1) フラグはすべて保持。									

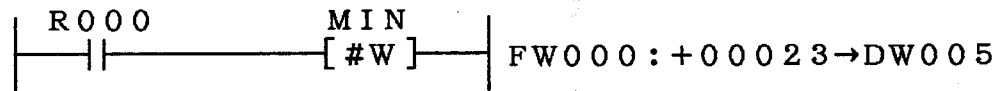
■ MIN



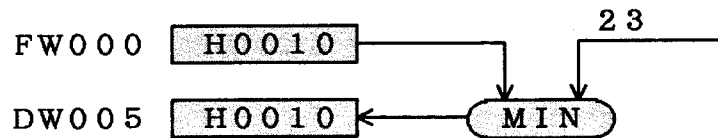
入力条件R000がOFF→ONの変化時、1回のみFW000の内容とFW001の内容の大小比較を行い、小さい値をDW000へ格納します。



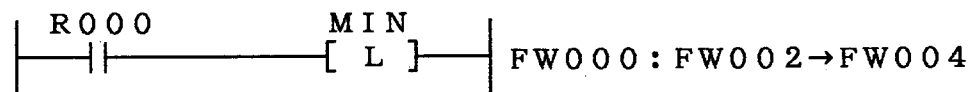
■ MIN #



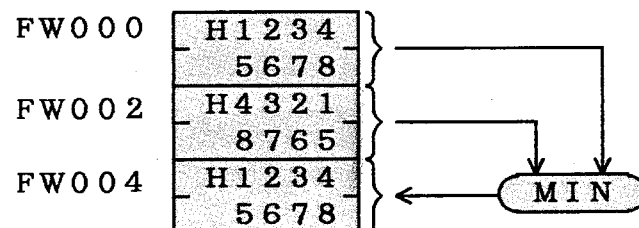
入力条件R000がONの時、FW000の内容とイミディエートデータ23の大小比較を行い、小さい値をDW005へ格納します。



■ MIN !



入力条件R000がONの時、FW000の内容とFW002の内容の大小比較を行い、小さい値をFW004へ格納します。

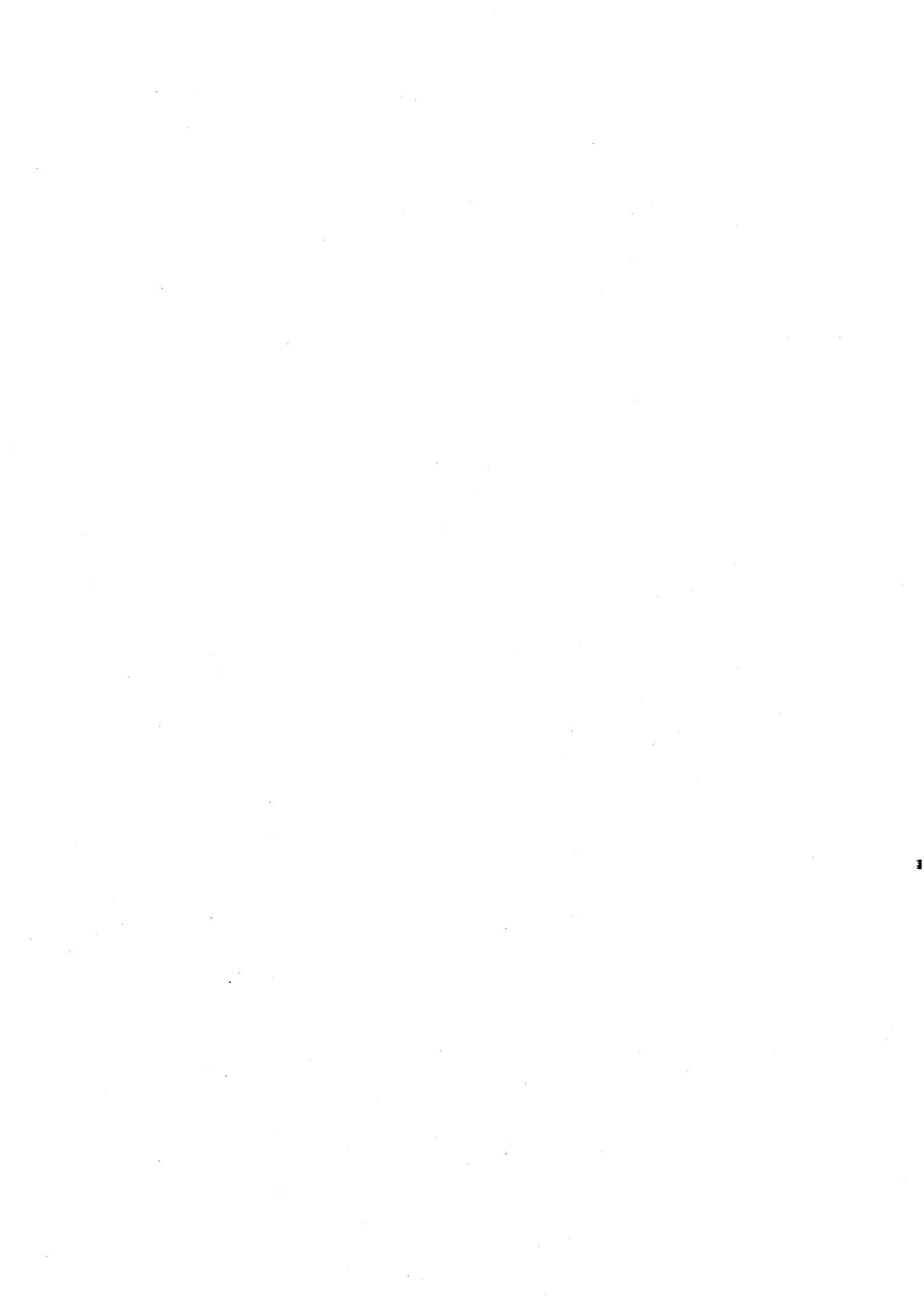


CLR クリア: CLEAR

特殊

機能		フ ラ グ						
		X	E	P	N	Z	V	
シンボル (注)	XCLR		-	-	-	-	-	-
	YCLR		-	-	-	-	-	-
	GCLR		-	-	-	-	-	-
	RCLR		-	-	-	-	-	-
	KCLR		-	-	-	-	-	-
	TCLR		-	-	-	-	-	-
	UCLR		-	-	-	-	-	-
	CCLR		-	-	-	-	-	-
	VCLR		-	-	-	-	-	-
	ECLR		-	-	-	-	-	-
	FCLR		0	0	0	0	0	0
シンボル 入力手順	<div style="text-align: center;"> </div> <p>クリアしたいエリアのコード (X, Y, G, R, ...) を入力 します。</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> パラメータはありません。 </div>							

クリア エリア	Xエリア	← XCLRがクリアするエリア。		
	Yエリア	← YCLRがクリアするエリア。		
	Gエリア	← GCLRがクリアするエリア。		
	Rエリア	← RCLRがクリアするエリア。		
	Kエリア	← KCLRがクリアするエリア。		
	Tエリア	← TCLRが クリアするエリア。	→ T計数値エリア	
	Uエリア	← UCLRが クリアするエリア。	→ U計数値エリア	
	Cエリア	← CCLRが クリアするエリア。	→ C計数値エリア	
	Vエリア	← VCLRがクリアするエリア。		
	Eエリア	← ECLRがクリアするエリア。		
	SW000	ファンクションフラグエリア (X, E, P, N, Z, V, 等)	← FCLRがクリアするエリア。	



3

処理時間

3 処理時間

3.1 2αの処理時間

ラダー回路プログラムの処理時間はラダープログラムの実行時間と、演算命令、プロセスレジスタ、ネスティングコイル等のOS処理時間の合計となります。

(1) ラダー回路命令の処理時間

シンボル	処理時間 μs/命令
↑↑, ↓↓, , ↑, ↓	0.33
↑↑, ↓↓, (), (S), (R)	0.66
END命令 (プログラミング時 自動セット)	300

(2) 演算ファンクションの処理時間

図にOSによる演算処理の概要を示します。

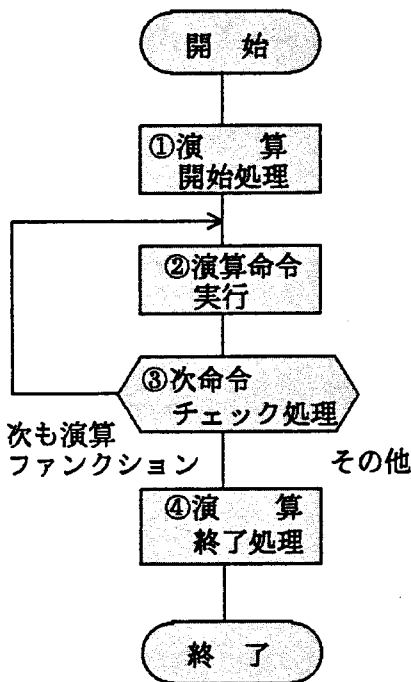


図 3-1

②の処理時間を演算ファンクション一覧表 (P. 31 参照) に示します。

①③④のOS処理時間は下記の様になります。

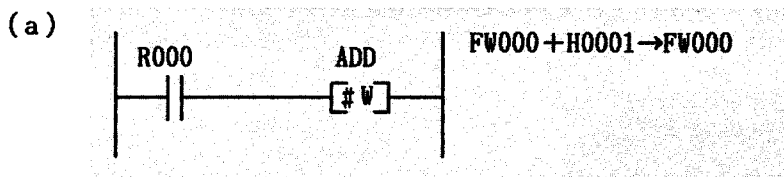
No	処理時間 [msec]
①	0.12
③	次が演算命令の場合=0.04 その他 =0.01
④	0.12

[概略計算]

- ・演算ファンクション1命令単独の場合
0.3ms/命令
- ・演算ファンクション4命令以上連続の場合
0.15ms/命令

上記時間に演算ファンクションの使用個数をかけた時間が、演算ファンクションの概略処理時間になります。

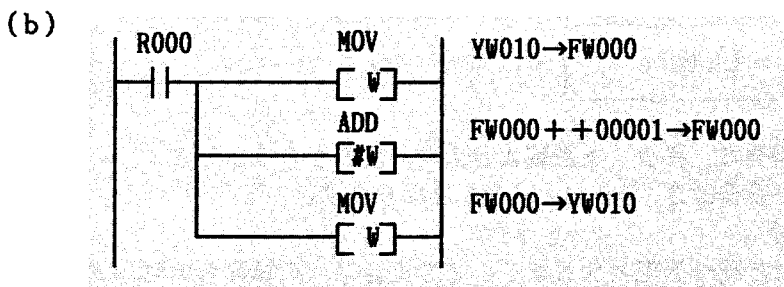
次の回路を例に計算方法を示します。



- ① = 0.12
- ② = 0.06
- ③ = 0.01
- ④ = 0.12

計 0.31 [msec]

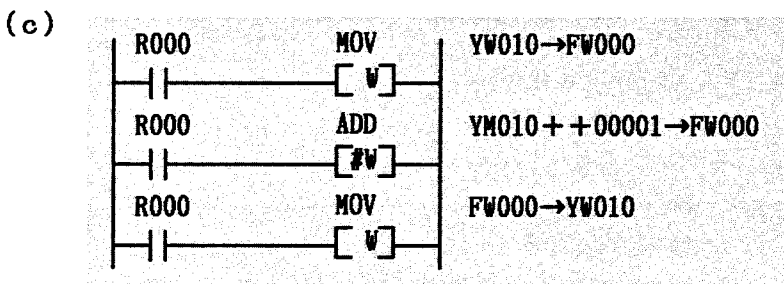
処理時間=①+②+③+④



- ① = 0.12
- 3 × ③ = 2 × 0.04 + 1 × 0.01 = 0.09
- ④ = 0.12
- 2 × ②MOV = 2 × 0.04 = 0.08
- ②ADD# = 0.06
- 計 = 0.47 [msec]

処理時間=①+3×③+④
+2×②MOV+②ADD

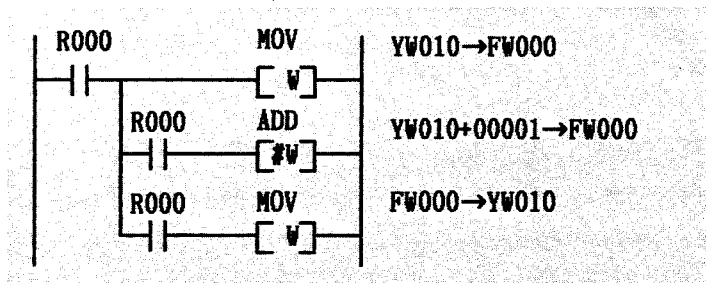
上記 (b) の回路は次の様にも作成できます。しかし、この場合 (b) よりも処理時間が長くなります。



- 3 × ① = 3 × 0.12 = 0.36
- 3 × ③ = 3 × 0.01 = 0.03
- 3 × ④ = 3 × 0.12 = 0.36
- 2 × ②MOV = 2 × 0.04 = 0.08
- ②ADD# = 0.06
- 計 = 0.89 [msec]

処理時間 = [①+②MOV+③+④]
+ [①+②ADD#+③+④]
+ [①+②MOV+③+④]
= 3 × ① + 3 × ③ + 3 × ④
+ 2 × ②MOV + ②ADD#

※ 次の回路の場合は (c) と同じ計算となります。
演算ファンクションの次に1つでも接点等が入れば③はその他で計算して下さい。



● 演算ファンクションがONしていない場合は、ラダー回路命令の-()-と同じ0.66 μs/命令とパラメータの実行時間 (0.33 μs/1パラメータ) の合計となります。

3 処理時間

(3) 特殊コイルの処理時間

プロセスレジスタ (P) , ネスティングコイル (N) は、演算ファンクションと同様にONしている場合にはOSの処理が行われます。以下にそれぞれの処理時間を示します。

種 別	処理時間 [msec]
P -()	0.5
N -()	0.5

● CPU待行列が多くある場合は、多少、上記値より大きくなる場合があります。

● 各コイルがONしていない場合は $0.66 \mu s$ / 命令となります。

(4) アップダウンカウンタの処理時間

アップダウンカウンタは演算命令と同一処理となります。この場合は、演算命令の計算式の②に次の値をセットして計算してください。

カウンタ処理時間 $\approx 70 \mu s$ / 命令

(5) タイマ (T) ワンショット (U) の処理時間

タイマ及びワンショットはプログラムの有無に関係なく、100msec毎にエディション点数分だけ処理されます。

またこの100msecのタイミングはOS内部の10msec毎の割込によって図に示すように作成されます。

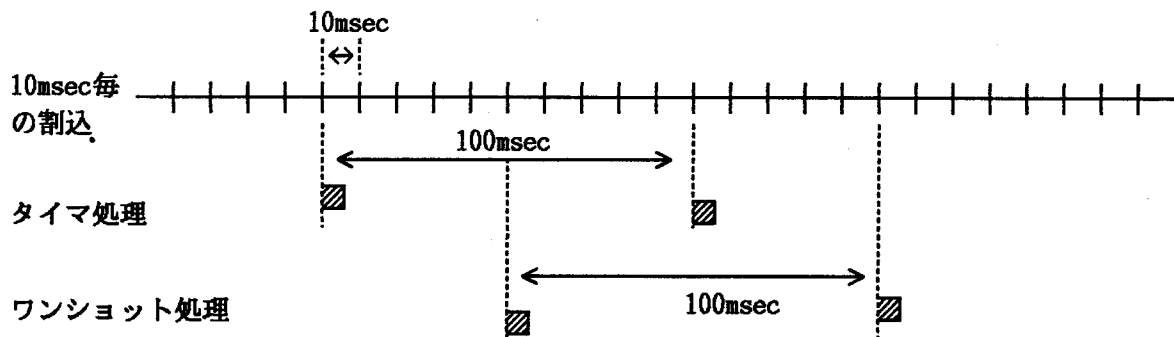


図3-2 OS処理タイミング

さて、タイマ及びワンショットは16点単位に処理されますが、それぞれの16点単位の処理時間は次のようになります。

タイマ16点の処理時間 … 250 μ s / 16点

ワンショット16点の処理時間 … 300 μ s / 16点

また10msec毎に発生するOSの基本処理時間は

10msec毎の処理時間 … 約250 μ s

従って、100msec毎に発生するそれぞれの処理時間は

$$\text{タイマ} = 250 + 250 \times \left(\frac{\text{タイマ点数}}{16} \right)$$

$$\text{ワンショット} = 250 + 300 \times \left(\frac{\text{ワンショット点数}}{16} \right)$$

となります。

- 10msecタイマを使用した場合は10msec毎に次の処理時間が必要になります。

10msecタイマ処理 = 250 + 250 = 500 μ s / 16点

3 処理時間

(6) リモートI/Oサポート時間

リモートI/Oの1サイクル分の転送が終了するとOSはリモートI/Oのエラー情報を退避した後、再起動を行います。この時のリモートI/Oの処理時間とOS処理の関係を図3.3に示します。

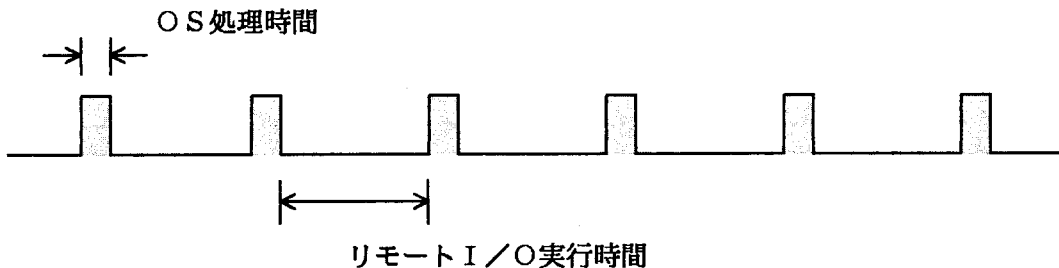


図3-3 リモートI/OとOS処理

リモートI/Oはエディション設定により点数を変更できますが、この時の点数とOSの処理時間は次の様になります。

リモートI/O点数	OS処理時間 (msec)	リモートI/O実行時間 (msec)
512	$0.2+0.2=0.4$	5.0
1024	$0.2+0.4=0.6$	10.0
1536	$0.2+0.6=0.8$	15.0
2048	$0.2+0.8=1.0$	20.0

- アナログ、パルスカウンタ等を御使用になる場合各サポートOSの処理時間を上記データに加算してください。
 なお、サポートOSの処理時間については、アナログ、パルスカウンタ等のマニュアルを参照ください。

3.2 各命令語の処理時間（4αの場合）

3.2.1 スキャンタイム

PC_sは、シーケンスRUN時I/O転送（I/Oメモリと入出力モジュール間のデータ転送）やシーケンスプログラムの実行等を、高速に繰り返し実行しています。

図3.4.1に示します様に、1回分の実行時間、1スキャンタイム（T）は、下式により求められます。

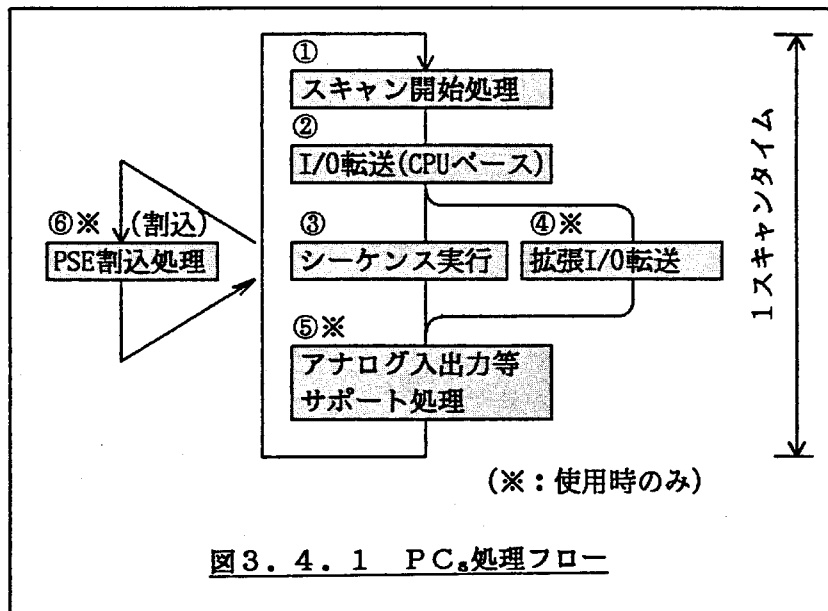


図3.4.1 PC_s処理フロー

$$T = ① + ② + \max(③, ④) + ⑤ + ⑥$$

→ ③, ④の大きい方の値を使用します。

①～⑥のそれぞれの処理時間を表3.1.1に示します。

3.2.2 スキャンタイム計算例

図3.5にスキャンタイムの計算例を示します。

（但し、CPUベースは基本8スロット、アナログ入出力等特殊I/Oモジュール、10msタイマ未使用、PSEモニタなしとします。）

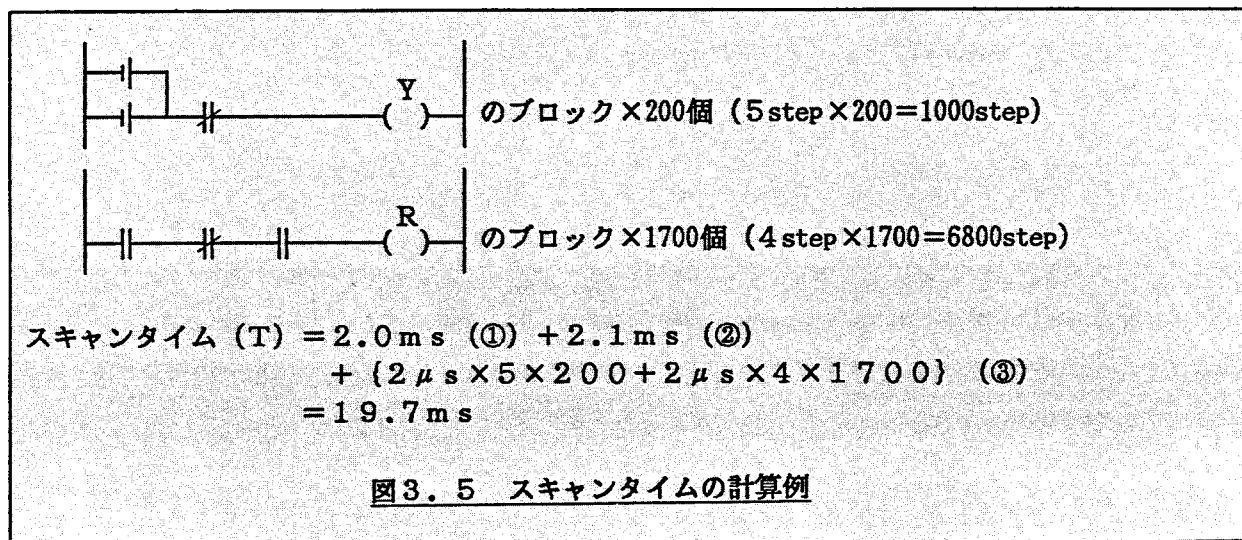


図3.5 スキャンタイムの計算例

3 処理時間

表3.1 処理時間一覧

No	項目	処理時間 (MAX)	備考	
①	スキャン開始処理	2.0 ms		
②	I/O転送 (CPUベース)	4スロット	1.1 ms	
		8スロット	2.1 ms	
CPUベース内のDI/OとのI/O転送。				
③	シ ー ケ ン ス 実 行		2 μs	
			2 μs	
			2 μs	
			2 μs	
			2 μs	
			T	(ON時) 210 μs (OFF時) 80 μs
			U	(ON時) 300 μs (OFF時) 180 μs
		CU CD CR	CU	(OFF→ON時) 240 μs
			CD	(他) 4 μs
			CR	(他) 4 μs
他	2 μs			
	(ON時)	→		
	(OFF時)	8 μs		
(演算ファンクション一覧表参照)				
④	拡張 I/O転送	CPUベース 4スロット時	11.6 ms	
		CPUベース 8スロット時	8.0 ms	
拡張スロット付CPUベースにて、拡張I/Oモジュール (LWE800又はLWE805) 実装の場合のみ。				
⑤	アナログ入出力等 サポート処理	1.2 ms	アナログ入出力、プリセットカウンタモジュール使用時のみ。	
⑥	PSE割込処理	PSEのリモート接続開始時や、読出し処理、各種モニタ処理により数ms～数100msスキャンタイムが伸びることがあります。		
⑦	その他			
	(1) 10msタイマ (2) 上位計算機	T = T × 1.1 約5ms	(10msタイマ使用時のみ) (上位計算機接続時のみ)	

3.3 平均スキャンタイムの表示

2α, 4αのプログラム平均スキャンタイムは図3.6に示すプログラムを追加することにより、PSEで見ることができます。この値はユーザプログラムRUN時、8秒ごとの平均スキャンタイムを表示することができます。

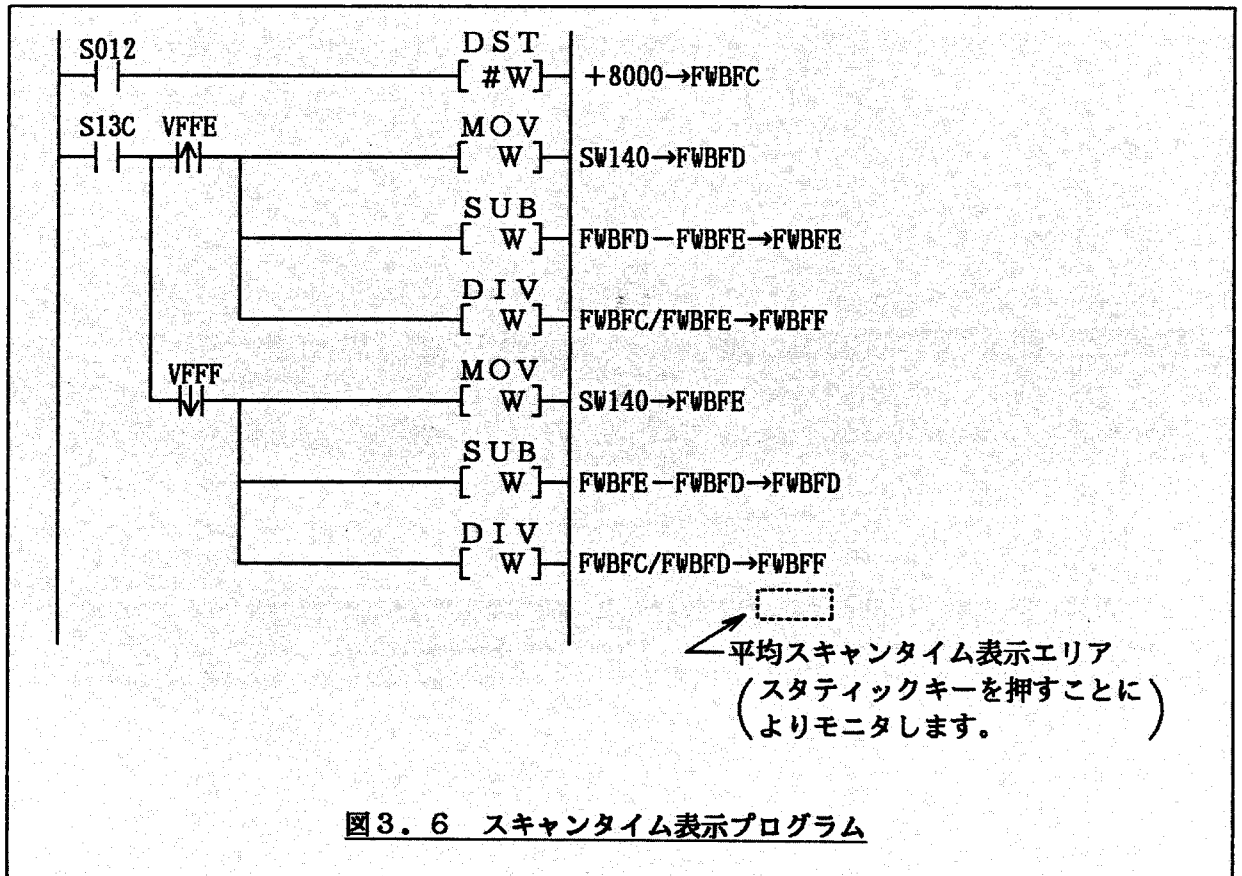
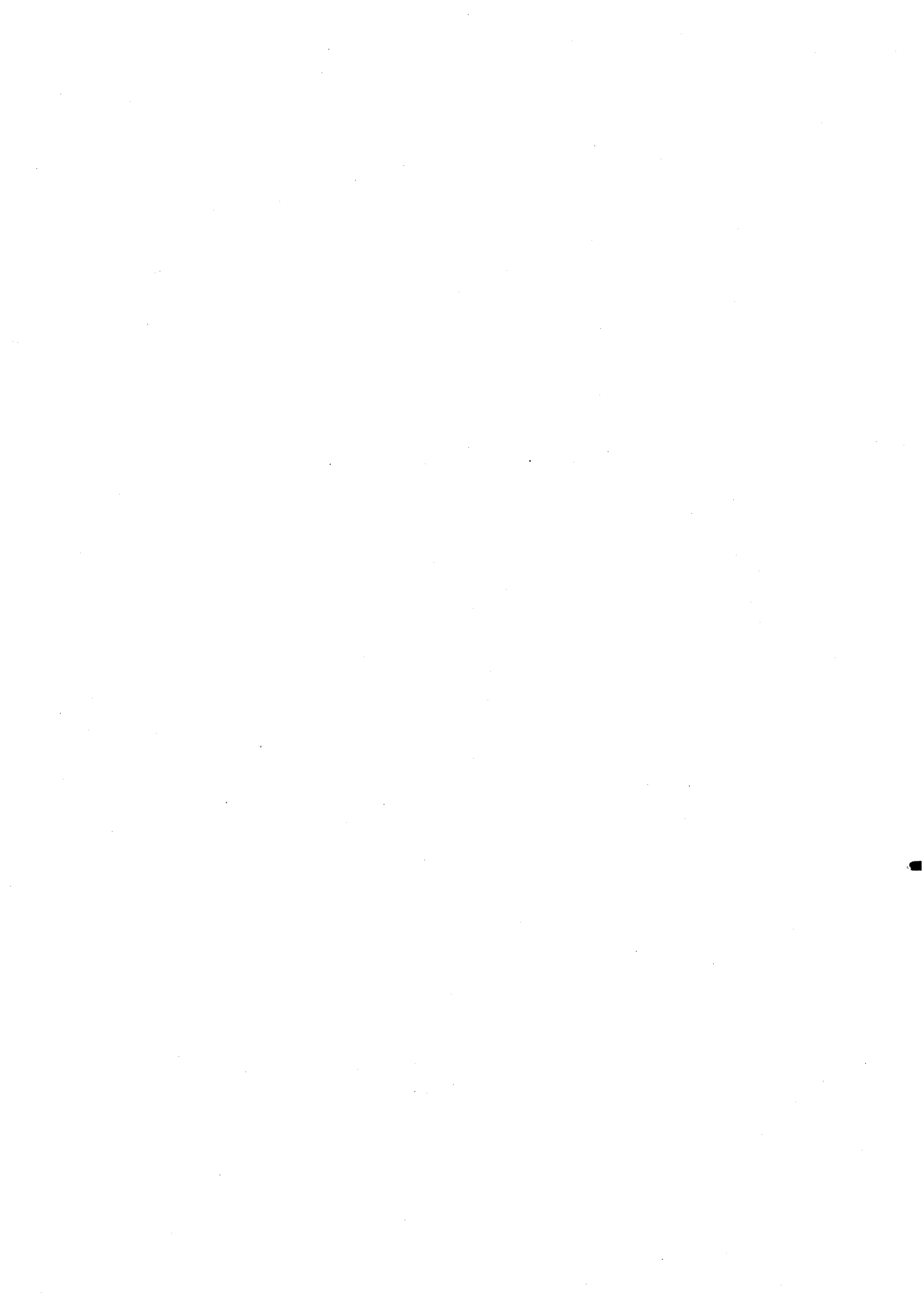


図3.6 スキャンタイム表示プログラム



ご利用者各位

〒319-1293

茨城県日立市大みか町五丁目2番1号

株式会社 日立製作所 情報制御システム事業部

お 願 い

各位にはますますご清栄のことと存じます。

さて、この資料をより良くするために、お気付きの点はどんなことでも結構ですので、下欄にご記入の上、弊社営業担当または弊社所員に、お渡しくさせていただきますようお願い申し上げます。なお、製品開発、サービス、その他についてもご意見を併記して頂ければ幸甚に存じます。

ご住所 〒	_____
貴会社名 (団体名)	_____
芳名	_____
製品名	_____
ご意見欄	_____ _____