

TOSHIBA

汎用プログラマブル コントローラ
PROSEC EX SERIES

EX2000

プログラミング説明書

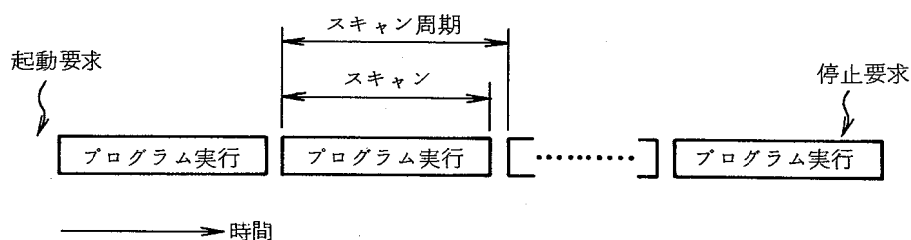
第 1 章 機能説明	4
1 - 1 プログラム実行方法	4
1 - 2 プログラム構成	5
1 - 2 - 1 メインプログラム	5
1 - 2 - 2 サブプログラム	5
1 - 2 - 3 割込プログラム	8
1 - 2 - 4 サブルーチン	8
1 - 3 プログラムページ割付	10
第 2 章 システム設計	11
2 - 1 システム設計手順	11
2 - 2 本体運転モード	12
2 - 3 デバイスとレジスタ	15
2 - 4 入出力割付	17
2 - 4 - 1 デバイス / レジスタ番号の割付	18
2 - 4 - 2 特殊指定	19
2 - 4 - 3 割込プログラム割付	19
2 - 5 入出力割付例	22
2 - 5 - 1 16 点入出力モジュールを装着した場合	22
2 - 5 - 2 32 点入出力モジュールを装着した場合	23
2 - 5 - 3 16 点 / 32 点入出力モジュールを混在し装着する場合	24
2 - 5 - 4 各入出力モジュールをレジスタとして扱う場合	25
2 - 5 - 5 特殊指定の使用方法	26
2 - 6 機能種別とデバイス / レジスタ番号	28
2 - 6 - 1 外部入力デバイス (X)	28
2 - 6 - 2 外部出力デバイス (Y)	29
2 - 6 - 3 外部入力レジスタ (XW)	29
2 - 6 - 4 外部出力レジスタ (YW)	29
2 - 6 - 5 補助リレーデバイス / レジスタ (R, RW)	29
2 - 6 - 6 データレジスタ (D)	30

2-6-7	リンクデバイス / レジスタ (Z, ZW)	30
2-6-8	タイマレジスタ (T)	32
2-6-9	カウンタレジスタ (C)	32
2-6-10	タイマデバイス (T)	32
2-6-11	カウンタデバイス (C)	32
2-6-12	特殊コイル	33
第3章 プログラム設計		51
3-1	メモリ構成	51
3-1-1	メモリ書込プロテクト機能	52
3-1-2	イニシャルロード機能	53
3-2	プログラム容量の計算	54
3-3	プログラムの実行順序	55
3-3-1	直接ジャンプ命令を含む場合の実行順序	56
3-3-2	サブルーチンコール命令を含む場合の実行順序	57
3-3-3	割込プログラム実行時の実行順序	58
3-4	データ入出力処理	59
3-5	タイマ更新処理	60
3-6	内部処理サイクル	61
3-6-1	パワーアップシーケンス	62
3-6-2	システムイニシャライズ処理	63
3-6-3	スキャンサイクル	63
3-6-4	各種自己診断処理	63
3-6-5	ウォッチドックタイマリセット	64
3-6-6	スキャン動作	65
3-6-7	HOLD入力時の処理	66
3-6-8	サブプログラム起動方法	66
3-6-9	割込プログラムの実行禁止	67
3-7	RAS情報	69

第4章 SFC (Sequential Function Chart)	70
4-1 シーケンシャル ファンクション チャート概要	70
4-2 SFC 構成要素	71
4-2-1 ステップ (STEP)	71
4-2-2 遷移 (TRANSITION)	71
4-2-3 条件分岐	73
4-2-4 並列実行	74
4-3 SFC プログラム構成	75
4-3-1 リピート実行	76
4-3-2 サブルーチン	76
4-3-3 コネクタステップ	77
4-4 SFC プログラム実行順序	78
4-5 SFC プログラム制御	80
4-5-1 起動, イニシャライズ	80
4-5-2 一時停止	80
4-5-3 ステップ処理の出力カット	80
4-6 プログラミング	81
4-7 SFC シンボル一覧	82
第5章 命 令 語	83
5-1 命令語一覧	83
5-1-1 命令語一覧表の見方	83
5-1-2 命令語一覧表	84
5-2 各命令語	90
5-2-1 各命令語仕様の見方	90
5-2-2 各命令語仕様	91

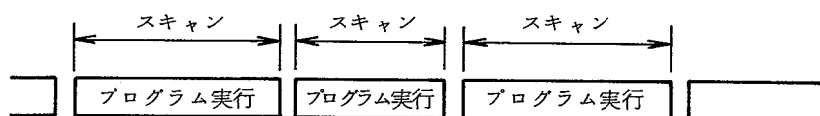
1-1 プログラム実行方法

EX2000 は、ユーザの作成したシーケンスプログラムを繰返し実行します。（スキャン実行と呼びます。）プログラムのスキャン実行は、本体モードスイッチ又はグラフィックプログラマ（GP110AP2）からの起動要求にて開始し停止要求がくるまで継続されます。

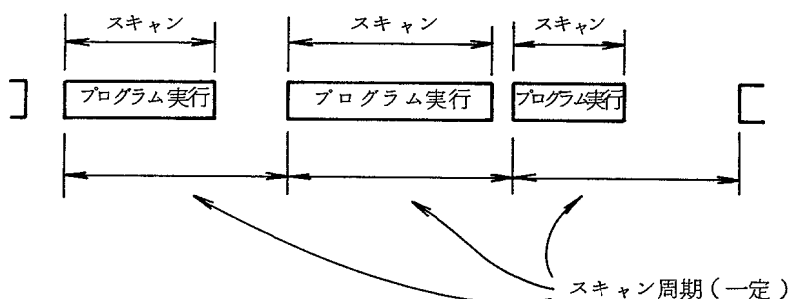


各スキャンの実行周期（スキャン周期と呼びます。）は、スキャン終了次第、直ちに次のスキャンを実行するフローティングスキャンモードと、一定の時間周期でスキャンを実行する定刻スキャンモードが設定できます。

○フローティングスキャンモード



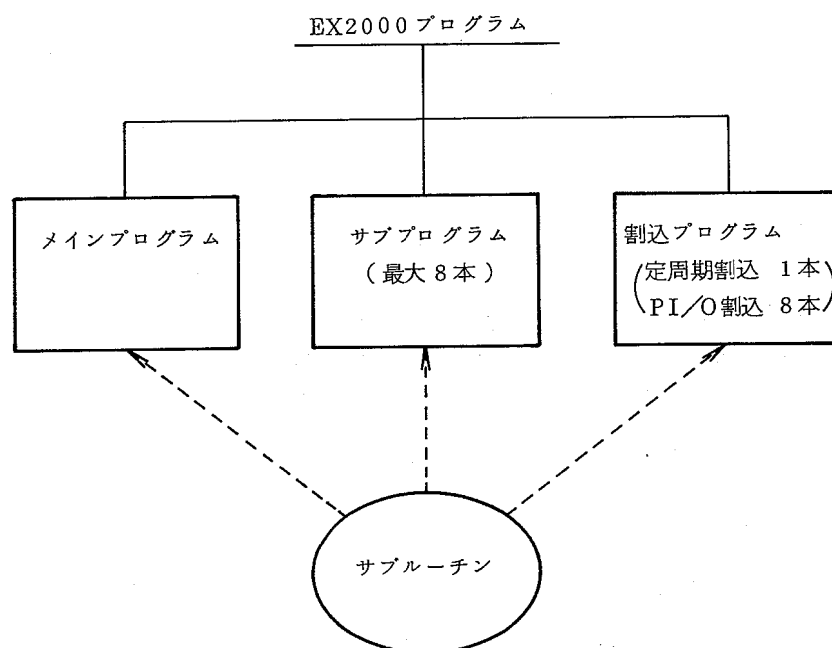
○定刻スキャンモード



1-2 プログラム構成

EX 2000 は、制御目的によりタイプの異なる複数のプログラムを同時に実行できます。（マルチプログラム機能）

EX 2000 で実行できるプログラムのタイプは下図の様に 4 種類に分けられます。



以下、各プログラムについて説明します。

1-2-1 メインプログラム

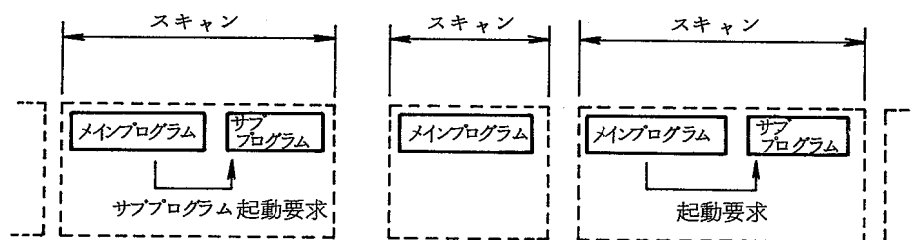
メインプログラムは、各スキャンで必ず、全部実行されるプログラムです。

基本的なシーケンス制御プログラムは、メインプログラムとする必要があります。

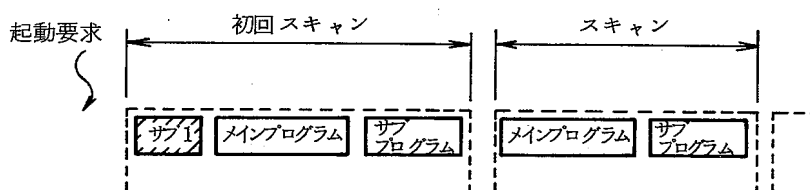
1-2-2 サブプログラム

サブプログラムは、各スキャンで実行するか否かを制御することができるプログラムです。

実行の制御は、メインプログラム、他のサブプログラム、割込プログラムから行え、メインプログラム終了後に起動要求のある場合、該当するサブプログラムを実行します。



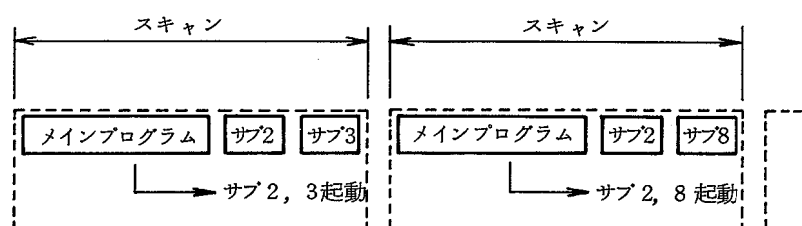
サブプログラムは、8本（サブ1～8）まで登録できますが、サブ1は初期設定用プログラムでプログラム起動時にメインプログラム実行前に1回だけ実行されます。



サブ2～8は、サブプログラム起動要求によりメインプログラム終了後実行されますが、一度に2つ以上のサブプログラムに起動要求があった場合には、優先度制御を行います。又、サブプログラムの実行時間には制限があり、制限時間以内に起動要求のあったサブプログラムが実行できない場合、次スキャン以降に実行されます。

(1) サブプログラムの優先度制御方法

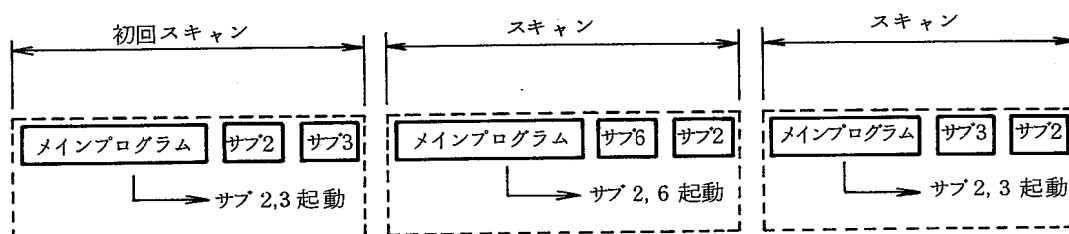
サブプログラムの優先度制御方法は、固定優先度、回転優先度の2通りの設定ができます。固定優先度では、サブプログラム番号の小さいものが常に優先度が高くなります。つまり、サブ2が一番優先度が高く、サブ8が一番優先度が低くなるので、サブ2とサブ3に起動要求があった場合には、まずサブ2が実行され、次にサブ3が実行されます。



優先度 2 → 3 → 4 → 5 → 6 → 7 → 8

2 → 3 → 4 → 5 → 6 → 7 → 8

回転優先度では、初回スキャンはサブプログラム番号の小さいものが優先度が高くなっていますが、サブプログラムの実行により、前回スキャンで最後に実行したサブプログラムの次の番号のものが、そのスキャンでは最も優先度が高くなります。



優先度 2→3→4→5→6→7→8

初回スキャンでは優先度は番号順となる。

4→5→6→7→8→2→3

前回スキャンにてサブ3まで実行したのでサブ4が最高優先度となる。

3→4→5→6→7→8→2

前回スキャンにてサブ2まで実行したのでサブ3が最高優先度となる。

。どちらの優先度制御方法でも、1回のスキャンでサブプログラム番号一巡までしか起動要求は調べません。

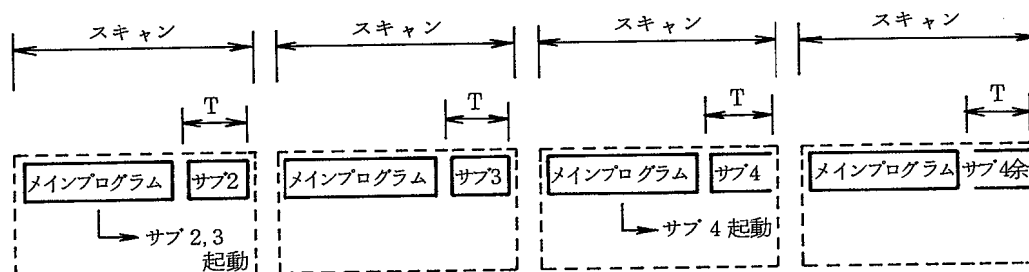
(2) サブプログラムの実行時間制限

サブプログラムの実行制限時間は、プログラムのスキャン実行モードにより異なります。

フローティングスキャンモードでは、サブプログラム実行割当て時間としてプログラマから設定します。

定刻スキャンモードでは、メインプログラム実行後のスキャン周期の余り時間となります。

。フローティングスキャンモード



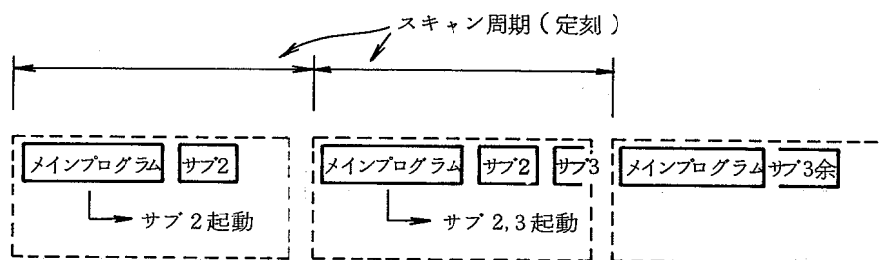
T : サブプログラム割当て時間

前回スキャンにて実行できなかったサブ3が実行される。

サブ4が制限時間内で終了しな
いたため一度中断され、2スキャンに跨って実行される。

◎割当て時間内にサブプログラム実行が全て終了した場合には、直ちに次スキャンに入ります。

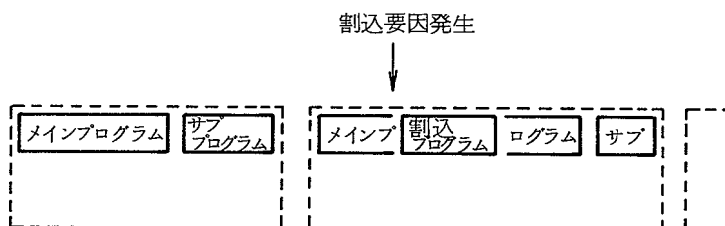
。定刻スキャンモード



スキャンの定刻周期になってもサブ3が終了しないため中断され、次スキャンで継続実行される。

1-2-3 割込プログラム

割込プログラムは、割込要因発生時に最優先度で実行されるプログラムで、メインプログラム、サブプログラムが実行中でも、一時中断して割込プログラムが実行されます。

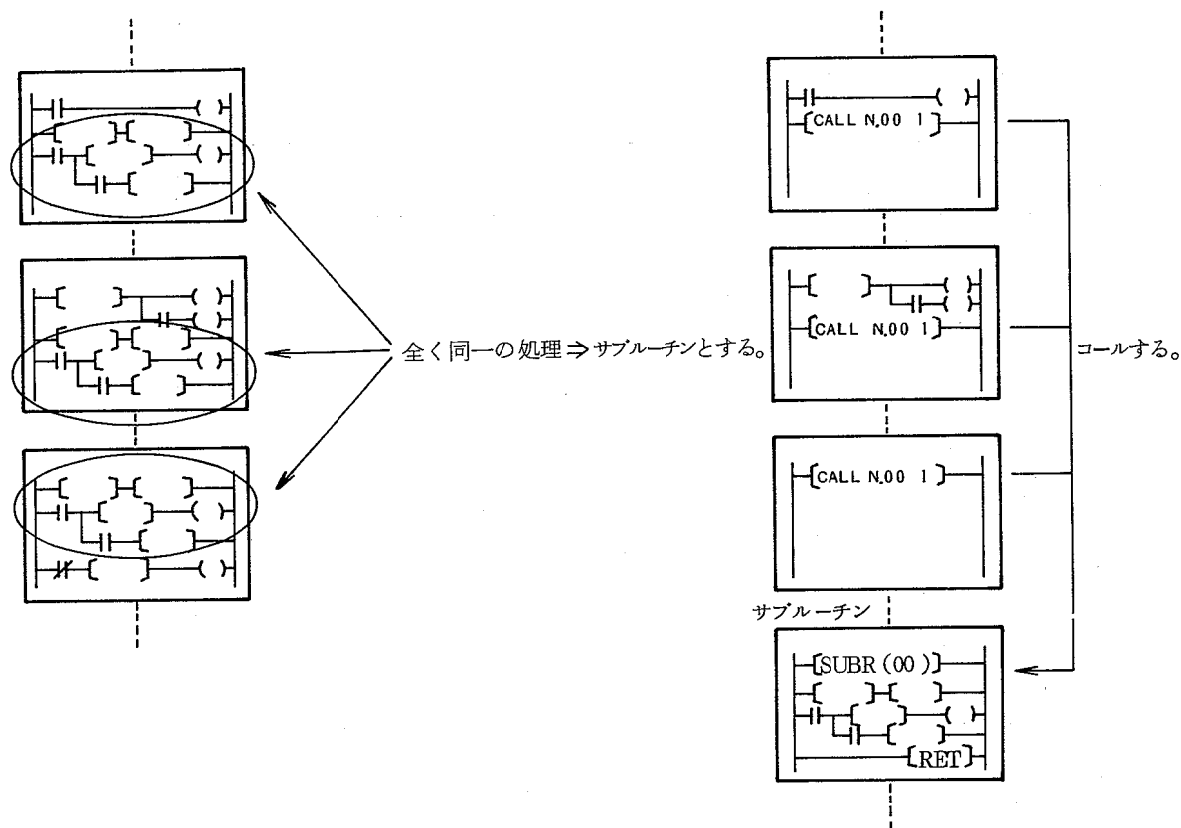


メインプログラム実行中に割込要因が発生したため中断し、優先的に割込プログラムが実行される。

割込プログラムは9本まで登録可能で、1本はEX 2000本体にて定周期で発生する要因により起動され（周期はプログラマより設定）、残りの8本は外部の割込機能付入出力カード（CDDI, PI等）からの割込要因により起動されます。

1-2-4 サブルーチン

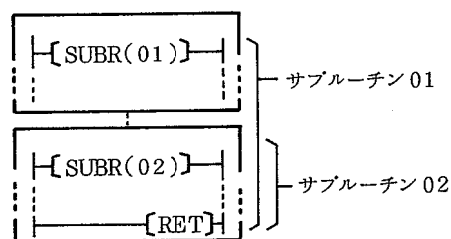
プログラム中に同じ処理が何度も繰返し使われている場合、この処理ブロックをサブルーチンとして登録すれば、サブルーチンのCALL命令を繰返し使用すればよくなり、ステップ数が短縮できます。



サブルーチンは、メインプログラム、サブプログラム、割込プログラムから呼び出すことができます。

サブルーチンからサブルーチンを呼び出すこともできます。(ただし、4重までです。)

サブルーチン中にサブルーチンを含むこともできます。(下図)



☆各プログラムタイプ間で同一サブルーチンを使用した場合に、割込プログラム又はサブプログラムの実行時間制限でサブルーチン実行中に中断され、その状態で同一サブルーチンが実行されると、中断されたサブルーチンが継続実行後、正しい演算結果が得られません。

又、サブプログラムを固定優先度制御に設定していて、かつ実行時間制限でサブプログラム実行が中断される場合にも、別々のサブプログラム間で同一サブルーチンを使用していると、正常な演算結果が得られないことがあります。

1-3 プログラムページ割付

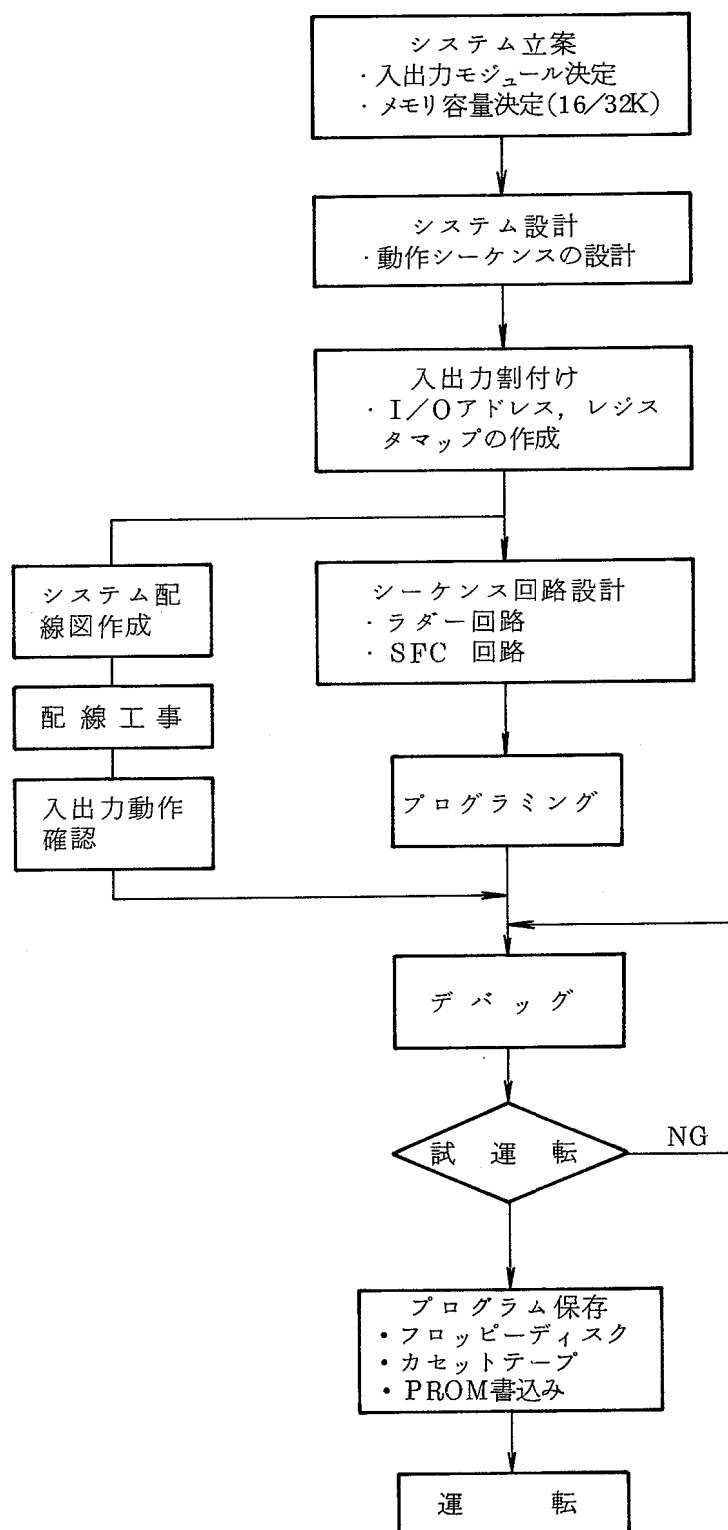
EX 2000 は、プログラムページを最大 7899 ページ持ち、下表の様なページ範囲に各プログラムタイプが割付けられています。

ページ範囲	プログラムタイプ	備考
1 ～ 1999	メインプログラム	
2000 ～ 2499	サブプログラム (サブ 1)	イニシャライズ用
2500 ～ 2999	〃 (サブ 2)	
3000 ～ 3499	〃 (サブ 3)	
3500 ～ 3999	〃 (サブ 4)	
4000 ～ 4499	〃 (サブ 5)	
4500 ～ 4999	〃 (サブ 6)	
5000 ～ 5499	〃 (サブ 7)	
5500 ～ 5999	〃 (サブ 8)	
6000 ～ 6999	サブルーチン	
7000 ～ 7099	割込プログラム (定周期)	
7100 ～ 7199	〃 (PIO 1)	
7200 ～ 7299	〃 (PIO 2)	
7300 ～ 7399	〃 (PIO 3)	
7400 ～ 7499	〃 (PIO 4)	
7500 ～ 7599	〃 (PIO 5)	
7600 ～ 7699	〃 (PIO 6)	
7700 ～ 7799	〃 (PIO 7)	
7800 ～ 7899	〃 (PIO 8)	

表のページ範囲内にプログラミングすれば、自動的にプログラムタイプが判別されます。(ただし、メイン、サブ、割込プログラムは、それぞれのページ範囲内の一番若いページから、END 命令までをプログラムと見なします。サブルーチンは、SUBR 命令から RET 命令を 1 つのサブルーチンと見なし、ページ範囲内に 100 個まで登録可能です。)

2-1 システム設計手順

EX 2000を用いたシステムの設計手順は次のフローチャートに準じて行って下さい。



※ SFC: Sequential Function Chart

(詳細は第 4 章を参照下さい。)

2-2 本体運転モード

EX 2000 本体の運転モードには、次の4つのモードがあります。

1. RUN ; 運転モード切換キースイッチ (MPU モジュールの正面にあるキースイッチ。以下モードキースイッチと呼びます。) が RUN 側に設定され、電源投入時正常なら通常の運転状態に入ります。

プログラムページの追加、削除、修正はできます (オンラインプログラミング機能) が、END, JUMP, JCS/R, MCS/R 等の、実行制御命令の追加、削除、システム情報の変更はできません。レジスタ/デバイスデータの設定は可能です。

HOLD 入力信号 (電源端子台の HOLD 入力端子) を ON すると、プログラムの実行を停止し、一括入出力動作のみ実行します。従って入力デバイス/レジスタには最新の入力情報が入り、出力デバイス/レジスタの内容は常時出力されます。またタイマーの更新も停止、保持されます。

2. SIM ; モードキースイッチが SIM 側に設定されたとき、動作は RUN のときと同じですが、出力モジュールの出力は全て OFF 状態になり、PI/O 割込も受け付けません。(但し、定周期割込は受け付けます。)

このモードは、プログラムのデバック時に出力を出したくないときなどに使用します。

3. HALT ; モードキースイッチが HALT 側に設定されたとき、または RUN 側にてプログラマより本体停止コマンド (CNTL 8 0) を実行したとき、本体は運転を停止し、出力モジュールへの出力は全て OFF となります。

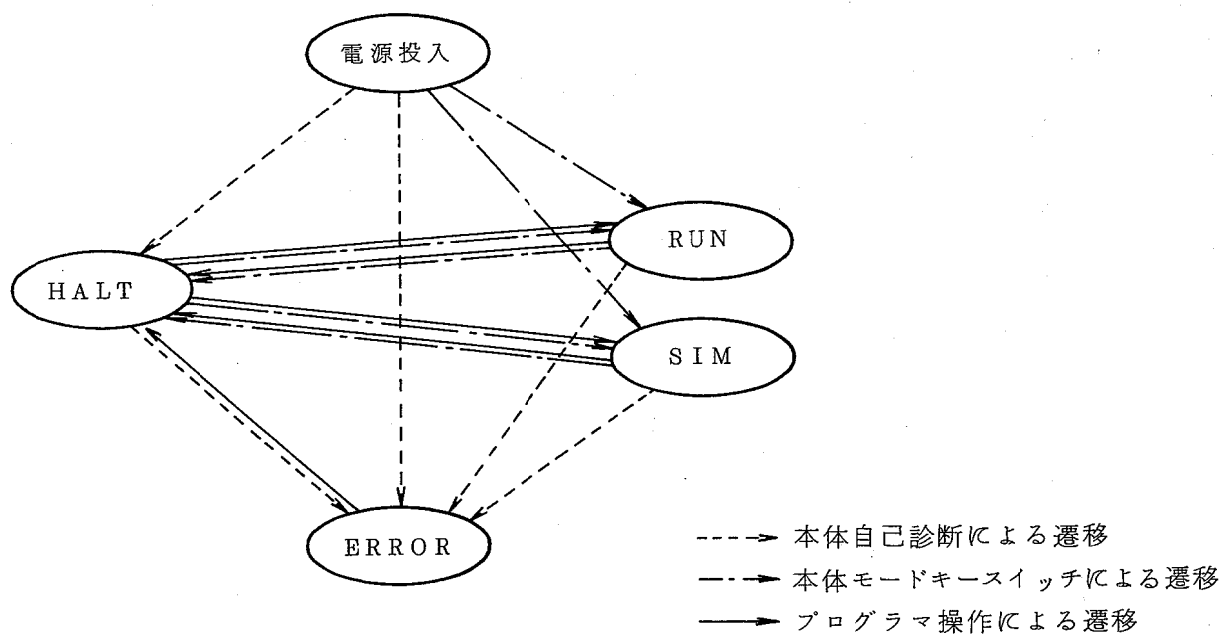
基本的にプログラムの編集はこのモードで行います。

4. ERROR ; RUN・HALT・SIM モード中に自己診断の結果、続行不能のとき ERROR モードとなります。プログラムの実行、一括入出力処理は行わず、出力モジュールへの出力は全て OFF となります。

本モードの解除は、グラフィックプログラマからのエラーリセットコマンド

(CNTL 9 3) の実行か電源再投入で可能です。

本体運転モード遷移図



本体運転モード遷移時の処理

遷移前	遷移後	遷移条件	プログラム実行	一括入出力	タイマー更新	I/O出力	ヘルシー出力	備考
電源投入	ERROR	イニシャライズ処理でエラー検出※	×	×	×	×	×	
	HALT	モードキースイッチがHALT側	×	×	×	×	×	
	RUN	モードキースイッチがRUN側	○	○	○	○	○	HOLD入力ON時、プログラム実行およびタイマー更新しない。
	SIM	モードキースイッチがSIM側	○	○	○	×	×	
HALT	RUN	モードキースイッチがHALT→RUNまたは、スイッチがRUN側でプログラマよりRUN起動	○	○	○	○	○	HOLD入力ON時、プログラム実行およびタイマー更新しない。
	SIM	モードキースイッチがHALT→SIMまたは、スイッチがSIM側でプログラマよりRUN起動	○	○	○	×	×	
	ERROR	本体自己診断でエラー検出	×	×	×	×	×	
RUN	HALT	モードキースイッチがRUN→HALTまたは、スイッチがRUN側でプログラマよりHALT実行	×	×	×	×	×	
	ERROR	本体自己診断でエラー検出	×	×	×	×	×	
SIM	HALT	モードキースイッチがSIM→HALTまたは、スイッチがSIM側でプログラマよりHALT実行	×	×	×	×	×	
	ERROR	本体自己診断でエラー検出	×	×	×	×	×	
ERROR	HALT	プログラマよりエラーリセット実行	×	×	×	×	×	

○：ON（又は実行），×：OFF（又は不実行）

※電源投入時にエラーが発生した場合、処理は次の3通りになります。

- (i) MPU モジュールハードウェアエラー：ダウンモード (" RUN ", " I/O ", " PRG " の LED 点滅)
ーの場合 となりグラフィックプログラマ操作できません。
 - (ii) SPU, メモリモジュールハードウ：ERROR モードになり、プログラマ操作もできますが、エ
ェアエラーの場合 ラーリセットできません。
 - (iii) メモリモジュールソフトウェアエラー：ERROR モードとなります。エラーリセット後メモリク
ーの場合 リアしてRUN起動できます。
- (パリティエラー、ROM BCC エラー)

注) ヘルシー出力は電源モジュール端子台より接点出力として出力されます。

2-3 デバイスとレジスタ

EX 2000 のシーケンスプログラムには接点やコイル命令のビット単位の演算や、データ転送、四則演算等のファンクション命令のレジスタ単位のデータ処理があります。

デバイス………接点やコイルの ON, OFF 情報を格納しておくところをデバイスと呼びます。デバイス(ビット) は機能種別とデバイス番号によりなります。

<機能種別>

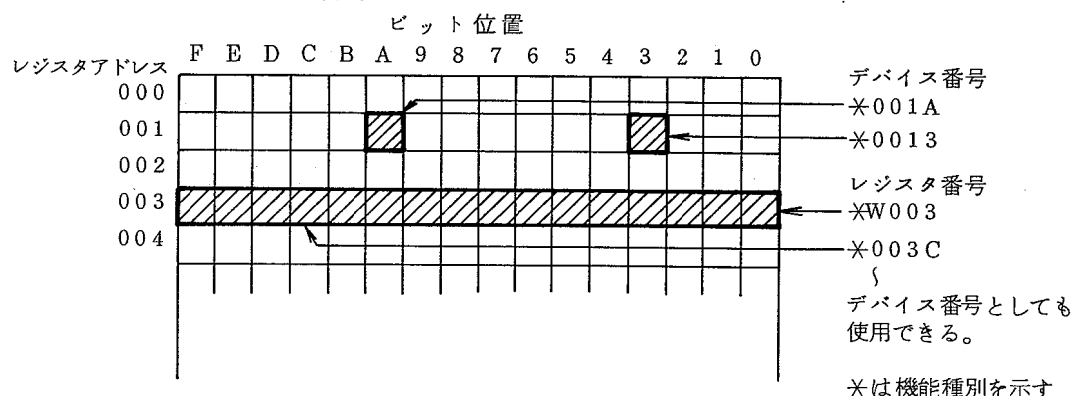
- X ; 外部入力デバイス(接点)
- Y ; 外部出力デバイス(コイル)
- R ; 補助リレーデバイス(接点, コイル)
- Z ; リンクデバイス(接点, コイル)
- T ; タイマーデバイス(接点)
- C ; カウンターデバイス(接点)

レジスタ………ファンクション命令にて使用される 16 ビットのデータを格納するところをレジスタ(ワード) 呼びます。レジスタは機能種別とレジスタ番号によりなります。

<機能種別>

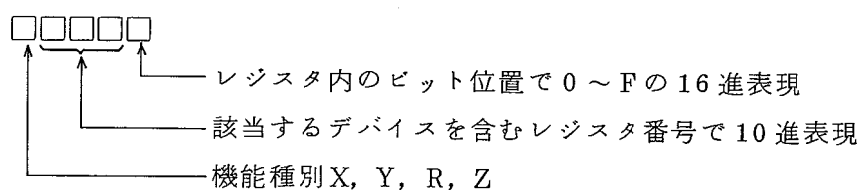
- | | | |
|----------------|---|--------------------|
| XW ; 外部入力レジスタ | } | デバイスとしても
使用できる。 |
| YW ; 外部出力レジスタ | | |
| RW ; 補助リレーレジスタ | | |
| ZW ; リンクレジスタ | | |
| D ; データレジスタ | | |
| T ; タイマレジスタ | | |
| C ; カウンタレジスタ | | |

。デバイス番号とレジスタ番号



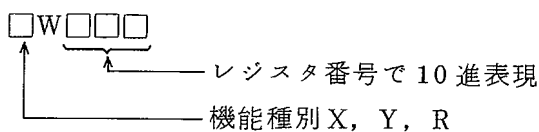
。表現方法

<デバイス>



- ・タイマデバイス T. □□□ T. 000～499
- ・カウンタデバイス C. □□□ C. 000～499

<レジスタ>



- ・リンクレジスタ ZW□□□□ ZW0000～1999
(ZW0000～ZW0999はデバイスとしても使用できます。ZW1000～1999はレジスタとしてのみ使用します。)
- ・タイマレジスタ T□□□ T000～499
- ・カウンタレジスタ C□□□ C000～499
- ・データレジスタ D□□□□□ D00000～D08191(8K語時)
D00000～D16383(16K語時)

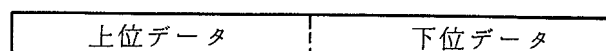
レジスタで扱える数値を次表に示します。

レジスタ種別	10進表現	10進表現符号付	16進表現
タイマレジスタ	0～32767	—	H 0000～H7FFF
カウンタレジスタ	0～65535	—	H 0000～HFFFF
単長レジスタ	0～65535	—32768～32767	H 0000～HFFFF
倍長レジスタ	0～4294967295	—2147483648 ～2147483647	H 00000000～HFFFFFFF

倍長レジスタは2つの連続した番号のレジスタを連結して使用したものです。構成は下図のようになります。

*W□□□

*W□□□+1



2-4 入出力割付

EX 2000 では、各入出力モジュールの種別とユニットへの装着位置から、機能種別を判定し、レジスタ／デバイス番号を自動的に決定します。（自動割付機能）

入出力モジュールと機能種別、レジスタ数の対応は次表の通りです。

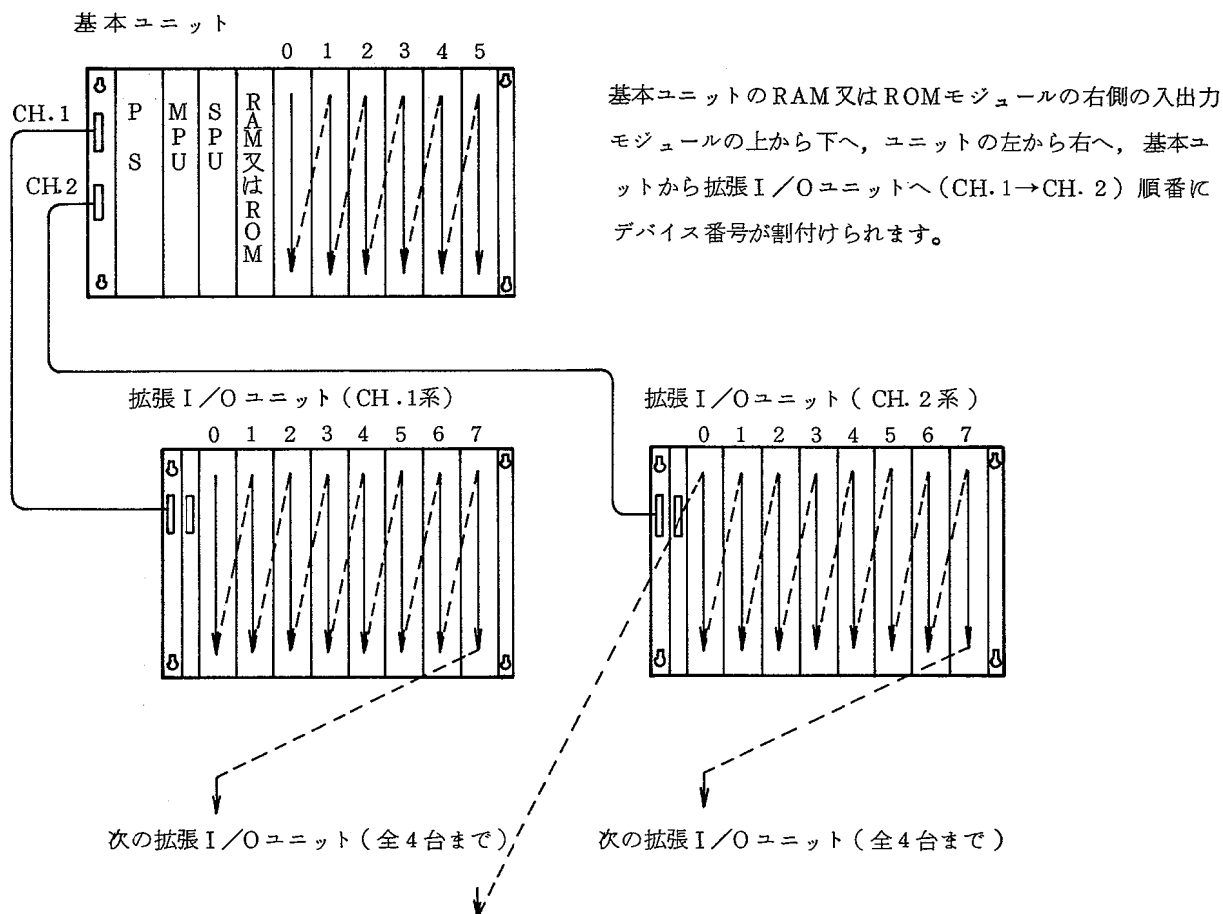
機能種別	レジスタ数	入出力モジュール種別
Y	1	16点／1レジスタ出力モジュール
Y	2	32点／2レジスタ "
Y	4	64点／4レジスタ "
Y	8	128点／8レジスタ "
X	1	16点／1レジスタ入力モジュール
X	2	32点／2レジスタ "
X	4	64点／4レジスタ "
X	8	128点／8レジスタ "
X+Y	2	32点／2レジスタ入出力モジュール
X+Y	4	64点／4レジスタ "
X+Y	8	128点／8レジスタ "
Z	8	8レジスタ リンクモジュール
Z	16	16レジスタ "
Z	32	32レジスタ "
OPT	—	各種オプションモジュール

割込機能付カード（CDDI, PI など）では機能種別に "i" が付き、"iX", "iY", "iX+Y" となります。

入出力モジュールの機能種別とレジスタ数はグラフィックプログラマ（GP110AP2）からの入出力設定コマンド（CNTL 0 5）により、EX 本体に自動登録されます。また、システム情報2（入出力種別読出し／設定画面）にて登録状態の表示及び編集ができます。

2-4-1 デバイス / レジスタ番号の割付

各入出力モジュールの入出力デバイス (X, Y) への割付けは、ユニットへの装着位置と各入出力モジュールの点数の関係で自動的に決まります。



注) 1. 各ユニットに実装した入出力モジュールは、プログラマからの入出力設定コマンド

() により自動的にデバイス/レジスタ番号が決定し、EX内部に登録されます。

登録内容はシステム情報2にて見ることができます。

2. 装着されない空スペースは無視されます。レジスタ/デバイス割付にも空スペースを設けたい場合はプログラマよりSP設定を行って下さい。(SP設定は次ページ参照下さい。)

2-4-2 特殊指定

入出力モジュールの登録には、グラフィックプログラマから次の様な特殊指定が可能です。

○スペース指定 (SP)

割付け時に空スロットに対してXYレジスタエリアを確保する場合に用いる指定で、1スロット当たり確保するエリアサイズは1, 2, 4, 8, 16, 32 Wが設定できます。

○ユニット先頭レジスタNo指定

ユニット毎に、割付ける先頭XYレジスタNoの指定ができます。

○一括入出力禁止指定

I/Oカード毎に、一括入出力の禁止をセットできます。

X, YおよびX+Y指定にiを付ける(iX, iY, iX+Y)ことにより、対応するカードは一括入出力の対象外となります。割込機能付I/Oカードは入出力設定コマンドにてi指定となります。

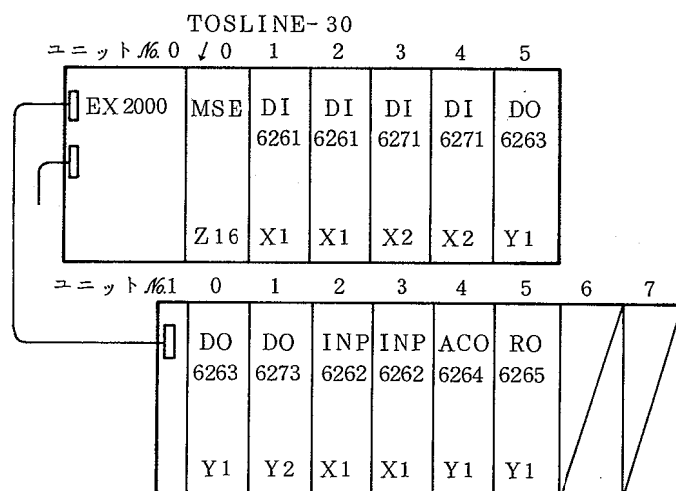
2-4-3 割込プログラムの割付け

割込機能付入出力モジュールに対するPI/O割込プログラムの割付けは、デバイス番号の割付け順で、割込機能付入出力モジュール検出順にPI/O割込プログラム1から8へ自動的に割付けられます。但し、割込機能付入出力モジュールを検出するのは基本ユニットとCH. 1系の拡張I/Oユニット(IFスロット側、オプションスロット側とも有効)のみで、CH. 2系の拡張I/Oユニットでは検出されません。(i指定はされません。)

注) 割込機能付入出力モジュール、使用時、割込機能付入出力モジュール検出前に、空スロットを設けると、割込機能が正常に働きません。但し、割込機能は基本ユニット、各CH. 1系拡張I/Oユニットで独立していますので、基本ユニットー拡張I/Oユニット又は拡張I/Oユニットー拡張I/Oユニット間で空スロットが存在してもかまいません。

つまり基本ユニットでスロット4に割込機能付入出力モジュールを実装し、スロット0～3に空スロットを設けることはできませんが、拡張I/Oユニット#1～4に実装した場合には、基本ユニットのスロット0～4に空スロットが存在しても大丈夫です。但し、拡張I/Oユニット#1～0～3に空スロットは設けられません。

(例)



入出力設定コマンド (CNTL 0 5) により自動的に機能種別, デバイス/レジスタ設定が実行され, システム情報 2 にて下記の情報が表示されます。

```

#0-0  Z  16 W
#0-1  X  01 W
#0-2  X  01 W
#0-3  X  02 W
#0-4  X  02 W
#0-5  Y  01 W
#1-0  Y  01 W
#1-1  Y  02 W
#1-2  X  01 W
#1-3  X  01 W
#1-4  Y  01 W
#1-5  Y  01 W
#1-6
#1-7

```

以上の設定により次の様に機能種別，レジスタ番号が決まります。

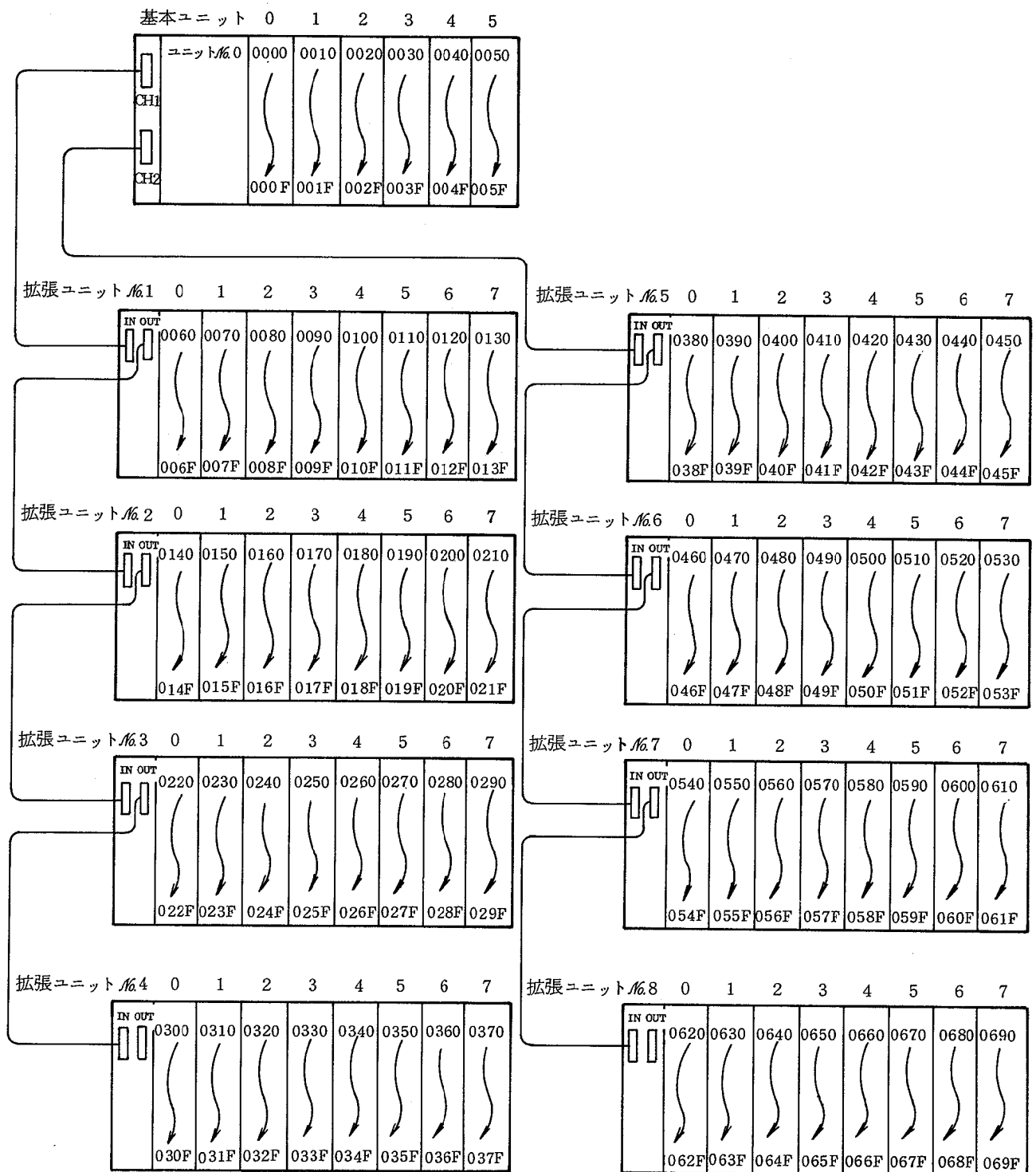
レジスタアドレス		レジスタ番号	モジュール	ユニット・ スロット番号
00		XW00	DI - 6261	# 0 - 1
01		XW01	DI - 6261	# 0 - 2
02		XW02	DI - 6271	# 0 - 3
03		XW03	}	
04		XW04	DI - 6271	# 0 - 4
05		XW05	}	
06		YW06	DO - 6263	# 0 - 5
07		YW07	DO - 6263	# 1 - 0
08		YW08	DO - 6273	# 1 - 1
09		YW09	}	
10		XW10	INP - 6262	# 1 - 2
11		XW11	INP - 6262	# 1 - 3
12		YW12	ACO - 6264	# 1 - 4
13		YW13	RO - 6265	# 1 - 5

00		ZW00	MSE	# 0 - 0
			(TOSLINE-30)	
15		ZW15		

2-5 入出力割付例

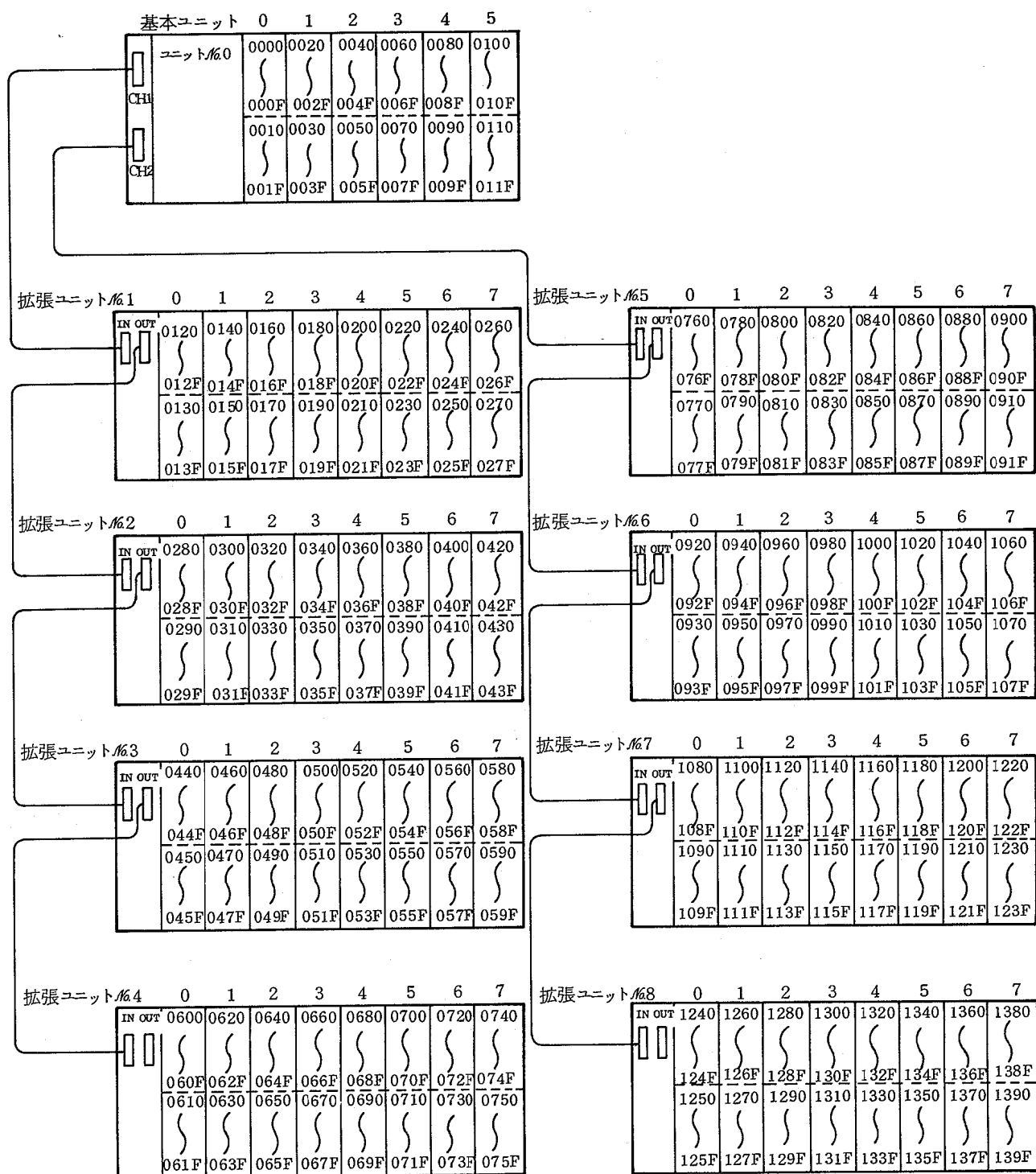
2-5-1 16点入出力モジュールを装着した場合

16点入出力モジュールを順序よくユニットへ装着した場合の装着位置とデバイス番号の関係は次のようになります。



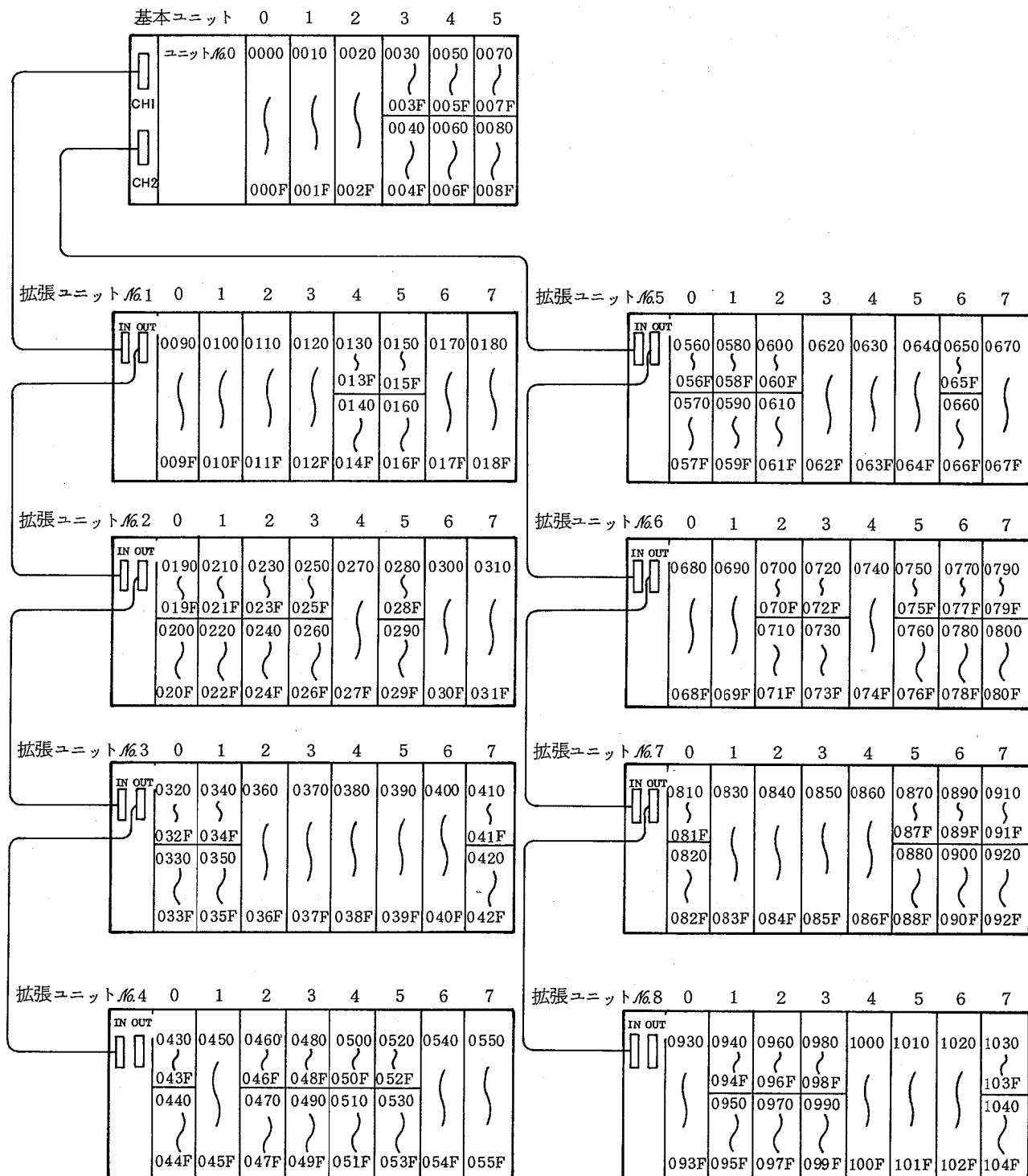
2-5-2 32点入出力モジュールを装着した場合

32点入出力モジュールを順序よくユニットへ装着した場合の装着位置とデバイス番号の関係は次のようになります。



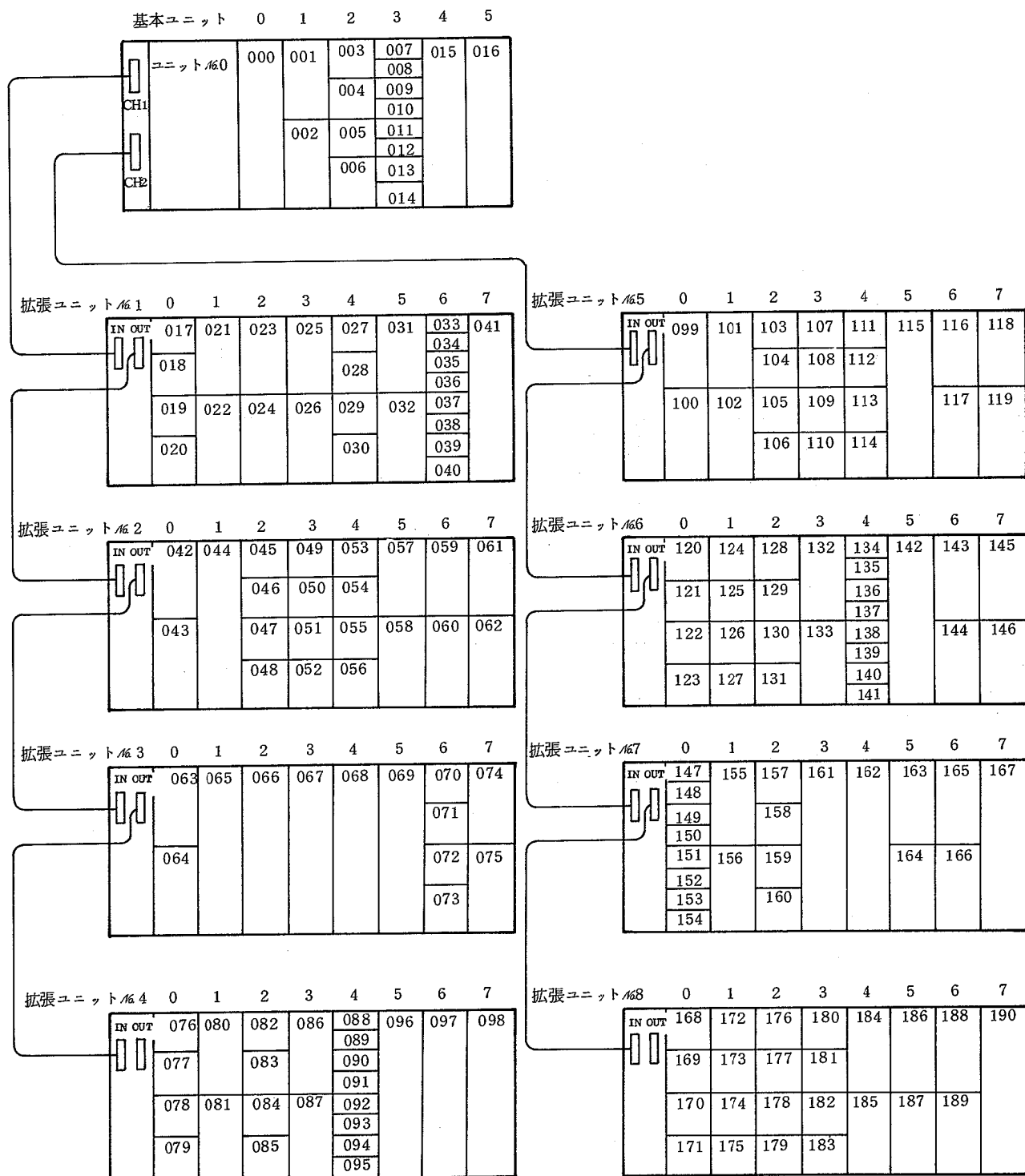
2-5-3 16点/32点入出力モジュールを混在し装着する場合

16点/32点入出力モジュールを混在し順序よくユニットへ装着する場合の装着位置とデバイス番号の関係は次のようになります。



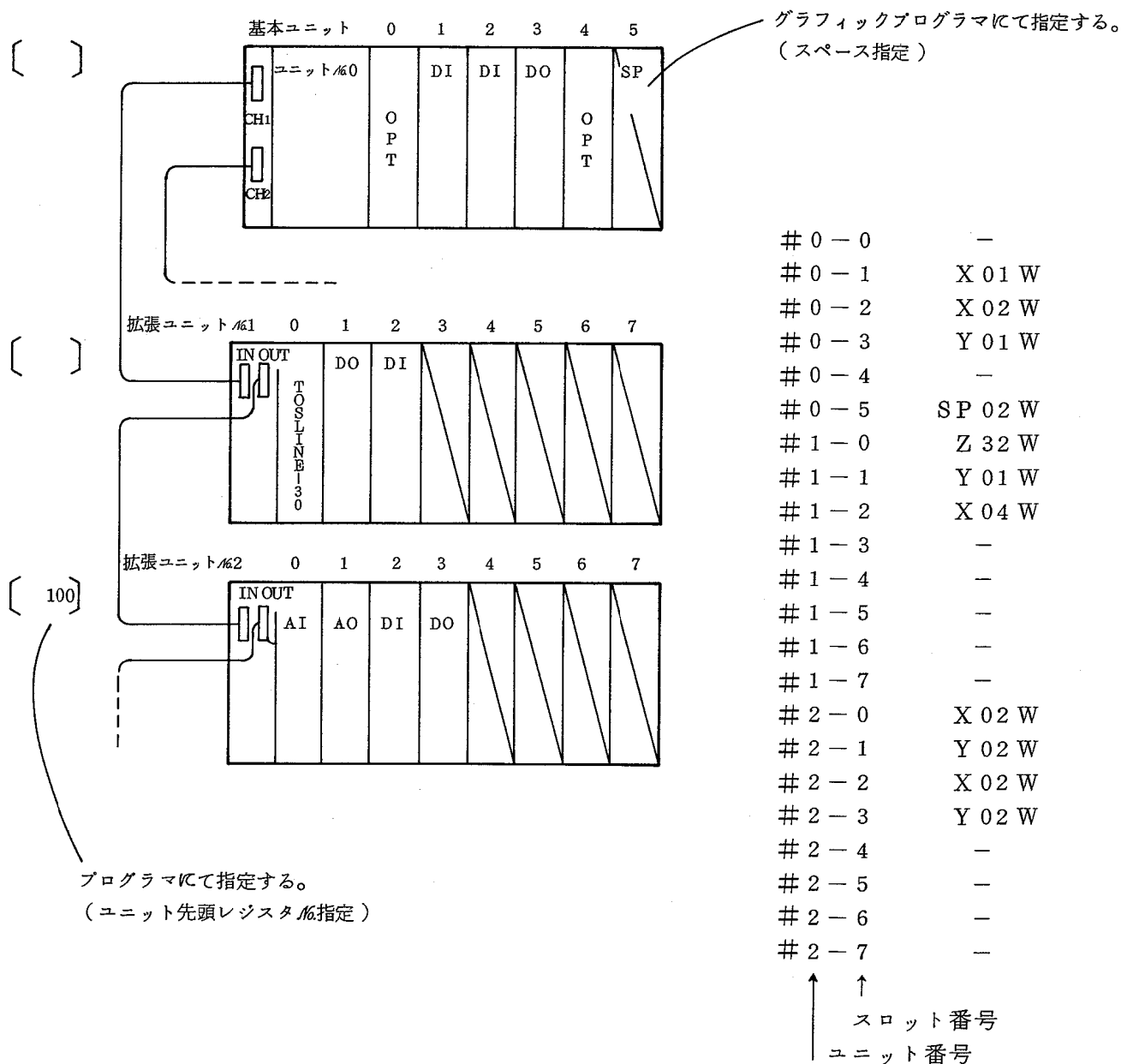
2-5-4 各入出力モジュールをレジスタとして扱う場合

各入出力モジュールをデバイスとして扱わず、レジスタとして扱う場合のユニットへの装着位置とレジスタ番号の関係は次のようになります。



2-5-5 特殊指定の使用方法

グラフィックプログラマからの指定にて、スペース指定 (SP) とユニット先頭レジスタ NO 指定を行う場合の装着位置とデバイス、レジスタ番号の関係は次のようになります。



— 27 —

2-6 機能種別とデバイス／レジスタ番号

EX 2000 は次の機能，種別とデバイス・レジスタ番号をもっています。

名 称	記号	EX 2000	
		点 数	番 号
外部入力 デバイス	X	合 計 8000 デバイス	X 0000 ～ X 499 F
外部出力 デバイス	Y		Y 0000 ～ Y 499 F
外部入力 レジスタ	XW	合 計 500 レジスタ	XW 000 ～ XW 499
外部出力 レジスタ	YW		YW 000 ～ YW 499
補助リレー デバイス	R	16000 デバイス	R 0000 ～ R 999 F (特殊コイル R 9000 ～ R 999 F)
補助リレー レジスタ	RW	1000 レジスタ	RW 000 ～ RW 999 (特殊コイル RW 900 ～ RW 999)
データレジスタ 16KS／8K 語 32KS／16K 語	D	8192 レジスタ	D 00000 ～ D 08191
		16384 レジスタ	D 00000 ～ D 16383
リンクリレー デバイス	Z	16000 デバイス	Z 0000 ～ Z 999 F
リンクリレー レジスタ	ZW	2000 レジスタ	ZW 0000 ～ ZW 1999
タイマ レジスタ	T	500 レジスタ	T 000 ～ T 499
カウンタ レジスタ	C	500 レジスタ	C 000 ～ C 499
タイマ デバイス	T.	500 デバイス	T.000 ～ T.499
カウンタ デバイス	C.	500 デバイス	C.000 ～ C.499

注 1) 各レジスタは 16 個の連続したデバイスの集合とも見なせデータレジスタ (D),
タイマ (T), カウンタ (C) のレジスタ以外はデバイスとレジスタは兼用です。

2-6-1 外部入力デバイス (X)

入力モジュールを介して入力される外部接点の ON / OFF 状態を示します。外部入力デバイスはラダー回路中何度でも使用できます。

各種入力モジュールへの割付は，16 点デバイス単位で行われます。

例) [X 1W] モジュールは X □□□ 0 ～ X □□□ F の 16 デバイスに割付けられる。

2-6-2 外部出力デバイス (Y)

出力モジュールを介して外部機器を駆動するコイルの ON / OFF 信号を格納しているのが外部出力デバイスです。

シーケンスのコイルに指定する外に演算命令の結果の格納エリアにも使用できます。各種出力モジュールへの割付は、16 点デバイス単位で行われます。

例) [Y 1W] モジュールは Y□□□ 0 ~ Y□□□ F の 16 デバイスに割付けられる。

2-6-3 外部入力レジスタ (XW)

外部機器からの数値 (アナログ入力, パルス入力, 外部設定器等) を格納する 16 ビットのメモリを外部入力レジスタと呼びます。

各種入力モジュールへの割付は、モジュールのレジスタ数により決まります。

例) [X 2W] モジュールは XW□□ n ~ XW□□ n+1 の 2 レジスタに割付けられる。

2-6-4 外部出力レジスタ (YW)

外部機器に出力する為の数値 (アナログ出力, 数値表示器等) を格納する 16 ビットのメモリを外部出力レジスタ (YW) と呼びます。

各種出力モジュールへの割付はモジュールのレジスタ数により決まります。

例) [Y 2W] モジュールは YW□□ n ~ XW□□ n+1 の 2 レジスタに割付けられる。

2-6-5 補助リレーデバイス / レジスタ (R, RW)

シーケンスの中間結果を格納するデバイスとして (R), また演算命令の結果の一時記憶用のレジスタ (RW) として使用します。

このデバイス / レジスタの内容は直接外部に出力することができません。出力する場合は一旦外部出力デバイス (Y), 外部出力レジスタ (YW) に移して下さい。

尚, 補助リレー後半の 100 レジスタにはタイミングリレーや本体自己診断結果などのシステム情報を格納したり, サブプログラムの起動要求, 割込プログラムの受付禁止設定などをする特殊リレーエリアがあり, 補助リレーとして参照や設定ができます。詳細については, 2-6-12 特殊コイルを参照して下さい。

本デバイス / レジスタは停電保持指定にて停電保持領域の指定ができます。

2-6-6 データレジスタ (D)

演算命令の結果の一時記憶用のレジスタ, 又は固定データの保存用エリアとして使用します。デバイスとしては使用できませんが, 書込プロテクトスイッチ設定でデータの書込みを禁止することができます。(書込プロテクトについては3-1-1項を参照して下さい。)

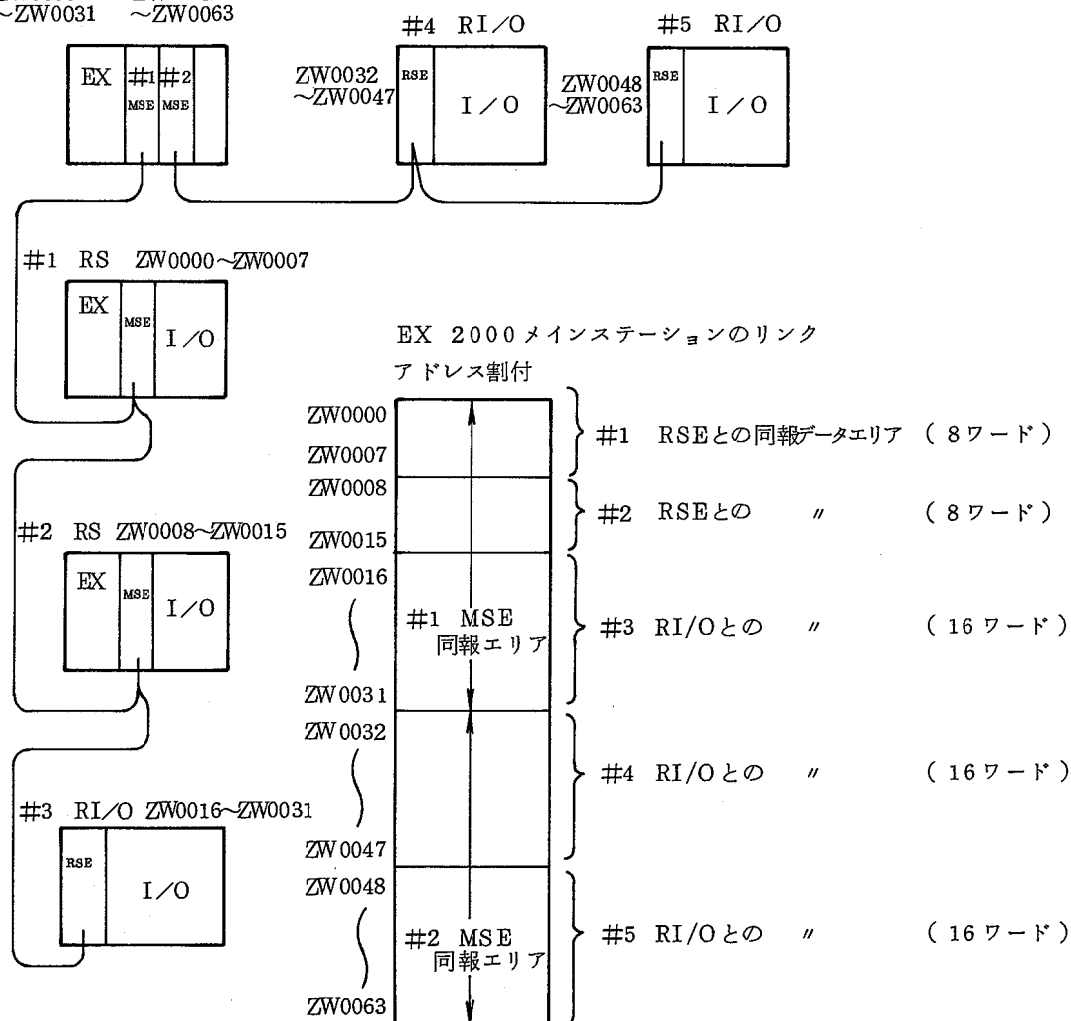
メッセージ付診断表示のメッセージ登録テーブル等に使用できます。本レジスタは停電保持指定にて停電保持領域の指定ができます。

2-6-7 リンクデバイス/レジスタ (Z, ZW)

データ伝送装置TOSLINE-30(以下TL-30), TOSLINE-2000E(以下TL-2000E)を使用してEX間の情報リンク及びリモートI/O構成を行うための情報エリアで2000レジスタ準備されています。(但し, TL-30では最大512レジスタまでの使用となります。)

MSE(TL-30)は, リンクエリアの先頭から伝送ワード数分づつ自動的に割付けられます(最大16台)

(例) ZW0000 ZW0032
~ZW0031 ~ZW0063



MSE; EXステーションモジュール
RSE; リモートI/Oステーションモジュール

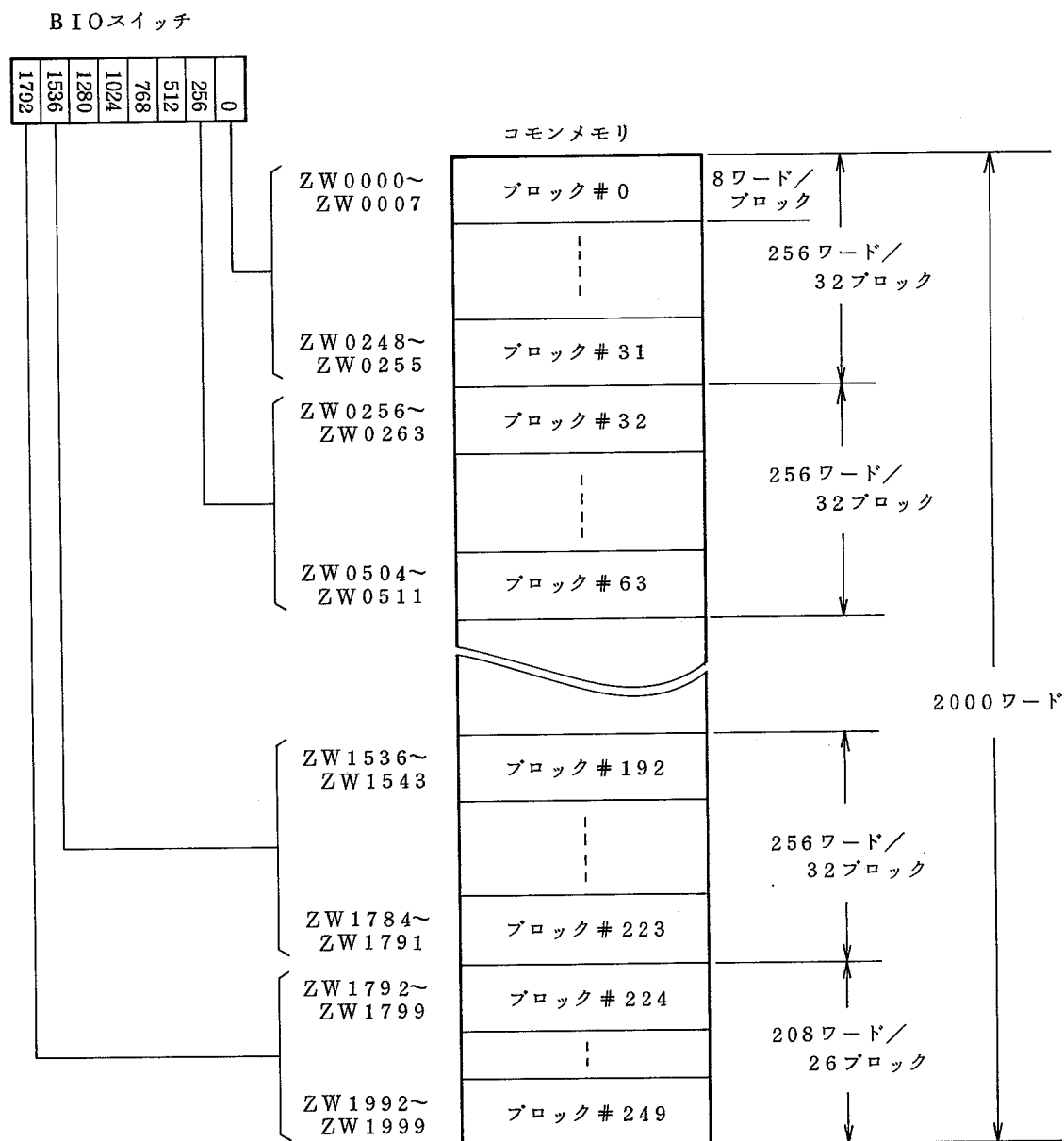
RS ; リモートステーション
RI/O; リモートI/Oステーション

TL-2000Eは、2000ワードのリンクエリアを持っていて、ZW0000～ZW1999に対応しています。

リンクレジスタに割付けるか否か（注）はTL-2000EステーションのBIOスイッチで指定します。スイッチとリンクレジスタの対応を下图に示します。

注) リンクレジスタの割付け

リンクレジスタは共有レジスタであるため、2台のステーションが実装された場合、リンクレジスタをどちらのステーションに割付けるか、又は、割付けないかをBIOスイッチで指定します。2台実装し両方とも割付けた場合、そのエリアはチャンネル#1側に割付けられます。



BIOスイッチ………スキャン伝送用コモンメモリにリンクレジスタ（ZW）を割付けるか否かを指定します。

ON：割付け許可 OFF：割付け禁止

2-6-8 タイマレジスタ (T)

オンディレイタイマ, オフディレイタイマ, シングルショット等タイマ命令の経過時間や残り時間を格納するためのレジスタです。

タイマカウント値の単位はレジスタ番号によって異なり, T 000 ~ T 449 が 100 ms 単位 (100ms タイマと呼びます。), T 450 ~ T 499 が 10 ms 単位 (10ms タイマと呼びます。) となっています。

本レジスタはデバイスとして使用できません。また演算命令の結果を格納するレジスタとして使用できません。

停電保持指定にて停電保持領域の指定ができます。

2-6-9 カウンタレジスタ (C)

カウンタの現在カウント値を格納する為のメモリがカウンタレジスタです。本レジスタはデバイスとして使用できません。また演算結果を格納するレジスタとして使用できません。

停電保持指定にて停電保持領域の指定ができます。

2-6-10 タイマデバイス (T.)

オンディレイタイマ, オフディレイタイマ, シングルショット等タイマ命令の演算結果が格納されるデバイスがタイマデバイスです。

本デバイスは参照のみ可能で, タイマ命令以外の演算結果の格納エリアとしては使用できません。

2-6-11 カウンタデバイス (C.)

カウンタ命令の演算結果が格納されるデバイスがカウンタデバイスです。

本デバイスは参照のみ可能で, カウンタ命令以外の演算結果の格納エリアとしては使用できません。

2-6-12 特殊コイル

補助リレーデバイス／レジスタの R 9000 / RW 900 以降は、EX 本体の各種自己診断結果の格納エリアや、サブプログラム、割込プログラムの制御情報の設定エリアとして用いられています。

(1) 特殊コイルエリアレジスタ構成

レジスタ No.	名 称	備 考
RW 900 { RW 945	TOSLINE-2000E ステータスエリア	
RW 946 { RW 977	TOSLINE-30 リンクデータ ステータスエリア	
RW 978	予 備	
RW 979	"	
RW 980	割込プログラム制御エリア	
RW 981	サブプログラム制御エリア	
RW 982	SFC プログラム制御エリア	
RW 983	PC リンク書込みプロテクト	SEND 命令を参照下さい
RW 984	予 備	
RW 985	カレンダー 年 (西暦下 2 桁)	下位 8 ビットに 2 桁 BCD データにて格納
RW 986	" 月	
RW 987	" 日	
RW 988	" 時	
RW 989	" 分	
RW 990	" 秒	
RW 991	タイミングデバイス	
RW 992	システムステータス 1	
RW 993	システムステータス 2	
RW 994 { RW 999	オプションモジュール用ステータスエ リア	

(2) TL-2000 E ステータスエリア (参照のみ)

レジスタ№	名 称	備 考
RW 900 } RW 915	TL-2000 E CH. 1 スキャンステータス	
RW 916 } RW 931	TL-2000 E CH. 2 スキャンステータス	
RW 932 } RW 935	TL-2000 E CH. 1 オンラインステータス (STN№ 1 ~ 63)	
RW 936 } RW 939	TL-2000 E CH. 2 オンラインステータス (STN№ 1 ~ 63)	
RW 940 } RW 945	(将来用予備)	

(3) TL-2000E スキャンステータス(参照のみ)

レジスタNo	名 称	備 考
RW900	TL-2000E ブロック 0～ 15 スキャン正常	伝送装置CH.1 スキャンステータス 1:正常、0:スキャンデータ更新無 ・ TL-2000E ブロック n は $ZW_n \times 8 \sim ZW_n \times 8 + 7$ の8ワード分のステータスを示す。 ・ ブロック n のステータスは $RW900 + (n/16 \text{ の整数部})$ のビット $(n/16 \text{ の余り})$ が示す。
RW901	" ブロック 16～ 31 "	
RW902	" ブロック 32～ 47 "	
RW903	" ブロック 48～ 63 "	
RW904	" ブロック 64～ 79 "	
RW905	" ブロック 80～ 95 "	
RW906	" ブロック 96～111 "	
RW907	" ブロック 112～127 "	
RW908	" ブロック 128～143 "	
RW909	" ブロック 144～159 "	
RW910	" ブロック 160～175 "	
RW911	" ブロック 176～191 "	
RW912	" ブロック 192～207 "	
RW913	" ブロック 208～223 "	
RW914	" ブロック 224～239 "	
RW915	" ブロック 240～255 "	
RW916	TL-2000E ブロック 0～ 15 スキャン正常	伝送装置CH.2 スキャンステータス 1:正常、0:スキャンデータ更新無 ・ TL-2000E ブロック o は $ZW_n \times 8 \sim ZW_n \times 8 + 7$ の8ワード分のステータスを示す。 ・ ブロック n のステータスは $RW900 + (n/16 \text{ の整数部})$ のビット $(n/16 \text{ の余り})$ が示す。
RW917	" ブロック 16～ 31 "	
RW918	" ブロック 32～ 47 "	
RW919	" ブロック 48～ 63 "	
RW920	" ブロック 64～ 79 "	
RW921	" ブロック 80～ 95 "	
RW922	" ブロック 96～111 "	
RW923	" ブロック 112～127 "	
RW924	" ブロック 128～143 "	
RW925	" ブロック 144～159 "	
RW926	" ブロック 160～175 "	
RW927	" ブロック 176～191 "	
RW928	" ブロック 192～207 "	
RW929	" ブロック 208～223 "	
RW930	" ブロック 224～239 "	
RW931	" ブロック 240～255 "	

・スキャンステータスの更新は一括入出力実行後に行われます。従って、HALTモード時は更新されません。

又、入出力設定画面にて伝送装置に対して「OPT」指定がない場合、一括入出力は行いませんので、本ステータスの更新も行われません。

(4) TL-2000E オンラインステータス (参照のみ)

レジスタNo	名 称		備 考
RW932	TL-2000E	ステーション 1~15 オンラインステータス	伝送装置CH.1 ステータス 1：オンライン，0：オフライン ・ステーション1~63 RW932+nのビットmは ステーション n×16+mのステータ スを示す。
RW933	〃	〃 16~31 〃	
RW934	〃	〃 32~47 〃	
RW935	〃	〃 48~63 〃	
RW936	〃	〃 1~15 オンラインステータス	伝送装置CH.2 ステータス 1：オンライン，0：オフライン ・ステーション1~63 RW936+nのビットmは ステーション n×16mのステータ スを示す。
RW937	〃	〃 16~31 〃	
RW938	〃	〃 32~47 〃	
RW939	〃	〃 48~63 〃	

- ・本ステータスは、自局がオンラインのとき100ms周期で更新設定されます。自局がオフライン又はH/W異常が発生した場合には、更新しません。(システムステータスの伝送装置異常R9932がセットされています。)
- ・ステーション0は存在しません。

(5) TOSLINE - 30 リンクデータステータスエリア (参照のみ)

デバイスNo.	名 称	備 考
R 9460	データリンク正常 ZW0000	• 伝送モジュール正常にてON。 • 伝送エラーのときOFFで正常になると自動的にONとなる。 • 該当リンクレジスタがリンク用に使用されていない場合OFF。
R 9461	" ZW0001	
R 9462	" ZW0002	
R 9463	" ZW0003	
R 9464	" ZW0004	
R 9465	" ZW0005	
R 9466	" ZW0006	
R 9467	" ZW0007	
R 9468	" ZW0008	
R 9469	" ZW0009	
R 946A	" ZW0010	
R 946B	" ZW0011	
R 946C	" ZW0012	
R 946D	" ZW0013	
R 946E	" ZW0014	
R 946F	" ZW0015	
R 9470	" ZW0016	
R 9471	" ZW0017	
R 9472	" ZW0018	
R 9473	" ZW0019	
R 9474	" ZW0020	
R 9475	" ZW0021	
R 9476	" ZW0022	
R 9477	" ZW0023	
R 9478	" ZW0024	
R 9479	" ZW0025	
R 947A	" ZW0026	
R 947B	" ZW0027	
R 947C	" ZW0028	
R 947D	" ZW0029	
R 947E	" ZW0030	
R 947F	" ZW0031	
R 9480	" ZW0032	
R 9481	" ZW0033	
R 9482	" ZW0034	
R 9483	" ZW0035	
R 9484	" ZW0036	
R 9485	" ZW0037	
R 9486	" ZW0038	
R 9487	" ZW0039	
R 9488	" ZW0040	

(次ページへ続く)

(続 き)

R 9489	データリンク正常	ZW 0041	• 伝送モジュール正常にて ON。 • 伝送エラーのとき OFF で、正常になると自動的に ON となる。 • 該当リンクレジスタがリンク用に使用されていなければ OFF。
R 948A	"	ZW 0042	
R 948B	"	ZW 0043	
R 948C	"	ZW 0044	
R 948D	"	ZW 0045	
R 948E	"	ZW 0046	
R 948F	"	ZW 0047	
R 9490	"	ZW 0048	
R 9491	"	ZW 0049	
R 9492	"	ZW 0050	
R 9493	"	ZW 0051	
R 9494	"	ZW 0052	
R 9495	"	ZW 0053	
R 9496	"	ZW 0054	
R 9497	"	ZW 0055	
R 9498	"	ZW 0056	
R 9499	"	ZW 0057	
R 949A	"	ZW 0058	
R 949B	"	ZW 0059	
R 949C	"	ZW 0060	
R 949D	"	ZW 0061	
R 949E	"	ZW 0062	
R 949F	"	ZW 0063	
R 9500	"	ZW 0064	
R 9501	"	ZW 0065	
R 9502	"	ZW 0066	
R 9503	"	ZW 0067	
R 9504	"	ZW 0068	
R 9505	"	ZW 0069	
R 9506	"	ZW 0070	
R 9507	"	ZW 0071	
R 9508	"	ZW 0072	
R 9509	"	ZW 0073	
R 950A	"	ZW 0074	
R 950B	"	ZW 0075	
R 950C	"	ZW 0076	
R 950D	"	ZW 0077	
R 950E	"	ZW 0078	
R 950F	"	ZW 0079	
R 9510	"	ZW 0080	
R 9511	"	ZW 0081	
R 9512	"	ZW 0082	

(次 ページへ続く)

(続き)

R 9513	データリンク正常	ZW0083	・伝送モジュール正常にてON。 ・伝送エラーのときOFFで、正常になると自動的にONとなる。 ・該当リンクレジスタがリンク用に使用されていなければOFF。
R 9514	"	ZW0084	
R 9515	"	ZW0085	
R 9516	"	ZW0086	
R 9517	"	ZW0087	
R 9518	"	ZW0088	
R 9519	"	ZW0089	
R 951A	"	ZW0090	
R 951B	"	ZW0091	
R 951C	"	ZW0092	
R 951D	"	ZW0093	
R 951E	"	ZW0094	
R 951F	"	ZW0095	
R 9520	"	ZW0096	
R 9521	"	ZW0097	
R 9522	"	ZW0098	
R 9523	"	ZW0099	
R 9524	"	ZW0100	
R 9525	"	ZW0101	
R 9526	"	ZW0102	
R 9527	"	ZW0103	
R 9528	"	ZW0104	
R 9529	"	ZW0105	
R 952A	"	ZW0106	
R 952B	"	ZW0107	
R 952C	"	ZW0108	
R 952D	"	ZW0109	
R 952E	"	ZW0110	
R 952F	"	ZW0111	
R 9530	"	ZW0112	
R 9531	"	ZW0113	
R 9532	"	ZW0114	
R 9533	"	ZW0115	
R 9534	"	ZW0116	
R 9535	"	ZW0117	
R 9536	"	ZW0118	
R 9537	"	ZW0119	
R 9538	"	ZW0120	
R 9539	"	ZW0121	
R 953A	"	ZW0122	
R 953B	"	ZW0123	
R 953C	"	ZW0124	
R 953D	"	ZW0125	

(次ページへ続く)

(続き)

R 953E	データリンク正常	ZW0126	・伝送モジュール正常にてON。 ・伝送エラーのときOFFで、正常になると自動的にONとなる。 ・該当リンクレジスタがリンク用に使用されていなければOFF。
R 953F	"	ZW0127	
R 9540	"	ZW0128	
R 9541	"	ZW0129	
R 9542	"	ZW0130	
R 9543	"	ZW0131	
R 9544	"	ZW0132	
R 9545	"	ZW0133	
R 9546	"	ZW0134	
R 9547	"	ZW0135	
R 9548	"	ZW0136	
R 9549	"	ZW0137	
R 954A	"	ZW0138	
R 954B	"	ZW0139	
R 954C	"	ZW0140	
R 954D	"	ZW0141	
R 954E	"	ZW0142	
R 954F	"	ZW0143	
R 9550	"	ZW0144	
R 9551	"	ZW0145	
R 9552	"	ZW0146	
R 9553	"	ZW0147	
R 9554	"	ZW0148	
R 9555	"	ZW0149	
R 9556	"	ZW0150	
R 9557	"	ZW0151	
R 9558	"	ZW0152	
R 9559	"	ZW0153	
R 955A	"	ZW0154	
R 955B	"	ZW0155	
R 955C	"	ZW0156	
R 955D	"	ZW0157	
R 955E	"	ZW0158	
R 955F	"	ZW0159	
R 9560	"	ZW0160	
R 9561	"	ZW0161	
R 9562	"	ZW0162	
R 9563	"	ZW0163	
R 9564	"	ZW0164	
R 9565	"	ZW0165	
R 9566	"	ZW0166	
R 9567	"	ZW0167	
R 9568	"	ZW0168	

(次ページへ続く)

(続き)

R 9569	データリンク正常	ZW 0169	• 伝送モジュール正常にてON。 • 伝送エラーのときOFFで、正常になると自動的にONとなる。 • 該当リンクレジスタがリンク用に使用されていなければOFF。
R 956A	"	ZW 0170	
R 956B	"	ZW 0171	
R 956C	"	ZW 0172	
R 956D	"	ZW 0173	
R 956E	"	ZW 0174	
R 956F	"	ZW 0175	
R 9570	"	ZW 0176	
R 9571	"	ZW 0177	
R 9572	"	ZW 0178	
R 9573	"	ZW 0179	
R 9574	"	ZW 0180	
R 9575	"	ZW 0181	
R 9576	"	ZW 0182	
R 9577	"	ZW 0183	
R 9578	"	ZW 0184	
R 9579	"	ZW 0185	
R 957A	"	ZW 0186	
R 957B	"	ZW 0187	
R 957C	"	ZW 0188	
R 957D	"	ZW 0189	
R 957E	"	ZW 0190	
R 957F	"	ZW 0191	
R 9580	"	ZW 0192	
R 9581	"	ZW 0193	
R 9582	"	ZW 0194	
R 9583	"	ZW 0195	
R 9584	"	ZW 0196	
R 9585	"	ZW 0197	
R 9586	"	ZW 0198	
R 9587	"	ZW 0199	
R 9588	"	ZW 0200	
R 9589	"	ZW 0201	
R 958A	"	ZW 0202	
R 958B	"	ZW 0203	
R 958C	"	ZW 0204	
R 958D	"	ZW 0205	
R 958E	"	ZW 0206	
R 958F	"	ZW 0207	
R 9590	"	ZW 0208	
R 9591	"	ZW 0209	
R 9592	"	ZW 0210	
R 9593	"	ZW 0211	

(次ページへ続く)

(続き)

R 9594	データリンク正常	ZW0212	• 伝送モジュール正常にてON。 • 伝送エラーのときOFFで、正常になると自動的にONとなる。 • 該当リンクレジスタがリンク用に使用されなければOFF。
R 9595	"	ZW0213	
R 9596	"	ZW0214	
R 9597	"	ZW0215	
R 9598	"	ZW0216	
R 9599	"	ZW0217	
R 959A	"	ZW0218	
R 959B	"	ZW0219	
R 959C	"	ZW0220	
R 959D	"	ZW0221	
R 959E	"	ZW0222	
R 959F	"	ZW0223	
R 9600	"	ZW0224	
R 9601	"	ZW0225	
R 9602	"	ZW0226	
R 9603	"	ZW0227	
R 9604	"	ZW0228	
R 9605	"	ZW0229	
R 9606	"	ZW0230	
R 9607	"	ZW0231	
R 9608	"	ZW0232	
R 9609	"	ZW0233	
R 960A	"	ZW0234	
R 960B	"	ZW0235	
R 960C	"	ZW0236	
R 960D	"	ZW0237	
R 960E	"	ZW0238	
R 960F	"	ZW0239	
R 9610	"	ZW0240	
R 9611	"	ZW0241	
R 9612	"	ZW0242	
R 9613	"	ZW0243	
R 9614	"	ZW0244	
R 9615	"	ZW0245	
R 9616	"	ZW0246	
R 9617	"	ZW0247	
R 9618	"	ZW0248	
R 9619	"	ZW0249	
R 961A	"	ZW0250	
R 961B	"	ZW0251	
R 961C	"	ZW0252	
R 961D	"	ZW0253	
R 961E	"	ZW0254	

(次ページへ続く)

(続き)

R 961F	データリンク正常	ZW 0255	• 伝送モジュール正常にてON。 • 伝送エラーのときOFFで、正常になると自動的にONとなる。 • 該当リンクレジスタがリンク用に使用されていなければOFF。
R 9620	"	ZW 0256	
R 9621	"	ZW 0257	
R 9622	"	ZW 0258	
R 9623	"	ZW 0259	
R 9624	"	ZW 0260	
R 9625	"	ZW 0261	
R 9626	"	ZW 0262	
R 9627	"	ZW 0263	
R 9628	"	ZW 0264	
R 9629	"	ZW 0265	
R 962A	"	ZW 0266	
R 962B	"	ZW 0267	
R 962C	"	ZW 0268	
R 962D	"	ZW 0269	
R 962E	"	ZW 0270	
R 962F	"	ZW 0271	
R 9630	"	ZW 0272	
R 9631	"	ZW 0273	
R 9632	"	ZW 0274	
R 9633	"	ZW 0275	
R 9634	"	ZW 0276	
R 9635	"	ZW 0277	
R 9636	"	ZW 0278	
R 9637	"	ZW 0279	
R 9638	"	ZW 0280	
R 9639	"	ZW 0281	
R 963A	"	ZW 0282	
R 963B	"	ZW 0283	
R 963C	"	ZW 0284	
R 963D	"	ZW 0285	
R 963E	"	ZW 0286	
R 963F	"	ZW 0287	
R 9640	"	ZW 0288	
R 9641	"	ZW 0289	
R 9642	"	ZW 0290	
R 9643	"	ZW 0291	
R 9644	"	ZW 0292	
R 9645	"	ZW 0293	
R 9646	"	ZW 0294	
R 9647	"	ZW 0295	
R 9648	"	ZW 0296	

(次ページへ続く)

(続き)

R 9649	データリンク正常	ZW 0297	• 伝送モジュール正常にて ON。 • 伝送エラーのとき OFF で，正常になると自動的に ON となる。 • 該当リンクレジスタがリンク用に使用されていなければ OFF。
R 964A	"	ZW 0298	
R 964B	"	ZW 0299	
R 964C	"	ZW 0300	
R 964D	"	ZW 0301	
R 964E	"	ZW 0302	
R 964F	"	ZW 0303	
R 9650	"	ZW 0304	
R 9651	"	ZW 0305	
R 9652	"	ZW 0306	
R 9653	"	ZW 0307	
R 9654	"	ZW 0308	
R 9655	"	ZW 0309	
R 9656	"	ZW 0310	
R 9657	"	ZW 0311	
R 9658	"	ZW 0312	
R 9659	"	ZW 0313	
R 965A	"	ZW 0314	
R 965B	"	ZW 0315	
R 965C	"	ZW 0316	
R 965D	"	ZW 0317	
R 965E	"	ZW 0318	
R 965F	"	ZW 0319	
R 9660	"	ZW 0320	
R 9661	"	ZW 0321	
R 9662	"	ZW 0322	
R 9663	"	ZW 0323	
R 9664	"	ZW 0324	
R 9665	"	ZW 0325	
R 9666	"	ZW 0326	
R 9667	"	ZW 0327	
R 9668	"	ZW 0328	
R 9669	"	ZW 0329	
R 966A	"	ZW 0330	
R 966B	"	ZW 0331	
R 966C	"	ZW 0332	
R 966D	"	ZW 0333	
R 966E	"	ZW 0334	
R 966F	"	ZW 0335	
R 9670	"	ZW 0336	
R 9671	"	ZW 0337	
R 9672	"	ZW 0338	
R 9673	"	ZW 0339	

(次ページへ続く)

(続き)

R 9674	データリンク正常	ZW 0340	• 伝送モジュール正常にて ON。 • 伝送エラーのとき OFF で、正常になると自動的に ON となる。 • 該当リンクレジスタがリンク用に使用されていなければ OFF。
R 9675	"	ZW 0341	
R 9676	"	ZW 0342	
R 9677	"	ZW 0343	
R 9678	"	ZW 0344	
R 9679	"	ZW 0345	
R 967A	"	ZW 0346	
R 967B	"	ZW 0347	
R 967C	"	ZW 0348	
R 967D	"	ZW 0349	
R 967E	"	ZW 0350	
R 967F	"	ZW 0351	
R 9680	"	ZW 0352	
R 9681	"	ZW 0353	
R 9682	"	ZW 0354	
R 9683	"	ZW 0355	
R 9684	"	ZW 0356	
R 9685	"	ZW 0357	
R 9686	"	ZW 0358	
R 9687	"	ZW 0359	
R 9688	"	ZW 0360	
R 9689	"	ZW 0361	
R 968A	"	ZW 0362	
R 968B	"	ZW 0363	
R 968C	"	ZW 0364	
R 968D	"	ZW 0365	
R 968E	"	ZW 0366	
R 968F	"	ZW 0367	
R 9690	"	ZW 0368	
R 9691	"	ZW 0369	
R 9692	"	ZW 0370	
R 9693	"	ZW 0371	
R 9694	"	ZW 0372	
R 9695	"	ZW 0373	
R 9696	"	ZW 0374	
R 9697	"	ZW 0375	
R 9698	"	ZW 0376	
R 9699	"	ZW 0377	
R 969A	"	ZW 0378	
R 969B	"	ZW 0379	
R 969C	"	ZW 0380	
R 969D	"	ZW 0381	
R 969E	"	ZW 0382	

(次ページへ続く)

(続き)

R 969F	データリンク正常	ZW 0383	• 伝送モジュール正常にて ON。 • 伝送エラーのとき OFF で、正常になると自動的に ON となる。 • 該当リンクレジスタがリンク用に使用されていなければ OFF。
R 9700	"	ZW 0384	
R 9701	"	ZW 0385	
R 9702	"	ZW 0386	
R 9703	"	ZW 0387	
R 9704	"	ZW 0388	
R 9705	"	ZW 0389	
R 9706	"	ZW 0390	
R 9707	"	ZW 0391	
R 9708	"	ZW 0392	
R 9709	"	ZW 0393	
R 970A	"	ZW 0394	
R 970B	"	ZW 0395	
R 970C	"	ZW 0396	
R 970D	"	ZW 0397	
R 970E	"	ZW 0398	
R 970F	"	ZW 0399	
R 9710	"	ZW 0400	
R 9711	"	ZW 0401	
R 9712	"	ZW 0402	
R 9713	"	ZW 0403	
R 9714	"	ZW 0404	
R 9715	"	ZW 0405	
R 9716	"	ZW 0406	
R 9717	"	ZW 0407	
R 9718	"	ZW 0408	
R 9719	"	ZW 0409	
R 971A	"	ZW 0410	
R 971B	"	ZW 0411	
R 971C	"	ZW 0412	
R 971D	"	ZW 0413	
R 971E	"	ZW 0414	
R 971F	"	ZW 0415	
R 9720	"	ZW 0416	
R 9721	"	ZW 0417	
R 9722	"	ZW 0418	
R 9723	"	ZW 0419	
R 9724	"	ZW 0420	
R 9725	"	ZW 0421	
R 9726	"	ZW 0422	
R 9727	"	ZW 0423	
R 9728	"	ZW 0424	
R 9729	"	ZW 0425	

(次ページへ続く)

(続き)

R 972A	データリンク正常	ZW 0426	• 伝送モジュール正常にてON。 • 伝送エラーのときOFFで、正常になると自動的にONとなる。 • 該当リンクレジスタがリンク用に使用されてなければOFF。
R 972B	"	ZW 0427	
R 972C	"	ZW 0428	
R 972D	"	ZW 0429	
R 972E	"	ZW 0430	
R 972F	"	ZW 0431	
R 9730	"	ZW 0432	
R 9731	"	ZW 0433	
R 9732	"	ZW 0434	
R 9733	"	ZW 0435	
R 9734	"	ZW 0436	
R 9735	"	ZW 0437	
R 9736	"	ZW 0438	
R 9737	"	ZW 0439	
R 9738	"	ZW 0440	
R 9739	"	ZW 0441	
R 973A	"	ZW 0442	
R 973B	"	ZW 0443	
R 973C	"	ZW 0444	
R 973D	"	ZW 0445	
R 973E	"	ZW 0446	
R 973F	"	ZW 0447	
R 9740	"	ZW 0448	
R 9741	"	ZW 0449	
R 9742	"	ZW 0450	
R 9743	"	ZW 0451	
R 9744	"	ZW 0452	
R 9745	"	ZW 0453	
R 9746	"	ZW 0454	
R 9747	"	ZW 0455	
R 9748	"	ZW 0456	
R 9749	"	ZW 0457	
R 974A	"	ZW 0458	
R 974B	"	ZW 0459	
R 974C	"	ZW 0460	
R 974D	"	ZW 0461	
R 974E	"	ZW 0462	
R 974F	"	ZW 0463	
R 9750	"	ZW 0464	
R 9751	"	ZW 0465	
R 9752	"	ZW 0466	
R 9753	"	ZW 0467	
R 9754	"	ZW 0468	

(次ページへ続く)

(続き)

R 9755	データリンク正常	ZW 0469	• 伝送モジュール正常にて ON。 • 伝送エラーのとき OFF で、正常になると自動的に ON となる。 • 該当リンクレジスタがリンク用に使用されてなければ OFF。
R 9756	"	ZW 0470	
R 9757	"	ZW 0471	
R 9758	"	ZW 0472	
R 9759	"	ZW 0473	
R 975A	"	ZW 0474	
R 975B	"	ZW 0475	
R 975C	"	ZW 0476	
R 975D	"	ZW 0477	
R 975E	"	ZW 0478	
R 975F	"	ZW 0479	
R 9760	"	ZW 0480	
R 9761	"	ZW 0481	
R 9762	"	ZW 0482	
R 9763	"	ZW 0483	
R 9764	"	ZW 0484	
R 9765	"	ZW 0485	
R 9766	"	ZW 0486	
R 9767	"	ZW 0487	
R 9768	"	ZW 0488	
R 9769	"	ZW 0489	
R 976A	"	ZW 0490	
R 976B	"	ZW 0491	
R 976C	"	ZW 0492	
R 976D	"	ZW 0493	
R 976E	"	ZW 0494	
R 976F	"	ZW 0495	
R 9770	"	ZW 0496	
R 9771	"	ZW 0497	
R 9772	"	ZW 0498	
R 9773	"	ZW 0499	
R 9774	"	ZW 0500	
R 9775	"	ZW 0501	
R 9776	"	ZW 0502	
R 9777	"	ZW 0503	
R 9778	"	ZW 0504	
R 9779	"	ZW 0505	
R 977A	"	ZW 0506	
R 977B	"	ZW 0507	
R 977C	"	ZW 0508	
R 977D	"	ZW 0509	
R 977E	"	ZW 0510	
R 977F	"	ZW 0511	

(6) 割込プログラム制御エリア (参照/書込み可能)

デバイスNO.	名 称	備 考
R 9800	割込禁止/許可 割込プログラム(定周期)	OFFにて割込許可。
R 9801	" " (PIO1)	ONにて割込禁止。
R 9802	" " (PIO2)	※割込禁止中に発生した割込要因は捨てられます。
R 9803	" " (PIO3)	
R 9804	" " (PIO4)	
R 9805	" " (PIO5)	
R 9806	" " (PIO6)	
R 9807	" " (PIO7)	
R 9808	" " (PIO8)	
R 9809	不 使 用	
R 980A	"	
R 980B	"	
R 980C	"	
R 980D	"	
R 980E	"	
R 980F	"	

(7) サブプログラム制御エリア (参照/書込み可能)

デバイスNO.	名 称	備 考
R 9810	サブプログラム制御モード	OFF:固定優先度, ON:回転優先度
R 9811	不 使 用	
R 9812	サブプログラム 2 起動要求	ONにて起動要求。 ※要求受付にて自動的に OFF します。
R 9813	" 3 "	
R 9814	" 4 "	
R 9815	" 5 "	
R 9816	" 6 "	
R 9817	" 7 "	
R 9818	" 8 "	
R 9819	" 2 実行ステータス	ONにて実行中。(中断中も含む) ※本エリアは書込み禁止。
R 981A	" 3 "	
R 981B	" 4 "	
R 981C	" 5 "	
R 981D	" 6 "	
R 981E	" 7 "	
R 981F	" 8 "	

(8) SFC プログラム制御エリア (参照/書込み可能)

デバイスNO.	名 称	備 考
R 9820	SFC ステップ歩進禁止/許可	OFF:歩進許可, ON:歩進禁止
R 9821 R 982F	(将来用予備)	

※SFCについては第4章参照下さい。

(9) タイミングデバイス (参照のみ)

デバイスNO.	名 称	備 考
R 9910	タイミングリレー 0.1 S	• バイナリカウンタ構成 (下図) 0.1 S (1 S) 0.2 S (2 S) 0.4 S (4 S) 0.8 S (8 S)
R 9911	" 0.2 S	
R 9912	" 0.4 S	
R 9913	" 0.8 S	
R 9914	" 1.0 S	
R 9915	" 2.0 S	
R 9916	" 4.0 S	
R 9917	" 8.0 S	
R 9918	不 使 用	
R 9919		
R 991A		
R 991B		
R 991C		
R 991D		
R 991E	常時 OFF 接点	
R 991F	常時 ON 接点	

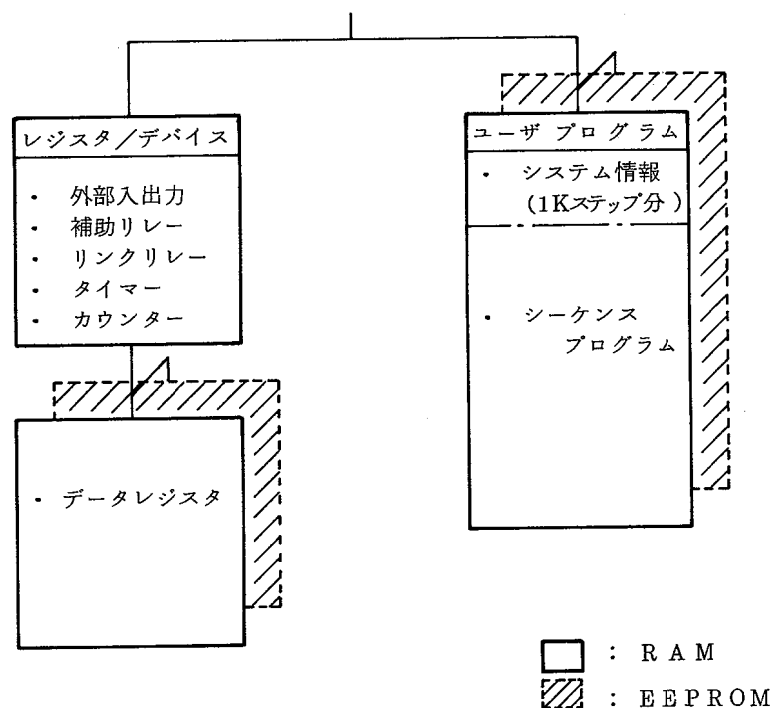
(10) システムステータス 1.2 (参照のみ)

デバイスNO.	名 称	備 考
R 9920	SPU モジュール異常	SPU ハード異常, 演算機能異常。
R 9921	カレンダー異常	カレンダー LSI 読出し時エラー発生。(カレンダー更新停止中。)
R 9922	(将来用予備)	
R 9923	I/O 異常	I/O バス異常, I/O モジュール異常。
R 9924	I/O 照合異常	設定 I/O と実装 I/O が不一致。
R 9925	プログラム異常	イリーガル命令検出, プログラム内容異常。
R 9926	メモリパリティエラー	メモリパリティエラー発生。
R 9927	(将来用予備)	
R 9928	I/O 無応答	I/O モジュール異常, I/O 未実装。
R 9929	スキャンタイムオーバー	スキャンタイムが 250ms を越えた。
R 992A	スキャン実行渋滞	定刻スキャンモードで定刻スキャンをオーバーしている。
R 992B	TOSLINE 異常	TOSLINE-30にて異常を検出した。
R 992C	プログラマ伝送異常	プログラマとの交信で異常を検出した。
R 992D	コンピュータリンク異常	コンピュータとの交信で異常を検出した。
R 992E	故障診断命令実行	故障診断表示命令実行にて ON。
R 992F	バッテリー電圧異常	バッテリー電圧が低下, または未装着。
R 9930	本体制御モード	0: HALTモード 1: RUNモード
R 9931		2: ERRORモード 3: SIMモード
R 9932	伝送装置異常	オフライン又は異常(CH1とCH2のオア)を検出した
R 9933	" 割付け オーバーラップ	スキャンデータ割付けがCH1とCH2で重複

※ R 9920 ~ F, R 9932 は, 自己診断にてエラー発生時 ON となります。R 992A ~ F, 9932 は正常時自動復帰します。

3-1 メモリ構成

レジスタ／デバイスデータ，およびユーザープログラムはメモリモジュールに格納されています。
メモリモジュール内は下図の様にデータメモリとプログラムメモリから構成されています。



メモリモジュール内構成

メモリモジュールは形式によりデータレジスタ，ユーザープログラム容量が違います。また，タイプにRAMとROMがあり，ROMタイプのメモリモジュールではEEPROM（電氣的消去可能プログラマブルリードオンリメモリ）を内蔵していますので，プログラマからのコマンドでPROMにデータレジスタ，ユーザプログラムの内容を保存できます。

次表にメモリモジュールの形式と容量を示します。

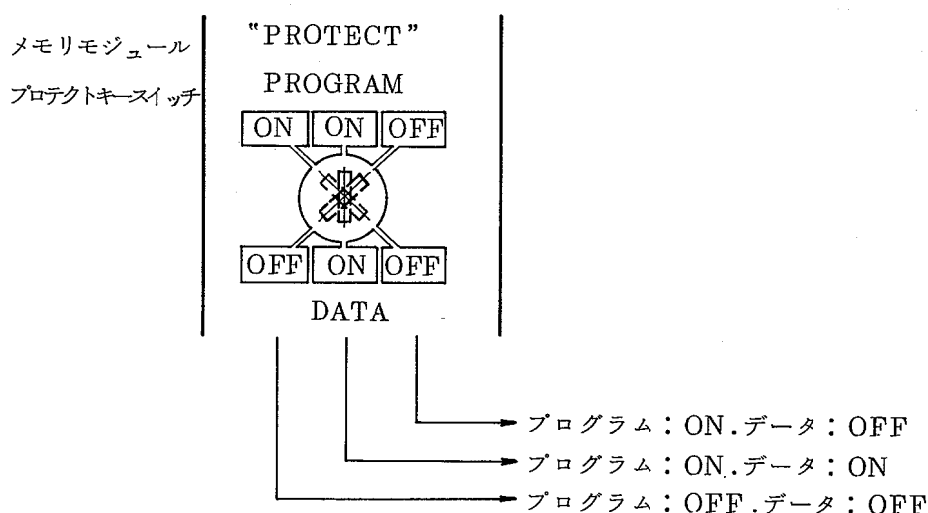
タイプ	形 式	容 量	備 考
RAM	RAM-6624	・ データレジスタ 8K語 ・ ユーザプログラム16Kステップ	
	RAM-6625	・ データレジスタ 16K語 ・ ユーザプログラム32Kステップ	
ROM	ROM-6634	・ データレジスタ 8K語 ・ ユーザプログラム16Kステップ	EEPROM内蔵
	ROM-6635	・ データレジスタ 16K語 ・ ユーザプログラム32Kステップ	

注) ユーザプログラムエリアには，プログラム実行制御用のシステム情報エリアが1Kステップ分含まれています。

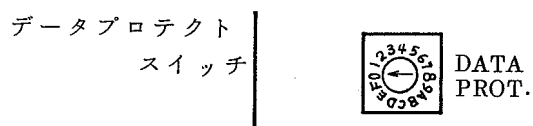
3-1-1 メモリ書込プロテクト機能

メモリモジュール正面のプロテクトモード切換（以下プロテクトキースイッチと呼びます。）キースイッチおよびデータプロテクト領域設定スイッチ（以下データプロテクトスイッチと呼びます。）によりデータレジスタメモリ，ユーザプログラムメモリへの書込のプロテクトをかけることができます。

プロテクトキースイッチは，データレジスタまたはユーザプログラム全エリアに対するプロテクトの指定で，「ON」の位置でプロテクトがかかります。（下図参照）



データレジスタエリアのプロテクト指定が「ON」のとき，データプロテクトスイッチにてプロテクト範囲を細かく指定できます。（下図参照）



データプロテクトスイッチ位置	データレジスタ書込みプロテクト範囲
0	プロテクトなし
1	D02048～最終Dレジスタ
2	D04096～ “
3	D06144～ “
4	D08192～ “ ※1
5	D12288～ “ ※1
6	プロテクトなし
7	
8	
F	

※1 データレジスタ8K語のメモリモジュールではプロテクトなしとなります。

メモリに書込プロテクトをかけておくことにより，プログラマからのプログラム変更，データレジスタ内容変更またユーザプログラムによるデータレジスタ変更が禁止されます。（固定データへの上書，破壊を防ぐ場合などに使用できます。）

3-1-2 イニシャルロード機能

メモリモジュールにROMモジュールを使用し、プロテクト指定してRUNモードにした場合、プロテクト指定した領域のEEPROMの内容がRAMに転送された後に、制御が開始されます。

(イニシャルロード指定)

プロテクトキースイッチにて、ユーザプログラムエリアをプロテクト指定した場合、プログラムエリア全てが(ROMモジュールのサイズ分)イニシャルロードされます。又、データレジスタエリアをプロテクト指定した場合、データプロテクトスイッチにてプロテクト指定された範囲のみイニシャルロードされ、残りは以前のデータを保持します。

注) ROMモジュールを使用して、プログラムを変更し、PROMライトコマンド(CNTL , 9 4)を実行せずに、プロテクト指定してRUNモードへ立ち上げると、イニシャルロード機能により変更が無効になってしまう場合があります。プログラム変更後はプロテクト指定をせずに立ち上げるか、必ずPROMライトコマンド実行後に立ち上げる様にして下さい。

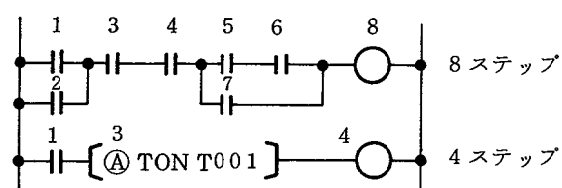
3-2 プログラム容量の計算

EX 2 0 0 0 の適用に先立って所要メモリの計算が必要です。

- ・ リレーシーケンス回路の接点，コイルは全て1ステップとして計算します。
- ・ タイマー，カウンタ，ファンクション命令については2～7ステップ必要とします。

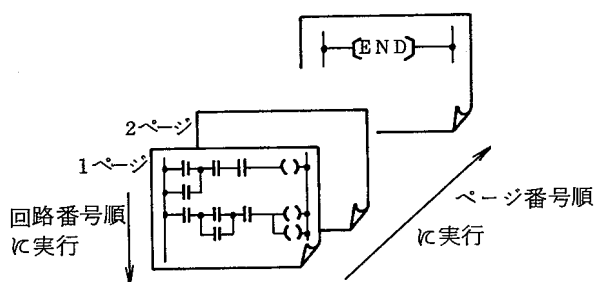
詳細は3・1命令語一覧のステップ数の項を御参照下さい。

- ・ 分流，合流等の接続数は計算する必要ありません。



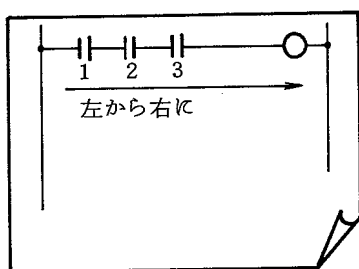
3-3 プログラムの実行順序

プログラムメモリに格納されているラダー回路は、ページ番号順、回路番号順に実行してEND命令まで実行します。(但し、ジャンプ、サブルーチンなどの実行制御を伴う命令を含む場合を除きます。)同一回路番号の実行順序は次のルールに従って実行します。

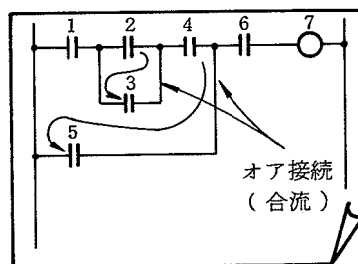


- ・ルール①; 上下接続のないラインの中では、左から右の順に実行します。
- ・ルール②; 上下接続のある1つのつながり回路ではオア接続(信号の合流)がある毎に、オア論理部を先に実行します。
- ・ルール③; また右方向に流れ出し(分流)があるときは、先ず上のラインを実行してから下のラインを実行します。

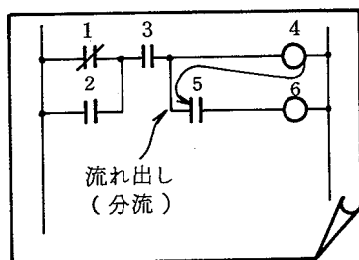
ルール ①



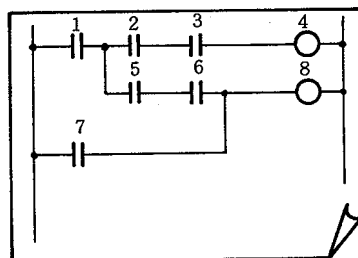
ルール ②



ルール ③

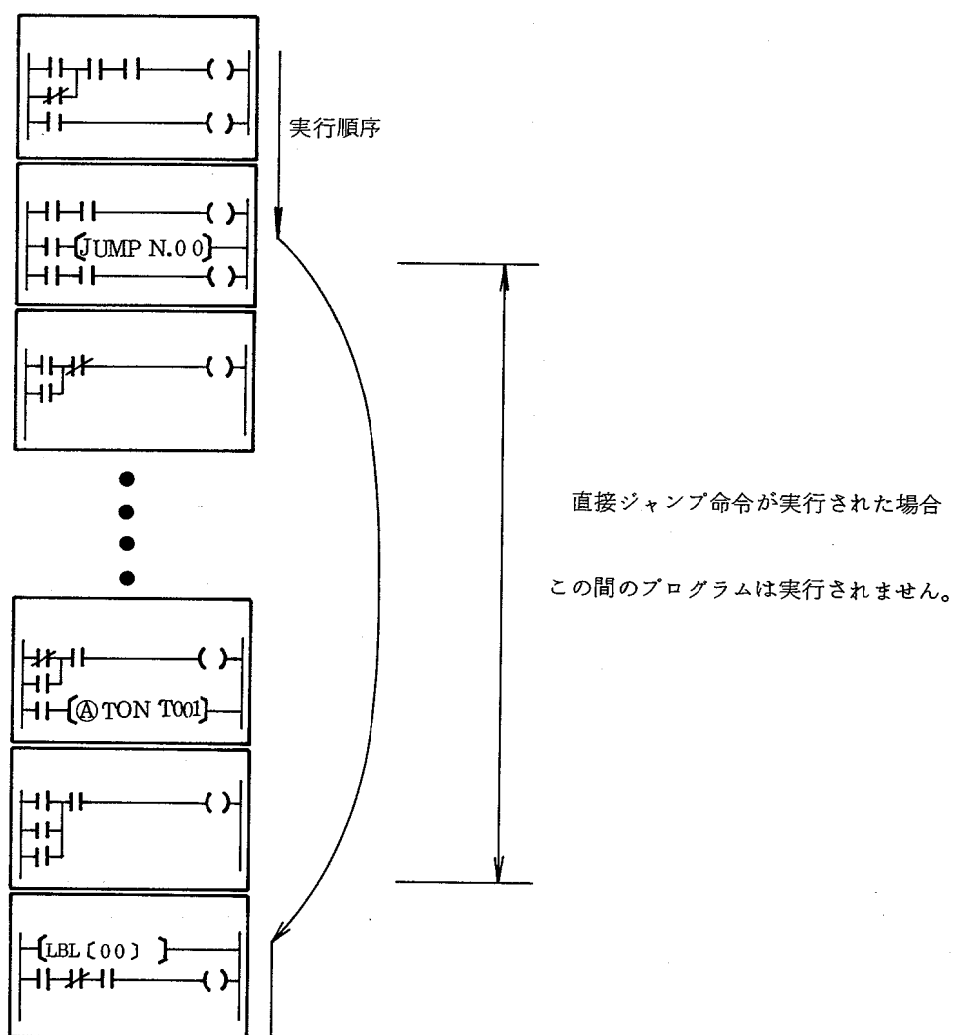


ルール ②③の組合せ



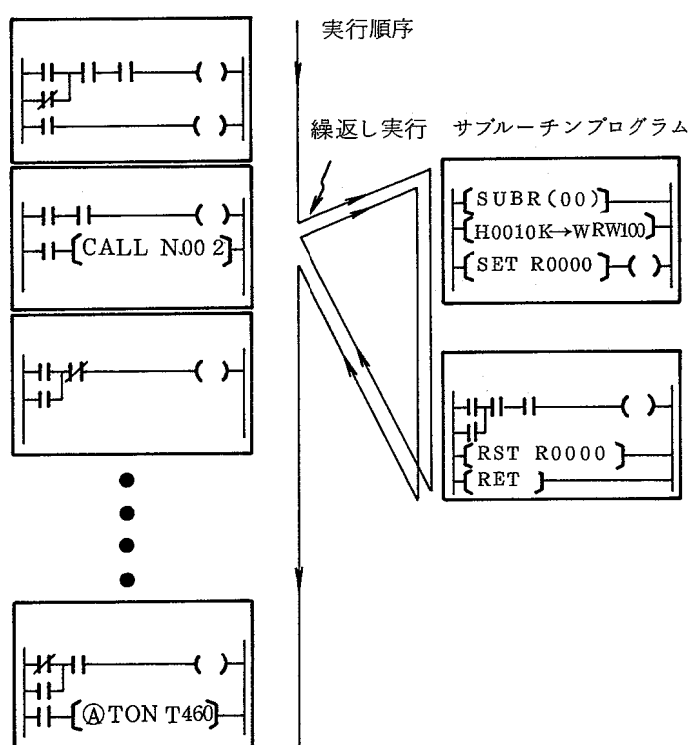
3-3-1 直接ジャンプ命令を含む場合の実行順序

プログラム中に直接ジャンプ命令がある場合、直接ジャンプ命令が実行されると、ラベル命令までのプログラムは実行されません。



3-3-2 サブルーチンコール命令を含む場合の実行順序

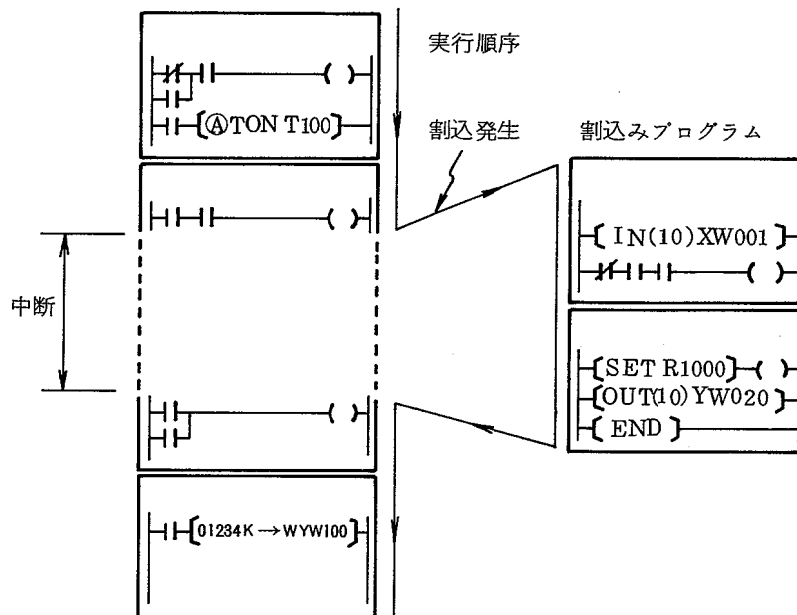
プログラム中にサブルーチンコール命令がある場合、サブルーチンコール命令が実行されると、直ちにサブルーチンプログラムを実行し、サブルーチンプログラム終了後、続きのプログラムが実行されます。



サブルーチンコール命令実行により
サブルーチンプログラムを指定回数
(左図では2回)実行後、続きのプロ
グラムを実行します。

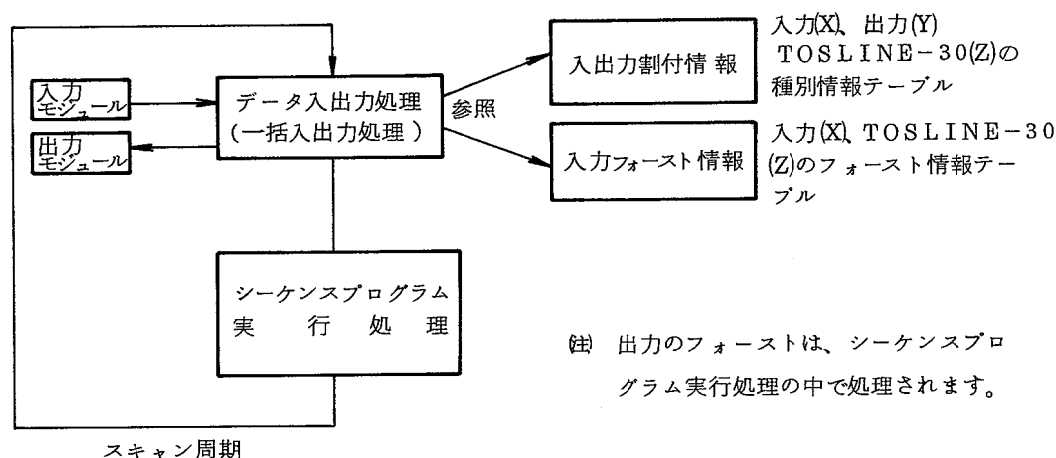
3-3-3 割込プログラム実行時の実行順序

割込プログラムが起動された場合、実行中のプログラムは直ちに中断され、割込プログラムが実行され、割込プログラム終了後、続きのプログラムが実行されます。



3-4 データ入出力処理

外部入出力機器のセンサ入力やアクチュエータとの信号のやりとりは、一旦入出力モジュールで中継され、コントローラ内のタイミングにより入出力割付情報を基に、入力はフォースト指定なしのデバイスに対して、入力デバイス／レジスタに格納し、この入力データにてプログラム実行を行い出力はシーケンスプログラム実行結果が出力デバイス／レジスタに格納されていますので、このデータを出力機器に出力します。これを一括入出力処理と言います。



又、ユーザ命令の「IN」、「OUT」を使用することにより、命令実行時に直接、入出力モジュールと入出力デバイス／レジスタ間のデータ転送が行えます。これらの命令を直接入出力命令と呼びます。直接入出力命令を使用すると一括入出力に加え、直接入出力が行われますが、入出力を行うモジュールの入出力割付にグラフィックプログラマより「i」設定を付けることにより、その入出力モジュールに対する一括入出力は禁止することができます。

☆割込機能付入出力モジュールには「i」設定を付けないとI/O照合エラーとなるので、入出力デバイス／レジスタとのデータ転送に必ず直接入出力命令を使用して下さい。

☆フォーストは各エリアの先頭128Wのみ設定可能です。

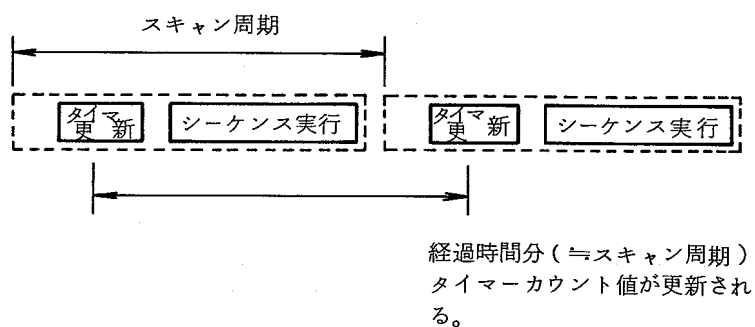
X/Y0000~X/Y127F
R0000~R127F
Z0000~Z127F

☆HALT → RUN遷移

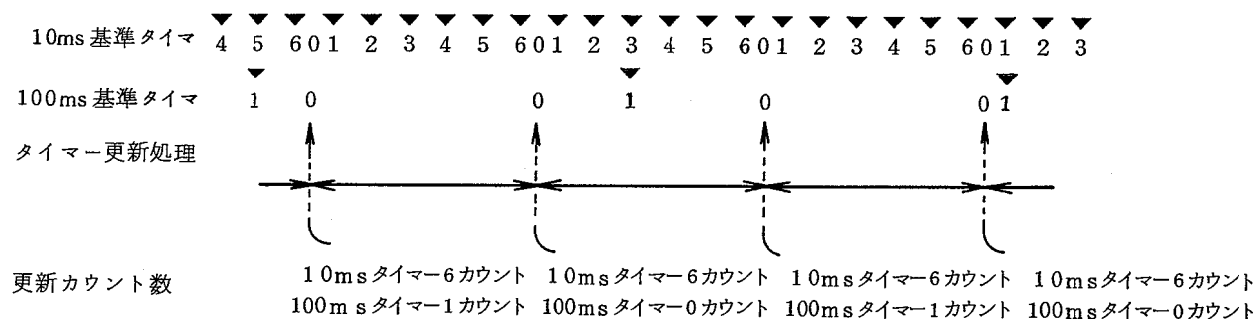
第1回目スキャンでは、入力のみ行い、出力は不実行になります。また、直接出力命令も不実行になります。

3-5 タイマー更新処理

タイマー命令の実行により、タイマーレジスタにセットされたタイマカウント値は、ユーザプログラム実行直前に経過時間分更新されます。

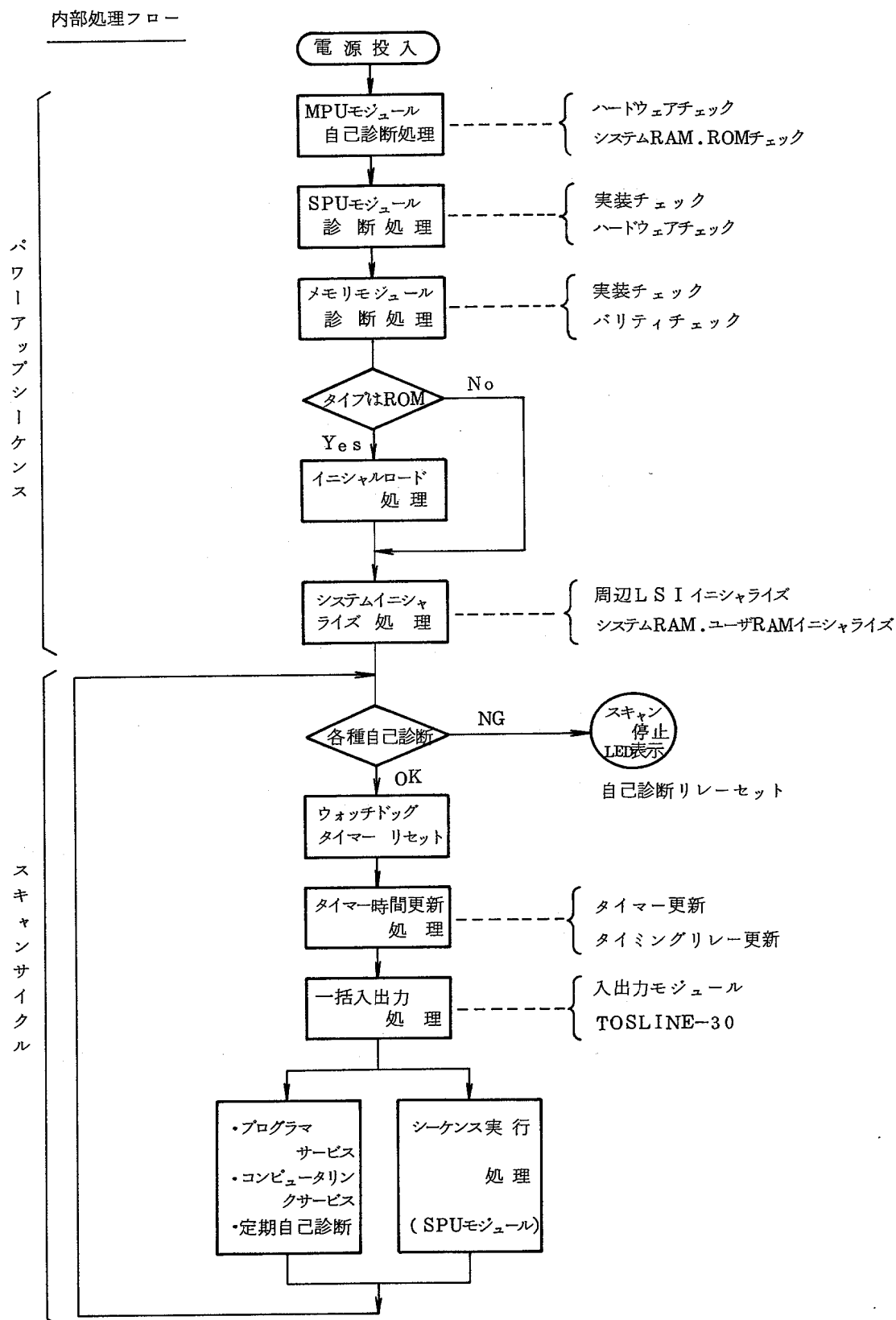


タイマーカウント値の更新値は基準タイマを用いて算出しているので、誤差が積算されることはありません。(下図参照)



タイマー更新カウント値算出方法

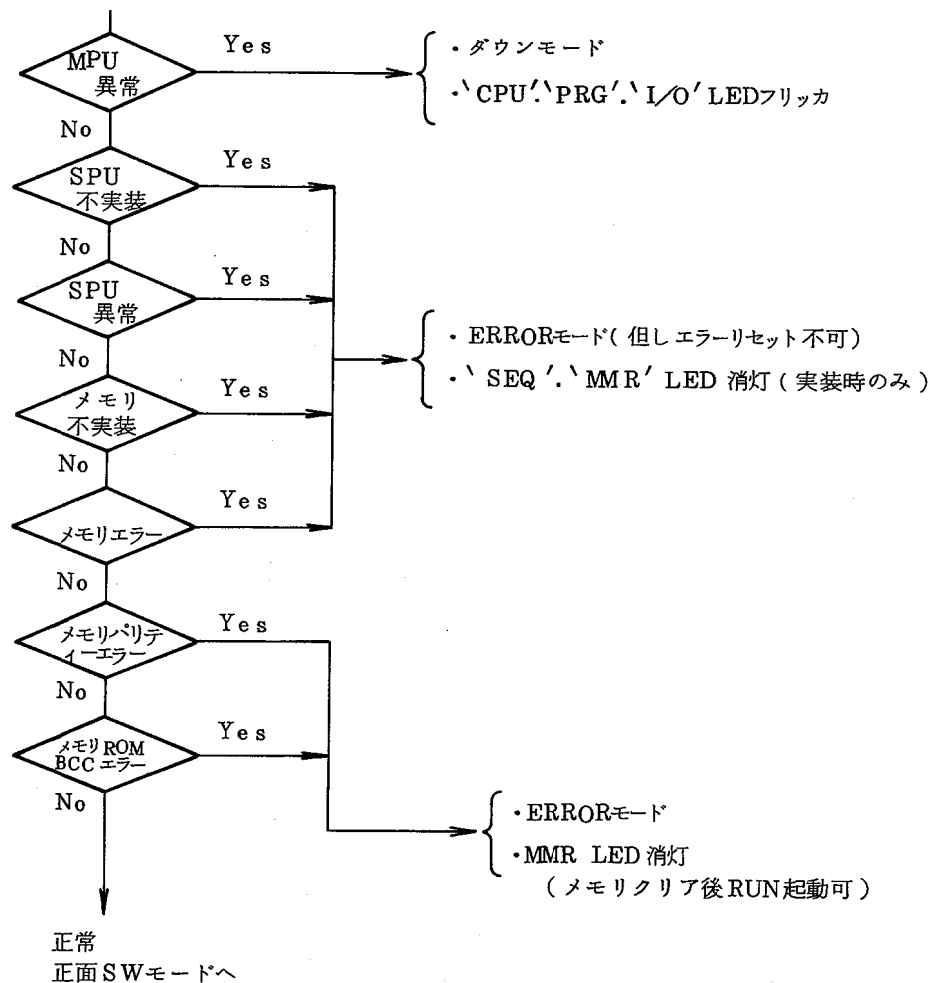
3-6 内部処理サイクル



3-6-1 パワーアップシーケンス

電源投入と同時にMPUモジュール、SPUモジュール、メモリモジュールのチェックを行い正常であれば、システムイニシャライズ処理を行います。

エラー発生時は、エラー発生箇所により処理は下図フローの様になります。



注) SPU、メモリモジュールハードウェアエラー発生時はERRORモードとなりプログラマサービスも行いますが、エラーリセットコマンドは効きません。(異常を取除いた後電源再投入して下さい。)

3-6-2 システムイニシャライズ処理

レジスタ／デバイスデータのイニシャライズおよびフォースト指定のイニシャライズを行います。

レジスタ	電源投入後の状態
外部入出力 (XW,YW)	入力フォースト指定されている入力デバイス(X)はそのデータを保持し、それ以外は全て0。
内部出力 (RW)	停電記憶指定レジスタN ₀ 以後はすべて保持。それ以外は全て0。
タイマ (T,T.)	停電記憶指定レジスタN ₀ 以後は、すべて保持。それ以外は全て0。
カウンタ (C,C.)	
データレジスタ (D)	
リンクレジスタ (ZW)	入力フォースト指定されている入力デバイス(Z)はそのデータを保持し、それ以外は全て0。

3-6-3 スキャンサイクル

パワーアップシーケンス実行後、スキャンサイクルへ移ります。

スキャンサイクルでは、ハードウェアチェックを行った後ユーザシーケンスプログラムを実行し、実行完了にて1スキャンを終了します。その後再びスキャンサイクルの先頭へ戻り以後繰返し実行します。

3-6-4 各種自己診断処理

ユーザプログラム実行前に行う自己診断処理で、RUN(SIM)モード遷移時初回にだけ行われる処理と毎スキャン行われる処理があります。

(1) I/O照合チェック

RUN(SIM)モード遷移初回スキャンに実行され、内部I/O設定テーブルと実装I/Oの照合チェックを行います。

(2) ユーザプログラムチェック

RUN (SIM)モード遷移初回スキャンに実行され、ユーザプログラムの文法チェックを行います。'END命令'の存在、JCS/JCR, MCS/MCR命令のチェック, JUMP, CALL命令のチェック, 出力オペランドのチェック, イリーガル命令の検出等を行います。

(3) 入出力データバスチェック

各入出力ユニットのデータバスが正常かどうかチェックします。

(4) SPU演算機能チェック

SPUモジュールの演算機能が正常かどうかチェックします。

(5) イリーガル命令チェック

ユーザプログラム実行中に、イリーガル命令が存在しないかをチェックします。

(6) パリティチェック

ユーザプログラム実行中に、レジスタ/デバイスデータ、ユーザプログラムでパリティエラーがないかチェックします。

3-6-5 ウォッチドッグタイマリセット

MPUモジュールの正常動作監視およびスキャン周期の監視を行い、スキャン毎にこれをリセットします。

MPUモジュールの動作監視には350msのタイマーを使用し、350ms以内にリセットできない場合、CPUエラーとなります。

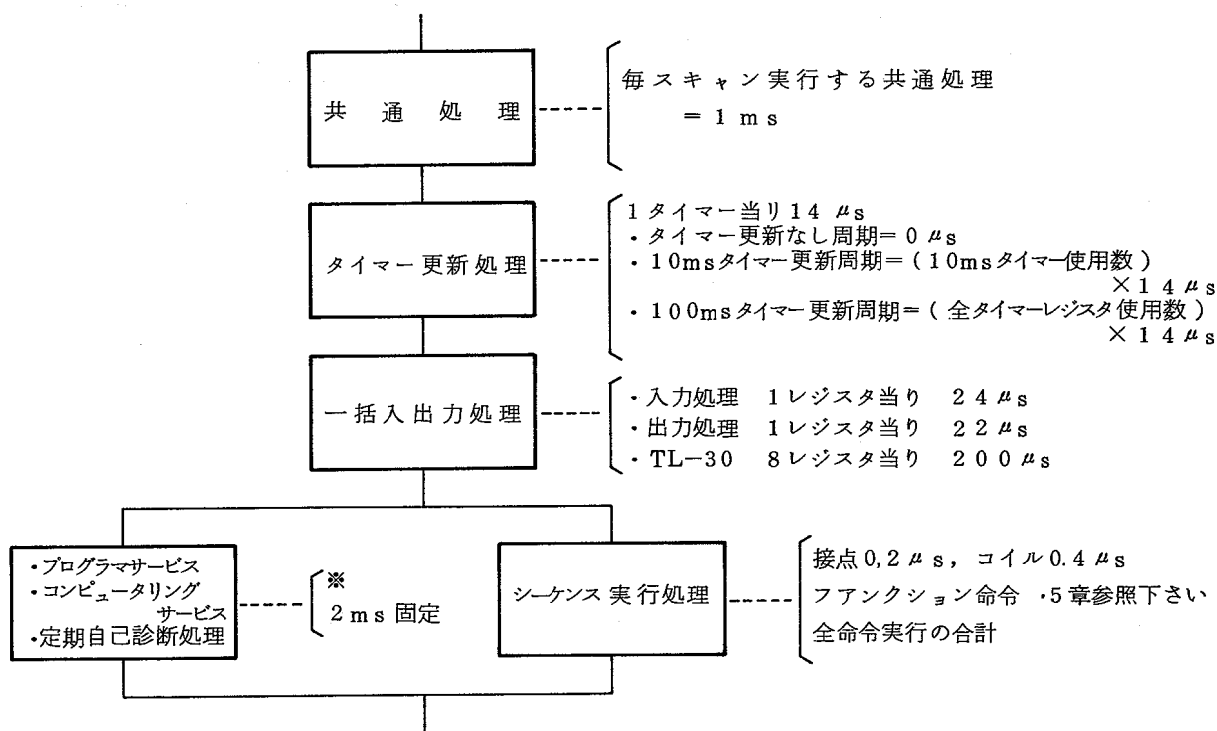
スキャン周期の動作監視には250msのタイマーを使用し、250ms以内にリセットできないときは、スキャンタイムオーバーとなります。

3-6-6 スキャン動作

本体のスキャン動作は、タイマ時間の更新、データの入出力、ユーザプログラムの実行により1スキャンサイクルを構成しています。

プログラムのサービス処理、コンピュータリンク処理などの周辺サポートは、ユーザプログラム実行中に並列して行われます。また、MPUモジュール内のハードウェアチェック、システムRAMリード/ライトチェック、システムROM、BCCチェック、メモリモジュール内ユーザメモリリード/ライトチェックも定期診断処理としてユーザプログラム実行と並行して行っています。

スキャン時間の概算は次の様に行います。



※ フローティングスキャンモードのみ
定刻スキャンモードでは0です。

・ シーケンス実行処理とプログラマサービス処理等は
並列処理されるので、処理時間は長い方になります。

<計算例>

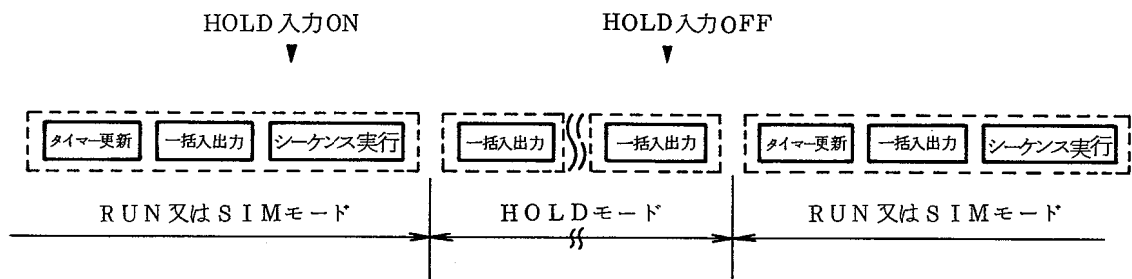
・ 共通処理	1.0 ms
・ 入力 128 点 + 出力 128 点	0.37 ms
・ TL-30 32W	0.8 ms
・ シーケンス 2ks … 接点 1ks, コイル 0.25ks	
ファンクション 0.75ks (1命令3ステップ, 10μs)	2.7 ms
・ タイマー使用数 100msタイマー 32ケ	(0.45 ms)
	5.0 ms (タイマー更新周期)
	5.5 ms)

・ 実際のスキャン時間は、本体の状態によりバラつきます。上記計算は目安として扱って下さい。

3-6-7 HOLD入力時の処理

電源カード(P S - 6 6 1 8)のHOLD入力端子を'ON'にした場合、本体がRUN又はSIMモードで運転中であれば、HOLDモードに入り、ユーザプログラムの実行が一時停止します。このときタイマー更新も停止しますが、一括入出力は実行されます。

HOLD入力端子を'OFF'とすれば、ユーザプログラムおよびタイマー更新は継続して処理を再開します。



- HOLD入力が'ON'になった次のスキャンからHOLDモードとなり、HOLD入力が'OFF'となった次のスキャンからRUN又はSIMモードに戻ります。
- HOLDモード中は基準タイマーの更新もしないので、タイマー更新も継続されます。
- HOLDモード中の割込要求は保留され、HOLDモード解除後に実行されます。
- 定刻スキャンモードでHOLDモードに入った場合、定刻スキャン周期は維持します。

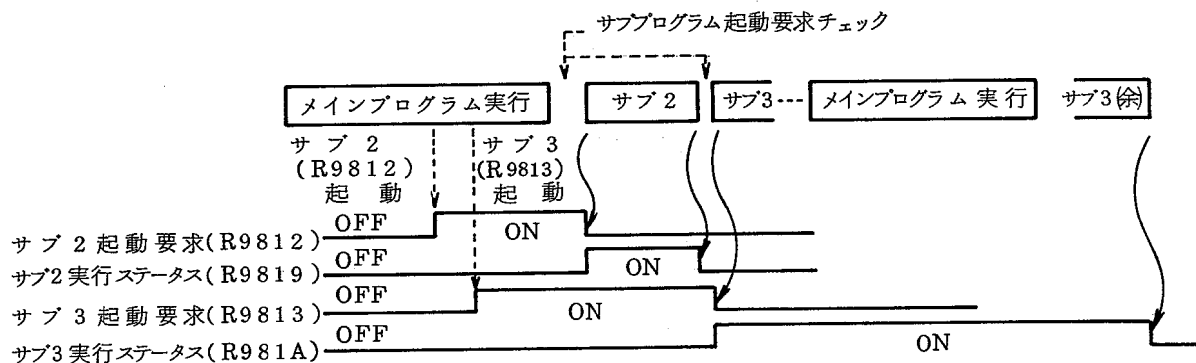
3-6-8 サブプログラム起動方法

サブプログラムは8本まで登録できます。(登録方法は1-3プログラムページ割付を参照下さい。)

8本のプログラムはサブ1～サブ8と呼びます。

サブ1は、初期設定用サブプログラムで、登録があればHALT→RUN(又はSIM)へモード移行時の第1スキャンでのみ実行されます。

サブ2～サブ8は、特殊コイルエリアのR 9 8 1 2～8に'ON'をセットすることにより起動要求がセットされ、メインプログラム実行完了後に起動要求のあるものが実行されます。又、サブプログラム実行中はR 9 8 1 9～Fのサブプログラム実行ステータスがセットされます。起動要求と実行ステータスのタイミングを次ページの図に示します。(サブプログラム動作モード等は1-2-2サブプログラムを、特殊コイル構成は2-6-12(7)項を参照下さい。)



。サブプログラム2～8は特殊コイルの起動要求をセットすることにより1回だけ起動されます。また、起動要求は受付け時に自動リセットします。

。サブプログラム実行中は特殊コイルの実行ステータスがセットされます。サブプログラムが中断された場合でも実行ステータスはセットされたままとなります。

。サブプログラムの起動要求チェックは、メインプログラム実行後およびサブプログラム実行完了後に行うので、サブプログラム内でより優先度の低いサブプログラムに起動要求をかけた場合、同一スキャン内で実行されます。(但し、時間制限で実行されない場合があります。)

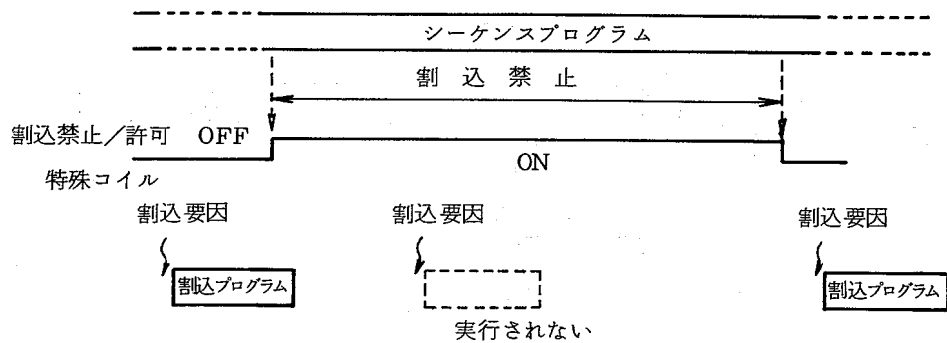
3-6-9 割込プログラムの実行禁止

割込プログラムは、定周期および割込機能付入出力カード用8本の計9本が登録できます。

(登録方法は1-3プログラムページ割付を参照下さい。)

割込要因発生時に割込プログラムを起動するかしないかは特殊コイルエリアR9800～8の割込禁止/許可により制御することができます。(1-2-3割込プログラム, 特殊コイルについては2-6-1 2(6)を参照下さい。)

割込禁止/許可特殊コイルに'ON'をセットすることにより、割込要因発生時でも割込プログラムは起動されません。



◦ 割込禁止状態で発生した割込要因は、保留されません。

◦ HALT→RUN（又はSIM）へモード移行時の第1スキャンは割込要因の受付けを禁止していますので、割込禁止の設定はイニシャルプログラム（サブ1）で行うか、メインプログラムの第1スキャンで行う様にして下さい。

3-7 RAS 情報

EX 2000 では、メンテナンス用の RAS 情報として、エラー履歴テーブル、プログラム実行時間テーブル及びシステムトレース機能を持ち、プログラマより読出し・操作が可能です。

(1) エラー履歴テーブル

自己診断処理にて検出されたエラーを時系列的に登録するテーブルです。登録内容はエラー種別及び関連情報、発生時刻、エラーの程度などです。

本テーブルはプログラマのエラー履歴テーブル画面 (**SYS** , **3**) にて読出すことができます。

※ プログラマからの RUN 起動要求時のチェックにて発生したエラーは、プログラマに表示され、エラー履歴テーブルには登録されません。

エラー以外に電源 ON/OFF のイベント登録も行われます。

(2) プログラム実行時間テーブル

RUN (又は SIM) モードにてプログラム実行時のメイン、サブ、割込プログラムの実行時間及びスキャン時間を計測・登録します。

スキャン時間、メイン、サブプログラムの実行時間は 1ms 単位で、割込プログラムの実行時間は 0.1ms 単位に計測されます。

本テーブルは、プログラマのプログラム実行時間テーブル (**SYS** , **4**) にて読出すことができます。

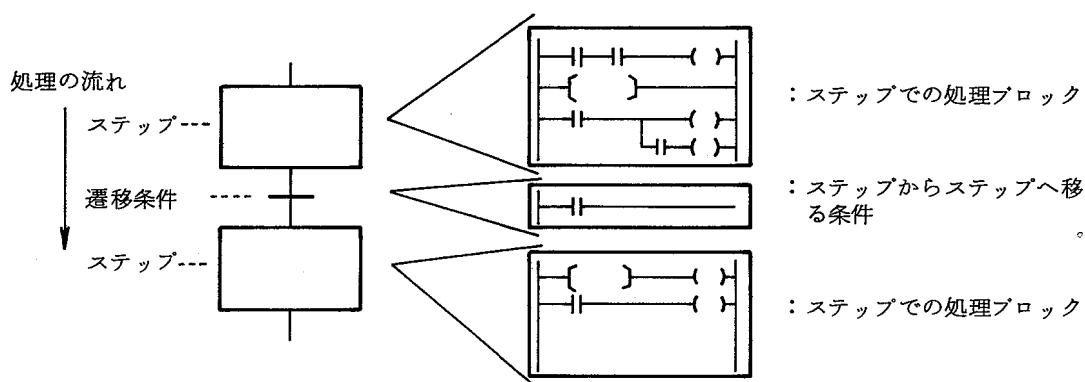
※ 時間の計測は全て切上げで行われます。又、プログラム起動などのオーバーヘッドも含まれていますので命令実行時間の合計とは一致しない場合があります。

メイン・サブプログラムにて実行中に割込プログラムが入った場合、割込プログラムの実行時間も合計されて計測されます。

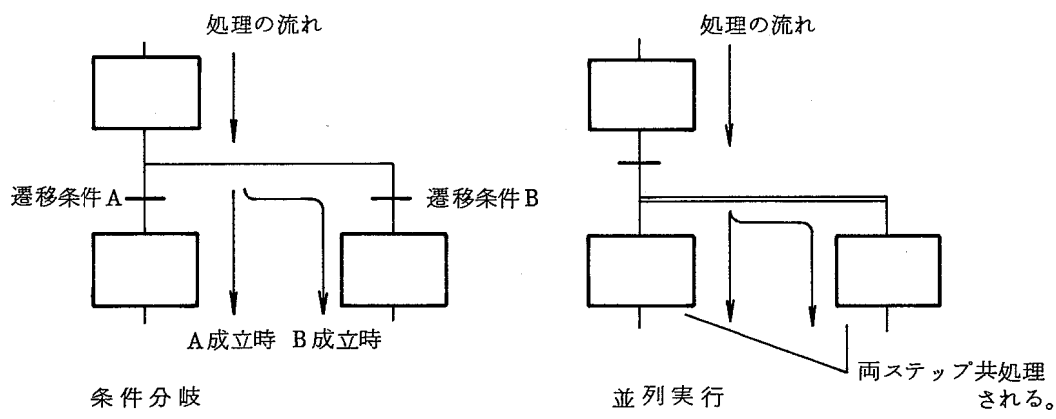
4-1 シーケンシャルファンクションチャート概要

EX2000では、ラダープログラムの構造記述にシーケンシャルファンクションチャート（以下SFCと記す）を使用できます。

SFCは、ラダープログラムの機能単位のブロックを1つのステップとし、ステップからステップへの処理の流れを遷移条件（ラダープログラム）により規定したものです。



SFCの記述には条件分岐、並列実行等の構造化プログラミングが使用できるので、ラダープログラムでは複雑になる工程歩進形プログラムでも分かり易く、簡単に作成することができます。

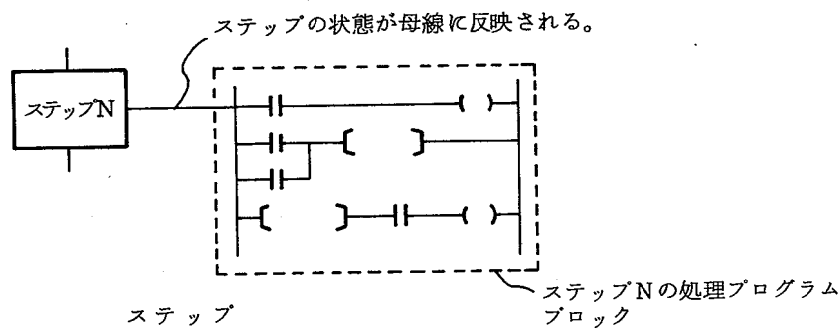


4-2 SFC 構成要素

以下SFCの構成要素について説明します。

4-2-1 ステップ (STEP)

ステップは、ラダープログラムの1つの機能ブロックを示すもので、箱で表わされます。

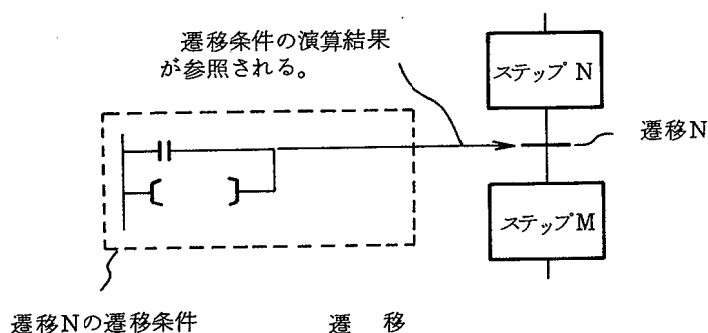


ステップは、アクティブ／イナクティブの2つの状態を持ち、アクティブな状態のステップだけ対応するラダープログラムブロックが実行されます。

図で説明しますと、ステップNの状態が対応するプログラムブロックの母線の状態となり、ステップNがアクティブなときには、対応するプログラムは母線が“活きた”状態で、実行となり、イナクティブなときには、母線が“切れた”状態で、不実行（実際には母線OFFでの実行）となります。

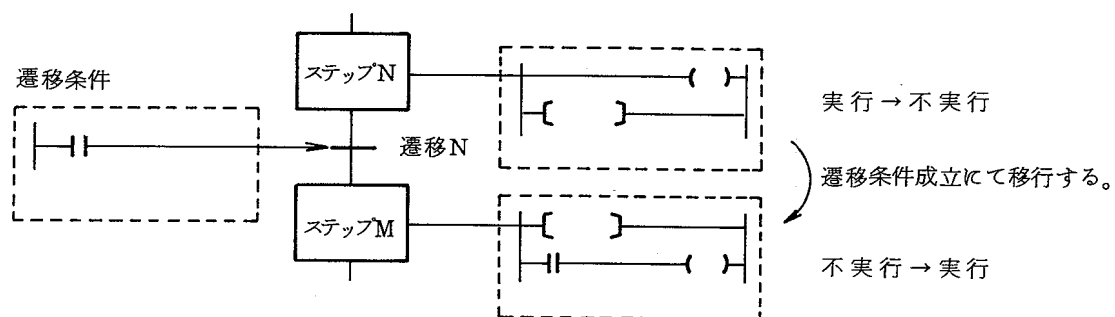
4-2-2 遷移 (TRANSITION)

遷移は、ステップから次のステップへの移行時の処理を行う部分で、横線で表わされます。



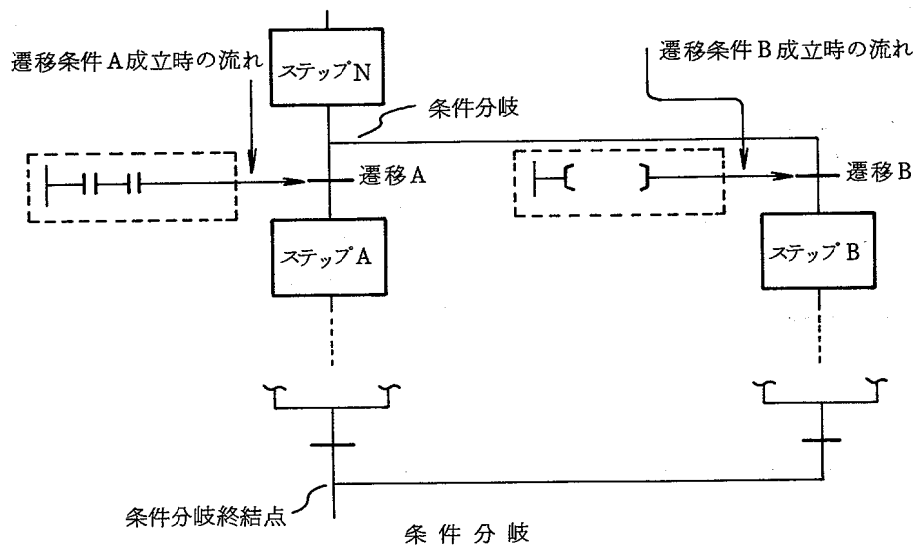
遷移には、遷移条件が付属していて、遷移条件成立時に遷移処理を行います。

図で説明しますと、ステップNがアクティブな状態で遷移Nの遷移条件が成立（遷移条件の演算結果がON）した場合、遷移NではステップNをイナクティブな状態とし、ステップMをアクティブな状態とします。この処理により、ステップNに対応するプログラムブロックの実行が不実行となり、かわりにステップMに対応するプログラムブロックが不実行から実行となります。



4-2-3 条件分岐

条件分岐は、ステップの流れを選択制御するもので、一本の水平線で表わされます。

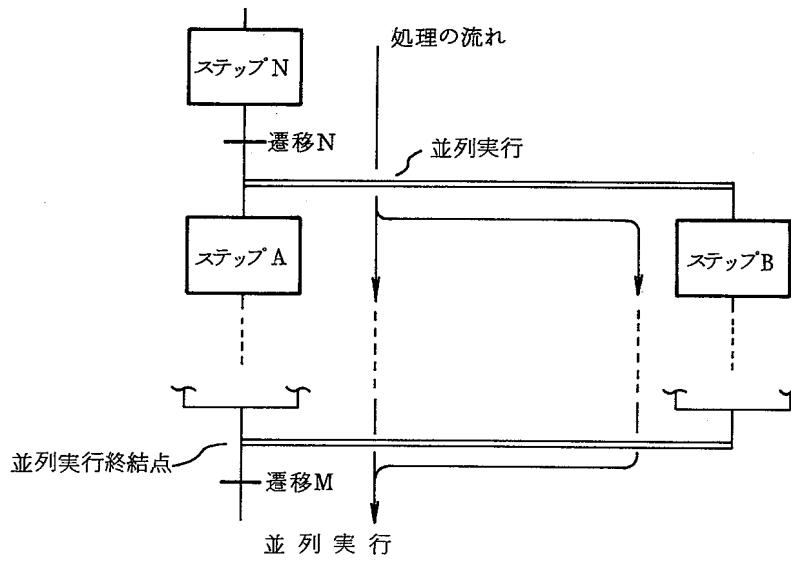


条件分岐では水平線につながっている複数の遷移のうち遷移条件の成立している1つの列へアクティブなステップが流れます。図では、遷移Aの遷移条件が成立した場合にはステップNからステップAへ、遷移Bの遷移条件が成立した場合にはステップNからステップBへ、ステップが移ります。複数遷移条件が同時に成立している場合には、左側の遷移が優先されます。

条件分岐の終結点では、水平線上の遷移のうち1つの遷移条件が成立し、その前のステップがアクティブであれば、条件分岐を抜け出し次のステップへと移ります。

4-2-4 並列実行

並列実行は、複数ステップを同時実行するもので、二重の水平線で表わされます。



並列実行では、ステップが移ったときに、二重水平線につながっている全てのステップが同時にアクティブとなり、複数の処理を同時に行います。図では、遷移Nの遷移条件が成立してステップが並列実行に移った場合、ステップAとステップBが同時にアクティブとなり、以降、並列実行終結点までの処理が並列して実行されます。

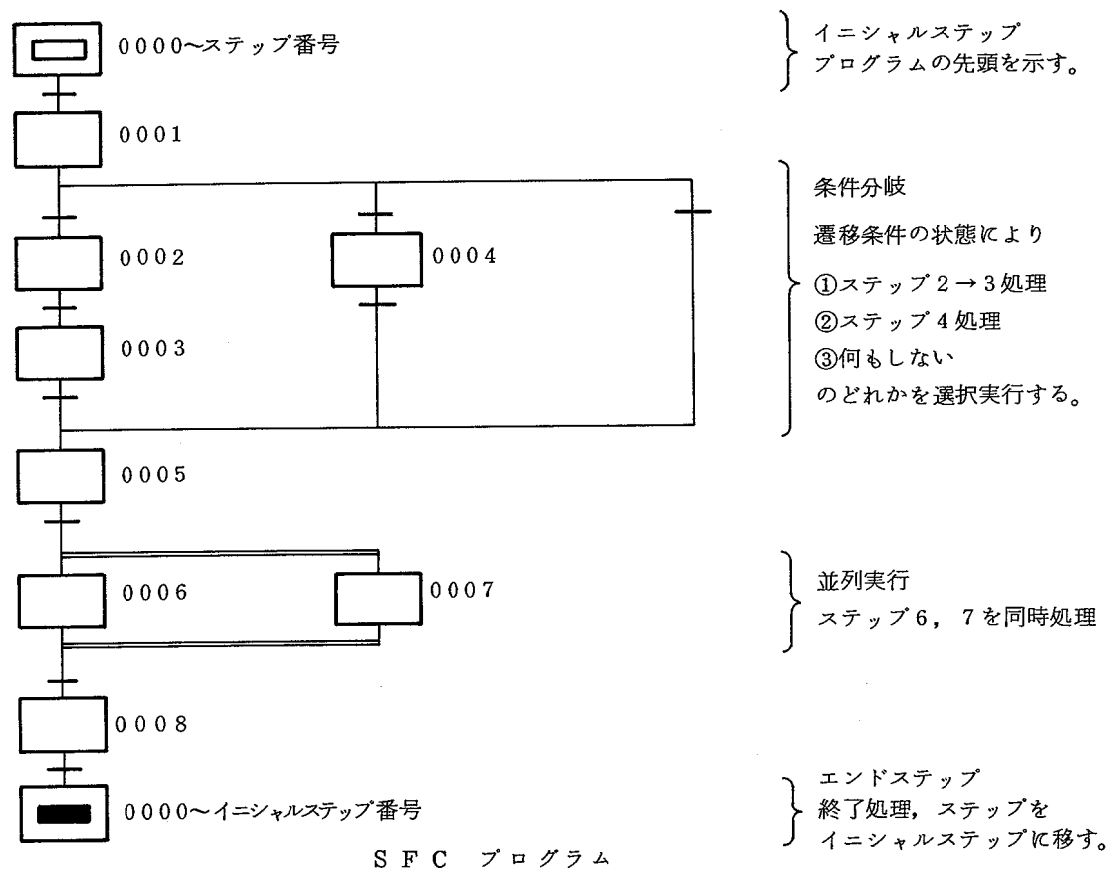
並列実行の終結点では、並列処理している全ての列が終了するまで待ちます。つまり、終結点の二重水平線上の全てのステップがアクティブで、かつ遷移Mの遷移条件が成立したとき、次ステップへ移行します。

4-3 SFC プログラム構成

EX2000では、SFCプログラムはラダープログラムと同様に、プログラムページ内に置くことができます。

下図にプログラムの例を示します。基本的には4.2項で説明した構成要素により構成されますが、ステップに特定の意味も持たせたもの、遷移の特別な使い方もあります。

下図のイニシャルステップ、エンドステップはステップに特定の意味を持たせたもので、イニシャルステップはSFCプログラムの始めを意味するステップで、エンドステップは終りを意味します。

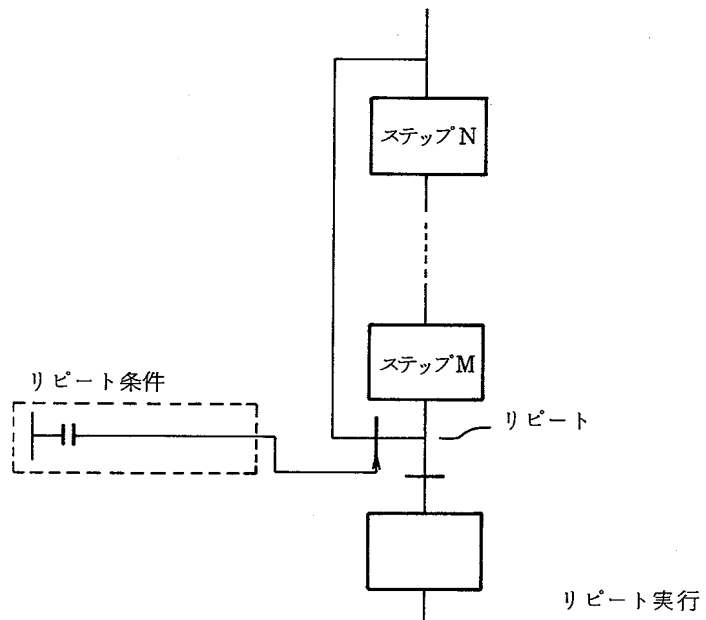


ステップはステップ番号により管理され、0～3999まで番号が設定できます。つまりSFCプログラム全体で4000個までのステップが使用できますが、1ページ内では100個までです。(但し、エンドステップはエンド処理と繰返し処理するためにステップをイニシャルステップに戻す処理をしますので、自分自身のラダー処理ブロックがなく、ステップ番号を使用しません。)

4-3-1 リピート実行

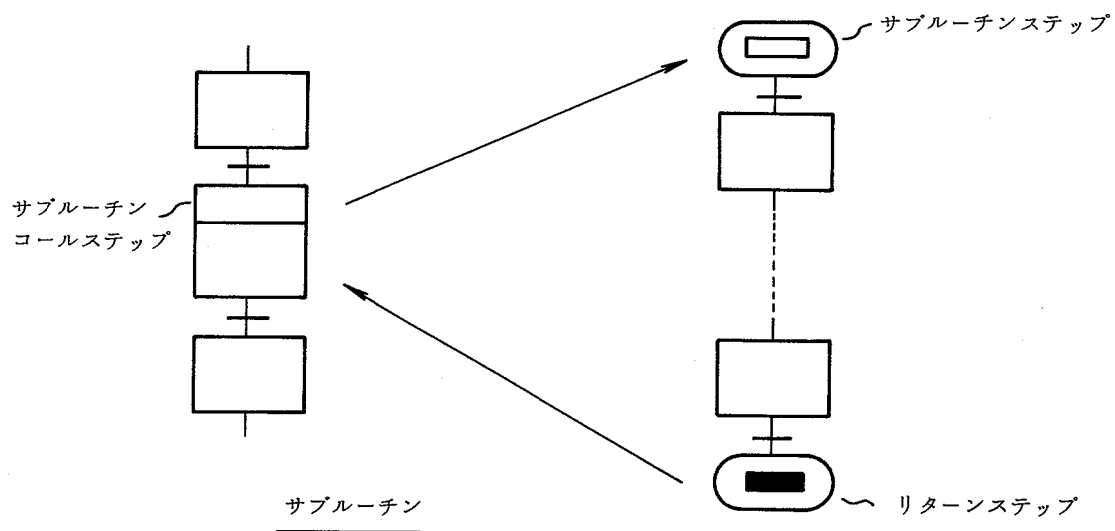
遷移の特別な使い方でリピート（繰り返し）ができます。

図では，リピート条件成立時，ステップMからステップNへステップが移ります。つまり，リピート条件成立中は繰り返しステップN～Mを処理します。



4-3-2 サブルーチン

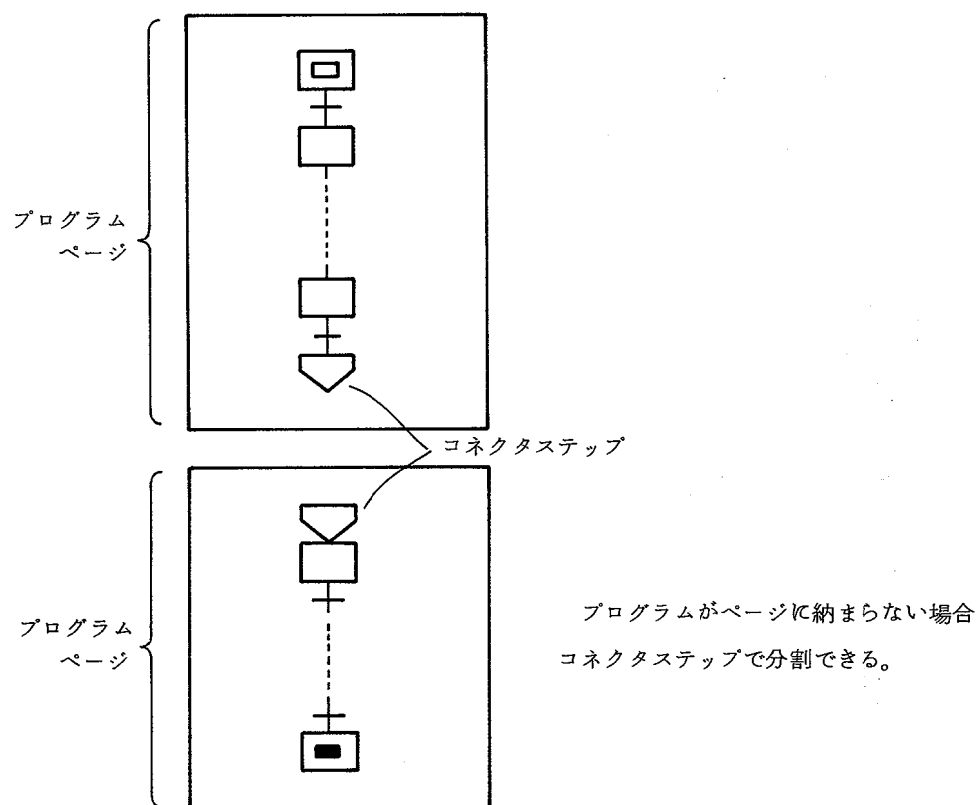
SFCプログラムの一つの機能ブロックをサブルーチンモジュールとすることができます。



サブルーチンは，サブルーチンステップで始まり，リターンステップで終わる1つのSFCプログラムです。サブルーチンモジュールの起動はサブルーチンコールステップを用います。サブルーチンコールステップでは，起動したサブルーチンステップが実行完了（リターンステップまで実行）し，かつ次の遷移条件が成立するまでステップが流れません。

4-3-3 コネクタステップ

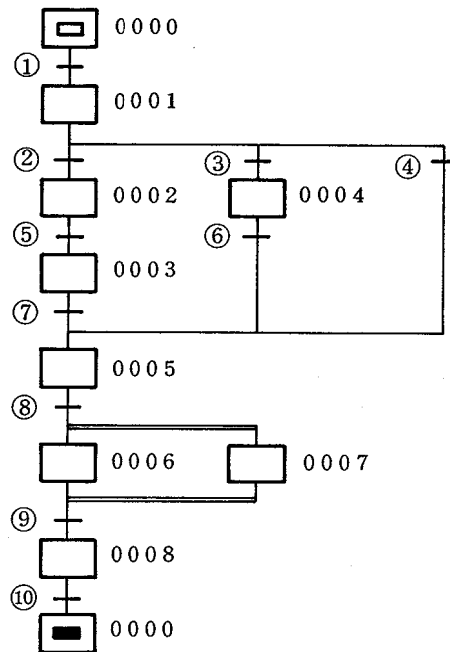
SFCプログラムが1ページ内に納まらない場合に、ページ間での接続をするステップです。



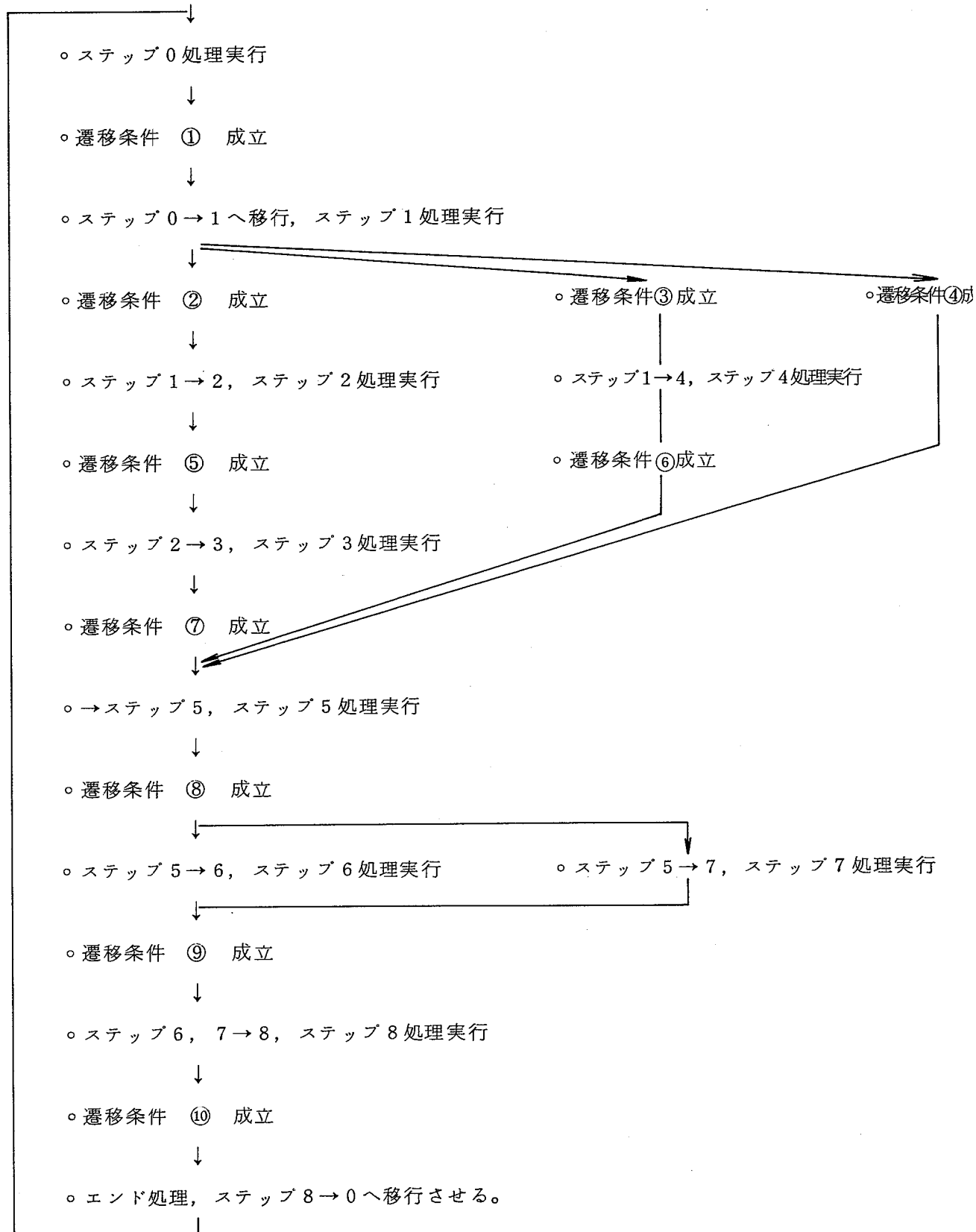
但し、コネクタステップで接続できるのは、ページの最後の遷移と次ページの先頭ステップの間で、1箇所だけです。(分岐中の複数箇所での接続できません。)

4-4 SFCプログラム実行順序

図のSFCプログラムで実行順序を説明します。(図では各ステップの処理, 遷移条件は詳略します。)



◦ SFC プログラムに起動



4-5 SFCプログラム制御

4-5-1 起動, イニシャライズ

SFCプログラムの起動, イニシャライズはラダー命令の「SFIZ…ステップフラグイニシャライズ」により行います。SFIZ命令により各ステップの状態をイナクティブとし, イニシャルステップに起動をかけます。

4-5-2 一時停止

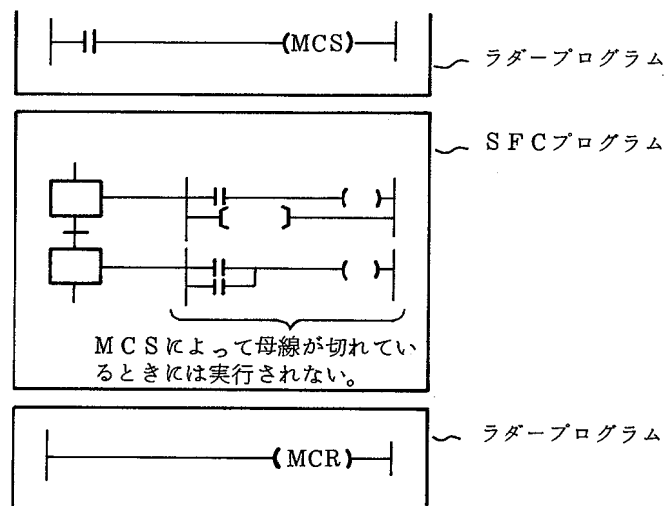
特殊コイルR9820でステップの歩進禁止/許可のコントロールができます。

R9820=ONにて, ステップの歩進が禁止となり, 遷移条件が成立してもステップは移行しません。

注) R9820はSFCプログラム全てに共通となっておりますので, 複数のSFCプログラムを入れた場合, 全てのステップの歩進が停止します。

4-5-3 ステップ処理の出力カット

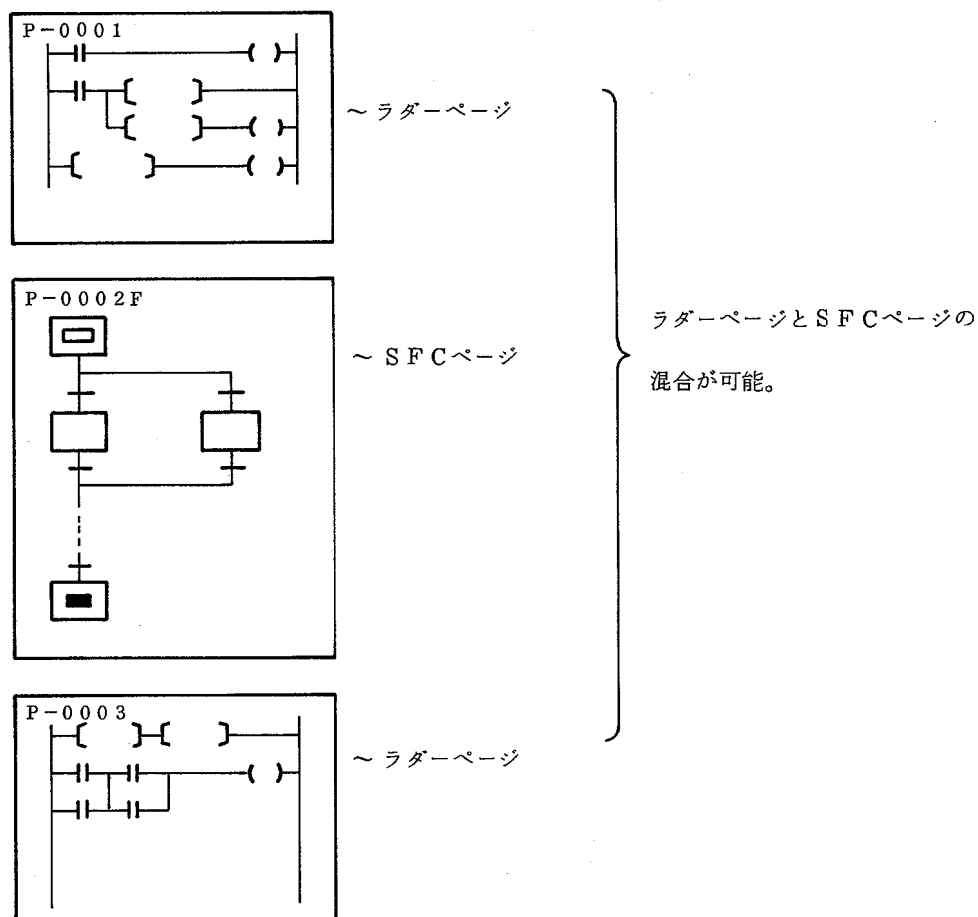
SFCプログラムの前後をMCS/MCRで挟むことにより, ステップ処理の出力をコントロールできます。



MCSにより母線が切れた状態では, ステップは移行していきませんが, ステップの処理は実行されません。

4-6 プログラミング

SFCプログラムはページ単位になっていますから、ラダーのページプログラムと混合してのプログラミングが可能です。



注) ◦ SFC ページがコネクタステップにより複数ページに跨っている場合、途中でラダーページを入れることはできません。

◦ サブルーチンプログラムも SFC ページとしてプログラムします。

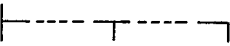
◦ SFC ページには遷移条件、ステップ処理プログラムも含めてプログラムのステップ数 800 までプログラミングできます。

◦ SFC プログラムページ数に制限はありません。(但し、ステップ 4000 個まで、サブルーチン 100 個までです。)

◦ SFC プログラムはメインプログラムエリア (P-1~P-1999) にしか作成できません。

プログラミングの詳細はグラフィックプログラマ (GP110AP2) 取扱説明書を参照下さい。

4-7 SFCシンボル一覧

シンボル名称	シンボル	備考
イニシャルステップ	 n n n n n n n n : ステップ№ (0 ~ 3 9 9 9)	
ステップ	 n n n n n n n n : ステップ№ (0 ~ 3 9 9 9)	
サブルーチン コールステップ	 n n n n n n n n : ステップ№ (0 ~ 3 9 9 9) l l l l l l l l : サブルーチンステップ№ mm (0 ~ 3 9 9 9) mm : サブルーチン№ (0 ~ 9 9)	
エンドステップ	 n n n n n n n n : イニシャルステップ ステップ№ (0 ~ 3 9 9 9)	
コネクタステップ	 mm mm : コネクタ№ (0 ~ 9 9)	
サブルーチンスタートステップ	 n n n n n n n n : ステップ№ (0 ~ 3 9 9 9)	
リターンステップ	 mm mm : サブルーチン№ (0 ~ 9 9)	
遷移	+	
条件分岐		
並列実行		

5-1 命令語一覧

5-1-1 命令語一覧表の見方

5-1-2の命令語一覧表の説明は次のようになっています。

第5章 命令語

TOSHIBA

5-1-2 命令語一覧表

	命 令	FUN 番号	機 能	ページ
リレー	A接点		リレーのA接点(常時閉)です。	91
	B接点		リレーのB接点(常時閉)です。	92
	コイル		リレーのコイルです。	93
	フォースト・コイル		コイルの値をON又はOFFに固定します。	94
	ON時微分接点		入力がOFFからONに変化した時に出力がONになります。	95
	OFF時微分接点		入力がONからOFFに変化した時に出力がONになります。	96
	MCS		MCSの入力がOFFの時、MCS～MCRの間の母線入力が	97
	MCR		OFFになります。	97
	JCS		JCSの入力がONの時、JCS～JCRの間の命令は、高速	98
	JCR		読みとばしを行います。	98
ス	オン・ディレイ・タイム		ON出力を指定時間だけディレイさせます。	99
	オフ・ディレイ・タイム		OFF出力を指定時間だけディレイさせます。	100
命	シングル・ショット		指定時間巾のパルスを発生するシングル・ショット・タイムです。	101
	カウンタ		入力がONする回数をカウントします。	102
令	エンド		プログラムの終了を示します。	103
デ ー タ 転 送	レジスタ転送	000	レジスタ間のデータの転送を行います。	104
	数値転送	001	数値をレジスタに転送します。	105
	テーブル初期化	002	テーブル上のデータの初期化を行います。	106
	マルチプレクサ	003	テーブル上のデータを指定レジスタに転送します。	107
	デマルチプレクサ	004	ワードデータをテーブル上に転送します。	109
	テーブル転送	005	テーブルデータのブロック転送を行います。	111
四 則 演 算	レジスタ加算	010	レジスタ同士の加算を行います。	112
	レジスタ減算	011	レジスタ同士の減算を行います。	113
	レジスタ乗算	012	レジスタ同士の乗算を行います。	114
	レジスタ除算	013	レジスタ同士の除算を行います。	115

①

②

③

④

⑤

〔 説 明 〕

- ①……………各命令語を用途別に分類しています。
- ②……………各命令語の名称を示しています。
- ③……………各命令語に対応したファンクション番号です。(グラフィックプログラマにてファンクション番号を入力しますと番号に対応した命令が画面に表示されます。)
- ④……………各命令語の機能概要について説明しています。
- ⑤……………各命令語を詳細に説明しているページを示します。

5-1-2 命令語一覧表

命 令		機 能	ページ	
	FUN No			
シ ン ケ ン ス 命 令	A 接点		リレーのA接点（常時開）です。	91
	B 接点		リレーのB接点（常時閉）です。	92
	コイル		リレーのコイルです。	93
	フォースト・コイル		コイルの値をON又はOFFに固定します。	94
	ON時微分接点		入力がOFFからONに変化した時に出力がONになります。	95
	OFF時微分接点		入力がONからOFFに変化した時に出力がONになります。	96
	MCS		MCSの入力がOFFの時、MCS～MCRの間の母線入力がOFFになります。	97
	MCR			97
	JCS		JCSの入力がONの時、JCS～JCRの間の命令は、高速読みとばしを行います。	98
	JCR			98
	オン・ディレイ・タイマ		ON出力を指定時間だけディレーさせます。	99
	オフ・ディレイ・タイマ		OFF出力を指定時間だけディレーさせます。	100
	シングル・ショット		指定時間巾のパルスを発生するシングル・ショット・タイマです。	101
カウンタ		入力がONする回数をカウントします。	102	
エンド		プログラムの終了を示します。	103	
デ ー タ 転 送	レジスタ転送	000	レジスタ間のデータの転送を行います。	104
	数値転送	001	数値をレジスタに転送します。	105
	テーブル初期化	002	テーブル上のデータの初期化を行います。	106
	マルチプレクサ	003	テーブル上のデータを指定レジスタに転送します。	107
	デマルチプレクサ	004	ワードデータをテーブル上に転送します。	109
	テーブル転送	005	テーブルデータのブロック転送を行います。	111
四 則 演 算	レジスタ加算	010	レジスタ同士の加算を行います。	112
	レジスタ減算	011	レジスタ同士の減算を行います。	113
	レジスタ乗算	012	レジスタ同士の乗算を行います。	114
	レジスタ除算	013	レジスタ同士の除算を行います。	115
比 較	レジスタ比較 >	014	レジスタ同士の比較を行い、>が成立する時出力がONになります。	116
	レジスタ比較 =	015	レジスタ同士の比較を行い、=が成立する時出力がONになります。	117
	レジスタ比較 <	016	レジスタ同士の比較を行い、<が成立する時出力がONになります。	118
四 則 演 算	倍長加算	017	レジスタ同士の倍長加算を行います。	119
	倍長減算	018	レジスタ同士の倍長減算を行います。	120

	命 令	FUN 番号	機 能	ページ
四 則 演 算	数値加算	020	レジスタと数値の加算を行います。	121
	数値減算	021	レジスタと数値の減算を行います。	122
	数値乗算	022	レジスタと数値の乗算を行います。	123
	数値除算	023	レジスタと数値の除算を行います。	124
比 較	数値比較 >	024	レジスタと数値の比較を行い、>が成立する時出力がONになります。	125
	数値比較 =	025	レジスタと数値の比較を行い、=が成立する時出力がONになります。	126
	数値比較 <	026	レジスタと数値の比較を行い、<が成立する時出力がONになります。	127
論 理 演 算	レジスタ論理積	030	レジスタ同士のビット毎の論理積をとります。	128
	レジスタ論理和	031	レジスタ同士のビット毎の論理和をとります。	129
	レジスタ排他的論理和	032	レジスタ同士のビット毎の排他的論理和をとります。	130
	レジスタ反転	034	レジスタのビット毎のデータの反転を行います。	131
	右ローテート	035	指定ビット数だけ右方向に回転させます。	132
	左ローテート	036	指定ビット数だけ左方向に回転させます。	133
	数値論理積	040	レジスタと数値のビット毎の論理積をとります。	134
	数値論理和	041	レジスタと数値のビット毎の論理和をとります。	135
	数値排他的論理和	042	レジスタと数値のビット毎の排他的論理和をとります。	136
	ビット・テスト	043	レジスタと数値の論理積をとり、結果が0の時にOFFになります。	137
デ ー タ 変 換	2の補数	046	レジスタの2の補数をとります。	138
	バイナリ変換	050	B C Dコードを2進数に変換します。	139
	単長B C D変換	051	1ワードの2進数をB C Dコードに変換します。	140
	倍長B C D変換	052	2ワードの2進数をB C Dコードに変換します。	141
	エンコード	053	データの最上位の"1"データのビット位置を格納します。	142
	デコード	054	指定されたビット位置に"1"を格納します。	143
特 殊 関 数	ビット・カウンタ	055	データ内の"1"がセットされているビットの数を格納します。	144
	上限リミット	060	指定値で上限リミットします。	145
	下限リミット	061	指定値で下限リミットします。	146
	最大値	062	テーブル上のデータの最大値を格納します。	147
	最小値	063	テーブル上のデータの最小値を格納します。	148
	平均値	064	テーブル上のデータの平均値を格納します。	149
	関数発生器	065	折線関数を発生させます。	150

	命 令		機 能	ページ
		FUN No.		
特 殊 関 数	開平	070	開平演算($\sqrt{\quad}$)を実行します。	151
	正弦関数	071	正弦関数を実行します。	152
	逆正弦関数	072	逆正弦関数を実行します。	153
	余弦関数	073	余弦関数を実行します。	154
	逆余弦関数	074	逆余弦関数を実行します。	155
	デバイスセット	080	デバイスをセットします。	156
	デバイスリセット	081	デバイスをリセットします。	157
診 断	診断表示	090	エラーコードの登録をし、特殊コイルをONにします。	158
	診断表示メッセージ付	091	エラーコードの登録をし、特殊コイルをONにします(メッセージ付)	159
入 出 力	直接入力	096	入力モジュールから直接データを入力します。	160
	直接出力	097	出力モジュールへ直接データを出力します。	161
A S C I I	アスキー入力	098	EX側からBASICプログラムを起動させ、そのデータをEX側へ書込みます。	162
	アスキー出力	099	EX側からデータを書込み、BASICプログラムを起動させます。	163
ス テ ッ ク エ ン ・ ス	ステップ・シーケンス イニシャライズ	100	ステップ・シーケンスのイニシャライズをします。	164
	ステップ・シーケンス入力	101	ステップ・シーケンスの入力です。	165
	ステップ・シーケンス出力	102	ステップ・シーケンスの出力です。	166
特 殊 命 令	フリップ・フロップ	110	リセット優先のフリップ・フロップを行います。	167
	アップ・ダウン・カウンタ	111	入力がONする回数をカウントアップ又はカウントダウンします。	168
	シフト・レジスタ	112	任意のビット長のデータを1ビット左へシフトします。	169

	命 令	FUN No.	機 能	ページ
スタック操作	PUSH	200	スタックにデータをプッシュします。	170
	POPL	201	最後にプッシュされたデータをポップします。	171
	POPF	202	最初にプッシュされたデータをポップします。	172
データ転送	テーブル→ビット転送	203	テーブル上のビットデータを転送します。	173
	ビット→テーブル転送	204	ビットデータをテーブル上に転送します。	174
	インデックス付ワード→ワード	206	インデックス修飾後のレジスタからデータを転送します。	175
	ワード→インデックス付ワード	207	データをインデックス修飾後のレジスタへ転送します。	176
	インデックス付ビット→ビット	208	インデックス修飾後のデバイスからビットデータを転送します。	177
	ビット→インデックス付ビット	209	ビットデータをインデックス修飾後のデバイスへ転送します。	178
四則演算	符号付加算	210	符号付データの加算を行います。	179
	符号付減算	211	符号付データの減算を行います。	180
	符号付乗算	212	符号付データの乗算を行います。	181
	符号付除算	213	符号付データの除算を行います。	182
比較	符号付比較>	214	符号付データの比較を行い>が成立する時出力がONになります。	183
	符号付比較＝	215	符号付データの比較を行い＝が成立する時出力がONになります。	184
	符号付比較<	216	符号付データの比較を行い<が成立する時出力がONになります。	185
	符号付比較≥	217	符号付データの比較を行い≥が成立する時出力がONになります。	186
	符号付比較<>	218	符号付データの比較を行い<>が成立する時出力がONになります。	187
	符号付比較≤	219	符号付データの比較を行い≤が成立する時出力がONになります。	188
四則演算	符号付倍長加算	220	符号付倍長データの加算を行います。	189
	符号付倍長減算	221	符号付倍長データの減算を行います。	190
	符号付倍長乗算	222	符号付倍長データの乗算を行います。	191
	符号付倍長除算	223	符号付倍長データの除算を行います。	192
比較	符号付倍長比較>	224	符号付倍長データの比較を行い>が成立する時, 出力がONになります。	193
	符号付倍長比較＝	225	符号付倍長データの比較を行い＝が成立する時, 出力がONになります。	194
	符号付倍長比較<	226	符号付倍長データの比較を行い<が成立する時, 出力がONになります。	195
	符号付倍長比較≥	227	符号付倍長データの比較を行い≥が成立する時, 出力がONになります。	196
	符号付倍長比較<>	228	符号付倍長データの比較を行い<>成立する時, 出力がONになります。	197
	符号付倍長比較≤	229	符号付倍長データの比較を行い≤が成立する時, 出力がONになります。	198

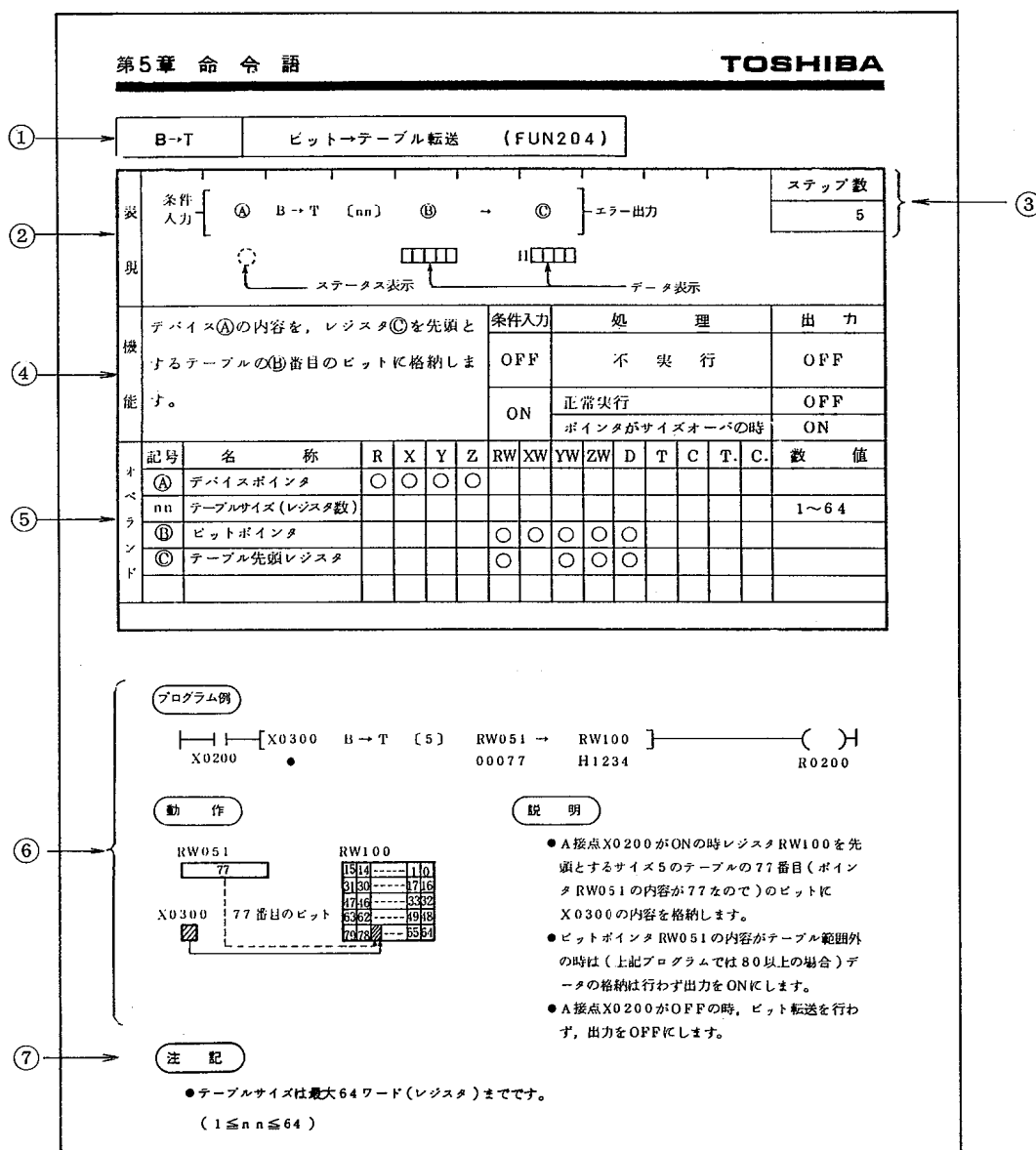
	命 令	FUN No.	機 能	ページ
フ ァ ィ ル 演 算	ビットファイル論理積	230	2つのファイルのビット毎の論理積をとります。	199
	ビットファイル論理和	231	2つのファイルのビット毎の論理和をとります。	200
	ビットファイル排他的論理和	232	2つのファイルのビット毎の排他的論理和をとります。	201
	ビットファイル反転	233	ファイルのビット毎のデータの反転を行います。	202
	ビットファイル右ローテート	234	ファイルを指定ビット数だけ右方向に回転させます。	203
	ビットファイル左ローテート	235	ファイルを指定ビット数だけ左方向に回転させます。	204
	ビットファイルビットセット	236	ファイル内の指定されたビットをセットします。	205
	ビットファイルビットリセット	237	ファイル内の指定されたビットをリセットします。	206
	ビットファイルビットテスト	238	ファイル内の指定されたビットのテストを行います。	207
	ビットファイル比較	239	テーブル間のビットデータの比較を行います。	208
デ バ ー タ 変 換	倍長バイナリ変換	240	倍長のBCDコードを2進数に変換します。	209
	ダブルワード変換	241	単長データを倍長データに拡張します。	210
特 殊 命 令	ドラムシーケンサ	246	ドラムシーケンス処理を行います。	211
	タイマートリガ	247	入力OFFからONに変化したら0からカウントアップを始めます。	215
	SFC用 ステップ・イニシャライズ	248	SFC使用時のイニシャライズ処理を行います。	216
伝 送	伝送送信	250	連続したデータの送信を行います。	217
	伝送受信	251	連続したデータの受信を行います。	220
特 殊 関 数	符号付上限リミット	260	符号付の指定値で上限リミットします。	222
	符号付下限リミット	261	符号付の指定値で下限リミットします。	223
	符号付最大値	262	テーブル上の符号付データの最大値を格納します。	224
	符号付最小値	263	テーブル上の符号付データの最小値を格納します。	225
	符号付平均値	264	テーブル上の符号付データの平均値を格納します。	226
	符号付関数発生器	265	符号付の折線関数を発生させます。	227
	符号付デッドバンド	266	デッド・バンドの判定と処理を行います。	228
デ バ ー タ 変 換	絶対値	267	データの絶対値をとります。	229
	倍長絶対値	268	倍長データの絶対値をとります。	230

	命 令	FUN 記号	機 能	ページ
特殊関数	符号付開平	270	符号付データの開平演算($\sqrt{\quad}$)を実行します。	231
	符号付正弦関数	271	符号付で正弦関数を実行します。	232
	符号付逆正弦関数	272	符号付で逆正弦関数を実行します。	233
	符号付余弦関数	273	符号付で余弦関数を実行します。	234
	符号付逆余弦関数	274	符号付で逆余弦関数を実行します。	235
	積分	275	積分を実行します。	236
	ランプ関数	276	ランプ関数を発生させます。	237
	速度型PID	277	微分先行速度型PID演算を行います。	238
	偏差2乗型PID	278	偏差2乗型PID演算を行います。	239
ベリ ジ 制 御	くり返しコールループ		サブルーチンプログラムを呼び出します(くり返し回数指定可)	241
	サブルーチン・エントリ		サブルーチンプログラムの入口を示します。	242
	サブルーチン・リターン		サブルーチンプログラムの終了を示します。	243
	直接ジャンプ		指定ページにジャンプします。	244
	ジャンプ・ラベル		ジャンプする入口を示します。	245

5-2 各命令語

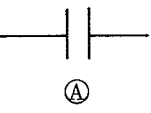
5-2-1 各命令語仕様の見方

5-2-2の各命令語仕様の説明は次のようになっています。



5-2-2 各命令語仕様

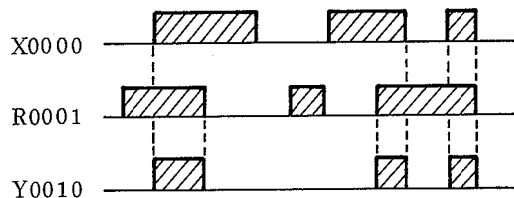
— —	A 接 点
-----	-------

表 現															ステップ数
															1
機 能	デバイス番号(A)に相当するリレーコイルの A接点(常時開接点)です。						条件入力	処 理						出 力	
							OFF	デバイスの内容にかかわらず						OFF	
							ON	デバイスの内容が“0”(OFF)の時 “ ” “1”(ON) “ ”						OFF	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	(A)	デバイス番号	○	○	○	○								○ ○	

プログラム例



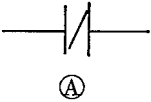
動 作



説 明

- デバイスX0000, R0001の内容が共に
1(ON)の時のみコイルY0010がONになります。

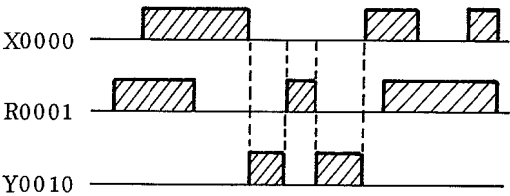
	B 接 点
---	-------

表 現																ステップ数
																1
機 能	デバイス番号①に相当するリレーコイルの B 接点（常時閉接点）です。					条件入力		処 理							出 力	
						OFF		デバイスの内容にかかわらず							OFF	
						ON		デバイスの内容が“0”(OFF)の時 “1”(ON)の時							ON OFF	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	デバイス番号	○	○	○	○								○	○	

プログラム例

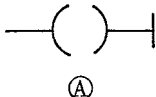


動 作

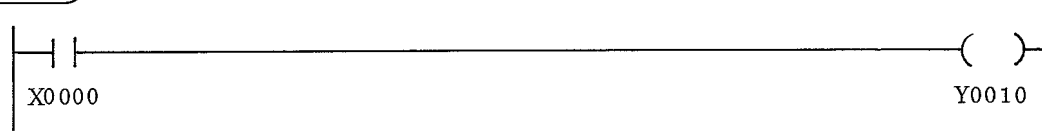


説 明

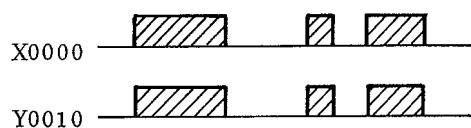
- デバイス X0000, R0001 の内容が共に 0 (OFF) の時のみコイル Y0010 の入力 が ON になります。

-()-		コ イ ル														
表 現															ステップ数	
															1	
機 能	デバイス番号①のリレーコイルです。						条件入力		処 理						出 力	
							OFF		デバイスに 0 を格納							
							ON		" 1 を "							
オ ペ ラ ン ド	記号	名 称		R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	①	デバイス番号		○		○	○									

プログラム例



動 作



説 明

- A 接点 X0000 が ON すると、デバイス Y0010 に 1 が格納され (リレーコイル ON) OFF すると 0 が格納されます。(リレーコイル OFF)

注 記

- コイルの右側に他の命令を書く事はできません。
- SFC の遷移部には、プログラムできません。

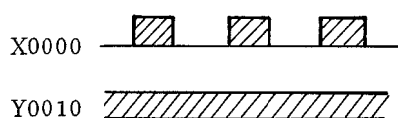
※()	フ ェ ー ス ト ・ コ イ ル
------	-------------------

表 現	入力 $\times \left(\begin{array}{c} \text{ } \end{array} \right) \text{---} $ ①														ステップ数		
															1		
機 能	入力がONであってもOFFであっても前の状態を保持します。						条件入力		処 理						出 力		
							OFF		前の状態を保持								
							ON		〃								
オ ペ ラ ン ド	記号	名 称		R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	デバイス番号		○		○	○										

プログラム例



動 作

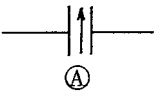


説 明

- フォーストされたコイルは入力条件にかかわらず、前の状態を保持します。
- 左の例では、ONの状態を保持していますが、プログラムのデータ設定によって強制的にリセットされると、OFFの状態を保持します。

注 記

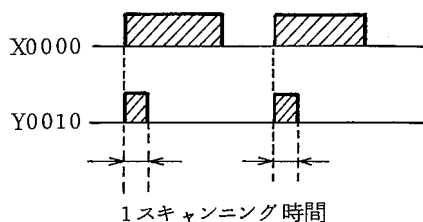
- デバイス①が二重コイルとして使用されている場合は、フォースト・コイルとしての動作は保証できません。

— ↑ — パルス (ON時微分接点)															
表 現		ステップ数 1													
機 能	入力 OFF から ON に変化した時のみ, 1 スキャン時間だけ出力を ON にします。					条件入力		処 理						出 力	
						OFF		以前の出力状態にかかわらず						OFF	
						ON		前回スキャン時の入力が OFF の時 // ON の時						ON OFF	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	①	デバイス番号	○												

プログラム例



動 作



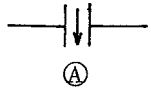
説 明

- A 接点 X0000 が ON になった時から 1 スキャン時間だけ、パルス接点 R0100 の出力が ON になります。

注 記

- 入力が ON となる変化を取り出してパルス信号を作るのに使用します。
- デバイス①は前回スキャン時の入力状態を貯えるのに用いられます。
- デバイス①の 2 重使用はしないで下さい。

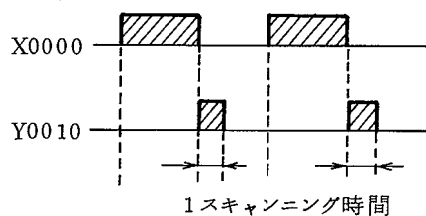
	パルス (OFF時微分接点)
---	----------------

表 現																ステップ数
																1
機 能	入力が入力ONからOFFに変化した時のみ1 スキャンニング時間だけ出力をONにしま す。						条件入力		処 理						出 力	
							OFF		前回スキャン時の入力が入力OFFの時						OFF	
									" ONの時						ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値

プログラム例



動 作



説 明

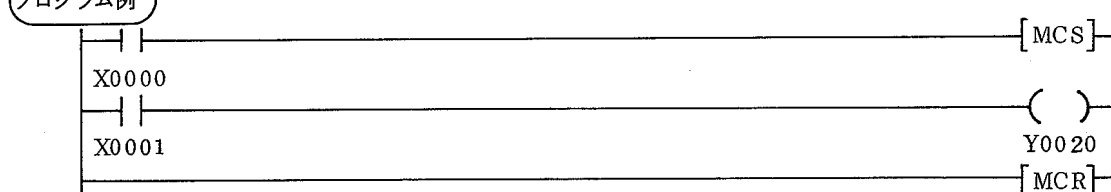
- A接点X0000がONからOFFになってから1スキャンニング時間だけ、パルス接点R0100の出力がONになります。

注 記

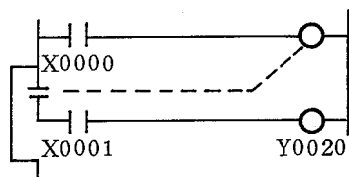
- 入力が入力ONからOFFとなる変化を取り出してパルス信号を作るのに使用します。
- デバイスAは前回スキャン時の入力状態を貯えるのに用いられます。
- デバイスAの、2重使用はしないで下さい。
- SFCの遷移部には、プログラムできません。

MCS MCR		マスターコントロールセット・リセット																		
表 現	条件 入力	マスターコントロールセット														ステップ数				
		マスターコントロールリセット														1				
機 能	MCSの入力がONの時は通常動作を行い、 MCSの入力がOFFの時は、MCS～MCR の間の母線入力がOFFになります。							条件入力		処 理							出 力			
								OFF		MCS - MCR間の母線入力をOFFにする。										
								ON		通 常 動 作										
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値

プログラム例



動 作



等価回路

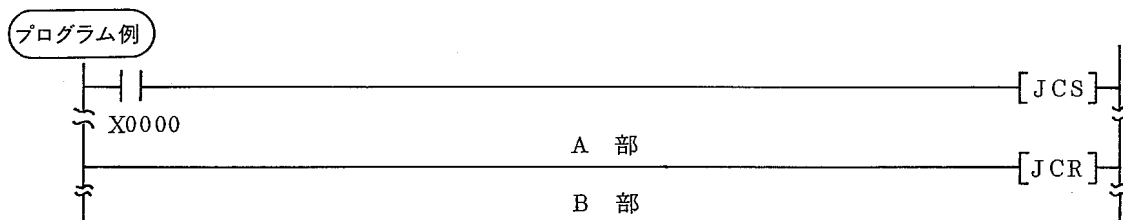
説 明

- デバイスX0000がONの時は通常動作を行います。つまり、X0001がONの時はY0020がONでX0001がOFFの時はY0020はOFFになります。
- デバイスX0000がOFFの時はMCSの入力がOFFになるので、MCS-MCR間の母線入力がOFFとなり、デバイスX0001がONであってもOFFであってもコイルY0020はOFFとなります。

注 記

- MCS, MCRは必ずペアで使用する必要があります。
- MCS, MCRをネスティングすることはできません。(MCSが連続で2回以上プログラミングされた場合)
- MCRの入力条件は不要です。
- SFCの実行部、遷移部にはプログラムできません。

JCS JCR		ジャンプコントロールセット・リセット																		
表 現	条件 入力 JCS JCR ジャンプコントロールセット ジャンプコントロールリセット															ステップ数				
																1				
機 能	入力がONになった時，JCS～JCRの間 の命令は高速読み飛ばしを行い，JCR 以降の命令から実行を始めます。							条件入力		処 理						出 力				
								OFF		不 実 行										
								ON		ジャンプ命令を実行										
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値



説 明

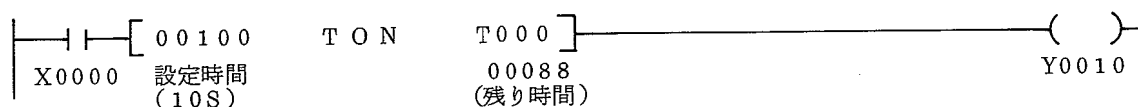
- A接点X0000（入力条件）がONになった場合は，A部の命令は実行せず，B部の命令まで高速読み飛ばしを行い，B部の命令から実行を始めます。
- A接点X0000がOFFの場合は，JCS命令，JCR命令を無視し，A部，B部の命令を実行します。

注 記

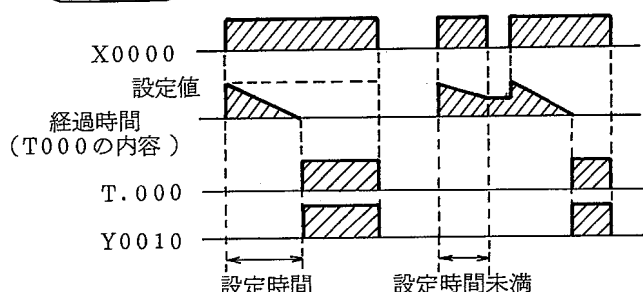
- JCS，JCRは必ずペアで使用することが必要です。
- JCSとJCRの順序が逆になるとエラーになります。
- JCSは連続で2回以上プログラミングすることはできません。
- JCR命令の入力条件は不要です。条件が入っても動作上コイル単独の場合と同じです。
- SFCの実行部，遷移部にはプログラムできません。

TON		オンディレイタイマ														
表 現	機 能	条件入力 [① T O N ②] タイマ出力 符号なし データ表示														ステップ数
																2
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	条件入力		処 理						出 力	
							OFF	不 実 行						OFF		
ON	残り時間 > 0						OFF									
	残り時間 = 0						ON									
①	設定時間					○	○	○	○	○					0~32767	
②	残り時間											○				

プログラム例



動 作



説 明

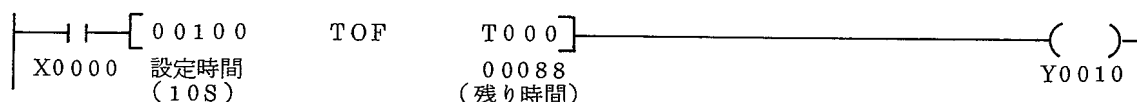
- A接点X000がONになってから10秒経過すると、TONの出力(タイマデバイスT.000)及びリレーコイルY0010はONになります。
残り時間はT000に格納されています。
- X0000が設定時間未満でOFFした場合は、T000は現在値を保持し、出力はOFFのままです。

注 記

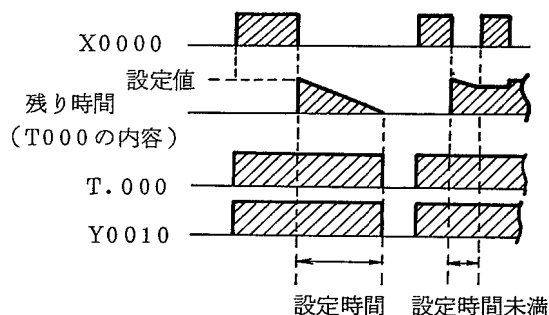
- 設定時間はT000~T449は0.1秒単位(MAX.3276.7S)
T450~T499は0.01秒単位(MAX.3276.7S)になります。
- 残り時間が0に達した後、残り時間は更新されず、0のままとなります。
- 設定時間①は数値、レジスタどちらでも使用できます。
- 停電記憶指定されたタイマーに関しては、電源OFF時に残り時間及び出力は保持されます。

TOF		オフディレイタイマ														ステップ数	
表 現	条件 入力	① TOF ② タイマ出力															

プログラム例



動 作



説 明

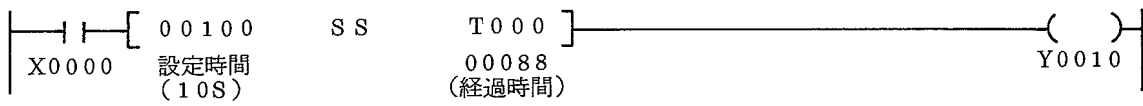
- A 接点 X000 が ON→OFF になってから 10 秒経過すると、TOF の出力 (タイマデバイス T.000) 及びリレーコイル Y0010 は ON→OFF になります。
- A 接点 X0000 が ON の時は、T000 に設定時間をロードし、出力も ON にします。
また、X0000 が設定時間未満で OFF した場合は再び T000 に設定時間をロードし、出力は ON のままです。

注 記

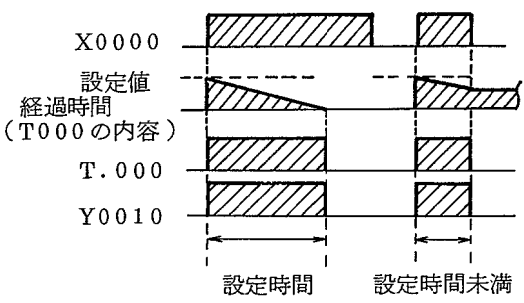
- 設定時間は T000 ~ T449 は 0.1 秒単位 (MAX. 3276.7S)
T450 ~ T499 は 0.01 秒単位 (MAX. 3276.7S) になります。
- 残り時間が 0 に達した後、残り時間は更新されず、0 のままとなります。
- 設定時間①は数値、レジスタどちらでも使用できます。
- 停電記憶指定されたタイマーに関しては電源 OFF 時、残り時間及び出力は保持されます。
- SFC の遷移部にはプログラムできません。

SS		シングルショット																
表 現	条件入力 [① SS ②] タイマ出力 符号なしデータ表示															ステップ数		
																2		
機 能	入力がONになった時，設定時間の間出力をONにします。							条件入力	処 理							出 力		
								OFF	不 実 行							OFF		
								ON	残り時間 > 0							ON		
残り時間 = 0							OFF											
オ ペ ラ ン ド	記号	名 称		R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	設 定 時 間						○	○	○	○	○						0～32767
	②	残 り 時 間											○					

プログラム例



動作



説明

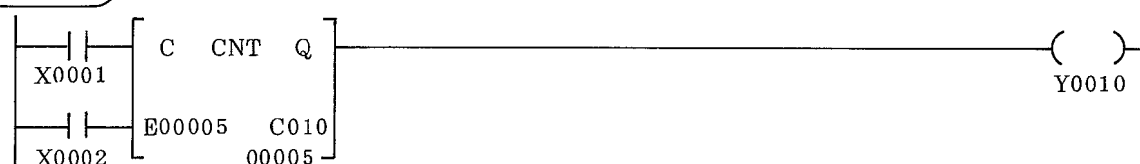
- A接点X0000がONになってから10秒間だけ，SSの出力（タイマデバイスT.000）及びリレーコイルY0010がONになります。残り時間はT000に格納されています。
- A接点X0000がOFFの時は，T000を0にクリアして出力もOFFにします。
- 残り時間が設定時間に達しないうちに，入力X0000がOFFになった時は，T000を保持し，出力をOFFにします。（出力は入力がONの時間だけONにします。）

注 記

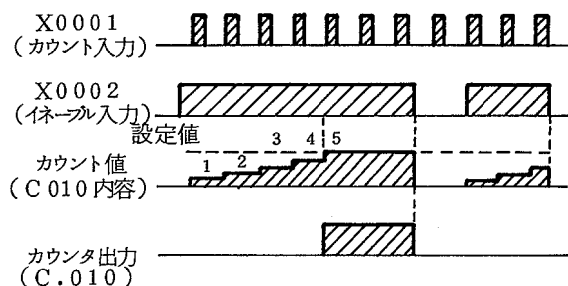
- 設定時間は，T000～T449は0.1秒単位（MAX.3276.7S）
T450～T449は0.01秒単位（MAX.327.67S）になります。
- 残り時間が0に達した後，残り時間は更新されず0のままとなります。
- 設定時間①は数値，レジスタどちらでも使用できます。
- 停電記憶指定されたタイマーに関しては電源がOFF時，残り時間及び出力は保持されます。

C N T		カ ウ ン タ														ステップ数	
表 現																2	
		イネーブル入力 ON の時、入力の OFF→ON への変化回数をカウントし、カウント値 B が設定値 A に達すると、出力を ON にします。															
機 能	条件入力	処 理												出 力			
	E = OFF	不 実 行 (カ ウ ン ト 値) ← 0												OFF			
	E = ON	カ ウ ン ト 値 < 設 定 値												OFF			
		カ ウ ン ト 値 = 設 定 値												ON			
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	A	設 定 値					○	○	○	○	○					0 ~ 65535	
	B	カ ウ ン ト 値											○				

プログラム例



動 作

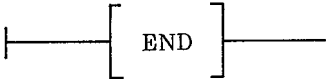


注 記

- カウンタ命令の前にパルス接点をおく必要はありません。(入力の OFF から ON への立上がりは CNT 命令自身が検出します。)
- カウンタをカスケード接続することにより、MAX 設定値以上のカウント値を得られます。
- 設定値 ① は、数値、レジスタどちらでも使用できます。設定範囲は 0 ~ 65535 です。

説 明

- A 接点 X0002 (イネーブル入力) が ON の時のみ、A 接点 X0001 (カウント入力) が OFF から ON になる回数をカウントします。カウント値は C010 に格納されます。カウント値が設定値 (00005) に達すると CNT の出力 (カウンタデバイス C.010) が ON になります。以後イネーブル入力 ON のうちは、カウント入力があってもカウント値は更新されず、出力は ON のままです。
- A 接点 X0002 (イネーブル入力) が OFF になると、カウント値をクリアし、出力も OFF します。また、カウント中にイネーブル入力が OFF した時はカウント値はクリアし、出力も OFF のままです。

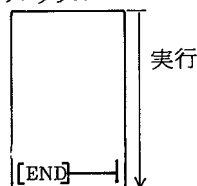
E N D		エ ン ド															
表 現																ステップ数	
																1	
機 能		メイン, サブ, 割込みプログラムの終了命令です。						条件入力				処 理				出 力	
								/				/				/	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値		

プログラム例

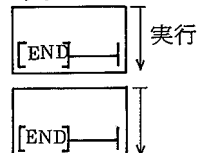


動 作

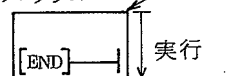
メインプログラム



サブプログラム



割込プログラム



説 明

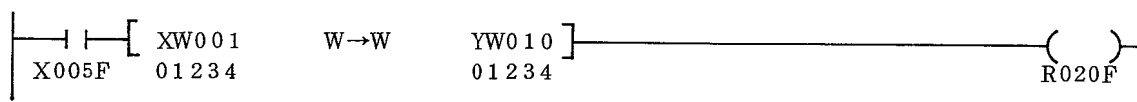
- 各プログラムの先頭ページから, END命令までのプログラムを実行します。
- END命令以降にプログラムを追加してもかまいませんが, 実行はしません。ただしステップ数としては, カウントしますので注意して下さい。

注 記

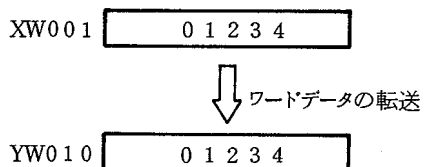
- END命令は, プログラム中必ず1コ入れて下さい。
(2つ以上あってもかまいません。)
- SFCの実行部, 遷移部にはプログラムできません。

W→W		レジスタ転送 (FUN000)																		
表 現	条件入力 [① W→W ②] 実行出力 データ表示															ステップ数				
																3				
機 能	レジスタ①のデータをレジスタ②に格納します。						条件入力	処　　理							出　　力					
							OFF	不　実　行							OFF					
							ON	実　　行							ON					
オ ペ ラ ン ド	記号	名　　　　称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数　　値
	①	転送データ									○	○	○	○	○	○				
	②	転送先レジスタ									○		○	○	○					

プログラム例



動 作



説 明

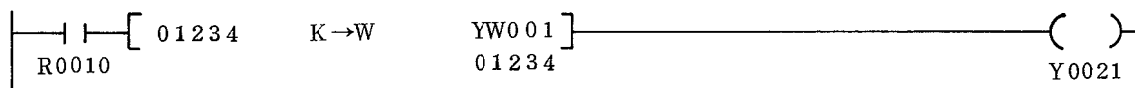
- A接点X005FがONの時、レジスタXW001の内容(01234)をレジスタYW010に格納し、出力をONにします。
- A接点X005FがOFFの時は実行せず、出力をOFFにします。

注 記

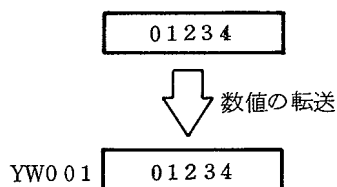
- 転送データ①はレジスタの転送のため、数値は使用できません。

K→W		数 値 転 送 (FUN001)															
表 現	条件 入力	① K→W ② — 実行出力														ステップ数	
		↑ データ表示														3	
機 能	数値①をレジスタ②に格納します。							条件入力	処 理							出 力	
								OFF	不 実 行							OFF	
								ON	実 行							ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値		
	①	転送データ													-32768~65535		
	②	転送先レジスタ					○		○	○	○						

プログラム例



動 作



説 明

- A接点R0010がONの時，数値1234をレジスタYW01に格納し，出力をONにします。
- A接点R0010がOFFの時は，実行せず，出力をOFFにします。

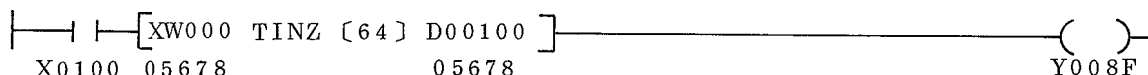
注 記

- この命令は数値（イミディエート）の転送ですので，レジスタの転送はできません。

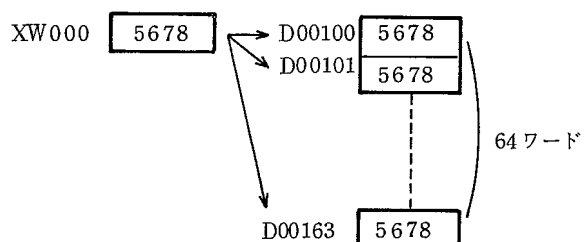
TINZ	テーブル初期化 (FUN002)
------	------------------

表 現	条件 入力 ① TINZ [nn] ② 実行出力 データ表示														ステップ数						
															4						
機 能	レジスタ②を先頭としたサイズ[nn]の テーブルの内容を全て、レジスタ①の内容 でイニシャライズを行います。						条件入力		処 理						出 力						
							OFF		不 実 行						OFF						
							ON		実 行						ON						
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	イニシャライズレジスタ									○	○	○	○	○	○	○				
	②	テーブル先頭レジスタ									○		○	○	○						
	nn	テーブルサイズ																		1～64	

プログラム例



動 作



説 明

- A接点X0100がONの時、レジスタXW000の内容5678を、レジスタD00100を先頭としたサイズ64のテーブルに転送を行ない、出力をONにします。
- A接点X0100がOFFの時、実行はせず、出力をOFFにします。

注 記

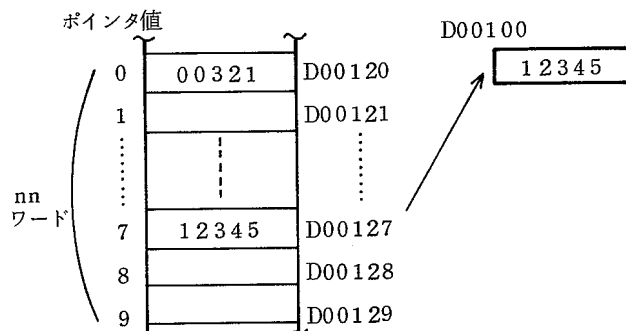
- テーブルサイズは最大64ワード(レジスタ)迄です。($1 \leq nn \leq 64$)
- テーブルは、先頭レジスタから1ワードとして数えたものになります。従ってテーブルの最終は(B+nn-1)になります。
- 数値をオペランド①に使用することはできません。
- イニシャライズのテーブルは、各レジスタ空間の有効範囲内であることが必要です。

T→W		マルチプレクサ (FUN003)														
表 現	条件 入力	[① T→W [nn] ② → ③] テーブルフレーム オーバー出力													ステップ数	
		データ表示 <td>5</td>													5	
機 能	レジスタ①を先頭としたサイズ[nn]のテ ーブルの②番目のレジスタの内容をレジス タ③に格納します。							条件入力	処 理							出 力
								OFF	不 実 行							OFF
								ON	正 常 実 行 時							OFF
							テーブル・フレーム・オーバー時							ON		
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	テーブル先頭レジスタ					○	○	○	○	○	○				
	②	テーブルポインタ					○	○	○	○	○	○				
	③	転送先レジスタ					○		○	○	○					
	nn	テーブルサイズ														1～64

プログラム例

$\left[\begin{array}{c} \text{D00120} \\ \text{X0070} \end{array} \right. \text{T→W [10]XW008} \rightarrow \left. \begin{array}{c} \text{D00100} \\ \text{00007} \end{array} \right] \text{R0301}$

動 作



注 記

- テーブルサイズは最大64ワード(レジスタ)迄です。($1 \leq nn \leq 64$)
- テーブルのレジスタは、テーブルの先頭を0として数えられたものになります。従って転送を行うレジスタは①+②となります。
- テーブルは、各レジスタ空間の有効範囲内である必要があります。

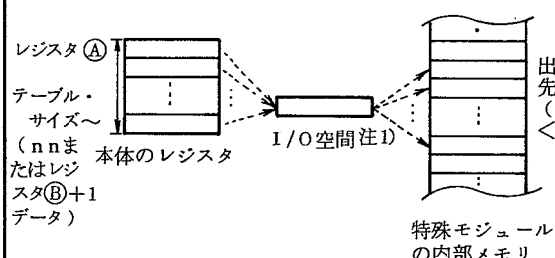
説 明

- A接点X0070がONの時、レジスタD00120を先頭とした、サイズ10ワードのテーブルの7番目(ポインタXW008の内容が7なので)レジスタD00127の内容を、D00100に格納します。この時、正常にデータ転送を行ったので、出力をOFFにします。
- XW008の内容がテーブル外のレジスタを指している時(上記プログラムの例では、XW008の内容が9を越えた時)、データ転送を行わずに出力をONにします。
- A接点X0070がOFFの時、データ転送を行わず、出力をOFFにします。

T → W

マルチプレクサ (特殊モジュール用データ出力命令)

動 作



注1) I/O割付けによって決定するYWレジスタであるレジスタ③と対応するI/Oデータ空間です。

説 明

特殊モジュール用データ出力命令は、マルチプレクサ命令内の特殊モードです。このモードは特殊モジュール(4ch PIモジュール等)へのデータ出力を行なうモードです。

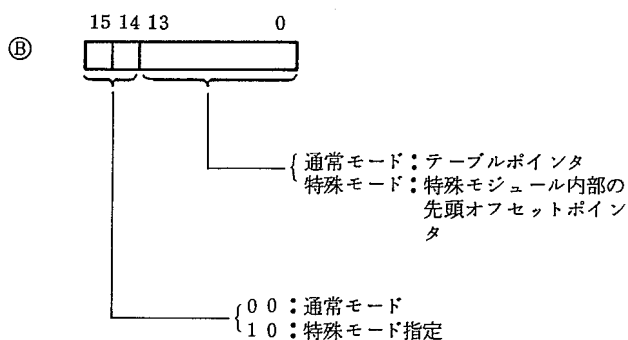
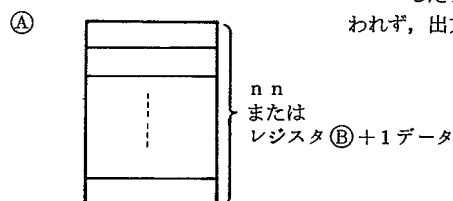
各モード切替はレジスタ②のビット<15>、<14>により替ります。(この命令はEX2000“V 2.1”以降でしたら使用できます。)

○モード

- 1.通常モード：前頁を参照して下さい。
- 2.特殊モード：レジスタ①をテーブルの先頭としたサイズnn(または、レジスタ②+1データ)のテーブルデータをレジスタ③にて指定された特殊モジュールに対して出力します。
転送元である特殊モジュール内部の先頭アドレスはレジスタ②のビット<13>～<0>にて示されます。命令出力は、入力OFF時、または正常実行時にOFFにします。

オペランド説明

注) 本テーブルがレジスタ空間をオーバーした場合、実行処理は行われず、出力はONにします。



nn：テーブルサイズ

通常モード： $1 \leq nn \leq 64$

特殊モード： $1 \leq nn \leq 63$ 、 $nn=64$ のとき、レジスタ②+1データがテーブルサイズです。

但し、 $1 \leq (\text{レジスタ②}+1 \text{ データ}) \leq 128$ が有効範囲です。範囲外の場合、実行処理は行われず、出力をONにします。また、レジスタ②+1がレジスタ空間をオーバーした場合も同様です。

③：転送先指定

通常モード：転送先レジスタ

特殊モード：YWレジスタのみ有効で、レジスタ番号に対応する特殊モジュールをさします。

命令実行後は、特殊モジュールに出力した最終データが格納されます。

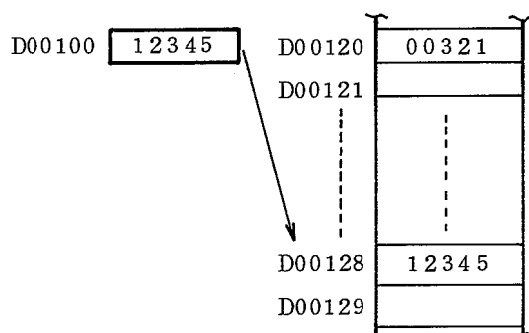
YWレジスタ以外の場合、実行処理は行われず、出力をONにします。

W→T		デマルチプレクサ (FUN004)												ステップ数						
表 現	条件 入力	① W→T [nn] ② → ③ テーブルフレーム オーバー出力 データ表示												5						
機 能	レジスタ①の内容をレジスタ③を先頭とするサイズ [nn] のテーブルの②番目のレジスタに格納します。						条件入力		処 理						出 力					
							OFF		不 実 行						OFF					
							ON		正 常 実 行 時						OFF					
									テーブル・フレーム・オーバー時						ON					
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	転送レジスタ									○	○	○	○	○	○	○			
	②	テーブルポインタ									○	○	○	○	○	○	○			
	③	テーブル先頭レジスタ									○		○	○	○					
	nn	テーブルサイズ																		1～64

プログラム例

$\left[\begin{array}{c} \text{D00100} \\ \text{X0071} \end{array} \right] \text{W} \rightarrow \text{T} [10] \text{XW008} \rightarrow \text{D00120} \left[\begin{array}{c} \text{00321} \\ \text{R0302} \end{array} \right]$

動 作



説 明

- A接点X0071がONの時、レジスタD00100の内容12345を、レジスタD00120を先頭としたサイズ10のテーブルの8番目（ポインタXW008の内容が8なので）のレジスタD00128に格納します。この時出力はOFFにします。
- ポインターレジスタXW008の内容がテーブル外を指している時（上記のプログラム例では9を越える時）、データの転送を行わずに、出力をONにします。
- A接点X0071がOFFの時、データ転送を行わず、出力をOFFにします。

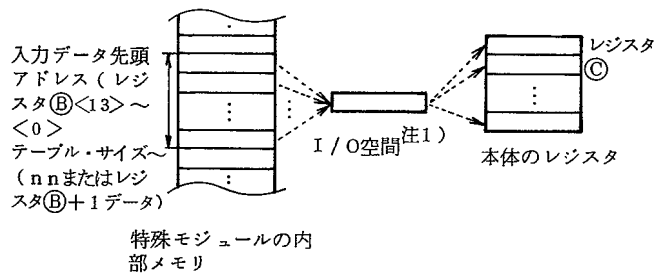
注 記

- テーブルサイズは最大64ワード（レジスタ）迄です。（ $1 \leq nn \leq 64$ ）
- 格納先レジスタは、テーブルの先頭レジスタを0として数えたものになります。従って格納先レジスタは③+②となります。
- 転送先のテーブルは、各レジスタ空間の有効範囲内である必要があります。

W → T

デマルチプレクサ (特殊モジュール用データ入力命令)

動作



注1) I/O割付けによって決定するXWレジスタであるレジスタ $\textcircled{A}$ に対応するI/Oデータ空間です。

説明

特殊モジュール用データ入力命令は、デマルチプレクサ命令内の特殊モードです。このモードは、特殊モジュール(4ch PIモジュール等)からデータ入力を行なうモードです。

各モード切替は、レジスタ $\textcircled{B}$ のビット<15>、<14>により替ります。(この命令はEX2000“V2.1”以降でしたら使用できます。)

モード

- 1.通常モード：前頁を参照下さい。
- 2.特殊モード：レジスタ $\textcircled{A}$ にて指定された特殊モジュールのレジスタ $\textcircled{B}$ のビット<13>~<0>にて示された特殊モジュール内部の先頭アドレスからサイズnn(またはレジスタ $\textcircled{B}$ +1データ)の内部データを、レジスタ $\textcircled{C}$ を先頭とするテーブルに入力します。命令出力は入力OFF時または、正常実行時にOFFにします。

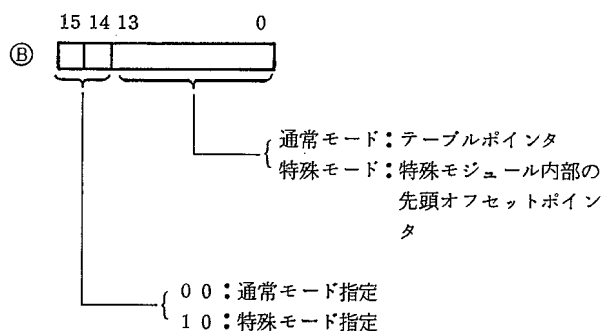
オペランド説明

 $\textcircled{A}$ ：転送元指定

通常モード：転送元レジスタ

特殊モード：XWレジスタのみ有効で、レジスタ番号に対応する特殊モジュールをさします。

命令実行後は、特殊モジュールから入力した最終データが格納されます。XWレジスタ以外の場合実行処理は行われず出力をONにします。



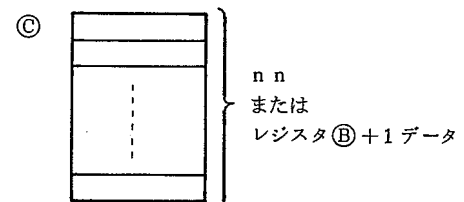
nn：テーブルサイズ

通常モード： $1 \leq nn \leq 64$

特殊モード： $1 \leq nn \leq 63$ 、 $nn=64$ のとき、レジスタ $\textcircled{B}$ +1データがテーブルサイズになります。

但し、 $1 \leq (\text{レジスタ}\textcircled{B}+1\text{データ}) \leq 128$ が有効範囲です。

範囲外の場合、実行処理は行われず出力をONにします。また、レジスタ $\textcircled{B}$ +1がレジスタ空間をオーバーした場合も同様になります。



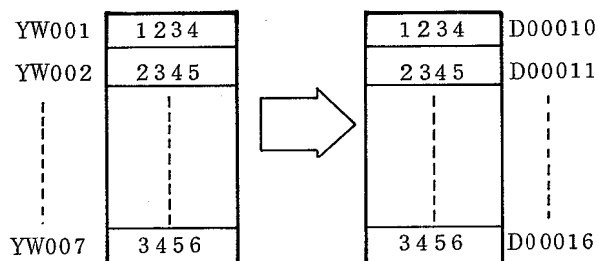
注)本テーブルがレジスタ空間をオーバーした場合、実行処理は行われず出力をONにします。

T→T	テーブル転送 (FUN005)
-----	-----------------

表現	条件入力 ① T → T [nn] ② 実行出力 符号なし データ表示														ステップ数	
															4	
機能	レジスタ①を先頭とするサイズ [nn] のテーブルデータを、レジスタ②以降にブロック転送します。						条件入力		処 理						出 力	
							OFF		不 実 行						OFF	
							ON		実 行						ON	
オペランド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	転送元テーブル先頭レジスタ					○	○	○	○	○	○				
	②	転送先テーブル先頭レジスタ					○		○	○	○					
	nn	テーブルサイズ														1 ~ 64

プログラム例

動 作



説 明

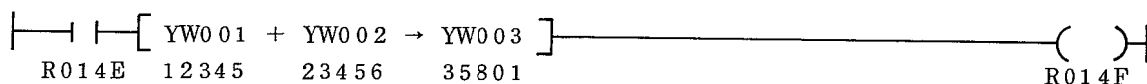
- A接点X000EがONの時、レジスタ YW001から7ワードのレジスタの内容を、レジスタD00010以降にブロック転送し、出力をONにします。
- A接点X000EがOFFの時、転送を行わず出力をOFFにします。

注 記

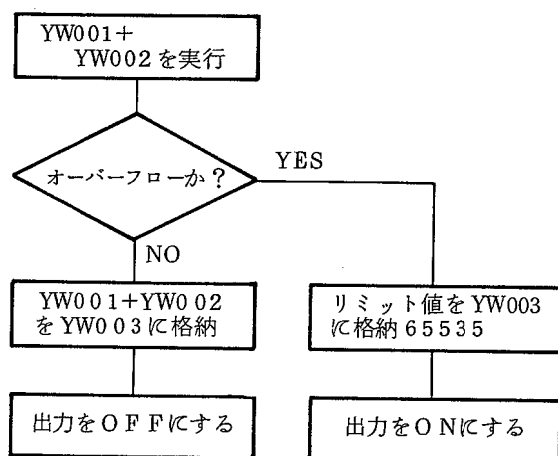
- テーブルサイズは最大64ワード(レジスタ)迄です。($1 \leq nn \leq 64$)
- 転送元及び転送先のテーブルは、各レジスタ空間の有効範囲であることが必要です。
- 転送元と転送先のレジスタが同種であっても、また、テーブル同士が重なってかまいません。

+		レジスタ加算 (FUN010)																		
表 現	条件 入力	① + ② → ③ オーバーフロー出力 符号なしデータ表示														ステップ数				
																4				
機 能	レジスタ①の内容と、レジスタ②の内容の和を求め、レジスタ③に格納します。							条件入力	処 理							出 力				
								OFF	不 実 行							OFF				
								ON	正 常 実 行 時							OFF				
							オーバーフロー							ON						
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	被加算数レジスタ									○	○	○	○	○	○	○			
	②	加数レジスタ									○	○	○	○	○	○	○			
	③	和									○		○	○	○					

プログラム例



動 作



注 記

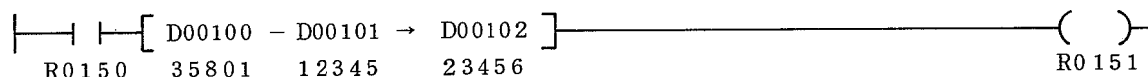
- この命令はレジスタの加算ですので数値は使用できません。
- オペランド①, ②, ③は同一レジスタとなってもかまいません。

説 明

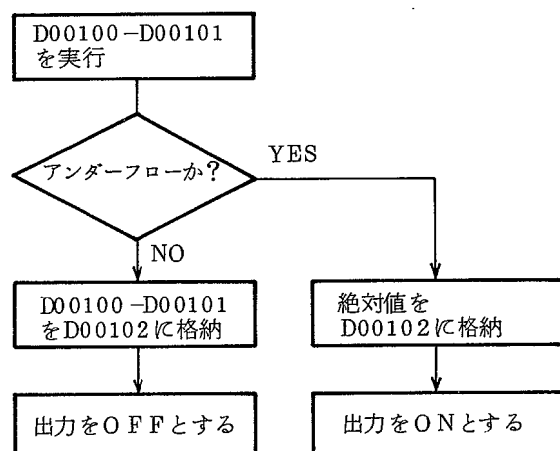
- A接点R014EがONの時、レジスタ YW001の内容12345とレジスタ YW002の内容23456の和を求め、レジスタ YW003に35801を格納します。この時オーバーフローは発生しないので、出力はOFFにします。
- 演算中はオーバーフローが発生すると、リミット値65535を格納します。また、この時出力をONにします。
- A接点R014EがOFFの時、演算を行わず、出力をOFFにします。

[illegible]

プログラム例



動作



說明

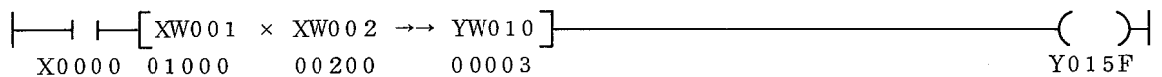
- A接点R0150がONの時、レジスタD00100の内容からレジスタD00101の内容を引き、差をD00102に格納します。アンダーフローが無い時は、出力をOFFにします。
- 演算実行時にアンダーフローが発生すると、絶対値をレジスタCに格納します。また、出力をONにします。
- A接点R0150がOFFの時、演算を行わず、出力をOFFにします。

注 記

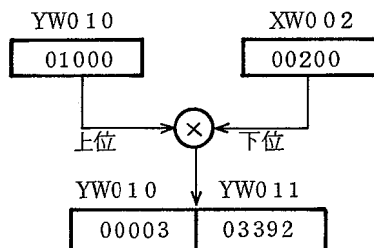
- この命令はレジスタの減算ですので、数値は使用できません。
- オペランド(A, B, C)は同一レジスタとなってもかまいません。

×		レジスタ乗算 (FUN012)															
表 現	条件 入力	条件入力 { ① × ② →→ ③ } 実行出力														ステップ数	
		符号なし データ表示														4	
機 能	レジスタ①の内容とレジスタ②の内容の積を求め、レジスタ③・③+1に格納します。							条件入力	処 理							出 力	
								OFF	不 実 行							OFF	
								ON	実 行							ON	
オ ペ ラ ン ド	記号	名 称		R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	被乗数レジスタ						○	○	○	○	○	○	○			
	②	乗数レジスタ						○	○	○	○	○	○	○			
	③	積						○		○	○	○					

プログラム例



動 作



説 明

- A接点X0000がONの時、レジスタXW001の内容（1000）とレジスタXW002の内容（200）との積を求め、レジスタYW010とYW011の連続したレジスタに積を格納します。YW010に上位、YW011に下位を格納し、出力はONにします。
- A接点X0000がOFFの時は、演算を行わず出力をOFFにします。

注 記

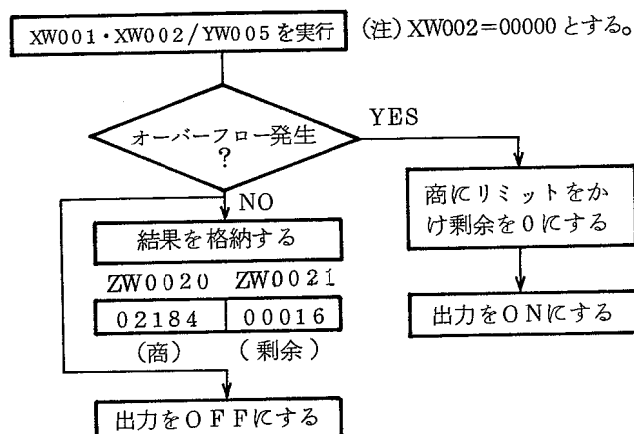
- この命令はレジスタの乗算ですので、数値は使用できません。
- レジスタ③は、③+1も考慮し、レジスタ範囲を越えないよう注意して下さい。
- 演算結果は（上位レジスタ）× 65536 +（下位レジスタ）で表現されます。
例では、 $3 \times 65536 + 3392 = 200000$ となります。
- 内部演算は全て16進で扱っています。

／		レジスタ除算 (FUN013)															
表 現	条件 入力	条件入力 { ① ／ ② → ③ } オーバーフロー出力 符号なし データ表示														ステップ数 4	
		レジスタ①・①+1の内容をレジスタ②の内容で除し、商をレジスタ③へ、剰余をレジスタ③+1へ格納します。															
機 能	条件入力	処 理												出 力			
	OFF	不 実 行												OFF			
	ON	正 常 実 行 時 オーバーフロー発生時												OFF ON			
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	被除数レジスタ					○	○	○	○	○	○					
	②	除数レジスタ					○	○	○	○	○	○					
	③	商					○		○	○	○						

プログラム例

$\left[\begin{array}{c} \text{XW001} / \text{YW005} \rightarrow \text{ZW0020} \\ \text{X0000} \quad 00001 \quad 00030 \quad 02184 \end{array} \right] \xrightarrow{\text{Y0100}}$

動 作



説 明

- A接点X0000がONの時、レジスタXW001の内容(00001)とXW002の内容(00000)の倍長データをレジスタYW005の内容(30)で除します。オーバーフローがなければZW0020に商(02184)が、ZW0021に剰余(16)が格納され、出力はOFFになります。オーバーフロー(除数が0の場合も含む)発生時商にリミット値65535を格納し、剰余は0とし、出力をONにします。
- A接点X0000がOFFの時は、演算を行わず出力をOFFにします。
- 演算結果は $((\text{XW001}) \times 65536 + (\text{XW002})) \div (\text{YW005}) = (\text{ZW0020}) \text{ 余り } (\text{ZW0021})$
 $(1 \times 65536 + 00000) \div 30 = 2184 \text{ 余り } 16$ となります。

注 記

- この命令はレジスタの除算ですので、数値は使用できません。
- レジスタ③は、③+1も考慮し、レジスタ範囲を越えないよう注意して下さい。
- 被除数は①・①+1の倍長データとして扱います。

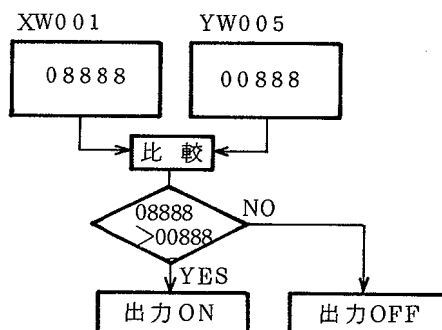
>	比較・より大きい (FUN014)
---	-------------------

表 現	条件入力 ① > ② 判定出力 符号なし データ表示														ステップ数					
															3					
機 能	レジスタ①とレジスタ②の内容を比較し、 ①>②の時に出力をONにします。						条件入力		処 理						出 力					
							OFF		不 実 行						OFF					
							ON		①>②						ON					
①≤②						OFF														
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	比 較 値									○	○	○	○	○	○				
	②	基 準 値									○	○	○	○	○	○				

プログラム例



動 作



説 明

- A接点X0000がONの時、レジスタXW001の内容(8888)とレジスタYW005の内容(888)を比較します。
XW001>YW005の時、出力をONにします。
XW001≤YW005の時、出力をOFFにします。
表示している値では出力はONにします。
- A接点X0000がOFFの時、比較を行わず、出力をOFFにします。

注 記

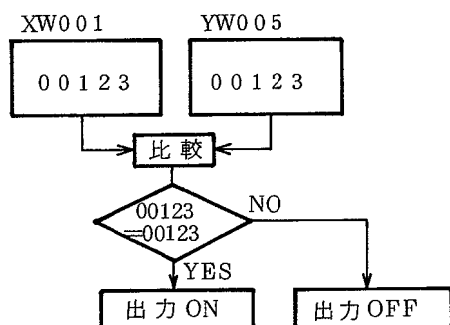
- 本命令はレジスタの比較ですので、数値は使用できません。
- 比較値①，基準値②は比較によって変化しません。

=		比較・等しい (FUN015)														
表 現	条件入力 $\left[\begin{array}{c} \text{A} \\ \text{B} \end{array} \right] =$ 判定出力 符号なし データ表示														ステップ数	
															3	
機 能	レジスタAとレジスタBの内容を比較し、 A=Bの時に、出力をONにします。						条件入力		処 理						出 力	
							OFF		不 実 行						OFF	
							ON		A=B						ON	
A≠B						OFF										
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	A	比 較 値					○	○	○	○	○	○	○			
	B	基 準 値					○	○	○	○	○	○	○			

プログラム例



動 作



説 明

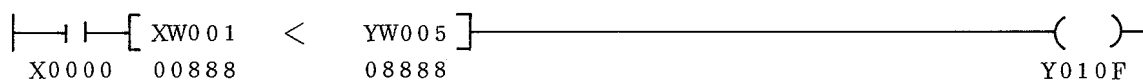
- A接点X0000がONの時、レジスタXW001の内容(00123)とレジスタYW005の内容(00123)を比較します。
XW001=YW005の時、出力をONにします。
XW001≠YW005の時、出力をOFFにします。
表示している値では出力はONにします。
- A接点X0000がOFFの時、比較を行わず、出力をOFFにします。

注 記

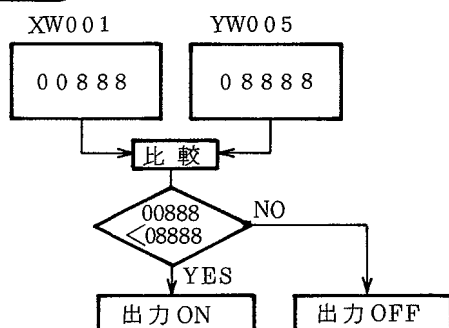
- 本命令はレジスタの比較ですので、数値は使用できません。
- 比較値A, 基準値Bは比較によって変化しません。

<		比較・より小さい (FUN016)														ステップ数	
表 現	機 能	条件入力 [④ < ⑤] 判定出力 符号なし データ表示														3	
		レジスタ④とレジスタ⑤の内容を比較し、 ④<⑤の時に出力をONにします。															
オ ペ ラ ン ド	機 能	条件入力		処 理										出 力			
		OFF		不 実 行										OFF			
		ON		④<⑤										ON			
				④≥⑤										OFF			
		記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値	
④	比 較 値					○	○	○	○	○	○	○					
⑤	基 準 値					○	○	○	○	○	○	○					

プログラム例



動 作



説 明

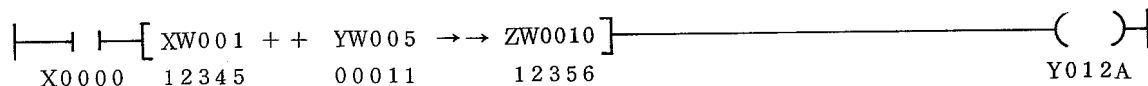
- A接点X0000がONの時、レジスタXW001の内容(888)とレジスタYW005の内容(8888)を比較します。
XW001<YW005の時、出力をONにします。
XW001≥YW005の時、出力をOFFにします。
表示している値では出力をONにします。
- A接点X0000がOFFの時、比較を行わず、出力をOFFにします。

注 記

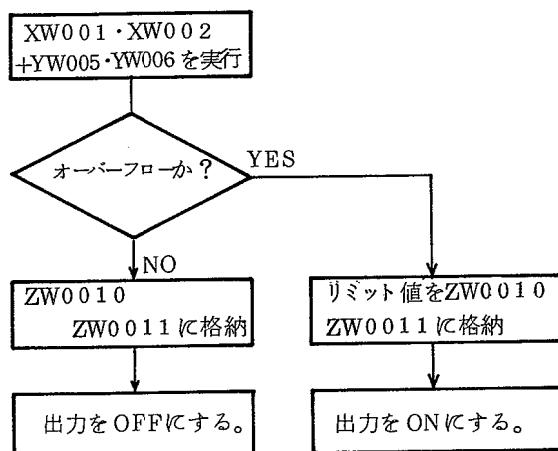
- 本命令はレジスタの比較ですので、数値は使用できません。
- 比較値④, 基準値⑤は比較によって変化しません。

++		レジスタ倍長加算 (FUN017)														ステップ数	
表 現	機 能	条件入力 { ① ++ ② → → ③ } オーバーフロー出力 符号なしデータ表示														4	
		レジスタ①・①+1の内容とレジスタ②・②+1の内容の和を求め、レジスタ③・③+1に格納します。															
オペ ラ ン ド	記号	名 称		R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
		① 被加算数レジスタ						○	○	○	○	○					
		② 加数レジスタ						○	○	○	○	○					
		③ 和						○		○	○	○					

プログラム例



動 作



説 明

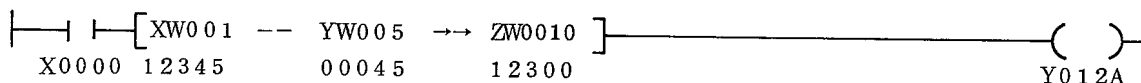
- A接点X0000がONの時、レジスタXW001・XW002の連続したレジスタの内容をYW005・YW006の連続したレジスタの内容に加え、オーバーフローがなければ和をZW0010・ZW0011の連続したレジスタに格納し出力をOFFにします。
- 演算中オーバーフローが発生するとリミット値“HFFFFFFF”をレジスタ③・③+1に格納し出力をONにします。
- A接点X0000がOFFの時、演算を行わず出力をOFFにします。

注 記

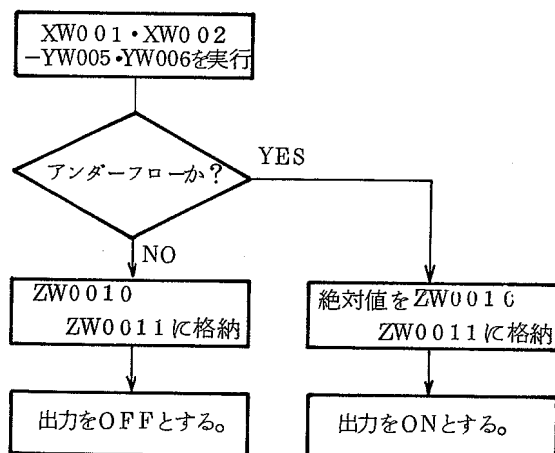
- 本命令はレジスタの加算ですので、数値は使用できません。
- レジスタの範囲を超えてしまう入力はできません。
- ①・①+1, ②・②+1, ③・③+1はそれぞれ、32ビットの2進表現の倍長データとして扱われます。
- ①, ②, ③共、偶数、奇数レジスタを問いません。

--		レジスタ倍長減算 (FUN018)																		
表 現	条件 入力	① -- ② →→ ③ アンダーフロー出力 □□□□ □□□□ □□□□ ↑ ↑ ↑ 符号なし データ表示														ステップ数				
																4				
機 能	レジスタ①・①+1の内容からレジスタ②・②+1の内容を引き、レジスタ③・③+1に差を格納します。						条件入力		処 理						出 力					
							OFF		不 実 行						OFF					
							ON		正 常 実 行 時 アンダー・フロー時						OFF ON					
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	被減数レジスタ									○	○	○	○	○					
	②	減数レジスタ									○	○	○	○	○					
	③	差									○		○	○	○					

プログラム例



動 作



説 明

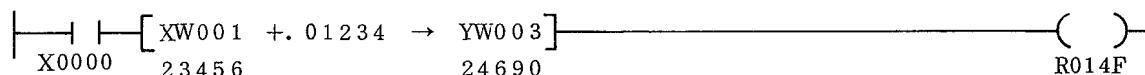
- A接点X0000がONの時、XW001・XW002の連続したレジスタの内容から、YW005・YW006の連続したレジスタの内容を引き、アンダーフローがなければ差をZW0010・ZW0011の連続したレジスタに格納し出力をOFFにします。
- 演算中アンダーフローが発生すると、絶対値をレジスタ③・③+1に格納し、出力をONにします。
- A接点X0000がOFFの時演算を行わず出力をOFFにします。

注 記

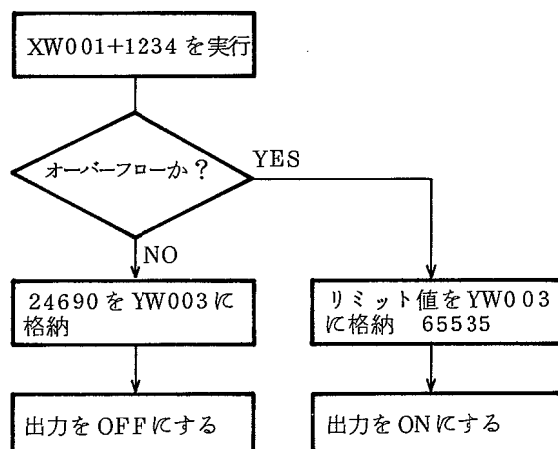
- 本命令はレジスタの減算ですので、数値は使用できません。
- レジスタの範囲を越えてしまう入力はありません。
- ①・①+1, ②・②+1, ③・③+1はそれぞれ、32ビットの2進表現の倍長データとして扱われます。
- ①, ②, ③共に、偶数, 奇数レジスタを問いません。

+.		数 値 加 算 (FUN020)															
表 現	条件入力 ① +. ② → ③ オーバーフロー出力 ④ ⑤ ⑥ ⑦ 符号なしデータ表示														ステップ数		
															4		
機 能	レジスタ①の内容と数値②の和を求めレジスタ③に格納します。							条件入力	処 理							出 力	
								OFF	不 実 行							OFF	
								ON	正 常 実 行 時							OFF	
							オーバー・フロー							ON			
オペランド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	被加算数レジスタ					○	○	○	○	○	○					
	②	加 数														0~65535	
	③	和					○		○	○	○						

プログラム例



動 作



説 明

- A接点X0000がONの時、レジスタXW001の内容23456と数値1234の和を求め、レジスタYW003に24690を格納します。この時オーバーフローは発生しないので出力はOFFにします。
- 演算中オーバーフローが発生するとリミット値65535をレジスタ③に格納します。また出力をONにします。
- A接点X0000がOFFの時、演算を行わず出力をOFFにします。

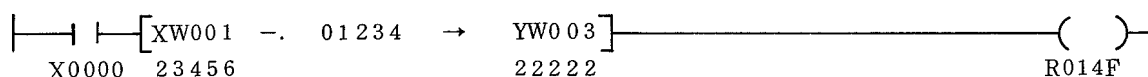
注 記

- 本命令は数値の加算ですので、加数②にはレジスタは使用できません。
- 被加算数レジスタ①、和③は同一レジスタでもかまいません。

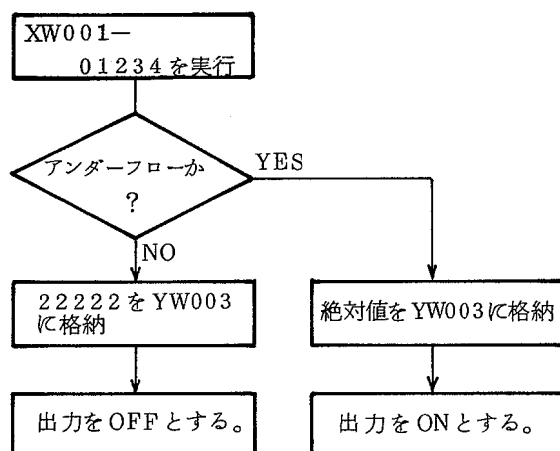
例) D00100 +.1 → D00100 (インクリメント)

- .		数 値 減 算 (FUN021)															
表 現	機 能	条件入力 [① - ② → ③] アンダーフロー出力 符号なしデータ表示														ステップ数	
		4															
機 能	レジスタ①の内容から数値②の内容を引きレジスタ③に格納します。	条件入力		処 理										出 力			
		OFF		不 実 行										OFF			
		ON		正 常 実 行 時										OFF			
				アンダー・フロー時										ON			
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	被減数レジスタ					○	○	○	○	○	○					
	②	減 数														0~65535	
	③	差					○		○	○	○						

プログラム例



動 作



説 明

- A 接点 X0000 が ON の時、レジスタ XW001 の内容 23456 から数値 1234 を引き、差を求めレジスタ YW003 に 22222 を格納します。この時アンダーフローが発生しないので出力は OFF にします。
- 演算中アンダーフローが発生すると絶対値をレジスタ ③ に格納します。また、出力を ON にします。
- A 接点 X0000 が OFF の時演算を行わず、出力を OFF とします。

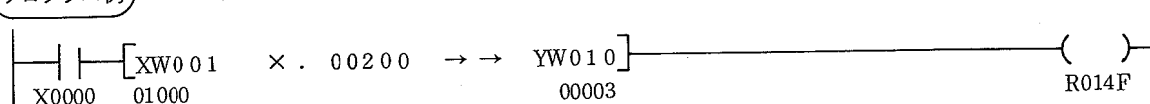
注 記

- 本命令は数値の減算ですので、減算 ② にはレジスタは使用できません。
- 被減数レジスタ ①、差 ③ は同一レジスタでもかまいません。

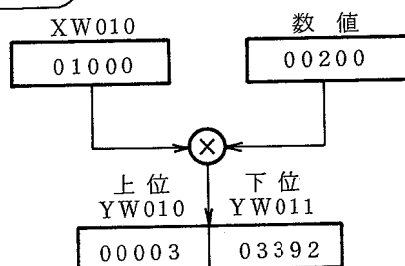
例) D00100 - .1 → D00100 (デクリメント)

x.		数 値 乗 算 (FUN022)															
表 現	条件 入力 ① x. ② →→ ③ 実行出力 符号なし データ表示	ステップ数 4															
機 能	レジスタ①の内容と数値②の積を求め、レジスタ③・③+1に格納します。							条件入力	処 理							出 力	
								OFF	不 実 行							OFF	
								ON	実 行							ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	被乗数レジスタ					○	○	○	○	○	○					
	②	乗 数														0~65535	
	③	積					○		○	○	○						

プログラム例



動 作



説 明

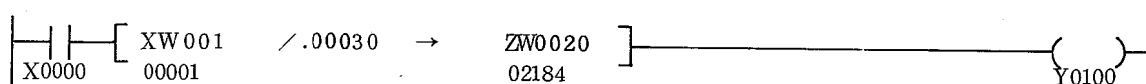
- A 接点X0000がONの時、レジスタXW001の内容(1000)と数値200との積を求め、レジスタYW010とYW011の連続したレジスタに積を格納します。
YW010に上位、YW011に下位を格納し出力はONにします。
- A 接点X0000がOFFの時は、演算を行わず出力をOFFにします。
- 演算結果は(上位レジスタ)×65536+(下位レジスタ)で表現されます。
例では、 $3 \times 65536 + 3392 = 200000$ となります。

注 記

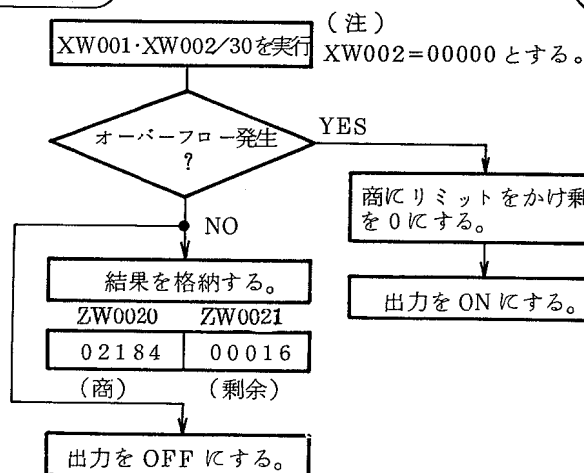
- 本命令は数値の乗算ですので、乗数Bにはレジスタは使用できません。
- レジスタCは、C+1も考慮し、レジスタ範囲を越えないよう注意して下さい。
- 内部演算は全て16進で扱っています。

／.		数 値 除 算 (FUN023)															
表 現	条件 入力	① ／. ② → ③ オーバーフロー 出 力 符号なし データ表示												ステップ数			
														4			
機 能	レジスタ①・①+1の内容を数値②で除し、 商をレジスタ③に、剰除をレジスタ③+1 に格納します。	条件入力		処 理										出 力			
		OFF		不 実 行										OFF			
		ON		正 常 実 行 時 オーバー・フロー発生時										OFF ON			
オ ペ ラ ン ド	記号	名 称		R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	被除数レジスタ						○	○	○	○	○	○	○			
	②	除 数															0~65535
	③	商						○		○	○	○					

プログラム例



動 作



説 明

- A接点X0000がONの時、レジスタXW001の内容(00001)とXW002の内容(00000)の倍長データを数値30で除します。オーバーフローがなければZW0020に商2184が、ZW0021に剰余16が格納され、出力はOFFにします。オーバーフロー(除数が0の場合も含む)発生時商にリミット値65535を格納し、剰余は0とし、出力をONにします。
- A接点X0000がOFFの時は、演算を行わず出力をOFFにします。
- 演算結果は((XW001)×65536+(XW002))÷30=ZW0020)余り(ZW0021)
(1×65536+0)÷30=2184 余り16となります。

注 記

- 本命令は数値の除算ですので、除数②にはレジスタは使用できません。
- レジスタ③は、③+1も考慮し、レジスタ範囲を越えないよう注意して下さい。
- 被除数は①・①+1の倍長データとして扱います。

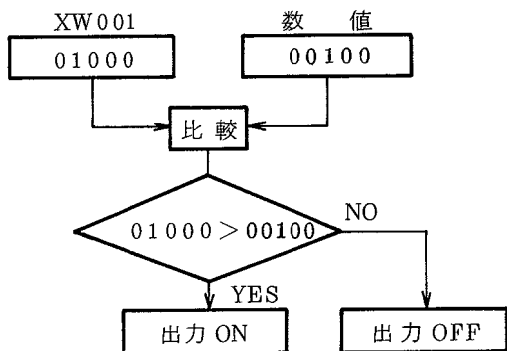
>.	数値比較・より大きい (FUN024)
----	---------------------

表 現	条件入力 ① >. ② 判定出力														ステップ数
	□ □ □ □ □ 符号なしデータ表示														3
機 能	レジスタ①と数値②の内容を比較し、①>②の時に出力をONにします。					条件入力		処 理							出 力
						OFF		不 実 行							OFF
						ON		① > ② ① ≤ ②							ON
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	①	比 較 値					○	○	○	○	○	○			
	②	基 準 値													0~65535

プログラム例



動 作



説 明

- A 接点X0000がONの時、レジスタXW001の内容1000と数値100を比較します。
XW001>100の時、出力をONにします。
XW001≤100の時、出力をOFFにします。
表示している値では出力はONにします。
- A 接点X0000がOFFの時、比較を行わず出力をOFFにします。

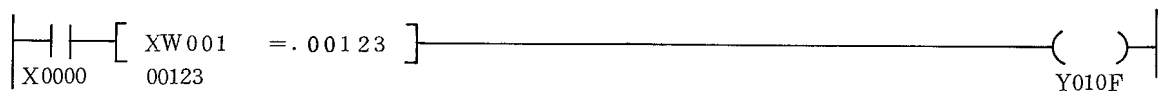
注 記

- 本命令は数値の比較ですので、基準値②にはレジスタは使用できません。

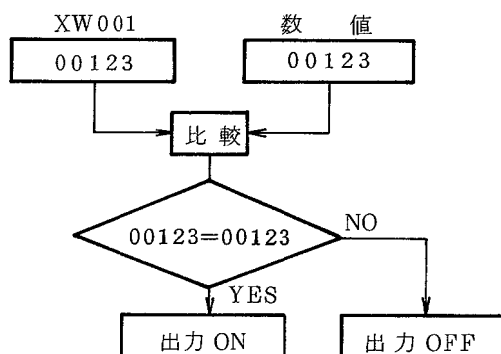
=.	数 値 比 較 ・ 等 し い (FUN025)
----	--------------------------

表 現	条件入力 [① = ②] 判定出力															ステップ数				
	↑ 符号なし データ表示															3				
機 能	レジスタ①の内容と数値②を比較し、① = ②の時に出力を ON にします。							条件入力	処 理							出 力				
								OFF	不 実 行							OFF				
								ON	① = ②							ON				
① ≠ ②							OFF													
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	比 較 値									○	○	○	○	○	○				
	②	基 準 値																		0～65535

プログラム例



動 作



説 明

- A接点X0000がONの時、レジスタXW001の内容00123と数値123を比較します。
XW001 = 123の時、出力をONにします。
XW001 ≠ 123の時、出力をOFFにします。
表示している値では出力はONにします。
- A接点X0000がOFFの時、比較を行わず出力をOFFにします。

注 記

- 本命令は数値の比較ですので、基準値②にはレジスタは使用できません。

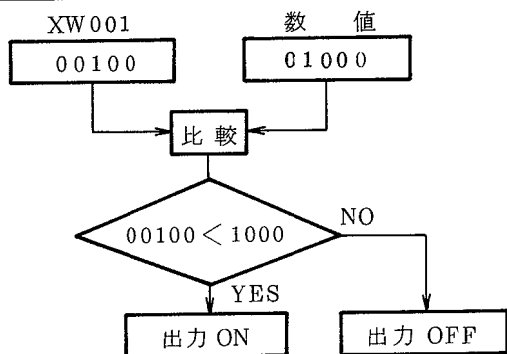
< .	数値比較・より小さい (FUN026)
-----	---------------------

表 現	条件入力 ① < . ② 判定出力														ステップ数																																																														
	□ □ □ □ □ 符号なしデータ表示														3																																																														
機 能	レジスタ①の内容と数値②を比較し、①<②の時に出力をONにします。														出力																																																														
	<table> <tr> <th>条件入力</th><th colspan="14">処 理</th><th>出 力</th></tr> <tr> <td>OFF</td><td colspan="14">不 実 行</td><td>OFF</td></tr> <tr> <td rowspan="2">ON</td><td colspan="14">① < ②</td><td>ON</td></tr> <tr> <td colspan="14">① ≥ ②</td><td>OFF</td></tr> </table>														条件入力	処 理														出 力	OFF	不 実 行														OFF	ON	① < ②														ON	① ≥ ②														OFF
条件入力	処 理														出 力																																																														
OFF	不 実 行														OFF																																																														
ON	① < ②														ON																																																														
	① ≥ ②														OFF																																																														
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値																																																														
	①	比 較 値					○	○	○	○	○	○																																																																	
	②	基 準 値													0~65535																																																														

プログラム例



動 作



説 明

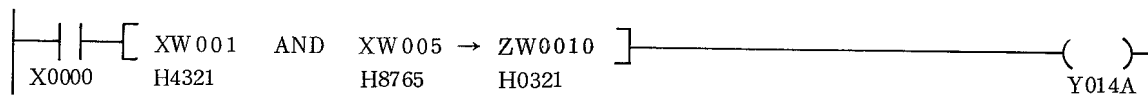
- A接点X0000がONの時、レジスタXW001の内容100と数値1000を比較します。
XW001<1000の時、出力をONにします。
XW001≥1000の時、出力をOFFにします。
表示している値では出力はONにします。
- A接点X0000がOFFの時、比較を行わず出力をOFFにします。

注 記

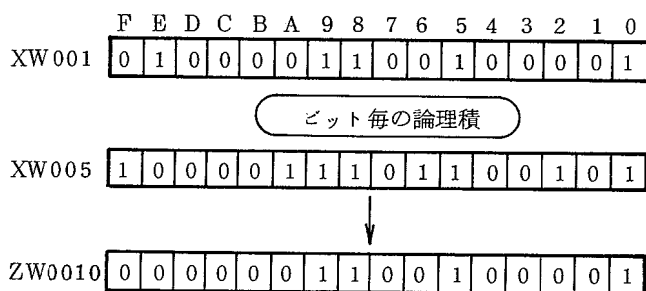
- 本命令は数値の比較ですので、基準値②にはレジスタは使用できません。

AND		レジスタ論理積 (FUN030)																		
表 現	条件入力 ① AND ② → ③ 実行出力 H H H データ表示	ステップ数																		
		4																		
機 能	レジスタ①の内容とレジスタ②の内容のビット毎の論理積を求め、レジスタ③に格納します。						条件入力	処 理							出 力					
							OFF	不 実 行							OFF					
							ON	実 行							ON					
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	演算データ									○	○	○	○	○	○				
	②	"									○	○	○	○	○	○				
	③	論 理 積									○		○	○	○					

プログラム例



動 作



説 明

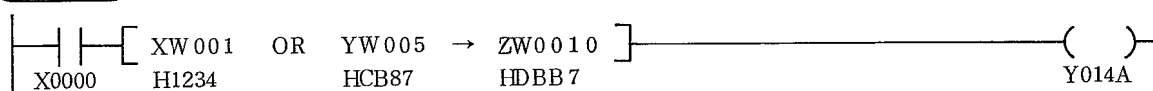
- A 接点X0000がONの時、レジスタXW001とXW005のワードデータのビット毎の論理積をとり、レジスタZW0010に格納します。また、この時出力をONにします。
- A 接点X0000がOFFの時、演算を行わず、出力をOFFにします。

注 記

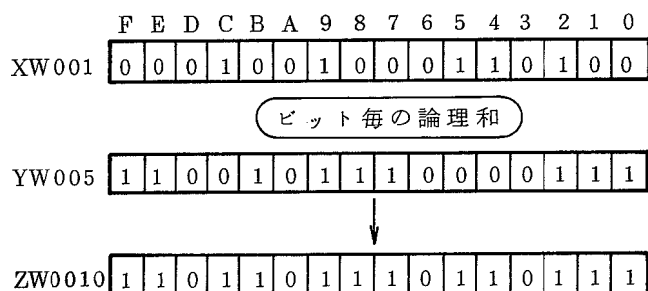
- 本命令はレジスタの論理積ですので、数値は使用できません。

OR		レジスタ論理和 (FUN031)																		
表 現	条件入力 ① OR ② → ③ 実行出力 H□□□ H□□□ H□□□ データ表示														ステップ数					
															4					
機 能	レジスタ①の内容とレジスタ②の内容のビット毎の論理和を求め、レジスタ③に格納します。						条件入力		処 理						出 力					
							OFF		不 実 行						OFF					
							ON		実 行						ON					
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	演算データ									○	○	○	○	○	○	○			
	②	"									○	○	○	○	○	○	○			
	③	論 理 和									○		○	○	○					

プログラム例



動作



說明

- A接点X0000がONの時、レジスタXW001とYW005のワードデータのビット毎の論理和をとり、レジスタZW0010に格納します。
また、この時出力をONにします。
- A接点X0000がOFFの時、演算を行わず、出力をOFFにします。

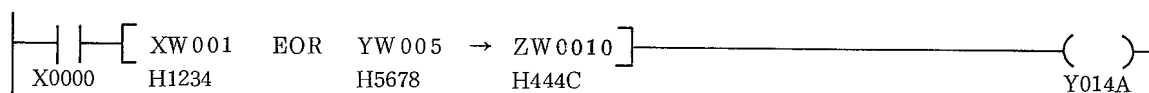
注 記

- 本命令はレジスタの論理和ですので、数値は使用できません。

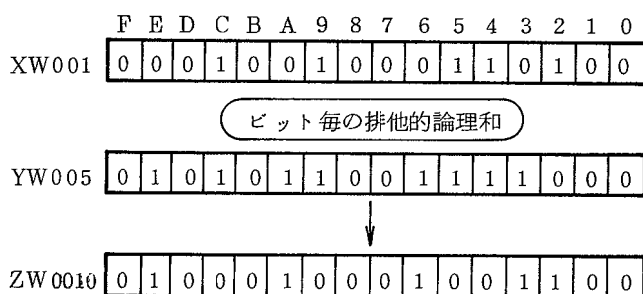
E O R	レジスタ排他的論理和 (FUN032)
-------	---------------------

表 現															ステップ数
															4
機 能	レジスタ①の内容とレジスタ②の内容のビット毎の排他的論理和を求めレジスタ③に格納します。														条件入力
															処 理
															出 力
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	①	演算データ					○	○	○	○	○	○			
	②	"					○	○	○	○	○	○			
	③	排他的論理和					○		○	○	○				

プログラム例



動 作



説 明

- A接点X0000がONの時、レジスタXW001とYW005のワードデータのビット毎の排他的論理和をとり、レジスタZW0010に格納します。また、この時出力をONにします。
- A接点X0000がOFFの時、演算を行わず出力をOFFにします。

注 記

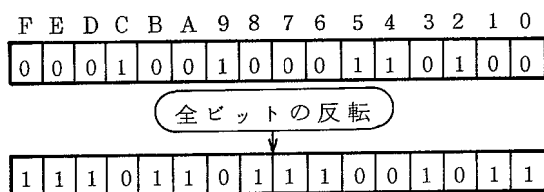
- 本命令はレジスタの排他的論理和ですので、数値は使用できません。

NOT		反 転 (FUN034)																							
表 現	条件入力 ① NOT ② 実行出力 H <table><tr><td></td><td></td><td></td><td></td></tr></table> H <table><tr><td></td><td></td><td></td><td></td></tr></table> データ表示																							ステップ数	
3																									
機 能	レジスタ①の内容を反転しレジスタ②へ格納します。							条件入力		処 理						出 力									
								OFF		不 実 行						OFF									
								ON		実 行						ON									
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値									
	①	演算データ					○	○	○	○	○	○													
	②	反転データ					○		○	○	○														

プログラム例



動 作



説 明

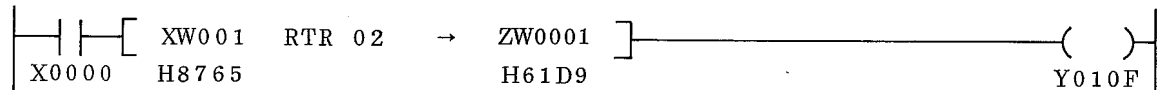
- A 接点 X0000 が ON の時、レジスタ XW001 全ビットの反転データをレジスタ YW005 に格納し、出力を ON にします。
- A 接点 X0000 が OFF の時、演算を行わず、出力を OFF にします。

注 記

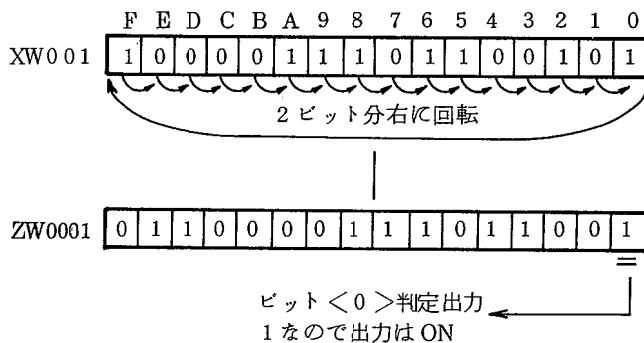
- 本命令はレジスタの反転ですので、数値は使用できません。

RTR		右 ロ ー テ ー ト (FUN035)																		
表 現	条件 入力	① RTR ② → ③														ステップ数				
		ビット<0> 判定出力														4				
		H□□□□ H□□□□																		
		データ表示																		
機 能	レジスタ①の内容を②で示すビット数だけ 右へ回転させたものをレジスタ③へ格納し ます。						条件入力	処 理								出 力				
							OFF	不 実 行								OFF				
							ON	実 行						ビット0が1の時 ビット0が0の時		ON OFF				
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	演算データ									○	○	○	○	○	○				
	②	ローテートビット数																		0～15
	③	ローテート結果									○		○	○	○					

プログラム例



動 作



説 明

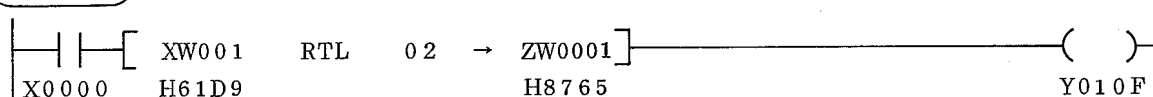
- A接点X0000がONの時、レジスタXW001の内容H8765を2ビット分だけ右に回転させたデータH61D9を、結果のビット<0>=1なので出力はONにします。結果のビット<0>=0の時は出力はOFFにします。
- A接点X0000がOFFの時、ローテートは行わず、出力をOFFにします。

注 記

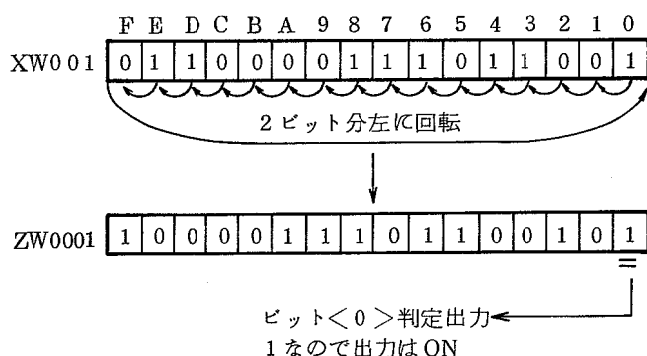
- ローテートビット数②の範囲は0~15が有効です。
- 演算データ①はローテートによって変化しません。

RTL		左 ロ ー テ ー ト (FUN036)												ステップ数											
表 現	機 能	条件入力 [① RTL ② → ③] ビット<0>判定出力 H <table><tr><td></td><td></td><td></td><td></td></tr></table> H <table><tr><td></td><td></td><td></td><td></td></tr></table> データ表示																						4	
レジスタ①の内容を②で示すビット数だけ左へ回転させたものをレジスタ③に格納します。						条件入力		処 理						出 力											
						OFF		不 実 行						OFF											
						ON		実 行		ビット0が1の時		ON													
										ビット0が0の時		OFF													
オ ペ ラ ン ド	記号	名 称				R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値							
	①	演算データ								○	○	○	○	○	○										
	②	ローテートビット数																0~15							
	③	ローテート結果								○		○	○	○											

プログラム例



動 作



説 明

- A接点X0000がONの時、レジスタXW001の内容H61D9を2ビット分だけ左に回転させたデータH8765を、レジスタZW0001に格納します。結果のビット<0>=1なので出力はONにします。結果のビット<0>=0の時は、出力はOFFにします。
- A接点X0000がOFFの時、ローテートは行わず、出力をOFFにします。

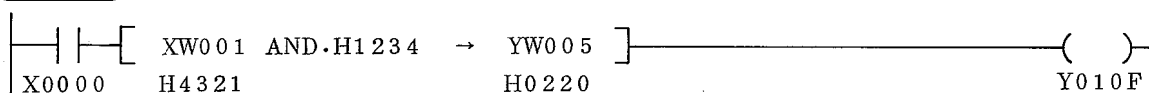
注 記

- ローテートビット数②の範囲は0~15が有効です。
- 演算データ①はローテートによって変化しません。

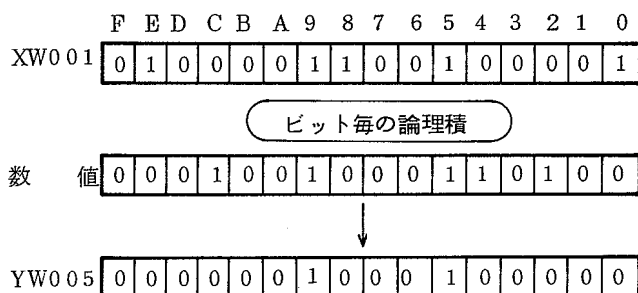
AND.	数 値 論 理 積 (FUN040)
------	--------------------

表 現	条件入力 [① AND. ② → ③] 実行出力														ステップ数		
	H <table><tr><td></td><td></td><td></td><td></td></tr></table>																

プログラム例



動 作



説 明

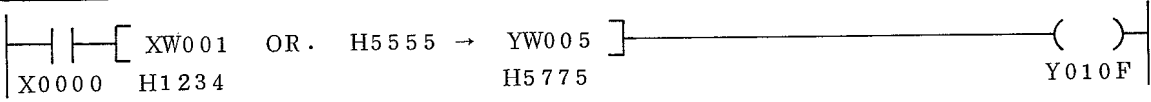
- A接点X0000がONの時、レジスタXW001と数値H1234のワードデータのビット毎の論理積をとり、レジスタYW005に格納します。また、この時出力をONにします。
- A接点X0000がOFFの時、演算を行わず、出力をOFFにします。

注 記

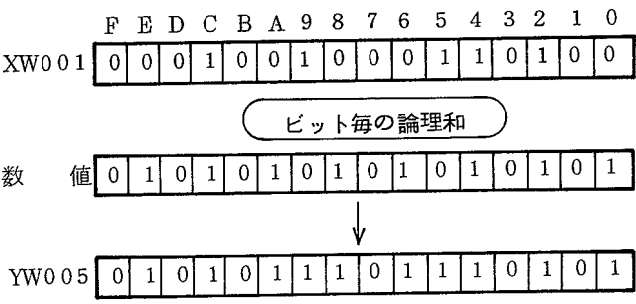
- 本命令は数値の論理積ですので、演算データ②にはレジスタは使用できません。

OR.		数 値 論 理 和 (FUN041)													ステップ数	
表 現															4	
機 能	レジスタ①の内容と数値②のビット毎の論理和を求め、レジスタ③に格納します。					条件入力		処 理							出 力	
						OFF		不 実 行							OFF	
						ON		実 行							ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	演算データ					○	○	○	○	○	○				
	②	"														○
	③	論 理 和					○		○	○	○					

プログラム例



動 作



説 明

- A接点X0000がONの時、レジスタXW001と数値H5555のワードデータのビット毎の論理和をとり、レジスタYW005に格納します。また、この時出力をONにします。
- A接点X0000がOFFの時、演算を行わず出力をOFFにします。

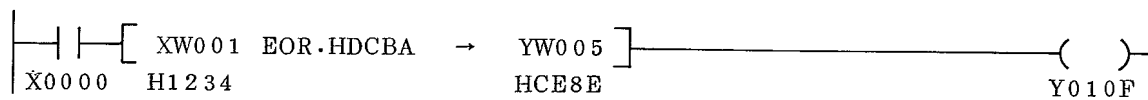
注 記

- 本命令は数値の論理和ですので、演算データ②にはレジスタは使用できません。

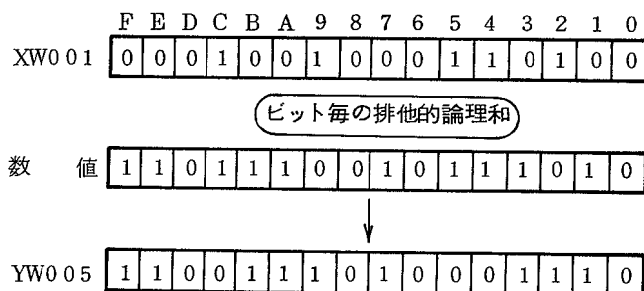
EOR.	数 値 排 他 的 論 理 和 (FUN042)
------	--------------------------

表現	条件入力 ① EOR. ② → ③ 実行出力 H H データ表示														ステップ数	
															4	
機能	レジスタ①の内容と数値②のビット毎の排他的論理和を求めレジスタ③に格納します。						条件入力		処 理						出 力	
							OFF		不 実 行						OFF	
							ON		実 行						ON	
オペランド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	演算データ					○	○	○	○	○	○				
	②	"														○
	③	排他的論理和					○		○	○	○					

プログラム例



動 作



説 明

- A接点X0000がONの時、レジスタXW001と数値HDCBAのワードデータのビット毎の排他的論理値をとり、レジスタYW005に格納します。また、この時出力をONにします。
- A接点X0000がOFFの時、演算を行わず、出力をOFFにします。

注 記

- 本命令は数値の排他的論理和ですので、演算データ②にはレジスタは使用できません。

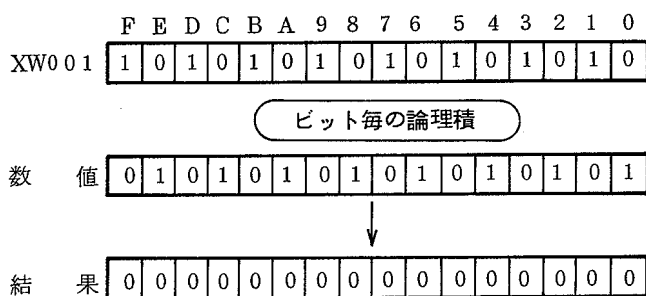
TEST	ビットテスト (FUN043)
------	-----------------

表 現	条件 入力 ① TEST ② } 判定出力 ステップ数 3 H データ表示																
	機 能	レジスタ①の内容と数値②の内容のビット 毎の論理積を求め、0以外であれば出力を ONにし、0であれば出力をOFFにします。						条件入力	処 理						出 力		
OFF								不 実 行						OFF			
ON								AND結果≧0の時						ON			
								AND結果=0の時						OFF			
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	演算データ					○	○	○	○	○	○	○				
	②	テストデータ															○

プログラム例



動作



說明

- A接点X0000がONの時、レジスタXW001と数値H5555のワードデータのビット毎の論理積をとります。
結果＝0なので出力をOFFにします。
結果≠0の時は出力をONにします。
- A接点X0000がOFFの時、演算を行わず、出力をOFFにします。

注 記

- テストデータ⑧は数値入力のみです。

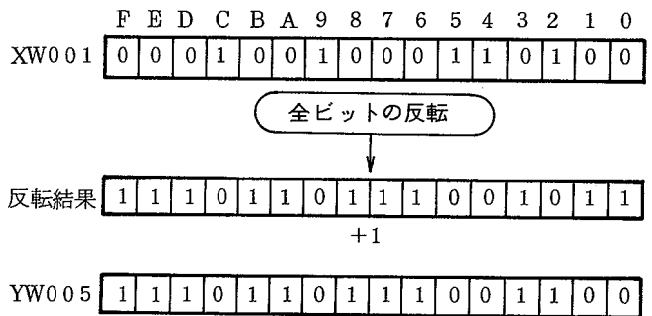
NEG	補 数 (FUN046)
-----	--------------

表 現	条件入力 ① NEG ② 実行出力 H <table><tr><td></td><td></td><td></td><td></td></tr></table> H <table><tr><td></td><td></td><td></td><td></td></tr></table> データ表示																						ステップ数
														3									
機 能	レジスタ①の内容の補数を取りレジスタ②に格納します。						条件入力	処 理							出 力								
							OFF	不 実 行							OFF								
							ON	実 行							ON								
オ ペ ラ ン ド	記号	名 称				R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値				
	①	演算データ								○	○	○	○	○	○	○							
	②	補数データ								○		○	○	○									

プログラム例



動 作



説 明

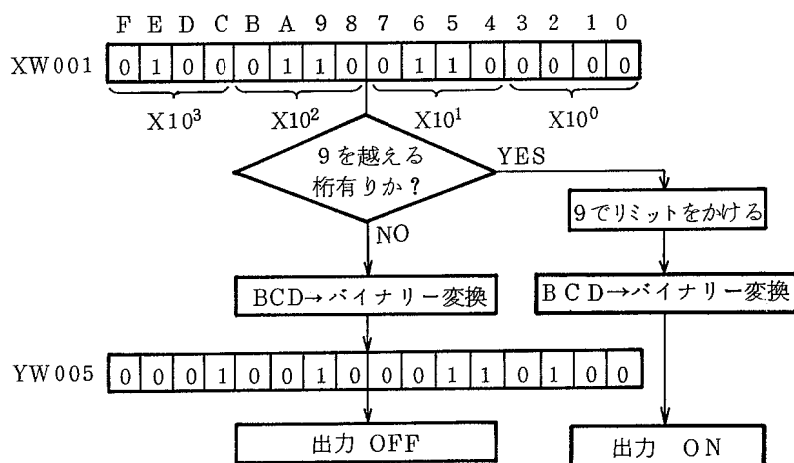
- A 接点X0000がONの時、レジスタXW001の補数（反転データ+1）を取りレジスタYW005に格納し、出力をONにします。
- A 接点X0000がOFFの時、演算を行わず、出力をOFFにします。

BIN		バイナリ変換 (FUN050)														ステップ数	
表 現	機 能	条件入力 [① BIN ②] エラー出力 H 符号なしデータ表示														3	
		レジスタ①の4桁のBCD(2進化10進)データを, 2進数に変換してレジスタ②に格納します。															
オ ペ ラ ン ド	機 能	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値	
		①	演算データ					○	○	○	○	○	○	○			
		②	バイナリデータ					○		○	○	○					

プログラム例



動 作



説 明

- A接点X0000がONの時, XW001の内容を4桁の2進化10進数と見做し, 2進数に変換してYW005に格納します。
正常実行の場合出力をOFFにします。
- 変換するデータを4ビット毎に区切った各桁の内容に9を越えるものがある時は, 9でリミットをかけ演算を行います。この時は出力をONにします。
- A接点X0000がOFFの時, 演算を行わず出力をOFFにします。

注 記

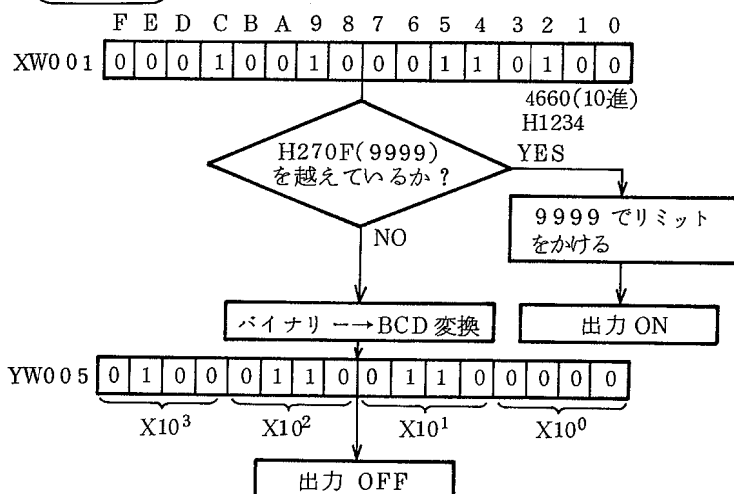
- HABCD等をバイナリ変換する場合は, H9999とし, 変換を行い, 出力をONにします。

BCD 1		B C D 変 換 (FUN051)																		
表 現	条件 入力 ④ BCD 1 ⑤ エラー出力 H 符号なし データ表示	ステップ数 3																		
		レジスタ④の内容を4桁のBCDデータに変換して、レジスタ⑤に格納します。						条件入力		処 理						出 力				
機 能							OFF		不 実 行						OFF					
							ON		実 行				正 常 時 エラー有りの時		OFF ON					
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	④	演算データ (バイナリ)									○	○	○	○	○	○	○			
	⑤	BCD データ									○		○	○	○					

プログラム例



動 作



注 記

- HABCD等をBCD変換する場合は、H9999とし変換を行い、出力をONにします。
- この命令の変換範囲は0~9999(0~H270F)です。
- この命令は単長のBCD変換のみ使用できます。(倍長データを変換する場合はFUN52 BCD2を参照下さい)

説 明

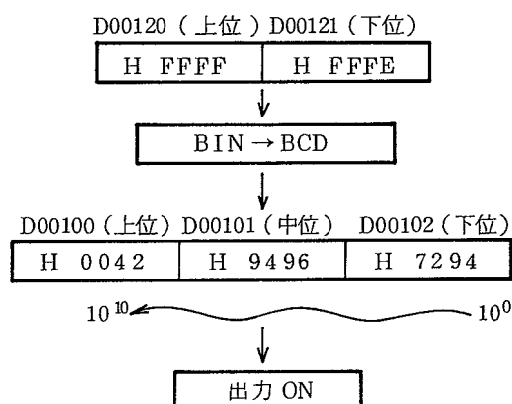
- A接点X0000がONの時、XW001の内容を4桁のBCDデータに変換してYW005に格納します。
正常実行の場合、出力をOFFにします。
- 変換するデータが、9999を越えている時は、9999でリミットをかけ演算を行います。
この時は出力をONにします。
- A接点X0000がOFFの時、演算を行わず出力をOFFにします。

BCD2		倍長 B C D 変換 (FUN052)										ステップ数 3			
表 現	条件入力 ① BCD2 ② 実行出力 H 符号なし データ表示														
	機 能	レジスタ①・①+1の倍長データを最大10桁のBCDデータに変換してレジスタ②・②+1・②+2に格納します。						条件入力	処 理						出 力
OFF								不 実 行						OFF	
ON								実 行						ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	①	演算データ (バイナリ)					○	○	○	○	○				
	②	BCD データ					○		○	○	○				

プログラム例



動 作



説 明

- A接点X0000がONの時、D00120・D00121の倍長データを最大10桁のBCDデータに変換して、D00100・D00101・D00102に格納します。
実行を行うと出力をONにします。
- A接点X0000がOFFの時は演算を行わず、出力をOFFにします。

注 記

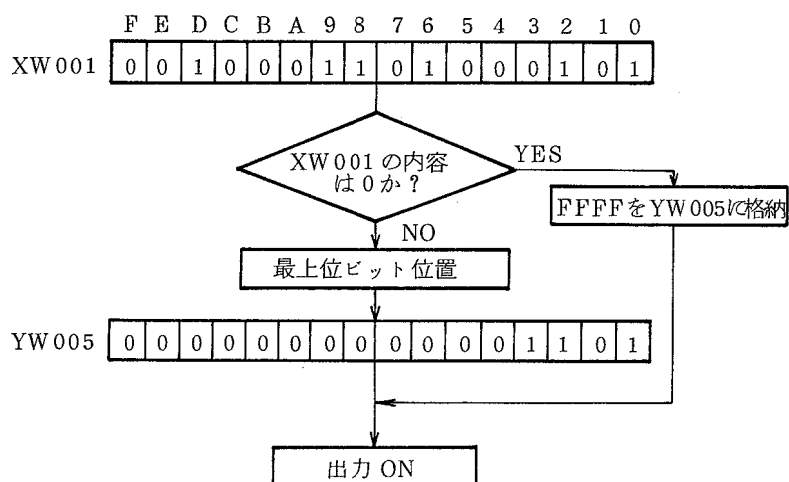
- この命令の変換範囲は0~4,294,967,295(0~FFFFFFFF)までです。
- 上位レジスタを0とすれば、単長の変換もできます。

ENC		エンコード (FUN053)													
表 現	機 能														
		ステップ数 3													
オ ペ ラ ン ド	機 能	レジスタ①の内容の最上位の ON ビット位置 N をレジスタ②に格納します。													
		条件入力		処 理										出 力	
		OFF		不 実 行										OFF	
		ON		実 行										ON	
記 号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	① エンコードデータ					○	○	○	○	○	○				
	② ビット位置データ					○		○	○	○					

プログラム例



動 作



説 明

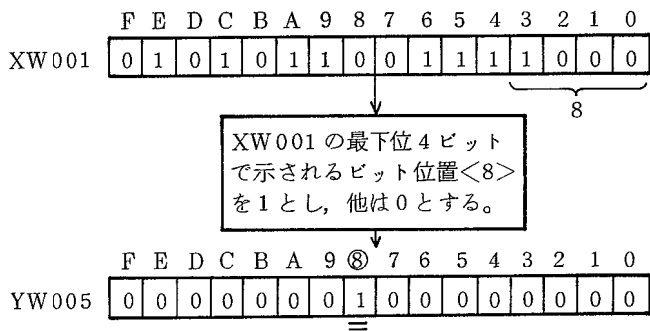
- A接点X0000がONの時、レジスタXW001の内容の最上位のONビット位置 N をレジスタYW005に格納し、出力をONにします。
- レジスタXW001の内容が0の時は、FFFFをYW005に格納し、出力をONにします。
- A接点X0000がOFFの時、エンコードは行わず、出力をOFFにします。

DEC		デ コ ー ド (FUN054)														ステップ数	
表 現	機 能	条件入力 ① DEC ② 実行出力 H H データ表示														3	
		レジスタ①最下位4ビットの内容で示すビット位置のみ1とし、他は全て0とするデータをレジスタ②に格納します。															
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値		
	①	ビット位置データ					○	○	○	○	○	○					
	②	デコードデータ					○		○	○	○						

プログラム例



動 作



説 明

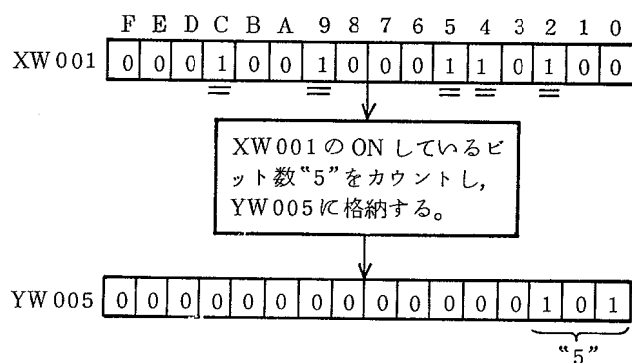
- A接点X0000がONの時、レジスタXW001の内容で最下位4ビットで示される、ビット位置<8>を1とし、他は全て0とするデータをレジスタYW005に格納し、出力をONにします。
- レジスタXW001の①データは最下位4ビットのみ有効で、他のビットは無効です。
- A接点X0000がOFFの時、デコードは行わず出力をOFFにします。

BITC		ビットカウント (FUN055)													
表 現	条件入力 ① BITC ② 実行出力 H														

プログラム例



動 作

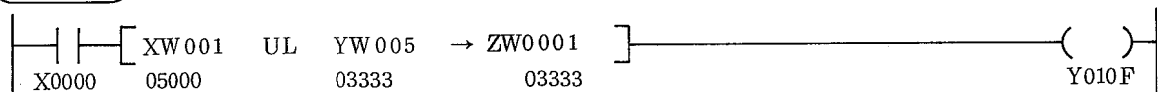


説 明

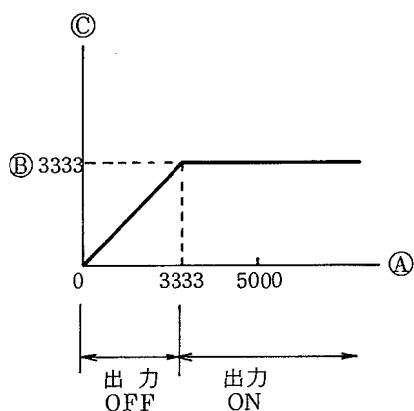
- A接点X0000がONの時、レジスタXW001の内容でON (1) しているビット数“5”をカウントし、レジスタYW005に格納します。また、この時出力をONにします。
- A接点X0000がOFFの時、ビットカウントは行わず出力をOFFにします。

UL		上 限 リ ミ ッ ト (FUN060)												ステップ数		
表 現	条件入力 ① UL ② → ③ 判定出力 符号なし データ表示															4
機 能	レジスタ①の内容を、レジスタ②の内容で 上限リミットし、レジスタ③に格納します。	条件入力	処 理												出 力	
		OFF	不 実 行												OFF	
		ON	実 行												OFF	
			①≤②の時												OFF	
			①>②の時												ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	演算データ					○	○	○	○	○	○				
	②	上限リミット値					○	○	○	○	○	○				
	③	リミット結果					○		○	○	○					

プログラム例



動 作

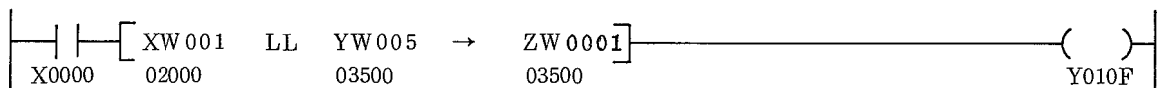


説 明

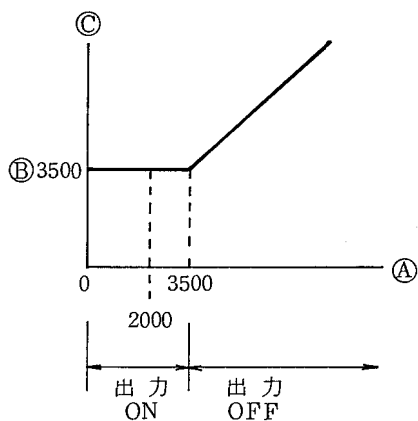
- A接点X0000がONの時、レジスタXW001の内容5000をレジスタYW005の内容3333で上限リミットした値をレジスタZW0001に格納し、5000は3333より大きいので、レジスタZW0001には3333を格納し、出力をONにします。
- XW001の内容がリミット値を越える時は出力をONにし、リミット値以下の時は出力をOFFにします。
- A接点X0000がOFFの時、上限リミットを行わず、出力をOFFにします。

LL		下 限 リ ミ ッ ト (FUN061)																			
表 現	条件入力 ① LL ② → ③ 判定出力 符号なしデータ表示																ステップ数				
																	4				
機 能	レジスタ①の内容をレジスタ②の内容で下限リミットし、レジスタ③に格納します。							条件入力		処 理							出 力				
								OFF		不 実 行							OFF				
								ON		実 行				①≥②の時			OFF				
										①<②の時				ON							
オ ペ ラ ン ド	記号	名 称						R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	演算データ										○	○	○	○	○	○	○			
	②	下限リミット値										○	○	○	○	○	○	○			
	③	リミット結果										○		○	○	○					

プログラム例

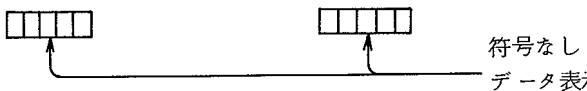


動 作

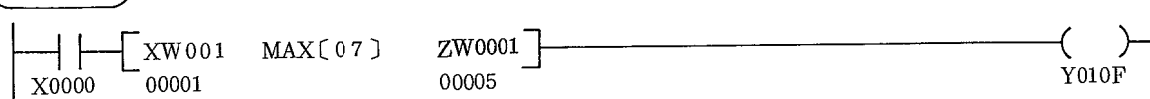


説 明

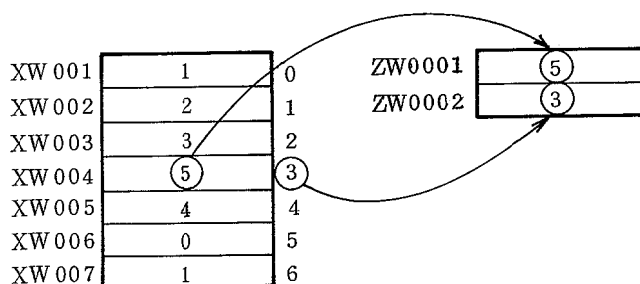
- A 接点X0000がONの時、レジスタXW001の内容2000をレジスタYW005の内容3500で下限リミットした値をレジスタZW0001に格納します。2000は3500より小さいので、レジスタZW0001には3500を格納し、出力をONにします。
- XW001の内容がリミット値より小さい時は出力をONにし、リミット値以上の時は出力をOFFにします。
- A 接点X0000がOFFの時、下限リミットを行わず出力をOFFにします。

MAX		最 大 値 (FUN062)														ステップ数	
表 現	機 能	条件入力 [① MAX [nn] ②] 実行出力 														4	
		レジスタ①から始まる nn 個のテーブルから最大値を検索しレジスタ②に格納, 又, そのテーブルポインタをレジスタ②+1 に格納します。															
オ ペ ラ ン ド	機 能	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値	
		①	テーブル先頭レジスタ					○	○	○	○	○	○				
		②	最 大 値					○		○	○	○					
		nn	テーブルサイズ													1~64	

プログラム例



動 作



説 明

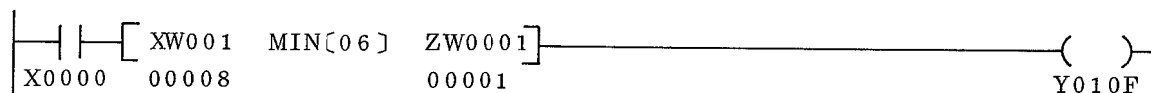
- A 接点 X0000 が ON の時, レジスタ XW001 から始まる 7 テーブル (0~6) から最大値 “5” を検索し, レジスタ ZW0001 に格納します。また, 最大値がデータ先頭番地から何番目のデータであるかを示すデータポインタ “3” を ZW0002 に格納し, 出力を ON にします。
- A 接点 X0000 が OFF の時, 演算を行わず出力を OFF にします。

注 記

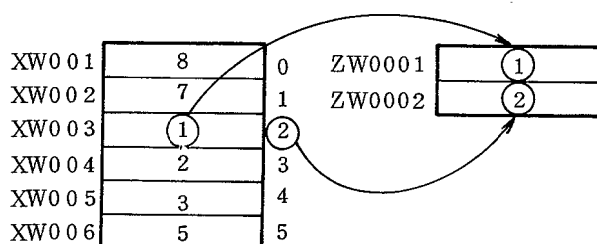
- 最大値が 2 つ以上存在する場合はポインタの若い番号を出力します。
- テーブルサイズ [nn] の指定範囲は 1 ~ 64 です。

MIN		最 小 値 (FUN063)																	
表 現	条件入力 ① MIN [nn] ② 実行出力 符号なし データ表示														ステップ数				
															4				
機 能	レジスタ①から始まる nn 個のテーブルから最小値を検索しレジスタ②に格納, 又, そのテーブルポインタをレジスタ②+1 に格納します。							条件入力	処 理							出 力			
								OFF	不 実 行							OFF			
								ON	実 行							ON			
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	①	テーブル先頭レジスタ									○	○	○	○	○	○			
	②	最 小 値									○		○	○	○				
	nn	テーブルサイズ																	1~64

プログラム例



動 作



説 明

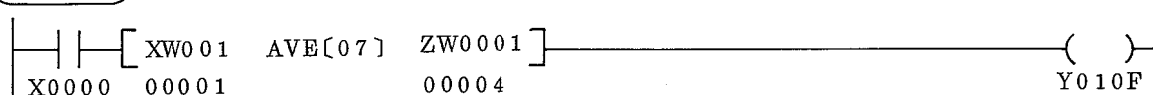
- A接点X0000がONの時, レジスタXW001 から始まる6テーブル(0~5)から最小値“1”を検索し, レジスタZW0001に格納します。また, 最小値がデータ先頭番地から何番目のデータであるかを示すデータポインタ“2”をZW0002に格納し, 出力をONにする。
- A接点X0000がOFFの時, 演算を行わず 出力をOFFにします。

注 記

- 最小値が2つ以上存在する場合はポインタの若い番号を出力します。
- テーブルサイズ[nn]の指定範囲は1~64です。

AVE		平 均 値 (FUN064)												ステップ数			
表 現	条件 入力	[① AVE [nn] ②] 実行出力												4			
		符号なし データ表示															
機 能	レジスタ①から始まる nn 個のテーブルの 平均値を求め、レジスタ②に格納します。							条件入力	処 理							出 力	
								OFF	不 実 行							OFF	
								ON	実 行							ON	
オ ペ ラ ン ド	記号	名 称		R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	テーブル先頭レジスタ						○	○	○	○	○	○				
	②	平 均 値						○		○	○	○					
	nn	テーブルサイズ															1～64

プログラム例



動 作

XW001	1	0
XW002	2	1
XW003	3	2
XW004	4	3
XW005	5	4
XW006	6	5
XW007	7	6

→ ZW0001

4

平均値

説 明

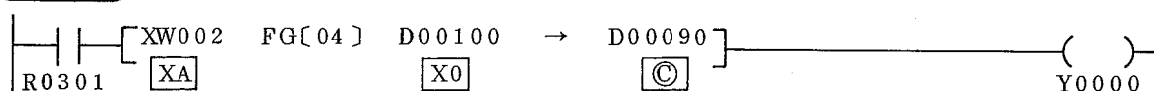
- A接点X0000がONの時、レジスタXW001から始まる7テーブル(0～6)の平均値“4”を演算し、レジスタZW0001に格納します。また、この時出力をONにします。
- A接点X0000がOFFの時、演算は行わず、出力をOFFにします。

注 記

- テーブルサイズ[nn]の指定範囲は1～64です。

FG		関 数 発 生 器														(FUN065)			
表 現	条件 入力	条件入力 [① FG [nn] ② → ③] 実行出力														ステップ数			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn] ② → ③] 実行出力														5			
		[① FG [nn]																	

プログラム例

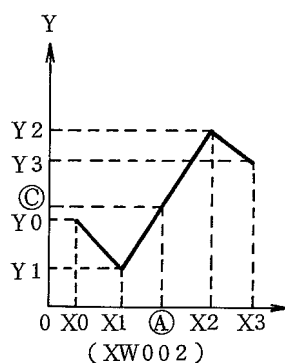


動 作

関数データテーブル

D00100	X0
D00101	X1
D00102	X2
D00103	X3
D00104	Y0
D00105	Y1
D00106	Y2
D00107	Y3

2×nn
=8ワード



説 明

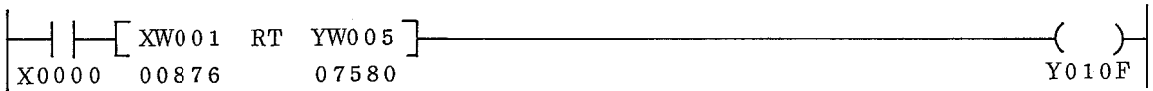
- A接点R0301がONの時、演算を実行します。
- レジスタXW002で指定される任意のX値 X_A が $X_0 < X_A < X_i$ で、 X_A に対応した関数 Y_i が $Y_{i-1} < Y_i$ の時（右上がりの直線）
レジスタD00090には
出力データ③ $= Y_{i-1} + (X_A - X_{i-1}) \cdot \frac{Y_i - Y_{i-1}}{X_i - X_{i-1}}$
 Y_i が $Y_{i-1} > Y_i$ の時（右下がりの直線）
出力データ③ $= Y_i + (X_i - X_A) \cdot \frac{Y_{i-1} - Y_i}{X_i - X_{i-1}}$
を格納します。
- $X_A \geq X_3$ の時は Y_3 の値を出力します。
- $X_A \leq X_0$ の時は Y_0 の値を出力します。

注 記

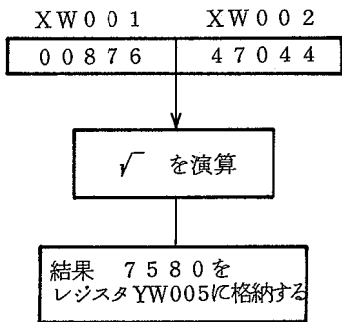
- $X_0 \leq X_1 \leq X_2 \dots \leq X_n$ を前提としているため、 $X_{i-1} > X_i$ の時は X_i 、 Y_i を無視します。
- Y は $Y_{i-1} \leq Y_i$ であっても $Y_{i-1} > Y_i$ でもかまいません。

RT		開 平 (FUN070)															
表 現	条件入力 [① RT ②] 実行出力 □□□□ □□□□ 符号なし データ表示															ステップ数	
																3	
機 能	レジスタ①・①+1の倍長データの平方根を、レジスタ②に格納します。							条件入力	処 理							出 力	
								OFF	不 実 行							OFF	
								ON	実 行							ON	
オペランド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	演算データ					○	○	○	○	○	○					
	②	平方根					○		○	○	○						

プログラム例



動 作

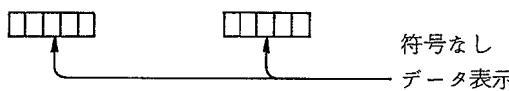


説 明

- A接点X0000がONの時、レジスタXW001・XW002の32ビット2進数(倍長データ)の平方根を求め、レジスタYW005に格納し、出力はONにします。
- A接点X0000がOFFの時は、演算を行わず、出力をOFFにします。

注 記

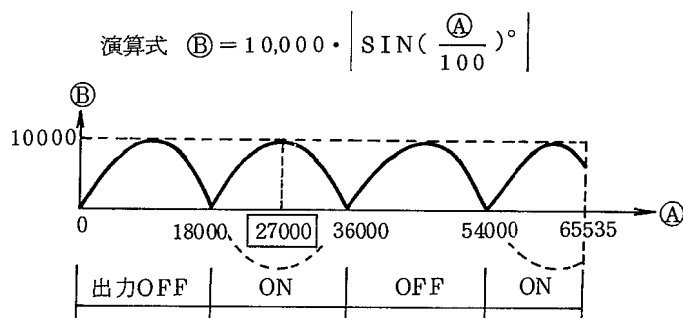
- レジスタ①は偶数、奇数レジスタを問いません。
- レジスタ①・①+1は、開平演算により変化しません。
- レジスタ①・①+1は倍長データとして扱います。

SIN		正 弦 関 数 (FUN071)																ステップ数	
表 現	機 能	条件入力 [① SIN ②] 実行出力 																3	
		データ①の内容の1/100を角度(deg)とする正弦の値の10000倍をレジスタ②に格納します。																	
オ ペ ラ ン ド	機 能	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値		
		①	演算データ					○	○	○	○	○	○	○					
		②	正弦関数値					○		○	○	○							

プログラム例



動 作



注 記

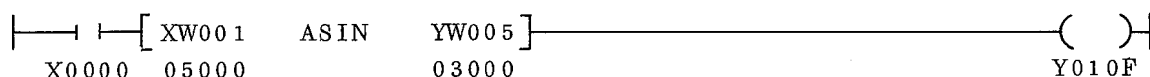
- 演算データ①の範囲は0～65535です。
- 相対誤差は±0.8%以下です。

説 明

- A接点X0000がONの時、レジスタXW001の内容27000のSINを左記演算式によって求め、レジスタYW005に格納します。
演算式は絶対値をとっているため負にはなりません、 $\sin \left(\frac{27000}{100} \right)$ は-1負なので出力はONにします。
- $\sin \left(\frac{①}{100} \right)$ が正の時、出力はOFFにします。
- A接点X0000がOFFの時は、演算を行わず出力をOFFにします。

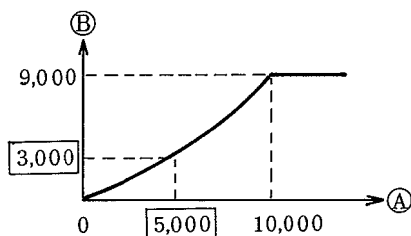
ASIN		逆 正 弦 関 数 (FUN072)												ステップ数		
表 現	機 能	条件入力 { ① ASIN ② } 実行出力														3
		↑ 符号なし データ表示 ↑														
機 能	データ①の内容の 1/10000 の逆正弦値 (deg) を 100 倍してレジスタ②に格納します。	条件入力	処 理												出 力	
		OFF	不 実 行												OFF	
		ON	実 行												ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	演算データ					○	○	○	○	○	○				
	②	逆正弦関数値					○		○	○	○					

プログラム例



動 作

$$\text{演算式 } \textcircled{B} = 100 \cdot \sin^{-1} \left(\frac{\textcircled{A}}{10,000} \right)$$



説 明

- A接点X0000がONの時、レジスタXW001の内容5000のASINを左記演算式によって求め、レジスタYW005に格納し、出力をONにします。
- A接点X0000がOFFの時は、演算を行わず、出力をOFFにします。

注 記

- 演算データ①の範囲は0～10,000です。従って、演算結果②も0～9,000となります。
- ① > 10,000のとき演算結果②は9,000でリミットします。
- 相対誤差は±1%以下です。

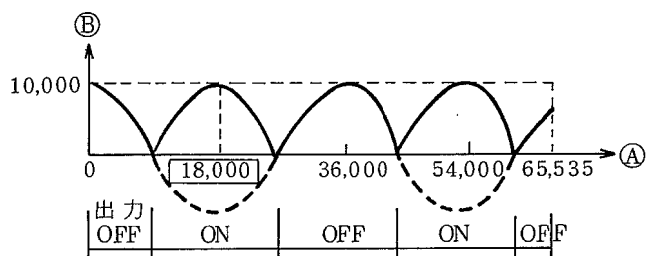
COS		余 弦 関 数 (FUN073)														
表 現	条件 入力 ①COS② 実行出力 符号なし データ表示	ステップ数 3														
機 能	データ①の内容の1/100を角度(deg)と余弦の値の1000倍をレジスタ②に格納します。						条件入力		処 理						出 力	
							OFF		不 実 行						OFF	
							ON		実 行		結果が負の時		ON		結果が正の時	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	演算データ					○	○	○	○	○	○				
	②	余弦関数値					○		○	○	○					

プログラム例



動 作

$$\text{演算式 } \textcircled{2} = 10,000 \cdot \left| \cos \left(\frac{\textcircled{1}}{100} \right)^\circ \right|$$



注 記

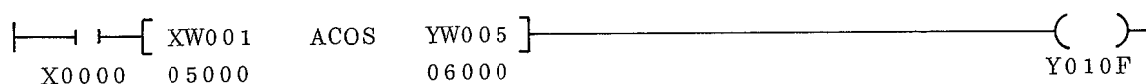
- 演算データ①の範囲は0～65535です。
- 相対誤差は±0.8%以下です。

説 明

- A接点X0000がONの時、レジスタXW001の内容18000のCOSを左記演算式によって求め、レジスタYW005に格納します。
演算式は絶対値をとっているため負にはなりません。が、 $\cos \left(\frac{18000}{100} \right)$ は-1負なので出力はONにします。
- $\cos \left(\frac{\textcircled{1}}{100} \right)$ が正の時、出力はOFFにします。
- A接点X0000がOFFの時は、演算を行わず、出力をOFFにします。

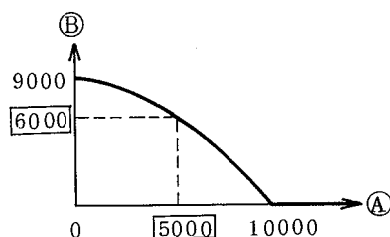
[illegible]

プログラム例



動作

演算式 ③ = 100 · COS⁻¹ ($\frac{\textcircled{A}}{10000}$)



說明

- A 接点X0000がONの時，レジスタXW001の内容5000のACOSを左記演算式によって求め，レジスタYW005に格納し，出力をONにします。
- A 接点X0000がOFFの時は，演算を行わず，出力をOFFにします。

注 記

- 演算データ④の範囲は0～10000です。従って、演算結果⑤も0～9000となります。
- 演算データ④>10000の時は、演算結果⑤は0でリミットします。
- 相対誤差は±1%以下です。

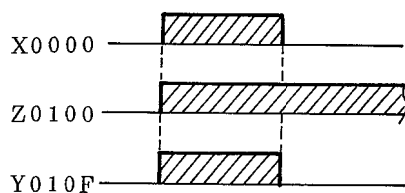
SET	デバイスセット (FUN080)
-----	------------------

表 現	条件入力 [SET ①] 実行出力														ステップ数					
	ステータス表示														2					
機 能	デバイス①をONにセットします。						条件入力		処 理						出 力					
							OFF		不 実 行						OFF					
							ON		実 行						ON					
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	セ ッ ト デ バ イ ス					○		○	○										

プログラム例



動 作



説 明

- A接点X0000がONの時、デバイスZ0100をONにセットし、出力をONにします。X0000がOFFにしても、セット状態を継続します。
- A接点X0000がOFFの時、デバイスセットは行わず、出力をOFFにします。

注 記

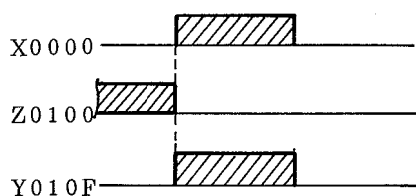
- デバイスセットはデバイスリセットとペアで使うと便利です。

RST		デバイスリセット (FUN081)															
表 現	条件入力 $\left[\begin{array}{c} \text{RST} \quad \textcircled{A} \end{array} \right]$ 実行出力  ステータス表示																ステップ数
																	2
機 能	デバイス $\textcircled{A}$ をOFFにリセットします。							条件入力	処 理							出 力	
								OFF	不 実 行							OFF	
								ON	実 行							ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	$\textcircled{A}$	リセットデバイス	○		○	○											

プログラム例



動 作



説 明

- A接点X0000がONの時、デバイスZ0100をOFFにリセットし、出力をONにします。
X0000がOFFしても、リセット状態を継続します。
- A接点X0000がOFFの時、デバイスリセットは行わず出力をOFFにします。

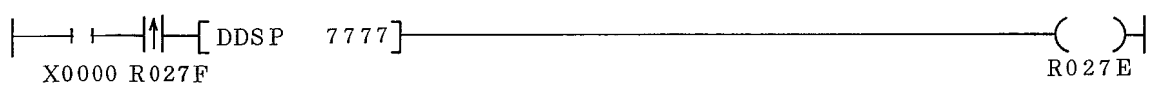
注 記

- デバイスリセットはデバイスセットとペアで使うと便利です。

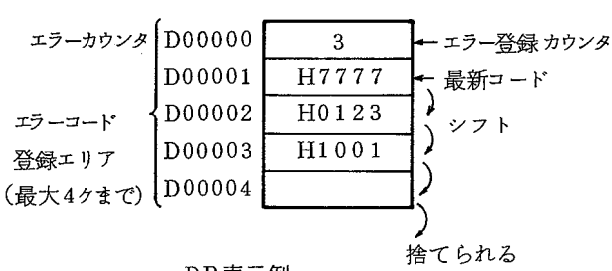
DDSP	診 断 表 示 (FUN090)
------	--------------------

表	故障条件 入力	DDSP ㉔	}	実行出力	ステップ数																
					2																
現																					
機 能	エラー登録カウンタ(D00000)が“0”もしくは前回と異なるエラーコード(㉔≠D00001)が発生したとき、㉔のコードをBCD変換してデータレジスタD00001に格納し、特殊コイルR992EをONにします。 最大4つのエラーコードが登録できます。						条件入力	処 理								出 力					
							OFF	不 実 行								OFF					
							ON	D00000≠0かつ㉔=D00001の時不実行								OFF					
								D00000=0または㉔≠D00001の時実行								ON					
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	㉔	エラーコードデータ																		1～9999 (BCD)	

プログラム例



動 作



DP表示例
C06 1 DIAG. 7777

説 明

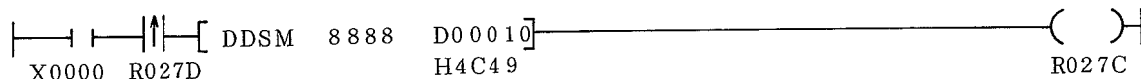
- データレジスタD00000～D00004をリザーブしておく必要があります。
- A接点X0000がONの時、データH7777をBCD変換後コード化し、エラーカウンタ(D00000)が0もしくは入力データ≠D00001の時、エラーコードH7777をD00001に登録し、特殊コイルR992E及び、出力をONとし、エラー登録カウンタの更新とエラーコードのシフトを行います。エラーカウンタ≠0かつ入力データ④=D00001(最新コードと今回発生したエラーコードが一致している時)は不実行で出力をOFFにします。

注 記

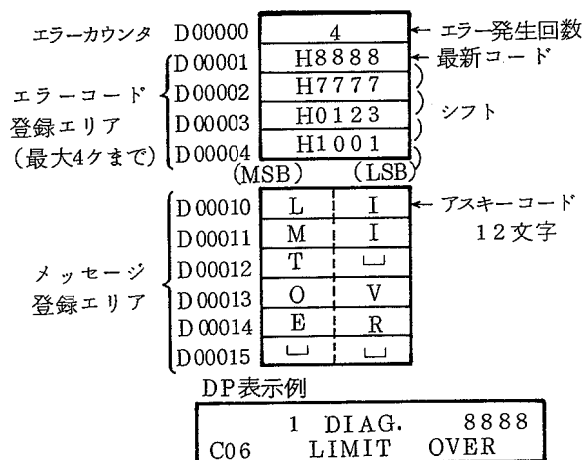
- 本命令の前に↑↓等の命令を入れないとスキャンごと実行してしまう場合もありますので、エラー条件(プログラム例ではX0000)の立ち上り、もしくは立ち下りでトリガさせます。
- グラフィックプログラムまたはデータアクセスパネルはエラーコードを表示できます。また、特殊コイルR992Eを外部出力すれば警報出力に使用できます。
- エラーカウンタ(D00000)及び特殊コイルR992Eのクリアはユーザーがプログラムまたはプログラムのデータ設定で行います。
- D00000～D00004までをプログラム中で使用すると、グラフィックプログラムのシステム情報(ユーザ診断命令のメッセージテーブル)に無意味な表示がでることがあります。

DDSM		診断表示・メッセージ付 (FUN091)														ステップ数	
表 現	故障条件 入力	DDSM ① ② 実行出力														3	
		H データ表示															
機 能	DDSP機能に、メッセージ出力機能を加えたものです。														条件入力	処 理	出 力
															OFF	不 実 行	OFF
															ON	D00000≠0かつ①=D00001の時不実行 D00000=0または①=D00001の時実行	OFF ON
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値		
	①	エラーコードデータ													1~9999 (BCD)		
	②	メッセージデータ先頭レジスタ									○						

プログラム例



動 作



説 明

- 下記以外は DDSP 命令と同じです。
 - メッセージデータ先頭レジスタ②に格納されているアスキーコードをエラーコードといっしょにプログラミング装置 GP または DP に表示することができます。
- 左記例では、エラーコードデータ 8888 に対応したメッセージ「LIMIT OVER」を D00010 から D00015 に次の様なアスキーコードで登録し、プログラマに表示しています。
- | | |
|--------------------|----------|
| D00010←H4C49[L I] | アスキーコードは |
| D00011←H4D49[M, I] | プログラム (K |
| D00012←H5420[T,] | →W) またはプ |
| D00013←H4F56[O, V] | ログラムのデー |
| D00014←H4552[E, R] | タ設定で行いま |
| D00015←H2020[,] | す。 |

注 記

- この命令がプログラム中に複数ある場合は、それぞれのメッセージデータを 12 文字、メッセージ登録エリアに登録しておく必要があります。
- DDSP 命令参照。

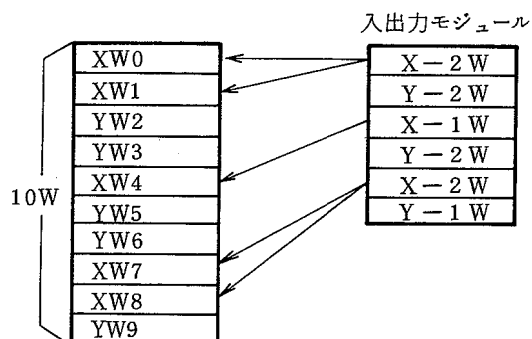
IN	直接入力 (FUN096)
----	---------------

[illegible]

プログラム例



動作

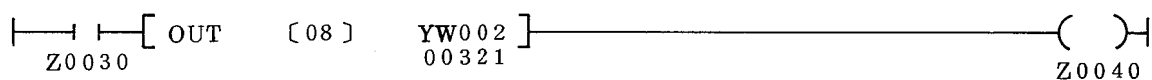


說明

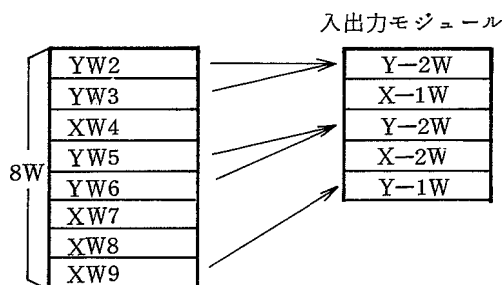
- A 接点 Z0010 が ON の時 XW000 から 10 ワード分のエリアに対応する I/O モジュールから入力カードの所のデータを入力して、出力を ON にします。10 ワードの内 XW 指定のエリアのみ行い、YW 指定は無視する。
- A 接点 Z0010 が OFF の時、入力を行わず、出力を OFF にします。

OUT		直 接 出 力 (FUN097)														
表 現	条件入力 [OUT [nn] ①] 実行出力														ステップ数	
	データ表示														3	
機 能	レジスタ①から nn ワード分のデータを、 対応する I/O モジュールへ出力します。						条件入力		処 理						出 力	
							OFF		不 実 行						OFF	
							ON		実 行						ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値	
	nn	テーブルサイズ													1 ~ 16	
	①	出力先頭レジスタ							○							

プログラム例



動 作



説 明

- A 接点 Z0030 が ON の時，YW002 から 8 ワード分のエリアに対応する I/O モジュールから出力カードの所へデータを出力して出力を ON にします。8 ワードの内 YW 指定のエリアのみ行い，XW 指定は無視する。
- A 接点 Z0030 が OFF の時，データの出力は行わず，出力を OFF にします。

READ	ASCII リード	(FUN098)
------	-----------	----------

表 現	条件入力 — [CH. ① READ [nnn] ②] —→ ③ — 完了出力														ステップ数	
	H データ表示														5	
機 能	チャンネル①で示す ASCII モジュールに対し、③で示す番号の BASIC プログラムを実行させ、③+1 で示すサイズのデータを③+2 で示すレジスタ以後に読み込みます。プログラムの完了ステータスは②で示されます。						条件入力				処 理				出 力	
							OFF		不実行						OFF	
							ON		実行		実行中				OFF	
										完了(エラー完了も含む)				ON		
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	チャンネル No														1～8
	nnn	テーブルサイズ														2～502
	②	完了ステータスレジスタ					○		○	○						
	③	パラメータ先頭レジスタ					○		○	○	○					

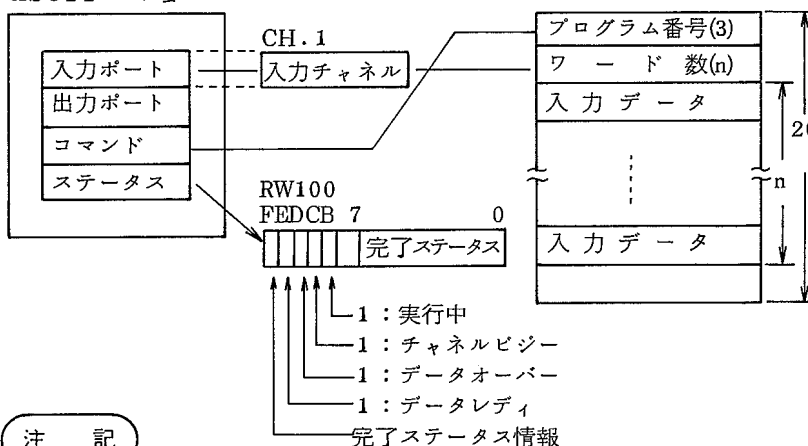
プログラム例



動 作

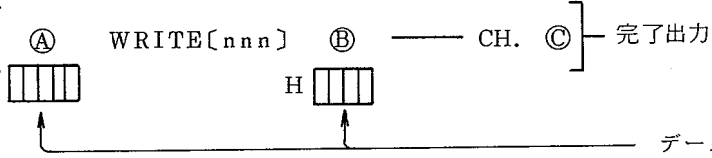
説 明

ASCII モジュール

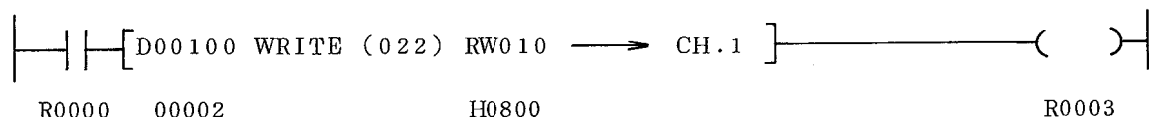


注 記

- 条件入力は ASCII モジュールが処理を実行している間、保持しておいて下さい。
- チャンネル No, テーブルサイズ正しくない場合には、ASCII モジュールを起動することなくエラー出力します。
- ワード数がテーブルサイズを越す場合は、テーブルの範囲内だけをリードしてエラー出力します。
- 1 台の ASCII モジュールに複数の READ 命令が起動をかけた場合、2 番目以後に起動をかけた READ ではチャンネルビジーのステータスがセットされます。このまま入力を ON にし続けることにより実行中の READ 命令が完了次第ビジーステータスが解除され命令が実行されます。

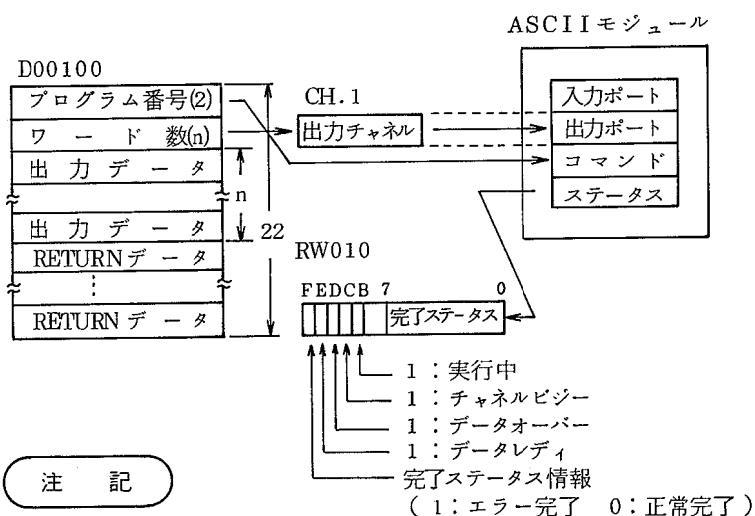
WRITE		ASC II ライト														(FUN099)				
表 現	条件 入力															ステップ数				
		データ表示														5				
機 能	①+2レジスタからの①+1サイズ分の出力データをチャンネル③で示すASCIIモジュールに出力した後①で示す番号のBASICプログラムを実行させます。プログラムの完了ステータスは②で示されます。						条件入力		処 理								出 力			
							OFF		不実行								OFF			
							ON		実行		実行中						OFF			
								完了(エラー完了も含む)						ON						
オ ペ ラ ン ド	記号 名 称						R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	① パラメータ先頭レジスタ										○		○	○	○					
	nn テーブルサイズ																		2~502	
	② 完了ステータスレジスタ										○		○	○						
	③ チャンネルNo																		1~8	

プログラム例



動 作

説 明



注 記

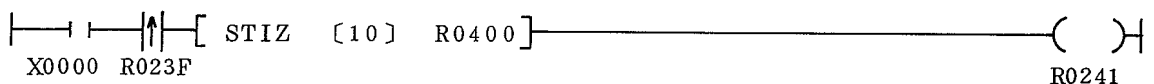
- 出力データのサイズnもASCIIモジュールへ転送されます。
- この命令を実行した時、ASCIIモジュールよりRETURNデータがある場合にはそのデータを読み込み、出力データの後に格納します。また、RETURNデータのサイズは①+1レジスタに格納します。

- A接点R0000がONの時、CH.1のASCIIモジュールに対しD00101の内容で示されるサイズのD00102を先頭とするテーブルのデータを出力します。
- そしてD00100の内容で指定される番号のBASICプログラムを実行させます。
- 完了ステータスはRW010に示されます。

STIZ	ステップシーケンス・イニシャライズ (FUN100)
------	----------------------------

表 現	条件入力 $\left[\begin{array}{c} \text{STIZ} \quad [nn] \quad \textcircled{A} \end{array} \right]$ 実行出力															ステップ数
																3
機 能	デバイス $\textcircled{A}$ をONにして、それ以降の $\textcircled{A} + nn - 1$ までのデバイスをイニシャライズ(OFF)します。							条件入力	処 理							出 力
								OFF	不 実 行							OFF
								ON	実 行							ON
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	$\textcircled{A}$	ステップシーケンス先頭デバイス	○													
	nn	ステップシーケンスデバイスサイズ														1 ~ 64

プログラム例



動 作

R0400	ON	0
R0401	OFF	1
R0402	OFF	2
⋮	⋮	⋮
R0408	OFF	8
R0409	OFF	9 (nn-1)

説 明

- ステップシーケンスを開始する前にこのイニシャライズを行います。
- A接点X0000の立上りで、ステップシーケンス先頭デバイスR0400をONとし、それ以降のR0409までのデバイスをすべてOFFにします。
([nn]=10なのでR0400+10-1=R0409)
この時出力をONにします。
- A接点X0000がOFFの時、イニシャライズは行わず、出力をOFFにします。

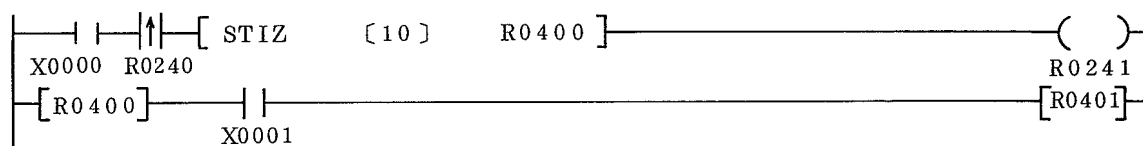
注 記

- この命令はFUN101 (ステップシーケンス入力), FUN102 (ステップシーケンス出力) と併用します。
- FUN101 (ステップシーケンス入力), FUN102 (ステップシーケンス出力) を参照して下さい。

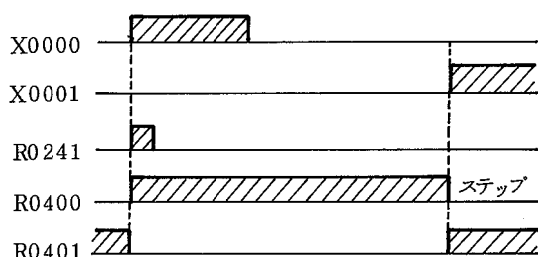
STIN	ステップシーケンス・入力 (FUN101)
------	-----------------------

表 現	条件入力 [㉔] 実行出力														ステップ数
															2
機 能	入力がONで、デバイス㉔がONの時、出力をONにします。						条件入力	処 理						出 力	
							OFF	不 実 行						OFF	
							ON	㉔=ON						ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	㉔	ステップシーケンスデバイス	○												

プログラム例



動 作



説 明

- X0000の立ち上りでFUN100ステップシーケンス・イニシャライズを実行し、R0400をONとしてR0401～R0409をOFFにします。
- X0001がOFFの間は、R0400がONでR0401がOFFの状態がつづき、X0001がONになると、R0400をOFFにしてR0401がONになります。

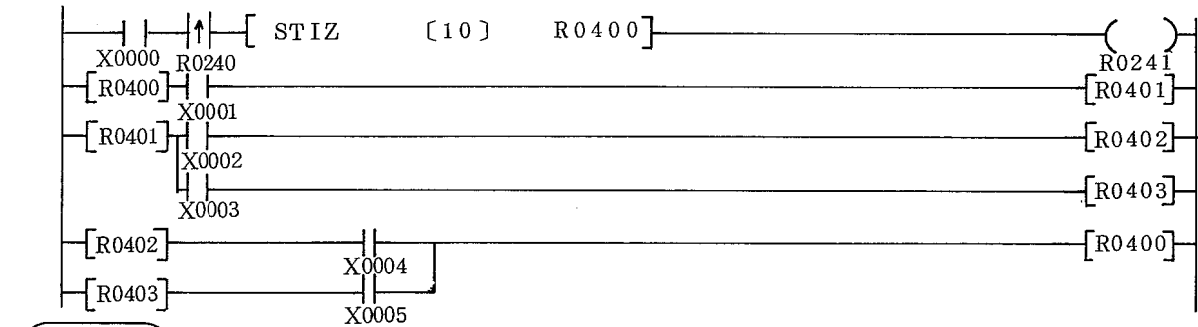
注 記

- この命令はFUN100(ステップシーケンス・イニシャライズ) FUN102(ステップシーケンス出力)と併用します。
- 入力の形は、 $\begin{bmatrix} \text{H} \\ \text{H} \end{bmatrix}$ または $\begin{bmatrix} \text{H} \\ \text{H} \end{bmatrix}$ などを直並列に最大14コまで接続できます。
- FUN100(ステップシーケンス・イニシャライズ)、FUN102(ステップシーケンス出力)を参照して下さい。

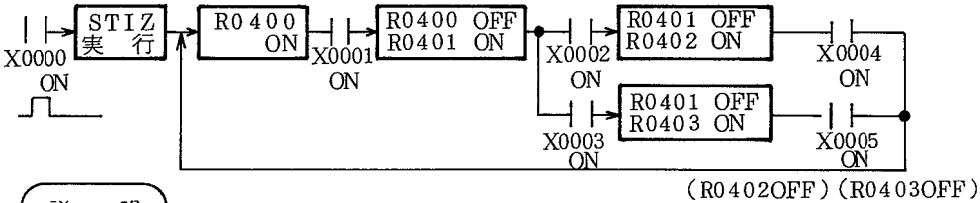
STOT	ステップシーケンス・出力 (FUN102)
------	-----------------------

表 現	条件 入力 Ⓐ															ステップ数		
	2																	
機 能	入力がONの時，同一回路上のステップシーケンス入力のデバイスを全てクリアし，デバイスⒶをONにします。							条件入力	処 理							出 力		
								OFF	不 実 行									
								ON	実 行									
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値		
	Ⓐ	ステップシーケンスデバイス	○															

プログラム例



動作



説明

- この命令の入力がONすると，同一回路上のステップシーケンス入力のデバイスを全てOFFにクリアし，デバイス①をONにします。
- 上記プログラム例を実行すると，R0400からR0403まで段階的に動作が移動します。

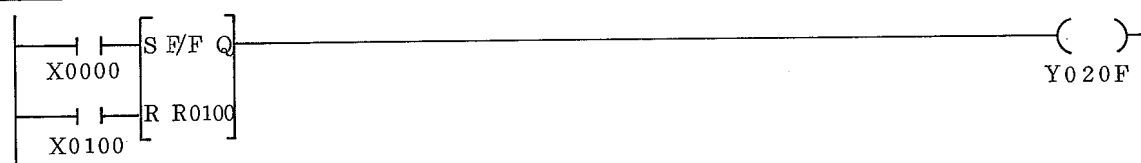
注 記

- この命令は FUN100 (ステップシーケンス・イニシャライズ), FUN101 (ステップシーケンス入力) と併用します。
- この命令をプログラム例の様に並列で使用してもかまいません。
- FUN100 (ステップシーケンス・イニシャライズ), FUN101 (ステップシーケンス入力) を参照ください。

F/F	フリップフロップ (FUN110)
-----	-------------------

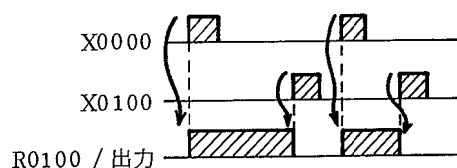
表 現	セット入力 $\left[\begin{array}{c} S \\ F/F \\ Q \end{array} \right]$ フリップフロップ出力 リセット入力 $\left[\begin{array}{c} R \\ \textcircled{A} \end{array} \right]$															ステップ数
																2
機 能	セット入力がONの時、デバイス $\textcircled{A}$ をON にセットし、リセット入力がONの時、デ バイス $\textcircled{A}$ をOFFにリセットします。リセ ット優先のF/Fです。					条件入力		処 理							出 力	
						—		— (下記参照)							デバイス $\textcircled{A}$ の出力状態 と同じです。	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	$\textcircled{A}$	フリップフロップデバイス	○		○	○										

プログラム例



動 作

セット入力S X0000の状態	リセット入力R X0100の状態	R0100 / 出力Q
OFF	OFF	前の状態
OFF	ON	OFF
ON	OFF	ON
ON	ON	OFF



注 記

- SFCの遷移部にはプログラムできません。

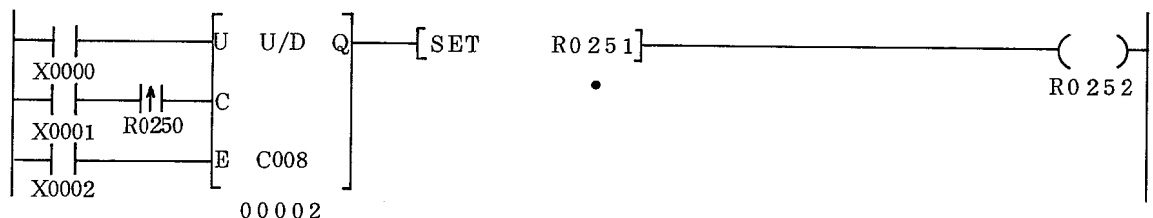
説 明

- セット入力X0000がON、リセット入力X0100がOFFの時のみ、R0100及び出力がONします。リセット入力X0100がONの時は、セット入力X0000の状態にかかわらず、R0100及び出力をOFFにします。
- セット入力、リセット入力ともOFFの時はR0100及び出力は前の状態のままです。

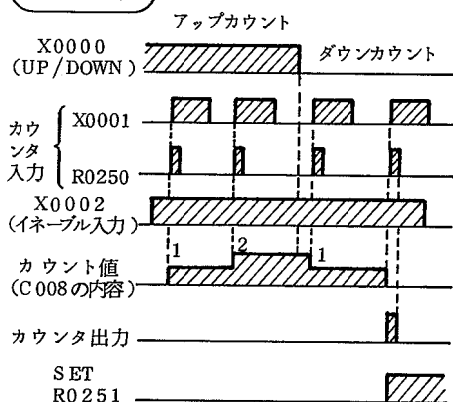
U/D	アップダウンカウンタ (FUN111)
-----	---------------------

表 現	UP/DOWN 選択入力 U U/D Q カウンタ 入力 C イネーブル 入力 E ④ 符号なし データ表示														ステップ数					
															2					
機 能	イネーブル入力がONの時、カウンタ入力CのOFF→ONへの変化回数をカウントし、カウント値がリミットに達すると出力をONにします。UP/DOWN選択によりカウント方向を選択可能です。							条件入力		処 理						出 力				
								E=OFF		カウント不実行（カウント値クリア）						OFF				
								E=ON		カウント値がリミット値に達していない時 カウント値＝リミット値、C＝ON						OFF ON				
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	④	カウンタレジスタ															○			

プログラム例



動 作



注 記

- カウンタ入力は微分接点 $\uparrow\downarrow$ を入れないとスキャン毎にカウントしてしまいます。

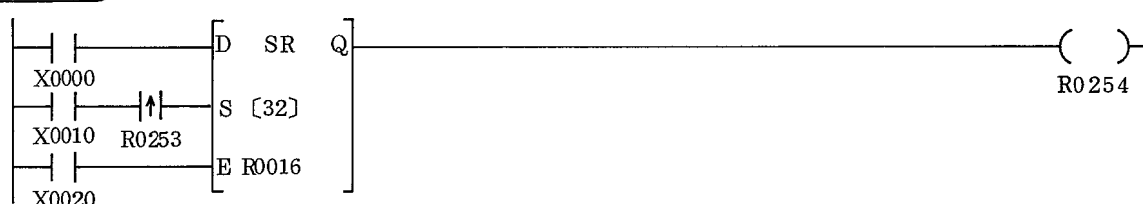
説 明

- A接点X0000がONの時 (U=ON) アップカウント, OFFの時 (U=OFF) ダウンカウントとなります。
- イネーブル入力 (A接点X0002) がONの時, アップまたはダウンカウントを実行し, OFFの時はカウント値をクリアし, 出力をOFFにします。
また, イネーブル入力がONでもカウンタがOFFしている場合はカウントを実行せず, 出力をOFFにします。
- アップカウントの場合はリミット値65535に達すると出力をONします。ダウンカウントの場合リミット値0で出力をONにします。
- 左記例の場合, ダウンカウントでリミット値0に達した時のみカウンタ出力がONとなります。また, 出力がONするのはイネーブル入力がONでカウンタ入力ONの時のみです。

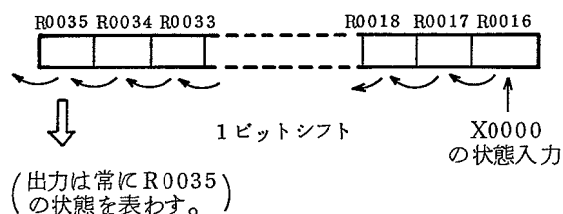
SR	シフトレジスタ (FUN112)
----	------------------

表 現	データ入力	D	SR	Q	シフトレジスタ出力														ステップ数
	シフトレジスタ入力	S	[nn]														3		
機 能	イネーブル入力	E	Ⓐ																
	イネーブル入力 が ON の時、デバイス Ⓐ を 1 ビットシフトします。						条件入力		処 理								出 力		
							E=OFF		対象となるデバイスを全て クリアします。								出力は常に、 デバイス (Ⓐ+ nn-1 の状態 を表わす。		
							E=ON		S=ON シフト実行 S=OFF 不実行										
オ ペ ラ ン ド	記号	名 称				R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	Ⓐ	先頭デバイス				○		○	○										
	nn	ビット長																	1〜64

プログラム例



動 作



注 記

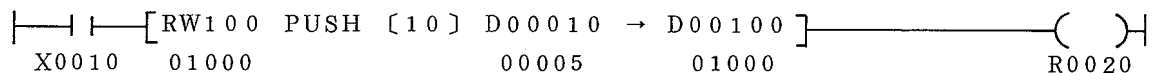
- シフトレジスタ入力には微分接点 $\uparrow$ を入れないとスキャン毎にシフトしています。
- ビット長 nn の範囲は 1~64 です。

説 明

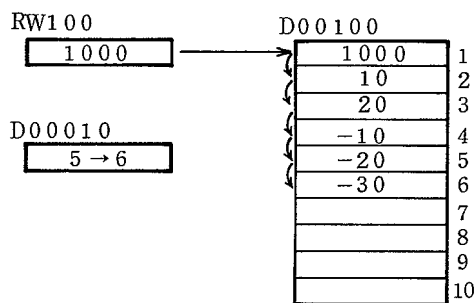
- イネーブル入力(A 接点 X0020)が OFF の時対象デバイス R0016~R0035 を全てクリアします。
- イネーブル入力(A)が ON で、シフトレジスタ入力(S)が OFF の時は実行しません。
- イネーブル入力(A)が ON で、シフトレジスタ入力(S)が ON の時、その時のデータ入力(X0000 の状態)を R0016 に入力し、1 ビットシフトします。
シフトレジスタ入力(S)がパルス入力の場合は、パルスの立上り毎に 1 ビットシフトすることになります。
- 出力は常に(A)+ nn-1 (R0035) の状態を表わします。

[illegible]

（プログラム例）



動作

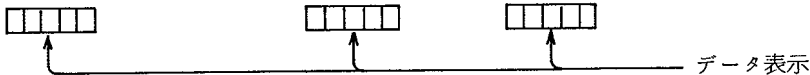


記 注

- テーブルサイズは、最大64ワード（レジスタ）までです。
($1 \leq nn \leq 64$)
- POPL 命令または POPF 命令と組合せて使
用します。

說明

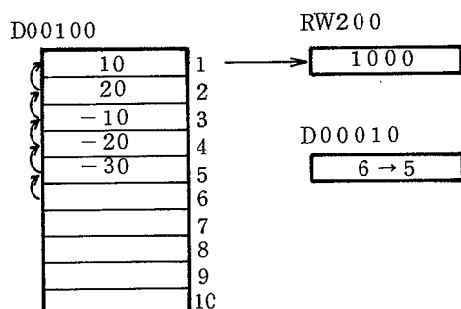
- A接点X0010がONの時、レジスタRW100の内容をレジスタD00100を先頭とするサイズ10Wのスタックの先頭にプッシュします。
以前に入っていたデータは、1つずつ下にシフトされ、データ総数は、インクリメントされます。
- プッシュの結果、データ総数（レジスタD00010の内容）がスタックサイズに達すると出力をONにします。
- スタック・フルの状態では、プッシュを実行するとスタックの底のデータを捨てて、シフトダウンした後新データをプッシュします。データ総数はインクリメントせず出力をONにします。
- データ総数レジスタD00010がスタックサイズ範囲外の場合は、（上記プログラムでは11以上の場合）プッシュは行わず出力をONにします。
- A接点X0010がOFFの時、プッシュは行わず出力をOFFにします。

POPL		ポ プ プ ・ ラ ス ト (FUN201)														ステップ数	
表 現	条件 入力	条件 ① POPL [nn] ② → ③ スタックエンブティ出力														5	
																	
機 能	レジスタ①から始まるサイズ[nn]のスタックから、最後にプッシュされたデータをポップし③に格納し、②の内容を1デクリメントします。							条件入力		処 理						出 力	
								OFF		不 実 行						OFF	
								ON		データ有り時, 実行		スタック空の時		データ有り時		ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	スタック先頭レジスタ					○		○	○	○						
	nn	スタックサイズ														1～64	
	②	データ総数レジスタ					○		○	○	○						
	③	格納先レジスタ					○		○	○	○						

プログラム例

$\overline{X0020} \quad 01000 \quad [D00100 \text{ POPL } [10] \quad D00010 \rightarrow RW200] \quad R0030$

動 作



注 記

- テーブルサイズは最大64ワード(レジスタ)までです。
($1 \leq nn \leq 64$)
- PUSH命令と組合せて使用します。

説 明

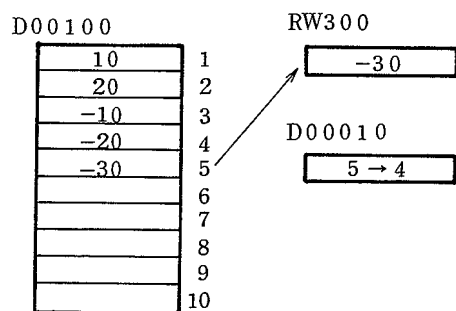
- A接点X0020がONの時レジスタD00100を先頭とするスタックから一番最後にプッシュされたデータ、つまり、スタックの先頭データをレジスタRW200に格納します。データ総数レジスタD00010の内容をデクリメントして、残りのデータをシフトアップし、出力をOFFにします。
- ポップを実行したあとデータ総数が0になったら出力をONにします。
- スタック内にデータが無い状態でポップを実行するとポップは行わず出力をONにします。
- データ総数レジスタD00010がスタックサイズ範囲外の時は(上記プログラムでは11以上の場合)ポップは行わず出力をONにします。
- A接点X0020がOFFの時ポップは行わず出力をOFFにします。

[illegible]

プログラム例

$\left| \begin{array}{c} \text{X0030} \\ \text{00010} \end{array} \right| \left[\begin{array}{c} \text{D00100 POPF [10] D00010} \\ \text{00005 -00030} \end{array} \right] \rightarrow \text{RW300} \left(\begin{array}{c} \text{R0040} \end{array} \right)$

動作



記 注

- テーブルサイズは最大 64 ワード (レジスタ) までです。
($1 \leq nn \leq 64$)
- PUSH 命令と組合せて使用します。

說明

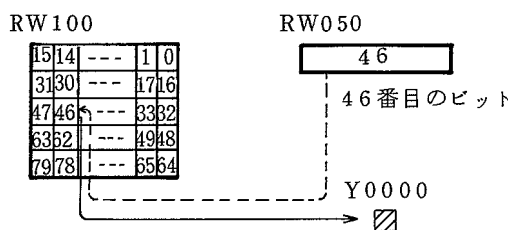
- A 接点 X0030 が ON の時、レジスタ D00100 を先頭とするスタックから一番最初にプッシュされたデータをレジスタ RW300 に格納します。データ総数レジスタ D00010 の内容をデクリメントして、出力を OFF にします。
- ポップを実行したあと、データ総数が 0 になったら出力を ON にします。
- スタック内にデータが無い状態でポップを実行すると、ポップは行わず出力を ON にします。
- データ総数レジスタ D00010 がスタックサイズ範囲外の時は、（上記プログラムでは 11 以上の場合）ポップは行わず出力を ON にします。
- A 接点 X0030 が OFF の時ポップは行わず出力を OFF にします。

T→B		テーブル→ビット転送 (FUN203)												ステップ数		
表 現	条件入力 ① T→B [nn] ② → ③ エラー出力														5	
	H ← ステータス表示 ← データ表示															
機 能	①レジスタを先頭とするテーブルの、先頭から②番目のビットを、デバイス③に格納します。						条件入力		処 理						出 力	
							OFF		不 実 行						OFF	
							ON		正常実行 ポインタがサイズオーバの時						OFF ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値	
	①	テーブル先頭レジスタ					○	○	○	○	○					
	nn	テーブルサイズ (レジスタ数)													1~64	
	②	ビットポインタ					○	○	○	○	○					
	③	結果格納デバイス	○			○	○									

プログラム例

$\begin{array}{|c|c|} \hline \text{X0100} & \text{H1234} \\ \hline \end{array} \begin{array}{|c|c|} \hline \text{RW100} & \text{T→B} \\ \hline \end{array} \begin{array}{|c|c|} \hline [5] & \text{RW050} \\ \hline \end{array} \begin{array}{|c|c|} \hline \text{00046} & \text{→ Y0000} \\ \hline \end{array} \begin{array}{|c|c|} \hline \text{●} & \text{R0100} \\ \hline \end{array}$

動 作



説 明

- A接点X0100がONの時、レジスタRW100を先頭とするサイズ5のテーブルの46番目 (ポインタRW050の内容が46なので) のビットをY0000に格納します。
- ビットポインタRW050の内容が、テーブル範囲外の場合は、(上記プログラムでは80以上の場合) データの格納は行わず出力をONにします。
- A接点X0100がOFFの時、ビット転送を行わず出力をOFFにします。

注 記

- テーブルサイズは、最大64ワード (レジスタ) までです。
($1 \leq nn \leq 64$)

B→T		ビット→テーブル転送 (FUN204)																		
表 現	条件 入力	① B → T [nn] ② → ③ } エラー出力												ステップ数						
		ステータス表示 H データ表示												5						
機 能	デバイス①の内容を，レジスタ③を先頭とするテーブルの②番目のビットに格納します。						条件入力	処 理								出 力				
							OFF	不 実 行								OFF				
							ON	正常実行 ポインタがサイズオーバーの時								OFF ON				
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	デバイスポインタ					○	○	○	○										
	nn	テーブルサイズ (レジスタ数)																		1～64
	②	ビットポインタ									○	○	○	○	○					
	③	テーブル先頭レジスタ									○		○	○	○					

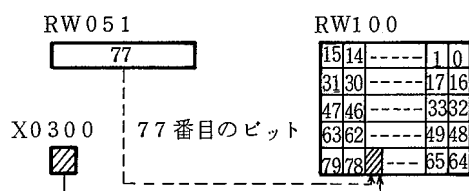
プログラム例

```

┌──┴──┐ ┌──┴──┐ [X0300 B→T [5] RW051 → RW100 ] ┌──┴──┐
X0200      00077      H1234      R0200

```

動 作



説 明

- A接点X0200がONの時レジスタRW100を先頭とするサイズ5のテーブルの77番目 (ポインタRW051の内容が77なので) のビットにX0300の内容を格納します。
- ビットポインタRW051の内容がテーブル範囲外の場合は (上記プログラムでは80以上の場合) データの格納は行わず出力をONにします。
- A接点X0200がOFFの時，ビット転送を行わず，出力をOFFにします。

注 記

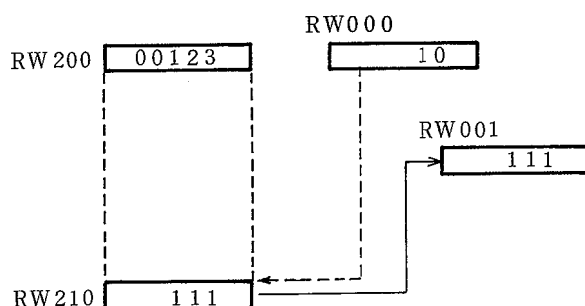
- テーブルサイズは最大64ワード (レジスタ) までです。
($1 \leq nn \leq 64$)

[W]→W		インデックス付ワード→ワード(FUN206)															
表 現	条件 入力	[① @ ② [W]→W ③] エラー出力														ステップ数	
		* ↑↑↑ データ表示 *符号付表示														4	
機 能	レジスタ①から②番目の内容をレジスタ③へ格納します。						条件入力	処 理								出 力	
							OFF	不 実 行								OFF	
							ON	正常実行								OFF	
エリアオーバー時								ON									
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	転送元先頭レジスタ					○	○	○	○	○	○					
	②	転送元インデックスレジスタ					○		○	○	○						
	③	格納先レジスタ					○		○	○	○						

プログラム例

$\overline{R1000} \quad 00123 \quad 00010 \quad 00111 \quad \text{[RW200 @ RW000 [W]→W RW001]}$

動 作



注 記

- インデックスレジスタの指定範囲は、-32768から32767までです。

説 明

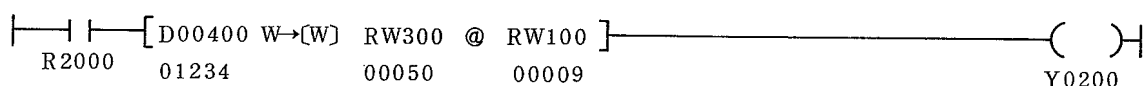
- A接点R1000がONの時、インデックスRW000の内容10を先頭レジスタRW200に加えた番地RW210の内容をRW001へ格納し、出力をOFFにします。
- インデックス修飾後のレジスタが該当レジスタ範囲を超えた場合は(上記プログラムでは、RW000より小さくRW999より大きくなるような場合)、転送は行わず出力をONにします。
- A接点R1000がOFFの時、データの転送は行わず出力をOFFにします。

W→[W]

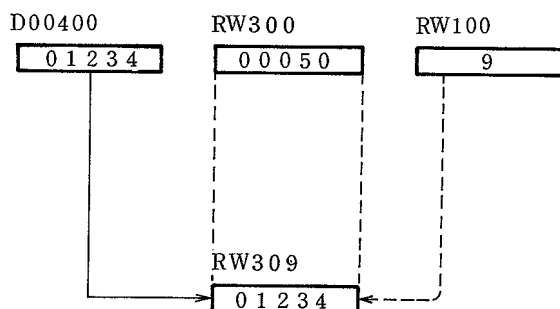
ワード→インデックス付ワード (FUN207)

表現	条件入力 ① W→[W] ② @ ③ エラー出力 データ表示 *符号付表示														ステップ数					
															4					
機能	レジスタ①の内容を，レジスタ②から③番目のレジスタに格納します。						条件入力		処 理						出 力					
							OFF		不 実 行						OFF					
							ON		正常実行						OFF					
エリアオーバー時						ON														
オペランド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	転送元レジスタ									○	○	○	○	○	○				
	②	格納先先頭レジスタ									○		○	○	○					
	③	格納先インデックスレジスタ									○		○	○	○					

プログラム例



動 作



説 明

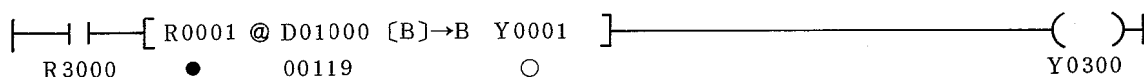
- A接点R2000がONの時D00400の内容をインデックスRW100の内容9を先頭レジスタRW300に加えた番地RW309へ格納し，出力をOFFにします。
- インデックス修飾後のレジスタが該当レジスタ範囲を超えた場合は（上記プログラムではRW000より小さく，RW999より大きくなるような場合）転送は行わず，出力をONにします。
- A接点R2000がOFFの時データの転送は行わず出力をOFFにします。

注 記

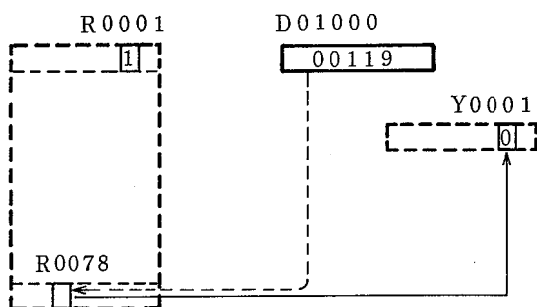
- インデックスレジスタの指定範囲は-32768から32767までです。

[illegible]

プログラム例



動作



說明

- A 接点 R 3000 が ON の時、インデックス D01000 の内容 119 を先頭デバイス R0001 に加えたデバイス R0078 の内容を Y0001 へ格納し、出力を OFF にします。
- インデックス修飾後のデバイスが該当デバイス範囲を超えた場合は（上記プログラムでは R0000 より小さく、R999F より大きくなるような場合）転送は行わず、出力を ON にします。
- A 接点 R 3000 が OFF の時、デバイスの転送は行わず、出力を OFF にします。

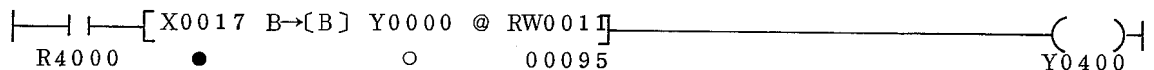
注 記

- インデックスレジスタの指定範囲は-15999から15999までです。

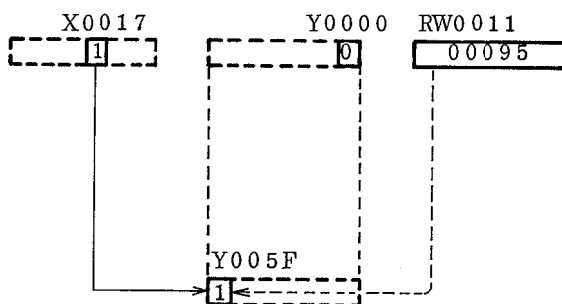
B→[B]	ビット→インデックス付ビット (FUN209)
-------	-------------------------

表 現	条件入力 [① B→[B] ② @ ③] エラー出力														ステップ数					
	○ ○ □□□□□ データ表示 ステータス表示 *符号付表示														4					
機 能	デバイス①の内容を、デバイス②から③番目のデバイスへ格納します。						条件入力	処 理								出 力				
							OFF	不 実 行								OFF				
							ON	正常実行								OFF				
エリアオーバー時								ON												
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	転送元デバイス					○	○	○	○								○	○	
	②	格納先先頭デバイス					○		○	○										
	③	格納先インデックスレジスタ									○		○	○	○					

プログラム例



動 作



説 明

- A接点R4000がONの時，X0017の内容を，インデックスRW0011の内容95を先頭デバイスY0000に加えたデバイスY005Fへ格納し，出力をOFFにします。
- インデックス修飾後のデバイスが該当デバイス範囲を越えた場合は（上記プログラムではY0000より小さくY499Fより大きくなるような場合）転送は行わず，出力をONにします。
- A接点R4000がOFFの時，デバイスの転送は行わず，出力をOFFにします。

注 記

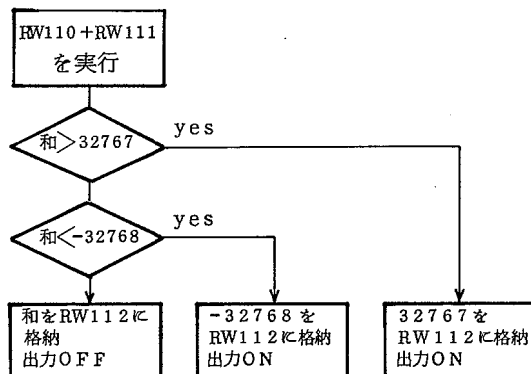
- インデックスレジスタの指定範囲は-15999から15999までです。

+		符 号 付 加 算 (FUN210)														ステップ数	
表 現	機 能	条件入力 ① + ② → ③ オーバーフロー/アンダーフロー出力 符号付データ表示														4	
		データ①と②の和を求め、レジスタ③に格納します。															
オ ペ ラ ン ド	機 能	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値	
		①	被加算数					○	○	○	○	○	○	○		-32768 ~32767	
		②	加数					○	○	○	○	○	○	○		-32768 ~32767	
		③	和					○		○	○	○					

プログラム例

$\left[\begin{array}{c} \text{R010F} \\ \text{01000} \end{array} \right] \left[\begin{array}{c} \text{RW110} + \text{RW111} \rightarrow \text{RW112} \\ \text{00400} \quad \text{01400} \end{array} \right] \text{R010E}$

動 作



説 明

- A接点R010FがONの時、レジスタRW110の内容1000とレジスタRW111の内容400の和を求め、レジスタRW112に1400を格納し、出力をOFFにします。
- 演算中にオーバーフロー、アンダーフローが発生するとリミット値を格納し、出力をONにします。
- A接点R010FがOFFの時、演算を行わず出力をOFFにします。

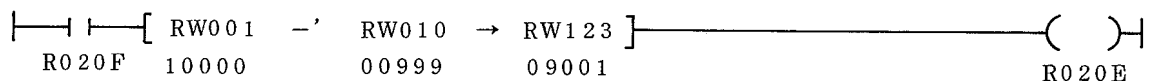
注 記

- この命令では、データ①と②にレジスタ、数値のどちらでも使用可能です。

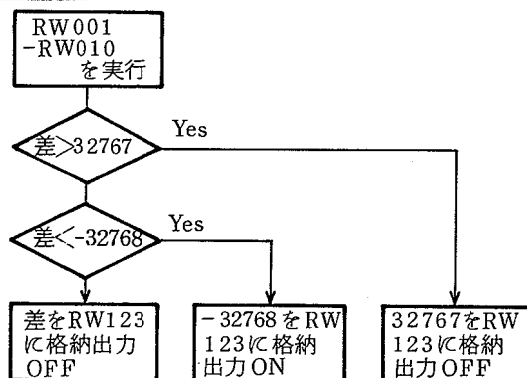
—'	符 号 付 減 算 (FUN211)
----	--------------------

表 現	条件入力 [① —' ② → ③] オーバーフロー/アンダーフロー出力 														ステップ数
															4
機 能	データ①からデータ②を引き、差をレジスタ③に格納します。						条件入力		処 理						出 力
							OFF		不 実 行						OFF
							ON		実行		正常時 オーバーフロー/ アンダーフロー時				OFF ON
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	①	被減数					○	○	○	○	○	○	○		-32768 ~32767
	②	減数					○	○	○	○	○	○	○		-32768 ~32767
	③	差					○		○	○	○				

プログラム例



動 作



説 明

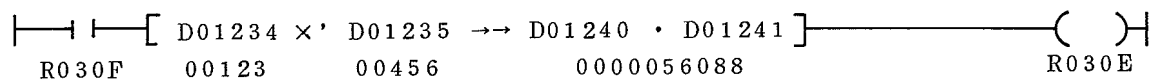
- A接点R020FがONの時、レジスタRW001の内容10000とレジスタRW010の内容999の差を求め、レジスタRW123に9001を格納し、出力をOFFにします。
- 演算中にオーバーフロー、アンダーフローが発生すると、リミット値を格納し、出力をONにします。
- A接点R020FがOFFの時、演算を行わず、出力をOFFにします。

注 記

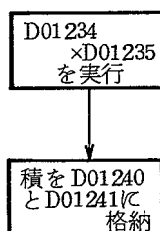
- この命令では、データ①と②にレジスタ、数値のどちらでも使用可能です。

x'		符 号 付 乗 算 (FUN212)														ステップ数	
表 現	条件 入力	$\left[\textcircled{A} \times' \textcircled{B} \rightarrow \rightarrow \textcircled{C} \cdot \textcircled{C} + 1 \right]$ 実行出力														4	
	現																
機 能	データ $\textcircled{A}$ と $\textcircled{B}$ の積を，レジスタ $\textcircled{C} \cdot \textcircled{C} + 1$ に格納します。										条件入力			処 理			出 力
											OFF			不 実 行			OFF
											ON			実 行			ON
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	$\textcircled{A}$	被乗数					○	○	○	○	○	○	○			-32768 ~32767	
	$\textcircled{B}$	乗数					○	○	○	○	○	○	○			-32768 ~32767	
	$\textcircled{C}$ $\textcircled{C}+1$	積					○		○	○	○						

プログラム例



動 作



説 明

- A接点R030FがONの時，レジスタD01234の内容123とレジスタD01235の内容456の積を求めD01240・D01241に56088を格納し，出力をONにします。
- A接点R030FがOFFの時，演算は行わず，出力をOFFにします。

注 記

- この命令ではデータ $\textcircled{A}$ ・ $\textcircled{B}$ にレジスタ，数値のどちらでも使用可能です。

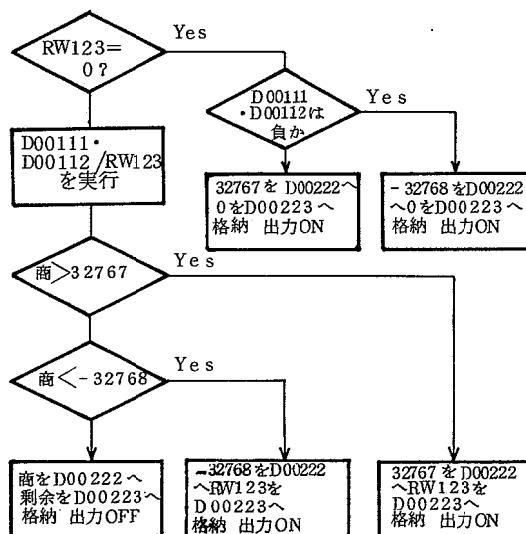
/ ,	符 号 付 除 算 (FUN213)
-----	--------------------

表 現	条件入力 $\left[\begin{array}{c} \textcircled{A} \cdot \textcircled{A} + 1 \quad / \quad \textcircled{B} \end{array} \rightarrow \textcircled{C} \cdot \textcircled{C} + 1 \right]$ オーバーフロー出力												ステップ数			
													$\begin{array}{l} R/R \\ R/K \end{array} \quad 4$ $\begin{array}{l} K/R \\ K/K \end{array} \quad 5$			
機 能	データ $\textcircled{A} \cdot \textcircled{A} + 1$ （倍長）をデータ $\textcircled{B}$ で除し、商をレジスタ $\textcircled{C}$ に、剰余をレジスタ $\textcircled{C} + 1$ に格納します。						条件入力	処　　理						出　　力		
							OFF	不　　実　　行						OFF		
							ON	実　　行	正　　常　　時				OFF			
								オーバーフロー時				ON				
オ ペ ラ ン ド	記号	名　　称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数　　値
	$\textcircled{A}$ $\textcircled{A} + 1$	被除算					○	○	○	○	○	○				-2147483648 ~2147483647
	$\textcircled{B}$	除数					○	○	○	○	○	○				-32768 ~32767
	$\textcircled{C}$	商					○		○	○	○					
	$\textcircled{C} + 1$	剰余														

プログラム例

$\left[\begin{array}{c} \text{D00111} \cdot \text{D00112} / , \text{RW123} \rightarrow \text{D00222} \cdot \text{D00223} \end{array} \right] \text{---} () \text{---}$
 R040F 0000123456 01000 00123 00456 R040E

動 作



説 明

- A接点R040FがONの時、レジスタD00111・D00112の内容123456をレジスタRW123の内容1000で除算を行い、商123をD00222に、剰余456をD00223に格納し、出力をOFFにします。
- 演算中にオーバーフロー、アンダーフローが発生すると、商としてリミット値を、剰余として除数を格納し、出力をONにします。
- 除数が0の時には被除数の符号により商としてリミット値を剰余に0を格納し、出力をONにします。
- A接点R040FがOFFの時、演算は行わず出力をOFFにします。

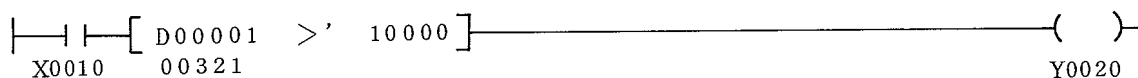
注 記

- この命令ではデータA・A+1、Bに、レジスタ、数値のどちらでも使用可能です。

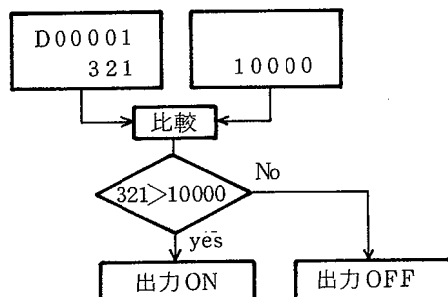
>'	符号付比較・より大きい (FUN214)
----	----------------------

表 現	条件入力 $\left[\begin{array}{c} \text{㉑} > \text{㉒} \end{array} \right]$ 判定出力 符号付 データ表示														ステップ数		
															3		
機 能	データ㉑とデータ㉒を比較して、㉑>㉒の時に出力をONにします。							条件入力		処 理						出 力	
								OFF		不 実 行						OFF	
								ON		㉑ > ㉒ の時 ㉑ ≤ ㉒ の時						ON OFF	
オ ペ ラ ン ド	記号	名 称		R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値	
	㉑	比 較 値						○	○	○	○	○	○			-32768 ~32767	
	㉒	基 準 値						○	○	○	○	○	○			-32768 ~32767	

プログラム例



動 作



説 明

- A接点X0010がONの時、レジスタD00001の内容321と数値10000を比較します。この時、数値10000の方が大きいので、出力をOFFにします。
- A接点X0010がOFFの時、比較は行わず 出力をOFFにします。

注 記

- この命令では、データ①と②にレジスタ、数値のどちらでも使用可能です。

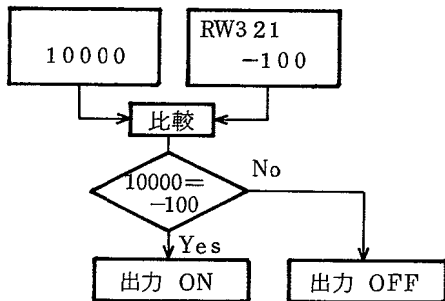
='	符号付比較・等しい (FUN215)
----	--------------------

表 現	条件入力 [① = ②] 判定出力														ステップ数					
	符号付 データ表示														3					
機 能	データ①とデータ②を比較して、 ①＝②の時に出力をONにします。							条件入力	処 理							出 力				
								OFF	不 実 行							OFF				
								ON	①＝② の 時							ON				
							①≠② の 時							OFF						
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	比 較 値									○	○	○	○	○	○	○			-32768 ~32767
	②	基 準 値									○	○	○	○	○	○	○			-32768 ~32767

プログラム例



動 作

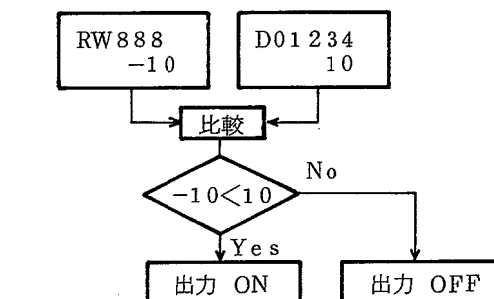
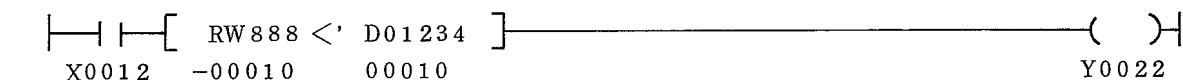


説 明

- A接点X0011がONの時、数値10000とレジスタRW321の内容-100を比較します。
この場合は等しくないので出力をOFFにします。
- A接点X0011がOFFの時、比較は行わず出力はOFFにします。

注 記

- この命令では、データ①と②にレジスタ、数値のどちらでも使用可能です。

[illegible]

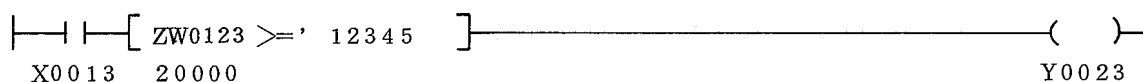
- A接点X0012がONの時、レジスタRW888の内容-10とレジスタD01234の内容10を比較し-10の方が小さいので、出力をONにします。
- A接点X0012がOFFの時比較は行わず出力をOFFにします。

- この命令では、データ④と⑤にレジスタ、数値のどちらでも使用可能です。

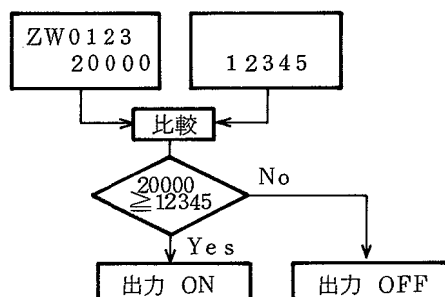
>= '	符号付比較・より大きい又は等しい (FUN217)
------	-----------------------------

表 現	条件入力 { ① >= ' ② } 判定出力 														ステップ数
															3
機 能	データ①とデータ②を比較して、 ①≥②の時に出力をONにします。							条件入力	処 理						出 力
								OFF	不 実 行						OFF
								ON	① ≥ ② の 時						ON
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	①	比 較 値					○	○	○	○	○	○	○		-32768 ~32767
	②	基 準 値					○	○	○	○	○	○	○		-32768 ~32767

プログラム例



動 作



説 明

- A接点X0013がONの時レジスタ ZW0123の内容 20000 と数値12345を比較し、20000の方が大きいので、出力をONにします。
- A接点X0013がOFFの時、比較は行わず出力をOFFにします。

注 記

- この命令では、データ①と②にレジスタ、数値のどちらでも使用可能です。

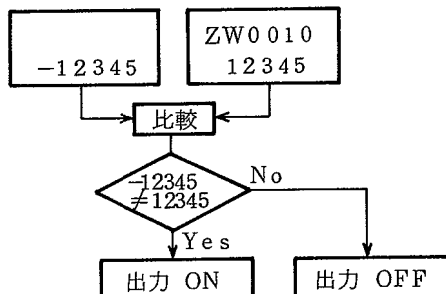
<>'	符号付比較・等しくない (FUN218)
-----	----------------------

表 現	条件入力 $\left[\textcircled{A} \quad <> \quad \textcircled{B} \right]$ 判定出力 符号付 データ表示															ステップ数				
																3				
機 能	データ $\textcircled{A}$ とデータ $\textcircled{B}$ を比較して、 $\textcircled{A} \neq \textcircled{B}$ の時に出力をONにします。						条件入力		処 理						出 力					
							OFF		不 実 行						OFF					
							ON		$\textcircled{A} \neq \textcircled{B}$ の 時						ON					
								$\textcircled{A} = \textcircled{B}$ の 時						OFF						
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	$\textcircled{A}$	比 較 値									○	○	○	○	○	○	○			-32768 ~32767
	$\textcircled{B}$	基 準 値									○	○	○	○	○	○	○			-32767 ~32767

プログラム例



動 作



説 明

- A接点X0014がONの時、数値-12345とレジスタZW0010の内容12345を比較し等しくないので、出力をONにします。
- A接点X0014がOFFの時比較は行わず出力をOFFにします。

注 記

- この命令ではデータ①と②にレジスタ、数値のどちらでも使用可能です。

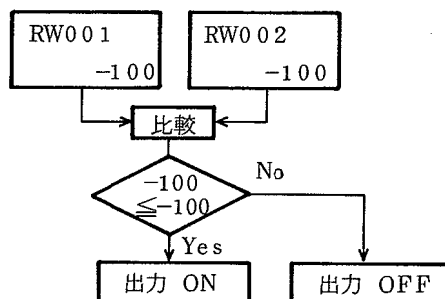
<='	符号付比較・より小さい又は等しい (FUN219)
-----	---------------------------

表 現	条件入力 ① <= ② 判定出力 符号付 データ表示														ステップ数					
															3					
機 能	データ①とデータ②を比較して， ①≤②の時に出力をONにします。						条件入力		処 理						出 力					
							OFF		不 実 行						OFF					
							ON		① ≤ ② の 時						ON					
① > ② の 時						OFF														
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	比 較 値									○	○	○	○	○	○	○			-32768 ～32767
	②	基 準 値									○	○	○	○	○	○	○			-32768 ～32767

プログラム例



動 作



説 明

- A接点X0015がONの時レジスタRW001の内容-100とレジスタRW002の内容-100を比較し等しいので、出力をONにします。
- A接点X0015がOFFの時比較は行わず出力をOFFにします。

注 記

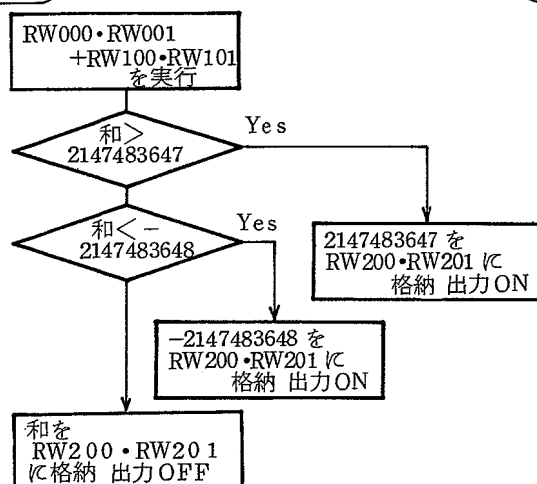
- この命令ではデータ①と②にレジスタ、数値のどちらでも使用可能です。

++'		符 号 付 倍 長 加 算 (FUN220)											
表 現	条件 入力	オーバー フロー/ アンダー フロー出 力 ①・①+1 ++' ②・②+1 →→ ③・③+1											

プログラム例

$\begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline \text{Z0107} & \text{RW000} & \text{RW001} & ++' & \text{RW100} & \text{RW101} & \rightarrow \rightarrow & \text{RW200} & \text{RW201} & \text{Z0207} \\ \hline 0001000000 & & & & 0000234567 & & & 0001234567 & & \end{array}$

動 作



説 明

- A接点Z0107がONの時レジスタRW000・RW001の倍長データ1000000とレジスタRW100・RW101の倍長データ234567の和を求めRW200・RW201に格納し、出力をOFFにします。
- 演算中にオーバーフロー、アンダーフローが発生した場合は、リミット値を格納し出力をONにします。
- A接点Z0107がOFFの時、演算を行わず出力をOFFにします。

注 記

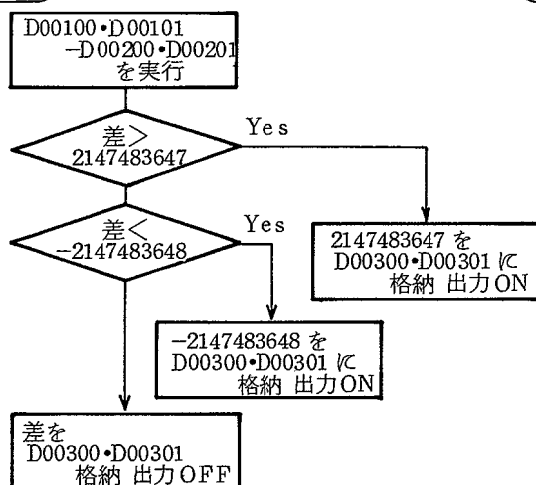
- この命令ではデータ(A)・(A)+1, (B)・(B)+1にレジスタ、数値のどちらでも使用可能です。

--'		符 号 付 倍 長 減 算 (FUN221)															
表 現	条件 入力	①・①+1 --' ②・②+1 →→ ③・③+1 オーバー フロー/ アンダー フロー出 力 符号付 データ表示														ステップ数	
		R--R 4															
		R--K 5															
		K--R 6															
機 能	データ①・①+1 からデータ②・②+1 を引き、差をレジスタ③・③+1 に格納 します。						条件入力		処 理						出 力		
	OFF						OFF		不 実 行						OFF		
	ON						実 行		正 常 時 オーバーフロー/ アンダーフロー時						OFF ON		
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	① ①+1	被減数					○	○	○	○	○	○	○			-2147483648 ~2147483647	
	② ②+1	減 数					○	○	○	○	○	○	○			-2147483648 ~2147483647	
	③ ③+1	差					○		○	○	○						

プログラム例

$\text{Z0108} \left[\text{D00100} \cdot \text{D00101} \text{ --' } \text{D00200} \cdot \text{D00201} \rightarrow \rightarrow \text{D00300} \cdot \text{D00301} \right] \text{H} \text{H}$
 $2000000000 \quad 2100000000 \quad -0100000000 \quad \text{Z0208}$

動 作



説 明

- A接点Z0108がONの時、レジスタD00100・D00101の内容2000000000とレジスタD00200・D00201の内容2100000000の差を求め、D00300・D00301に-1000000000を格納し、出力をOFFにします。
- 演算中にオーバーフロー、アンダーフローが発生すると、リミット値を格納し、出力をONにします。
- A接点Z0108がOFFの時演算を行わず出力をOFFにします。

注 記

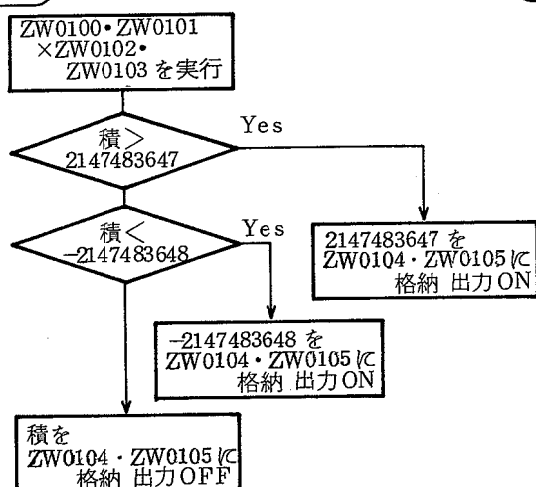
- この命令では、データ④・④+1と⑤・⑤+1にレジスタ、数値のどちらでも使用可能です。

xx'		符 号 付 倍 長 乗 算 (FUN222)															
表 現	条件 入力	① · ①+1 xx' ② · ②+1 →→ ③ · ③+1 オーバーフロー出力 符号付データ表示														ステップ数	
		R×R	4														
		R×K K×R	5														
		K×K	6														
機 能	データ① · ①+1 とデータ② · ②+1 の積をレジスタ③ · ③+1 に格納します。							条件入力		処 理						出 力	
								OFF		不 実 行						OFF	
								ON		実 行		正 常 時				OFF	
											オーバーフロー時				ON		
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T	C	数 値	
	①	被 乗 数					○	○	○	○	○	○	○			-2147483648	
	①+1														~2147483647		
	②	乗 数					○	○	○	○	○	○	○			-2147483648	
	②+1														~2147483647		
	③	積					○		○	○	○						
③+1																	

プログラム例

$\text{Z0109} \quad \text{ZW0100} \cdot \text{ZW0101} \times \times' \text{ZW0102} \cdot \text{ZW0103} \rightarrow \rightarrow \text{ZW0104} \cdot \text{ZW0105} \quad \text{Z0209}$
 $\quad \quad \quad -0000101010 \quad \quad \quad -0000000005 \quad \quad \quad 0000505050$

動 作



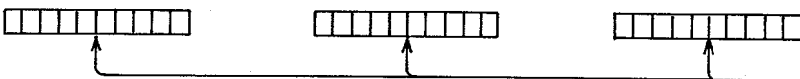
説 明

- A接点Z109がONの時レジスタZW0100・ZW0101の内容-101010とレジスタZW0102・ZW0103の内容-5の積を求め、ZW0104・ZW0105に505050を格納し、出力をOFFにします。
- 演算中にオーバーフロー、アンダーフローが発生した場合は、リミット値を格納し出力をONにします。
- A接点Z0109がOFFの時演算は行わず出力をOFFにします。

注 記

- この命令では、データ $\textcircled{A} \cdot \textcircled{A}+1$ 、 $\textcircled{B} \cdot \textcircled{B}+1$ にレジスタ、数値のどちらでも使用可能です。

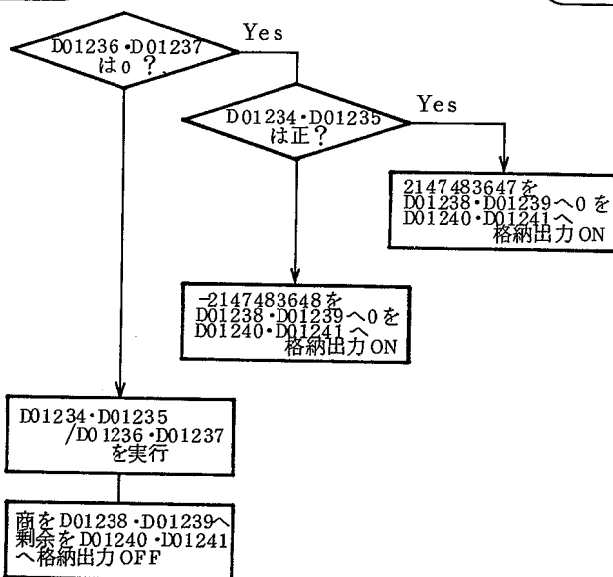
//'	符号付倍長除算 (FUN223)
-----	------------------

表現	条件入力 $\left[\begin{array}{c} \textcircled{A} \cdot \textcircled{A}+1 \quad // \quad \textcircled{B} \cdot \textcircled{B}+1 \quad \rightarrow \rightarrow \quad \textcircled{C} \cdot \textcircled{C}+1 \end{array} \right]$ オーバーフロー出力														ステップ数	
															R//R 4	
	符号付データ表示														R//K 5 K//R 5	
															K//K 6	
機能	データ $\textcircled{A} \cdot \textcircled{A}+1$ をデータ $\textcircled{B} \cdot \textcircled{B}+1$ で除し、商をレジスタ $\textcircled{C} \cdot \textcircled{C}+1$ に、剰余を $\textcircled{C}+2 \cdot \textcircled{C}+3$ に格納します。						条件入力		処 理						出 力	
							OFF		不 実 行						OFF	
							ON		実 行		正 常 時				OFF	
										オーバーフロー時				ON		
オペランド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	$\textcircled{A}$ $\textcircled{A}+1$	被 除 数					○	○	○	○	○	○	○			-2147483648 ~2147483647
	$\textcircled{B}$ $\textcircled{B}+1$	除 数					○	○	○	○	○	○	○			-2147483648 ~2147483647
	$\textcircled{C}$ $\textcircled{C}+1$	商					○		○	○	○					

プログラム例

$\left[\begin{array}{c} \text{D01234} \cdot \text{D01235} \quad // \quad \text{D01236} \cdot \text{D01237} \quad \rightarrow \rightarrow \quad \text{D01238} \cdot \text{D01239} \end{array} \right] \text{H} \text{H}$
 Z010A 0000123456 0000010000 0000000012 Z020A

動 作



説 明

- A接点Z010AがONの時レジスタD01234・D01235の内容123456をレジスタD01236・D01237の内容10000で除算を行い、商12をD01238・D01239に剰余3456をD01240・D01241に格納し、出力をOFFにします。
- 除数が0の時には、被除数の符号により商としてリミット値、剰余として0を格納し出力をONにします。
- A接点Z010AがOFFの時演算は行わず出力をOFFにします。

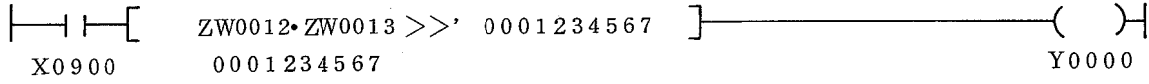
注 記

- この命令では、データA・A+1とB・B+1にレジスタ、数値のどちらでも使用可能です。

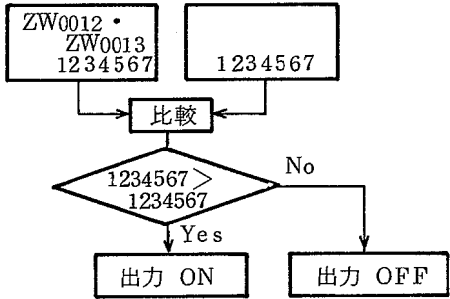
>>'	符号付倍長比較・より大きい (FUN224)
-----	--------------------------

表 現	条件入力 $\textcircled{A} \cdot \textcircled{A} + 1 >>' \textcircled{B} \cdot \textcircled{B} + 1$ 判定出力 符号付データ表示														ステップ数
															R>>R 3
															R>>K K>>R 4
															K>>K 5
機 能	データ $\textcircled{A} \cdot \textcircled{A} + 1$ とデータ $\textcircled{B} \cdot \textcircled{B} + 1$ を比較して、 $\textcircled{A} \cdot \textcircled{A} + 1 > \textcircled{B} \cdot \textcircled{B} + 1$ の時に出力をONにします。						条件入力	処 理							出 力
							OFF	不 実 行							OFF
							ON	$\textcircled{A} \cdot \textcircled{A} + 1 > \textcircled{B} \cdot \textcircled{B} + 1$							ON
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	$\textcircled{A}$ $\textcircled{A} + 1$	比 較 値					○	○	○	○	○	○			-2147483648 ~2147483647
	$\textcircled{B}$ $\textcircled{B} + 1$	基 準 値					○	○	○	○	○	○			-2147483648 ~2147483647

プログラム例



動 作



説 明

- A接点X0900 がONの時レジスタ ZW0012・ZW0013 の内容1234567と数値1234567を比較し等しいので、出力をOFFにします。
- A接点X0900 がOFFの時、比較は行わず出力をOFFにします。

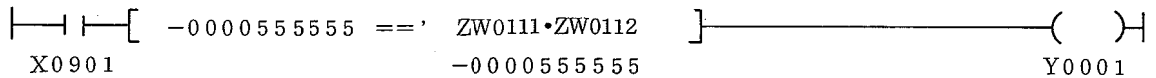
注 記

● この命令では、データ $\textcircled{A} \cdot \textcircled{A} + 1$ と $\textcircled{B} \cdot \textcircled{B} + 1$ にレジスタ、数値のどちらでも使用可能です。

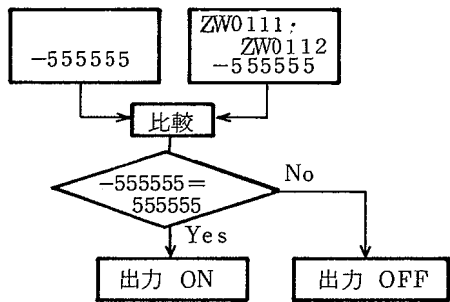
=='	符号付倍長比較・等しい (FUN225)
-----	----------------------

表現	条件入力 $\textcircled{A} \cdot \textcircled{A} + 1 ==' \textcircled{B} \cdot \textcircled{B} + 1$ 判定出力 符号付データ表示														ステップ数
															R==R 3
															R==K K==R 4
															K==K 5
機能	データ $\textcircled{A} \cdot \textcircled{A} + 1$ とデータ $\textcircled{B} \cdot \textcircled{B} + 1$ を比較して、 $\textcircled{A} \cdot \textcircled{A} + 1 = \textcircled{B} \cdot \textcircled{B} + 1$ の時に出力をONにします。														出力
	条件入力														処 理
	OFF														不 実 行
	ON														$\textcircled{A} \cdot \textcircled{A} + 1 = \textcircled{B} \cdot \textcircled{B} + 1$
															$\textcircled{A} \cdot \textcircled{A} + 1 \neq \textcircled{B} \cdot \textcircled{B} + 1$
オペランド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	$\textcircled{A}$	比 較 値					○	○	○	○	○	○			-2147483648 ~2147483647
	$\textcircled{B}$	基 準 値					○	○	○	○	○	○			-2147483648 ~2147483647
	$\textcircled{A}+1$														
	$\textcircled{B}+1$														

プログラム例



動作



説明

- A接点X0901 がONの時数値-555555とレジスタZW0111・ZW0112の内容-555555を比較し等しいので、出力をONにします。
- A接点X0901 がOFFの時、比較は行わず出力をOFFにします。

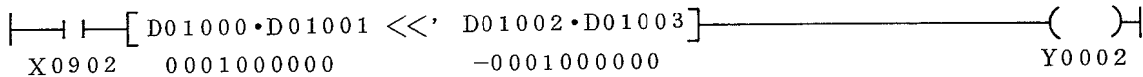
注 記

- この命令ではデータ $\textcircled{A} \cdot \textcircled{A} + 1$ と $\textcircled{B} \cdot \textcircled{B} + 1$ にレジスタ、数値のどちらでも使用可能です。

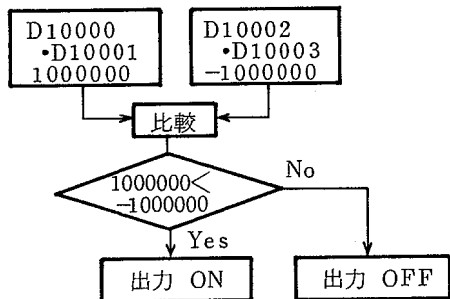
<<'	符号付倍長比較・より小さい (FUN226)
-----	------------------------

表 現	条件入力 $\left[\begin{array}{c} \textcircled{A} \cdot \textcircled{A} + 1 << \textcircled{B} \cdot \textcircled{B} + 1 \end{array} \right]$ 判定出力															ステップ数
																$R << R$ 3
																$R << K$ $K << R$ 4
																$K << K$ 5
機 能	データ $\textcircled{A} \cdot \textcircled{A} + 1$ とデータ $\textcircled{B} \cdot \textcircled{B} + 1$ を比較して、 $\textcircled{A} \cdot \textcircled{A} + 1 < \textcircled{B} \cdot \textcircled{B} + 1$ の時に出力をONにします。							条件入力	処 理							出 力
								OFF	不 実 行							OFF
								ON	$\textcircled{A} \cdot \textcircled{A} + 1 < \textcircled{B} \cdot \textcircled{B} + 1$							ON
									$\textcircled{A} \cdot \textcircled{A} + 1 \geq \textcircled{B} \cdot \textcircled{B} + 1$							OFF
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	$\textcircled{A}$ $\textcircled{A} + 1$	比 較 値					○	○	○	○	○	○				-2147483648 ~2147483647
	$\textcircled{B}$ $\textcircled{B} + 1$	基 準 値					○	○	○	○	○	○				-2147483648 ~2147483647

プログラム例



動 作



説 明

- A接点X0902がONの時レジスタD01000・D01001の内容1000000とレジスタD01002・D01003の内容-1000000を比較し、1000000の方が大きいので出力をOFFにします。
- A接点X0902がOFFの時比較は行わず出力をOFFにします。

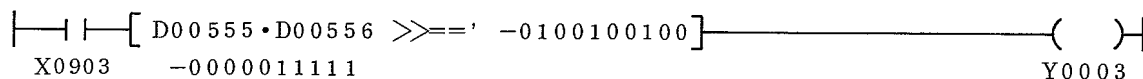
注 記

- この命令ではデータ $\textcircled{A} \cdot \textcircled{A} + 1$ と $\textcircled{B} \cdot \textcircled{B} + 1$ にレジスタ、数値のどちらでも使用可能です。

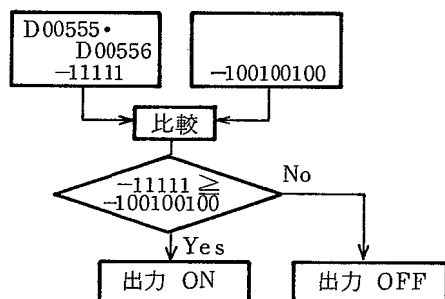
>>=='	符号付倍長比較・より大きい又は等しい(FUN227)
-------	----------------------------

表 現	条件入力 $\left[\textcircled{A} \cdot \textcircled{A} + 1 >>==', \textcircled{B} \cdot \textcircled{B} + 1 \right]$ 判定出力														ステップ数	
															R>>==R 3	
															R>>==K 4	
															K>>==R 4	
														K>>==K 5		
機 能	データ $\textcircled{A} \cdot \textcircled{A} + 1$ とデータ $\textcircled{B} \cdot \textcircled{B} + 1$ を比較して、 $\textcircled{A} \cdot \textcircled{A} + 1 \geq \textcircled{B} \cdot \textcircled{B} + 1$ の時に出力をONにします。						条件入力		処 理						出 力	
							OFF		不 実 行						OFF	
							ON		$\textcircled{A} \cdot \textcircled{A} + 1 \geq \textcircled{B} \cdot \textcircled{B} + 1$						ON	
		$\textcircled{A} \cdot \textcircled{A} + 1 < \textcircled{B} \cdot \textcircled{B} + 1$						OFF								
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	$\textcircled{A}+1$	比 較 値					○	○	○	○	○	○	○			-2147483648 ~2147483647
	$\textcircled{B}+1$	基 準 値					○	○	○	○	○	○	○			-2147483648 ~2147483647

プログラム例



動 作



説 明

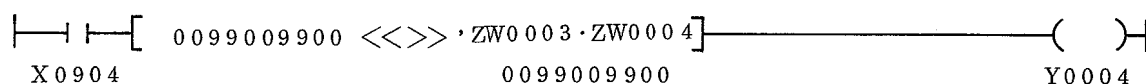
- A接点X0903 がONの時レジスタD00555・D00556の内容-11111と数値-100100100を比較し、-11111の方が大きいので出力をONにします。
- A接点X0903 がOFFの時、比較は行わず出力をOFFにします。

注 記

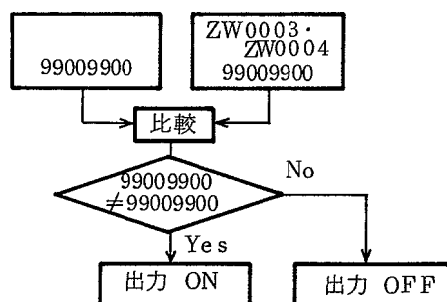
- この命令ではデータ $\textcircled{A} \cdot \textcircled{A} + 1$ と $\textcircled{B} \cdot \textcircled{B} + 1$ にレジスタ、数値のどちらでも使用可能です。

<<>>'		符号付倍長比較・等しくない (FUN228)															
表現	条件入力 $\textcircled{A} \cdot \textcircled{A} + 1$ <<>>' $\textcircled{B} \cdot \textcircled{B} + 1$ 判定出力 符号付データ表示														ステップ数 R<<>>R 3 R<<>>K 4 K<<>>R 4 K<<>>K 5		
	データ $\textcircled{A} \cdot \textcircled{A} + 1$ とデータ $\textcircled{B} \cdot \textcircled{B} + 1$ を比較して、 $\textcircled{A} \cdot \textcircled{A} + 1 \neq \textcircled{B} \cdot \textcircled{B} + 1$ の時に出力をONにします。							条件入力		処 理					出 力		
								OFF		不 実 行					OFF		
								ON		$\textcircled{A} \cdot \textcircled{A} + 1 \neq \textcircled{B} \cdot \textcircled{B} + 1$ $\textcircled{A} \cdot \textcircled{A} + 1 = \textcircled{B} \cdot \textcircled{B} + 1$					ON OFF		
オペランド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	$\textcircled{A}$ $\textcircled{A} + 1$	比 較 値					○	○	○	○	○	○				-2147483648 ~2147483647	
	$\textcircled{B}$ $\textcircled{B} + 1$	基 準 値					○	○	○	○	○	○				-2147483648 ~2147483647	

プログラム例



動作



說明

- A接点X0904がONの時数値99009900とレジスタZW0003・ZW0004の内容99009900を比較し、等しいので、出力をOFFにします。
- A接点X0904がOFFの時、比較は行わず出力をOFFにします。

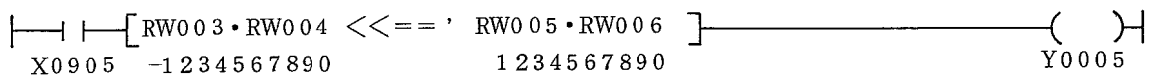
注 記

- この命令ではデータ④・④+1と⑤・⑤+1にレジスタ，数値のどちらでも使用可能です。

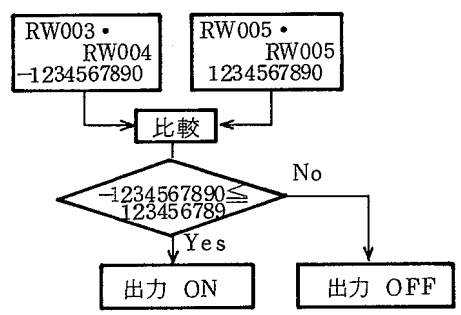
<<== ' 符号付倍長比較・より小さい又は等しい (FUN229)

表現	条件入力 $\left[\begin{array}{c} \textcircled{A} \cdot \textcircled{A} + 1 <<== ' \quad \textcircled{B} \cdot \textcircled{B} + 1 \end{array} \right]$ 判定出力														ステップ数	
															R<<==R 3	
	符号付 データ表示														R<<==K 4	
															K<<==R 5	
機能	データ $\textcircled{A} \cdot \textcircled{A} + 1$ とデータ $\textcircled{B} \cdot \textcircled{B} + 1$ を 比較して、 $\textcircled{A} \cdot \textcircled{A} + 1 \leq \textcircled{B} \cdot \textcircled{B} + 1$ の時に 出力をONにします。							条件入力	処 理							出 力
								OFF	不 実 行							OFF
								ON	$\textcircled{A} \cdot \textcircled{A} + 1 \leq \textcircled{B} \cdot \textcircled{B} + 1$							ON
								$\textcircled{A} \cdot \textcircled{A} + 1 > \textcircled{B} \cdot \textcircled{B} + 1$							OFF	
オペランド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	$\textcircled{A}$	比 較 値					○	○	○	○	○	○				-2147483648
	$\textcircled{A}+1$														~2147483647	
	$\textcircled{B}$	基 準 値					○	○	○	○	○	○				-2147483648
	$\textcircled{B}+1$														~2147483647	

プログラム例



動 作



説 明

- A接点X0905がONの時レジスタRW003・RW004の内容-1234567890とレジスタRW005・RW006の内容1234567890を比較し-1234567890の方が小さいので出力をONにします。
- A接点X0905がOFFの時比較は行わず出力をOFFにします。

注 記

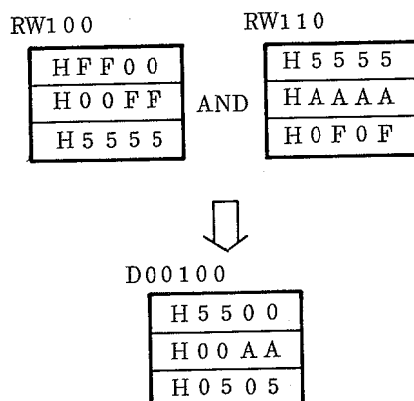
- この命令ではデータ $\textcircled{A} \cdot \textcircled{A} + 1$ と $\textcircled{B} \cdot \textcircled{B} + 1$ にレジスタ、数値のどちらでも使用可能です。

AND*		ビットファイル 論理積 (FUN230)														ステップ数											
表 現	機 能	条件入力 [① AND* [nn] ②] → ③ 実行出力														5											
		H <table><tr><td></td><td></td><td></td><td></td></tr></table> H <table><tr><td></td><td></td><td></td><td></td></tr></table> H <table><tr><td></td><td></td><td></td><td></td></tr></table> データ表示																									
機 能	レジスタ①, ②で指定される2組のテーブルサイズ [nn] のデータのビット毎の論理積を求め、レジスタ③以降に格納します。	条件入力		処 理										出 力													
		OFF		不 実 行										OFF													
		ON		実 行										ON													
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値											
	①	テーブル先頭レジスタ					○	○	○	○	○	○	○														
	nn	テーブルサイズ (レジスタ数)														1 ~ 64											
	②	テーブル先頭レジスタ					○	○	○	○	○	○	○														
	③	論理積格納先頭レジスタ					○		○	○	○																

プログラム例

$\begin{array}{|c|} \hline \text{R8000} \\ \hline \end{array} \left[\begin{array}{l} \text{RW100 AND* [3] RW110} \\ \text{HFF00 H5555} \end{array} \right] \rightarrow \begin{array}{|c|} \hline \text{D00100} \\ \hline \end{array} \left[\begin{array}{l} \text{H5500} \\ \hline \end{array} \right] \text{R9010}$

動 作



説 明

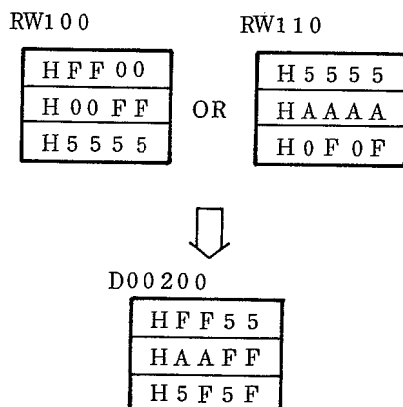
- A接点R8000がONの時RW100を先頭とする3ワードのテーブルとRW110を先頭とする3ワードのテーブルのビット毎の論理積をとってD00100を先頭とする3ワードのテーブルに格納し、出力をONにします。
- A接点R8000がOFFの時演算は行わず出力をOFFにします。

OR*		ビットファイル 論理和 (FUN231)																		
表 現	条件 入力	[① OR* [nn] ② → ③] 実行出力												ステップ数						
		H H H データ表示												5						
機 能	レジスタ①, ②で指定される2組のテーブルサイズ [nn] のデータのビット毎の論理和を求め、レジスタ③以降に格納します。						条件入力		処 理						出 力					
							OFF		不 実 行						OFF					
							ON		実 行						ON					
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	テーブル先頭レジスタ									○	○	○	○	○	○	○			
	nn	テーブルサイズ(レジスタ数)																		1 ~ 64
	②	テーブル先頭レジスタ									○	○	○	○	○	○	○			
	③	論理和格納先頭レジスタ									○		○	○	○					

プログラム例

$\left[\begin{array}{c} \text{RW100} \\ \text{HFF00} \end{array} \text{ OR* } [3] \begin{array}{c} \text{RW110} \\ \text{H5555} \end{array} \rightarrow \begin{array}{c} \text{D00200} \\ \text{HFF55} \end{array} \right] \text{ () } \text{ R8011}$

動 作



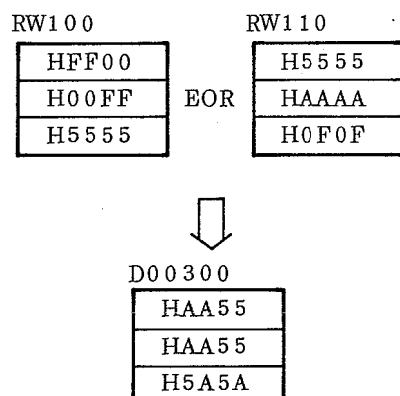
説 明

- A接点R8001がONの時、RW100を先頭とする3ワードのテーブルとRW110を先頭とする3ワードのテーブルのビット毎の論理和をとってD00200を先頭とする3ワードのテーブルに格納し、出力をONにします。
- A接点R8001がOFFの時、演算は行わず出力をOFFにします。

EOR*		ビットファイル排他的論理和 (FUN232)														ステップ数													
表 現	機 能	条件入力 [① EOR* [nn] ②] → ③ 実行出力 H <table><tr><td></td><td></td><td></td><td></td></tr></table> H <table><tr><td></td><td></td><td></td><td></td></tr></table> H <table><tr><td></td><td></td><td></td><td></td></tr></table> データ表示																										5	
レジスタ①, ②で指定される 2 組のテーブルサイズ [nn] のデータのビット毎の排他的論理和を求め、レジスタ③以降に格納します。														OFF		ON													
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値													
	①	テーブル先頭レジスタ					○	○	○	○	○	○	○																
	nn	テーブルサイズ(レジスタ数)														1~64													
	②	テーブル先頭レジスタ					○	○	○	○	○	○	○																
	③	排他的論理和先頭レジスタ					○		○	○	○																		

プログラム例

動 作

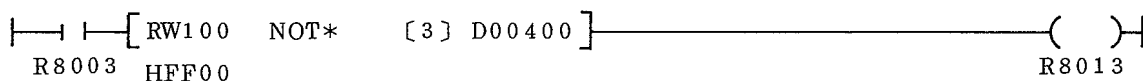


説 明

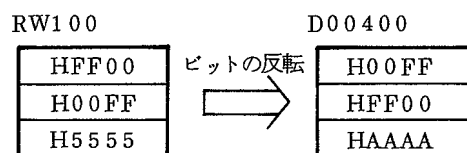
- A接点 R8002 が ON の時、RW100 を先頭とする 3 ワードのテーブルと、RW110 を先頭とする 3 ワードのテーブルのビット毎の排他的論理和をとって D00300 を先頭とする 3 ワードのテーブルに格納し、出力を ON にします。
- A接点 R8002 が OFF の時、演算は行わず出力を OFF にします。

[illegible]

プログラム例



動作

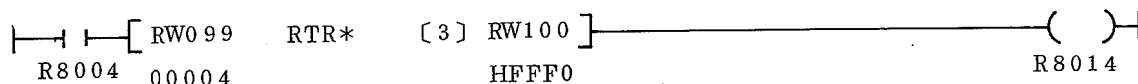


說明

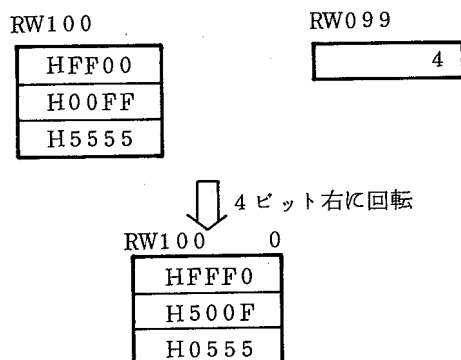
- A接点R8003がONの時 RW100を先頭とする
3ワードのテーブルの反転データをD00400を
先頭とする3ワードのテーブルに格納し、出力を
ONにします。
- A接点R8003がOFFの時、演算は行わず、出
力をOFFにします。

[illegible]

プログラム例



動作



說明

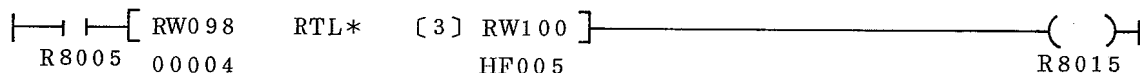
- A 接点 R8004 が ON の時、RW100 を先頭とする 3 ワードのテーブルを RW099 で指定されたビット数分だけ右に回転します。回転の結果ビット 0 が 1 で出力が ON、0 で出力が OFF になりますので、この場合の出力は OFF にします。
- ローテートビット数がレジスタで、その内容が 16 を超えた時は、実行せず出力を OFF にします。
- A 接点 R8004 が OFF の時、実行は行わず出力を OFF にします。

注 記

- テーブルサイズは最大 64 ワード (レジスタ) までです。
($1 \leq n \leq 64$)
- ローテートビット数は、レジスタでも数値でも使用できます。

RTL*		ビットファイル左ローテート (FUN235)														ステップ数	
表 現	機 能	条件入力 ① RTL* [nn] ② 最終ローテートビットステータス出力 H データ表示														4	
		レジスタ②で指定されるテーブルサイズ [nn] のデータをデータ①のビット数分だけ左ローテートします。 条件入力 処 理 出 力 OFF 不 実 行 OFF ON 実 行 ビット0='1' ビット0='0' ON OFF OFF エラー時 OFF															
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値		
	①	ローテートビット数					○	○	○	○	○	○			1~16		
	nn	テーブルサイズ(レジスタ数)													1~64		
	②	テーブル先頭レジスタ					○		○	○	○						

プログラム例



動 作

RW100

HF00
H00FF
H5555

RW098

4

↓ 4ビット左に回転

RW100 0
HF005
H0FFF
H5550

説 明

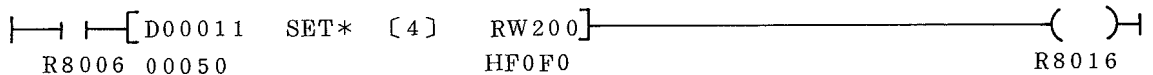
- A接点R8005がONの時、RW100を先頭とする3ワードのテーブルをRW098で指定されたビット数分だけ左に回転します。回転の結果ビット0が1で出力がON、0で出力がOFFになりますので、この場合の出力はONにします。
- ローテートビット数がレジスタでその内容が16を超えた時は、実行せず、出力をOFFにします。
- A接点R8005がOFFの時、実行は行わず出力をOFFにします。

注 記

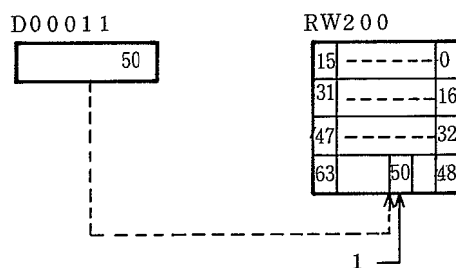
- テーブルサイズは最大64ワード(レジスタ)までです。
($1 \leq nn \leq 64$)
- ローテートビット数はレジスタでも数値でも使用できます。

SET*		ビットファイルビットセット (FUN236)														ステップ数	
表 現	機 能	条件入力 [㉠ SET* [nn] ㉡] 実行出力 4 H 4 データ表示														4	
		レジスタ㉡で指定されるビットテーブルサイズ [nn] のレジスタ㉠番目のビットを ON にセットします。															
オ ペ ラ ン ド	機 能	条件入力	処 理												出 力		
		OFF	不 実 行												OFF		
		ON	正 常 実 行												ON		
			エ ラ ー 時												OFF		
		記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
		㉠	セットビットポインタ					○	○	○	○	○	○				
nn	テーブルサイズ (レジスタ数)														1 ~ 6 4		
㉡	テーブル先頭レジスタ					○		○	○	○							

プログラム例



動 作



説 明

- A接点 R8006 が ON の時、RW200 を先頭とする 4 ワードのテーブルの D00011 で表されるビットポインタ 50 番目のデバイスにセットし、出力を ON にします。
- ビットポインタが、テーブル範囲外の時は (この場合は 64 以上の時) セットは行わず出力を OFF にします。
- A接点 R8006 が OFF の時、セットは行わず出力を OFF にします。

注 記

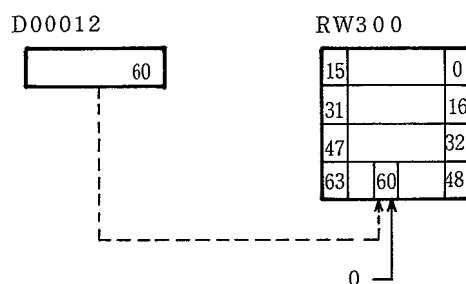
- テーブルサイズは、最大 64 ワード (レジスタ) までです。
($1 \leq nn \leq 64$)

[illegible]

プログラム例

|—|—[D00012 RST* [4] RW300]————()—
R8007 00060 HF0F0 R8017

動作

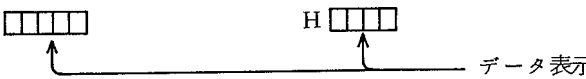


說明

- A 接点 R 8 0 0 7 が ON の時、 RW 3 0 0 を先頭とする 4 ワードのテーブルの D 0 0 0 1 2 で示されるビットポインタ 6 0 番目のデバイスをリセットし、出力を ON にします。
- ビットポインタがテーブル範囲外の時は、（この場合は 6 4 以上の時）リセットは行わず出力を OFF にします。
- A 接点 R 8 0 0 7 が OFF の時、リセットは行わず出力を OFF にします。

記 注

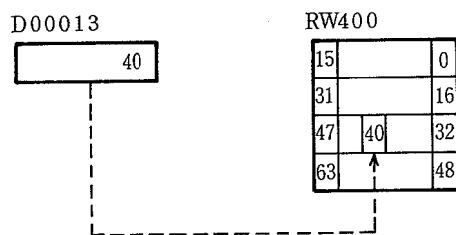
- テーブルサイズは最大 64 ワード (レジスタ) までです。
($1 \leq n_n \leq 64$)

TEST*		ビットファイルビットテスト (FUN238)														ステップ数			
表 現	機 能	条件入力 [① TEST* [nn] ②] ビットステータス出力														4			
																			
オ ペ ラ ン ド	機 能	レジスタ②で指定されるビットテーブルサイズ[nn]のレジスタ①番目のビットのON/OFF状態を判定します。								条件入力		処 理						出 力	
										OFF		不 実 行						OFF	
										ON		正常実行 エラー時		結果 = '1'		ON			
		結果 = '0'		OFF															
				OFF															
		記号		名 称		R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
①		テストビットポインタ						○	○	○	○	○	○						
nn		テーブルサイズ(レジスタ数)																1~64	
②		テーブル先頭レジスタ						○	○	○	○	○	○	○					

プログラム例

$\begin{array}{|c|c|} \hline \text{R8008} & \text{00040} \\ \hline \end{array} \left[\begin{array}{|c|c|c|c|} \hline \text{D00013} & \text{TEST*} & [4] & \text{RW400} \\ \hline \end{array} \right] \text{---} \left(\begin{array}{|c|} \hline \text{R8018} \\ \hline \end{array} \right)$

動 作



説 明

- A接点R8008がONの時、RW400を先頭とする4ワードのテーブルのD00013で示されるビットポインタ40番目のビットを調べてこのビットが1の時は出力をONに、ビットが0の時は、出力をOFFにします。
- ビットポインタがテーブル範囲外の時は(この場合は64以上の時)テストは行わず、出力をOFFにします。
- A接点R8008がOFFの時、テストは行わず、出力をOFFにします。

注 記

- テーブルサイズは最大64ワード(レジスタ)までです。
($1 \leq nn \leq 64$)

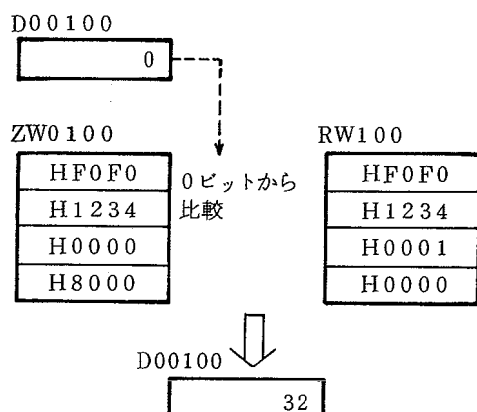
CMP*

ビットファイル比較 (FUN239)

表現	条件入力 ① CMP* [nn] ② → ③ 不一致検出出力 H H データ表示															ステップ数
																5
機能	レジスタ①, ②で指定される2組のテーブルサイズ [nn] のデータを比較し、不一致ビット No.をレジスタ③に格納します。						条件入力		処 理						出 力	
							OFF		不 実 行						OFF	
							ON		実 行		不一致ビット有り				ON	
オペランド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	テーブル先頭レジスタ					○	○	○	○	○	○	○			
	nn	テーブルサイズ(レジスタ数)														1～64
	②	テーブル先頭レジスタ					○	○	○	○	○	○	○			
	③	不一致ビットポインタ					○		○	○	○					

プログラム例

動作



注 記

- テーブルサイズは最大64ワード(レジスタ)までです。
($1 \leq n \leq 64$)

說明

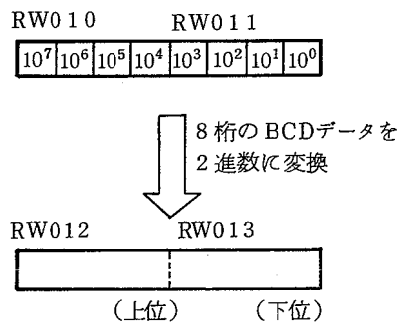
- A接点R8009がONの時、ZW0100を先頭とする4ワードのテーブルとRW100を先頭とする4ワードのテーブルを比較します。
不一致ビットポイントが0ですので、比較は1から始め、32ビット目のビットが不一致なので、32を不一致ビットポイントとしてD00100に格納し、出力をONにします。
- 途中から比較を始めて、テーブルの最後まで行ったら、先頭にもどって比較をします。
- 元のポイント値まで比較を行い、不一致が無かった時は、そこで実行を停止し、ビットポイントは不変のまま、出力をOFFにします。
- 実行開始前にポイントがテーブル範囲外の時は、（この場合は64以上の時）0ビット目から比較を行ないます。
- A接点R8009がOFFの時は、比較は行わず出力をOFFにします。

BIN2		倍長 バイナリ変換 (FUN240)													
表 現	条件 入力	条件 [① ・ ①+1 BIN2 ② ・ ②+1] エラー出力												ステップ数	
		H 符号なし データ表示 <td colspan="2">3</td>												3	
機 能	①・①+1 レジスタの 8 桁の BCD データを, 2 進数に変換して②・②+1 レジスタに格納します。						条件入力	処 理						出 力	
							OFF	不 実 行						OFF	
							ON	実 行				正常時		OFF	
											エラー時		ON		
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	① ①+1	演算データ					○	○	○	○	○				
	② ②+1	バイナリデータ					○		○	○	○				

プログラム例

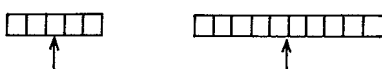


動 作



説 明

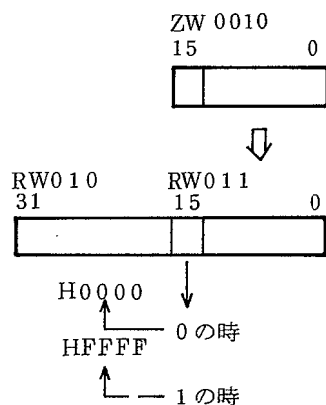
- A接点Z1000がONの時, レジスタ RW010・RW011の内容の8桁のBCDデータH12345678を2進数に変換して, RW012・RW013に格納し, 出力をOFFにします。
- 変換するデータのなかに9を超える桁があった時は, 9でリミットをかけて, 出力をONにします。
- A接点Z1000がOFFの時, 変換を行わず, 出力をOFFにします。

DW'		ダブルワード変換 (FUN241)														ステップ数	
表 現	機 能	条件入力 $\left[\textcircled{A} \quad DW' \quad \textcircled{B} \cdot \textcircled{B} + 1 \right]$ 実行出力 														3	
		①レジスタの内容（又は、数値①）の符号を延長し、倍長データとして②・②+1レジスタに格納します。															
オ ペ ラ ン ド	機 能	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
		①	被変データ					○	○	○	○	○	○	○			-32768 ~32767
		② ②+1	倍長データ					○		○	○	○					

プログラム例

$\left[\begin{array}{c} \text{ZW0010} \\ \text{Z2000} \end{array} \quad \begin{array}{c} DW' \\ 10000 \end{array} \quad \begin{array}{c} RW010 \cdot RW011 \\ 0000010000 \end{array} \right] \longrightarrow () \leftarrow R2000$

動 作



説 明

- ・ A 接点 Z2000 が ON の時、レジスタ ZW0010 の内容を RW011 に格納し、15 ビット目が 0 なので、RW010 には H0000 を格納し、出力を ON にします。
- ・ A 接点 Z2000 が OFF の時、変換は行わず、出力を OFF にします。

注 記

- ・ この命令では、入力データにレジスタ、数値のどちらでも使用可能です。

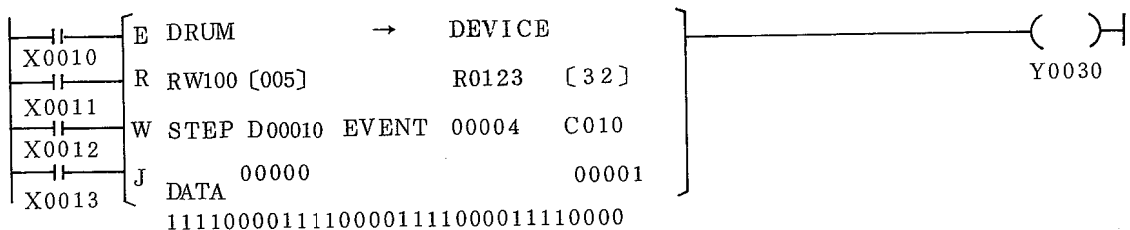
DRUM		ドラム シーケンサ (FUN246)													
------	--	--------------------	--	--	--	--	--	--	--	--	--	--	--	--	--

表 現	イベント入力 E DRUM → DEVICE 最終ステップ 実行出力 リセット入力 R ① [n n n] ④ [n n] ウェイト入力 W STEP ② EVENT □ □ □ □ □ ③ 歩進入力 J □ □ イベント数字 □ □ □ □ □ DATA □ 現在ステップ16表示 イベントカウンタ内容表示 データ表示 (2進表示)														ステップ数
	7														

機 能	ドラムシーケンス処理						条件入力		処 理						出 力	
							—		(次 頁 参 照)						最終ステップ 初回実行時ON 上記以外OFF	

オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	①	ステップデータ先頭レジスタ					○				○				
	nnn	ステップ数													1~256
	②	現在ステップ16登録レジスタ					○		○	○	○				
	③	イベントカウンタ											○		
	④	データ出力先頭デバイス	○			○	○								
	nn	デバイス数													1~32

プログラム例



説 明

RW 100				
ステップ0	H	F0F0	F0F0	データ (倍長)
1	H	FF00	FF00	
2	H	1234	5678	
3	H	8765	4321	
4	H	5555	AAAA	
ステップ0		4	イベント数	
1		3		
2		4		
3		5		
4		3		

● イベント入力 (A接点 X0010) が ON の時に、C010 を 1 づつ増加します。各ステップでイベント数を決めておいて、C010 とイベント数が等しくなったら、次のステップに歩進します。現在ステップ番号は、D00010 に格納され、現在ステップのデータが R0123 から R0142 まで 32 ビット分格納されます。この場合には現在ステップは 0 であり、C010 が 4 になるとステップ 1 に歩進します。データ HF0F0F0F0 が R0123 から格納されます。

11110000111100001111000011110000
 ↑ R0142 ↑ R0124 ↑ R0123

DRUM	ドラムシーケンサ
------	----------

動 作

各入出力は、以下のとおりです。

- (i) イベント入力 E …… '1' にてイベント入力有。イベントカウンタのカウンタアップ。
- (ii) リセット入力 R …… '1' にてリセット入力有。リセット入力にてステップ 0 にイニシャライズ。
- (iii) 歩進入力 J …… '1' にて歩進入力有。歩進入力にてイベントカウンタの内容・ウェイト
に関係なく、次ステップへ歩進。
- (iv) ウェイト入力 W …… '1' にてウェイト入力有。ウェイト入力中は、イベントカウンタがカウン
タアップしてもステップは歩進。
- (v) 出 力 …… 最終ステップへ歩進した 1 スキャンのみ '1'

各入力条件に対する処理を下に示します。

入 力				処 理
E	R	J	W	
*	1	*	*	ステップ 0 にイニシャライズ
*	0	1	*	次ステップへ歩進
1	0	0	0	イベントカウンタカウンタアップ。イベントカウンタが設定イベント数にて次ステップへ歩進。
1	0	0	1	イベントカウンタカウンタアップ。イベントカウンタが設定イベント数になっても歩進しない。
0	0	0	0	W が 1 → 0 にてイベントカウンタが設定イベント数になっているとき、次ステップへ歩進
0	0	0	1	NOP

注 記

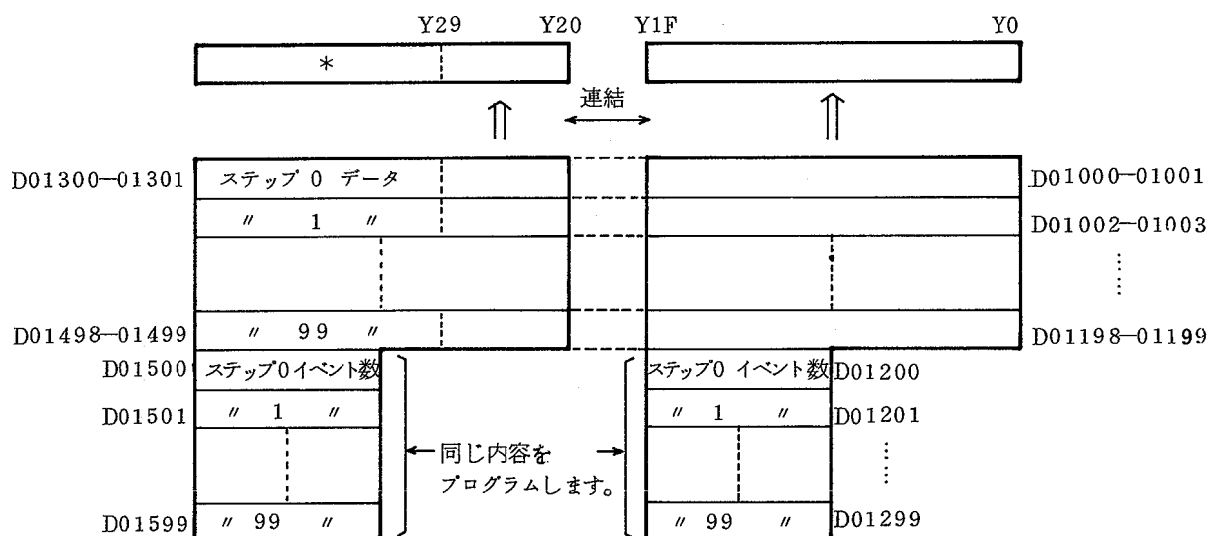
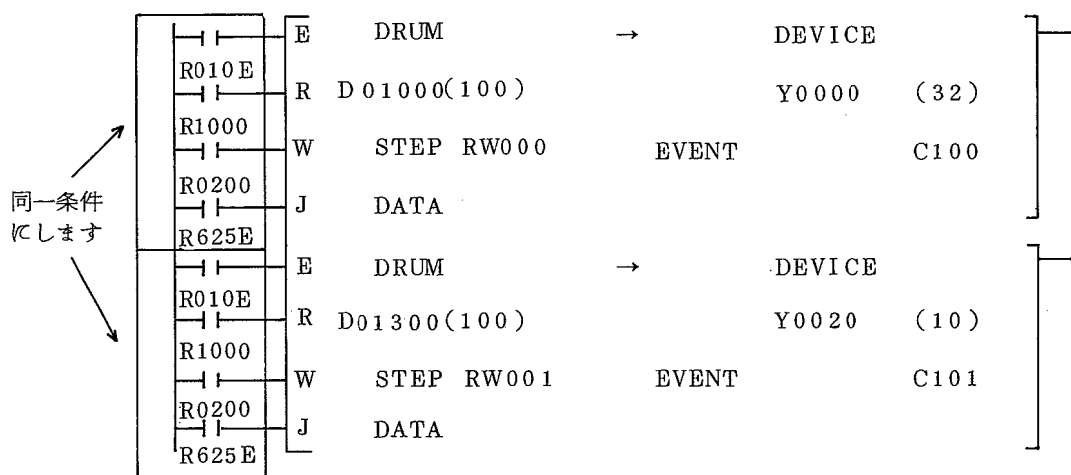
- イベント数の範囲は 1 から 32767 までです。(HFFFF は直列連結用)
- 現在ステップ *n* に指定ステップ以上の数値が入っている時は、ステップ 0 から処理を行ないます。
- イベント数に 0 が入っている場合は、1 スキャン後に次のステップに進みます。

DRUM	ドラムシーケンサ
------	----------

複数のドラム命令を連結させて、出力デバイス数、ドラムステップ数を拡張させる事ができます。

プログラム例1 並列連結

出力デバイスを32以上に拡張する場合に用いる方法です。複数のドラムシーケンサ命令の入力条件を同一にして出力デバイスを連続して指定します。



DRUM

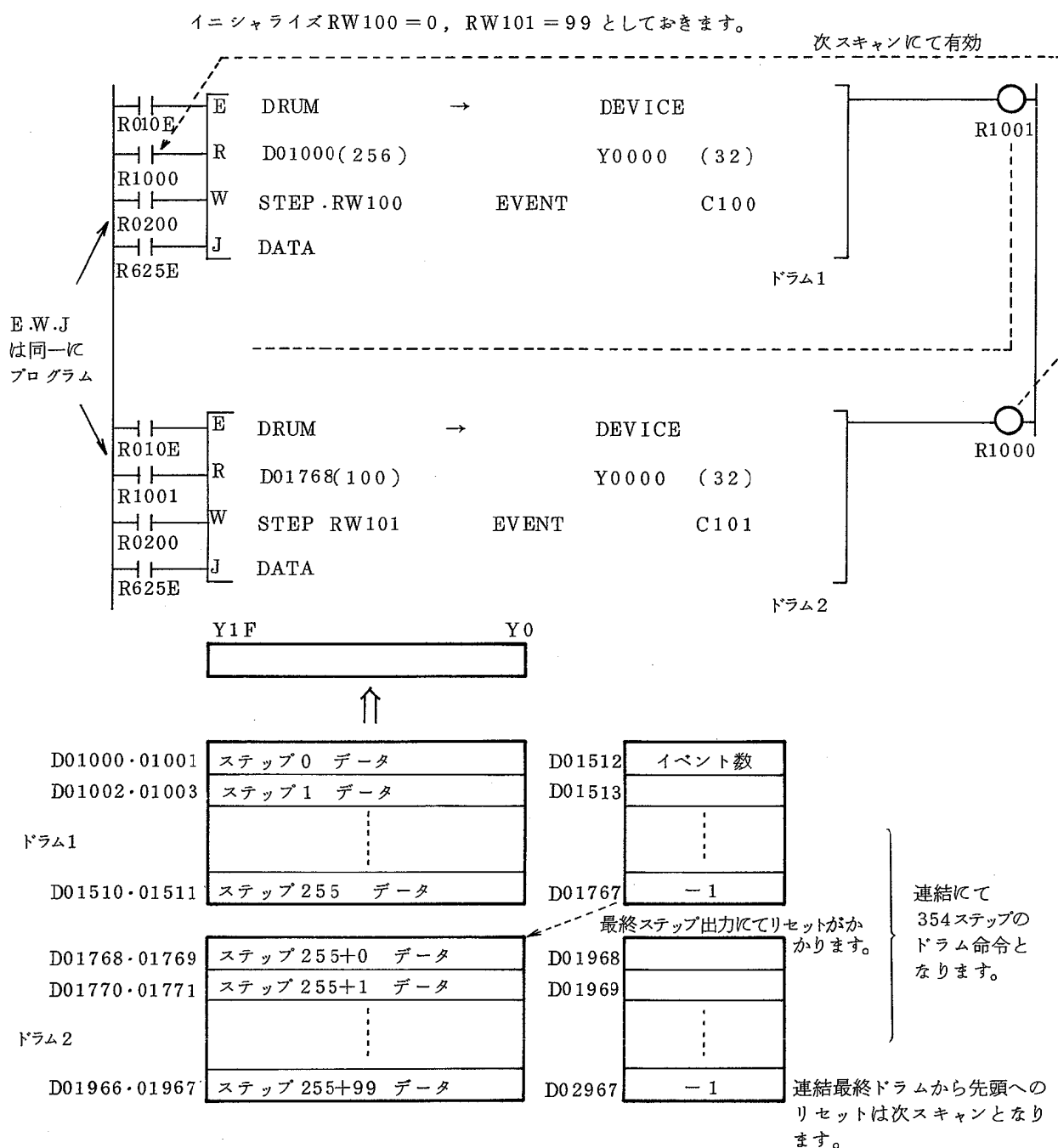
ドラムシーケンサ

プログラム例2

直 列 連 結

ドラム・ステップ数を 256 以上に拡張する場合に用いる方法です。

ドラム・シーケンス命令では、イベント数=-1(HFFFF)にプログラムされている、ステップがあるとそれ以降の歩進がストップし、出力デバイスへのデータセットも行わず、リセット入力しか受け付けなくなります。



動作は、イニシャライズにて事前に RW100=0, RW101=99 と設定しておくことにより、ドラム1はステップ0から実行し、ドラム2は、ステップ99となり、イベント数=-1でストップ状態となります。(出力なし)ドラム1がステップ255まで実行すると、イベント数=-1なのでストップ状態となり、又最終ステップへ移ったのでドラム命令出力=1となります。(R1001)これによりドラム2にリセットがかかり、ステップ0から実行し始め、以後ドラム1, 2でこの動作を交互に繰返し連結して実行されます。

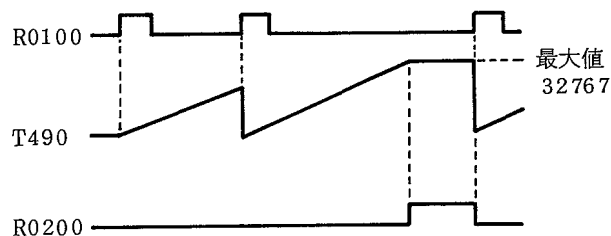
TRG	タイマトリガ (FUN247)
-----	-----------------

表 現	条件入力 TRG ① タイムオーバ出力														ステップ数				
	↑ 符号なし データ表示														2				
機 能	入力OFF→ONの変化によりタイマ値を0から32767までカウントアップします。						条件入力		処 理						出 力				
							OFF		下記参照										
							ON		#										
オ ペ ラ ン ド	記号	名 称				R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	積算タイマレジスタ													○				

プログラム例



動 作



説 明

- A接点R0100がOFFからONに変わると、T490が0からカウントアップを始めます。途中で入力がOFFになっても、カウントアップは続けます。出力は、T490がリミット値(32767)に達した時にONになります。

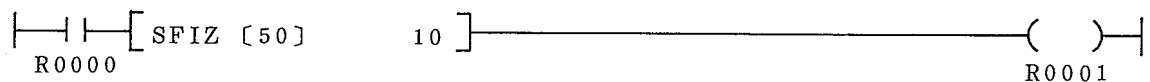
注 記

- 積算タイマ使用レジスタとしては、T490～T499が使用可能です。
(10mSタイマです)
- SFCの遷移部にはプログラムできません。

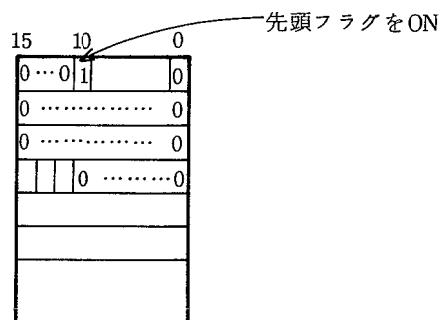
SFIZ	SFC用ステップイニシャライズ (FUN248)
------	--------------------------

表 現	条件入力 $\left[\text{SFIZ} \quad [\text{nnnn}] \quad \textcircled{\text{A}} \right]$ 実行出力														ステップ数		
															3		
機 能	ステップフラグ $\textcircled{\text{A}}$ をONにして、それ以降の $\textcircled{\text{A}} + \text{nnnn} - 1$ までのステップフラグをイニシャライズ(OFF)します。						条件入力	処 理								出 力	
							OFF	不 実 行								OFF	
							ON	実 行								ON	
オ ペ ラ ン ド	記号	名 称		R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	$\textcircled{\text{A}}$	ステップ先頭フラグ №															0000 ~ 3999
	nnnn	イニシャライズステップサイズ															0001 ~ 4000

プログラム例



動 作



説 明

- A接点R0000がONの時ステップ先頭フラグ10をONにして、それ以降の49コのフラグを全てOFFにし、出力をONにします。
- A接点R0000がOFFの時、イニシャライズは行わず、出力をOFFにします。

SEND
伝 送 送 信 (FUN250)

表 現

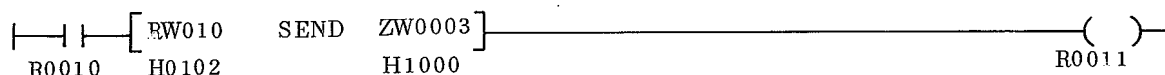
ステップ数

3

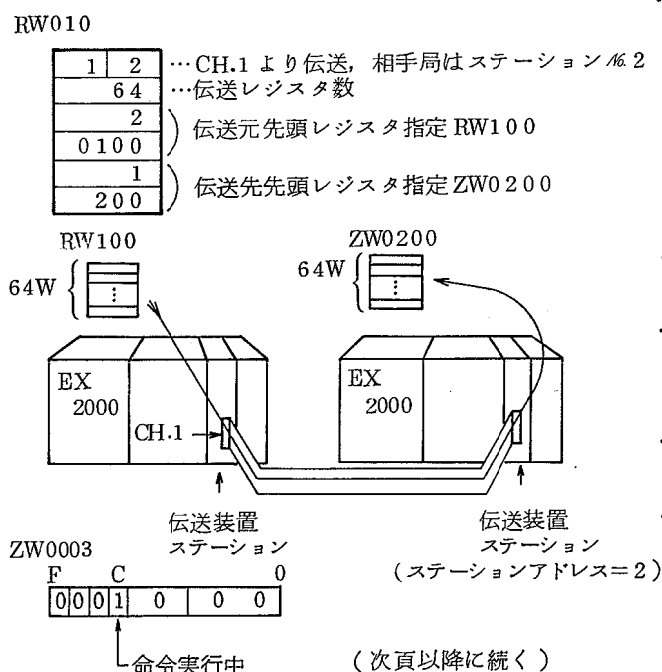
機能	伝送パラメータ①に従って、自局の連続するレジスタデータを相手局のレジスタエリアへ書込みす。							条件入力		処 理					出 力		
								OFF		不実行					OFF		
								ON		実 行					動 作 中		OFF
															動作完了		ON

オペランド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
		①	伝送パラメータ先頭レジスタ					○	○	○	○	○				
	②	完了ステータス					○		○	○						

プログラム例



動作

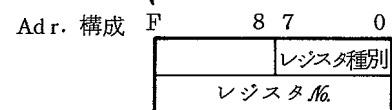
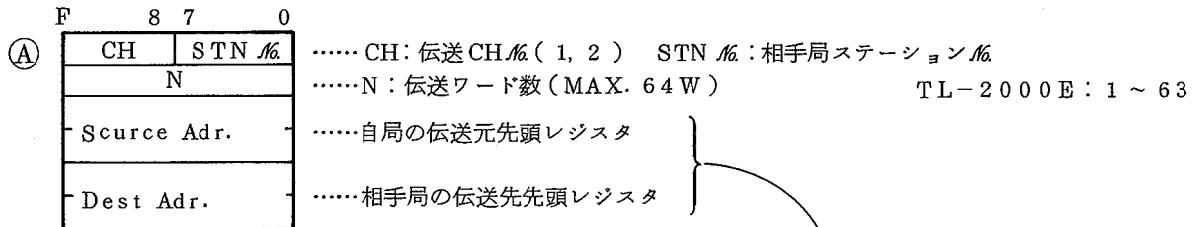


說明

- ・ A 接点 R0010 が ON の時，レジスタ RW010 から はじまる 6 W の伝送パラメータに従って，データの 伝送を行います。この場合は，自局の RW100 から 6 4 W のデータを CH. 1 の伝送装置からステーション アドレス 2 の伝送装置の ZW0200 以降に伝送を行 います。このとき，ステータスには，動作中のフラ グをセットし，出力は OFF のままです。
- ・ 6 4 W の伝送が完了すると，完了ステータスをセッ トし，出力は ON にします。
- ・ 伝送パラメータ，又は，伝送に異常があった場合， 要因を完了ステータスにセットし，出力を ON にし ます。
- ・ A 接点 R0010 が OFF の時，伝送は行わず出力を OFF にします。
- ・ 伝送装置は，TOSLINE-2000E。

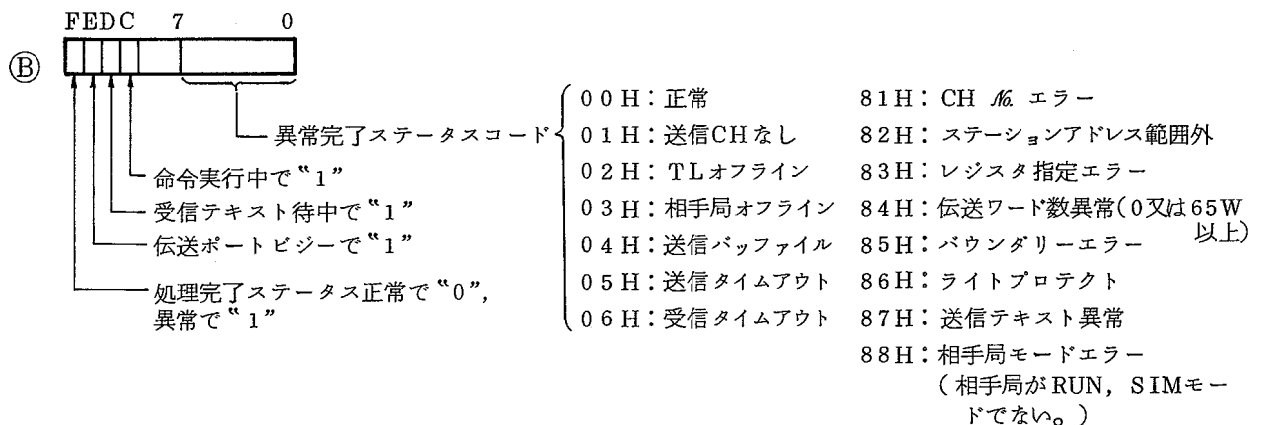
SEND	伝 送 送 信
------	---------

• 伝送パラメータ

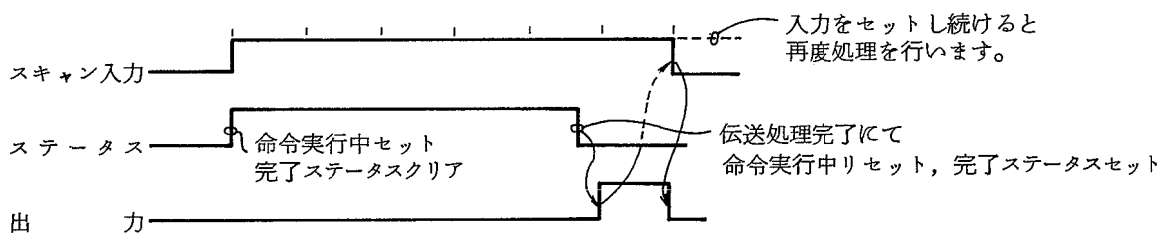


レジスタ種別……0: XW / YW
 1: ZW
 2: RW
 3: D

• 完了ステータス



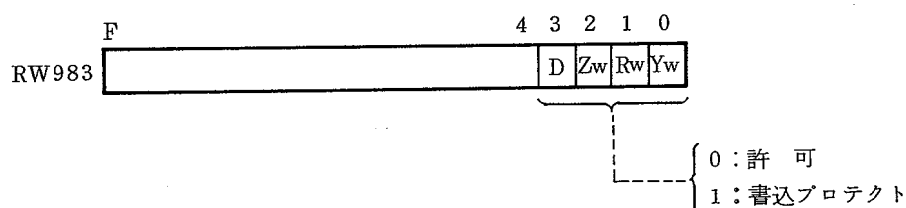
• 動作タイミング



SEND	伝 送 送 信
------	---------

注 記

- SEND 命令起動後，完了出力が ON になるまで条件入力，ON 状態を保持して下さい。
完了出力が ON する前に条件入力を OFF にすると，それ以降の伝送命令処理が正常に実行されなくなります。
- 伝送パラメータの伝送元，伝送先にてワード数がエリアを越える場合は，エラー完了になります。（バウンダリエラー）
- 伝送先へのレジスタデータ書込みは，伝送データ受信後，即座に行うのでスキャンとは同期しません。
- 特殊コイル設定にて，SEND 命令によるレジスタ書込みをプロテクトにすることができます。



- 伝送ポートビジー（他の命令にて既に伝送要求を行っている。）場合，条件入力を ON し続けることにより他の伝送命令の処理が終了ししだい，命令処理が行われます。これにより伝送ポートビジーにても再要求をかける必要がありません。
- 完了ステータスとして使用するレジスタは，停電記憶指定を行わないで下さい。停電記憶指定を行う場合は，起動初回スキャンで完了ステータスレジスタをクリアして下さい。

RECV		伝 送 受 信 (FUN251)																		
表 現	条件入力 ① RECV ② 完了出力 H H データ表示																ステップ数			
																	3			
機 能	伝送パラメータ①に従って、相手局の連続するレジスタデータを読み込み、自局のレジスタエリアへ格納します。										条件入力		処 理						出 力	
											OFF		不 実 行						OFF	
											ON		実 行		動 作 中		OFF			
												動 作 完 了						ON		
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	伝送パラメータ先頭レジスタ									○	○	○	○	○					
	②	完了ステータス									○		○	○						

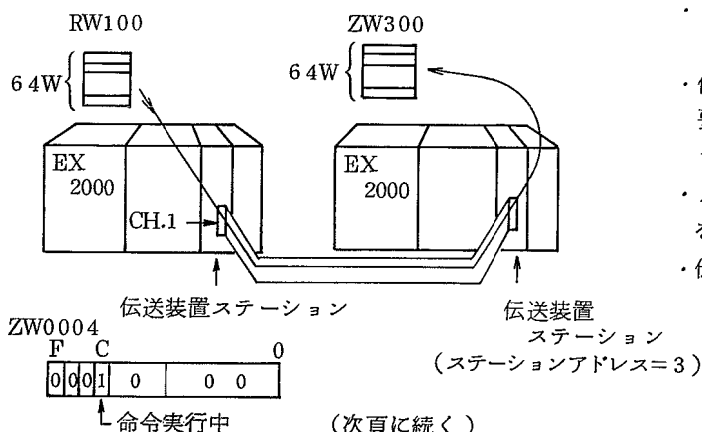
プログラム例



動 作

RW020

1	3	…CH.1より伝送、相手局はステーション#3
64		…伝送レジスタ数
2		伝送先頭レジスタ指定 RW100
100		
2		伝送元先頭レジスタ指定 ZW300
300		



説 明

- ・ A接点R0020がONの時、レジスタRW020からはじまる6Wの伝送パラメータに従って、データの伝送を行います。この場合は、ステーションアドレス3の伝送装置のZW300から64Wのデータを、伝送装置のCH.1からRW100以降へ伝送を行います。このとき、ステータスには動作中のフラグをセットし、出力はOFFのままです。
- ・ 64Wの伝送が完了すると、完了ステータスをセットし、出力はONになります。
- ・ 伝送パラメータ、又は、伝送に異常があった場合、要因を完了ステータスにセットし、出力をONにします。
- ・ A接点R0020がOFFの時、伝送は行わず、出力をOFFにします。
- ・ 伝送装置はTOSLINE-2000E。

(次頁に続く)

RECV	伝 送 受 信
------	---------

・ 伝送パラメータ

④	F 87 0		
	CH	STN NO.	CH:伝送CH №(1, 2) STN №:相手局ステーション №(1 ~ 63)
	N		N:伝送ワード数(MAX. 64 W)
	Dest. Adr.		自局の伝送先先頭レジスタ
	Source Adr.		相手局の伝送元先頭レジスタ

・ 完了ステータス

⑤	F 0

※ SEND 命令を参照して下さい。

(但しエラーコード 86 H ライトプロテクトはありません。)

88 H 相手局モードエラー

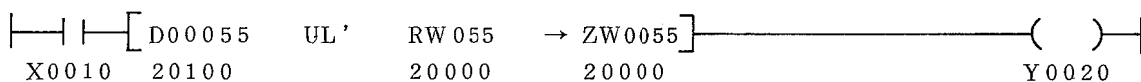
・ 動作タイミング..... SEND 命令を参照して下さい。

注 記

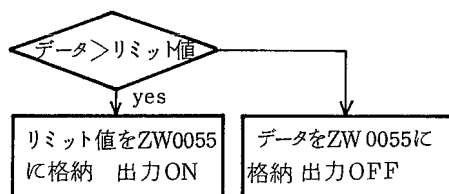
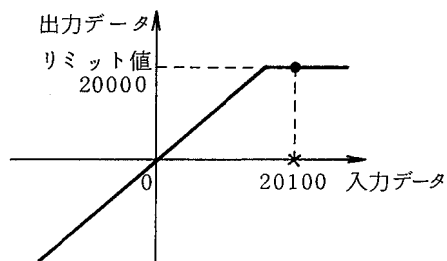
- ・ RECV 命令起動後、完了出力が ON になるまで条件入力、ON 状態を保持して下さい。
完了出力が ON する前に条件入力を OFF にすると、それ以降の伝送命令処理が正常に実行されなくなります。
- ・ 伝送パラメータの伝送元、伝送先にてワード数がエリアを超える場合は、エラー完了になります。(パウンダリエラー)
- ・ 伝送先へのレジスタデータの読み込みは、伝送データ送信後、即座に行うのでスキャンとは同期しません。
- ・ 伝送ポートビジー(他の命令にて既に伝送要求を行っている。)の場合、条件入力を ON し続けることにより他の伝送命令の処理が終了ししだい、命令処理が行われます。
- ・ 完了ステータスとして使用するレジスタは、停電記憶指定を行わないで下さい。停電記憶指定を行う場合は、起動初回スキャンで完了ステータスレジスタをクリアして下さい。

[illegible]

プログラム例



動作



說明

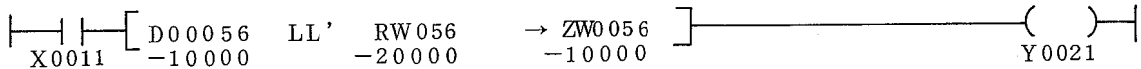
- A接点X0010がONの時、レジスタD00055の内容20100を、リミット値RW055の内容20000で上限リミットして20000をZW055に格納し、出力をONにします。
- A接点X0010がOFFの時、演算は行わず出力をOFFにします。

注 記

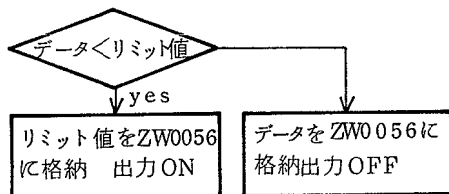
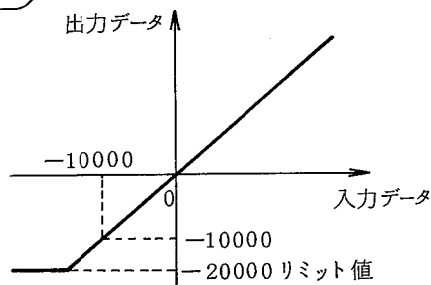
- この命令では、入力データ、リミット値にレジスタ、数値のどちらでも使用可能です。

LL'		符号付下限リミット (FUN261)														ステップ数	
表 現	機 能	条件入力 ① LL' ② → ③ リミットオーバー判定出力														4	
		データ①をデータ②で下限リミットし、③に格納します。 符号付データ表示															
オペ ラ ン ド	機 能	条件入力 処 理 出 力															
		OFF 不 実 行 OFF															
		ON 実 行 ① ≥ ② の時 OFF															
		ON ① < ② の時 ON															
		記号 名 称 R X Y Z RW XW YW ZW D T C T. C. 数 値															
オペ ラ ン ド	機 能	①	演算データ					○	○	○	○	○	○	○			-32768 ~32767
		②	下限リミット値					○	○	○	○	○					-32768 ~32767
		③	リミット結果					○		○	○	○					

プログラム例



動 作



注 記

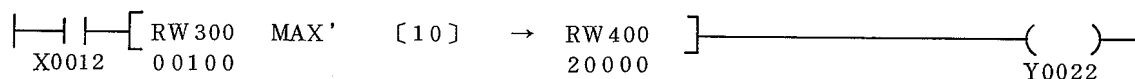
- この命令では、入力データ、リミット値にレジスタ、数値のどちらでも使用可能です。

説 明

- A接点X0011がONの時、レジスタD00056の内容-10000は、リミット値RW056の内容-20000よりも大きいので、そのまま出力データとしてZW0056に格納し出力をOFFにします。
- A接点X0011がOFFの時、演算は行わず出力をOFFにします。

MAX'		符号付最大値 (FUN262)																		
表 現	条件入力 ① MAX' [nn] ② 実行出力 符号付 データ表示															ステップ数				
																4				
機 能	レジスタ①からデータnnで指定されたテーブルサイズ分のテーブル中で最大の値をレジスタ②に、そのポインタをレジスタ②+1に格納します。						条件入力	処 理								出 力				
							OFF	不 実 行								OFF				
							ON	実 行								ON				
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	テーブル先頭レジスタ									○	○	○	○	○	○				
	nn	テーブルサイズ																		1 ~ 64
	②	最 大 値									○		○	○	○					
	②+1	最大値ポインタ																		

プログラム例



動 作

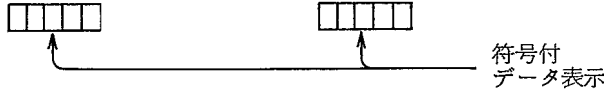
RW300	ポインタ
100	0
10000	1
-10000	2
10	3
0	4
200	5
-300	6
20000	7
-30	8
20	9

RW400
20000
最大値

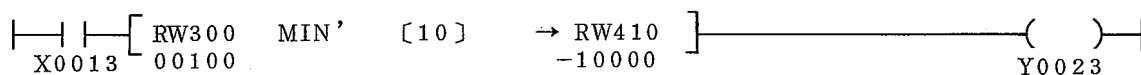
RW401
7
ポインタ

説 明

- A接点X0012がONの時、レジスタRW300を先頭とする10ワードのテーブルのなかから、最大値をさがして、最大値20000をRW400にポインタ7をRW401に格納し、出力をONにします。
- 最大値が複数個あった場合には、ポインタは若い番号の方を格納します。
- A接点X0012がOFFの時、演算は行わず出力をOFFにします。

MIN'		符号付最小値 (FUN263)												ステップ数			
表 現	条件 入力	[① MIN' [nn] ②] 実行出力												4			
																	
機 能	レジスタ①からデータ nn で指定されたテーブルサイズ分のテーブル中で、最小の値をレジスタ②に、そのポインタをレジスタ②+1に格納します。						条件入力	処 理								出 力	
							OFF	不 実 行								OFF	
							ON	実 行								ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	テーブル先頭レジスタ					○	○	○	○	○	○					
	nn	テーブルサイズ														1 ~ 64	
	②	最 小 値					○		○	○	○						
	②+1	最小値ポインタ															

プログラム例



動 作

RW300	ポインタ
100	0
10000	1
-10000	2
-10	3
0	4
200	5
-300	6
20000	7
-30	8
20	9

RW410
-10000
最小値

RW411
2
ポインタ

説 明

- A接点X0013がONの時、レジスタRW300を先頭とする10ワードのテーブルのなかから、最小値をさがして、最小値-10000をRW410にポインタ2をRW411に格納し、出力をONにします。
- 最小値が複数個あった場合には、ポインタは若い番号の方を格納します。
- A接点X0013がOFFの時演算は行わず出力をOFFにします。

AVE'		符号付平均値 (FUN264)												
------	--	-----------------	--	--	--	--	--	--	--	--	--	--	--	--

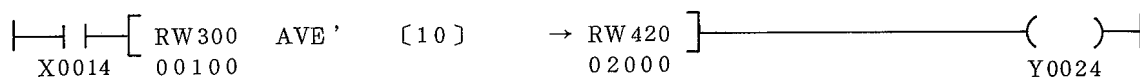
表現

ステップ数	
4	

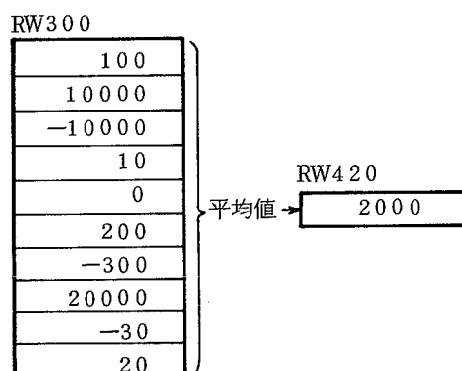
機能	レジスタ①からデータ nn で指定されるテーブルサイズ分のテーブルから平均値を求めてレジスタ②に格納します。	条件入力	処 理	出 力
		OFF	不 実 行	OFF
		ON	実 行	ON

オペランド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値
	①	テーブル先頭レジスタ					○	○	○	○	○	○			
	nn	テーブルサイズ													1 ~ 64
	②	平 均 値					○		○	○	○				

プログラム例



動作



說明

- A 接点 X0014 が ON の時、レジスタ RW300 を先頭とする 10 ワードのテーブルの平均値 2000 を求めて、RW420 に格納し、出力を ON にします。
- A 接点 X0014 が OFF の時、演算は行わず出力を OFF にします。

FG'		符号付関数発生器 (FUN265)												ステップ数		
表 現	機 能	条件入力 [① FG' [nn] ② → ③] 実行出力 符号付データ表示														5
		レジスタ②から始まる2×[nn]テーブルのデータにより発生する関数に対応するデータ①の演算値を③に格納します。														
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	入力データ					○	○	○	○	○	○				-32768 ~32767
	nn	関数データ数														1 ~ 32
	②	関数データテーブル 先頭レジスタ					○	○	○	○	○					
	③	出力データ					○		○	○	○					

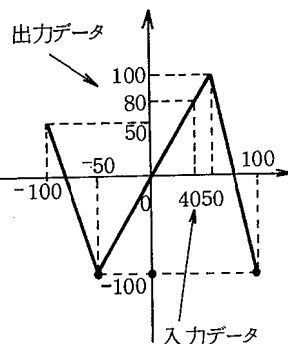
プログラム例

$\left[\begin{array}{l} \text{D00100} \\ \text{X0015} \end{array} \right] \text{FG' [05]} \left[\begin{array}{l} \text{D00200} \\ \text{00040} \end{array} \right] \rightarrow \left[\begin{array}{l} \text{RW111} \\ \text{00080} \end{array} \right] \text{Y0025}$

動 作

D00200

-100	X ₀
- 50	X ₁
0	X ₂
50	X ₃
100	X ₄
50	Y ₀
-100	Y ₁
0	Y ₂
100	Y ₃
-100	Y ₄



説 明

- A接点X0015がONの時下記演算を行ない出力をONにします。
 - $X_{n-1} \leq \text{入力データ} \leq X_n$ の場合

$$\text{出力データ} = Y_{n-1} + \frac{(Y_n - Y_{n-1}) \times (\text{入力データ} - X_{n-1})}{X_n - X_{n-1}}$$
 - 入力データ $\leq X_0$ の場合

$$\text{出力データ} = Y_0$$
 - 入力データ $\geq X_n$ の場合

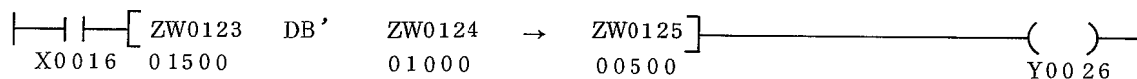
$$\text{出力データ} = Y_n$$
- A接点X0015がOFFの時、演算を行わず出力をOFFにします。

注 記

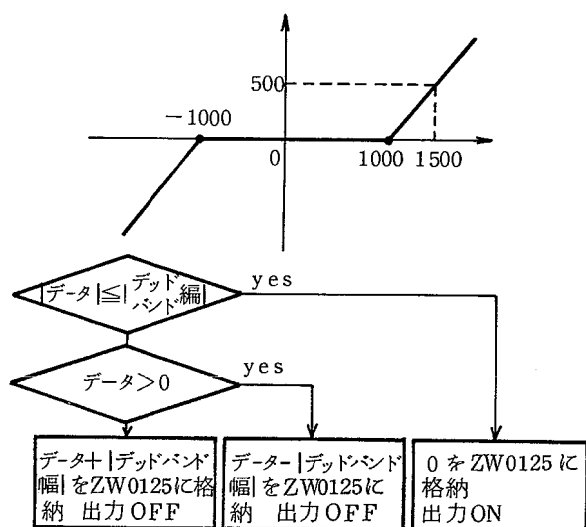
- $X_0 \leq X_1 \leq X_2 \dots \leq X_{n-1}$ を前程としているため $X_{i-1} > X_i$ の時は X_i, Y_i を無視します。
- Yは $Y_{i-1} \leq Y_i$ であっても $Y_{i-1} > Y_i$ でもかまいません。

[illegible]

プログラム例



動作



說明

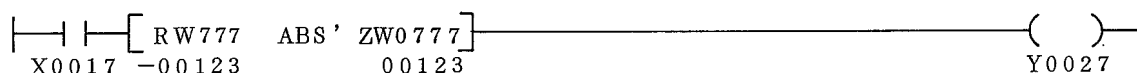
- A 接点X0016 が ON の時下記演算を実行します。
 - $| \text{入力データ} | \leq | \text{デッドバンド幅} |$ の場合
 $0 \rightarrow$ 演算結果, 出力ON
 - $| \text{入力データ} | > | \text{デッドバンド幅} |$
 - 1) $\text{入力データ} < 0$ の場合
 $\text{入力データ} + | \text{デッドバンド幅} | \rightarrow$ 演算結果
 出力OFF
 - 2) $\text{入力データ} > 0$ の場合
 $\text{入力データ} - | \text{デッドバンド幅} | \rightarrow$ 演算結果
 出力OFF
- A 接点X0016 が OFF の時演算は行わず出力を OFF にします。

注 記

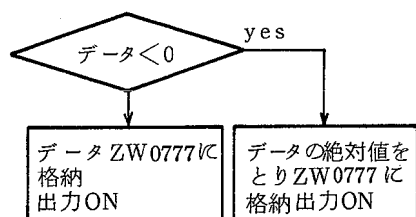
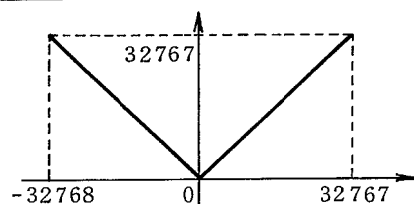
- この命令では、入力データ、デッドバンド幅にレジスタ、数値のどちらでも使用可能です。

ABS'		絶 対 値 (FUN267)														
表 現	条件入力 [① ABS' ②] 実行出力 符号付データ表示														ステップ数	
															3	
機 能	データ①の絶対値をレジスタ②に格納します。						条件入力		処 理						出 力	
							OFF		不 実 行						OFF	
							ON		実 行						ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値	
	①	被演算データ					○	○	○	○	○	○			-32768 ~32767	
	②	絶対値格納レジスタ					○		○	○	○					

プログラム例



動 作



注 記

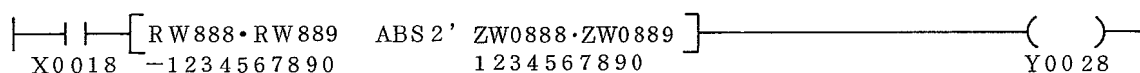
- この命令では、入力データにレジスタ、数値のどちらでも使用可能です。

説 明

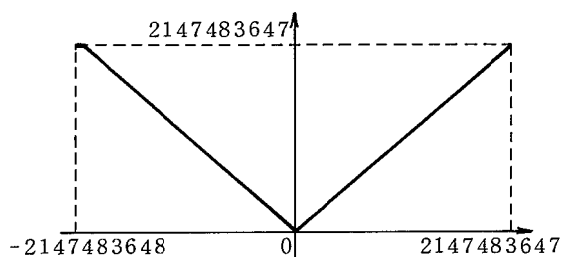
- A接点X0017がONの時、レジスタRW777の内容-123の絶対値123をZW0777に格納し、出力をONにします。
- 入力データが-32768の時は32767を絶対値として格納します。
- A接点X0017がOFFの時、演算を行わず出力をOFFにします。

ABS 2'		倍 長 絶 対 値 (FUN268)														
表 現	条件 入力 ④ ・ ④+1 ABS 2' ④ ・ ④+1 実行出力 符号付 データ表示	ステップ数														
		R ABS2R 3														
		K ABS2R 4														
機 能	倍長データ④・④+1の絶対値をレジスタ④・④+1に格納します。						条件入力		処 理						出 力	
							OFF		不 実 行						OFF	
							ON		実 行						ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	④ ④+1	被演算データ					○	○	○	○	○	○				-2147483648 ~2147483647
	④ ④+1	絶対値格納レジスタ					○		○	○	○					

プログラム例

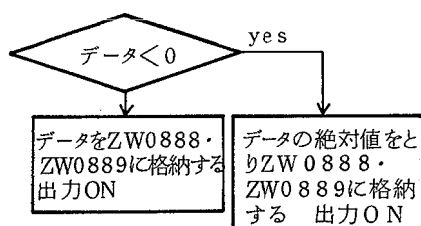


動 作



説 明

- A接点X0018がONの時、レジスタRW888・RW889の内容-1234567890の絶対値1234567890をZW0888・ZW0889に格納し、出力をONにします。
- 入力データが-2147483648の時は、2147483647を絶対値として格納します。
- A接点X0018がOFFの時演算を行わず出力をOFFにします。

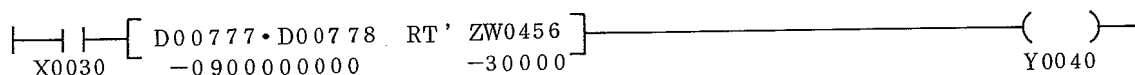


注 記

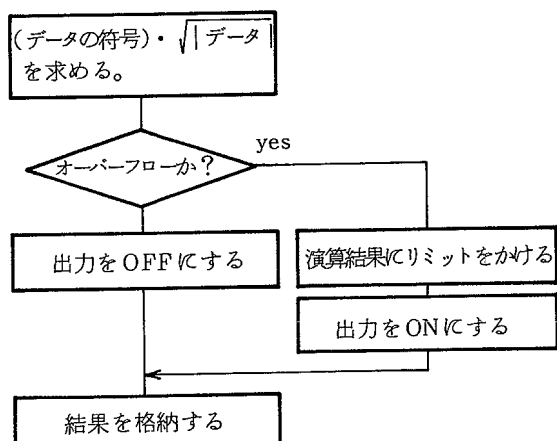
- この命令では、入力データにレジスタ、数値のどちらでも使用可能です。

RT'		符号付開平 (FUN270)													
表 現	機 能	条件入力 $\left[\textcircled{A} \cdot \textcircled{A} + 1 \quad \text{RT}' \quad \textcircled{B} \right]$ オーバーフロー出力 												ステップ数	
														R RT R 3	
														K RT R 4	
オ ペ ラ ン ド	機 能	データ $\textcircled{A} \cdot \textcircled{A} + 1$ の平方根を，レジスタ $\textcircled{B}$ に格納します。												出力	
		条件入力 OFF 処 理 不 実 行 出 力 OFF													
		ON 実 行 正 常 時 オーバフロー時 OFF ON													
記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
						○	○	○	○	○	○				-2147483648 ~2147483647
						○		○	○	○					

プログラム例



動 作



注 記

- この命令では，入力データにレジスタ，数値のどちらでも使用可能です。

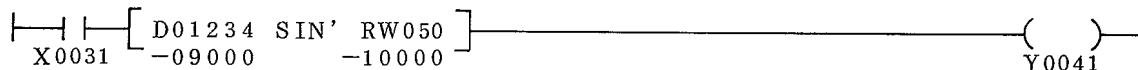
説 明

- A接点X0030がONの時，レジスタD00777・D00778の内容-9000000000の絶対値の開平を求め，符号を負にした-30000を結果としてZW0456に格納し，出力をOFFにします。
- A接点X0030がOFFの時，演算は行わず出力をOFFにします。

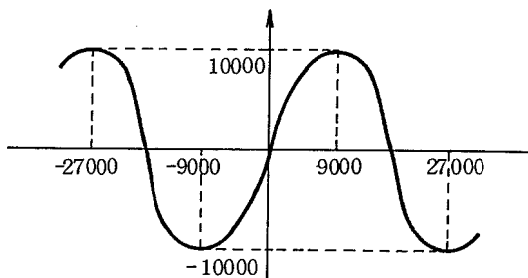
SIN'	符号付正弦関数 (FUN271)
------	------------------

表 現	条件入力 ① SIN' ② 実行出力 符号付データ表示														ステップ数					
															3					
機 能	データ①の内容の1/100を角度(deg)とする正弦の値の10000倍をレジスタ②に格納します。						条件入力		処 理						出 力					
							OFF		不 実 行						OFF					
							ON		実 行						ON					
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	角度入力									○	○	○	○	○	○	○			-32768 ~32767
	②	正弦関数値									○		○	○	○					

プログラム例



動 作



$$\text{出力データ} = 10000 \times \sin\left(\frac{\text{入力データ}}{100}\right)^{\circ}$$

注 記

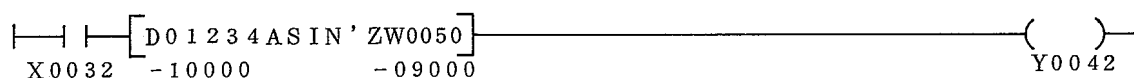
- この命令では、入力データにレジスタ、数値のどちらでも使用可能です。
- 相対誤差は、±0.8%以下です。

説 明

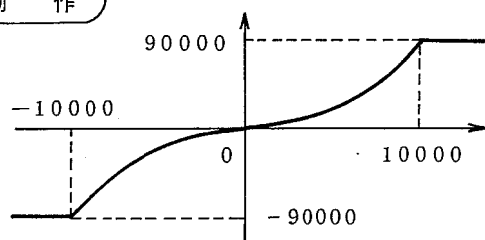
- A接点X0031がONの時、レジスタD01234の内容-9000の $\frac{1}{100}$ を角度とするSINの値を求め、それを10000倍してレジスタRW050に格納し、出力をONにします。
- A接点X0031がOFFの時、演算は行わず出力をOFFにします。

ASIN'		符号付逆正弦関数 (FUN272)														ステップ数	
表 現	条件 入力	[① ASIN' ②] 実行出力														3	
		符号付 データ表示															
機 能	データ①の内容の1/10000の逆正弦値 (deg)を100倍してレジスタ②に格納し ます。							条件入力		処 理						出 力	
								OFF		不実行						OFF	
								ON		実 行						ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	入力データ					○	○	○	○	○	○				-32768 ~32767	
	②	逆正弦関数値					○		○	○	○						

プログラム例



動 作



$$\text{出力データ} = 100 \times \sin^{-1} \left(\frac{\text{入力データ}}{10000} \right)$$

ただし

入力データ > 10000 の時

出力データ = 9000

入力データ < -10000 の時

出力データ = -9000

注 記

- この命令では、入力データにレジスタ、数値のどちらでも使用可能です。
- 相対誤差は±1%以下です。

説 明

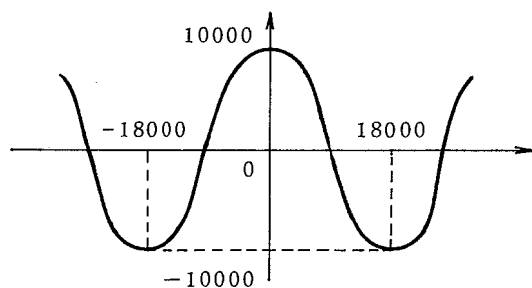
- A接点 X0032 がONの時レジスタ D01234 の内容 -10000 の $\frac{1}{10000}$ の $\sin^{-1}$ を求め、それを 100 倍してレジスタ ZW0050 に格納し、出力を ON にします。
- A接点 X0032 がOFFの時演算は行わず 出力を OFF にします。

[illegible]

プログラム例



動作



$$\text{出力データ} = 10000 \times \cos\left(\frac{\text{入力データ}}{100}\right)^\circ$$

說明

- A 接点 X0032 が ON の時レジスタ D 01234 の内容 18000 の $\frac{1}{100}$ を角度とする COS の値を求め、それを 10000 倍してレジスタ RW051 に格納し、出力を ON にします。
- A 接点 X0032 が OFF の時、演算は行わず出力を OFF にします。

注 記

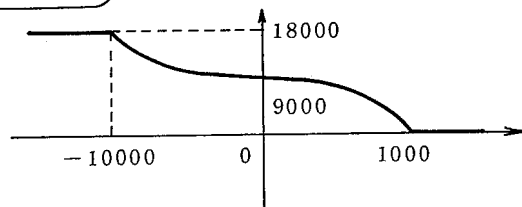
- この命令では、入力データにレジスタ、数値のどちらでも使用可能です。
- 相対誤差は、 $\pm 0.8\%$ 以下です。

ACOS'		符号付逆余弦関数										(FUN274)									
表 現	条件 入力	① ACOS' ② 実行出力 符号付 データ表示														ステップ数					
		3																			
機 能	データ①の内容の1/10000の逆余弦値 (deg)を100倍してレジスタ②に格納し ます。							条件入力		処 理							出 力				
								OFF		不 実 行							OFF				
								ON		実 行							ON				
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	入力データ									○	○	○	○	○	○			-32768 ~32767		
	②	逆余弦関数									○		○	○	○						

プログラム例



動 作



$$\text{出力データ} = 100 \times \cos^{-1} \left(\frac{\text{入力データ}}{10000} \right)$$

ただし,

入力データ > 10000 の時

出力データ = 0

入力データ < -10000 の時

出力データ = 18000

注 記

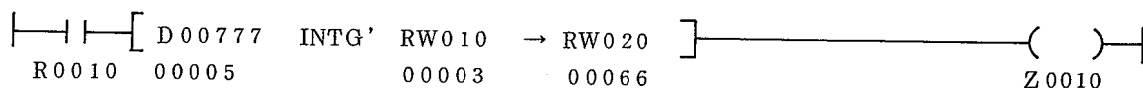
- この命令では入力データに, レジスタ数値のどちらでも使用可能です。
- 相対誤差は±1%以下です。

説 明

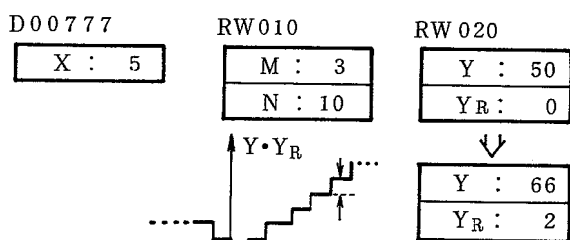
- A接点 X0033 が ON の時 レジスタ D01234 の内容 -10000 の $\frac{1}{10000}$ の $\cos^{-1}$ を求め, それを100倍してレジスタZW0051に格納し, 出力をONにします。
- A接点 X0033 が OFF の時, 演算は行わず 出力をOFFにします。

INTG'		積 分 (FUN275)														
表 現	条件 入力	① INTG' ② → ③ 実行出力 符号付 データ表示														ステップ数
																4
機 能	入力値①に積分定数②を演算し、レジスタ③に積分します。							条件入力	処 理							出 力
								OFF	不 実 行							OFF
								ON	M≠0の時、正常実行							ON
							M=0の時、エラー							OFF		
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	入 力 値					○	○	○	○	○					
	②	積分定数先頭レジスタ					○	○	○	○	○					
	③	積 分 値					○		○	○	○					

（プログラム例）

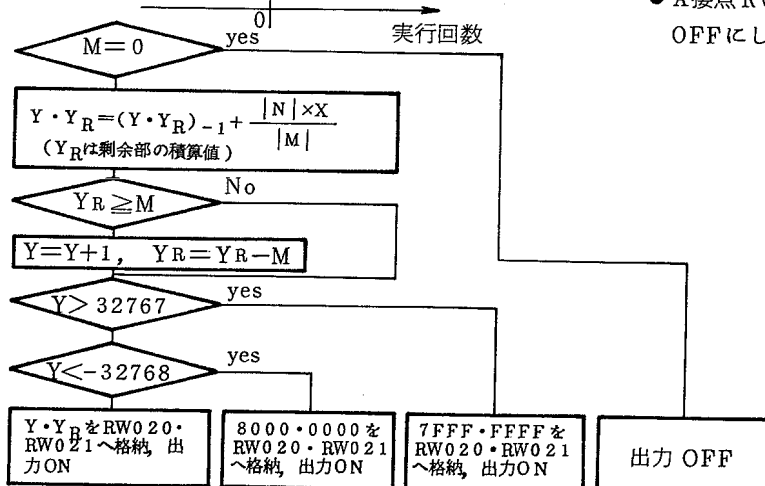


動作



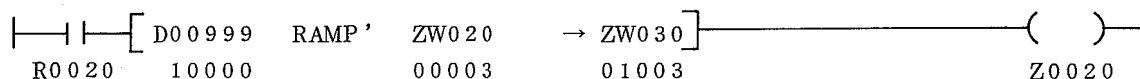
説 明

- A 接点 R0010 が ON の時、レジスタ D00777 の内容 5，M：3 と N：10 を使って積分を行います。結果の 66 と 2 を RW020，RW021 に格納し出力を ON にします。
- M が 0 の時には，実行は行わず出力を OFF にします。
- A 接点 R0010 が OFF の時，演算は行わず出力を OFF にします。

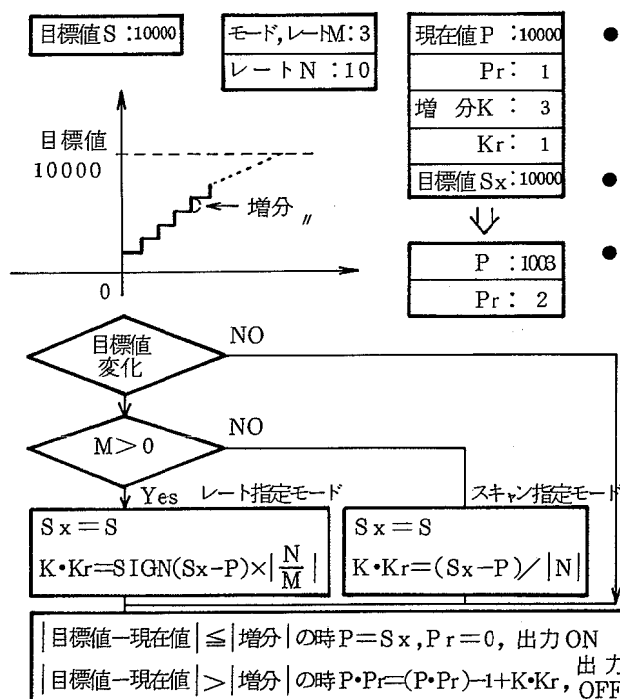


RAMP'		ラ ン プ 関 数 (FUN276)														ステップ数	
表 現	条件 入力	① RAMP' ② → ③ 完了出力 符号付 データ表示														4	
		ランプ関数を Ⅰ.レート指定モード Ⅱ.スキャン指定モード のいずれかで実行します。															
機 能	条件入力	処 理										出 力					
	OFF	不 実 行										OFF					
能	ON	実 行										未 完 了		OFF			
												完 了		ON			
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	目 標 値					○	○	○	○	○						
	②	モード指定レジスタ					○	○	○	○	○						
	③	現在値格納ポインタ					○		○	○	○						

プログラム例



動 作

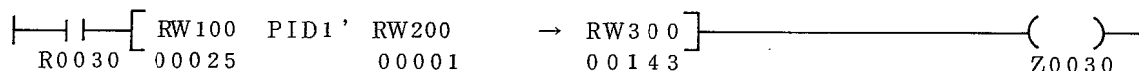


説 明

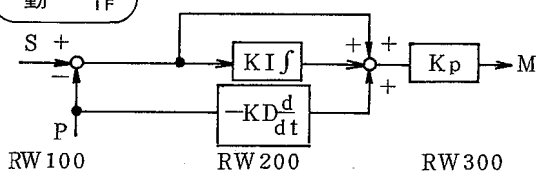
- A接点 R0020 が ON の時、レジスタ D00999 の内容 10000 を目標値として現在値 P : 1000 に増分 K : 3 を、Pr : 1 に Kr : 1 を加えて、出力を OFF にします。
- この関数をくり返し行い、現在値が目標値に達したら、出力を ON にします。
- 目標値が変わったら、増分を計算しなおして演算を実行します。
- A接点 R0020 が OFF の時、演算は行わず出力を OFF にします。

PID1'		速度型PID (FUN277)															
表現	条件入力	① PID1' ② → ③ 実行出力 符号付データ表示														ステップ数	
		4															
機能	微分先行速度型PID演算を行います。							条件入力	処 理						出 力		
								OFF	不実行(下記参照)						OFF		
								ON	実 行						ON		
オペランド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	①	入力データ先頭レジスタ					○	○	○	○	○						
	②	パラメータ先頭レジスタ					○	○	○	○	○						
	③	出力データ先頭レジスタ					○		○	○	○						

プログラム例



動 作



入力 P: 25
設定値 S: 100

比例係数 Kp: 1
積分係数 KIH: 5
KIL: 10
微分係数 KDH: 20
KDL: 5
ギャップ定数 G: 0
リミット定数 EL: 100

出力データ ΔM: -
前回入力偏差 e-1: 78
前回入力 P-1: 22
前前回入力 P-2: 20

ΔM: 143
e-1: 75
P-1: 25
P-2: 22

説 明

● A接点 R0030 が ON の時、入力値 RW100 の内容を使って速度型 PID 演算を行い出力を ON にします。

● A接点 R0030 が OFF の時、演算は行わず、

$$e-1 = e$$

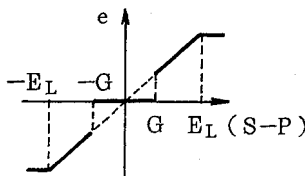
$$P-2 = P-1 = P$$

$$\Delta M = 0$$

にして出力を OFF にします。

$$\Delta M = K_p \left\{ (e - e_{-1}) + \frac{1}{\left| \frac{K_{IH}}{K_{IL}} \right|} \cdot e + \left| \frac{K_{DH}}{K_{DL}} \right| (2P_{-1} - P - P_{-2}) \right\}$$

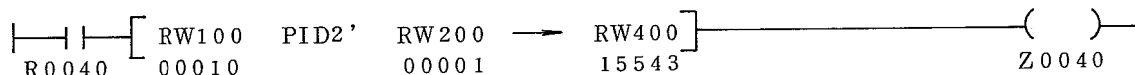
ここで e は (S-P) にリミット,
Gap 演算を施したものです。
(右図参照)



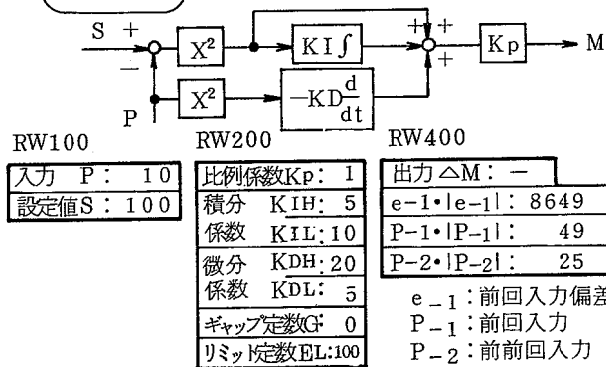
* 詳細は補足説明を参照して下さい。

PID2'		偏差 2 乗型 PID (FUN278)													ステップ数	
表 現	条件 入力	① PID2' ② → ③ 実行出力 符号付 データ表示													4	
機 能	偏差 2 乗型 PID 演算を行います。						条件入力		処 理						出 力	
							OFF		不実行 (下記参照)						OFF	
							ON		実 行						ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	①	入力データ先頭レジスタ					○	○	○	○	○					
	②	パラメータ "					○	○	○	○	○					
	③	出力データ "					○		○	○	○					

プログラム例



動 作



説 明

● A 接点 R0040 が ON の時、入力値 RW100 の内容を使って、偏差 2 乗型 PID 演算を行い、出力を ON にします。

● A 接点 R0040 が OFF の時、演算は行わず、

$$e-1 \cdot |e-1| = e \cdot |e|$$

$$P-2 \cdot |P-2| = P-1 \cdot |P-1| = P \cdot |P|$$

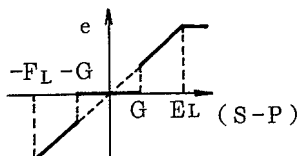
$$\Delta M = 0$$

にして出力を OFF にします。

ΔM: 15543
e-1・ e-1 : 8100 (倍長)
P-1・ P-1 : 100 (倍長)
P-2・ P-2 : 49 (倍長)

$$\Delta M = K_p \left\{ (e \cdot |e| - e-1 \cdot |e-1|) + \left[\frac{K_{IH}}{K_{IL}} \right] e \cdot |e| + \left[\frac{K_{DH}}{K_{DL}} \right] \cdot (2 \times P-1 \cdot |P-1| - P \cdot |P| - P-2 \cdot |P-2|) \right\}$$

ここで e は (S-P) にリミット Gap 演算を施したものです。
(右図参照)



* 詳細は補足説明を参照して下さい。

PID	速度型PID〔FUN277〕, 偏差2乗型PID〔FUN278〕
-----	----------------------------------

補足説明

一般にPID演算は式(1)の様に表します。

$$\Delta M_v = K_p \{ (e_n - e_{n-1}) \} + \frac{\Delta T}{T_I} e_n - \frac{T_D}{\Delta T} (PV_n - 2PV_{n-1} + PV_{n-2}) \} \dots\dots\dots (1)$$

ΔM_v : 操作量の変化分, e_n : 偏差 ($SV - PV_n$)

T_I : 積分時間, ΔT : スキャン周期, T_D : 微分レート時間

PV_n : n スキャン目の現在値

EX2000のPID命令では式(2)の様に係数を設定しています。

$$\Delta M = K_p \{ (e - e_{-1}) + \frac{1}{\left| \frac{K_I H}{K_I L} \right|} e + \left| \frac{K_D H}{K_D L} \right| (2P_{-1} - P - P_{-2}) \} \dots\dots\dots (2)$$

ΔM : PID命令の出力データ, e : 偏差リミット, Cap演算を施したもの

K_P : 比例係数, K_I : 積分係数, K_D : 微分係数, P : 入力,

P_{-1} : 前回入力, P_{-2} : 前々回入力

上式から分かる様に, 微分係数, 積分係数はスキャン周期の影響を受けますので, 定周期スキャン実行で使用して下さい。

PIDの応答時間は, スキャン周期により決定されます。定周期スキャンを使用する場合には, 設定したスキャン周期 (10ms ~ 200ms, 10ms 単位) により応答が決まります。

PID命令を含む処理があまり多くない場合は, その処理を定周期割込み処理ページ (7000~7099 ページ) にも書くこともできます。定周期割込みの周期は 5ms ~ 1000 ms までの範囲で 1 ms 単位に設定できます。

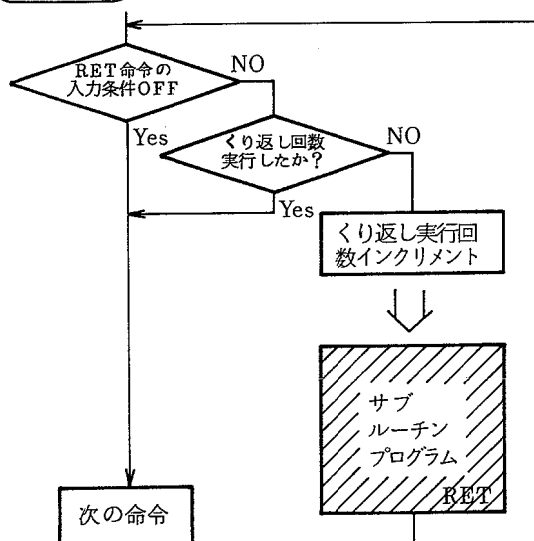
PID命令使用時には, 予め任意の出力カレジスタに初期値 (バイアス値) を格納しておき, それにPID命令の出力を積算したものを操作量として出力して下さい。

CALL		くり返しコールループ														ステップ数	
表 現	条件 入力	CALL N.XX ①														3	
		実行出力															
機 能	①回数, 又はサブルーチンのリターン命令 がOFFになるまでサブルーチンのくり返 し実行を行います。														データ表示		
	条件入力		処 理										出 力				
	OFF		不 実 行										OFF				
ON		実 行										ON					
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値	
	XX	コール先サブルーチンエントリー%														0 ~ 99	
	①	くり返し指定回数					○	○	○	○	○		○			1 ~ 255	

プログラム例



動 作



注 記

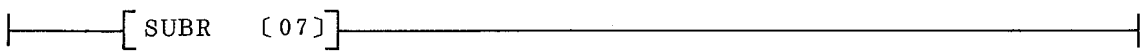
- コールのネスティングは、4段までです。

説 明

- A接点 Z0100 が ON の時、番号 7 のサブルーチンをくり返し実行して、出力を ON にします。リターン命令の入力条件が ON のままだったら、サブルーチンを 10 回くり返しますが、途中で入力条件が OFF になると、10 回に満たなくてもその時点でサブルーチンの実行を止めます。
- 256 以上だった場合は、255 でリミットをかけます。
- コール命令は、メインプログラム、サブプログラム、サブルーチン、割込プログラムにて使用可能ですが、割込プログラムと他のプログラムからの同一サブルーチンコールはできません。
- A接点 Z0100 が OFF の時コールは行わず出力を OFF にします。

SUBR		サブルーチンエントリー																		
表 現																ステップ数				
																2				
機 能	サブルーチンプログラムの入口を示します						条件入力	処 理							出 力					
								続くプログラムの実行を開始します。							ON					
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	XX	サブルーチンエントリー No.																		0 ~ 99

プログラム例

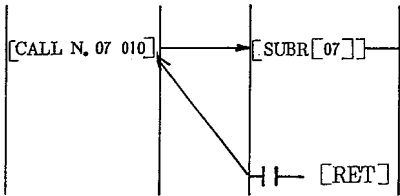


動 作

説 明

メインプログラム サブルーチンプログラム

● サブルーチンプログラムの番号を定義し入口を示します。



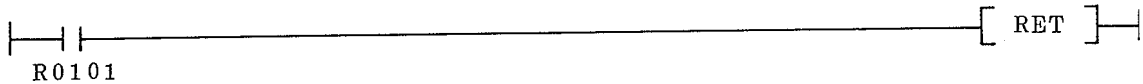
注 記

- この命令は、ページの先頭にしかプログラムできません。
- SFCの実行部、遷移部にはプログラムできません。

RET	サブルーチンリターン
-----	------------

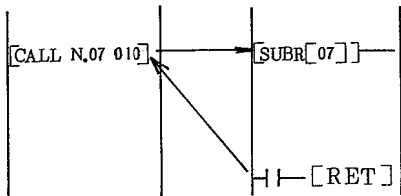
表 現	条件入力 RET														ステップ数
															1
機 能	サブルーチンプログラムの実行を終了し、元のプログラムへ復帰させます。						条件入力	処 理						出 力	
							OFF	くり返しサブルーチン実行を強制完了します。						ON	
							ON	元のコール命令に復帰します。(くり返し有り)						ON	
オ ペ ラ ン ド	記号	名 称	R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T. C.	数 値

プログラム例



動 作

メインプログラム サブルーチンプログラム



説 明

- サブルーチンのプログラムの実行を終了して、元のプログラムのコール命令に戻ります。
A接点 R0101 が OFF の場合は、くり返し回数にまだ達していなくても強制的にサブルーチンを終了します。ON の場合は、コール命令にてくり返し回数をチェックします。
出力を ON にします。

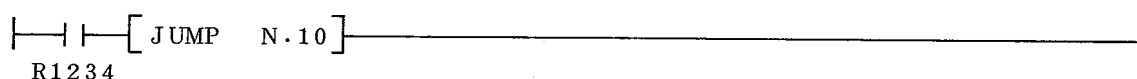
注 記

- SFC の実行部、遷移部にはプログラムできません。

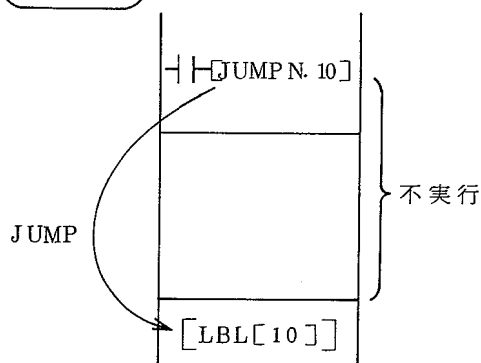
JUMP	直接ジャンプ
------	--------

表 現	条件 入力														ステップ数					
	[JUMP N. XX] 実行出力														2					
機 能	指定された飛び先 No. のラベルへ直接ジャンプします。							条件入力		処 理						出 力				
								OFF		不 実 行						OFF				
								ON		実 行						ON				
オ ペ ラ ン ド	記号	名 称					R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	XX	飛び先ページラベル No.																		0 ~ 99

プログラム例



動 作



説 明

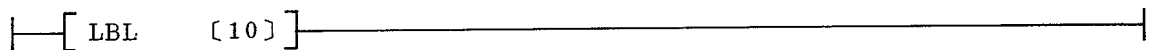
- A 接点 R 1 2 3 4 が ON の時、指定された番号のページへ直接ジャンプをし出力を ON にします。途中の命令は、不実行になります。
- A 接点 R 1 2 3 4 が OFF の時、ジャンプは行わず、出力を OFF にします。

注 記

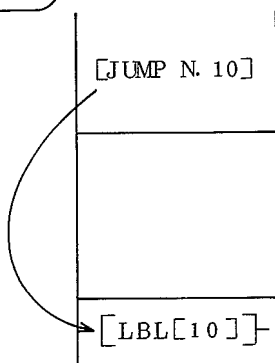
- 逆戻りのジャンプはできません。
- メイン、サブ、サブルーチン、割込みプログラム間でのジャンプはできません。
- SFC の実行部、遷移部にはプログラムできません。

LBL		ジャンプラベル															
表 現																	ステップ数
																	2
機 能	直接ジャンプ命令のジャンプ先を定義します。						条件入力	処 理								出 力	
								続くプログラムの実行を開始								O N	
オ ペ ラ ン ド	記号	名 称		R	X	Y	Z	RW	XW	YW	ZW	D	T	C	T.	C.	数 値
	XX	ジャンプラベル No.															0 ~ 99

プログラム例



動 作



説 明

- 直接ジャンプ命令のジャンプ先を示し、出力を ON にします。

注 記

- この命令はページの先頭にしかプログラムできません。
- SFC の実行部、遷移部にはプログラムできません。

株式会社 **東芝**
