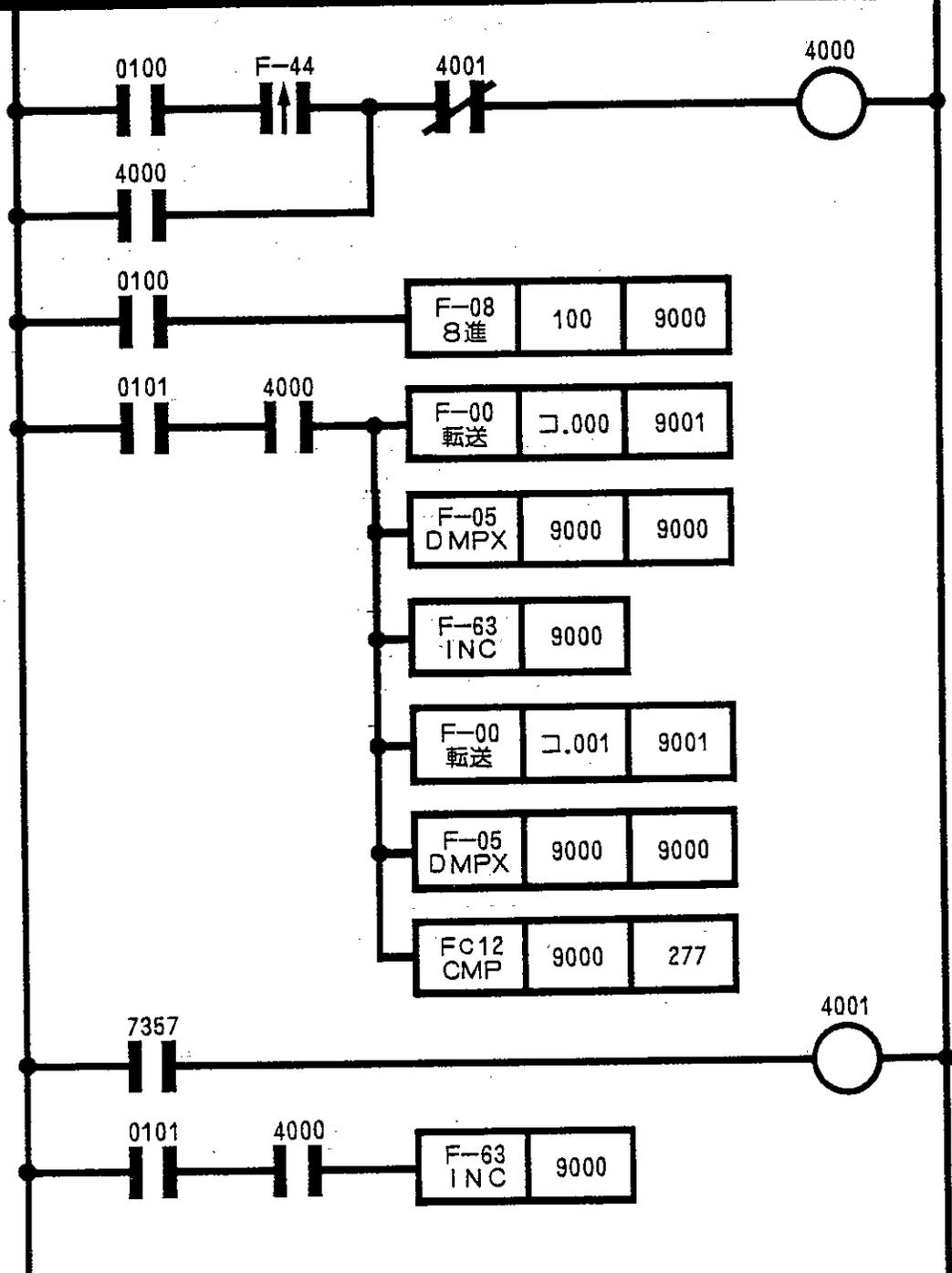


シャーププログラマブルコントローラ

## ニューサテライト-W16/W51

### プログラミングマニュアル



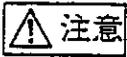
# 安全上のご注意

取付、運転、保守・点検の前に必ずこの取扱説明書とその他の付属書類をすべて熟読し、正しくご使用ください。機器の知識、安全の情報そして注意事項のすべてについて習熟してからご使用ください。この取扱説明書では、安全注意事項のランクを「危険」「注意」として区分してあります。



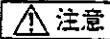
**危険**

：取扱を誤った場合に、危険な状況が起こりえて、死亡または重傷を受ける可能性が想定される場合。



**注意**

：取扱を誤った場合に、危険な状況が起こりえて、中程度の傷害や軽傷を受ける可能性が想定される場合および物的損害だけの発生が想定される場合。

なお、 **注意** に記載した事項でも、状況によっては重大な結果に結びつく可能性があります。いずれも重要な内容を記載していますので必ず守ってください。

禁止、強制の絵表示の説明を次に示します。



：禁止（してはいけないこと）を示します。例えば、分解厳禁の場合は となります。



：強制（必ずしなければならないこと）を示します。例えば、接地の場合は となります。

## ■ 取付について

### 注意

- ・カタログ、取扱説明書に記載の環境で使用してください。  
高温、多湿、じんあい、腐食性ガス、振動、衝撃がある環境で使用すると感電、火災、誤動作の原因となることがあります。
- ・取扱説明書に従って取り付けてください。  
取付に不備があると落下、故障、誤動作の原因となることがあります。
- ・電線くずなどの異物を入れないでください。  
火災、故障、誤動作の原因となることがあります。

## ■ 配線について

### 強制

- ・必ず接地を行ってください。  
接地しない場合、感電、誤動作のおそれがあります。

### 注意

- ・定格にあった電源を接続してください。  
定格と異った電源を接続すると、火災の原因となることがあります。
- ・配線作業は、資格のある専門家が行ってください。  
配線を誤ると火災、故障、感電のおそれがあります。

## ■ 使用について

### 危険

- ・ 通電中は端子に触れないでください。  
感電のおそれがあります。
- ・ 非常停止回路、インターロック回路等はプログラマブルコントローラの外部で構成してください。プログラマブルコントローラの故障により、機械の破損や事故のおそれがあります。

### 注意

- ・ 運転中のプログラム変更、強制出力、RUN、STOP等の操作は十分安全を確認して行ってください。操作ミスにより機械の破損や事故のおそれがあります。
- ・ 電源投入順序に従って投入してください。  
誤動作により機械の破損や事故のおそれがあります。

## ■ 保守について

### 危険

- ・ 電池の $\oplus$  $\ominus$ の逆接続、充電、分解、加熱、火中に投入、ショートはしないでください。  
破裂、発火のおそれがあります。

### 禁止

- ・ 分解、改造はしないでください。  
火災、故障、誤動作の原因となります。

### 注意

- ・ ユニットの着脱は電源をOFFしてから行ってください。  
感電、誤動作、故障の原因となることがあります。
- ・ ヒューズは指定品と交換してください。  
火災、故障の原因となります。

# 目 次

§ 1	はじめに	1
§ 2	システム設計	2
2-1	システム設計手順	2
2-2	システム設計に際しての留意事項	3
2-3	入力ユニット、出力ユニットのリレー番号について	4
(1)	W51の場合	4
(2)	W16の場合	5
2-4	電源の容量について	6
(1)	W51の場合	6
(2)	W16の場合	8
§ 3	コントロールユニットの構成と動作	10
3-1	コントロールユニットの構成	10
3-2	データメモリ	11
(1)	データメモリの種類	11
(2)	データメモリの機能	11
(3)	キーリレーの特殊領域	13
(4)	TMR、CNT、MDのデータ格納領域	14
(5)	リレー領域のバイトアドレスと絶対アドレス	14
(6)	データメモリのアドレスマップ	15
3-3	システムメモリ	21
(1)	コントロールユニットの各種機能を設定する領域	21
(2)	PCの異常コードを格納する領域	23
(3)	コンピュータリンク、データリンク、リモートI/Oの各オプションが使用する領域	23
3-4	プログラムメモリとファイルレジスタ	24
(1)	プログラムメモリ	24
(2)	ファイルレジスタ	24
(1)	ファイルレジスタ容量の設定	24
(2)	ファイルレジスタを使用できる命令	25
(3)	プログラマからの書込	25
(3)	ROM運転	25
3-5	システムメモリの設定とメモリクリア	26
3-6	運転サイクル	27
(1)	動作フローチャート	27
(2)	パワーON処理	28
(3)	スキャンサイクル	28
(1)	ハードウェアチェック	28
(2)	フラグのクリア	29
(3)	ゼロクロス同期	29
(4)	入出力処理	30
(5)	ウォッチドグタイマ	30
(6)	プログラマ、オプションからのリクエストに対する処理	30
(7)	0.1秒 1秒クロックのセット/リセット	30
(8)	ユーザプログラム処理	30
(9)	スキャンタイム	32

3-7	自己診断	34
(1)	自己診断内容	34
(2)	停止出力	35
(3)	特殊リレー	35
(4)	異常コード	35
(5)	異常時の出力ユニットのON/OFF状態	35
§ 4	命令語の説明	36
4-1	命令語一覧表	36
4-2	ビット処理部の動作	39
4-3	基本命令の説明	40
(1)	STR OUT	40
(2)	STR NOT	40
(3)	AND	41
(4)	AND NOT	41
(5)	OR	41
(6)	OR NOT	42
(7)	AND STR	42
(8)	OR STR	42
(9)	TMR	43
(10)	CNT	45
(11)	MD	46
4-4	ラダー設計に関する留意事項	49
(1)	リレー盤用ラダー図から書換えを必要とする回路	49
(2)	入出力一括処理方式	50
(3)	プログラム順序による影響	51
(4)	プログラムの簡略化	52
(5)	直並列回路のプログラム	52
4-5	応用命令に関する留意事項	54
(1)	数値の表現方法	54
(2)	ソースとテストネーション	56
(3)	応用命令とスタックレジスタ	56
(4)	演算実行条件	59
(5)	データ処理命令とフラグ	59
(6)	倍長演算	63
(7)	リレー領域を使用するとき	65
(8)	データメモリのブロックと基準アドレス	66
(9)	数値信号の入出力方法	69
4-6	応用命令の説明	73
F-00	1バイトデータの転送	73
F-01	BCD定数の転送	74
F-02	レジスタ間のデータ交換	75
F-03	BCD→BINARY変換	76
F-04	BINARY→BCD変換	77
F-05	1バイトデータの分配	78
F-06	1バイトデータの抽出	80
F-07	10進定数の転送	81

F-08	8進定数の転送	82
F-09	8ビットデータの反転	83
F-10	レジスタ間(BCD2桁)の加算	84
F c10	レジスタとBCD定数(2桁)の加算	86
F-11	レジスタ間(BCD2桁)の減算	87
F c11	レジスタとBCD定数(2桁)の減算	89
F-12	レジスタ間の比較	90
F c12	レジスタと定数の比較	91
F-13	レジスタ間の論理積	92
F c13	レジスタと定数の論理積	93
F-14	レジスタ間の論理和	94
F c14	レジスタと定数の論理和	95
F-15	レジスタ(4桁)間の乗算	96
F-15	レジスタ(8桁)間の乗算(W51のみ)	97
F c15	レジスタ(4桁)とBCD定数(3桁)の乗算	98
F c15	レジスタ(8桁)とBCD定数(3桁)の乗算(W51のみ)	99
F-16	レジスタ(4桁)とレジスタ(2桁)の除算	100
F-16	レジスタ(8桁)とレジスタ(8桁)の除算(W51のみ)	102
F c16	レジスタ(4桁)とBCD定数(2桁)の除算	103
F-17	レジスタ間的一致	104
F c17	レジスタと定数的一致	105
F-18	レジスタ間の排他的論理和	106
F c18	レジスタと定数の排他的論理和	107
F-30	マスタコントロールセット	108
F-31	マスタコントロールリセット	108
F-40	エンド命令	111
F-41	ジャンプコントロールセット	112
F-42	ジャンプコントロールリセット	112
F-43	ビット反転	114
F-44	ON時微分	115
F-45	OFF時微分	116
F-50	4→16デコーダ	117
F-51	16→4エンコーダ	118
F-52	7SEGデコーダ	119
F-53	BCD(4桁)→BIN(16ビット)変換	120
F-54	BIN(16ビット)→BCD(6桁)変換	121
F-55	上位4ビットと下位4ビットの交換	122
F-60	両方向シフトレジスタ	123
F-61	非同期両方向シフトレジスタ	125
F-62	BCD2桁のアップダウンカウンタ	126
F-63	加算カウンタ	127
F-64	減算カウンタ	128
F-70	nバイト一括転送	129
F-71	8進定数一括転送	131
F-72	ファイルレジスタへの分配	132
F-73	ファイルレジスタからの抽出	134
§ 5	プログラム例	136

## § 1 はじめに

ニューサテライトW16/W51は、AND、ORのようなシーケンス制御命令ばかりではなく、四則演算、データ変換、転送、ファイル命令などの高度な演算機能をもった強力なプログラマブルコントローラです。本書によりプログラミングの方法をマスタしていただき、機能を十二分に引出していただきますようお願いいたします。

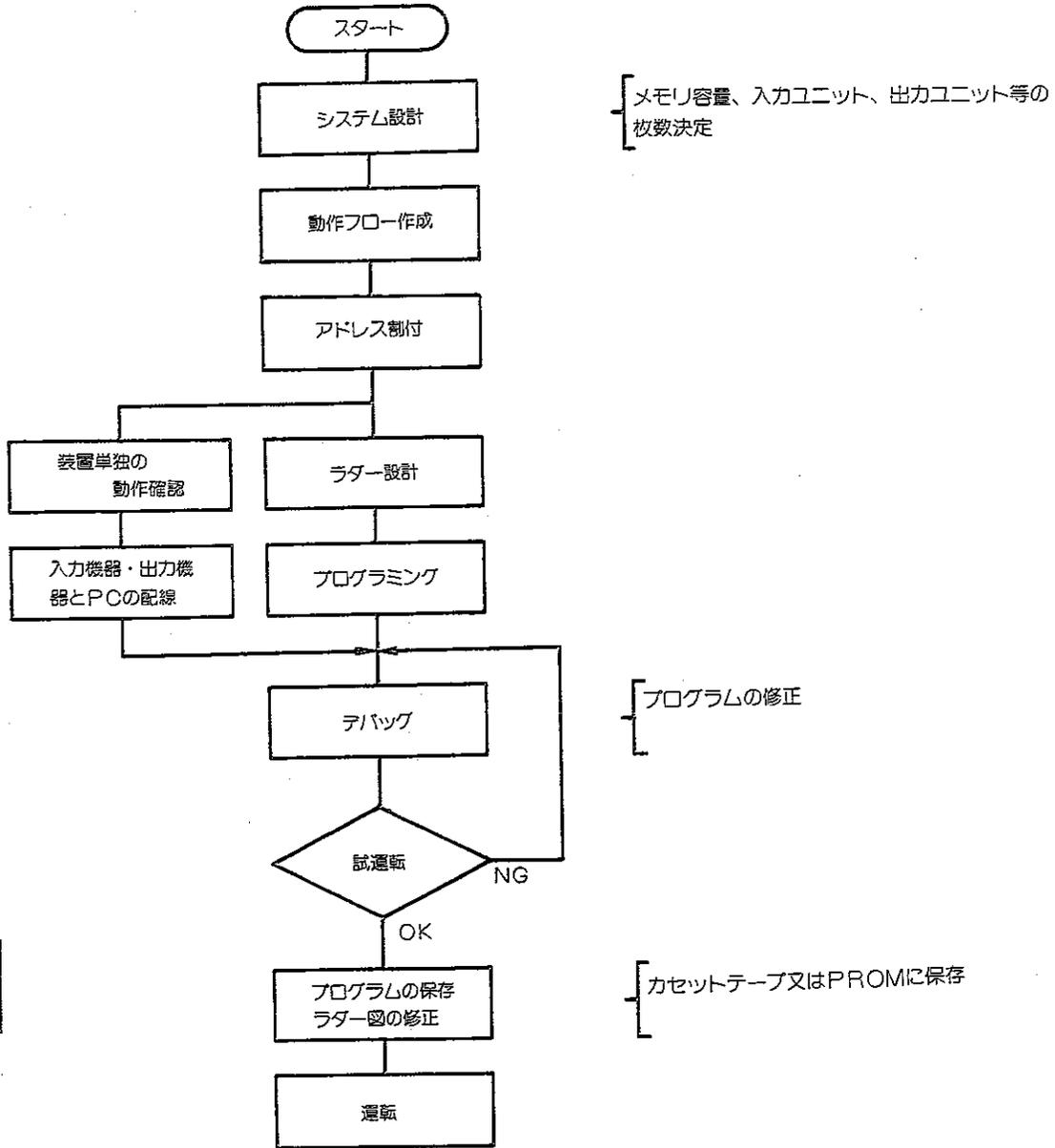
なお、取付け、配線方法などは「W16取扱説明書」「W51取扱説明書」を、プログラマ等の周辺装置の操作方法は各周辺装置の「取扱説明書」を、データリンク等のオプションに関しては各オプションの「取扱説明書」をご参照ください。

## § 2 システム設計

### 2-1 システム設計手順

PCを用いた制御装置の設計手順は、一般のリレーシーケンス制御装置の設計とほぼ同じです。下図に、PCを用いた装置の設計手順の例を示します。

W16とW51のソフトウェアは完全コンパチブルです。ただし、メモリ容量、制御入出力点数の違いにはご注意ください。



2

## 2-2 システム設計に際しての留意事項

PCと従来のリレー回路との本質的な相違点は、PCが制御内容のプログラムをサイクリック（直列）に処理しているのに対して、リレー回路は並列処理をしているといえます。

したがってリレー回路の場合は、故障がおこってもその異常動作は限定されますが、PCの場合は、システム全体の異常動作につながります。

フェイルセーフの観点から、すべての制御をPCに任せるのは良策ではなく、機械の破壊や人身事故につながる部分、たとえば

非常停止回路

保護回路

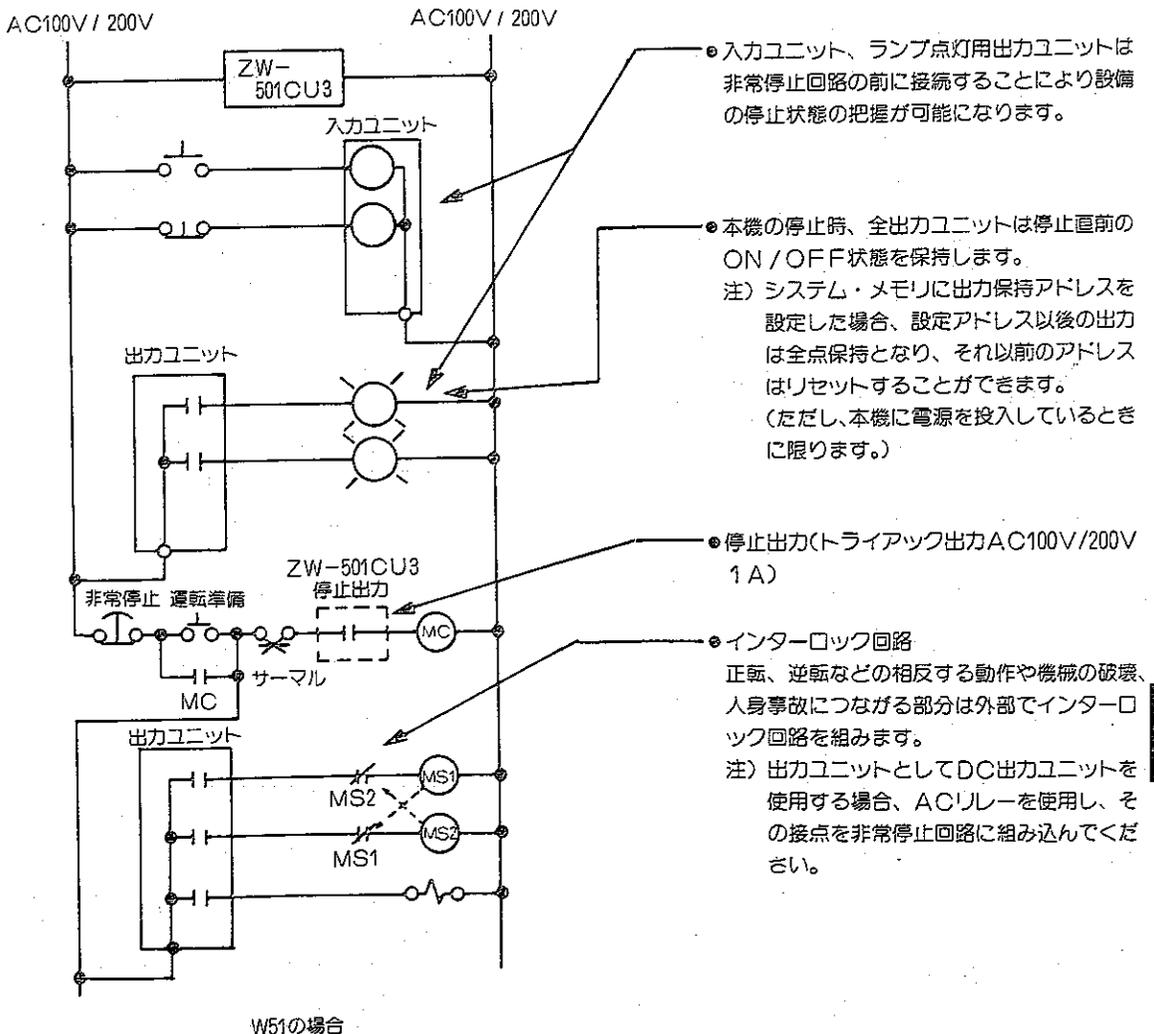
高電圧機器の操作回路

などは、PCの外部で構成してください。

また、サイクリック処理のため、応答時間にも注意する必要があります。

さらに、PCに電源を投入した瞬間に出力ユニットの出力が瞬時ONすることがありますので、これより外部出力機器が動作することを防止するため、次図のように運転準備回路にPCの停止出力を直列に接続してください。

（PCに電源を投入して約1秒後に停止出力がONの状態になります。）

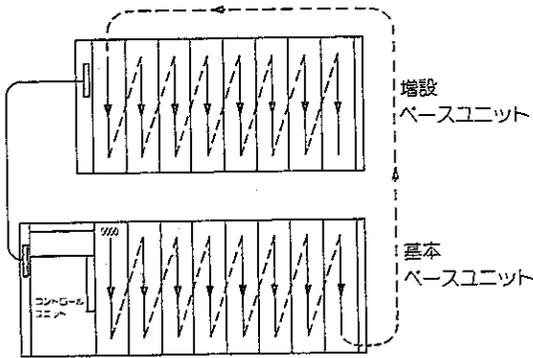


2

## 2-3 入力ユニット、出力ユニットのリレー番号について

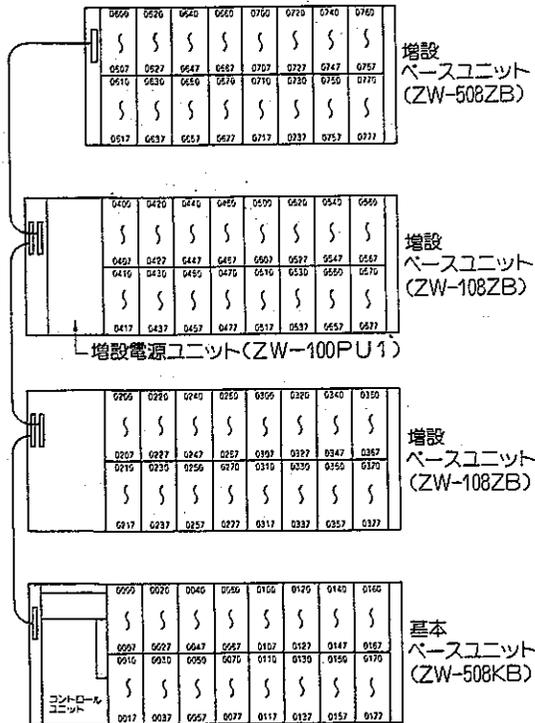
### (1) W51の場合

入力ユニット、出力ユニットのリレー番号はベースユニットへの装着順に追番方式で決まります。

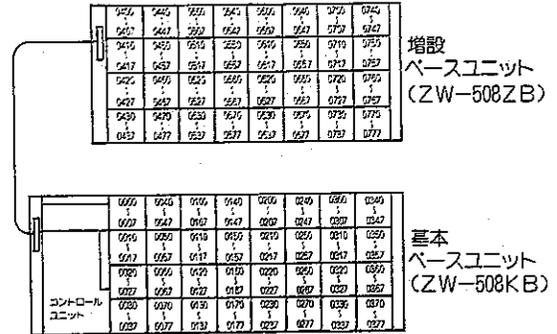


コントロールユニットの右隣に装着した入出力ユニットの最上段を基点(0000)として上から下へ、左から右へ順次8進数でアドレスが決定されます。

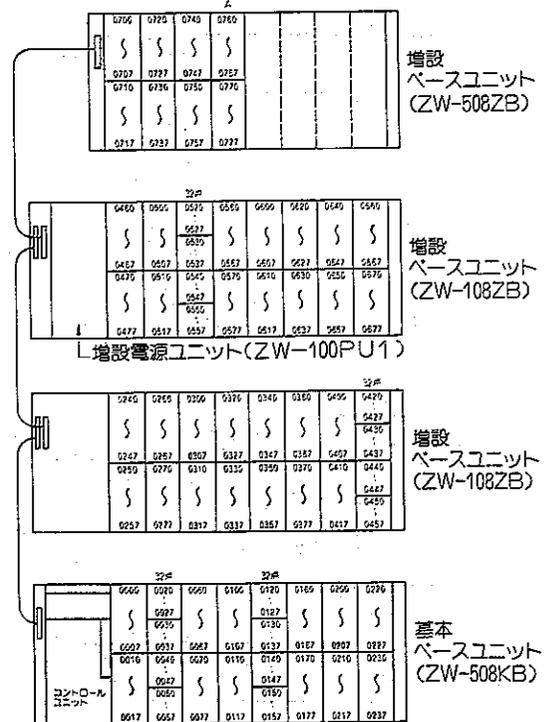
(1) 入力ユニット、出力ユニットとして16点ユニットだけを使用する場合、装着位置とリレー番号の関係はつぎようになります。



(2) 入力ユニット、出力ユニットとして32点ユニットだけを使用する場合、装着位置とリレー番号の関係はつぎようになります。



(3) 入力ユニット、出力ユニットとして16点ユニット、32点ユニットを混合して使用する場合、装着位置とリレー番号の関係はつぎようになります。

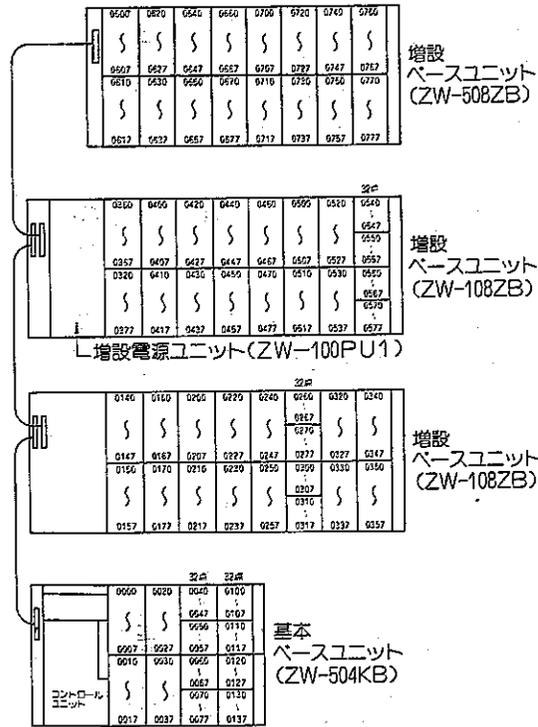


注1 W51の最大入出力点数は512点(0000~0777)です。512点を越えるように入出力ユニットを装着しても(0000~0777)の512点のみ有効です。例えば、Aの位置に32点ユニットを装着してもAのユニットは(0760~0777)の16点のみ有効です。

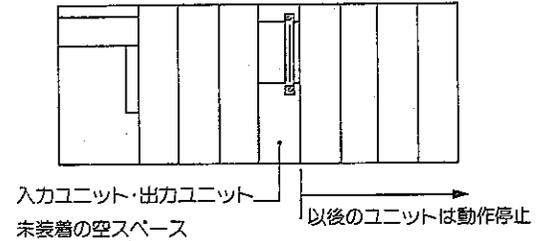
コントロールユニットの右隣の入出力ユニットの最上段を基点として、あくまで上から下へ、左から右へという追番方式の原則に従ってリレー番号が決定されます。コントロールユニットに付属のアドレス表示ラベルをご使用いただくと動作チェックの際に便利です。実装位置に合わせて番号を選び、入出力ユニットカバーに貼り付けてください。

アドレス表示ラベルはリレー番号の2桁目、3桁目を示します。(0230~0237の場合、**23**を使用)

- (4) 基本ベースユニットとしてZW-504KBをご使用の場合、入力ユニット、出力ユニットの装着位置とリレー番号の関係はつぎのようになります。

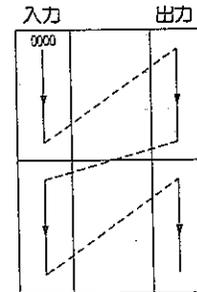


- (6) 基本ベースユニット、増設ベースユニットに入力ユニット、出力ユニットを装着する場合、空スペースを設けず、左からつめて装着してください。空スペースより右に装着された入力ユニット、出力ユニットは動作しません。基本ベースユニットに空スペースがあると、増設ベースユニットに装着された入力ユニットも動作しません。



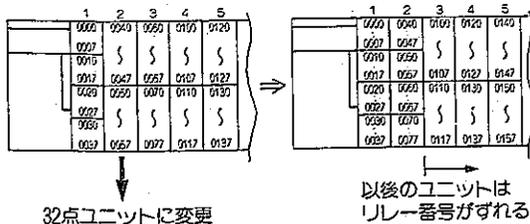
## (2) W16の場合

入出力ユニットのリレー番号は入力側から出力側に追番方式で決まります。

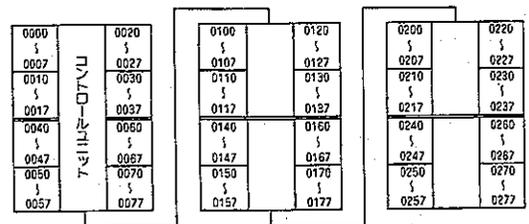


- (5) 点数の異なる入力ユニット、出力ユニットに変更すると、以後の入力ユニット、出力ユニットのリレー番号がずれますのでご注意ください。

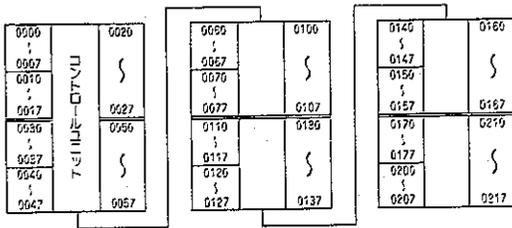
(例)



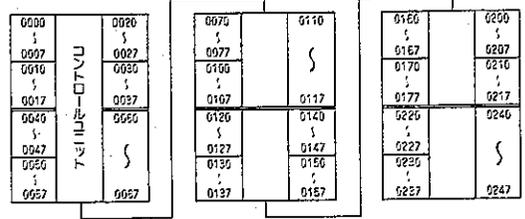
- (1) 入出力ユニットとして32点ユニットだけを使用する場合、リレー番号はつぎのようになります。



(2) 入力ユニットとして24点ユニットだけを使用する場合、リレー番号はつぎようになります。



(3) 入力ユニットとして32点ユニット、24点ユニットを混合して使用する場合、リレー番号はつぎようになります。



## 2-4 電源の容量について

### (1) W 51の場合

コントロールユニット (ZW-501CU3) に内蔵されたDC 5V電源の定格電流は7Aです。また、増設電源ユニット (ZW-100PU1/ZW-100PU2) のDC 5V電源の定格電流は7A/PU1、12A/PU2です。この範囲を越えて入力ユニット、出力ユニット等を使用

しますと、DC5V電源の電流制限機能が働き、PCは運転を停止します。システム設計をする場合、入力ユニット、出力ユニット等の消費電流の合計が7A又は12A (ZW-100PU2使用時)以下であることを確認してください。

機種名	全点OFF時の消費電流 I(OFF) (mA)	全点ON時の消費電流 I(ON) (mA)	n点ON時の消費電流 I(n) (mA)
ZW-501CU3(コントロールユニット)		700	
ZW-101PG1(プログラマ)		700	
JW-10PG(プログラマ)		200	
ZW-100PG2(図示プログラマ)		900	
ZW-16N1(AC100V入力)	50	120	50+ 4.4n
ZW-16N2(DC12/24V入力)	50	120	50+ 4.4n
ZW-16N3(AC200V入力)	50	120	50+ 4.4n
ZW-32N1T(AC100V入力)	75	200	75+ 3.8n
ZW-32N2/N2T(データ入力)	85	85	85
ZW-64N2(データ入力)	40	170	40+4.0n1+0.03n2
ZW-8S1(AC100V出力)	80	240	80+20n
ZW-8S2(DC12/24V出力)	80	160	80+10n
ZW-16S1(AC100V出力)	80	400	80+20n
ZW-16S2(DC12/24V出力)	80	240	80+10n
ZW-16S3(AC200V出力)	80	400	80+20n
ZW-16S4(リレー出力)	85	180	85+ 6n
ZW-16S4D(リレー出力)	20	80	20+ 3.8n
ZW-32S1T(AC100V出力)	85	600	85+16.1n
ZW-32S2/S2T/S2TD(データ出力)	100	320	100+ 6.9n
ZW-32S4T(リレー出力)	100	220	100+ 3.7n
ZW-32S5(ソースタイプデータ出力)	85	185	85+ 3.1n
ZW-64S2(データ出力)	80	420	80+7.0n1+3.0n2
ZW-32IO2(DC5/12/24V入出力)	180	320	180+5nIN+3.5nOUT
ZW-1HC5(高速カウンタ)		600	
ZW-1HC6(高速カウンタII)		740	
ZW-4AD2(アナログ入力)		400	
ZW-2DA2(アナログ出力)		300	
ZW-14PC2(リレースキャッチ)		170	

n1 : ランプ点灯ON点数  
n2 : ランプ消灯ON点数

機種名	全点OFF時の消費電流 (OFF) (mA)	全点ON時の消費電流 (ON) (mA)	n点ON時の消費電流 I(n) (mA)
ZW-1PO2(パルス出力)		600	
ZW-232SU(シリアルI/O)		900	
ZW-112PM(位置決め基本)		600	
ZW-202PM(位置決め増設)		280	
ZW-100DM(タミー)		60	
ZW-501CL2(コンピュータリンク)		600	
ZW-501DL1(テータリンク)		600	
ZW-501DL0(テータリンク)		600	
ZW-501RM1(リモートI/O親局)		600	
ZW-501RS1(リモートI/O子局)		600	
ZW-501BL(BRAINリンク)		600	

(例1)

ZW-501CU3(コントロールユニット)	
ZW-101PG1(プログラマ)	
ZW-16N1(AC100V入力)——10ユニット	
ZW-16S1(AC100V出力)——6ユニット	
ZW-501CL2(コンピュータリンク)	
ZW-501DL1(テータリンク)	

全点同時ONとして計算すると

ZW-501CU3	0.7(A)
ZW-101PG1	0.7(A)
ZW-16N1	$0.12 \times 10 = 1.2(A)$
ZW-16S1	$0.40 \times 6 = 2.4(A)$
ZW-501CL2	0.6(A)
ZW-501DL1	0.6(A)
合計	6.2(A)

7A以下であり問題ありません。

(例2)

ZW-501CU3(コントロールユニット)	
ZW-100PG2(図示プログラマ)	
ZW-16N1(AC100V入力)——6ユニット	
ZW-16S1(AC100V出力)——10ユニット	
ZW-501CL2(コンピュータリンク)	
ZW-501RM1(リモートI/O親局)	

全点同時ONとして計算すると

ZW-501CU3	0.7(A)
ZW-100PG2	0.9(A)
ZW-16N1	$0.12 \times 6 = 0.72(A)$
ZW-16S1	$0.40 \times 10 = 4.0(A)$
ZW-501CL2	0.6(A)
ZW-501RM1	0.6(A)
合計	7.52(A)

全点同時ONの条件では7A以上となります。このような場合、実使用における同時ONとなる最大の入出力点数を調べます。

入力96点中65点、出力160点中100点が最大の同時ONの場合を計算すると

ZW-501CU3	0.7(A)
ZW-100PG2	0.9(A)
ZW-16N1	$0.05 \times 6 + 0.0044 \times 65 = 0.59(A)$
ZW-16S1	$0.08 \times 10 + 0.02 \times 100 = 2.8(A)$
ZW-501CL2	0.6(A)
ZW-501RM1	0.6(A)
合計	6.19(A)

7A以下となり問題ありません。

(例3)

ZW-501CU3(コントロールユニット)	} 内蔵電源
ZW-101PG1(プログラマ)	
ZW-16N1(AC100V入力)——8ユニット	
ZW-16S1(AC100V出力)——8ユニット	} 増設電源
ZW-16N1(AC100V入力)——8ユニット	
ZW-16S1(AC100V出力)——8ユニット	

全点同時ONとして計算すると

(1)内蔵電源	
ZW-501CU3	0.7(A)
ZW-101PG1	0.7(A)
ZW-16N1	$0.12 \times 8 = 0.96(A)$
ZW-16S1	$0.40 \times 8 = 3.2(A)$
合計	5.56(A)・・・①

(2)増設電源

ZW-16N1	$0.12 \times 8 = 0.96(A)$
ZW-16S1	$0.40 \times 8 = 3.2(A)$
合計	4.16(A)・・・②

①、②ともに7A以下となり問題ありません。

2

## (2) W16の場合

コントロールユニット(ZW-160CU)に内蔵の電源の定格出力は、DC5V:4.3A, DC24V:0.8Aです。また増設電源ユニット(ZW-160PU1)の定格出力はDC24V:0.8Aです。定格出力電流値を越えて入出力ユニット等を使用しますと電源ユニットの異常発熱(ヒューズ

が作動します)・電流制限機能による出力電圧の低下など誤動作の原因となります。システム設計をする場合、入出力ユニット等の消費電流の合計が使用各電源の定格以下である事を確認してください。

機種名	DC 5 V 電源		DC 24 V 電源	
	全点ON時の消費電流 I(ON) (mA)	n点ON時の消費電流 I(n) (mA)	全点ON時の消費電流 I(ON) (mA)	n点ON時の消費電流 I(n) (mA)
ZW-160CU(コントロールユニット)	700		—	
ZW-101PG1(プログラマ)	700		—	
JW-10PG(プログラマ)	200		—	
ZW-100PG2(図示プログラマ)	900		—	
ZW-32NS1 (16AC 16トリアック)	入力	155	155	—
	出力	272	17n	—
ZW-32NS2 (16DC 16トランジスタ)	入力	130	130	160
	出力	80	5n	10n
ZW-32NS3 (16DC 16トリアック)	入力	130	130	160
	出力	272	17n	—
ZW-32NS4 (16DC 16リレー)	入力	130	130	160
	出力	30	30	240
ZW-32NS5 (16AC 16リレー)	入力	155	155	—
	出力	30	30	240
ZW-24NS1 (16AC 8トリアック)	入力	155	155	—
	出力	152	19n	—
ZW-24NS2 (16DC 8トランジスタ)	入力	130	130	160
	出力	48	6n	—
ZW-24NS3 (16DC 8トリアック)	入力	130	130	160
	出力	152	19n	—
ZW-24NS4 (16DC 8リレー)	入力	130	130	160
	出力	20	20	120
ZW-24NS5 (16AC 8リレー)	入力	155	155	—
	出力	20	20	120
ZW-160CL2(コンピュータリンク)	550		—	
ZW-160DL1(データリンク)	550		—	
ZW-160RM1(リモートI/O親局)	550		—	
ZW-160RS1(リモートI/O子局)	550		—	

**注1** コントロールユニット1台に対し、入出力ユニットは6台まで接続可能ですが、上表より入出力の同時ON点数に制限がある場合があります。

**注2** ZW-32NS4、ZW-32NS5、ZW-24NS4、ZW-24NS5をご使用の場合は、2ユニットを越える毎に増設電源ユニットが1台必要です。

(例1)

ZW-160CU(コントロールユニット)	
ZW-101PG1(プログラマ)	
ZW-32NS1(16AC16トリアック)	— 4ユニット
ZW-24NS4(16DC8リレー)	— 2ユニット

(1)DC5V電源

ZW-160CU	0.7 (A)
ZW-101PG1	0.7 (A)
ZW-32NS1	$(0.155+0.272) \times 4 = 1.71(A)$
ZW-24NS4	$(0.13+0.02) \times 2 = 0.3 (A)$

合計 3.41(A)…①

(2)DC24V電源

ZW-24NS4	$(0.16+0.12) \times 2 = 0.56(A)$ …②
----------	-------------------------------------

①, ②ともに定格電流以下であり問題ありません。

(例2)

ZW-160CU(コントロールユニット)	
ZW-100PG2(図示プログラマ)	
ZW-32NS3(16DC16トリアック)	— 3ユニット
ZW-32NS4(16DC16リレー)	— 3ユニット

(1)DC5V電源

ZW-160CU	0.7 (A)
ZW-100PG2	0.9 (A)
ZW-32NS3	$(0.13+0.272) \times 3 = 1.21(A)$
ZW-32NS4	$(0.13+0.03) \times 3 = 0.48(A)$

合計 3.29(A)…①

(2)DC24V電源

ZW-32NS3	$0.16 \times 3 = 0.48(A)$
ZW-32NS4	$(0.16+0.24) \times 3 = 1.2 (A)$

合計 1.68(A)…②

注2 に記載のようにZW-32NS4 を3ユニット使用する場合、増設電源ユニットが1台必要であり増設電源ユニットを1台使用時のDC24V定格電流値は(0.8+0.8)Aの計1.6Aとなります。

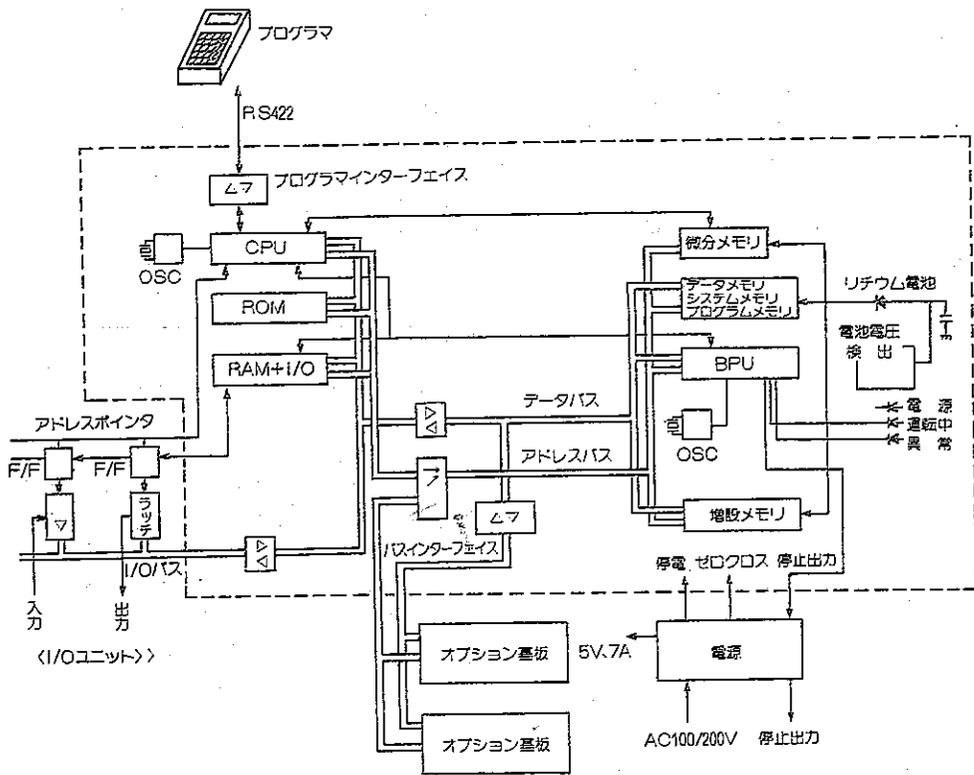
②はこの定格電流値1.6Aを越えています。このような場合、増設電源ユニットをもう1台使用するか実使用における同時ONとなる最大の入出力点数を調べます。

全DC入力96点中80点が最大の同時ONの場合を計算すると  $1.68 - \{(96-80) \times 0.01\} = 1.52(A)$  1.6A以下となり問題ありません。

但し、電源個々について定格電流値0.8Aを越えないように注意する必要があります。

# § 3. コントロールユニットの構成と動作

## 3-1 コントロールユニットの構成



ブロック図の説明

名称	内容
CPU	W16/W51 全体をコントロールしているCPUで、主に下記内容を行なっています。 ① I/O ユニットとのデータ転送 ② プログラマとのデータ通信 ③ ユーザメモリへのプログラム転送 ④ BPUのコントロール ⑤ シーケンス演算 (基本命令以外)
BPU	カスタムLSIを使用し、シーケンスの基本命令を実行させています。
I/O	I/O ユニットとCPUのデータ交換をおこなうポートです。
プログラムメモリ	シーケンスプログラム・ファイルレジスタ用メモリでリチウム電池で停電保護しています。
データメモリ	I/O、タイマ、カウンタ、レジスタ等のメモリでリチウム電池で停電保護しています。
微分メモリ	プログラム上、リレーの立上りを検出するためのメモリです。
システムメモリ	CPUの機能指定、異常コードの格納、オプションモジュールの機能指定用のメモリです。

## 3-2 データメモリ

### (1) データメモリの種類

種類	容量	リレー番号(ビットアドレス)	バイトアドレス	絶対アドレス	停電後の状態
入出力リレー	512点 (64/バイト)	0000~0777	コ.000~コ.077	000~077	クリア [注1]
リンクリレー	512点 (64/バイト)	2000~2777	コ.200~コ.277	100~177	クリア [注2]
補助リレー	512点 (64/バイト)	4000~4777	コ.400~コ.477	200~277	クリア [注2]
キーブリレー	256点 (32/バイト)	7000~7377	コ.700~コ.737	300~337	保持 [注2]
TMR、CNT、MD	128点(現在値256/バイト)		b.000~b.377		TMRクリア CNT・MD 保持 [注3]
レジスタ	256/バイト		9000~9377		保持
ファイルレジスタ	最大 4096/バイト		30000~37777 [注4]		保持

**注1** データメモリのアドレスはビットアドレス、バイトアドレス、絶対アドレスとも、8進数で扱います。(ただしレジスタ領域の4桁目の9は例外)したがって0007の次は0008ではなく、0010となります。(詳細は(6)「データメモリのアドレスマップ」をご参照ください。8進数に関しては、4-5(1)「数値の表現方法」をご参照ください。

**注2** キーブリレーとは停電後の電源投入時、停電直前の状態を保持するデータメモリのリレー領域です。システムメモリの#200にキーブリレー領域を指定することにより、キーブリレー領域の拡大、

縮小が可能です。

出荷時、キーブ機能をもつ領域を(7000~7377)に設定しているため、この領域をキーブリレーと呼んでいます。詳細は3-3「システムメモリ」の項をご参照ください。キーブ指定していない領域は電源投入時クリアされます。

**注3** TMRはシステムメモリ#201の設定により、停電時保持とすることができます。

**注4** ファイルレジスタの容量はシステムメモリ#205の設定により決まります。詳細は3-3「システムメモリ」の項をご参照ください。

### (2) データメモリの機能

入出力リレー	入力ユニットを装着した領域	<ul style="list-style-type: none"> <li>● 毎スキャンサイクルの入出力処理で入力ユニットのON/OFF状態を読み込み、1スキャンサイクル中保持します。[注1]</li> <li>● プログラムで入力情報(接点、データ)として使用します。</li> </ul>
	出力ユニットを装着した領域	<ul style="list-style-type: none"> <li>● ユーザプログラムで、コイル、テストティネーションとして演算結果を書込みます。</li> <li>● 入出力処理で出力ユニットにON/OFF状態が転送されます。</li> <li>● 演算結果はプログラム中で接点、ソースとして使用できます。</li> </ul>
	ユニット未装着領域	<ul style="list-style-type: none"> <li>● ユーザプログラムで、演算結果を書込みます。</li> <li>● 補助リレー、リモートI/O用リレーとして使用できます。</li> </ul>
リンクリレー	データリンク受信用に使用している領域	<ul style="list-style-type: none"> <li>● データリンクモジュールからのリクエストに対する処理時、他のPCからの受信データを格納します。</li> <li>● プログラムで入力情報(接点、データ)として使用します。</li> </ul>
	データリンク送信用に使用している領域	<ul style="list-style-type: none"> <li>● ユーザプログラムで、コイル、テストティネーションとして演算結果を書込みます。</li> <li>● データリンクモジュールからのリクエストに対する処理時、データリンクモジュールのバッファメモリに転送され、他のPCに送信されます。</li> <li>● 演算結果は、プログラム中、接点、ソースとして使用できます。</li> </ul>
	リモートI/O入力用に使用している領域	<ul style="list-style-type: none"> <li>● リモートI/Oモジュールからのリクエストに対する処理時、リモートI/Oモジュールから子局の入力ユニットのON/OFF状態を読み込みます。</li> <li>● プログラムで入力情報(接点、データ)として使用します。</li> </ul>
	リモートI/O出力用に使用している領域	<ul style="list-style-type: none"> <li>● ユーザプログラムで演算結果を書込みます。</li> <li>● リモートI/Oモジュールからのリクエストに対する処理時、リモートI/Oモジュールのバッファメモリに転送され、リモートI/O子局の出力ユニットに送信されます。</li> <li>● 演算結果はプログラム中、接点、ソースとして使用できます。</li> </ul>

リンク	データリンク、リモートI/Oに使用していない領域	●補助リレーとして使用できます。
補助リレー	リモートI/O入力用に使用している領域	リンクリレーをリモートI/O入力用に使用したときと同様です。
	リモートI/O出力用に使用している領域	リンクリレーをリモートI/O出力用に使用したときと同様です。
	リモートI/Oに使用していない領域	●ユーザプログラムでコイル、テストステーションとして演算結果を書込みます。 ●外部に出力する必要のない演算結果の一時記憶に使用します。 ●演算結果はプログラム中、接点、ソースとして使用できます。
キーブリレー	リモートI/O入力用に使用している領域	リンクリレーをリモートI/O入力用に使用したときと同様です。
	リモートI/O出力用に使用している領域	リンクリレーをリモートI/O出力用に使用したときと同様です。
	リモートI/Oに使用していない領域	●ユーザプログラム処理でコイル、テストステーションとして演算結果を書込みます。 ●外部に出力する必要のない演算結果の一時記憶に使用します。 ●演算結果はプログラム中、接点、ソースとして使用できます。 ●停電時保持する必要のある接点、データの記憶用に使用します。
	特殊領域 (7340~7377)	●異常コードの格納、各種フラグの領域でプログラム中、コイル、テストステーションとしては使用できません。接点、ソースとして使用できます。
TMR・CNT・MD領域	TMRとして使用している場合	●現在値が0になるとTMR接点がONします。 ●TMR接点はプログラム中何度でも使用できます。 ●現在値はプログラム中、ソース、(特殊用途としてテストステーション)として使用できます。
	CNTとして使用している場合	●現在値が0になるとCNT接点がONします。 ●CNT接点はプログラム中何度でも使用できます。 ●現在値はプログラム中、ソース(特殊用途としてテストステーション)として使用できます。
	MDとして使用している場合	●出力指示条件がONのとき、現在値領域にMD情報が書き込まれます。 ●現在値領域のMD情報はプログラム中ソースとして使用できます。
	TMR、CNT、MDとして使用していない領域	●現在値領域 (b. x x x) をレジスタとして使用できます。
レジスタ		●ユーザプログラムでテストステーションとして演算結果を書込みます。 ●演算結果はプログラム中、ソースとして使用します。 ●設定値変更モードで、プログラマ等からデータを書込むこともできます。
ファイルレジスタ		●レジスタ領域 (9000~9377) の256バイト以上にレジスタ領域が必要な場合に使用します。 ●ユーザプログラム処理でF-05、06、70、71、72、73の命令により読み出し、書き込みが可能です。 ●設定値変更モードでプログラマ等からデータを書込むこともできます。

**注1** 入力ユニットを装着している領域は、入出力処理で読み込んだON/OFF状態を次のサイクルの入出力処理まで保持しますが、プログラム中でこれをコイル、テストステーションとして使用すると、そのスキャンサイクル中は演算結果によりデータメモリが書換えられます。

**注2** ソース、テストステーションとは応用命令で、演算結果を入れるレジスタをテストステーション、演算前のデータを入れるレジスタをソースと呼び

ます。  
詳細は、4-5「応用命令に関する留意事項」の項をご参照ください。

**注3** データリンク、リモートI/Oに関する詳細は各オプションの取扱説明書をご参照ください。

**注4** キーブリレーの特殊領域は、3-2(3)「キーブリレーの特殊領域」の項をご参照ください。

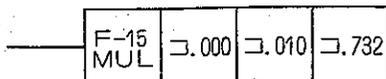
**注5** ファイルレジスタの詳細は、3-4「プログラムメモリとファイルレジスタ」の項をご参照ください。

### (3) キープリレーの特殊領域

キープリレーの7340~7377の32点は、以下のような特殊な領域になっています。

7340	自己診断結果の異常コードを収納する特殊レジスタでバイトアドレスコ.734として扱います。
7341	
7342	
7343	
7344	
7345	
7346	
7347	
7350	乗除算命令桁数選択(W51のみ)
7351	
7352	
7353	
7354	ノンキャリーフラグ
7355	
7356	
7357	
7357	
7360	0.1秒クロック
7361	
7362	
7363	
7364	
7365	
7366	
7367	
7370	メモリ異常
7371	
7372	
7373	
7374	
7375	
7376	
7377	
7377	
7377	

これらのキープリレーは、7365、7367の2点を除き、CPUから書込まれる領域で、ユーザプログラムでは接点、ソースとして使用します。ユーザプログラムでコイル、テストネーションとして使用することの無いようご注意ください。2バイト以上のデータメモリを扱う命令や、分配・抽出命令、一括転送命令では特に注意が必要です。(例)



(コ.001、コ.000) × (コ.011、コ.010) の演算結果を(コ.735、コ.734、コ.733、コ.732)の4バイトに書込む命令です。

コ.734、コ.735の特殊領域に演算結果が書込まれてしまいます。

#### ① 7340~7347 (コ.734)

- 現在発生している異常内容のコードが格納される特殊レジスタです。
- 複数の異常が同時に発生した場合は優先順位の高い方の異常コードが入ります。
- 異常が回復すると異常コードはクリアされます。
- 異常コードについては3-7“自己診断”の項をご参照ください。

#### ② 7350 (乗除算命令桁数選択: W51のみ)

- 乗算命令(F-15, Fc15)、除算命令(F-16)の桁数の選択を行ないます。

命令	7350 OFF	7350 ON
F-15	レジスタ(4桁)間の乗算	レジスタ(8桁)間の乗算
Fc15	レジスタ(4桁)とBCD定数(3桁)の乗算	レジスタ(8桁)とBCD定数(3桁)の乗算
F-16	レジスタ(4桁)とレジスタ(2桁)の除算	レジスタ(8桁)とレジスタ(8桁)の除算

注1 Fc16は7350のON/OFFにかかわらず、「レジスタ(4桁)とBCD定数(2桁)の除算」です。

#### ③ 7354~7357 (フラグ)

- フラグに影響を与える応用命令の実行時、演算内容に応じてセットされます。
- 詳細は4-5(5)“データ処理命令とフラグ”をご参照ください。

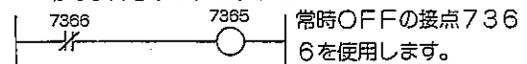
#### ④ 7360 (0.1秒クロック)、7364 (1秒クロック)

- CNT命令のクロックや各種応用命令のクロックとして使用します。

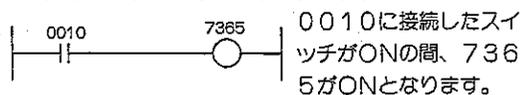


#### ⑤ 7365 (設定値変更スイッチ)

- 設定値変更モードではキープリレー(7000~7337)のセット、リセットが可能ですが、キープリレー以外のリレー(0000~4777)のセット、リセットやレジスタ(コ.000~コ.733、b.000~b.377、9000~9377)ファイルレジスタ(30000~37777)にプログラマ等の周辺装置からデータを書込むためには、7365をプログラム上ONとする必要があります。
- 常時ONとするプログラム



- 外部スイッチでONとするプログラム



#### ⑥ 7366 (常時OFFの接点)

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

3

⑦ 7367 (ゼロクロススイッチ)

○ゼロクロス同期が不要な場合、7367をONとします。

○ゼロクロス同期に関しては3-6(3)“スキャンサイクル”の項をご参照ください。

⑧ 7370~7377 (自己診断結果)

○自己診断の結果、異常であれば、異常内容に応じた接点がONとなります。

○詳細は、3-7 “自己診断” の項をご参照ください。

**注2** コンピュータリンク、リモートI/O、タータリンクの名オプションモジュールを使用しますと、7300~7337のキーリレー領域も特殊領域となります。詳細は各オプションの取扱説明書をご参照ください。

(4) TMR、CNT、MDのデータ格納領域

b.000~b.377の256バイトはTMR、CNTの現在値、MD命令のMD情報を格納する領域です。TMR、CNT、MDは合計128点で、1点当り、2バイトを使用します。TMR、CNT、MD番号とb. x xの領域の関係は次のようになります。

TMR、CNT、MD番号	データ格納領域
000	b.000、b.001
001	b.002、b.003
002	b.004、b.005
003	b.006、b.007
...	...
176	b.374、b.375
177	b.376、b.377

b.000~b.377をデータ処理命令(F-00等)で指定すれば、TMR、CNTの現在値を演算に使用することができます。

b.000~b.377のデータフォーマットを下表に示します。

	7	6	5	4	3	2	1	0
TMR	( $\times 10^0$ )				( $\times 10^{-1}$ )			
	"8"	"4"	"2"	"1"	"8"	"4"	"2"	"1"
	TMR (CNT) のとき 0	TMR のとき 0	リセット 注1	( $\times 10^2$ ) "1"	( $\times 10^1$ ) "8"	"4"	"2"	"1"
CNT	( $\times 10^1$ )				( $\times 10^0$ )			
	"8"	"4"	"2"	"1"	"8"	"4"	"2"	"1"
	TMR (CNT) のとき 0	CNT のとき 1	リセット 注1	( $\times 10^3$ ) "1"	( $\times 10^2$ ) "8"	"4"	"2"	"1"
MD	( $\times 10^1$ )				( $\times 10^0$ )			
	"8"	"4"	"2"	"1"	"8"	"4"	"2"	"1"
	MD のとき 1	入力情報 S <sub>1</sub> S <sub>2</sub> S <sub>3</sub>			( $\times 10^2$ ) "8"	"4"	"2"	"1"

**注1** 設定値変更モードで強制リセットすると0 (OFF) になります。通常1 (ON) となっています。

**注2** b.000~b.377では数値をBCDで扱います。

(5) リレー領域のバイトアドレスと

絶対アドレス

W16/W51は、AND、ORといったビット単位の演算のみではなく、四則演算や転送といったデータ処理の機能を豊富に備えたプログラマブルコントローラです。データ処理命令は、バイト単位で扱います。

入出力リレー、リンクリレー、補助リレー、キーリレーの各領域をデータ処理の対象とするとき、これらの領域をバイト単位で指定するバイト単位のアドレスが必要となります。W16/W51ではこれらの領域をバイト単位で指定するとき、バイトアドレスと絶対アドレスの2つのアドレス方式があります。

A. バイトアドレス

リレー番号と対応したバイト単位のアドレスで、4桁のリレー番号の最下位桁を捨てた上3桁にバイト単位であることを明確にするためコ。(コードの意味)を付加したものです。

(例) 

0137	0136	0135	0134	0133	0132	0131	0130
------	------	------	------	------	------	------	------

 のバイトアドレスはコ.013となります。

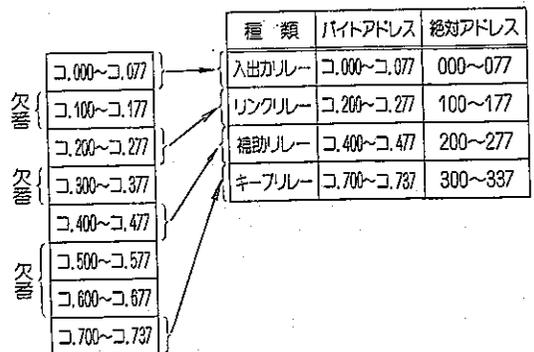
応用命令でソース、テストディネーションとして、リレー領域をバイト指定するとき、このバイトアドレスを使用します。

B. 絶対アドレス

リレー番号は、入出力リレー (0000~0777)、リンクリレー (2000~2777)、補助リレー (4000~4777)、キーリレー (7000~7377) と各グループごとに一連の番号が割当てられていますが、各グループ間に欠番となっているリレー番号があります。(たとえば入出力リレーとリンクリレーの間の1000~1777)

ところが実際のデータメモリ用RAMでは、各グループ間に空間を設けず、連続して配列しています。

絶対アドレスはグループ間の空間を詰めたとときのバイト単位のアドレスです。



(a) キーリレー領域、異常時の出力保持領域の設定には、この絶対アドレスを使用します。

(b) 応用命令には、コ. x x xの領域の欠番や、絶対アドレスを考慮する必要のある命令があります。詳細は4-5(7) “リレー領域を使用するとき” をご参照ください。

# (6) データメモリのアドレスマップ

1) 入力リレー、出力リレー (0000~0777)

1バイト(8ビット)  
単位の名称

絶対アドレス

0007	0006	0005	0004	0003	0002	0001	0000
0017	0016	0015	0014	0013	0012	0011	0010
0027	0026	0025	0024	0023	0022	0021	0020
0037	0036	0035	0034	0033	0032	0031	0030
0047	0046	0045	0044	0043	0042	0041	0040
0057	0056	0055	0054	0053	0052	0051	0050
0067	0066	0065	0064	0063	0062	0061	0060
0077	0076	0075	0074	0073	0072	0071	0070
0107	0106	0105	0104	0103	0102	0101	0100
0117	0116	0115	0114	0113	0112	0111	0110
0127	0126	0125	0124	0123	0122	0121	0120
0137	0136	0135	0134	0133	0132	0131	0130
0147	0146	0145	0144	0143	0142	0141	0140
0157	0156	0155	0154	0153	0152	0151	0150
0167	0166	0165	0164	0163	0162	0161	0160
0177	0176	0175	0174	0173	0172	0171	0170
0207	0206	0205	0204	0203	0202	0201	0200
0217	0216	0215	0214	0213	0212	0211	0210
0227	0226	0225	0224	0223	0222	0221	0220
0237	0236	0235	0234	0233	0232	0231	0230
0247	0246	0245	0244	0243	0242	0241	0240
0257	0256	0255	0254	0253	0252	0251	0250
0267	0266	0265	0264	0263	0262	0261	0260
0277	0276	0275	0274	0273	0272	0271	0270
0307	0306	0305	0304	0303	0302	0301	0300
0317	0316	0315	0314	0313	0312	0311	0310
0327	0326	0325	0324	0323	0322	0321	0320
0337	0336	0335	0334	0333	0332	0331	0330
0347	0346	0345	0344	0343	0342	0341	0340
0357	0356	0355	0354	0353	0352	0351	0350
0367	0366	0365	0364	0363	0362	0361	0360
0377	0376	0375	0374	0373	0372	0371	0370
0407	0406	0405	0404	0403	0402	0401	0400
0417	0416	0415	0414	0413	0412	0411	0410
0427	0426	0425	0424	0423	0422	0421	0420
0437	0436	0435	0434	0433	0432	0431	0430
0447	0446	0445	0444	0443	0442	0441	0440
0457	0456	0455	0454	0453	0452	0451	0450
0467	0466	0465	0464	0463	0462	0461	0460
0477	0476	0475	0474	0473	0472	0471	0470
0507	0506	0505	0504	0503	0502	0501	0500
0517	0516	0515	0514	0513	0512	0511	0510
0527	0526	0525	0524	0523	0522	0521	0520
0537	0536	0535	0534	0533	0532	0531	0530
0547	0546	0545	0544	0543	0542	0541	0540
0557	0556	0555	0554	0553	0552	0551	0550
0567	0566	0565	0564	0563	0562	0561	0560
0577	0576	0575	0574	0573	0572	0571	0570
0607	0606	0605	0604	0603	0602	0601	0600
0617	0616	0615	0614	0613	0612	0611	0610
0627	0626	0625	0624	0623	0622	0621	0620
0637	0636	0635	0634	0633	0632	0631	0630
0647	0646	0645	0644	0643	0642	0641	0640
0657	0656	0655	0654	0653	0652	0651	0650
0667	0666	0665	0664	0663	0662	0661	0660
0677	0676	0675	0674	0673	0672	0671	0670
0707	0706	0705	0704	0703	0702	0701	0700
0717	0716	0715	0714	0713	0712	0711	0710
0727	0726	0725	0724	0723	0722	0721	0720
0737	0736	0735	0734	0733	0732	0731	0730
0747	0746	0745	0744	0743	0742	0741	0740
0757	0756	0755	0754	0753	0752	0751	0750
0767	0766	0765	0764	0763	0762	0761	0760
0777	0776	0775	0774	0773	0772	0771	0770

コ. 000
コ. 001
コ. 002
コ. 003
コ. 004
コ. 005
コ. 006
コ. 007
コ. 010
コ. 011
コ. 012
コ. 013
コ. 014
コ. 015
コ. 016
コ. 017
コ. 020
コ. 021
コ. 022
コ. 023
コ. 024
コ. 025
コ. 026
コ. 027
コ. 030
コ. 031
コ. 032
コ. 033
コ. 034
コ. 035
コ. 036
コ. 037
コ. 040
コ. 041
コ. 042
コ. 043
コ. 044
コ. 045
コ. 046
コ. 047
コ. 050
コ. 051
コ. 052
コ. 053
コ. 054
コ. 055
コ. 056
コ. 057
コ. 060
コ. 061
コ. 062
コ. 063
コ. 064
コ. 065
コ. 066
コ. 067
コ. 070
コ. 071
コ. 072
コ. 073
コ. 074
コ. 075
コ. 076
コ. 077

000
001
002
003
004
005
006
007
010
011
012
013
014
015
016
017
020
021
022
023
024
025
026
027
030
031
032
033
034
035
036
037
040
041
042
043
044
045
046
047
050
051
052
053
054
055
056
057
060
061
062
063
064
065
066
067
070
071
072
073
074
075
076
077

コ. 036は左記の  
0360~0367の  
8ビットを示し  
ています。



2) リンクリレー (2000~2777)

1バイト(8ビット) 絶対アドレス  
単位の名称

2007	2006	2005	2004	2003	2002	2001	2000
2017	2016	2015	2014	2013	2012	2011	2010
2027	2026	2025	2024	2023	2022	2021	2020
2037	2036	2035	2034	2033	2032	2031	2030
2047	2046	2045	2044	2043	2042	2041	2040
2057	2056	2055	2054	2053	2052	2051	2050
2067	2066	2065	2064	2063	2062	2061	2060
2077	2076	2075	2074	2073	2072	2071	2070
2107	2106	2105	2104	2103	2102	2101	2100
2117	2116	2115	2114	2113	2112	2111	2110
2127	2126	2125	2124	2123	2122	2121	2120
2137	2136	2135	2134	2133	2132	2131	2130
2147	2146	2145	2144	2143	2142	2141	2140
2157	2156	2155	2154	2153	2152	2151	2150
2167	2166	2165	2164	2163	2162	2161	2160
2177	2176	2175	2174	2173	2172	2171	2170
2207	2206	2205	2204	2203	2202	2201	2200
2217	2216	2215	2214	2213	2212	2211	2210
2227	2226	2225	2224	2223	2222	2221	2220
2237	2236	2235	2234	2233	2232	2231	2230
2247	2246	2245	2244	2243	2242	2241	2240
2257	2256	2255	2254	2253	2252	2251	2250
2267	2266	2265	2264	2263	2262	2261	2260
2277	2276	2275	2274	2273	2272	2271	2270
2307	2306	2305	2304	2303	2302	2301	2300
2317	2316	2315	2314	2313	2312	2311	2310
2327	2326	2325	2324	2323	2322	2321	2320
2337	2336	2335	2334	2333	2332	2331	2330
2347	2346	2345	2344	2343	2342	2341	2340
2357	2356	2355	2354	2353	2352	2351	2350
2367	2366	2365	2364	2363	2362	2361	2360
2377	2376	2375	2374	2373	2372	2371	2370
2407	2406	2405	2404	2403	2402	2401	2400
2417	2416	2415	2414	2413	2412	2411	2410
2427	2426	2425	2424	2423	2422	2421	2420
2437	2436	2435	2434	2433	2432	2431	2430
2447	2446	2445	2444	2443	2442	2441	2440
2457	2456	2455	2454	2453	2452	2451	2450
2467	2466	2465	2464	2463	2462	2461	2460
2477	2476	2475	2474	2473	2472	2471	2470
2507	2506	2505	2504	2503	2502	2501	2500
2517	2516	2515	2514	2513	2512	2511	2510
2527	2526	2525	2524	2523	2522	2521	2520
2537	2536	2535	2534	2533	2532	2531	2530
2547	2546	2545	2544	2543	2542	2541	2540
2557	2556	2555	2554	2553	2552	2551	2550
2567	2566	2565	2564	2563	2562	2561	2560
2577	2576	2575	2574	2573	2572	2571	2570
2607	2606	2605	2604	2603	2602	2601	2600
2617	2616	2615	2614	2613	2612	2611	2610
2627	2626	2625	2624	2623	2622	2621	2620
2637	2636	2635	2634	2633	2632	2631	2630
2647	2646	2645	2644	2643	2642	2641	2640
2657	2656	2655	2654	2653	2652	2651	2650
2667	2666	2665	2664	2663	2662	2661	2660
2677	2676	2675	2674	2673	2672	2671	2670
2707	2706	2705	2704	2703	2702	2701	2700
2717	2716	2715	2714	2713	2712	2711	2710
2727	2726	2725	2724	2723	2722	2721	2720
2737	2736	2735	2734	2733	2732	2731	2730
2747	2746	2745	2744	2743	2742	2741	2740
2757	2756	2755	2754	2753	2752	2751	2750
2767	2766	2765	2764	2763	2762	2761	2760
2777	2776	2775	2774	2773	2772	2771	2770

コ. 200
コ. 201
コ. 202
コ. 203
コ. 204
コ. 205
コ. 206
コ. 207
コ. 210
コ. 211
コ. 212
コ. 213
コ. 214
コ. 215
コ. 216
コ. 217
コ. 220
コ. 221
コ. 222
コ. 223
コ. 224
コ. 225
コ. 226
コ. 227
コ. 230
コ. 231
コ. 232
コ. 233
コ. 234
コ. 235
コ. 236
コ. 237
コ. 240
コ. 241
コ. 242
コ. 243
コ. 244
コ. 245
コ. 246
コ. 247
コ. 250
コ. 251
コ. 252
コ. 253
コ. 254
コ. 255
コ. 256
コ. 257
コ. 260
コ. 261
コ. 262
コ. 263
コ. 264
コ. 265
コ. 266
コ. 267
コ. 270
コ. 271
コ. 272
コ. 273
コ. 274
コ. 275
コ. 276
コ. 277

100
101
102
103
104
105
106
107
110
111
112
113
114
115
116
117
120
121
122
123
124
125
126
127
130
131
132
133
134
135
136
137
140
141
142
143
144
145
146
147
150
151
152
153
154
155
156
157
160
161
162
163
164
165
166
167
170
171
172
173
174
175
176
177

3) 補助リレー (4000~4777)

1バイト(8ビット) 絶対アドレス  
単位の名称

4007	4006	4005	4004	4003	4002	4001	4000
4017	4016	4015	4014	4013	4012	4011	4010
4027	4026	4025	4024	4023	4022	4021	4020
4037	4036	4035	4034	4033	4032	4031	4030
4047	4046	4045	4044	4043	4042	4041	4040
4057	4056	4055	4054	4053	4052	4051	4050
4067	4066	4065	4064	4063	4062	4061	4060
4077	4076	4075	4074	4073	4072	4071	4070
4107	4106	4105	4104	4103	4102	4101	4100
4117	4116	4115	4114	4113	4112	4111	4110
4127	4126	4125	4124	4123	4122	4121	4120
4137	4136	4135	4134	4133	4132	4131	4130
4147	4146	4145	4144	4143	4142	4141	4140
4157	4156	4155	4154	4153	4152	4151	4150
4167	4166	4165	4164	4163	4162	4161	4160
4177	4176	4175	4174	4173	4172	4171	4170
4207	4206	4205	4204	4203	4202	4201	4200
4217	4216	4215	4214	4213	4212	4211	4210
4227	4226	4225	4224	4223	4222	4221	4220
4237	4236	4235	4234	4233	4232	4231	4230
4247	4246	4245	4244	4243	4242	4241	4240
4257	4256	4255	4254	4253	4252	4251	4250
4267	4266	4265	4264	4263	4262	4261	4260
4277	4276	4275	4274	4273	4272	4271	4270
4307	4306	4305	4304	4303	4302	4301	4300
4317	4316	4315	4314	4313	4312	4311	4310
4327	4326	4325	4324	4323	4322	4321	4320
4337	4336	4335	4334	4333	4332	4331	4330
4347	4346	4345	4344	4343	4342	4341	4340
4357	4356	4355	4354	4353	4352	4351	4350
4367	4366	4365	4364	4363	4362	4361	4360
4377	4376	4375	4374	4373	4372	4371	4370
4407	4406	4405	4404	4403	4402	4401	4400
4417	4416	4415	4414	4413	4412	4411	4410
4427	4426	4425	4424	4423	4422	4421	4420
4437	4436	4435	4434	4433	4432	4431	4430
4447	4446	4445	4444	4443	4442	4441	4440
4457	4456	4455	4454	4453	4452	4451	4450
4467	4466	4465	4464	4463	4462	4461	4460
4477	4476	4475	4474	4473	4472	4471	4470
4507	4506	4505	4504	4503	4502	4501	4500
4517	4516	4515	4514	4513	4512	4511	4510
4527	4526	4525	4524	4523	4522	4521	4520
4537	4536	4535	4534	4533	4532	4531	4530
4547	4546	4545	4544	4543	4542	4541	4540
4557	4556	4555	4554	4553	4552	4551	4550
4567	4566	4565	4564	4563	4562	4561	4560
4577	4576	4575	4574	4573	4572	4571	4570
4607	4606	4605	4604	4603	4602	4601	4600
4617	4616	4615	4614	4613	4612	4611	4610
4627	4626	4625	4624	4623	4622	4621	4620
4637	4636	4635	4634	4633	4632	4631	4630
4647	4646	4645	4644	4643	4642	4641	4640
4657	4656	4655	4654	4653	4652	4651	4650
4667	4666	4665	4664	4663	4662	4661	4660
4677	4676	4675	4674	4673	4672	4671	4670
4707	4706	4705	4704	4703	4702	4701	4700
4717	4716	4715	4714	4713	4712	4711	4710
4727	4726	4725	4724	4723	4722	4721	4720
4737	4736	4735	4734	4733	4732	4731	4730
4747	4746	4745	4744	4743	4742	4741	4740
4757	4756	4755	4754	4753	4752	4751	4750
4767	4766	4765	4764	4763	4762	4761	4760
4777	4776	4775	4774	4773	4772	4771	4770

コ. 400
コ. 401
コ. 402
コ. 403
コ. 404
コ. 405
コ. 406
コ. 407
コ. 410
コ. 411
コ. 412
コ. 413
コ. 414
コ. 415
コ. 416
コ. 417
コ. 420
コ. 421
コ. 422
コ. 423
コ. 424
コ. 425
コ. 426
コ. 427
コ. 430
コ. 431
コ. 432
コ. 433
コ. 434
コ. 435
コ. 436
コ. 437
コ. 440
コ. 441
コ. 442
コ. 443
コ. 444
コ. 445
コ. 446
コ. 447
コ. 450
コ. 451
コ. 452
コ. 453
コ. 454
コ. 455
コ. 456
コ. 457
コ. 460
コ. 461
コ. 462
コ. 463
コ. 464
コ. 465
コ. 466
コ. 467
コ. 470
コ. 471
コ. 472
コ. 473
コ. 474
コ. 475
コ. 476
コ. 477

200
201
202
203
204
205
206
207
210
211
212
213
214
215
216
217
220
221
222
223
224
225
226
227
230
231
232
233
234
235
236
237
240
241
242
243
244
245
246
247
250
251
252
253
254
255
256
257
260
261
262
263
264
265
266
267
270
271
272
273
274
275
276
277



4) キープリレー (7000~7377)

1バイト(8ビット)

絶対アドレス

単位の名称

7007	7006	7005	7004	7003	7002	7001	7000
7017	7016	7015	7014	7013	7012	7011	7010
7027	7026	7025	7024	7023	7022	7021	7020
7037	7036	7035	7034	7033	7032	7031	7030
7047	7046	7045	7044	7043	7042	7041	7040
7057	7056	7055	7054	7053	7052	7051	7050
7067	7066	7065	7064	7063	7062	7061	7060
7077	7076	7075	7074	7073	7072	7071	7070
7107	7106	7105	7104	7103	7102	7101	7100
7117	7116	7115	7114	7113	7112	7111	7110
7127	7126	7125	7124	7123	7122	7121	7120
7137	7136	7135	7134	7133	7132	7131	7130
7147	7146	7145	7144	7143	7142	7141	7140
7157	7156	7155	7154	7153	7152	7151	7150
7167	7166	7165	7164	7163	7162	7161	7160
7177	7176	7175	7174	7173	7172	7171	7170
7207	7206	7205	7204	7203	7202	7201	7200
7217	7216	7215	7214	7213	7212	7211	7210
7227	7226	7225	7224	7223	7222	7221	7220
7237	7236	7235	7234	7233	7232	7231	7230
7247	7246	7245	7244	7243	7242	7241	7240
7257	7256	7255	7254	7253	7252	7251	7250
7267	7266	7265	7264	7263	7262	7261	7260
7277	7276	7275	7274	7273	7272	7271	7270
7307	7306	7305	7304	7303	7302	7301	7300
7317	7316	7315	7314	7313	7312	7311	7310
7327	7326	7325	7324	7323	7322	7321	7320
7337	7336	7335	7334	7333	7332	7331	7330
7347	7346	7345	7344	7343	7342	7341	7340
7357	7356	7355	7354	7353	7352	7351	7350
7367	7366	7365	7364	7363	7362	7361	7360
7377	7376	7375	7374	7373	7372	7371	7370

コ. 700
コ. 701
コ. 702
コ. 703
コ. 704
コ. 705
コ. 706
コ. 707
コ. 710
コ. 711
コ. 712
コ. 713
コ. 714
コ. 715
コ. 716
コ. 717
コ. 720
コ. 721
コ. 722
コ. 723
コ. 724
コ. 725
コ. 726
コ. 727
コ. 730
コ. 731
コ. 732
コ. 733
コ. 734
コ. 735
コ. 736
コ. 737

300
301
302
303
304
305
306
307
310
311
312
313
314
315
316
317
320
321
322
323
324
325
326
327
330
331
332
333
334
335
336
337

特殊  
レジスタ  
特殊  
プリレー

特殊レジスタ

(キープリレーの特殊領域)

7340	自己診断結果の異常コードを収納する特殊レジスタでバイトアドレスコ.734として扱います。	7360	0.1秒フロック
7341		7361	
7342		7362	
7343		7363	
7344		7364	1秒フロック
7345		7365	設定値変更スイッチ
7346		7366	常時OFFの接点
7347	7367	ゼロクロススイッチ	
7350	乗除算命令桁数選択(W51のみ)	7370	メモリ異常
7351		7371	CPU異常
7352		7372	電池異常
7353		7373	入出力異常
7354	ノンキャリアフラグ エラーフラグ キャリアフラグ ゼロフラグ	7374	オプション異常
7355		7375	
7356		7376	ROM異常
7357		7377	電源異常

オプションモジュール使用時は、7300~7337の領域も特殊領域となります。詳細は各オプションの取扱説明書をご参照ください。

5) TMR、CNT、MD

000	b. 000	040	b. 100	100	b. 200	140	b. 300
	b. 001		b. 101		b. 201		b. 301
001	b. 002	041	b. 102	101	b. 202	141	b. 302
	b. 003		b. 103		b. 203		b. 303
002	b. 004	042	b. 104	102	b. 204	142	b. 304
	b. 005		b. 105		b. 205		b. 305
003	b. 006	043	b. 106	103	b. 206	143	b. 306
	b. 007		b. 107		b. 207		b. 307
004	b. 010	044	b. 110	104	b. 210	144	b. 310
	b. 011		b. 111		b. 211		b. 311
005	b. 012	045	b. 112	105	b. 212	145	b. 312
	b. 013		b. 113		b. 213		b. 313
006	b. 014	046	b. 114	106	b. 214	146	b. 314
	b. 015		b. 115		b. 215		b. 315
007	b. 016	047	b. 116	107	b. 216	147	b. 316
	b. 017		b. 117		b. 217		b. 317
010	b. 020	050	b. 120	110	b. 220	150	b. 320
	b. 021		b. 121		b. 221		b. 321
011	b. 022	051	b. 122	111	b. 222	151	b. 322
	b. 023		b. 123		b. 223		b. 323
012	b. 024	052	b. 124	112	b. 224	152	b. 324
	b. 025		b. 125		b. 225		b. 325
013	b. 026	053	b. 126	113	b. 226	153	b. 326
	b. 027		b. 127		b. 227		b. 327
014	b. 030	054	b. 130	114	b. 230	154	b. 330
	b. 031		b. 131		b. 231		b. 331
015	b. 032	055	b. 132	115	b. 232	155	b. 332
	b. 033		b. 133		b. 233		b. 333
016	b. 034	056	b. 134	116	b. 234	156	b. 334
	b. 035		b. 135		b. 235		b. 335
017	b. 036	057	b. 136	117	b. 236	157	b. 336
	b. 037		b. 137		b. 237		b. 337
020	b. 040	060	b. 140	120	b. 240	160	b. 340
	b. 041		b. 141		b. 241		b. 341
021	b. 042	061	b. 142	121	b. 242	161	b. 342
	b. 043		b. 143		b. 243		b. 343
022	b. 044	062	b. 144	122	b. 244	162	b. 344
	b. 045		b. 145		b. 245		b. 345
023	b. 046	063	b. 146	123	b. 246	163	b. 346
	b. 047		b. 147		b. 247		b. 347
024	b. 050	064	b. 150	124	b. 250	164	b. 350
	b. 051		b. 151		b. 251		b. 351
025	b. 052	065	b. 152	125	b. 252	165	b. 352
	b. 053		b. 153		b. 253		b. 353
026	b. 054	066	b. 154	126	b. 254	166	b. 354
	b. 055		b. 155		b. 255		b. 355
027	b. 056	067	b. 156	127	b. 256	167	b. 356
	b. 057		b. 157		b. 257		b. 357
030	b. 060	070	b. 160	130	b. 260	170	b. 360
	b. 061		b. 161		b. 261		b. 361
031	b. 062	071	b. 162	131	b. 262	171	b. 362
	b. 063		b. 163		b. 263		b. 363
032	b. 064	072	b. 164	132	b. 264	172	b. 364
	b. 065		b. 165		b. 265		b. 365
033	b. 066	073	b. 166	133	b. 266	173	b. 366
	b. 067		b. 167		b. 267		b. 367
034	b. 070	074	b. 170	134	b. 270	174	b. 370
	b. 071		b. 171		b. 271		b. 371
035	b. 072	075	b. 172	135	b. 272	175	b. 372
	b. 073		b. 173		b. 273		b. 373
036	b. 074	076	b. 174	136	b. 274	176	b. 374
	b. 075		b. 175		b. 275		b. 375
037	b. 076	077	b. 176	137	b. 276	177	b. 376
	b. 077		b. 177		b. 277		b. 377

TMR、CNT、MD番号160  
の現在値はb. 340、b. 341の2  
バイトに格納されます。



6) レジスタ (9000~9377)

9007	9006	9005	9004	9003	9002	9001	9000
9017	9016	9015	9014	9013	9012	9011	9010
9027	9026	9025	9024	9023	9022	9021	9020
9037	9036	9035	9034	9033	9032	9031	9030
9047	9046	9045	9044	9043	9042	9041	9040
9057	9056	9055	9054	9053	9052	9051	9050
9067	9066	9065	9064	9063	9062	9061	9060
9077	9076	9075	9074	9073	9072	9071	9070
9107	9106	9105	9104	9103	9102	9101	9100
9117	9116	9115	9114	9113	9112	9111	9110
9127	9126	9125	9124	9123	9122	9121	9120
9137	9136	9135	9134	9133	9132	9131	9130
9147	9146	9145	9144	9143	9142	9141	9140
9157	9156	9155	9154	9153	9152	9151	9150
9167	9166	9165	9164	9163	9162	9161	9160
9177	9176	9175	9174	9173	9172	9171	9170

9207	9206	9205	9204	9203	9202	9201	9200
9217	9216	9215	9214	9213	9212	9211	9210
9227	9226	9225	9224	9223	9222	9221	9220
9237	9236	9235	9234	9233	9232	9231	9230
9247	9246	9245	9244	9243	9242	9241	9240
9257	9256	9255	9254	9253	9252	9251	9250
9267	9266	9265	9264	9263	9262	9261	9260
9277	9276	9275	9274	9273	9272	9271	9270
9307	9306	9305	9304	9303	9302	9301	9300
9317	9316	9315	9314	9313	9312	9311	9310
9327	9326	9325	9324	9323	9322	9321	9320
9337	9336	9335	9334	9333	9332	9331	9330
9347	9346	9345	9344	9343	9342	9341	9340
9357	9356	9355	9354	9353	9352	9351	9350
9367	9366	9365	9364	9363	9362	9361	9360
9377	9376	9375	9374	9373	9372	9371	9370

7) ファイルレジスタ (30000~37777)

30007	30006	30005	30004	30003	30002	30001	30000
§	§	§	§	§	§	§	§
30377	30376	30375	30374	30373	30372	30371	30370
30407	30406	30405	30404	30403	30402	30401	30400
§	§	§	§	§	§	§	§
30777	30776	30775	30774	30773	30772	30771	30770
31007	31006	31005	31004	31003	31002	31001	31000
§	§	§	§	§	§	§	§
31377	31376	31375	31374	31373	31372	31371	31370
31407	31406	31405	31404	31403	31402	31401	31400
§	§	§	§	§	§	§	§
31777	31776	31775	31774	31773	31772	31771	31770
32007	32006	32005	32004	32003	32002	32001	32000
§	§	§	§	§	§	§	§
32377	32376	32375	32374	32373	32372	32371	32370
32407	32406	32405	32404	32403	32402	32401	32400
§	§	§	§	§	§	§	§
32777	32776	32775	32774	32773	32772	32771	32770
33007	33006	33005	33004	33003	33002	33001	33000
§	§	§	§	§	§	§	§
33377	33376	33375	33374	33373	33372	33371	33370
33407	33406	33405	33404	33403	33402	33401	33400
§	§	§	§	§	§	§	§
33777	33776	33775	33774	33773	33772	33771	33770

34007	34006	34005	34004	34003	34002	34001	34000
§	§	§	§	§	§	§	§
34377	34376	34375	34374	34373	34372	34371	34370
34407	34406	34405	34404	34403	34402	34401	34400
§	§	§	§	§	§	§	§
34777	34776	34775	34774	34773	34772	34771	34770
35007	35006	35005	35004	35003	35002	35001	35000
§	§	§	§	§	§	§	§
35377	35376	35375	35374	35373	35372	35371	35370
35407	35406	35405	35404	35403	35402	35401	35400
§	§	§	§	§	§	§	§
35777	35776	35775	35774	35773	35772	35771	35770
36007	36006	36005	36004	36003	36002	36001	36000
§	§	§	§	§	§	§	§
36377	36376	36375	36374	36373	36372	36371	36370
36407	36406	36405	36404	36403	36402	36401	36400
§	§	§	§	§	§	§	§
36777	36776	36775	36774	36773	36772	36771	36770
37007	37006	37005	37004	37003	37002	37001	37000
§	§	§	§	§	§	§	§
37377	37376	37375	37374	37373	37372	37371	37370
37407	37406	37405	37404	37403	37402	37401	37400
§	§	§	§	§	§	§	§
37777	37776	37775	37774	37773	37772	37771	37770

ファイルレジスタ容量 { 2kバイト設定時 30000~33777  
4kバイト設定時 30000~37777

### 3-3 システムメモリ

システムメモリは#000~#377の256バイトのメモリで、電池でバックアップされています。

- #000~#177の128バイトは、コントロールユニット内のCPUが使用する領域で、プログラマ等の周辺装置から書込むことはできません。(読出しは可能)
- #200~#377の128バイトは、ユーザ開放領域で、PCの各種機能の設定、異常コードの格納、データリンク等のオプション類の機能設定等に使用します。

ユーザに非開放の領域		ユーザ開放領域	
#000	#100	#200	#300
#001	#101	#201	#301
#002	#102	#202	#302
#003	#103	#203	#303
#004	#104	#204	#304
#005	#105	#205	#305
#006	#106	#206	#306
#007	#107	#207	#307
#010	#110	#210	#310
}	}	}	}
#074	#174	#274	#374
#075	#175	#275	#375
#076	#176	#276	#376
#077	#177	#277	#377

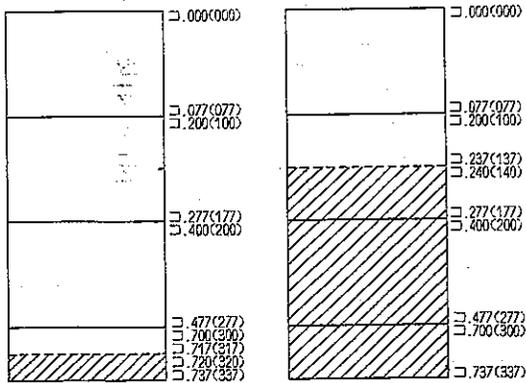
(1) コントロールユニットの各種機能を設定する領域

アドレス	設定項目	内容																																
#200	キーブリレー領域	キーブリレー領域を初期の状態から増減したい場合に設定します。 設定数値はデータメモリの絶対アドレスを8進数で設定します。 なお、初期状態は7000~のため300が設定されています。																																
		<table border="1"> <thead> <tr> <th>リレー</th> <th>リレーアドレス</th> <th>バイトアドレス</th> <th>絶対アドレス</th> </tr> </thead> <tbody> <tr> <td rowspan="2">入出力リレー</td> <td>0000</td> <td>コ.000</td> <td>000</td> </tr> <tr> <td>0777</td> <td>コ.077</td> <td>077</td> </tr> <tr> <td rowspan="2">リンクリレー</td> <td>2000</td> <td>コ.200</td> <td>100</td> </tr> <tr> <td>2777</td> <td>コ.277</td> <td>177</td> </tr> <tr> <td rowspan="2">補助リレー</td> <td>4000</td> <td>コ.400</td> <td>200</td> </tr> <tr> <td>4777</td> <td>コ.477</td> <td>277</td> </tr> <tr> <td rowspan="2">キーブリレー 特殊リレー、特殊 レジスタを除く</td> <td>7000</td> <td>コ.700</td> <td>300</td> </tr> <tr> <td>7337</td> <td>コ.733</td> <td>333</td> </tr> </tbody> </table>	リレー	リレーアドレス	バイトアドレス	絶対アドレス	入出力リレー	0000	コ.000	000	0777	コ.077	077	リンクリレー	2000	コ.200	100	2777	コ.277	177	補助リレー	4000	コ.400	200	4777	コ.477	277	キーブリレー 特殊リレー、特殊 レジスタを除く	7000	コ.700	300	7337	コ.733	333
		リレー	リレーアドレス	バイトアドレス	絶対アドレス																													
		入出力リレー	0000	コ.000	000																													
			0777	コ.077	077																													
		リンクリレー	2000	コ.200	100																													
2777	コ.277		177																															
補助リレー	4000	コ.400	200																															
	4777	コ.477	277																															
キーブリレー 特殊リレー、特殊 レジスタを除く	7000	コ.700	300																															
	7337	コ.733	333																															
		8点(1バイト)単位で設定できます。																																
#201	TMRのリセット 条件設定	TMR命令の復電時の状態を設定します。 設定数値は8進数で設定し、 000……復電時リセット 001……停電時の状態記憶 のようになります。 初期状態は000に設定されています。																																
#202	CNTのリセット 条件設定	CNT命令、F-60(F/B SFR)、F-62(U/D CNT)のリセット入力条件を設定します。 設定数値は8進数で設定し、 000……ONでリセット 001……OFFでリセット のようになります。 初期状態は000に設定されています。																																

3

アドレス	設定項目	内容
#203	出力保持アドレス	本体停止時に出力ユニットの出力を保持する出力リレーの絶対アドレスを8進数で設定します。 初期状態は“000”のためコ.000以降の全出力ユニットが出力を保持しています。 設定を“002”としたときはコ.000、コ.001のみ本体停止時に出力がOFFとなりコ.002以降は出力を保持します。 初期状態は000に設定されています。
#204	プログラムメモリ容量	プログラムメモリの容量を8進数で設定します。 000……2.5K語 004……6.5K語 001……3.5K語 005……7.5K語 002……4.5K語 003……5.5K語 初期状態は001に設定されています。
#205	ファイルレジスタ容量	ファイルレジスタの容量を8進数で設定します。 000……使用せず 001……2kバイト 002……4kバイト 初期状態は000に設定されています。

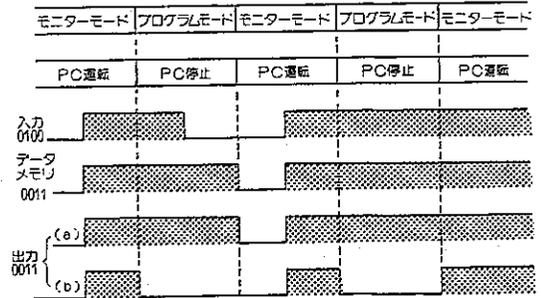
- 注1** キーブリレー領域を縮小する場合、コ.734(絶対アドレス334)(データリンク、コンピュータリンク、リモートI/O等のオプションモジュールを使用するときはコ.731)以後の領域のキー機能を解除しないようにしてください。
- 注2** キーブリレー領域の設定を変更した場合、停電後の復電時に保持される領域(キー領域)は斜線の範囲となります。( )内は絶対アドレスを示します。



#200に320を設定

#200に140を設定

転が停止したときに出力ユニットのラッチがリセットされ、出力がOFFとなります。ただし、データメモリはリセットされませんのでご注意ください。



- (a)……0011が出力保持領域にあるとき  
(b)……0011が出力保持解除領域にあるとき

- 3 注3** #202のカウンタのリセット条件を変更すると、CNT命令以外にF-60、F-62の各命令のリセット条件も同時に影響を受けます。
- 注4** #203の出力保持アドレスの設定はコ.737(絶対アドレス337)まで書込めますが、コ.201(絶対アドレス101)以後の設定は無意味です。出力保持を解除した領域は、PCをプログラムモードに切換えたり、自己診断の結果PCの運

自己診断の結果、異常が検知されPCの運転が停止する場合、異常内容によっては、出力保持を解除した領域の出力ユニットをOFFにできない場合があります。PC異常時にOFFにする必要がある出力は、コントロールユニット(ZW-501CU)の停止出力(トライアック出力、AC100V/200V、1A)を直列に接続してください。(2-2 “システム設計に際しての留意事項”参照)

**注5** #204と#205の設定は、増設RAMユニットの装着状態をCPUが判断し、メモリ容量以上の設定ができないようになっています。3-4“プログラムメモリとファイルレジスタ”の項をご参照ください。

② PCの異常コードを格納する領域

#210 ～#217	自己診断結果の 異常コード	自己診断の結果、異常と判断した場合、異常内容に応じ、異常コードが格納されます。
---------------	------------------	---

#210～#217はシフトレジスタとして働き、8回の異常発生を記憶することができます。異常コードの詳細は3-7“自己診断”の項をご参照ください。

**注1** 異常が解消してもクリアされませんので、クリアする必要があるときは、プログラマ等の周辺装置で00を書き込んでください。

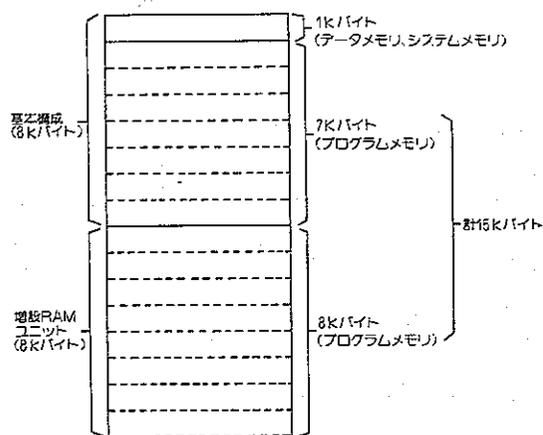
③ コンピュータリンク、テータリンク、リモートI/Oの各オプションが使用する領域

#206	コンピュータリンク使用時のPC番号	コンピュータリンクモジュールからモジュール内のPC番号設定スイッチの内容が書き込まれます。
#207	テータリンク使用時のPC番号	テータリンクモジュールからモジュール内のPC番号設定スイッチの内容が書き込まれます。
#220	テータリンク、リモートI/Oの設定ミス	各モジュールの設定スイッチの設定が異常のとき、30 (BCD) が各モジュールから書き込まれます。
	リモートI/Oの通信エラー	リモートI/O子局との通信が異常のとき、31 (BCD) がリモートI/Oモジュールから書き込まれます。
#221	リモートI/O異常子局の局番	異常発生中のリモート子局の局番が、リモートI/Oモジュールから書き込まれます。
#300 ～#357	テータリンクの運転情報	テータリンクの親局となるPCに、プログラマから転送/バイト数、受信局の指定情報を登録します。子局のPCには親局から伝送され自動的に書き込まれます。
#360	テータリンク局数	テータリンクの親局となるPCに、プログラマからテータリンクで接続されるPCの局数を書き込みます。
#361	テータリンクスタートスイッチ	テータリンクの親局となるPCでプログラマから0を書き込むとリンク動作がストップ、1を書き込むとスタートします。

## 3-4 プログラムメモリとファイルレジスタ

### (1) プログラムメモリ

プログラムメモリとは、ユーザプログラムを書込む領域で、PCが運転中はプログラムの先頭アドレスから順次読み出され、プログラム内容に応じて演算が行われます。W16/W51では基本構成時、電池で停電保護された8kバイトのRAMが標準実装されています。このうち1kバイトはデータメモリおよびシステムメモリに、残りの7kバイトがプログラムメモリに割当てられます。W51では増設RAMユニット(ZW-501MA)を使用しますと、プログラムメモリを15kバイトまで拡張できます。(W16では増設RAMユニットを使用できません)



命令には1語命令、2語命令、3語命令、4語命令があり、1語は2バイトで構成されます。

	代表的な命令	使用バイト数
1語命令	STR、AND等	2
2語命令	TMR、CNT等	4
3語命令	F-00、F-01等	6
4語命令	F-10、F-11等	8

プログラムメモリの容量を表現するとき、一般にバイト数ではなく、語数を用います。従って1kバイトは0.5k語と表現されます。

	プログラムメモリ容量	
	バイト数	語数
基本構成時	7kバイト	3.5k語
増設RAMユニット使用時	15kバイト	7.5k語

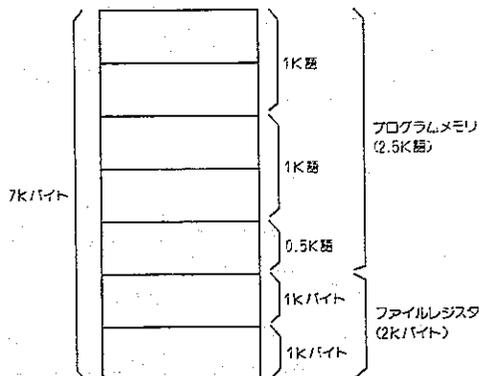
注1 1kは1024を示します。従って3.5k語は正確には3584語となります。

### (2) ファイルレジスタ

#### (1) ファイルレジスタ容量の設定

ファイルレジスタはプログラムメモリの一部をデータメモリに転用するものです。

たとえば、標準実装状態の7kバイトのうち、プログラムメモリを5kバイト(2.5k語)に制限すると、残りの2kバイトをファイルレジスタでできます。



システムメモリ#204にプログラムメモリの容量を、#205にファイルレジスタの容量を設定します。

プログラムメモリの容量		ファイルレジスタの容量			
システムメモリ #204	000	2.5k語	システムメモリ #205	000	使用せず
	001	3.5k語		001	2kバイト
	002	4.5k語		002	4kバイト
	003	5.5k語			
	004	6.5k語			
	005	7.5k語			

プログラムメモリ容量と、設定可能なファイルレジスタ容量の関係は次のようになります。

	プログラムメモリ		ファイルレジスタ	
	容量	アドレス	容量	アドレス
基本構成時	3.5k語	00000~06777	使用不可	
	2.5k語	00000~04777	2kバイト	30000~33777
増設RAMユニット (ZW-501MA) 使用時	7.5k語	00000~16777	使用不可	
	6.5k語	00000~14777		
	5.5k語	00000~12777	2kバイト	30000~33777
	4.5k語	00000~10777		
	3.5k語	00000~06777		
2.5k語	00000~04777			

注1 増設RAMユニット使用時、プログラムメモリ容量を4.5K語、3.5K語、2.5K語とした場合も、ファイルレジスタ容量は最大4kバイトに制限しています。またプログラムメモリ容量を6.5K語とした場合、ファイルレジスタを設定できません。

注2 前頁以外の組合せでプログラムメモリ容量、ファイルレジスタ容量を設定しようとした場合、プログラマ等の周辺装置がこれを判断し設定を受付けません。システムメモリ#204と#205では#204(プログラムメモリ容量)が優先されます。

(例1)基本構成時、#204に001(3.5K語)設定後、#205に001(2kバイト)を書込みません。逆に#205に001((2kバイト)設定後、#204に001(3.5K語)を書込むと、#205は自動的に000(ファイル使用せず)に変更されます。

(例2)増設RAMユニット装着状態で#204に004(6.5K語)設定後、#205に001(2kバイト)を書込みません。

### ②ファイルレジスタを使用できる命令

ファイルレジスタは9000~9377のレジスタと同じくバイト指定可能なデータメモリですが、以下の6種の応用命令でのみ使用可能です。

- ①F-05(分配)
- ②F-06(抽出)
- ③F-70(nバイト一括転送)
- ④F-71(8進定数一括転送)
- ⑤F-72(ファイルレジスタへの分配)
- ⑥F-73(ファイルレジスタからの抽出)

F-10等の命令でファイルレジスタを直接指定できませんので、上記6種の命令で一般のデータメモリ領域(C, XXX, b. XXX, 9XXX)に転送後、演算する必要があります。

### ③プログラマからの書込

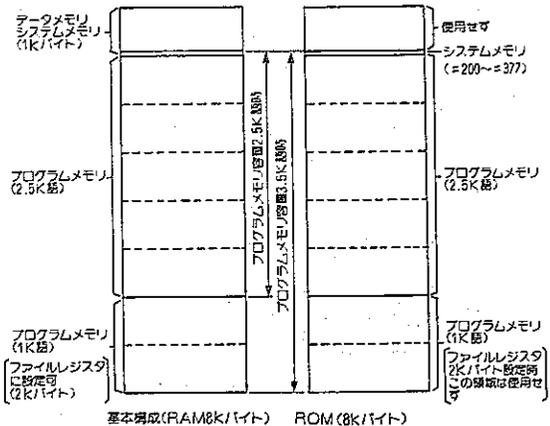
ファイルレジスタにプログラマ等の周辺装置から直接データを書込むことができます。キープリューの特殊領域の7365(設定値変更スイッチ)をONとするプログラムを書込んだ状態で、設定値変更モードとすると、周辺装置のキー操作によりデータの書込みが可能となります。7365(設定値変更スイッチ)に関しては、3-2(3)“キープリューの特殊領域”をご参照ください。

## (3)ROM運転

W16ではROM単品をW51ではROMユニット(ZW-501MO/ZW-501MO2)を使用しますと、プログラムメモリ化できます。プログラムメモリをROM化した場合にも、RAM領域を使ってファイルレジスタを使用できます。ROM運転時のプログラムメモリ容量、ファイルレジスタ容量は次の組合せが可能です。

	プログラムメモリ		ファイルレジスタ	
	容量	アドレス	容量	アドレス
ROMまたはROMユニット(ZW-501MO)使用時	3.5K語	00000~06777	使用せず	—
	2.5K語	00000~04777	使用せず	—
	2.5K語	00000~04777	2kバイト	30000~33777
ROMユニット(ZW-501MO2)使用時	7.5K語	00000~16777	使用せず	—
	6.5K語	00000~14777		
	5.5K語	00000~12777	2kバイト	30000~33777
	4.5K語	00000~10777		
	3.5K語	00000~06777	4kバイト	30000~37777
	2.5K語	00000~04777		

ROMへの書込みには、PROMライタ(ZW-100WR1)をご使用願います。PROMライタ(ZW-100WR1)によりROMへの書込みを行いますと、システムメモリ領域の#200~377もプログラムメモリとともにROMに書込まれます。



書込み完了後のROMまたはROMユニットをそれぞれW16またはW51のコントロールユニットに装着しますと、

①電源投入時、システムメモリの#200~#377の領域はROMから基本構成のRAMに転送され、以後RAMのシステムメモリ領域をCPUは参照して運転が行われます。

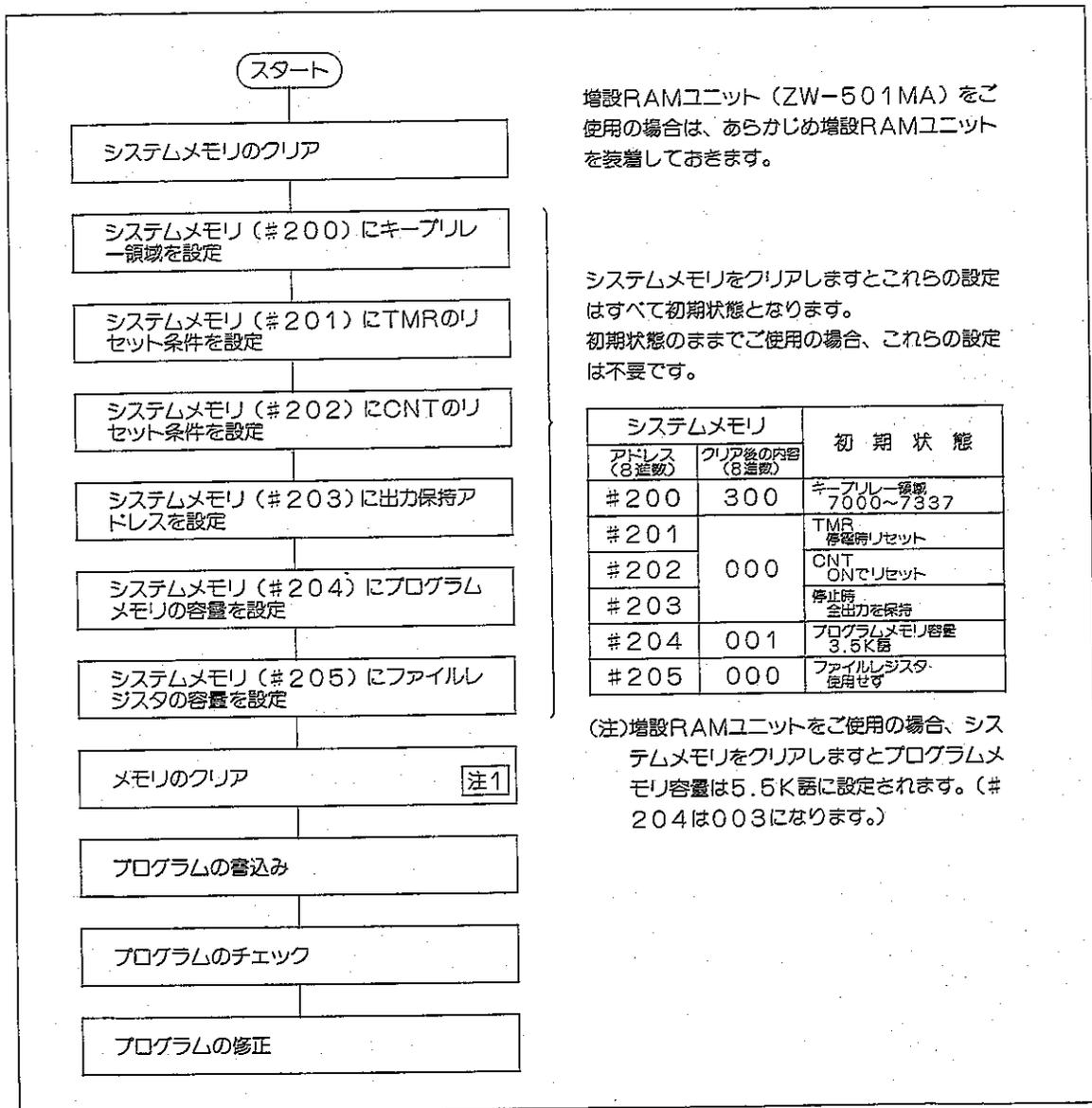
②プログラムメモリはROM側の内容に基づき演算が実行されます。

注1 システムメモリの#210~#217(異常コード格納)はROMには書込まれません。

注2 ROM運転中にプログラムモードにしてシステムメモリ#200~#377の内容を変更することができますが、一旦電源が切れると電源投入時ROMの#200~#377の内容が再度転送されるため、変更が無効となります。システムメモリの内容を変更する場合、必ずROMに再書込みする必要があります。

### 3-5 システムメモリの設定とメモリクリア

ラダー設計が完了しましたら、プログラマ（ZW-101PG1）等の周辺装置を使って、次の手順でプログラムを書込んでください。



増設RAMユニット（ZW-501MA）をご使用の場合は、あらかじめ増設RAMユニットを装着しておきます。

システムメモリをクリアするとこれらの設定はすべて初期状態となります。初期状態のままでご使用の場合、これらの設定は不要です。

システムメモリ		初 期 状 態
アドレス (8進数)	クリア後の内容 (8進数)	
#200	300	キーアラー領域 7000~7337
#201		TMR 停止時リセット
#202	000	CNT ONでリセット
#203		停止時 全出力を保持
#204	001	プログラムメモリ容量 3.5K語
#205	000	ファイルレジスタ 使用せず

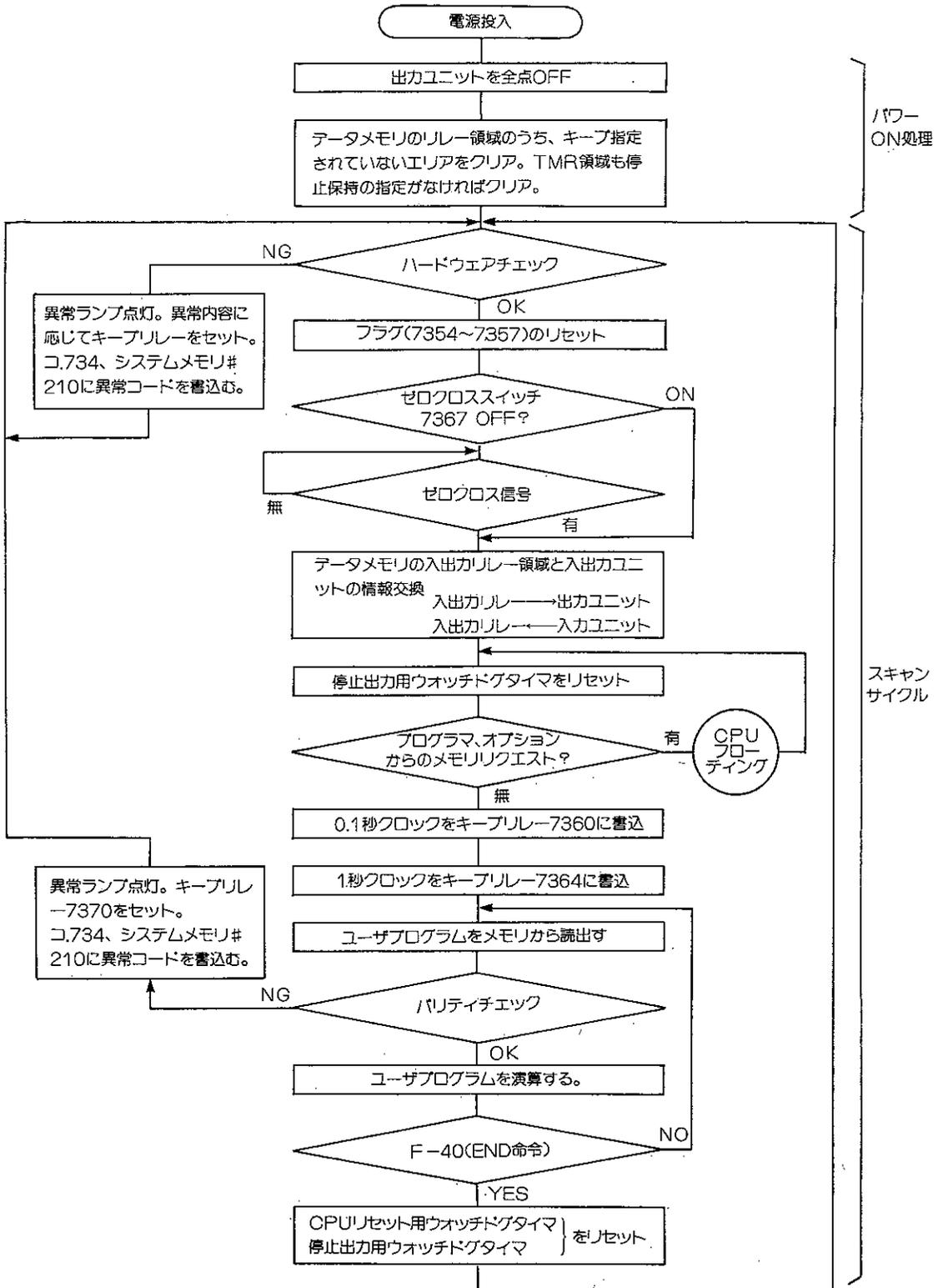
(注)増設RAMユニットをご使用の場合、システムメモリをクリアするとプログラムメモリ容量は5.5K語に設定されます。(＃204は003になります。)

**注1** “メモリのクリア” はシステムメモリの＃204、＃205の設定状態に基づき、クリアする範囲を判断し、次の処理を行います。

- プログラムメモリはすべてNOP命令（何もしない命令）が書込まれ、最終アドレスにはEND命令が書込まれます。
- データメモリおよびファイルレジスタをすべて0にします。したがってシステムメモリの設定後、メモリのクリアを行う必要があります。

### 3-6 運転サイクル

(1) 動作フローチャート



## (2) パワーON処理

電源が投入されると、停電信号（5V電源が完全に立上ってから約10msの間“L”）をチェックし、停電信号が無くなれば、データメモリをイニシャライズします。この

イニシャライズの結果、データメモリは次の様になります。

データメモリ	アドレス	イニシャライズ処理後の状態
入出力リレー	0000~0777	システムメモリ#200にキーブ機能の開始アドレスを指定することができます。 キーブ機能指定以前のアドレス→全てOFF キーブ機能指定以後のアドレス→停電前のON/OFF状態を保持
リンクリレー	2000~2777	
補助リレー	4000~4777	
キーブリレー	7000~7377	
TMR, CNT, MD	000~177	TMR システムメモリ#201に電源投入時の状態を指定することができます。 000—現在値は設定値になります。TMR接点はリセットされます。 001—現在値は停電前の値を保持。TMR接点は停電前のON/OFF状態を保持
		CNT 現在値は停電前の状態を保持。 CNT接点も停電前の状態を保持。
		MD MDデータ、入力情報とも停電前の状態を保持
レジスタ	9000~9377	停電前の値を保持
ファイルレジスタ	30000~37777	停電前の値を保持

**注1** 電源投入時、上記のイニシャライズ処理の前に、各出力ユニット内の出力データ用ラッチがリセットされ、全出力はOFFとなります。

**注2** 電源投入時、データメモリは上記の如くイニシャライズされますが、最初のスキャンサイクルの入出力処理によってデータメモリの入出力リレー領域は次の様に変化します。

- (1) 入力ユニットが装着されている領域  
入力ユニットに接続された入力機器（リミットスイッチ等）のON/OFF状態に従ってON又はOFFとなります。
- (2) 出力ユニットが装着されている部分および入力ユニット未装着の領域  
ユーザプログラムの演算に入るまでイニシャライズ処理の状態から変化しません。

## (3) スキャンサイクル

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

- (1) ハードウェアチェック  
コントロールユニットのハードウェアが正常に機能す

ることを自己診断します。

- a. ROMチェック  
CPU基板内のROM（マイクロプログラムが収納されていて、CPU基板の内部処理を決定するもの）のCRCコードをチェックします。ROMチェックは電源投入時の1サイクルのみ実行され、以後のスキャンサイクルでは実行しません。  
**注1** ROM運転用のROMユニット（ZW-501MO）のチェックではありません。
- b. RAMチェック  
データメモリ用RAMが書込、読み出し可能であるかチェックします。  
**注1** データメモリ用RAMのチェック専用領域を使います。
- c. ハードウェア動作チェック  
ビット処理（AND、OR等の演算）用のアキュムレータ、スタックが正しく動作するかチェックします。
- d. パリティチェック機能の動作チェック  
命令の演算実行時にプログラムメモリのパリティチェックを行います。このチェックはハードウェアで行っています。このハードウェアが正しく機能しているかチェックします。
- e. 入出力データバスのチェック  
ハードウェアチェックの段階では入出力ユニットとデータの交換を行う入出力データバスはフローティング状態になっているのが正常です。もしフローテ

インク状態でなければ入出力データバス異常として処理されます。

**注1** 自己診断としては、上記の5種類以外に次の各項目があります。

- ①入出力信号チェック
- ②ウォッチドグタイマ
- ③パリティチェック
- ④電源異常
- ⑤オプション異常
- ⑥電池異常

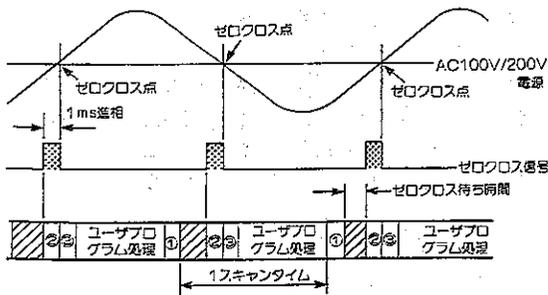
3-7 “自己診断” の項をご参照ください。

**②フラグのクリア**

データ処理命令には、演算の結果、フラグ(Flag)に影響を与えるものがあります。毎スキャンサイクルのユーザプログラム処理の前にフラグをクリアします。フラグに関しては4-5(5)“データ処理命令とフラグ”をご参照ください。

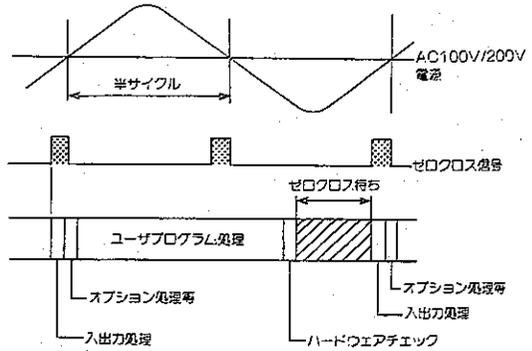
**③ゼロクロス同期**

ゼロクロス同期は入出力処理をAC電源のゼロクロス付近で行うためのものです。入出力処理によりデータメモリの内容が出力ユニットに書込まれると、出力ユニットはデータメモリの内容に従ってON又はOFFとなります。AC100V用またはAC200V用の出力ユニットを使用する場合、ゼロクロス点で入出力処理を行うと、サージの防止や相反動作をするソレノイドの焼損防止に効果があります。ゼロクロス同期処理では、電源のゼロクロス点が来るまで入出力処理に入らず待機します。



- ① ハードウェアチェック
- ② 入出力処理
- ③ オプション処理等

**注1** 入出力処理、オプション処理、ユーザプログラム処理、ハードウェアチェックの合計処理時間が電源の半サイクルに納まらない場合は次のようになります。

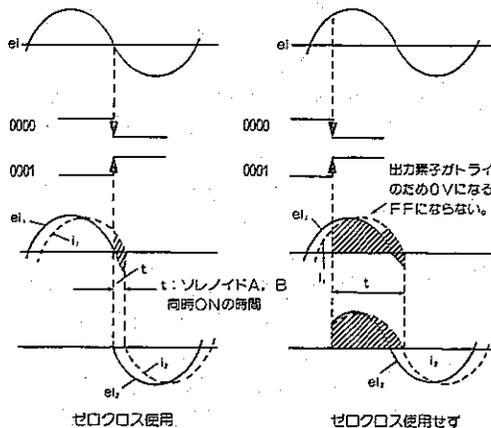
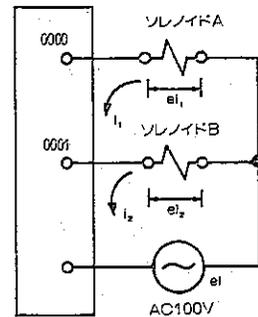


**注2** 出力ユニットとしてDC出力ユニットを使用する場合等ゼロクロス同期の必要が無いときは、データメモリのゼロクロススイッチ(7367)をプログラムでONにすることでゼロクロス待ち時間を0にできます。



7366は常時OFFの接点であり、7367は常時ONとなります。

**参考** ソレノイドAのON→OFF、ソレノイドBのOFF→ONが同時に起きる時、ゼロクロスを用いる場合とそうでない場合を例に説明します。下図の通りに同時オンの時間が短くなり、ソレノイドの焼損を防ぎます。



#### (4)入出力処理

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

##### a. 入出力信号異常チェック

選択された入出力ユニットからは、入力ユニットであるか、出力ユニットであるかをCPUに知らせる「入出力信号」を発生します。CPUはこの信号に基づき、入力処理（入力ユニットの情報をデータメモリへ）あるいは出力処理（データメモリの内容を出力ユニットへ）を実行します。

入出力ユニットの選択前に、この「入出力信号」を発生しているユニットがあれば「入出力信号異常」として処理します。

このチェックは1バイト(8点)ごとに行われ、16点ユニットの場合は前半8点の選択前と、後半8点の選択前の2回実行されます。

##### b. 入出力ユニット処理

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

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

**注1** 電源投入後の1サイクル目では、「パワーON処理」でイニシャライズされたデータメモリの内容が、出力ユニットに書込まれ、以後のサイクルでは、1回前のサイクルの演算結果が出力ユニットに書込まれます。

**注2** 選択されたユニットから「入出力信号」が返送されない場合、CPUはユニット未装着と判断し、入出力処理を終り次の処理（ウォッチドグタイマをリセット）に移ります。従って入出力処理に要する時間は入出力ユニットの装着数により変化します。

**注3** 入力ユニット装着領域で、入力機器が接続されていない部分は、入出力処理でOFFとしてデータメモリに読込まれます。従って補助リレーには使えません。

**注4** 出力ユニット装着領域で、出力機器が接続されていない部分は、補助リレーとして使えます。（ただし、入出力処理で出力ユニットにはデータメモリの内容が書込まれ、出力ユニットのLEDは点灯します。）

**注5** 入出力リレー領域のユニット未装着部分および出力ユニット装着領域で出力機器の未接続部分を補助リレーとして使用した場合、将来入出力機器の追加でこの領域を使用すると、プログラムの大巾変更（他の領域に補助リレーを移す）となる場合がありますのでご注意ください。

#### (5)ウォッチドグタイマ

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

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

何らかの原因でスキャンが異常となるとCPUからのリセットが掛らずウォッチドグタイマがタイムアップし、「停止出力」がOFFとなります。またパネル面の「運転中」のLEDは消灯、「異常」のLEDが点灯します。（3-7「自己診断」参照）

#### (6)プログラマ、オプションからのリクエストに対する処理

プログラマからのモニタ/設定値変更、コンピュータリンク、テータリンク、リモートI/Oとのデータの交換を行います。コントロールユニットに対してプログラマやオプションモジュールからメモリリクエスト（コントロールユニット内のデータメモリ、ユーザプログラムメモリに対して書込み、読出しを要求する信号）があれば、コントロールユニット内のCPUはHOLD状態となり、バスラインを開放します。

この間ウォッチドグタイマがタイムアップすることの無いようウォッチドグタイマをリセットします。

#### (7)0.1秒クロック(7360)、1秒クロック(7364)のセット/リセット

ハードウェアで作成した0.1秒クロックのON/OFF状態をキーブリレー7360に、0.1秒クロックをソフト的に分周した1秒クロックのON/OFF状態をキーブリレー7364に書込みます。

#### (8)ユーザプログラム処理

ユーザプログラムメモリ先の先頭からプログラムを順次読み出し、パリティチェックがOKの場合、プログラム内容に従い演算を実行します。

##### a. ユーザプログラムのパリティチェック

ユーザプログラムは1語当り2バイトで構成されています。プログラマ等でプログラムを書込むとき、1語ごとにパリティが生成されユーザプログラムに付加されます。ユーザプログラム処理ではユーザプログラムメモリを1語読出すごとにパリティのチェックを行い、異常の場合、パリティエラーの処理に入ります。ユーザプログラムメモリが何らかの原因で変化した場合、このパリティチェックで検定されるため、演算のステップには進みません。

##### b. ユーザプログラムの演算

パリティチェックがOKの場合、ユーザプログラムの内容に従い演算します。

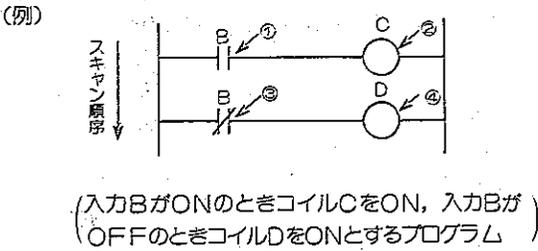
STR、STR NOT、AND、AND NOT、OR、OR NOT、AND STR、OR STRの各命令では演算結果をアキュムレータ、スタックレジスタに格納します。

OUT、TMR、CNT、MD、及び殆どどの応用命令（F-××）では演算結果をデータメモリに書込みます。

注1 各命令の詳細は「§4命令語の説明」をご参照ください。

注2 ユーザプログラムの演算に先だち、「入出力処理」において入力ユニットのON/OFF状態を一括してデータメモリに読み込み、各命令の演算時にはデータメモリの内容を参照する方式を採用しているため「入力のレーシング」といった異常現象は発生しません。

参考 入力のレーシング現象  
命令の演算時にその都度入力ユニットのON/OFF状態を読み込む方式のPCの場合、次のような現象が起ります。



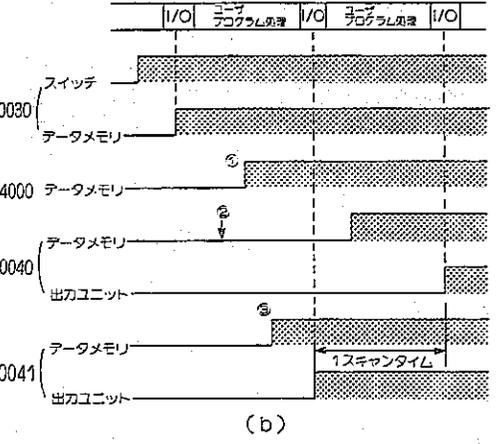
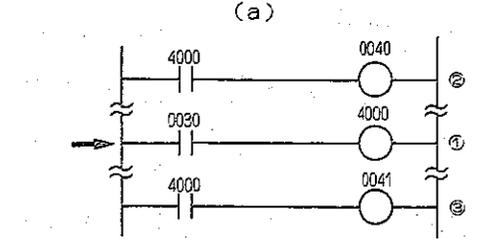
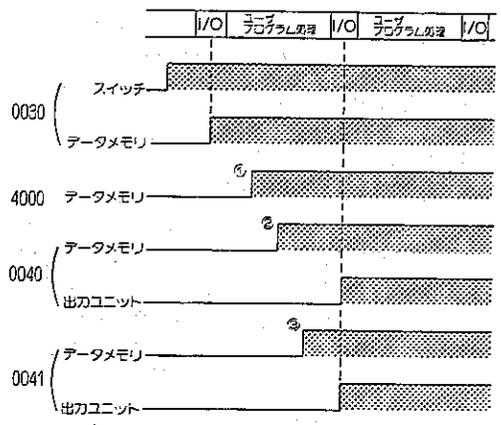
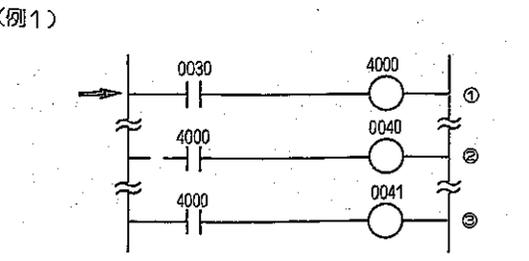
上図のプログラムではC= $\bar{D}$ となるはずですが、①で入力Bの状態を入力ユニットからアキュムレータに入れたときBはONであったとします(CはON)。ところが③の演算までの間に入力Bの状態がOFFに変化すると、③の演算ではBはOFFとして扱われ、コイルDがONし、C、DともにONという論理的に矛盾した結果になります。

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

「入出力処理」を一括して行うPCではこのような現象は起りません。

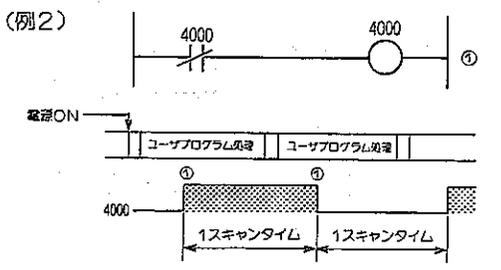
注3 OUT命令のようにデータメモリに演算結果を書込む命令では演算の都度、データメモリに演算結果を書込みます。ただし出力ユニットの状態は次のスキャンサイクルの入出力処理までは変化しません。

注4 OUT命令のようにデータメモリに演算結果を書込む命令の後に、当該データメモリを接点として使用する命令があると、OUT命令で書き換えられた内容に基づき演算されます。



(a)と(b)に示すようにプログラムの書き込み順を入れ替えると、演算結果が異なったものとなります。並列に処理されるリレー盤では、(a)も(b)も差はありませんが、直列処理形のプログラマブルコントローラ（現在市販されているプログラマブルコントローラは殆んど直列処理形です）では上記のような現象が起ります。したがってコイルの補助接点を使うとき(例1では4000)、次の事項に注意してプログラムを作成する必要があります。

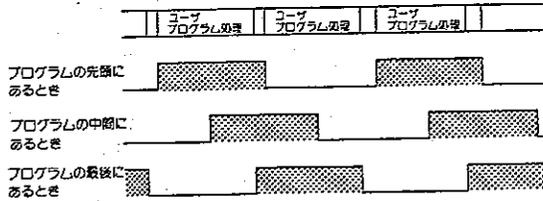
「コイルの前に言われた補助接点の状態変化は、コイルの状態が変わった次のスキャンに生ずる。」



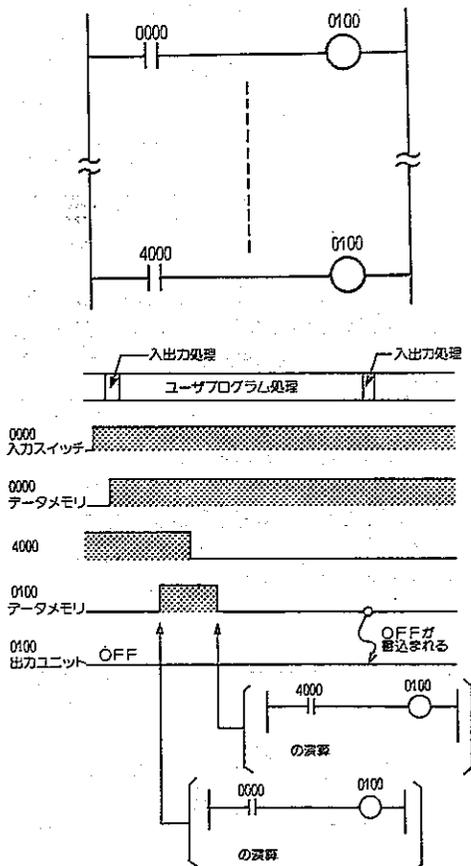
例2は「コイルの前に蓄かれた補助接点の状態変化は、コイルの状態が変化した次のスキャンに生ずる」ことを逆に応用したもので、1スキャンサイクルごとにON/OFFを繰り返します。(発振回路)

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

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



**注5** プログラム上、同一リレー番号をコイルとして複数回使用すると、プログラムENDでは、最後にコイルとして使用したプログラムの演算結果がデータメモリに残っていて、これが次のスキャンサイクルの入出力処理で出力ユニットに書き込まれるため、目的と違った動作となることがあります。



W16/W51用プログラマ(ZW-101PG1)では、プログラムチェックを行うと、このようなコイルの二重使用があるとき“DOUBLE OUT”として表示されます。

### ⑨スキャンタイム

ハードウェアチェックからEND命令(F-40)の演算までの1スキャンに要する時間をスキャンタイムと呼び、次のようにして概略計算することができます。ただしプログラマ、オプションからのリクエストに対する処理に要する時間はリクエストの有無、処理内容により異なるため、一般にスキャンタイムに含めません。またハードウェアチェック、0.1秒・1秒クロックのキーブリレーへの蓄込、フラグのリセット等に要する時間は他の処理時間と比べ短いため無視して計算します。

$$1 \text{ スキャンタイム}(T) = \text{入出力処理時間}(t_1) + \text{ユーザプログラム処理時間}(t_2)$$

#### ①入出力処理時間( $t_1$ )

CPUが入力ユニットの入力情報を読み込み、出力ユニットへ出力情報を書き込むのに必要とする時間です。本機の場合、ベースユニットへの入力ユニット、出力ユニットの実装数により、入出力処理時間が決定されます。

$$t_1 = \text{入力ユニットの実装数} \times t(\text{in}) + \text{出力ユニットの実装数} \times t(\text{out})$$

	16点ユニットのとき	32点ユニットのとき
$t(\text{in})$	34.72 ( $\mu\text{s}$ )	69.44 ( $\mu\text{s}$ )
$t(\text{out})$	39.06 ( $\mu\text{s}$ )	78.12 ( $\mu\text{s}$ )

(例)

16点入力ユニット	4ユニット
32点入力ユニット	1ユニット
16点出力ユニット	9ユニット
32点出力ユニット	2ユニット

$$t_1 = 34.72 \times 4 + 69.44 \times 1 + 39.06 \times 9 + 78.12 \times 2 = 716.1(\mu\text{s})$$

#### ②ユーザプログラム処理時間( $t_2$ )

プログラムアドレス00000からEND命令までの全命令の処理時間の合計となります。各命令の処理時間は4-1“命令語一覧表”をご参照ください。

**注1** 応用命令の処理時間は実行時と非実行時で異なります。

#### ③END命令

プログラムメモリをクリアすると、プログラムメモリにすべてNOP命令を書き込み、最終アドレスにはF-40 (END命令) が書き込まれます。この状態でプログラムメモリの途中まで命令を書き込んだ場合、NOP命令の処理時間がスキャンタイムに加算されます。

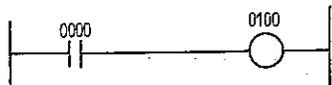
最後のプログラムを書き込んだアドレスの次にF-4

0を書き込みますと、そのアドレスでユーザープログラムの処理が終了、スキャンタイムを短くすることができます。

●NOP命令の処理時間 1.21μs

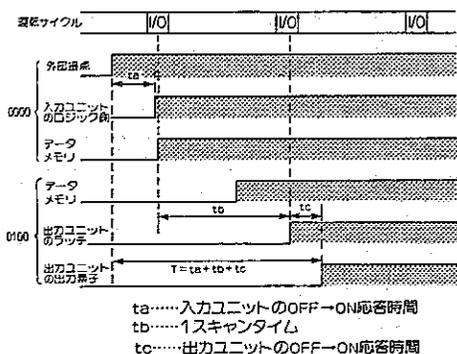
4-6 “応用命令の説明”のF-40をご参照ください。

注2 入出力ユニットの応答時間を含めたPC全体の応答時間は次のようになります。

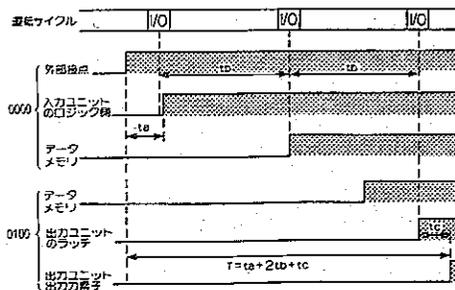


上記のプログラムで、外部接点0000が変化してから、出力ユニットの出力素子（トランジスタ、トライアック、リレー）が変化するまでの時間を示します。

(a)最も短時間の場合

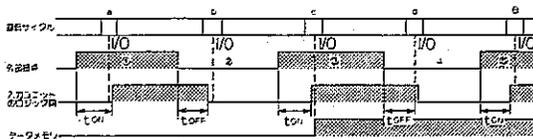


(b)最も長時間の場合



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

注3 外部接点のON/OFF状態を確実にデータメモリに取込むためには、入力ユニットのロジック側のONまたはOFFの時間として、1スキャンタイム以上が必要です。



tON……入力ユニットのOFF→ON応答時間  
tOFF……入力ユニットのON→OFF応答時間

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

②の外部接点のONでは、cの入出力処理の直前に入力ユニットのロジック側もONになっているため、データメモリにはONが書込まれます。

④の外部接点のOFFは、dの入出力処理の時、入力ユニットのロジック側は未だONのため、データメモリはONのままとなります。eの入出力処理では、入力ユニットが再びONのため、データメモリはONを維持します。

このように入力ユニットのロジック側のON/OFFの時間が1スキャンタイムより短いと、データメモリに取込まれたり、取込まれなかったりします。

入力ユニットのロジック側のON/OFF時間、出力ユニットの応答時間については“ニューサテライトW51取扱説明書”の“入力ユニットご使用時の留意事項”をご参照ください。

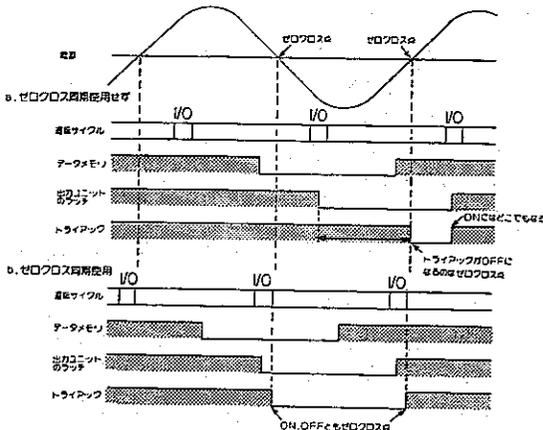
④ゼロクロス同期を使う場合

1スキャンタイムは電源周波数により決定されます。

$T = t_1 + t_2$	50Hz	60Hz
電源の半サイクル以下のとき	10ms	約 8.3ms
電源の1サイクル以下のとき	20ms	約16.7ms

3-6(3) “スキャンサイクル”の③ゼロクロス同期をご参照ください。

注4 出力ユニットとしてトライアック出力のもの（ZW-16S1、ZW-32NS1等）を使用すると、出力のトライアックがON→OFFとなるのは、電源のゼロクロス点となります。



### 3-7 自己診断

本機で行っている自己診断は次表のとおりです。

項目	内容	PCの運転状態	停止出力	パネルの表示灯			特殊リレー	異常コード											
				運転中	異常	電源		特殊レジスタコ.734	システムメモリ	優先順位									
メモリ異常	パリティチェック	停止	開	消灯	点灯	点灯	7370	20	21	6									
CPU異常	ウォッチドグタイマ						30	7371	7371 7376	7371	7371 7376	40	44	31	1				
	RAMチェック (R/W)													32	4				
	パリティチェック													33	4				
	ROMチェック													34	2				
	ハードウェアチェック													35	4				
入出力異常	入出力データバス						7373	7373	7373	7373	7373	40	44	45	5				
	入出力信号													45	5				
電源異常	停電 電源電圧低下						7377	7377	7377	7377	7377	10	13	13	3				
オプション異常	オプションモジュールの異常															7374	7374	7374	7374
		7372	20	22	8														
電池異常	電池電圧低下	運転	閉	点灯			7372	20	22	8									

注1 異常コードはすべてBCDコードです。

#### (1) 自己診断内容

1. メモリのパリティチェック  
すべての命令の演算実行時プログラムメモリのパリティをチェックします。
2. ウォッチドグタイマ  
入出力処理実行後、ウォッチドグタイマをリセット。運転サイクルが異常になるとタイムアップします。
3. RAMチェック  
毎運転サイクルごとにデータメモリ用RAMが書込、読み出し可能であるかチェックします。
4. CPUによるパリティチェック  
メモリのパリティチェックはハードウェアで実行します。  
毎運転サイクルごとにハードウェアのパリティチェック機能が正常かどうかCPUがチェックします。
5. ROMチェック  
電源投入時にマイクロROMのCRCコードをチェックします。
6. ハードウェア動作  
毎運転サイクルごとにアキュムレータ、スタックが正しく設定されることをチェックします。
7. 入出力データバス  
入出力処理の前に入出力データバスがフローティング状態であることを確認します。
8. 入出力信号  
入出力処理サイクルに入ると入出力ユニットが順次選択されますが、選択された入出力ユニットはCPUに対し入力であるか出力であるかを知らせる信号を発生します。  
入出力処理の前にこの信号が発生している入出力ユニットが無いことをチェックします。
9. 電源  
●本機は約10ms以下の瞬時停電の場合、これにตอบสนองせず運転を続行します。  
これ以上の停電の場合、CPUが停止し停止出力が開放となります。  
停電が復旧すると自動的に運転を再開します。

- 電源電圧が徐々に低下（スローダウン）してきた場合、定格電圧の80%以下になるとCPUが停止し、停止出力が開放となります。  
この場合も電源電圧が復旧すれば自動的に運転を再開します。

#### 10. オプション異常

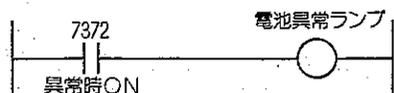
コンピュータリンク、テータリンク、リモート I/O を使用時、これらのオプションモジュールに異常が無いがチェックします。

#### 11. 電池

メモリバックアップ用電池の電圧が正常であるかチェックします。

特殊リレー7372を使って、電池異常時ランプを点灯させたり、ブザーを鳴らすことができます。

PCに電源が投入されている限り、電池異常状態でもPCの運転には影響はありませんが、万一の停電にそなえ、できるだけ速やかに電池を交換する必要があります。



## (2) 停止出力

- 自己診断により異常と判断されたとき、「開」となる出力で正常運転中は「閉」となります。
- システムの非常停止回路に本機の停止出力を接続することにより、PC異常時システムを非常停止させることができます。
- トライアック出力 AC100V又はAC200V、1A

## (3) 特殊リレー

データメモリの特殊リレー領域に自己診断結果を書き込みます。

自己診断の結果異常が検知されPCが停止した場合、周辺装置により特殊リレー（7370～7377）を検索し異常内容を知ることができます。

**[注1]** 自己診断は毎演算サイクルごとに行われますので、異常が回復すればPCは運転を再開し停止出力も閉となります。  
また自己診断用特殊リレーもリセットされます。

## (4) 異常コード

### 1. 特殊レジスタ

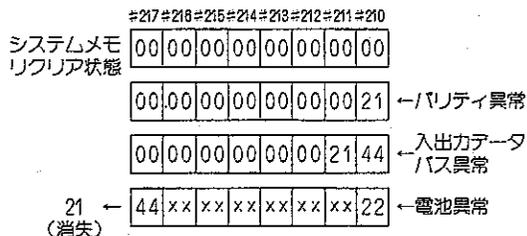
自己診断の結果、異常と判断された場合、データメモリの特殊レジスタ（バイトアドレスコ.734）に異常コードを書き込みます。

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

### 2. システムメモリ

自己診断の結果、異常と判断された場合、システムメモリ（#210～#217）にも異常コードが書き込まれます。

#210～#217はシフトレジスタとして働き、8回の異常発生を記憶することができます。異常が8回以上になると、最初に書き込まれた異常コードから順に消失して行きます。



- 特殊レジスタには代表コードを書き込みますが、システムメモリには異常内容をさらに分類した個別コードを書き込みます。
- システムメモリの異常コードは異常回復後もクリアされません。クリアするときは、プログラマ等の周辺装置でシステムメモリ（#210～#217）に“00”を書き込んでください。
- 同じ異常が連続して発生した場合、異常コードは書き込まれません。

## (5) 異常時の出力ユニットの

### ON/OFF状態

システムメモリ（#203）の出力保持アドレスの設定内容により、自己診断の結果PCが運転を停止する場合の出力ユニットのON/OFF状態が決まります。

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

ただし、異常内容によっては出力保持アドレス以前の出力ユニットをOFFにできない場合があります。PC異常時にOFFにする必要がある出力は、コントロールユニット（ZW-160CU/ZW-501CU2）の停止出力を直列に接続してください。2-2 “システム設計に際しての留意事項”、3-3 “システムメモリ”の項をご参照ください。



# §4 命令語の説明

## 4-1 命令語一覧表

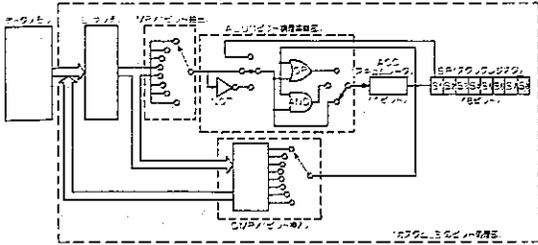
命令語	シンボル	語数	機能	演算内容	実行条件	スタック	フラグ	処理時間(μs)		ページ
								W16	W51	
STR		1	a 接点で論理を閉結。中間結果の記憶	DM → ACC → S				1.63	40	
STR NOT		1	b 接点で論理を閉結。中間結果の記憶	DM → ACC → S				1.63	40	
AND		1	論理積	DM → AND → ACC → S				1.63	41	
AND NOT		1	論理積否定	DM → AND → ACC → S				1.63	41	
OR		1	論理和	DM → OR → ACC → S				1.63	41	
OR NOT		1	論理和否定	DM → OR → ACC → S				1.63	42	
AND STR		1	中間結果との論理積	DM → AND → ACC → S				1.63	42	
OR STR		1	中間結果との論理和	DM → OR → ACC → S				1.63	42	
OUT		1	演算結果の出力	DM → ACC				1.63	40	
TMR		2	タイマ ①スタート入力(ONでON) ②TMR設定(000~177) ③ON時間(0~99.99) ④内部クロック 0.1μs	スタート入力のON時間、0.1μsごとに設定値を-1。現在値=0でTMR動作。	ON			193 (168)	191 (81)	43
CNT		2	カウンタ ①リセット入力 ②CNT設定(000~177) ③ON時間(0~99.99)	リセット入力のOFFの間、計数入力の上昇で現在値を-1。現在値=0でCNT動作。	ON			194 (168)	123 (95)	45
MD		2	メンデナンス ディスプレイ ①出力指示端子 ②出力指示端子 ③電源出力 ④MD設定(000~177) ⑤MDデータ(000~999)	出力指示端子ONの時、①、②の出力指示と、④のMDデータと、⑤で指定のMDデータのデータメモリに書き込まれる。	ON			79 (82)	62 (28)	46
F-00		3	データレジスタ間の1バイト転送	S → D				112 (81)	82 (46)	73
F-01		3	BCD定数の転送	n → D n = 00~99				112 (81)	76 (46)	74
F-02		3	レジスタ間のデータ交換	D1 ↔ D2				119 (89)	90 (46)	75
F-03		3	BCD → BIN変換	S → D				151 (88)	103 (46)	76
F-04		3	BIN → BCD変換	S → D				128 (86)	100 (46)	77
F-05		3	分配	S + 1 → D + (S)				104 (81)	76 (46)	78
F-06		3	抽出	S + (D) → D + 1				107 (81)	81 (46)	80
F-07		3	10進定数の転送	n → D n = 000~255				112 (91)	75 (46)	81
F-08		3	8進定数の転送	n → D n = 000~377(o)				112 (81)	75 (46)	82
F-09		3	反転データの転送	S → D				113 (91)	84 (46)	83





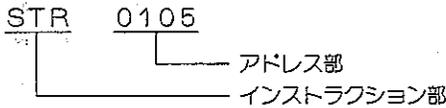
## 4-2 ビット処理部の動作

ビット処理とは、接点信号の論理演算のことで、W16/W51ではカスタムLSIがこの処理を行っています。TMR、CNT、MD命令および応用命令はCPUが処理しますが、これら命令の演算条件であるビット演算も、このカスタムLSIが行い、以後のデータ処理はCPUが実行します。カスタムLSIのビット処理部の概略ブロック図を下図に示します。



### (1) L(ラッチ)

ビット処理命令は、インストラクション部とアドレス部で構成されます。

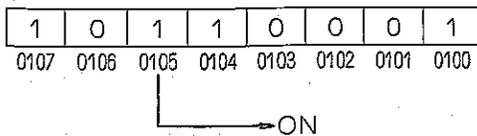


アドレス部はデータメモリのリレー領域(入出力リレー、リンクリレー、補助リレー、キーブリレー)のリレー番号を表わします。データメモリからリレーのON/OFF情報を読み出す場合、そのリレー番号が含まれる1バイト(8ビット)の内容をまとめてL(ラッチ)に読み出します。

STR 0105の場合、0100~0107の8ビットが読み出されることになります。

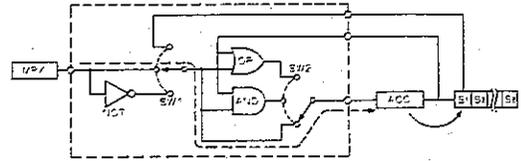
### (2) MPX(マルチプレクサ)

L(ラッチ)に読み出された8ビットのうち、必要な1ビットを抽出します。STR 0105の場合0100~0107から0105のON/OFF情報が抽出されます。



### (3) ALU(1ビット論理演算部)

命令のインストラクション部の内容に従い論理演算を行います。



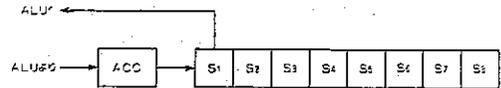
上図はSTR命令の場合の演算状態を示します。インストラクション部の内容により、SW1、SW2が切換えられます。

### (4) ACC(アキュムレータ)

ALUの演算結果を格納する1ビットのレジスタです。

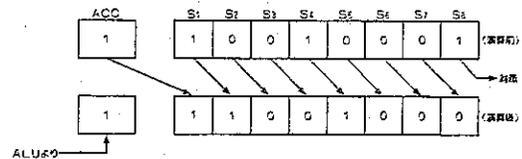
### (5) SR(スタックレジスタ)

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



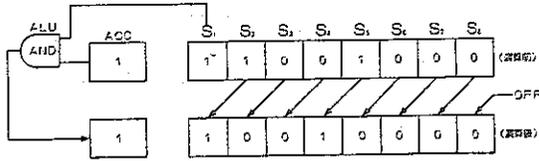
### ① STR、STR NOT命令実行時のSRの動き

- STR、STR NOT命令では、データメモリから読み出された1ビットのON/OFF情報がACCに入ります。(STR NOTでは反転後ACCに入ります)
- それ以前にACCに入っていたON/OFF情報はS1に、S1の情報はS2に、以後S2→S3、S3→S4、S4→S5、S5→S6、S6→S7、S7→S8とシフトされ、S8に入っていた情報は消滅します。

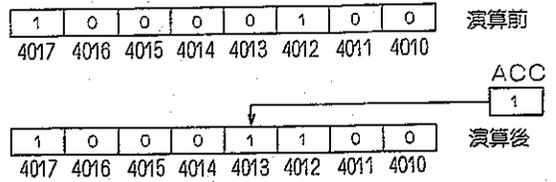


### ② AND STR、OR STR命令実行時のSRの動き

- AND STR、OR STR命令では、S1のON/OFF情報がALUに入り、ACCの内容との間でAND又はORの演算が行われ、演算結果はACCに格納されます。
- 演算後不要となったS1のON/OFF情報は消滅し、S1にはS2の情報が、S2にはS3が、以後S3→S4、S4→S5、S5→S6、S6→S7、S7→S8とシフトされ、S8にはOFFの情報が入ります。



(OUT 4013で、演算結果がONの場合)



(6) DMPX(デマルチプレクサ)  
OUT命令では、L(ラッチ)に読出された8ビットのうち、命令のアドレス部で示される1ビットを、演算結果(ACCの内容)に書き換え、データメモリに1バイト分を転送します。

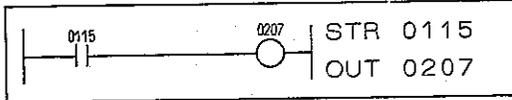
注1 各ビット処理命令におけるビット処理部の動作は4-3“基本命令の説明”をご参照ください。

### 4-3 基本命令の説明

#### (1) STR/OUT

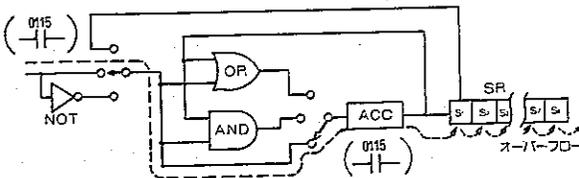
**STR** 指定されたデータメモリの内容(ON/OFF状態)をアキュムレータ(ACC)に格納します。また、以前にあったACCの内容をスタックレジスタ(SR)のS1にシフトします。

**OUT** アキュムレータ(ACC)の内容を指定されたデータメモリへ送ります。



#### STR 0115

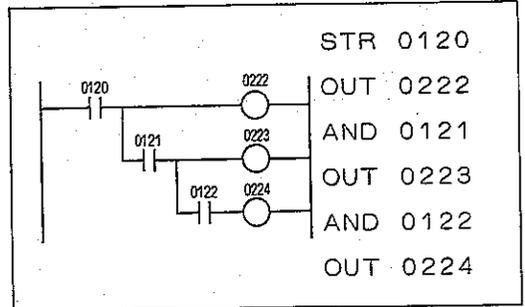
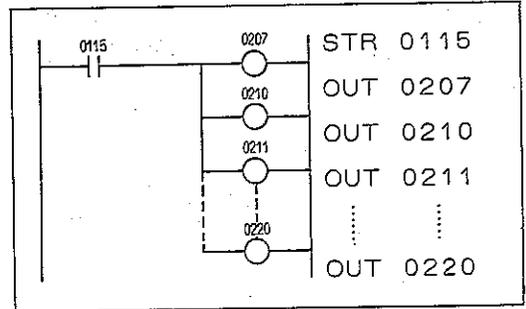
- L(ラッチ)……データメモリから(0110)~(0117)の8ビットが読出されます。
- MPX……L(ラッチ)内の8ビットから(0115)の1ビットを抽出します。
- ALU, ACC, SR……MPXの出力をそのままACCに書込みます。  
また、以前のACCの内容はSRのS1にシフトします。



#### OUT 0207

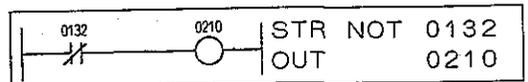
- L(ラッチ)……データメモリから(0200)~(0207)の8ビットが読出されます。
- MPX……OUT命令では関与しません。
- ALU, ACC, SR……ACC、SRの内容は不変です。
- DMPX……L(ラッチ)内の8ビットの内(0207)の1ビットをACCの内容に書換え(0200)~(0207)の8ビットをデータメモリに送ります。

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



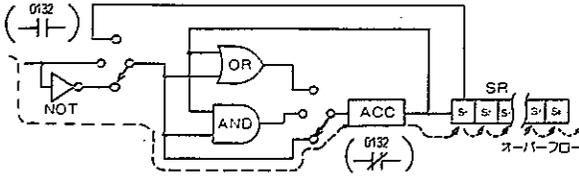
#### (2) STR NOT

- 指定されたデータメモリの内容を反転してACCに格納します。  
また、以前にあったACCの内容をSRのS1にシフトします。



### STR NOT 0132

- L(ラッチ)…データメモリから(0130)~(0137)の8ビットが読出されます。
- MPX…L(ラッチ)内の8ビットから(0132)の1ビットを抽出します。
- ALU、ACC、SR…MPXの出力を反転してACCに書込みます。  
また、以前のACCの内容はSRのS1にシフトします。

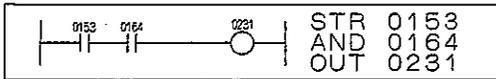


### OUT 0210

データメモリの(0210)は  $\overline{0132}$  の演算結果に書換えられます。

### (3) AND

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

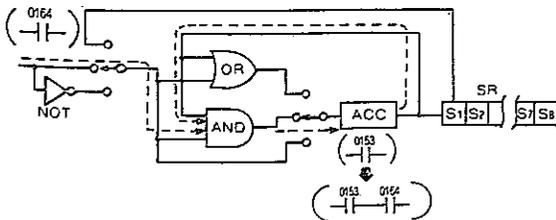


### STR 0153

ACCにデータメモリの(0153)の内容が記憶されます。

### AND 0164

- L(ラッチ)…データメモリから(0160)~(0167)の8ビットが読出されます。
- MPX…L(ラッチ)内の8ビットから(0164)の1ビットを抽出します。
- ALU、ACC、SR…ACCの内容(0153)とMPXの出力(0164)のANDを演算し、ACCに書込みます。SRの内容は保持されます。

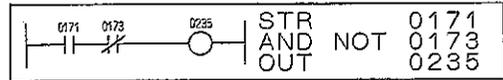


### OUT 0231

データメモリの(0231)は  $\overline{0153}$  の演算結果に書換えられます。

### (4) AND NOT

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

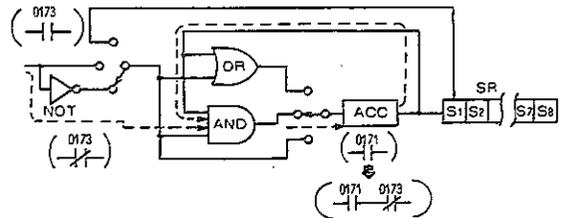


### STR 0171

ACCにデータメモリの(0171)の内容が記憶されます。

### AND NOT 0173

- L(ラッチ)…データメモリから(0170)~(0177)の8ビットが読出されます。
- MPX…L(ラッチ)内の8ビットから(0173)の1ビットを抽出します。
- ALU、ACC、SR…ACCの内容(0171)と、MPXの出力(0173)の反転したもののANDを演算しACCに書込みます。SRの内容は保持されます。

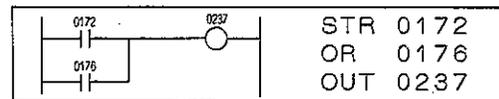


### OUT 0235

データメモリの(0235)は  $\overline{0171}$  の演算結果に書換えられます。

### (5) OR

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

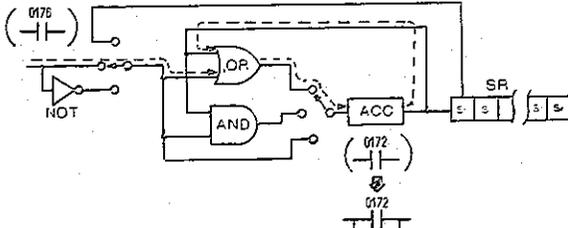


### STR 0172

ACCにデータメモリの(0172)の内容が記憶されます。

### OR 0176

- L(ラッチ)…データメモリから(0170)~(0177)の8ビットが読出されます。
- MPX…L(ラッチ)内の8ビットから(0176)の1ビットを抽出します。
- ALU、ACC、SR…ACCの内容(0172)とMPXの出力(0176)のORを演算し、ACCに書込みます。SRの内容は保持されます。



### OUT 0237

データメモリの(0237)は  $\begin{matrix} 0172 \\ | \\ 0176 \end{matrix}$  の演算結果に書換えられます。

## (6) OR NOT

●指定されたデータメモリの内容を反転し、ACCの内容とOR演算して、その結果をACCに格納します。

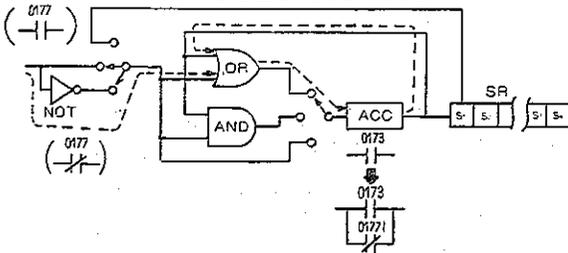


### STR 0173

ACCにデータメモリの(0173)の内容が記憶されます。

### OR NOT 0177

- L(ラッチ)…データメモリから(0170)~(0177)の8ビットが読出されます。
- MPX…L(ラッチ)内の8ビットから(0177)の1ビットを抽出します。
- ALU、ACC、SR…ACCの内容(0173)とMPXの出力(0177)を反転したもののORを演算しACCに書込みます。SRの内容は保持されます。

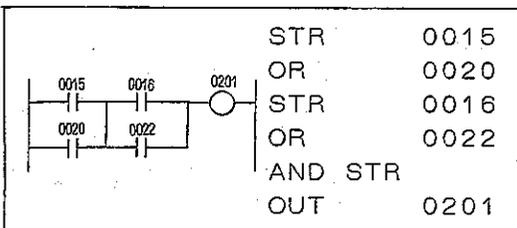


### OUT 0240

データメモリの(0240)は  $\begin{matrix} 0173 \\ | \\ 0177 \end{matrix}$  の演算結果に書換えられます。

## (7) AND STR

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



### STR 0015

ACCにデータメモリの(0015)の内容が記憶されません。

### OR 0020

ACCには  $\begin{matrix} 0015 \\ | \\ 0020 \end{matrix}$  の演算結果が記憶されます。

### STR 0016

ACCに入っているそれ迄の演算結果  $\begin{matrix} 0020 \\ | \\ 0015 \end{matrix}$  をSRのS1に待避させ、データメモリ(0016)の内容をACCに書込みます。

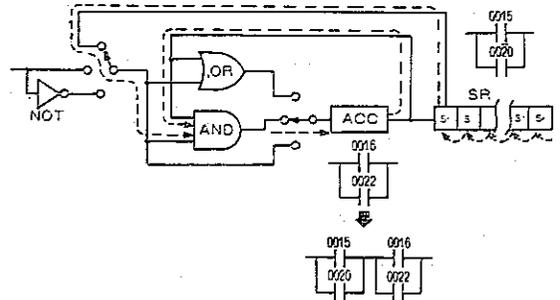
### OR 0022

ACCには  $\begin{matrix} 0015 \\ | \\ 0022 \end{matrix}$  の演算結果が記憶されます。

## AND STR

- L(ラッチ)…AND STR命令の場合関与しません。
- MPX…AND STR命令の場合関与しません。

- ALU、ACC、SR…SRのS1の内容  $\begin{matrix} 0015 \\ | \\ 0016 \end{matrix}$  とACCの内容  $\begin{matrix} 0015 \\ | \\ 0022 \end{matrix}$  のANDを演算し、ACCに書込みます。

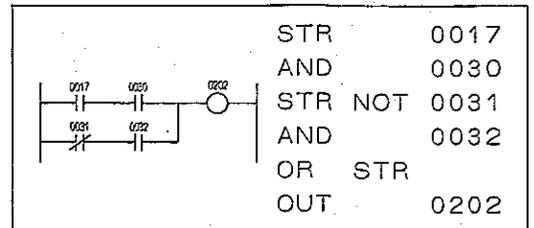


### OUT 0201

データメモリの(0201)は  $\begin{matrix} 0015 \\ | \\ 0022 \end{matrix}$  の演算結果に書換えられます。

## (8) OR STR

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



### STR 0017

ACCにデータメモリの(0017)の内容が記憶されません。

### AND 0030

ACCには  $\overline{0017} \cdot \overline{0030}$  の演算結果が記憶されます。

### STR NOT 0031

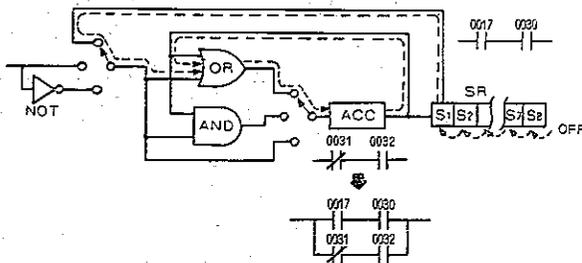
ACCに入っているそれ迄の演算結果  $\overline{0017} \cdot \overline{0030}$  をSRのS1に待避させ、データメモリ (0031) の内容を反転してACCに書込みます。

### AND 0032

ACCには  $\overline{0031} \cdot \overline{0032}$  の演算結果が記憶されます。

### OR STR

- L(ラッチ)……OR STR命令の場合  
関与しません。
- MPX……OR STR命令の場合  
関与しません。
- ALU、ACC、SR……SRのS1の内容  $\overline{0017} \cdot \overline{0030}$  とACCの内容  $\overline{0031} \cdot \overline{0032}$  のORを演算し、ACCに書込みます。



### OUT 0202

データメモリの(0202)は  $\overline{0031} \cdot \overline{0032}$  の演算結果に書換えられます。

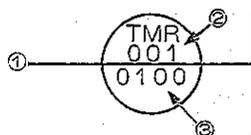
### (9) TMR(タイマ命令)

TMR命令は0.1秒クロックを内部クロックとする減算式のタイマーです。スタート入力OFFの間、計数は行われず、現在値=設定値を維持し、TMR接点はOFFになっています。スタート入力ONになると0.1秒ごとに現在値は-1され、現在値が0になるとTMR接点がONし、スタート入力ONの間この状態を保持します。

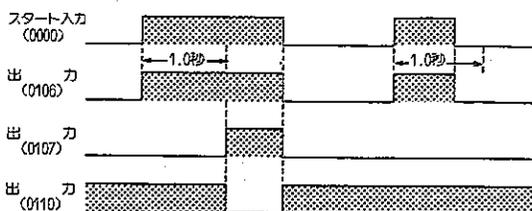
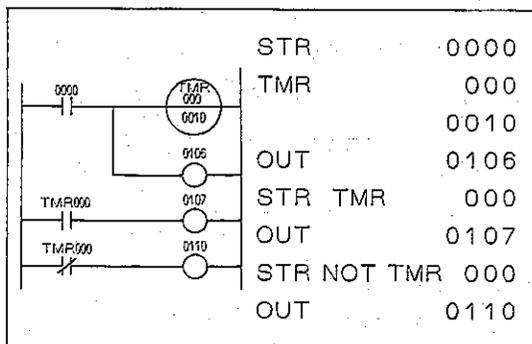
スタート入力	現在値	TMR接点
OFF	設定値	OFF
ON(現在値>0)	0.1秒ごとに-1される	OFF
ON(現在値=0)	0	ON
ON→OFF(現在値>0)	設定値にもどる	OFF
ON→OFF(現在値=0)	設定値にもどる	ON→OFF

TMR命令のシンボル

TMR接点のシンボル



- ①スタート入力 (ONでスタート)
- ②TMR番号 000~177(8進)……CNT、MDと共通使用
- ③設定値 0001~1999(BOD)……0.1秒単位 (000.1~199.9秒)  
精度 (設定値 $\pm$ 0.1秒) + スキャンタイム



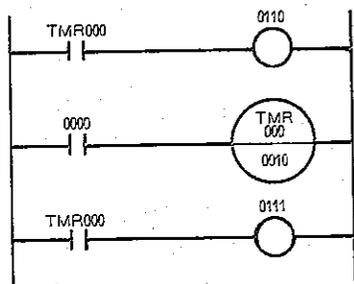
注1 TMR番号は、CNT、MDに共通使用していますので、CNT、MDに使用した番号は、TMRに使用しないでください。又、同一TMR番号の使用も避けてください。万一、同一番号を使用した場合、プログラマ(ZW-101PG1)等のプログラムチェックによって、エラー表示します。

注2 TMR接点はTMR番号と同じ番号を指定し、a接点、b接点を何個でも使用できます。

注3 TMRの現在値は、b.000~b.377の256バイトに格納されます。3-2〔4〕“TMR、CNT、MDのデータ格納領域”をご参照ください。

注4 PCの電源投入時、タイマはリセットされます。従って、タイマのスタート入力ON状態で、PCの電源が入っても、リセット機能が働き、現在値は設定値となります。

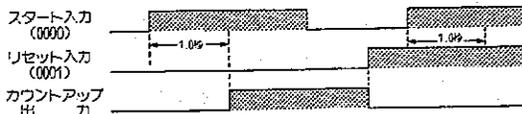
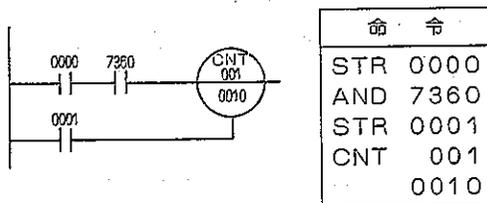
注5 PCの電源投入時、タイマはリセットされるにもかかわらず、TMR命令より前にあるTMR接点が1スキャンタイムONすることがあります。



上記のプログラムにおいて、電源断時、接点0000がONしており、TMR000がタイムアップ(ON)状態のとき、電源再投入時、出力0110が1スキャンタイムの間ONします。この現象を回避するためには、TMR命令の後にTMR接点をプログラムするようにしてください。(出力0111はONしません。)

注6 タイマ命令はシステムメモリ(≒201)にタイマリセット条件を設定することにより停電時の状態を記憶することもできます。3-3 “システムメモリ”の項をご参照ください。

注7 接点7360(0.1秒クロック)とCNT命令を利用して停電記憶のタイマや、スタート条件とリセット条件の違うタイマを実現することができます。



注8 ZW501CU3では、タイマ接点とカウンタ接点の処理が、つぎのようになっています。

- 1) タイマ入力条件が、ON中0.1秒クロックの立上りによってタイマ接点処理を行ないます。よって、タイマの現在値を“0000”にし0.1秒クロック立上りを検知しない、短時間だけタイマ入力条件が、ONしても、タイマ接点はONしません。
- 2) カウンタ接点  
カウントパルスの立上りの1スキャン又は、リセット条件が入っている間の毎スキャンにカウンタ接点処理を行ないます。リセット条件と、カウントパルスの両方が、入力されない時はカウンタ現在値が“0000”に変えられてもカウンタ接点は、ONしません。

## (10) CNT(カウンタ命令)

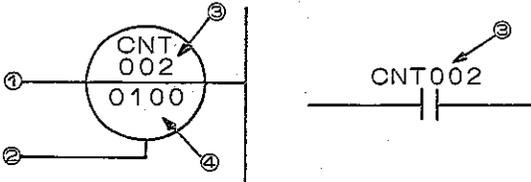
CNT命令は計数入力の立上りで1回計数する減算式のカウンタです。

リセット入力ONの間、計数入力OFF→ONに変化しても計数は行われず、現在値=設定値を維持し、CNT接点はOFFになっています。

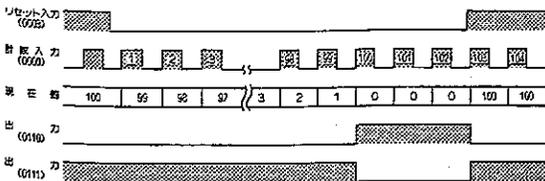
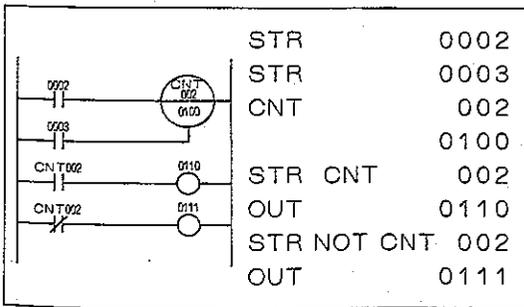
リセット入力OFFの間に、計数入力OFF→ONに変化することに現在値は-1され、現在値が0になるとCNT接点がONし、リセット入力OFFの間この状態を保持します。

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

CNT命令のシンボル      CNT接点のシンボル

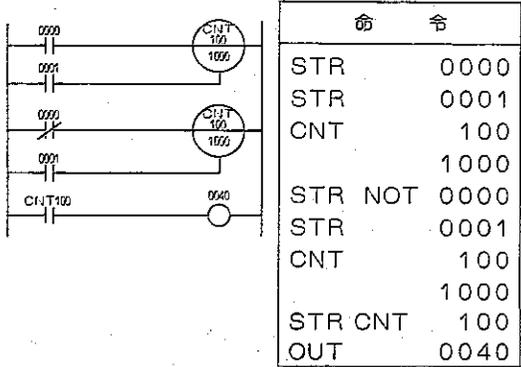


- ①計 数 入 力    (OFF→ONを検知)
- ②リセット入力    (ONでリセット)
- ③CNT 番号    000~177(8道)…… TMR、MDと共通使用
- ④設 定 値    0001~1999(BCD)

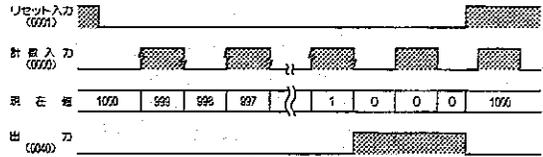


**注1** CNT番号は、TMR、MDに共通使用していますので、TMR、MDに使用した番号は、CNTに使用しないでください。万一、同一番号を使用した場合、プログラム(ZW-101PG1)等のプログラムチェックによってエラー表示します。又、同一CNT番号を使用してもエラー表示しますが意図的に同一番号を使用する場合、この警告は無視してください。

(例) 計数入力の立上り、立下りで計数するカウンタ。



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

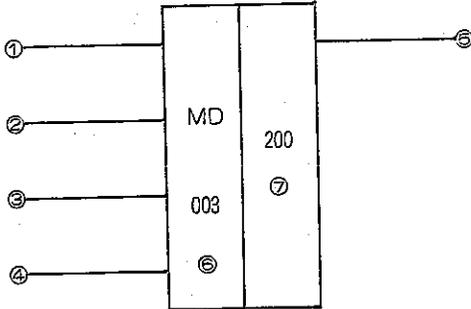


- 注2** CNT接点は、CNT番号と同じ番号を指定し、a接点、b接点を何個でも使用できます。
- 注3** カウントアップすると以後の入力を無視します。再び、計数をはじめるときはリセット入力を一旦ONした後、再びOFFにするか、プログラム(ZW-101PG1)等により、強制リセット後に計数を開始してください。
- 注4** 計数入力と、リセット入力と同時にONの場合、リセットが優先されます。
- 注5** CNTの現在値はb.000~b.377の256バイトに格納されます。3-2(4)“TMR、CNT、MDのデータ格納領域”の項をご参照ください。
- 注6** 停電時カウンタは現在値を記憶しています。ただしリセット入力が電源投入時ONとなる場合、現在値がリセットされてしまいます。停電時にも現在値を記憶する必要がある場合、電源投入時OFFとなるリセット入力を加えてください。
- 注7** リセット入力はシステムメモリ(#202)にリセット条件を設定することにより“OFFでリセット”することもできます。
- 注8** ZW-501CU3のとき4-3-(9)項の**注8**を参照ください。

## (11) MD(メンテナンスディスプレイ)

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

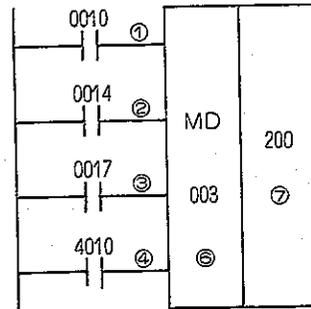
### (1) シンボルの説明



① ② ③	入力情報	⑦のMDデータと共に外部に出力する接点情報で0000~7337の各リレー、TMR・CNTの接点を使用できます。
④	出力指示条件	⑥で指定したMD番号のデータメモリ又はリレー領域に、①、②、③の接点情報および⑦のMDデータを出力するかどうか指示する入力で、0000~7337の各リレー、TMR・CNTの接点を使用できます。ONのとき出力されます。OFFになっても接点情報、MDデータは変化しません。
⑤	MD拡張出力	MD命令を同一出力指示条件で連続して使用するとき、それぞれのMD命令に④の条件をプログラムする必要はありません。詳細は次項“(2)MD命令のプログラム手順”の項をご参照ください。
⑥	MD番号	MD命令は①、②、③の接点情報、⑦のMDデータの各情報を格納するデータメモリ領域としてTMR、CNTの現在値格納領域(b.000~b.377)またはリレー領域(0000~7337)を使用します。 (1)TMR、CNT領域を使用するとき TMR、CNTと同様000~177の番号でプログラムし、情報はプログラマ等でモニタします。 (2)TMR、CNTで使用した番号と重複して使用することはできません。

		②リレー領域を使用するとき バイトアドレス Co.×××でプログラムします。たとえば Co.000とプログラムすると、Co.000、Co.001の2バイトがMD用の領域となります。出力ユニットが装着されている領域を使用すると、①、②、③の接点情報と、⑦のMDデータを外部出力(表示)することができます。
⑦	MDデータ	BCDコードで000~999の任意の数値を使用することができます。工程番号、リレー番号、外部機器番号等と関連付けてプログラムします。

### (2) MD命令のプログラム手順



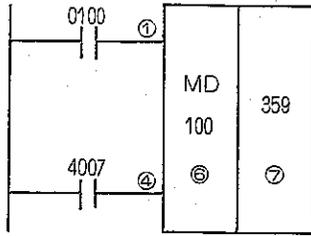
```

STR    0010  —①
STR    0014  —②  入力情報
STR    0017  —③
STR    4010  —④  出力指示
MD     003   —⑥  MD番号
       200   —⑦  MDデータ
    
```

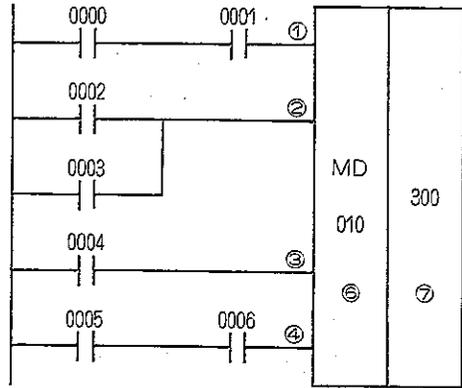
注1 入力情報をモニタ（外部出力）する必要のない場合、プログラムする必要はありません。

注2 入力情報、出力指示条件とも単一条件でない複雑な論理演算結果でもかまいません。

(例1)

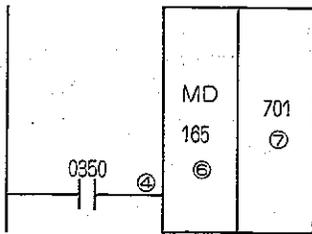


STR 0100 —① 入力情報  
 STR 4007 —④ 出力指示  
 MD 100 —③ MD番号  
 359 —② MDデータ



STR 0000 —①  
 AND 0001 —①  
 STR 0002 —②  
 OR 0003 —②  
 STR 0004 —③  
 STR 0005 —④  
 AND 0006 —④  
 MD 010 —⑤ MD番号  
 300 —⑥ MDデータ

(例2)

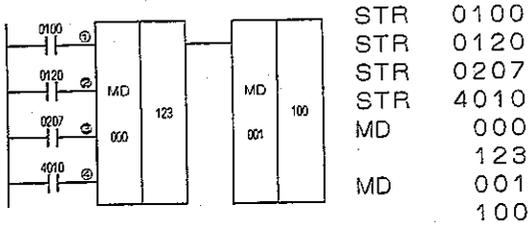


STR 0350 —④ 出力指示  
 MD 165 —⑤ MD番号  
 701 —⑥ MDデータ

MD命令演算時のスタックレジスタの推移

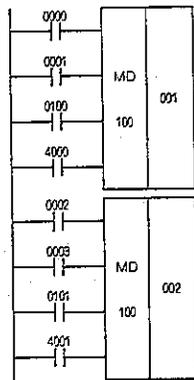
	アキュムレータ		スタックレジスタ		
	ACC		S1	S2	S3
STR 0000					
AND 0001					
STR 0002					
OR 0003					
STR 0004					
STR 0005					
AND 0006					
MD 010 300	出力指示④		入力情報③	入力情報②	入力情報①

注3 MD命令の演算実行後もアキュムレータおよびスタックレジスタの状態は変化しません。従って同一出力指示条件でMD命令を連続使用するときには次のようにプログラムすることができます。

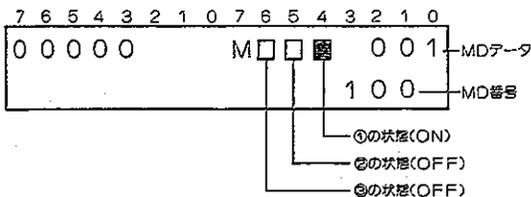


(3) MD情報のモニタ

プログラマ(ZW-101PG1)でMD情報をモニタすると、次のように表示されます。



(MD番号100をモニタ)



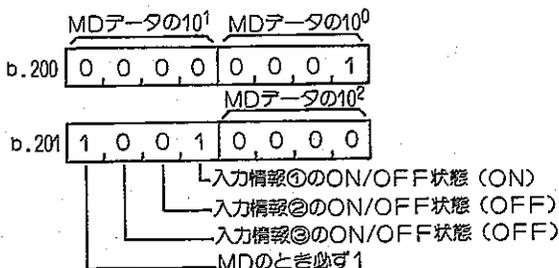
プログラマの表示から次のような情報が得られます。

MDデータが001であるから

- a. 補助リレー4000がONで4001はOFF
- b. 表示中の入力情報は
  - ①……0000 (ON)
  - ②……0001 (OFF)
  - ③……0100 (OFF)

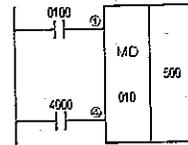
参考 MD番号100のMD情報はデータメモリのb.200、

b.201に格納されています。



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

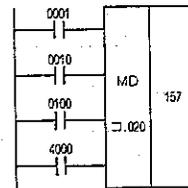
下図のような場合、MD 010の演算時入力情報①はスタックレジスタS1に、出力指示条件④はアキュムレータに格納されています。スタックレジスタS2、S3にはそれ以前の演算で使用された中間結果が残っているため、MD情報としては全く無意味なものです。



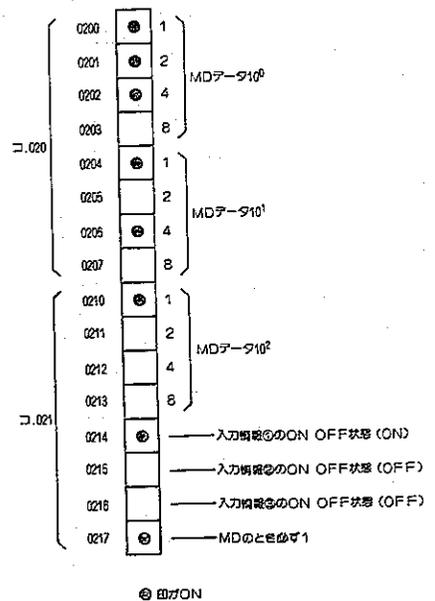
注2 MD番号は本例のように重複して使用することができますが、出力指示条件が同時にONになった場合、プログラム順が後の方の出力指示条件が有効となります。(例では4001が有効)

(4) MD情報の外部出力

MD番号の替りにデータメモリのリレー領域をバイトアドレスで指定することにより、MD情報を外部に出力したり、データリンク機能を使って他のPCに伝達したりすることができます。



Co.020と指定することで、Co.020、Co.021の2バイトにMD情報が出力されます。Co.020、Co.021には出力ユニットを装着しておきます。



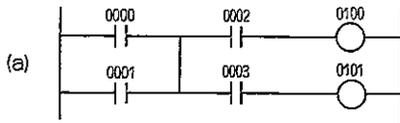
## 4-4 ラダー設計に関する留意事項

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

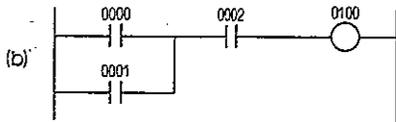
以下のリレー盤でのラダー設計とPCでのラダー設計の相違点を十分理解し、効率の良いラダー図を設計してください。

### (1) リレー盤用ラダー図から書換えを必要とする回路

(例1)



(a)のラダー図は、このままではPCでは使用できません。



(b)の部分は

命 令	
STR	0000
OR	0001
AND	0002
OUT	0100

というプログラムで演算可能です。

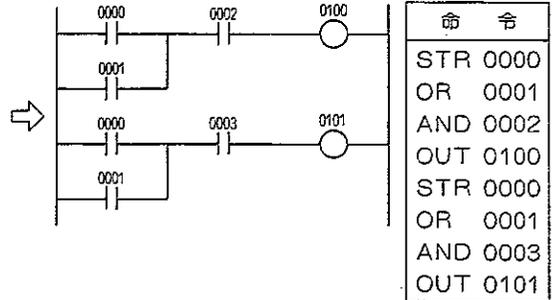
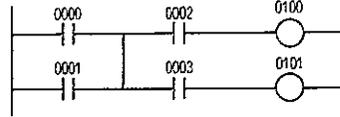
(b)のプログラムを演算する場合のACC (アキュムレータ) の状態推移は、次のようになります。

命 令	ACCの内容
STR 0000	
OR 0001	の演算結果
AND 0002	の演算結果
OUT 0100	の演算結果

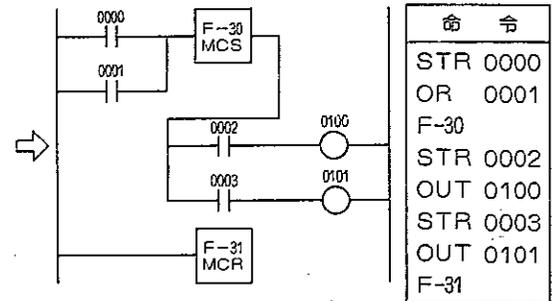
ACCにはプログラムの1命令を演算することに演算結果が0または1で入ります。

したがってAND 0002まで演算すると の演算結果はすでに消滅して、これを0003に反映することはできません。

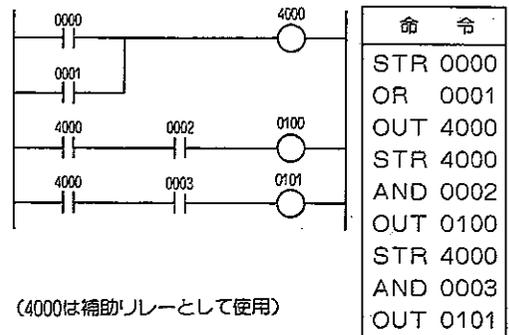
PC用のラダー図としては次のように書換えます。



または



または

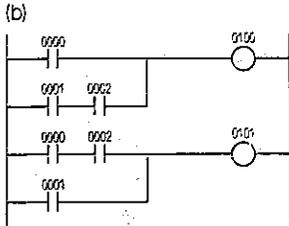
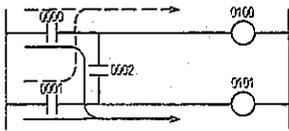


(4000は補助リレーとして使用)

F-30 (MCS)、F-31 (MCR)に関しては、4-6 “応用命令の説明” をご参照ください。

(例2)

(a)のリレー盤のラダー図では、0002に0000からと、0001からの両方向に電流が流れ、(b)のPC用に書き換えたラダー図と同様の動作をします。

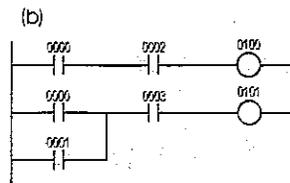
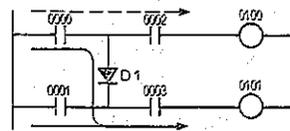


命 令	
STR	0000
STR	0001
AND	0002
OR STR	
OUT	0100
STR	0000
AND	0002
OR	0001
OUT	0101

PCでは(a)の0002のようにラダー図上の1つの接点シンボルに両方向に電流が流れるような考え方は成り立ちません。PCの演算はプログラムメモリをアドレス0からEND命令まで順次スキャンする方式のため、ラダー図上の同一接点シンボルを2度通るような処理は行いません。

(例3)

(a)のリレー盤の回路は通り込み防止ダイオードD1の働きにより、0001から0002には電流は流れず、(b)のPC用に書き換えたラダー図と同様の動作をします。



命 令	
STR	0000
AND	0002
OUT	0100
STR	0000
OR	0001
AND	0003
OUT	0101

PCでは(a)のD1のような通り込み防止ダイオードをプログラムすることはできません。

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

リレーでは1接点増えると  
①リレーの追加  
②配線の追加

無限

PCのデータメモリは「接点数が無数のリレー」。  
1接点増えても、スペース、コスト、信頼性等への影響はありません。

(2) 入出力一括処理方式

3-6 “運転サイクル” で説明しましたように、W16/W51では毎スキャンサイクルに“入出力処理”というデータメモリと入出力ユニット間でデータの交換を行う処理があります。

入出力処理では、ベースユニットに装着された入出力ユニットをリレー番号の若い方から順にスキャンし、

①入力ユニットであれば

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

②出力ユニットであれば

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

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

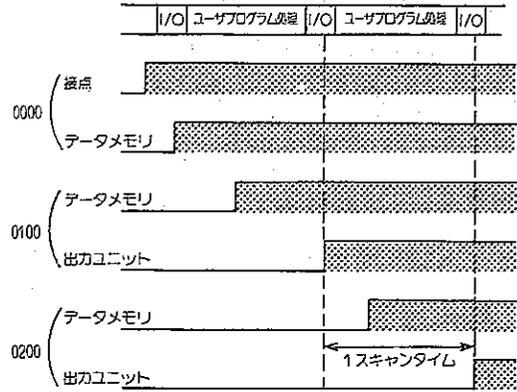
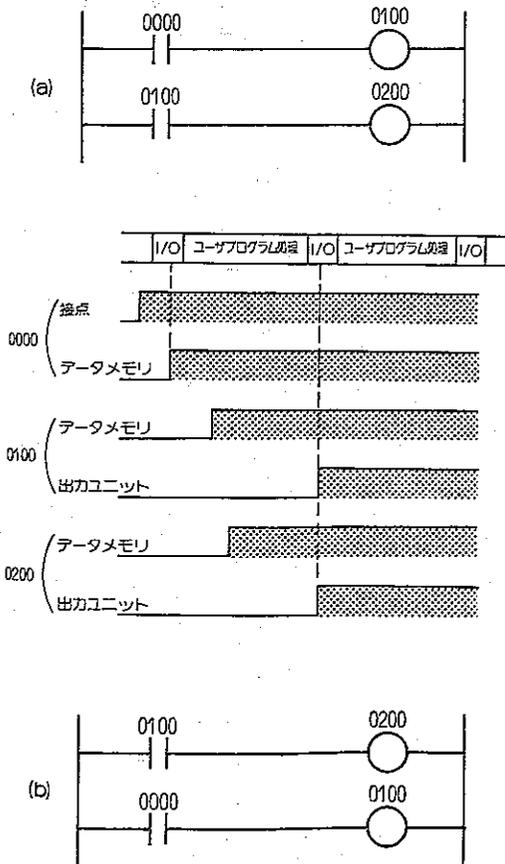
このように入出力ユニットに対する処理を一括して行うPCでは次の事項を念頭に置いてラダー設計をする必要があります。

- (1) 外部接点のON/OFF状態の変化は1スキャンに1度の入出力処理でデータメモリに取り込まれます。したがって、ユーザプログラム処理中に外部機器のON/OFF状態が変化しても、そのスキャンサイクル中はデータメモリ（入力として割当てられているもの）の内容は変化しません。このため“入力レーシング現象”(3-6(3)⑧)“ユーザプログラム処理”参照)は発生しません。
- (2) 演算結果のON/OFF状態をデータメモリから出力ユニットに書き込むのは1スキャンに1度の入出力処理で行われます。したがって演算結果が出力ユニットに出力されるのは、次のスキャンの入出力処理ということになります。

### (3) プログラム順序による影響

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

- (1) プログラム順を入れ替えると異なった動作をする場合があります。

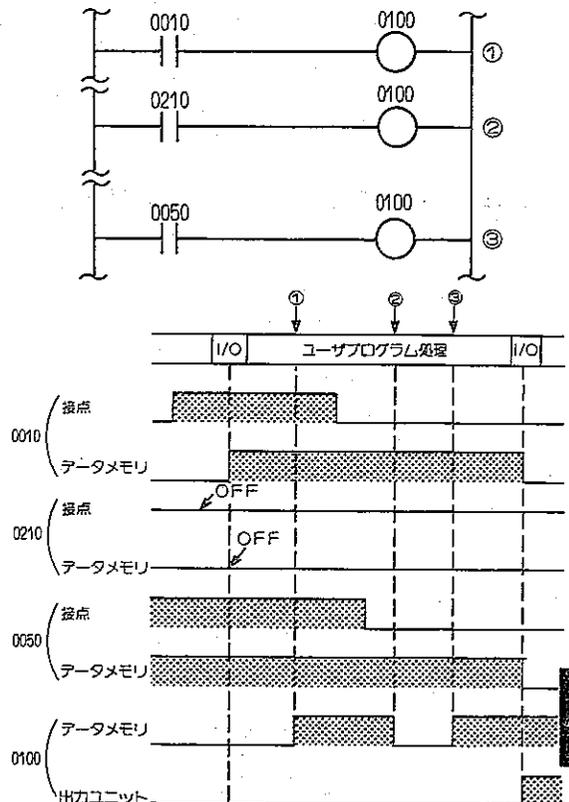


- (a)のプログラムでは、入力0000がONになると、出力0100、0200は同一スキャン内でONとなりますが、(b)では1スキャン遅れて0200がONになります。

コイルの補助接点を使う場合、「コイルの前に書かれた補助接点の状態変化は、コイルの状態が変わった次のスキャンに生じる」ということを考慮してプログラムする必要があります。

- (2) コイルの2重使用

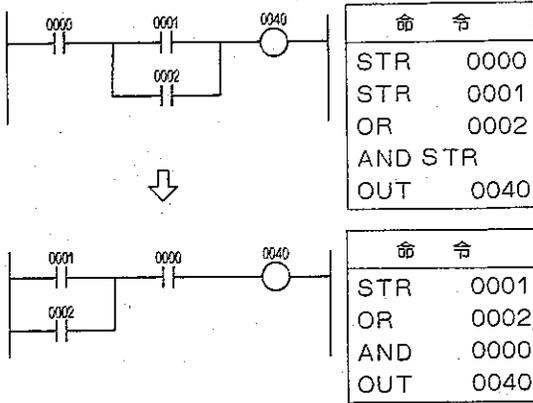
同一のリレー番号をコイルとして複数回使用すると、それぞれのプログラム内容に応じデータメモリの内容は変化し、出力ユニットには一番最後に書かれたプログラムの演算結果がデータメモリから書き込まれます。



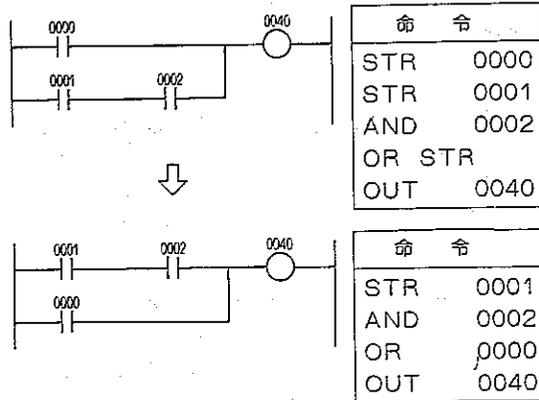
### (4) プログラムの簡略化

シーケンス回路によっては、回路を書換えることによってプログラムが簡単になることがあります。

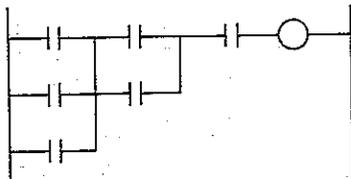
(例1)



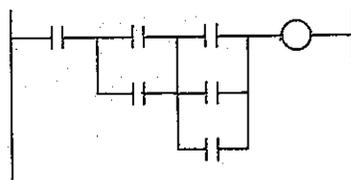
(例2)



一般に左下りの回路を作るとプログラムが簡単になります。



左下りの回路

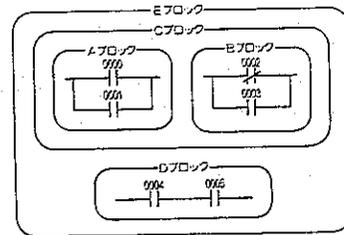
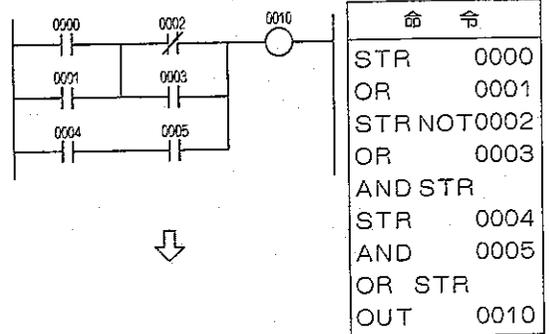


右下りの回路

### (5) 直並列回路のプログラム

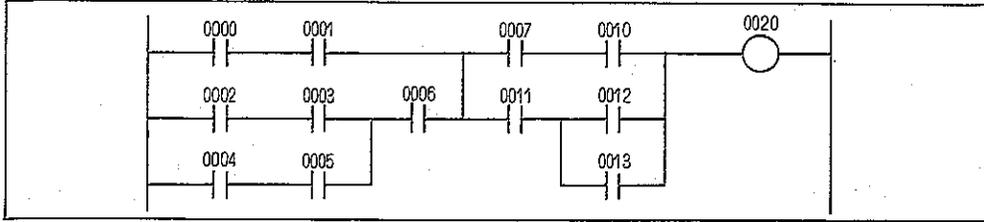
直並列回路をプログラムする場合にはまず、小さなブロックに分割し、その小さなブロック毎にプログラムし、最終的に1つの大きなブロックになるようにします。

(例1)



命 令	アキュムレータ ACC	スタックレジスタ S1
A	STR 0000	← 直前のACCの状態
	OR 0001	
B	STR NOT 0002	←
	OR 0003	←
C	AND STR	←
D	STR 0004	←
	AND 0005	←
E	OR STR	←
OUT	0010	

(例2)



命令	アキュムレータ ACC	スタックレジスタ		
		S1	S2	S3
STR 0000	0000	直前のACCの状態		
AND 0001	0001			
STR 0002	0002	0000 0001		
AND 0003	0003	0000 0001		
STR 0004	0004	0002 0003	0000 0001	
AND 0005	0005	0002 0003	0000 0001	
OR STR	0002 0003 0004 0005	0000 0001		
AND 0006	0002 0003 0006 0004 0005	0000 0001		
OR STR	0000 0001 0002 0003 0006 0004 0005			
STR 0007	0007	0000 0001 0002 0003 0006 0004 0005		
AND 0010	0007 0010	0000 0001 0002 0003 0006 0004 0005		
STR 0011	0011	0007 0010	0000 0001 0002 0003 0006 0004 0005	
STR 0012	0012	0011	0007 0010	0000 0001 0002 0003 0006 0004 0005
OR 0013	0012 0013	0011	0007 0010	0000 0001 0002 0003 0006 0004 0005
AND STR	0011 0012 0013	0007 0010	0000 0001 0002 0003 0006 0004 0005	
OR STR	0007 0010 0011 0012 0013	0000 0001 0002 0003 0006 0004 0005		
AND STR	0000 0001 0007 0010 0002 0003 0006 0011 0012 0004 0005 0013			
OUT 0020	0000 0001 0007 0010 0002 0003 0006 0011 0012 0004 0005 0013			

## 4-5 応用命令に関する留意事項

### (1) 数値の表現方法

#### (1) 2進数 (Binary Code)

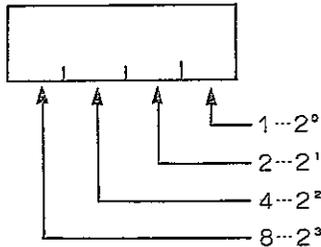
我々が日常使用している10進数では0~9の数字を使用します。ロジックの世界では0 (OFF)と1 (ON)の2つの状態しか存在しませんが、この0と1であらゆる数値を表現することができます。

0と1で表現した数値を2進数といいます。

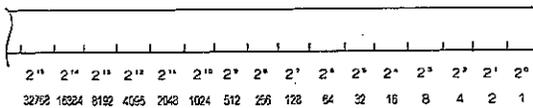
10進数では0、1、2、...8、9と数字が増えると、次に10と桁上げが起こりますが、2進数ではこの桁上げが0、1の次に10という形で起こります。したがって10 (イチゼロと読む) は10進数の2を意味します。以下同様に11→100、111→1000と桁上げが起こります。

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

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

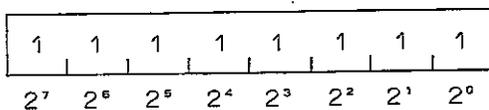


以下同様にして、各桁は $2^n$ の重みを持ちます。



2進数の各桁を「ビット」と呼びます。

W16/W51ではレジスタは8ビットで構成されています。8ビットがすべて1のときの様子を調べると次のようになります。



それぞれのビットの重みを合計すると

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

すなわち8ビットで0~255の10進数を表現することができます。

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

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

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

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

#### (3) 8進数と16進数

PCの内部では、数値はすべて2進数 (バイナリコード) 又はBCDコードで処理されます。しかし、プログラムの書込みや、演算結果のモニタを2進数 (0と1のビットパターン) で行うとキー操作や重み計算が面倒なため、プログラムに2進→10進変換機能 (BCD→10進変換機能) を持たせ、10進数でプログラムの書込み、モニタを可能にしています。

ただし、PCをビット演算機能を中心に考えたとき、ビットパターンを直感的に連想できる他の数値表現方法の方が望ましい場合が多々あります。8進数および16進数は、ビットパターンとの相性がよくPCやコンピュータでよく使われます。

##### a. 8進数

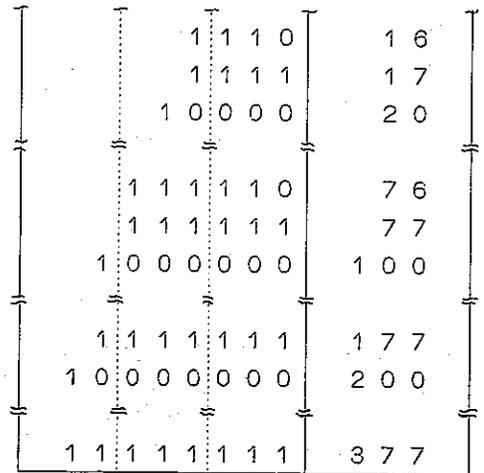
10進数では9→10、2進数では1→10と桁上げが起こりますが、8進数では7→10と桁上げが起こります。

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

桁上げ

桁上げ

桁上げ



レジスタは8ビットで構成されますので、0~377<sub>10</sub>の範囲を取り得ます。

●データメモリのアドレス、システムメモリのアドレス、プログラムメモリのアドレスも8進数で表現されます。

b、16進数

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

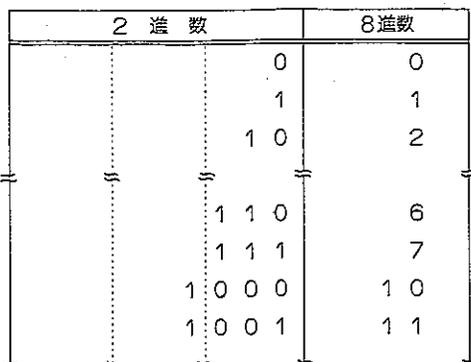
すなわち、0、1、2...7の次は8ではなく、10と桁上げが起こります。同様にして17→20、77→100と桁上げが起こります。

●8進数と2進数は次のように対応します。

2進数は3桁で0~7を表わし、111→1000と桁上げが起こります。

8進数は1桁で0~7の範囲をとり、7→10と桁上げが起こります。

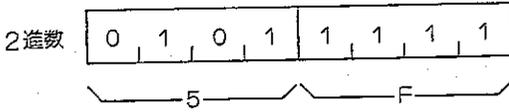
2進数、8進数がともに7の次に桁上げが起こる性質から、2進数を3桁ごとに区切ると、これに1桁の8進数を当てはめることができます。



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



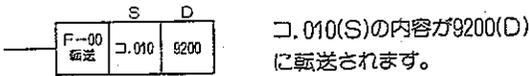
- 16進数と2進数は次のように対応します。  
2進数を4ビットごとに区切り、これに16進数の1桁を割り当てます。



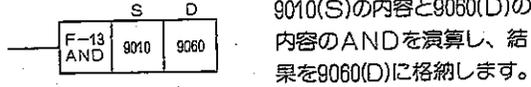
## (2) ソースとデスティネーション

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

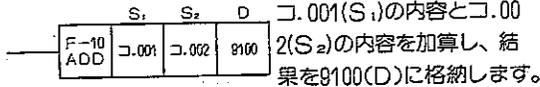
(例1)



(例2)

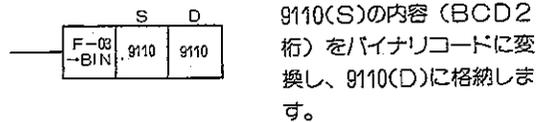


(例3)



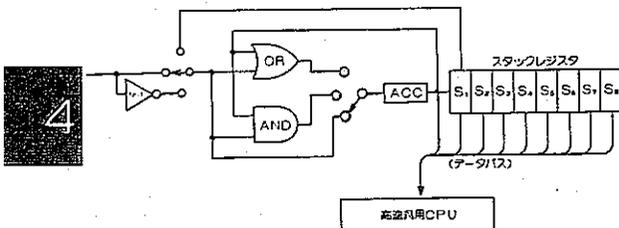
ソース側のレジスタの内容は演算実行後も変化しません。ただしソースとデスティネーションに同一レジスタを使用することも可能で、この場合は命令によってはソース(すなわちデスティネーション)の内容が変化します。

(例4)

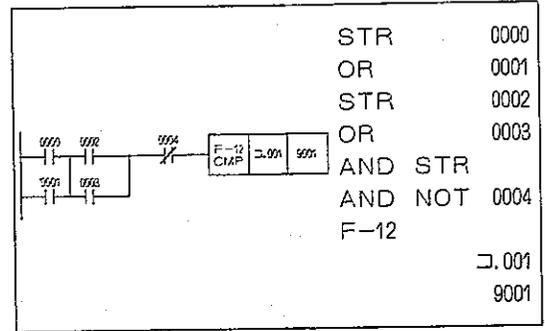


## (3) 応用命令とスタックレジスタ

W16/W51ではTMR、CNT、MDの各命令とF00~F73の応用命令は高速度用CPUで処理しています。これらの命令はACC(アキュムレータ)とSR(スタックレジスタ)の内容がデータバスを經由してCPUに送られ、これを演算条件として実行されます。

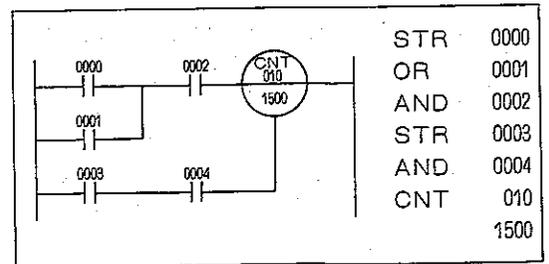


- (例1) CNT、MD、F-60、F-61、F-62を除く応用命令は、ACCの内容のみを演算条件として実行されます。



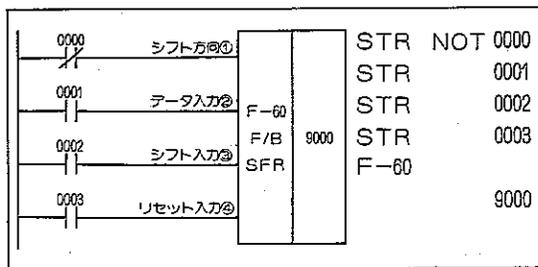
命令	アキュムレータ ACC	スタックレジスタ S <sub>i</sub>
STR 0000	0000	
OR 0001	0001	
STR 0002	0002	0000
OR 0003	0002 0003	0000 0001
AND STR	0001 0002	
AND NOT 0004	0001 0002 0004	
F-12	条件成立のとき演算	

(例2) CNT命令の場合



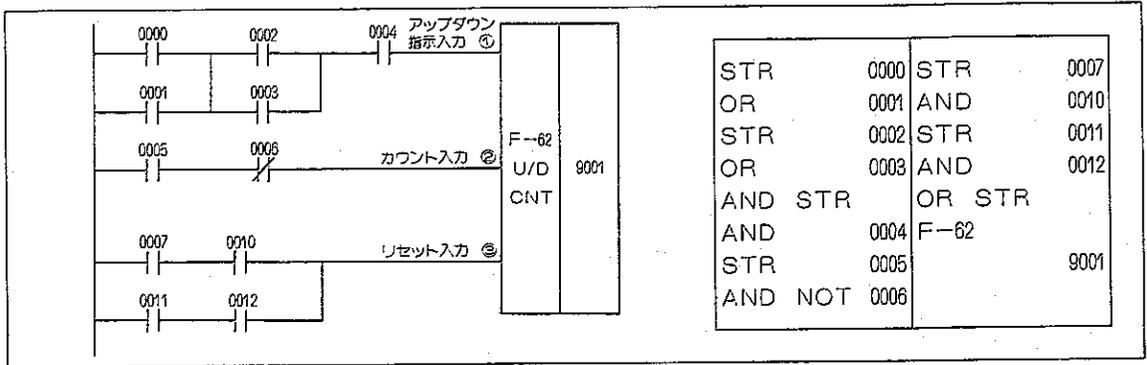
命 令	アキュムレータ ACC	スタックレジスタ S <sub>1</sub>
STR 0000	0000 — / —	
OR 0001	0000 — / — 0001 — / —	
AND 0002	0000 0002 — / — 0001 — / —	
STR 0003	0003 — / —	0000 0002 — / — 0001 — / —
AND 0004	0003 0004 — / —	0000 0002 — / — 0001 — / —
CNT 010 1500	リセット入力	計数入力

(例3) F-60ではACC、スタックレジスタ(S<sub>1</sub>~S<sub>3</sub>)が演算条件となります。



命 令	アキュムレータ ACC	スタックレジスタ		
		S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>
STR NOT 0000	0000 — / —			
STR 0001	0001 — / —	0000 — / —		
STR 0002	0002 — / —	0001 — / —	0000 — / —	
STR 0003	0003 — / —	0002 — / —	0001 — / —	0000 — / —
F-60	リセット入力 ④	シフト入力 ③	データ入力 ②	シフト方向 ①

(例4) スタックの内容は複雑な直並列回路でもかまいません。



命 令	アキュムレータ ACC	スタックレジスタ		
		S <sub>1</sub>	S <sub>2</sub>	S <sub>3</sub>
STR 0000	0000			
OR 0001	0000 0001			
STR 0002	0002	0000 0001		
OR 0003	0002 0003	0000 0001		
AND STR	0002 0001			
AND 0004	0000 0001 0002 0003 0004			
STR 0005	0005	0000 0001 0002 0003 0004		
AND NOT 0006	0005 0006	0000 0001 0002 0003 0004		
STR 0007	0007	0005 0006	0000 0001 0002 0003 0004	
AND 0010	0007 0010	0005 0006	0000 0001 0002 0003 0004	
STR 0011	0011	0007 0010	0005 0006	0000 0001 0002 0003 0004
AND 0012	0011 0012	0007 0010	0005 0006	0000 0001 0002 0003 0004
OR STR	0007 0010 0011 0012	0005 0006	0000 0001 0002 0003 0004	
F-62	リセット入力 ③	カウント入力 ②	アップダウン指示入力 ①	

この例ではSTR0011演算時スタックレジスタを3段目(S<sub>3</sub>)まで使います。

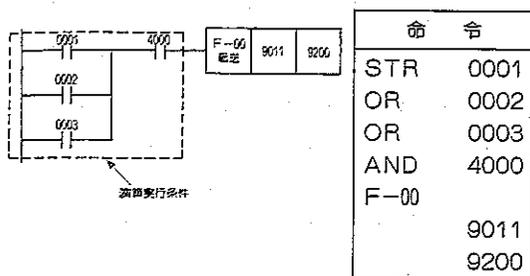
4

## (4) 演算実行条件

(1) 応用命令の演算実行条件（演算を実行するかしないかの条件）は、1接点のON/OFFに限らず、複雑な直並列回路を用いることが可能です。

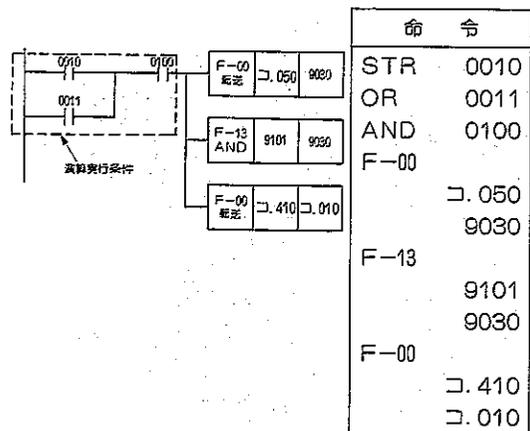
（4-5(3) “応用命令とスタックレジスタ” 参照）

(例)



(2) 演算実行条件が共通の場合、次のように続けてプログラムすることができます。

(例)



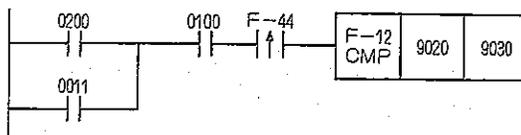
注1 4-5(6) “倍長演算”の項をご参照ください。

(3) 応用命令には、演算実行条件が成立した場合の処理方式に次の2種類の形態があります。(F-30、F-31、F-40、F-41、F-42、F-43は除く)

①	演算実行条件が成立している間、毎スキャンサイクル演算を実行するもの	F-12 (レジスタ間の比較) F c12 (レジスタと定数の比較) F-81 (非同期シフトレジスタ)
②	演算実行条件が成立した最初の1スキャンサイクルのみ演算を実行するもの	上記以外のデータ処理命令

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

注2 F-12、F c12、F-81で、演算実行条件のOFF→ONの変化時のみ演算させる必要がある場合、F-44(立上り微分命令)を使用します。



(4) 演算実行条件が不成立の場合（演算実行条件がOFF→ONへの変化時のみ演算を実行する命令では、ON中の以降のスキャンサイクルも含まず）、演算は実行されず、テストディネーション側のレジスタの内容は不変です。またフラグに影響を与える命令の場合、フラグはクリアされます。

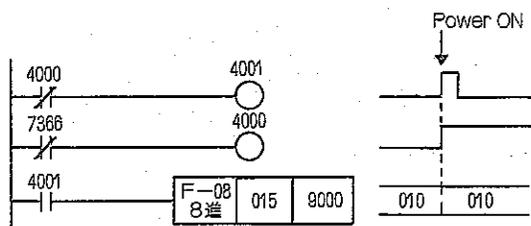
注3 フラグに関しては4-5(5) “データ処理命令とフラグ”をご参照ください。

(5) 演算実行条件のOFF→ONを検出するための1ビットのメモリを微分メモリといいます。

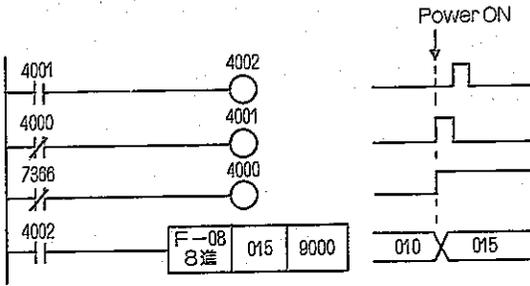
微分メモリは、停電後の電源再投入時、停電直前の状態を保持しています。

ただし、プログラムの挿入・削除直後は、ONの状態となります。

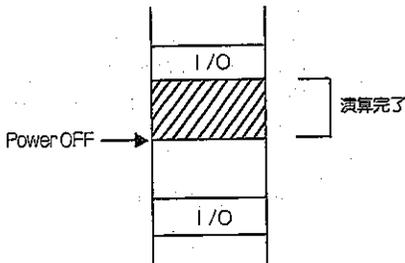
従って次の回路では電源投入時、F-08命令は実行されません。



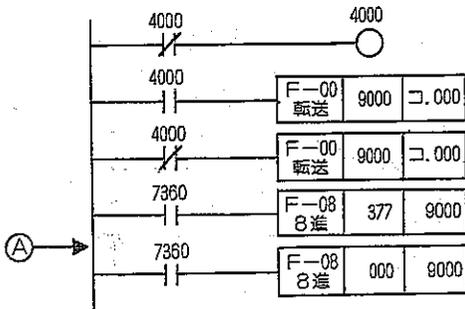
この場合、1回目のスキャンで微分メモリをOFFしてから2回目のスキャンでF-08命令を実行するようにします。



(6) W51の演算では、演算途中で電源をOFFすると、その時点で演算を中止します。



従って、電源再投入時、演算の完了している部分のデータは保持されています。



例えば上記回路では、コ.000の値は000になりますが、(A)の部分で電源をOFFした場合、レジスタ9000には、377 (B)のデータがセットされたままとまります。そのため、電源が再投入されると、コ.000は1スキャンタイムだけ、全点ONします。

## (5) データ処理命令とフラグ

### (1) フラグの種類

フラグ(Flag…旗)は、演算結果を以降のステップの演算に反映させるための信号で、W16/W51にはノンキャリーフラグ、エラーフラグ、キャリーフラグ、ゼロフラグの4種類のフラグがあり、データメモリの7354~7357の4ビットに割当てられています。

ノンキャリーフラグ	エラーフラグ	キャリーフラグ	ゼロフラグ
7354	7355	7356	7357

### (2) フラグに影響を与える命令

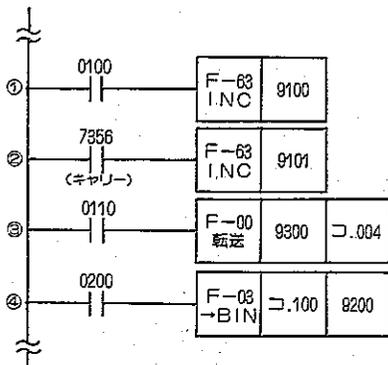
次の17種の命令では演算結果に従いフラグがセットされます。

1	F-03	BCD(2桁)→BIN変換
2	F-10	レジスタ間の加算(BCD)
3	F c10	レジスタとBCD定数の加算
4	F-11	レジスタ間の減算(BCD)
5	F c11	レジスタとBCD定数の減算
6	F-12	レジスタ間の比較
7	F c12	レジスタと定数の比較
8	F-15	レジスタ間の乗算
9	F c15	レジスタとBCD定数の乗算
10	F-16	レジスタ間の除算
11	F c16	レジスタとBCD定数の除算
12	F-53	BCD(4桁)→BIN変換
13	F-60	両方向シフトレジスタ
14	F-61	非同期シフトレジスタ
15	F-62	BCDアップダウンカウンタ
16	F-63	加算カウンタ
17	F-64	減算カウンタ

上記以外の命令ではフラグは影響を受けません。

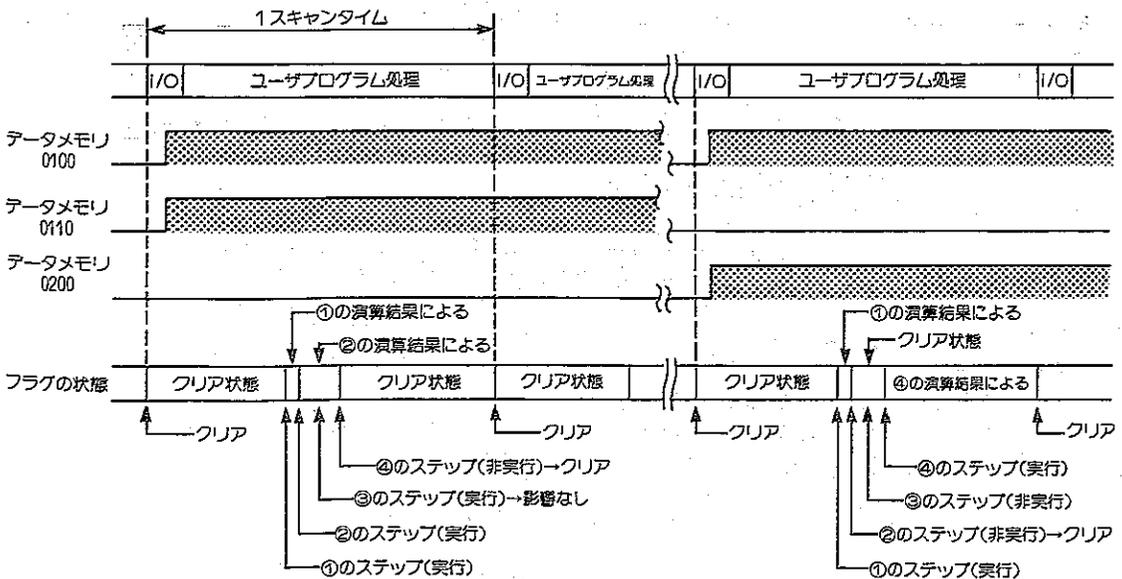
- ③ スキャンサイクル中でのフラグの推移
- ① 毎スキャンサイクルのユーザプログラム処理に先立ち、フラグはクリアされます。3-6 “運転サイクル” の項を参照願います。
- ② フラグに影響を与える命令の処理に入ると、
- a. その命令の実行条件が成立しているとき  
命令の演算結果によりフラグがセットされます。
- b. その命令の実行条件が不成立のとき  
フラグをクリアします。
- ③ フラグに影響を与えない命令の処理では、実行・非実行にかかわらず、フラグの状態は変化しません。

以前にフラグに影響のある命令なしとする。



以後フラグに影響のある命令なしとする。

アドレス	命令	
01000	STR	0100
01001	F-63	
01002		9100
01003	STR	7356
01004	F-63	
01005		9101
01006	STR	0110
01007	F-00	
01010		9300
01011		コ.004
01012	STR	0200
01013	F-03	
01014		コ.100
01015		9200

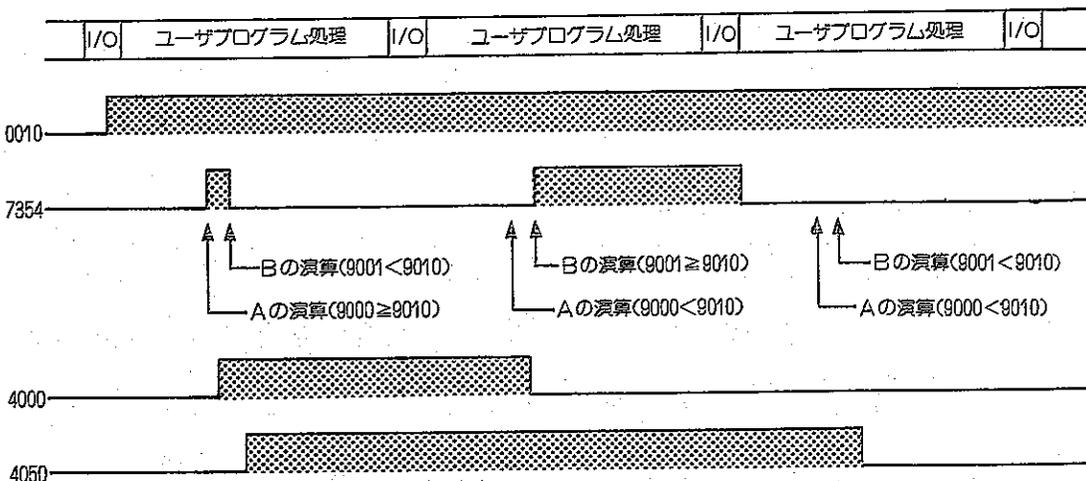
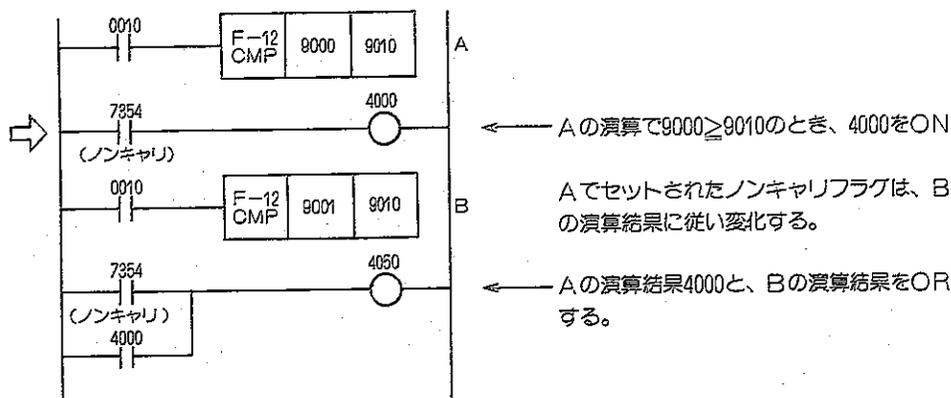


(4) フラグを保持する方法

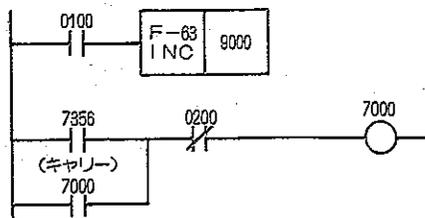
以上のように演算の結果セットされたフラグは、そのスキャンサイクル中、次にフラグに影響を与える命令の処理により変化したりクリアされてしまいます。また次のスキャンサイクルに入るとユーザプログラム処理の前にクリアされてしまいます。

フラグを保持する必要がある場合、以下のように当該命令の直後にフラグの状態をコイル（補助リレー等）に書込んでおきますと次のスキャンサイクルの当該命令の演算まで保持できます。

(例1)  $9000 \geq 9010$  又は  $9001 \geq 9010$  のとき、4050をONにするプログラム

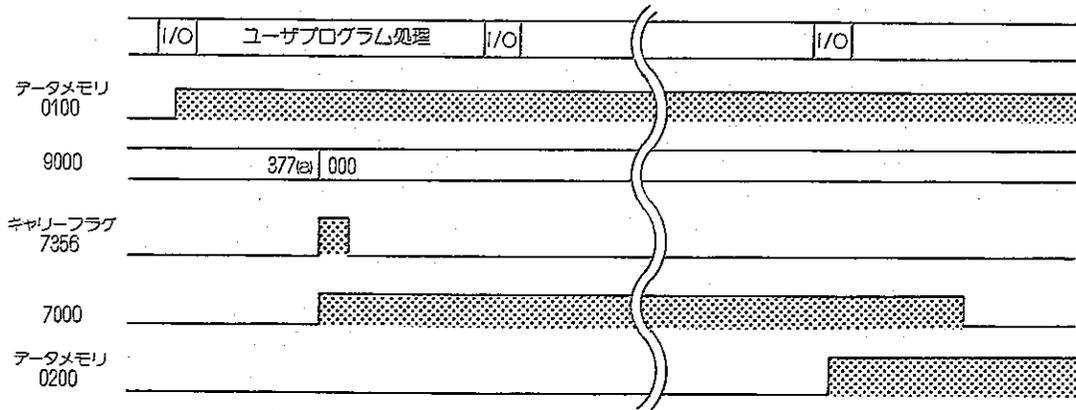


フラグの状態をプログラマ等の周辺装置でモニタしたり、外部に表示する場合は、例1のようにフラグの状態をコイルに書くだけでは1スキャンサイクルしか保持できないため、目で確認するのは困難です。このような場合、右図のようにフラグを自己保持する必要があります。



0200をONにするまで、①の演算によるキャリアフラグ(7356)の状態を自己保持します。





## (6) 倍長演算

(1) 倍長演算機能をもつ命令

次の6種の命令には、2バイト以上のデータの演算を可能とする倍長演算の機能があります。

- ① F-10 レジスタ間の加算
- ② Fc 10 レジスタとBCD定数の加算
- ③ F-11 レジスタ間の減算
- ④ Fc 11 レジスタとBCD定数の減算

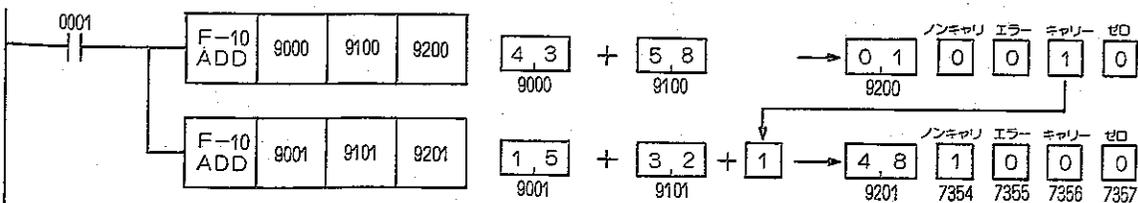
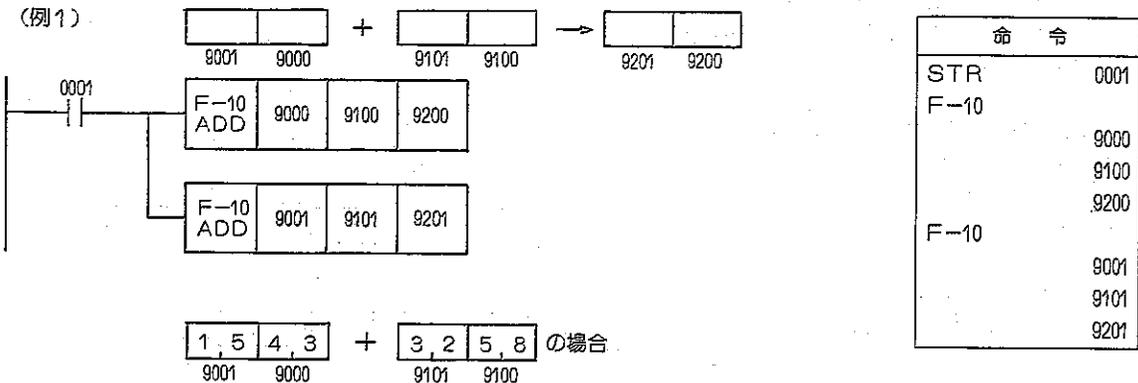
⑤ F-12 レジスタ間の比較

⑥ Fc 12 レジスタと定数の比較

(2) 倍長演算時のプログラム

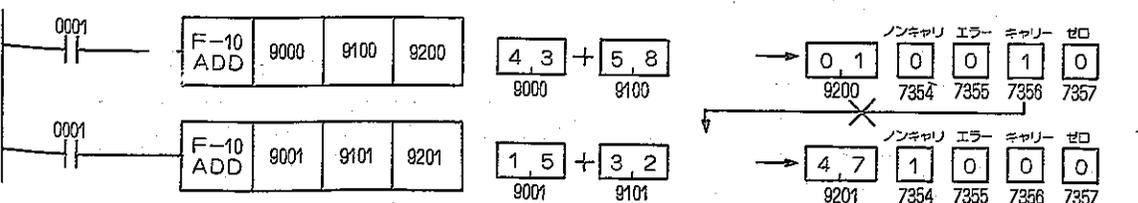
倍長演算は下の桁の演算により発生した桁上げ、桁下げ信号を次の桁の演算に自動的に反映させるもので、次のように演算実行条件に続けて下の桁からプログラムを書込みます。

(例1)

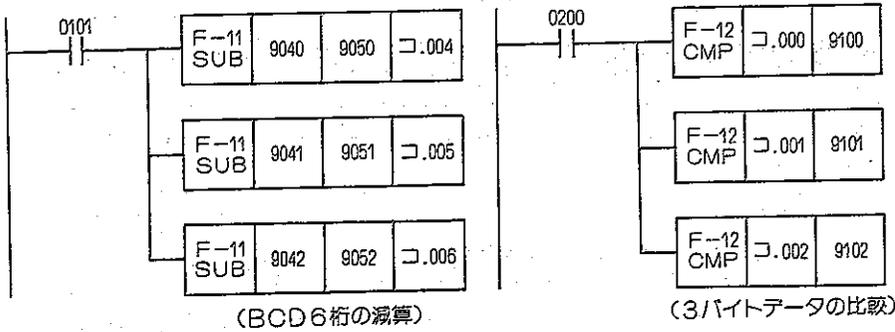


下の桁のキャリーフラグが上の桁の演算時に加算されます。

参 尋次のようにプログラムすると倍長演算になりません。



(例2) 3バイト以上の倍長演算も同様にして可能です。



(3) 倍長演算時の内部処理

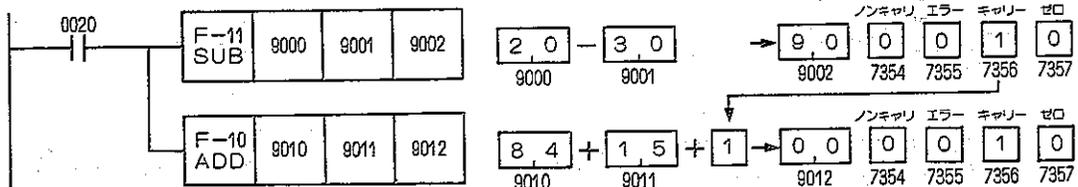
- ① 演算実行条件以後、最初に現われるF-10(Fc10)、F-11(Fc11)、F-12(Fc12)の各命令の演算時は、それ以前のフラグの状態を含めずに演算が行われます。
- ② 共通演算実行条件中、次にF-10(Fc10)、F-11(Fc11)、F-12(Fc12)のいずれかの命令があると次のように演算が行われます。
  - a、直前のキャリーフラグの状態を含めて演算が実行されます。
  - b、ゼロフラグは、直前のゼロフラグの状態と、当該命令の演算によるゼロフラグの状態のANDをとり、いずれも1のときにゼロフラグがセットされます。

F-10(Fc10)	直前のキャリーフラグの状態を加算
F-11(Fc11)	直前のキャリーフラグの状態を消算
F-12(Fc12)	直前のキャリーフラグの状態を消算

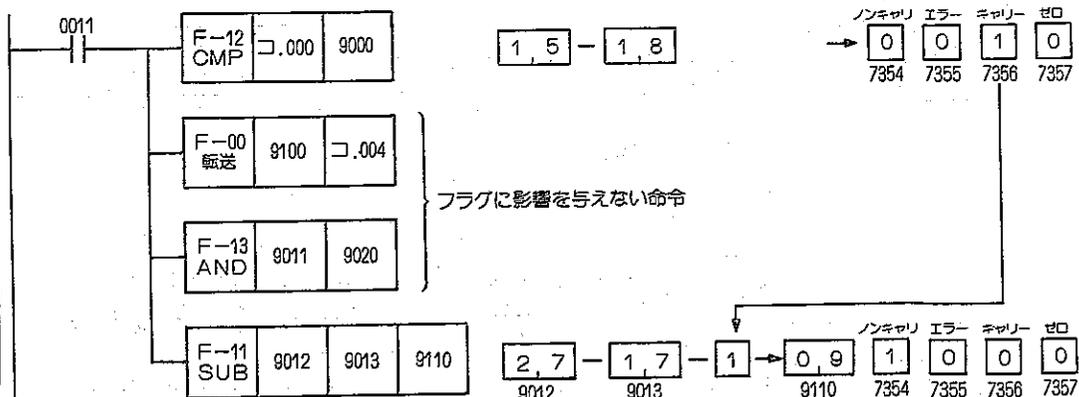
注1 F-12(Fc12)命令は $S_1-S_2$ 又は $S_1-n$ の演算を行い、結果をフラグに格納します。

(4) 倍長演算に関する注意事項

- ① F-10(Fc10)、F-11(Fc11)、F-12(Fc12)は、共通演算条件の形式でプログラムされていると、具種命令間でもフラグを含めた演算が行われます。

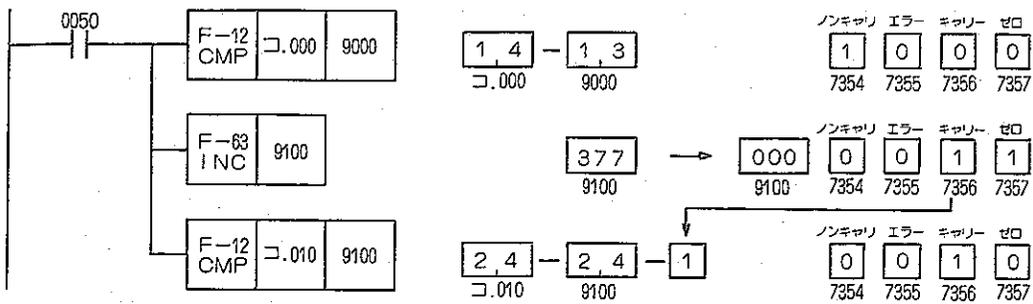


- ② F-10(Fc10)、F-11(Fc11)、F-12(Fc12)の間に、フラグに影響を与えない命令があっても倍長演算として演算されます。



注1 多数の命令が間に入る場合、特にご注意ください。

- ③ F-10(Fc10)、F-11(Fc11)、F-12(Fc12)の間に、フラグに影響を与える命令があると、その命令の演算によるフラグを含めた演算が行われます。



### (7) リレー領域を使用するとき

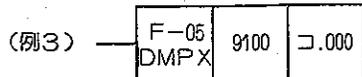
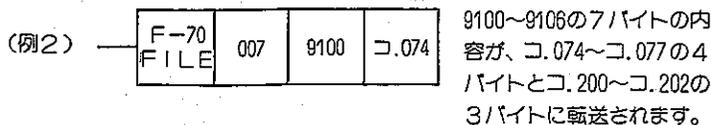
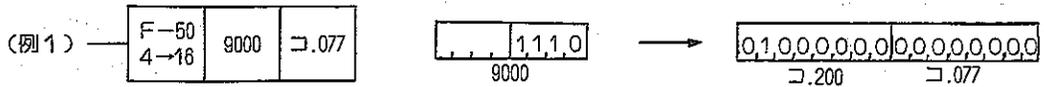
3-2〔5〕“リレー領域のバイトアドレスと絶対アドレス”で説明しましたように入力リレー(0000~0777)、リンクリレー(2000~2777)、補助リレー(4000~4777)、キーブリレー(7000~7377)のリレー領域をバイトアドレスコ.×××としてバイト指定することによりb.000~b.377、9000~9377と同様にレジスタとして扱うことができます。

(1) リレー領域の欠番

リレー番号はグループ間に欠番がありますが、実際

のメモリ上では連続して配列されています。以下の各命令ではソース、デスティネーションが2バイト以上のデータメモリを指示しますが、グループをまたがるようなプログラムをすると、グループ間の欠番を無視した形で演算が行われます。グループ間の欠番を詰めたとときのアドレスを絶対アドレスと呼びます。

F-05	分配	F-53	BCD4桁→16ビットBIN
F-06	抽出	F-54	16ビットBIN→BCD6桁
F-15	レジスタ間の乗算	F-61	非同期シフトレジスタ
Fc15	レジスタとBCD定数の乗算	F-70	nバイト一括転送
F-16	レジスタ間の除算	F-71	8進定数一括転送
Fc16	レジスタとBCD定数の除算	F-72	ファイルレジスタへの分配
F-50	4→16アコーダ	F-73	ファイルレジスタからの抽出
F-51	16→4エンコーダ		



レジスタ9100の内容(データポインタ)は絶対アドレスを使用します。コ.461を分配先とすると、9100の内容は261回となります。

	バイトアドレス	絶対アドレス
入力リレー	コ.000~コ.077	000~077
リンクリレー	コ.200~コ.277	100~177
補助リレー	コ.400~コ.477	200~277
キーブリレー	コ.700~コ.737	300~337

(2) キーブリレーの特殊領域

3-2〔3〕“キーブリレーの特殊領域”で説明しましたようにコ.734~コ.737の4バイトは特殊な領域であり、プログラムでコイル、デスティネーションとしては使用できません。

注1 7365(設定値変更スイッチ)、7367(ゼロクロススイッチ)はコイルとして使用します。

ソース、デスティネーションが2バイト以上のデータメモリを意味する命令では特にご注意ください。

(8) データメモリのブロックと基準アドレス

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

基準アドレス	ブロック	範囲
コ.000	入出力リレー、リンクリレー、補助リレー、キーアリレー	コ.000~コ.737
b.000	TMR、CNTの現在値、MD情報	b.000~b.377
9000	レジスタ	9000~9377
30000	ファイルレジスタ①	30000~30377
30400	ファイルレジスタ②	30400~30777
31000	ファイルレジスタ③	31000~31377
31400	ファイルレジスタ④	31400~31777
32000	ファイルレジスタ⑤	32000~32377
32400	ファイルレジスタ⑥	32400~32777
33000	ファイルレジスタ⑦	33000~33377
33400	ファイルレジスタ⑧	33400~33777
34000	ファイルレジスタ⑨	34000~34377
34400	ファイルレジスタ⑩	34400~34777
35000	ファイルレジスタ⑪	35000~35377
35400	ファイルレジスタ⑫	35400~35777
36000	ファイルレジスタ⑬	36000~36377
36400	ファイルレジスタ⑭	36400~36777
37000	ファイルレジスタ⑮	37000~37377
37400	ファイルレジスタ⑯	37400~37777

注1 コ.000~コ.737のブロックは224バイトです。

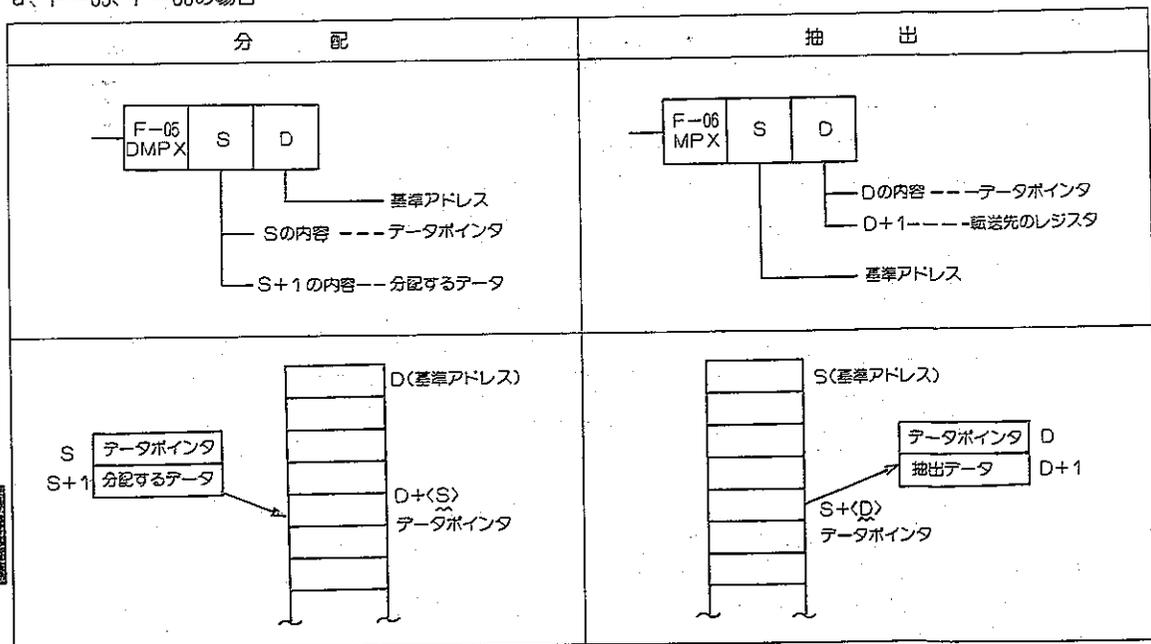
注2 ファイルレジスタ容量の設定が2Kバイトのときは、30000~33777までが使用可能です。

(1) 基準アドレスを用いる命令  
次の各命令では基準アドレスを用います。

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

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

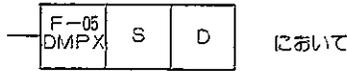
a、F-05、F-06の場合



4

◎基準アドレス

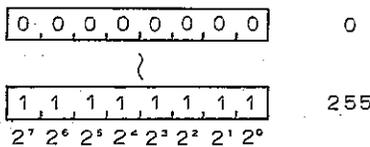
F-05のD、F-06のSが基準アドレスであり、各ブロックの先頭アドレス(コ.000, b.000, 9000, 30000, 30400, 31000, ...37400)を使用します。基準アドレスとしてブロックの先頭アドレス以外のアドレスもプログラム上設定することもできますが、PCの演算ではそのアドレスの含まれるブロックの先頭アドレスを基準アドレスとして処理します。



Dの設定	演算上の基準アドレス
コ.200	コ.000
b.110	b.000
9005	9000
30210	30000

◎データポイント

F-05のS、F-06のDの内容がデータポイントとなります。S、Dは8ビットで構成されますので、0~255の値を取り得ます。



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

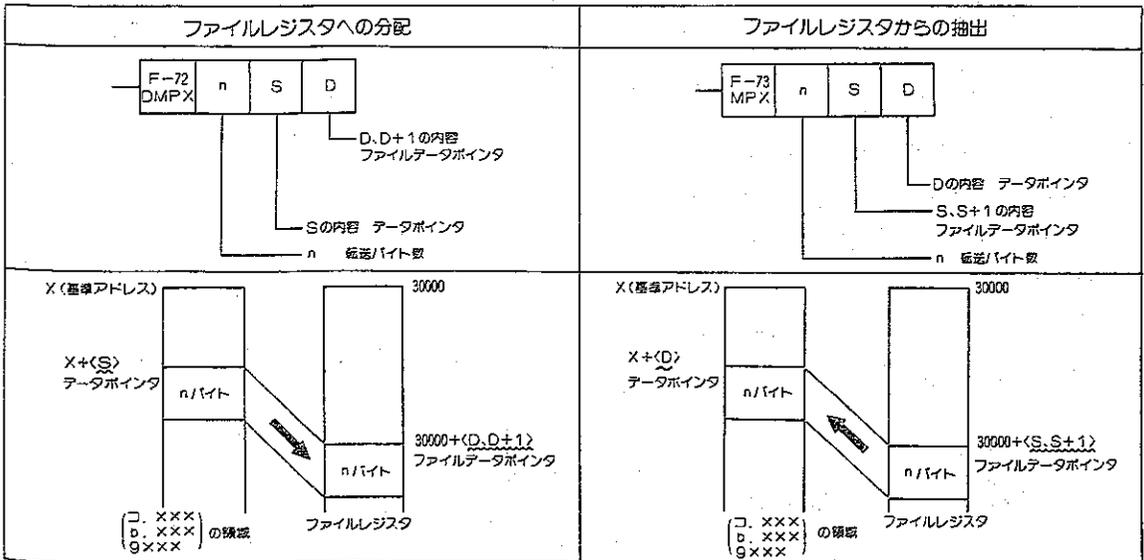
(基準アドレス+データポイント)で各ブロック内の任意のデータメモリを分配先、抽出元とすることができます。データメモリのバイトアドレスは8進数で扱いますので、データポイントの内容も8進数と見なすと、対象のレジスタのアドレスが直接判断できます。データポイントとなるレジスタの内容をF-63(INC命令)で変化させたり、外部機器(デジタルスイッチ等)で指定することにより、分配先、抽出元を変化させることができます。

基準アドレス.....9000

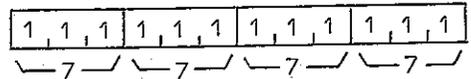
データポイントの内容... (10進では179)

9000+263=9263が分配先(抽出元)

b、F-72、F-73の場合



(10進では4095)



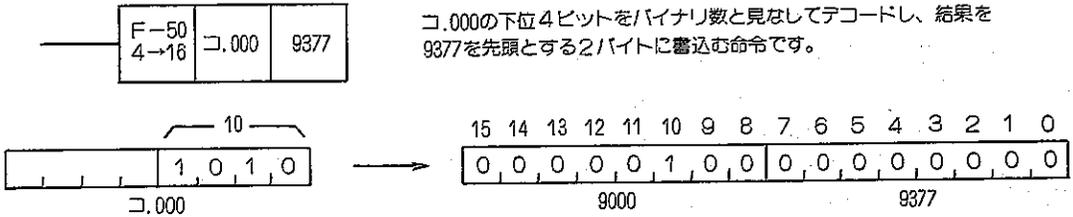
従って0~37777のファイルレジスタ全域を指示することができます。

F-72、F-73は分配先、抽出元をファイルレジスタに限定し、複数バイト一括の分配、抽出が可能な命令です。転送が開始されるアドレスは、レジスタ側では基準アドレス+データポインタで、ファイルレジスタ側は30000+ファイルデータポインタで決定されます。ファイルデータポインタは2バイトのレジスタ（16ビット）中の12ビットで示されます。

(2) テータメモリのブロックを考慮する必要がある命令  
次の各命令では、テータメモリが256バイト単位のブロックに分割されている事を考慮する必要があります。

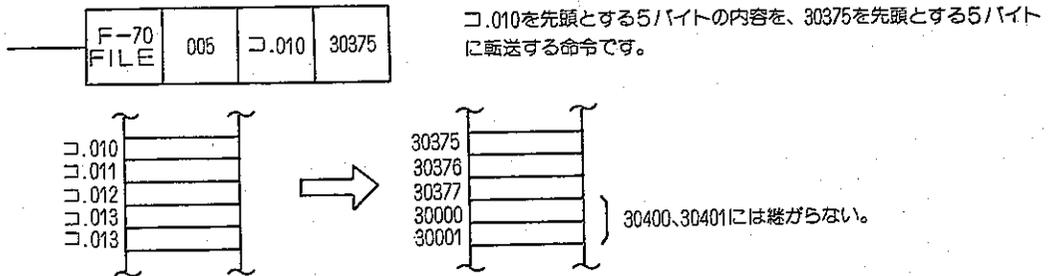
a、ソース、テストレーションが2バイト以上のテータメモリを意味する命令で、ブロックをまたがるようなアドレスをプログラムすると、次のブロックには継がらず、同一ブロックの先頭アドレスに戻ります。

(例)



b、nバイト一括転送命令(F-70)でブロックをまたがるようなアドレスをプログラムすると、次のブロックには継がらず、同一ブロックの先頭アドレスに戻ります。

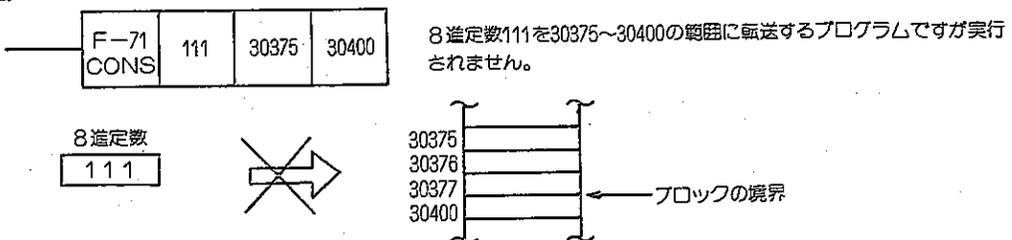
(例)



F-72、F-73でも同様の注意が必要です。

c、8進定数一括転送命令(F-71)で、ブロックをまたがるようなアドレスをプログラムすると、演算を行いません。

(例)



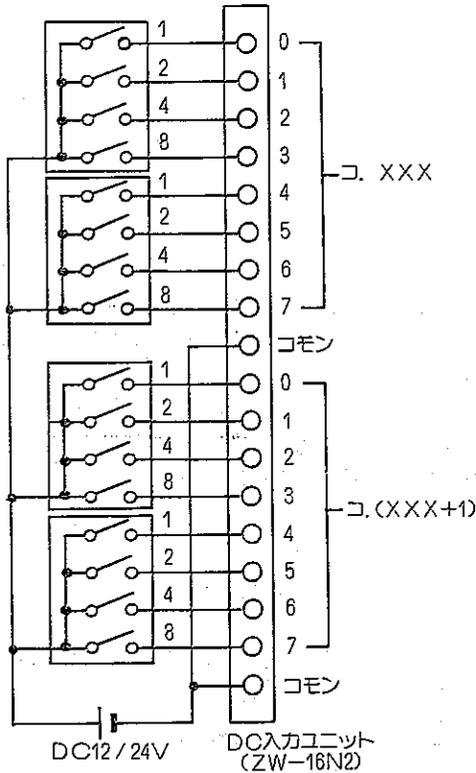
4

## (9) 数値信号の入出力方法

デジタルスイッチ等の外部機器から数値信号を読み込んでW16/W51のデータ処理命令で演算したり、演算結果を数字表示器に出力する場合の外部機器との接続例を示します。

### (1) 数値信号の入力方法

#### a. デジタルスイッチとの接続



- 入力ユニットとして、DC入力ユニット（ZW-16N2）を使用すると、1ユニット当たりBCD4桁の数値信号を読み込めます。
- デジタルスイッチとしてはリアルコードのものを使用します。

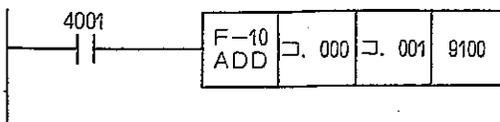
読み	数値	0	1	2	3	4	5	6	7	8	9
1			●		●		●		●		●
2				●	●			●	●		
4						●	●	●	●		
8										●	●

●印—スイッチON

**注1** コンプリメンタルコードのデジタルスイッチを使用するときは、F-09(INV命令)で反転させます。

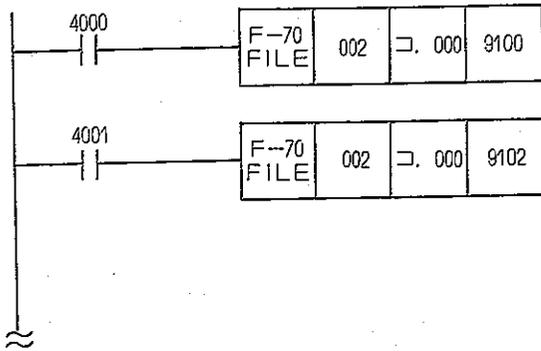
- 上記の接続で毎スキャンサイクルの入出力処理で、データメモリの入出力リレー領域に読み込まれます。16ビットのデータはコ. XXXの1バイト(8点)と、コ. (XXX+1)の1バイト(8点)としてデータ処理命令で直接指定することができます。

(例)



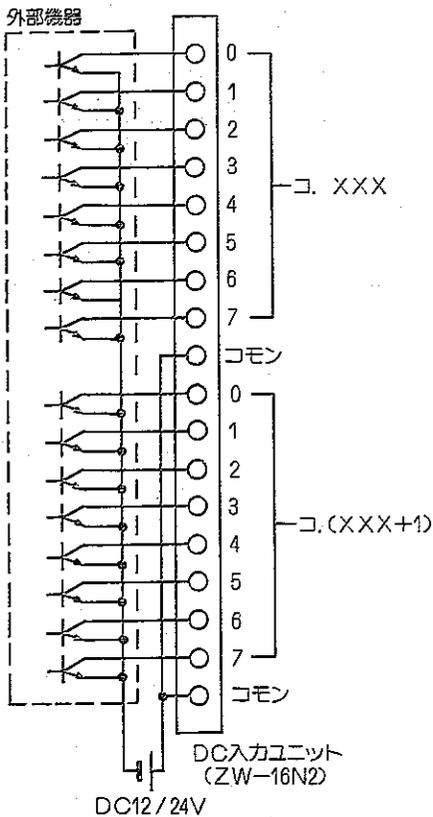
コ. 000(BCD2桁)と、コ. 001(BCD2桁)を加算し、レジスタ9100に格納。  
また、転送命令により一旦レジスタ領域に転送後、データ処理命令に使うこともできます。

(例)



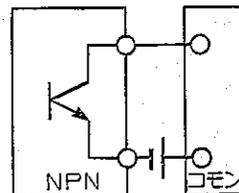
上記の例では、1組のデジタルスイッチで複数の設定値を読み込んでいます。

b. オープンコレクタ出力の外部機器との接続

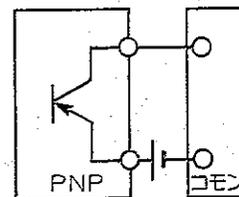


- 4000をONにすると、コ. 000, コ. 001の2バイト(BCD4桁)が9100, 9101に転送されます。
- 4001をONにすると、コ. 000, コ. 001の2バイト(BCD4桁)が9102, 9103に転送されます。

- 入力ユニットとして、DC入力ユニット (ZW-16N2) を使用すると、1ユニット当たり16ビットの数値信号が読み取れます。
- 外部機器の出力トランジスタがNPNかPNPかで接続を変更する必要があります。



NPNトランジスタ



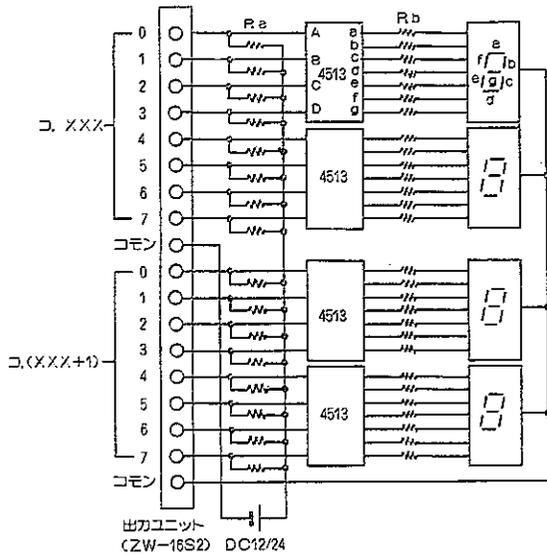
PNPトランジスタ

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



(2) 数値信号の出力方法

a. 数字表示器との接続(1)



⑥ 7セグメントLED数字表示器としては、カソードコ  
モンのものを使用します。

⑦ デコーダ・ドライバーICとしては、C-MOS  
MC4513相当品を使用します。

デコーダ・ドライバーICとしては、

$V_{DD}$  — DC12~18V

$V_{SS}$  — 0V

LE — 0V

RB1 — 0V

$\overline{BI}$  —  $V_{DD}$ と同電位

$\overline{LT}$  —  $V_{DD}$ と同電位

⑧  $R_a$ はプルアップ抵抗で5~10k $\Omega$ とします。

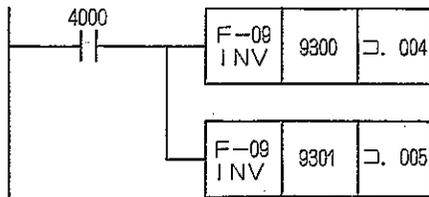
⑨  $R_b$ は電流制限抵抗で、LED数字表示器の  $I_{Fmax}$ 、 $V_F$ より算出します。

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

ただし4513の出力電流の制限から  $I_F < 25$  mAと  
してください。

⑩ 上記の表示回路は正論理で動作します。

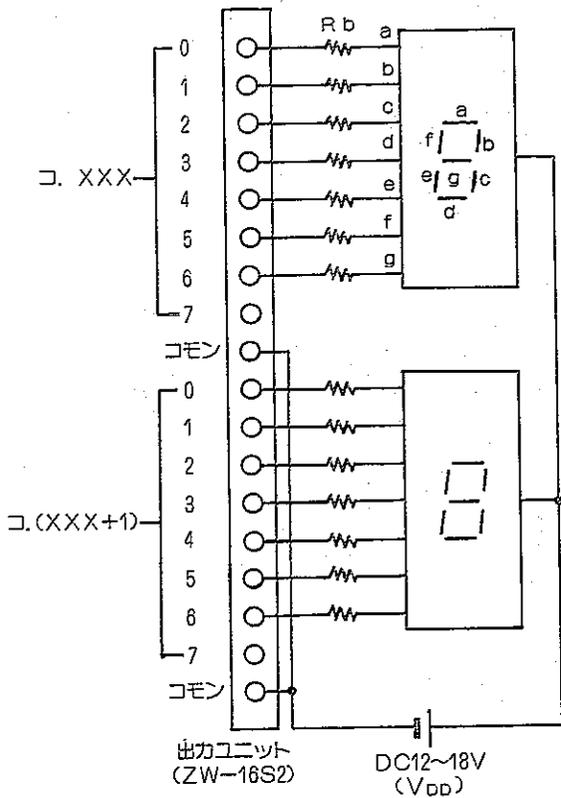
出力するデータはF-09(INV命令)で論理を反転  
してからデータメモリの入出力リレー領域に転送する  
必要があります。



⑪ レジスタ9300の内容を論理反転し、コ. 004(数  
字表示器下2桁接続)に格納

⑫ レジスタ9301の内容を論理反転し、コ. 005(数  
字表示器の上2桁接続)に格納

b. 数字表示器との接続(F-52使用)



- F-52(7SEGデコーダ命令)を使用すると、数字表示を簡単な配線で実現できます。
- 出力ユニットとしてZW-16S2を用いると2桁の数値が表示できます。
- 7セグメントLED数字表示器としてはアノードコモンのもので使用します。
- R<sub>b</sub>は電流制限抵抗で次式で算出します。

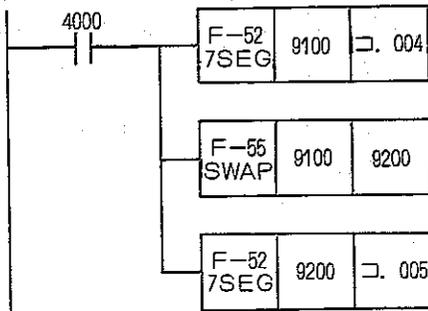
$$R_b = \frac{V_{DD} - V_F - V_{ON}}{I_{FMAX}}$$

V<sub>DD</sub>—電源電圧

V<sub>F</sub>—LED数字表示器の順電圧

V<sub>ON</sub>—出力ユニットのON電圧(1Vで計算)

- 1バイトのBCD2桁の数値を表示する場合、次の様にプログラムします。



- レジスタ9100の下位4ビット (BCD2桁のうち下位1桁) を7セグメントデータに変換し、コ. 004に出力
- レジスタ9100の上位4ビットと下位4ビットを交換し、レジスタ9200に格納
- レジスタ9200の下位4ビット (BCD2桁のうち上位1桁) を7セグメントデータに変換し、コ. 005に出力

# 4-6 応用命令の説明

## F-00 転送

### 1バイトデータの転送

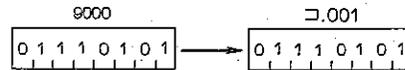
シンボル		
機能	レジスタSの内容(1バイトデータ)をレジスタDに転送する。	
演算内容	S→D	
Sの使用範囲	コ.000~コ.733 <sup>※1)</sup> b.000~b.377 9000~9377	
Dの使用範囲	コ.000~コ.733 <sup>※1)</sup> b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Sの内容	不変
	Dの内容	レジスタSの内容
	フラグ	不変

(解説)



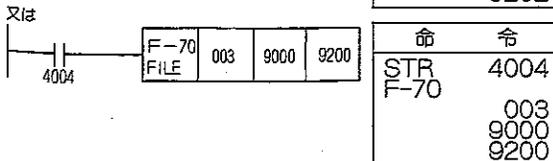
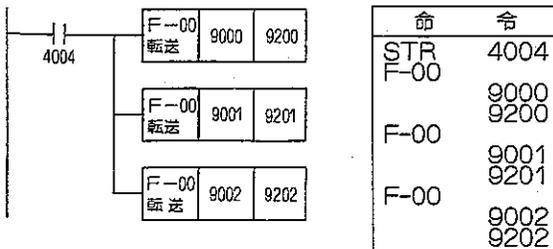
命令	
STR	4004
F-00	9000
	コ.001

入力条件4004がOFF→ONの変化時にレジスタ9000の内容をレジスタコ.001に転送します。



**注1** コ.734~コ.737は特殊領域です。(3-2(3)“キーブリレーの特殊領域”参照)

**参考** 2バイト以上同時に転送する場合は、下記のようにF-00の命令を続けるか、あるいはF-70のnバイト一括転送命令を使用して下さい。



F-00

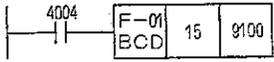
4

**F-01**  
**BCD**

**BCD定数の転送**

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>F-01</td> <td>n</td> <td>D</td> </tr> <tr> <td>BCD</td> <td></td> <td></td> </tr> </table>		F-01	n	D	BCD		
F-01	n	D						
BCD								
機能	2桁のBCD定数nをレジスタDに転送する。							
演算内容	n→D							
nの使用範囲	00~99							
Dの使用範囲	c.000~c.733 <sup>#1)</sup> b.000~b.377 9000~9377							
演算条件	入力信号の立上り(OFF→ON)							
演算後	Dの内容	n(00~99)						
	フラグ	不変						

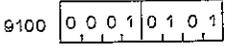
(解説)



命 令	
STR	4004
F-01	15
	9100

入力条件4004がOFF→ONの変化時に、レジスタ9100にBCD定数15を転送します。

レジスタ9100は転送時、下の数値になります。

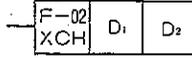


注1) c.734~c.737は特殊領域です。(3-2(3))"キーブリレーの特殊領域"参照)

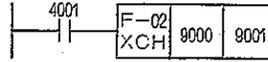


**F-02**  
**XCH**

**レジスタ間のデータ交換**  
**(eXCHange)**

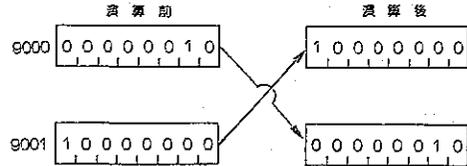
シンボル		
機能	レジスタD <sub>1</sub> の内容とレジスタD <sub>2</sub> の内容を交換する。	
演算内容	D <sub>1</sub> ← D <sub>2</sub>	
D <sub>1</sub> の使用範囲	c.000~c.733※1) b.000~b.377 9000~9377	
D <sub>2</sub> の使用範囲	c.000~c.733※1) b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	D <sub>1</sub> の内容	レジスタD <sub>2</sub> の内容
	D <sub>2</sub> の内容	レジスタD <sub>1</sub> の内容
	フラグ	不変

(解説)



命 令	
STR	4001
F-02	9000 9001

入力条件4001がOFF→ONの変化時に、レジスタ9000の内容とレジスタ9001の内容が交換されます。



注1) c.734~c.737は特殊領域です。(3-2(3)「キープレリー」の特殊領域」参照)

F-02

4

F-03  
→BIN

BCD→BINARY変換

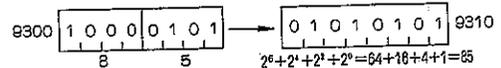
シンボル																
機能	レジスタSの内容(8ビットデータ)をBCDコードと見なしBinary(2進数)コードに変換して、レジスタDに格納する。															
演算内容	S→D															
Sの使用範囲	コ.000~コ.733※1) b.000~b.377 9000~9377															
Dの使用範囲	コ.000~コ.733※1) b.000~b.377 9000~9377															
演算条件	入力信号の立上り(OFF→ON)															
演算後	Sの内容	不変														
	Dの内容	・演算結果 ・レジスタSの内容がBCDコードでない時不変														
	フラグ	<table border="1"> <tr> <td>レジスタSの内容</td> <td>キャリー 7354</td> <td>エラー 7355</td> <td>キャリー 7356</td> <td>ゼロ 7357</td> </tr> <tr> <td>BCDコード</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>BCDコードでない時</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> </table>	レジスタSの内容	キャリー 7354	エラー 7355	キャリー 7356	ゼロ 7357	BCDコード	0	0	0	0	BCDコードでない時	0	1	0
レジスタSの内容	キャリー 7354	エラー 7355	キャリー 7356	ゼロ 7357												
BCDコード	0	0	0	0												
BCDコードでない時	0	1	0	0												

(解説)

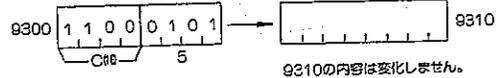


命 令	
STR	4006
F-03	9300 9310

- 入力条件4006がOFF→ONの変化時に、レジスタ9300の8ビットのデータをBCDコードと見なし、Binary(2進数)のコードに変換して、レジスタ9310に転送します。レジスタ9300の内容は不変です。9300の内容がBCDコード以外の場合9310の内容は変化せず、エラーフラグ(7355)が1になります。
- レジスタの内容とフラグの推移



キャリー 7354	エラー 7355	キャリー 7356	ゼロ 7357
0	0	0	0



キャリー 7354	エラー 7355	キャリー 7356	ゼロ 7357
0	1	0	0

注1) コ.734~コ.737は特殊領域です。(3-2(3))“キープリレーの特殊領域”参照  
 注2) フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5))“データ処理命令とフラグ”参照

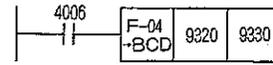
F-03

**F-04**  
→BCD

**BINARY→BCD変換**

シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-04 →BCD</td> <td style="padding: 2px;">S</td> <td style="padding: 2px;">D</td> </tr> </table>		F-04 →BCD	S	D
F-04 →BCD	S	D			
機能	レジスタSの内容(8ビットデータ)をBinary(2進数)コードと見なしBCDコードに変換してレジスタDに格納する。				
演算内容	S→D				
Sの使用範囲	コ.000~コ.733 <sup>注1)</sup> b.000~b.377 9000~9377				
Dの使用範囲	コ.000~コ.733 <sup>注1)</sup> b.000~b.377 9000~9377				
演算条件	入力信号の立上り(OFF→ON)				
演算後	Sの内容	不変			
	Dの内容	演算結果			
	フラグ	不変			

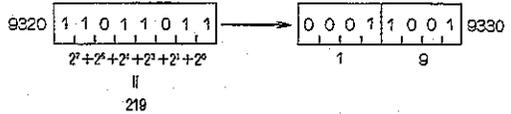
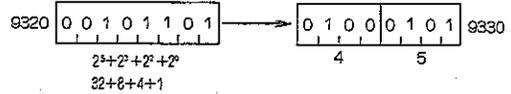
(解説)



命令	
STR	4006
F-04	9320
	9330

●入力条件4006がOFF→ONの変化時に、レジスタ9320の8ビットのデータをBinary(2進数)のコードと見なし、BCDコードに変換してレジスタ9330に転送します。レジスタ9320の内容は不変です。

変換したBCD値が100を越える場合、100以上の数値は無視されます。



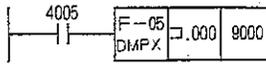
注1) コ.734~コ.737は特殊領域です。(3-2(3)キープリレーの特殊領域参照)

F-05  
DMPX

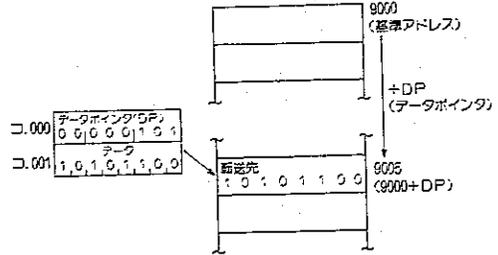
1バイトデータの分配  
(DeMultiPleXer)

シンボル		
機能	レジスタS+1の内容をレジスタD (基準アドレス)からレジスタSの内容 (データポイント)だけ変位したレジスタに転送する。	
演算内容	$S+1 \rightarrow D + \langle S \rangle$  ↳データポイント(DP) ↳基準アドレス	
Sの使用範囲	コ.000~コ.732 <sup>注1)</sup> b.000~b.377 9000~9377	
Dの使用範囲	コ.000 <sup>注1)</sup> b.000 9000 30000 30400 31000 31400 32000 32400 33000 33400 34000 34400 35000 35400 36000 36400 37000 37400	
演算条件	入力信号の立上り (OFF→ON)	
演算後	S, S+1の内容	不変
	Dの内容	不変
	D+⟨S⟩の内容	S+1のレジスタの内容
	フラグ	不変

(解説)



命 令	
STR	4005
F-05	コ.000
	9000



- 入力条件4005がOFF→ONの変化時に、以下の転送をします。  
 コ.000+1すなわちコ.001にあるデータを、基準アドレス9000からデータポイント、コ.000の内容(005)だけ変位したアドレス9005に転送します。
- データポイントは、8進数で000から377迄の値を取ります。従って、上記の例では、基準アドレスを9000とすると、データポイントを変えることにより、9000~9377の番地にデータの分配が可能です。

注1) コ.734~コ.737は特殊領域です。(3-2(3)“キープリューの特殊領域”参照)

- (1)Sをコ.733とすると、S+1がコ.734になり、特殊領域に入ってしまいます。
- (2)Dにコ.000を使用するとき、Sの内容(データポイント)が333回を越えないようにご注意ください。絶対アドレス334回はコ.734となります。

注2) 30000~37777のファイルレジスタはシステムメモリ(#205)にファイルレジスタ容量の設定をした場合に使用可能です。

3-4(2)“ファイルレジスタ”の項をご参照ください。  
 注3) D (基準アドレス) にブロックの先頭アドレス以外のアドレスもプログラム上設定できますが、PCの演算ではそのアドレスの含まれるブロックの先頭アドレスを基準アドレスとして処理されます。

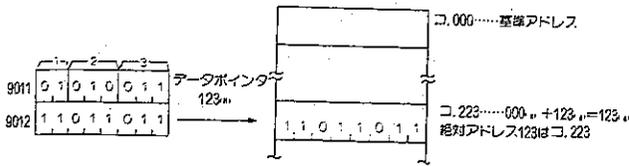
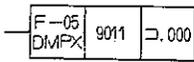
(例)

Dの設定	演算上の基準アドレス
コ.400	コ.000
b.210	b.000
9105	9000
33210	33000

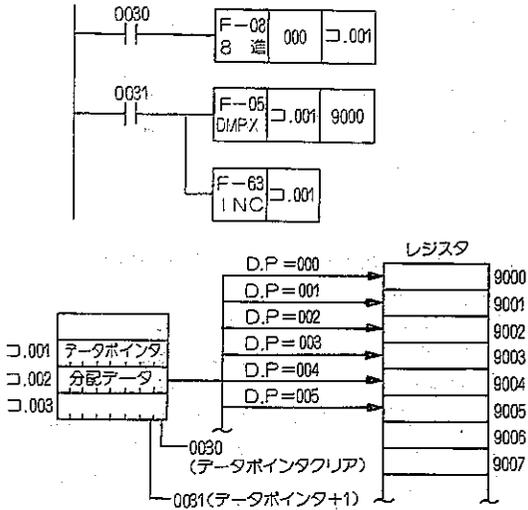
4-5(8)“データメモリのブロックと基準アドレス”をご参照ください。

注4) Dにコ.×××の領域を使用するとき、分配先のレジスタのアドレスはコ.000を基準アドレスとした絶対アドレスで決定されます。

4-5(7)“リルー領域を使用するとき”をご参照ください。



**参考** データポインタを変化させ、分配先を移動させるプログラム例



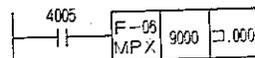
- ① 0030をOFF→ONにするとコ.001に8進定数000が転送されます。(データポインタ000)
  - ② 0031をOFF→ONにすると、コ.002の内容が(9000+000=9000)に転送されます。コ.001の内容はF-63により+1され001となります。
  - ③ 0031を再びOFF→ONにすると、コ.002の内容が(9000+001=9001)に転送されます。コ.001の内容はF-63により+1され002となります。
- 以後これと同様にして9377までのレジスタにコ.002の内容が分配されます。

F-06  
MPX

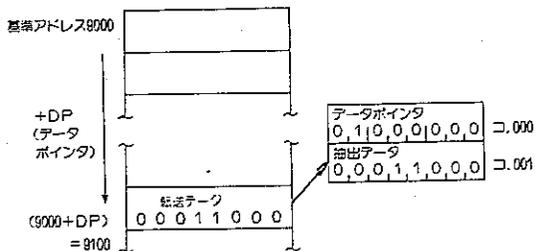
1バイトデータの抽出  
(MultiPleXer)

シンボル		
機能	レジスタS(基準アドレス)からレジスタDの内容(データポイント)だけ変位したレジスタの内容をレジスタD+1に転送する。	
演算内容	$S+(D) \rightarrow D+1$ └─データポイント(DP) └─基準アドレス	
Sの使用範囲	コ.000 b.000 9000 30000 30400 31000 31400 32000 32400 33000 33400 34000 34400 35000 35400 36000 36400 37000 37400	
Dの使用範囲	コ.000~コ.732 <sup>注1)</sup> b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Sの内容	不変
	Dの内容	不変(データポイント)
	D+1の内容	S+(D)のレジスタの内容
	フラグ	不変

(解説)



命 令	
STR	4005
F-06	9000 コ.000



- 入力条件4005がOFF→ONの変化時に、以下の転送をします。  
基準アドレス9000からコ.000のデータポイントの内容 100(8) だけ変位したアドレス9100の内容を、コ.000+1(コ.001)に転送します。
- データポイントは、8進数で000から377の値をとります。従って、上記の例では、データポイントを変えることにより、9000~9377の番地からデータの抽出が可能です。

注1) コ.734~コ.737は特殊領域です。(3-2(3)“キーブリレーの特殊領域”参照)

(1)Dをコ.733とすると、D+1はコ.734になり、特殊領域に入ってしまいます。

(2)Sにコ.000を使用するとき、Dの内容(データポイント)が333(8)を越えないようにご注意ください。絶対アドレス334(8)はコ.734となります。

注2) 30000~37777のファイルレジスタは、システムメモリ(≠205)にファイルレジスタ容量の設定した場合に使用可能です。

3-4(2)“ファイルレジスタ”の項をご参照ください。

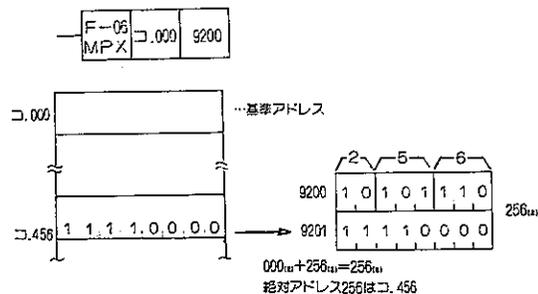
注3) S(基準アドレス)にブロックの先頭アドレス以外のアドレスもプログラム上設定できますが、PCの演算ではそのアドレスの含まれるブロックの先頭アドレスを基準アドレスとして処理されます。

(例)

Sの設定	演算上の基準アドレス
コ.211	コ.000
b.106	b.000
9023	9000
31257	31000

4-5(8)“データメモリのブロックと基準アドレス”をご参照ください。

注4) Sにコ.×××の領域を使用するとき、抽出元のレジスタのアドレスはコ.000を基準アドレスとした絶対アドレスで決定されます。



F-06

**F-07**  
**10進**

**10進定数の転送**

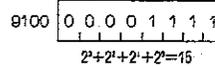
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-07</td> <td style="padding: 2px;">n</td> <td style="padding: 2px;">D</td> </tr> <tr> <td style="padding: 2px;">10進</td> <td></td> <td></td> </tr> </table>		F-07	n	D	10進		
F-07	n	D						
10進								
機能	10進定数 n をレジスタ D に転送する。							
演算内容	n → D							
n の使用範囲	000~255							
D の使用範囲	コ.000~コ.733 <sup>注1)</sup> b.000~b.377 9000~9377							
演算条件	入力信号の立上り(OFF→ON)							
演算後	Dの内容	n (000~255)						
	フラグ	不変						

(解説)



命 令	
STR	4004
F-07	015
	9100

- 入力条件4004がOFF→ONの変化時に、レジスタ9100に10進定数15を転送します。
- レジスタ9100は転送時、バイナリコードで下の数値になります。



**注1)** コ.734~コ.737は特殊領域です。(3-2(3))“キーブリレーの特殊領域”参照)

F-08  
8進

8進定数の転送

シンボル		
機能	8進定数nをレジスタDに転送する。	
演算内容	n → D	
nの使用範囲	000~377	
Dの使用範囲	コ.000~コ.733 <sup>※1)</sup> b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Dの内容	n (000~377)
	フラグ	不変

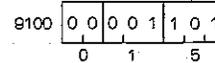
(解説)



命 令	
STR	4004
F-08	015 9100

● 入力条件4004がOFF→ONの変化時に、レジスタ9100に8進定数015を転送します。

● レジスタ9100は転送時、下の数値になります。

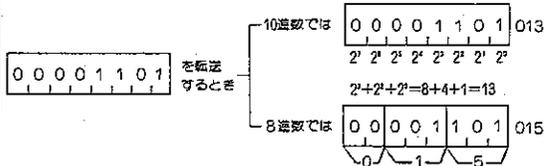


注1) コ.734~コ.737は特殊領域です。(3-2(3)“キーブリレーの特殊領域”参照)

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



F-08



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

**F-09  
INV**

**8ビットデータの反転**

シンボル		<p>(解説)</p>	<table border="1"> <thead> <tr> <th colspan="2">命 令</th> </tr> </thead> <tbody> <tr> <td>STR</td> <td>4002</td> </tr> <tr> <td>F-09</td> <td>9000</td> </tr> <tr> <td></td> <td>9003</td> </tr> </tbody> </table>	命 令		STR	4002	F-09	9000		9003
命 令											
STR	4002										
F-09	9000										
	9003										
機能	レジスタSの内容を反転してレジスタDに格納する。										
演算内容	$\bar{S} \rightarrow D$	<p>入力条件4002がOFF→ONの変化時に、レジスタ9000の8ビットの内容を反転させ、その内容をレジスタ9003に格納します。</p> <p>レジスタ9000の内容は不変です。</p>									
Sの使用範囲	コ.000~コ.733 <sup>#1)</sup> b.000~b.377 9000~9377										
Dの使用範囲	コ.000~コ.733 <sup>#1)</sup> b.000~b.377 9000~9377										
演算条件	入力信号の立上り(OFF→ON)										
演算後	Sの内容	不変									
	Dの内容	レジスタSの内容の反転データ									
	フラグ	不変									

注1) コ.734~コ.737は特殊領域です。(3-2(3)"キーブリレーの特殊領域"参照)

F-09

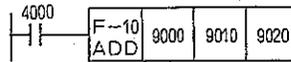
4

**F-10  
ADD**

**レジスタ間(BCD2桁)の加算  
(ADD)**

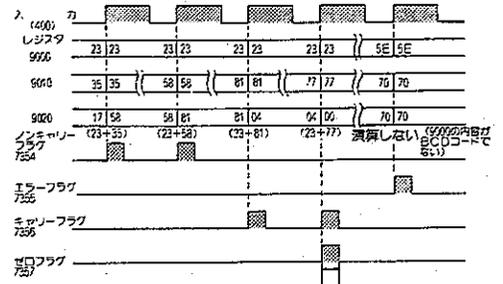
シンボル	<table border="1" style="display: inline-table;"> <tr> <td>F-10 ADD</td> <td>S<sub>1</sub></td> <td>S<sub>2</sub></td> <td>D</td> </tr> </table>				F-10 ADD	S <sub>1</sub>	S <sub>2</sub>	D
F-10 ADD	S <sub>1</sub>	S <sub>2</sub>	D					
機能	レジスタS <sub>1</sub> の内容とレジスタS <sub>2</sub> の内容を加算(BCD2桁加算)してレジスタDに格納する。							
演算内容	S <sub>1</sub> + S <sub>2</sub> → D							
S <sub>1</sub> の使用範囲	コ.000~コ.733 <sup>(注1)</sup> b.000~b.377 9000~9377							
S <sub>2</sub> の使用範囲	コ.000~コ.733 <sup>(注1)</sup> b.000~b.377 9000~9377							
Dの使用範囲	コ.000~コ.733 <sup>(注1)</sup> b.000~b.377 9000~9377							
演算条件	入力信号の立上り(OFF→ON)							
演算後	S <sub>1</sub> の内容	不変						
	S <sub>2</sub> の内容	不変						
演算後	Dの内容	●演算結果(下位2桁) ●レジスタS <sub>1</sub> 、S <sub>2</sub> の内容がBCDコードでないとき不変						
	フラグ	演算結果	キャリー 7354	エラー 7355	キャリー 7356	ゼロ 7357		
		0	1	0	0	1		
		1~99	1	0	0	0		
		100	0	0	1	1		
		101以上	0	0	1	0		
S <sub>1</sub> 、S <sub>2</sub> の内容がBCDコードでない時	0	1	0	0				

(解説)



命 令	
STR	4000
F-10	9000
	9010
	9020

- 入力条件4000がOFF→ONの変化時に、レジスタ9000の内容とレジスタ9010の内容を加算して、レジスタ9020に格納します。レジスタ9000、9010の内容は不変です。
- 演算結果とフラグの推移



12ナントタイム以内  
プログラム中、フラグに影響を与えない命令まで有効

F-10

4

注1 コ. 734~コ. 737は特殊領域です。(3-2(3) "キープリレーの特殊領域" 参照)

注2 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。

(4-5(5) "データ処理命令とフラグ" 参照)

注3 S<sub>1</sub>、S<sub>2</sub>の内容がBCDコード以外の数値の場合、エラーフラグ(7355)をONし、加算は実行しません。

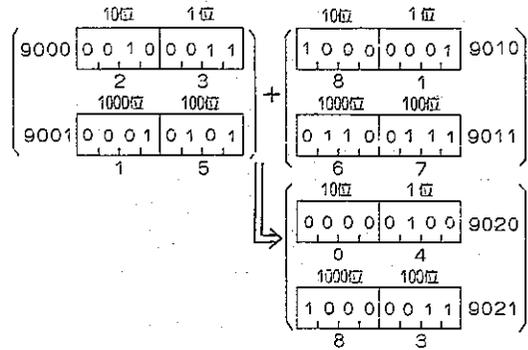
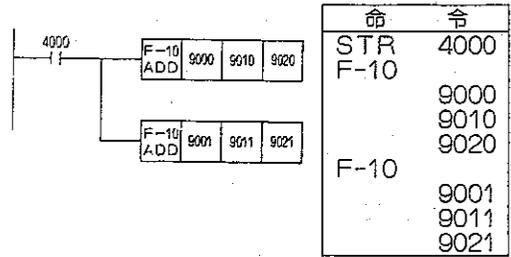
(例) S<sub>1</sub>

0	1	0	1	1	1	1	0
				E <sub>1</sub> (H)			

 1110はBCDでは禁止のコードです。

参考 BCDで3桁以上の加算をする場合、F-10 命令を続けて設定します。

連続してF-10命令を設定すると、2つ目以降のF-10命令では、キャリーフラグ(7356)の内容も加算されるようになります。STR命令に続く最初のF-10命令では、キャリーフラグ(7356)の内容は加算されません。



- 上記の演算は1523+6781=8304を示しています。
  - 下の桁から順次プログラムをしていくと、桁上げの情報が上位桁に入ってきます。
- 4-5(6) "倍長演算"の項をご参照ください。

F-10

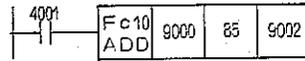
4

**Fc10  
ADD**

**レジスタとBCD定数(2桁)の加算  
(ADD)**

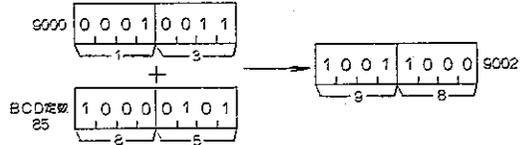
シンボル	$\overline{\text{Fc10}} \text{ ADD } S_1 \quad n \quad D$				
機能	レジスタS <sub>1</sub> の内容と2桁のBCD定数nを加算してレジスタDに格納する。				
演算内容	$S_1 + n \rightarrow D$				
S <sub>1</sub> の使用範囲	コ.000~コ.733※1) b.000~b.377 9000~9377				
nの使用範囲	00~99				
Dの使用範囲	コ.000~コ.733※1) b.000~b.377 9000~9377				
演算条件	入力信号の立上り (OFF→ON)				
演算	S <sub>1</sub> の内容	不変			
	Dの内容	<ul style="list-style-type: none"> <li>演算結果(下位2桁)</li> <li>レジスタS<sub>1</sub>の内容がBCDコードでないとき不変</li> </ul>			
後フラグ	演算結果	ノンキャリー 7354	エラー 7355	キャリー 7356	ゼロ 7357
	0	1	0	0	1
	1~99	1	0	0	0
	100	0	0	1	1
	101以上	0	0	1	0
S <sub>1</sub> 内容がBCDでない時	0	1	0	0	

(解説)



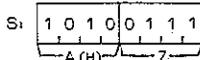
命令	
STR	4001
Fc10	9000
	85
	9002

入力条件4001がOFF→ONの変化時に、レジスタ9000の内容とBCD定数85を加算して、レジスタ9002に格納します。  
タイミング関係はF-10と同様な動きをします。



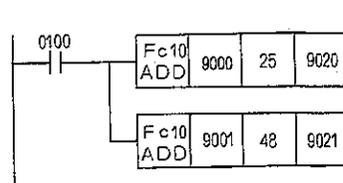
- 注1) コ.734~コ.737は特殊領域です。(3-2(3))“キーブリレーの特殊領域”参照)
- 注2) フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5))“データ処理命令とフラグ”参照)
- 注3) S<sub>1</sub>の内容がBCDコード以外の数値の場合、エラーフラグ(7355)をONし、加算は実行しません。

(例)

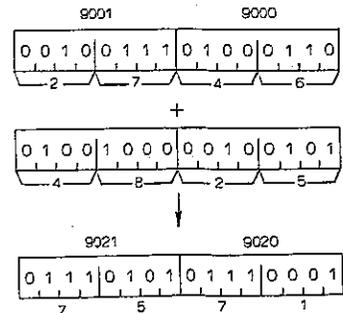


1010はBCDでは禁止のコードです。

参考) F-10と同様にBCD3桁以上の加算が可能です。



命令	
STR	0100
Fc10	9000
	25
Fc10	9020
	9001
	48
	9021



F010

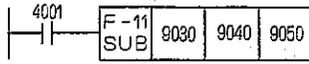
**F-11  
SUB**

**レジスタ間(BCD2桁)の減算**

(SUBtract)

シンボル						
機能	レジスタS <sub>1</sub> の内容からレジスタS <sub>2</sub> の内容を減算(BCD2桁減算)してレジスタDに格納する。					
演算内容	S <sub>1</sub> - S <sub>2</sub> → D					
S <sub>1</sub> の使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377					
S <sub>2</sub> の使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377					
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377					
演算条件	入力信号の立上り(OFF→ON)					
演算後	S <sub>1</sub> の内容	不変				
	S <sub>2</sub> の内容	不変				
	Dの内容	●演算結果 ●レジスタS <sub>1</sub> 、S <sub>2</sub> の内容がBCDコードでないとき不変				
	フラグ	演算結果	ゼロ	エラー	キャリー	ゼロ
	0	1	0	0	0	1
	1~99	1	0	0	0	0
	負の数値	0	0	1	0	0
	S <sub>1</sub> 、S <sub>2</sub> がBCDでない時	0	1	0	0	0

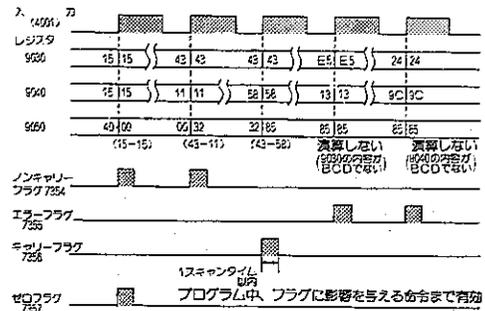
(解説)



命令	アドレス
STR	4001
F-11	9030
	9040
	9050

入力条件4001がOFF→ONの変化時に、レジスタ9030の内容から、レジスタ9040の内容を減算して、レジスタ9050に格納します。レジスタ9030、9040の内容は不変です。

演算結果とフラグの推移



F-11

注1 コ. 734~コ. 737は特殊領域です。(3-2(3) "キープリューの特殊領域"参照)

注2 (S<sub>1</sub>の内容) < (S<sub>2</sub>の内容)の演算を行うと、答は100の補数で得られます。

(例) 23-85=-62は、62の100の補数38が答となります。

(123-85=38と考えてください。)

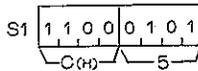
注3 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。

(4-5(5) "データ処理命令とフラグ"参照)

注4 S<sub>1</sub>、S<sub>2</sub>の内容がBCDコード以外の数値の場合、エラーフラグ(7355)をONし、減算は実行しません。

(Dの内容は不変です。)

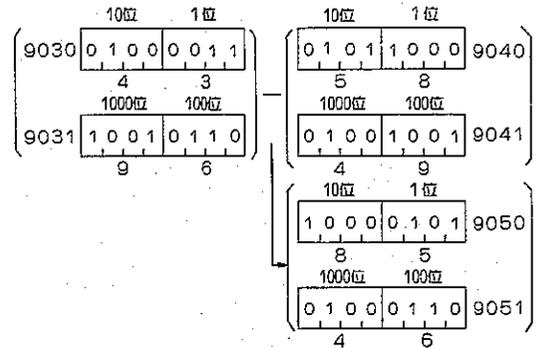
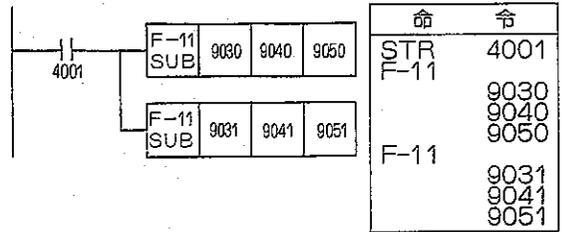
(例)



1100はBCDでは禁止のコードです。

参考 3桁以上のBCD減算する場合、F-11命令を続けて設定します。

連続して、F-11命令を設定すると、2つ目以降のF-11命令では、キャリーフラグ(7356)の内容も減算されるようになります。STR命令に続く最初のF-11命令では、キャリーフラグ(7356)の内容は減算されません。



●上記の演算は、9643-4958=4685を示しています。

●下の桁から順次プログラムをしていくと、桁下げの情報が上位桁に入ってきます。

4-5(6) "倍長演算"の項をご参照ください。

F-11

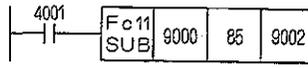
4

**Fc11  
SUB**

**レジスタとBCD定数(2桁)の減算  
(SUBtract)**

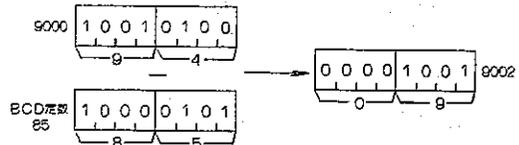
シンボル		
機能	レジスタSiの内容から2桁のBCD定数nを減算してレジスタDに格納する。	
演算内容	Si → n → D	
Siの使用範囲	コ.000~コ.733 <sup>※1)</sup> b.000~b.377 9000~9377	
nの使用範囲	00~99	
Dの使用範囲	コ.000~コ.733 <sup>※1)</sup> b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Siの内容	不変
	演算結果	ノキャリ 7354   エラー 7355   キャリ 7356   ゼロ 7357
	0	1   0   0   1
	1~99	1   0   0   0
	負の数値	0   0   1   0
SiがBCDでない時	0   1   0   0	

(解説)

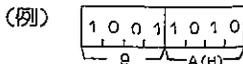


命 令	
STR	4001
Fc11	9000
	85
	9002

入力条件4001がOFF→ONの変化時に、レジスタ9000の内容からBCD定数85を減算して、レジスタ9002に格納します。  
タイミング関係はF-11と同様な動きをします。

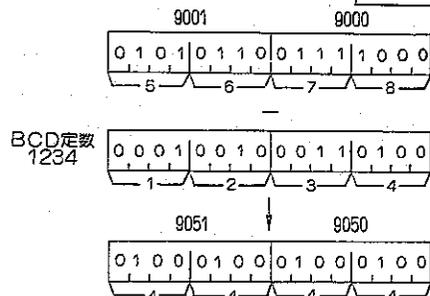
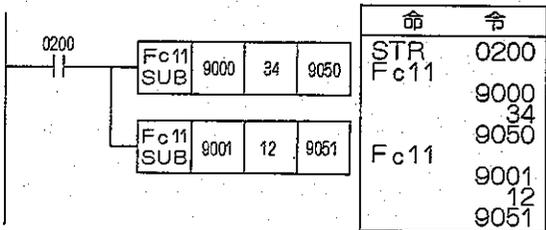


- 注1 コ.734~コ.737は特殊領域です。(3-2(3)“キープリレーの特殊領域”参照)
- 注2 (Siの内容) < nの演算を行うと、答は100の補数で得られます。  
(例) 23-85 = -62は、62の100の補数38が答となります。  
(123-85=38と考えてください。)
- 注3 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です (4-5(5)“データ処理命令とフラグ”参照)
- 注4 Siの内容がBCDコード以外の数値の場合、エラーフラグ(7355)をONし、減算は実行しません。(Dの内容は不変です)



1010はBCDでは禁止のコードです。

参考 F-11と同様にBCD3桁以上の減算が可能です。



11011

4

**F-12  
CMP**

**レジスタ間の比較  
(CoMPare)**

シンボル																						
機能	レジスタS <sub>1</sub> の内容とレジスタS <sub>2</sub> の内容を大小比較する。																					
演算内容	S <sub>1</sub> <=> S <sub>2</sub>																					
S <sub>1</sub> の使用範囲	C.000~C.733≠1) b.000~b.377 9000~9377																					
S <sub>2</sub> の使用範囲	C.000~C.733≠1) b.000~b.377 9000~9377																					
演算条件	入力信号がONの時(OFF→ONの変化時に限定されない)																					
演算後	S <sub>1</sub> の内容	不変																				
	S <sub>2</sub> の内容	不変																				
	フラグ	<table border="1"> <thead> <tr> <th>レジスタの内容</th> <th>ノキャリー 7354</th> <th>エラー 7355</th> <th>キャリー 7356</th> <th>ゼロ 7357</th> </tr> </thead> <tbody> <tr> <td>S<sub>1</sub> &gt; S<sub>2</sub></td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>S<sub>1</sub> = S<sub>2</sub></td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>S<sub>1</sub> &lt; S<sub>2</sub></td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	レジスタの内容	ノキャリー 7354	エラー 7355	キャリー 7356	ゼロ 7357	S <sub>1</sub> > S <sub>2</sub>	1	0	0	0	S <sub>1</sub> = S <sub>2</sub>	1	0	0	1	S <sub>1</sub> < S <sub>2</sub>	0	0	1	0
	レジスタの内容	ノキャリー 7354	エラー 7355	キャリー 7356	ゼロ 7357																	
S <sub>1</sub> > S <sub>2</sub>	1	0	0	0																		
S <sub>1</sub> = S <sub>2</sub>	1	0	0	1																		
S <sub>1</sub> < S <sub>2</sub>	0	0	1	0																		

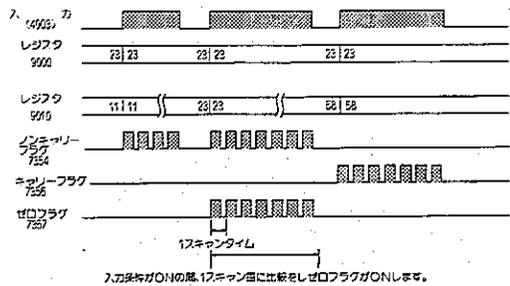
(解説)



命	令
STR	4003
F-12	9000
	9010

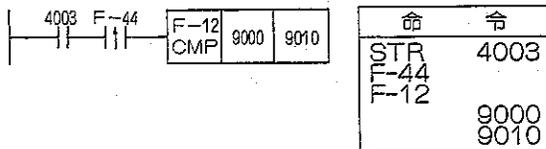
入力条件4003がONの時、レジスタ9000とレジスタ9010の内容を大小比較して、その結果をノンキャリフラグ (7354)、キャリーフラグ (7356) とゼロフラグ (7357) に設定します。この時、レジスタ9000とレジスタ9010の内容は不変です。

●レジスタの内容とフラグの推移

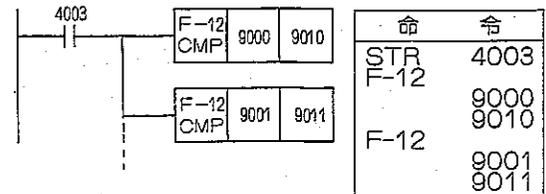


入力条件がONの毎、12チャンネル毎に比較をしゼロフラグONします。

- 注1 C.734~C.737は特殊領域です。(3-2(3)“キープリレーの特殊領域”参照)
- 注2 入力信号がONの間、毎スキャンサイクル演算が実行されます。(4-5(4)“演算実行条件”参照)
- 注3 エラーフラグ (7355) は常に“0”となります。
- 注4 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5)“データ処理命令とフラグ”参照)
- 参考 入力条件のOFF→ONの変化時にのみ、大小比較をする場合は、入力条件に微分命令を組合せて下さい。



参考 2バイト以上のデータの大小比較をする場合は、加算・減算 (F-10・F-11) の場合と同様に、下位の数値から比較するようにプログラムします。連続して、F-12命令を設定すると、2目以降のF-12命令では、キャリーフラグ (7356) の内容も比較対象に入ります。(STR命令に続く最初のF-12命令では、キャリーフラグ (7356) の内容は比較対象から除外されます。)



下の桁から、順次プログラムをしていくと、桁下げの情報が上位桁に入ってきます。  
4-5(6)“倍長演算”の項をご参照ください。

F-12

4

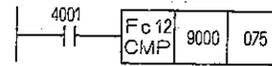
**Fc12  
CMP**

**レジスタと定数の比較**

(CoMPare)

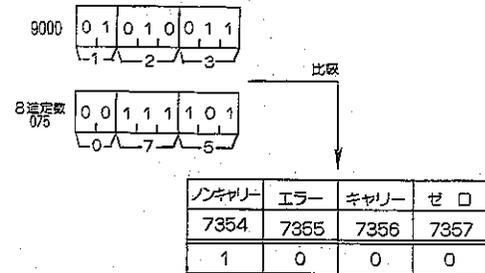
シンボル						
機能	レジスタSiの内容と8進定数nを大小比較する。					
演算内容	$S_i <=> n$					
Siの使用範囲	C.000~C.733 <sup>※1)</sup> b.000~b.377 9000~9377					
nの使用範囲	000~377					
演算条件	入力信号がONの時 (OFF→ONの変化時に限定されない)					
演算後	Siの内容	不変				
	フラグ	レジスタの内容	ノンキャリア 7354	エラー 7355	キャリア 7356	ゼロ 7357
		Si > n	1	0	0	0
		Si = n	1	0	0	1
Si < n	0	0	1	0		

(解説)



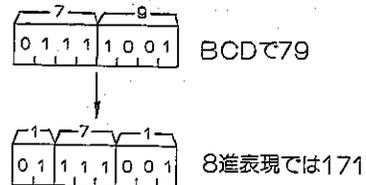
命 令	
STR	4001
Fc12	9000 075

入力条件4001がONの時に、レジスタ9000の内容と8進定数075を大小比較して、その結果をノンキャリアフラグ (7354)、エラーフラグ (7355)、ゼロフラグ (7357) に設定します。この時レジスタ9000の内容は不変です。  
タイミング関係はF-12と同様な動きをします。



- 注1 C.734~C.737は特殊領域です。(3-2(3)“キープリレーの特殊領域”参照)
- 注2 入力信号がONの間、毎スキャンサイクル演算が実行されます。(4-5(4)“演算実行条件”参照)
- 注3 エラーフラグ (7355) は常に“0”となります。
- 注4 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5)“データ処理命令とフラグ”参照)

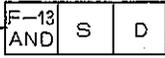
参考 Fc12はプログラムの書き込み時に8進数を用います。8進数は、あらゆるビットパターンを数値で表現することができ、面倒な重み計算も不要です。BC定数との比較をする場合、BCD定数を8進数に換し、プログラムを書き込んでください。



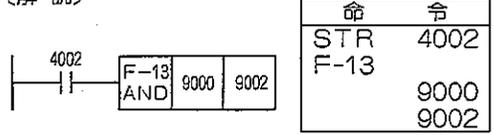
Fc12

4

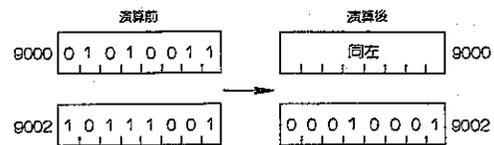
**F-13** レジスタ間の論理積  
**AND** (AND)

シンボル		
機能	レジスタSの内容(8ビットデータ)とレジスタDの内容(8ビットデータ)の論理積をとり、レジスタDに格納する。	
演算内容	S AND → D	
Sの使用範囲	コ.000 ~ コ.733(注1) b.000 ~ b.377 9000 ~ 9377	
Dの使用範囲	コ.000 ~ コ.733(注1) b.000 ~ b.377 9000 ~ 9377	
演算条件	入力信号の立上り(OFF → ON)	
演算後	Sの内容	不変
	Dの内容	演算結果
	フラグ	不変

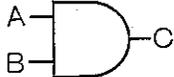
(解説)



入力条件4002がOFF → ONの変化時に、レジスタ9000の8ビットの内容とレジスタ9002の8ビットの内容の論理積(AND)をとり、レジスタの9002に格納します。レジスタ9000の内容は不変です。



ANDの真理値表

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

注1 コ.734 ~ コ.737は特殊領域です。(3-2(3) "キープリューの特殊領域"参照)

F-13

4

# Fc13 レジスタと定数の論理積 AND (AND)

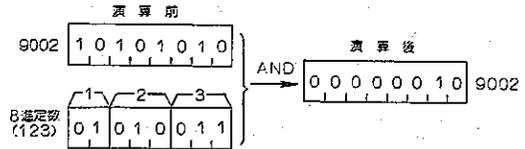
シンボル		
機能	8進定数nとレジスタDの内容の論理積をとり、レジスタDに格納する。	
演算内容	$n \cap D \rightarrow D$	
nの使用範囲	000~377	
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Sの内容	不変
	Dの内容	演算結果
	フラグ	不変

(解説)



命 令	
STR	4001
Fc13	123
	9002

入力条件4001がOFF→ONの変化時に、8進定数123とレジスタ9002の内容の論理積をとり、レジスタ9002に格納します。



ANDの真理値表

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

注1) コ.734~コ.737は特殊領域です。(3-2(3) "キープリレーの特殊領域"参照)

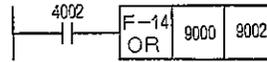
Fc13

4

# F-14 レジスタ間の論理和 (OR)

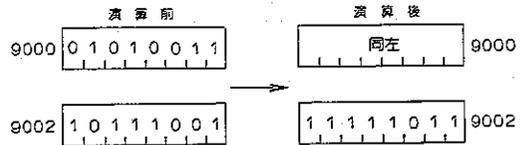
シンボル		
機能	レジスタSの内容(8ビットデータ)とレジスタDの内容(8ビットデータ)の論理和をとり、レジスタDに格納する。	
演算内容	SUD→D	
Sの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Sの内容	不変
	Dの内容	演算結果
	フラグ	不変

(解説)



命令	
STR	4002
F-14	9000 9002

入力条件4002がOFF→ONの変化時に、レジスタ9000の8ビットの内容とレジスタ9002の8ビットの内容の論理和(OR)をとり、レジスタ9002に格納します。レジスタ9000の内容は不変です。



ORの真理値表

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

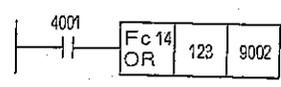
注1 コ.734~コ.737は特殊領域です。(3-2(3) "キープルーの特殊領域"参照)

F-14

**Fc14** レジスタと定数の論理和  
**OR** (OR)

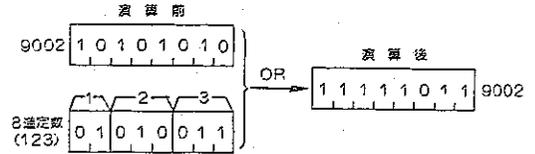
シンボル		
機能	8進定数nとレジスタDの内容の論理和をとりレジスタDに格納する。	
演算内容	nUD→D	
nの使用範囲	000~377	
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Sの内容	不変
	Dの内容	演算結果
	フラグ	不変

(解説)



命 令	
STR	4001
Fc14	123
	9002

入力条件4001がOFF→ONの変化時に、8進定数123とレジスタ9002の内容の論理和(OR)をとり、レジスタ9002に格納します。



ORの真理値表

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

注1 コ.734~コ.737は特殊領域です。(3-2(3) "キープリレーの特殊領域"参照)

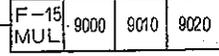
Fc14

**F-15  
MUL**

**レジスタ(4桁)間の乗算  
(MULTiPLY)**

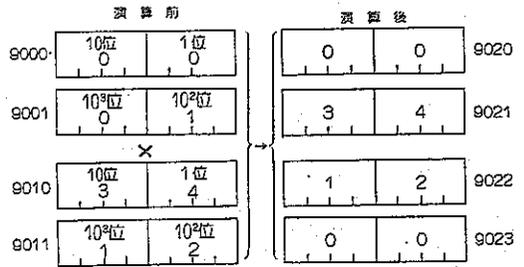
シンボル					
機能	レジスタS <sub>1</sub> 、S <sub>1</sub> +1の内容(BCD4桁)とレジスタS <sub>2</sub> 、S <sub>2</sub> +1の内容(BCD4桁)を乗算してレジスタDから4バイトに格納する。				
演算内容	$(S_1, S_1+1) \times (S_2, S_2+1) \rightarrow D, D+1, D+2, D+3$				
S <sub>1</sub> の使用範囲	コ.000~コ.732(注1) b.000~b.377 9000~9377				
S <sub>2</sub> の使用範囲	コ.000~コ.732(注1) b.000~b.377 9000~9377				
Dの使用範囲	コ.000~コ.730(注1) b.000~b.377 9000~9377				
演算条件	W16	入力信号の立上り(OFF→ON)			
	W51	特殊リレー7350がOFFの時 入力信号の立上り(OFF→ON)			
演算後	S <sub>1</sub> 、S <sub>1</sub> +1の内容	不変			
	S <sub>2</sub> 、S <sub>2</sub> +1の内容	不変			
	Dの内容	演算結果 (1の位と10の位)		レジスタS <sub>1</sub> 、 S <sub>1</sub> +1、S <sub>2</sub> 、 S <sub>2</sub> +1の内容 がBCDコードでない時不 変	
	D+1の内容	演算結果 (10 <sup>2</sup> の位と10 <sup>3</sup> の位)			
	D+2の内容	演算結果 (10 <sup>4</sup> の位と10 <sup>5</sup> の位)			
D+3の内容	演算結果 (10 <sup>6</sup> の位と10 <sup>7</sup> の位)				
フラグ	レジスタS <sub>1</sub> 、S <sub>1</sub> +1 S <sub>2</sub> 、S <sub>2</sub> +1の内容	ノンキャリア 7354	エラー 7355	キャリア 7356	ゼロ 7357
	BCDコード	0	0	0	0
BCDコードでない時	1				

(解説)



命令	
STR	4001
F-15	
	9000
	9010
	9020

入力条件4001がOFF→ONの変化時に、レジスタ9000、9001のBCD4桁とレジスタ9010、9011のBCD4桁を乗算をして、レジスタ9020からの4バイトに格納します。



- 上記の演算は100×1234=123400を示しています。
- W51の場合、特殊リレー7350がOFFのとき、本演算を実行します。7350がONのときは、レジスタ(8桁)間の乗算となります。次頁をご参照ください。

F-15 (4桁)

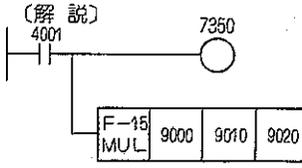
注1) コ.734~コ.737は特殊領域です。(3-2(3)“キープレーの特殊領域”参照)  
注2) フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5)“データ処理命令とフラグ”参照)

注3) S<sub>1</sub>、S<sub>1</sub>+1、S<sub>2</sub>、S<sub>2</sub>+1の内容がBCDコード以外の数値の場合、エラーフラグ(7355)をONし、乗算を実行しません。(D~D+3の内容は不変です。)

**F-15**  
**MUL**

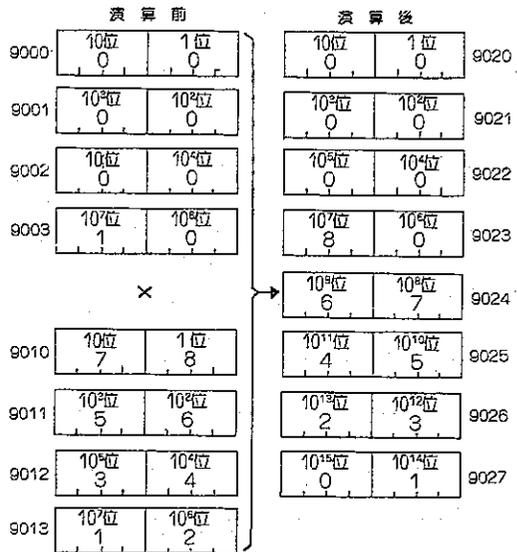
**レジスタ(8桁)間の乗算 (W51のみ)**  
**(MULTiply)**

シンボル	F-15 MUL S <sub>1</sub> S <sub>2</sub> D			
機能	レジスタS <sub>1</sub> ~S <sub>1</sub> +3の内容(BCD8桁)とレジスタS <sub>2</sub> ~S <sub>2</sub> +3の内容(BCD8桁)を乗算してレジスタDから8バイトに格納する。			
演算内容	(S <sub>1</sub> , S <sub>1</sub> +1, S <sub>1</sub> +2, S <sub>1</sub> +3) × (S <sub>2</sub> , S <sub>2</sub> +1, S <sub>2</sub> +2, S <sub>2</sub> +3) → D, D+1, D+2, D+3, D+4, D+5, D+6, D+7			
S <sub>1</sub> の使用範囲	コ.000~コ.730 (注2) b.000~b.377 9000~9377			
S <sub>2</sub> の使用範囲	コ.000~コ.730 (注2) b.000~b.377 9000~9377			
Dの使用範囲	コ.000~コ.724 (注2) b.000~b.377 9000~9377			
演算条件	特殊リレー7350がONの時、 入力信号の立上り (OFF→ON)			
演算後	S <sub>1</sub> ~S <sub>1</sub> +3の内容	不変		
	S <sub>2</sub> ~S <sub>2</sub> +3の内容	不変		
	Dの内容	演算結果 (1の位と10の位)		
	D+1の内容	演算結果 (10 <sup>2</sup> の位と10 <sup>3</sup> の位)		
	D+2の内容	演算結果 (10 <sup>4</sup> の位と10 <sup>5</sup> の位)		
	D+3の内容	演算結果 (10 <sup>6</sup> の位と10 <sup>7</sup> の位)		
	D+4の内容	演算結果 (10 <sup>8</sup> の位と10 <sup>9</sup> の位)		
	D+5の内容	演算結果 (10 <sup>10</sup> の位と10 <sup>11</sup> の位)		
	D+6の内容	演算結果 (10 <sup>12</sup> の位と10 <sup>13</sup> の位)		
	D+7の内容	演算結果 (10 <sup>14</sup> の位と10 <sup>15</sup> の位)		
フラグ	レジスタS <sub>1</sub> ~S <sub>1</sub> +3 S <sub>2</sub> ~S <sub>2</sub> +3の内容	キャリー 7354	エラー 7355	ゼロ 7357
	BCDコード	0	0	0
	BCDコードでない時	0	1	0



命 令	
STR	4001
OUT	7350
F-15	9000
	9010
	9020

入力条件4001がOFF→ONの変化時に、レジスタ9000~9003のBCD8桁とレジスタ9010~9013のBCD8桁を乗算して、レジスタ9020からの8バイトに格納します。



●上記の演算は10000000×12345678=1234567800000000を示します。

処理時間(最大の場合)

実行時	2203 μs
非実行時	62 μs

- 注1 W16では本演算機能はありません。
- 注2 コ.734~コ.737は特殊領域です。(3-2(3)“キープリレーの特殊領域”参照)
- 注3 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5)“データ処理命令とフラグ”参照)

- 注4 S<sub>1</sub>~S<sub>1</sub>+3, S<sub>2</sub>~S<sub>2</sub>+3の内容がBCDコード以外の数値の場合、エラーフラグ(7355)をONし、乗算を実行しません。(D~D+7の内容は不変です。)

F-15 (8桁)

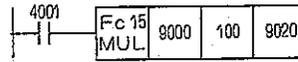
4

**Fc 15**  
**MUL**

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

シンボル	Fc15 MUL S <sub>i</sub> n D				
機能	レジスタS <sub>i</sub> 、S <sub>i</sub> +1の内容(BCD4桁)と3桁のBCD定数nを乗算してレジスタDからの4バイトに格納する。				
演算内容	(S <sub>i</sub> 、S <sub>i</sub> +1)×n → D、D+1、D+2、D+3				
S <sub>i</sub> の使用範囲	コ.000~コ.732 (注1) b.000~b.377 9000~9377				
nの使用範囲	000~999				
Dの使用範囲	コ.000~コ.730 (注1) b.000~b.377 9000~9377				
演算条件	W16	入力信号の立上り(OFF→ON)			
	W51	特殊リレー7350がOFFの時、 入力信号の立上り(OFF→ON)			
演算後	S <sub>i</sub> 、S <sub>i</sub> +1の内容	不変			
	Dの内容	演算結果 (1の位と10の位)		レジスタS <sub>i</sub> 、 S <sub>i</sub> +1の内容 がBCDコードでない時不変	
	D+1の内容	演算結果 (100の位と1,000の位)			
	D+2の内容	演算結果(10,000の位 と100,000の位)			
	D+3の内容	演算結果(1,000,000位 と10,000,000の位)			
フラグ	レジスタS <sub>i</sub> 、 S <sub>i</sub> +1の内容	リレー 7354	エラー 7355	キャリー 7356	ゼロ 7357
	BCDコード	0	0	0	0
	BCDコードで ない時	0	1		

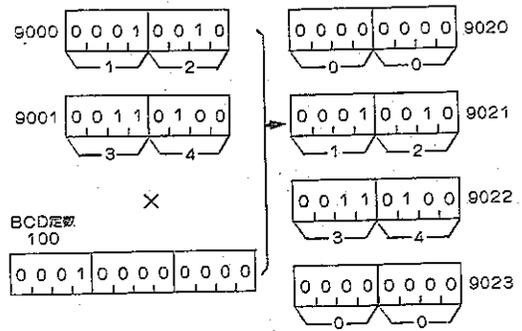
(解説)



命 令	
STR	4001
Fc15	9000
	100
	9020

入力条件4001がOFF→ONの変化時に、レジスタ9000、9001のBCD4桁とBCD定数100(3桁)の乗算をして、レジスタ9020から4バイトに格納します。

3412 × 100 = 341200



● W51の場合、特殊リレー7350がOFFのとき、本演算を実行します。7350がONのときは、レジスタ(8桁)とBCD定数(3桁)の乗算となります。次項をご参照ください。

Fc15 (4桁)

- 注1 コ.734~コ.737は特殊領域です。(3-2(3)“キープリレーの特殊領域”参照)
- 注2 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5)“データ処理命令とフラグ”参照)

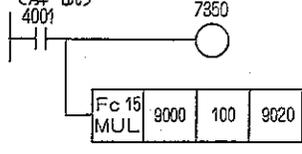
- 注3 S<sub>i</sub>、S<sub>i</sub>+1の内容がBCDコード以外の場合、エラーフラグ(7355)をONし、乗算を実行しません。(D~D+3の内容は不変です。)

# Fc15 MUL

## レジスタ(8桁)とBCD定数(3桁)の乗算(W51のみ) (MULTiPLY)

シンボル	Fc15 MUL S <sub>i</sub> n D				
機能	レジスタS <sub>i</sub> ~S <sub>i</sub> +3の内容(BCD8桁)と3桁のBCD定数nを乗算してレジスタDからの6バイトに格納する。				
演算内容	(S <sub>i</sub> , S <sub>i</sub> +1, S <sub>i</sub> +2, S <sub>i</sub> +3) × n → D, D+1, D+2, D+3, D+4, D+5				
S <sub>i</sub> の使用範囲	コ. 000~コ. 730(注2) b. 000~b. 377 9000~9377				
nの使用範囲	000~999				
Dの使用範囲	コ. 000~コ. 726(注2) b. 000~b. 377 9000~9377				
演算条件	特殊リレー7350がONの時、 入力信号の立上り(OFF→ON)				
演算後	S <sub>i</sub> ~S <sub>i</sub> +3の内容	不変			
	Dの内容	演算結果 (1の位と10の位)		レジスタS <sub>i</sub> ~ S <sub>i</sub> +3の内容 がBCDコードでない時不 変	
	D+1の内容	演算結果 (10 <sup>2</sup> の位と10 <sup>3</sup> の位)			
	D+2の内容	演算結果 (10 <sup>4</sup> の位と10 <sup>5</sup> の位)			
	D+3の内容	演算結果 (10 <sup>6</sup> の位と10 <sup>7</sup> の位)			
	D+4の内容	演算結果 (10 <sup>8</sup> の位と10 <sup>9</sup> の位)			
	D+5の内容	演算結果 (10 <sup>10</sup> の位と10 <sup>11</sup> の位)			
フラグ	レジスタS <sub>i</sub> ~ S <sub>i</sub> +3の内容	リギット 7354	エラー 7355		
	BCDコード	0	0	0	0
BCDコードでない時	1				

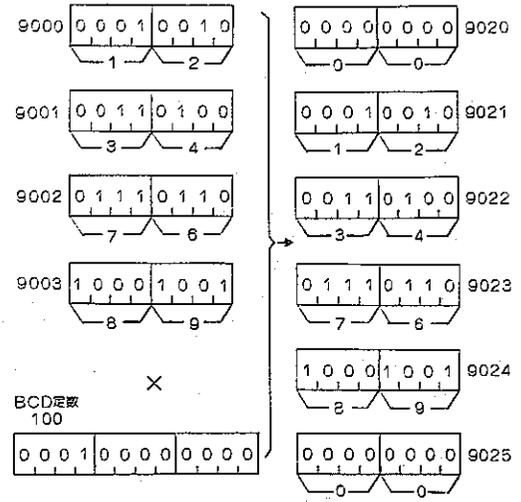
(解説)



命 令	
STR	4001
OUT	7350
Fc15	9000
	100
	9020

入力条件4001がOFF→ONの変化時に、レジスタ9000~9003のBCD8桁とBCD定数100(3桁)の乗算をして、レジスタ9020から6バイトに格納します。

$$89763412 \times 100 = 8976341200$$



処理時間(最大の場合)

実行時	1113 μs
非実行時	62 μs

- 注1 W16では本演算機能はありません。
- 注2 コ. 734~コ. 737は特殊領域です。(3-2(3)“キーブリレーの特殊領域”参照)
- 注3 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5)“データ処理命令とフラグ”参照)

- 注4 S<sub>i</sub>~S<sub>i</sub>+3の内容がBCDコード以外の場合、エラーフラグ(7355)をONし、乗算を実行しません。(D~D+5の内容は不変です。)

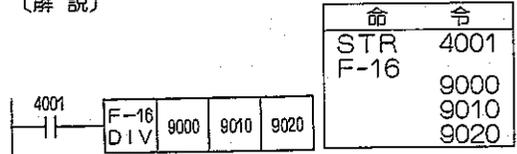
Fc15 (8桁)

**F-16**  
**DIV**

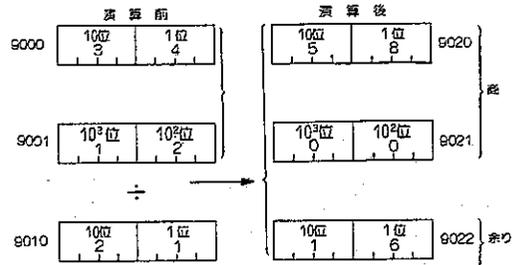
**レジスタ(4桁)とレジスタ(2桁)の除算**  
(DiVide)

シンボル						
機能	レジスタS <sub>1</sub> 、S <sub>1</sub> +1の内容(BCD4桁)をレジスタS <sub>2</sub> の内容(BCD2桁)で除算し、レジスタDからの2バイトに商を3バイト目に余を格納する。					
演算内容	$(S_1, S_1+1) \div S_2$ → D、D+1、D+2					
S <sub>1</sub> の使用範囲	コ.000~コ.732 (注1) b.000~b.377 9000~9377					
S <sub>2</sub> の使用範囲	コ.000~コ.733 (注1) b.000~b.377 9000~9377					
Dの使用範囲	コ.000~コ.731 (注1) b.000~b.377 9000~9377					
演算条件	W16	入力信号の立上り(OFF→ON)				
	W51	特殊リレー7350がOFFの時、入力信号の立上り(OFF→ON)				
演算	S <sub>1</sub> 、S <sub>1</sub> +1の内容	不変				
	S <sub>2</sub> の内容	不変				
	Dの内容	演算結果の商 (1の位と10の位)	レジスタS <sub>1</sub> 、S <sub>1</sub> +1、S <sub>2</sub> の内容がBCDコードでない時、S <sub>2</sub> の内容が00の時不変			
	D+1の内容	演算結果の商 (10 <sup>2</sup> の位と10 <sup>3</sup> の位)				
D+2の内容	演算結果の余					
後	フラグ	レジスタS <sub>1</sub> 、S <sub>1</sub> +1、S <sub>2</sub> の内容	ノキャリ 7354	エラー 7355	キャリ 7356	ゼロ 7357
		BCDコード	0	0	0	0
		●BCDコードでない時 ●S <sub>2</sub> の内容が00の時	0	1	0	0

(解説)



入力条件4001がOFF→ONの変化時に、レジスタ9000、9001のBCD4桁をレジスタ9010のBCD2桁で除算をし、レジスタ9020からの2バイトに商を入れ、3バイト目に余りを入れます。



- 上記の演算は1234÷21=58余り16を示しています。
- W51の場合、特殊リレー7350がOFFのとき、本演算を実行します。7350がONのときは、レジスタ(8桁)とレジスタ(8桁)の除算となります。次項をご参照ください。

F-16 (4桁)

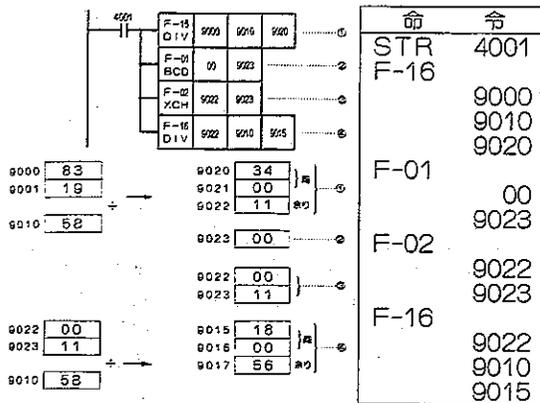
4

- 注1 コ.734~コ.737は特殊領域です。(3-2(3)“キープリレーの特殊領域”参照)
- 注2 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5)“データ処理命令とフラグ”参照)
- 注3 S<sub>1</sub>、S<sub>1</sub>+1、S<sub>2</sub>の内容がBCDコード以外の数値の

- 場合、エラーフラグ(7355)をONし、除算を実行しません。(D、D+1、D+2の内容は不変です。)
- 注4 分子<分母(S<sub>1</sub><S<sub>2</sub>、S<sub>1</sub>+1=0)の時、演算結果の商(D、D+1の内容)は0となり、余り(D+2の内容)は、分子(S<sub>1</sub>の内容)となります。例えば20÷30を実行すると、商は0余り20となります。

【参考】 小数点以下2桁を求めるときは次のようなプログラムを組むと求められます。

例  $1983 \div 58 = 34.18$  余り0.56



- ① 入力条件4001がOFF→ONのとき、レジスタ9000、9001の内容をレジスタ9010の内容で除算し結果は9020、9021に商が格納され、9022に余りが格納されます。
- ② 9023に00のデータを入れ
- ③ 9022と9023の内容を交換し、余りを、千、百の位に変換します。
- ④ ③のデータを再度9010の内容で除算し、9015、9016に商を9017に余りを格納します。  
9015に格納されたデータが小数点以下の2桁になります。

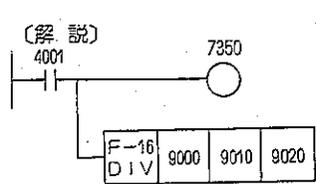
F-16

4

# F-16 MUL

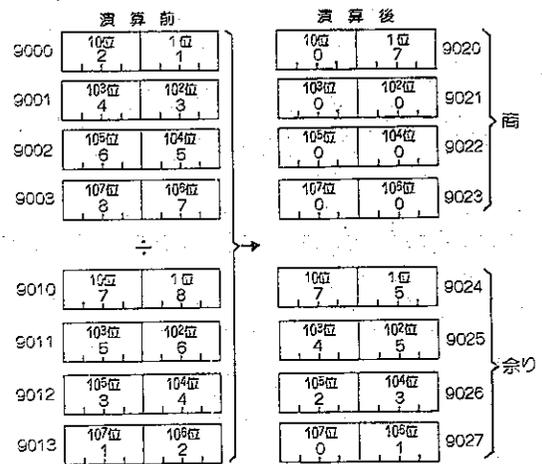
## レジスタ(8桁)とレジスタ(8桁)の除算 (W51のみ) (Divide)

シンボル	F-16 DIV S <sub>1</sub> S <sub>2</sub> D				
機能	レジスタS <sub>1</sub> ~S <sub>1</sub> +3の内容(BCD8桁)をレジスタS <sub>2</sub> ~S <sub>2</sub> +3の内容(BCD8桁)で除算し、レジスタDからの4バイトに商を5バイト目からの4バイトに余りを格納する。				
演算内容	$(S_1, S_1+1, S_1+2, S_1+3) \div (S_2, S_2+1, S_2+2, S_2+3)$ $\rightarrow D, D+1, D+2, D+3, D+4, D+5, D+6, D+7$				
S <sub>1</sub> の使用範囲	コ.000~コ.730 (注2) b.000~b.377 9000~9377				
S <sub>2</sub> の使用範囲	コ.000~コ.730 (注2) b.000~b.377 9000~9377				
Dの使用範囲	コ.000~コ.724 (注2) b.000~b.377 9000~9377				
演算条件	特殊リレー7350がONの時、入力信号の立上り(OFF→ON)				
演算後	S <sub>1</sub> ~S <sub>1</sub> +3の内容	不変			
	S <sub>2</sub> ~S <sub>2</sub> +3の内容	不変			
	Dの内容	演算結果の商 (1の位と10の位)			
	D+1の内容	演算結果の商 (10 <sup>2</sup> の位と10 <sup>3</sup> の位)			
	D+2の内容	演算結果の商 (10 <sup>4</sup> の位と10 <sup>5</sup> の位)			
	D+3の内容	演算結果の商 (10 <sup>6</sup> の位と10 <sup>7</sup> の位)			
	D+4の内容	演算結果の余 (1の位と10の位)			
	D+5の内容	演算結果の余 (10 <sup>2</sup> の位と10 <sup>3</sup> の位)			
D+6の内容	演算結果の余 (10 <sup>4</sup> の位と10 <sup>5</sup> の位)				
D+7の内容	演算結果の余 (10 <sup>6</sup> の位と10 <sup>7</sup> の位)				
フラグ	レジスタS <sub>1</sub> ~S <sub>1</sub> +3 S <sub>2</sub> ~S <sub>2</sub> +3の内容	ナンバー 7354	エラー 7355	キャリー 7356	ゼロ 7357
	BCDコード	0	0	0	0
	・BCDコードでない時 ・S <sub>1</sub> ~S <sub>1</sub> +3の内容が0の時	0	1	0	0



命令	
STR	4001
OUT	7350
F-16	9000
	9010
	9020

入力条件4001がOFF→ONの変化時に、レジスタ9000~9003のBCD8桁をレジスタ9010~9013のBCD8桁で除算をし、レジスタ9020からの4バイトに商を入れ、5バイト目からの4バイトに余りを入れます。



●上記の演算は87654321÷12345678=7余り1234575を示しています。

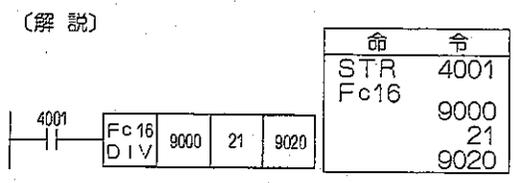
処理時間(最大の場合)

実行時	15977μs
非実行時	62μs

- 注1 W16では本演算機能はありません。
- 注2 コ.734~コ.737は特殊領域です。(3-2(3)"キープリレーの特殊領域"参照)
- 注3 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5)"データ処理命令とフラグ"参照)
- 注4 S<sub>1</sub>~S<sub>1</sub>+3、S<sub>2</sub>~S<sub>2</sub>+3の内容がBCDコード以外の数値の場合、エラーフラグ(7355)をONし、除算を実行しません。(D~D+7の内容は不変です。)
- 注5 分子<分母の時、演算結果の商(D~D+3の内容)は0となり、余り(D+4~D+7内容)は、分子(S<sub>1</sub>~S<sub>1</sub>+3の内容)となります。例えば2000÷3000を実行すると、商は0余り2000となります。

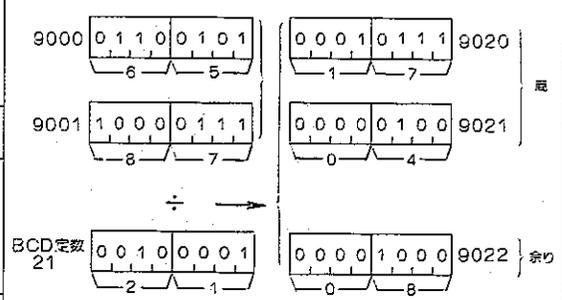
# Fc16 DIV レジスタ(4桁)とBCD定数(2桁)の除算 (DiVide)

シンボル	Fc16 DIV S <sub>i</sub> n D				
機能	レジスタS <sub>i</sub> 、S <sub>i</sub> +1の内容(BCD4桁)を2桁のBCD定数nで除算し、レジスタDから2バイトに商を3バイト目に余を格納する。				
演算内容	$(S_i, S_{i+1}) \div n \rightarrow D, D+1, D+2$				
S <sub>i</sub> の使用範囲	コ.000~コ.732(注1) b.000~b.377 9000~9377				
nの使用範囲	00~99				
Dの使用範囲	コ.000~コ.731(注1) b.000~b.377 9000~9377				
演算条件	入力信号の立上り(OFF→ON)				
演算後	S <sub>i</sub> 、S <sub>i</sub> +1の内容	不変			
	Dの内容	演算結果の商 (1の位と10の位)		レジスタS <sub>i</sub> 、S <sub>i</sub> +1の内容がBCDコードでない時、nが00の時不変	
	D+1の内容	演算結果の商 (100の位と1,000の位)			
	D+2の内容	演算結果の余			
フラグ	レジスタS <sub>i</sub> 、S <sub>i</sub> +1、nの内容	ノキャリー 7354	エラー 7355	キャリー 7356	ゼロ 7357
	BCDコード	0	0	0	0
	●BCDコードでない時 ●nが00の時	0	1	0	0



入力条件4001がOFF→ONの変化時に、レジスタ9000、9001のBCD4桁をBCD定数21で除算し、レジスタ9020から2バイトに商を入れ、3バイト目に余りを入れます。

$$8765 \div 21 = 417 \dots 8$$



- 注1 コ.734~コ.737は特殊領域です。(3-2(3)「キープリレーの特殊領域」参照)
- 注2 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5)「データ処理命令とフラグ」参照)
- 注3 S<sub>i</sub>、S<sub>i</sub>+1の内容がBCDコード以外の数値の場合、エラーフラグ(7355)をONし、除算を実行しません。(D、D+1、D+2の内容は不変です。)

- 注4 分子<分母(S<sub>i</sub><n、S<sub>i</sub>+1=0)の時、演算結果の商(D、D+1の内容)は0となり、余り(D+2の内容)は、分子(S<sub>i</sub>の内容)となります。例えば、20÷30を実行すると、答は0、余り20となります。

91011

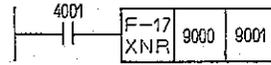
4

**F-17**  
**XNR**

**レジスタ間的一致**  
(eXclusive NoR)

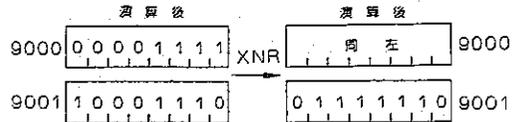
シンボル		
機能	レジスタSの内容とレジスタDの内容の否定排他的論理和をとりレジスタDに格納する。	
演算内容	$S \oplus D \rightarrow D$	
Sの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Sの内容	不変
	Dの内容	演算結果
	フラグ	不変

(解説)



命令	
STR	4001
F-17	9000
	9001

入力条件4001がOFF→ONの変化時に、レジスタ9000の内容とレジスタ9001の内容の否定排他的論理和(exclusive NOR)をとり、レジスタ9001に格納します。  
レジスタ9000の内容は不変です。



9000と9001で一致したビット(0と0、1と1)は1に、不一致のビット(0と1)は0になります。

Exclusive NOR 真理値表

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

注1) コ.734~コ.737は特殊領域です。(3-2(3)“キーアレーの特殊領域”参照)

F-17

**Fc17** レジスタと定数の一致  
**XNR** (eXclusive NoR)

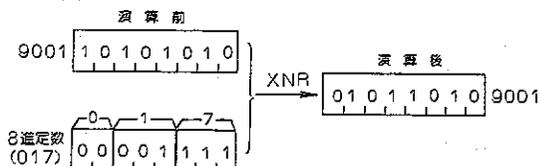
シンボル		
機能	8進定数nとレジスタDの内容の否定排他的論理和をとり、レジスタDに格納する。	
演算内容	$n \oplus D \rightarrow D$	
nの使用範囲	000~377	
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Dの内容	演算結果
	フラグ	不変

(解説)



命 令	
STR	4001
Fc17	017
	9001

入力条件4001がOFF→ONの変化時に、8進定数017とレジスタ9001の内容の否定排他的論理和(exclusive NOR)をとり、レジスタ9001に格納します。



Exclusive NOR 真理値表

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

注1) コ.734~コ.737は特殊領域です。(3-2(3)“キープリレーの特殊領域”参照)

Fc17

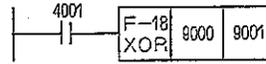
4

**F-18  
XOR**

**レジスタ間の排他的論理和  
(eXclusive OR)**

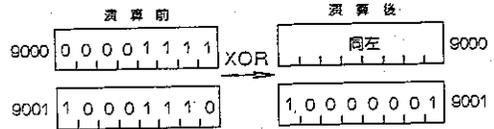
シンボル		
機能	レジスタSの内容とレジスタDの内容の排他的論理和をとり、レジスタDに格納する。	
演算内容	$S \oplus D \rightarrow D$	
Sの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Sの内容	不変
	Dの内容	演算結果
	フラグ	不変

(解説)



命 令	
STR	4001
F-18	9000 9001

入力条件4001がOFF→ONの変化時に、レジスタ9000の内容とレジスタ9001の内容の排他的論理和(exclusive OR)をとり、レジスタ9001に格納します。  
レジスタ9000の内容は不変です。



9000と9001で不一致のビット(0と1)は1に、一致のビット(0と0、1と1)は0になります。

Exclusive OR 真理値表

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

注1 コ.734~コ.737は特殊領域です。(3-2(3) "キープリレーの特殊領域"参照)

F-18

4

**Fc 18  
XOR**

**レジスタと定数の排他的論理和  
(eXclusive OR)**

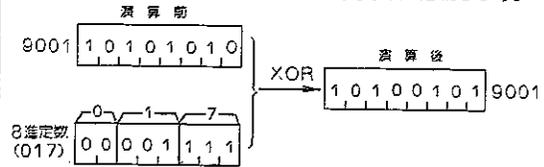
シンボル		
機能	8進定数nとレジスタDの内容の排他的論理和をとりレジスタDに格納する。	
演算内容	$n \oplus D \rightarrow D$	
nの使用範囲	000~377	
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Dの内容	演算結果
	フラグ	不変

〔解説〕



命 令	
STR	4001
Fc18	017 9001

入力条件4001がOFF→ONの変化時に、8進定数017とレジスタ9001の内容の排他的論理和(exclusive OR)をとり、レジスタ9001に格納します。



Exclusive OR 真理値表

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

〔注1〕 コ.734~コ.737は特殊領域です。(3-2(3)“キープリレーの特殊領域”参照)

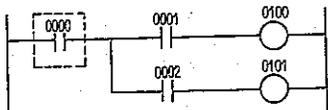
F-30  
MCS

マスターコントロールセット  
(Master Control Set)

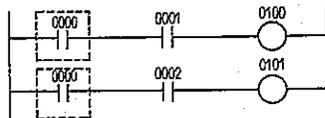
F-31  
MCR

マスターコントロールリセット  
(Master Control Reset)

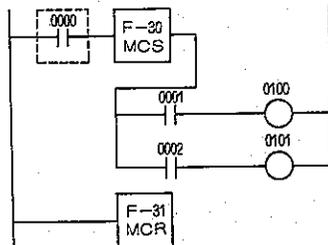
MCS、MCRは、共通演算条件以後の回路が複数の出力に分岐している場合に使用します。



(1) リレー盤の場合



(2) MCS、MCRを使わない場合



(3) MCS MCRを使用した場合

MCS →	STR	0000
	F-30	
	STR	0001
	OUT	0100
	STR	0002
	OUT	0101
MCR →	F-31	

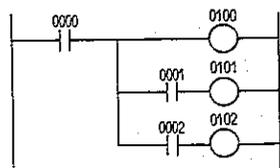
F-30 (MCS) を使用するとそれまでのACC (アキュムレータ) の内容が、CPU内部のレジスタに記憶され、F-31 (MCR) までの各命令の演算はCPU内部レジスタの内容とANDされたものとなります。F-31 (MCR) は、このANDする範

囲の終了を意味します。

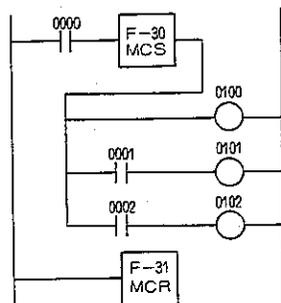
□内の共通演算条件が複雑な場合や、共通演算条件に続く演算の分岐が多い場合、プログラムを簡略できます。

注1 F-30 (MCS) で派生した母線に、直接OUT、TMR、CNTの各命令及び応用命令を接続しないでください。

(1) リレー盤の場合

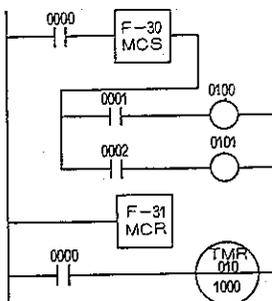
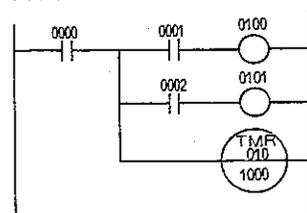


(2) MCS、MCRで禁止のプログラム

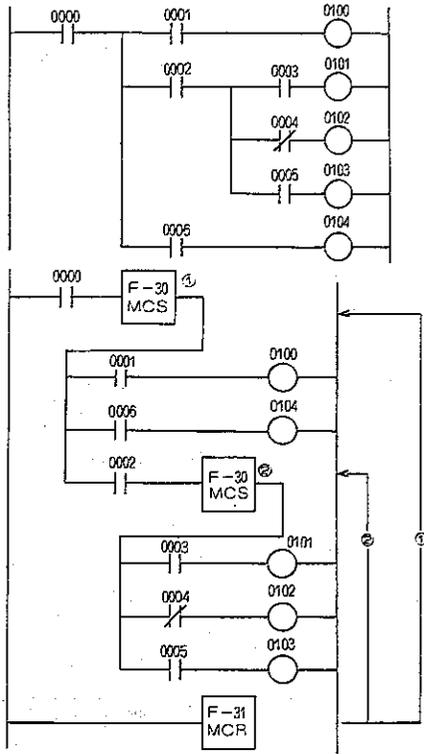


命 令	
STR	0000
F-30	
OUT	0100
STR	0001
OUT	0101
STR	0002
OUT	0102
F-31	

次のようにプログラムする必要があります。



MCS、MCRの間にさらにMCSを使用できます。

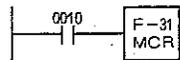


左図のリレー盤のラダー図はMCS、MCRを用いて次のようにプログラムできます。ただし本例のようにプログラム順を入れ替える必要がある場合があります。(※印部)

命	令
STR	0000
F-30	
STR	0001
OUT	0100
STR	0006
OUT	0104
STR	0002
F-30	
STR	0003
OUT	0101
STR NOT	0004
OUT	0102
STR	0005
OUT	0103
F-31	

● F-31 (MCR)はそれ以前のF-30(MCS)…左図の場合①、②…の終了を意味します。

注2 F-31 (MCR)は無条件命令です。

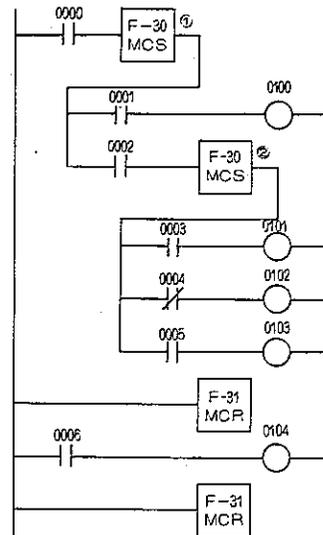


のようなプログラムはできません。

注3 次のようにプログラムすると、所期の回路にはなりません。

	命	令
	STR	0000
MCS→	F-30	
	STR	0001
	OUT	0100
MCS→	F-30	
	STR	0002
	STR	0003
	OUT	0101
	STR NOT	0004
	OUT	0102
	STR	0005
	OUT	0103
MCR→	F-31	
	STR	0006
	OUT	0104
MCR→	F-31	

このMCRは無意味なものです。  
このMCRで①、②のMCSは終了しています。

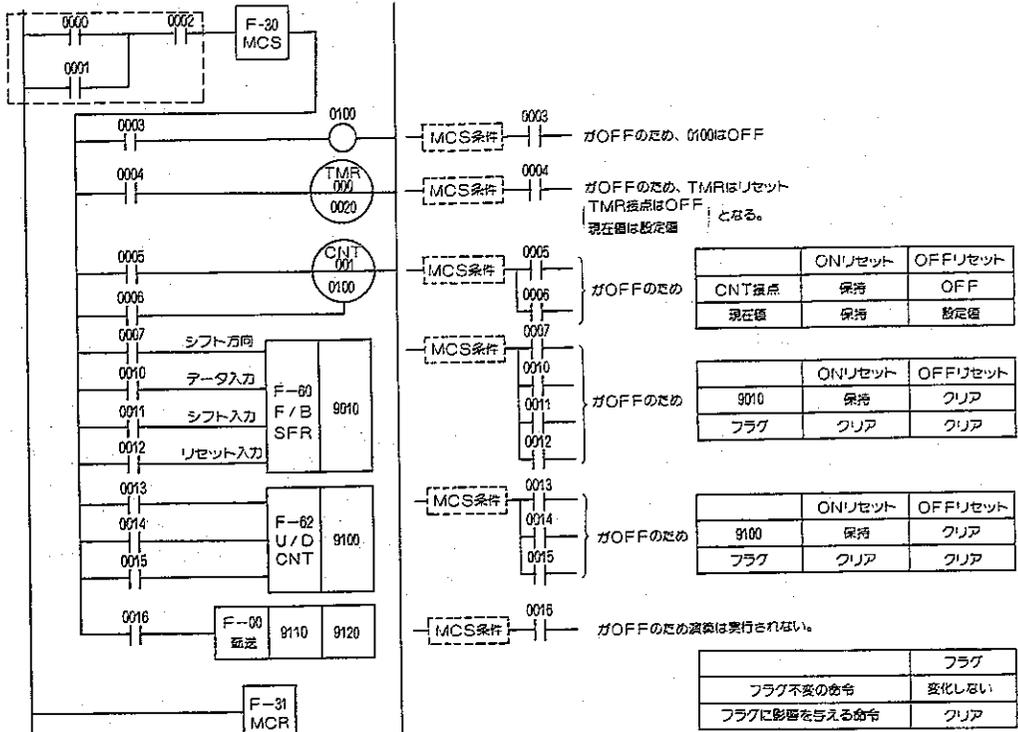


05  
000  
1.1  
11.1

4

注4) MCS、MCR \*の間にMCSを何度でも使用でき  
ますが、すべてのMCSの範囲は、\*のMCRで  
終了します。

MCSの条件(点線内)がOFFのとき、MCSとM  
CRの間にある命令は次のように処理されます。



注5) CNT、F-60、F-62の各命令はシステム  
メモリ#202でリセット条件をONリセット、  
OFFリセットのいずれかに設定できます。  
OFFリセットの場合、MCSによりリセットさ  
れます。

**F-40  
END**

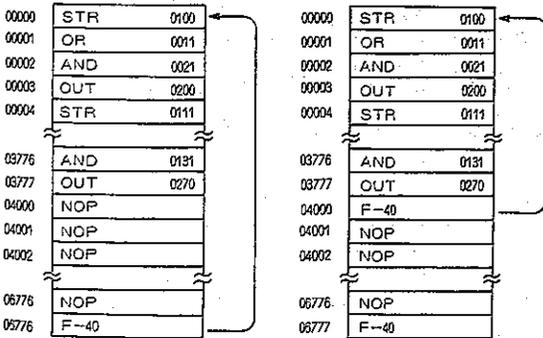
**エンド命令  
(END)**

F-40(エンド命令)はプログラムの終了を意味します。F-40は優先度が一番高い命令であり、F-41(JCS)とF-42(JCR)の間にあってもEND命令は実行され、ユーザプログラム処理を終了します。

END命令はプログラムメモリ容量の設定後メモリをクリアすると、プログラムメモリの最終アドレスに自動的に書込まれますので、次のような場合を除き特に書込む必要はありません。

(1) スキャンタイムを早くする場合

スキャンタイムは(入出力処理時間+ユーザプログラム処理時間)となります。ユーザプログラム処理時間はプログラムアドレス00000からEND命令までの全命令の処理時間の合計です。メモリのクリアで自動的に書込まれるEND命令の位置は、たとえばプログラムメモリ容量の設定が3.5K語の場合06777(3584語目)となります。設計完了したラダー図をプログラムで書込んだとき、その最終アドレスがたとえば03777(2048語目)であったとすると、04000~06776まではNOP命令、06777にEND命令が存在し、このNOP命令の処理時間(1語当り1.21μs)を空費することになります。少しでも演算時間を早くする必要があるとき、04000にF-40を書込むと以下のNOP命令を処理することなくユーザプログラムの演算を終了し、次のスキャンサイクルに移ります。



(a)メモリクリアによるEND(6777)のみ (b)04000にF-40(END)を書込み

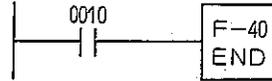
(2) 試運転でプログラムを部分的に実行させる場合

シーケンス動作の区切毎にF-40を挿入する事でプログラムを部分的に実行させ、OKであればF-40を削除します。

**注1** ゼロクロス同期を使っているときは、ゼロクロス同期によって決まるスキャンタイムになります。

**注2** メモリを追加したり(1)、(2)でEND命令を書込むと、F-40が複数個存在することがあります。このような場合、最初のF-40でユーザプログラムの演算を終了します。本運転の前にF-40の位置を検索して確認してください。

**注3** F-40(END)は無条件命令です。



のようなプログラムはできません。

F-40

4

F-41  
JCS

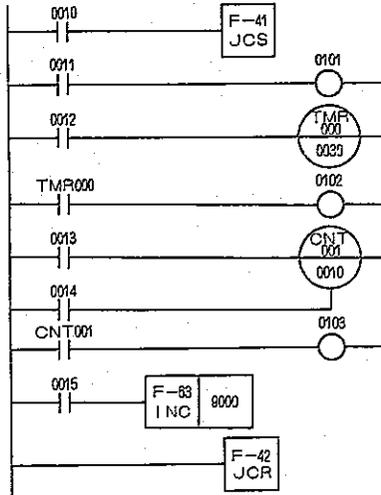
ジャンプコントロールセット  
(Jump Control Set)

F-42  
JCR

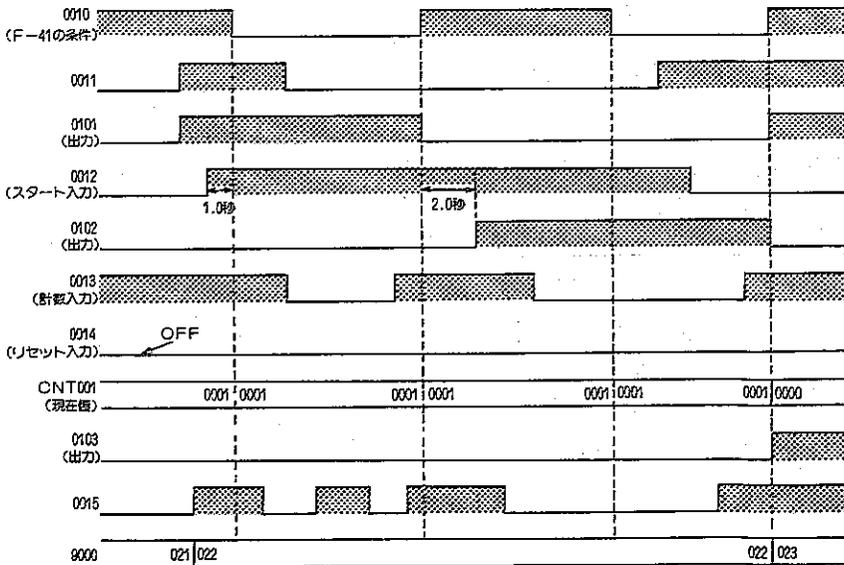
ジャンプコントロールリセット  
(Jump Control Reset)

F-41 (JCS)の条件がOFFの時、F-42 (JCR) までにあるEND命令を除くすべての命令を実行しません。したがってOUT命令、TMR・CNT・MD

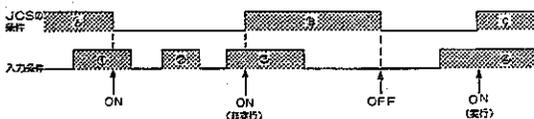
命令、応用命令等の演算結果をデータメモリに書き込む命令があってもデータメモリの内容は変化せず、JCSの条件がONの時の状態を保持します。



命 令	
STR	0010
F-41	
STR	0011
OUT	0101
STR	0012
TMR	000
	0030
STR TMR	000
OUT	0102
STR	0013
STR	0014
CNT	001
	0010
STR CNT	001
OUT	0103
STR	0015
F-63	
	8000
F-42	



注1 TMRの内部クロック(0.1秒クロック)、CNTの計数入力および応用命令の入力条件(入力条件のOFF→ONで演算を実行するもの)と、F-41(JCS)の条件のON/OFFのタイミングにご注意ください。



- ①の立上りでは、JCSの条件ONのため、演算を実行します。
- ②の立上りでは、JCSの条件OFFのため、演算を実行しません。
- ③の立上りでは、JCSの条件OFFのため、演算を実行しません。
- ④がONの間にJCSの条件がONとなりますが、④のJCSの条件がON→OFFとなるときの入力条件がON、⑤のJCSの条件がOFF→

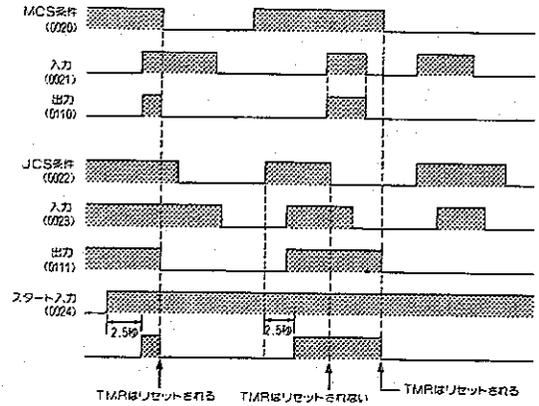
ONとなる時の入力条件もONであるため、入力条件がOFF→ONに変化したとは見なさず演算を実行しません。

④の立上りでは、JCSの条件がOFFのため、演算を実行しません。

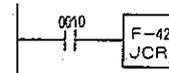
④がONの間にJCSの条件がONとなります。

⑤のJCSの条件がON→OFFとなる時の入力条件がOFF、④のJCSの条件がOFF→ONとなる時の入力条件はONと変化しているため、⑤のJCS条件がOFF→ONとなった直後に演算が実行されます。

- 注2** F-41(JCS)とF-42(JCR)の間に、フラグに影響を与える命令があるとき、JCSの条件がOFFであれば、演算は実行しませんが、7354~7357の各フラグはクリアされます。
- 注3** F-41(JCS)とF-42(JCR)の間に、F-40(END命令)があるとき、JCSの条件のON/OFFにかかわらずEND命令は実行され、ユーザープログラムの演算を終了し、次のスキャンサイクルに移ります。
- 注4** F-30(MCS)とF-31(MCR)の間に、F-41(JCS)、F-42(JCR)を入れ子構造でプログラムすることができません。ただし、MCSはJCSより優先度が高いため、MCSの条件がOFFになると、MCSの条件がOFFのときの状態(出力リレー等はOFF、TMRはリセット等)になります。

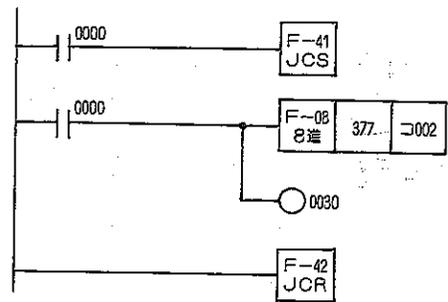
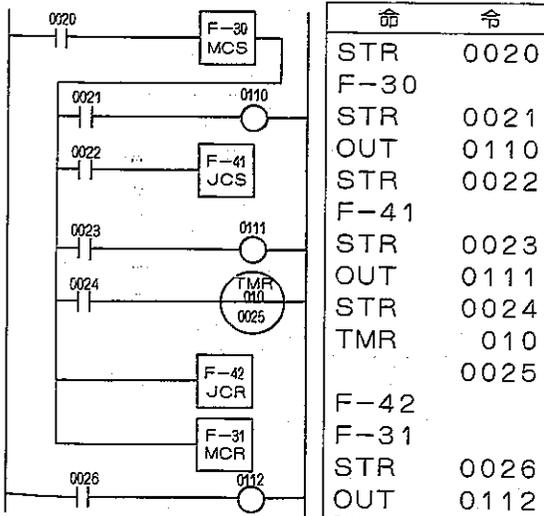


- 注5** F-41(JCS)とF-42(JCR)の間に、さらにF-41、F-42を入れることはできません。このようなプログラムを書込むと、プログラムチェックの際、プログラマ(ZW-101PG1)では“JCS ERROR”と表示されます。
- 注6** F-42(JCR)は無条件命令です。



のようなプログラムはできません。

- 注7** F-41(JCS)とF-42(JCR)の間に、作るプログラムの入力条件は、F-41(JCR)の入力条件と同じものを使わないで下さい。2回目から入力条件が、OFF→ONに変化したと見なさず、演算を実行しません。



のようなプログラムを作らないでください。

F-41  
F-42

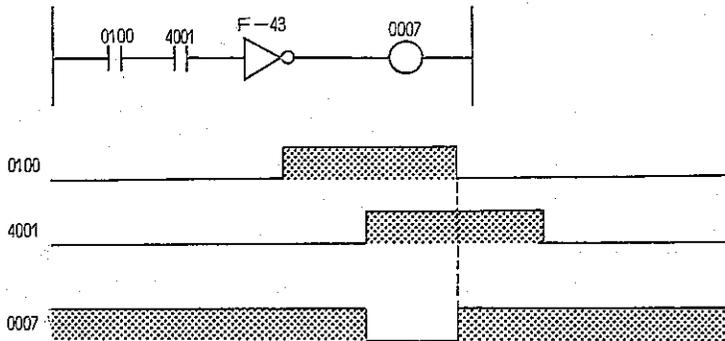
4

F-43  
CPL

ビット反転  
(ComPLement)

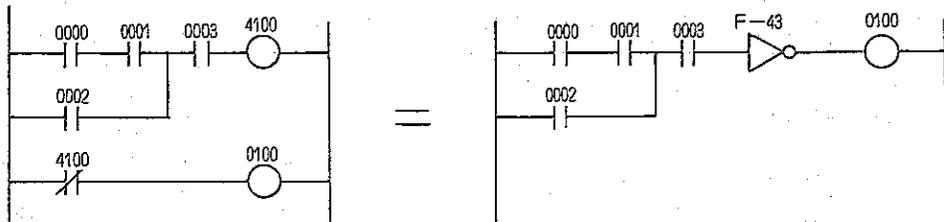
F-43は直前のACC(アキュムレータ)の内容を反転する命令です。

命	令
STR	0100
AND	4001
F-43	
OUT	0007



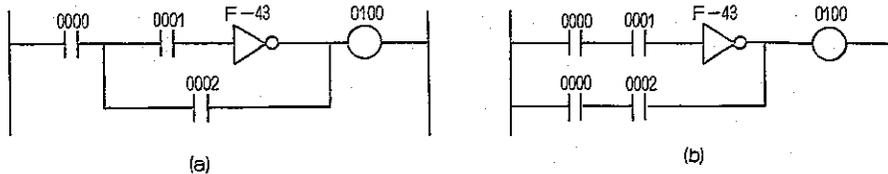
STR命令からF-43命令までの演算結果を反転し出力リレー0007に出力します。

F-43を使用すると補助接点を使うことなく、反転出力が得られます。



注1 F-43命令の条件は、1接点でも複数の接点でも構いません。

注2 F-43は直前のACCの内容を反転する命令のため、次の(a)と(b)のプログラムでは同じ演算結果が得られませんので注意してください。



F-43

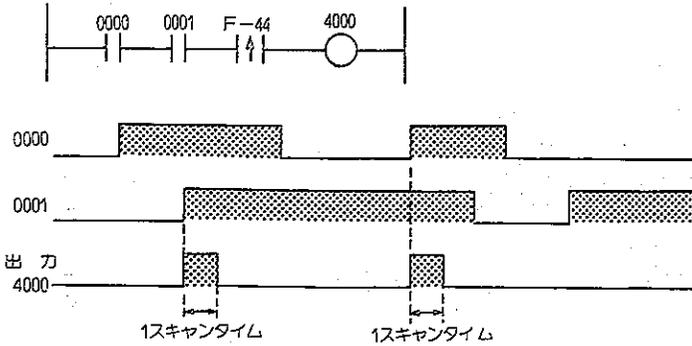
4

# F-44



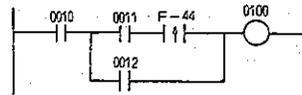
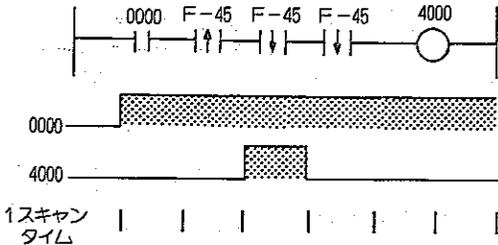
## ON時微分

F-44命令の直前のACC(アキュムレータ)の状態がOFF→ONと変化した時に1スキャンタイムのパルスが発生します。



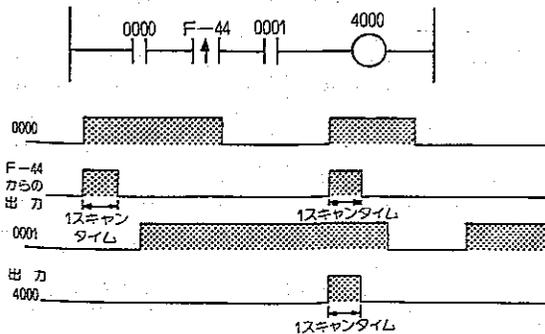
命 令	
STR	0000
AND	0001
F-44	
OUT	4000

注1 160CUでは、下記のような微分命令の連続使用は出来ません。ただし501CU3では、微分命令の連続使用数により、必要スキャンタイム遅らせたパルスを作ることが可能です。



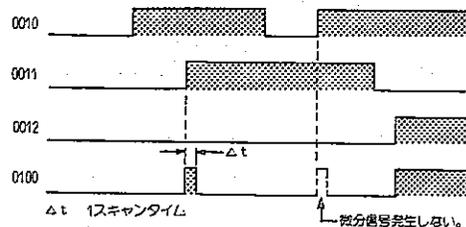
STR	0010
STR	0011
F-44	
OR	0012
AND STR	
OUT	0100

注2 上記ラダー図でF-44のプログラムする順序を変えると、結果が変わりますので注意してください。(F-45の場合も同様です)



	アキュムレータ ACC	スタックレジスタ S1
STR 0010	0010	
STR 0011	0011	0010
F-44	0011 F-44	0010
OR 0012	0011 F-44	0010
AND STR	0010 0011 F-44	
OUT 0100	0010 0011 F-44	

0011がOFF→ONに変化したスキャンサイクルのみACCがON



注3 F-44命令の条件は、1接点でも複数の接点でも構いません。

注4 F-44は、直前のACC(アキュムレータ)の内容がOFF→ONに変化した場合、これを検知して1スキャンタイムの間ONとなる信号を得る命令です。

上記の例の場合、AND STR命令で0010とのANDが演算されるため、0011がONのときに0010がOFF→ONとなっても微分信号は発生しません。

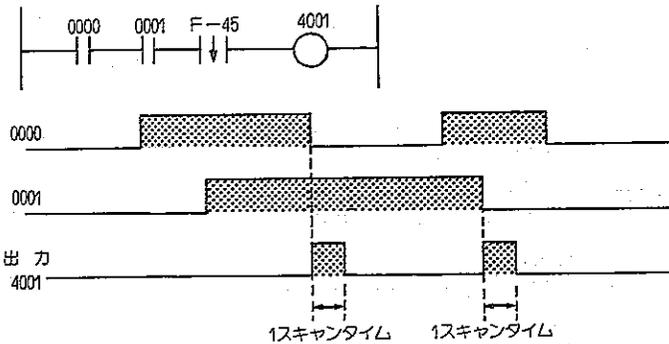
F-44

4

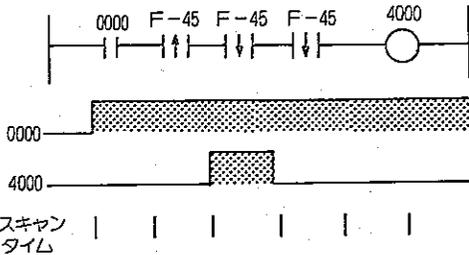
# F-45 OFF時微分

F-45命令の直前のACC(アキュムレータ)の状態が、ON→OFFと変化した時に、1スキャンタイムのパルスが発生します。

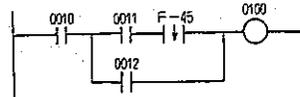
命 令	
STR	0000
AND	0001
F-45	
OUT	4001



注1 160CUでは、下記のような微分命令の連続使用は出来ません。ただし01CU3では、微分命令の連続使用数により、必要スキャンタイム遅らせたパルスを作ることが可能です。



て1スキャンタイムの間ONとなる信号を得る命令です。

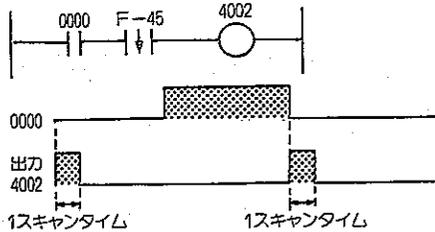


STR	0010
STR	0011
F-45	
OR	0012
AND STR	
OUT	0100

F-45

注2 微分命令 (F-44、F-45) は、プログラム中何回使っても構いません。

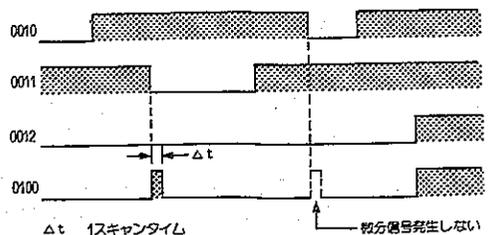
注3 OFF時微分命令を使用すると、プログラム書込 (F-45命令の書込又は挿入、削除などによりF-45命令のプログラムアドレスが移動する場合) 直後の運転時に1スキャンタイムのパルスが発生する場合があります。



↑  
プログラム書込直後の運転開始時に、入力 (0000) がOFF状態の場合に出力 (4002) がONとなります。

	アキュムレータ ACC	スタックレジスタ S <sub>n</sub>
STR 0010	0010	
STR 0011	0011	0010
F-45	0011 F-45	0010
OR 0012	0011 F-45 0012	0010
AND STR	0010 0011 F-45 0012	
OUT 0100	0010 0011 F-45 0012	

0011がON→OFFに変化したスキャンサイクルのみACCがON



上記の例の場合、AND STR命令で0010とのANDが演算されるため、0011がONのときに0010がON→OFFとなっても微分信号は発生しません。

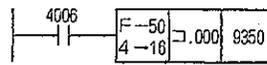
4

注4 F-45は、直前のACC(アキュムレータ)の内容がON→OFFに変化した場合、これを検知し

**F-50** 4→16デコーダ  
4→16

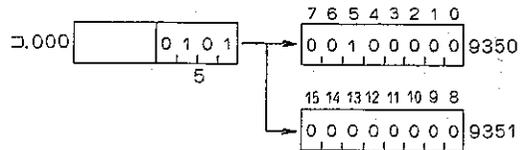
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td style="padding: 2px;">F-50</td> <td style="padding: 2px;">S</td> <td style="padding: 2px;">D</td> </tr> <tr> <td style="padding: 2px;">4→16</td> <td></td> <td></td> </tr> </table>		F-50	S	D	4→16		
F-50	S	D						
4→16								
機能	レジスタSの下位4ビットのデータをデコードし、レジスタD、D+1の2バイトに16ビットのデータとして格納する。							
演算内容	S→D、D+1							
Sの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377							
Dの使用範囲	コ.000~コ.732(注1) b.000~b.377 9000~9377							
演算条件	入力信号の立上り(OFF→ON)							
演算後	Sの内容	不変						
	Dの内容	演算結果(0~7)						
	D+1の内容	演算結果(8~15)						
	フラグ	不変						

(解説)



命 令	
STR	4006
F-50	
	コ.000
	9350

入力条件4006がOFF→ONの変化時に、レジスタ コ.000の下位4ビットのデータをデコードし、レジスタ9350と9351の2バイトに16ビットのデータとして格納します。



下位4ビットの数値0~15に相当するビットの位置のみがONし、その他のビットはOFFとなります。

注1) コ.734~コ.737は特殊領域です。(3-2(3)“キープリレーの特殊領域”参照)

注2) Sの上4ビットは演算上無視されます。

F-50

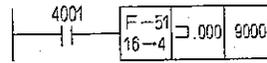
4

F-51  
16→4

# 16→4エンコーダ

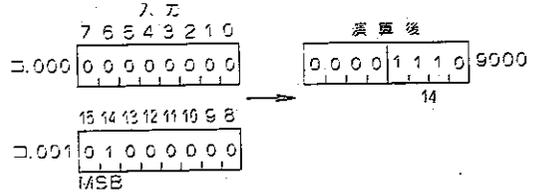
シンボル		
機能	レジスタS、S+1の2バイトのデータをエンコードし、レジスタDに格納する。	
演算内容	S、S+1→D	
Sの使用範囲	コ.000~コ.732 <sup>(注1)</sup> b.000~b.377 9000~9377	
Dの使用範囲	コ.000~コ.733 <sup>(注1)</sup> b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	S、S+1の内容	不変
	Dの内容	演算結果
	フラグ	不変

(解説)

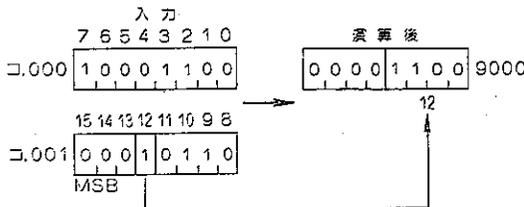


命令	
STR	4001
F-51	コ.000 9000

入力条件4001がOFF→ONの変化時に、レジスタコ.000とコ.001の2バイトのデータをエンコードし、レジスタ9000に格納します。



- 注1 コ.734~コ.737は特殊領域です。(3-2(3)“キープルーの特殊領域”参照)
- 注2 演算後、D(例の場合9000)の上位4ビットは常に0になります。
- 注3 エンコーダの入力はMSB側が優先されます。

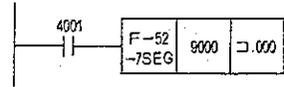


TIG

# F-52 7SEGデコーダ →7SEG

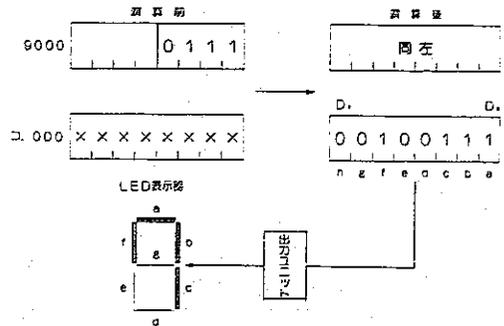
シンボル		
機能	レジスタSの下位4ビットのデータを7セグメントの表示データにデコードする。	
演算内容	S→D	
Sの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Sの内容	不変
	Dの内容	演算結果 (“7セグメントデコーダ表”参照)
	フラグ	不変

(解説)



命令	
STR	4001
F-52	9000 コ.000

入力条件4001がOFF→ONの変化時に、レジスタ9000の内容(下4ビット)が7セグメントの表示データにデコードされます。入力データと表示出力の関係は“7セグメントデコーダ表”をご覧ください。



- 注1 コ.734~コ.737は特殊領域です。(3-2(3)“キープリレーの特殊領域”参照)
- 注2 出力データD<sub>0</sub>~D<sub>6</sub>は7セグメント表示器のa~gに対応しています。D<sub>7</sub>は常に“0”出力です。

7セグメント デコーダ表

入力データ	出力データ	表示出力
	gfedcba	
0000000	00111111	0
0000001	00000110	1
0000010	01011011	2
0000011	01001111	3
0000100	01100110	4
0000101	01101101	5
0000110	01111101	6
0000111	00100111	7
0001000	01111111	8
0001001	01101111	9
0001010	01110111	A
0001011	01111100	b
0001100	00111001	c
0001101	01011110	d
0001110	01111001	e
0001111	01110001	f



F-52

4

F-53  
→BIN

# BCD(4桁)→BIN(16ビット)変換

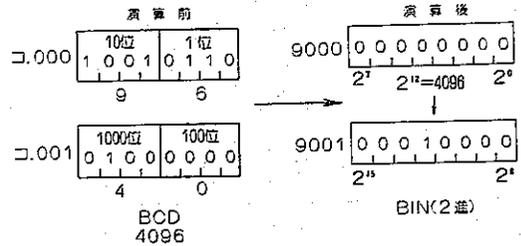
シンボル	<table border="1"> <tr> <td>F-53</td> <td>S</td> <td>D</td> </tr> <tr> <td colspan="3">→BIN</td> </tr> </table>					F-53	S	D	→BIN		
F-53	S	D									
→BIN											
機能	レジスタS、S+1の2バイトのBCD 4桁データを2進に変換し、レジスタD、D+1の2バイトに格納する。										
演算内容	S、S+1→D、D+1										
Sの使用範囲	コ.000~コ.732(注1) b.000~b.377 9000~9377										
Dの使用範囲	コ.000~コ.732(注1) b.000~b.377 9000~9377										
演算条件	入力信号の立上り(OFF→ON)										
演算後	S、S+1の内容	不変									
	Dの内容	演算結果 (0~255)	レジスタS、S+1の内容がBCDコードでない時不変								
	D+1の内容	演算結果 (256~9999)									
	フラグ	レジスタS、S+1の内容	ノギヤ 7354	エラー 7355	キャリー 7356	ゼロ 7357					
	BCDコード	0	0	0	0						
	BCDコードでない時		1								

(解説)



命 令	
STR	4001
F-53	
	コ.000
	9000

入力条件4001がOFF→ONの変化時に、レジスタコ.000とコ.001のBCD 4桁データを2進に変換し、レジスタ9000と9001の2バイトに変換データを格納します。



注1 コ.734~コ.737は特殊領域です。(3-2(3)“キープリレーの特殊領域”参照)

注2 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5)“データ処理命令とフラグ”参照)

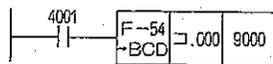
F-53

**F-54  
→BCD**

**BIN(16ビット)→BCD(6桁)変換**

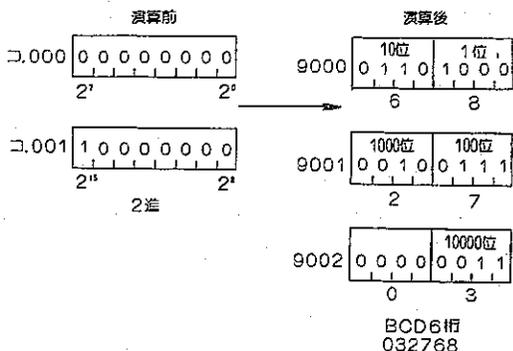
シンボル		
機能	レジスタS、S+1の2バイトの2進データをBCD6桁に変換し、レジスタD、D+1、D+2の3バイトに格納する。	
演算内容	S、S+1→D、D+1、D+2	
Sの使用範囲	コ.000~コ.732(注1) b.000~b.377 9000~9377	
Dの使用範囲	コ.000~コ.731(注1) b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	S、S+1の内容	不変
	Dの内容	演算結果(1の位と10の位)
	D+1の内容	演算結果(100の位と1,000の位)
	D+2の内容	演算結果(10,000の位)
	フラグ	不変

(解説)



命 令	
STR	4001
F-54	
	コ.000
	9000

入力条件4001がOFF→ONの変化時に、レジスタコ.000とコ.001の2バイト2進データをBCD6桁に変換し、レジスタ9000からの3バイトに変換データを格納します。



注1) コ.734~コ.737は特殊領域です。(3-2(3)“キープリレーの特殊領域”参照)

F-54

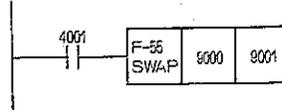
4

**F-55  
SWAP**

上位4ビットと下位4ビットの交換  
(SWAP)

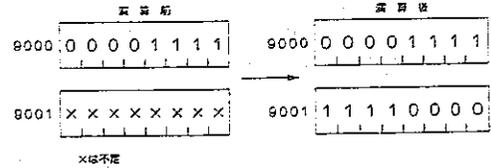
シンボル		
機能	レジスタSの内容の上下4ビットずつを交換し、レジスタDに格納する。	
演算内容	S → D	
Sの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	Sの内容	不変
	Dの内容	演算結果
	フラグ	不変

(解説)



命 令	
STR	4001
F-55	9000
	9001

入力条件4001がOFF→ONの変化時に、レジスタ9000の上下4ビットずつを交換し、レジスタ9001に格納します。  
レジスタ9000の内容は不変です。



1-1010

**注1** コ.734~コ.737は特殊領域です。(3-2(3)“キーブリレーの特殊領域”参照)

**参考** F-55命令は次のようなときに有効です。  
F-52命令(7SEGアコーダ)は、下4ビットが7セグメントデータにアコードされます。多桁の表示をするとき、F-55命令により上4ビットと下4ビットを交換し、再度F-52を使用します。

# F-60 SFR

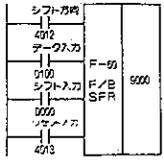
## 両方向シフトレジスタ (Forward/Backward Shift Register)

シンボル		①シフト方向指示入力 ②データ入力 ③シフト入力 ④リセット入力				
機能	レジスタDの8ビットデータをシフト方向指示入力①に従って上位ビット、又は下位ビットへシフトする。					
演算内容	・シフト方向指示入力①がONの場合  ・シフト方向指示入力①がOFFの場合 					
Dの使用範囲	0.000~0.733(注1) b.000~b.377 9000~9377					
演算条件	リセット入力④がOFFの時、シフト入力③の立上り(OFF→ON)でシフト					
演算後	Dの内容	・リセット入力④がOFFの時、演算結果 ・リセット入力④がONの時、全ビットOFF				
	フラグ	リセット入力④	ノンキャリアー 7354	エラー 7355	キャリアー 7356	ゼロ 7357
		OFF	1又は0	0	0又は1	0又は1
ON	0	0	0	0		

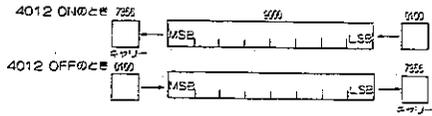
- 注1 0.734~0.737は特殊領域です。(3-2(3) "キープリレーの特殊領域" 参照)
- 注2 リセット入力④はシステムメモリ(#202)にリセット条件を設定することにより"OFFでリセット"とすることもできます。

- 注3 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令まで有効です。(4-5(5) "データ処理命令とフラグ" 参照)

### (解説)



シフト入力0000のOFF→ONの変化時、シフト方向指示入力4012の状態により、次のようにシフトされます。



命 令	
STR	4012
STR	0100
STR	0000
STR	4013
F-60	9000

入力条件	9000 (演算前)								9000 (演算後)								ノンキャリアー 7354	キャリアー 7356	ゼロ 7357
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
4012 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	○	●
0100 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
0000 上	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
4013 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
4012 ●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
0100 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
0000 上	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
4013 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
4012 ●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
0100 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
0000 上	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
4013 ○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
4013 ●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

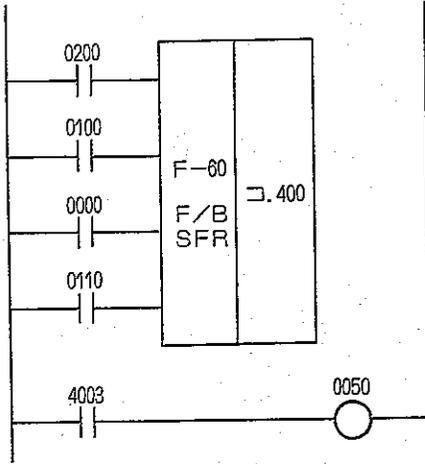
・エラーフラグ(7355)は常にOFFとなります。

○ OFF ● ON

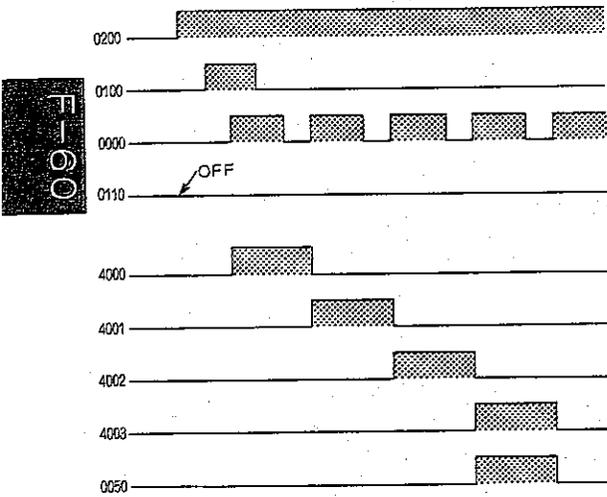
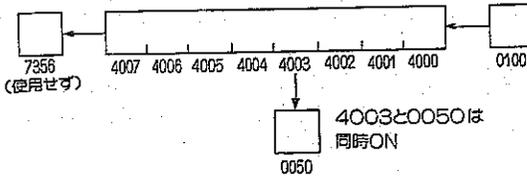
CO-11

4

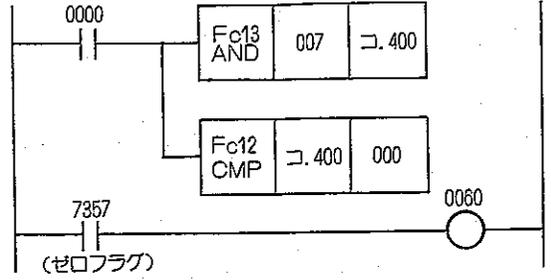
【参考】 Dにコ.×××の領域を使用すると、nビット(n<8)のシフトレジスタを構成できます。



(0200がONの場合)

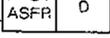


(注1) 4004~4007にもデータがシフトされます。  
 (注2) ゼロフラグは4000~4007が全て0のとき1となります。4000~4002が0であることを知る必要がある場合は、次のプログラムを追加します。



00000011 とANDすることで4003~4007をマスク(すべて0にする)しています。

# F-61 ASFR (Asynchronous Shift Register)

シンボル	①  ② 	①シフト方向指示入力 ②シフト入力				
機能	シフト方向指示入力①に従って、レジスタD-1 (①ON)またはレジスタD+1 (①OFF) の1バイトデータをレジスタDにシフトする。					
演算内容	<ul style="list-style-type: none"> <li>シフト方向指示入力ONのとき D-1 → D</li> <li>シフト方向指示入力OFFのとき D+1 → D</li> </ul>					
Dの使用範囲	コ.001~コ.732(注1) b.000~b.377 9000~9377					
演算条件	レジスタDの内容が0の時、シフト入力②がONでシフト(OFF→ONの変化時に限定されない)					
レジスタ	① ON		① OFF		① ON/OFF	
	演算前	演算後	演算前	演算後	演算前	演算後
D-1の内容	D1	0	D1	同左	D1	同左
Dの内容	0	D1	0	D2	0以外	同左
D+1の内容	D2	同左	D2	0	D2	同左
フ ラ グ	ノンキャリアー 7354	1 (D1=0) 0 (D1≠0)	1 (D2=0) 0 (D2≠0)	1		
	エラー 7355	0	0	0	0	0
	キャリアー 7356	0 (D1=0) 1 (D1≠0)	0 (D2=0) 1 (D2≠0)	0	0	0
	ゼロ 7357	0	0	0	0	0

- 注1 コ.734~コ.737は特殊領域です。(3-2(3) "キープリレーの特殊領域" 参照)
- 注2 演算が実行されると、シフトしたレジスタ (D-1又はD+1) の内容はクリアされます。
- 注3 Dの内容が0でないとき、演算は実行されません。
- 注4 D-1又はD+1から0以外のデータがシフトされた場合だけ、キャリアーフラグ(7356)がONします。

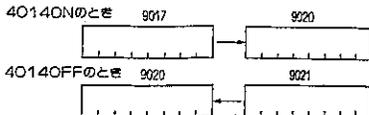
- 注5 シフト入力②がONの間、毎スキャンサイクル演算が実行されます。
- 注6 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5) "データ処理命令とフラグ" 参照)

〔解説〕



命 令	
STR	4014
STR	4015
F-61	9020

シフト入力4015がONの間、シフト方向指示入力4014の状態により、次のように1バイト単位のデータがシフトされます。



- 演算前9020の内容が0でないとき、シフトは実行されません。
- シフトしたレジスタ(9017または9021)の内容はクリアされます。

入力条件	演 算 前		演 算 後		ノンキャリアー 7354	キャリアー 7356	ゼロ 7357
	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0			
4014 ● 4015 ●	9017 9020 9021	● ●	● ●	● ●	●	○	○
4014 ● 4015 ○	9017 9020 9021	○ ○	○ ○	○ ○	○	●	○
4014 ○ 4015 ●	9017 9020 9021	○ ○	○ ○	○ ○	○	○	○

・エラーフラグ(7355)は常にOFFとなります。

○ OFF ● ON

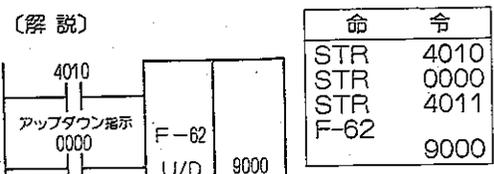
10-11

4

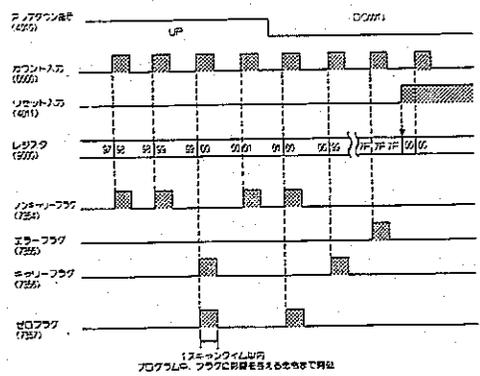
# F-62 BCD 2桁のアップ・ダウンカウンタ

## U/D CNT (Up/Down Counter)

シンボル		①アップ・ダウン指示入力 ②カウント入力 ③リセット入力																																																
機能	アップ・ダウン指示入力①に従ってレジスタDの内容(BCD2桁)を加算(①ON)または減算(①OFF)する。																																																	
演算内容	①アップダウン指示入力ONのとき $\langle D \rangle + 1 \rightarrow D$ ①アップダウン指示入力OFFのとき $\langle D \rangle - 1 \rightarrow D$																																																	
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377																																																	
演算条件	リセット入力③がOFFの時、カウント入力②の立上り(OFF→ON)																																																	
演算後	Dの内容	●リセット入力③がOFFの時 演算結果(BCDコード) ●リセット入力③がONの時 全ビットOFF																																																
	フラグ	<table border="1"> <thead> <tr> <th>アップ・ダウン指示入力①</th> <th>演算結果</th> <th>ノンキャリー 7354</th> <th>エラー 7355</th> <th>キャリー 7356</th> <th>ゼロ 7357</th> </tr> </thead> <tbody> <tr> <td rowspan="3">ON</td> <td>99+1 →00</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>00~98+1 →01~99</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>BCD以外の数値</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td rowspan="4">OFF</td> <td>00-1 →99</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>01-1 →00</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>02~99-1 →01~98</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>BCD以外の数値</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>リセット入力③ONの時</td> <td></td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </tbody> </table>	アップ・ダウン指示入力①	演算結果	ノンキャリー 7354	エラー 7355	キャリー 7356	ゼロ 7357	ON	99+1 →00	0	0	1	1	00~98+1 →01~99	1	0	0	0	BCD以外の数値	0	1	0	0	OFF	00-1 →99	0	0	1	0	01-1 →00	1	0	0	1	02~99-1 →01~98	1	0	0	0	BCD以外の数値	0	1	0	0	リセット入力③ONの時		0	0	0
アップ・ダウン指示入力①	演算結果	ノンキャリー 7354	エラー 7355	キャリー 7356	ゼロ 7357																																													
ON	99+1 →00	0	0	1	1																																													
	00~98+1 →01~99	1	0	0	0																																													
	BCD以外の数値	0	1	0	0																																													
OFF	00-1 →99	0	0	1	0																																													
	01-1 →00	1	0	0	1																																													
	02~99-1 →01~98	1	0	0	0																																													
	BCD以外の数値	0	1	0	0																																													
リセット入力③ONの時		0	0	0	0																																													



リセット入力4011がOFFで計数可能となります。(ON)リセットに設定時)  
 アップダウン指示入力4010がONの時加算、OFFの時減算カウンタとして動作します。他の命令により9000の内容がBCD以外のコードになったとき、エラーフラグ(7355)がONし、カウント動作は実行しません。(例では7F)



注1 コ.734~コ.737は特殊領域です。(3-2(3)“キープリレーの特殊領域”参照)  
 注2 リセット入力③はシステムメモリ(#202)にリセット条件を設定することにより“OFF”でリセットすることもできます。

注3 フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5)“一タ処理命令とフラグ”参照)

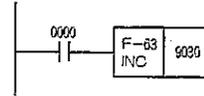
F-62

F-63  
INC

加算カウンタ  
(INCrement)

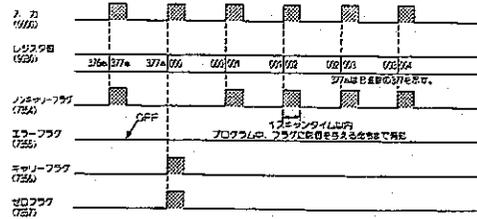
シンボル	— <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="padding: 2px;">F-63 INC</td><td style="padding: 2px;">D</td></tr></table>					F-63 INC	D
F-63 INC	D						
機能	レジスタDの内容(バイナリーデータ)を加算カウントする。						
演算内容	$\langle D \rangle + 1 \rightarrow D$						
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377						
演算条件	入力信号の立上り(OFF→ON)						
演算後	Dの内容	演算結果 (バイナリーコード)					
	フラグ	演算結果	ノン キャリー 7354	エラー 7355	キャリー 7356	ゼロ 7357	
		377 <sup>注2)</sup> →000	0	0	1	1	
上記以外	1	0	0	0			

(解説)



命 令	
STR	0000
F-63	9030

入力条件0000のOFF→ONを検知して、加算カウントします。



注1) コ.734~コ.737は特殊領域です。(3-2(3))"キーブリレー"の特殊領域参照)

注2) Dの内容はバイナリコードです。10進表現では000~255、8進表現では000~377<sub>8</sub>と見なすことができます。

注3) フラグの状態はそのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5))"データ処理命令とフラグ"参照)

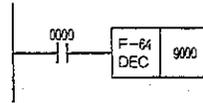
391

4

# F-64 減算カウンタ DEC (DECrement)

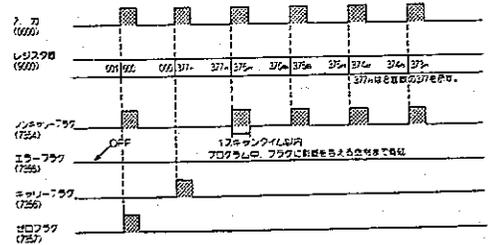
シンボル					
機能	レジスタDの内容(バイナリーデータ)を減算カウントする。				
演算内容	$\langle D \rangle - 1 \rightarrow D$				
Dの使用範囲	c. 000 ~ c. 733 (注1) b. 000 ~ b. 377 9000 ~ 9377				
演算条件	入力信号の立上り(OFF→ON)				
演算後	Dの内容	演算結果(バイナリーコード)			
	演算結果	ノン キャリー 7354	エラー 7355	キャリー 7356	ゼロ 7357
	001 <sub>10</sub> → 000 <sub>10</sub>	1	0	0	1
	000 <sub>10</sub> → 377 <sub>10</sub>	0	0	1	0
上記 以外	1	0	0	0	

(解説)



命令	
STR	0000
F-64	9000

入力条件0000のOFF→ONの変化を検知して減算カウントします。



**注1** c. 734 ~ c. 737は特殊領域です。(3-2(3)“キーブリレーの特殊領域”参照)

**注2** Dの内容はバイナリーコードです。10進表現では000 ~ 255、8進表現では000 ~ 377<sub>10</sub>と見なすことができます。

**注3** フラグの状態は、そのスキャンサイクル中の次のフラグに影響を与える命令までが有効です。(4-5(5)“ラータ処理命令とフラグ”参照)

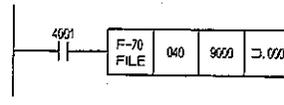
F-64

**F-70  
FILE**

**nバイト一括転送  
(FILE)**

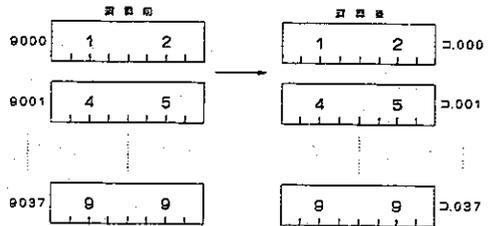
シンボル		
機能	レジスタSからS+n-1までのnバイト(8進数)のデータをレジスタDからD+n-1までのnバイトに一括転送する。	
演算内容	S, ..., S+n-1 → D, ..., D+n-1	
nの使用範囲	000~200 <sub>8</sub> (000とすると128バイトとなる)	
Sの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377 30000~37777	
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377 30000~37777	
演算条件	入力信号の立上り(OFF→ON)	
演算後	S~S+n-1の内容	不変
	Dの内容	レジスタSの内容
	D+1の内容	レジスタS+1の内容
	D+n-1の内容	レジスタS+n-1の内容
フラグ	不変	

〔解説〕



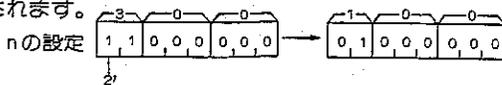
命令	
STR	4001
F-70	040
	9000
	コ.000

入力条件4001がOFF→ONの変化時に、レジスタ9000から9037までの040<sub>8</sub>バイト(10進数で32バイト)のデータをコ.000からコ.037までの32バイトに一括転送できます。レジスタ9000から9037までの内容は不変です。



注1 コ.734~コ.737は特殊領域です。(3-2(3)“キープリレーの特殊領域”参照)

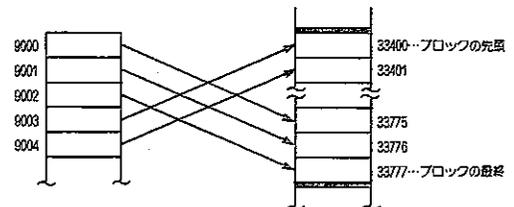
注2 転送バイト数nは8進数で000~200の範囲としてください。プログラムとしては300<sub>8</sub>等の設定も可能ですが、2<sup>7</sup>のビットは無視され、100<sub>8</sub>と見なし処理されます。



またnを000<sub>8</sub>とプログラムすると200<sub>8</sub>と見なし処理されます。

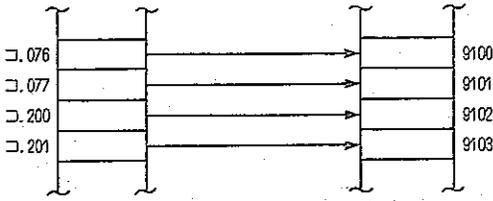
注3 データメモリのブロック(4-5(8)“基準アドレスとデータポインタ”参照)をまたがるようなn、S、Dの設定を行うと次のブロックには継がらず、同一ブロックの先頭アドレスに戻ります。

— とプログラムすると

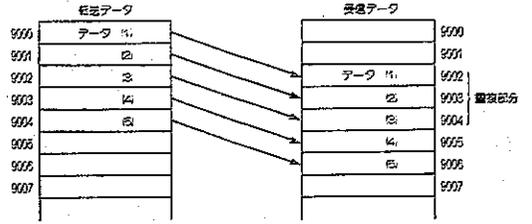
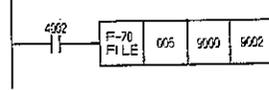


**注4** S、Dにコ. XXXの領域を使用するとき、リレー領域の欠番(4-5(7)“リレー領域を使用するとき”参照)を詰めた形で転送が行われます。

とプログラムすると、コ. 100  
 ~コ. 177は欠番のため、



**注6** 転送元、転送先が重複するようなn、S、Dの設定も可能です。



**注5** 30000~37777のファイルレジスタは、システムメモリ(#205)にファイルレジスタ容量を設定した場合に使用可能です。  
 3-4(2)“ファイルレジスタ”の項をご参照ください。

F-70

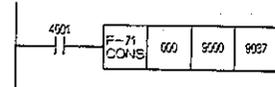
4

**F-71  
CONS**

**8進定数一括転送  
(CONSTant)**

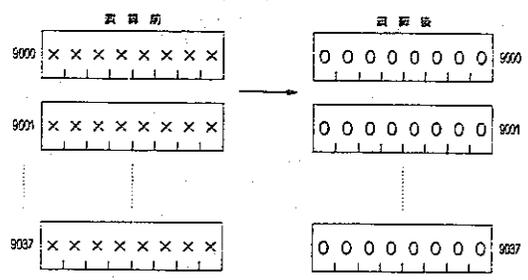
シンボル		
機能	レジスタD <sub>1</sub> からレジスタD <sub>2</sub> に8進定数nを一括転送する。	
演算内容	n → D <sub>1</sub> 、……、D <sub>2</sub>	
nの使用範囲	000~377(o)	
D <sub>1</sub> の使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377 30000~37777	
D <sub>2</sub> の使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377 30000~37777	
演算条件	入力信号の立上り(OFF→ON)	
演算後	D <sub>1</sub> の内容 D <sub>1</sub> +1の内容 ⋮ D <sub>2</sub> -1の内容 D <sub>2</sub> の内容	定数 n
	フラグ	不変

(解説)



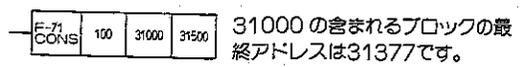
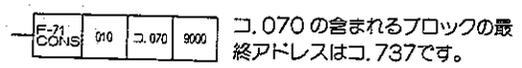
命 令	
STR	4001
F-71	
	000
	9000
	9037

入力条件4001がOFF → ONの変化時に、レジスタ9000から9037に8進定数000を一括転送します。

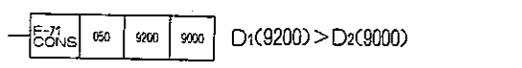


**注1** コ.734~コ.737は特殊領域です。(3-2(3)“キーブリレーの特殊領域”参照)

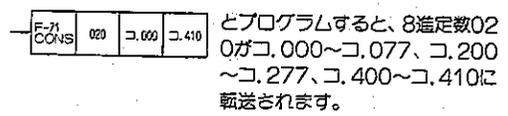
**注2** テータメモリのブロック(4-5(8)“基準アドレスとデータポインタ”参照)をまたがるようなD<sub>1</sub>、D<sub>2</sub>の設定をすると演算を実行しません。



**注3** D<sub>1</sub> > D<sub>2</sub>となるアドレスを設定すると、演算を実行しません。



**注4** D<sub>1</sub>、D<sub>2</sub>にコ.×××の領域を使用するとき、リレー番号の欠番(4-5(7)“リレー領域を使用するとき”参照)を詰められた形で転送が行われます。



**注5** 30000~37777のファイルレジスタはシステムメモリ(#205)にファイルレジスタ容量を設定した場合に使用可能です。3-4(2)“ファイルレジスタ”の項をご参照ください。

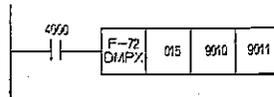
F-71

**F-72  
DMPX**

**ファイルレジスタへの分配  
(DeMultiPleXer)**

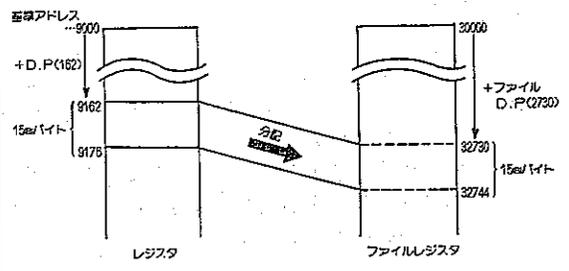
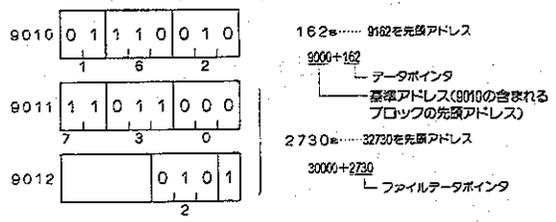
シンボル		
機能	Sが含まれるデータメモリブロックの先頭アドレス(基準アドレス)からSの内容(データポインタ)だけ変位したレジスタを先頭とするnバイトのレジスタ群の内容を、30000からD、D+1の内容(ファイルデータポインタ)だけ変位したファイルレジスタを先頭とするnバイトのファイルレジスタ群へ転送する。	
演算内容	$X + \langle S \rangle, \dots, X + \langle S \rangle + n - 1$ $\rightarrow 30000 + \langle D, D + 1 \rangle, \dots,$ $30000 + \langle D, D + 1 \rangle + n - 1$ $X \dots S$ が含まれるデータメモリブロックの先頭アドレス(基準アドレス) $\langle S \rangle \dots$ データポインタ $\langle D, D + 1 \rangle \dots$ ファイルデータポインタ	
nの使用範囲	000~200 <sub>ea</sub> (000とすると128バイトとなる)	
Sの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	ファイルレジスタ以外のレジスタ	不変
	30000+ $\langle D, D + 1 \rangle$	レジスタX+ $\langle S \rangle$ の内容
	30000+ $\langle D, D + 1 \rangle + 1$	レジスタX+ $\langle S \rangle + 1$ の内容
	30000+ $\langle D, D + 1 \rangle + n - 1$	レジスタX+ $\langle S \rangle + n - 1$ の内容
フラグ	不変	

(解説)



命 令	
STR	4000
F-72	015
	9010
	9011

入力条件4000がOFF→ONの変化時に、9000(9010が含まれるデータメモリブロックの先頭アドレス…基準アドレス)から9010の内容(データポインタ)だけ変位したレジスタを先頭とする015<sub>ea</sub>バイトのレジスタ群の内容を、30000から9011と9012の内容(ファイルデータポインタ)だけ変位したファイルレジスタを先頭とする015<sub>ea</sub>バイトのファイルレジスタ群へ転送します。



注1 コ.734~コ.737は特殊領域です。(3-2(3)“キープリューの特殊領域”参照)

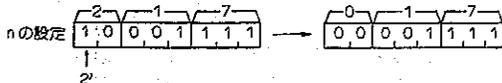
F-72

4

注2 レジスタ側の基準アドレスは、コ.000、b.000、9000で、Sの含まれるデータメモリのブロックの先頭アドレスとなります。(4-5(8)“基準アドレスとデータポインタ”参照)

注3 F-72はシステムメモリ(#205)にファイルレジスタ容量を設定した場合に使用可能です。(3-4(2)“ファイルレジスタ”参照)

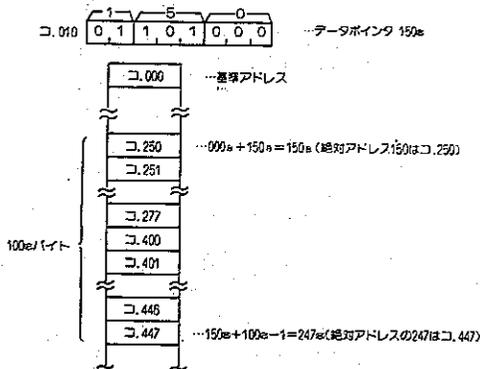
注4 転送バイト数nは8進数で000~200<sub>8</sub>の範囲としてください。  
プログラムとしては217<sub>8</sub>等の設定も可能ですが2<sup>7</sup>のビットは無視され017<sub>8</sub>と見なし処理されず。



またnを000<sub>8</sub>とプログラムすると200<sub>8</sub>と見なし処理されます。

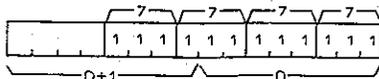
注5 Sにコ.×××の領域を使用するとき、コ.000を基準アドレスとした絶対アドレスで先頭アドレスが決定し、転送範囲も絶対アドレスで決まります。

— F-72 DMPX 100 コ.010 9100 とプログラムしたとき



4-5(7)“リレー領域を使用するとき”の項をご参照ください。

注6 ファイルデータポインタはD、D+1の2バイト(16ビット)中の下12ビットで示されます。



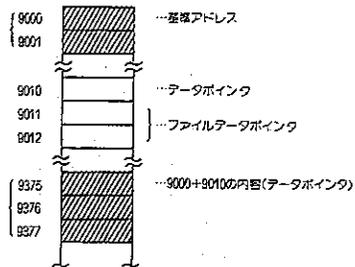
したがって0000~7777の範囲を取り得るため30000~37777のファイルレジスタ全域を指定できます。

D+1の上4ビットは演算上無視されます。

注7 (基準アドレス+データポインタ+n-1)がデータメモリのブロックの最終アドレスを越えると、次のブロックには継がらず、同一ブロックの先頭アドレスに戻ります。

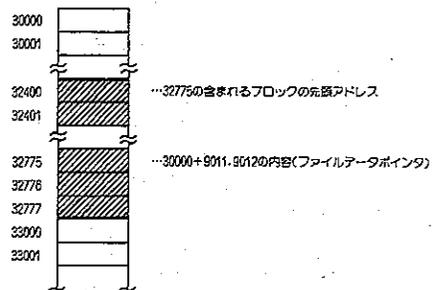


(1)レジスタ側の例



9010の内容(データポインタ)が375<sub>8</sub>のとき、9375.9376.9377の3バイトと9000.9001の2バイトの内容がファイルレジスタに転送されます。

(2)ファイルレジスタ側の例



9011.9012の内容が2775<sub>8</sub>のとき、32775.32776.32777の3バイトと32400.32401の2バイトにレジスタから転送されます。(33000.33001には継がりません。)

F-72

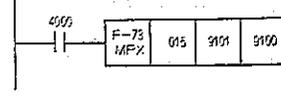
4

**F-73  
MPX**

**ファイルレジスタからの抽出  
(MultiPleXer)**

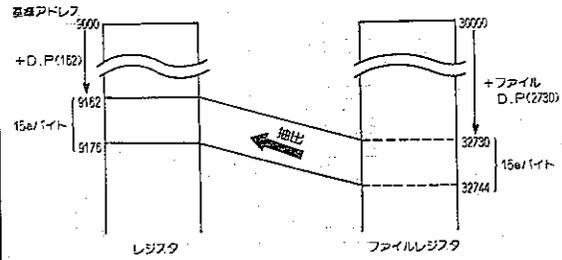
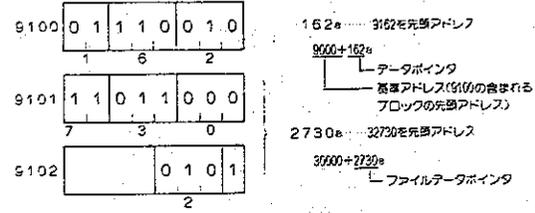
シンボル		
機能	30000からS、S+1の内容（ファイルテータポイント）だけ変位したファイルレジスタを先頭とするnバイトのファイルレジスタ群の内容をDが含まれるデータメモリブロックの先頭アドレス（基準アドレス）からDの内容（データポイント）だけ変位したレジスタを先頭とするnバイトのレジスタ群へ転送する。	
演算内容	$30000 + \langle S, S+1 \rangle, \dots$ $30000 + \langle S, S+1 \rangle + n - 1$ $\rightarrow X + \langle D \rangle, \dots X + \langle D \rangle + n - 1$ $\langle S, S+1 \rangle \dots$ ファイルデータポイント $X \dots D$ が含まれるデータメモリブロックの先頭アドレス（基準アドレス） $\langle D \rangle \dots$ データポイント	
nの使用範囲	000~200 <sub>16</sub> (000とすると128/バイトとなる)	
Sの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
Dの使用範囲	コ.000~コ.733(注1) b.000~b.377 9000~9377	
演算条件	入力信号の立上り(OFF→ON)	
演算後	ファイルレジスタ	不変
	X+⟨D⟩	ファイルレジスタ30000+⟨S, S+1⟩の内容
	X+⟨D⟩+1	ファイルレジスタ30000+⟨S, S+1⟩+1の内容
	X+⟨D⟩+n-1	ファイルレジスタ30000+⟨S, S+1⟩+n-1の内容
フラグ	不変	

(解説)



命 令	
STR	4000
F-73	015
	9101
	9100

入力条件4000がOFF→ONの変化時に30000から9101, 9102の内容(ファイルテータポイント)だけ変位したファイルレジスタを先頭とする015<sub>16</sub>バイトのファイルレジスタ群の内容を、9000(9100が含まれるデータメモリブロックの先頭アドレス…基準アドレス)から9100の内容(データポイント)だけ変位したレジスタを先頭とする015<sub>16</sub>バイトのレジスタ群へ転送します。



注1) コ.734~コ.737は特殊領域です。(3-2(3)“キープルーの特殊領域”参照)

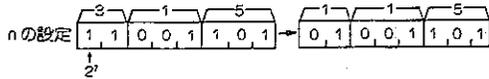
注2) レジスタ側の基準アドレスはコ.000、b.000、9000で、Dの含まれるデータメモリブロックの先頭アドレスとなります。(4-5(8)“基準アドレスとデータポイント”参照)

F-73

4

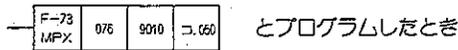
**注3** F-73はシステムメモリ(#205)にファイルレジスタ容量を設定した場合に使用可能です。(3-4(2)“ファイルレジスタ”参照)

**注4** 転送バイト数nは8進数で000~200<sub>8</sub>の範囲としてください。プログラムとしては315<sub>8</sub>等の設定も可能ですが、2<sup>7</sup>のビットは無視され115<sub>8</sub>と見なし処理されます。

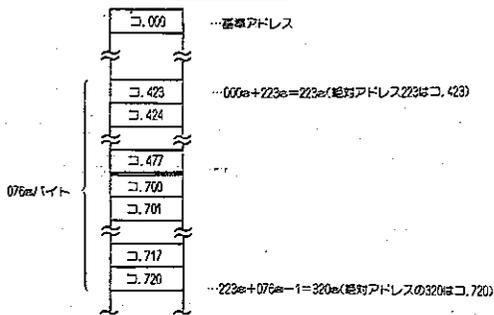


またnを000<sub>8</sub>とプログラムすると200<sub>8</sub>と見なしして処理されます。

**注5** Dにコ. x x xの領域を使用するとき、コ. 000を基準アドレスとした絶対アドレスで先頭アドレスが決定し、転送を受ける範囲も絶対アドレスで決まります。

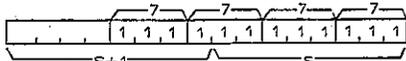


コ. 060 1 1 0 0 1 0 0 1 1 1 1 ...タークポイント 223<sub>a</sub>



レジスタ数の最終アドレスが、コ. 733を超えないようご注意ください。

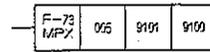
**注6** ファイルタークポイントはS, S+1の2バイト(16ビット)中の下12ビットで示されます。



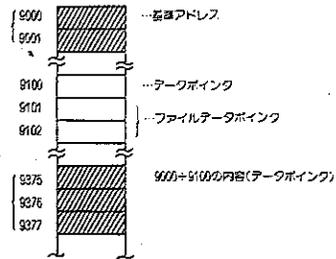
したがって0000~7777の範囲を取り得るため30000~37777のファイルレジスタの全域を指定できます。

S+1の上4ビットは演算上無視されます。

**注7** (基準アドレス+タークポイント+n-1)がタークメモリのブロックの最終アドレスを越えると、次のブロックには継がらず、同一ブロックの先頭アドレスに戻ります。

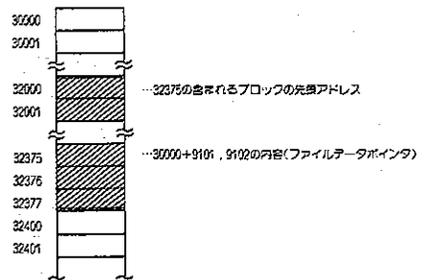


(1)レジスタ側の例



9100の内容が375<sub>8</sub>のとき、9375、9376、9377の3バイトと、9000、9001の2バイトにファイルレジスタから転送が行われます。

(2)ファイルレジスタ側の例



9101、9102の内容が2375<sub>8</sub>のとき、32375、32376、32377の3バイトと、32000、32001の2バイトの内容がレジスタに転送されます。(32400、32401には継がりません。)

F-73

4

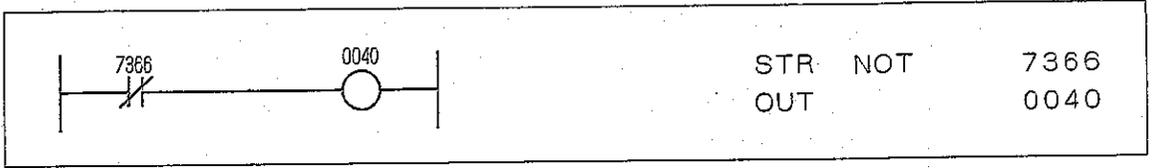
## § 5 プログラム例

5-1 基本命令の応用回路	138
〔1〕 常時ON回路	138
〔2〕 電源投入時に1パルス発生させる回路	138
〔3〕 発振回路	139
〔4〕 立上り微分	140
〔5〕 立下り微分	140
〔6〕 自己保持回路(リセット優先)	141
〔7〕 自己保持回路(セット優先)	141
〔8〕 優先回路	142
〔9〕 オルタネート回路	143
〔10〕 nビットシフトレジスタ	144
5-2 タイマの応用回路	145
〔1〕 オンディレイタイマ	145
〔2〕 オフディレイタイマ	145
〔3〕 オン・オフディレイタイマ	146
〔4〕 入力立上り時ワンショットタイマ(1)	146
〔5〕 入力立上り時ワンショットタイマ(2)	147
〔6〕 入力立下り時ワンショットタイマ	147
〔7〕 立上り・立下りワンショットタイマ	148
〔8〕 オンディレイワンショットタイマ	148
〔9〕 等間パルス発生回路	149
〔10〕 デューティ可変パルス発生回路(1)	150
〔11〕 デューティ可変パルス発生回路(2)	150
〔12〕 長時間タイマ(1)	151
〔13〕 長時間タイマ(2)	151
5-3 カウンタの応用回路	152
〔1〕 大容量カウンタ(1)	152
〔2〕 大容量カウンタ(2)	152
〔3〕 計数入力の立上り、立下りで計数するカウンタ	153
5-4 メンテナンスディスプレイ(MD)の応用	153
〔1〕 排他的入力の異常検知	153
〔2〕 入力機器のON時間の異常検知	154

5-5	データ処理命令の応用	154
(1)	レジスタのクリア	154
(2)	データのマスク	155
(3)	ビット挿入	155
(4)	数の分解	156
(5)	数の合成	156
(6)	設定値との比較	157
(7)	ウィンドウコンパレータ	158
(8)	不帯感をもつ比較回路	159
(9)	複数のセットポイントを持つタイマ	160
(10)	減算結果を符号付絶対値で求める	161
(11)	BCD 8桁の乗算	162
(12)	BCD 8桁÷BCD 2桁	164
(13)	BCD 4桁の除算	165
(14)	ドラムシーケンサ	169
(15)	タイマ現在値の外部出力	170
(16)	カウンタ現在値の外部出力	171
(17)	タイマの設定値を外部機器から入力	172
(18)	カウンタの設定値を外部機器から入力	173
(19)	複数のタイマ、カウンタの設定値を外部機器から入力	175
(20)	ダイナミック入力	178
(21)	ダイナミック出力	180
(22)	2バイトのデータをデータテーブルに格納	182
(23)	BCD 4桁の最小値・最大値を求める	183
(24)	BCD 2桁の数値の平均値を求める	185
(25)	機種コードにより複数の設定値・定数を切替える	186
(26)	同期型FIFOスタックレジスタ	189
(27)	テンキーからの数値の読み	190
(28)	8→256デコーダ	192
(29)	256→8エンコーダ	193
(30)	ファイルレジスタの領域指定クリア	195
(31)	BCD 4桁のアップダウンカウンタ	197
(32)	バイナリアップダウンカウンタ	198

## 5-1 基本命令の応用回路

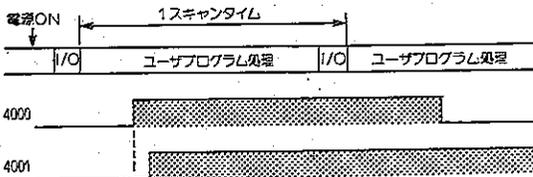
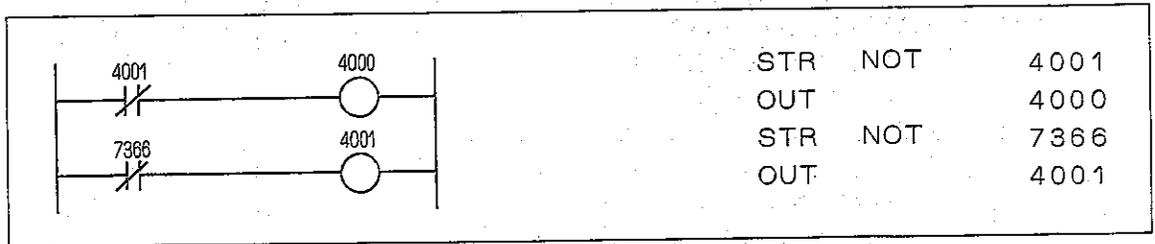
### (1) 常時ON回路



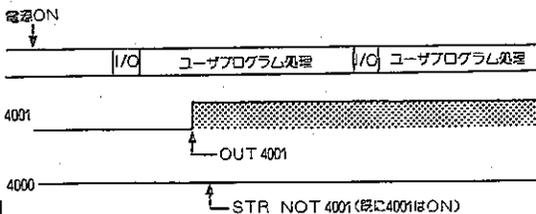
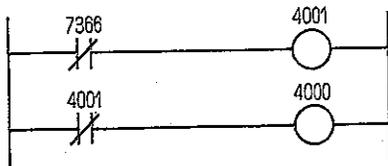
● キーブリーの7366は常時OFFの接点のため、コイル0040は常時ONとなります。0040を「停止時OFFとなる領域」に設定すると、プログラムモードに切換えたときや、自己診断の結果、本体が停止するとOFFになるため、W16/W51のRUN表示として使用できます。

**注1** システムメモリの#203で本体停止時の出力保持領域を設定できます。設定は絶対アドレスで行いません。システムメモリをクリアすると000<sub>0</sub>となり、全出力ユニットが「停止時保持」の状態になります。(出荷時システムメモリはクリアされています)  
上記の例の場合、システムメモリ#203に005を登録すると0000~0047の領域は、本体停止時リセットされます。

### (2) 電源投入時に1パルス発生させる回路



**注1** プログラム順を切換えると、パルスは発生しません。



**注2** 4001がキーブ機能指定領域のとき、パルスは発生しません。(システムメモリ#200にキーブ機能領域を設定します)

(理由) 前回の電源ON時に4001はONし、停電中もONを保持。

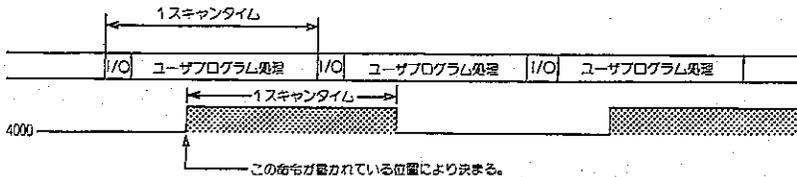
**注3** このパルスは電源投入時にレジスタをクリアしたり、ある定数をプリセットする時に用います。上記プログラムは、レジスタをクリアする命令より以前に書いておく必要があります。

(ただし は より後であればどこに書いてもかまいません。)

### (3) 発振回路

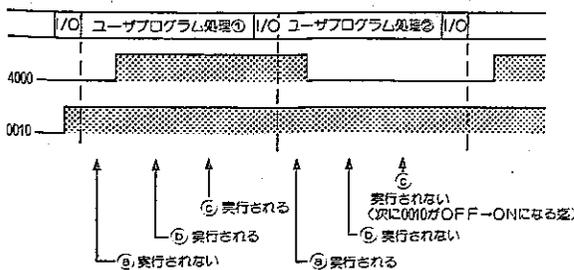
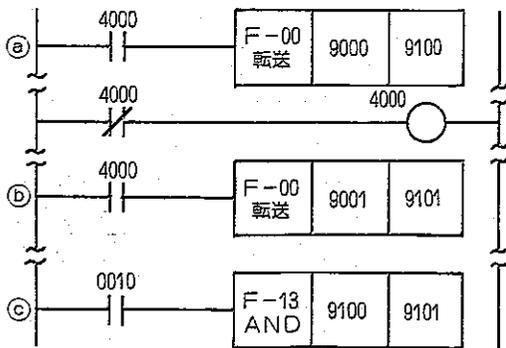


● 1スキャンごとにON/OFFを繰り返します。



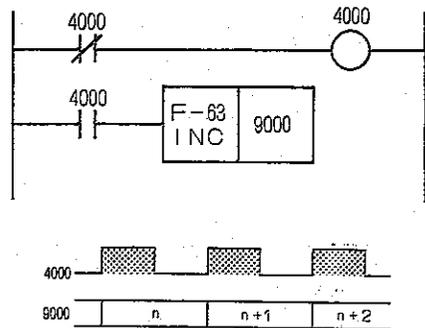
このパルスは点滅回路の基本クロックや、1スキャンおきの演算起動信号として使用します。

**注1** スキャンサイクルのどこでON→OFF、OFF→ONと変化するかは、この命令がプログラムステップのどこに書かれているかで決まります。このパルスを演算の起動信号として使う場合、注意が必要です。

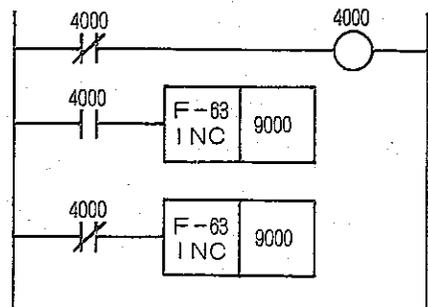


ユーザプログラム処理①のスキャンサイクルで②の演算を実行するとき、③は、このスキャンサイクルでは実行されないで、1つ前のスキャンサイクルで実行された②の演算結果が③の演算に使われてしまいます。

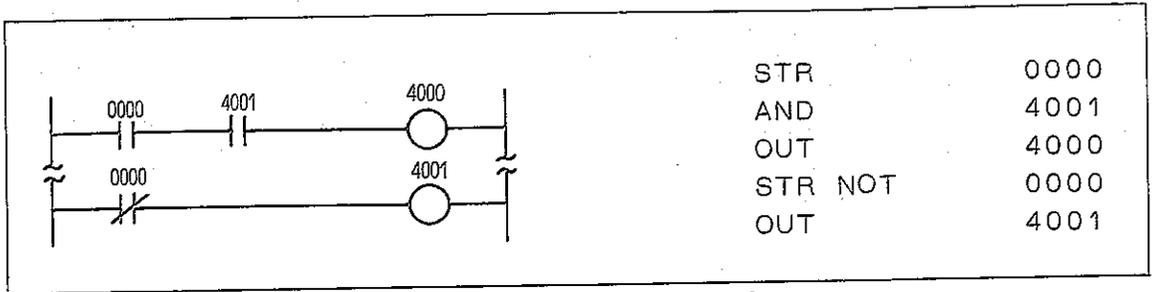
**参考** 毎スキャンサイクル演算を実行する方法  
発振回路のクロックをデータ処理命令の実行条件としてプログラムすると、1スキャンおきにしかな演算が実行されません。



毎演算サイクル演算を実行する必要がある場合、次のようにプログラムします。

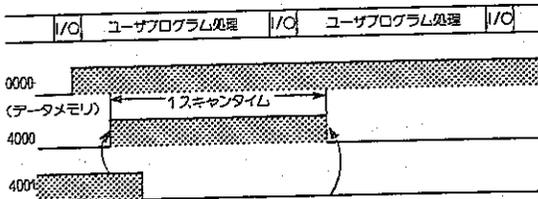


#### (4) 立上り微分

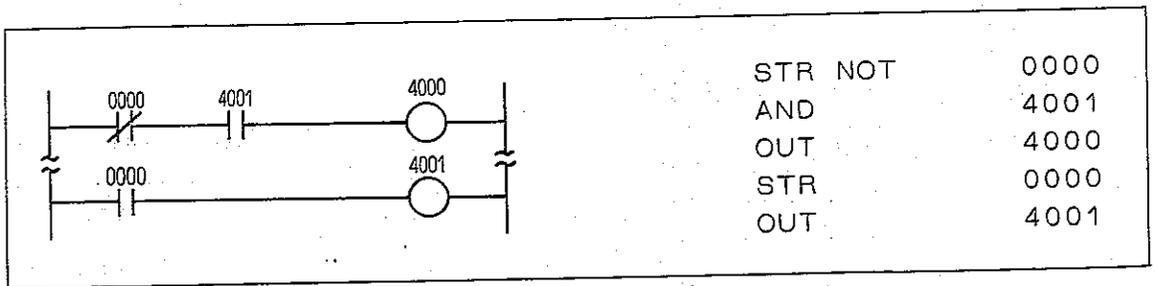


● 0000がOFF→ONとなるとき、1スキャンタイムの間4000がONとなります。

- 注1 F-44を使用すると一命令で実現できます。  
 注2 プログラム順を入換えるとバリスが発生しません。

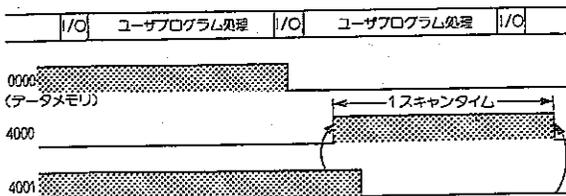


#### (5) 立下り微分

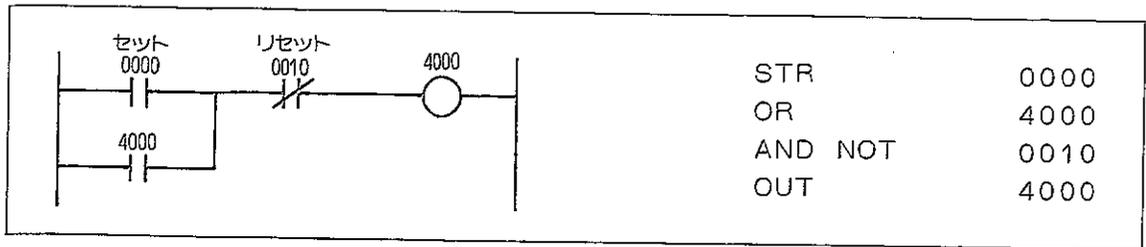


● 0000がON→OFFとなるとき、1スキャンタイムの間4000がONとなります。

- 注1 F-45を使用すると一命令で実現できます。  
 注2 プログラム順を入換えるとバリスが発生しません。

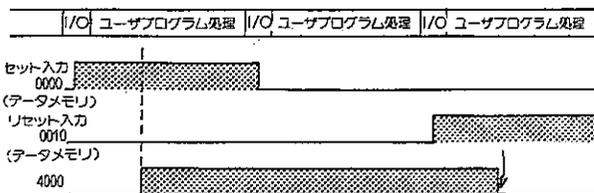


## (6) 自己保持回路 (リセット優先)

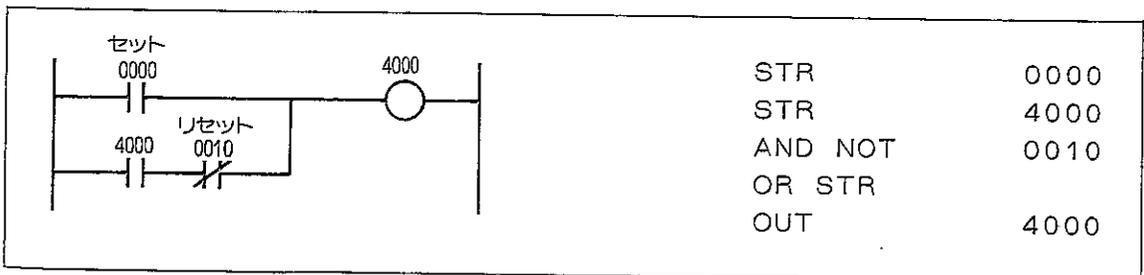


●リセット入力がOFF (ラダー図で導通) のとき、セット入力を一旦ONにすると、出力はONとなり、セット入力がOFFになってもこの状態を保持します。リセット入力をONにするか、電源を切らない限りONが継続します。

**注1** 出力として、キー機能指定領域のデータメモリを使うと、停電があっても、停電直前の状態を保持できます。  
ただしリセット入力の外部接点は a 接点を使用しプログラム上でAND NOTとしてください。  
外部接点に b 接点を使用し、プログラム上でANDを使用すると、入力用電源がW16/W51の電源より先に落ちると、自己保持がリセットされてしまいます。

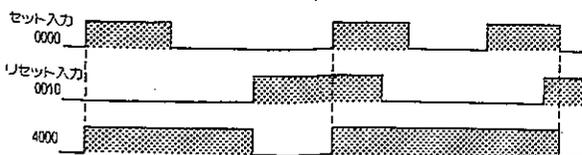


## (7) 自己保持回路 (セット優先)



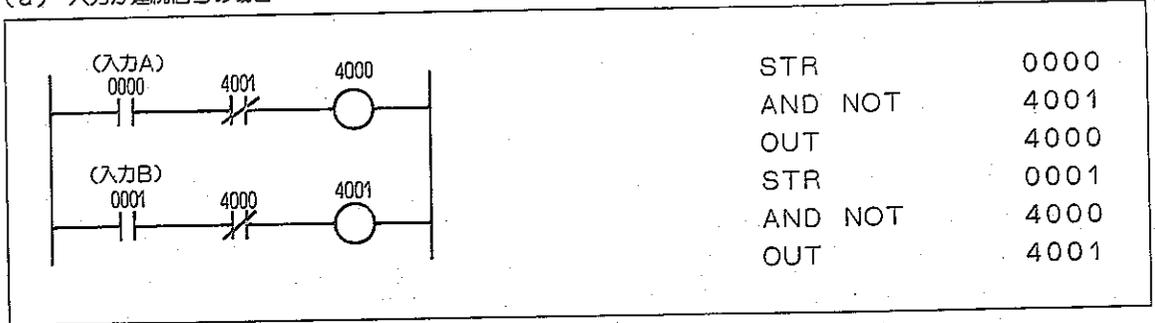
●リセット入力のON/OFFにかかわらず、セット入力を一旦ONにすると出力はONとなり、セット入力がOFFになってもこの状態を保持します。  
●セット入力がONのとき、リセット入力をON (ラダー図で非導通) にしてもリセットは無効で出力はONを保持します。  
●セット入力がOFFのときにリセットをONするか、一旦電源を切ると出力がOFFになります。

**注1** 出力としてキー機能指定領域のデータメモリを使うと、停電があっても停電直前の状態を保持できます。  
ただしリセット入力の外部接点は a 接点を使用しプログラム上でAND NOTとしてください。  
●外部接点に b 接点を使用し、プログラム上でANDを使用すると、  
① セット入力がOFFで停電したとき、入力用電源がW16/W51の電源より先に落ちると、リセット用外部接点が閉であってもリセットされます。  
② セット入力がOFFで復電したとき、入力用電源がW16/W51の電源より遅れて立上ると、リセット用外部接点が閉であってもリセットされます。



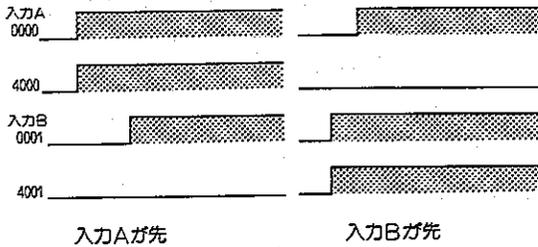
## (8) 優先回路

(a) 入力連続信号の場合

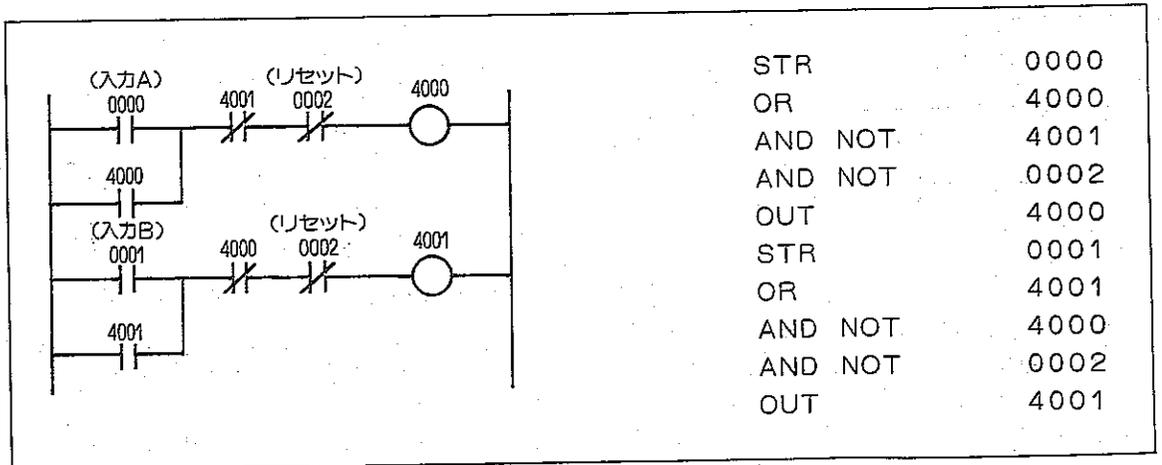


● 入力Aか入力Bのいずれか先に入った方を優先し、後で入った方の入力を無効にします。

注1 入力A、入力Bが一つのスキャンサイクルの入出力処理でONとなったとき、プログラム順が先の方が優先されます。

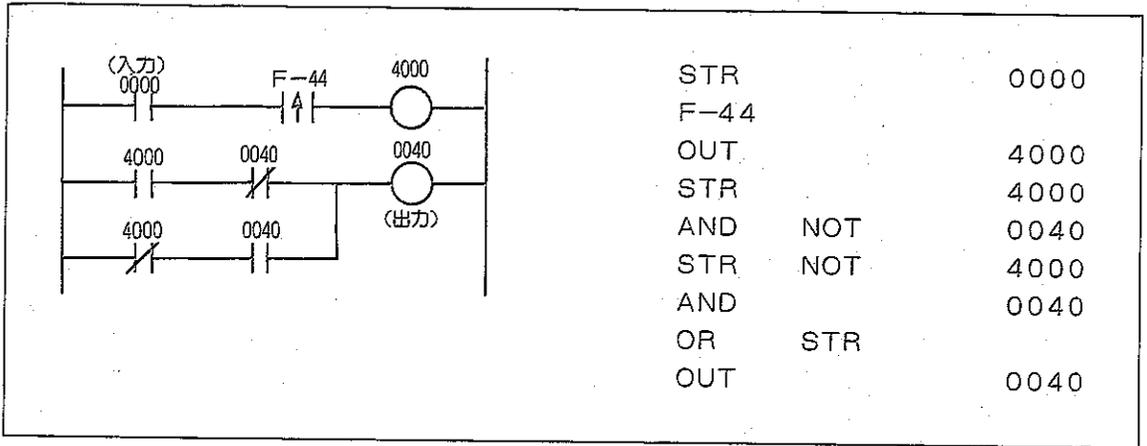


(b) 入力パルス信号の場合

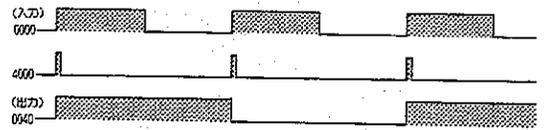


● この信号は、同時にONしては困る出力（モータの正転／逆転等）に使用します。

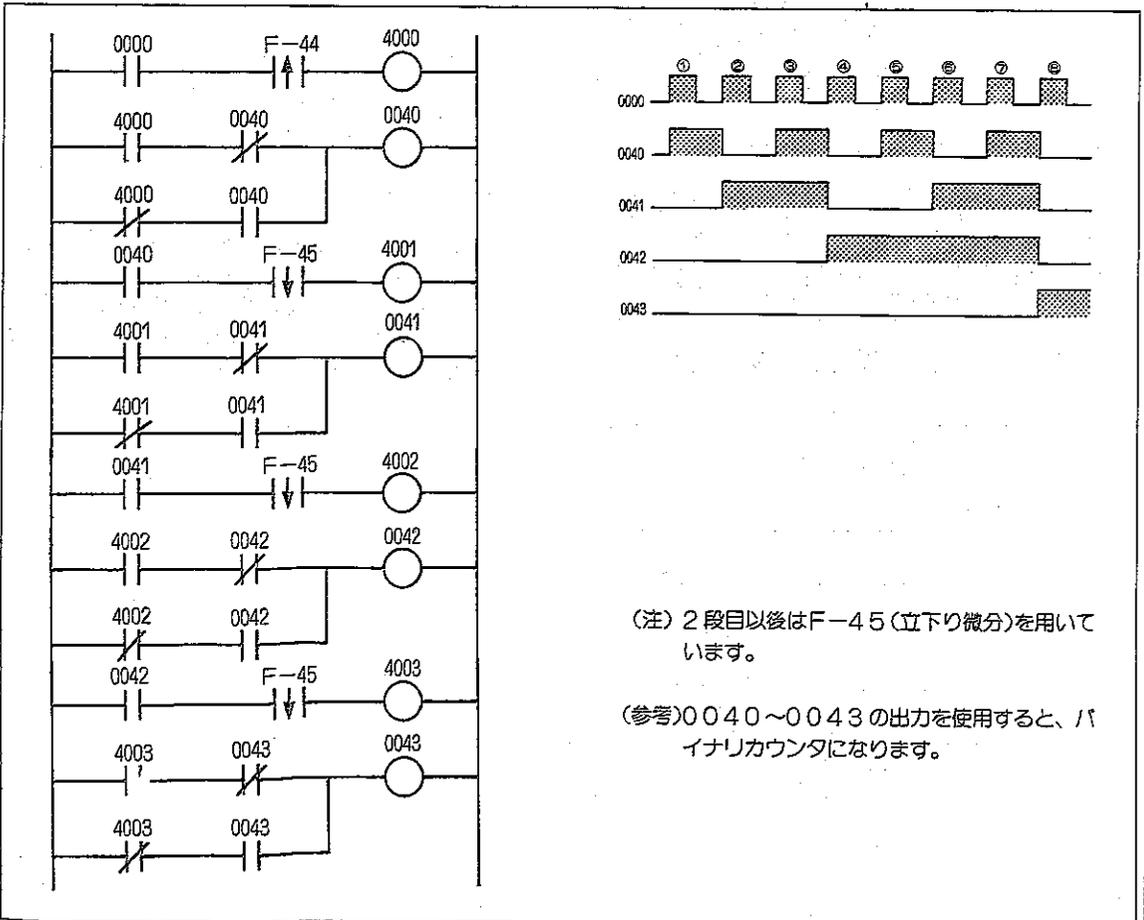
〔9〕オルタネート回路



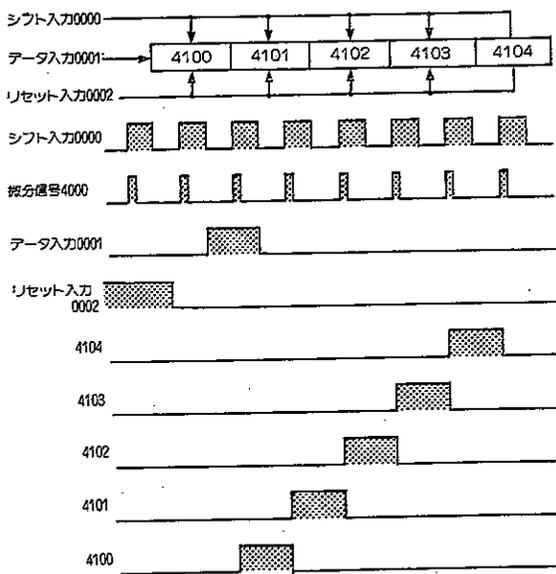
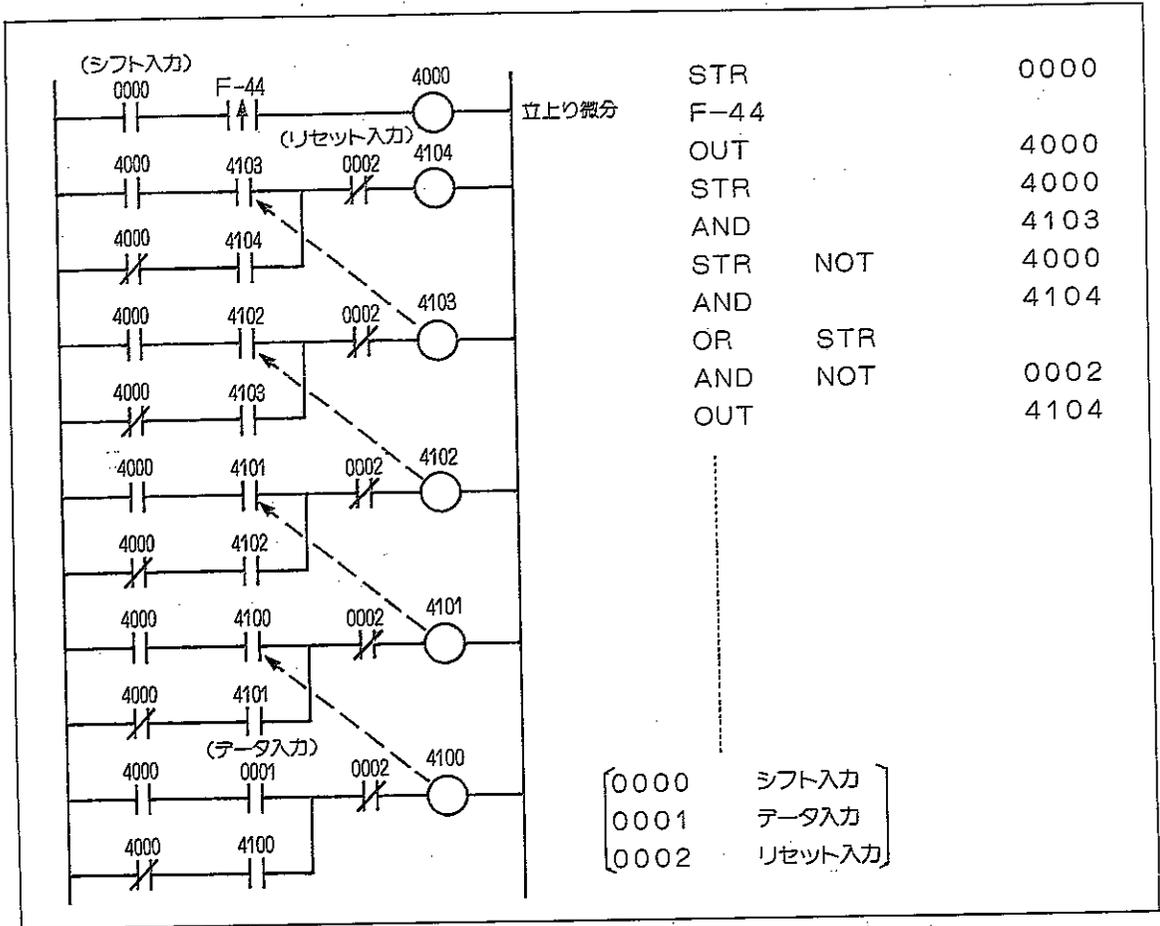
● 入力がONとなるごとに出力が反転します。モーメントリスイッチの接点を受けてオルタネート出力を取出すことができます。



● この回路を連続してn回使用するとn段の分周回路が構成できます。



# (10) nビットシフトレジスタ

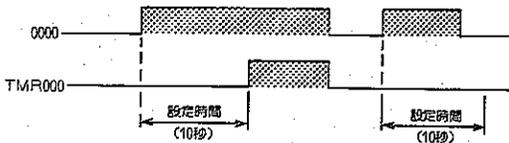
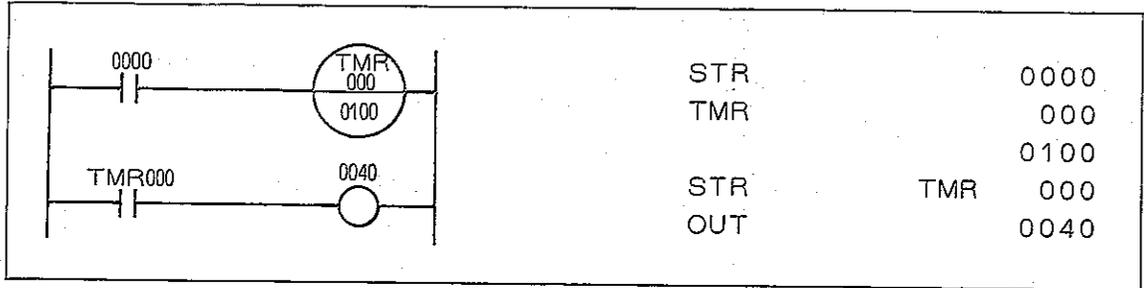


- F-60 を用いると1命令でシフトレジスタを実現できます。
- 停電時、シフト状態を保持する場合、4100~4104 はキー機能指定領域のデータメモリを使用する必要があります。



## 5-2 タイマの応用回路

### (1) オンディレイタイマ



● 入力がON後、設定時間だけ遅れて出力がONします。もし入力ONの時間が設定時間以下のとき出力はONしません。

● 入力がOFFになれば、出力もOFFとなります。

**注1** システムメモリ#201の設定により、停電時に現在値をリセットするか保持するかの選択ができます。

#201の設定	{	000 <sup>(a)</sup> 現在値をリセット (設定値になる)
		001 <sup>(b)</sup> 現在値を保持

**注2** 入力がONのとき停電があると#201の設定状態により、復電時の出力の状態が異なります。

(a) #201が000<sup>(a)</sup>——現在値リセット  
復電後、設定時間だけ遅れて出力がONとなります。

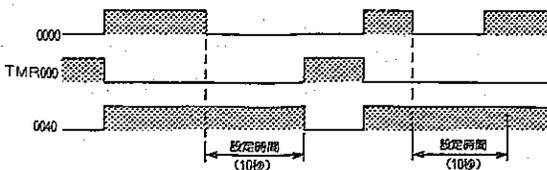
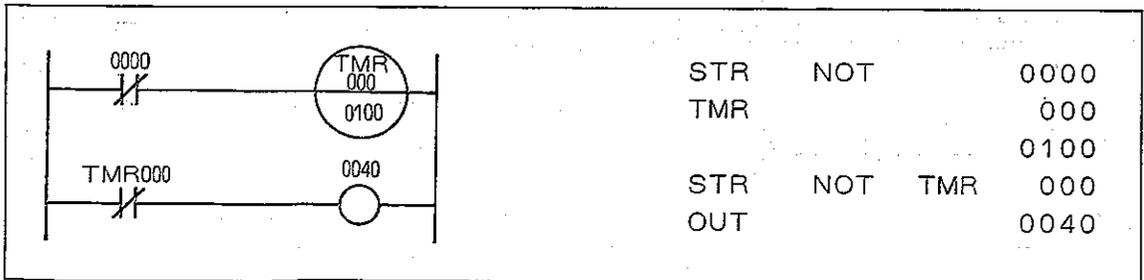
(b) #201が001<sup>(b)</sup>——現在値保持

(b)-1 停電前にタイムアップしていたとき  
復電後、最初のスキャンで出力がONとなります。

(b)-2 停電前にタイムアップしていなかったとき

復電後、(設定値-停電時の現在値)の時間だけ遅れて出力がONになります。

### (2) オフディレイタイマ



● 入力がOFF後、設定時間だけ遅れて出力がOFFとなります。もし入力OFFの時間が設定時間以下のとき出力はOFFになりません。

● 入力がONになれば、出力もONになります。

**注1** 入力がOFFのとき(タイマの入力はON)停電が

あると、システムメモリ#201の設定状態により、復電時の出力の状態が異なります。

(a) #201が000<sup>(a)</sup>——現在値リセット  
復電時、設定時間だけ出力がONします。

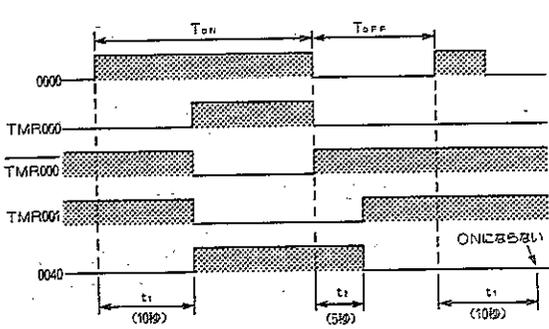
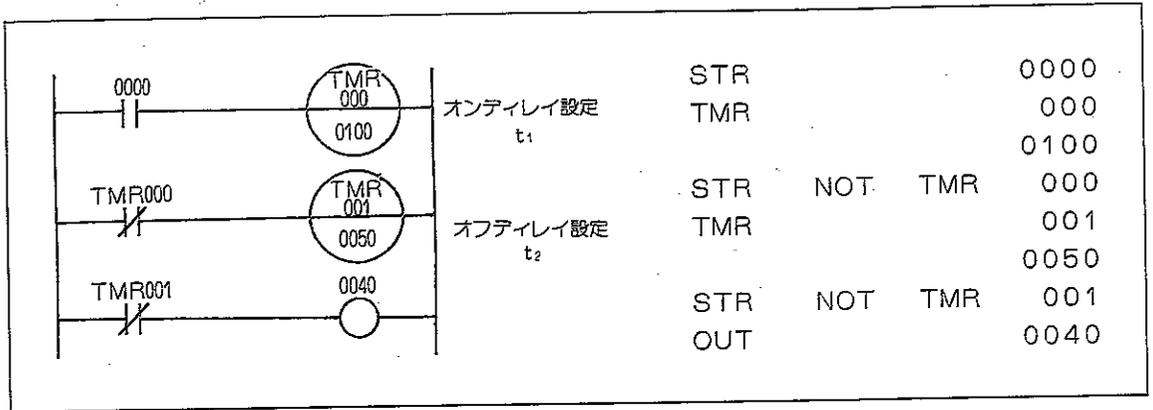
(b) #201が001<sup>(b)</sup>——現在値保持

(b)-1 停電前にタイムアップしていたとき  
復電時、出力はONしません。

(b)-2 停電前にタイムアップしていなかったとき

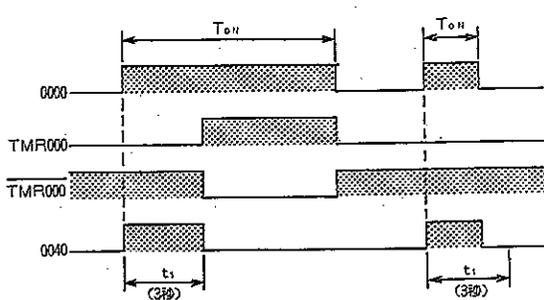
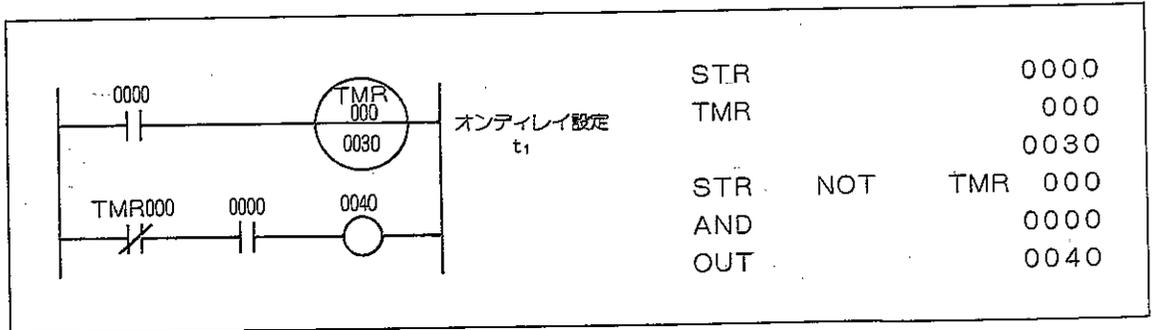
復電時、(設定値-停電時の現在値)の時間だけ出力がONします。

### (3) オン・オフディレイタイマ



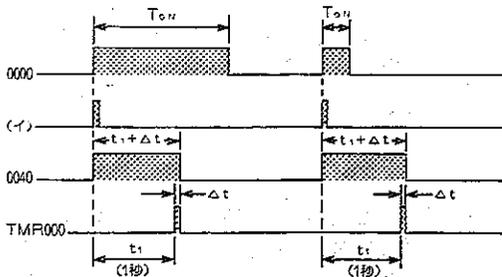
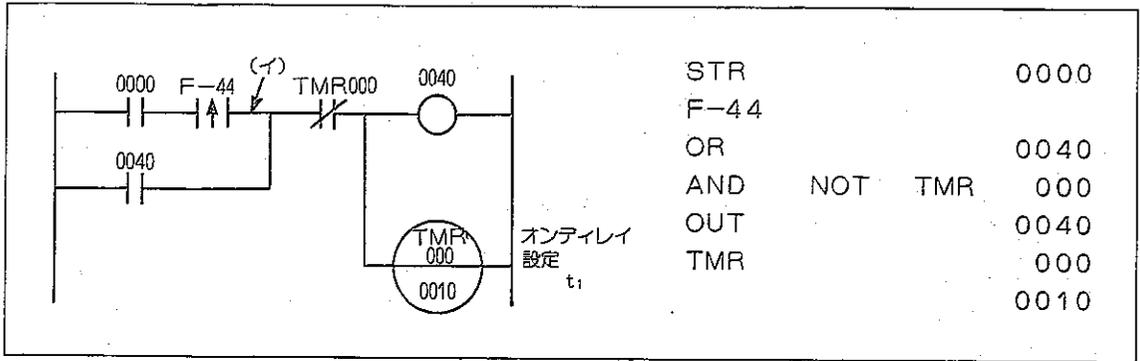
- 入力ONしてから $t_1$ だけ遅れて出力がONし、入力がOFFしてから $t_2$ だけ遅れて出力がOFFになります。
  - 入力のONの時間( $T_{ON}$ ) < オンディレイ設定時間( $t_1$ ) のとき、出力はONになりません。
- 注1** 停電時の出力の状態については〔1〕オンディレイタイマ、〔2〕オフディレイタイマの注意事項をご参照ください。

### (4) 入力立上り時ワンショットタイマ(1)



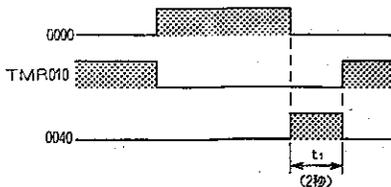
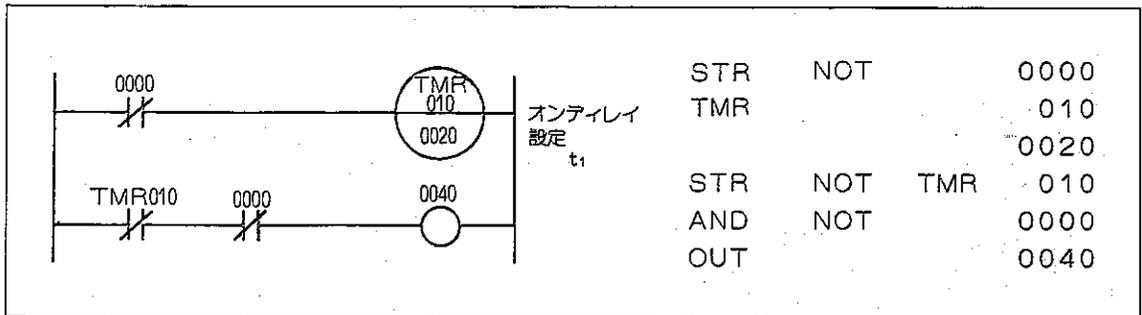
- 入力の立上り(OFF→ON)時に設定時間 $t_1$ の中のパルスが出力されます。
- 入力のONしている時間( $T_{ON}$ ) < 設定時間( $t_1$ ) のとき、出力パルスの中は $T_{ON}$ となります。

(5) 入力立上り時ワンショットタイム(2)



- 入力の立上り(OFF→ON)時に(設定時間  $t_1 + \Delta t$ ) の中のパルスが出力されます。  
 $\Delta t$  —— 1 スキャンタイム
- 入力のONしている時間( $T_{ON}$ )に関係なく、出力のパルス巾は  $t_1 + \Delta t$  となります。

(6) 入力立下り時ワンショットタイム

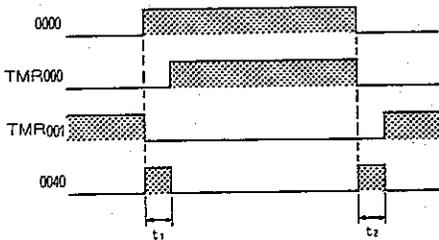
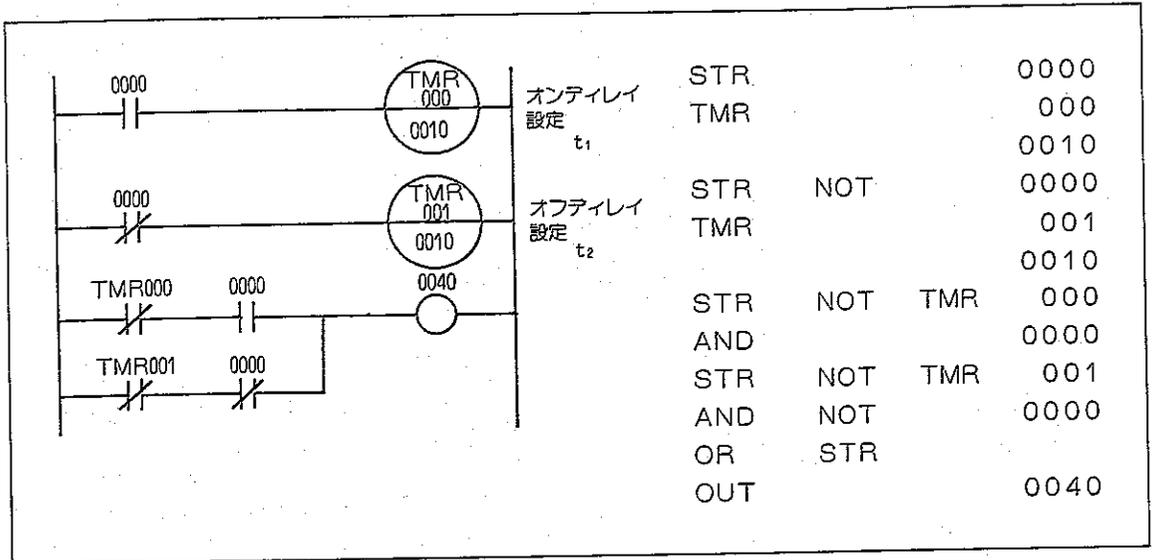


- 入力の立下り(ON→OFF)時に設定時間( $t_1$ ) の中のパルスが出力されます。

注1 入力がOFFのとき停電があるとシステムメモリ #201 の設定状態により、復電時の出力の状態が異なります。

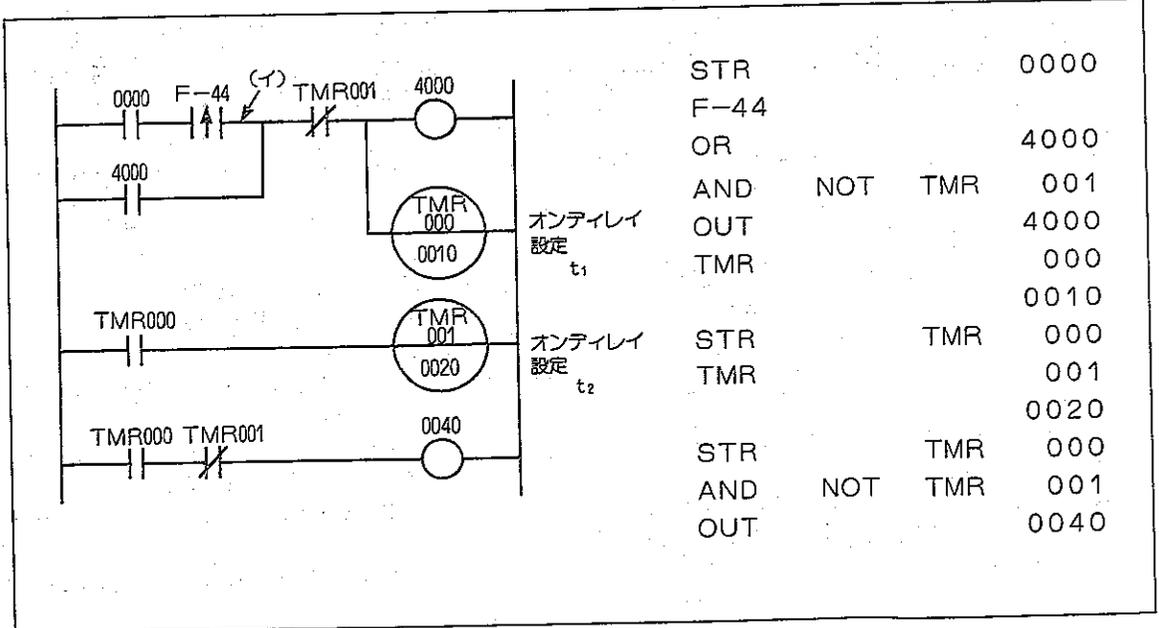
- (a) #201 が 000 (a) —— タイマの現在値  
リセット  
復電時、設定時間だけ出力がONします。  
このパルスは電源投入時のイニシャライズ用に使  
用できます。
- (b) #201 が 001 (a) —— タイマの現在値  
保持  
(b-1) 停電前にタイムアップしていたとき  
復電時、出力はONしません。  
(b-2) 停電前にタイムアップしていなかったと  
き  
復電時、(設定値 - 停電時の現在値)だけONします

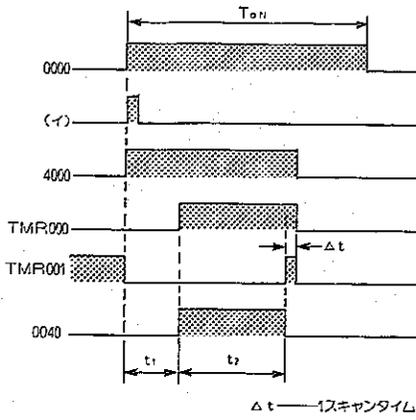
(7) 立上り、立下りワンショットタイマ



- 入力の立上り、立下りにそれぞれ  $t_1$ 、 $t_2$  のパルス中だけ出力がONします。
- 入力が状態変化があった場合、これを検出する信号として使用します。
- 入力パルスの周波数を過倍する場合にも使用できます。

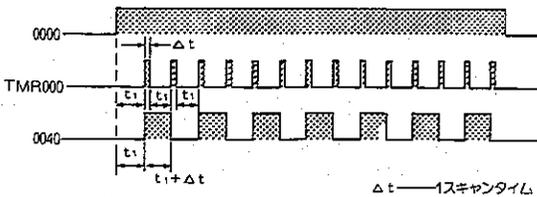
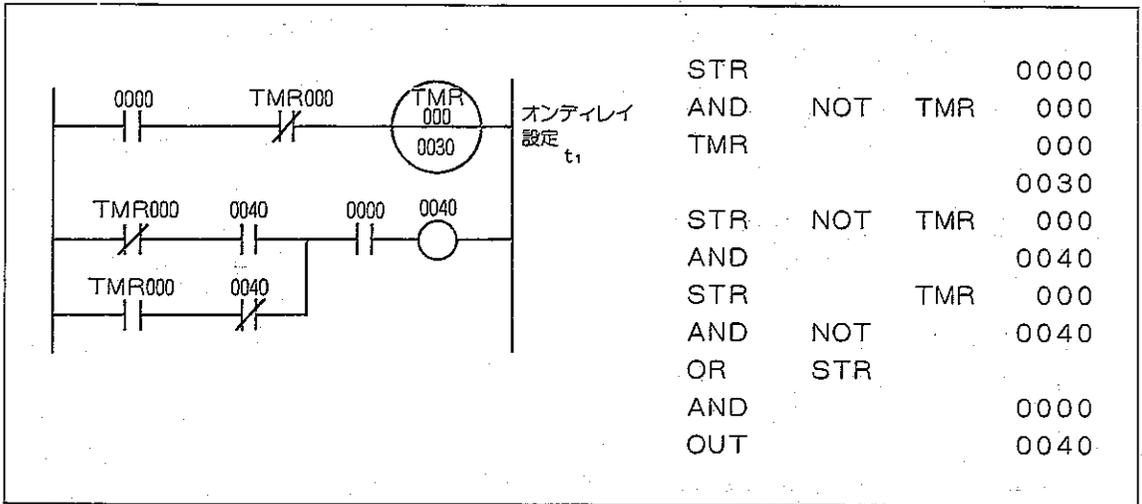
(8) オンティレイワンショットタイマ





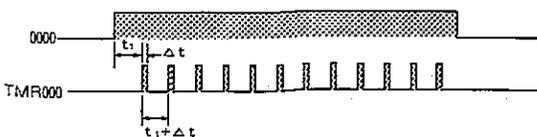
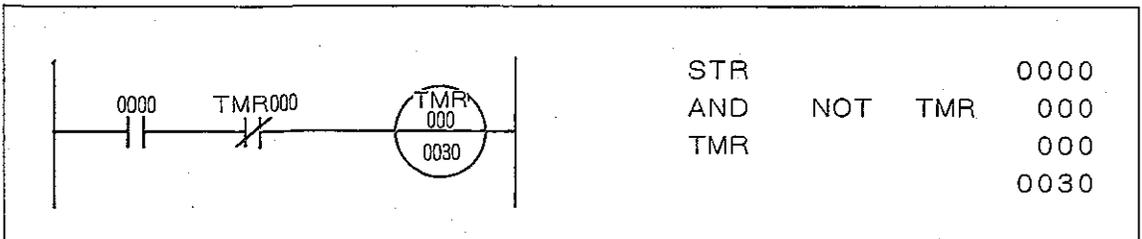
- 入力の立上りから設定値  $t_1$  だけ遅れて、パルス中  $t_2$  のパルスが出力されます。
- $(t_1 + t_2)$  の時間中に入力が ON/OFF しても無視されます。

### 〔9〕 等間パルス発生回路

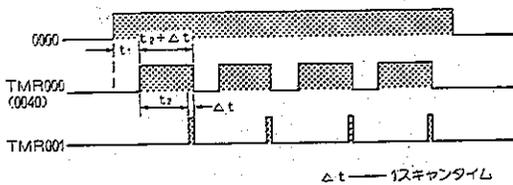
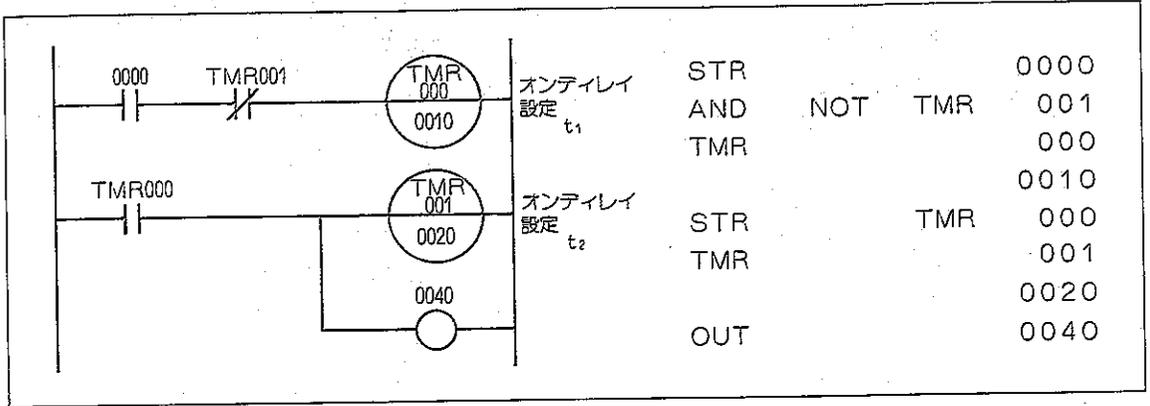


- 入力が ON の間、ON/OFF の時間が等しい (デューティサイクル 50%) パルスが出力されます。ON/OFF の時間は TMR の設定値 ( $t_1$ ) で任意に設定することができます。パルス中は  $t_1 + \Delta t$  となります。

【参】 ON 時間  $\Delta t$ 、OFF 時間  $t_1$  のパルスは下図で得られます。

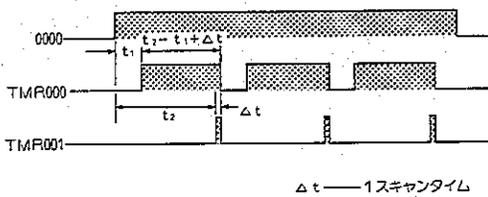
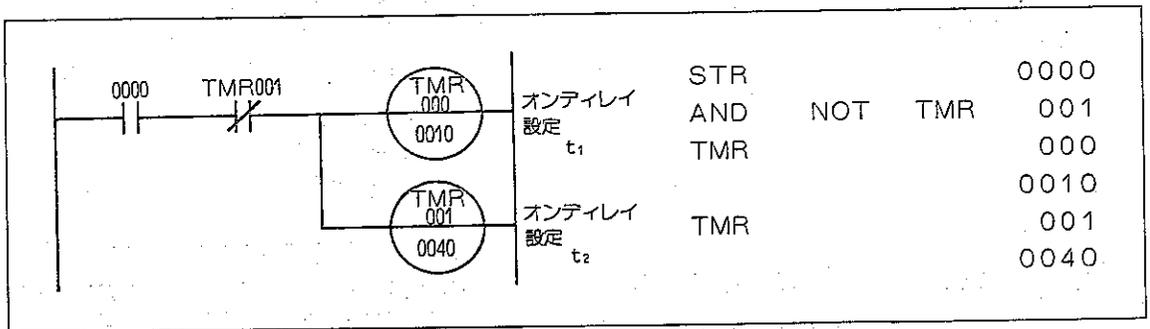


(10) デューティ可変パルス発生回路(1)



●入力がONの間、ON時間 ( $t_2 + \Delta t$ )、OFF時間 ( $t_1$ ) のパルスが発生します。

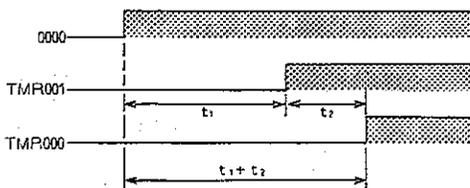
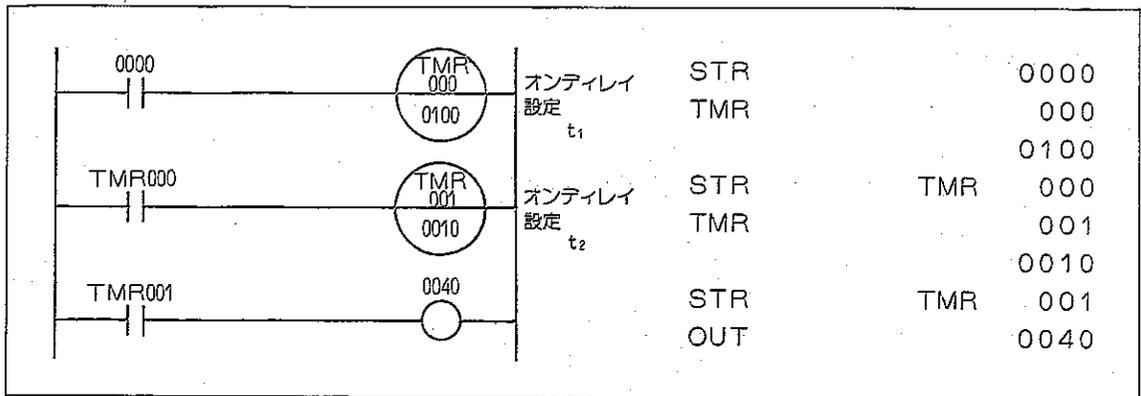
(11) デューティ可変パルス発生回路(2)



●入力がONの間、ON時間 ( $t_2 - t_1 + \Delta t$ )、OFF時間 ( $t_1$ ) のパルスが発生します。

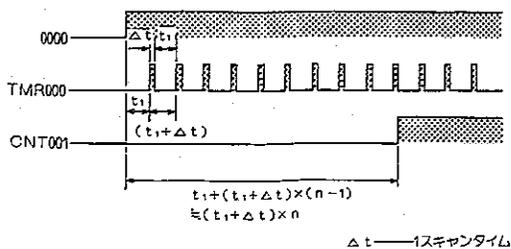
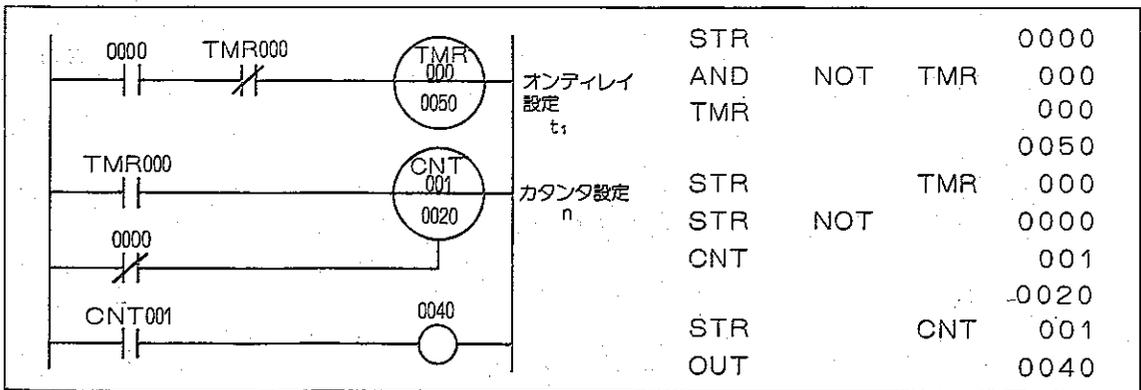
注1 必ず  $t_1 < t_2$  と設定してください。

(12) 長時間タイマ(1)



●入力ON後、 $t_1 + t_2$ 遅れて出力がONとなります。

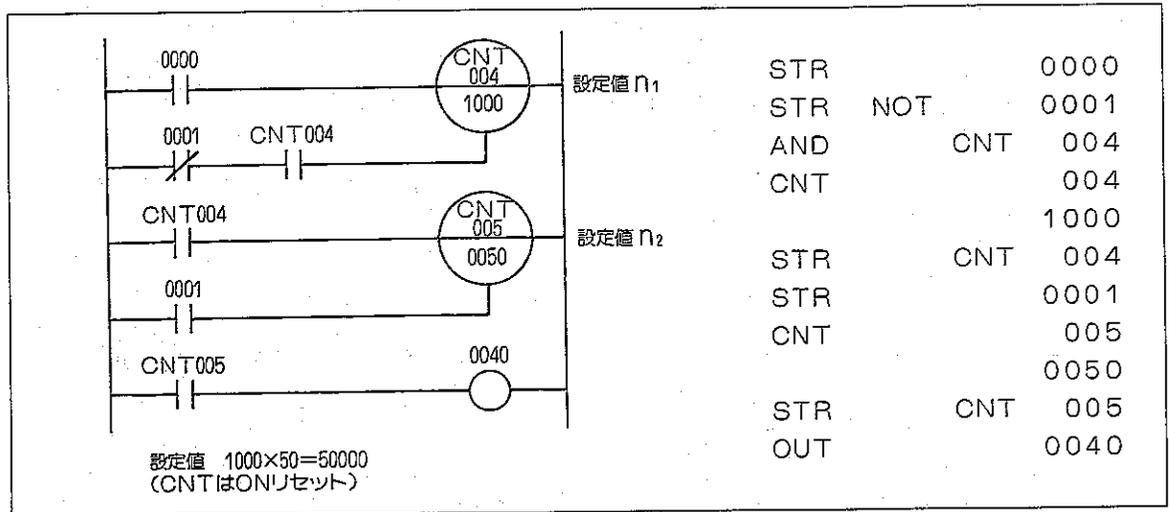
(13) 長時間タイマ(2)



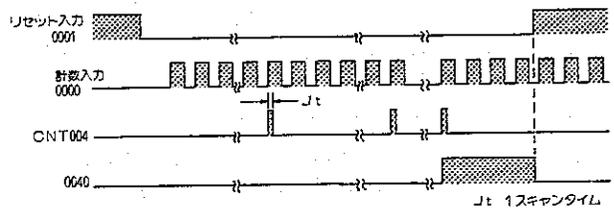
●入力ON後 $(t_1 + \Delta t) \times n$ 遅れて出力がONになります。

## 5-3 カウンタの応用

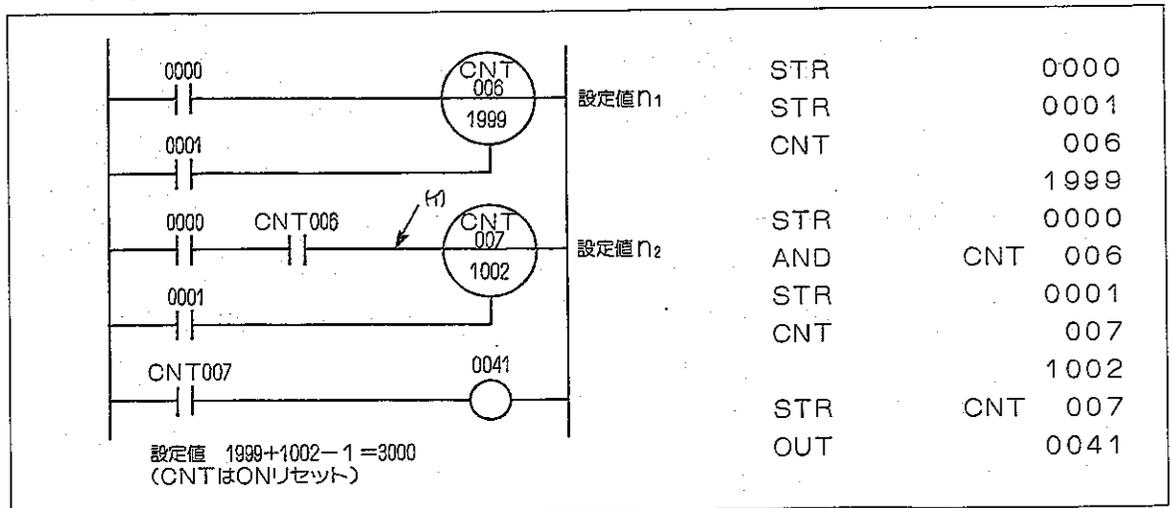
### (1) 大容量カウンタ(1)



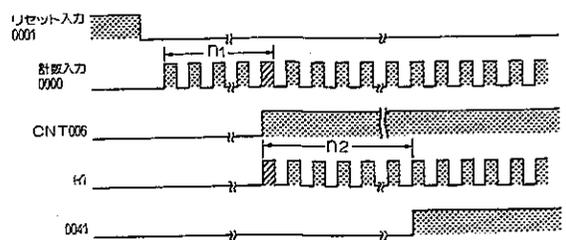
●設定値が1999を越える場合、上図の如くプログラムすると、設定値 ( $n_1 \times n_2$ ) のカウンタを実現できません。



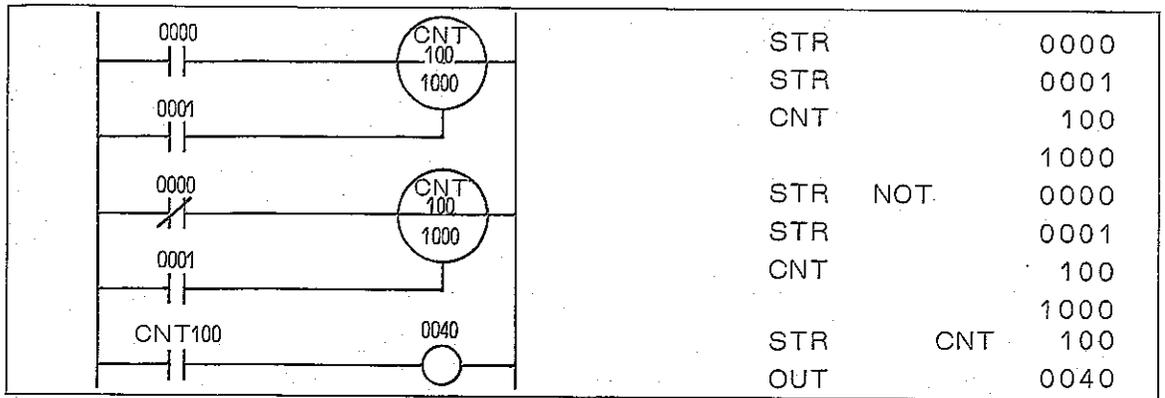
### (2) 大容量カウンタ(2)



●設定値 ( $n_1 + n_2 - 1$ ) のカウンタとなります。

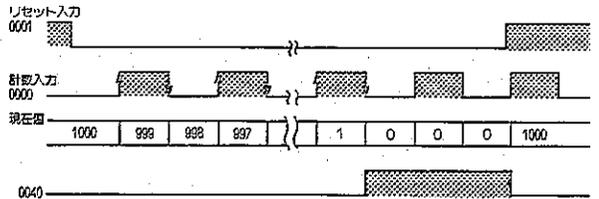


### (3) 計数入力の立上り、立下りで計数するカウンタ



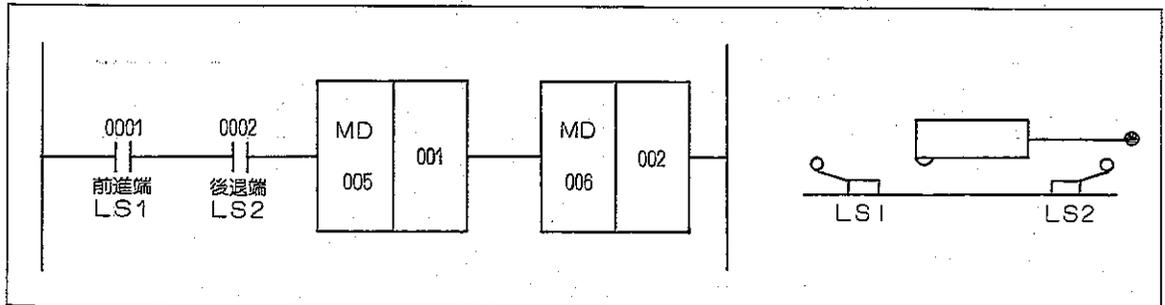
●計数入力がOFF→ONに変化したとき、ON→OFFに変化したときのいずれの場合も減算するカウンタです。

注1 プログラム(ZW-101PG1)等でプログラムチェックすると“DOUBLE NUMBER”と表示されますが、本例の様に意図的に同一番号を使用する場合、この警告は無視してもかまいません。



## 5-4 メンテナンスディスプレイ(MD)の応用

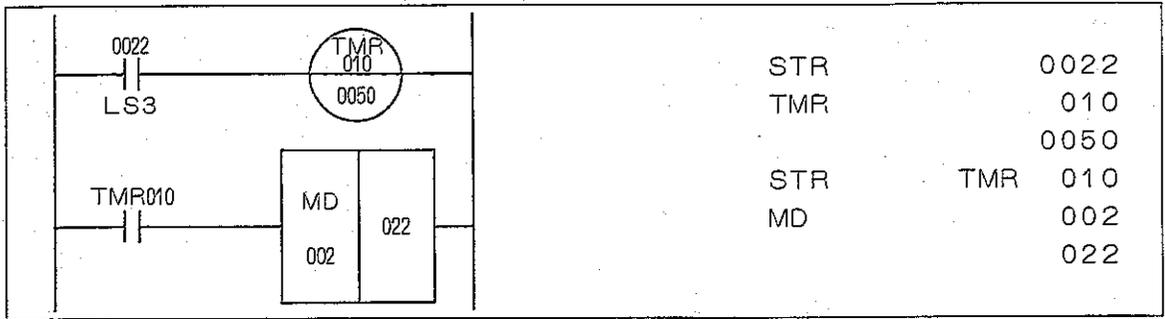
### (1) 排他的入力の異常検知



- 前進端リミットスイッチLS1(0001)と後退端リミットスイッチLS2(0002)が同時にONしない条件で、どちらかが溶着等の事故で同時にONした場合に記憶表示します。
- このプログラムでは、MDを2段に拡張して使用していますが、これは出力指示端子0001を出力データ

001に、0002を002にそれぞれ対応させ、該当するリミットスイッチの故障(この場合はLS1あるいはLS2)を判断できるようにしています。ただし、この場合、スタックS<sub>1</sub>、S<sub>2</sub>、S<sub>3</sub>の状態は無意味です。

## (2) 入力機器のON時間の異常検知

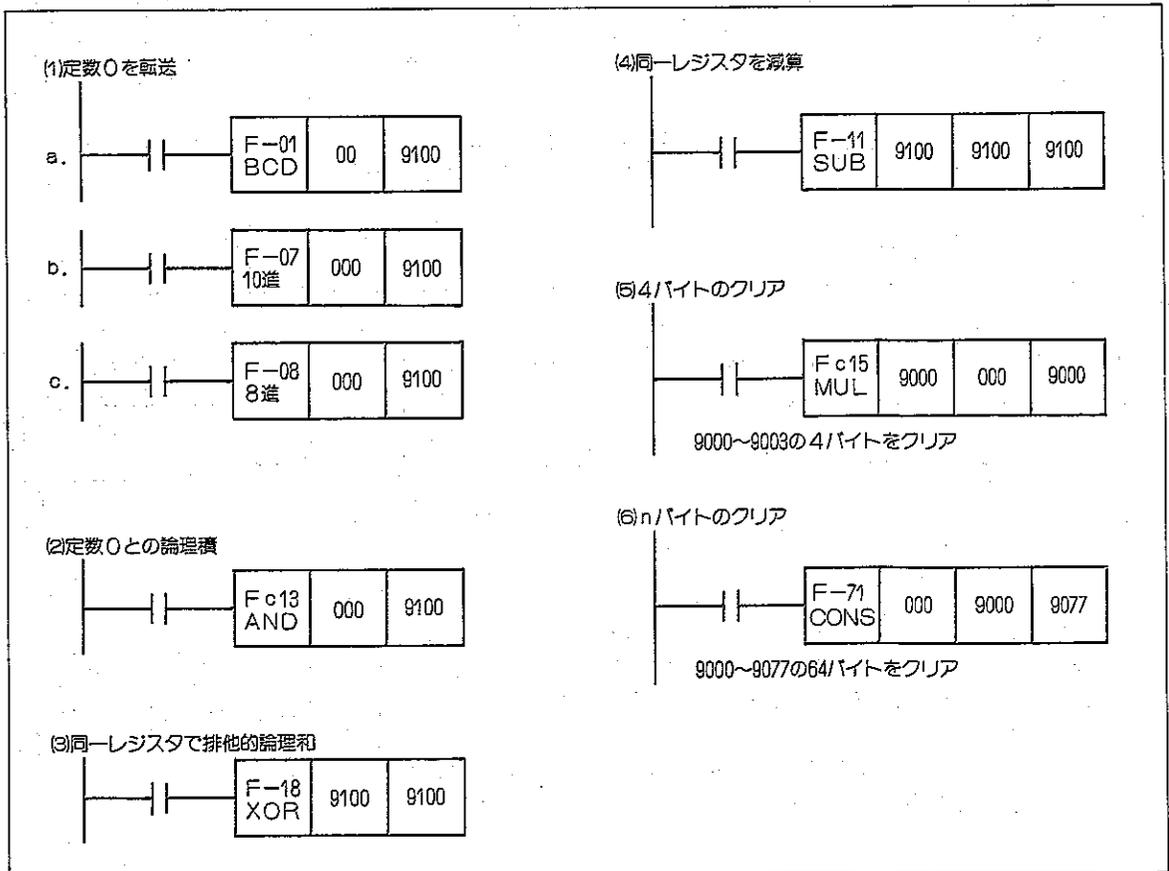


● 正常なシーケンスで、リミットスイッチLS3 (0022) のON時間は5.0秒未満とします。LS3のON時間が5.0秒以上になった場合にMDの現在値

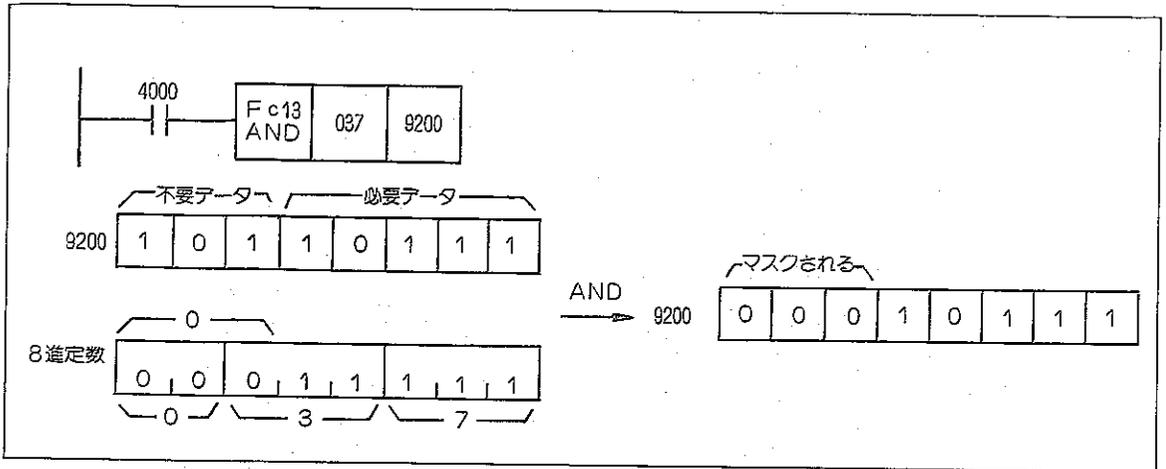
022を表示します。ただし、この場合、スタックS<sub>1</sub>、S<sub>2</sub>、S<sub>3</sub>の状態は無意味です。

## 5-5 データ処理命令の応用

### (1) レジスタのクリア

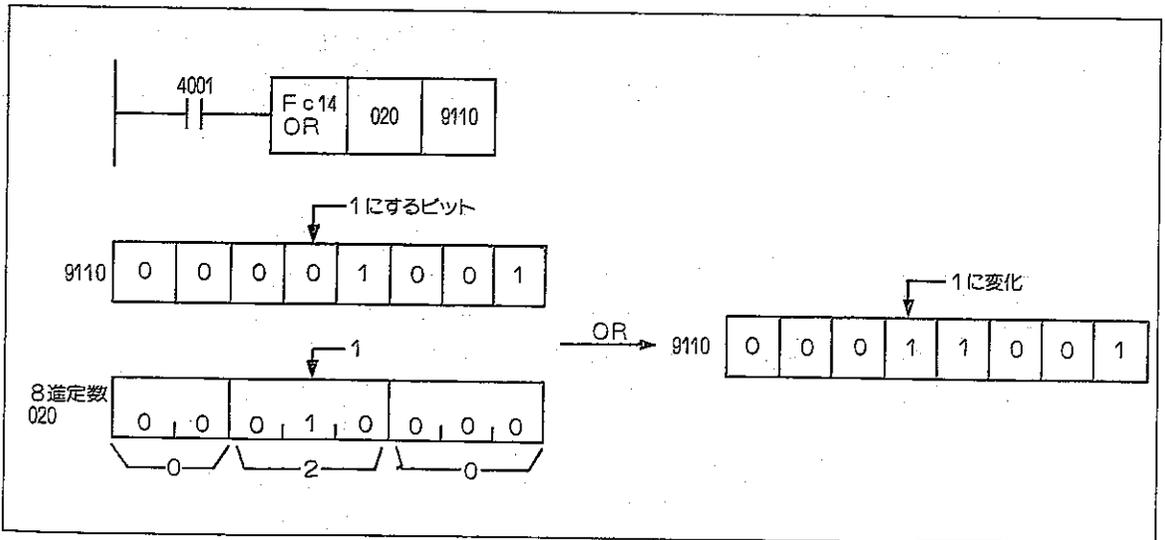


## (2) データのマスク



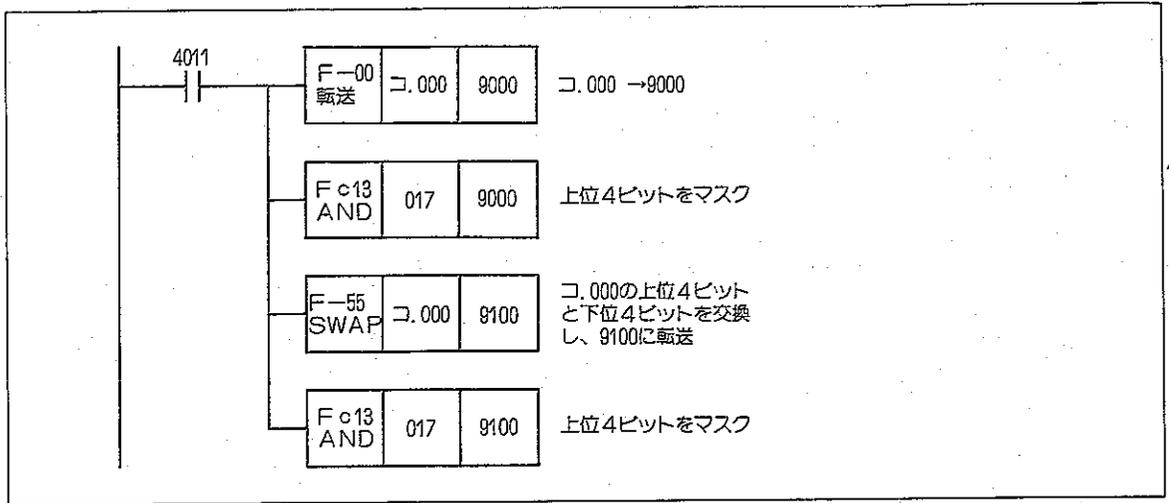
- 8ビットデータのうちの任意のビットを0にします。

## (3) ビット挿入

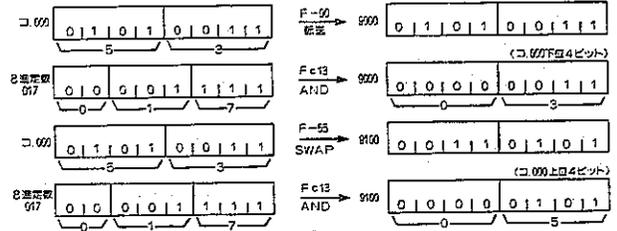


- 8ビットデータのうちの任意のビットを1にします。

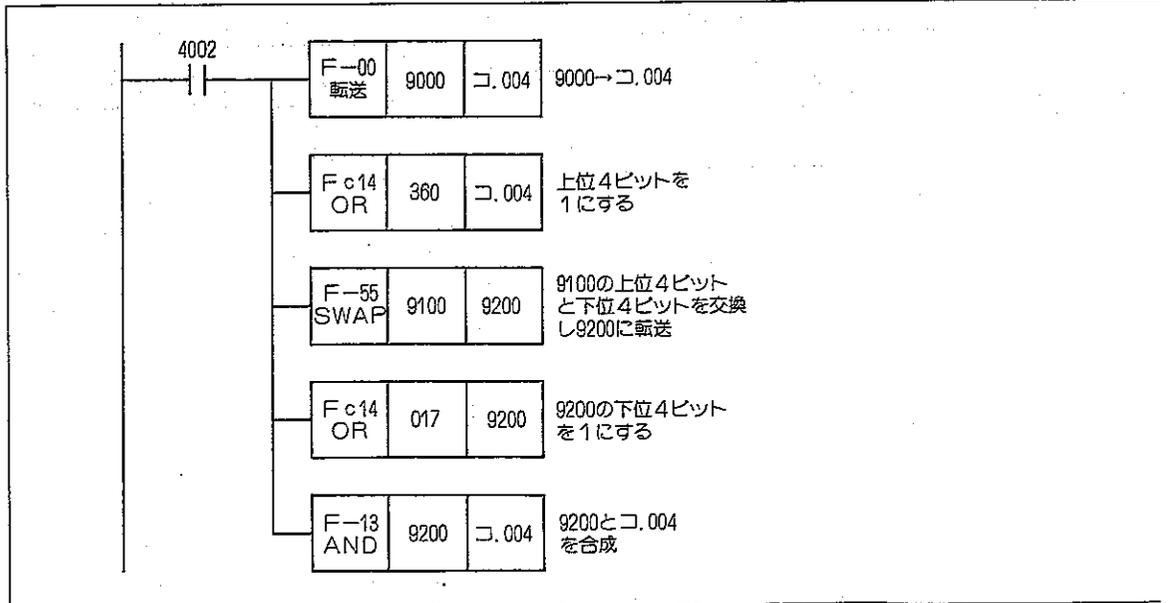
#### 〔4〕 数の分解



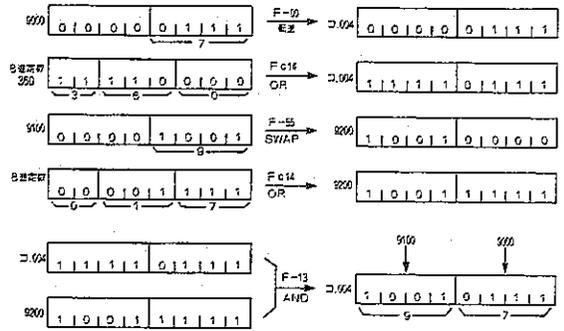
- 入力ユニットから入力されるBCD2桁の数値を1桁ずつに分解し別々のレジスタに格納します。  
例では、コ.000の下位4ビットを9000に、上位4ビットを9100に転送しています。
- 複数のBCD1桁の数値を入力する場合、2桁分まとめて入力ユニットを使用すると、入力ユニットの節約ができます。



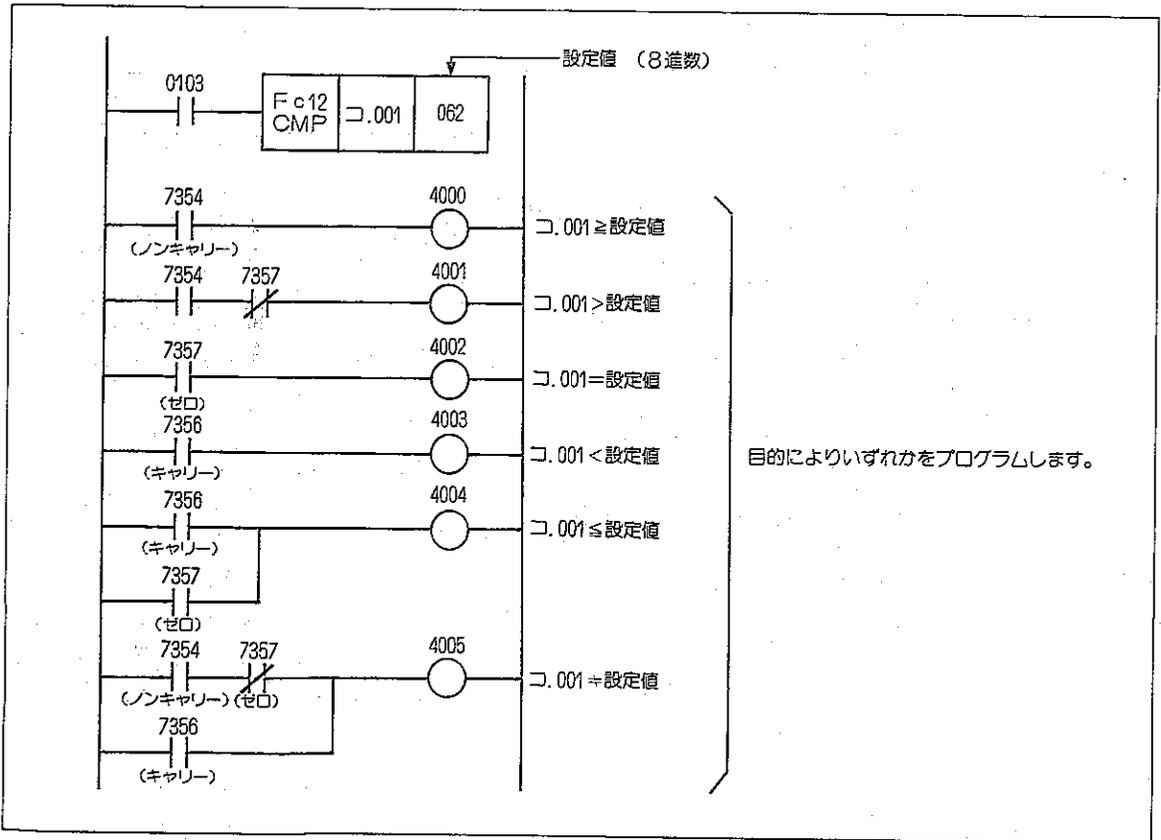
#### 〔5〕 数の合成



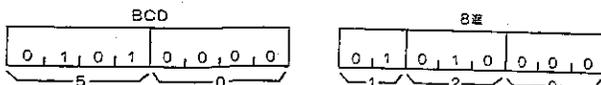
- 2つのレジスタ内のBCD1桁の数値を1つの出力ユニットにまとめて出力します。  
例では9000と9100のそれぞれの下位4ビットをコ.004に出力しています。
- 複数のBCD1桁の数値を出力する場合、2桁まとめて出力ユニットを使用すると、出力ユニットの節約ができます。



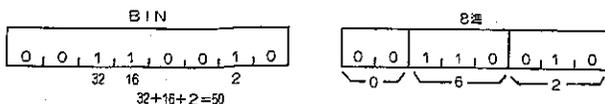
## 〔6〕 設定値との比較



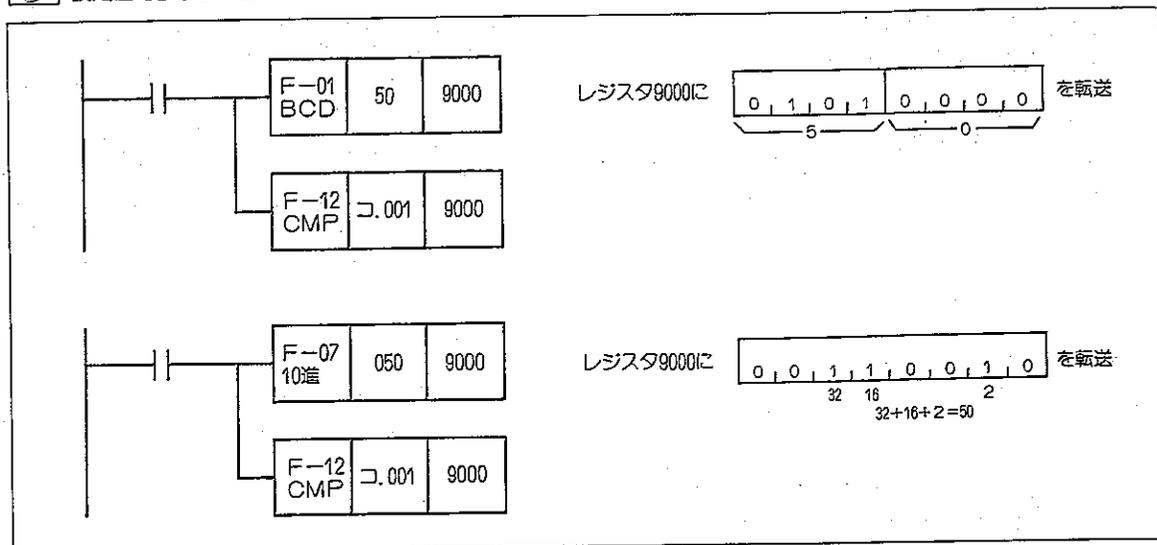
- 設定値は8進数でプログラムします。
- (例1) レジスタ内のデータがBCDの場合 (設定値50 (BCD))



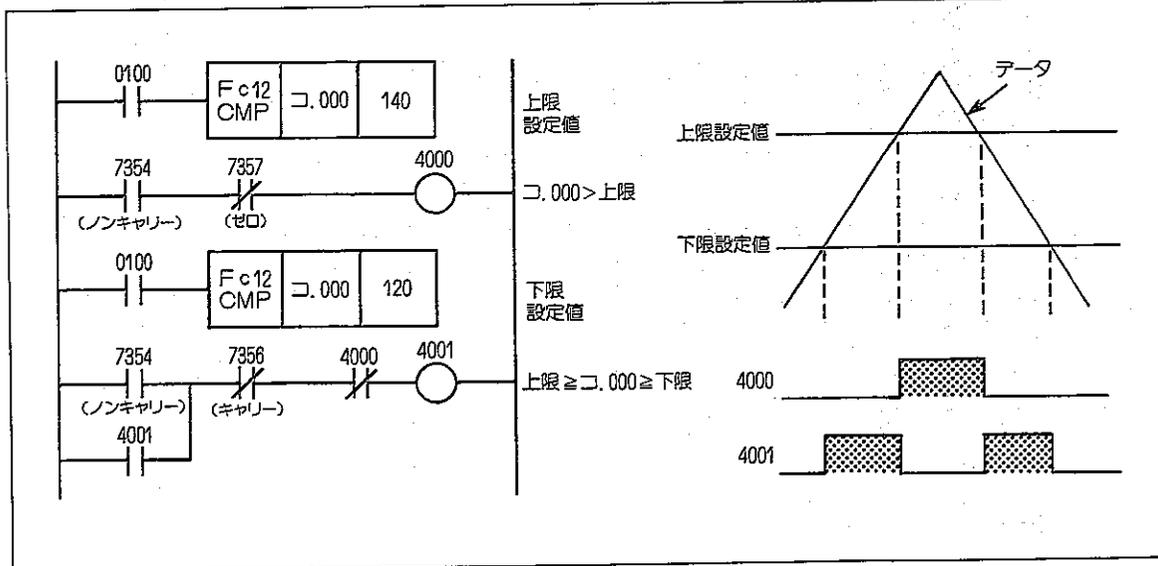
- (例2) レジスタ内のデータがBINの場合 (設定値50 (BIN))



参 設定値をBCD、BINでプログラムする方法



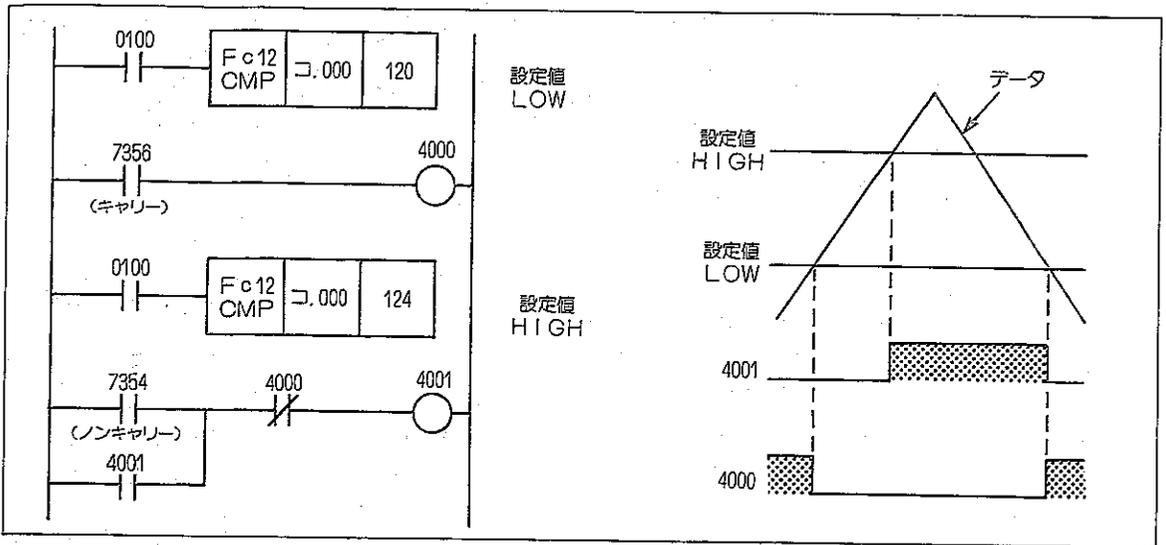
(7) ウィンドウコンパレータ



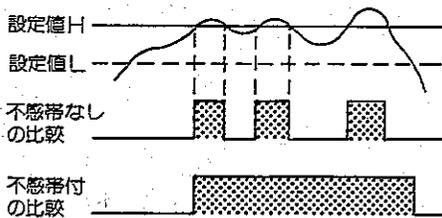
●データが下限設定値と上限設定値の間にあるとき、4001がONします。良品と不良品の判別 (GO/NG) に使用します。

●上記の例では上限設定値を140 (BCDで60) 下限設定値を120 (BCDで50) としています。4001がONとなるのは  $60 \geq 0.000 \geq 50$  となります。

〔8〕 不感帯をもつ比較回路

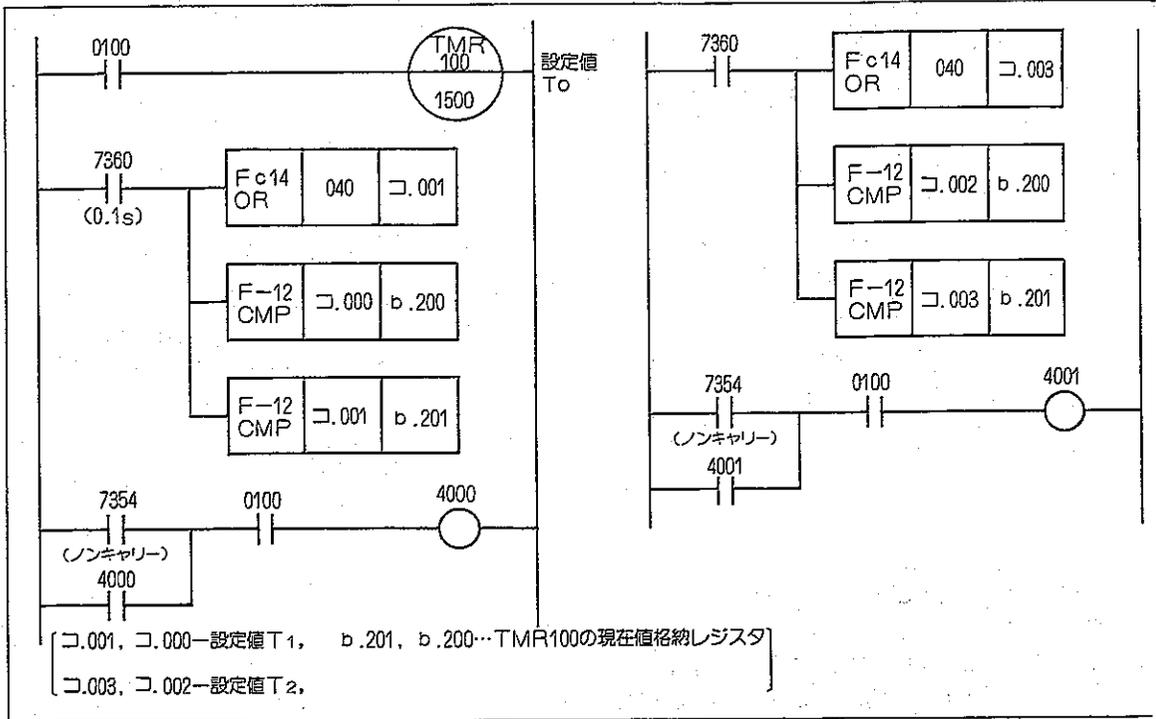


- A/D変換器の出力を入力ユニットに取り込み比較する場合、アナログ信号のわずかな変動でA/D変換器の出力が変化し、シーケンサーでの比較結果がON/OFFすることがあります。本プログラムを使用すると、A/D変換器の下位ビットのフラツキによる影響を受けない比較が行なえます。



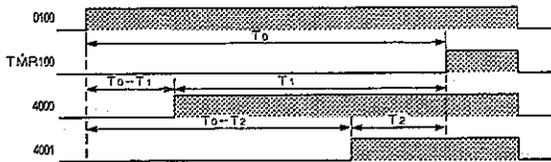
- 上記の例ではHIGH側設定値を124<sub>10</sub> (BCDで54)、LOW側設定値を120<sub>10</sub> (BCDで50)としています。コ.000の内容 (BCDコードとする)  $\geq 54$  のとき、4001はONとなりますが、一旦ONになるとコ.000の内容  $< 50$  まではONを維持します。

(9) 複数のセットポイントを持つタイマ



● TMRの現在値とレジスタ（または入力ユニット）の設定値を比較することにより、複数のセットポイントを持つタイマが実現できます。

注1 b. xxx, b. xxx+1には次のようなフォーマットでTMRの現在値が格納されています。



	7	6	5	4	3	2	1	0
b. xxx	(x10 <sup>2</sup> )			(x10 <sup>-1</sup> )				
	*8*	*4*	*2*	*1*	*8*	*4*	*2*	*1*
b. xxx+1	7	6	5	4	3	2	1	0
	OFF	OFF	*	(x10 <sup>2</sup> )	(10 <sup>1</sup> )			
				1	8	4	2	1

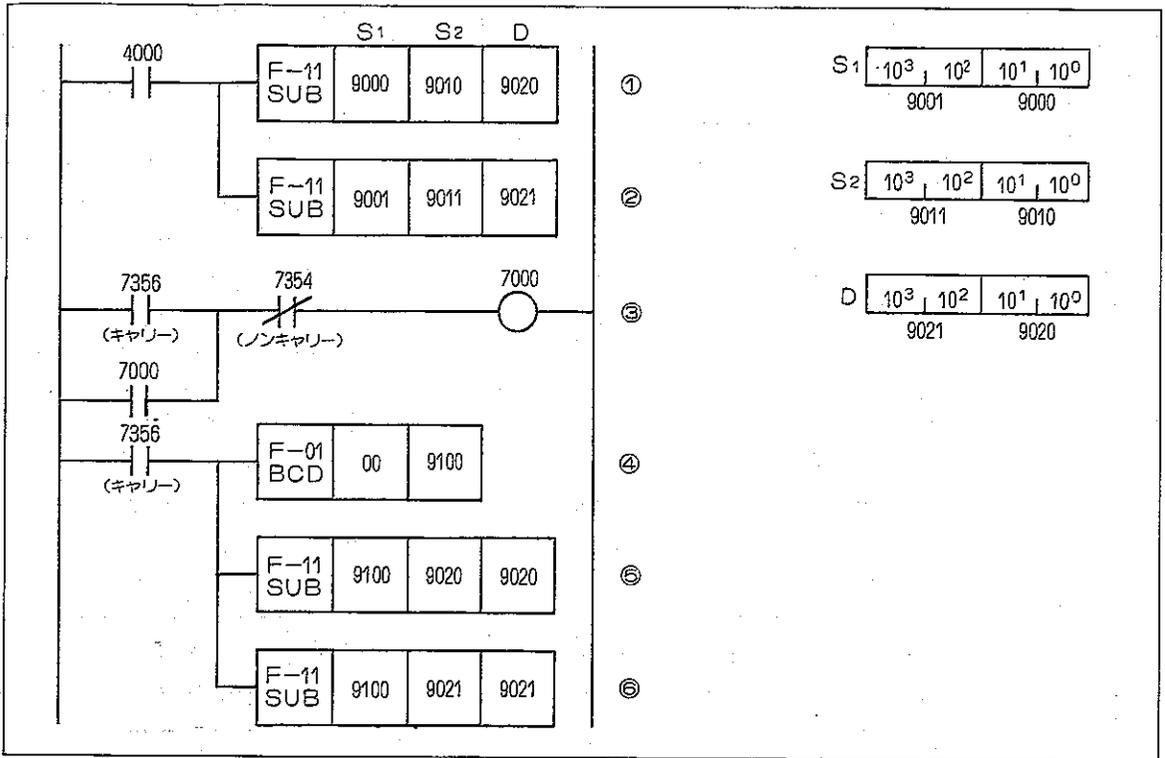
b. xxx+1の※のビットは1でタイマが動します。

Fc14で8進定数040

0,0 1,0,0 0,0,0

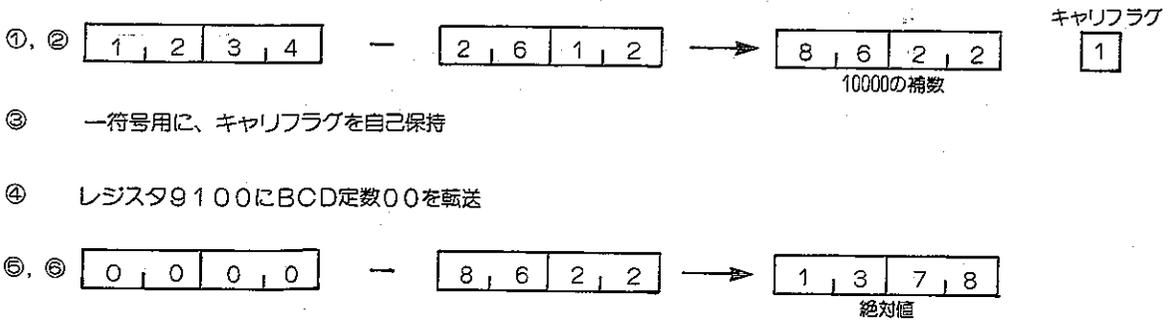
とORし、※に1を挿入しています

〔10〕減算結果を符号付絶対値で求める。

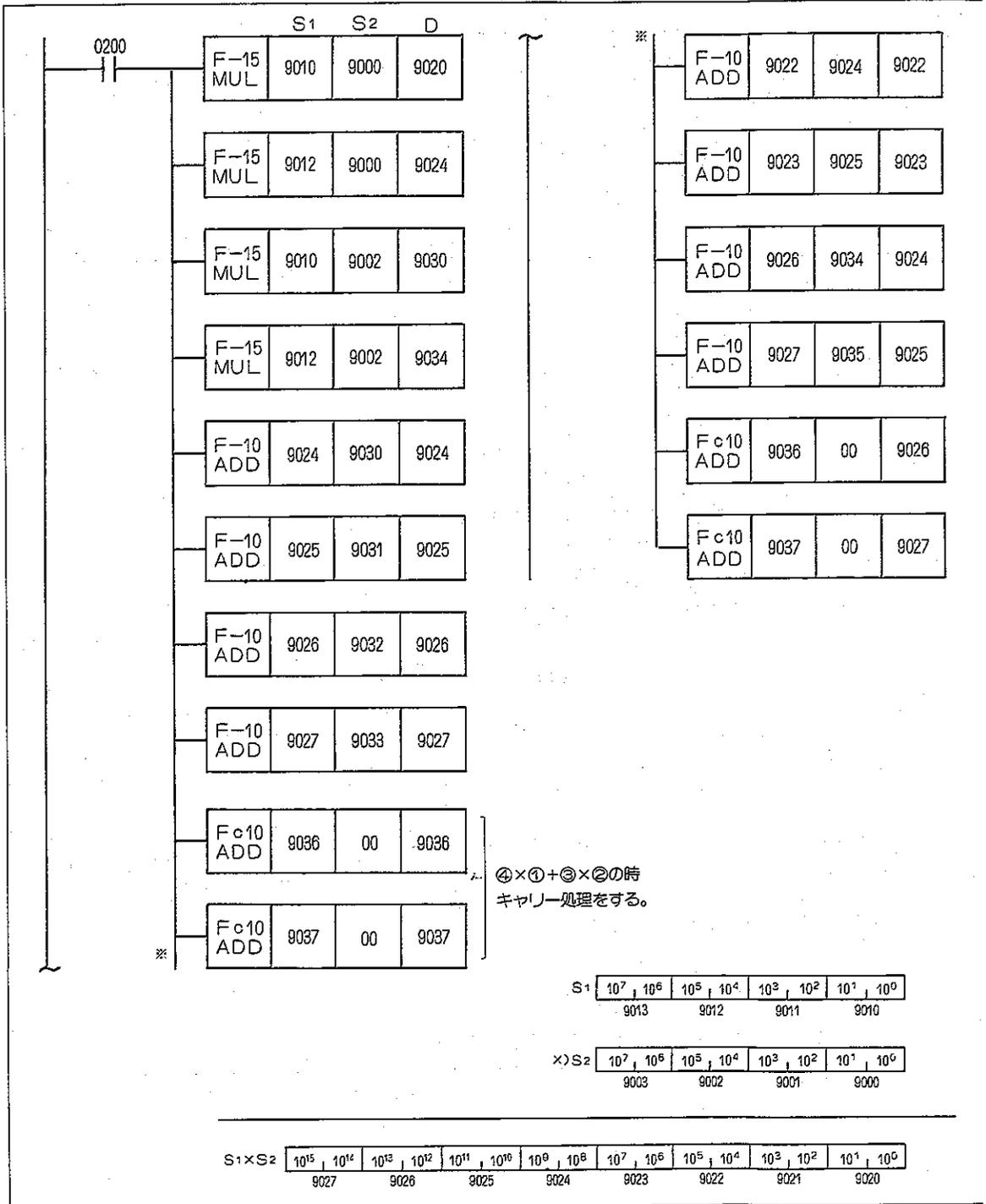


● F-11で  $\langle S_1 \rangle < \langle S_2 \rangle$  の減算を行うと、答は補数で得られます。  
 (例)  $1234 - 2612 \rightarrow 8622$  (10000の補数)

● 答を符号付の絶対値で得る場合、上記のようにプログラムします。この場合、答は  $|S_1 - S_2| = D$  となり、一符号は7000に出力されます。

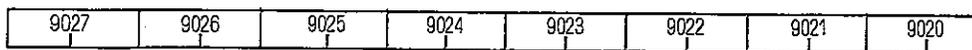
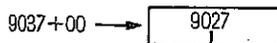
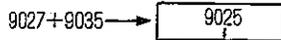
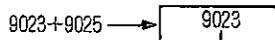
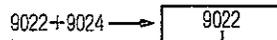
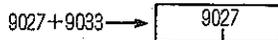
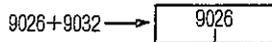
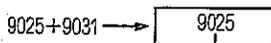
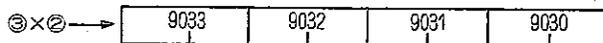
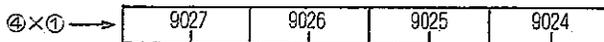
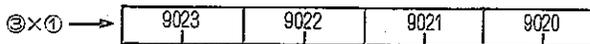
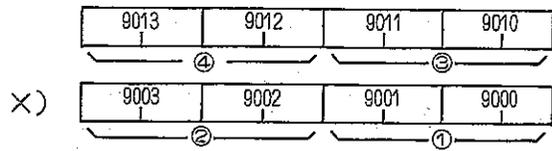


(11) BCD8桁の乗算



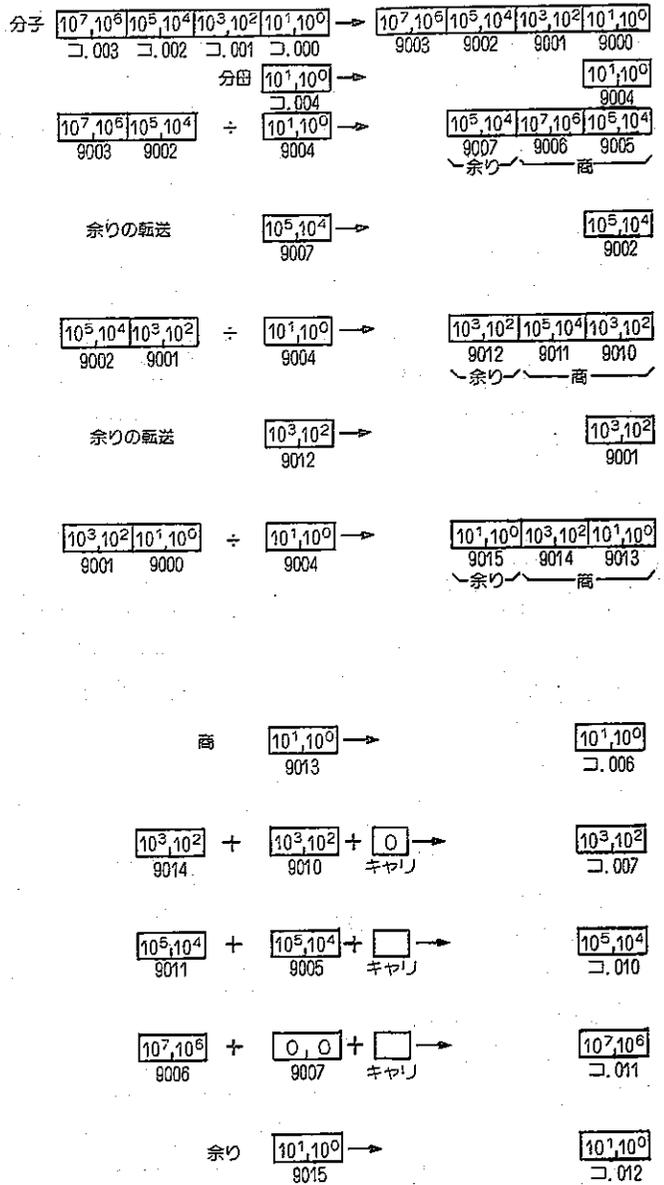
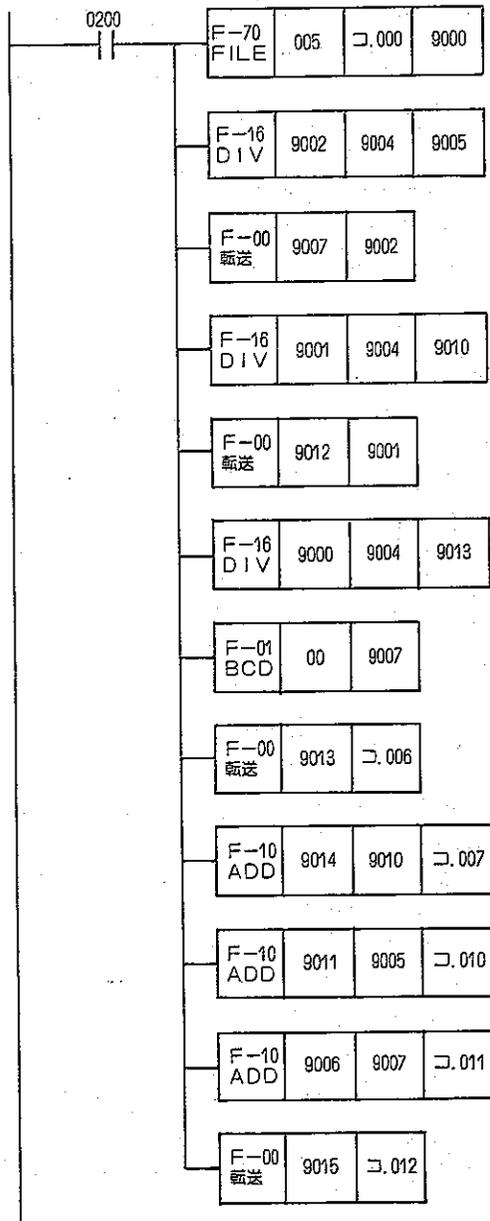
● F-15はBCD4桁の乗算命令ですが、上記のようにプログラムするとBCD8桁×BCD8桁の乗算ができます。

注1) W51の場合、特殊リレー7350がONの時、F-15命令1個でBCD8桁の乗算が可能です。



注1) レジスタの9030~9037は演算の中間結果を一時的に記憶するのに使用しています。

〔12〕 BCD8桁÷BCD2桁



$$\overbrace{\begin{array}{cccc} \hline 10^7, 10^6 & 10^5, 10^4 & 10^3, 10^2 & 10^1, 10^0 \\ \hline \end{array}}^{\text{分子}} \div \overbrace{\begin{array}{cc} \hline 10^1, 10^0 \\ \hline \end{array}}^{\text{分母}} \rightarrow \overbrace{\begin{array}{cc} \hline 10^1, 10^0 \\ \hline \end{array}}^{\text{余り}} \overbrace{\begin{array}{cccc} \hline 10^7, 10^6 & 10^5, 10^4 & 10^3, 10^2 & 10^1, 10^0 \\ \hline \end{array}}^{\text{商}}$$

$$\begin{array}{cccc} \hline 0.003 & 0.002 & 0.001 & 0.000 \\ \hline \end{array} \div \begin{array}{cc} \hline 0.004 \\ \hline \end{array} \rightarrow \begin{array}{cc} \hline 0.012 \\ \hline \end{array} \begin{array}{cccc} \hline 0.011 & 0.010 & 0.007 & 0.006 \\ \hline \end{array}$$

〔注1〕 W51の場合、特殊リレー7350がONの時、F-16命令1個でBCD8桁の除算が可能です。

### (13) BCD4桁の除算

F-16は分子がBCD4桁、分母がBCD2桁の除算命令です。分子、分母ともにBCD4桁の除算は次の方法で実現できます。

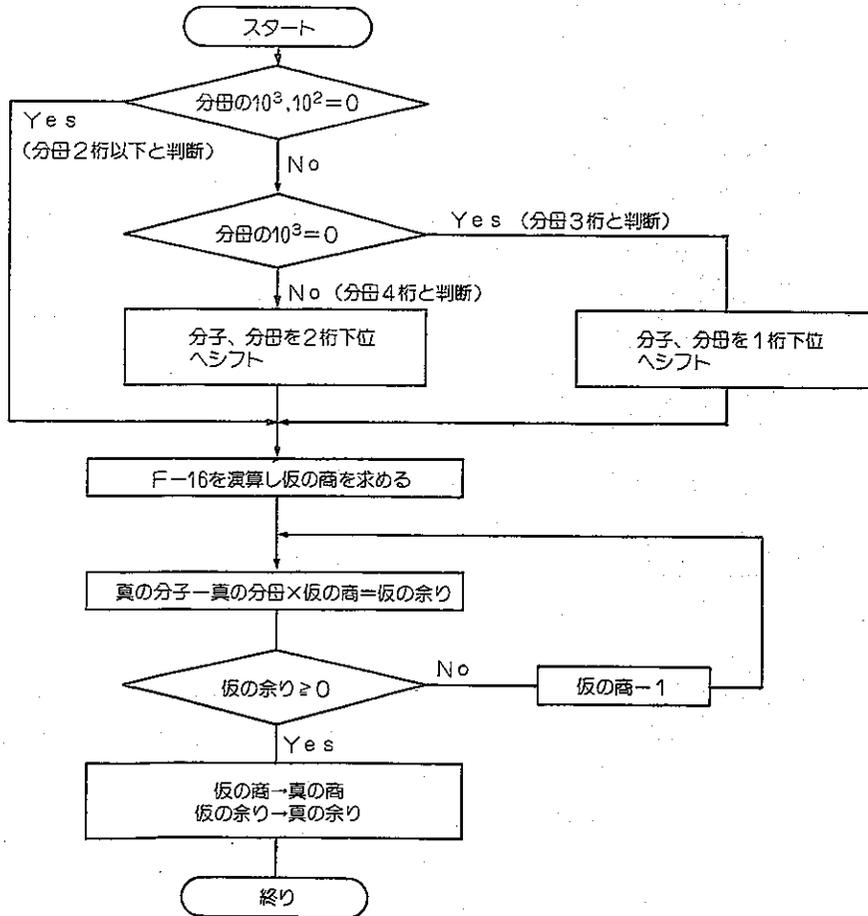
**注1** W51の場合、特殊リレー7350がONの時、F-16命令1個でBCD8桁の除算が可能です。

(考え方)

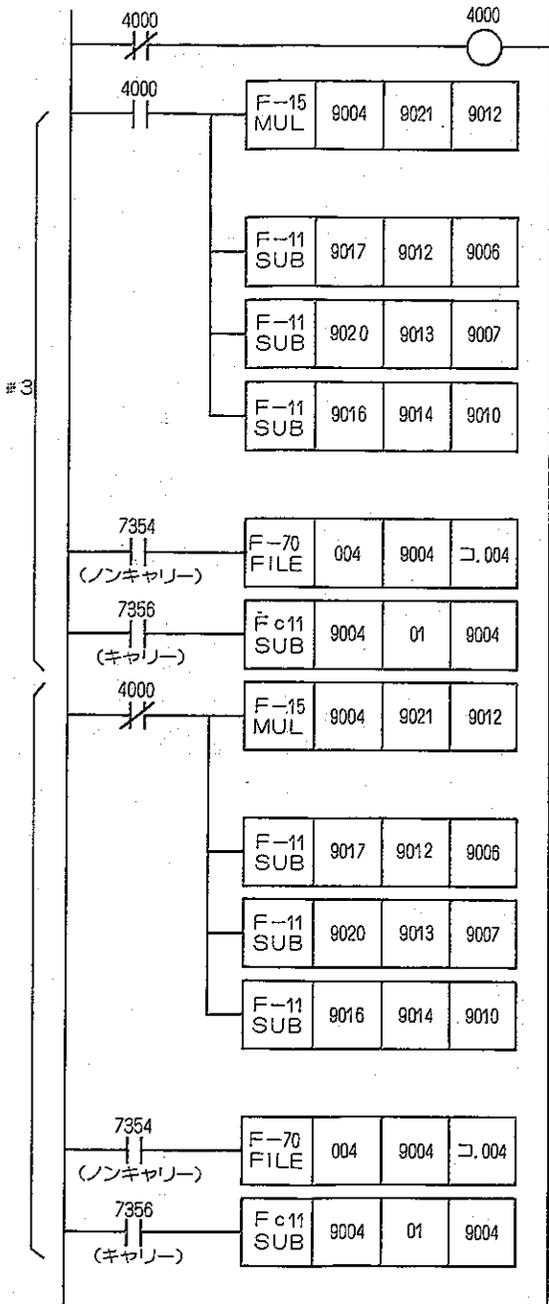
①F-16で演算可能な分母2桁となるように分子、分母を同じ桁数だけ移動しF-16で除算し、仮の商を求めます。

②(真の分子-真の分母×仮の商=仮の余り)を演算し、仮の余り $\geq 0$ の場合、これを真の余りとし、仮の商を真の商とします。仮の余り $< 0$ の場合桁移動のため分母を切り捨てたことにより、仮の商 $>$ 真の商となっているため、仮の商を-1し再度(真の分子-真の分母×仮の商=余り)の演算を行い余り $\geq 0$ となるまでこれを繰り返します。

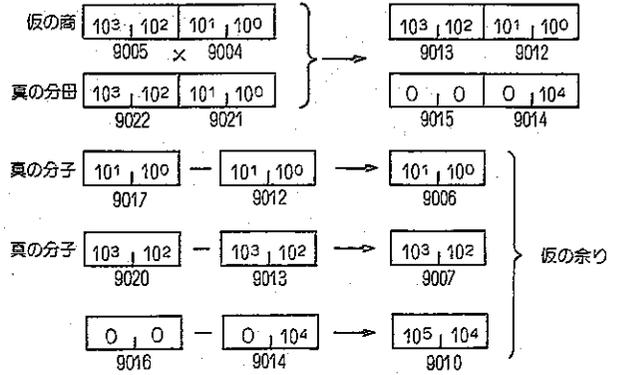
(フローチャート)







発振回路



9004 → 0.004 真の商 (10<sup>1</sup>, 10<sup>0</sup>)  
 9005 → 0.005 真の商 (10<sup>3</sup>, 10<sup>2</sup>)  
 9006 → 0.006 真の余り (10<sup>1</sup>, 10<sup>0</sup>)  
 9007 → 0.007 真の余り (10<sup>3</sup>, 10<sup>2</sup>)

仮の余り ≥ 0 のとき 9004 を -1

\*3と同じ演算を4000がOFFのスキャンサイクルで行う。

\*3 → \*4 → \*3 ……を仮の余り ≥ 0 となる迄繰り返す。

(レジスタ使用状況)

コ.000	(入力データ) 分子の下2桁
コ.001	(入力データ) 分子の上2桁
コ.002	(入力データ) 分母の下2桁
コ.003	(入力データ) 分母の上2桁
コ.004	(演算結果) 商の下2桁
コ.005	(演算結果) 商の上2桁
コ.006	(演算結果) 余りの下2桁
コ.007	(演算結果) 余りの上2桁

9000	W.R 分子の下2桁
9001	W.R 分子の上2桁
9002	W.R 分母の下2桁
9003	W.R 分母の上2桁
9004	W.R 板の商下2桁
9005	W.R 板の商上2桁
9006	W.R 板の余り下2桁
9007	W.R 板の余り中2桁

9010	W.R 板の余り上2桁
9011	W.R 分母3桁のとき1桁シフト用
9012	W.R 真の分母×板の商(10 <sup>7</sup> , 10 <sup>8</sup> )
9013	W.R 真の分母×板の商(10 <sup>8</sup> , 10 <sup>9</sup> )
9014	W.R 真の分母×板の商(10 <sup>9</sup> , 10 <sup>8</sup> )
9015	W.R 真の分母×板の商(10 <sup>7</sup> , 10 <sup>9</sup> )
9016	W.R 商の検定に使用
9017	W.R 分子の下2桁
9020	W.R 分子の上2桁
9021	W.R 分母の下2桁
9022	W.R 分母の上2桁

W.R;ワーキングレジスタ

(演算に要するスキャンサイクル数)

最大で10スキャンサイクル必要とします。

(例1) 1スキャンサイクルで完了する例

(①はスキャンサイクルを示します。)

1234÷1010の場合

①12÷10=1……余り2(板の余り)

①1234-1010×1=224>0

⇒商1, 余り224

(例2) 10スキャンサイクル必要とする例

(①~⑩はスキャンサイクルを示します。)

9900÷109の場合

①990÷10=99

①9900-109×99=-891<0

→ 99-1=98

②9900-109×98=-782<0

→ 98-1=97

③9900-109×97=-673<0

→ 97-1=96

④9900-109×96=-564<0

→ 96-1=95

⑤9900-109×95=-455<0

→ 95-1=94

⑥9900-109×94=-346<0

→ 94-1=93

⑦9900-109×93=-237<0

→ 93-1=92

⑧9900-109×92=-128<0

→ 92-1=91

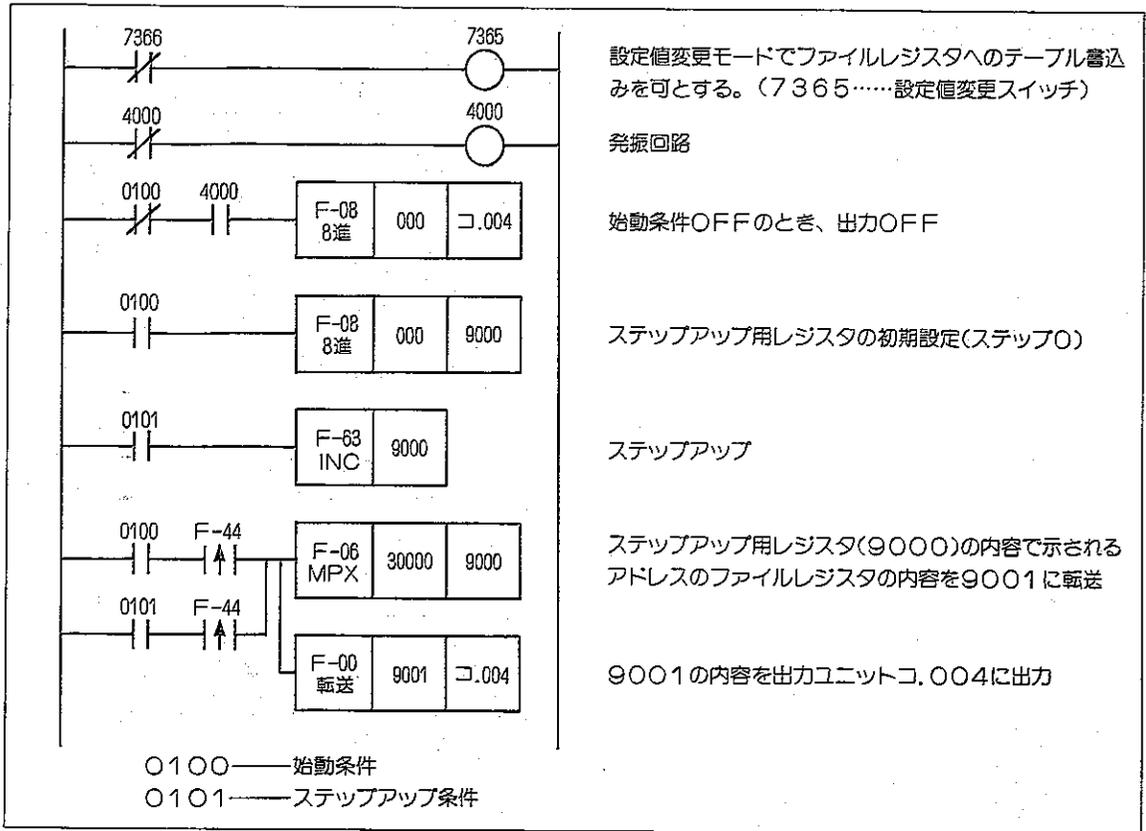
⑨9900-109×91=-19<0

→ 91-1=90

⑩9900-109×90=90>0

⇒商90, 余り90

(14)ドラムシーケンサ



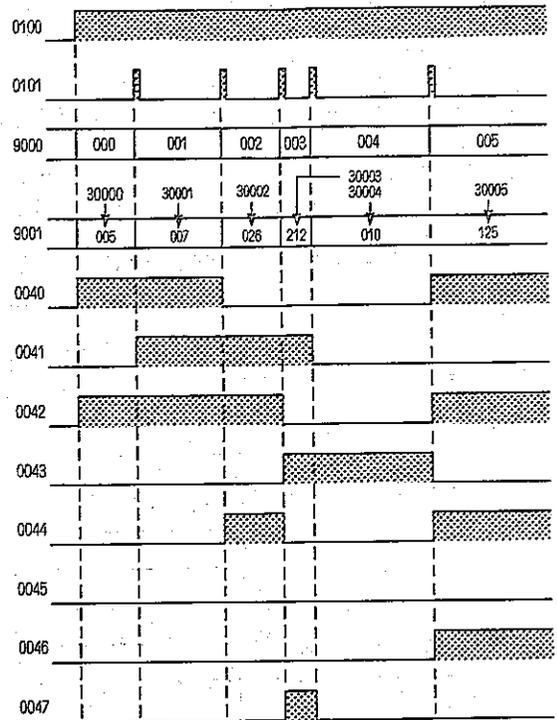
- ドラム型シーケンサと同じ機能を実現できます。
- 運転開始に先立ち、設定値変更モードでファイルレジスタの30000以後に、各ステップでの出力のON/OFF状態をテーブルとして書込みます。

出力	ステップ	0	1	2	3	4	5	6	7	8	9		255
0.004	0040	○	○					○	○			○	
	0041		○	○	○					○		○	
	0042	○	○	○				○	○	○		○	
	0043				○	○				○	○	○	
	0044			○				○	○	○	○		
	0045								○	○			○
	0046							○	○		○		
	0047				○								
ファイルレジスタNo		30000	30001	30002	30003	30004	30005	30006	30007	30010	30011		30377
データ		005	007	026	212	010	125	165	076	130	017		040

○印——出力ON

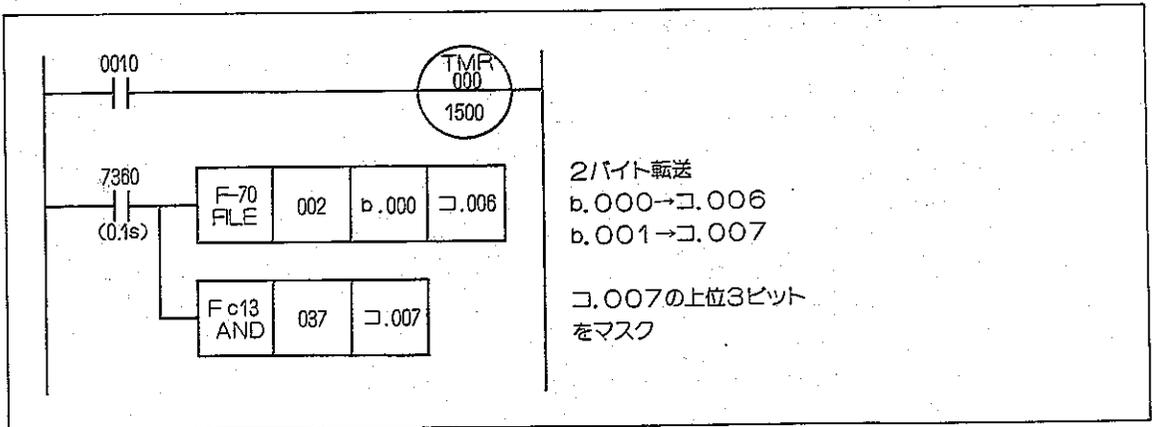
ファイルレジスタへの書き込みをビットパターンで行うと、出力のON/OFF状態を直接書込みます。

- 始動条件がONになるとステップ0の出力状態となり、以後、ステップアップ条件がONすることに各ステップの出力状態となります。



- レジスタ9000をモニタ(又は外部表示)すれば進行中のステップがわかります。

(15) タイマの現在値の外部出力



2バイト転送  
 b.000→コ.006  
 b.001→コ.007

コ.007の上位3ビットをマスク

- b.000, b.001の2バイトに格納されているTMR000の現在値をコ.006, コ.007の2バイトに転送します。
- TMRは0.1秒クロックでカウントダウンするため、F-70, Fc13の入力条件として7360(0.1秒クロック)を用いています。

- TMR, CNTの現在値とMDの設定値は、データメモリのb.000~b.377の領域に格納されています。TMR, CNT, MDの番号と、b.XXXの関係を下表に示します。

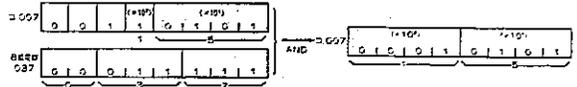
TMR, CNT, MD番号	000	001	177
b. XXX	b.000, b.001	b.002, b.003	
			b.376, b.377

- TMRの場合、b. XXX, b. (XXX+1)には次のようなデータフォーマットで現在値が格納されています。(数値はBCDコードで扱います。)

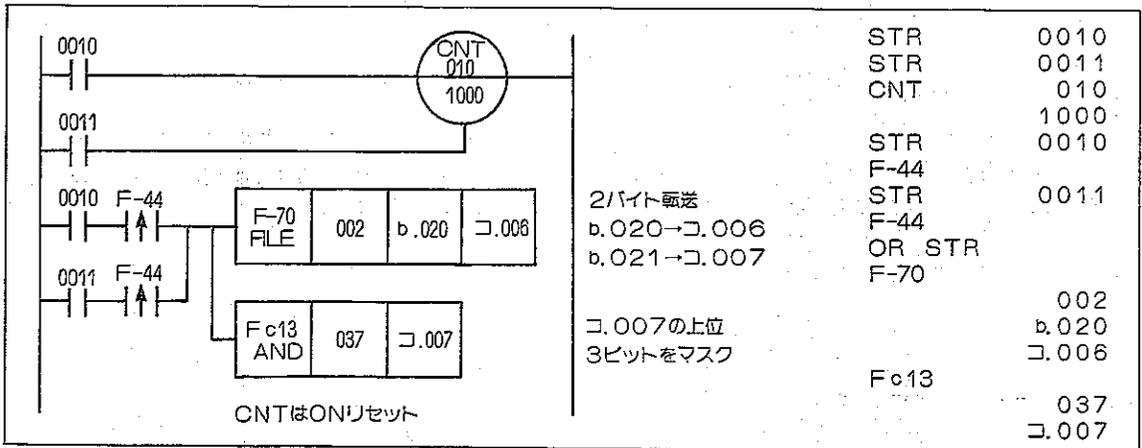
	7	6	5	4	3	2	1	0
b. XXX	(×10 <sup>0</sup> )				(×10 <sup>-1</sup> )			
	"8"	"4"	"2"	"1"	"8"	"4"	"2"	"1"
b.(XXX+1)	OFF	OFF	※1	×10 <sup>2</sup>	(×10 <sup>1</sup> )			
	OFF	OFF	ON	"1"	"8"	"4"	"2"	"1"

※1 設定値変更モードで強制リセットすると、OFFになります。

- b. 001の上位3ビットは現在値以外のデータのため、外部に出力する場合マスクする必要があります。F-70でb. 000をコ. 006に、b. 001をコ. 007に転送した後、コ. 007の上位3ビットをFc13でマスクしています。



### (16)カウンタの現在値の外部出力

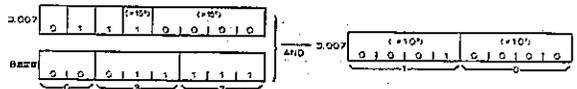


- 計数入力の立上り時(CNTはカウントダウン)、リセット入力の立上り時(CNTはリセット)に、CNT010の現在値が格納されているb. 020, b. 021の内容をCo. 006, Co. 007に転送します。
- CNTの場合、b. XXX, b. (XXX+1)には次のようなデータフォーマットで現在値が格納されています。(数値はBCDコードで扱います。)

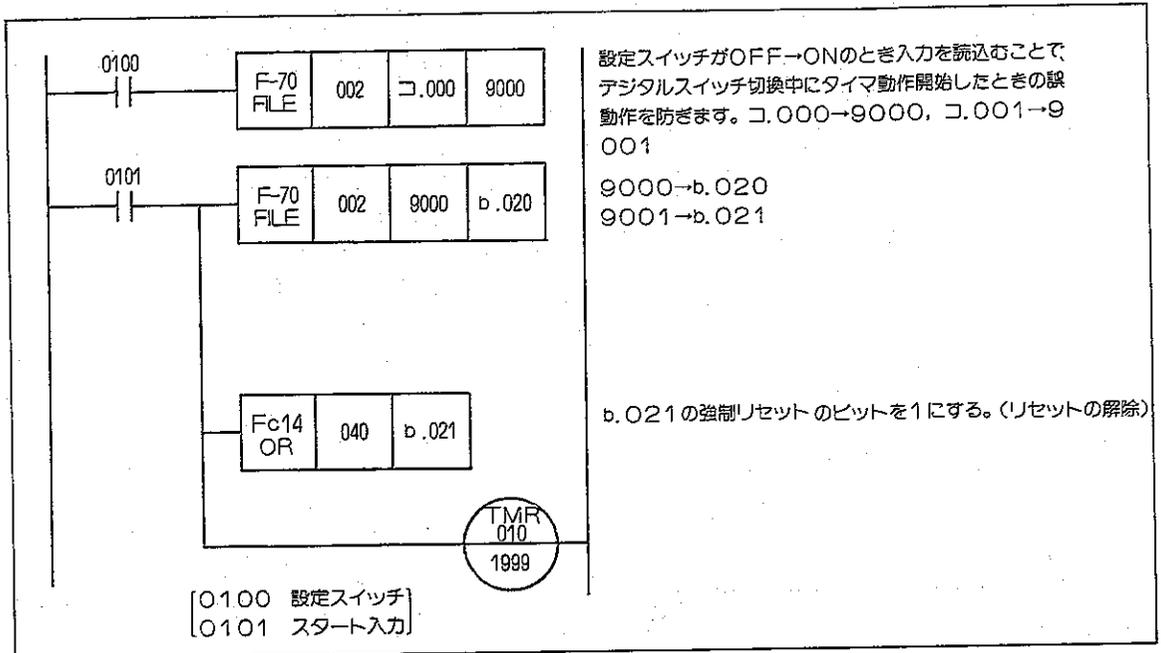
	7	6	5	4	3	2	1	0
b. XXX	(×10 <sup>1</sup> )				(×10 <sup>0</sup> )			
	"8"	"4"	"2"	"1"	"8"	"4"	"2"	"1"
b.(XXX+1)	OFF	ON	※1	(×10 <sup>2</sup> )	(×10 <sup>3</sup> )			
	OFF	ON	ON	"1"	"8"	"4"	"2"	"1"

※1 設定値変更モードで強制リセットすると、OFFになります。

- b. 021の上位3ビットは、現在値以外のデータのため、外部に出力する場合、マスクする必要があります。F-70でb. 020をCo. 006に、b. 021をCo. 007に転送した後、Fc13でマスクしています。

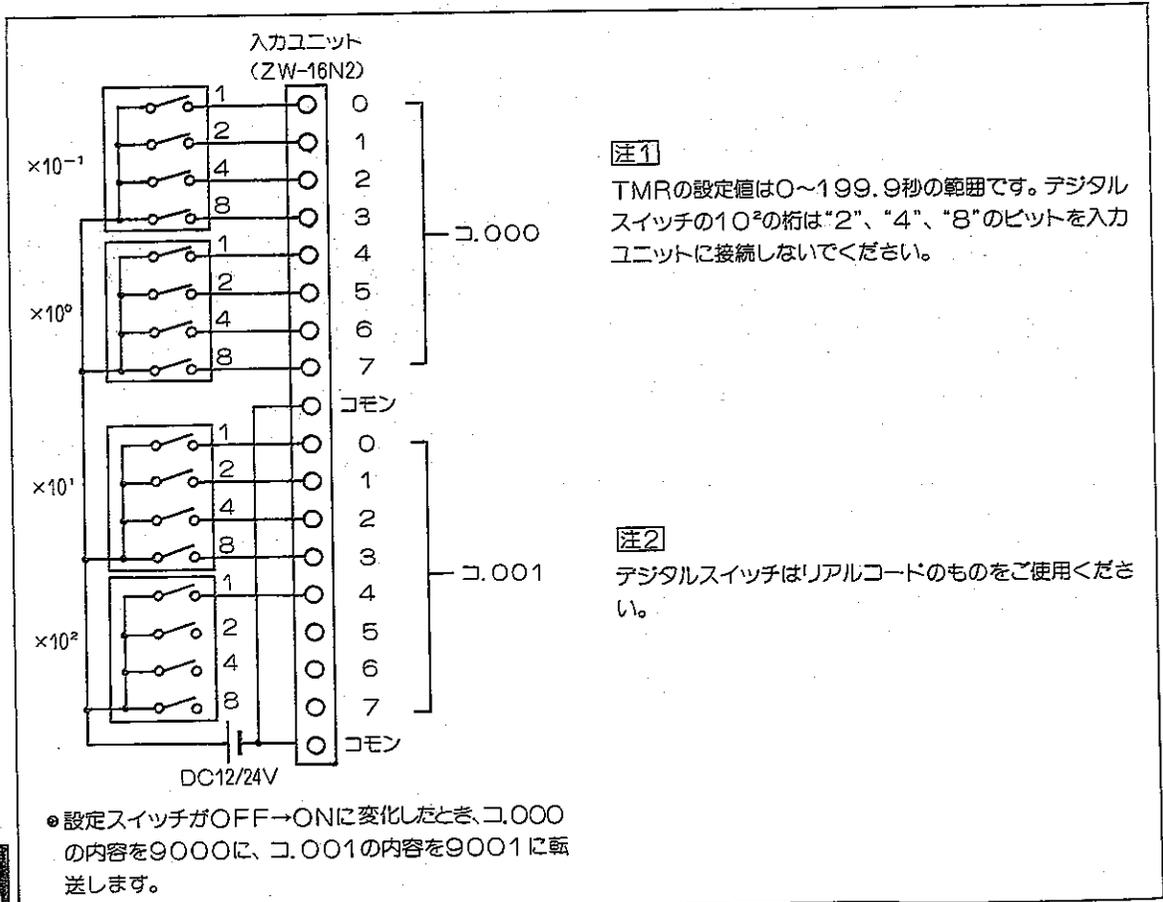


(17) タイマの設定値を外部機器から入力

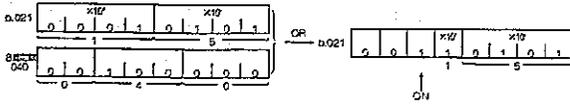


● W51 (W16)のRUN中にデジタルスイッチ等の外部機器からTMRの設定値を変更できます。(正しくは現在値を外部から書換えています。)

● コ.000, コ.001にはデジタルスイッチ等の外部機器を接続します。



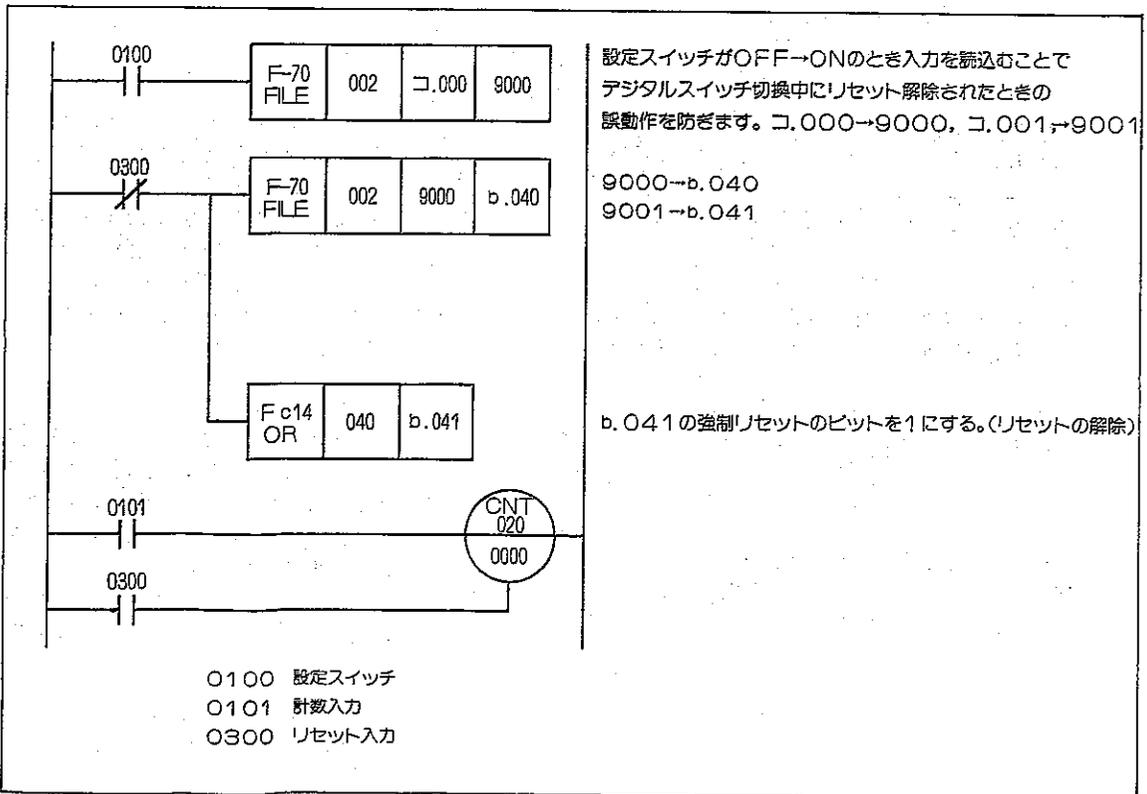
- スタート入力がOFF→ONに変化したとき、
  - ① 9000の内容をb.020に、9001の内容をb.021に転送します。
  - ② b.021の上から3ビット目をONにします。(Fc14)
 このビットがOFFの場合、TMR010のプログラムの設定値(本例の場合1999)がb.020、b.021に再度書き込まれ、外部設定が無効となります。



- スタート入力がONの間、0.1秒ごとに外部から書き込んだ現在値から減算され現在値が0になるとTMR接点ONとなります。
- スタート入力がOFFになると、TMRの現在値はプログラム上の設定値(本例の場合、1999)になり、TMR接点はOFFになります。スタート入力が再びONになると、データメモリの9000、9001の内容が現在値として転送されるため、プログラム上の設定値(本例の場合1999)は無視されます。(0~1999の任意の値をプログラムしてください。)

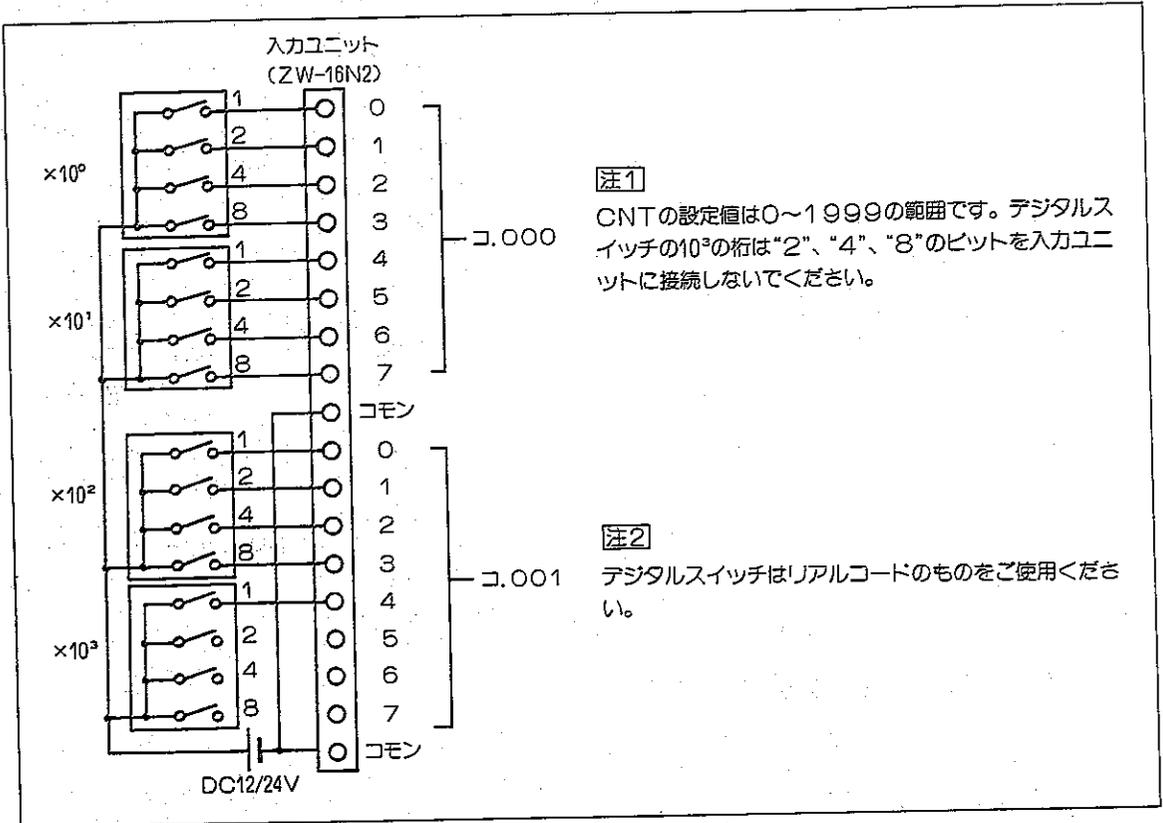
- 注1** スタート入力がONのとき、プログラマで強制セットすると、現在値は0となり、TMR接点をONとすることができます。強制リセットすると、現在値はプログラム上の設定値(本例の場合1999)になり、デジタルスイッチの設定値にはなりませんので御注意ください。
- 注2** スタート入力がONのとき(タイム動作中)に、デジタルスイッチの数値を変更しても現在値は変更されません。変更後の設定値が有効となるのは、一旦スタート入力がOFFとなり、次にONとなったときからとなります。

### (18)カウンタの設定値を外部機器から入力



●W51(W16)のRUN中にデジタルスイッチ等の外部機器からCNTの設定値を変更できます。(正しくは現在値を外部から書換えています。)

●コ.000, コ.001にはデジタルスイッチ等の外部機器を接続します。



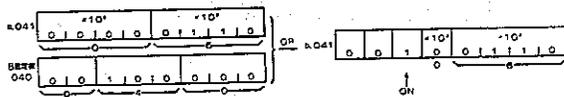
**注1**

CNTの設定値は0~1999の範囲です。デジタルスイッチの10<sup>3</sup>の桁は“2”、“4”、“8”のビットを入力ユニットに接続しないでください。

**注2**

デジタルスイッチはリアルコードのものをご使用ください。

- リセット入力がON→OFFに変化したとき、  
①9000の内容をb.040に、9001の内容をb.041に転送します。  
②b.041の上から3ビット目をONにします。(Fc14)  
このビットがOFFの場合、CNT020のプログラム上の設定値(本例の場合0000)がb.040, b.041に再度書込まれ、外部設定が無効となります。



- リセット入力がOFFの間、計数入力がOFF→ONに変化することに外部から書込んだ現在値から減算され、現在値が0になるとCNT接点がりONになります。

- リセット入力がONになると、CNTの現在値はプログラム上の設定値(本例の場合0000)になり、CNT接点はOFFになります。  
リセット入力が再びOFFになると、データメモリのコ.000, コ.001の内容が現在値として転送されるため、プログラム上の設定値(本例の場合0000)は無視されます。(0~1999の任意の値をプログラムしてください。)

**注1**

リセット入力がOFFのとき、プログラムで強制セットすると、現在値は0となり、CNT接点をONとすることができます。強制リセットすると、現在値はプログラム上の設定値(本例の場合0000)になりデジタルスイッチの設定値にはなりませんのでご注意ください。

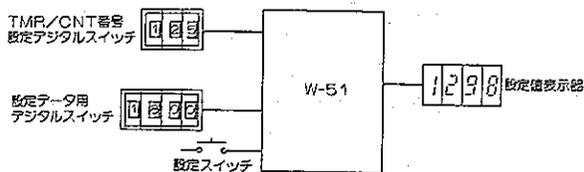
**注2**

リセット入力がOFF(カウンタ動作中)にデジタルスイッチの数値を変更しても現在値は変更されません。

変更後の設定値が有効となるのは、一旦リセット入力がONとなり次にOFFとなったときからとなります。

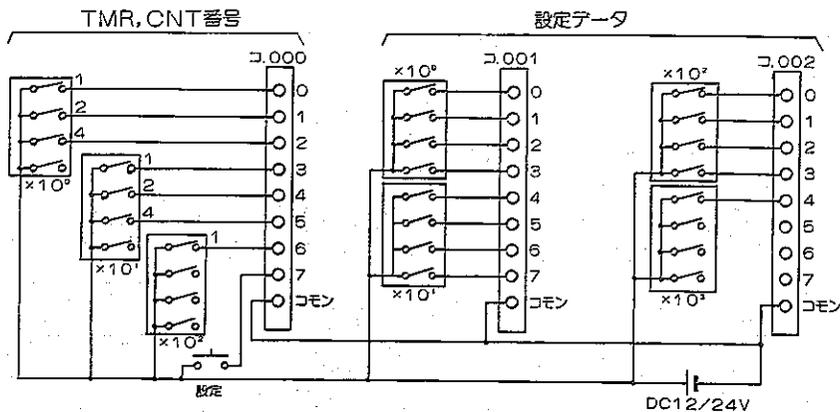
## (19) 複数のタイマ、カウンタの設定値を外部機器から入力

● 外部設定する必要があるタイマ、カウンタが多数あるとき、(17)、(18)の方法では、デジタルスイッチがタイマ、カウンタの数だけ必要となります。本例ではTMR/CNT番号指定デジタルスイッチ(3桁)と、設定データ用デジタルスイッチ(4桁)、設定値表示用表示器(4桁)で128点のTMR、CNTを全て外部設定式にすることができます。



### (1) 外部接続

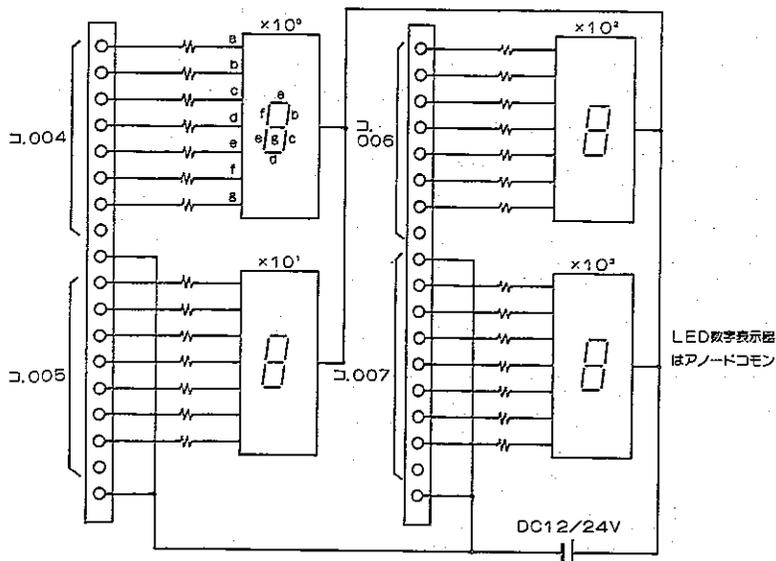
#### a. 設定用外部接続



コ.000	TMR, CNT 番号	8進数で設定(000~177)
コ.001	設定値下2桁	BCDで設定(00~99)
コ.002	設定値上2桁	BCDで設定(00~19)
0007	設定スイッチ	ONで設定

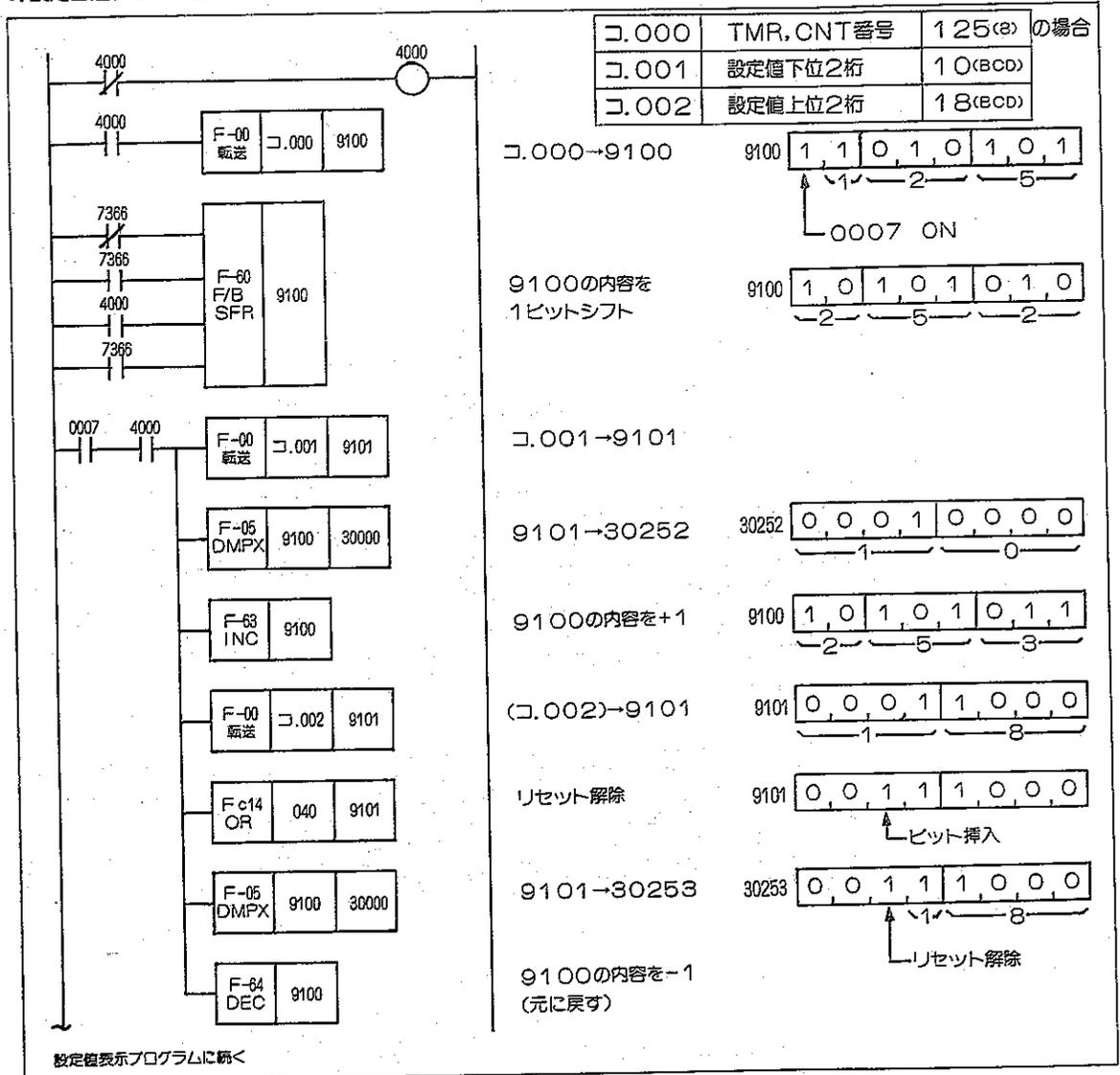
デジタルスイッチはリアルコード

#### b. 設定値表示用外部接続(F-52使用)



② プログラム

a. 設定番プログラム

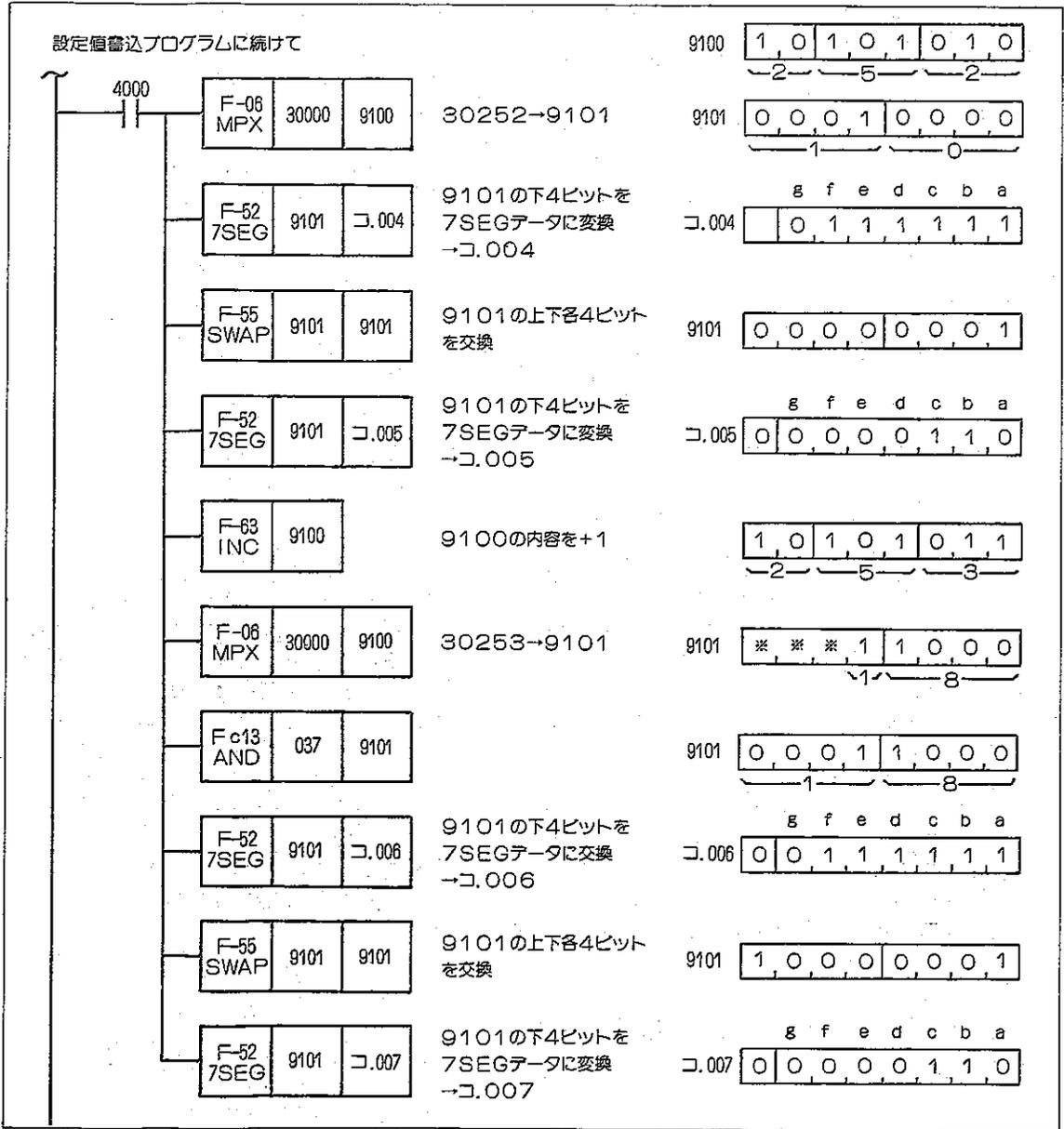


●設定スイッチをONにすると、TMR/CNT番号指定デジタルスイッチで指定されたTMR、CNTの設定値として設定値デジタルスイッチを読み込みファイルレジスタ(30000~30377)にデータテーブルとして分配します。

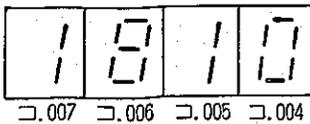
●F-60で1ビット左シフトすることにより、コ.000のTMR、CNT番号を当該の現在値(下位2桁)格納レジスタのアドレスに変換しています。

TMR/CNT番号	ファイルレジスタ	
000	30000	下位2桁
	30001	上位2桁
001	30002	下位2桁
	30003	上位2桁
≈	≈	≈
177	30376	下位2桁
	30377	上位2桁

b. 設定値表示プログラム



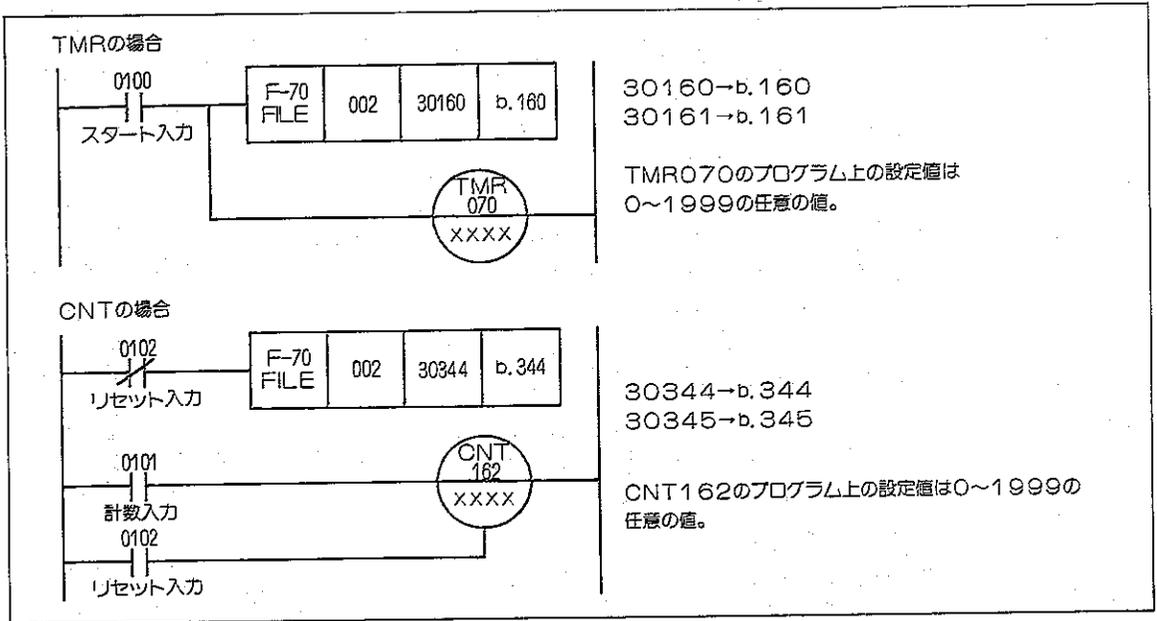
- TMR/CNT番号設定デジタルスイッチで指定したTMR, CNTの外部設定値をファイルレジスタから抽出し、7セグメント数字表示器に出します。
- 上記の例ではTMR125の設定値として、



が表示されます。



c. TMR, CNTのプログラム



● TMRの場合、スタート入力がOFF→ONに変化したとき、ファイルレジスタから外部設定値が転送されます。

● CNTの場合、リセット入力がON→OFFに変化したとき、ファイルレジスタから外部設定値が転送されます。

注1 システムメモリ#202でカウンタのリセット条件をOFFリセットとしたときは、F-70の演算条件にご注意ください。(リセット解除でF-70実行とします。)

注2 スタート入力がOFFのとき(タイマー停止中)又はカウンタのリセット入力がONのとき(カウン

タがリセット中)にプログラム等でTMRの現在値をモニターすると、プログラム上の設定値が表示されます。タイマ、カウンタが動作開始時に外部設定値に書き換えられます。

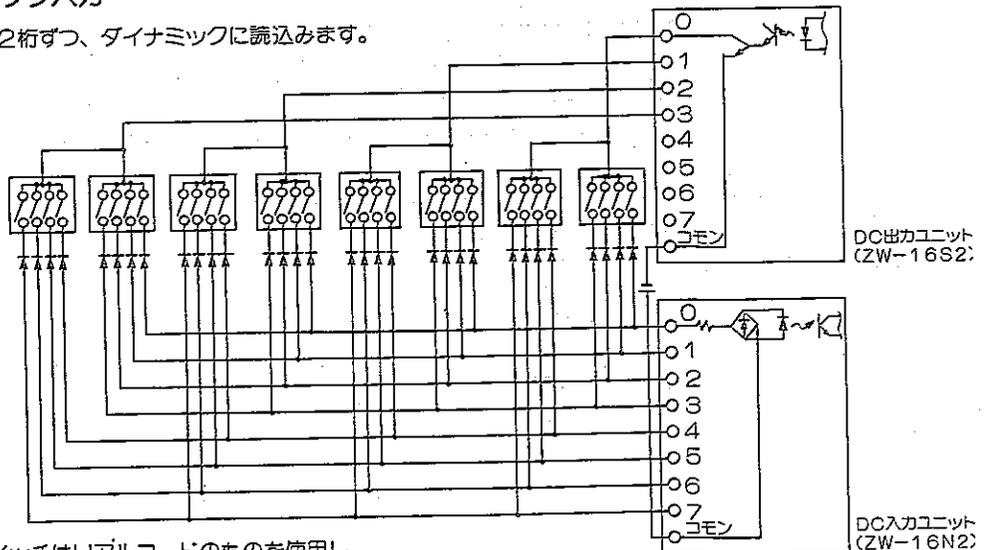
注3 スタート入力がON(タイマ動作中)、カウンタのリセット入力がOFF(カウンタ動作中)に外部設定値を変更しても受け付けられません。変更後の設定値が有効となるのは、次の動作時からとなります。

注4 外部設定する必要のない、TMR、CNTはF-70の転送命令を省略します。(一般のTMR、CNTのプログラム)

〔20〕 ダイナミック入力

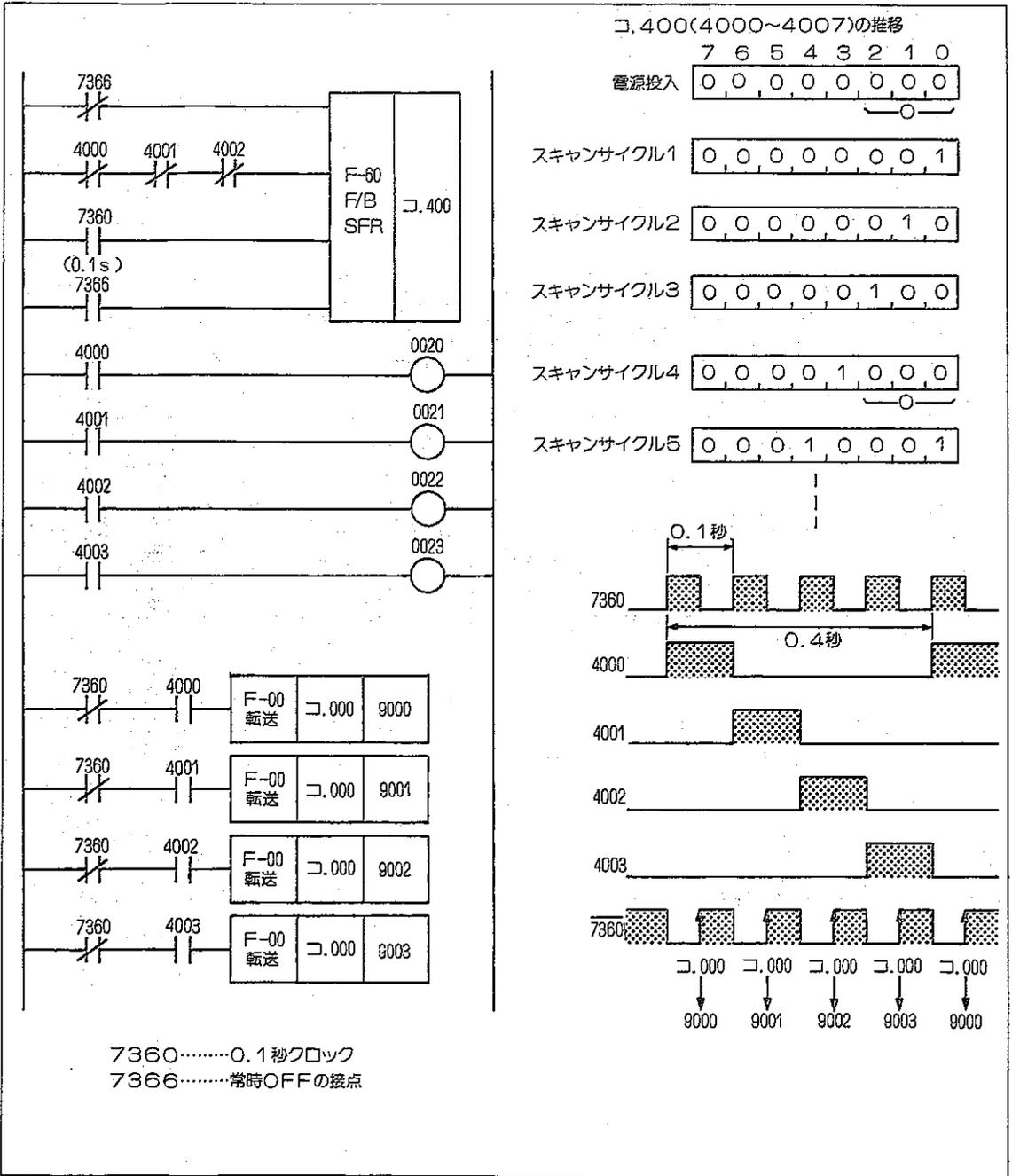
● 多桁の数値信号を2桁ずつ、ダイナミックに読み込みます。

(1) 外部接続



注1 デジタルスイッチはリアルコードのものを使用します。

② プログラム

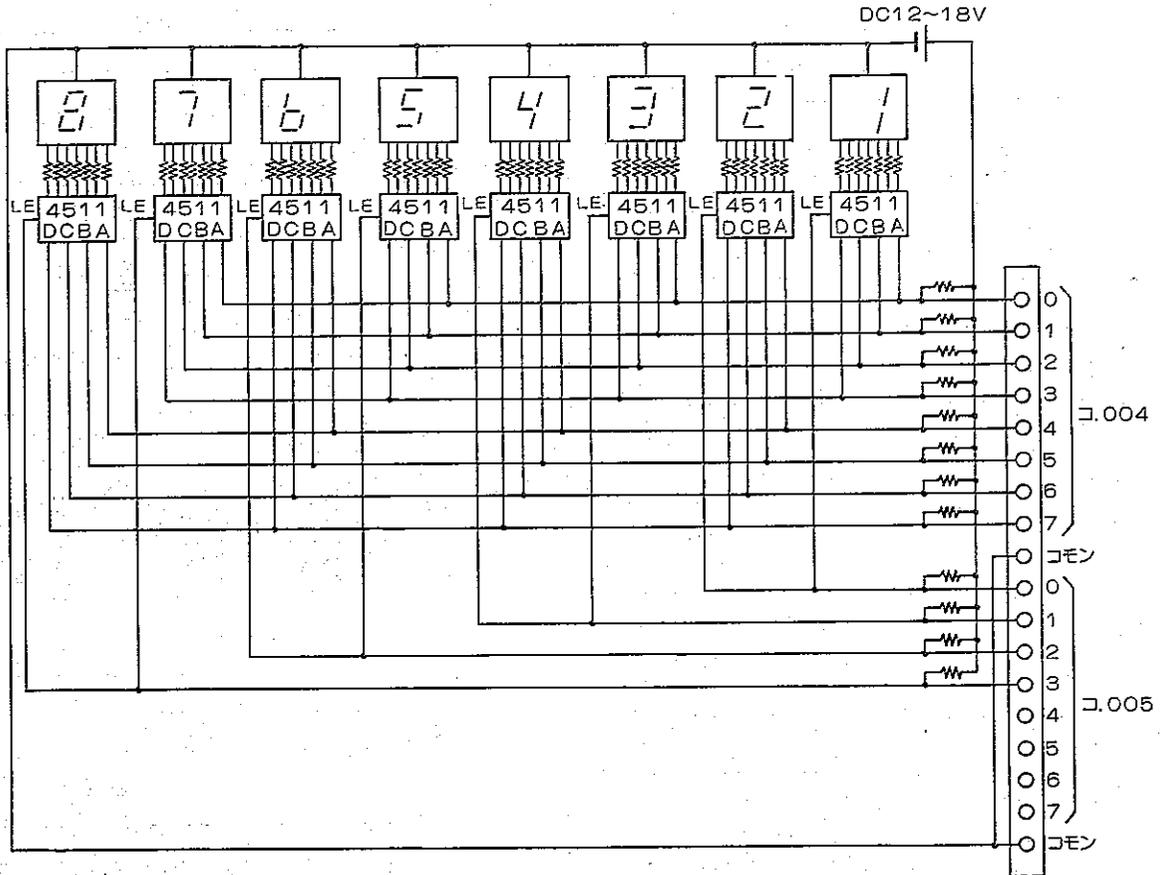


- 0.1秒クロック(7360)がOFF→ONに変化するごとに出力0020~0023のいずれか1ビットを順次ONします。
- 0.1秒クロックがON→OFFに変化するとき、コ.000のBCD2桁の数値をレジスタ9000~9003に順次格納します。
- 8桁の読みみに400msを要します。

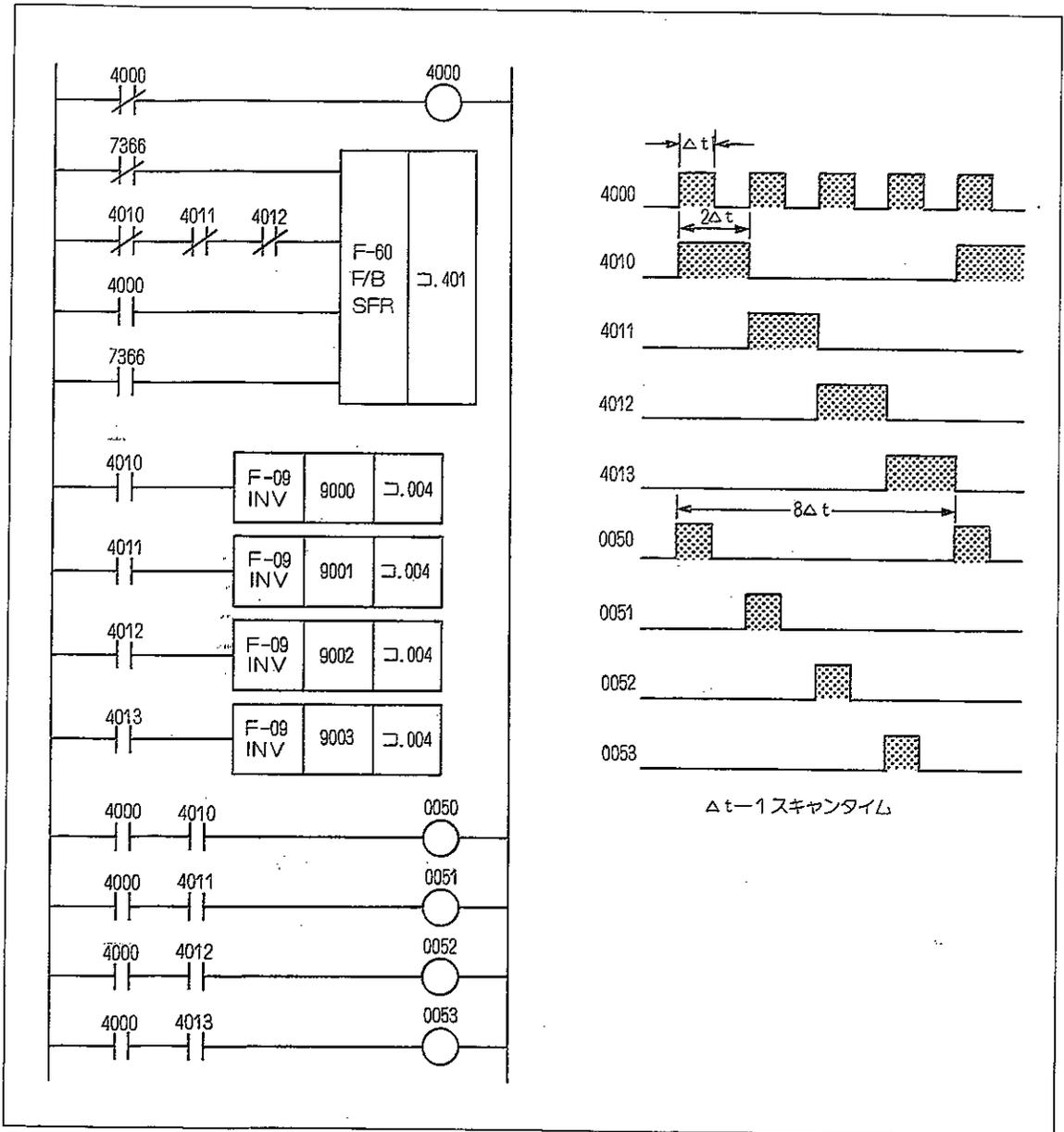
## (21) ダイナミック出力

●多桁の数値を時分割で出力し、ラッチ付数字表示回路を点灯させます。

### (1) 外部接続



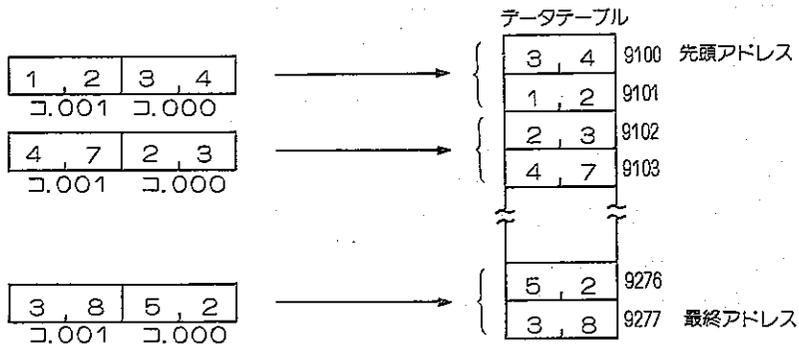
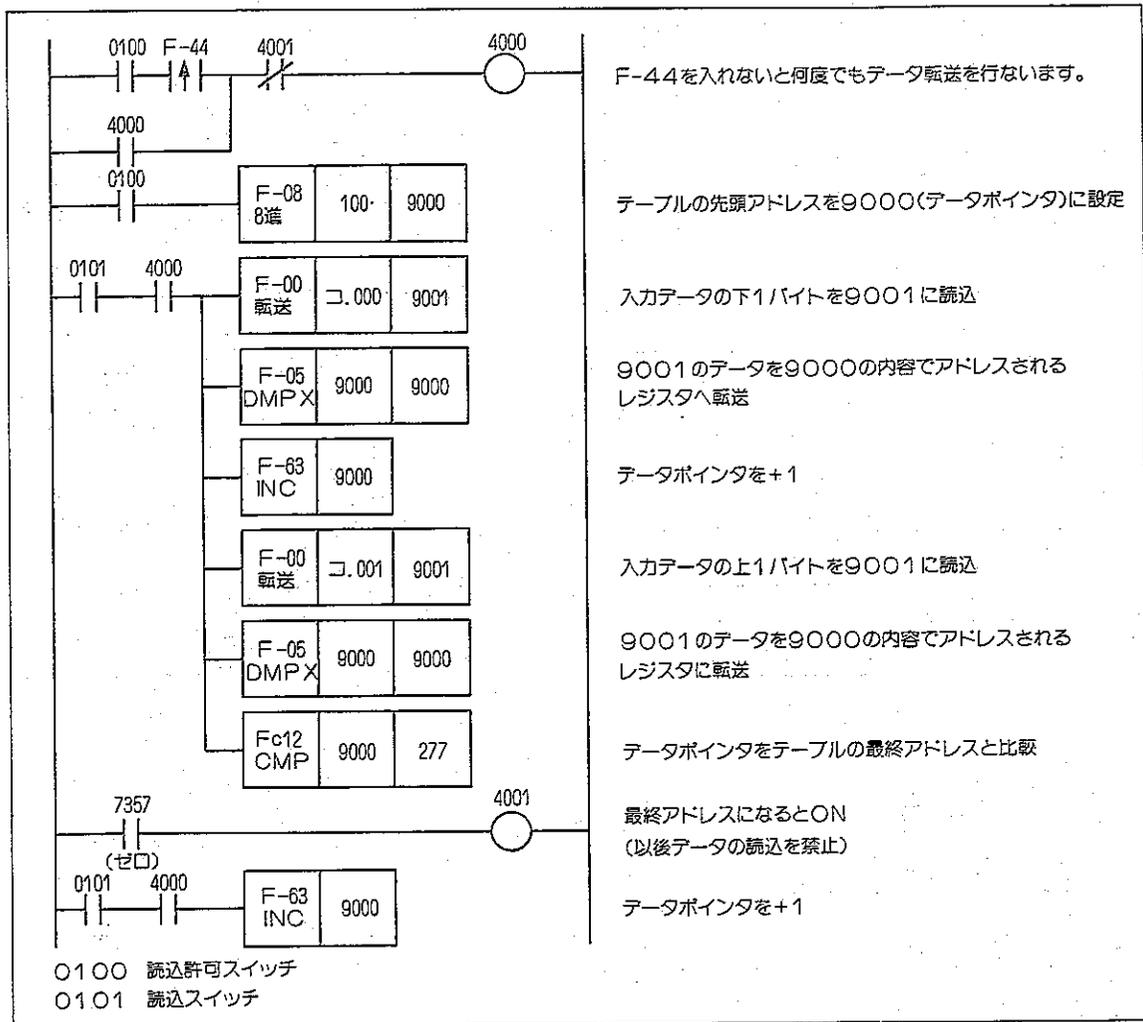
② プログラム



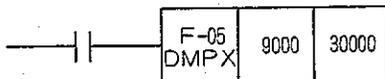
- $2\Delta t$ ごとにコ.004への出力データが9000→9001→9002→9003→9000と推移します。これに同期してストローブ信号(0050~0053)が出力されます。
- 8桁分出力するのに $8\Delta t$ の時間を要します。  
(1スキャンタイム( $\Delta t$ )が5msの場合で40ms)

〔22〕 2バイトのデータをデータテーブルに格納

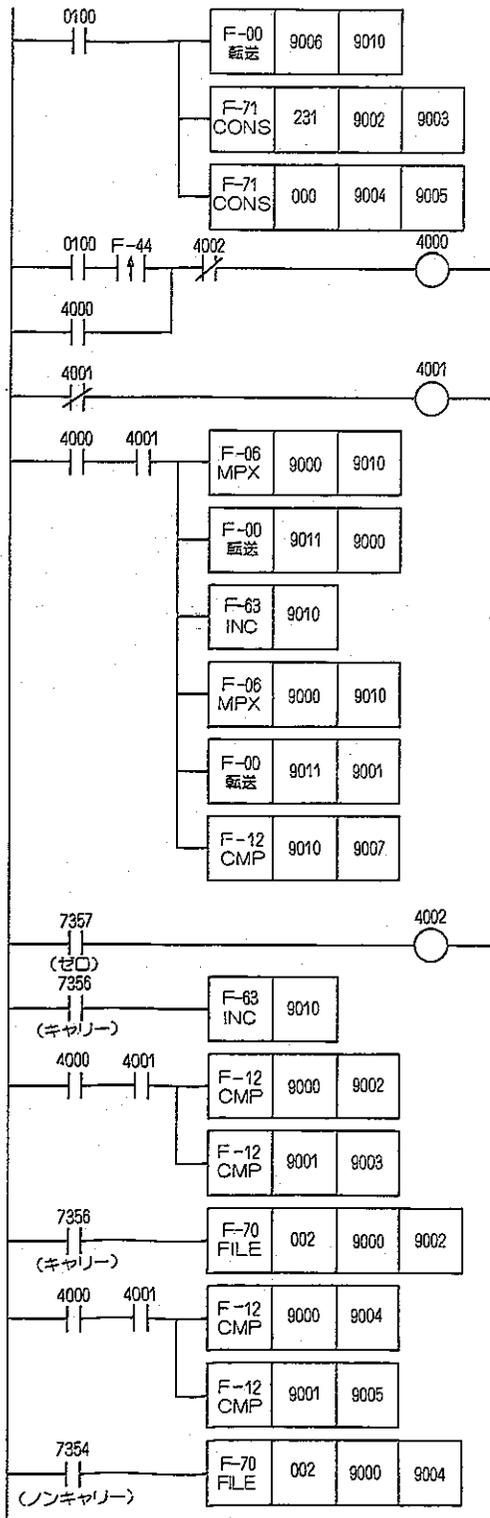
●コ.000, コ.001の2バイトデータをデータテーブルに順次格納します。



●ファイルレジスタをデータテーブルとして使用することもできます。



〔23〕 BCD 4桁の最小値・最大値を求める。



データテーブルの先頭アドレスを設定

最小値格納レジスタに231(8)(BCDで99)を転送

最大値格納レジスタに000(8)を転送

演算スタート(リセットは最終データ演算後)

発振回路

データテーブルより10<sup>1</sup>、10<sup>0</sup>のデータを抽出

比較用レジスタに格納

データポインタ(抽出元のレジスタアドレス)を+1

レジスタテーブルより10<sup>2</sup>、10<sup>1</sup>のデータを抽出

比較用レジスタに格納

データポインタはテーブルの最終アドレス以下か?

データポインタ=テーブル最終アドレスのとき演算終了

データポインタ<テーブル最終アドレスのとき、  
データポインタを+1

1 スキャン前の最小値と比較

1 回目は9999と比較

前回より小さいとき、抽出データを最小値格納  
レジスタへ転送

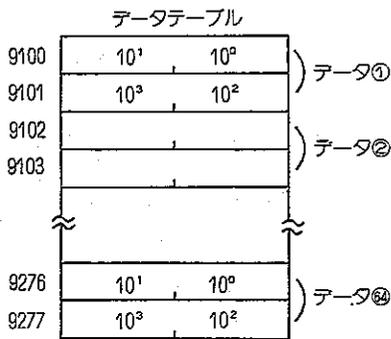
1 スキャン前の最大値と比較

1 回目は0000と比較

前回より大きいとき、抽出データを最大値格納  
レジスタへ転送

- データテーブルに格納されたBCD4桁の数値群から最大値、最小値を求めます。

9000	演算用ワーキングレジスタ
9001	演算用ワーキングレジスタ
9002	最小値格納用 (10 <sup>1</sup> , 10 <sup>0</sup> )
9003	最小値格納用 (10 <sup>3</sup> , 10 <sup>2</sup> )
9004	最大値格納用 (10 <sup>1</sup> , 10 <sup>0</sup> )
9005	最大値格納用 (10 <sup>3</sup> , 10 <sup>2</sup> )
9006	データテーブルの先頭アドレスを格納
9007	データテーブルの最終アドレスを格納
9010	データポインタ
9011	抽出データ格納



- データテーブル用レジスタには、BCD4桁の数値が格納されているものとします。(格納用プログラムは本例には含まれていません)。データテーブル用レジスタの先頭アドレスが9006に、最終アドレスが9007に設定されているものとします。(データ格納時に設定しておきます。)

データテーブル用レジスタの先頭アドレスが9100のとき、9006には100<sub>16</sub>を最終アドレスが9277のときは9007に277<sub>16</sub>を設定します。

- データテーブルはファイルレジスタ領域を使用することができます。ただしデータポインタの移動範囲は、0~377<sub>16</sub>の256バイトであり、データ数は128が最大となります。
- 2スキャンサイクルに1度ずつデータテーブルからデータを抽出し、演算するため、データテーブルの全ての数値を処理するには、(スキャンタイム×データ数×2)の時間を要します。



(レジスタ使用状況)

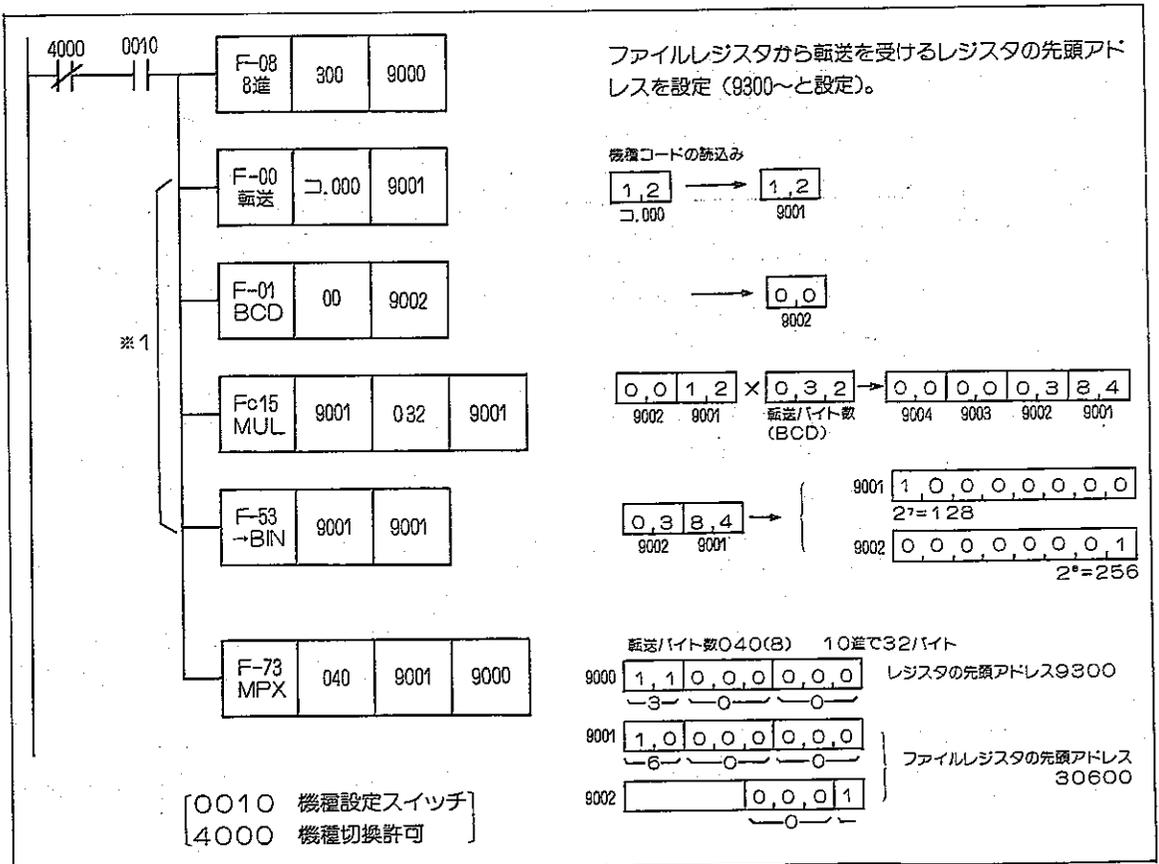
コ.004	出力 合計値(10 <sup>1</sup> , 10 <sup>0</sup> )
コ.005	出力 合計値(10 <sup>3</sup> , 10 <sup>2</sup> )
コ.006	出力 平均値(10 <sup>-1</sup> , 10 <sup>-2</sup> )
コ.007	出力 平均値(10 <sup>1</sup> , 10 <sup>0</sup> )
コ.010	出力 平均値(10 <sup>3</sup> , 10 <sup>2</sup> )

9000	抽出用データポインタ
9001	抽出データ格納
9002	整数部除算用
9003	合計値(10 <sup>1</sup> , 10 <sup>0</sup> )
9004	合計値(10 <sup>3</sup> , 10 <sup>2</sup> )
9005	平均値(10 <sup>-1</sup> , 10 <sup>-2</sup> )
9006	平均値(10 <sup>1</sup> , 10 <sup>0</sup> )
9007	平均値(10 <sup>3</sup> , 10 <sup>2</sup> )

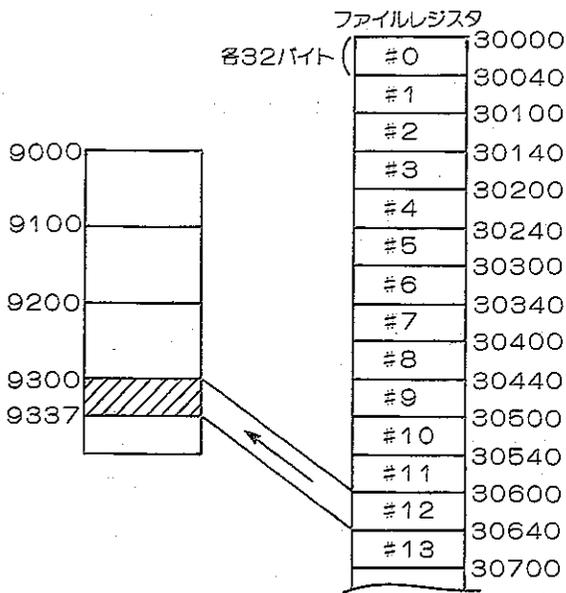
9010	整数部除算時の余り格納
9011	小数点以下2桁演算用
9012	小数点以下2桁演算用(商)
9013	小数点以下2桁演算用(商)
9014	小数点以下2桁演算用(余り)
9015	データ数
9016	データテーブルの先頭アドレス
9017	データテーブルの最終アドレス

(25) 機種コードにより複数の設定値・定数を切替える。

- 外部機器より入力される機種コードにより、TMR, CNTの設定値や、各種の設定値を機種に応じて切替えます。
- 各機種の設定値はファイルレジスタに予め書込んでおきます。
  - a. ファイルレジスタからの抽出用プログラム



(1) 9300~9377の領域に、機種コードに応じてファイルレジスタより、各種設定値を抽出します。



データメモリのブロックは256バイト単位で構成され、上記の例の場合、#5のデータが30000~30377のブロックと、30400~30777の2つのブロックにまたがるため、機種コード#5の転送を行おうとすると、30360~30377の16バイトと、30000~30037の32バイトがレジスタに転送されてしまいます。

●データ数×機種数はファイルレジスタ容量以内となるようにする必要があります。

(例)ファイルレジスタ容量4kバイトの場合、  
1機種当りのデータ数を32バイトとすると、機種数は64が最大となります。  
( $4096 \div 32 = 64$ )

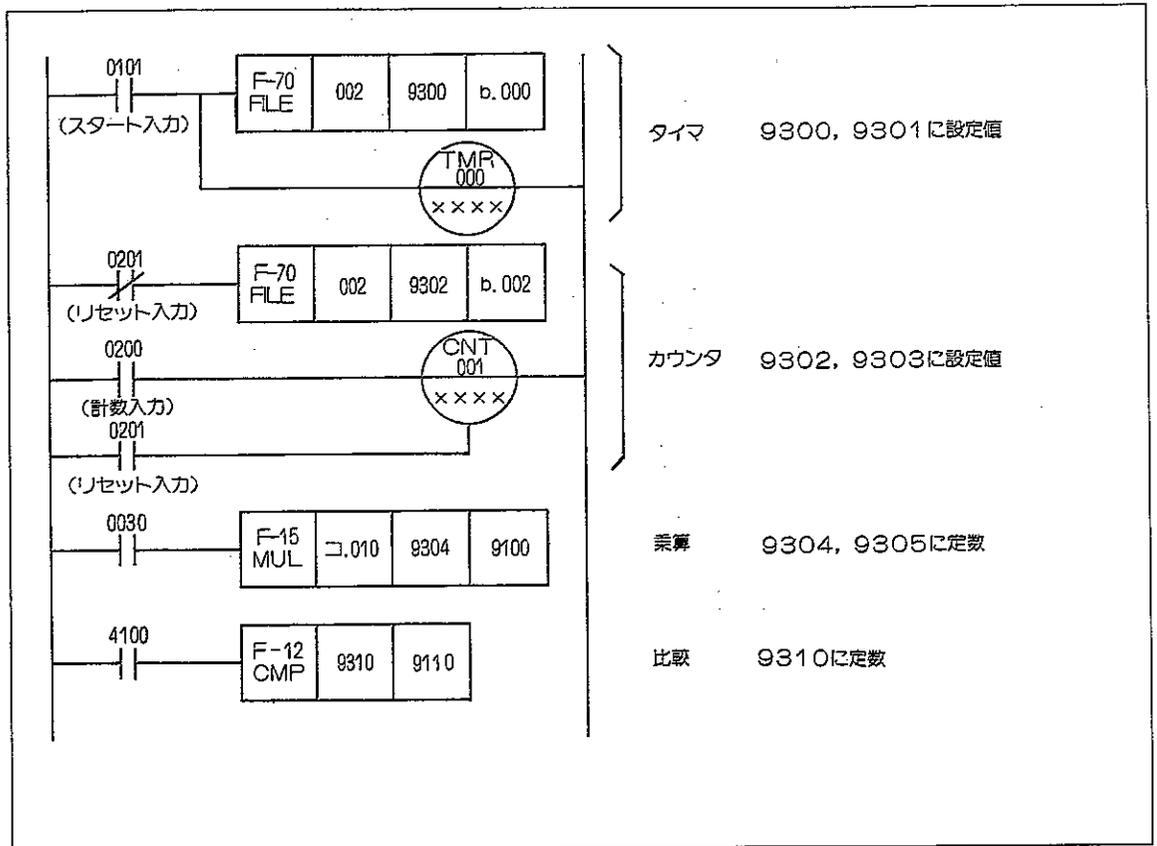
(2) ※1のプログラムで、機種コードからファイルレジスタ側の先頭アドレスを算出しています。

- 機種コードはBCD2桁の数値とします。  
(00~99)
- 1機種当りのデータ数は $2^n$ (2, 4, 8, 16……128)とします。  
最大128バイトとするのは、F-73で指定できる転送バイト数が128バイトに制限されているためです。  
(8進数で200<sub>8</sub>)  
 $2^n$ 以外の数値とすると、1つの機種データのデータが2つのファイルレジスタのブロックにまたがる場合があります。F-73は2つのブロックにまたがる場合、つぎのブロックに継がらず、最初のブロックの先頭に戻って転送が行われます。

(例)データ数を48バイトとしたとき。  
(8進数で060<sub>8</sub>)

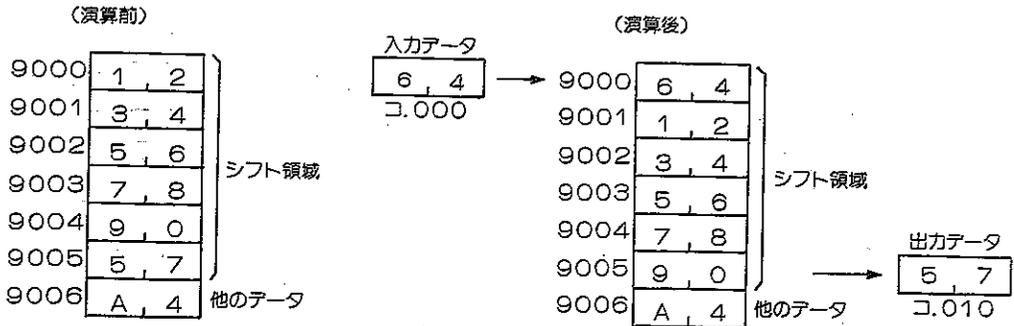
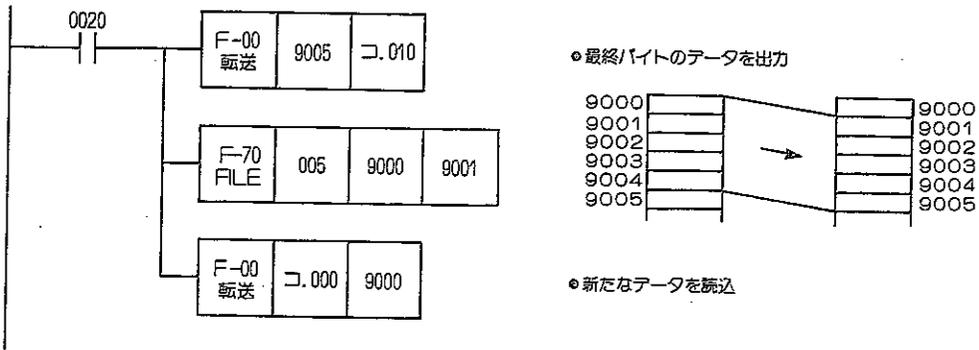
#0	30000~30057	
#1	30060~30137	
#2	30140~30217	
#3	30220~30277	
#4	30300~30357	
#5	30360~30437	←2つのブロック にまたがる
#6	30440~30517	

b. 機種コードにより設定値が変更される演算プログラム例

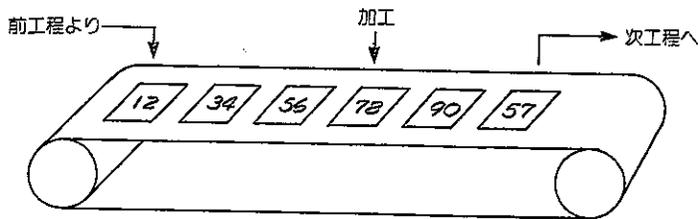


〔26〕 同期型FIFOスタックレジスタ

●任意バイト数(最大128バイト)のシフトレジスタを構成します。



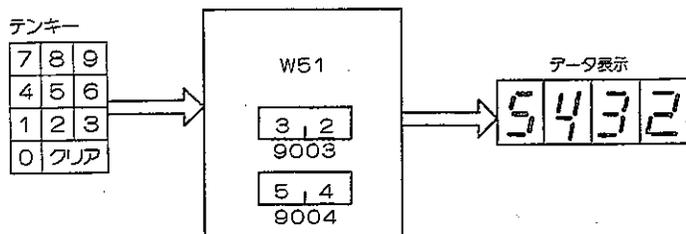
●9000～9005には常に最新のデータが格納されます。



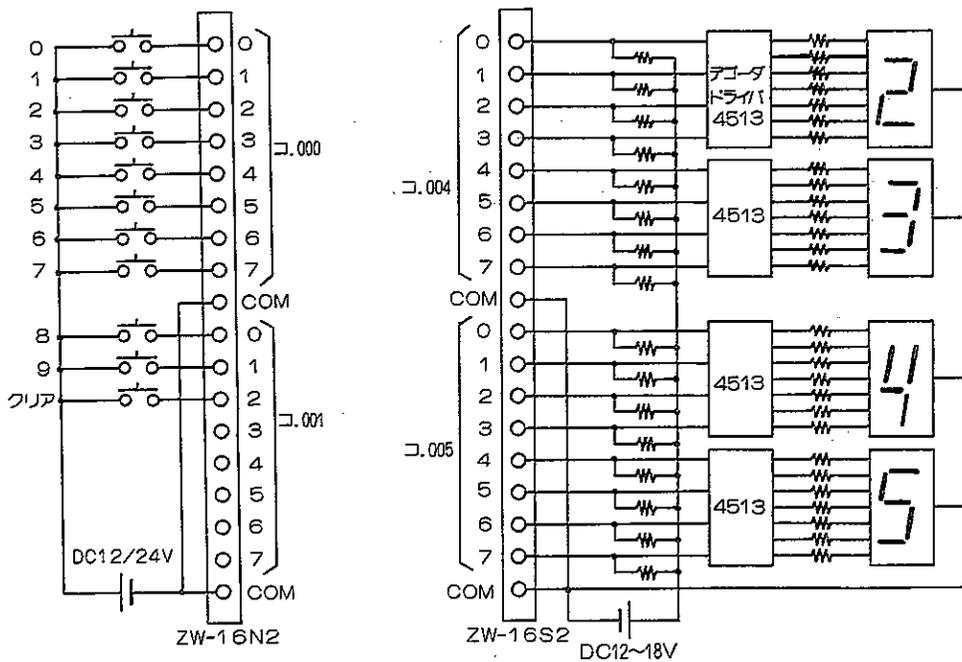
前工程から機種コードを受け、加工後、次の工程へ機種コードを伝えます。

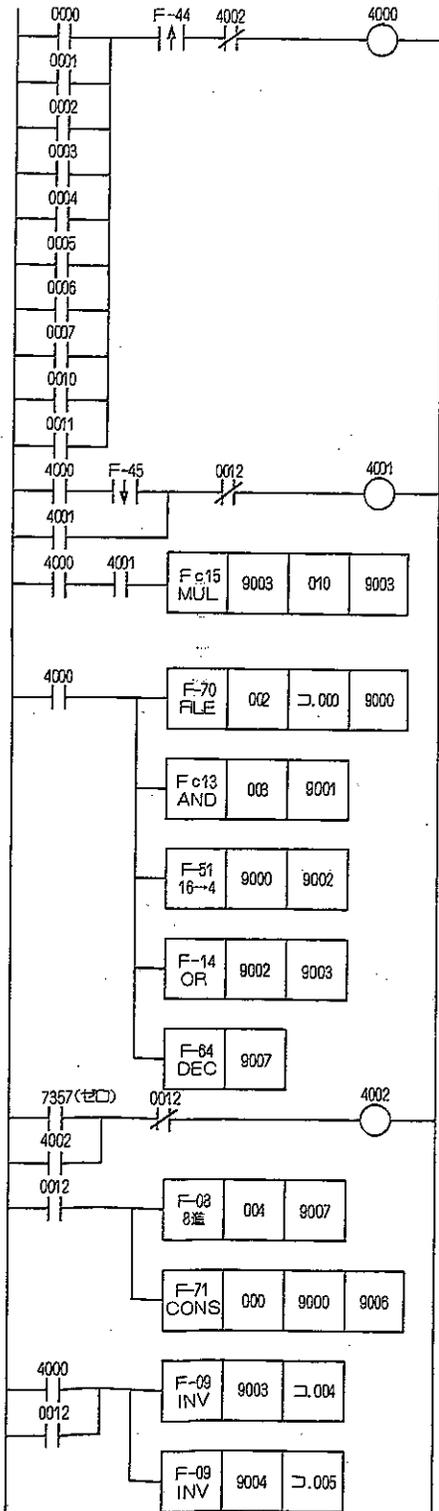
## (27) テンキーからの数値の読込

●テンキーから入力されるBCD4桁の数値をレジスタに読込みます。



### (1) 外部接続

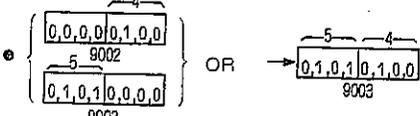
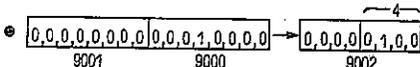
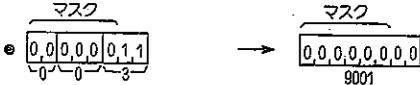
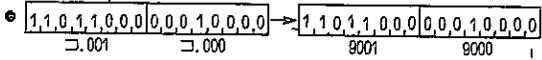
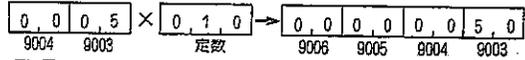




0~9のいずれかのキーを押すと1スキャンサイクルON  
(ただし4桁以上は禁止)

● 1桁目の処理の次のスキャンサイクルからON  
(クリアキーを押すとOFF)

● 2桁目~4桁目の入力でデータレジスタを1桁シフト



● 桁カウンタを-1

● 桁カウンタが000になるとON  
(クリアでOFF)

● クリアキーを押すと  
桁カウンタ(9007)に004を転送

● 9000~9006をクリア

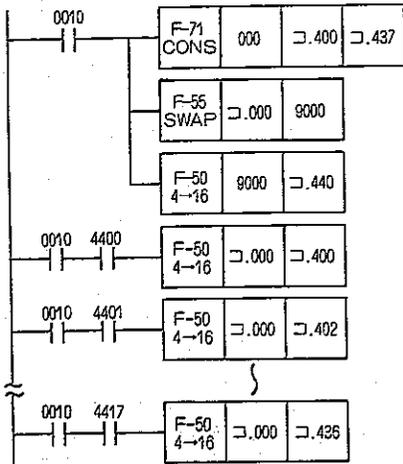
● データ表示

9007	クリア	0, 0, 4
1桁目	表示	0, 0, 3
2桁目	表示	0, 0, 2
3桁目	表示	0, 0, 1
4桁目	表示	0, 0, 0

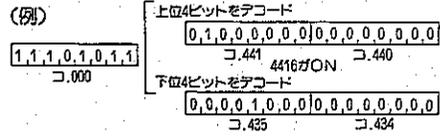
(28) 8→256デコーダ

●8ビットバイナリデータ(0~255)をアコードし、256ビット中の1ビットをONします。[F-50(4→16デコーダ)の拡張]

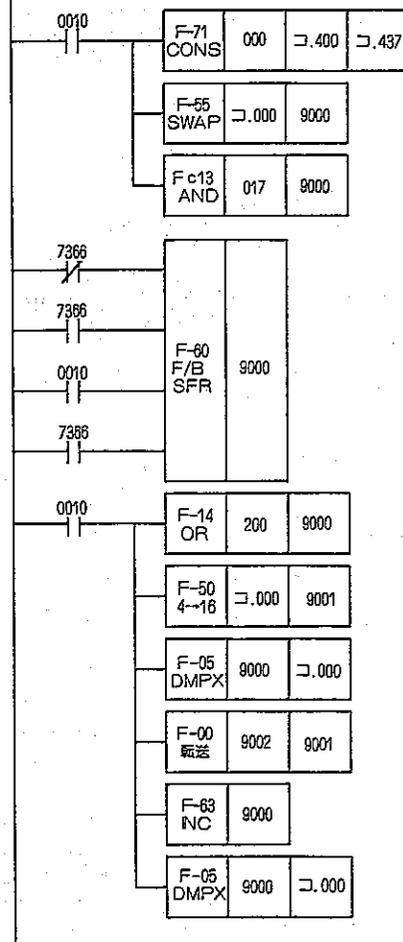
(1)プログラム例①



- デコード結果の書き込み先(4000~4377)をクリア
- 入力データの上位4ビットと下位4ビットを交換
- 9000の下位4ビット(入力データの上位4ビット)をアコード
- 入力データの上位4ビットのデコード結果により、入力データの下位4ビットのデコード結果の出力先を選択



(2)プログラム例②

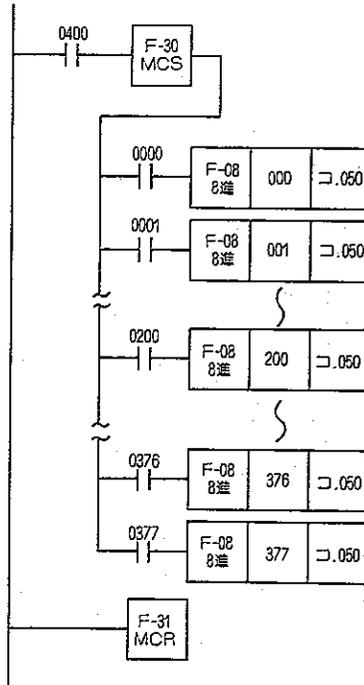


- デコード結果の書き込み先(4000~4377)をクリア
- 入力データの上位4ビットと下位4ビットを交換し、9000に転送
- 9000の上位4ビット(入力データの下位4ビット)をマスク
- 9000の内容を上位方向へ1ビットシフト(9000の内容を2倍する)
- 分配先をコ.400~とするため、コ.400の絶対アドレス200をプラス
- 入力データの下位4ビットをデコード→9001、9002
- 9000の内容でアドレスされるコ.×××の領域(コ.400~)に9001(デコード結果の下位8ビット)を転送
- デコード結果の上位8ビットを9001に転送
- 9000(データポインタ)を+1
- 9000の内容でアドレスされるコ.×××の領域(コ.400~)に9001(デコード結果の上位8ビット)を転送

## (29) 256→8エンコーダ

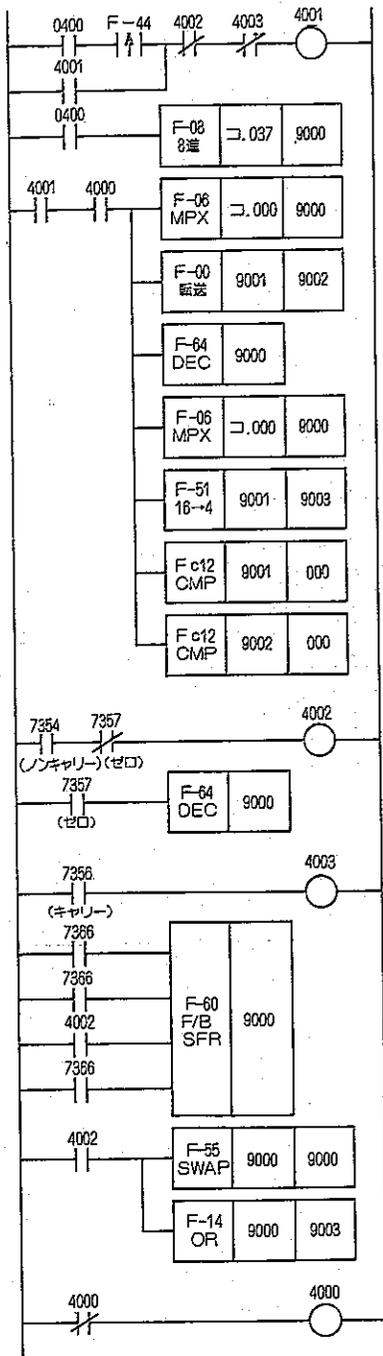
- コ.000~コ.037(0000~0377)の256点のデータをエンコードします。
- 同時に2点以上がONした場合、リレー番号の大きい方を優先し、演算します。

(1)プログラム例① (プログラムメモリ1026語使用)  
1スキャンサイクルで演算



- 0000~0377のうち、ONの入力に応じ8進定数をコ.050に転送。
- リレー番号の小さい方からプログラムすることでリレー番号の大きい方の入力の優先順位を高くしています。

②プログラム例② (プログラムメモリ52語使用)  
32スキャンサイクルで演算



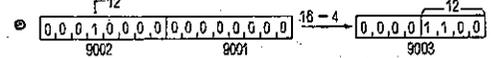
●データポインタに最終アドレス(コ.037)を設定

●データポインタで指示される入力(上8ビット)を9001に転送

●9001→9002

●データポインタを-1

●データポインタで指示される入力(下8ビット)を9001に転送



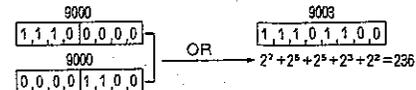
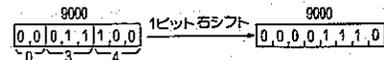
●9001、9002にONしているビットがあるか?

●ONのビットがあれば4002がON

●ONのビットがなければデータポインタを-1

●データポインタが000→377のときON

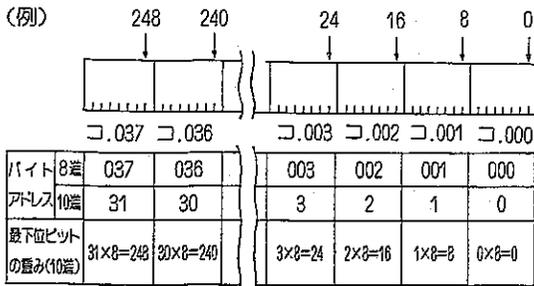
●9001、9002にONのビットがあるとき実行  
(例)データポインタ(9000)が034のときの演算で4002がON



●発振回路

【参考】本プログラムでは2バイトずつ抽出した16ビットのデータを16→4エンコードしたものと(下位4ビット)と、16ビット中の最下位ビットの重み(上位4ビット)を結合しています。入力デー

タをコ.000~コ.037の32バイトに割当てたとき、バイトアドレスとそのバイトの最下位ビットの重みには次の関係が成立します。  
最下位ビットの重み=バイトアドレス×8



本プログラムではこの計算を8進数のまま行っています。

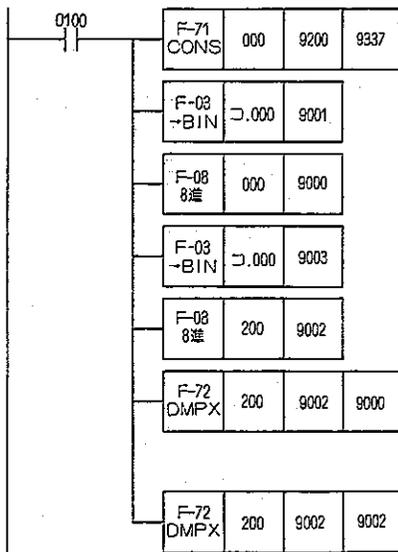
- ┌ F-60で1ビット右シフトすることで÷2
- └ F-55(SWAP)で上位4ビット(0)と下位4ビットの交換(左に4ビットシフト)で×16

$$\Rightarrow \frac{\text{バイトアドレス}}{2} \times 16 = \text{バイトアドレス} \times 8$$

### (30) ファイルレジスタの領域指定クリア

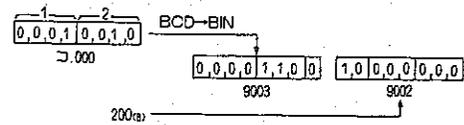
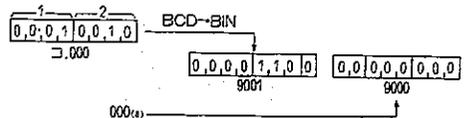
- プログラム等でファイルレジスタをクリアすると、全領域がクリアされますが、本プログラムを使用すると、ファイルレジスタの任意のブロック(256バイト単位)を指定してクリアできます。

#### (1) プログラム例①



コ.000の値	クリアされるファイルレジスタ	
0 0	30000~30177	30200~30377
0 1	30400~30577	30600~30777
0 2	31000~31177	31200~31377
0 3	31400~31577	31600~31777
0 4	32000~32177	32200~32377
0 5	32400~32577	32600~32777
0 6	33000~33177	33200~33377
0 7	33400~33577	33600~33777
0 8	34000~34177	34200~34377
0 9	34400~34577	34600~34777
1 0	35000~35177	35200~35377
1 1	35400~35577	35600~35777
1 2	36000~36177	36200~36377
1 3	36400~36577	36600~36777
1 4	37000~37177	37200~37377
1 5	37400~37577	37600~37777

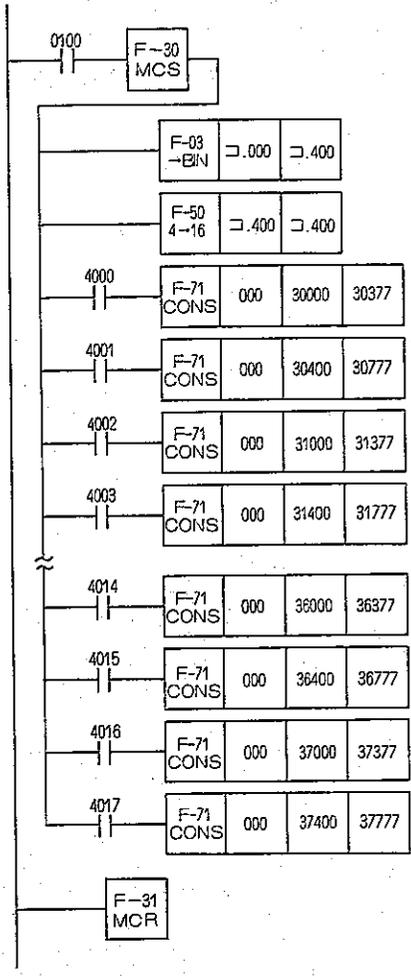
- レジスタ9200~9377の128バイトをクリア



- 9002で指示されるレジスタのアドレス...9200  
9001、9000で指示されるファイルレジスタのアドレス.....36000  
9200~9377の128バイト(内容は全て0)  
→36000~36177の128バイト
- 9002で指示されるレジスタのアドレス...9200  
9003、9002で指示されるファイルレジスタのアドレス.....36200  
9200~9377の128バイト(内容は全て0)  
→36200~36377の128バイト

注1 レジスタ9200~9377をファイルレジスタのクリア用として使用します。

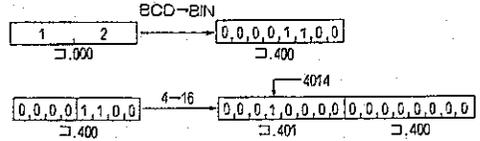
②プログラム例②



●コ.000のBCD数(00~15)をBINに変更

●さらに4→16デコード

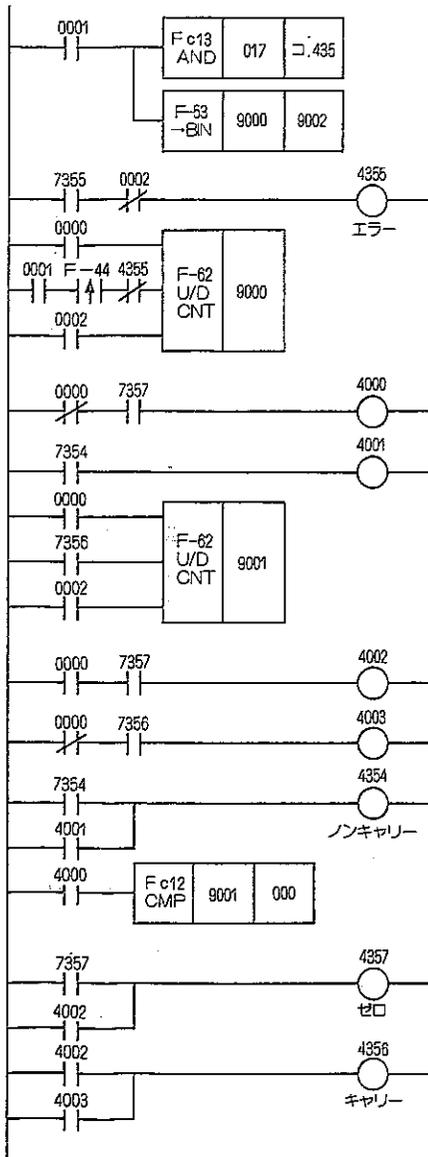
(例)



4014がONであるから36000~36377をクリア

### (31) BCD4桁のアップ・ダウンカウンタ

● アップ・ダウン指示入力に従ってBCD4桁(0~9999)を加算、減算し、演算結果によって補助リレーをフラグとして出力します。



● フラグのクリア

● BCD4桁か

● 0000 (ON 下2桁+1  
OFF 下2桁-1)

●  $\times \times 01-1$

● 0000 (ON 上2桁+1  
OFF 上2桁-1)

● 9999+1

● 0000-1

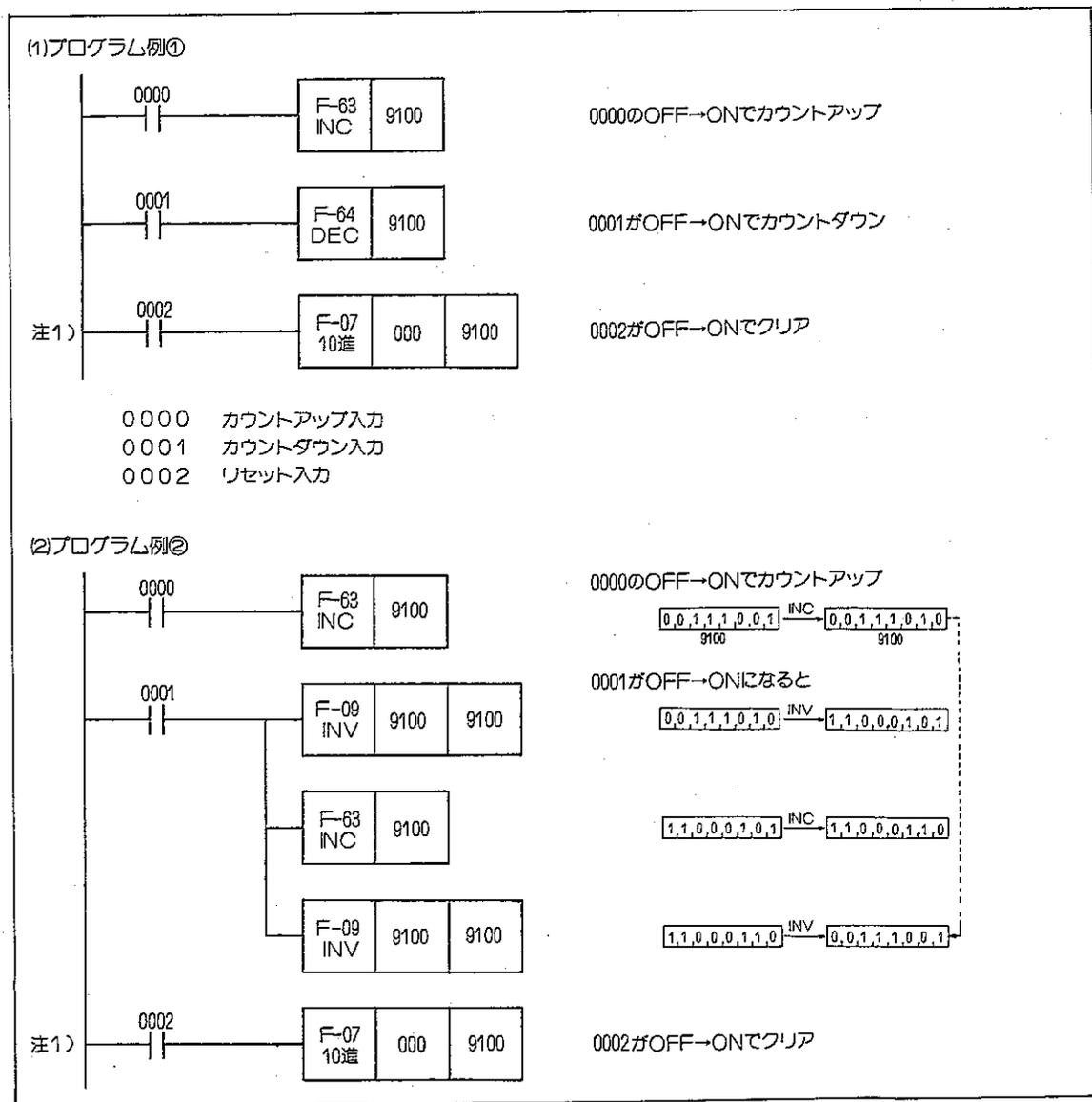
● 00001-1か

- 0000.....アップ・ダウン指示入力  
(ON — アップ (加算)  
OFF — ダウン (減算))
- 0001.....カウント入力(OFF→ONを検知)
- 0002.....リセット入力(ONでリセット)
- 9000.....下2桁(00~99)
- 9001.....上2桁(00~99)
- 4354.....ノンキャリーフラグ
- 4355.....エラーフラグ
- 4356.....キャリーフラグ
- 4357.....ゼロフラグ

アップ・ダウン 指示入力	演算結果	ノキャリー 4354	エラー 4355	キャリー 4356	ゼロ 4357
ON	9999+1	0	0	1	1
	0000~9998+1	1	0	0	0
	BCD以外の数値	0	1	0	0
OFF	0000-1	0	0	1	0
	0001-1	1	0	0	1
	0002~9999-1	1	0	0	0
	BCD以外の数値	0	1	0	0

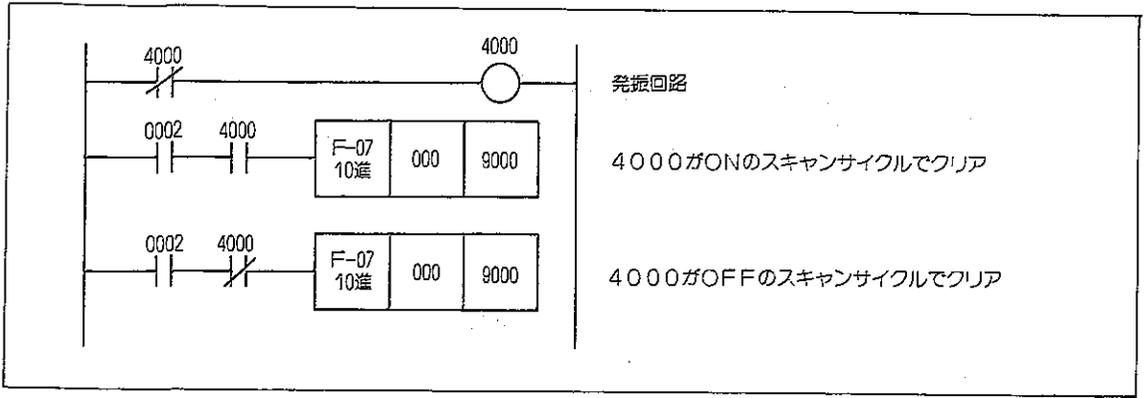
●リセット入力(0002)がOFF時に計数可能となります。アップ・ダウン指示入力(0000)がON時加算、OFF時減算のカウンタとして動作します。なお、使用のレジスタ(9000、9001)がBCDコードでない数値(A0、7F等)となった場合、エラーフラグ(4355)がONし、演算しません。

### (32) バイナリーアップダウンカウンタ



注1) 例①、例②とも0002のOFF→ONでクリアされるため、リセット優先にはなりません。0002がONの間でも0000、0001がOFF

→ONになるとカウントアップまたはカウントダウンします。リセット優先とするためには、注1)のプログラムを次のようにする必要があります。



# 付 録

## システムメモリ使用一覧表 (W16/W51)

#000 } #077	} PC内部処理に使用 }
#100 #101 #102 } #107	……周辺装置からの停止(D0) 書込許可(D1) フラグ ……オプションモジュールからの停止。書込許可要求フラグ } PCの内部処理に使用 }
#110 #111 #112 } #152	……リモート親局用接続子局数 (SW4 注. 固定割付) ……リモート親局用I/O先頭アドレス (SW2, SW1 注. 固定割付) } PCの内部処理に使用 }
#153 } #177	} PCの内部処理に使用 }
#200 #201 #202 #203 #204 #205 #206 #207	キープリレー領域設定 (300 <sub>0</sub> は7000から) TMRのリセット条件設定 CNTのリセット条件設定 出力保持アドレス設定 プログラムメモリ容量設定 ファイルレジスタ容量設定 ……CLリンク(コマンドモード)又はBLリンクの子局番号(SW2, SW1の値 8進数) ……DL1 PCの局番号 (SW2, SW1の値 8進数)
#210 } #217	} 自己診断 故障コード }
#220 #221 #222 #223	……オプションエラーコード (RM1……30,31,38 DL1…20 DL9…70,71) ……リモート I/O 異常局番 (使用予約) (使用予約)
#224 } #236	} リモート I/O親局 任意割付に使用 }
#237 } #257	} 使用予約 }
#260 } #377	} データリンク 親局 パラメータ設定 DL1…#300~#361 DL9…#260~#377 }

**注1** #000~#177の128バイトは、コントロールユニット内のCPUが、使用する領域です。  
データの読出だけにご利用下さい。

## データメモリ特殊リレー一覧表

### オプション用特殊リレー領域

7300	
7301	
7302	
7303	
7304	DL9 通信監視フラグ (子局側)
7305	DL9 イニシャルシーケンス完了フラグ (親局)
7306	DL9 個別読出フラグ (親局側)
7307	DL9 リンク動作フラグ (親局側)
7310	CL2 (文字列) エラー
7311	CL2 (文字列) 出力レディ
7312	CL2 (文字列) トリガ条件
7313	CL2 (文字列) エラー
7314	CL2 (文字列) 出力レディ
7315	CL2 (文字列) トリガ条件
7316	リモートI/O動作フラグ
7317	CL,BLリンク グローバルアドレス受信
7320	局番 00
7321	局番 01
7322	局番 02
7323	局番 03
7324	局番 04
7325	局番 05
7326	局番 06
7327	局番 07
7330	局番 10
7331	局番 11
7332	局番 12
7333	局番 13
7334	局番 14
7335	局番 15
7336	局番 16
7337	局番 17

### 特殊リレー領域

7340	
7341	
7342	
7343	自己診断結果の異常コードを収納する特殊レジスタでバイトアドレスコ.734として扱います。
7344	
7345	
7346	
7347	
7350	乗除算命令桁数選択(W51のみ)
7351	
7352	
7353	
7354	ノンキャリアフラグ
7355	エラーフラグ
7356	キャリアフラグ
7357	ゼロフラグ
7360	0.1秒クロック
7361	
7362	
7364	
7365	1秒クロック
7366	設定値変更スイッチ
7366	常時OFFの接点
7367	ゼロクロススイッチ
7370	メモリ異常
7371	CPU異常
7372	電池異常
7373	入出力異常
7374	オプション異常
7375	
7376	ROM異常
7377	電源異常

### オプション用特殊レジスタ領域

9370	出力フォーマットの先頭アドレス
9371	CL2 (文字列)
9372	データメモリの先頭アドレス
9373	CL2 (文字列)
9374	出力フォーマットの先頭アドレス
9375	CL2 (文字列)
9376	データメモリの先頭アドレス
9377	CL2 (文字列)

# リモート I/Oで使用するデータメモリー一覧

## (1) システムメモリ

#101 オプションカード停止要求 (リモート I/O親局モジュール SW3の4が、ONで、子局異常の時)



#110 接続局数 (任意割付の時は無視してください)

#111 先頭アドレス (任意割付の時は、無視してください)

#210~#217 PCの自己診断結果の異常コード格納領域  
(リモート I/O 親局モジュール等のオプションモジュールが、故障するとエラーコード53が入る)

#220 エラーコード (BCD) (データリングでも使用しています。)

30: 親局のスイッチ設定異常

31: 子局の異常

38: BCCチェックコードエラー (任意割付時)

#221 異常局番

#224 子局の台数

#225 子局1の I/Oバイト数

#226 // 2 //

#227 // 3 //

#230 // 4 //

#231 // 5 //

#232 // 6 //

#233 // 7 //

#234 リモート I/O先頭アドレス

#235 BCCチェックコード

#236 リモートスイッチ

} リモート I/O 任意割付時に使用  
(数値設定は、10進数と、16進数で設定してください)

## (2) キープリレー

7316……リモート I/O 動作フラグ (正常時 ON)

コ 000~コ 727……リモート子局用の I/O エリアに使えます。

## (3) ROMバージョン

リモート I/O 任意割付は、ROMバージョン(V3.0以上)で使えます。親局・子局とも、同一バージョンのROMをご使用下さい。

## データリンク(DL1)で使用するデータメモリー一覧

### (1) システムメモリー(親局のみ)

#207……PCの局番(SW1,SW2の状態)

#210～#217……PCの自己診断結果の異常コード格納領域

(データリンクモジュール等のオプションモジュールが、故障するとエラーコード「53」が入ります。)

#220……エラーコード(BCD)(リモートI/O親局でも使用しています。)

20:パラメータ設定ミス

#300～#357……各PCの転送バイト数および受信局の指定情報を登録します。

	7	6	5	4	3	2	1	0	(ビット)
#300	PC0送信バイト数								
#301	07	06	05	04	03	02	01	00	} どの局からのデータを 受信するか指定
#302	17	16	15	14	13	12	11	10	
	ON……受信				OFF……受信せず				

#360……接続局数(回線で接続されているPCの台数を蓄込みます。)

#361……リンクスタートスイッチ

0:リンク動作を停止します。

1:システムメモリーの内容を取込み、リンク動作を開始します。

リンクパラメータ設定

### (2) データメモリー

7320～7337……リンク動作フラグ(親局、子局とも通信が、正常に行なわれている局が、ON)

	7	6	5	4	3	2	1	0
コ 732	PC07	PC06	PC05	PC04	PC03	PC02	PC01	PC00
コ 733	PC17	PC16	PC15	PC14	PC13	PC12	PC11	PC10

# データリンク(DL9)で使用するデータメモリー一覧

## (1) システムメモリ (親局のみ)

#210～#217……PCの自己診断結果の異常コード格納領域  
 (データリンクモジュール等のオプションモジュールが、故障するとエラーコード“53”が入ります。)

#220……エラーコード〔BCD〕(リモートI/O親局でも使用しています。)  
 コード70〔BCD〕: 設定ミス  
 コード71〔BCD〕: BCCエラー

#260……接続子局数 1～17〔8進数〕(親局システムメモリのみ)

#261～#277……親局での各子局に対する送信エリアの先頭アドレス

#301～#317……親局での各子局からの受信エリアの先頭アドレス

#320……フラグ先頭アドレス(絶対アドレスで、000～322<sup>(a)</sup>の範囲としてください)

#321～#337……各子局でのリンクエリアの先頭アドレス

#340～#375……各子局との通信バイト数、通信エリアの設定(MSB/ビット)

	MSB/ビット			
#340	R	PC/親局 → 子局	R = 0	親局 コ.000～コ.727
#341	r	PC/子局 → 親局	r = 0	子局 コ.000～コ.727
			r = 1	親局 9000～9377
			r = 1	子局 9000～9377

#376……BCCチェックコード

#377……リンクスタートスイッチ  
 00 リンク停止  
 01 リンクスタート  
 80(H) BCC計算: 演算結果が、#376に入ります。

通信パラメータ設定

## (2) データメモリ

7304……通信監視フラグ(子局側だけのフラグ。親局が正常に運転中でかつ通信も正常な時 ON)

7305……イニシャルシーケンス完了フラグ(親局のフラグ。リンクパラメータを子局に送り終ると ON)

7306……個別読出フラグ(親局のフラグ。ON時には、#320で設定したアドレスからの6バイトにフラグを表示)

		7	6	5	4	3	2	1	0		
フラグ先頭アドレス→		07	06	05	04	03	02	01		}	
+1		17	16	15	14	13	12	11	10		①通信監視フラグ
+2		07	06	05	04	03	02	01			②PC運転状態監視 フラグ(I)
+3		17	16	15	14	13	12	11	10		③PC運転状態監視 フラグ(II)
+4		07	06	05	04	03	02	01			
+5		17	16	15	14	13	12	11	10		

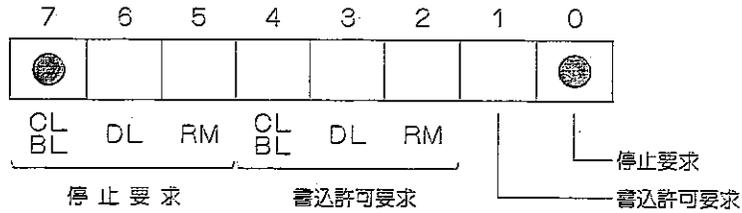
7307……リンク動作フラグ(親局フラグ。全局が、運転中で通信も正常な時 ON)

# コンピューターリンクとBRAINリンクで使用するデータメモリー一覧

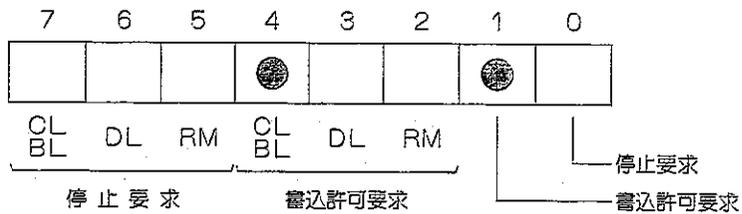
## (1) システムメモリー

#101……(コンピューターリンクのコマンドモードとBRAINリンクのみ)

◦HLTコマンドを受信すると、PCは停止し、下記のビットが立ちます。



◦EWRコマンドでモード2を受信すると、#101に下記のビットが立ちます。



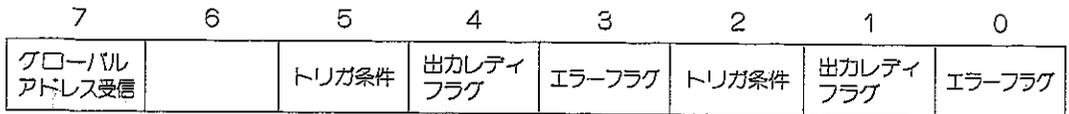
#206……コンピューターリンク又は、BRAINリンクの子局番号 (SW2, SW1の設定 8進数)

#210~#217……PCの自己診断結果の異常コード格納領域

(コンピューターリンクモジュール等のオプションモジュールが、故障するとエラーコード「53」が入ります。)

## (2) データメモリー

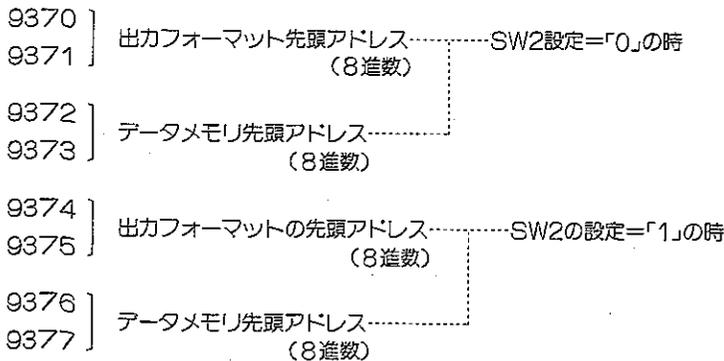
コ.731の6点分を使用 (7316はリモートI/O用フラグ)



(リンクモード)

SW2設定「1」  
(文字列出力モード)

SW2設定「0」  
(文字列出力モード)



- この製品に関するご意見・ご質問は下記へお寄せください。

### シャープ・マニファクチャリング・システム株式会社

本社	☎(0729) 91-0681	大阪府八尾市跡部本町4丁目1番33号
仙台	☎(022) 288-1131	仙台市若林区卸町東3丁目1番27号
東京	☎(03)3235-7351	東京都新宿区市谷八幡町8番地
名古屋	☎(052) 332-2691	名古屋市中川区山王3丁目5番5号
豊田	☎(0565) 29-0131	豊田市山之手8丁目124番コスモビル山之手1階
大阪	☎(06) 606-5459	大阪市阿倍野区西田辺町1丁目19番20号
広島	☎(082) 248-0131	広島市中区中町9番8号

- アフターサービスなどについてのお問い合わせ先

#### シャープお客様相談窓口

#### シャープシステムサービス㈱

仙台 技術センター	☎(022) 288-9161	仙台市若林区卸町東3丁目1番27号
宇都宮 技術センター	☎(0286) 34-0256	宇都宮市不動前4丁目2番41号
前橋 技術センター	☎(0272) 52-7311	前橋市問屋町1丁目3番7号
東京フィールド サポートセンター	☎(03)3810-9962	東京都北区田端新町2丁目2番12号
横浜 技術センター	☎(045) 753-9583	横浜市磯子区中原1丁目2番23号
静岡 技術センター	☎(054) 283-9497	静岡市曲金6丁目8番44号
名古屋 技術センター	☎(052) 332-2671	名古屋市中川区山王3丁目5番5号
金沢 技術センター	☎(0762) 49-9033	石川県石川郡野々市町字御経塚町1096の1
大阪フィールド サポートセンター	☎(06) 794-9671	大阪市平野区加美南3丁目7番19号
広島 技術センター	☎(082) 874-6100	広島市安佐南区西原2丁目13番4号
高松 技術センター	☎(0878) 23-4980	高松市朝日町6丁目2番8号
松山 技術センター	☎(0899) 73-0121	松山市高岡町178の1
福岡 技術センター	☎(092) 572-2617	福岡市博多区井相田2丁目12番1号

\*上記の所在地・電話番号などは変わることがあります。その節はご容赦願います。

## シャープ株式会社

本社	〒545	大阪市阿倍野区長池町22番22号
		電話 (06) 621-1221 (大代表)
東京支社	〒261	千葉市美浜区中瀬1丁目9番2号
		電話 (043)297-1221 (大代表)

お客様へ……お買いあげ日、販売店名を記入されますと、修理などの依頼のときに便利です。

お買いあげ日	年	月	日
販売店名			
	電話 ( )	局	番