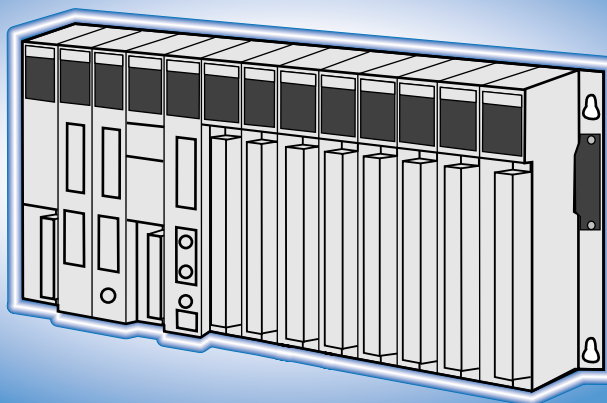
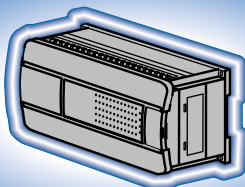
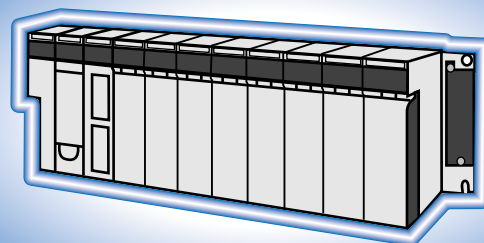
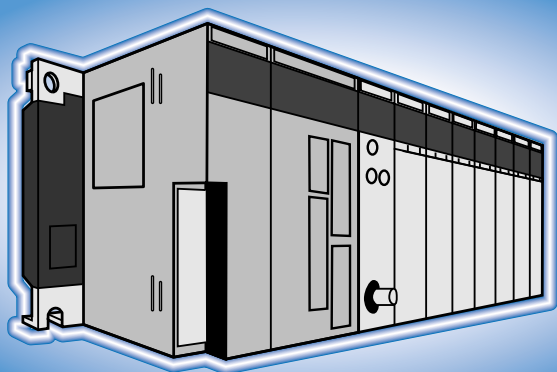


## シャーププログラマブルコントローラ ニューサテライト JW

### プログラム例集



# はじめに

本書は、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JWシリーズの応用命令の豊富さ、使いやすさをご理解いただくとともに、多種多様なプログラム例により、プログラム設計のヒントを提供するものです。

本書は、次の4つの独立した章と付録より構成されていますので、必要な箇所のみお読みください。

- 第1章 基本回路
- 第2章 データ処理回路
- 第3章 特殊命令の使い方
- 第4章 応用プログラム事例
- 付 録 命令語一覧

なお、命令の詳しい説明などは、各P Cの取扱説明書、ユーザズマニュアル、プログラミングマニュアルをご参照ください。

## おねがい

- ・ 本書の内容については十分注意して作成しておりますが、万一ご不審な点、お気づきのことがありましたら、本書の裏表紙に記載のシャープマニファクチャリングシステム株式会社までご連絡ください。
- ・ 本書の内容の一部または全部を、無断で複製することを禁止しています。
- ・ 本書の内容は、改良のため予告なしに変更することがありますので、あらかじめご了承ください。

# 目 次

本書を利用するにあたって

## 第1章 基本回路 ..... 1.1~1.18

1-1	常時ON回路	1.1
1-2	電源投入時に1パルス発生させる回路	1.2
1-3	発振回路	1.3
1-4	立上り微分	1.4
1-5	立下り微分	1.4
1-6	自己保持回路(リセット優先)	1.5
1-7	自己保持回路(セット優先)	1.6
1-8	優先回路	1.7
1-9	オルタネート回路	1.8
1-10	nビットシフトレジスタ	1.9
1-11	オンディレイタイマ	1.10
1-12	オフディレイタイマ	1.10
1-13	オン・オフディレイタイマ	1.11
1-14	入力立上り時ワンショットタイマ(1)	1.11
1-15	入力立上り時ワンショットタイマ(2)	1.12
1-16	入力立下り時ワンショットタイマ	1.12
1-17	立上り、立下りワンショットタイマ	1.13
1-18	オンディレイワンショットタイマ	1.13
1-19	等間パルス発生回路	1.14
1-20	デューティ可変パルス発生回路(1)	1.15
1-21	デューティ可変パルス発生回路(2)	1.15
1-22	長時間タイマ(1)	1.16
1-23	長時間タイマ(2)	1.16
1-24	大容量カウンタ(1)	1.17
1-25	大容量カウンタ(2)	1.17
1-26	計数入力の立上り、立下りで計数するカウンタ	1.18

## 第2章 データ処理回路 ..... 2.1~2.74

2-1	演算実行条件の変更	2.1
2-2	レジスタのクリア	2.2
2-3	データのマスク	2.2
2-4	ビットのセット/リセット	2.3
2-5	数の分解	2.4
2-6	数の合成	2.5
2-7	設定値との比較	2.6
2-8	ウィンドウコンパレータ	2.8
2-9	不感帯をもつ比較回路	2.9

2-10	複数のセットポイントを持つタイマ	2-10
2-11	BCD減算結果を符号付絶対値で求める	2-11
2-12	BCD 8桁の乗算	2-12
2-13	BCD 8桁÷BCD 2桁	2-14
2-14	BCD 4桁の除算	2-15
2-15	ドラムシーケンサ	2-19
2-16	タイマ現在値の外部出力	2-20
2-17	カウンタ現在値の外部出力	2-23
2-18	タイマの設定値を外部機器から入力	2-25
2-19	カウンタの設定値を外部機器から入力	2-27
2-20	複数のタイマ、カウンタの設定値を外部機器から入力	2-30
2-21	時・分・秒設定減算タイマ	2-34
2-22	ダイナミック入力	2-35
2-23	ダイナミック出力	2-37
2-24	同期型FIFOスタックレジスタ	2-39
2-25	データの分配	2-40
2-26	データの抽出	2-43
2-27	データの挿入	2-46
2-28	データの削除	2-48
2-29	データの検索(1)	2-50
2-30	データの検索(2)	2-51
2-31	データの照合	2-53
2-32	BCD 4桁の最小値・最大値を求める	2-55
2-33	BCD 2桁の数値の平均値を求める	2-57
2-34	ファイルレジスタの領域指定クリア	2-59
2-35	テンキーからの数値の読込	2-60
2-36	8→256デコーダ	2-62
2-37	256→8エンコーダ	2-63
2-38	7SEGエンコーダ	2-65
2-39	グレイコード→バイナリコード変換	2-66
2-40	BCD6桁のアップ・ダウンカウンタ	2-67
2-41	24ビットシフトレジスタ	2-69
2-42	スキャンタイムの測定	2-71
2-43	BCCコードの生成	2-72
2-44	高速カウンタの現在値を符号付きで表示する	2-73

### 第3章 特殊命令の使い方 ..... 3-1~3-22

3-1	I/Oリフレッシュ命令と割込処理	3-1
3-2	特殊I/Oデータリフレッシュ命令 (F-81)	3-5
3-3	特殊I/Oからの読込 (F-85)、特殊I/Oへの書込 (F-86)	3-8
3-4	送信命令 (F-204)、受信命令 (F-205)	3-12
3-5	MD (メンテナンスディスプレイ) 命令 (F-20)	3-17

第4章 応用プログラム事例 ..... 4.1~4.31

4-1	位置ずれ検出	4.1
4-2	鉄板の折り曲げ長さを求める	4.2
4-3	基板カセットの検索・削除処理	4.3
4-4	自動倉庫の入庫・出庫管理	4.5
4-5	アナログ出力電圧設定	4.8
4-6	データ照合	4.10
4-7	帳票用データの作成	4.14
4-8	スケール変換	4.17
4-9	昼夜判定処理	4.24
4-10	運転時間の切替	4.26
4-11	コンピュータリンクによるPC間通信	4.29

付録 命令語一覧 ..... 付.1~付.11

# 本書を利用するにあたって

## 1. 適応機種について

・本書のプログラムは、以下のJWシリーズPCで使用できます。

PC機種名	本体（CPU部）機種名
JW10	JW-1324K、JW-1342K、JW-1424K、JW-1442K、JW-1624K、JW-1642K
JW20H	JW-21CU、JW-22CU
JW30H	JW-31CUH、JW-32CUH、JW-33CUH JW-31CUH1、JW-32CUH1、JW-33CUH1、JW-33CUH2、JW-33CUH3
JW50H	JW-50CUH
JW70H	JW-70CUH
JW100H	JW-100CUH
J-board	Z-311J、Z-312J

(注1) プログラムの中には、適応PCが限定されるものがあります。適応PCを記述していないプログラムは上記の全PCで使用できます。

(注2) J-boardはJW20Hと命令語体系が同じですので、適応PCにJW20Hと記載されているプログラムはJ-boardでも使用できます。

(注3) 従来機種（W10/W16/W51/W100/W70H/W100H）についても、プログラムによっては適応できるものがあります。

## 2. 数値等の表現方法について

1) アドレス、設定値等の数値表現方法は下記を採用しています。

8進数 : 数値に (8) を付加 例 377<sub>(8)</sub>  
10進数 : 数値に (D) を付加または数値のみ 例 255<sub>(D)</sub>、255  
16進数 : 数値に (H) を付加 例 FF<sub>(H)</sub>

2) レジスタのデータ長の表現方法は下記を採用しています。

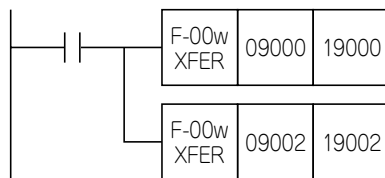
バイト（8ビット）処理 : レジスタ番号のみ  
例 09000  
ワード（16ビット）処理 : レジスタ番号にwを付加  
例 09000w (09000,09001使用)  
ダブルワード（32ビット）処理 : レジスタ番号にdを付加  
例 09000d (09000~09003使用)

### 3. プログラムについて

- 1) プログラムは必要最小限の構成となっています。適用するシステムに合わせて、指令信号を有効、無効にする条件信号やインターロック信号を付加する必要があります。
- 2) リレー番号などのデータメモリ番号は、JW10の番号を使用しています。これは、JW10の入力と出力のリレー割付が、固定されている（入力：00000～ 出力：00400～）ことと、JW10のレジスタなどのデータメモリ領域が他のPCのデータメモリ領域に含まれるからです。  
ただし、最大制御入出力点数が256点のPC（JW-21CU、J-board）は入出力リレー領域が00000～00377ですので、出力リレー番号は使用可能なリレー番号に置き換えてください。  
なお、リレー番号等は適当に付けてあります。使用するシステムに合わせて番号を付ける必要があります。
- 3) 応用命令は、ダブルワードの命令（F-00d、F-10d等）は使用していません。これは、JW10には、ダブルワードの命令がないためです。従って、JW30H等の機種において、ダブルワードの命令に書き換えるとプログラムが簡略されるものがあります。

例) レジスタ09000～09003のデータをレジスタ19000～19003に転送する場合

a) 本書のプログラム



または



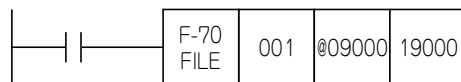
b) JW30Hで可能なプログラム



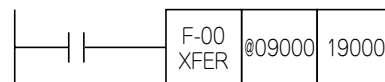
- 4) 応用命令は、間接アドレスの使用できる命令として、F-70（nバイト一括転送）とF-70w（nワード一括転送）のみ使用しています。これは、JW10とJW20Hには、他に間接アドレスの使用できる命令がないためです。従って、JW30H等の機種において、間接アドレスを使用するとプログラムが簡略されるものがあります。

例1) レジスタ09000～09002を間接アドレスとするレジスタの内容をレジスタ19000に転送する場合

a) 本書のプログラム

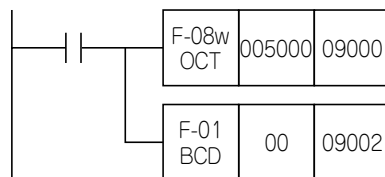


b) JW30Hで可能なプログラム

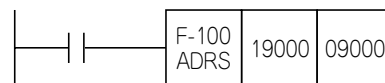


例2) レジスタ19000の間接アドレスをレジスタ09000～09002に設定する場合

a) 本書のプログラム



b) JW30Hで可能なプログラム



- 5) END命令（F-40）は省略しています。

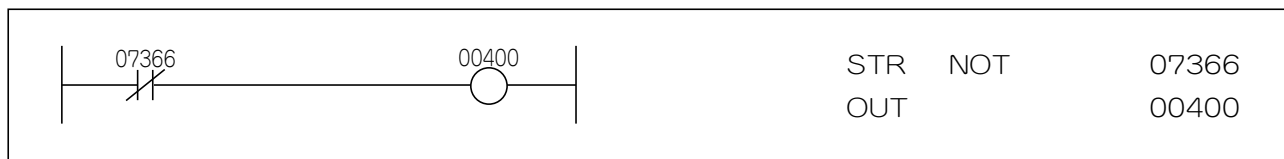
# 第1章 基本回路

この章では、ラダープログラムにおいて基本となる、接点、コイル、タイマ、カウンタ命令を使用した基本的な回路について紹介します。

1-1	常時ON回路	1-1
1-2	電源投入時に1パルス発生させる回路	1-2
1-3	発振回路	1-3
1-4	立上り微分	1-4
1-5	立下り微分	1-4
1-6	自己保持回路(リセット優先)	1-5
1-7	自己保持回路(セット優先)	1-6
1-8	優先回路	1-7
1-9	オルタネート回路	1-8
1-10	nビットシフトレジスタ	1-9
1-11	オンディレイタイマ	1-10
1-12	オフディレイタイマ	1-10
1-13	オン・オフディレイタイマ	1-11
1-14	入力立上り時ワンショットタイマ(1)	1-11
1-15	入力立上り時ワンショットタイマ(2)	1-12
1-16	入力立下り時ワンショットタイマ	1-12
1-17	立上り、立下りワンショットタイマ	1-13
1-18	オンディレイワンショットタイマ	1-13
1-19	等間パルス発生回路	1-14
1-20	デューティ可変パルス発生回路(1)	1-15
1-21	デューティ可変パルス発生回路(2)	1-15
1-22	長時間タイマ(1)	1-16
1-23	長時間タイマ(2)	1-16
1-24	大容量カウンタ(1)	1-17
1-25	大容量カウンタ(2)	1-17
1-26	計数入力の立上り、立下りで計数するカウンタ	1-18



# 1 - 1 常時 ON 回路



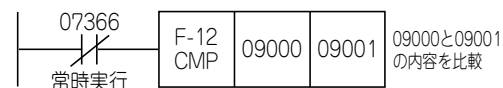
● 特殊リレーの07366は常時OFFの接点のため、コイル00400は常時ONとなります。00400を「停止時OFFとなる領域」に設定すると、プログラムモードに切換えたときや、自己診断の結果、本体が停止するとOFFになるため、PCのRUN表示として使用できます。

**注1** 出力が「停止時OFFとなる領域」の設定は、以下のシステムメモリの設定で行います。詳細は各PCのプログラミングマニュアルまたは、ユーザーズマニュアルをご参照ください。

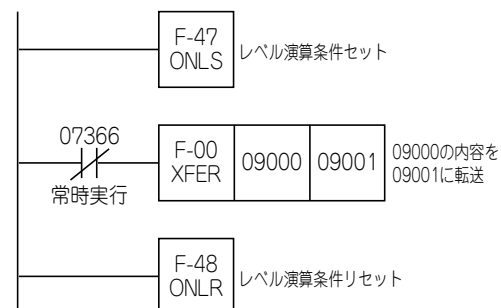
PC機種名	システムメモリ
JW10	#206
JW20H	#232, #233
JW30H	#232, #233, #252, #253
JW50H/70H/100H	#232, #233

**参考** 応用命令の演算を毎演算サイクル実行したい場合、応用命令の入力条件に07366のb接点を使用します。

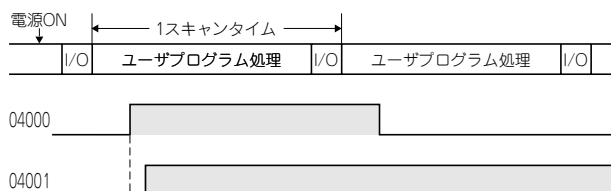
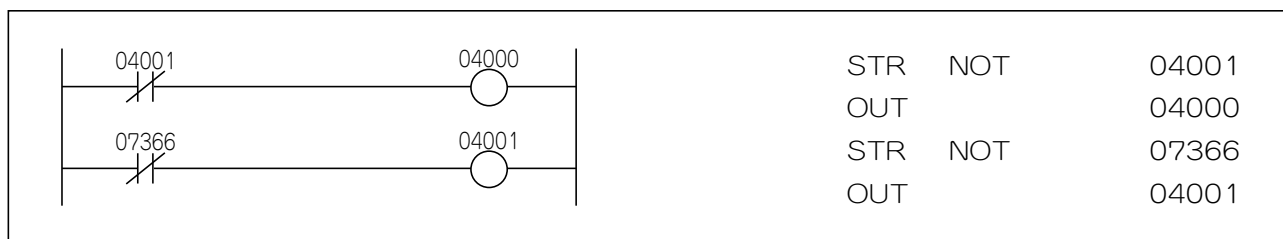
(例1) 入力信号がONの時、実行する命令の場合



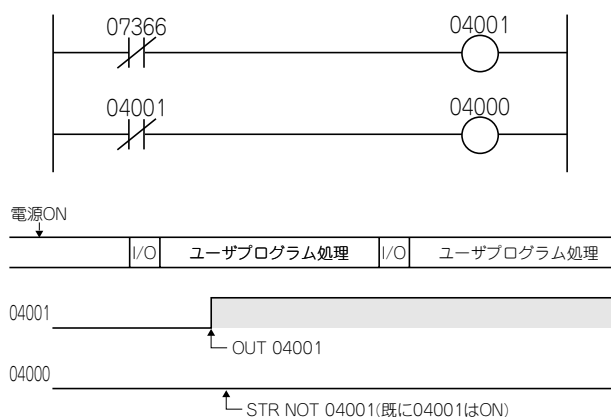
(例2) 入力信号がOFF→ONの変化時、実行する命令の場合



## 1-2 電源投入時に1パルス発生させる回路



**注1** プログラム順を入換えると、パルスは発生しません。



**注2** 04001がキープ機能指定領域のとき、パルスは発生しません。

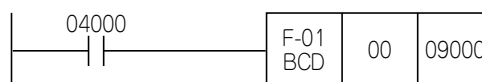
(理由) 前回の電源ON時に04001はONし、停電中もONを保持。

キープ機能指定領域の設定は以下のシステムメモリの設定で行います。詳細は各PCのプログラミングマニュアルまたは、ユーザズマニュアルをご参照ください。

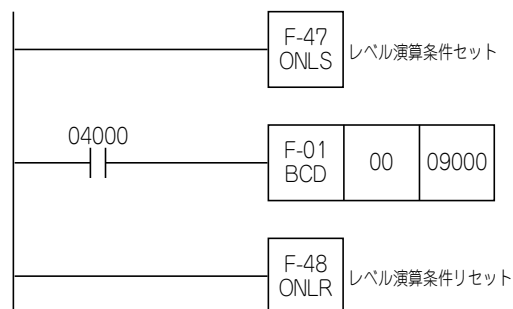
PC機種名	システムメモリ
JW10	#230, #231
JW20H	#230, #231
JW30H	#230, #231, #250, #251
JW50H/70H/100H	#230, #231

**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ときに用います。  
(例) 電源投入時、レジスタ09000の内容をクリア

・ JW50H/70H/100Hの場合



・ JW10、JW20H、JW30Hの場合



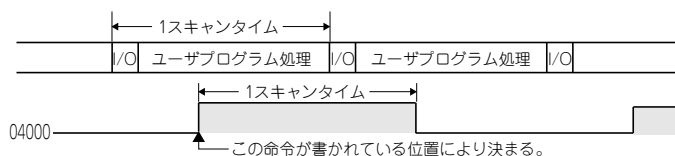
上記のように、JW10、JW20H、JW30Hの場合、入力信号の立上りで実行する命令(F-01等)の入力信号として本パルスを使用するときは、レベル演算条件内で使用する必要があります。

**参考** 電源投入時、1スキャンタイムONする特殊リレーとして、イニシャライズパルス(07362)があります。

# 1 - 3 発振回路

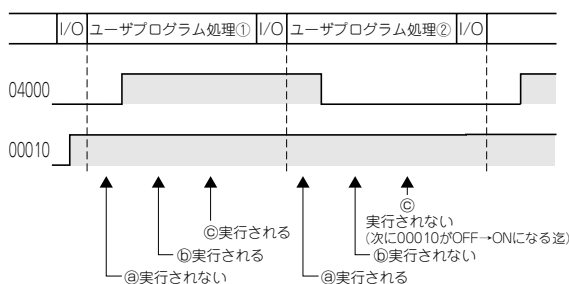
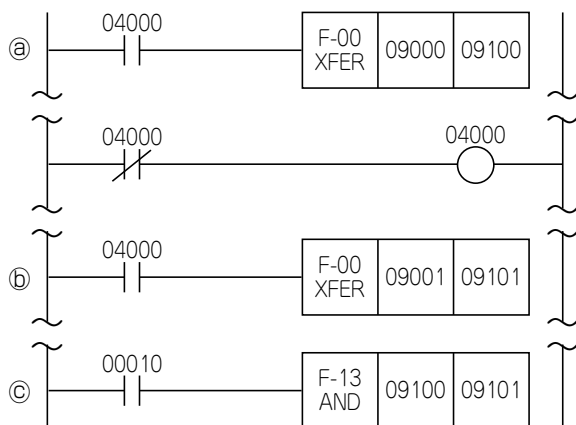


- 1スキャンごとにON/OFFを繰り返します。



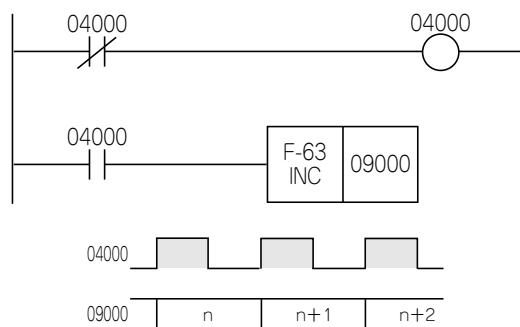
このパルスは点滅回路の基本クロックや、1スキャンおきの演算起動信号として使用します。

**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。

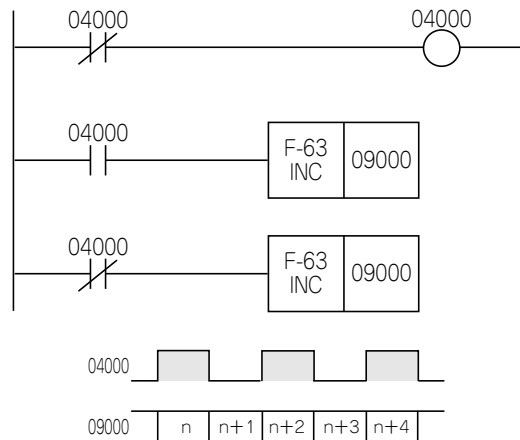


ユーザプログラム処理①のスキャンサイクルで◎の演算を実行するとき、◎は、このスキャンサイクルでは実行されないで、1つ前のスキャンサイクルで実行された◎の演算結果が◎の演算に使われてしまいます。

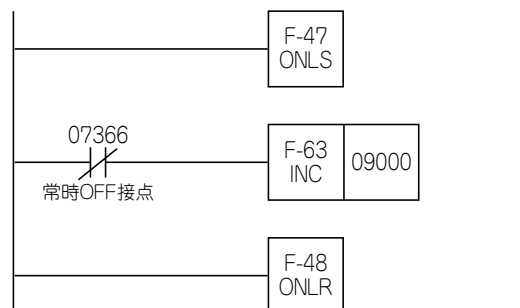
**参考** 毎スキャンサイクル演算を実行する方法  
発振回路のクロックをデータ処理命令の実行条件としてプログラムすると、1スキャンおきにしか演算が実行されません。



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。

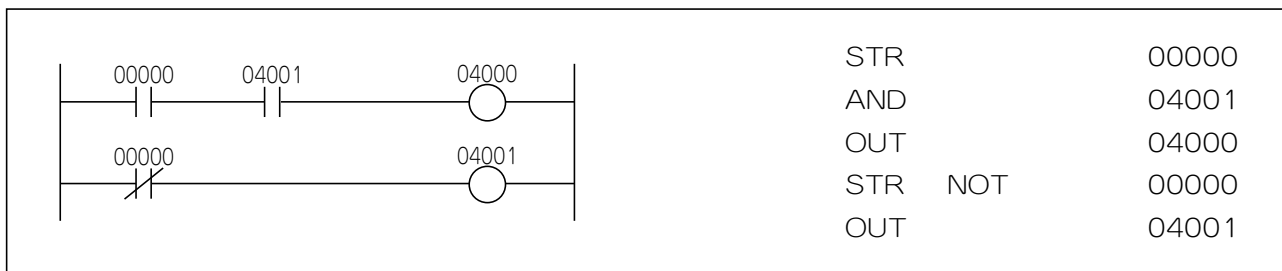


F-47(レベル演算条件セット)、F-48(レベル演算条件リセット)を使用すると毎演算サイクル演算を実行できます。



「2-1 演算実行条件の変更」をご参照ください。

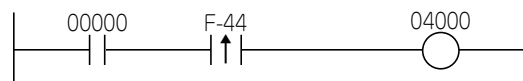
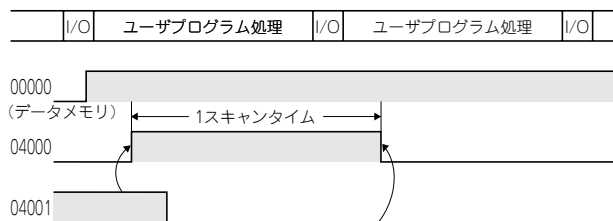
## 1 - 4 立上り微分



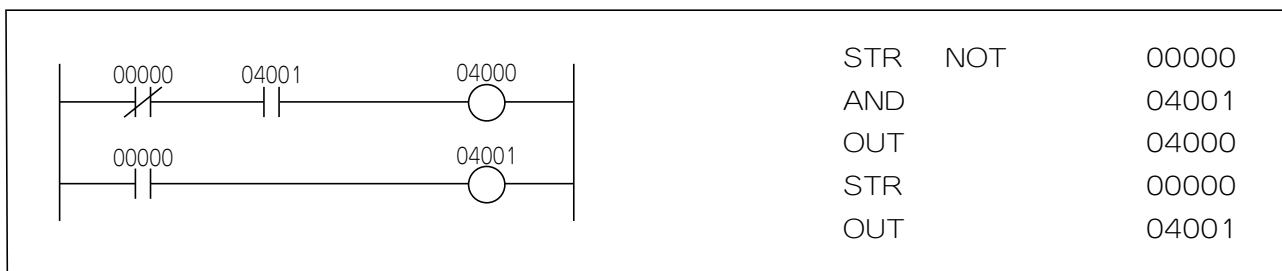
● 00000がOFF→ONとなる時、1スキャンタイムの間 04000がONとなります。

**注1** プログラム順を入換えるとパルスが発生しません。

**参考** F-44を使用すると1命令で実現できます。



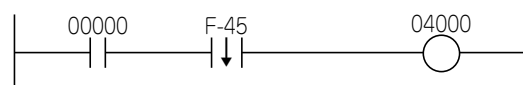
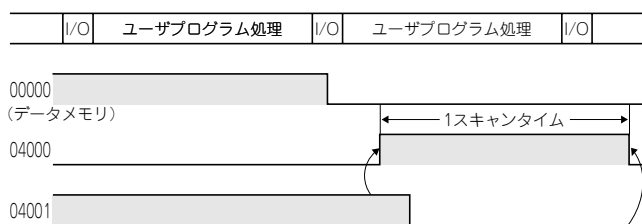
## 1 - 5 立下り微分



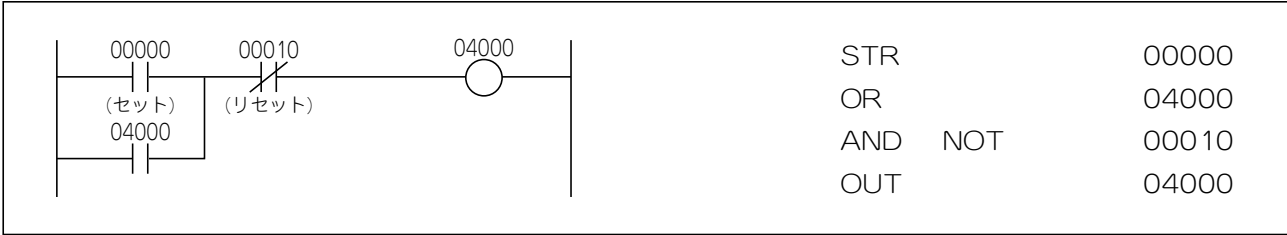
● 00000がON→OFFとなる時、1スキャンタイムの間 04000がONとなります。

**注1** プログラム順を入換えるとパルスが発生しません。

**参考** F-45を使用すると1命令で実現できます。

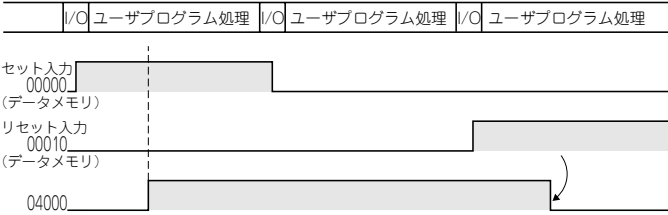


# 1 - 6 自己保持回路（リセット優先）

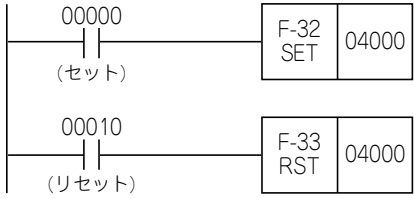


● リセット入力がOFF(ラダー図で導通)のとき、セット入力を一旦ONにすると、出力はONとなり、セット入力がOFFになってもこの状態を保持します。リセット入力をONにするか、電源を切らない限りONが継続します。

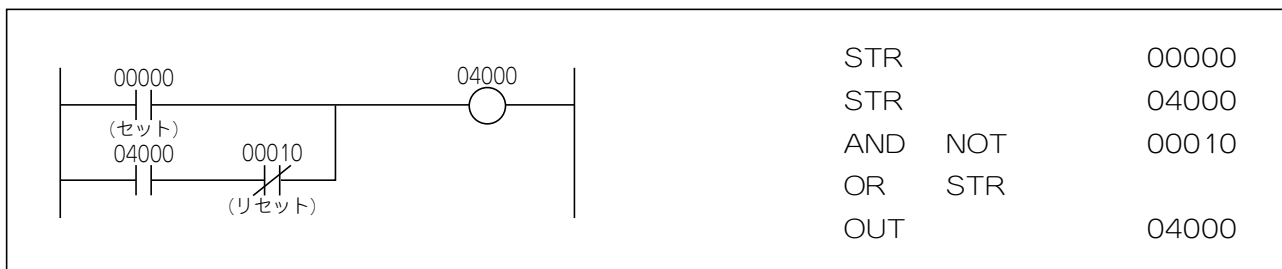
**注1** 出力として、キープ機能指定領域のデータメモリを使うと、停電があっても、停電直前の状態を保持できます。ただしリセット入力の外部接点はa接点を使用しプログラム上でAND NOTとしてください。外部接点にb接点を使用し、プログラム上でANDを使用すると、入力用電源がPCの電源より先に落ちると、自己保持がリセットされてしまいます。



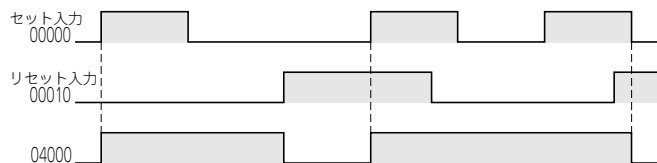
**参考** F-32(セットコイル)、F-33(リセットコイル)を使用しても自己保持回路を構成できます。



## 1-7 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** 出力としてキープ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。

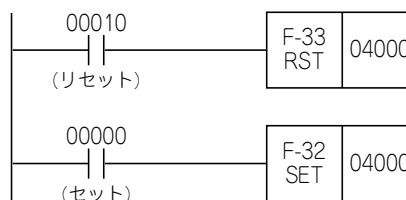
ただしリセット入力の外部接点はa接点を使用しプログラム上でAND NOTとしてください。

- 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PC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。

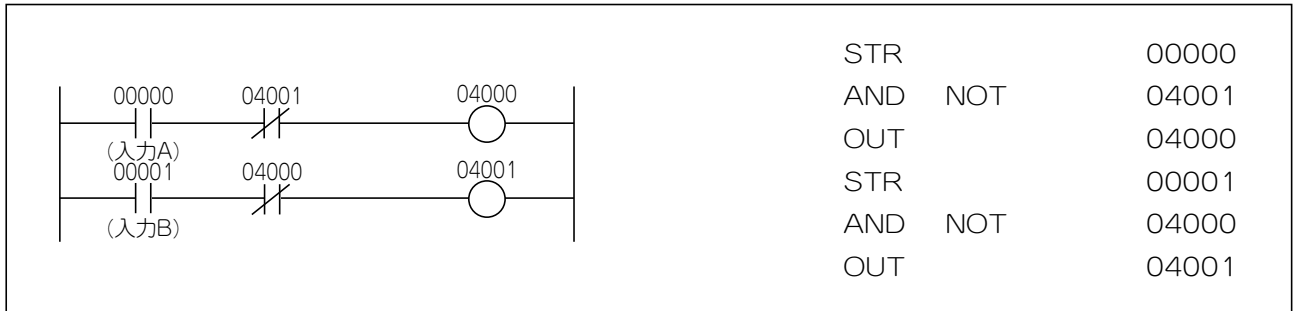
②セット入力がOFFで復電したとき、入力用電源がPCの電源より遅れて立上ると、リセット用外部接点が閉であってもリセットされます。

**参考** F-32(セットコイル)、F-33(リセットコイル)を使用しても自己保持回路を構成できます。



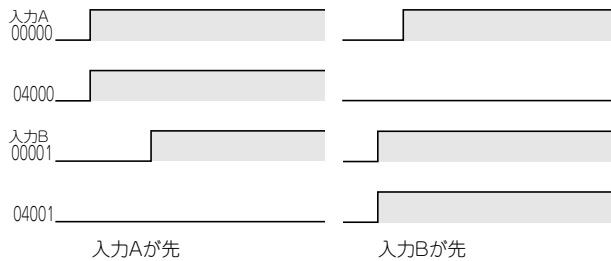
## 1 - 8 優先回路

(a) 入力が連続信号の場合

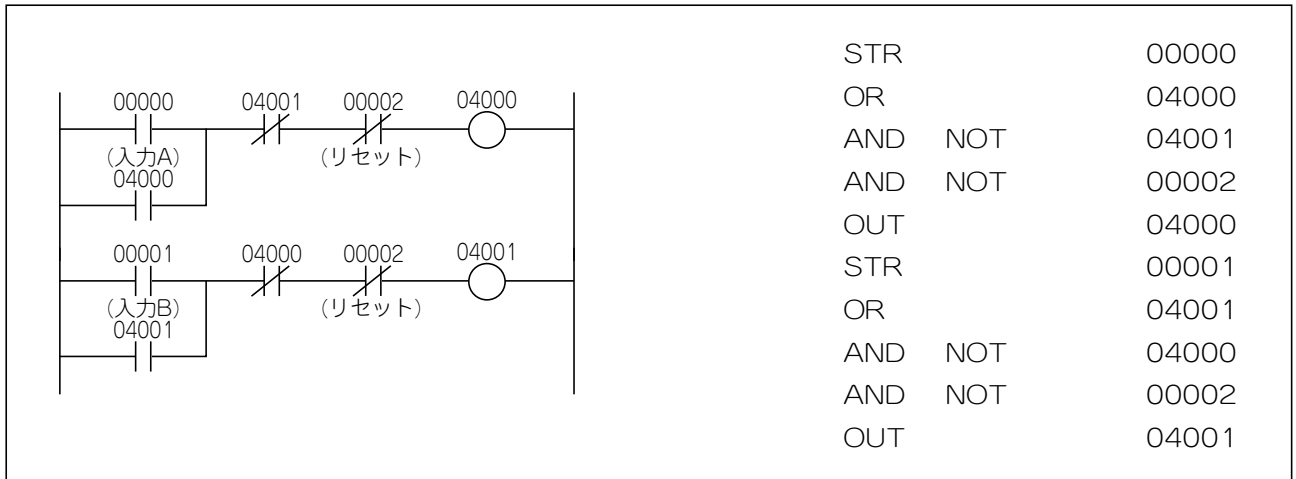


- 入力Aが入力Bのいずれか先に入った方を優先し、後で入った方の入力を無効にします。

**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。

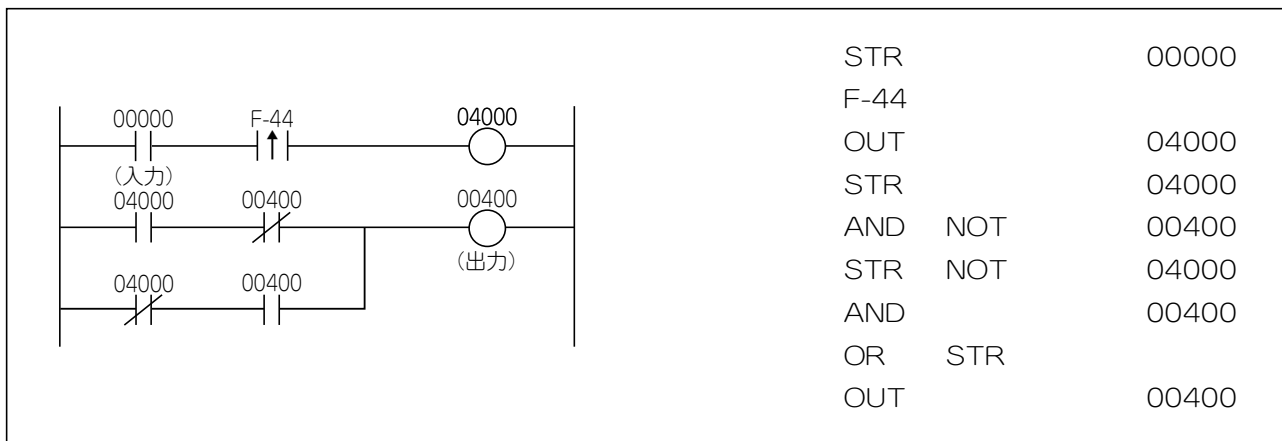


(b) 入力がパルス信号の場合

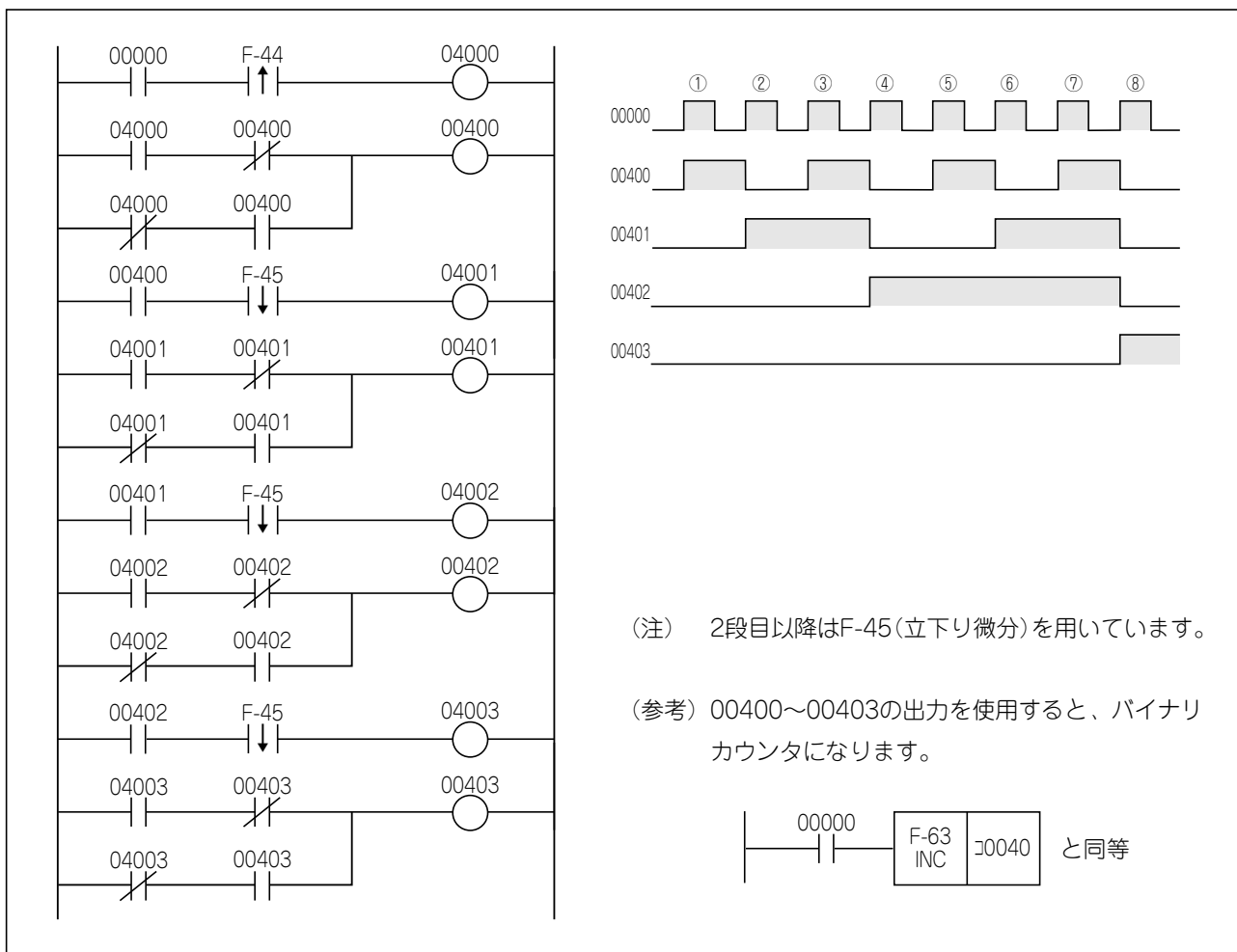
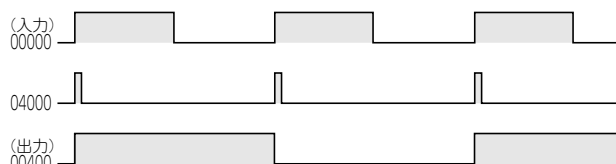


- この信号は、同時にONしては困る出力(モータの正転/逆転等)に使用します。

# 1-9 オルタネート回路

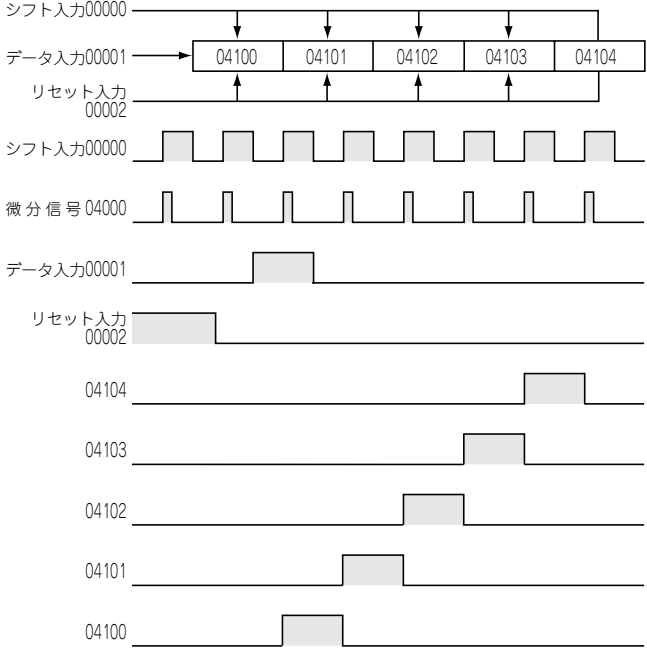
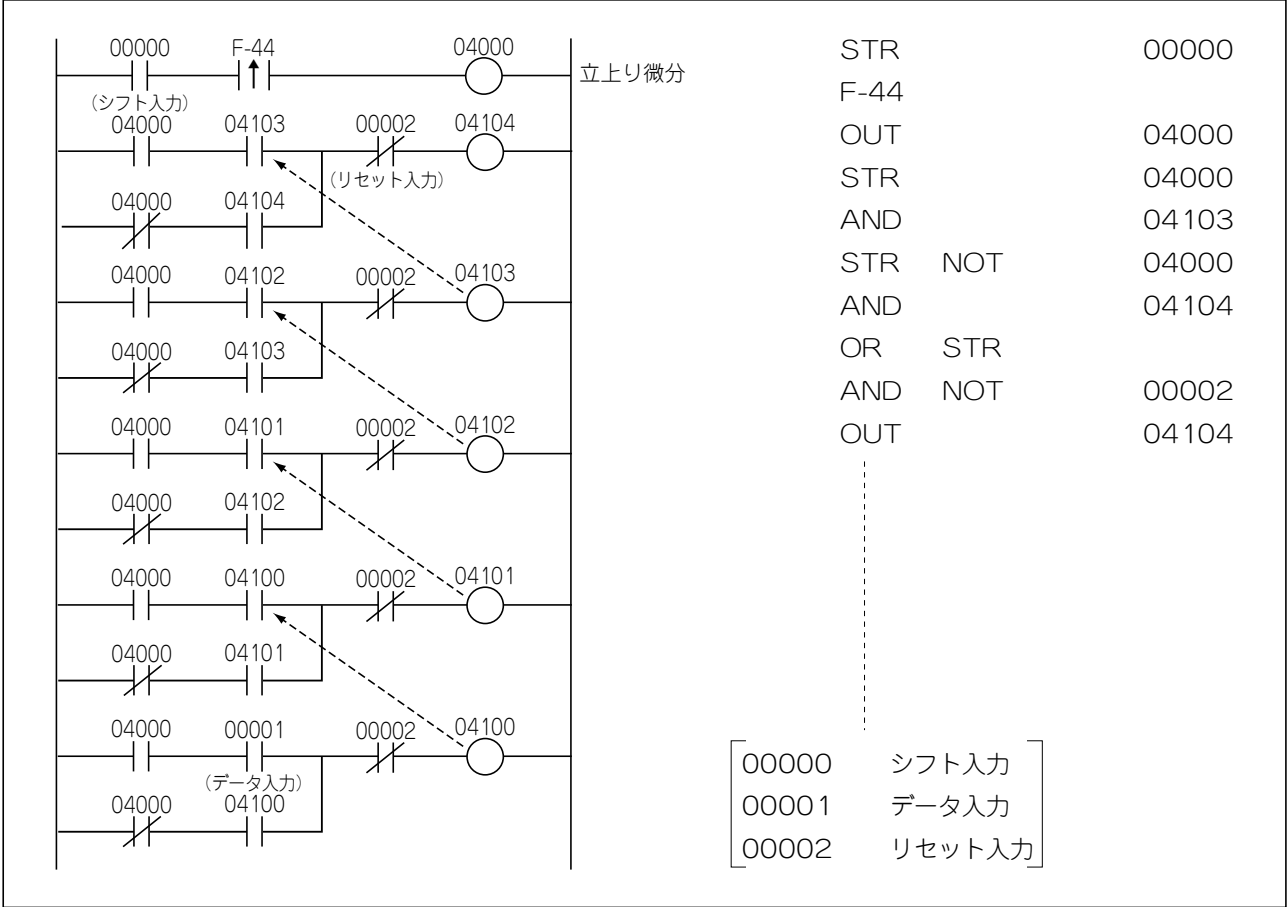


- 入力がONとなるごとに出力が反転します。モーメンタリスイッチの接点を受けてオルタネート出力を取ることができます。
- この回路を連続してn回使用するとn段の分周回路が構成できます。



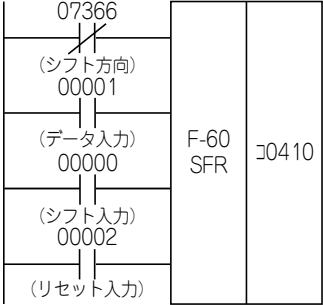


# 1 - 10 nビットシフトレジスタ

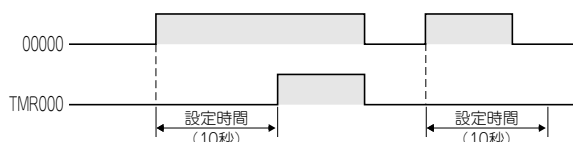
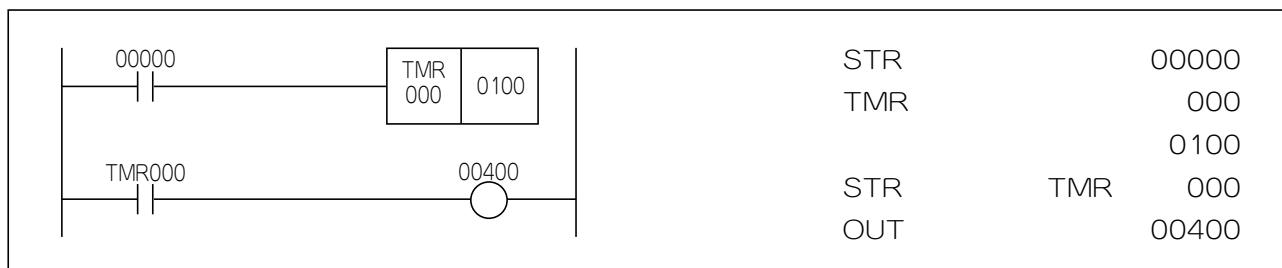


**注1** 停電時、シフト状態を保持する場合、04100～04104はキープ機能指定領域のデータメモリを使用する必要があります。

**参考** 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。



## 1-11 オンディレイタイマ



● 入力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。

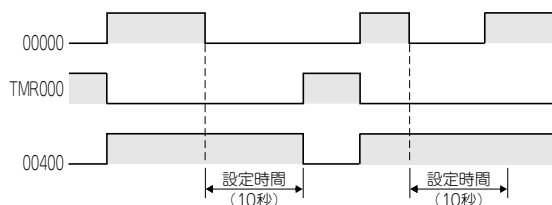
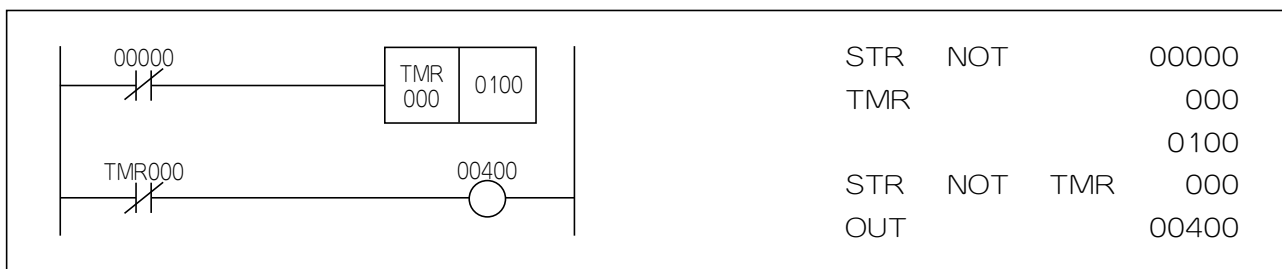
**注1** システムメモリ#201の設定により、停電時に現在値をリセットするか保持するかの選択ができます。

#201の設定	{	000 <sup>(8)</sup> 現在値をリセット (設定値になる)
		001 <sup>(8)</sup> 現在値を保持

**注2** 入力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>(8)</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>(8)</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。

## 1-12 オフディレイタイマ



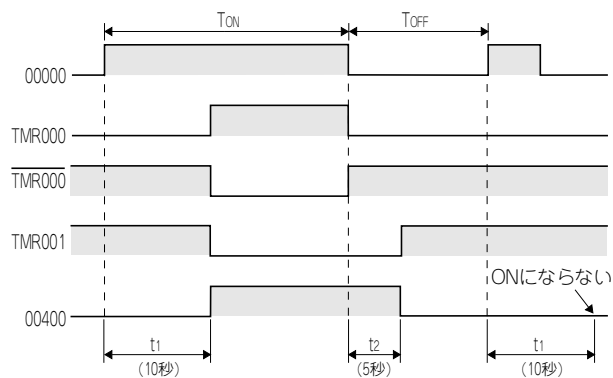
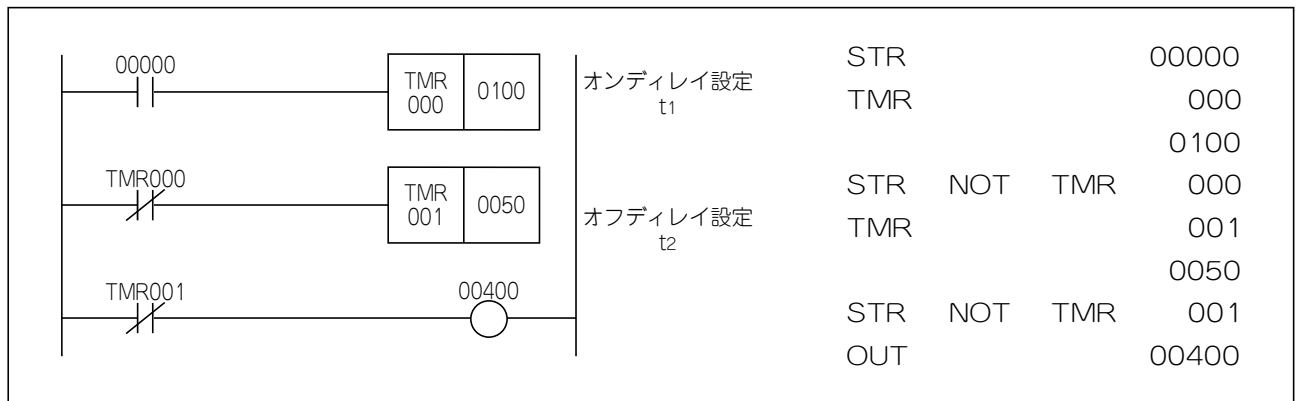
● 入力OFF後、設定時間だけ遅れて出力がOFFとなります。もし入力OFFの時間が設定時間以下のとき出力はOFFになりません。

● 入力ONになれば、出力もONになります。

**注1** 入力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>(8)</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>(8)</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。

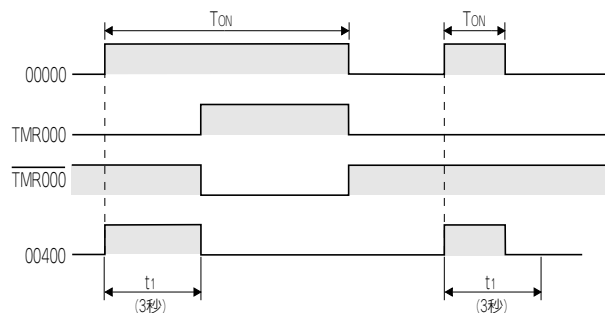
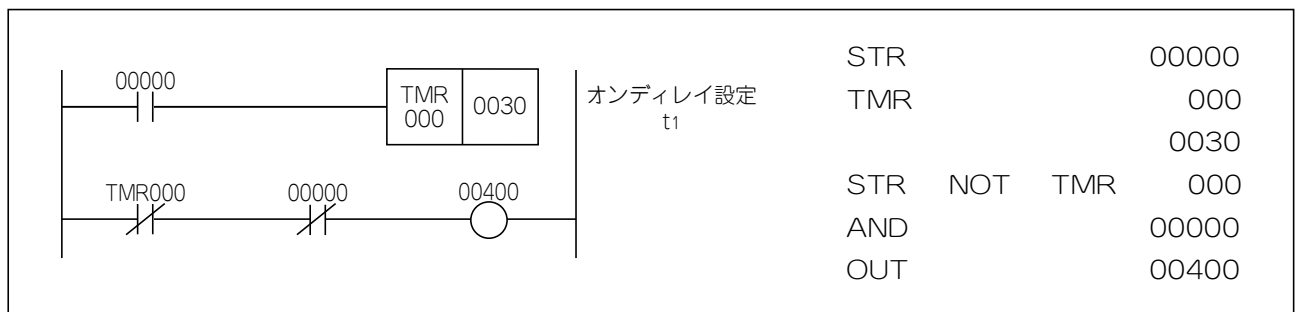
## 1 - 13 オン・オフディレイタイマ



- 入力が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。

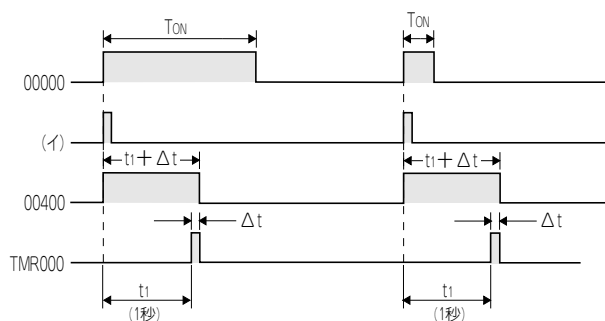
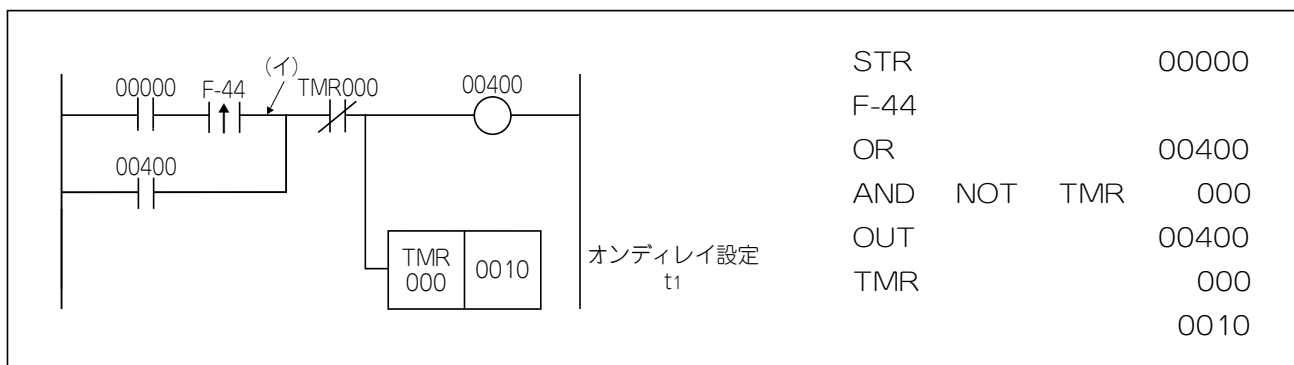
**注1** 停電時の出力の状態については「1-11 オンディレイタイマ」、「1-12 オフディレイタイマ」の注意事項をご参照ください。

## 1 - 14 入力立上り時ワンショットタイマ (1)



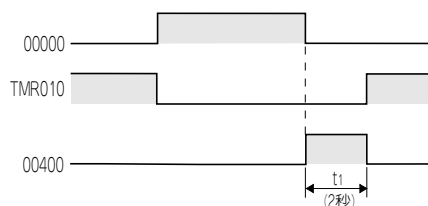
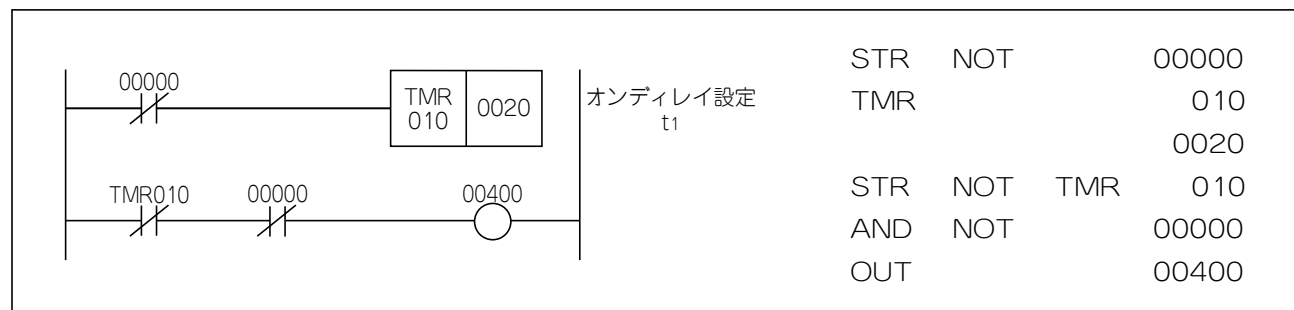
- 入力の立上り(OFF→ON)時に設定時間 $t_1$ の中のパルスが出力されます。
- 入力のONしている時間( $T_{ON}$ ) < 設定時間( $t_1$ )のとき、出力パルスの中は $T_{ON}$ となります。

## 1 - 15 入力立上り時ワンショットタイマ (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。

## 1 - 16 入力立下り時ワンショットタイマ



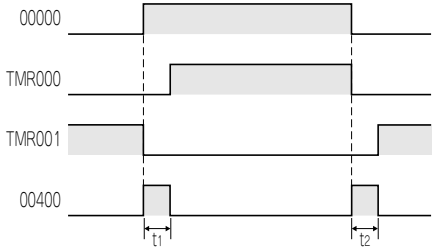
- 入力の立下り(ON→OFF)時に設定時間( $t_1$ )の中のパルスが出力されます。

**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(8) - 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えます。
- (b) #201が001(8) - 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。

# 1 - 17 立上り、立下りワンショットタイマ

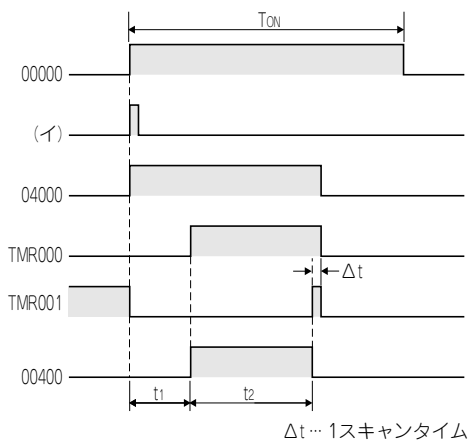
STR		00000
TMR		000
		0010
STR	NOT	00000
TMR		001
		0010
STR	NOT	TMR
AND		00000
STR	NOT	TMR
AND	NOT	00000
OR	STR	
OUT		00400



- 入力の立上り、立下りにそれぞれt1、t2のパルス巾だけ出力がONします。
- 入りに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。

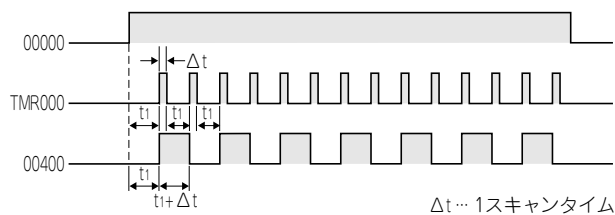
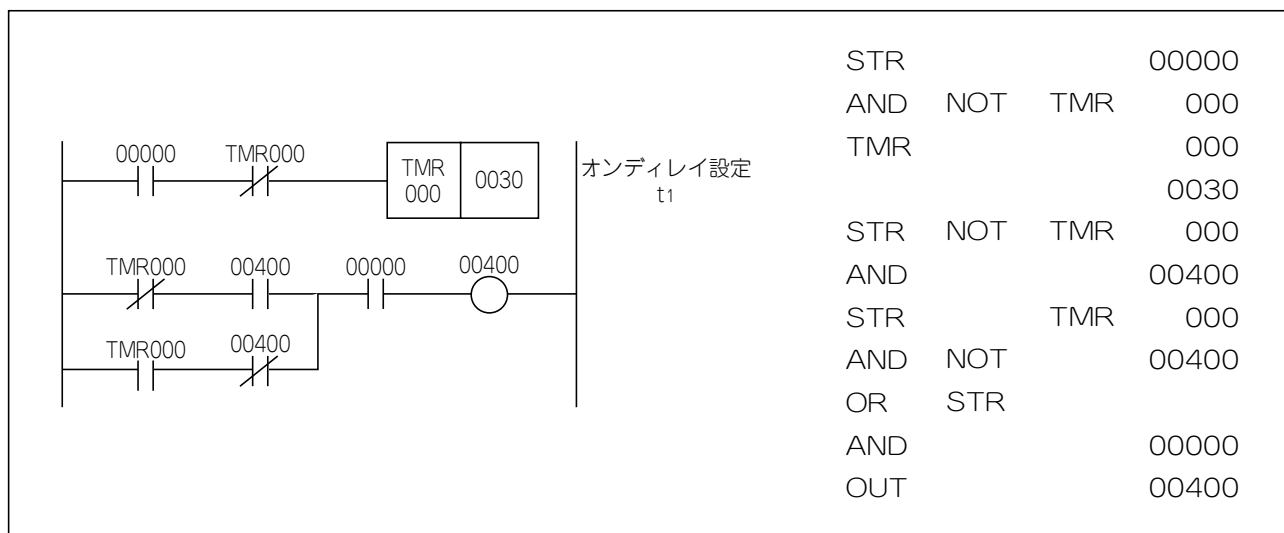
# 1 - 18 オンディレイワンショットタイマ

STR		00000
F-44		
OR		04000
AND	NOT	TMR
OUT		04000
TMR		000
		0010
STR		TMR
TMR		001
		0020
STR		TMR
AND	NOT	TMR
OUT		00400



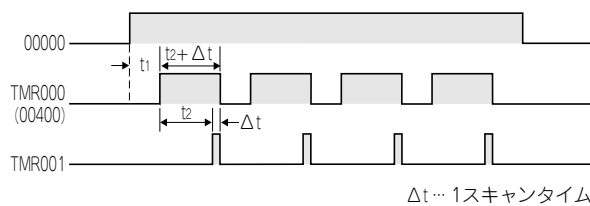
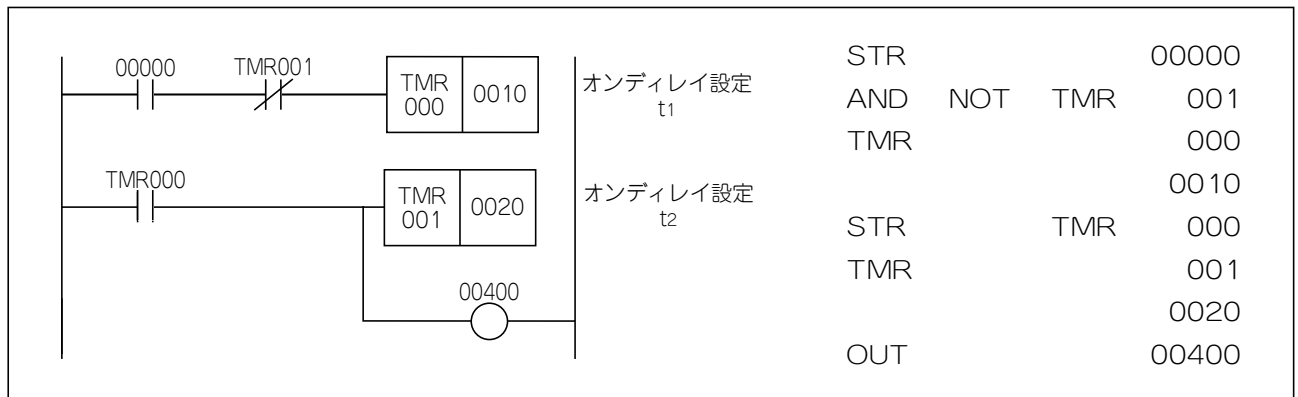
- 入力の立上りから設定時間 $t_1$ だけ遅れて、パルス巾 $t_2$ のパルスが出力されます。
- ( $t_1+t_2$ )の時間中に入力がON/OFFしても無視されます。

## 1 - 19 等間パルス発生回路



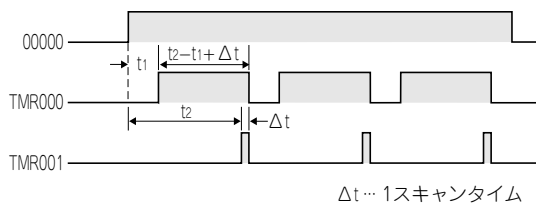
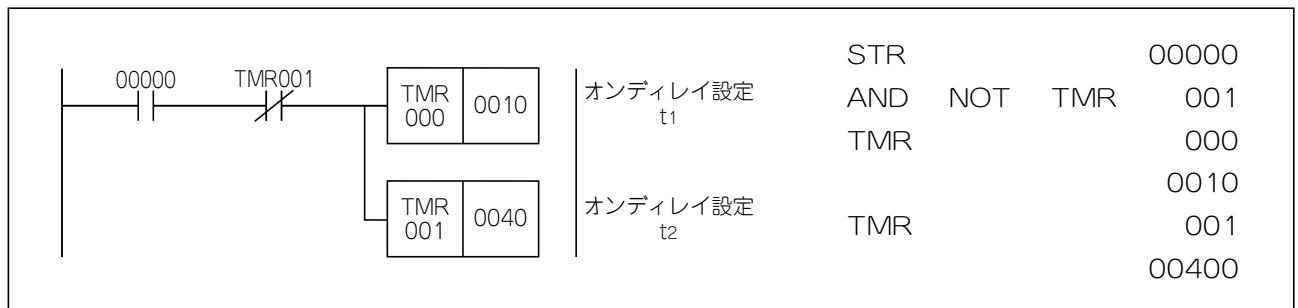
- 入力がONの間、ON/OFFの時間が等しい(デューティサイクル50%)パルスが出力されます。ON/OFFの時間はTMRの設定値( $t_1$ )で任意に設定することができます。パルス巾は $t_1 + \Delta t$ となります。

## 1 - 20 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。

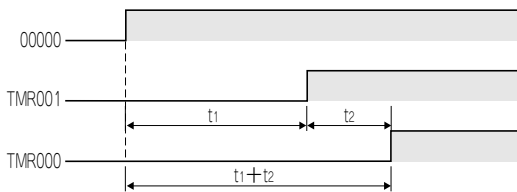
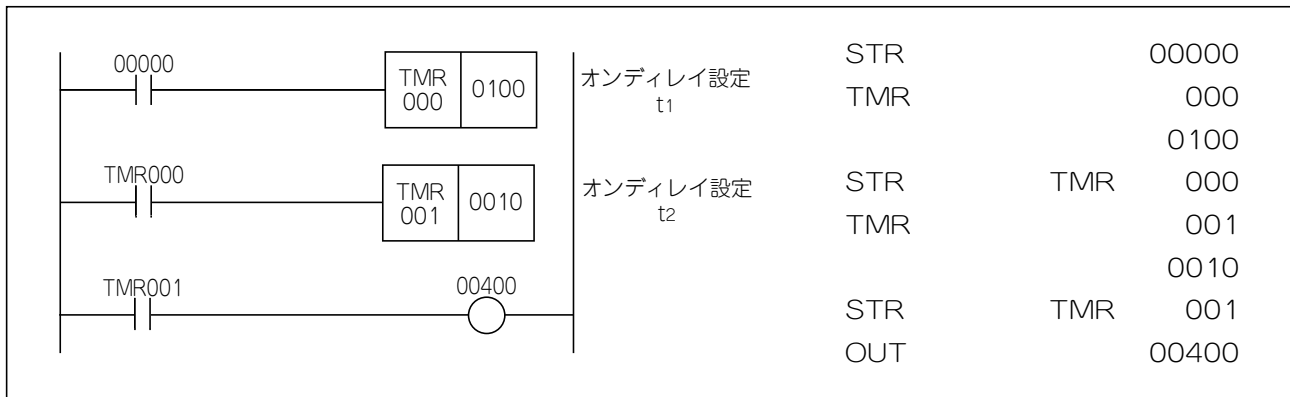
## 1 - 21 デューティ可変パルス発生回路 (2)



- 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。

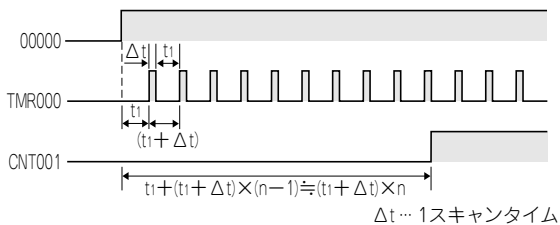
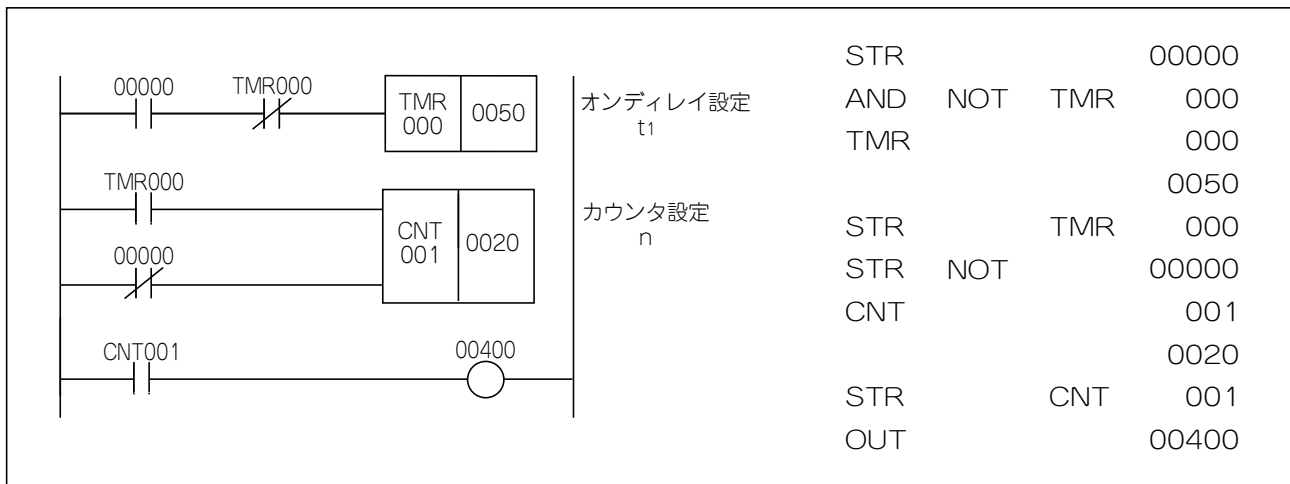
**注1** 必ず $t_1 < t_2$ と設定してください。

## 1 - 22 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。

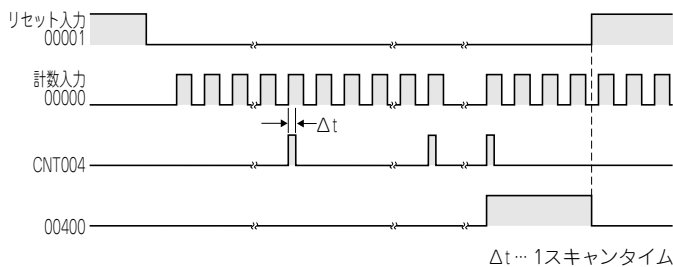
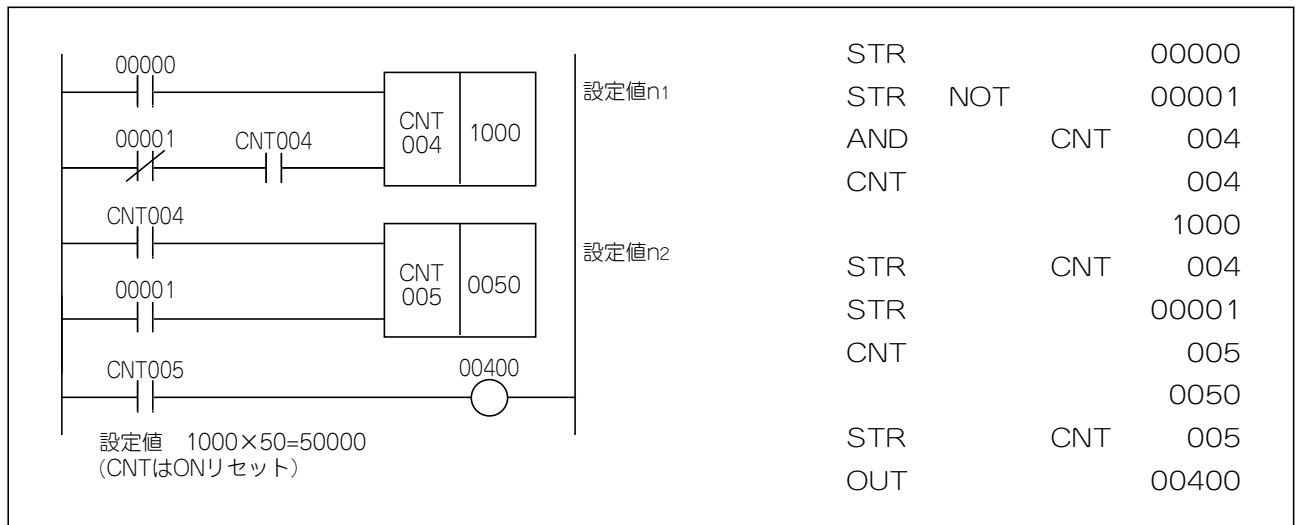
## 1 - 23 長時間タイマ (2)



● 入力ON後、 $(t_1+\Delta t) \times n$ 遅れて出力がONになります。

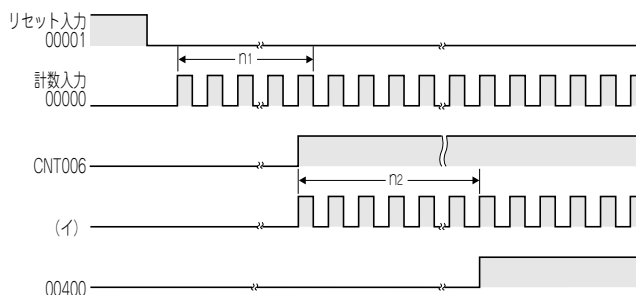
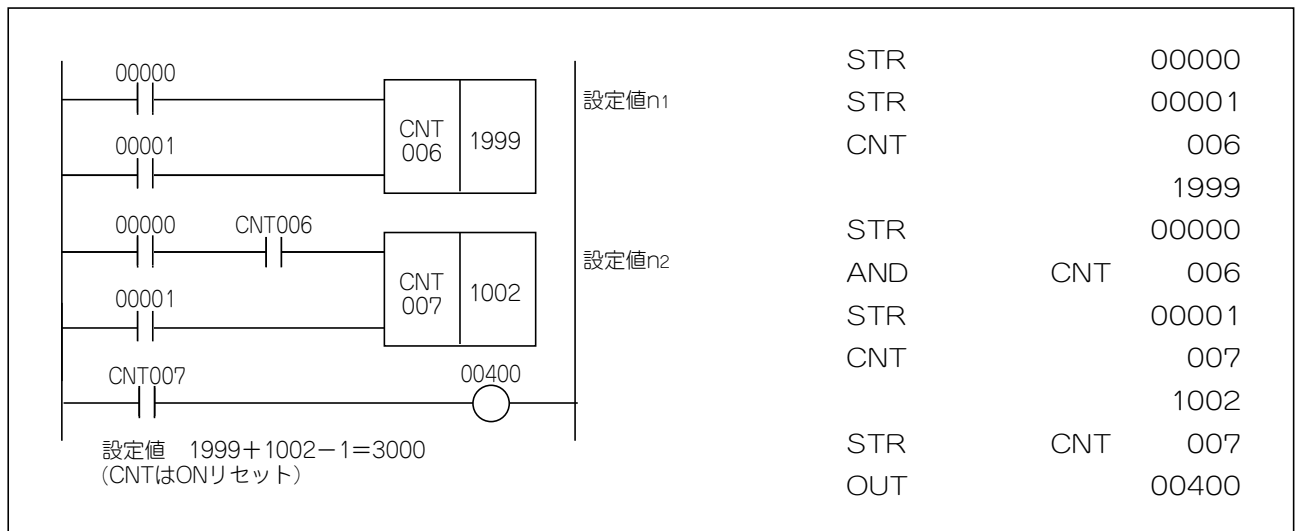


## 1 - 24 大容量カウンタ (1)



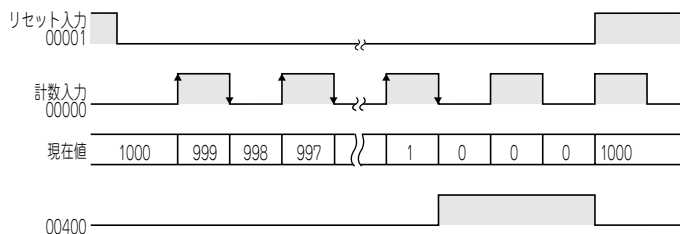
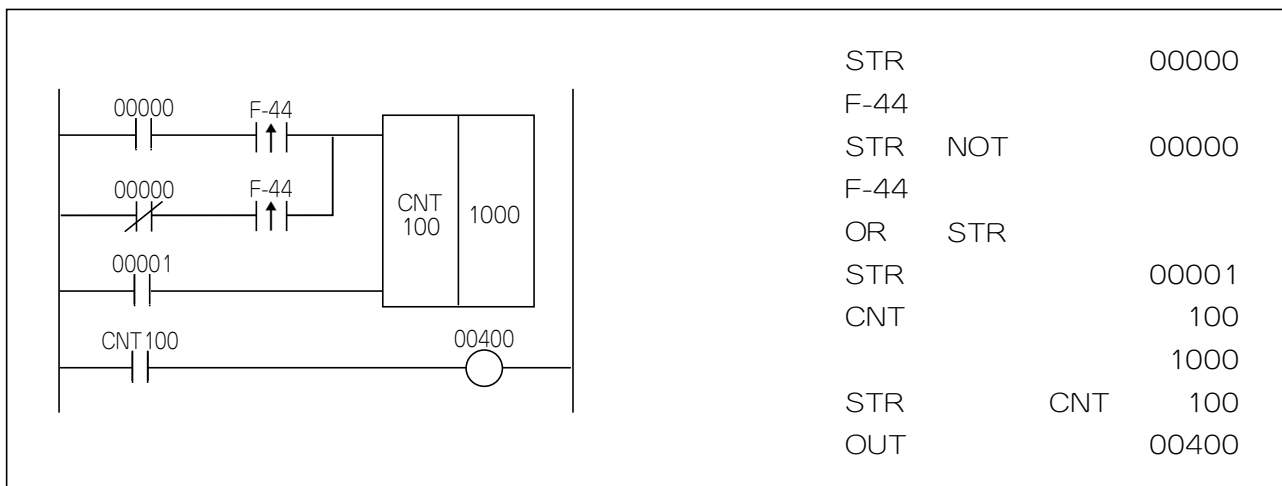
● 設定値が1999を越える場合、上図のようにプログラムすると、設定値( $n_1 \times n_2$ )のカウンタを実現できません。

## 1 - 25 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。

# 1 - 26 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。

# 第2章 データ処理回路

この章では、  
JWシリーズPCの豊富な応用命令を使用した一般的なデータ処理回路を中心に紹介します。

2-1	演算実行条件の変更	2-1
2-2	レジスタのクリア	2-2
2-3	データのマスク	2-2
2-4	ビットのセット/リセット	2-3
2-5	数の分解	2-4
2-6	数の合成	2-5
2-7	設定値との比較	2-6
2-8	ウィンドウコンパレータ	2-8
2-9	不感帯をもつ比較回路	2-9
2-10	複数のセットポイントを持つタイマ	2-10
2-11	BCD減算結果を符号付絶対値で求める	2-11
2-12	BCD 8桁の乗算	2-12
2-13	BCD 8桁÷BCD 2桁	2-14
2-14	BCD 4桁の除算	2-15
2-15	ドラムシーケンサ	2-19
2-16	タイマ現在値の外部出力	2-20
2-17	カウンタ現在値の外部出力	2-23
2-18	タイマの設定値を外部機器から入力	2-25
2-19	カウンタの設定値を外部機器から入力	2-27
2-20	複数のタイマ、カウンタの設定値を外部機器から入力	2-30
2-21	時・分・秒設定減算タイマ	2-34
2-22	ダイナミック入力	2-35
2-23	ダイナミック出力	2-37
2-24	同期型FIFOスタックレジスタ	2-39
2-25	データの分配	2-40
2-26	データの抽出	2-43
2-27	データの挿入	2-46
2-28	データの削除	2-48
2-29	データの検索(1)	2-50
2-30	データの検索(2)	2-51
2-31	データの照合	2-53

2-32	B C D 4桁の最小値・最大値を求める	2・55
2-33	B C D 2桁の数値の平均値を求める	2・57
2-34	ファイルレジスタの領域指定クリア	2・59
2-35	テンキーからの数値の読込	2・60
2-36	8→256デコーダ	2・62
2-37	256→8エンコーダ	2・63
2-38	7S E Gエンコーダ	2・65
2-39	グレイコード→バイナリコード変換	2・66
2-40	B C D 6桁のアップ・ダウンカウンタ	2・67
2-41	24ビットシフトレジスタ	2・69
2-42	スキャンタイムの測定	2・71
2-43	B C Cコードの生成	2・72
2-44	高速カウンタの現在値を符号付きで表示する	2・73

## 2 - 1 演算実行条件の変更

● データ処理用応用命令には、入力条件がOFF→ONの変化時に実行する命令と、入力条件がONの間、毎スキャンサイクル実行する命令があります。

a) 入力条件がOFF→ONの変化時実行する命令

F-00 (レジスタ間の転送)

F-63 (加算カウンタ) 等

b) 入力条件がONの間、毎スキャンサイクル実行(ONレベル演算)する命令

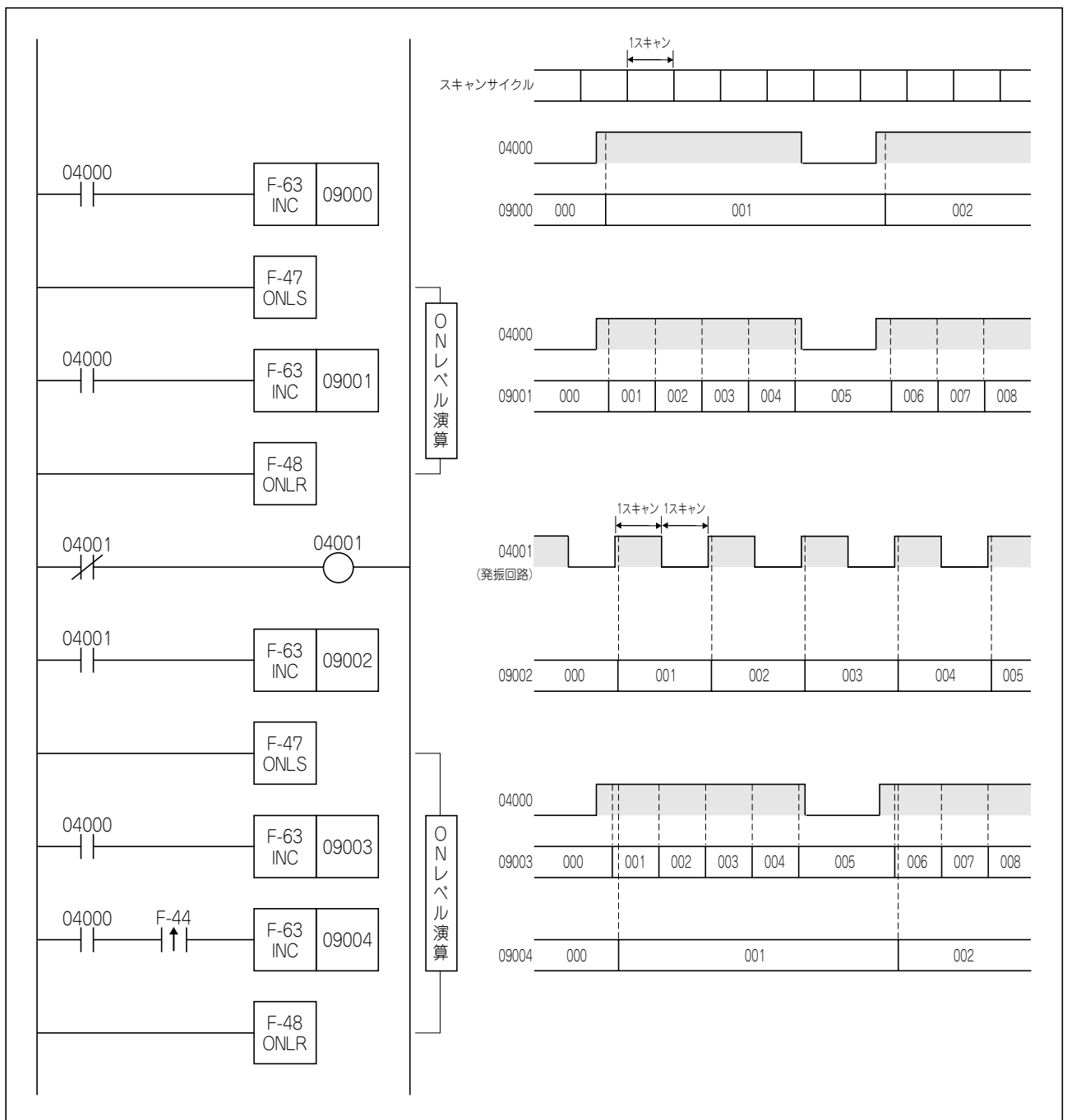
F-12 (レジスタ間の比較)

F-141(ラベルヘジャンプ) 等

● a)の命令を入力条件がONの間、毎スキャンサイクル実行させるには、当該命令をF-47(レベル演算条件セット)とF-48(レベル演算条件リセット)の間に入れます。

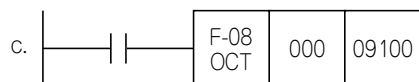
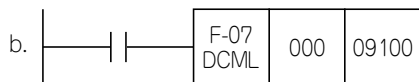
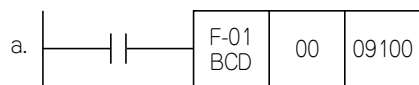
● b)の命令を入力条件がOFF→ONの変化時のみ実行させるには、F-44(ON時微分)を入力条件の後に入れます。

● F-47, F-48, F-44は、プログラム中に複数使用できますので、各命令毎に演算実行条件を変更できます。

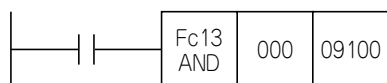


## 2-2 レジスタのクリア

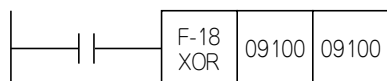
(1) 定数0を転送



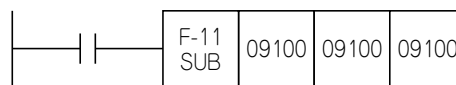
(2) 定数0との論理積



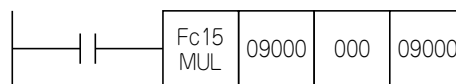
(3) 同一レジスタで排他的論理和



(4) 同一レジスタを減算

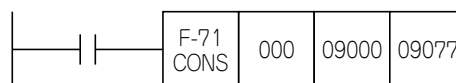


(5) 4バイトのクリア



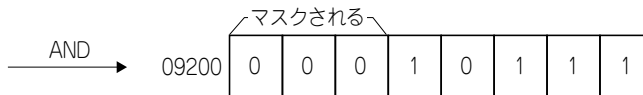
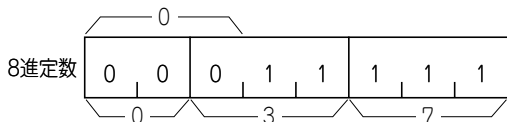
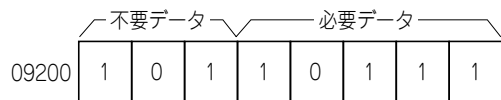
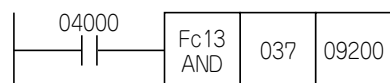
09000~09003の4バイトをクリア

(6) nバイトのクリア



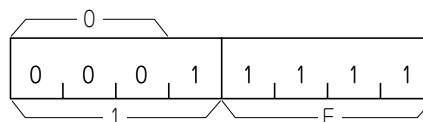
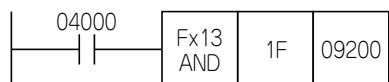
09000~09077の64バイトをクリア

## 2-3 データのマスク



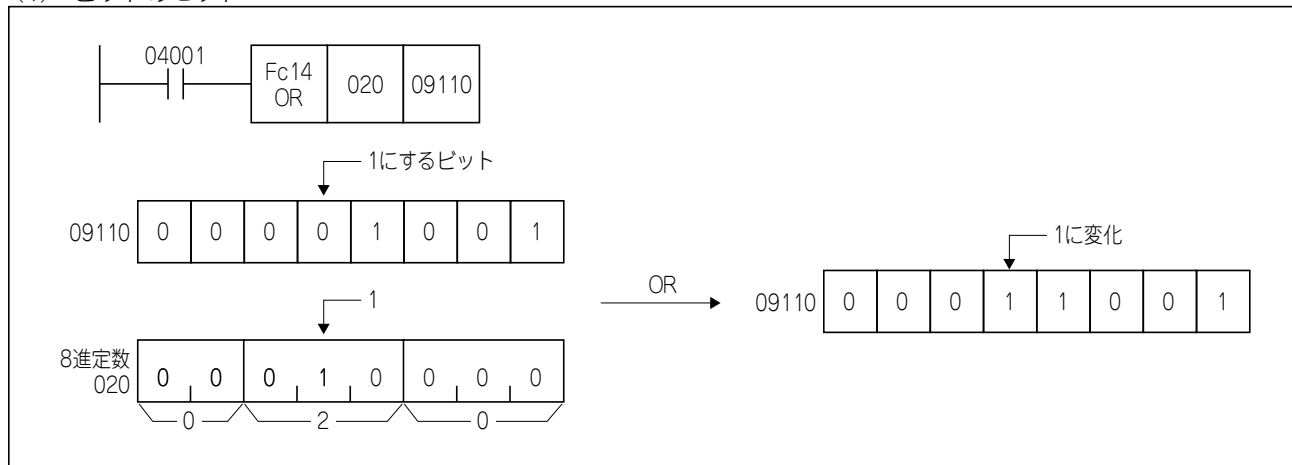
● 8ビットのデータのうちの任意のビットを0にします。

**参考** JW30H, JW50H/70H/100Hでは、16進定数によるデータのマスクも可能です。



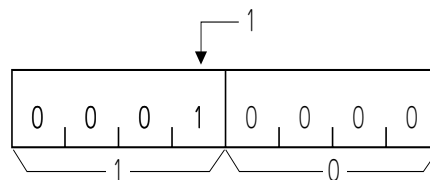
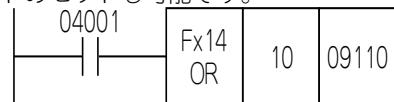
## 2-4 ビットのセット/リセット

### (1) ビットのセット

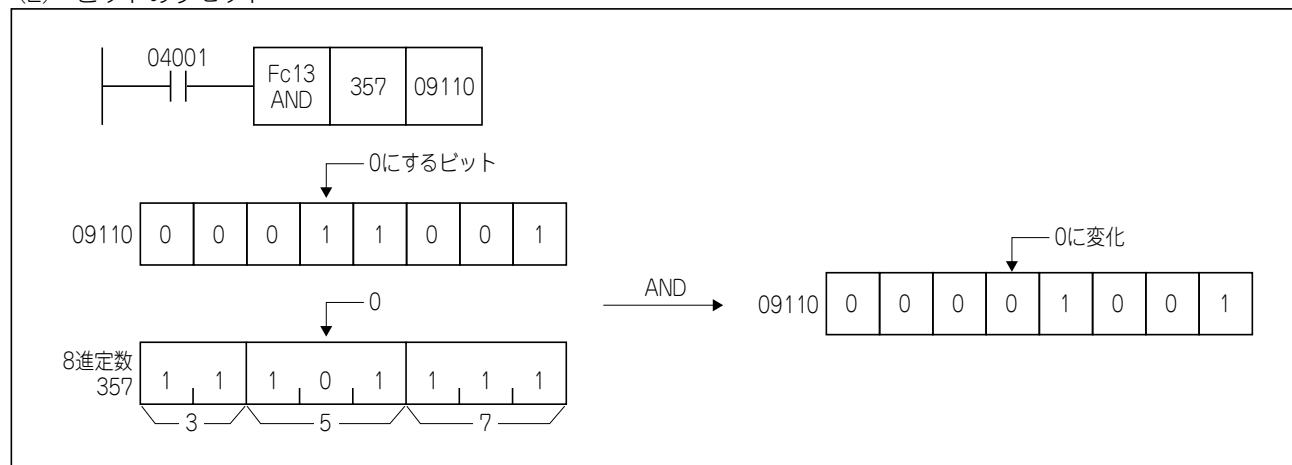


●8ビットのデータのうちの任意のビットを1にします。

【参考】 JW30H, JW50H/70H/100Hでは、16進定数によるビットのセットも可能です。

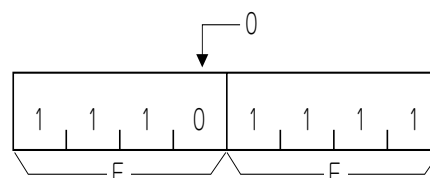
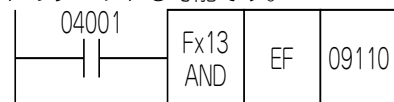


### (2) ビットのリセット

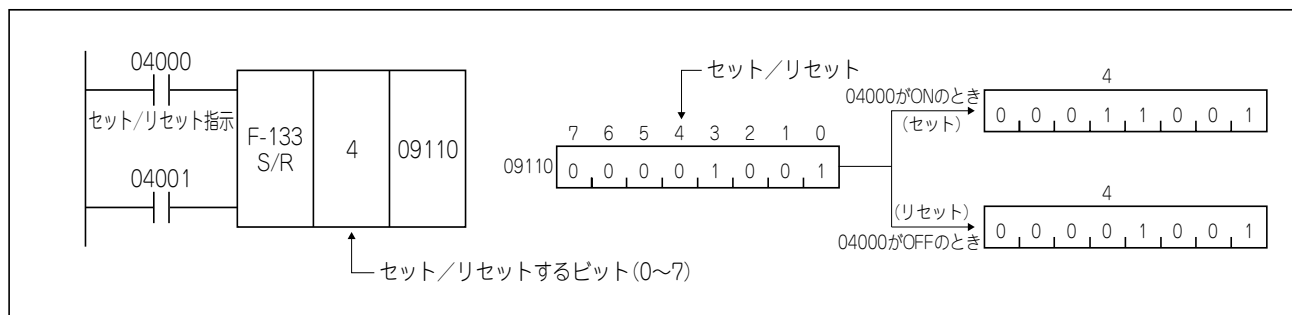


●8ビットのデータのうちの任意のビットを0にします。

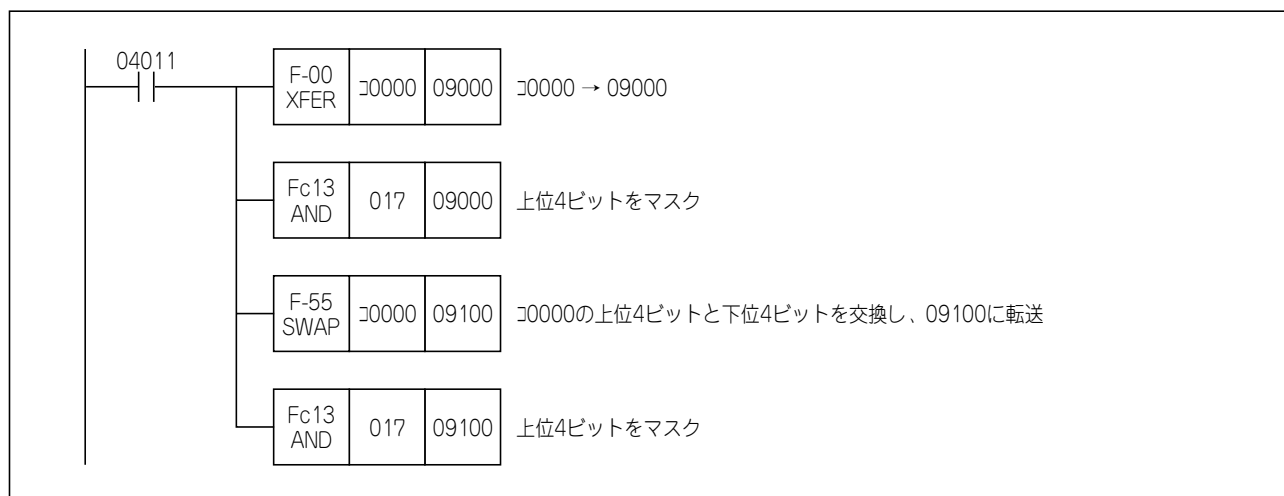
【参考】 JW30H, JW50H/70H/100Hでは、16進定数によるビットのリセットも可能です。



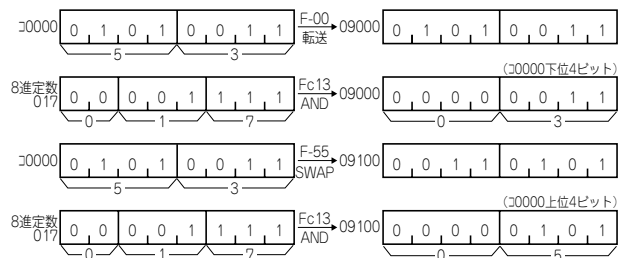
【参考】 JW20H, JW30H, JW50H/70H/100Hには、ビットセット/リセット命令(F-133)があります。



## 2 - 5 数の分解

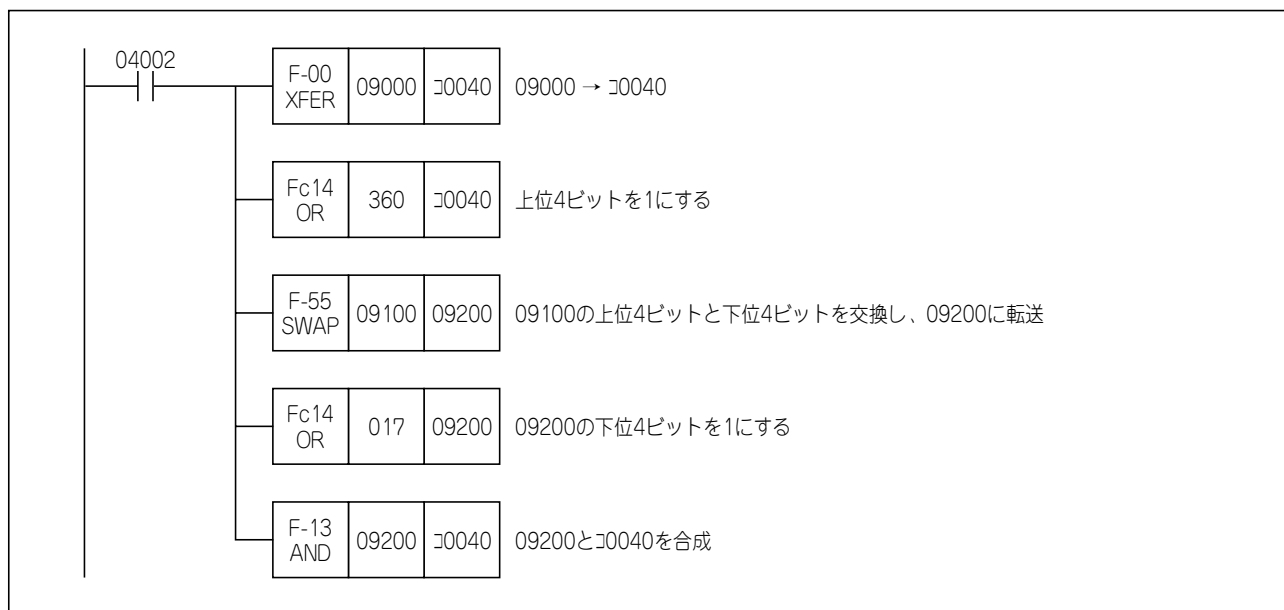


- 入力ユニットから入力されるBCD2桁の数値を1桁ずつに分解し別々のレジスタに格納します。  
例では00000の下位4ビットを09000に、上位4ビットを09100に転送しています。
- 複数のBCD1桁の数値を入力する場合、2桁分まとめて入力ユニットを使用すると、入力ユニットの節約ができます。

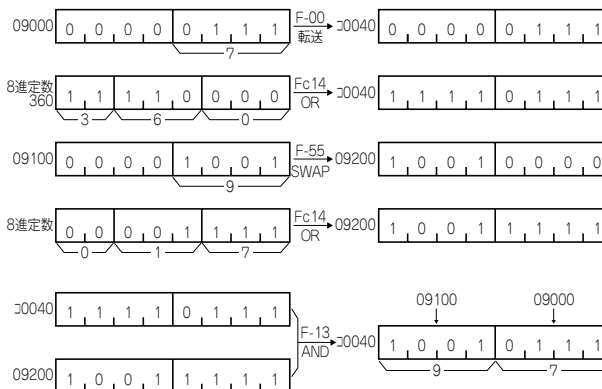




## 2-6 数の合成

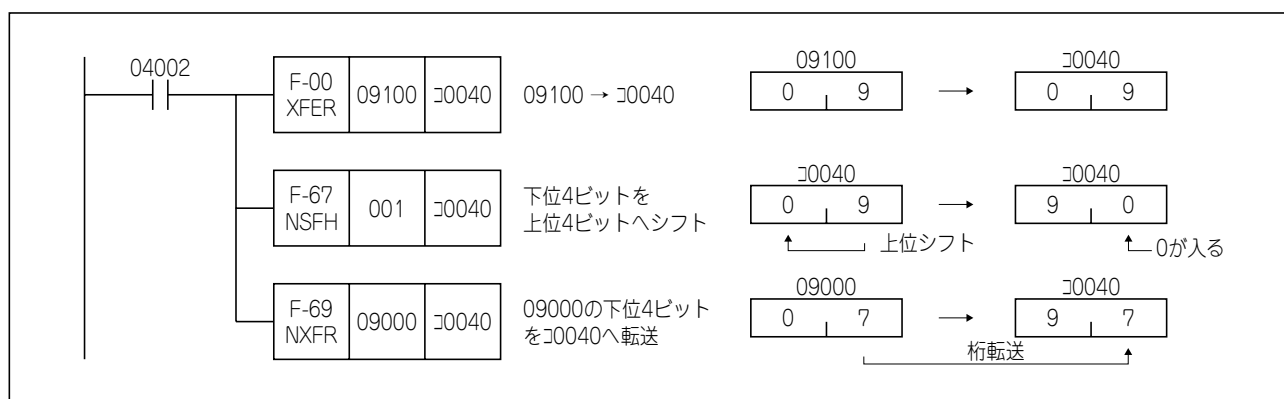


- 2つのレジスタ内のBCD1桁の数値を1つの出力ユニットにまとめて出力します。  
例では09000と09100のそれぞれの下位4ビットを0040に出力しています。

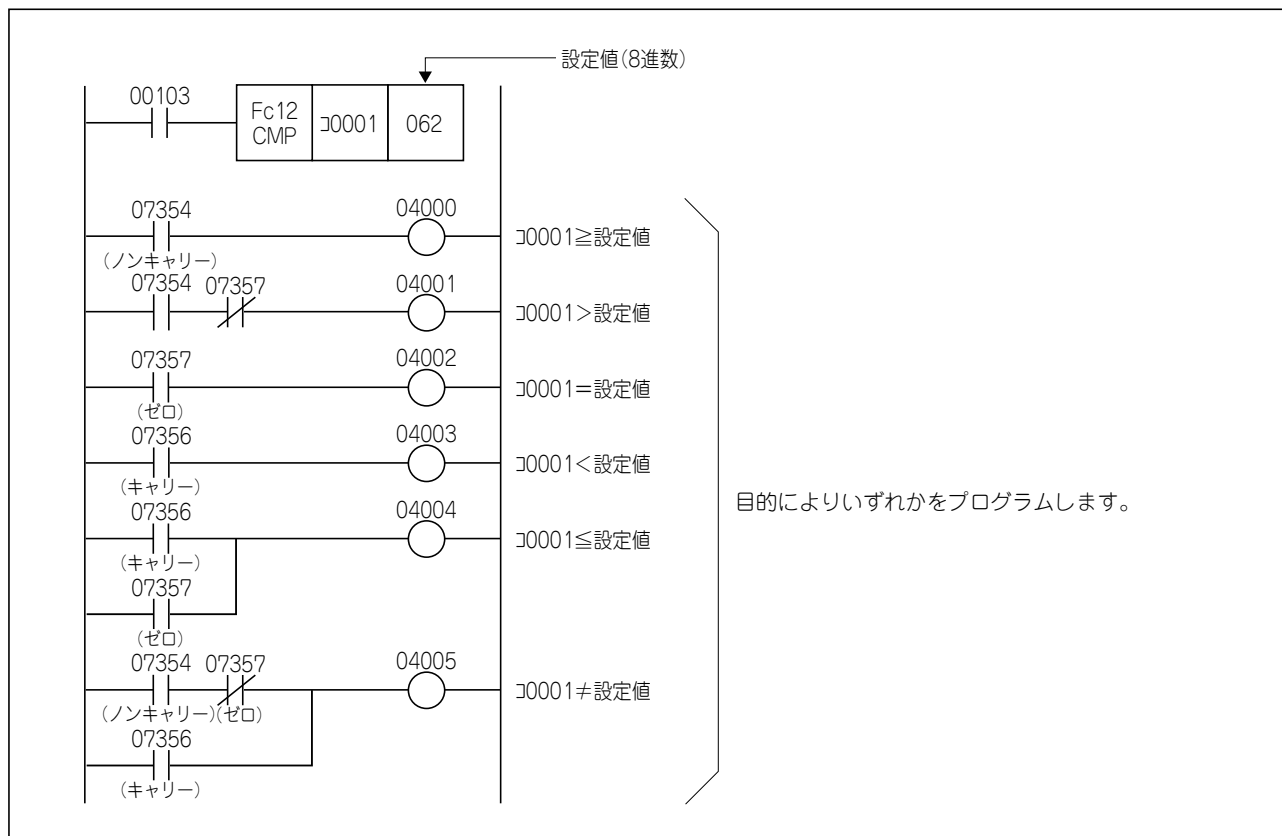


- 複数のBCD1桁の数値を出力する場合、2桁分まとめて出力ユニットを使用すると、出力ユニットの節約ができます。

**参考** JW20H, JW30H, JW50H/70H/100Hには、桁転送命令 (F-69) があります。

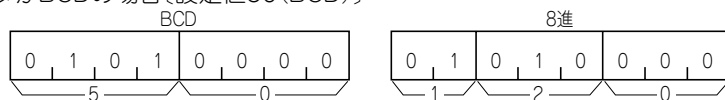


## 2-7 設定値との比較

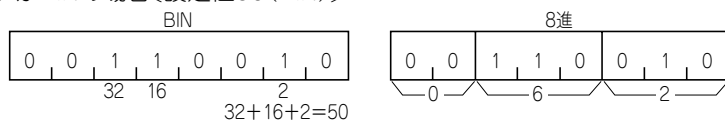


● 設定値は8進数でプログラムします。

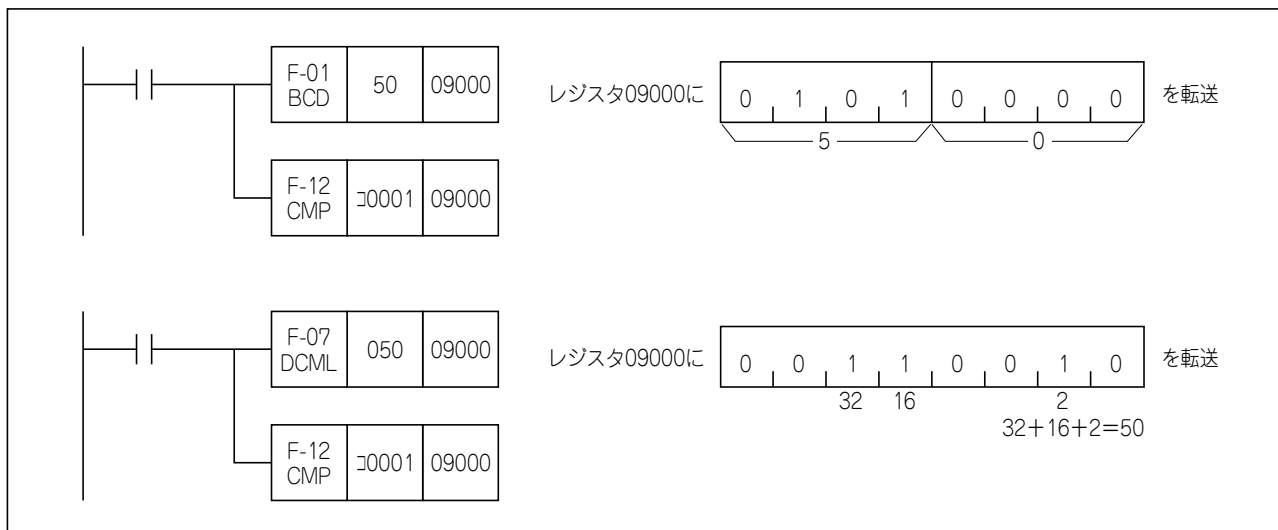
(例1) レジスタ内のデータがBCDの場合〔設定値50(BCD)〕



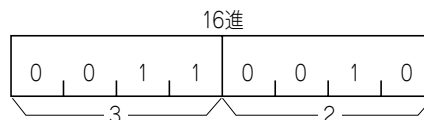
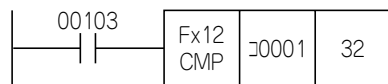
(例2) レジスタ内のデータがBINの場合〔設定値50(BIN)〕



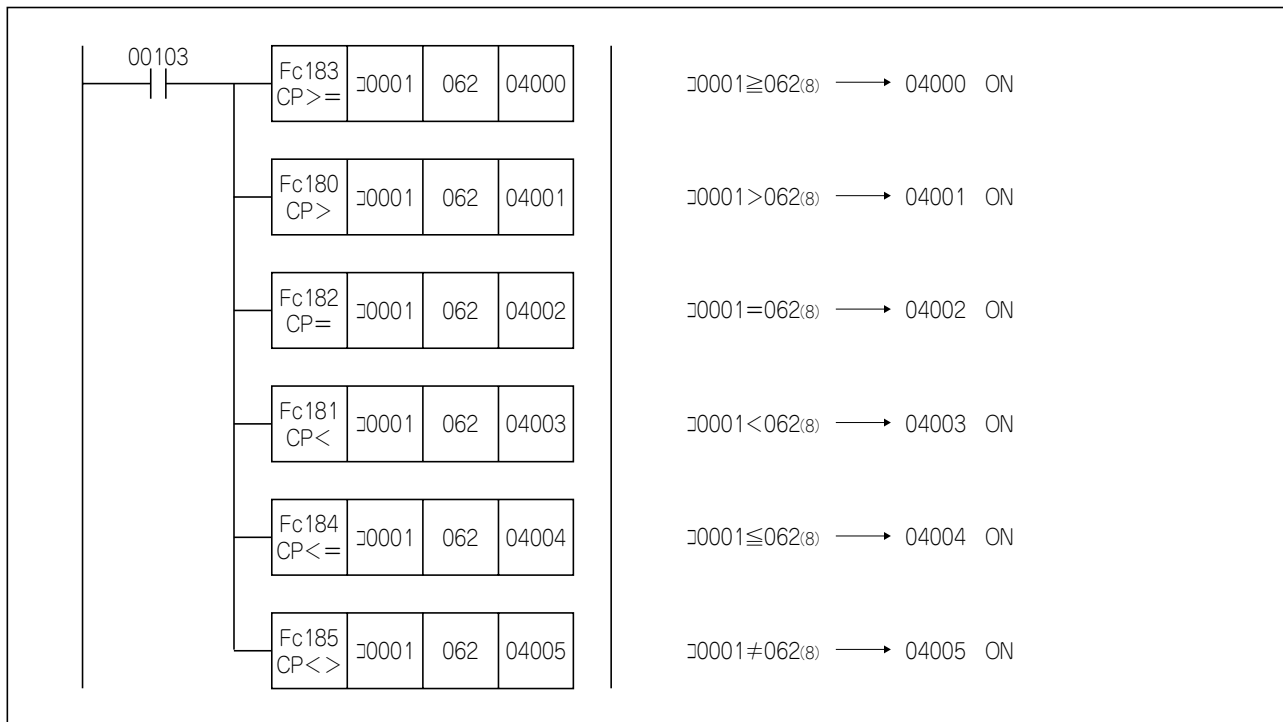
**参考** 設定値をBCD、BINでプログラムする方法



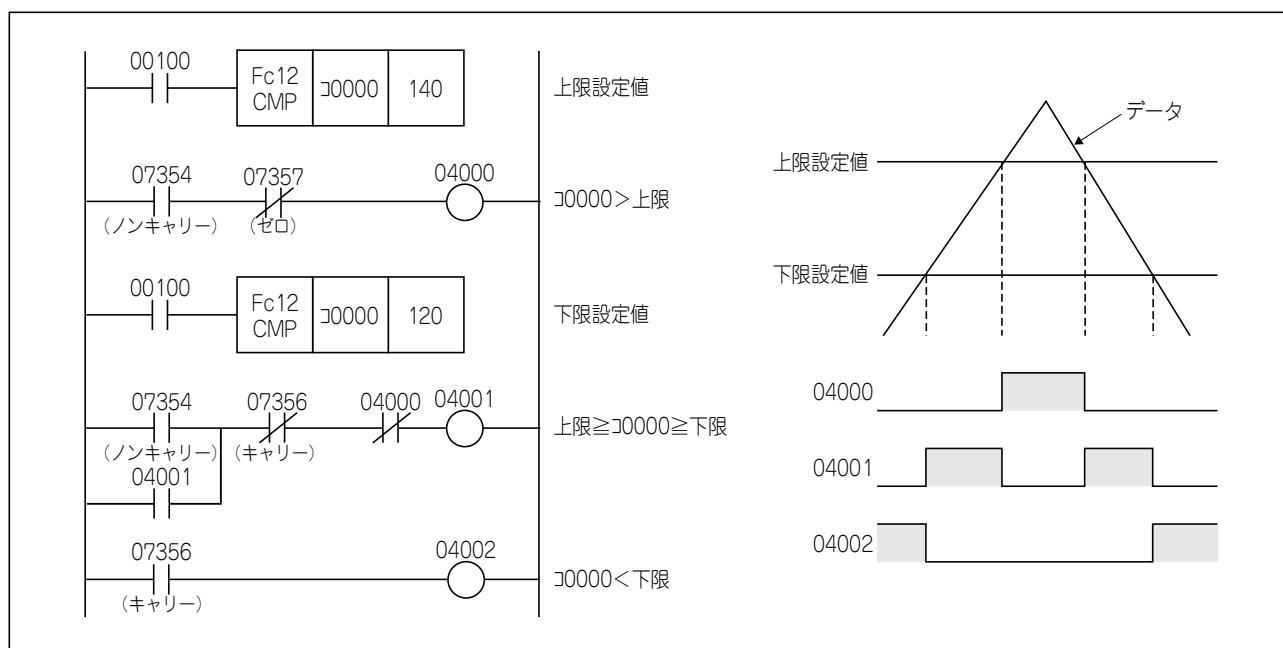
**参考** JW30H, JW50H/70H/100Hでは、16進数の設定値との比較も可能です。(Fx12使用)



**参考** JW30Hには、リレー出力付の比較命令(Fc180～Fc185)があります。



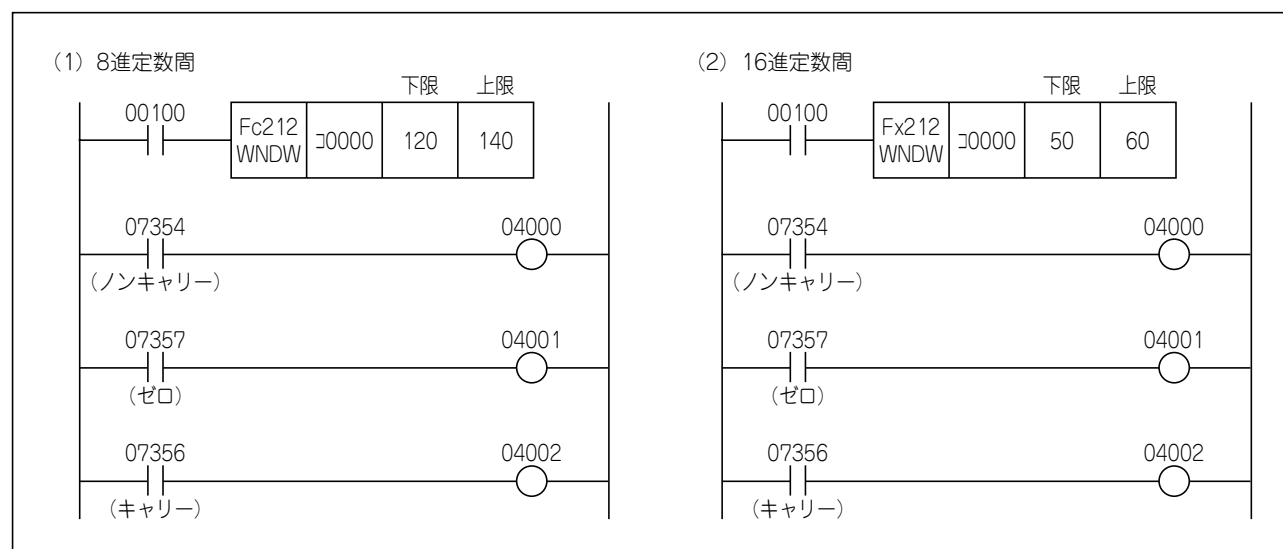
## 2-8 ウィンドウコンパレータ



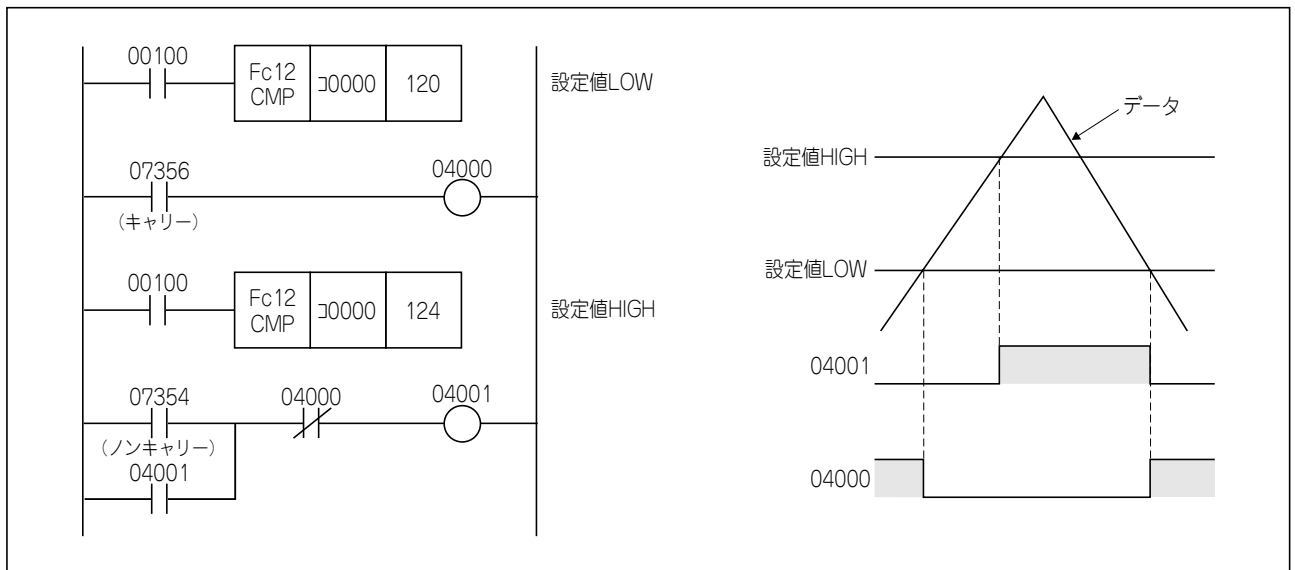
- データが下限設定値と上限設定値の間にあるとき、04001がONします。良品と不良品の判別(GO/NO GO)に使用します。

- 上記の例では上限設定値を140<sup>(8)</sup>(BCDで60)、下限設定値を120<sup>(8)</sup>(BCDで50)としています。04001がONとなるのは $60 \geq 30000 \geq 50$ となります。

**参考** Fc212(ウィンドウコンパレータ命令)を使用すると1命令で実現できます。  
 また、JW30H, JW50H/70H/100Hでは、16進定数を設定値とするウィンドウコンパレータ命令(Fx212)があります。

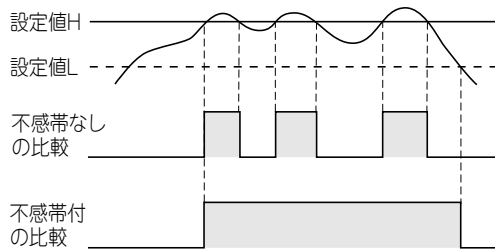


## 2-9 不感帯をもつ比較回路



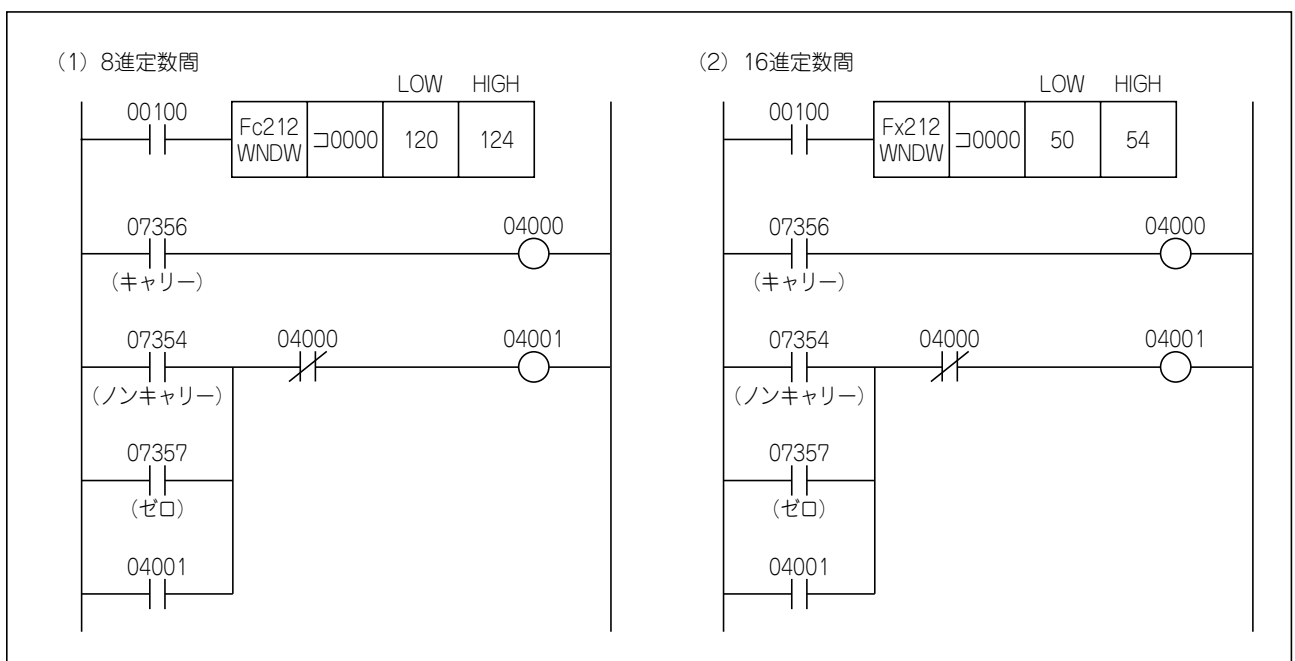
● A/D変換器の出力を入力ユニットに取り込み比較する場合、アナログ信号のわずかな変動でA/D変換器の出力が変化し、PCでの比較結果がON/OFFすることがあります。本プログラムを使用すると、A/D変換器の下位ビットのフラツキによる影響を受けない比較が行えます。

● 上記の例ではHIGH側設定値を124<sup>(8)</sup>(BCDで54)、LOW側設定値を120<sup>(8)</sup>(BCDで50)としています。00000の内容(BCDコードとする)≥54のとき、04001はONとなりますが、一旦ONになると00000の内容<50まではONを維持します。

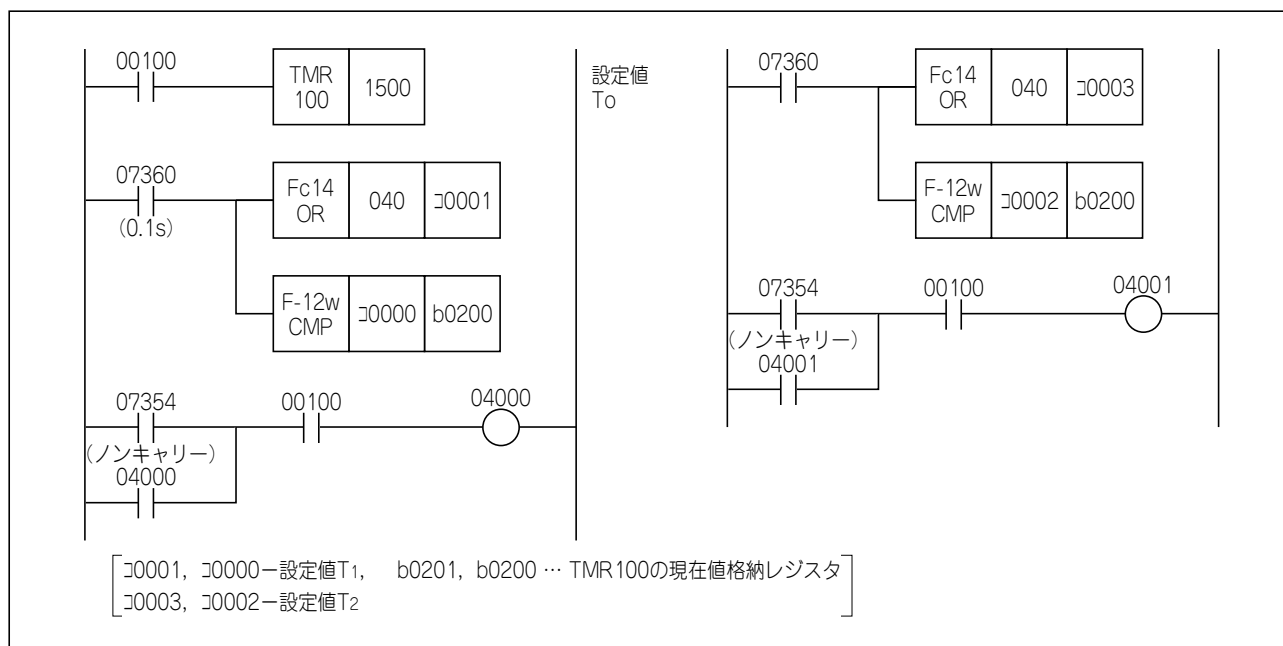


**参考** Fc212(ウィンドウコンパレータ命令)を使用すると、1命令で実現できます。

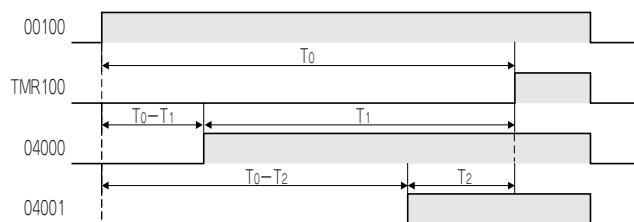
また、JW30H, JW50H/70H/100Hには、16進定数を設定値とするウィンドウコンパレータ命令(Fx212)があります。



## 2-10 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** bXXXX, bXXXX+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
bXXXX	(×10 <sup>0</sup> )				(×10 <sup>-1</sup> )			
	"8"	"4"	"2"	"1"	"8"	"4"	"2"	"1"
bXXXX+1	7	6	5	4	3	2	1	0
	OFF	OFF	*	1	8	4	2	1

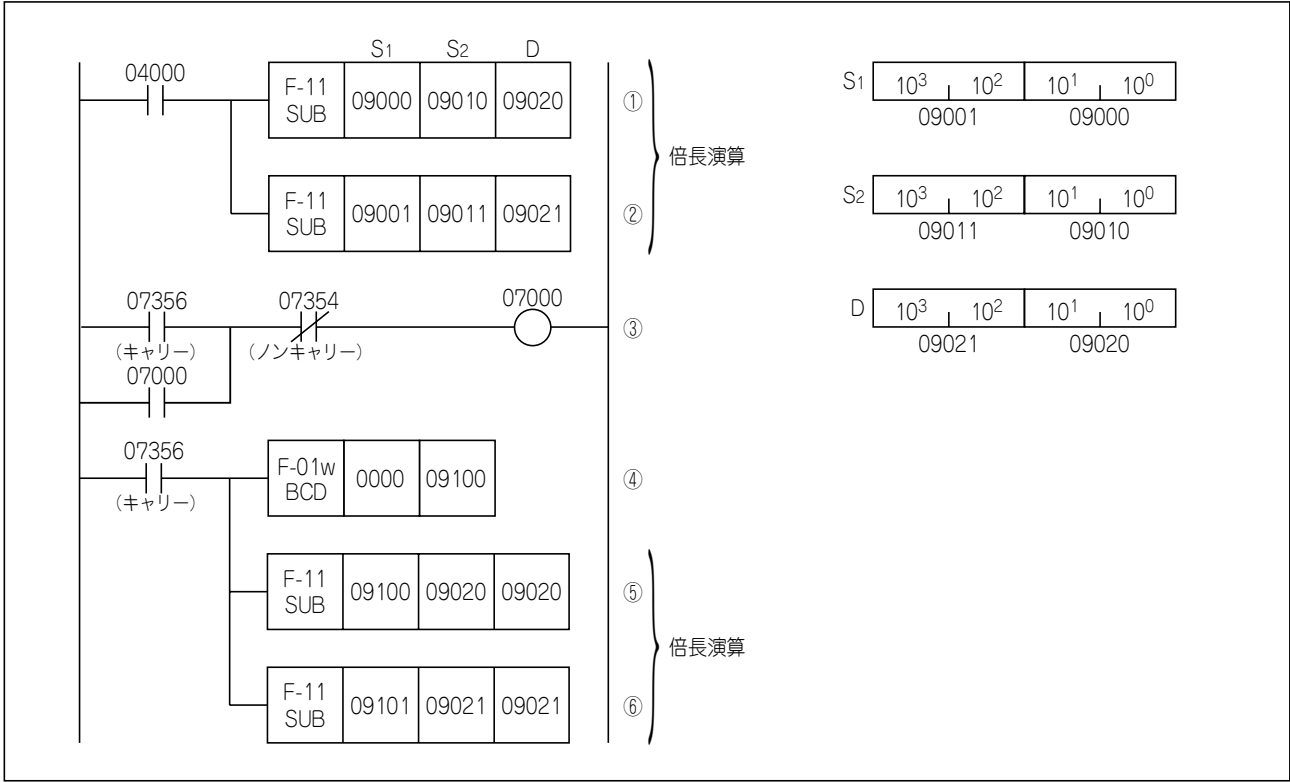
bXXXX+1の\*のビットは1でタイマが動作します。  
Fc14で8進定数040

0 0 1 0 0 0 0 0  
とORし、\*に1を挿入しています。

ただし、JW10の場合は\*のビットは、常に0 (OFF)です。従ってFc14命令は不要です。

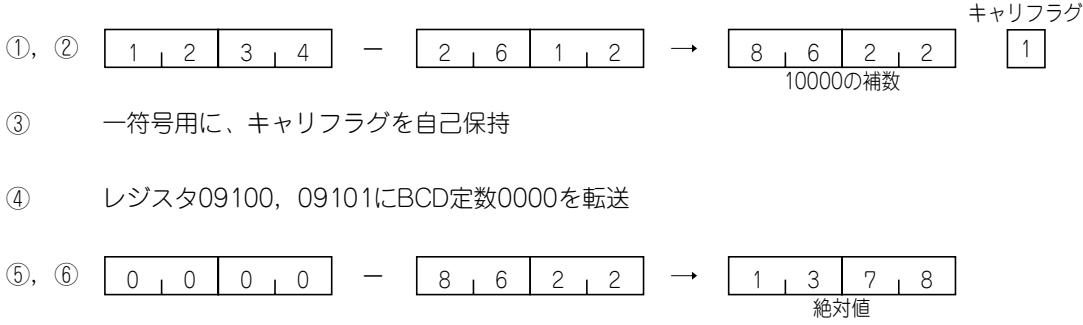
- 「2-16 タイマ現在値の外部出力」、「2-18 タイマ設定値を外部機器から入力」を合わせてご参照ください。

# 2 - 11 BCD 減算結果を符号付絶対値で求める



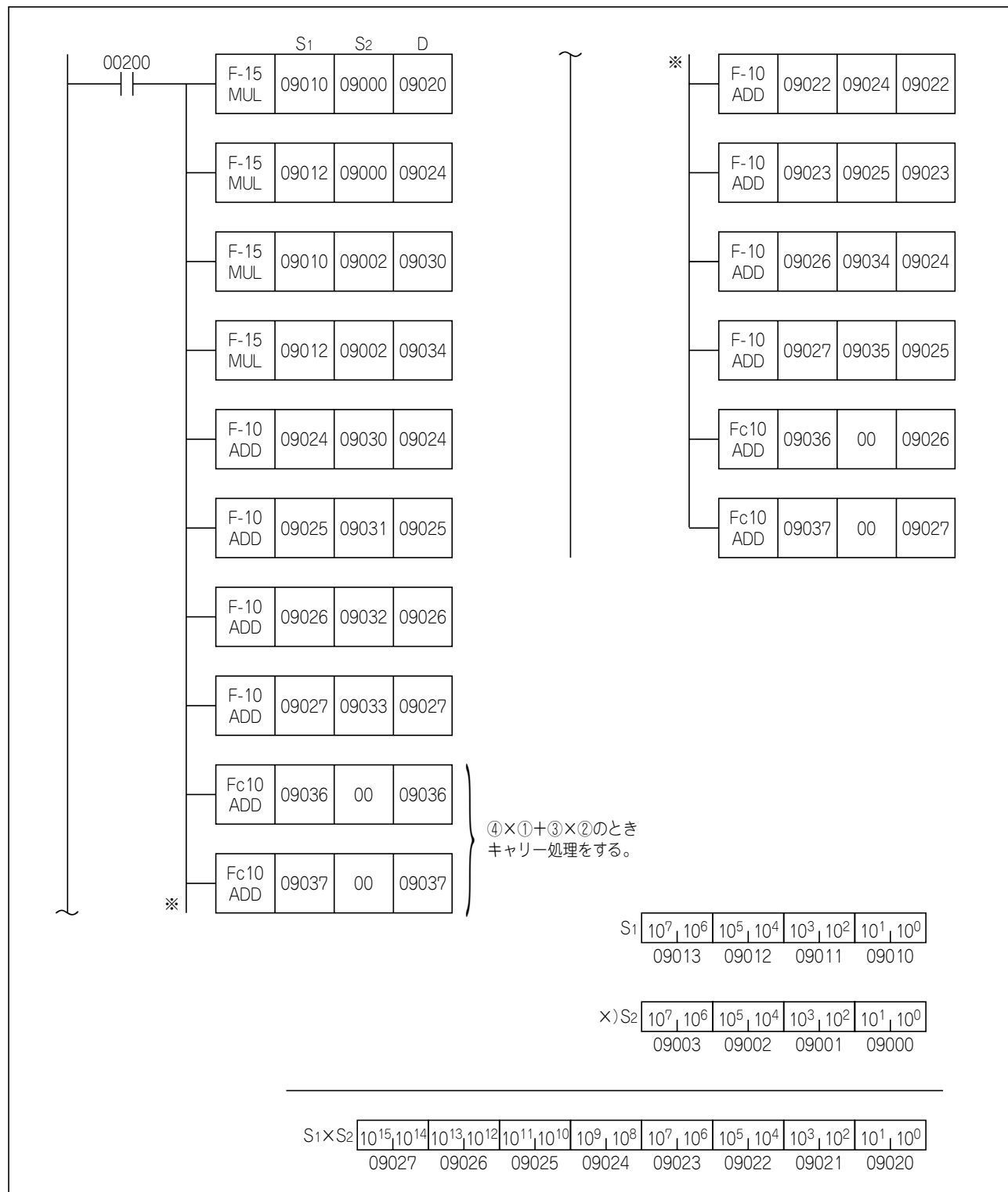
● F-11で<S1><S2>の減算を行うと、答は補数で得られます。  
 (例) 1234-2612→8622(10000の補数)

● 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|S1-S2|=Dとなり、一符号は07000に出力されます。

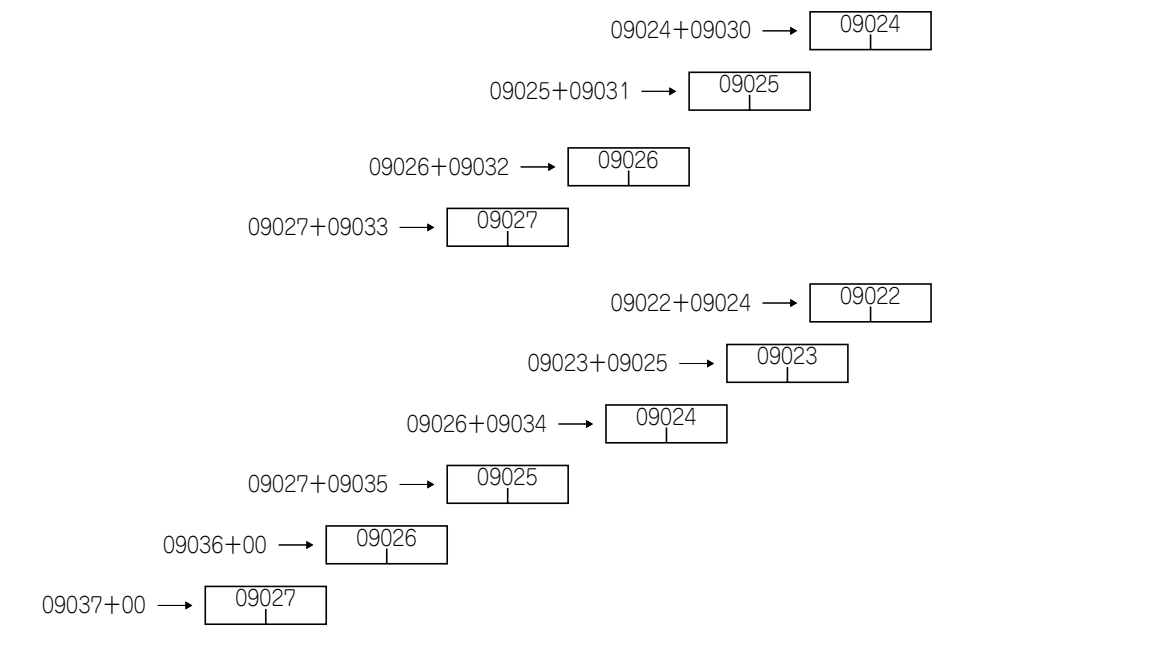
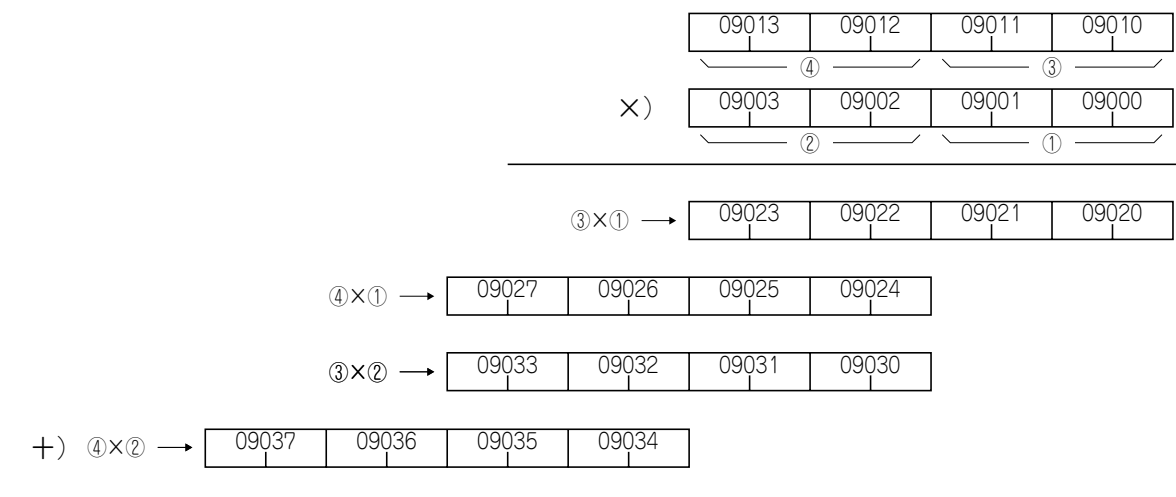


## 2 - 12 BCD8桁の乗算

- F-15はBCD4桁の乗算命令ですが、次のようにプログラムするとBCD8桁×BCD8桁の乗算ができます。

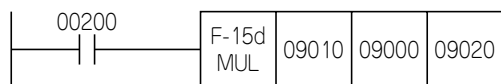






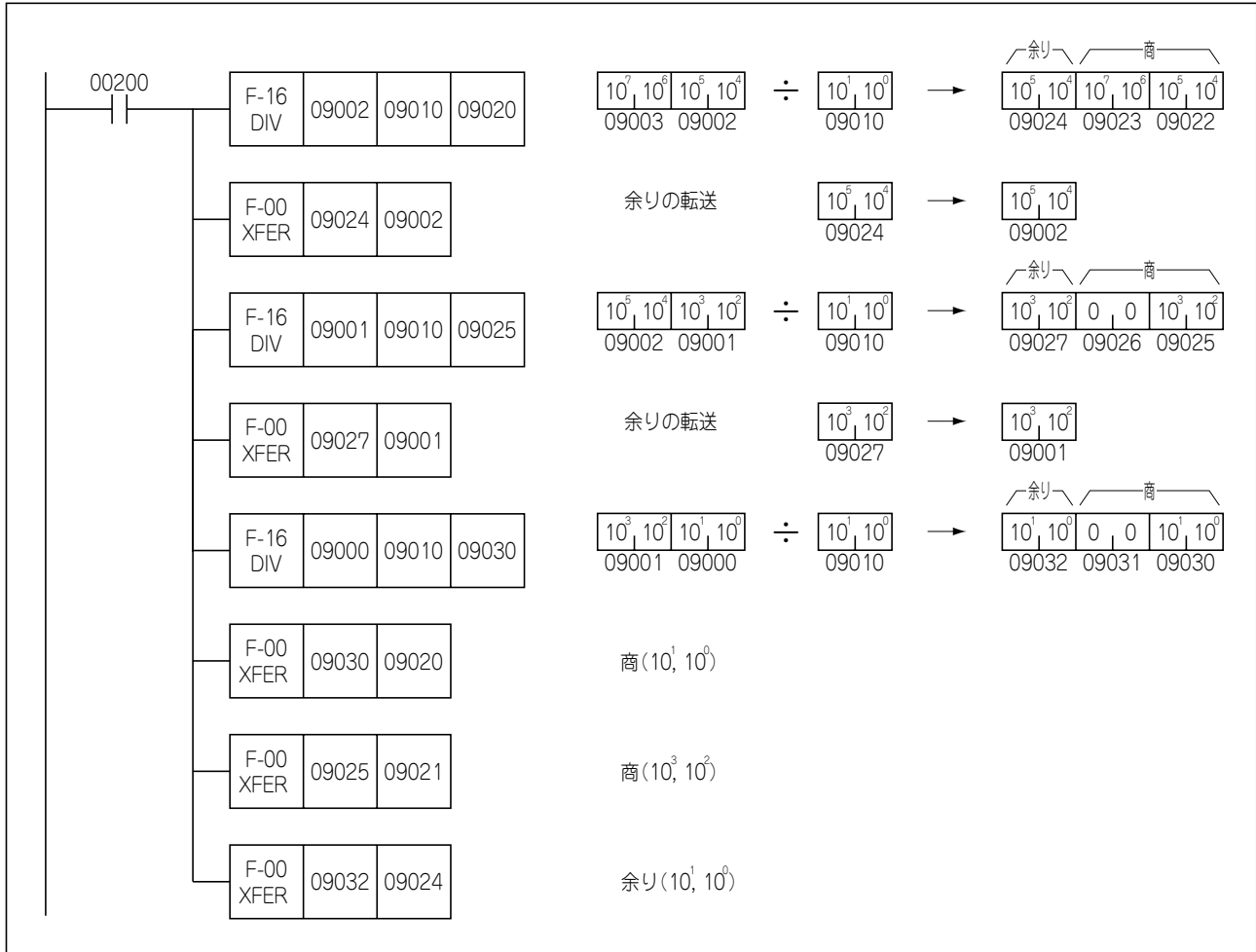
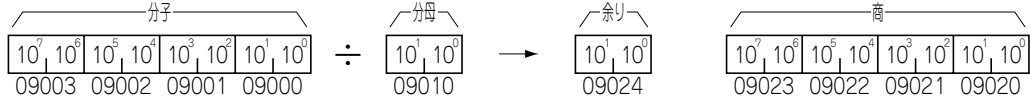
**注1** レジスタの09030~09037は演算の中間結果を一時的に記憶するのに使用しています。

**参考** JW20H, JW30H, JW50H/70H/100Hでは、F-15d命令1個でBCD8桁の乗算が可能です。

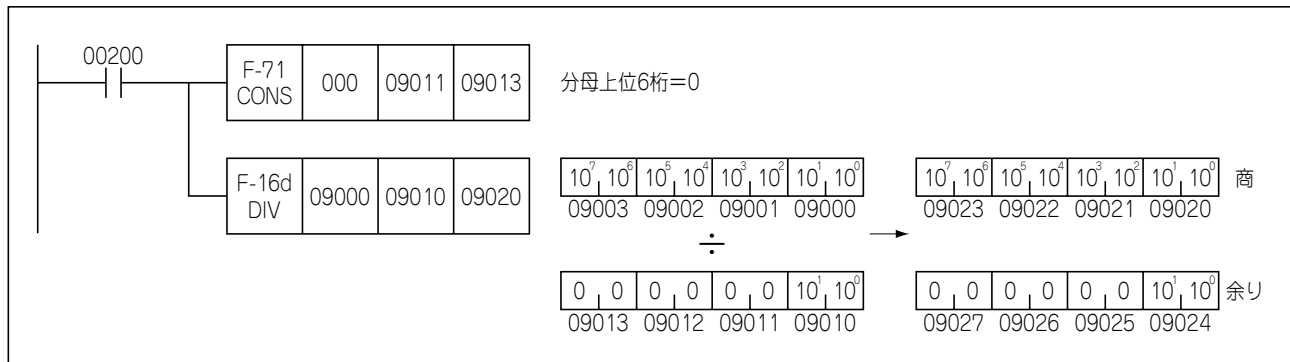


## 2 - 13 BCD8桁÷BCD2桁

- F-16は分子がBCD4桁、分母がBCD2桁の除算命令です。分子がBCD8桁、分母がBCD2桁の除算は次の方法で実現できます。



**参考** JW20H, JW30H, JW50H/70H/100Hでは、F-16d命令1個で分子BCD8桁、分母BCD8桁の除算が可能です。



## 2 - 14 BCD4桁の除算

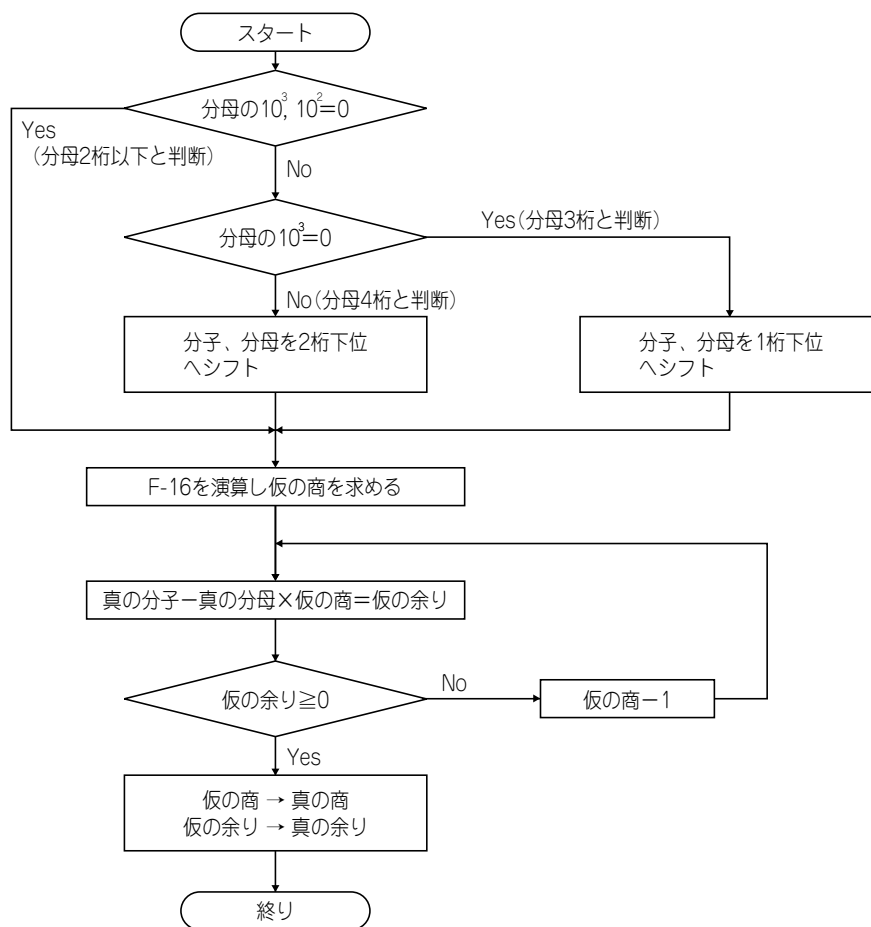
F-16は分子がBCD4桁、分母がBCD2桁の除算命令です。分子、分母ともにBCD4桁の除算は次の方法で実現できます。

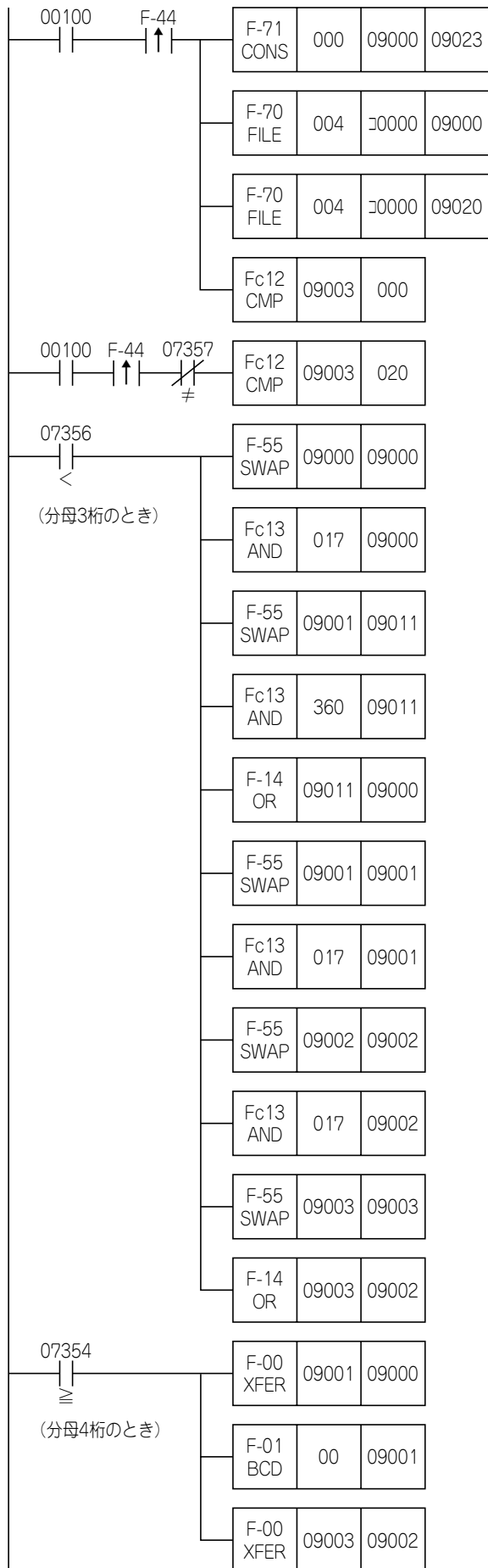
**参考** JW20H, JW30H, JW50H/70H/100Hでは、F-16d命令1個でBCD8桁の除算が可能です。

(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)

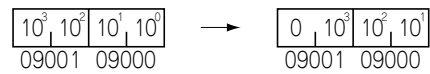




(次ページへ続く)

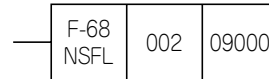
- 09000~09023をクリア
- 入力データを読み込
- 入力データを読み込(演算完了までのデータ変化に対処)
- 分母の $10^3, 10^2=0$ ?
- 分母の $10^3, 10^2 < 1000$ ? (3桁か4桁か)

分子を1桁シフト



(参考)

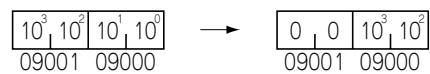
JW20H, JW30H, JW50H/70H/100Hは、桁シフト命令(F-68)で可能です。



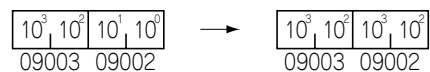
分母を1桁シフト

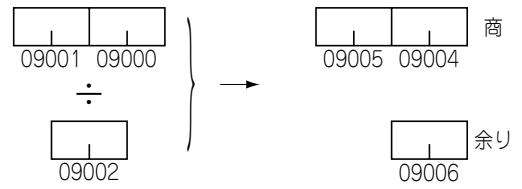
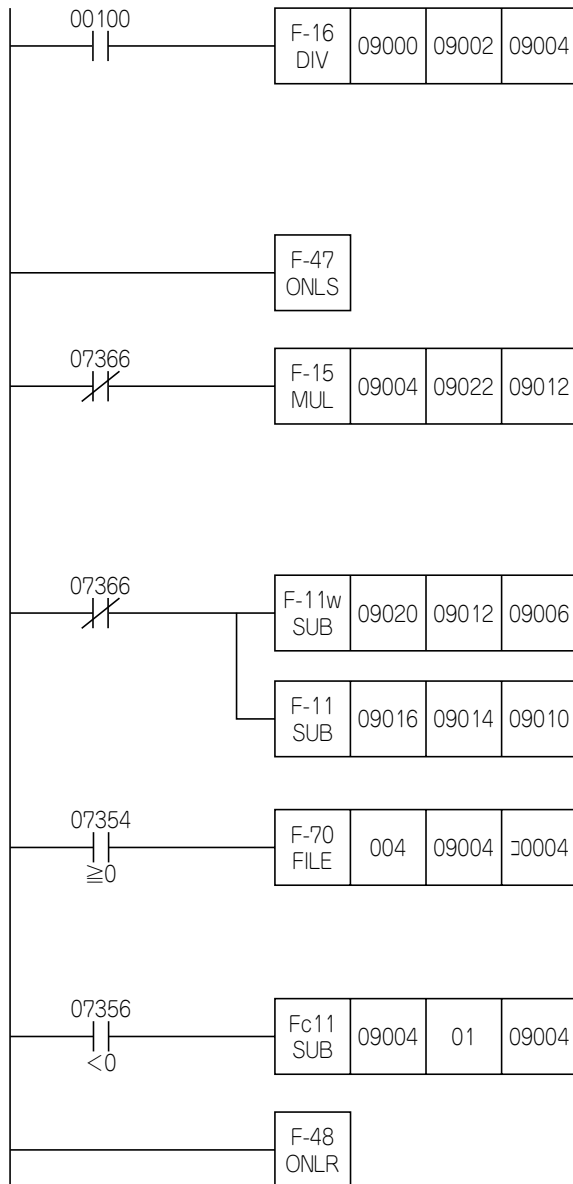


分子を2桁シフト

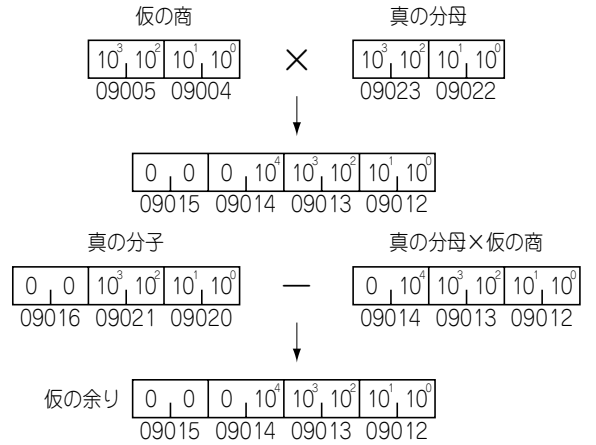


分母を2桁シフト





● レベル演算条件セット



- 仮の余り $\geq 0$ のとき
  - 09004 → 0004 真の商 ( $10^1, 10^0$ )
  - 09005 → 0005 真の商 ( $10^3, 10^2$ )
  - 09006 → 0006 真の余り ( $10^1, 10^0$ )
  - 09007 → 0007 真の余り ( $10^3, 10^2$ )

- 仮の余り $< 0$ のとき 09004(仮の商)を-1

● レベル演算条件リセット

(レジスタ使用状況)

30000	(入力データ) 分子の下2桁
30001	(入力データ) 分子の上2桁
30002	(入力データ) 分母の下2桁
30003	(入力データ) 分母の上2桁
30004	(演算結果) 商の下2桁
30005	(演算結果) 商の上2桁
30006	(演算結果) 余りの下2桁
30007	(演算結果) 余りの上2桁

09000	W.R 分子の下2桁
09001	W.R 分子の上2桁
09002	W.R 分母の下2桁
09003	W.R 分母の上2桁
09004	W.R 仮の商下2桁
09005	W.R 仮の商上2桁
09006	W.R 仮の余り下2桁
09007	W.R 仮の余り中2桁

W.R：ワーキングレジスタ

09010	W.R 仮の余り上2桁
09011	W.R 分母3桁のとき1桁シフト用
09012	W.R 真の分母×仮の商 (10 <sup>1</sup> , 10 <sup>0</sup> )
09013	W.R 真の分母×仮の商 (10 <sup>3</sup> , 10 <sup>2</sup> )
09014	W.R 真の分母×仮の商 (10 <sup>5</sup> , 10 <sup>4</sup> )
09015	W.R 真の分母×仮の商 (10 <sup>7</sup> , 10 <sup>6</sup> )
09016	W.R 商の検定に使用
09017	未使用
09020	W.R 分子の下2桁
09021	W.R 分子の上2桁
09022	W.R 分母の下2桁
09023	W.R 分母の上2桁

(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

①  $12 \div 10 = 1 \dots \text{余り} 2$  (仮の余り)

①  $1234 - 1010 \times 1 = 224 > 0$

⇒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

①  $990 \div 10 = 99$

①  $9900 - 109 \times 99 = -891 < 0$

→  $99 - 1 = 98$

②  $9900 - 109 \times 98 = -782 < 0$

→  $98 - 1 = 97$

③  $9900 - 109 \times 97 = -673 < 0$

→  $97 - 1 = 96$

④  $9900 - 109 \times 96 = -564 < 0$

→  $96 - 1 = 95$

⑤  $9900 - 109 \times 95 = -455 < 0$

→  $95 - 1 = 94$

⑥  $9900 - 109 \times 94 = -346 < 0$

→  $94 - 1 = 93$

⑦  $9900 - 109 \times 93 = -237 < 0$

→  $93 - 1 = 92$

⑧  $9900 - 109 \times 92 = -128 < 0$

→  $92 - 1 = 91$

⑨  $9900 - 109 \times 91 = -19 < 0$

→  $91 - 1 = 90$

⑩  $9900 - 109 \times 90 = 90 > 0$

⇒商90, 余り90

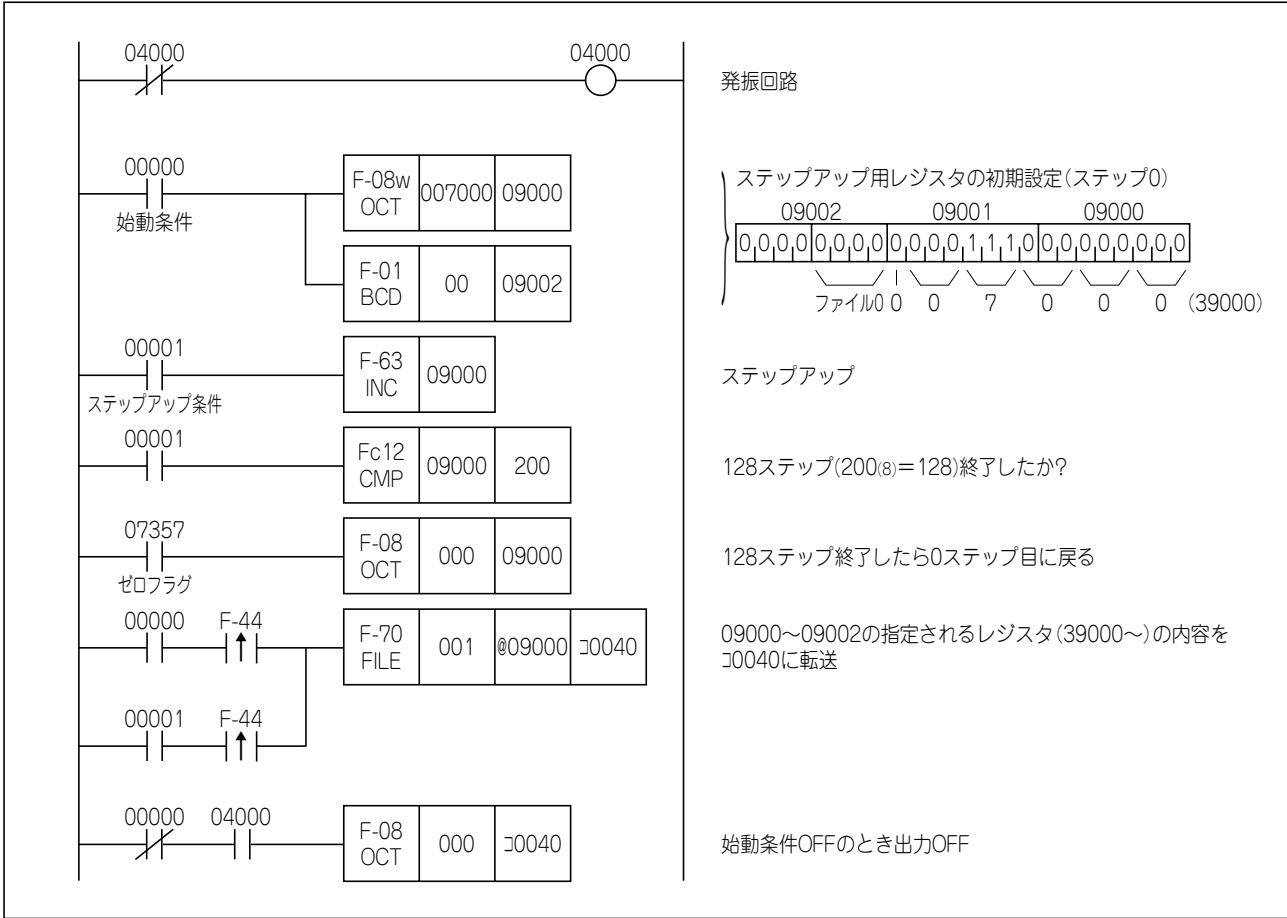
# 2 - 15 ドラムシーケンサ

- 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、レジスタの39000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。(本例では128ステップ書き込んでいます)

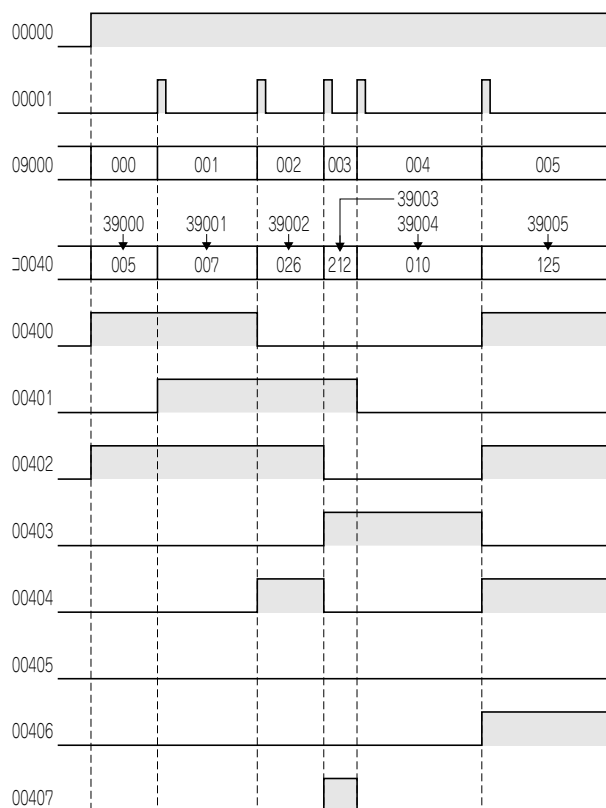
出力		ステップ											127	
		0	1	2	3	4	5	6	7	8	9			127
J0040	00400	○	○				○	○			○			
	00401		○	○	○				○		○			
	00402	○	○	○			○	○	○		○			
	00403				○	○			○	○	○			
	00404			○			○	○	○	○				
	00405							○	○		○			○
	00406							○	○		○			
00407				○										
レジスタNo.		39000	39001	39002	39003	39004	39005	39006	39007	39010	39011			39177
データ(8進)		005	007	026	212	010	125	165	076	130	017			040

○印 … 出力ON

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状態を直接書込めます。



- 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するごとに各ステップの出力状態となります。



- レジスタ09000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。

## 2 - 16 タイマ現在値の外部出力

- タイマ, カウンタ, MDの現在値は、b0000~b3777に格納されます。この現在値を使用すると、外部出力が可能です。
- タイマ, カウンタ, MD番号とbXXXXの現在値格納領域の関係は、次表のようになります。

タイマ、カウンタ、MD番号	現在値格納領域	JW10	JW20H	JW30H	JW50H/70H /100H			
0000	b0000, b0001	TMR CNT	TMR CNT MD DTMR(BCD), UTMR(BCD) DTMR(BIN), UTMR(BIN) DCNT(BCD), UCNT(BCD) DCNT(BIN), UCNT(BIN)					
0001	b0002, b0003							
0002	b0004, b0005							
}	}							
0377	b0776, b0777							
0400	b1000, b1001							
}	}							
0777	b1776, b1777							
1000	b2000, b2001							
}	}							
1377	b2776, b2777	TMR CNT						
1400	b3000, b3001							
}	}							
1777	b3776, b3777							

【注1】 JW50H/70H/100HはTMR1000~TMR1777, CNT1000~CNT1777の設定ができますが、現在値の格納領域は使用できません。



- タイマの場合、bXXXX, bXXXX+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ます。

(1) TMR命令(BCD, 0.1~199.9秒)

PC機種名	タイマ番号	現在値格納レジスタのデータフォーマット
JW10	000~277	$\begin{array}{c} \overbrace{bXXXX+1}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \quad \overbrace{bXXXX}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \\ \begin{array}{ c c c c c c c c } \hline 0 & 0 & 0 & \times 10^3 & \times 10^1 & \times 10^0 & \times 10^{-1} & \\ \hline 1 & 8 & 4 & 2 & 1 & 8 & 4 & 2 & 1 \\ \hline \end{array} \end{array}$
JW20H	000~777	$\begin{array}{c} \overbrace{bXXXX+1}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \quad \overbrace{bXXXX}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \\ \begin{array}{ c c c c c c c c } \hline 0 & 0 & *2 & \times 10^3 & \times 10^1 & \times 10^0 & \times 10^{-1} & \\ \hline 1 & 8 & 4 & 2 & 1 & 8 & 4 & 2 & 1 \\ \hline \end{array} \end{array}$
JW30H	0000~1777	
JW50H/70H/100H	000~777 *1	

- \*1: JW50H/70H/100Hは、TMR1000~TMR1777が設定できますが、現在値の格納領域は使用できません。
- \*2: リセットビット(タイマ動作中は1, 非計測またはリセット状態では0)

(2) TMR命令(BCD, 0.01~19.99秒)

PC機種名	タイマ番号	現在値格納レジスタのデータフォーマット
JW10	300~377	$\begin{array}{c} \overbrace{bXXXX+1}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \quad \overbrace{bXXXX}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \\ \begin{array}{ c c c c c c c c } \hline 0 & 0 & 0 & \times 10^3 & \times 10^0 & \times 10^{-1} & \times 10^{-2} & \\ \hline 1 & 8 & 4 & 2 & 1 & 8 & 4 & 2 & 1 \\ \hline \end{array} \end{array}$
JW20H	700~777 *1	$\begin{array}{c} \overbrace{bXXXX+1}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \quad \overbrace{bXXXX}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \\ \begin{array}{ c c c c c c c c } \hline 0 & 0 & *2 & \times 10^3 & \times 10^0 & \times 10^{-1} & \times 10^{-2} & \\ \hline 1 & 8 & 4 & 2 & 1 & 8 & 4 & 2 & 1 \\ \hline \end{array} \end{array}$
JW30H	0400~0777 *1	
JW50H/70H/100H	400~777 *1	

- \*1: JW20H, JW30H, JW50H/70H/100Hの10msタイマは、システムメモリ#227の設定で決まります。
- \*2: リセットビット(タイマ動作中は1, 非計測またはリセット状態では0)

(3) DTMR(BCD)命令、UTMR(BCD)命令(0.1~799.9秒)

PC機種名	タイマ番号	現在値格納レジスタのデータフォーマット
JW20H	000~777	$\begin{array}{c} \overbrace{bXXXX+1}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \quad \overbrace{bXXXX}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \\ \begin{array}{ c c c c c c c c } \hline *1 & \times 10^2 & \times 10^1 & \times 10^0 & \times 10^{-1} & & & \\ \hline 4 & 2 & 1 & 8 & 4 & 2 & 1 & 8 & 4 & 2 & 1 \\ \hline \end{array} \end{array}$
JW30H	000~777	
JW50H/70H/100H	000~777	

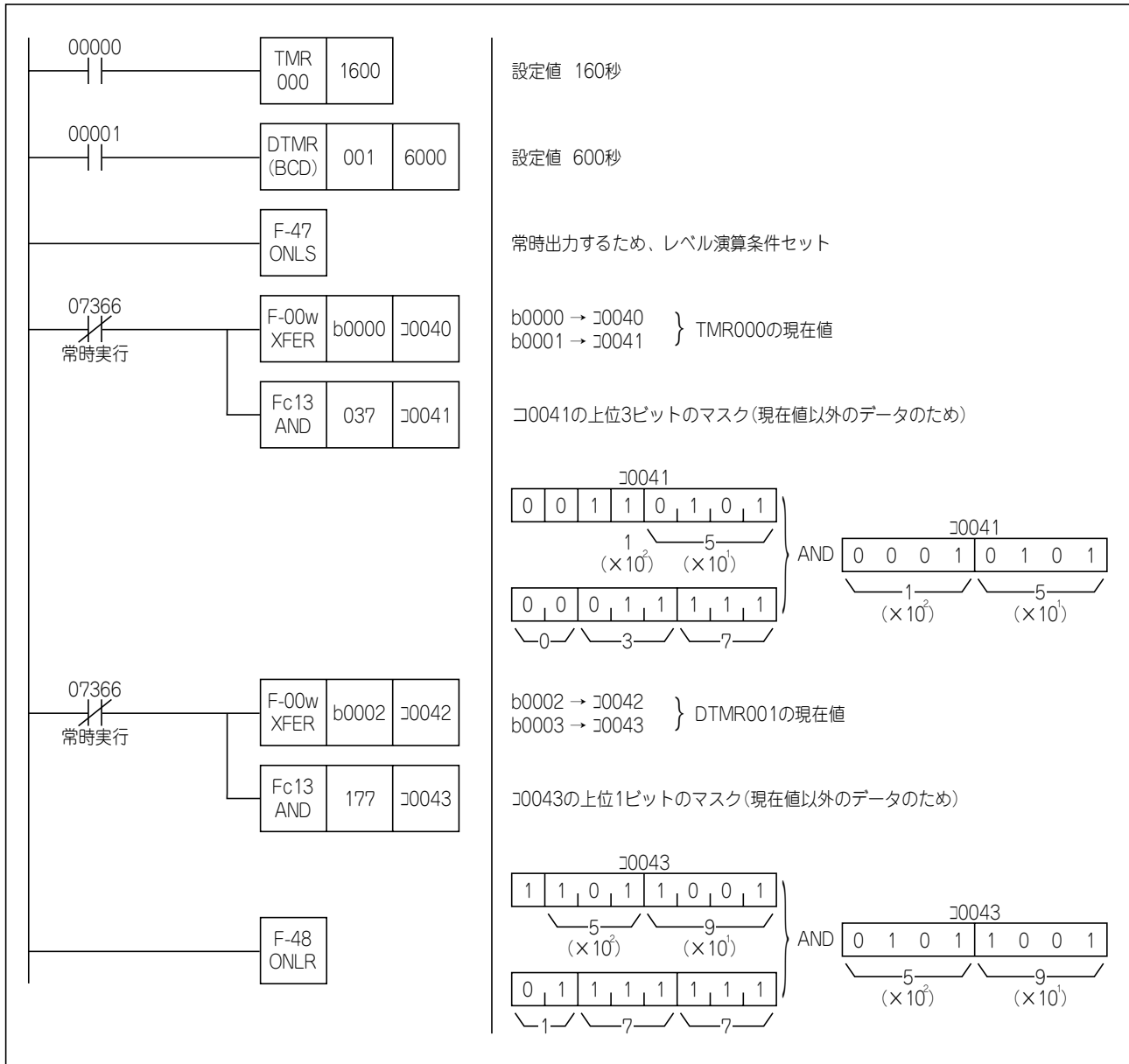
- \*1: リセットビット(タイマ動作中は1, 非計測またはリセット状態では0)

(4) DTMR(BIN)命令、UTMR(BIN)命令(0.1~3276.7秒)

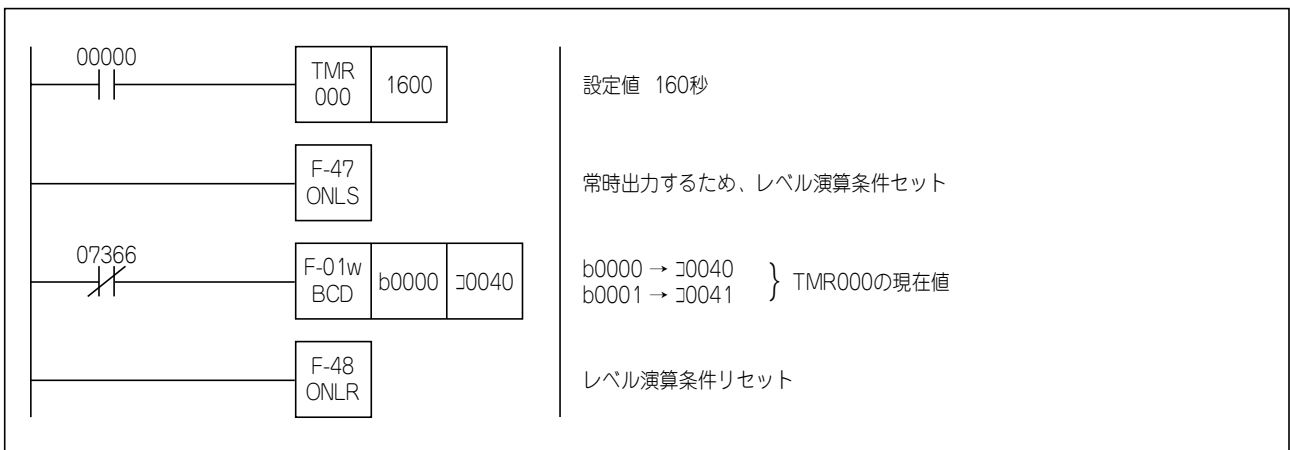
PC機種名	タイマ番号	現在値格納レジスタのデータフォーマット
JW20H	000~777	$\begin{array}{c} \overbrace{bXXXX+1}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \quad \overbrace{bXXXX}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \\ \begin{array}{ c c c c c c c c } \hline *1 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & 2^9 & 2^8 & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ \hline \end{array} \end{array}$
JW30H	000~777	
JW50H/70H/100H	000~777	

- \*1: リセットビット(タイマ動作中は1, 非計測またはリセット状態では0)

■ JW20H, JW30H, JW50H/70H/100Hの場合のプログラム



■ JW10の場合のプログラム



## 2 - 17 カウンタ現在値の外部出力

- カウンタの現在値もタイマと同様にb0000～b3777に格納されます。
- カウンタ番号とbXXXXの現在値格納領域の関係は、「2-16 タイマ現在値の外部出力」をご参照ください。
- カウンタの場合、bXXXX, bXXXX+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ます。

### (1) CNT命令(BCD 1～1999)

PC機種名	カウンタ番号	現在値格納レジスタのデータフォーマット
JW10	000～377	$\begin{array}{c} \overbrace{\text{bXXXXX}+1}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \quad \overbrace{\text{bXXXX}}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \\ \begin{array}{ c c c c c c c c } \hline 0 & 0 & 0 & & & & & \\ \hline & & & 10^3 & & 10^2 & & 10^1 & & 10^0 \\ \hline & & & 1 & 8 & 4 & 2 & 1 & 8 & 4 & 2 & 1 & 8 & 4 & 2 & 1 \\ \hline \end{array} \end{array}$
JW20H	000～777	$\begin{array}{c} \overbrace{\text{bXXXXX}+1}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \quad \overbrace{\text{bXXXX}}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \\ \begin{array}{ c c c c c c c c } \hline 0 & \text{不定} & *2 & & & & & \\ \hline & & & 10^3 & & 10^2 & & 10^1 & & 10^0 \\ \hline & & & 1 & 8 & 4 & 2 & 1 & 8 & 4 & 2 & 1 & 8 & 4 & 2 & 1 \\ \hline \end{array} \end{array}$
JW30H	0000～1777	
JW50H/70H/100H	000～777 *1	

\*1：JW50H/70H/100Hは、CNT 1000～CNT 1777が設定できますが、現在値の格納領域は使用できません。

\*2：リセットビット(カウンタ動作中は1, 非計測またはリセット状態では0)

### (2) DCNT(BCD)命令、UCNT(BCD)命令(1～7999)

PC機種名	カウンタ番号	現在値格納レジスタのデータフォーマット
JW20H	000～777	$\begin{array}{c} \overbrace{\text{bXXXXX}+1}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \quad \overbrace{\text{bXXXX}}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \\ \begin{array}{ c c c c c c c c } \hline *1 & & & 10^3 & & 10^2 & & 10^1 & & 10^0 \\ \hline & 4 & 2 & 1 & 8 & 4 & 2 & 1 & 8 & 4 & 2 & 1 & 8 & 4 & 2 & 1 \\ \hline \end{array} \end{array}$
JW30H	000～777	
JW50H/70H/100H	000～777	

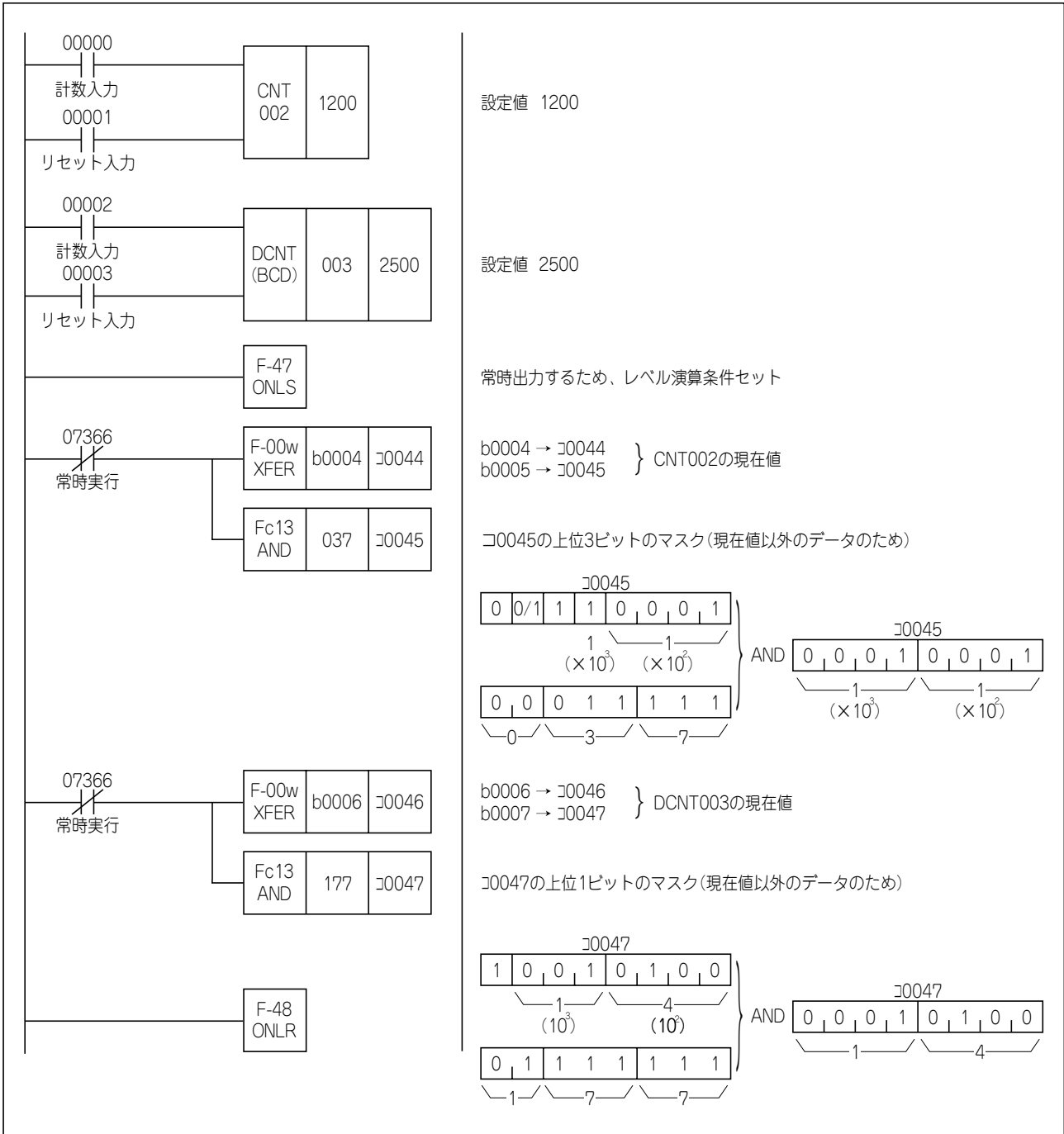
\*1：リセットビット(カウンタ動作中は1, 非計測またはリセット状態では0)

### (3) DCNT(BIN)命令、UCNT(BIN)命令(1～32767)

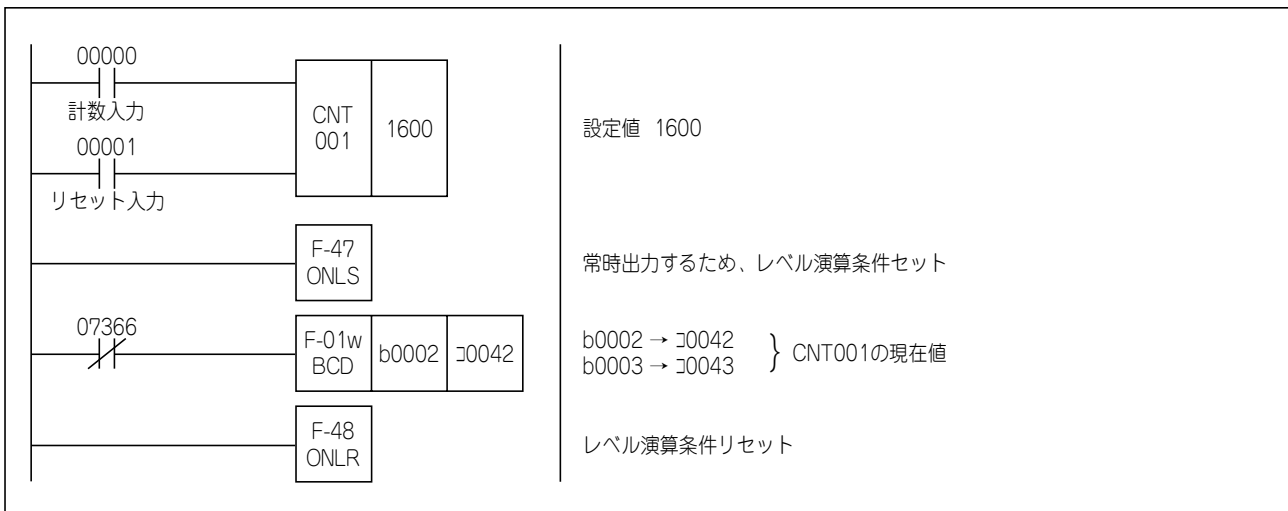
PC機種名	カウンタ番号	現在値格納レジスタのデータフォーマット
JW20H	000～777	$\begin{array}{c} \overbrace{\text{bXXXXX}+1}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \quad \overbrace{\text{bXXXX}}^{7\ 6\ 5\ 4\ 3\ 2\ 1\ 0} \\ \begin{array}{ c c c c c c c c } \hline *1 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 & 2^9 & 2^8 & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ \hline \end{array} \end{array}$
JW30H	000～777	
JW50H/70H/100H	000～777	

\*1：リセットビット(カウンタ動作中は1, 非計測またはリセット状態では0)

■ JW20H, JW30H, JW50H/70H/100Hの場合のプログラム

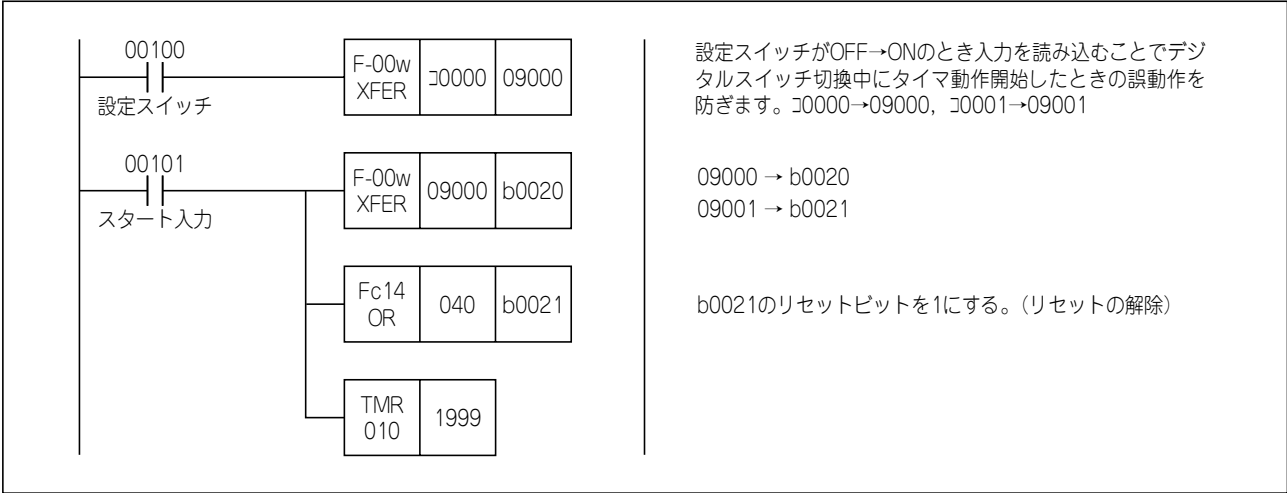


■ JW10の場合のプログラム



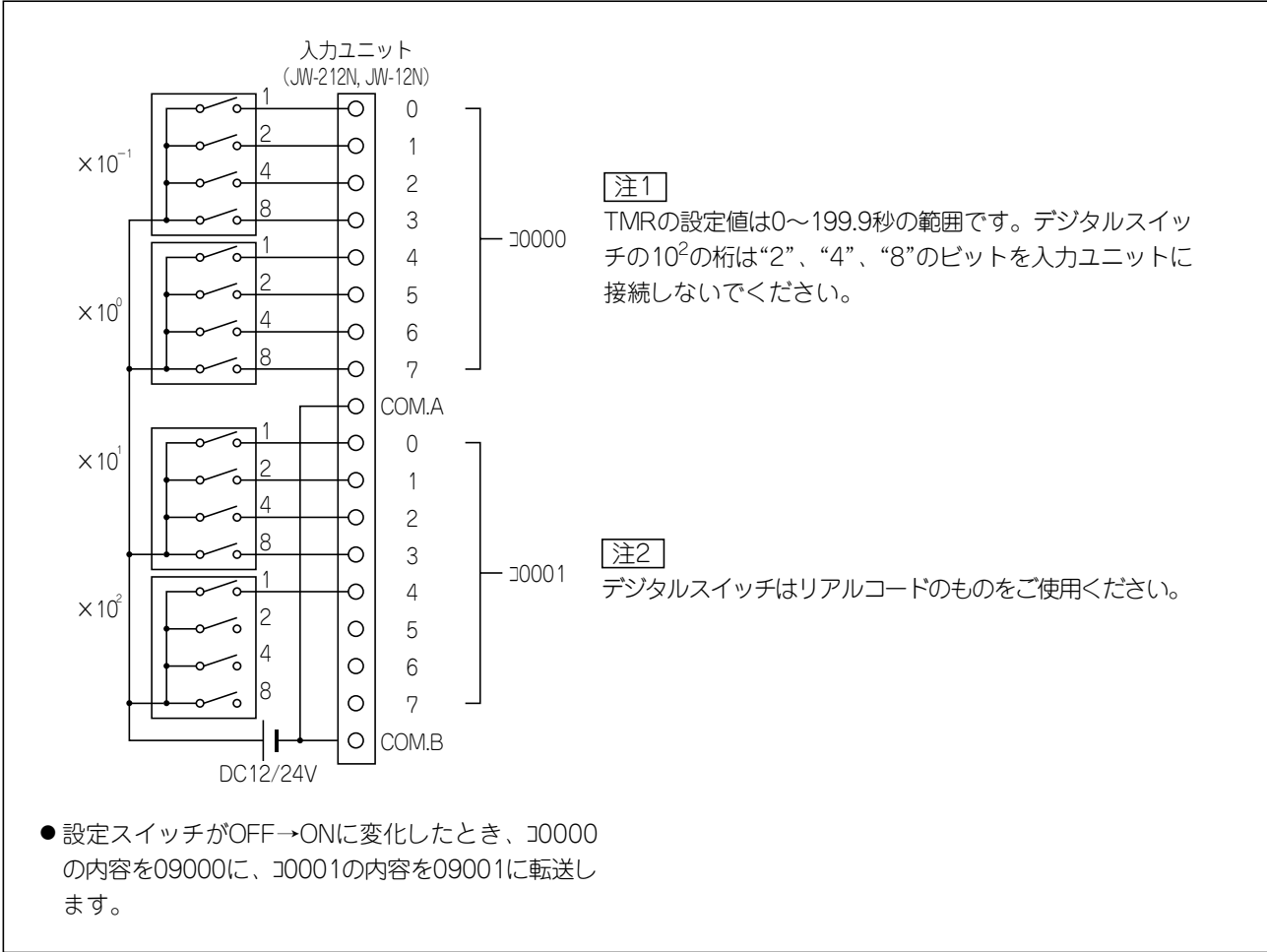
# 2 - 18 タイマの設定値を外部機器から入力

■ JW20H, JW30H, JW50H/70H/100Hの場合のプログラム

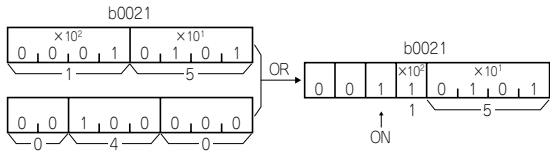


● PCのRUN中にデジタルスイッチ等の外部機器からTMRの設定値を変更できます。(正しくは現在値を外部から書換えています。)

● 00000, 00001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OFF→ONに変化したとき、
  - ①09000の内容をb0020に、09001の内容をb0021に転送します。
  - ②b0021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のプログラム上の設定値(本例の場合1999)がb0020、b0021に再度書込まれ、外部設定が無効となります。



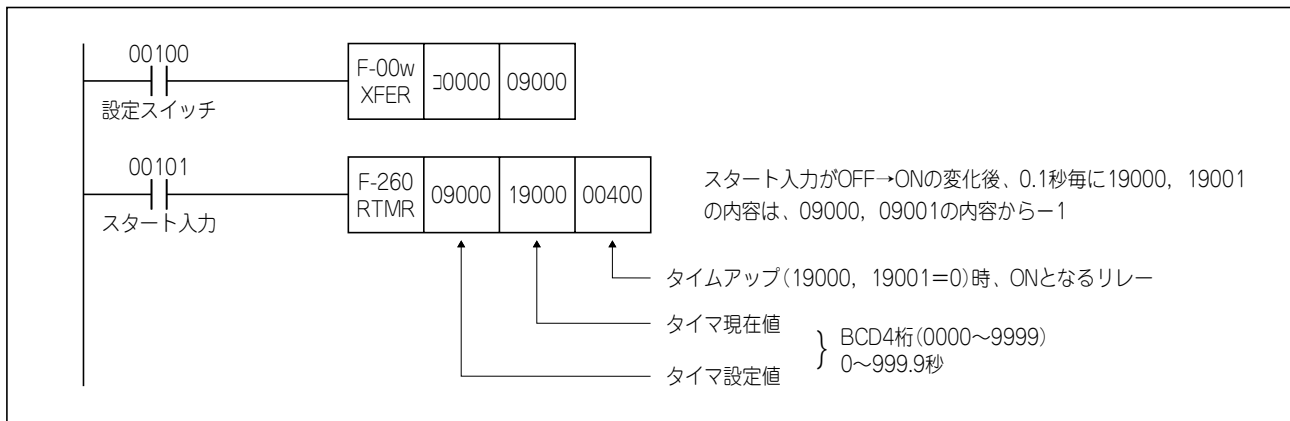
タイマの現在値格納レジスタ (b0020, b0021) のデータフォーマットについては、「2-16 タイマ現在値の外部出力」をご参照ください。

- スタート入力ONの間、0.1秒ご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TMR接点がONとなります。
- スタート入力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09000、09001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。)

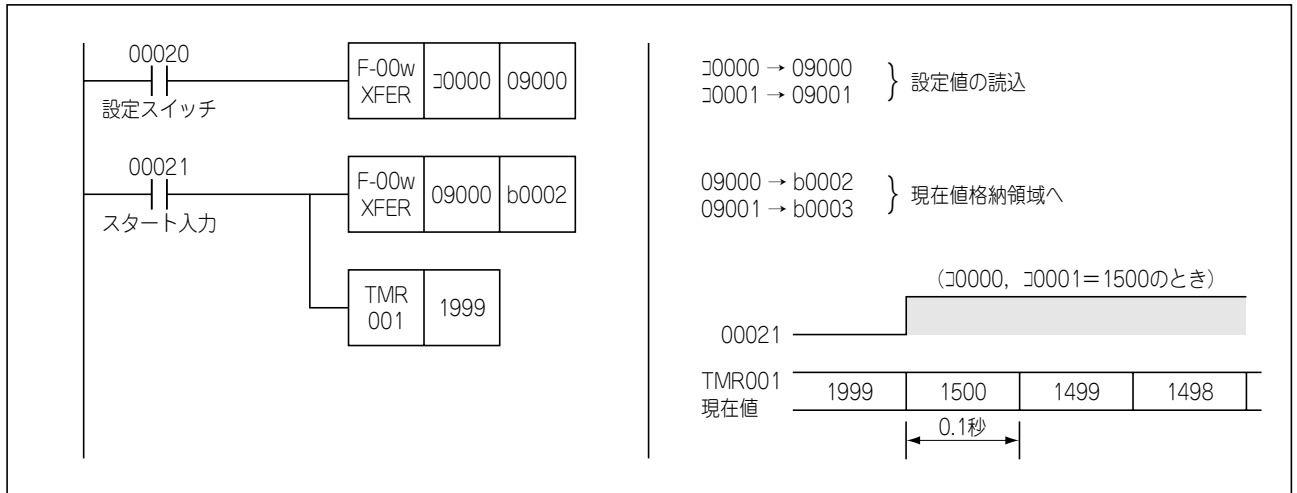
**注1** スタート入力ONのとき、プログラマで強制セットすると、現在値は0となり、TMR接点をONとすることができます。強制リセットすると、現在値はプログラム上の設定値(本例の場合1999)になり、デジタルスイッチの設定値にはなりませんのでご注意ください。

**注2** スタート入力ONのとき(タイマ動作中)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。

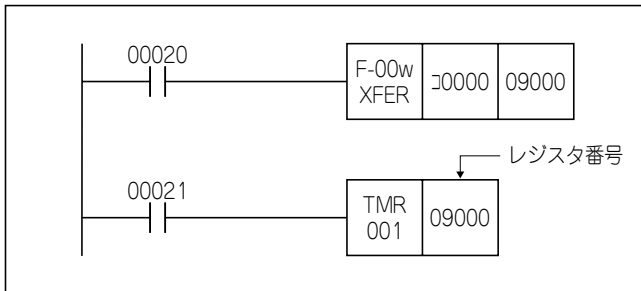
**参考** JW30H, JW50H/70H/100Hの場合、F-260(レジスタ指定減算タイマ)を使用してもタイマ設定値を変更できます。



■ JW10の場合のプログラム

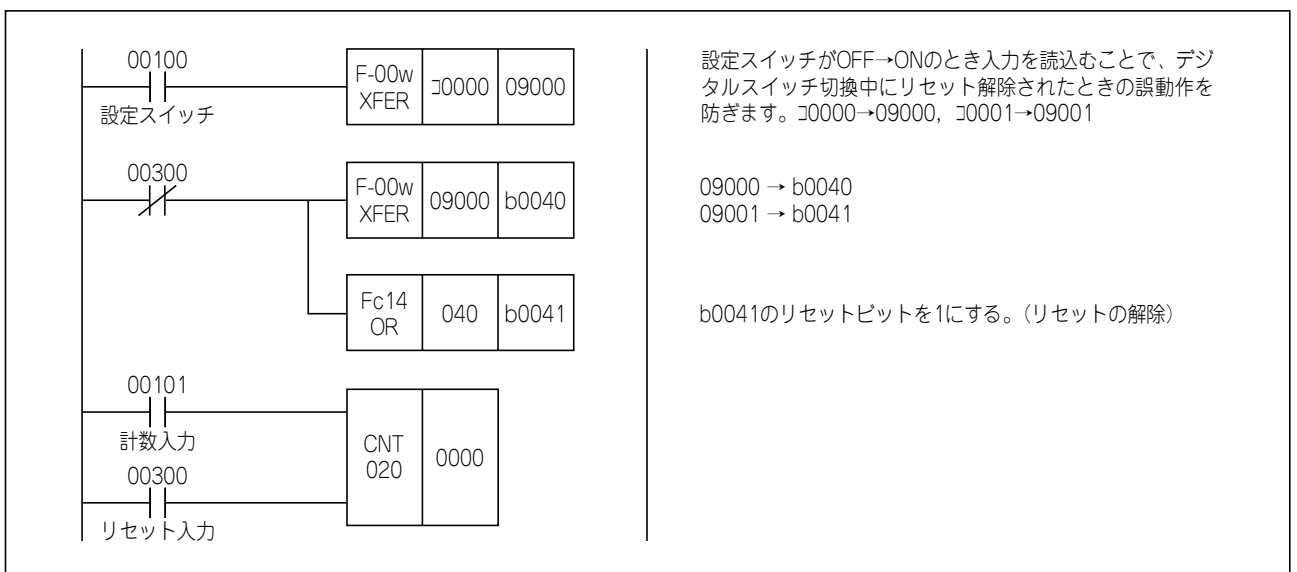


【参考】 JW10はタイマ設定値にレジスタ番号を設定すると、上記プログラムと同様の動作となります。



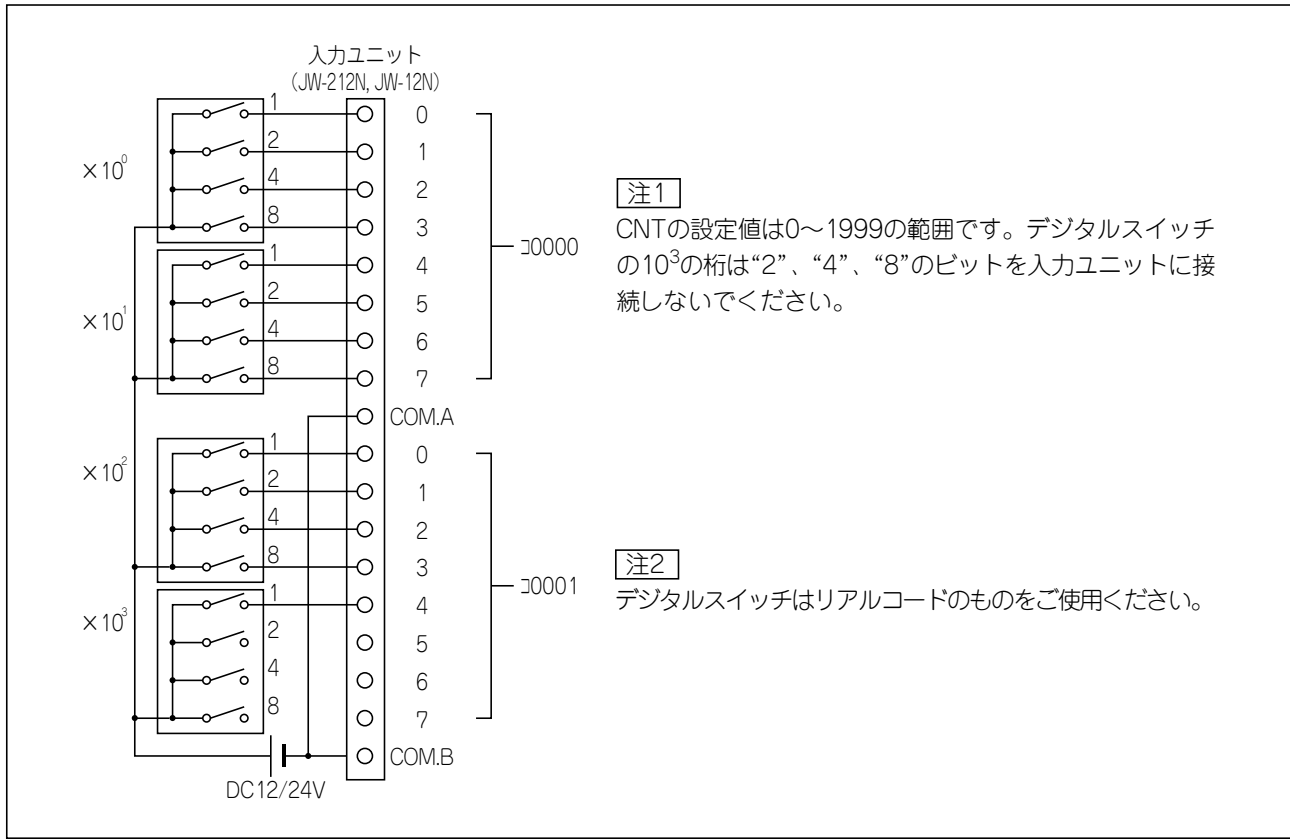
## 2 - 19 カウンタの設定値を外部機器から入力

■ JW20H, JW30H, JW50H/70H/100Hの場合のプログラム

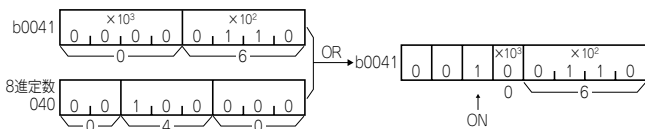


● PCのRUN中にデジタルスイッチ等の外部機器からCNTの設定値を変更できます。(正しくは現在値を外部から書換えています。)

● 30000, 30001にはデジタルスイッチ等の外部機器を接続します。



- リセット入力がON→OFFに変化したとき、
  - ①09000の内容をb0040に、09001の内容をb0041に転送します。
  - ②b0041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b0040, b0041に再度書込まれ、外部設定が無効となります。



カウンタの現在値格納レジスタ (b0040, b0041) のデータフォーマットについては、「2-17 カウンタ現在値の外部出力」をご参照ください。

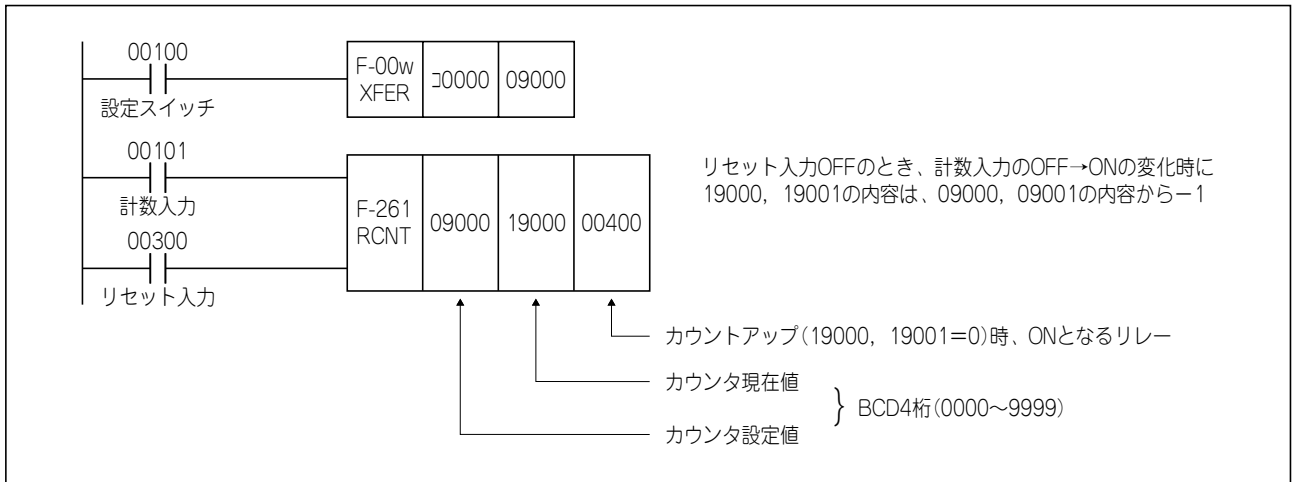
- リセット入力がOFFの間、計数入力がOFF→ONに変化するごとに外部から書込んだ現在値から減算され、現在値が0になるとCNT接点がONになります。
- リセット入力がONになると、CNTの現在値はプログラム上の設定値(本例の場合0000)になり、CNT接点はOFFになります。リセット入力が再びOFFになると、データメモリの30000, 30001の内容が現在値として転送されるため、プログラム上の設定値(本例の場合0000)は無視されます。(0～1999の任意の値をプログラムしてください。)

**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。

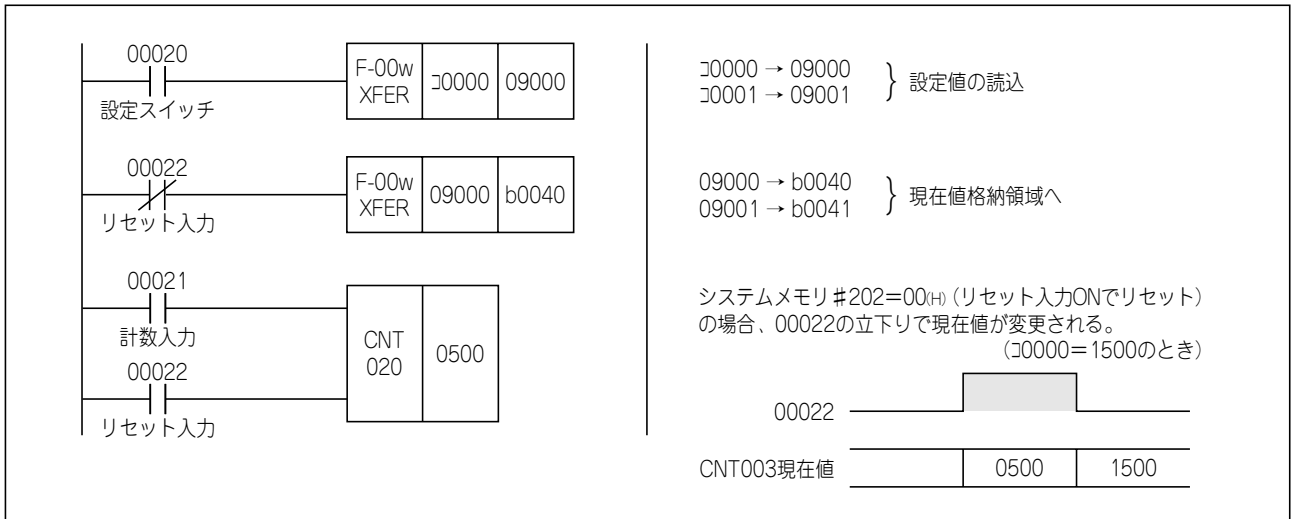
**注2** リセット入力がOFFのとき(カウンタ動作中)に、デジタルスイッチの数値を変更しても現在値は変更されません。変更後の設定値が有効となるのは、一旦リセット入力がONとなり、次にOFFとなったときからとなります。



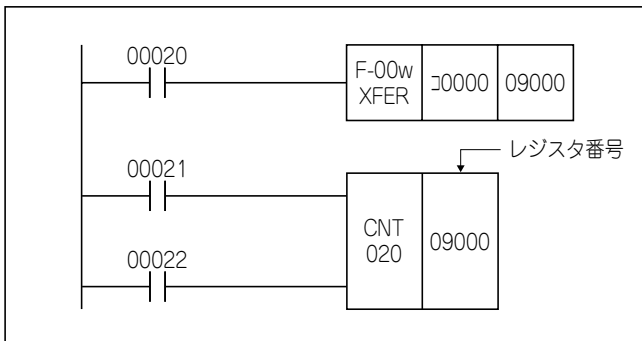
**参考** JW30H, JW50H/70H/100Hの場合、F-261(レジスタ指定減算カウンタ)を使用してもカウンタ設定値を変更できます。



■ JW10の場合のプログラム

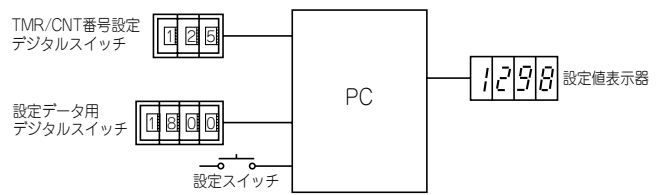


**参考** JW10はカウンタ設定値にレジスタ番号を設定すると、上記プログラムと同様の動作となります。

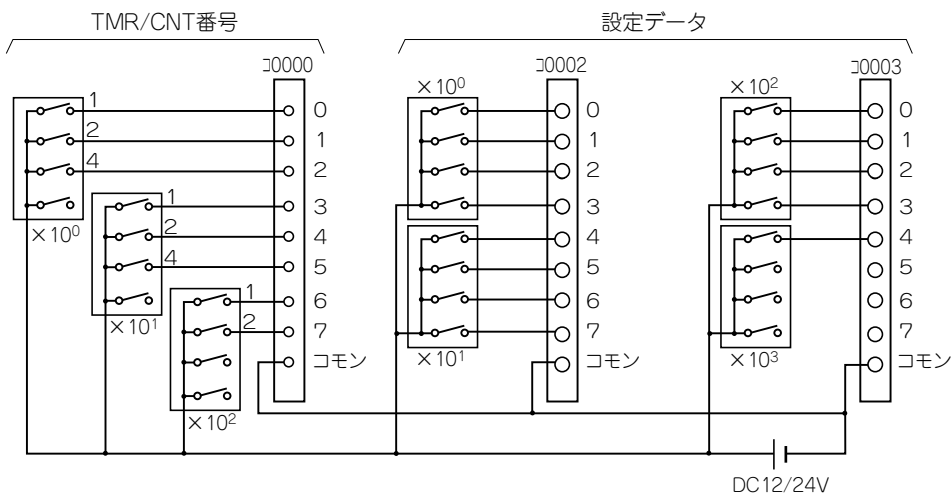


## 2 - 20 複数のタイマ、カウンタの設定値を外部機器から入力

- 外部設定する必要のあるタイマ、カウンタが多数あるとき、2-18, 2-19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256点のTMR/CNT(000~377)を外部設定式にすることができます。

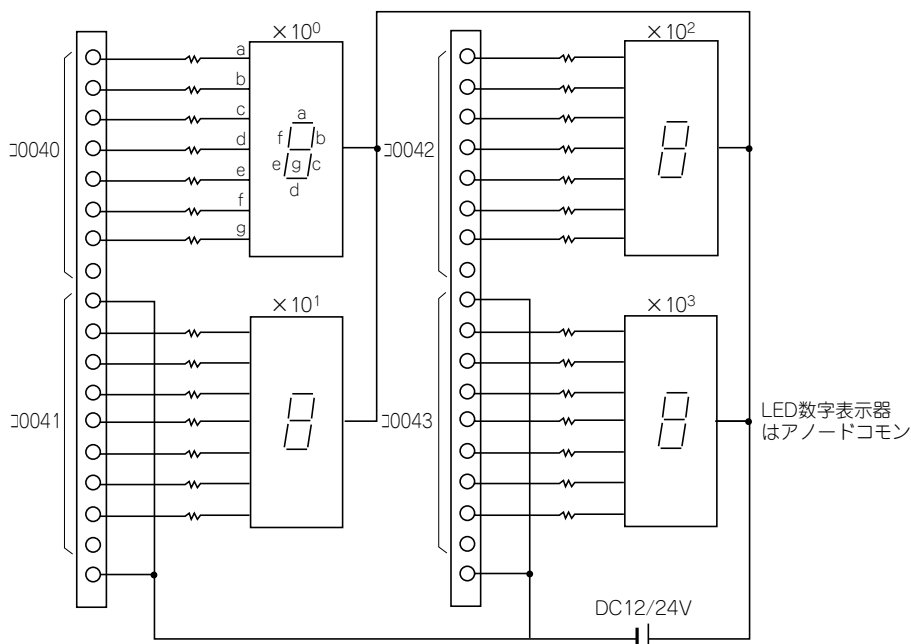


- (1) 外部接続(JW20H, JW30H, JW50H/70H/100H)
- a. 設定用外部接続



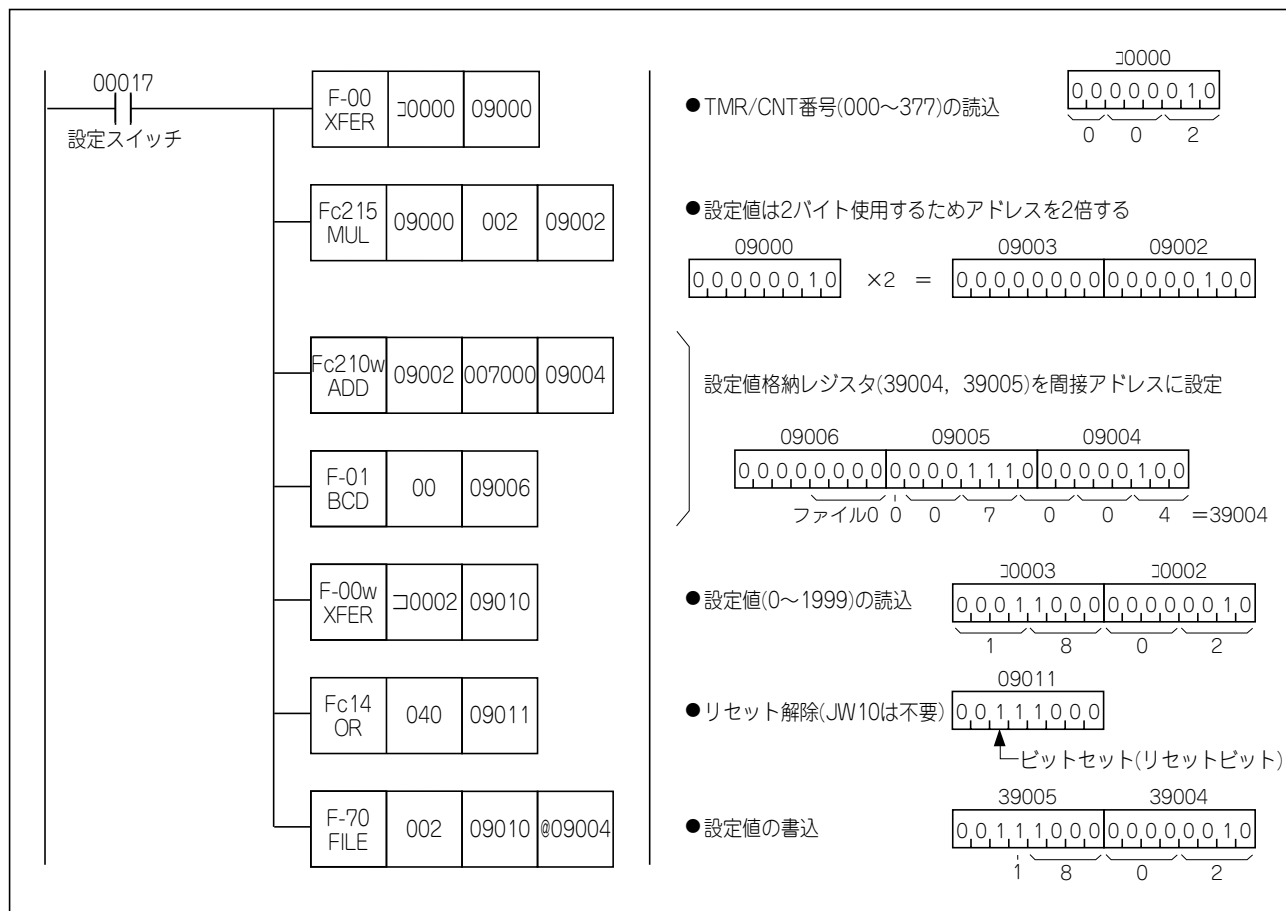
J0000	TMR/CNT番号	8進数で設定(000~377)	デジタルスイッチはリアルコード
J0002	設定値下2桁	BCDで設定(00~99)	
J0003	設定値上2桁	BCDで設定(00~19)	
00017	設定スイッチ	ONで設定	

- b. 設定値表示用外部接続(F-52使用)



(2) 設定値書込プログラム

- ・ TMR/CNT番号 002<sup>(8)</sup>
- ・ 設定値 1802<sup>(BCD)</sup> } の場合



- 設定スイッチをON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、レジスタ39000~39777にデータテーブルとして格納します。

**参考** JW30H(JW-32CUH/H1, JW-33CUH/H1/H2/H3), JW50H/70H/100Hでは、ファイルレジスタをデータテーブルに使用すると大量のデータを格納できます。  
(TMR/CNT400~777を指定する場合など)

TMR/CNT番号	レジスタ	
000	39000	下位2桁
	39001	上位2桁
001	39002	下位2桁
	39003	上位2桁
002	39004	下位2桁
	39005	上位2桁
⋮	⋮	⋮
377	39776	下位2桁
	39777	上位2桁

(3) 設定値表示プログラム

設定値書込プログラムに続けて

00017 設定スイッチ	F-70 FILE	002	@09004	09020
	F-52 →7SEG	09020	∩0040	
	F-55 SWAP	09020	09020	
	F-52 →7SEG	09020	∩0041	
	Fc13 AND	037	09021	
	F-52 →7SEG	09021	∩0042	
	F-55 SWAP	09021	09021	
	F-52 →7SEG	09021	∩0043	

- データテーブルから設定値の読み
 

09021	09020
0 0 1 1 1 0 0 0	0 0 0 0 0 0 0 1 0
1 8	0 2
- 09020の下4ビットを7SEGに変換→∩0040
 

2	g f e d c b a
	0 1 0 1 1 0 1 1
- 09020の上下4ビットを交換
 

09020
0 0 1 0 0 0 0 0
2 0
g f e d c b a
0 0 1 1 1 1 1 1
- 09020の下4ビットを7SEGに変換→∩0041
 

0	g f e d c b a
	0 0 1 1 1 1 1 1
- 09021の上位3ビットマスク
 

09021
0 0 0 1 1 0 0 0
1 8
g f e d c b a
0 1 1 1 1 1 1 1
- 09021の下4ビットを7SEGに変換→∩0042
 

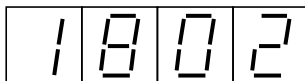
8	g f e d c b a
	0 1 1 1 1 1 1 1
- 09021の上下4ビットを交換
 

09021
1 0 0 0 0 0 0 1
8 1
g f e d c b a
0 0 0 0 0 1 1 0
- 09021の下4ビットを7SEGに変換→∩0043
 

!	g f e d c b a
	0 0 0 0 0 1 1 0

- TMR/CNT番号設定デジタルスイッチで指定したTMR/CNTの外部設定値をデータテーブル(レジスタ39000～39777)から読み出し、7セグメント数字表示器に出力します。

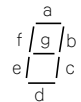
- 上記の例では、TMR002の設定値として、



∩0043 ∩0042 ∩0041 ∩0040

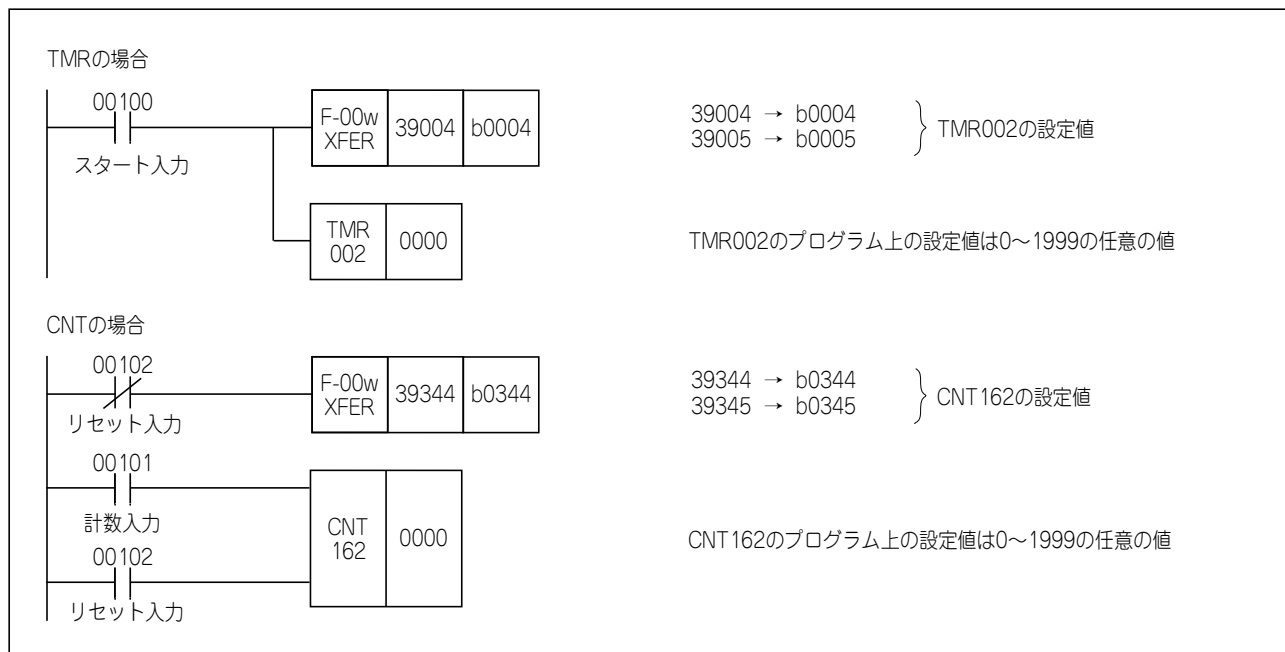
が表示されます。

7セグメント デコーダ表



入力データ	出力データ	表示出力
	g f e d c b a	
00000000	00111111	0
00000001	00000110	1
00000010	01011011	2
00000011	01001111	3
00000100	01100110	4
00000101	01101101	5
00000110	01111101	6
00000111	00100111	7
00001000	01111111	8
00001001	01101111	9
00001010	01110111	A
00001011	01111100	b
00001100	00111001	c
00001101	01011110	d
00001110	01111001	e
00001111	01110001	F

#### (4) TMR/CNTのプログラム



- TMRの場合、スタート入力OFF→ONに変化したとき、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ON→OFFに変化したとき、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リセットしたときは、F-00wの演算条件にご注意ください。(リセット解除でF-00w実行とします。)

**注2** スタート入力OFFのとき(タイマ停止中)又はカウンタのリセット入力ONのとき(カウンタがリセット中)にプログラマ等で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ONのとき(タイマ動作中)、カウンタのリセット入力OFFのとき(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-00wの転送命令を省略します。(一般のTMR/CNTのプログラム)

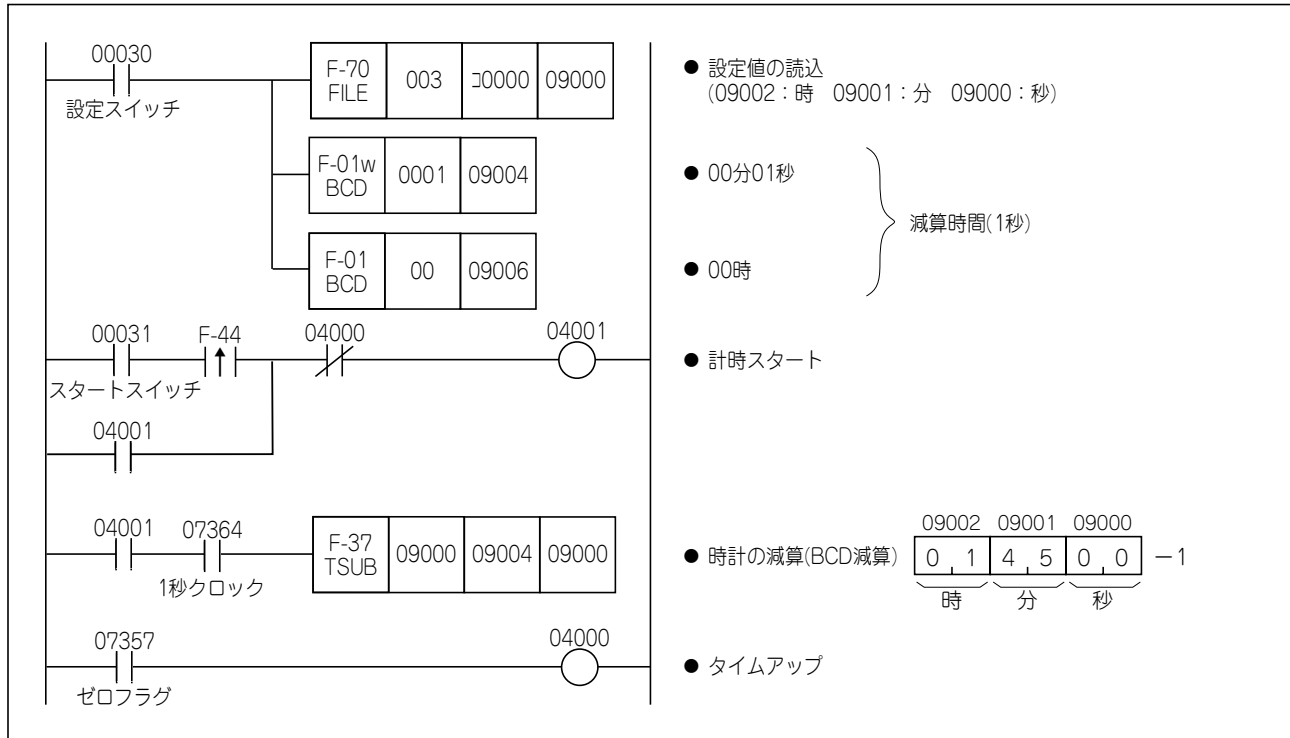
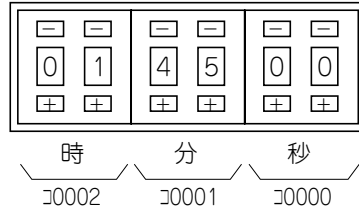
## 2-21 時・分・秒設定減算タイマ

- 通常のタイマ設定値は秒単位で設定しますが、本例では設定値を時・分・秒で外部設定式にすることができます。

適応PC

JW20H(JW-22CU)  
 JW30H(JW-32CUH/H1, JW-33CUH/H1/H2/H3)  
 JW50H/70H/100H

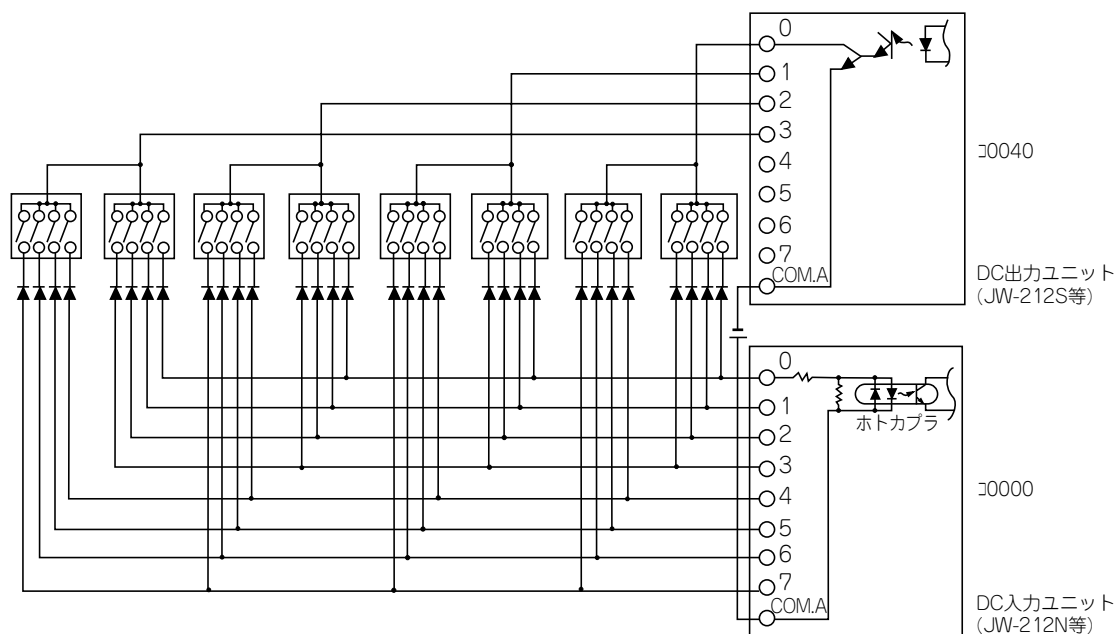
タイマ設定用デジタルスイッチ



## 2-22 ダイナミック入力

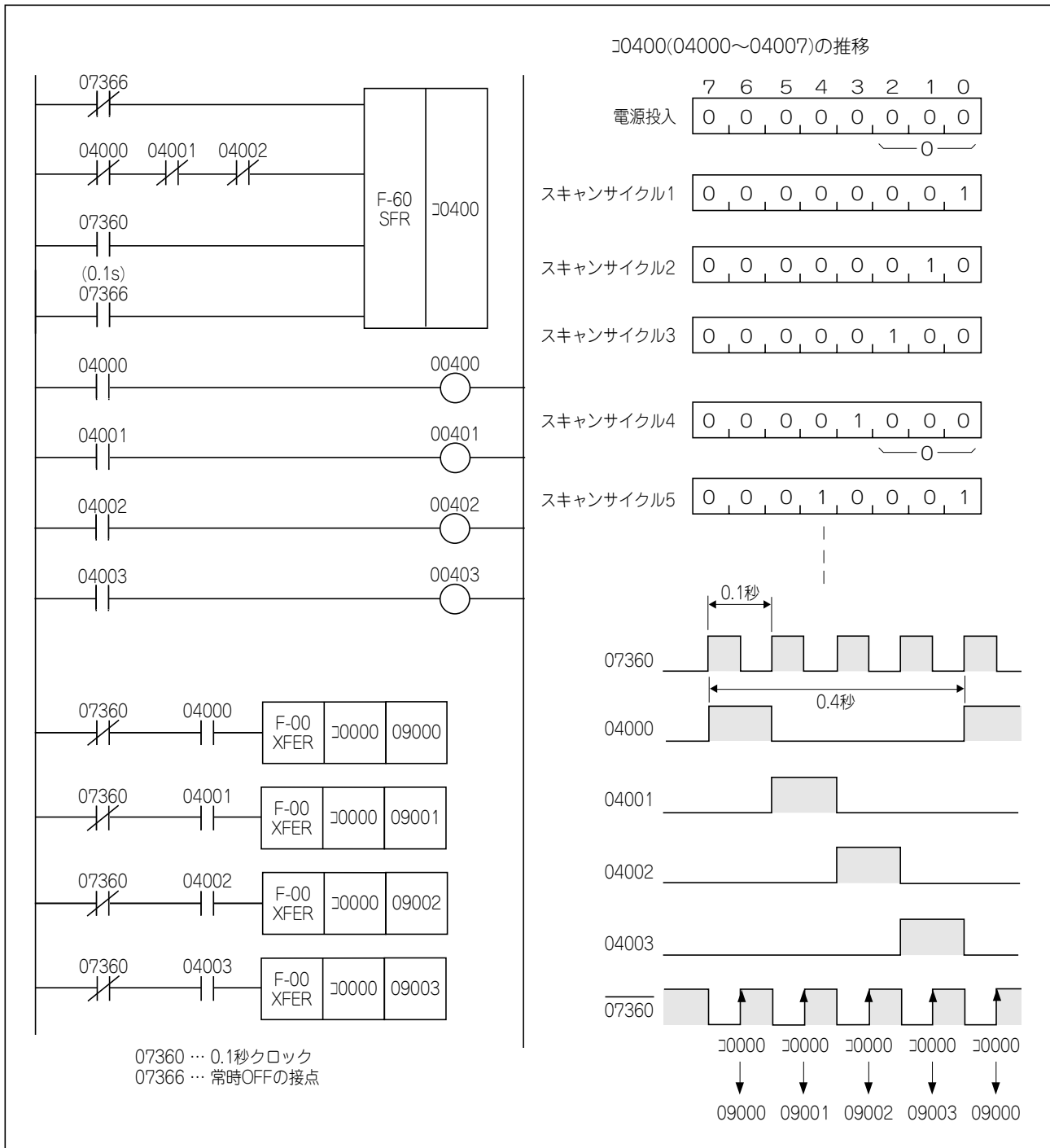
- 多桁の数値信号を2桁ずつ、ダイナミックに読み込みます。

### (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



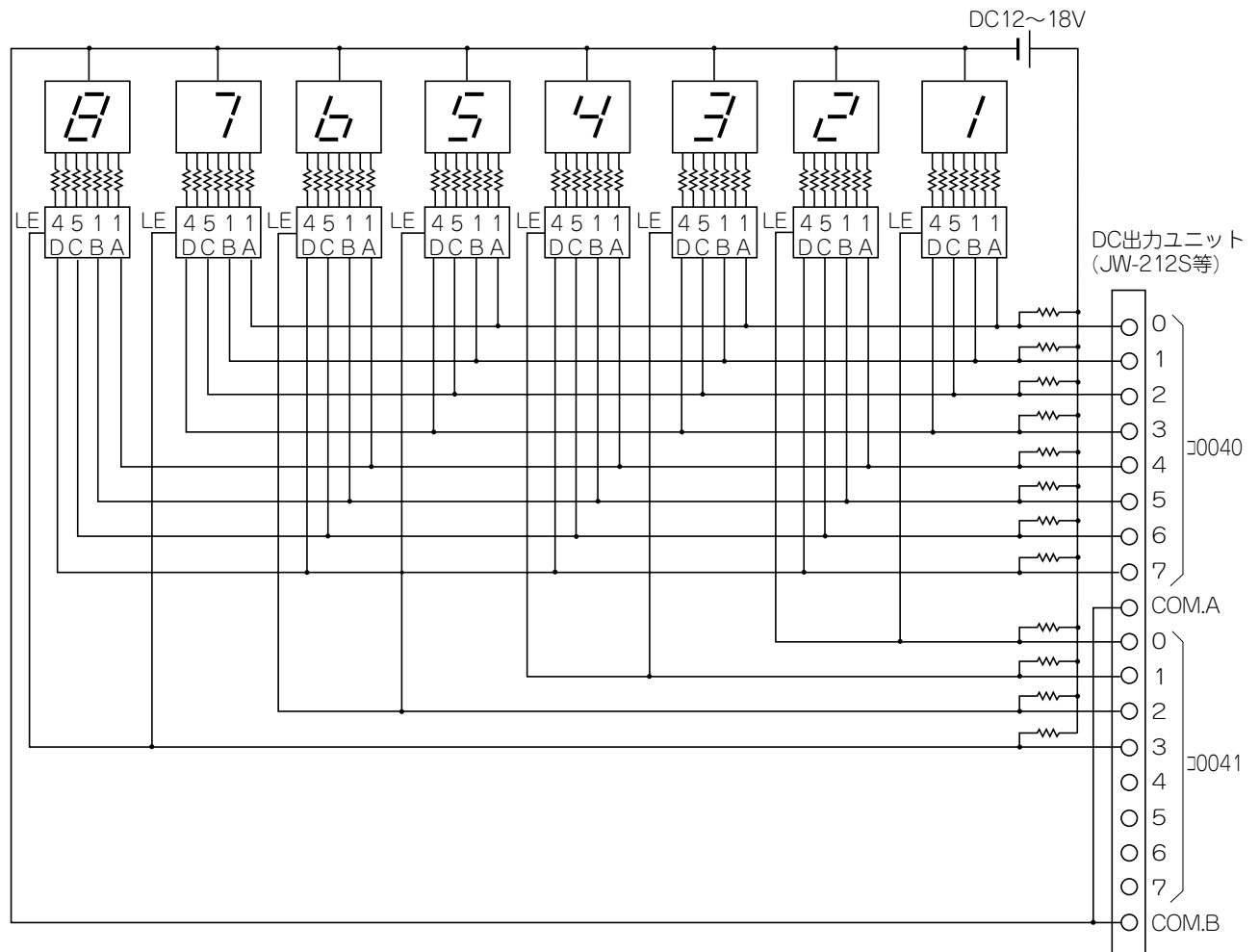
- 0.1秒クロック(07360)がOFF→ONに変化するごとに出力00400~00403のいずれか1ビットを順次ONします。
- 0.1秒クロックがON→OFFに変化するとき、00000のBCD2桁の数値をレジスタ09000~09003に順次格納します。
- 8桁の読み込みに400msを要します。



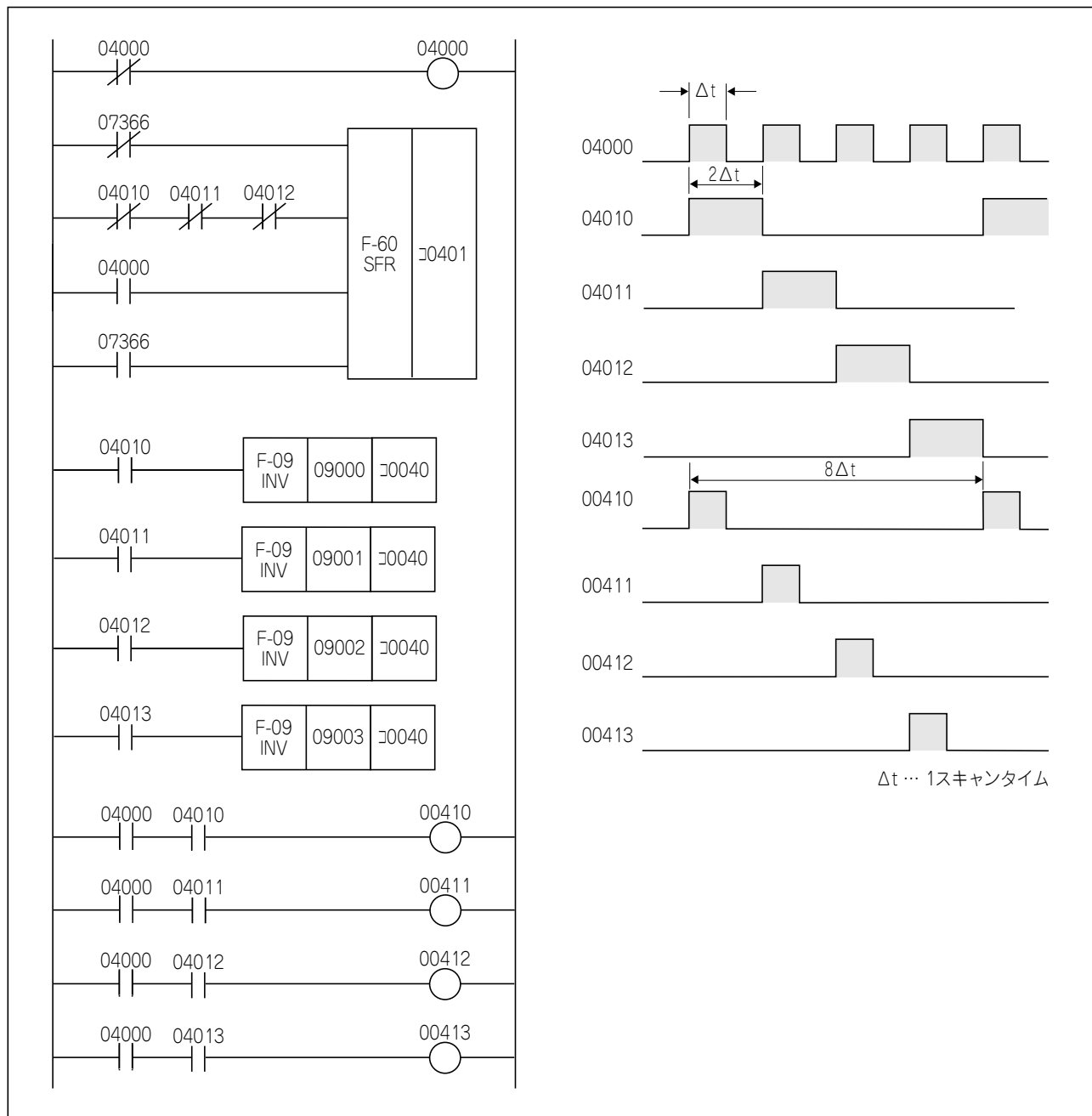
## 2 - 23 ダイナミック出力

- 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

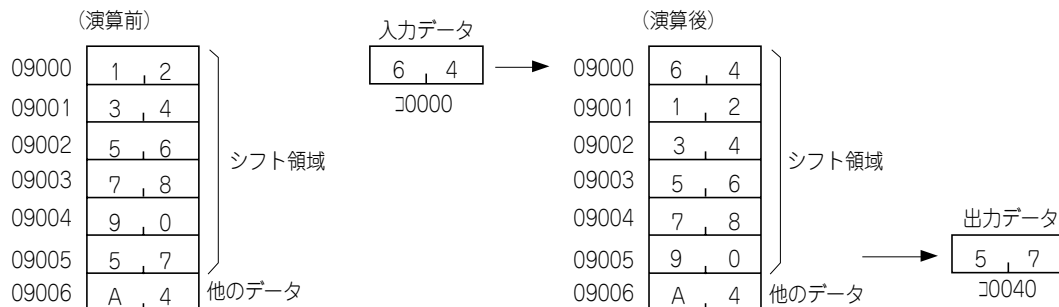


- $2\Delta t$ 毎に0040への出力データが09000→09001→09002→09003→09000と推移します。これに同期してストローブ信号(00410~00413)が出力されます。
- 8桁分出力するのに $8\Delta t$ の時間を要します。  
(1スキャンタイム( $\Delta t$ )が5msの場合で40ms)

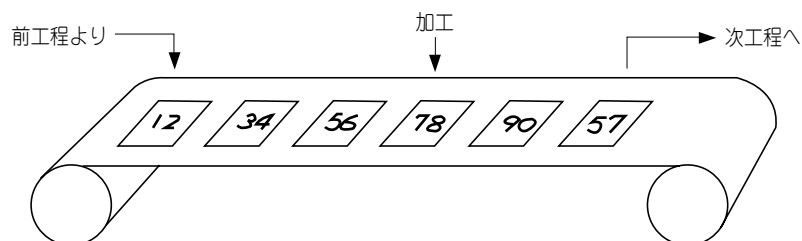
- 表示回路は正論理で動作するため、F-09命令でデータの論理を反転しています。

## 2 - 24 同期型 FIFO スタックレジスタ

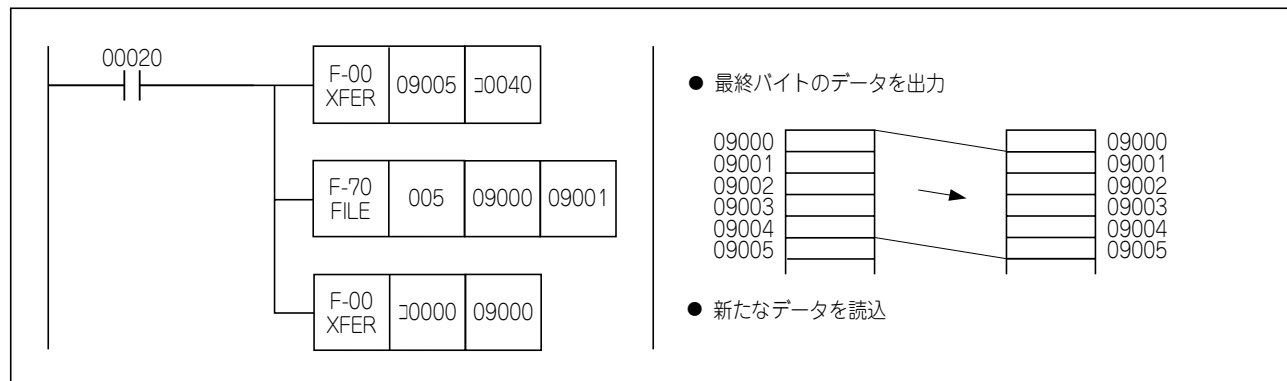
- 任意バイト数(最大256バイト)のシフトレジスタを構成します。



- 09000~09005には常に最新のデータが格納されます。

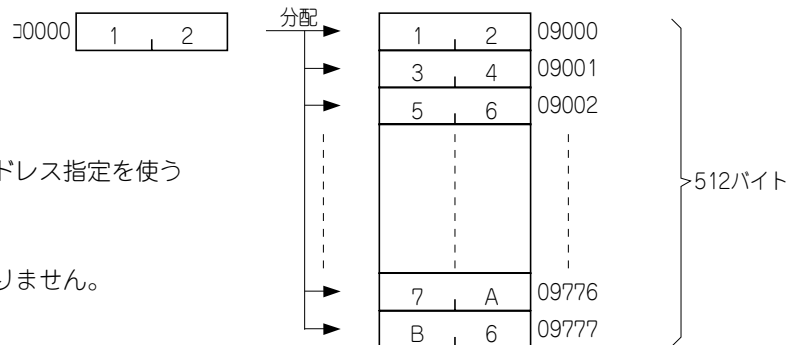


前工程から機種コードを受け、加工後、次の工程へ機種コードを伝えます。



## 2 - 25 データの分配 (データテーブルへの格納)

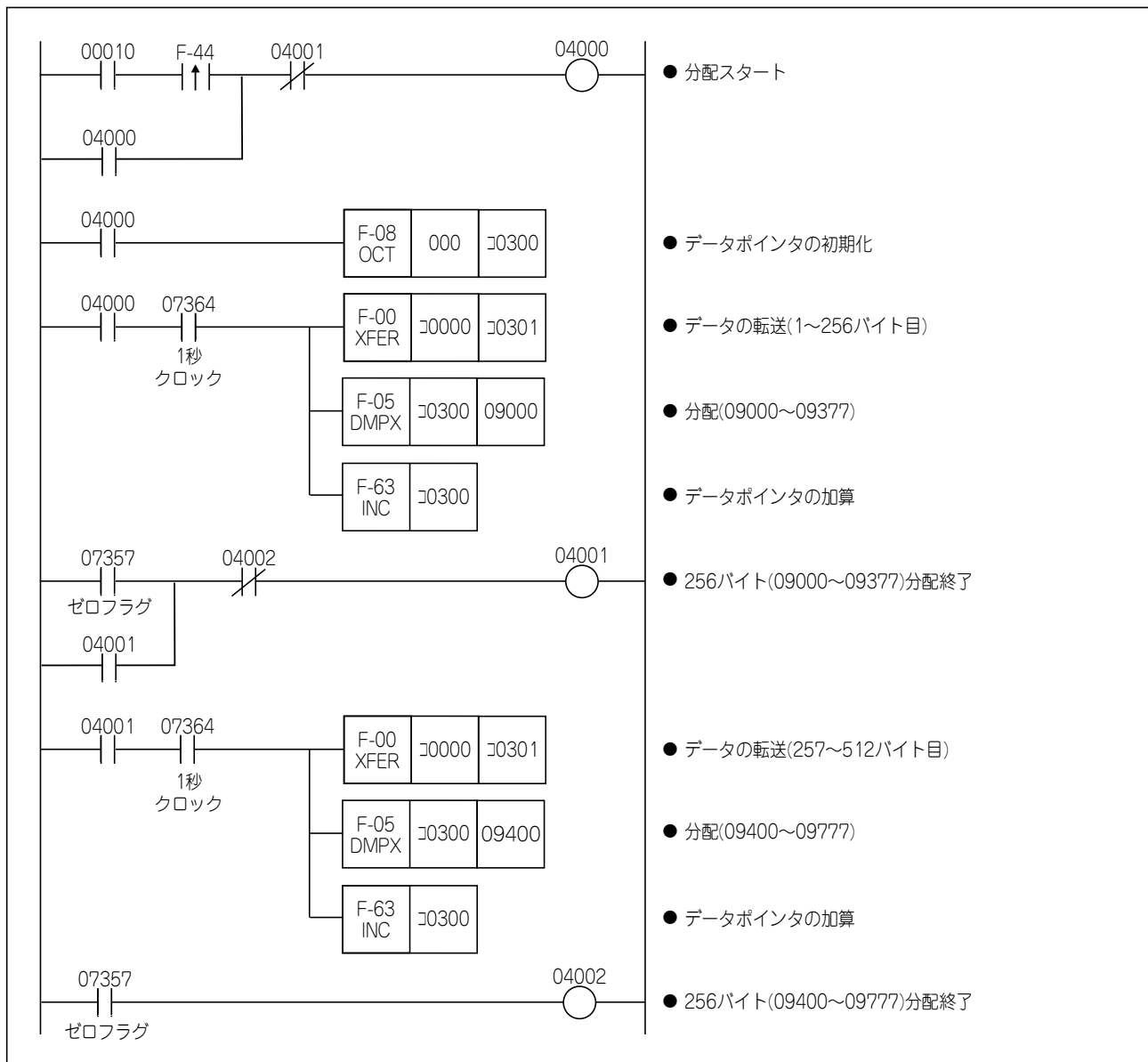
- 1秒毎に変化するレジスタ30000の内容をレジスタ09000~09777に分配します。



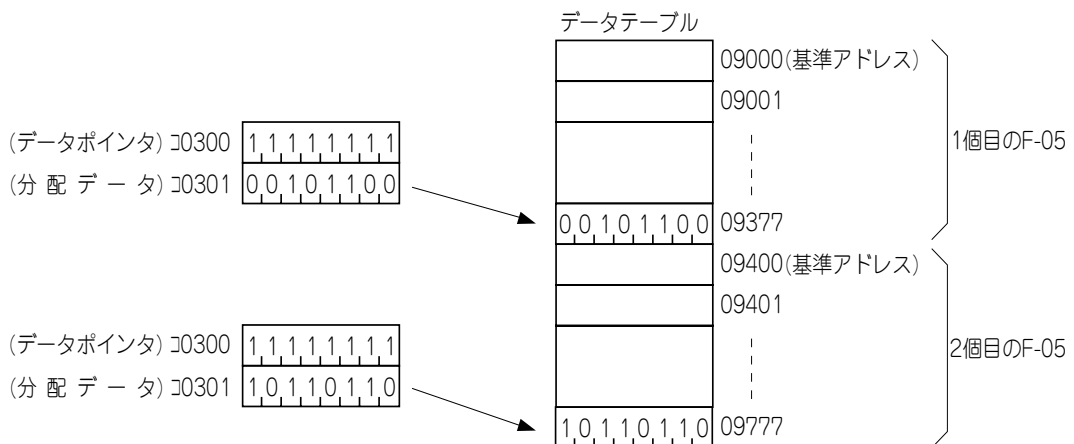
- 分配命令 (F-05) を使う方法と間接アドレス指定を使う方法があります。

**注1** JW10には分配命令 (F-05) はありません。

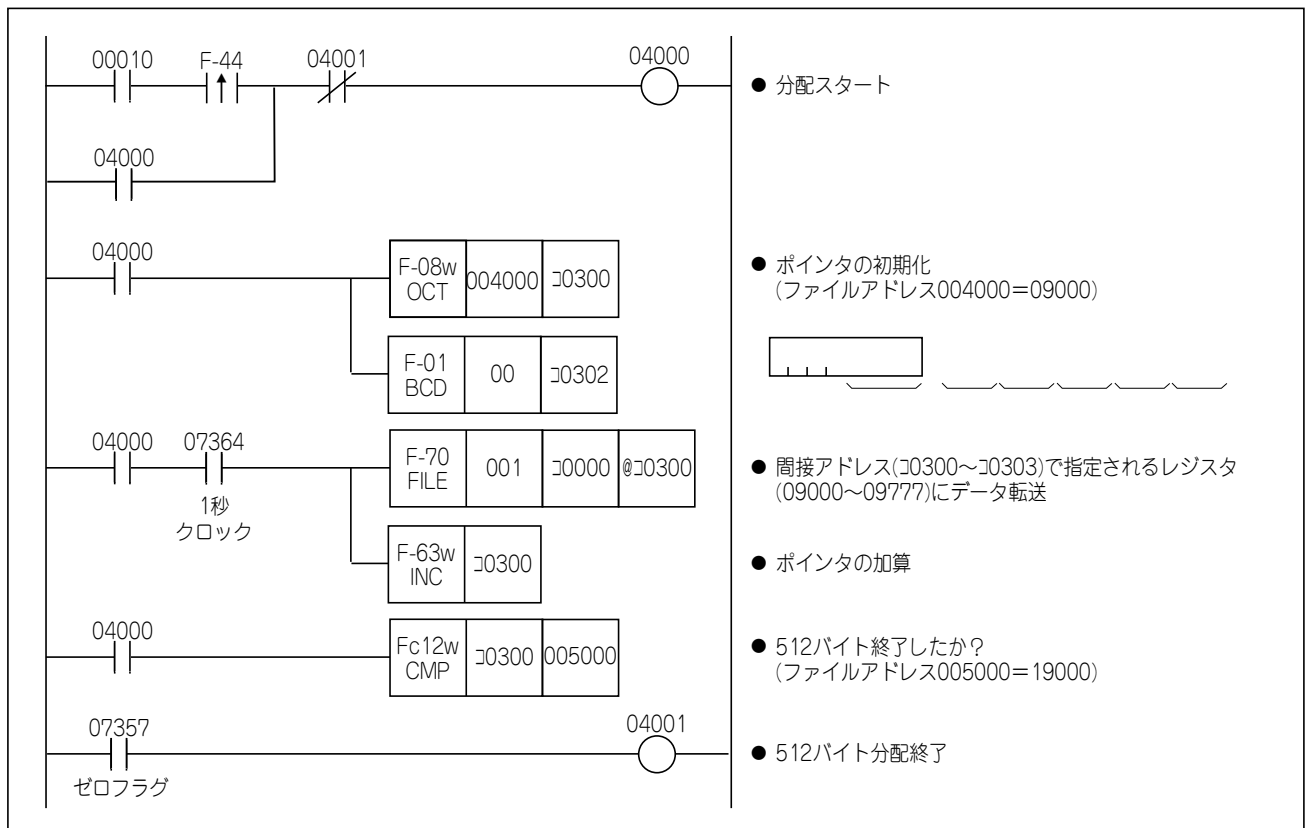
### (1) 分配命令 (F-05) を使用したプログラム



- 分配命令(F-05)は256バイト毎に基準アドレスが決められているため、512バイトの分配ではF-05を2個使用します。



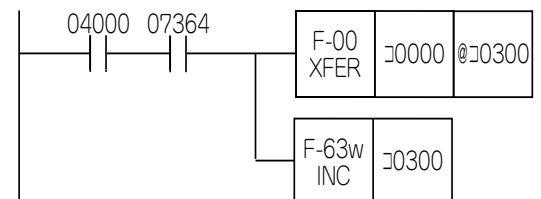
(2) 間接アドレスを使用したプログラム



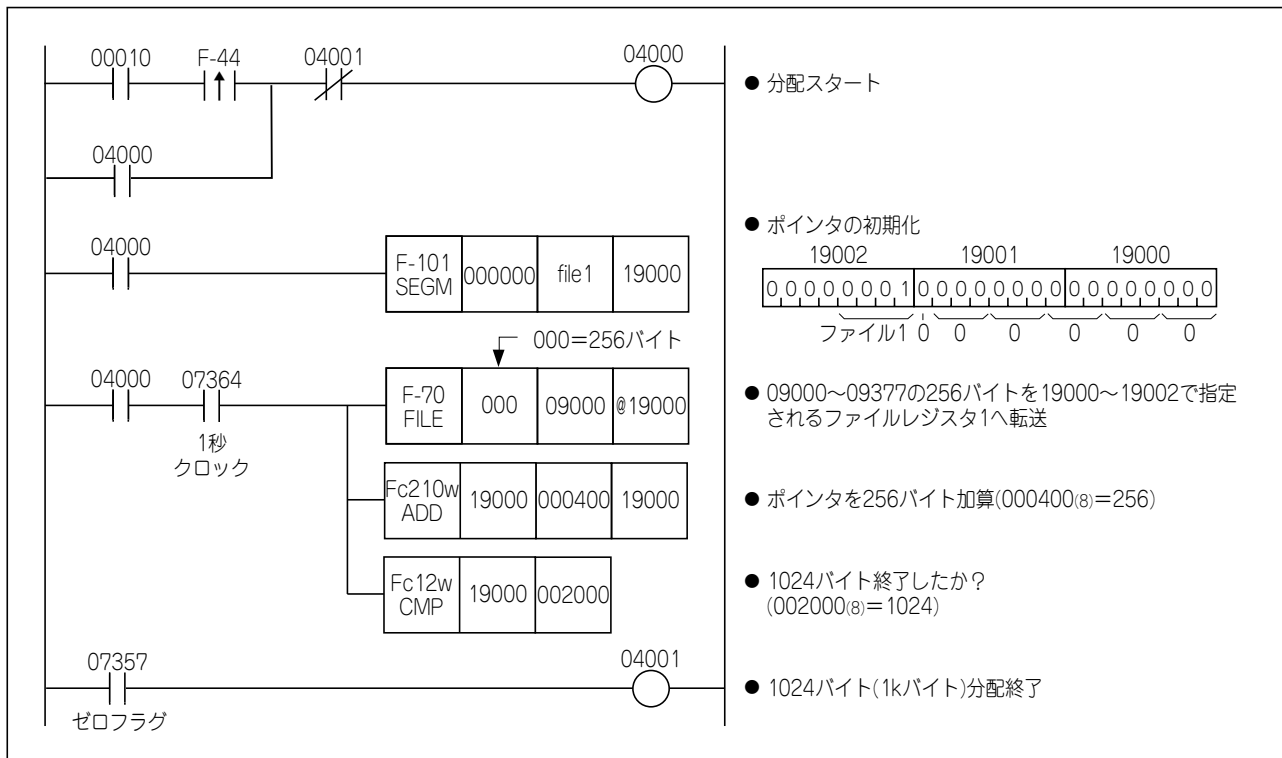
注1 JW30H, JW50H/70H/100Hには、間接アドレスの設定命令(F-100)があります。



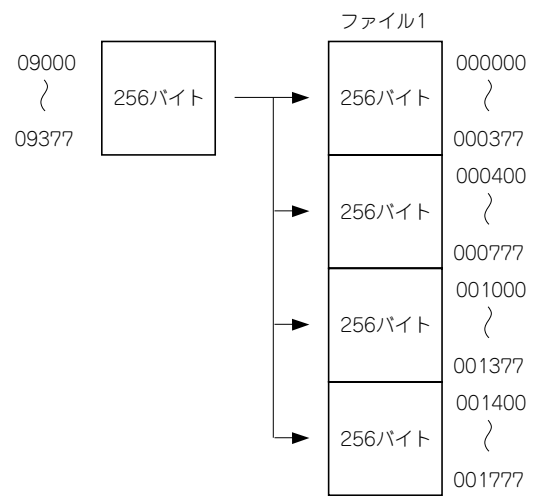
注2 JW30H, JW50H/70H/100Hは、F-00(レジスタ間の1バイト転送)も間接アドレス指定できます。



**参考** ファイルレジスタをデータテーブルとして使用すると、大量のデータを格納できます。  
 [JW30H (JW-32CUH/H1, JW-33CUH/H1/H2/H3), JW50H/70H/100Hのみ]

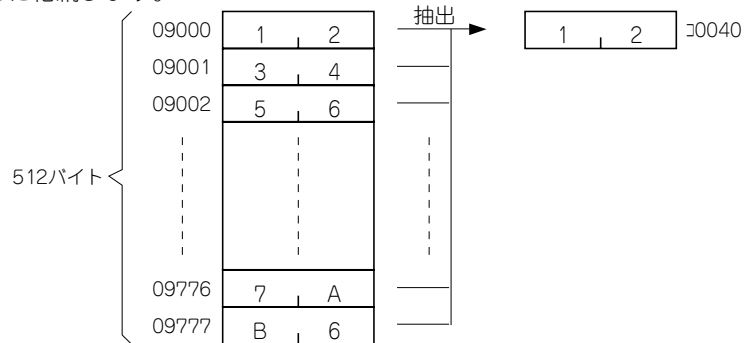


● 上記プログラムでは、1秒毎に変化するレジスタ09000~09377の256バイトの内容をファイルレジスタ1の000000~001777の1kバイトに分配しています。



## 2 - 26 データの抽出(データテーブルからの取り出し)

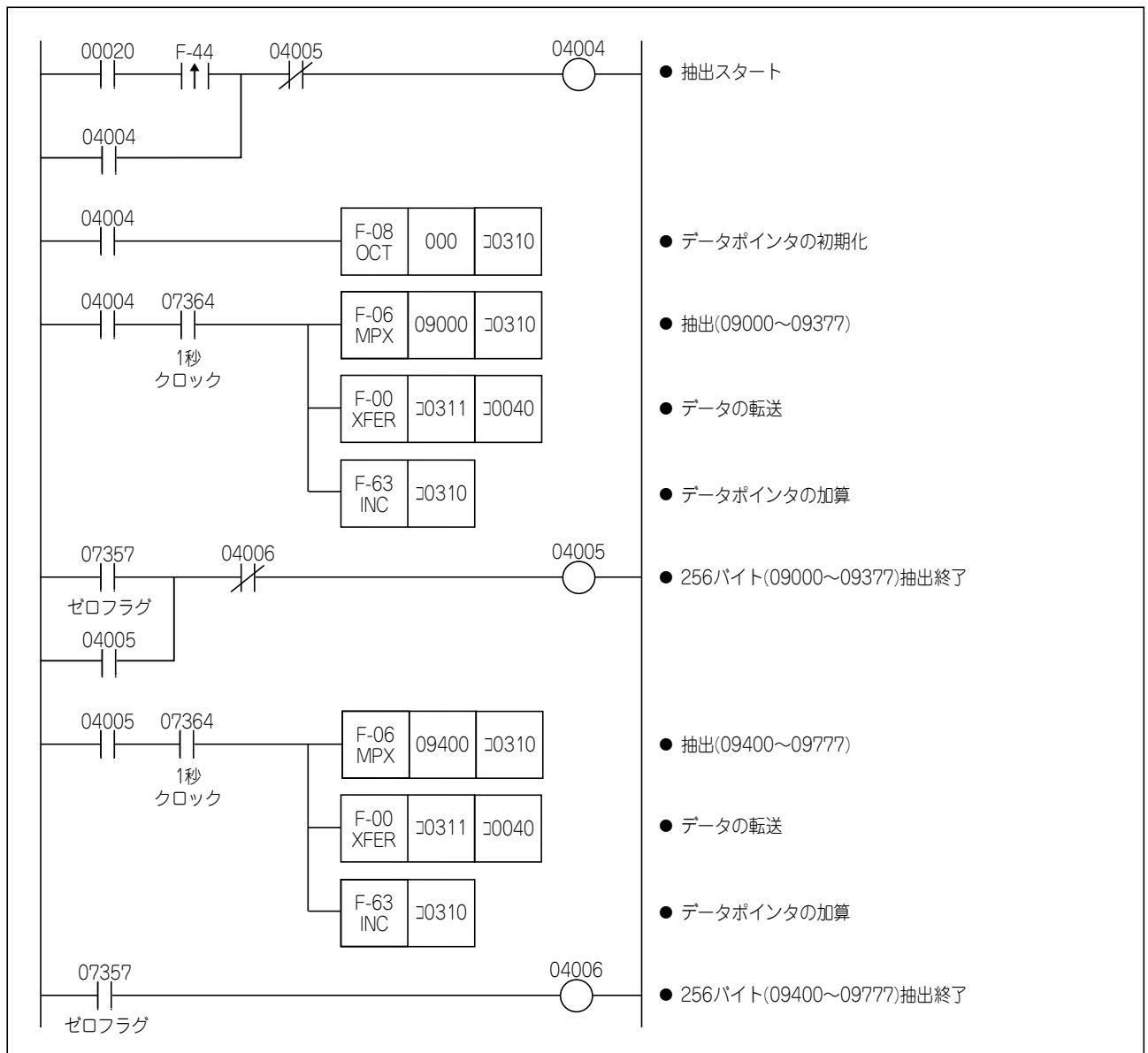
- レジスタ09000~09777に格納されたデータを1秒毎に順番に取り出し、レジスタ00040に格納します。



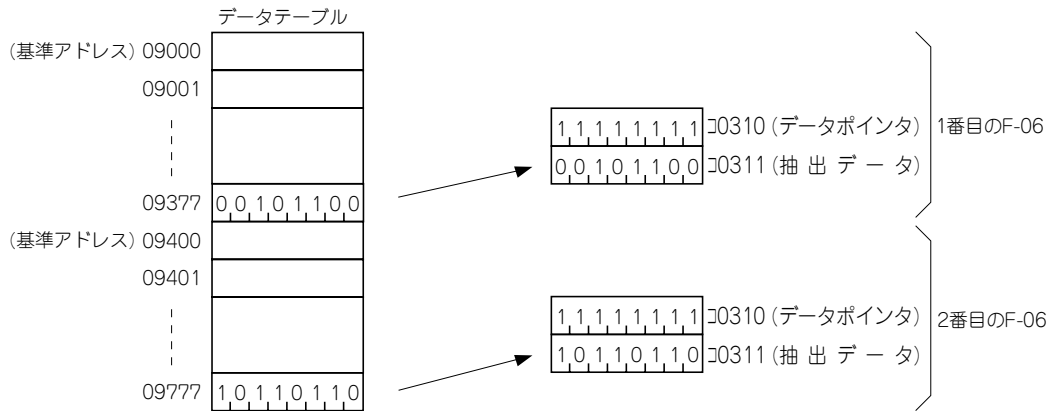
- 抽出命令(F-06)を使う方法と間接アドレス指定を使う方法があります。

注1 JW10には抽出命令(F-06)はありません。

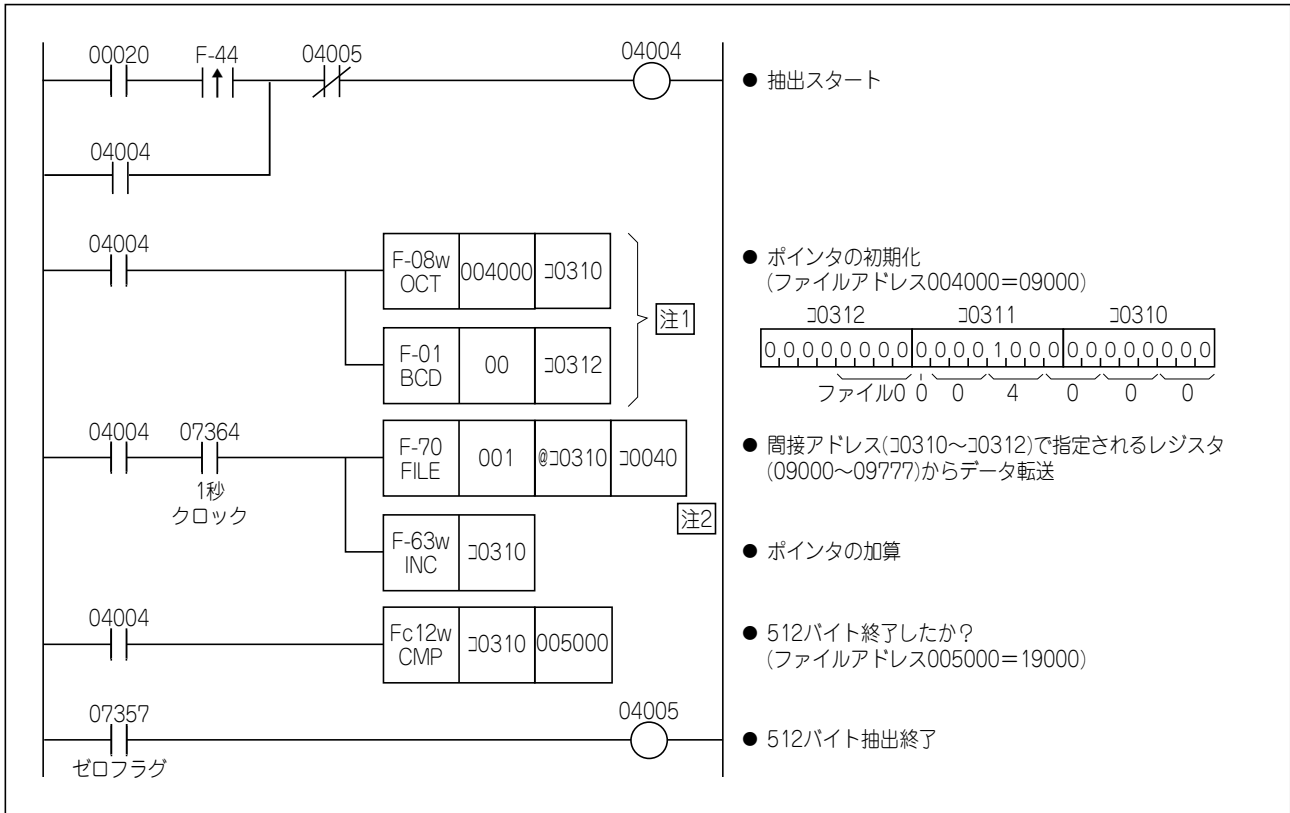
### (1) 抽出命令(F-06)を使用したプログラム



- 抽出命令 (F-06) は256バイト毎に基準アドレスが決められているため、512バイトの抽出ではF-06を2個使用します。



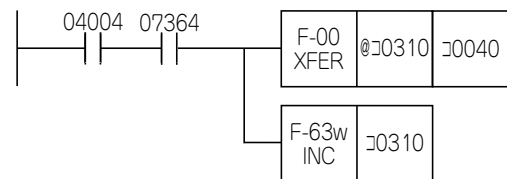
(2) 間接アドレスを使用したプログラム



注1 JW30H, JW50H/70H/100Hには間接アドレスの設定命令 (F-100) があります。

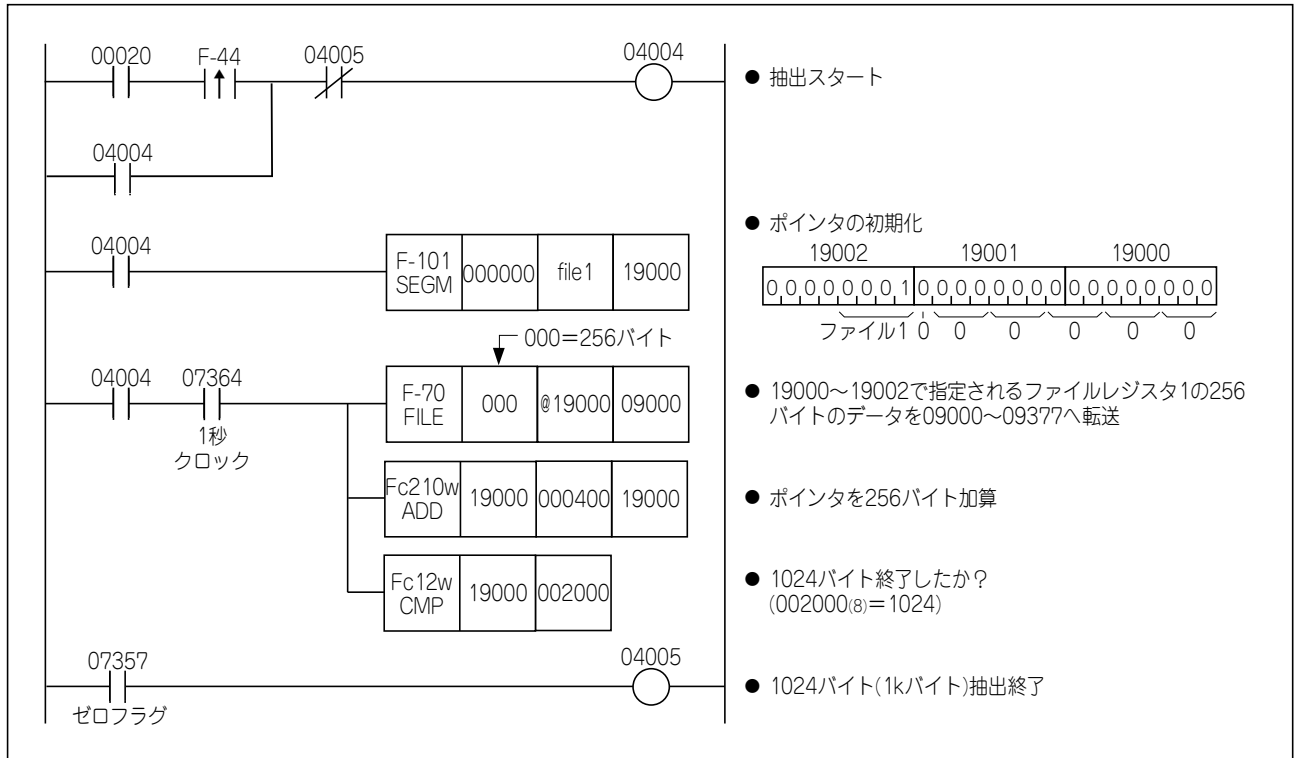


注2 JW30H, JW50H/70H/100Hは、F-00(レジスタ間の1バイト転送)も間接アドレス指定できます。

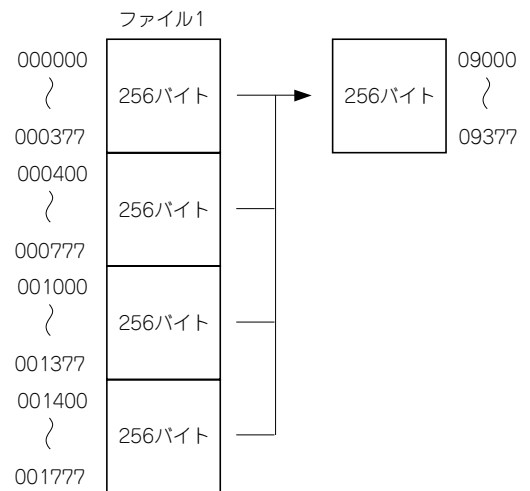




**参考** ファイルレジスタをデータテーブルとして使用すると大量のデータを格納できます。  
 [JW30H(JW-32CUH/H1, JW-33CUH/H1/H2/H3), JW50H/70H/100Hのみ]

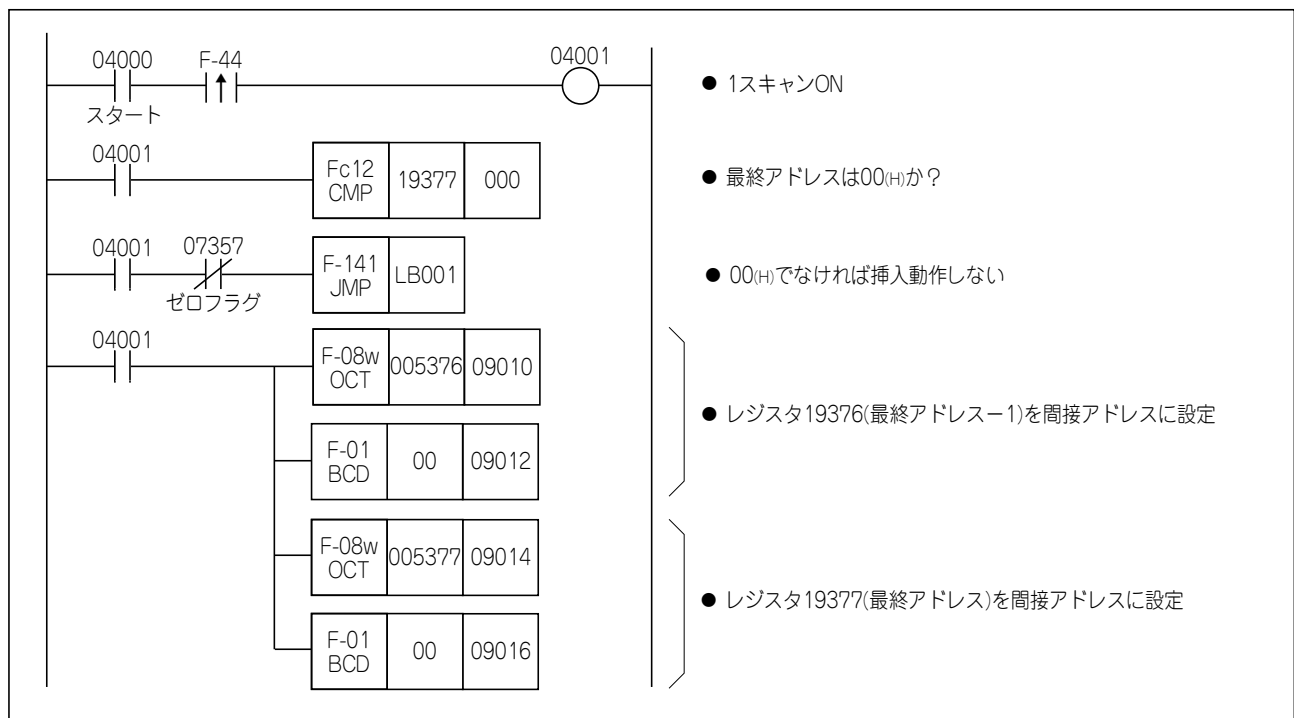
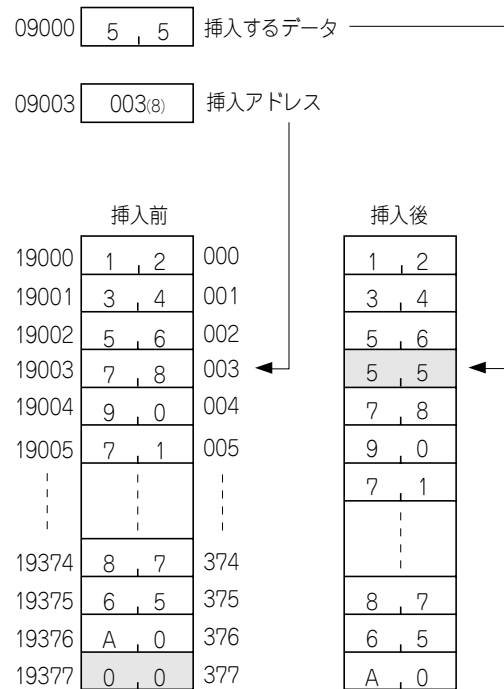


● 上記プログラムは、ファイルレジスタ1の0000000~001777の1kバイトに格納されているデータを256バイト単位で1秒毎にレジスタ09000~09377に抽出しています。

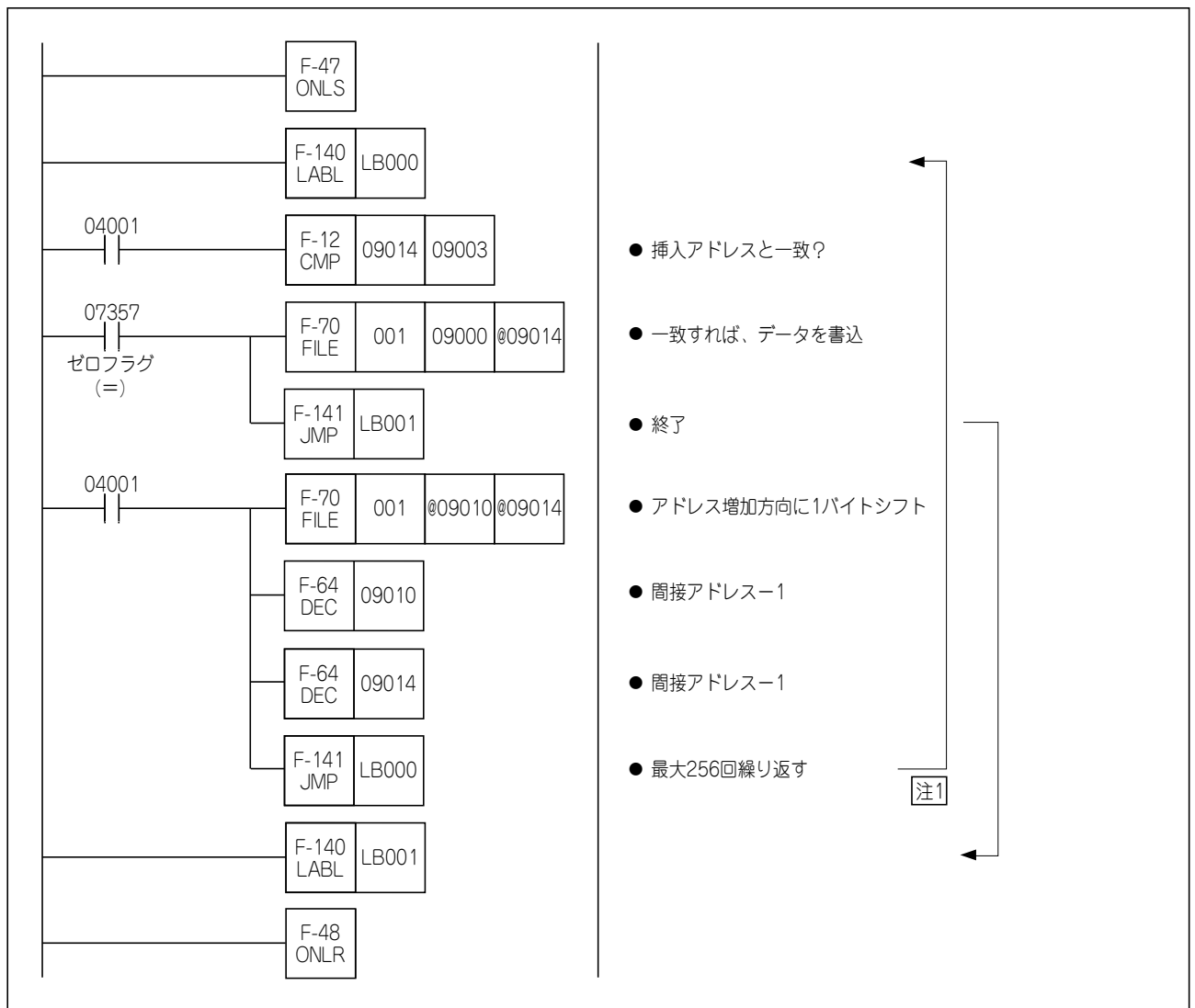


## 2 - 27 データの挿入

- 256個の1バイトデータを持つシフトレジスタにおいて、指定データを指定アドレスに挿入します。
- 挿入は最終アドレスの値が0のときのみ行います。

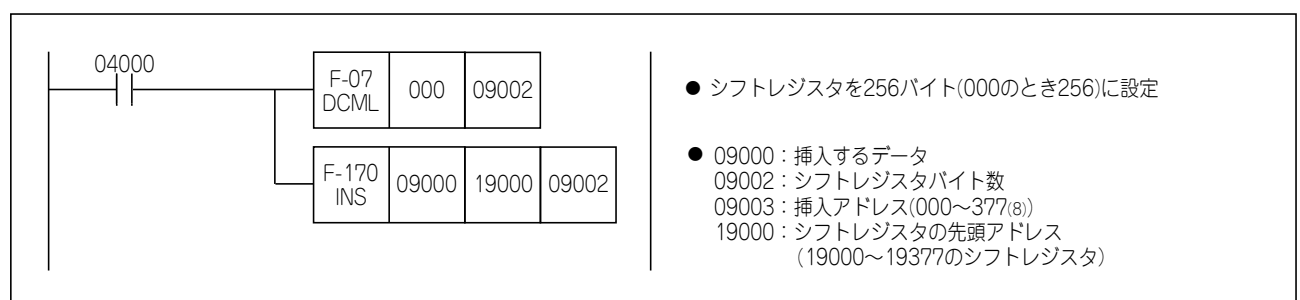


(次ページへ続く)



**注1** スキャンタイムにご注意ください。  
 (最大256回、LB000～LB001を繰り返すため)  
 ・ P C 機 種 : JW10(JW-1424K/1624K)  
 ・ 挿入アドレス : 000(09003=000<sup>(8)</sup>)  
 の場合、スキャンタイム≒71ms…最大  
 (最小6ms)

**参考** JW30H(JW-32CUH/H1, JW-33CUH/H1/H2/H3), JW50H/70H/100Hには、データの挿入命令(F-170)があります。



- シフトレジスタを256バイト(000のとき256)に設定
- 09000 : 挿入するデータ  
 09002 : シフトレジスタバイト数  
 09003 : 挿入アドレス(000～377<sup>(8)</sup>)  
 19000 : シフトレジスタの先頭アドレス  
 (19000～19377のシフトレジスタ)

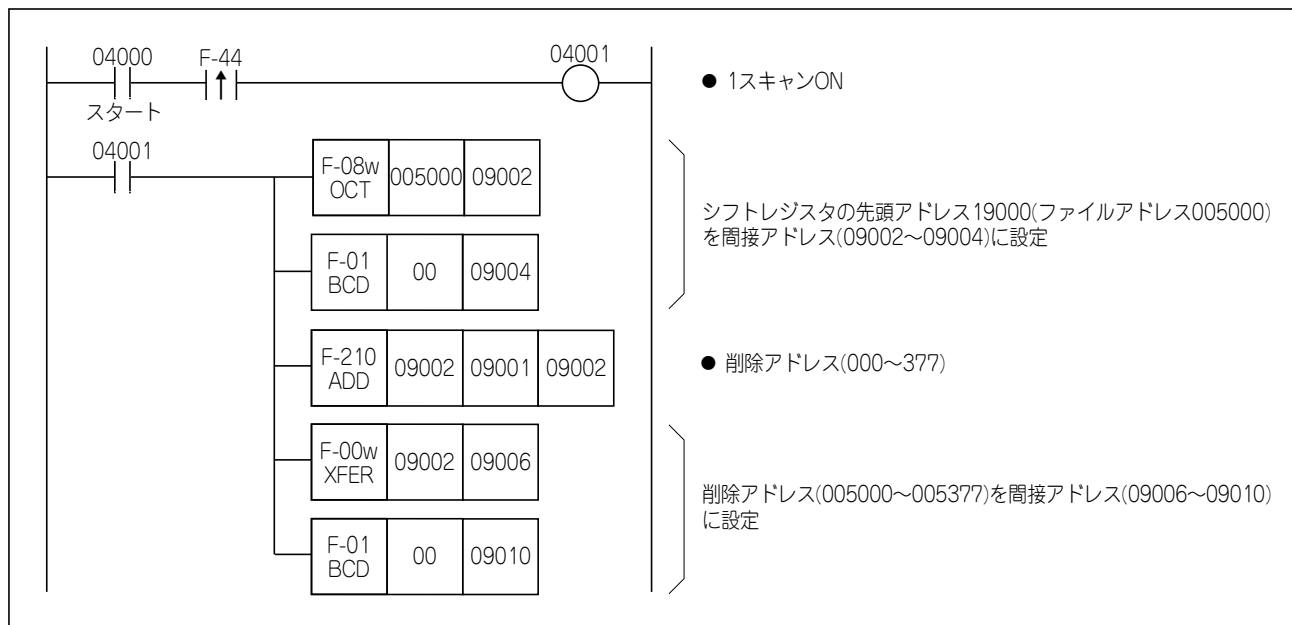
## 2 - 28 データの削除

- 256個の1バイトデータを持つシフトレジスタにおいて、指定アドレスのデータを削除します。
- 削除後、最終アドレスに0を書き込みます。

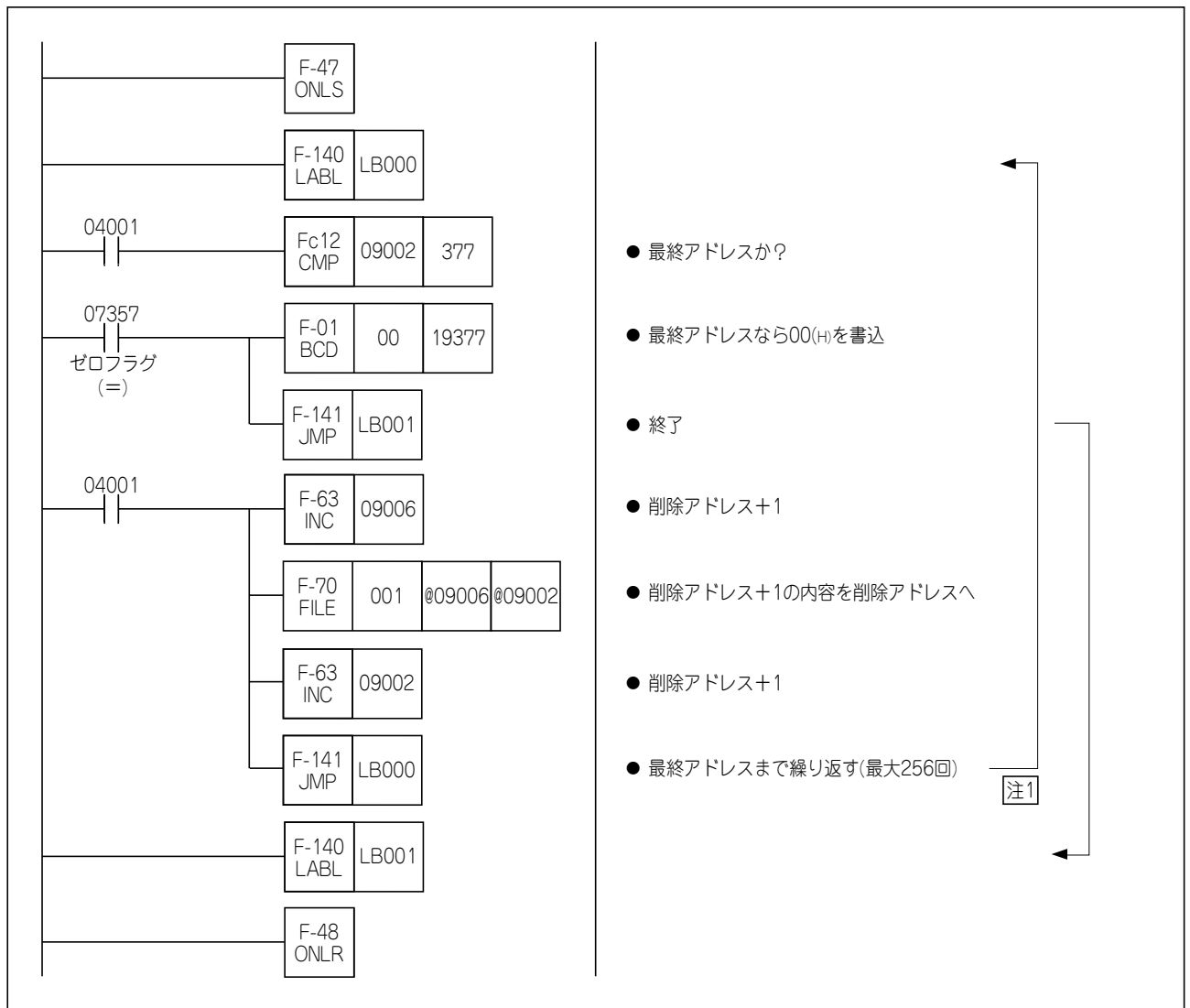
09001	003(8)	削除アドレス
-------	--------	--------

	削除前		削除後
19000	1 , 2	000	1 , 2
19001	3 , 4	001	3 , 4
19002	5 , 6	002	5 , 6
19003	5 , 5	003 ←	7 , 8
19004	7 , 8	004	9 , 0
19005	9 , 0	005	7 , 1
19006	7 , 1	006	⋮
⋮	⋮	⋮	⋮
19375	8 , 7	375	8 , 7
19376	6 , 5	376	6 , 5
19377	A , 0	377	0 , 0

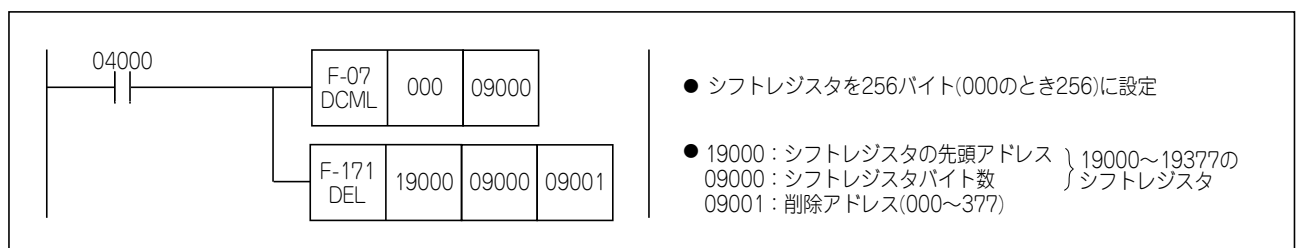


(次ページへ続く)



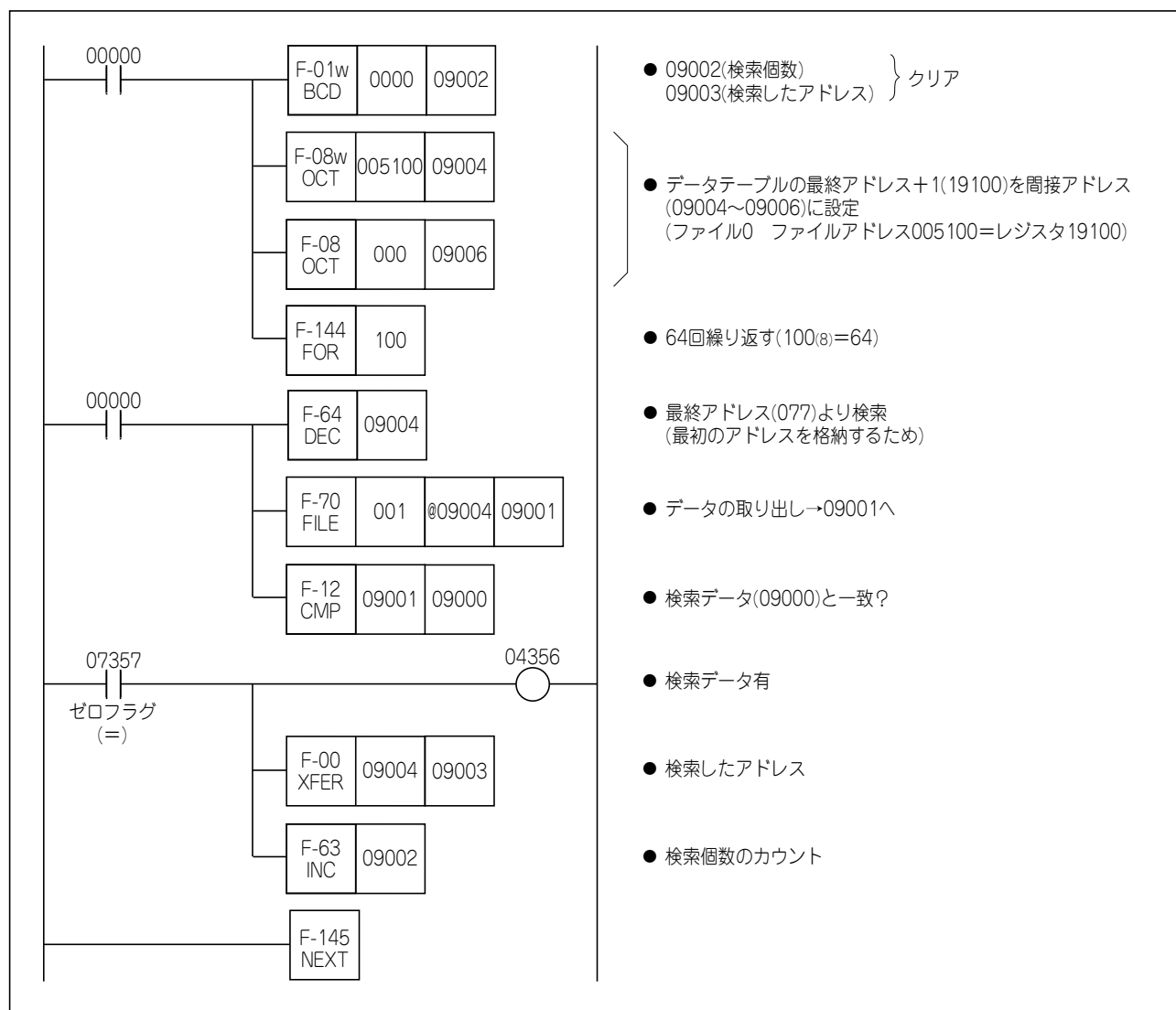
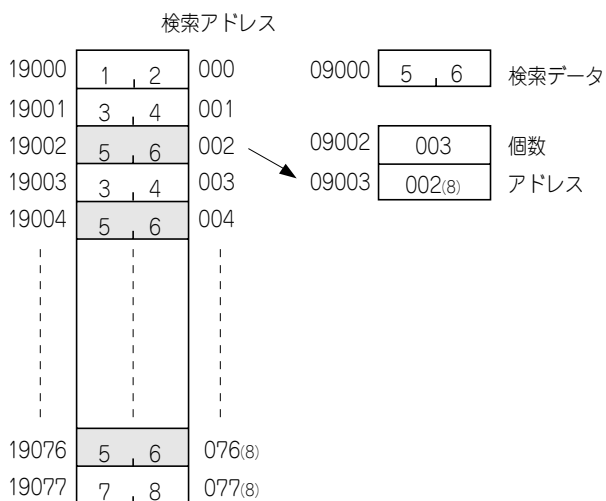
**注1** スキャンタイムにご注意ください。  
 (最大256回、LB000～LB001を繰り返すため)  
 ・ P C 機 種 : JW10(JW-1424K/1624K)  
 ・ 削除アドレス : 000(09001=000<sup>(8)</sup>)  
 の場合、スキャンタイム≒71ms…最大  
 (最小6ms)

**参考** JW30H(JW-32CUH/H1, JW-33CUH/H1/H2/H3), JW50H/70H/100Hには、データの削除命令 (F-171)があります。

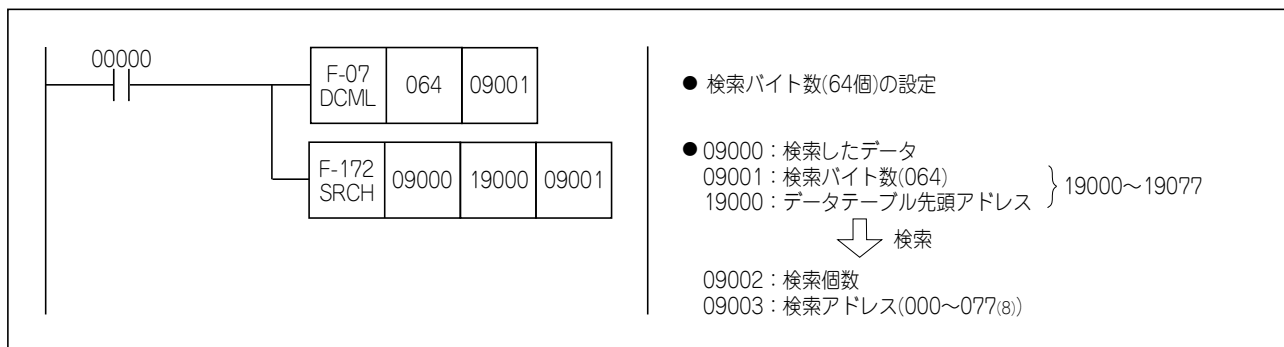


## 2 - 29 データの検索 (1)

- データテーブル(64個)に格納された1バイトデータの中から指定のデータを検索し、検索個数と検索した最初のアドレス(000~077<sup>(8)</sup>)を格納します。

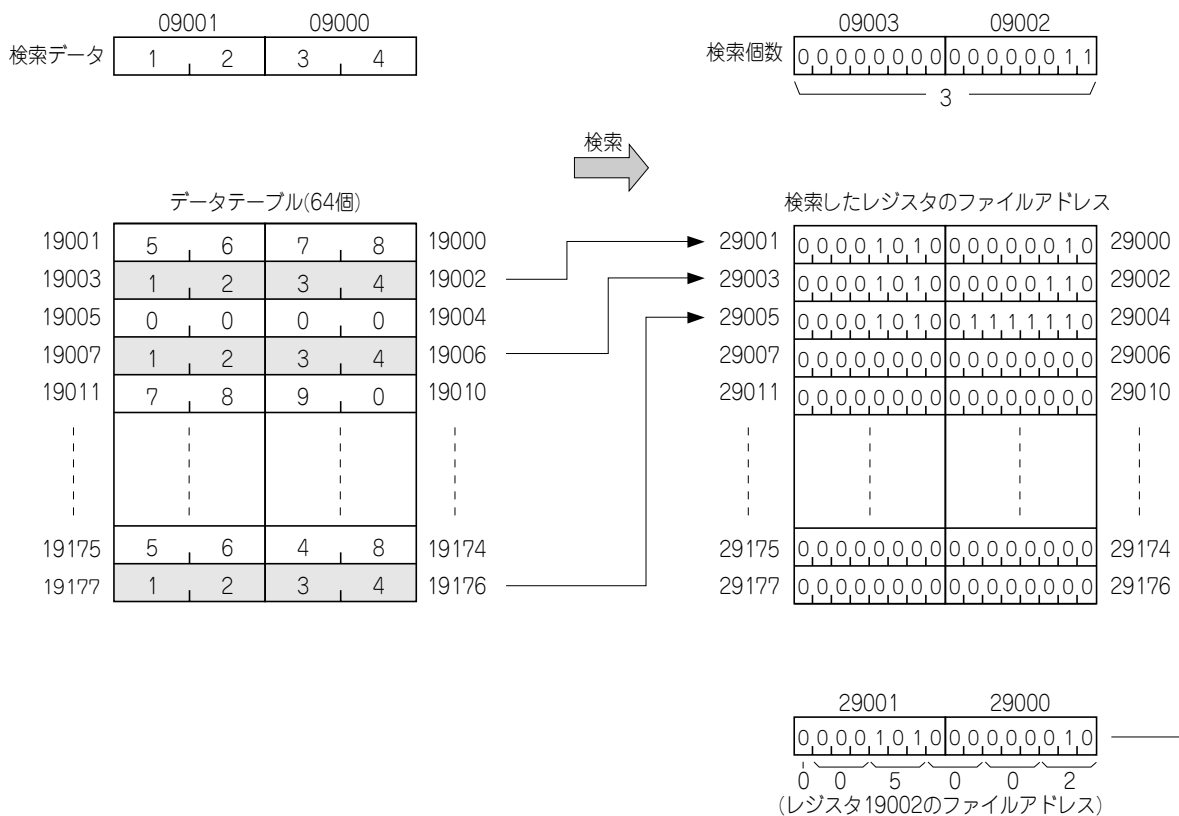


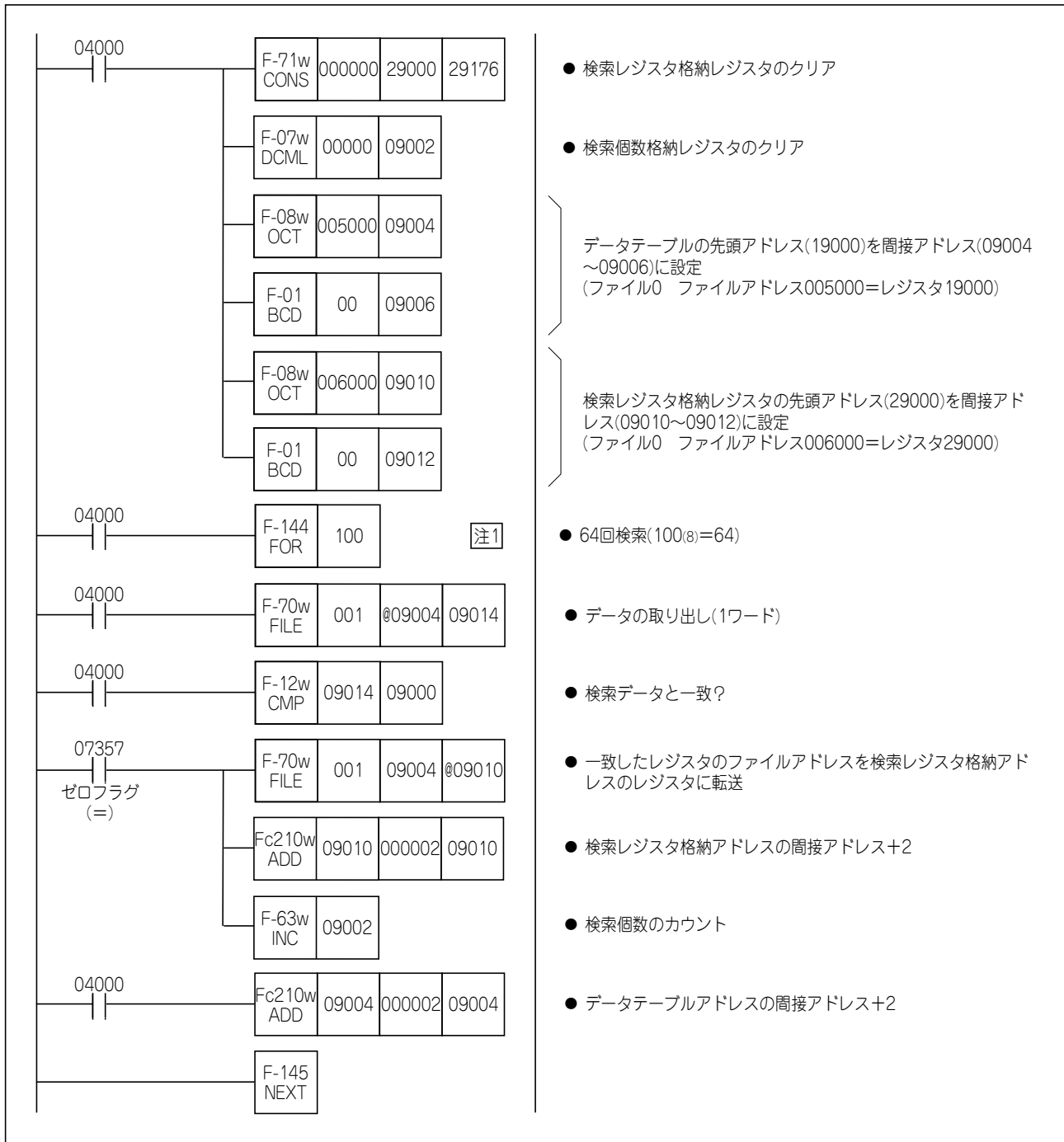
**参考** JW30H(JW-32CUH/H1, JW-33CUH/H1/H2/H3), JW50H/70H/100Hには、データの検索命令(F-172)があります。



## 2 - 30 データの検索 (2)

● データテーブル(64個)に格納された1ワードデータを検索し、検索個数および検索したレジスタのファイルアドレスを順番にレジスタに格納します。



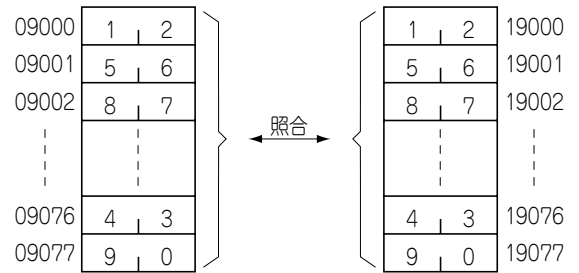


**注1** F-144 (FOR) の定数を000にすると最大256個のデータテーブルの検索ができます。ただし、スキャンタイムにご注意ください。

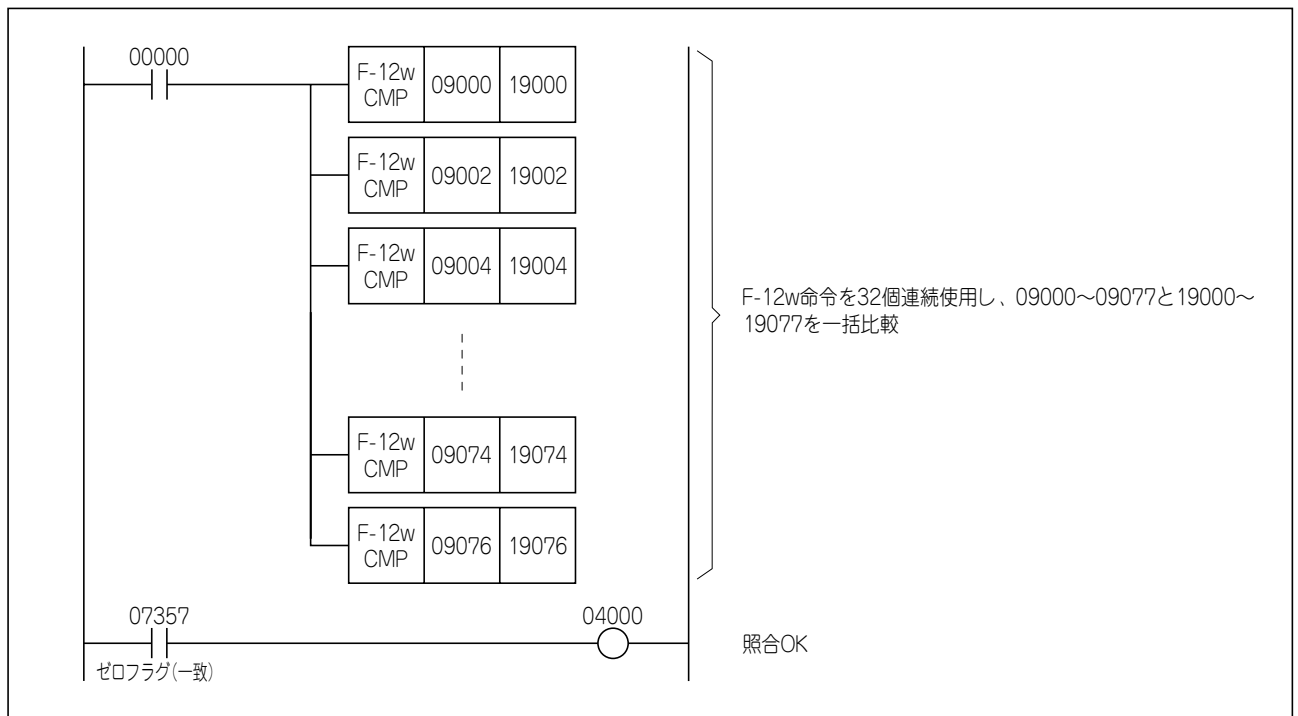


## 2-31 データの照合

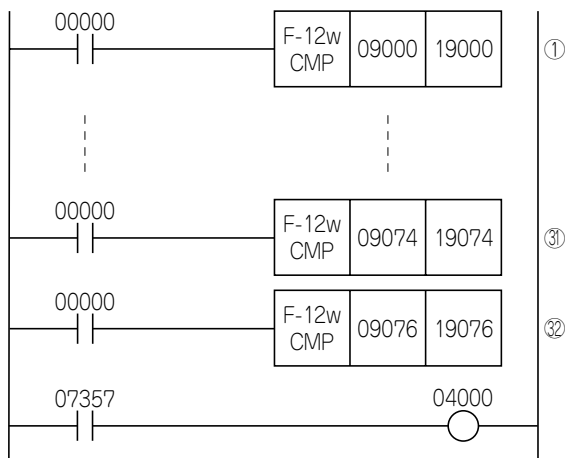
- レジスタ09000~09077とレジスタ19000~19077の各々64バイト(32ワード)のデータの照合を行います。



(1) 比較命令を連続使用する場合(倍長演算)

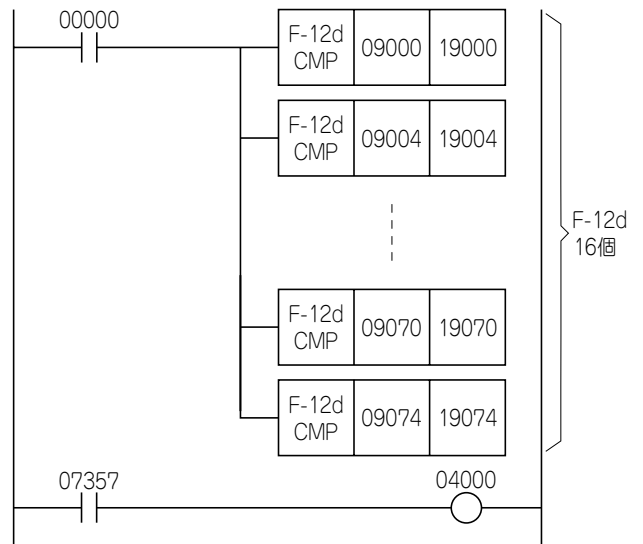


**注1** 入力条件が共通でないと倍長演算になりません。

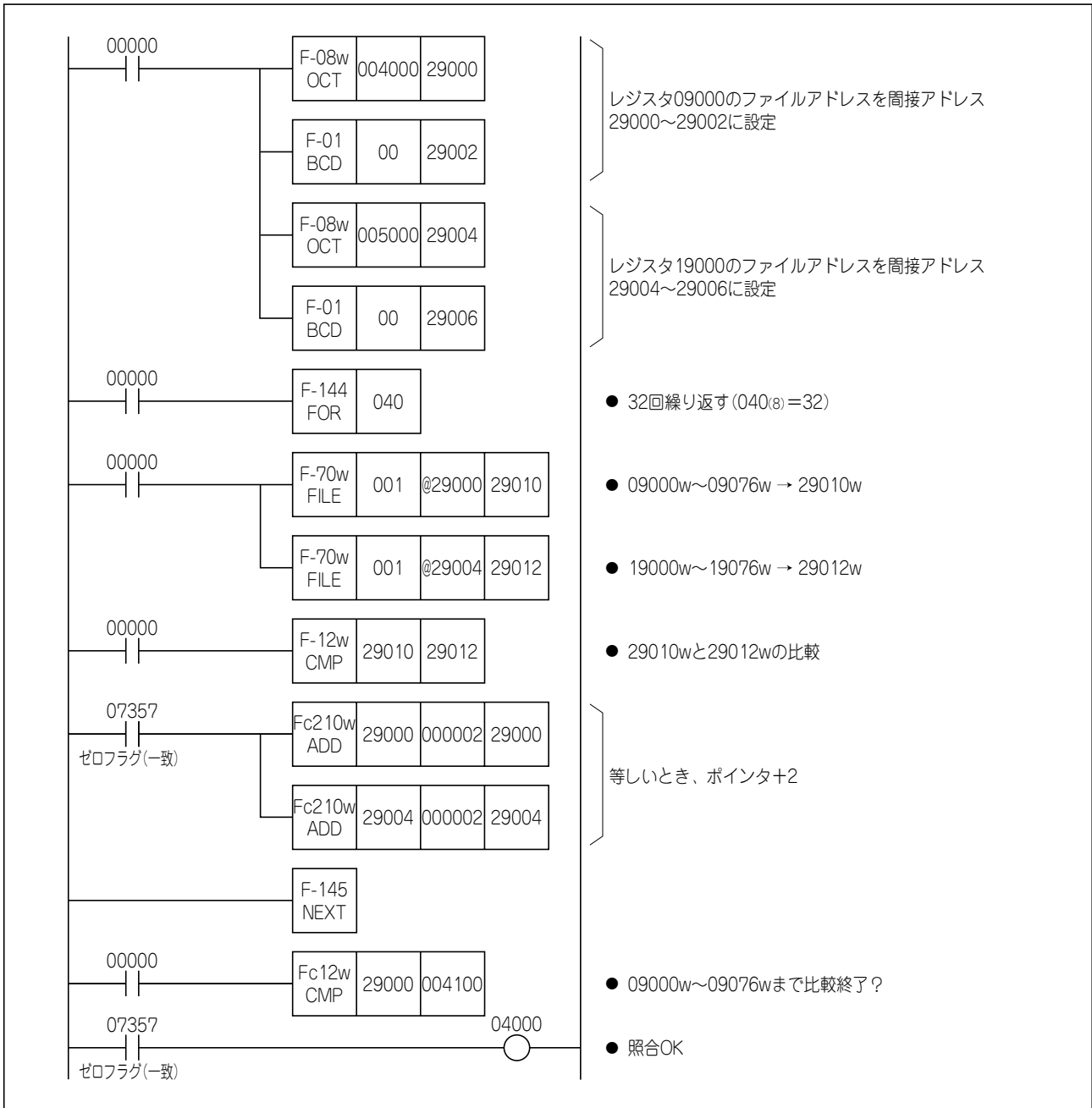


上記例の場合、09076wと19076wが等しいとき、04000はONします。(②の演算結果を反映し、①~③の演算結果は反映されない)

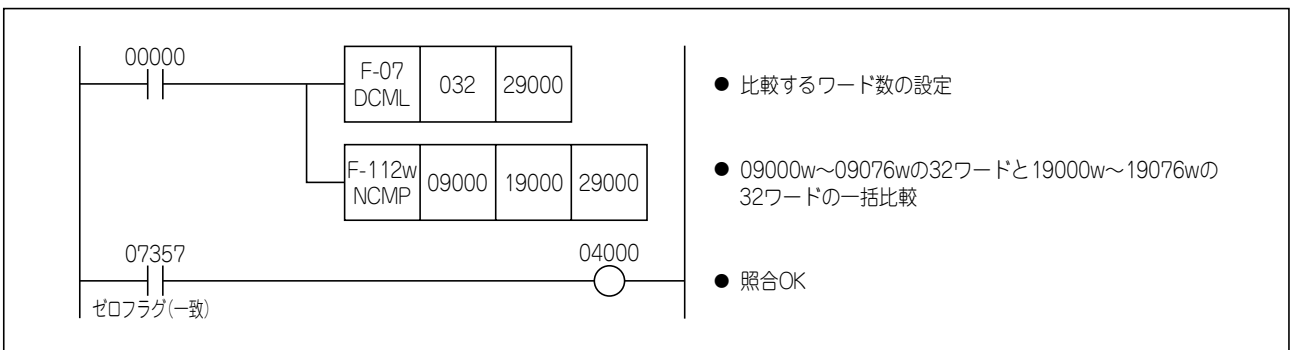
**参考** JW20H, JW30H, JW50H/70H/100Hでは、F-12d命令(ダブルワードの比較)の連続16個の使用が可能です。



(2) F-144(FOR)~F-145(NEXT)を使用する場合



(3) F-112w(nワード一括比較)を使用する場合[JW30H, JW50H/70H/100Hのみ]



**参考** 29000=000に設定すると256ワードの一括比較が可能です。

## 2 - 32 BCD4桁の最小値・最大値を求める

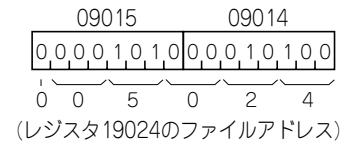
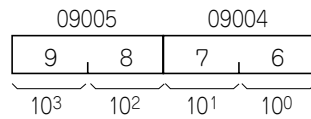
- データテーブル(256個)に格納されたBCD4桁の数値群から最小値、最大値およびそのデータが格納されたレジスタのファイルアドレスを求めます。
- 同一データがある場合は、アドレス番号の小さいアドレスを格納します。

レジスタ	内 容
09000	データテーブルより取り出したデータ(下位)
09001	〃 (上位)
09002	最小値格納用( $10^1, 10^0$ )
09003	〃 ( $10^3, 10^2$ )
09004	最大値格納用( $10^1, 10^0$ )
09005	〃 ( $10^3, 10^2$ )
09006	データテーブルより取り出す間接アドレス(下位)
09007	〃 (上位)
09010	〃 (ファイルNo.)
09011	未使用
09012	最小値のファイルアドレス(下位)
09013	〃 (上位)
09014	最大値のファイルアドレス(下位)
09015	〃 (上位)
09016	演算用ワーキングレジスタ
09017	〃

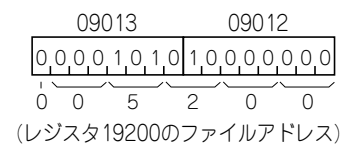
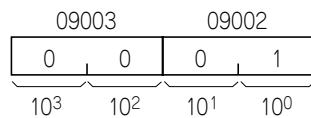
(例)

1	19000	1	2
	19001	3	4
2	19002	5	6
	19003	2	3
3	19004	0	4
	19005	1	6
⋮	⋮	⋮	⋮
	⋮	⋮	⋮
11	19024	7	6
	19025	9	8
⋮	⋮	⋮	⋮
	⋮	⋮	⋮
65	19200	0	1
	19201	0	0
⋮	⋮	⋮	⋮
	⋮	⋮	⋮
255	19774	0	1
	19775	0	0
256	19776	2	3
	19777	1	2

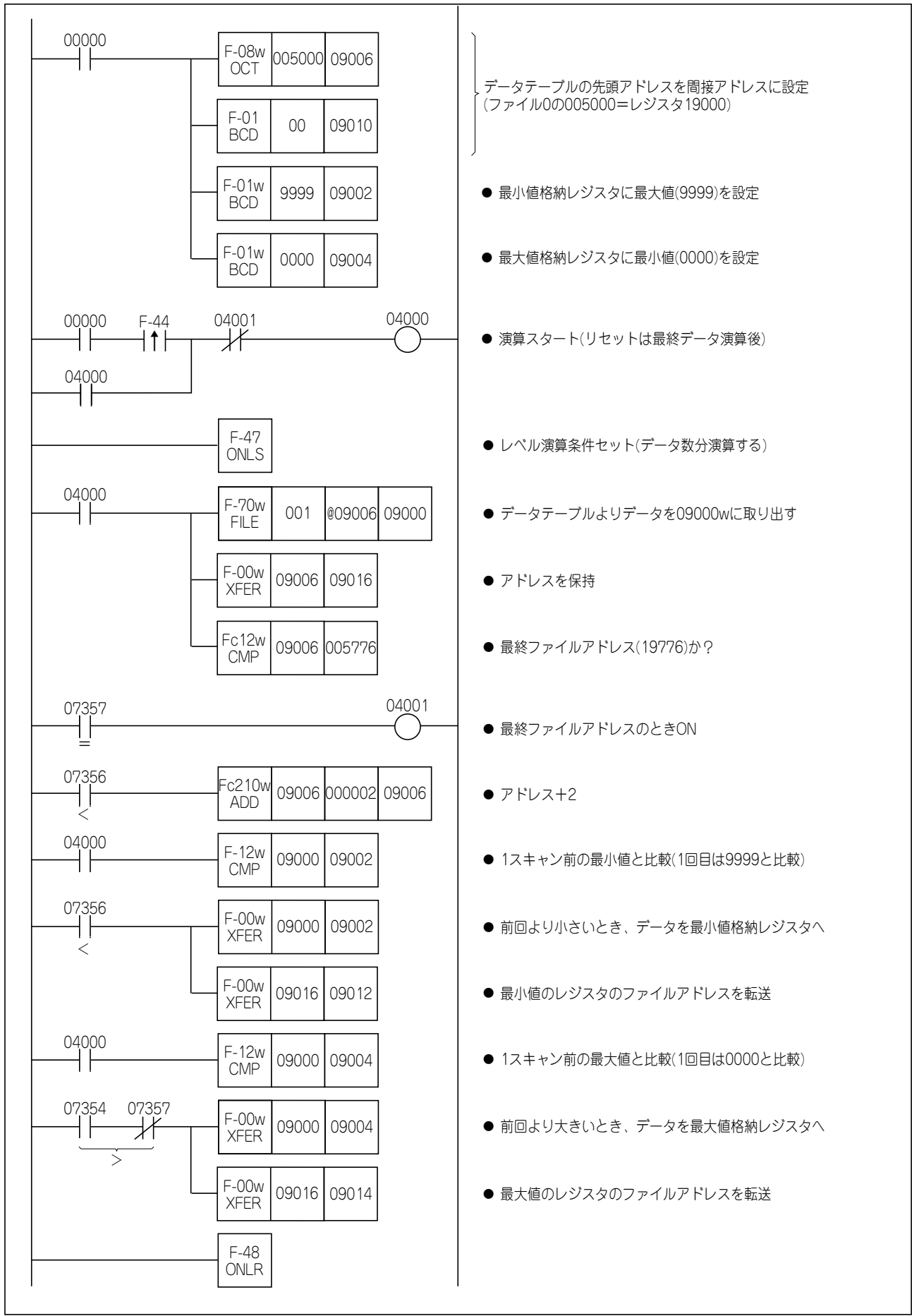
最大値



最小値



本例の場合19200wと19776wに共に最小値0001が格納されていますが、アドレスの小さい19200のファイルアドレスを格納します。

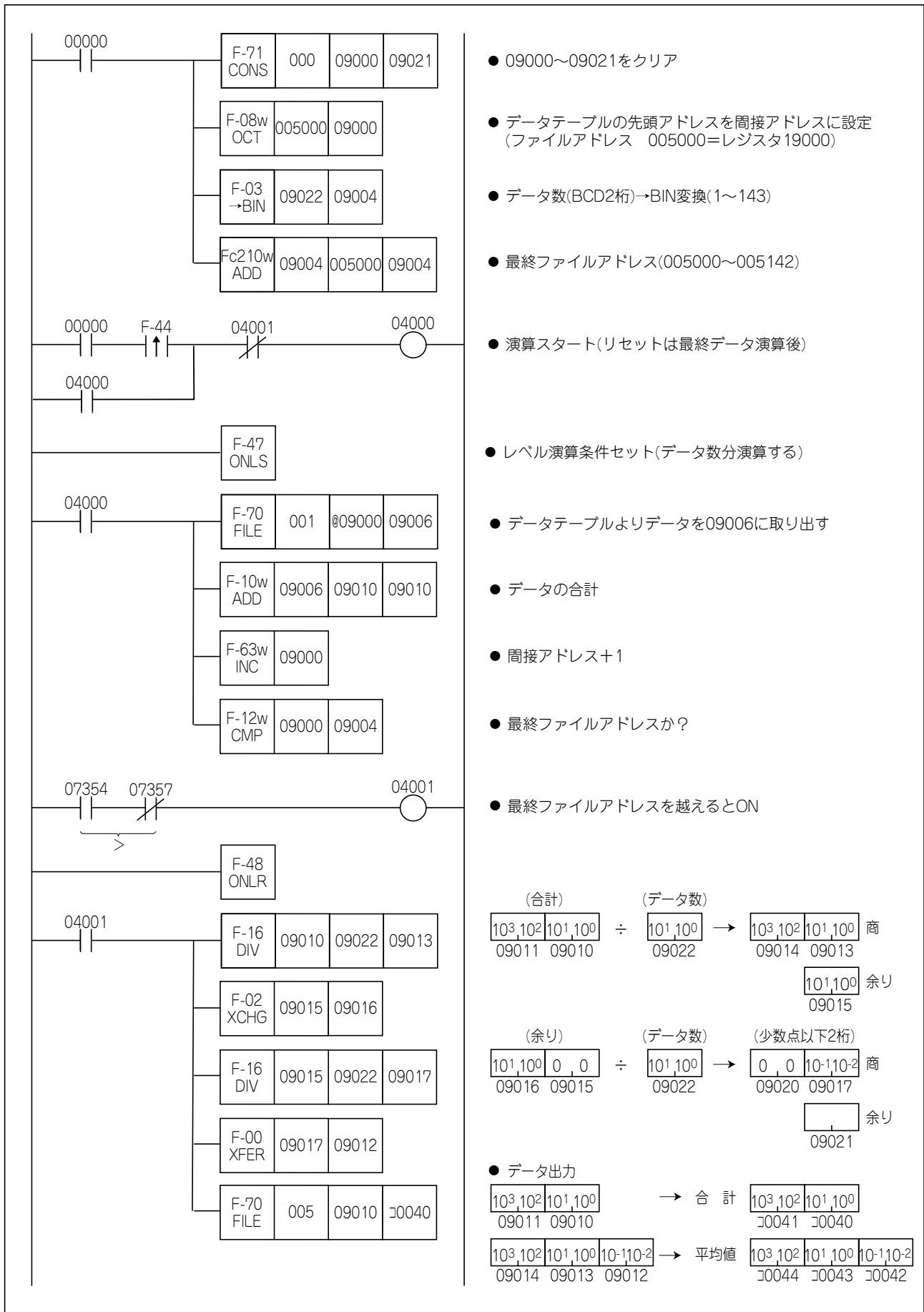


## 2 - 33 BCD2桁の数値の平均値を求める

- データテーブル(最大99個)に格納されているBCD2桁の数値の平均値を求めます。
- 平均値は小数点以下2桁まで求め、3桁以下は切り捨てます。

レジスタ	内 容
09000	データテーブルアドレスの間接アドレス(下位)
09001	〃 (上位)
09002	〃 (ファイル0)
09003	未使用
09004	データテーブルの最終アドレス(下位)
09005	〃 (上位)
09006	読出データ
09007	整数部除算用
09010	合計値( $10^1$ , $10^0$ )
09011	〃 ( $10^3$ , $10^2$ )
09012	平均値( $10^{-1}$ , $10^{-2}$ )
09013	〃 ( $10^1$ , $10^0$ )
09014	〃 ( $10^3$ , $10^2$ )
09015	整数部除算時の余り
09016	小数点以下2桁の演算用
09017	〃 (商)
09020	〃 (商)
09021	〃 (余り)
09022	データ数

データテーブル		
19000	1   2	1番目データ
19001	3   4	2番目データ
19002	5   8	3番目データ
19003	6   9	4番目データ
⋮	⋮	⋮
19141	7   3	98番目データ
19142	2   1	99番目データ
⌋0040	10 <sup>1</sup>   10 <sup>0</sup>	} 合計値
⌋0041	10 <sup>3</sup>   10 <sup>2</sup>	
⌋0042	10 <sup>-1</sup>   10 <sup>-2</sup>	} 平均値
⌋0043	10 <sup>1</sup>   10 <sup>0</sup>	
⌋0044	10 <sup>3</sup>   10 <sup>2</sup>	



- 09000~09021をクリア
- データテーブルの先頭アドレスを間接アドレスに設定 (ファイルアドレス 005000=レジスタ19000)
- データ数(BCD2桁)→BIN変換(1~143)
- 最終ファイルアドレス(005000~005142)
- 演算スタート(リセットは最終データ演算後)
- レベル演算条件セット(データ数分演算する)
- データテーブルよりデータを09006に取り出す
- データの合計
- 間接アドレス+1
- 最終ファイルアドレスか?
- 最終ファイルアドレスを越えるとON

(合計)                      (データ数)

$$\frac{\begin{array}{|c|c|c|} \hline 10^3 & 10^2 & 10^1, 10^0 \\ \hline 09011 & 09010 & \end{array}}{\begin{array}{|c|} \hline 10^1, 10^0 \\ \hline 09022 \\ \hline \end{array}} \rightarrow \begin{array}{|c|c|c|} \hline 10^3 & 10^2 & 10^1, 10^0 \\ \hline 09014 & 09013 & \end{array} \text{商}$$

$$\begin{array}{|c|} \hline 10^1, 10^0 \\ \hline 09015 \\ \hline \end{array} \text{余り}$$

(余り)                      (データ数)                      (少数点以下2桁)

$$\frac{\begin{array}{|c|c|c|} \hline 10^1, 10^0 & 0 & 0 \\ \hline 09016 & 09015 & \end{array}}{\begin{array}{|c|} \hline 10^1, 10^0 \\ \hline 09022 \\ \hline \end{array}} \rightarrow \begin{array}{|c|c|c|} \hline 0 & 0 & 10^{-1}, 10^{-2} \\ \hline 09020 & 09017 & \end{array} \text{商}$$

$$\begin{array}{|c|} \hline \phantom{0} \\ \hline 09021 \\ \hline \end{array} \text{余り}$$

● データ出力

$$\begin{array}{|c|c|c|} \hline 10^3 & 10^2 & 10^1, 10^0 \\ \hline 09011 & 09010 & \end{array} \rightarrow \text{合計} \begin{array}{|c|c|c|} \hline 10^3 & 10^2 & 10^1, 10^0 \\ \hline 00041 & 00040 & \end{array}$$

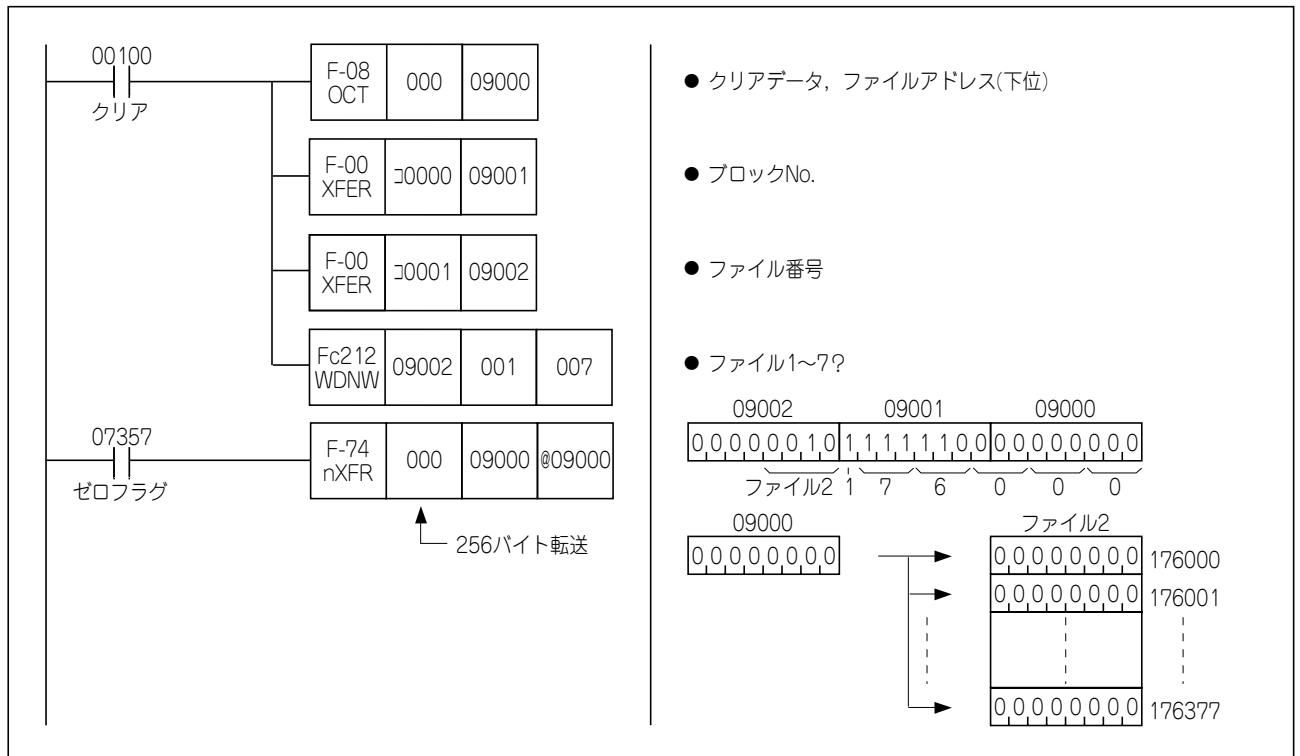
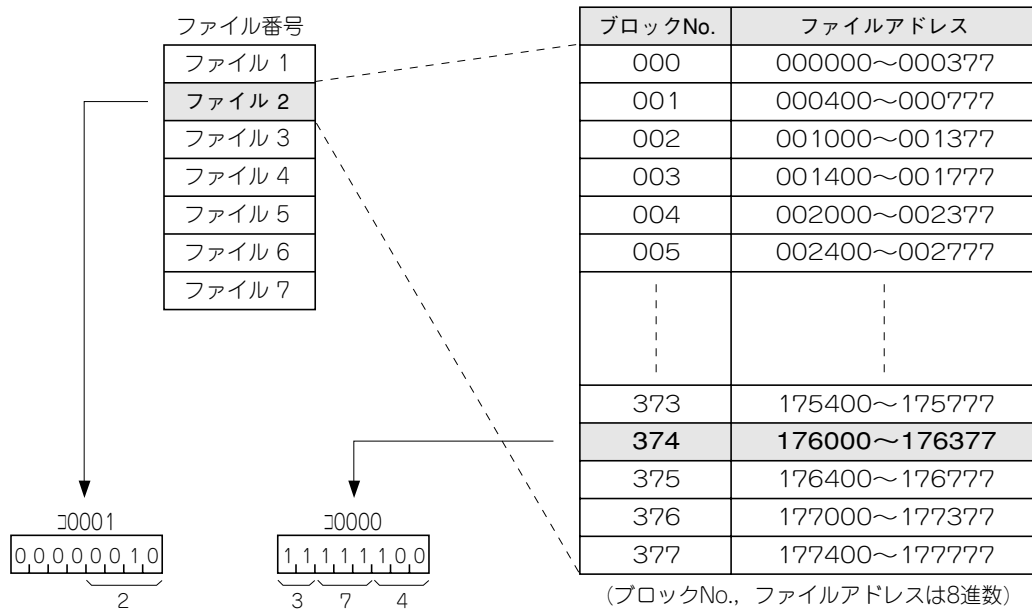
$$\begin{array}{|c|c|c|c|} \hline 10^3 & 10^2 & 10^1, 10^0 & 10^{-1}, 10^{-2} \\ \hline 09014 & 09013 & 09012 & \end{array} \rightarrow \text{平均値} \begin{array}{|c|c|c|c|} \hline 10^3 & 10^2 & 10^1, 10^0 & 10^{-1}, 10^{-2} \\ \hline 00044 & 00043 & 00042 & \end{array}$$

## 2 - 34 ファイルレジスタの領域指定クリア

- ハンディプログラマJW-13PG等でファイルレジスタをクリアすると、ファイル番号毎に全領域(64kバイト)がクリアされてしまいますが、本プログラムを使用すると、ファイルレジスタを任意のブロック(256バイト単位)を指定してクリアできます。

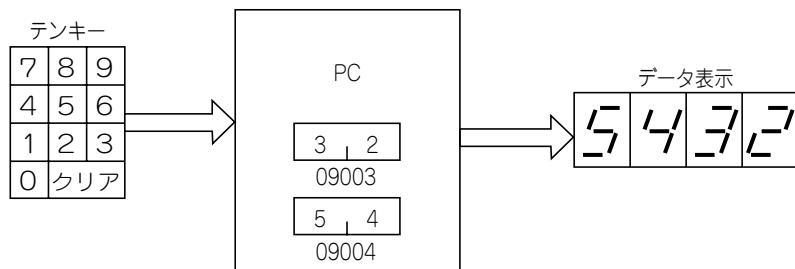
**適応PC** JW30H(JW-32CUH/H1  
JW-33CUH/H1/H2/H3)  
JW50H/70H/100H

**注1** PC機種により使用できるファイルレジスタ領域が異なります。各機種のマニュアルをご参照ください。

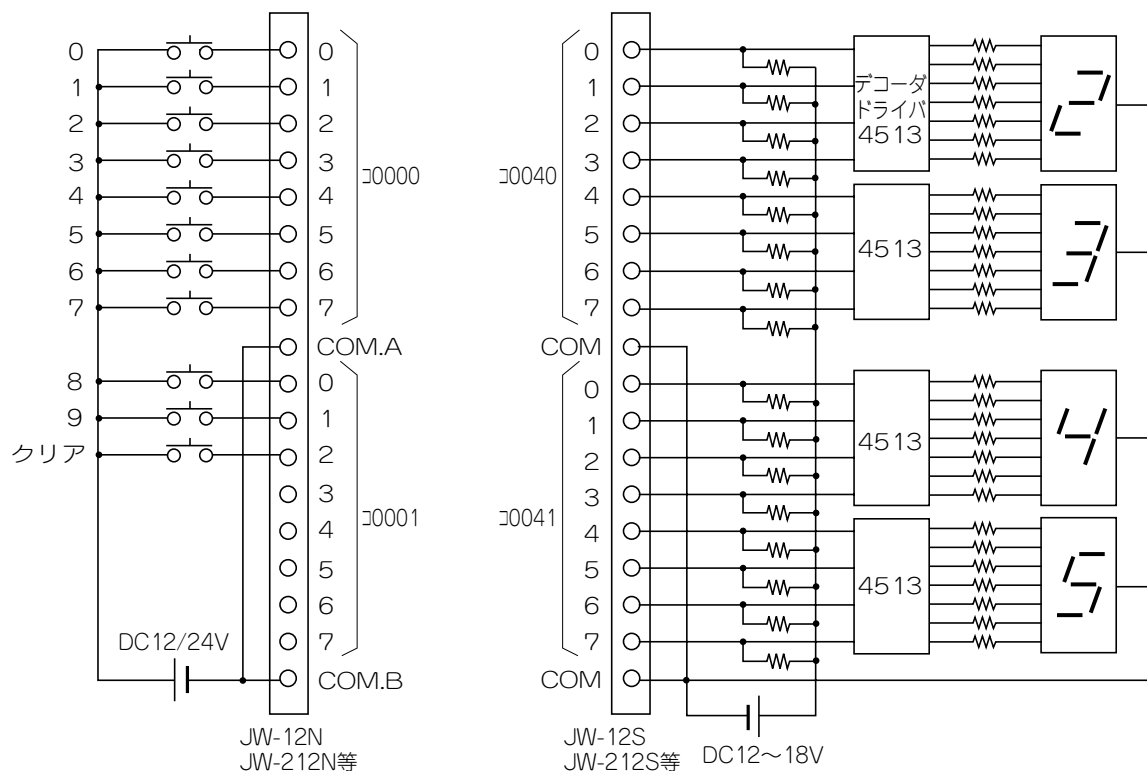


## 2 - 35 テンキーからの数値の読込

- テンキーから入力されるBCD4桁の数値をレジスタに読み込みます。

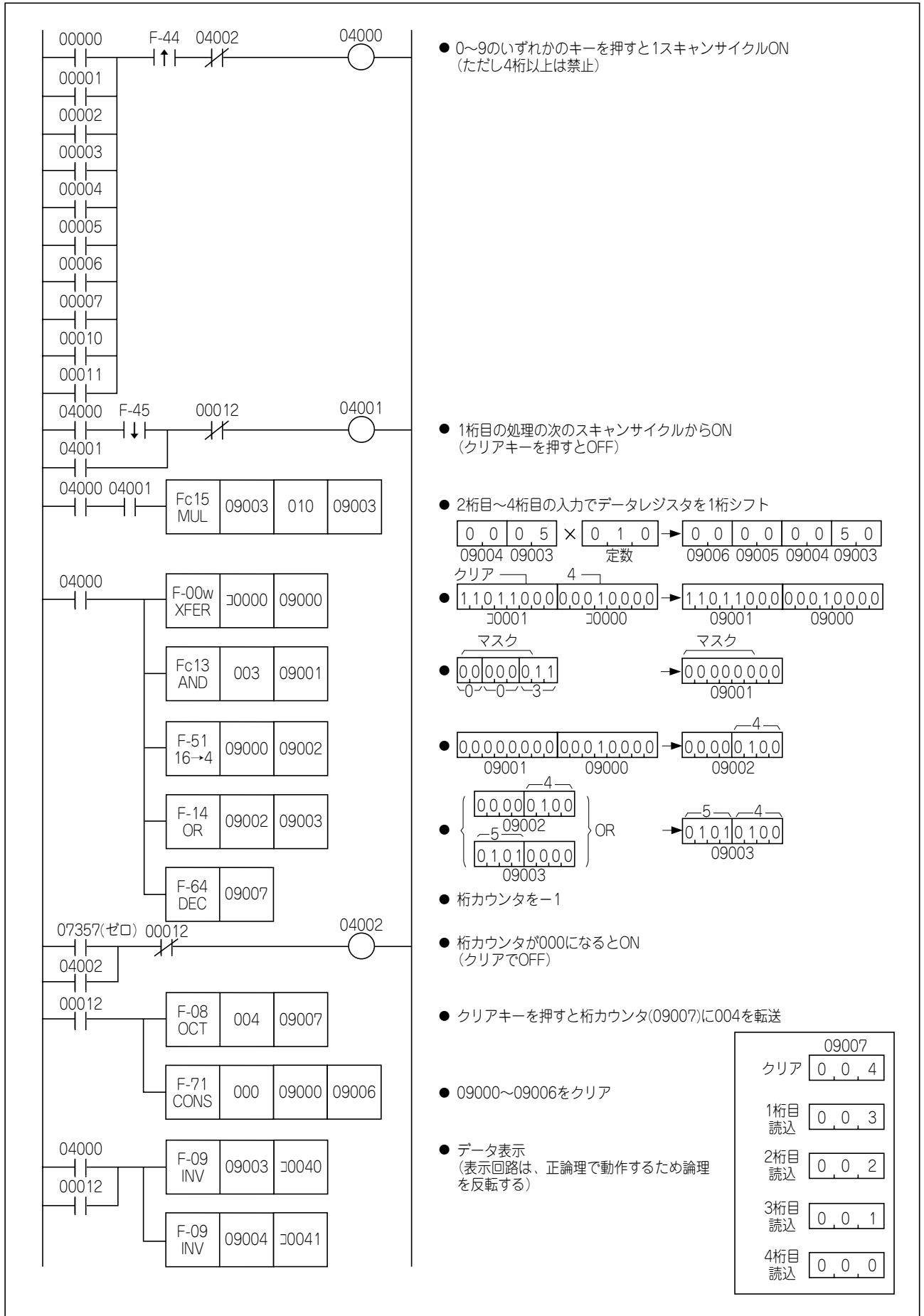


### (1) 外部接続





(2) プログラム

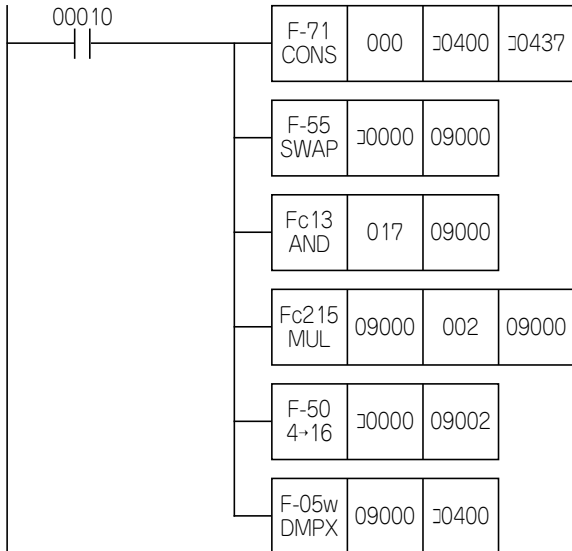


## 2 - 36 8 → 256 デコーダ

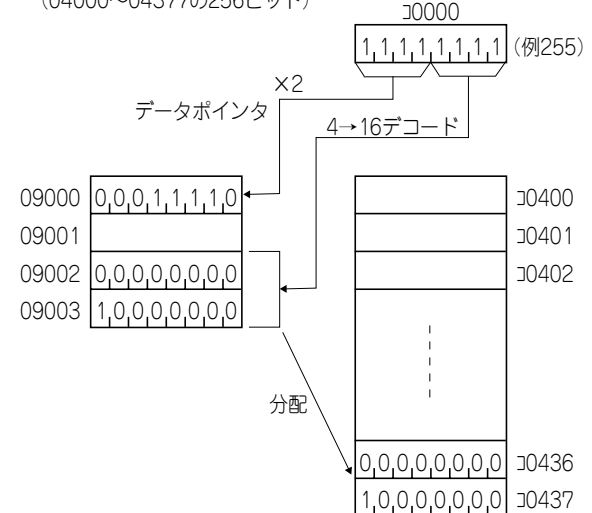
- 8ビットバイナリデータ(0~255)をデコードし、256ビット中の1ビットをONします。

[F-50(4→16デコーダ)の拡張]

(1) 分配命令(F-05w)を使用したプログラム(JW10は不可)

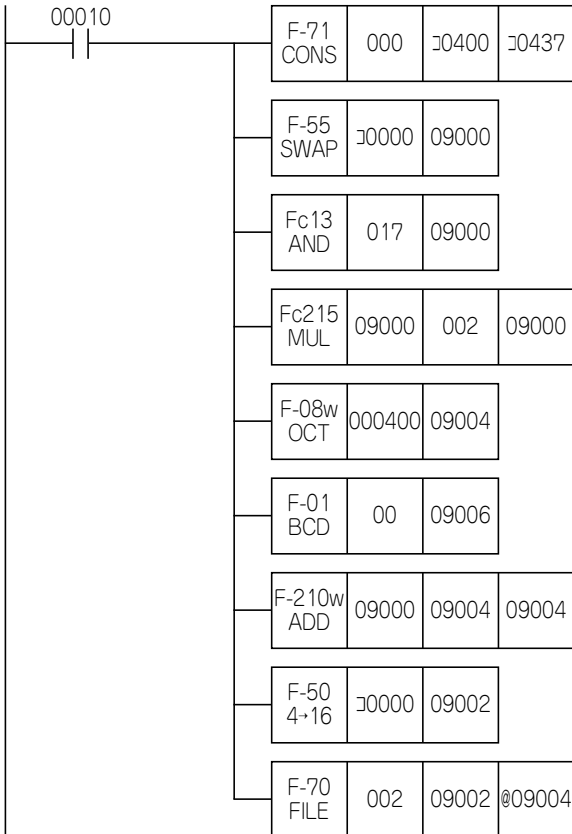


- デコード結果格納領域のクリア(04000~04377の256ビット)



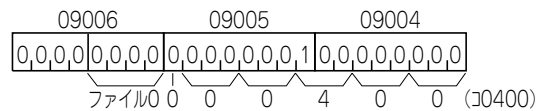
- JW10は、分配命令(F-05w)がありませんので、一括転送命令(F-70)を間接アドレスで使用します。

(2) 間接アドレスを使用したプログラム



- デコード結果格納領域のクリア(04000~04377の256ビット)
- 00000の上下4ビットを交換し、09000へ
- 下位4ビットのマスク
- 16ビットデータのため2倍する

- 0400をファイルアドレスで設定

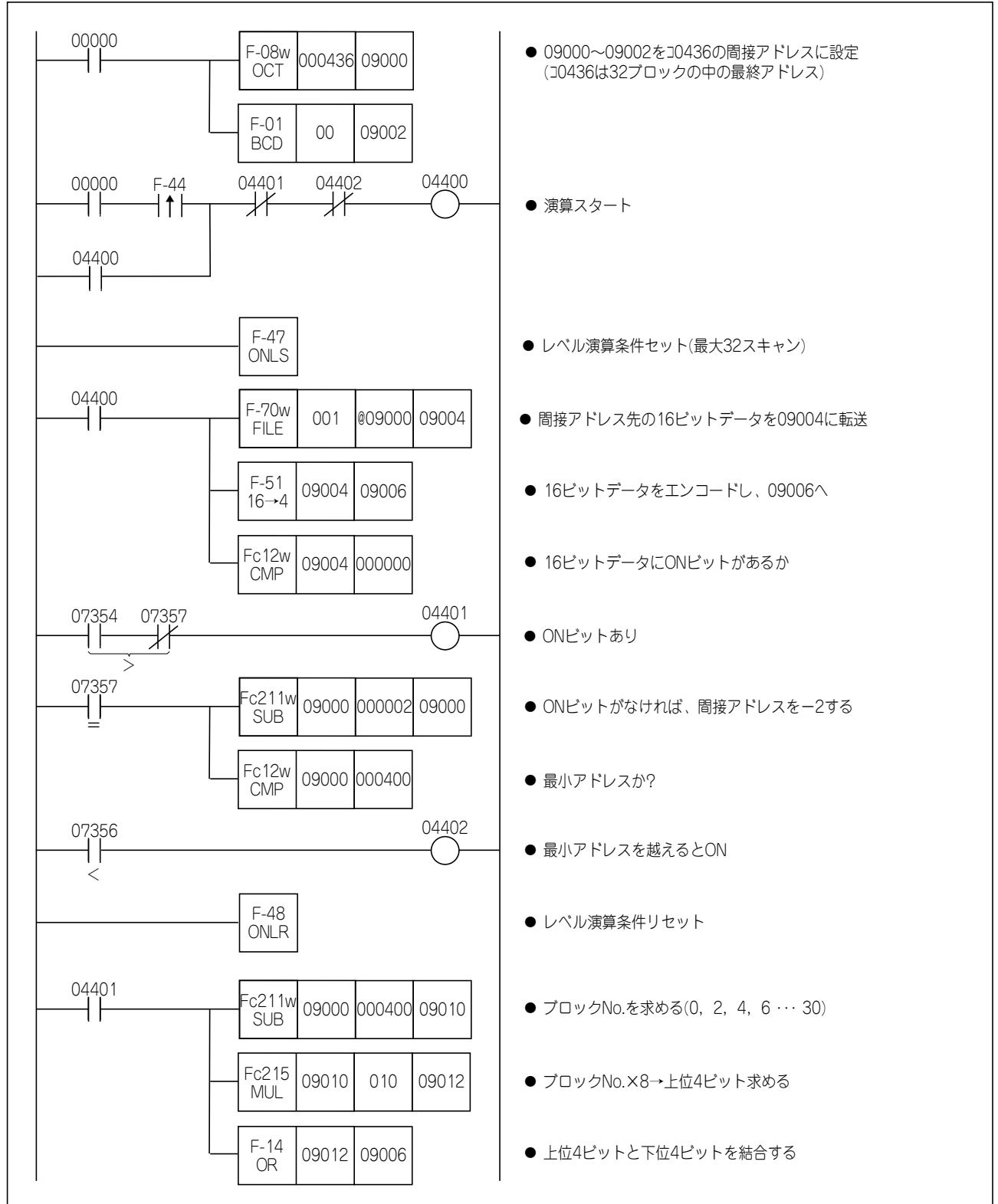


- 0400にデータポインタを加える
- 4→16デコード(下位4ビット)
- デコード値を間接指定されたアドレスのレジスタに転送

## 2 - 37 256 → 8 エンコーダ

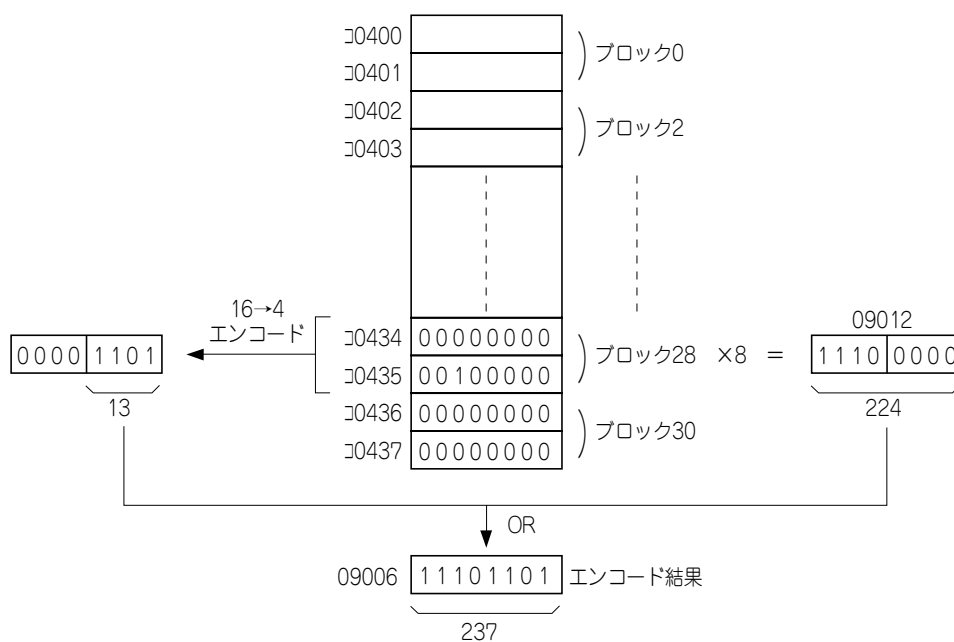
- 04000~04377の256点のデータをエンコードします。
- 同時に2点以上がONした場合、リレー番号の大きい方を優先し演算します。

[F-51(16→4エンコーダ)の拡張]



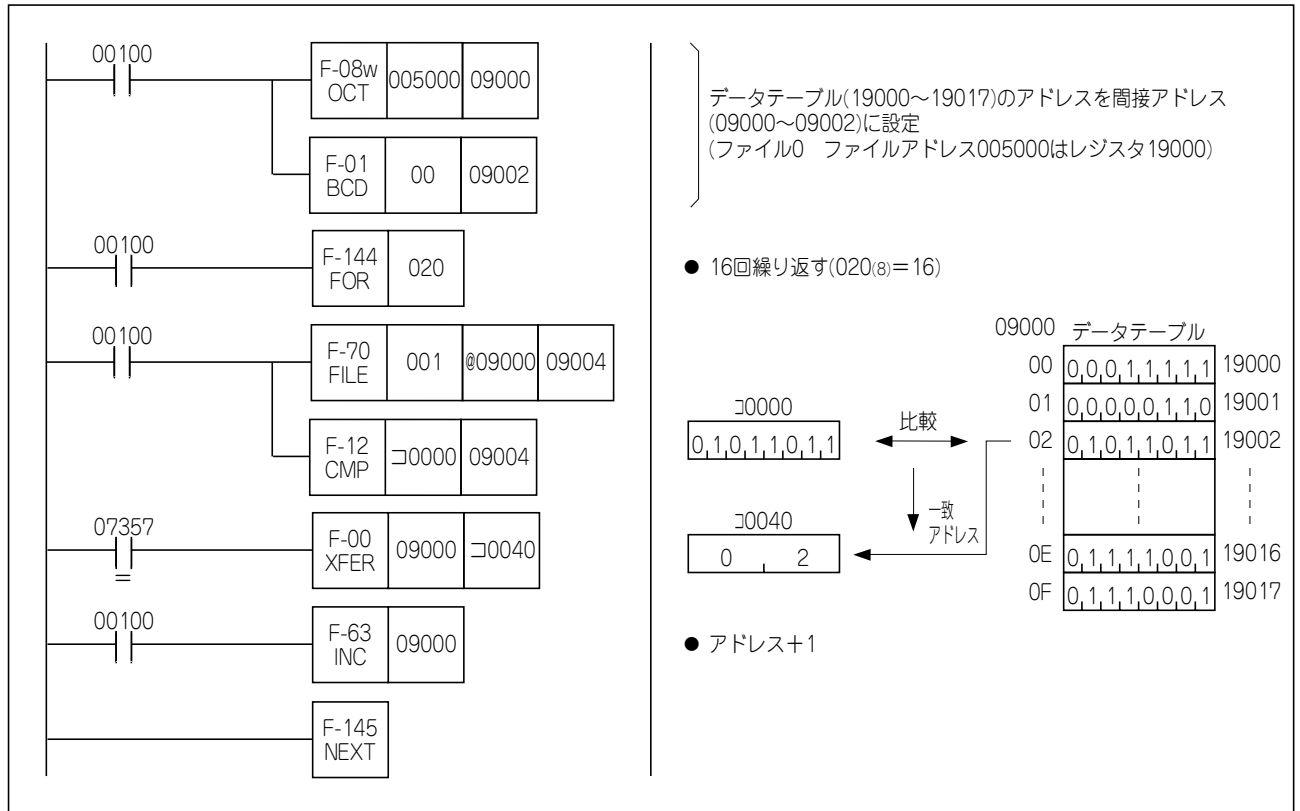
- 本プログラムでは、256ビットを16ビット(下位)×16(上位)として考えます。
- 間接アドレス指定により、2バイトずつ抽出した16ビットデータを16→4エンコードしたものの(下位4ビット)と16個のブロックNo.(偶数)から求めた上位4ビットを結合(OR)しています。

レジスタ	内 容
09000	00400~00436の間接アドレス(下位)
09001	〃 (上位)
09002	〃 (ファイルNo.)
09003	未使用
09004	1ブロックの16ビットデータ(下位)
09005	〃 (上位)
09006	エンコードした結果
09007	未使用
09010	ブロックNo.(下位)
09011	〃 (上位)=00(H)
09012	ブロックNo.×8=上位4ビット
09013	〃 =00(H)



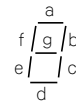
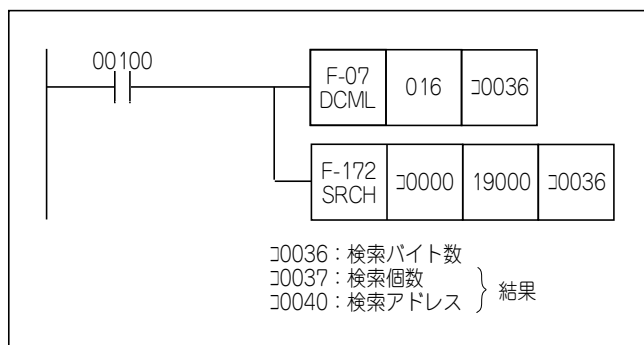
## 2 - 38 7SEG エンコーダ

- 7ビットの7セグメントデータより数値を求めます。
- 7セグメント表示器の点灯検査などに利用できます。



- 上記プログラムは、16個のデータテーブルにあらかじめ7セグメントデータを格納しておき、入力データと比較しています。

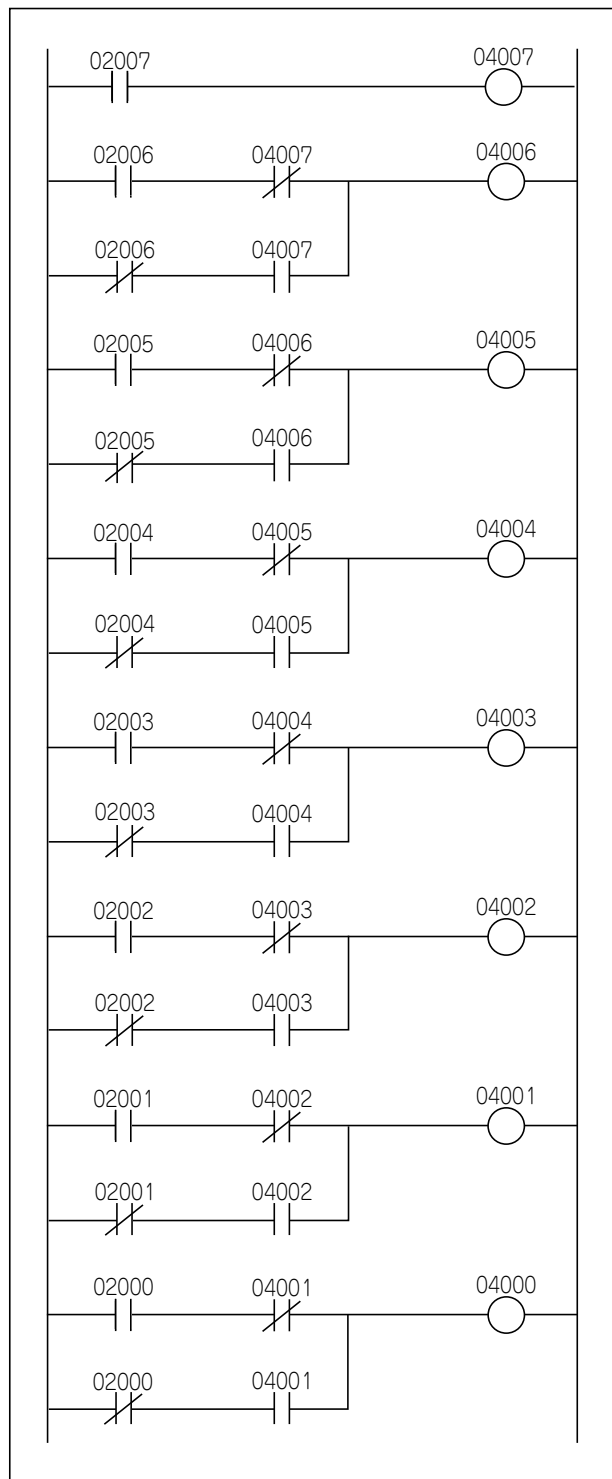
**参考** JW30H(JW-32CUH/H1, JW-33CUH/H1/H2/H3), JW50H/70H/100Hでは、データの検索命令(F-172)でも実現できます。



入力文字	入力データ	出力データ
	g f e d c b a	
0	0 0 1 1 1 1 1 1	0 0 0 0 0 0 0 0
1	0 0 0 0 0 1 1 0	0 0 0 0 0 0 0 1
2	0 1 0 1 1 0 1 1	0 0 0 0 0 0 1 0
3	0 1 0 0 1 1 1 1	0 0 0 0 0 0 1 1
4	0 1 1 0 0 1 1 0	0 0 0 0 0 1 0 0
5	0 1 1 0 1 1 0 1	0 0 0 0 0 1 0 1
6	0 1 1 1 1 1 0 1	0 0 0 0 0 1 1 0
7	0 0 1 0 0 1 1 1	0 0 0 0 0 1 1 1
8	0 1 1 1 1 1 1 1	0 0 0 0 1 0 0 0
9	0 1 1 0 1 1 1 1	0 0 0 0 1 0 0 1
A	0 1 1 1 0 1 1 1	0 0 0 0 1 0 1 0
b	0 1 1 1 1 1 0 0	0 0 0 0 1 0 1 1
c	0 0 1 1 1 0 0 1	0 0 0 0 1 1 0 0
d	0 1 0 1 1 1 1 0	0 0 0 0 1 1 0 1
E	0 1 1 1 1 0 0 1	0 0 0 0 1 1 1 0
F	0 1 1 1 0 0 0 1	0 0 0 0 1 1 1 1

## 2 - 39 グレイコード→バイナリコード変換

- 電子ハカリやロータリエンコーダ(アブソリュート形)で用いられるグレイコードをバイナリコードに変換します。
- J0200(グレイコード 8ビット)→J0400(バイナリコード 8ビット)変換のプログラム例です。



	[J0200]	[J0400]
	グレイコード (8ビット)	バイナリコード (8ビット)
0	00000000	00000000
1	00000001	00000001
2	00000011	00000010
3	00000010	00000011
4	00000110	00000100
5	00000111	00000101
6	00000101	00000110
7	00000100	00000111
8	00001100	00001000
9	00001101	00001001
10	00001111	00001010
11	00001110	00001011
12	00001010	00001100
13	00001011	00001101
14	00001001	00001110
15	00001000	00001111
16	00011000	00010000
17	00011001	00010001
18	00011011	00010010
19	00011010	00010011
20	00011110	00010100
21	00011111	00010101
22	00011101	00010110
23	00011100	00010111
24	00010100	00011000
25	00010101	00011001
⋮	⋮	⋮
249	10000101	11111001
250	10000111	11111010
251	10000110	11111011
252	10000010	11111100
253	10000011	11111101
254	10000001	11111110
255	10000000	11111111

## 2 - 40 BCD6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BCD6桁(000000～999999)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。

**参考** JW20H, JW30H, JW50H/70H/100Hには、BCD8桁のアップ・ダウンカウンタ(F-62d)があります。

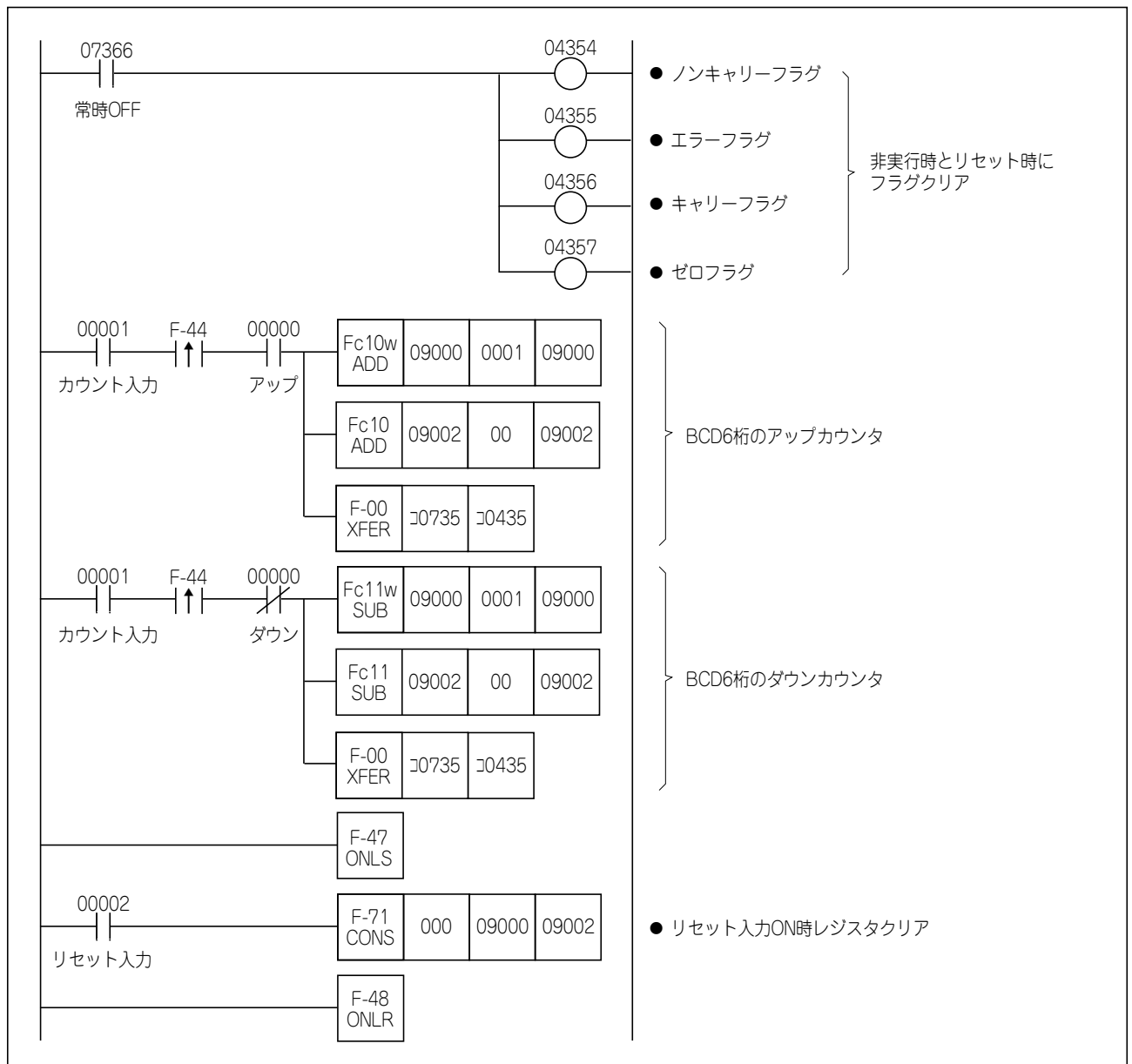
- 00000 : アップ・ダウン指示入力  
(ON : アップ OFF : ダウン)
- 00001 : カウント入力(OFF→ONの立上りで実行)
- 00002 : リセット入力(OFF→ONでレジスタクリア、ON中はカウント非実行)
- 04354 : ノンキャリーフラグ
- 04355 : エラーフラグ
- 04356 : キャリーフラグ
- 04357 : ゼロフラグ
- 04350～04353 : 不定(他回路での使用禁止)
- 09000～09002 : BCD6桁用レジスタ

演算フラグ →

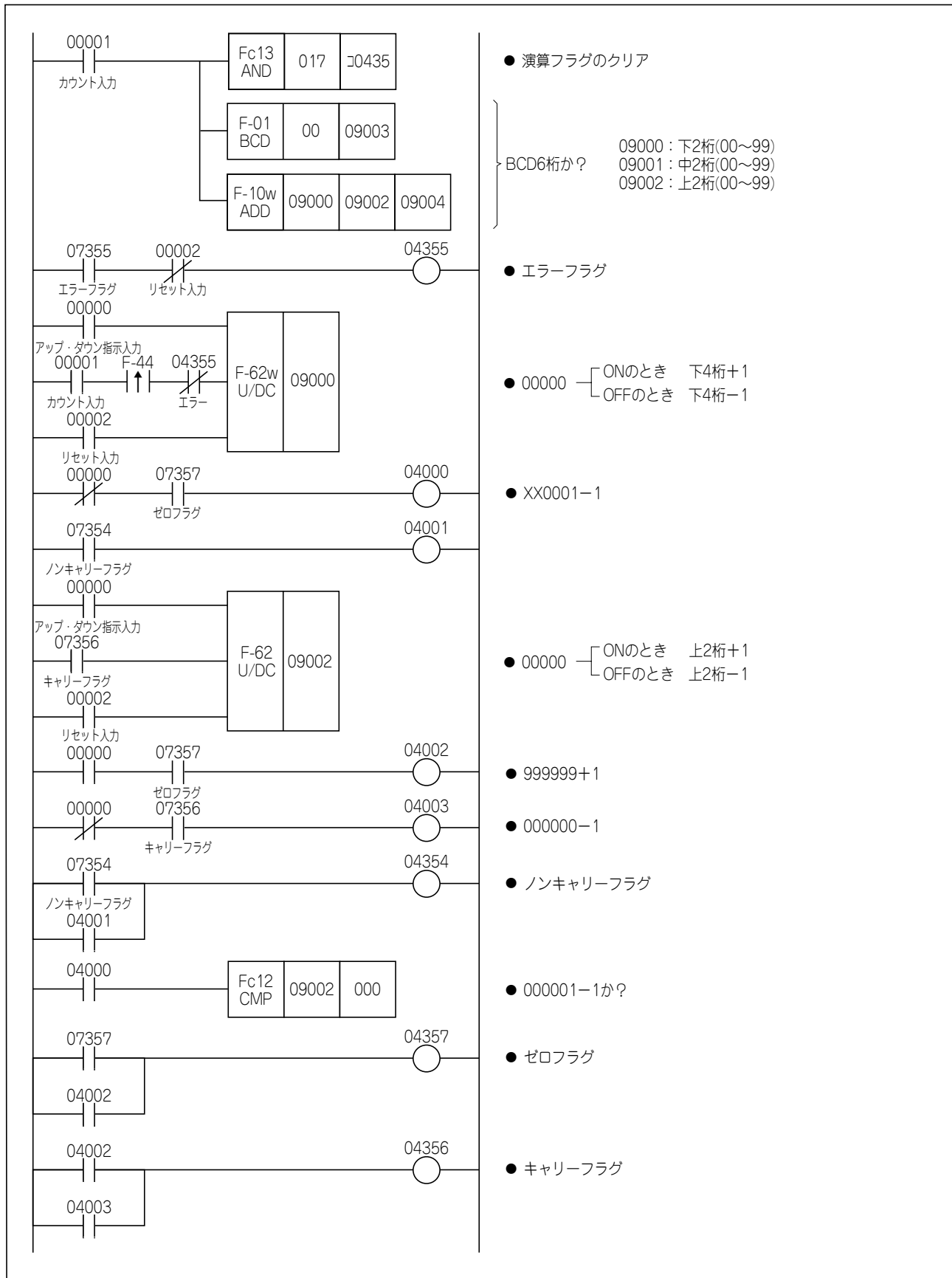
アップ・ダウン指示入力	演算結果	ノンキャリー 04354	エラー 04355	キャリー 04356	ゼロ 04357
ON	999999+1	0	0	1	1
	000000～999998+1	1	0	0	0
	BCD以外の数値	0	1	0	0
OFF	000000-1	0	0	1	0
	000001-1	1	0	0	1
	000002～999999-1	1	0	0	0
	BCD以外の数値	0	1	0	0

09002		09001		09000	
10 <sup>5</sup>	10 <sup>4</sup>	10 <sup>3</sup>	10 <sup>2</sup>	10 <sup>1</sup>	10 <sup>0</sup>

(プログラム例1)



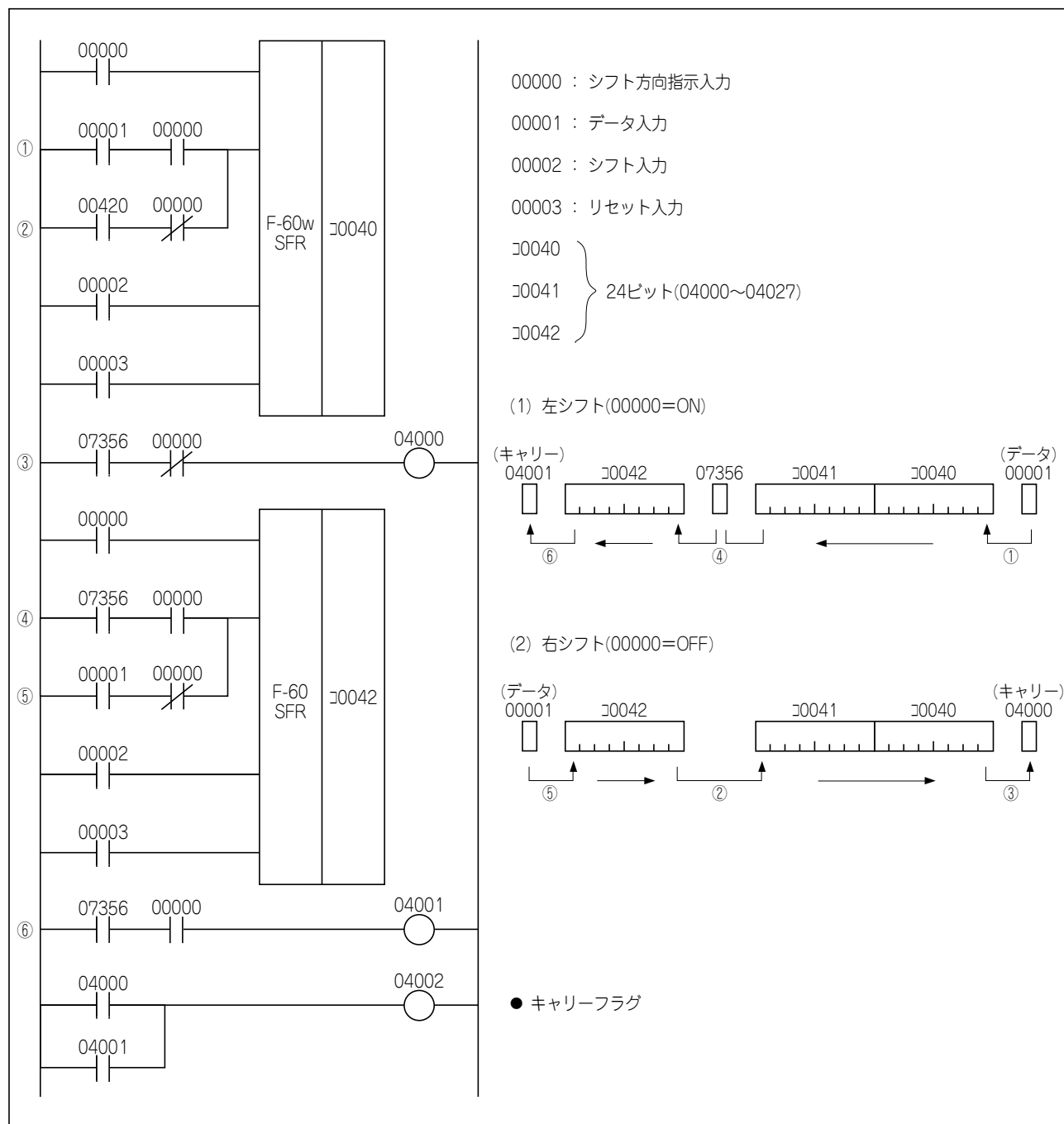
(プログラム例2)



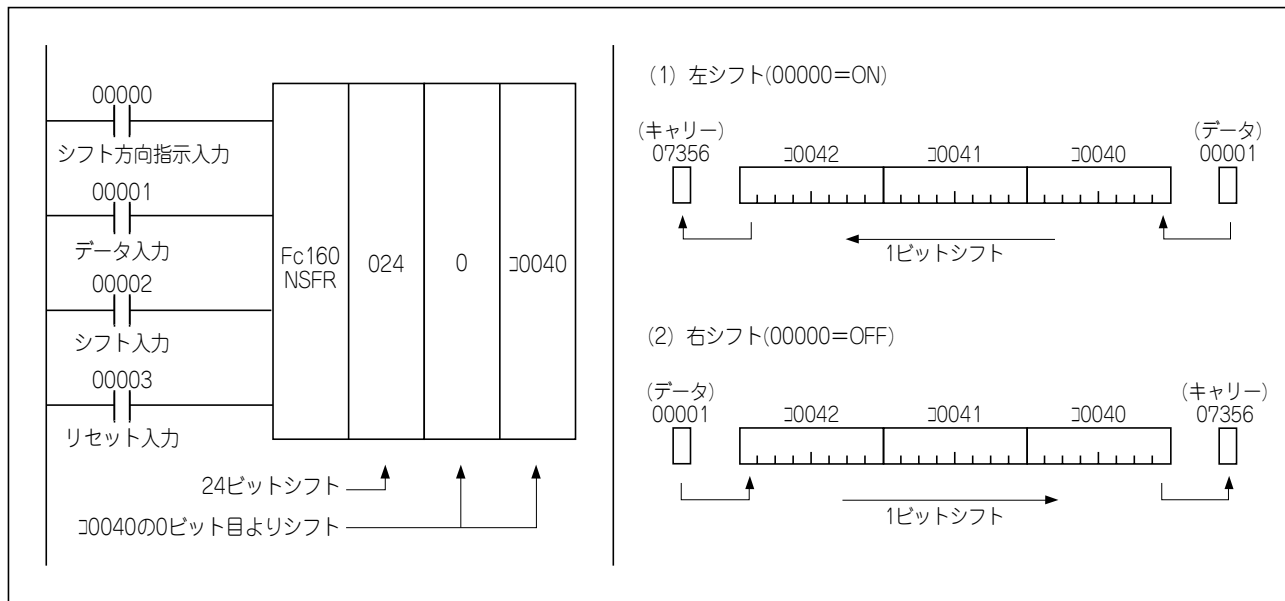


## 2 - 41 24ビットシフトレジスタ

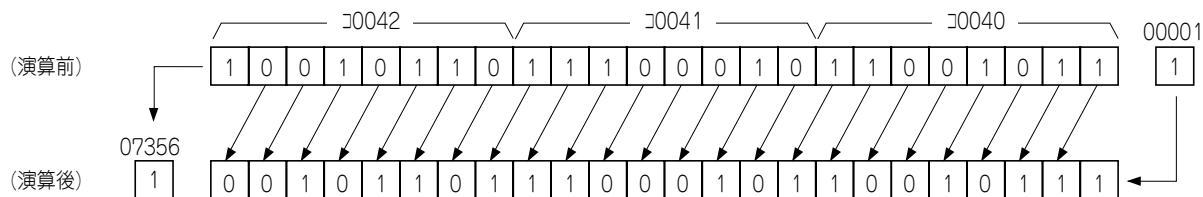
- シフトレジスタ命令(F-60)を複数使用することにより、nビットのシフトレジスタを構成できます。
- 以下に24ビットのシフトレジスタのプログラム例を示します。



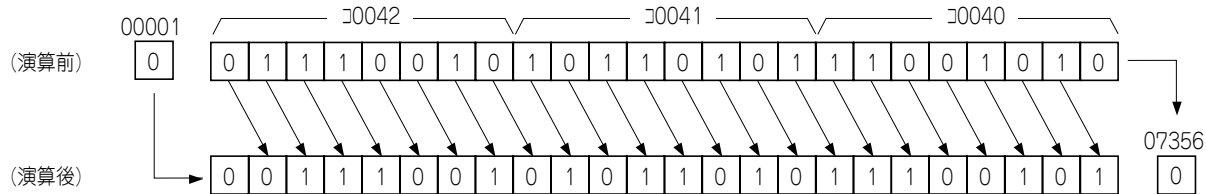
参考 JW30H, JW50H/70H/100Hの場合、nビットシフトレジスタ命令(Fc160)で同じ動作を実現できます。(最大256ビット)



(1) 左シフト例(00000=ON)

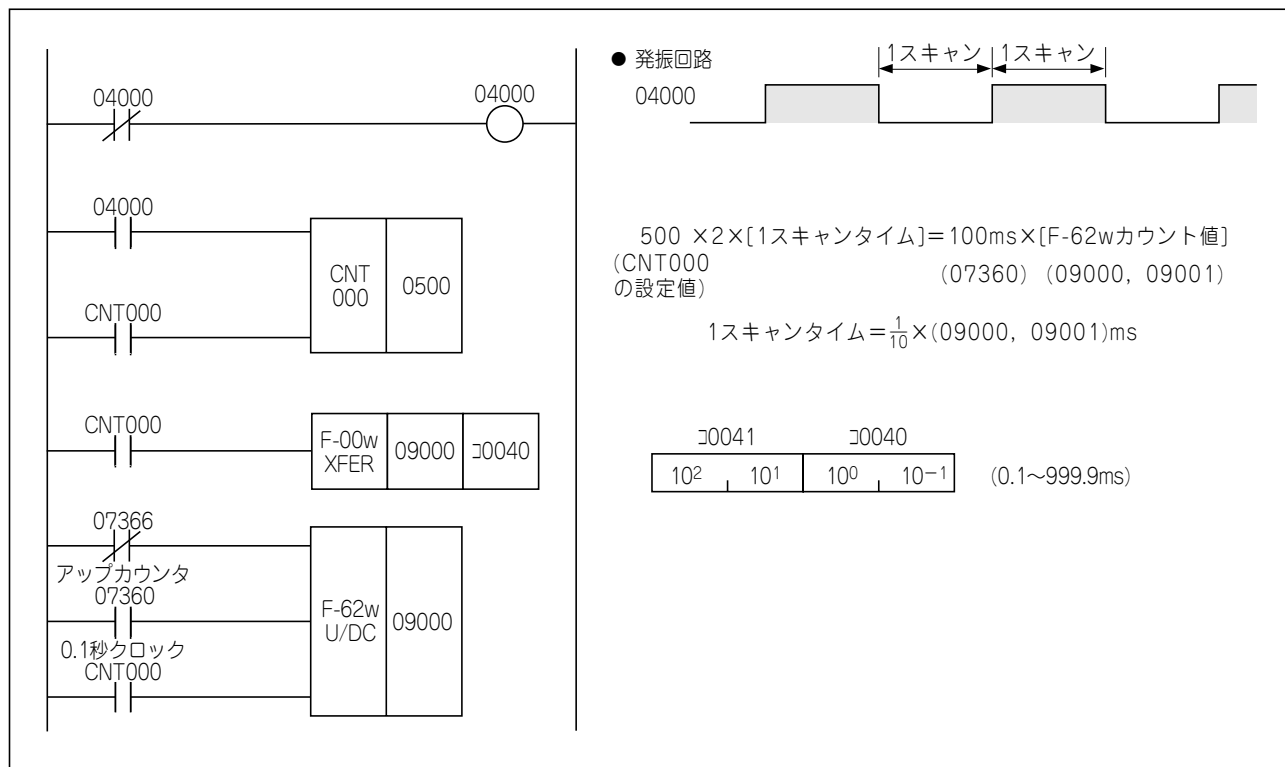


(2) 右シフト例(00000=OFF)



## 2-42 スキャンタイムの測定

- PCのスキャンタイムを発振回路を利用して求めます。
- レジスタに測定結果を格納できますのでスキャンタイムを外部に出力できます。

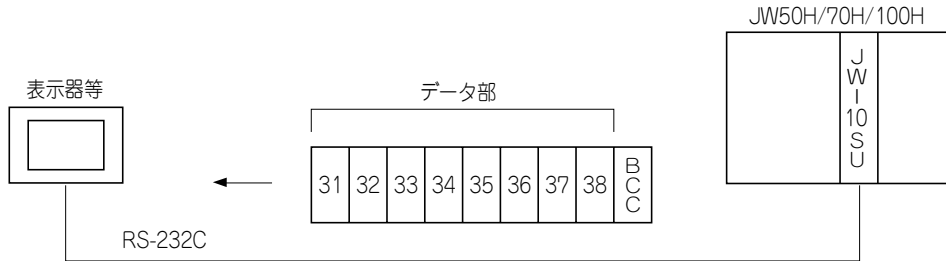


**参考** スキャンタイムは、システムメモリ #030～  
 #035で確認できます。(ms単位)

#030, #031	スキャンタイムの最小値
#032, #033	スキャンタイムの現在値
#034, #035	スキャンタイムの最大値

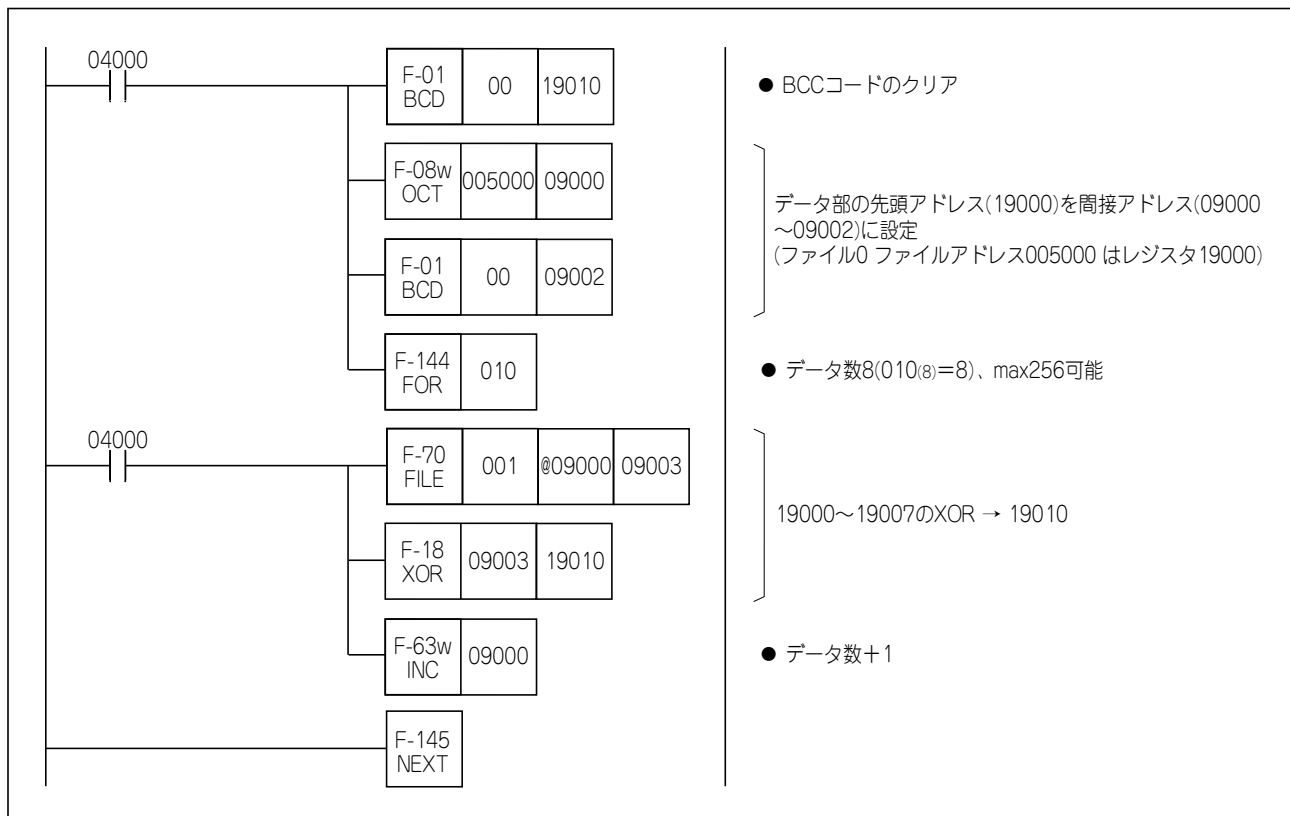
## 2 - 43 BCC コードの生成

- PCと表示器や計測器間でシリアル通信を行う場合、BCCコードを通信データの最後に付けて送受信する場合があります。



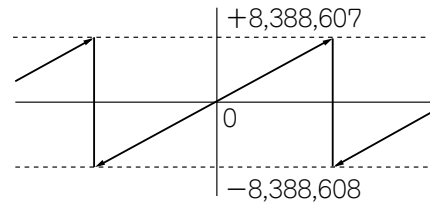
- BCCコードは、データ部の各文字のバイナリ値の排他的論理和(XOR)で生成されます。

レジスタ	データ (H)	バイナリ値			
19000	31	→ 00110001			
19001	32	→ 00110010	XOR	00110001	結果(1)
19002	33	→ 00110011	XOR	00000011	結果(2)
19003	34	→ 00110100	XOR	00110000	結果(3)
19004	35	→ 00110101	XOR	00000100	結果(4)
19005	36	→ 00110110	XOR	00110001	結果(5)
19006	37	→ 00110111	XOR	00000111	結果(6)
19007	38	→ 00111000	XOR	00110000	結果(7)
19010	08	←		00001000	----- BCCコード



## 2 - 44 高速カウンタの現在値を符号付きで表示する

- JW20H/30H用高速カウンタユニット(JW-21HC/22HC)のカウンタ現在値(-8,388,608~8,388,607)の-(マイナス)は、2の補数表現となります。従って-から+、または+から-への値が不連続値となります。
- 本プログラムでは、F-57d(2ワードデータの2の補数)命令を使用することにより、-から+、または+から-への不連続値を連続値に変換して表示します。



カウント値+8,388,607に+1すると-8,388,608になる  
 カウント値-8,388,608に-1すると+8,388,607になる

適応PC JW20H, JW30H

### ■ カウンタ現在値(JW20HでユニットNo.スイッチ0、CH1使用の場合)

	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
コ202	128	84	32	16	8	4	2	1
	2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>
コ203	32768	16384	8192	4096	2048	1024	512	256
	2 <sup>22</sup>	2 <sup>21</sup>	2 <sup>20</sup>	2 <sup>19</sup>	2 <sup>18</sup>	2 <sup>17</sup>	2 <sup>16</sup>	
コ204	符号	4194304	2097152	1048576	524288	262144	131072	65536

OFF(0): +  
ON(1): -

符号	2 <sup>22</sup>	2 <sup>21</sup>	2 <sup>20</sup>	2 <sup>19</sup>	2 <sup>18</sup>	2 <sup>17</sup>	2 <sup>16</sup>	2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	10進数
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	8,388,607
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	8,388,606
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	8,388,605
																								⋮
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	-2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	-3
																								⋮
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	-8,388,606
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-8,388,607
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-8,388,608

カウント値(不連続値)

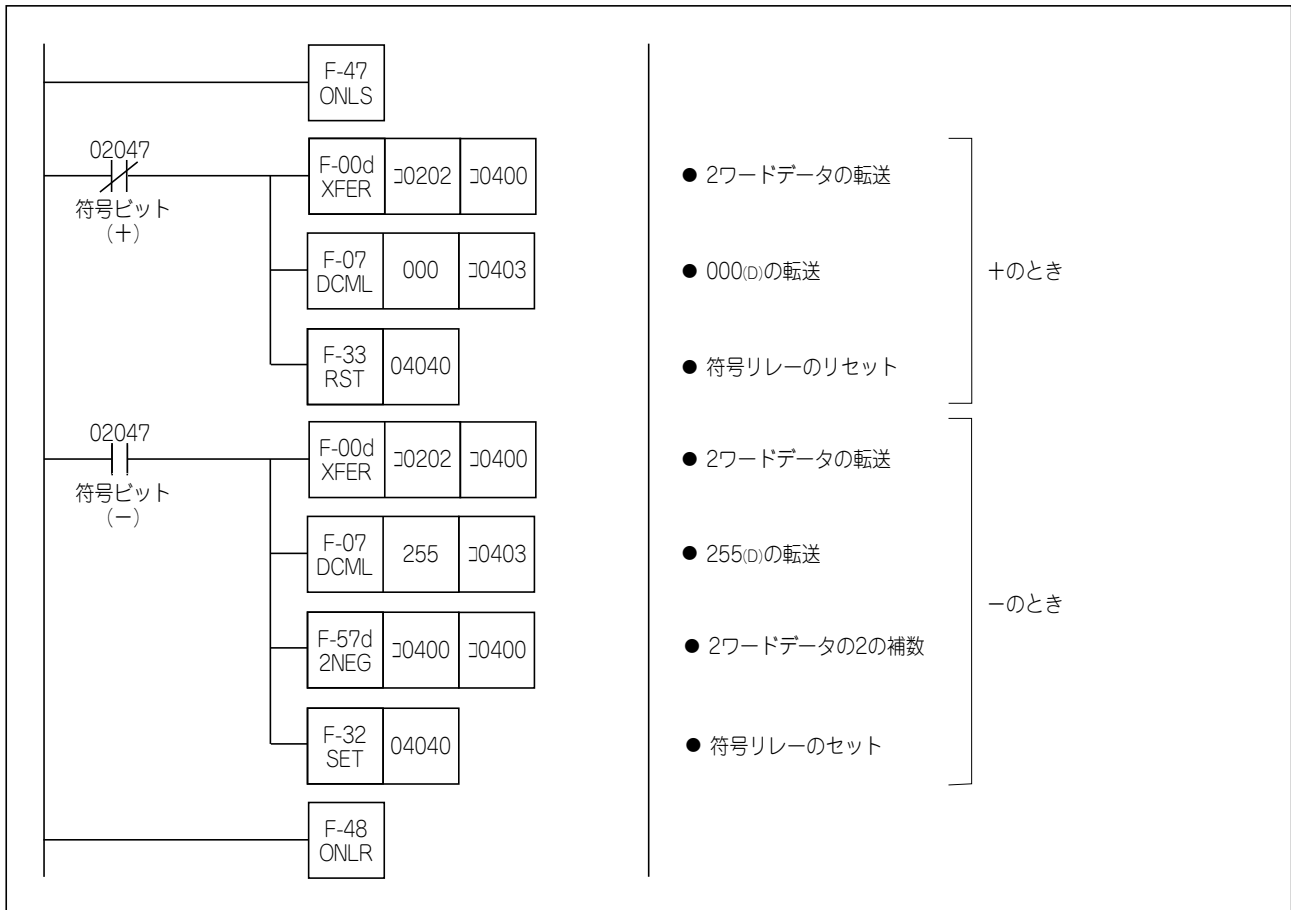
0202
0203
0204



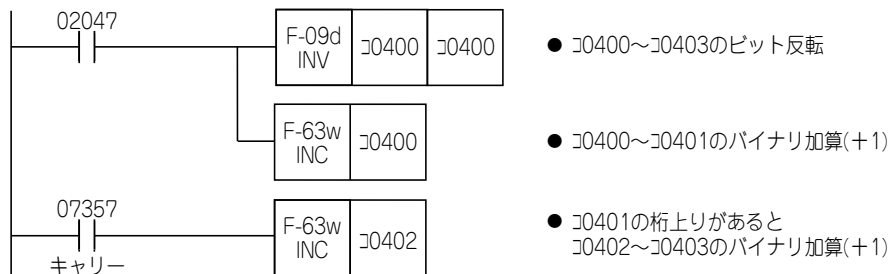
カウント値(連続値)

0400
0401
0402
0403

0404  
符号  (+ : OFF)  
           (- : ON)



**参考** 2の補数とは、データの全ビットを反転(0→1, 1→0)し、1を加えた値です。  
従ってF-57dは次の回路で実現できます。



# 第 3 章 特殊命令の使い方

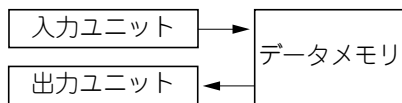
---

この章では、  
割込処理時に有効な命令や、特殊 I/O ユニット、通信ユニット使用時に使用する命令の使い方などについて説明します。

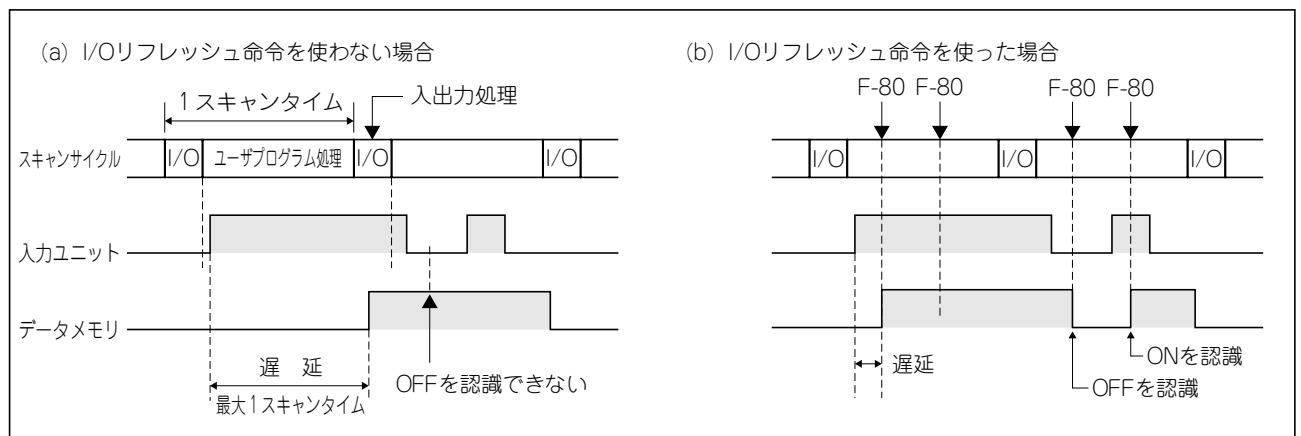
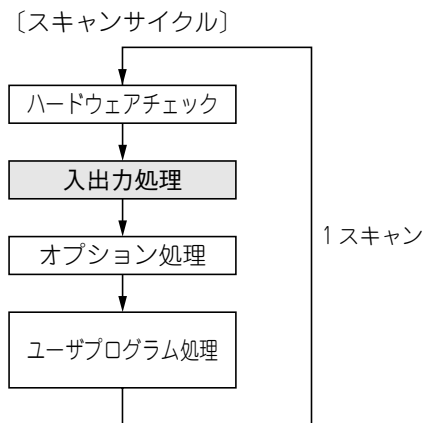
3-1	I/Oリフレッシュ命令と割込処理	3-1
3-2	特殊 I/O データリフレッシュ命令 (F-81)	3-5
3-3	特殊 I/O からの読出 (F-85)、特殊 I/O への書込 (F-86)	3-8
3-4	送信命令 (F-204)、受信命令 (F-205)	3-12
3-5	MD (メンテナンスディスプレイ) 命令 (F-20)	3-17

## 3-1 I/Oリフレッシュ命令と割込処理

- I/Oリフレッシュとは、PCの入出力ユニットとデータメモリ間で情報交換を行うことをいいます。



- 通常、I/Oリフレッシュ(入出力処理)は、1スキャンサイクルに1回行われます。従って、入出力ユニットとデータメモリ間の情報交換に最大1スキャンタイムの遅れが生じます。
- I/Oリフレッシュ命令を使用すると、その時点で入出力処理が行えますので、スキャンサイクルにとらわれない高速応答が可能です。



- I/Oリフレッシュ命令(F-80)はPC機種により次のように設定するデータが異なります。

PC機種名	シンボル	データ範囲	機能
JW10	F-80 IORF D	D 30000~30077	Dで指定される入出力部の1バイトデータのリフレッシュを行う。
	F-81 IORF n D	n 0~7 D 30000~30077	Dで指定される入出力部のnビット目の1ビットデータのリフレッシュを行う。
JW20H JW30H	F-80 IORF R-S	R 0~3(JW20H) 0~7(JW30H) S 0~7	R(ラック番号)とS(スロット番号)で指定される入出力ユニットの全データのリフレッシュ(最大32点)を行う。
JW50H JW70H JW100H	F-80 IORF R-S-B	R 0~7 S 0~F(H) (10進数では0~15) B 00~1F(H) (10進数では0~31)	R(ラック番号)とS(スロット番号)で指定される入出力ユニットのBバイト目の1バイトデータのリフレッシュを行う。

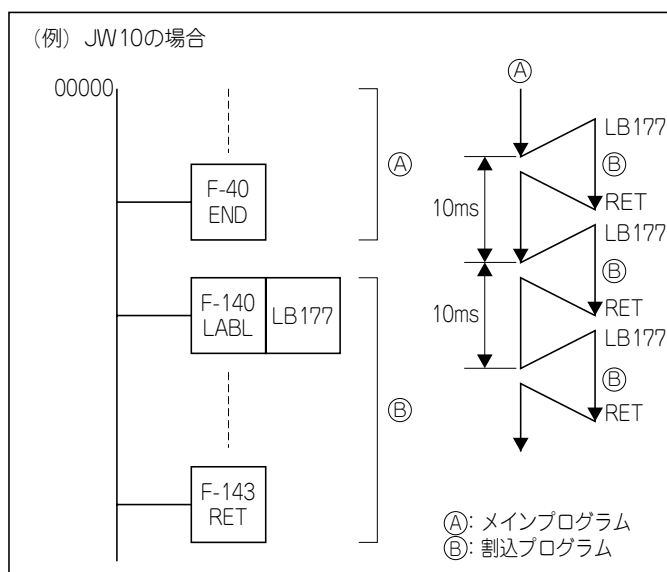


- I/Oリフレッシュ命令は割込機能と合わせて使用すると入出力の高速応答が実現できます。
- 割込にはタイマ割込と入力割込があります。

### [ 1 ] タイマ割込

- 割込周期(最小 1 ms)毎に割込ラベルで指定されるサブルーチンにプログラムの実行を移し、リターン命令(F-143)で割込前のプログラムに戻ります。
- 割込は、ユーザプログラム演算中だけでなく入出力処理中も実行します。

**注1** 割込プログラムは、割込周期以内の演算時間にしてください。



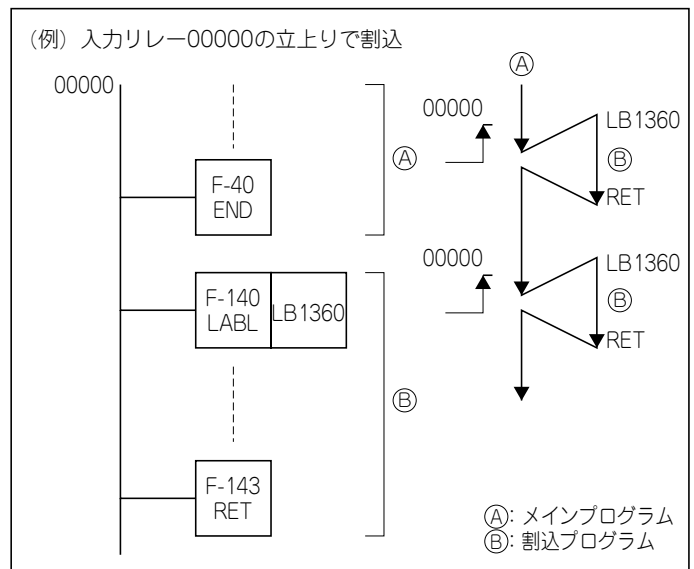
- タイマ割込の設定は、システムメモリへの設定で行います。

PC機種名	システムメモリの設定																				
JW10	#244=01 <sub>(H)</sub>	10ms割込	(ラベルLB177)																		
JW20H	#240	<table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>▨</td><td>▨</td><td>▨</td><td>▨</td><td>▨</td><td>▨</td><td>▨</td><td>▨</td> </tr> </table>	7	6	5	4	3	2	1	0	▨	▨	▨	▨	▨	▨	▨	▨	0: 割込なし    1: 割込あり 10ms割込 (ラベルLB1354) 20ms割込 (ラベルLB1355) 50ms割込 (ラベルLB1356) 100ms割込 (ラベルLB1357)	[優先度] 9 10 11 12	<b>注1</b>
7	6	5	4	3	2	1	0														
▨	▨	▨	▨	▨	▨	▨	▨														
JW30H JW50H JW70H JW100H	#240	<table border="1"> <tr> <td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>▨</td><td>▨</td><td>▨</td><td>▨</td><td>▨</td><td>▨</td><td>▨</td><td>▨</td> </tr> </table>	7	6	5	4	3	2	1	0	▨	▨	▨	▨	▨	▨	▨	▨	0: 割込なし    1: 割込あり 1ms割込 (ラベルLB1353) 2ms割込 (ラベルLB1354) 5ms割込 (ラベルLB1355) 10ms割込 (ラベルLB1356) 20ms割込 (ラベルLB1357)	[優先度] 17 18 19 20 21	<b>注1</b>
7	6	5	4	3	2	1	0														
▨	▨	▨	▨	▨	▨	▨	▨														

**注1** 複数の割込が同時に発生した場合は優先度の高い(数値の小さい)ものから処理します。  
 なお、優先度は、入力割込も含みます。

## 〔2〕 入力割込

- 指定された入力ユニットの入力信号の立上りまたは立下り時に割込ラベルで指定されたサブルーチンにプログラムの実行を移し、リターン命令(F-143)で割込前のプログラムに戻ります。
- 指定された入力信号のI/OリフレッシュはJW20Hの場合は10ms毎に、JW30H、JW50H/70H/100Hの場合は1ms毎に行います。



- 入力割込の設定は、システムメモリへの設定で行います。

PC機種名	システムメモリの設定																	
JW20H	● 割込用入力ユニットの設定	JW20H/30H : ラック番号 : 0~7(JW20Hは0~3)																
JW30H	#241	スロット番号 : 0~7																
JW50H		(#241=FF <sup>(H)</sup> で割込禁止)																
JW70H		JW50H/70H/100H : ラック番号 : 0~7, スロット番号 : 0~C																
JW100H		(#241=00 <sup>(H)</sup> で割込禁止)																
	● 割込条件(立下り/立上り)の設定																	
	#242	[優先度]																
	<table border="0"> <tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr> <tr><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr> </table>	7	6	5	4	3	2	1	0	□	□	□	□	□	□	□	□	入力0 (ラベルLB1360) 1 入力1 (ラベルLB1361) 2 入力2 (ラベルLB1362) 3 入力3 (ラベルLB1363) 4 入力4 (ラベルLB1364) 5 入力5 (ラベルLB1365) 6 入力6 (ラベルLB1366) 7 入力7 (ラベルLB1367) 8
7	6	5	4	3	2	1	0											
□	□	□	□	□	□	□	□											
	#243	[優先度]																
	<table border="0"> <tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr> <tr><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td><td>□</td></tr> </table>	7	6	5	4	3	2	1	0	□	□	□	□	□	□	□	□	入力10 (ラベルLB1370) 9 入力11 (ラベルLB1371) 10 入力12 (ラベルLB1372) 11 入力13 (ラベルLB1373) 12 入力14 (ラベルLB1374) 13 入力15 (ラベルLB1375) 14 入力16 (ラベルLB1376) 15 入力17 (ラベルLB1377) 16
7	6	5	4	3	2	1	0											
□	□	□	□	□	□	□	□											
	注1																	
		0 : 立下り(ON→OFF)で割込      1 : 立上り(OFF→ON)で割込																

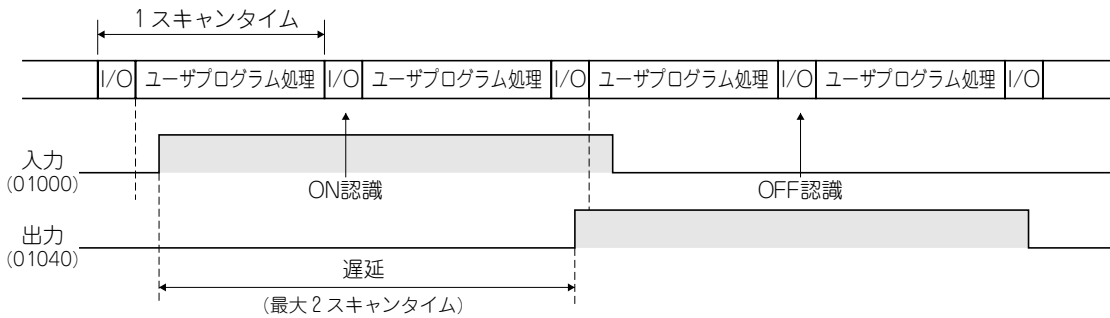
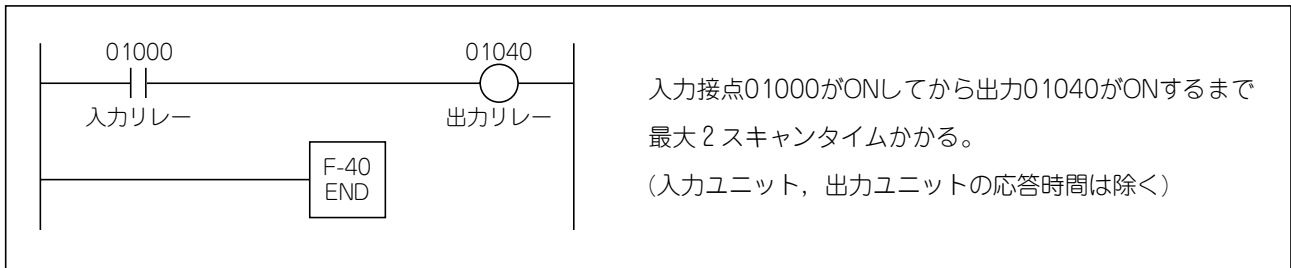
注1 JW20Hには#243の設定はありません。(8点のみ)

注2 JW20Hで16点、32点ユニットの場合は、前半8点、JW30H、JW50H/70H/100Hで32点ユニットの場合は、前半16点が割込入力として使用できます。

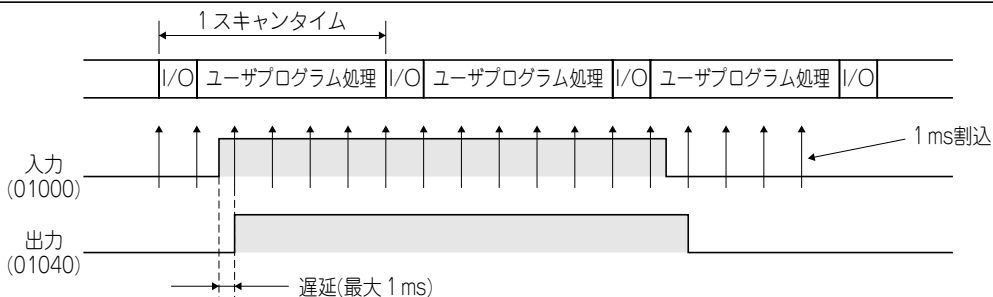
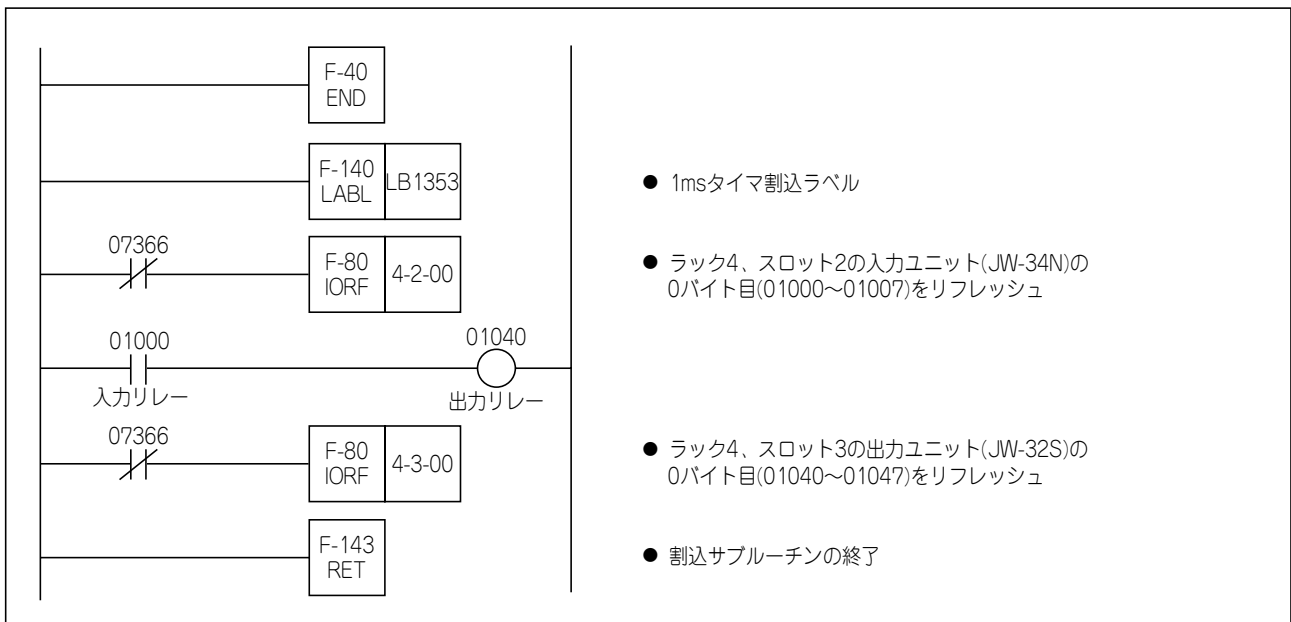
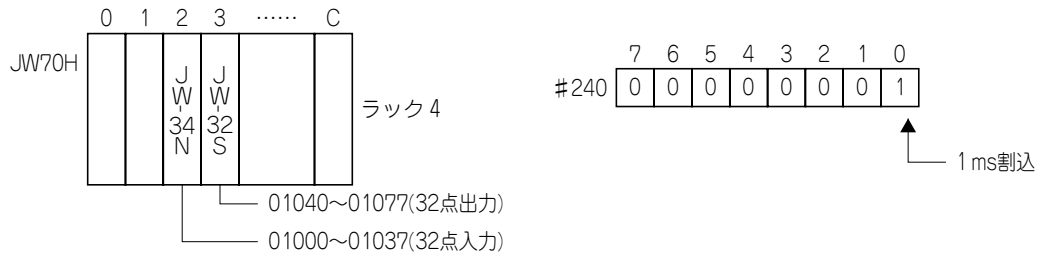
注3 複数の割込が同時に発生した場合は、優先度の高い(数値の小さい)ものから処理します。なお、優先度はタイマ割込も含まれます。

■ 入力ユニットで入力した情報を出力ユニットに出力する場合

(1) I/Oリフレッシュ命令を使わないとき



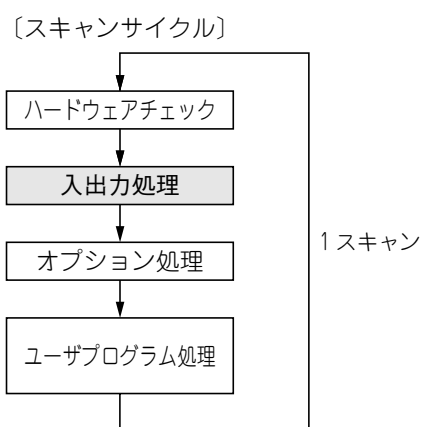
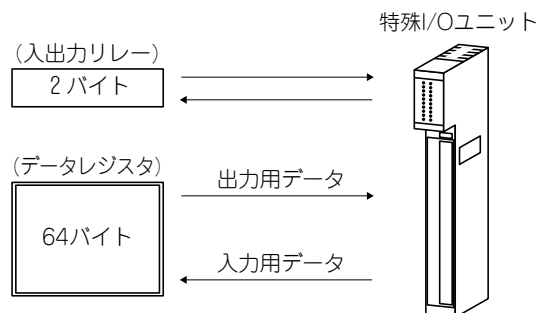
(2) I/Oリフレッシュ命令と1msタイマ割込を使用したとき



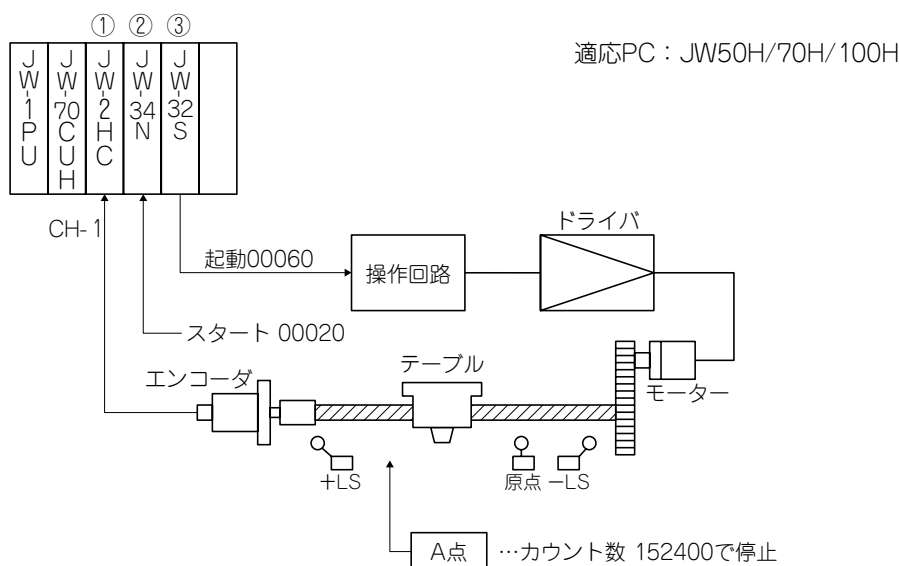
ただし、入力ユニット, 出力ユニットの応答時間は除く。

## 3-2 特殊I/Oデータリフレッシュ命令(F-81)

- JW50H/70H/100Hの高速カウンタユニット(JW-2 HC)等の特殊I/Oユニットは、入出力リレー2バイトとデータ用のデータレジスタの2つのデータメモリを使用します。
- 特殊I/Oユニットとデータレジスタ間のデータ変換(データリフレッシュ)は、スキャンサイクルの入出力処理で行われます。
- 通常、入出力処理は、1スキャンサイクルに1回行われます。従って、特殊I/Oユニットとデータレジスタ間のデータ交換に最大1スキャンタイムの遅れが生じます。
- データリフレッシュ命令(F-81)を使用すると、その時点で入出力処理が行えますのでスキャンサイクルにとられない高速応答が可能です。
- ここでは高速カウンタユニット(JW-2 HC)を使用し、データリフレッシュ命令とI/Oリフレッシュ命令、および割込処理を使用したプログラム例を示します。



### システム構成 1軸テーブルの位置決め



	ラックNo.	スロットNo.	入出力リレー	データレジスタ	
①	JW-2 HC	0	2	00000~00017	49000~49007
②	JW-34N	0	3	00020~00057	—
③	JW-32S	0	4	00060~00117	—

## JW-2HCのデータメモリ

### (1) 入出カリレー

30000	ORG	HLS (Z)	B	A	ORG	HLS (Z)	B	A
30001	ORG RST	Z SELECT	C-DIS	C-RST	ORG RST	Z SELECT	C-DIS	C-RST
	CH-2				CH-1			

### ● CH-1のリレー内容

リレー名称	リレー番号	内 容	入出力
A	00000	A相	入力
B	00001	B相	入力
HLS(Z)	00002	ホームポジションLS	入力
ORG	00003	原点	入力
C-RST	00010	カウンタリセット	出力
C-DIS	00011	カウント停止	出力
Z-SELECT	00012	HLS表示のかわりにZ相表示	出力
ORG RST	00013	ORGリセット	出力

### (2) データレジスタ

49000	CH-1 カウント値
49001	
49002	
49003	
49004	CH-2 カウント値
49005	
49006	
49007	

(49010~49077は未使用)

### ● CH-1のデータ内容

	7	6	5	4	3	2	1	0
49000	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
49001	2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>
49002	2 <sup>23</sup>	2 <sup>22</sup>	2 <sup>21</sup>	2 <sup>20</sup>	2 <sup>19</sup>	2 <sup>18</sup>	2 <sup>17</sup>	2 <sup>16</sup>
49003	0	0	0	0	0	0	0	0

カウント値は24ビットバイナリ値(0~16777215)

**注1** 入出カリレーとデータレジスタのアドレスはI/O登録により決まります。  
本例は、自動I/O登録を行った場合です。

## システムメモリ

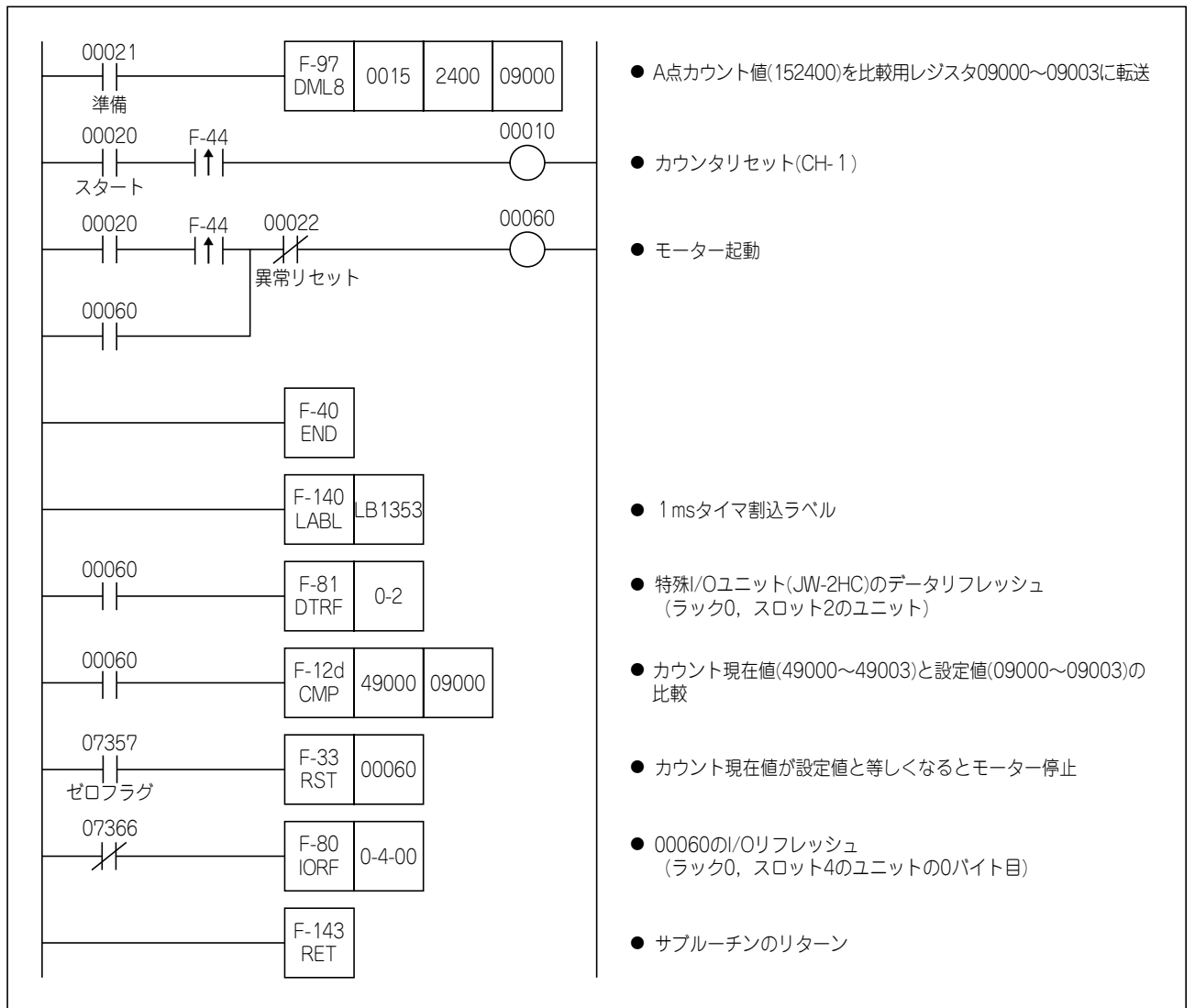
	7	6	5	4	3	2	1	0
#240	0	0	0	0	0	0	0	1

└── 1msタイマ割込

## プログラム

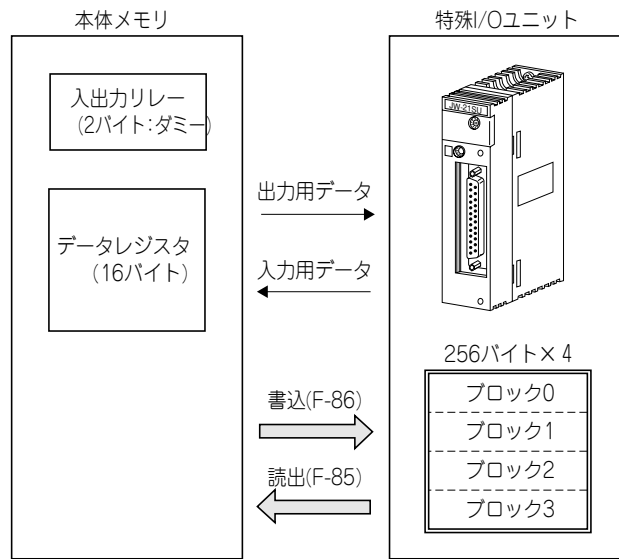
- カウント値が設定値(A点)と等しくなるとモータを停止します。
- カウント値の読み出し、モータ停止はデータリフレッシュ命令(F-81)、I/Oリフレッシュ命令(F-80)と1msタイマ割込を使用して行っています。

**注2** I/Oリフレッシュ命令(F-80)とタイマ割込については「3-1 I/Oリフレッシュ命令と割込処理」を参照してください。

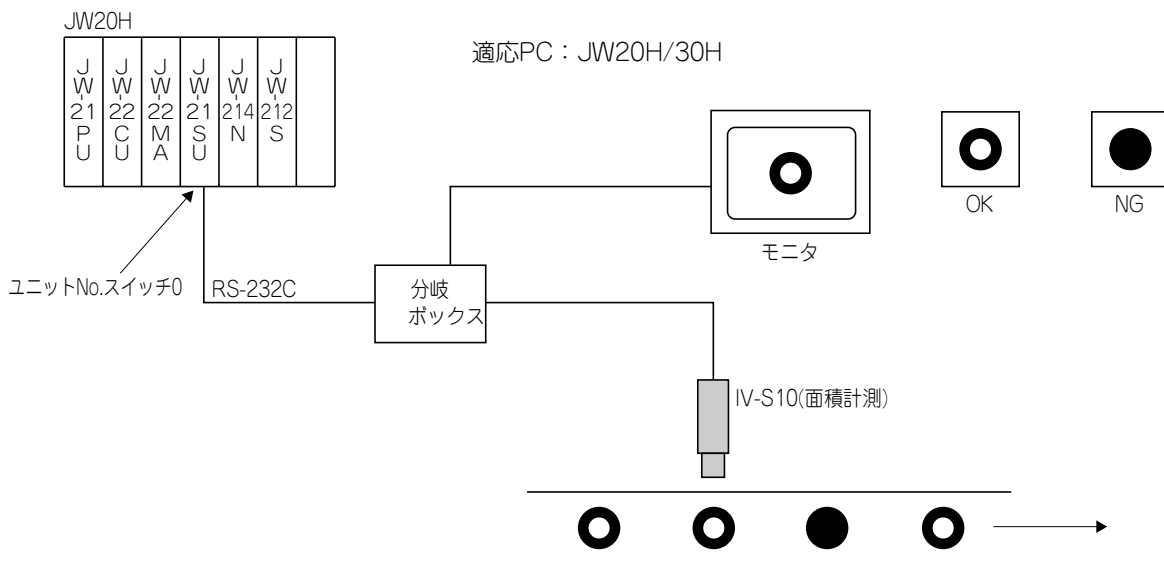


### 3 - 3 特殊I/Oからの読出(F-85)、特殊I/Oへの書込(F-86)

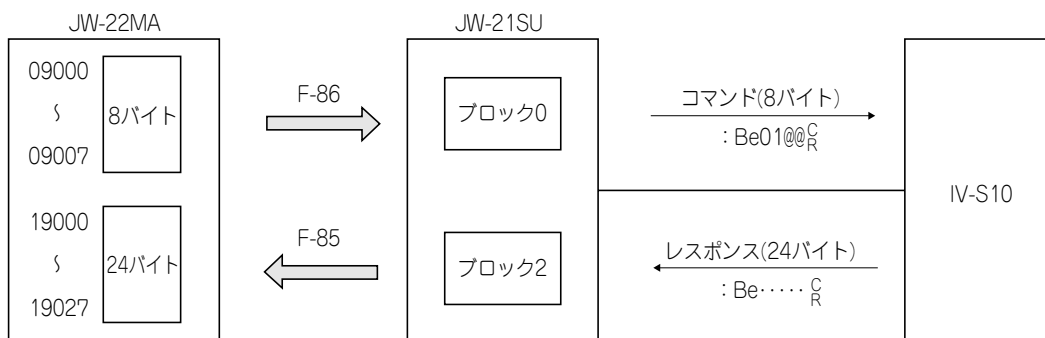
- JW20H/30Hのシリアルインターフェイスユニット (JW-21SU)等の特殊I/Oユニットは、入出力リレー2バイトとデータ用のデータレジスタの2つのデータメモリに加え、特殊I/Oユニット内部メモリを使用します。
- 特殊I/Oユニットの内部メモリは256バイトを1ブロックとするメモリで4ブロックあります。
- 特殊I/Oユニットの内部メモリと本体メモリ間のデータ変換を行う命令がF-85、F-86です。
- ここでは、シリアルインターフェイスユニット(JW-21SU)を使用し、小型画像センサカメラ(IV-S10)とのデータ通信を行うプログラム例を示します。

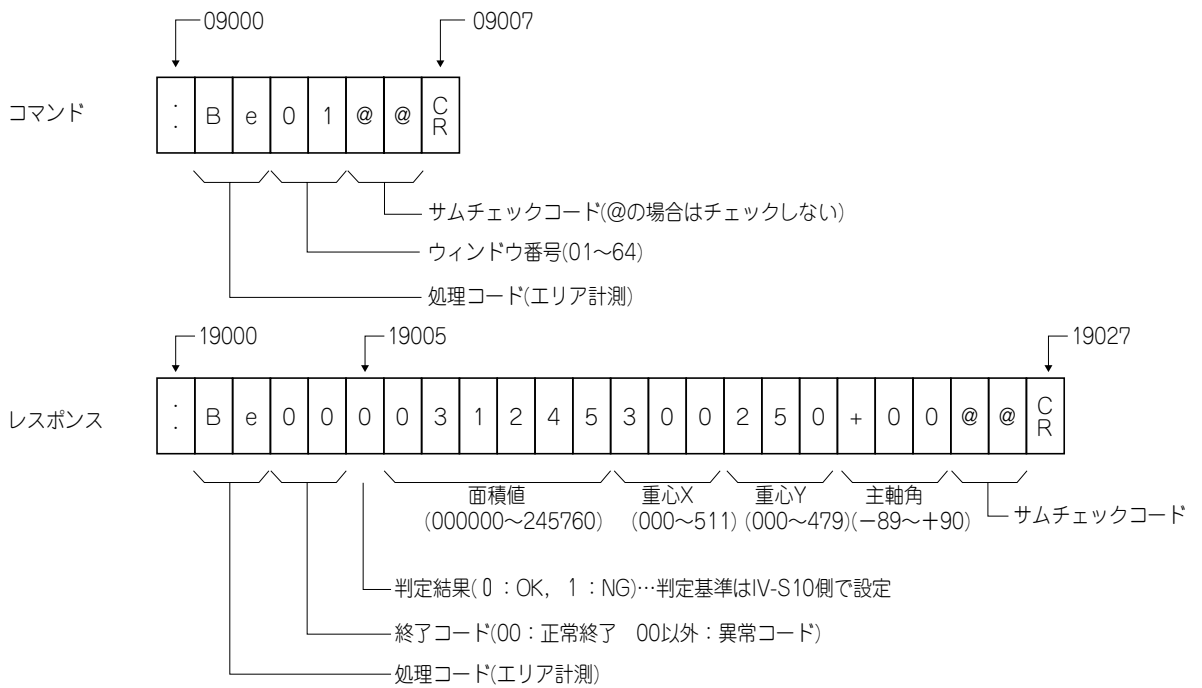


#### システム構成 IV-S10による面積計測



#### データの流れ



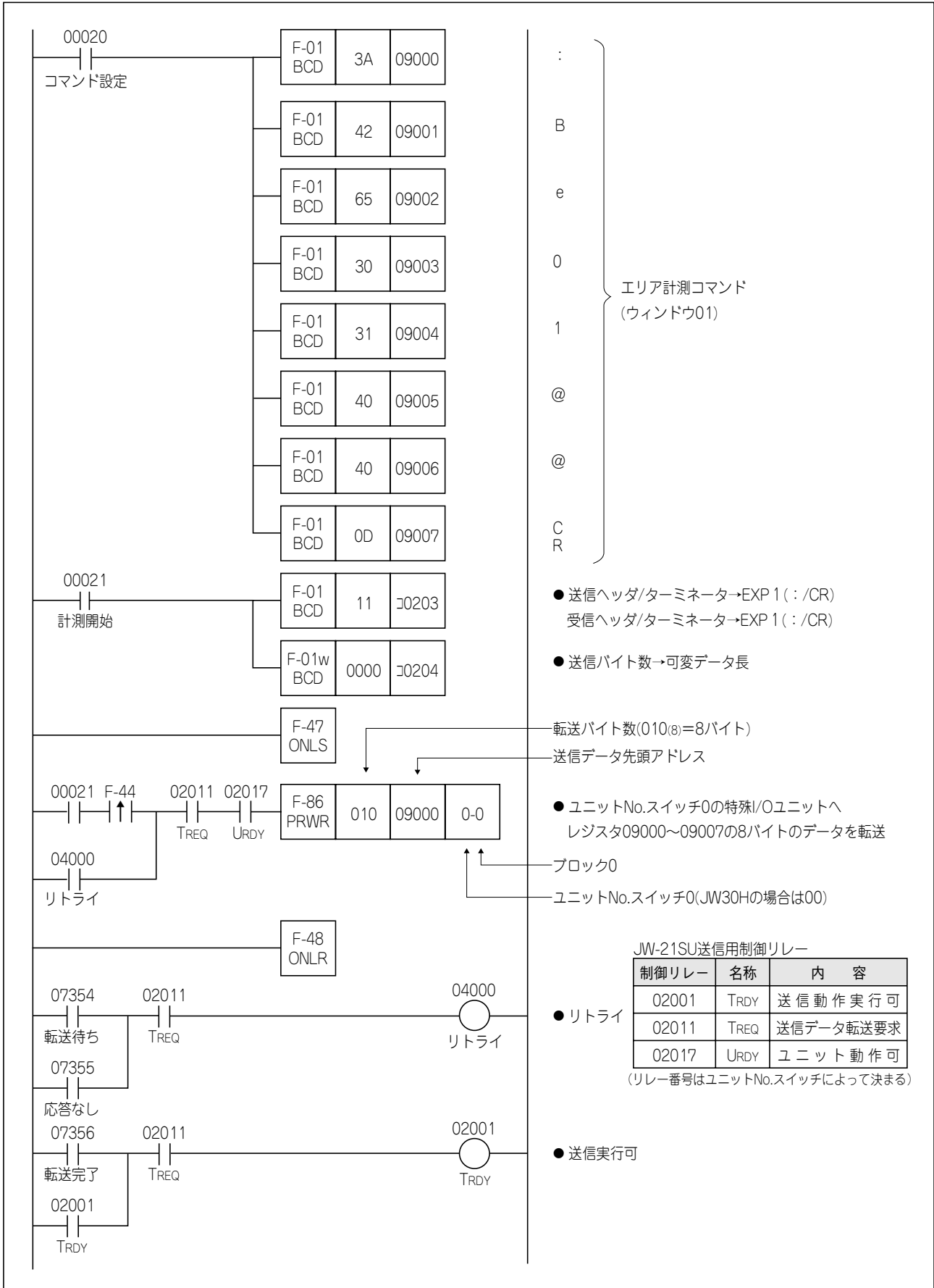


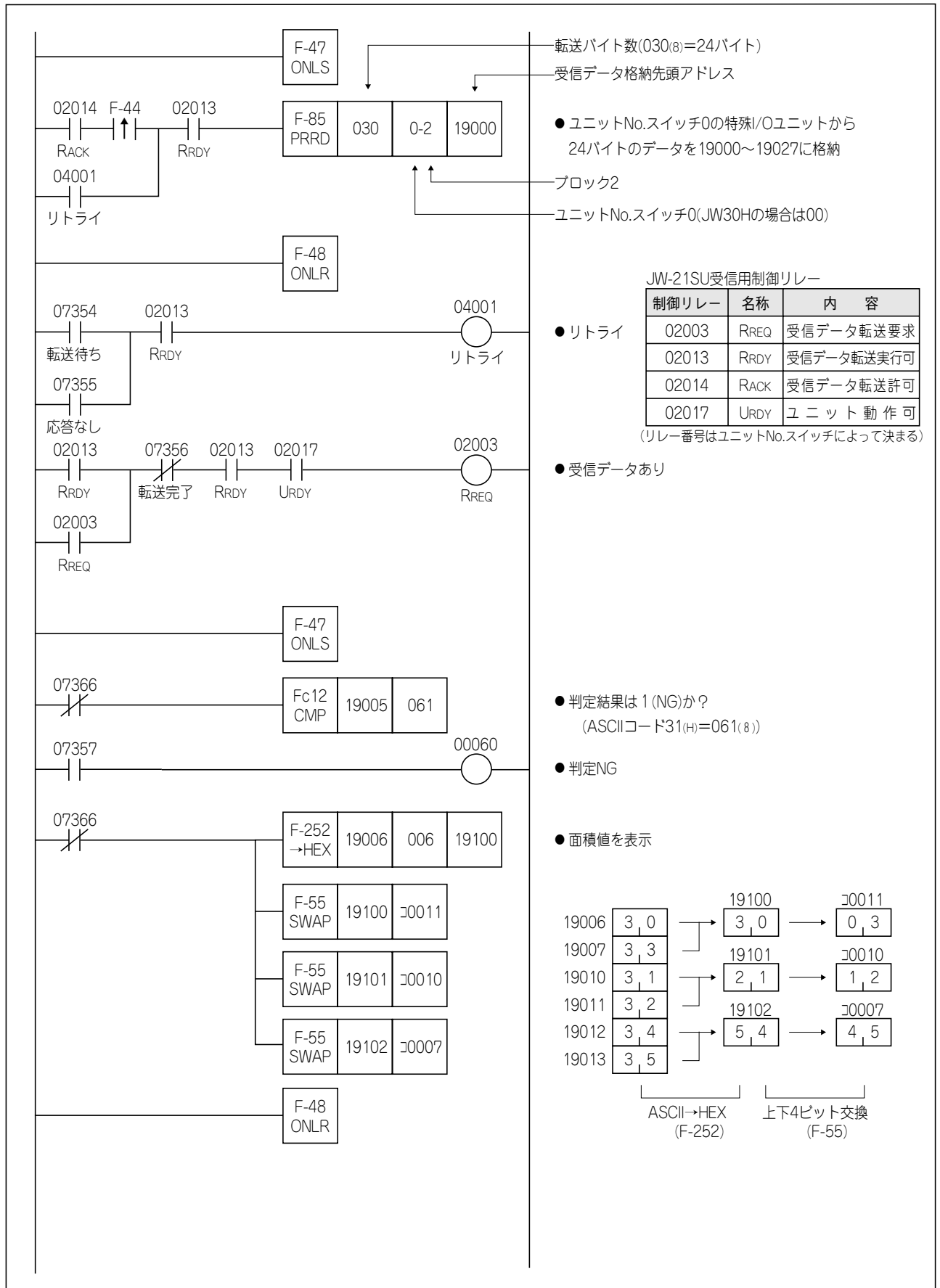
JW-21SUパラメータ

アドレス	パラメータ名	設定値(H)	内容	IV-S10の設定
000	パラメータ転送	22	実行	
001	BCC計算	01	実行	
002	通信方式	00	RS-232C	有
003	伝送方式	00	全二重	
004	伝送速度	01	9600ビット/s	有
005	データ長	01	7ビット	有
006	ストップビット	00	2ビット	有
007	パリティ	00	偶数	有
010	制御信号	00	なし	
011	伝送コード変換	00	無変換	
012	送信タイムアウト時間	01	100ms	
013	受信タイムアウト時間	01	100ms	
014	EXP 1 ヘッダ	3A	: (コロン)	
015	〃	00		
016	〃	00		
017	〃	00		
020	EXP 1 ターミネータ	0D	CR	
021	〃	00		
022				
〃	設定不要			
177				



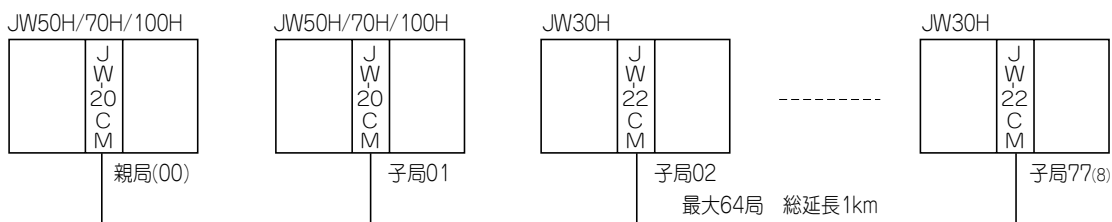
プログラム





## 3 - 4 送信命令(F-204)、受信命令(F-205)

- サテライトネットでPC間を接続すると、最大64局のPC間でのデータリンクが可能です。

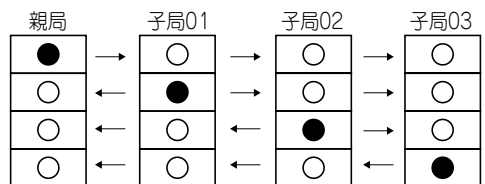


- データリンクには次の2つの通信方式があります。

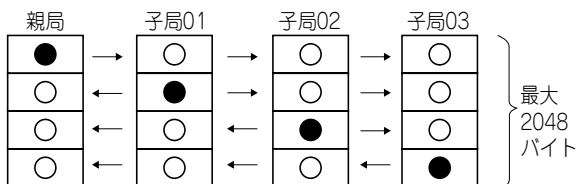
### 1) リレーリンク/レジスタリンク

- ・ 親局のパラメータに設定したリンク領域にて常時全局間で通信します。  
(プログラムレスリンク)

#### ①リレーリンク



#### ②レジスタリンク

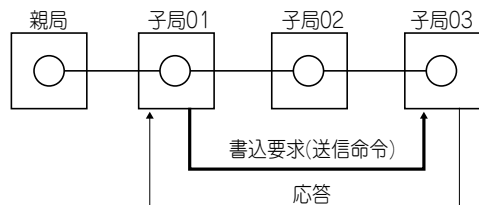


● : 送信 ○ : 受信

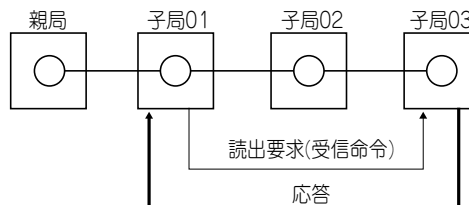
### 2) SEND/RECEIVE機能

- ・ SEND(送信)命令、RECEIVE(受信)命令を使用し、通信相手局との間で送受信したいときにのみ通信します。

#### ①SEND機能



#### ②RECEIVE機能



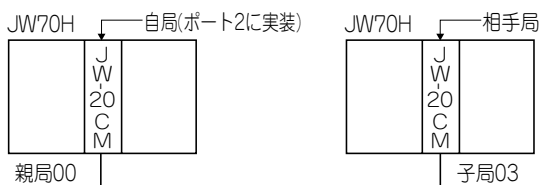
- ここでは、送信命令(F-204)、受信命令(F-205)を使用したプログラム例を示します。

適応PC

JW30H (JW-22CM実装)  
JW50H/70H/100H (JW-20CM実装)

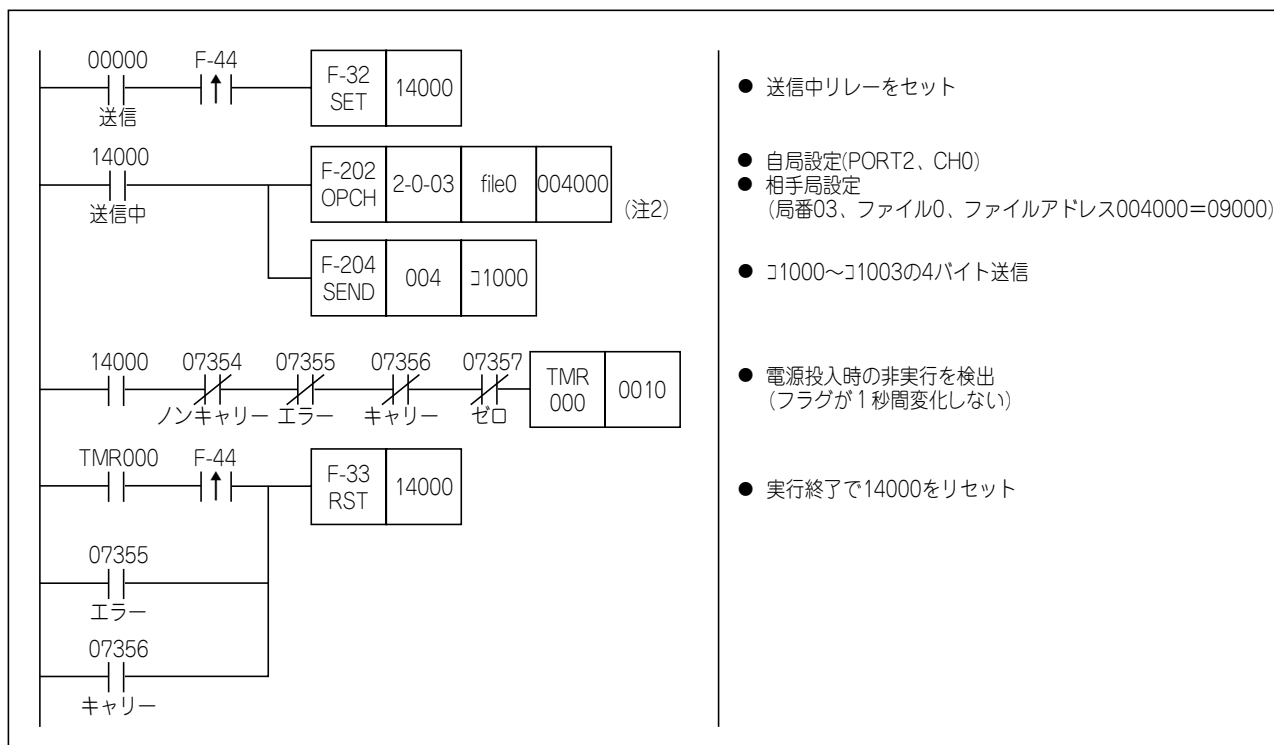
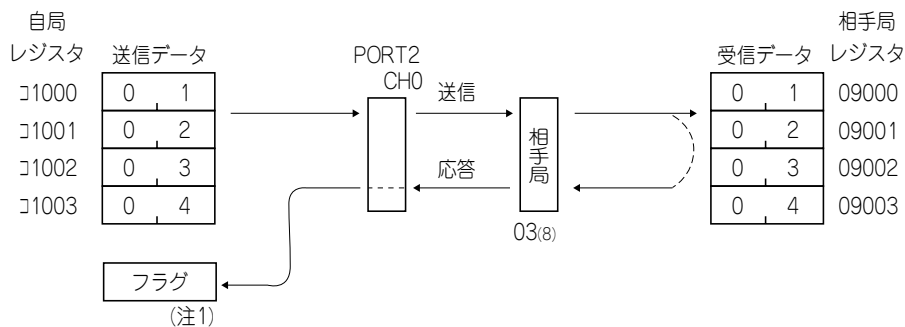
# 〔1〕送信命令(F-204)使用例

## (1) 1階層通信の場合



	JW30H	JW50H/70H/100H
自局	JW-22CM	JW-20CM
相手局		

- 子局03へ4バイトのデータを書き込みます。

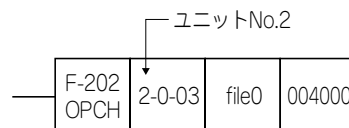


- 送信中リレーをセット
- 自局設定(PORT2、CH0)
- 相手局設定 (局番03、ファイル0、ファイルアドレス004000=09000)
- 01000~01003の4バイト送信
- 電源投入時の非実行を検出 (フラグが1秒間変化しない)
- 実行終了で14000をリセット

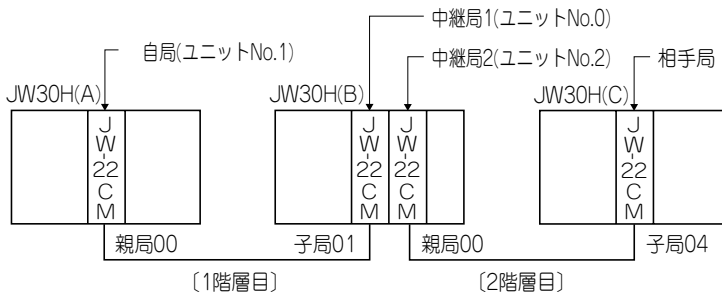
〔注1〕 F-204実行後、フラグは次のようになります。

通信内容	フラグ	ゼロ	キャリア	エラー	ノンキャリア
		07357	07356	07355	07354
ポートからの応答なし		0	0	1	0
通信渋滞		0	0	0	1
通信中		1	0	0	1
正常終了		0	1	0	0
異常終了		0	1	1	0
相手局書込禁止		1	1	1	0

〔注2〕 自局がJW30Hの場合、F-202にはユニットNo. (0~6)を設定します。

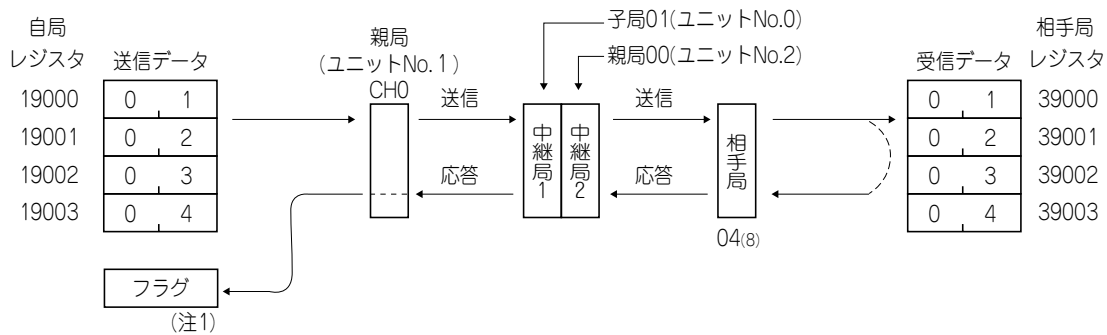


(2) 2階層通信の場合(自局はJW30Hのみ)



	JW30H	JW50H/70H/100H
自局	JW-22CM	—
中継局1		
中継局2	JW-22CM	JW-20CM
相手局		

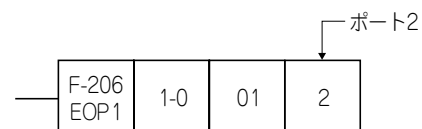
- JW30H(A)からJW30H(B)を経由してJW30H(C)に4バイトのデータを書き込みます。



- 送信中リレーをセット
- 自局の設定(ユニットNo.1、CHO)
- 中継局1の設定(局番01) ● 中継局2の設定(ユニットNo.2)
- 相手局の設定  
(局番04、ファイル0、ファイルアドレス006000=39000)
- 19000~19003の4バイトを送信
- 電源投入時の非実行を検出  
(フラグが1秒間変化しない)
- 実行終了で14002をリセット

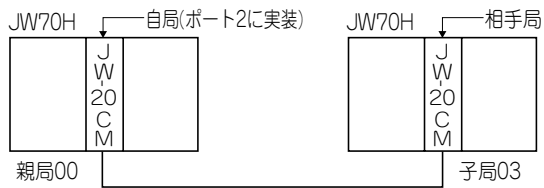
注1 F-204実行後のフラグの状態は1階層通信の場合と同じです。

注2 中継局がJW50H/70H/100Hの場合、F-206には、ポートNo.(2~7)を設定します。



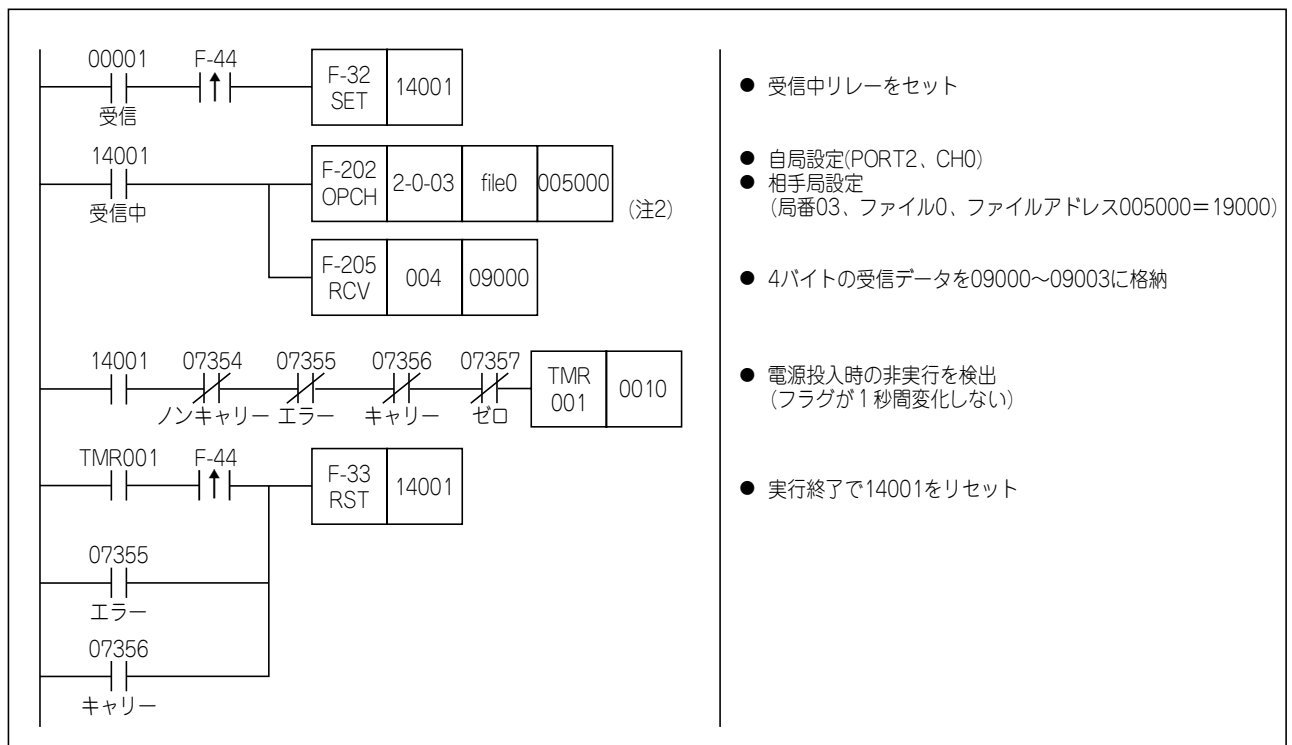
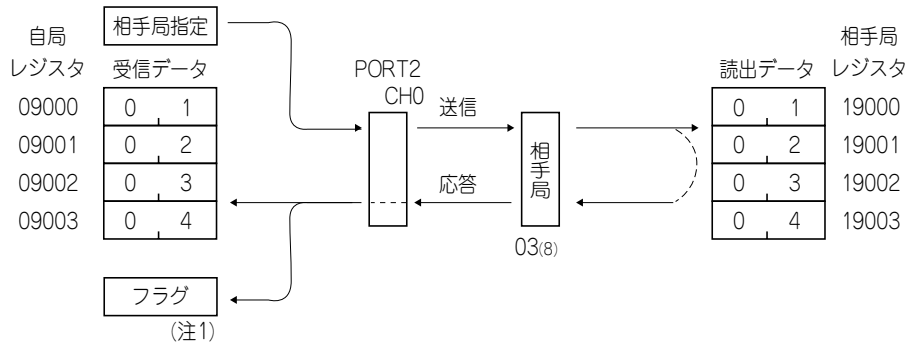
## 〔2〕受信命令(F-205)使用例

### (1) 1階層通信の場合



	JW30H	JW50H/70H/100H
自局	JW-22CM	JW-20CM
相手局		

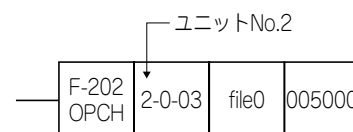
- 子局03から4バイトのデータを読み出します。



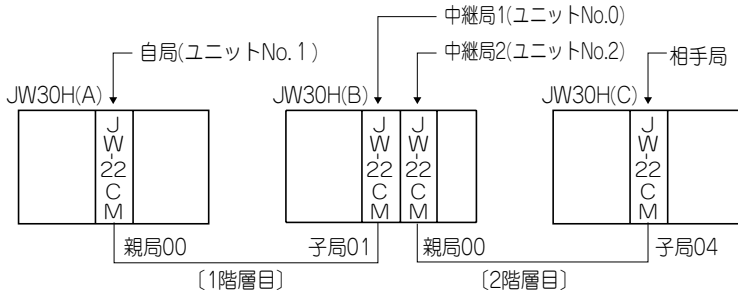
〔注1〕 F-205実行後、フラグは次のようになります。

通信内容	フラグ	ゼロ	キャリア	エラー	ノンキャリア
		07357	07356	07355	07354
ポートからの応答なし		0	0	1	0
通信渋滞		0	0	0	1
通信中		1	0	0	1
正常終了		0	1	0	0
異常終了		0	1	1	0

〔注2〕 自局がJW30Hの場合、F-202にはユニットNo. (0~6)を設定します。

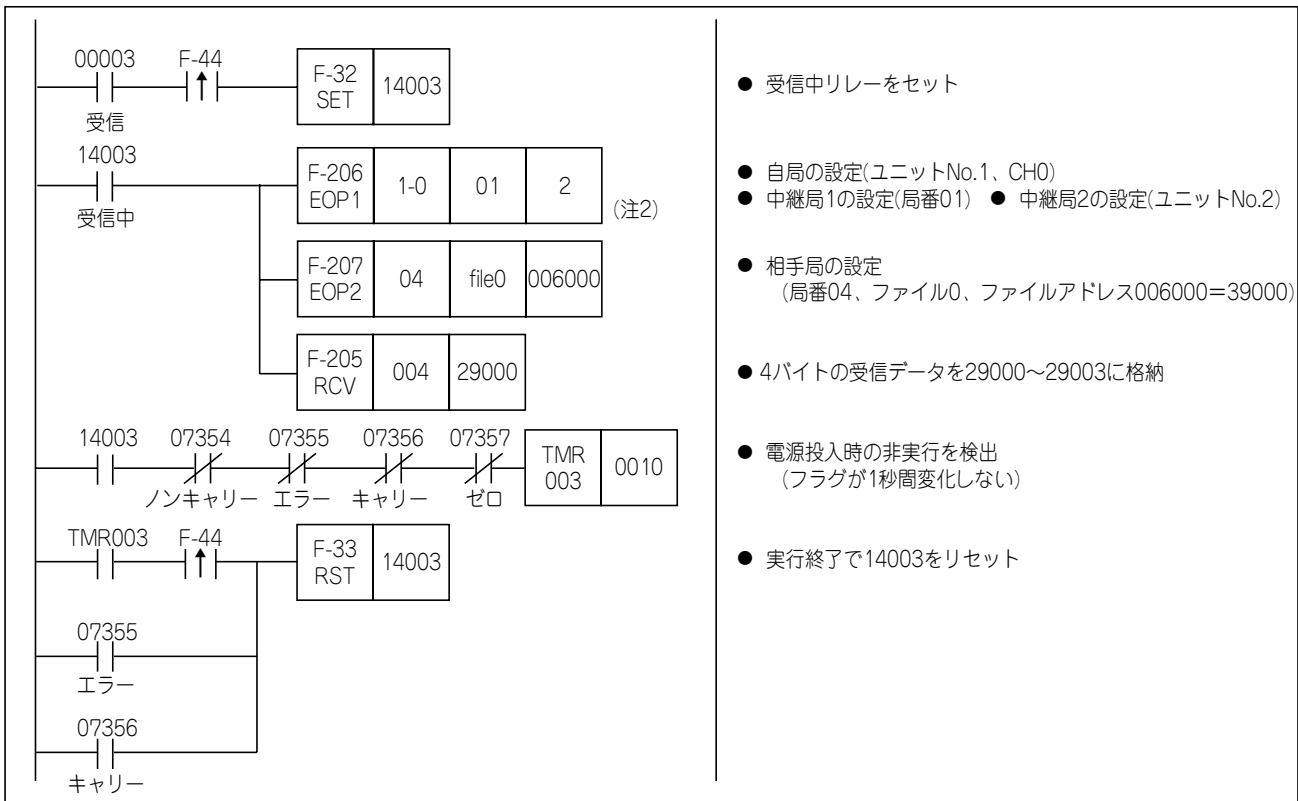
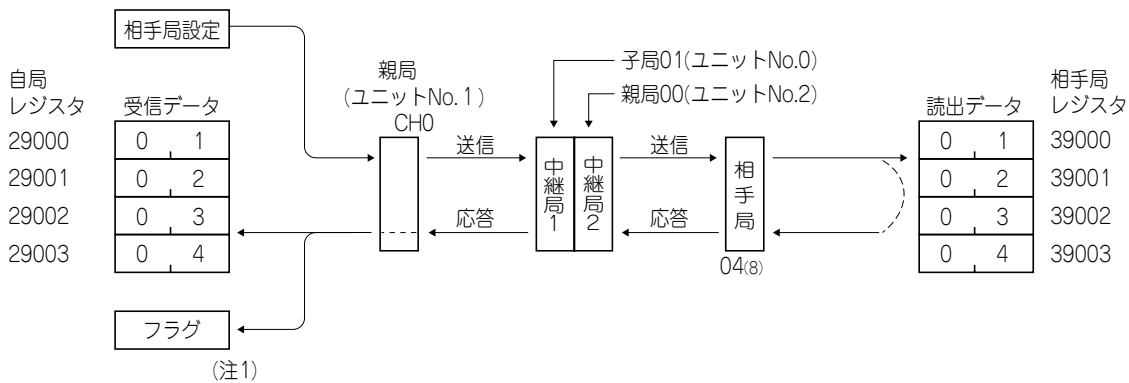


(2) 2階層通信の場合(自局はJW30Hのみ)



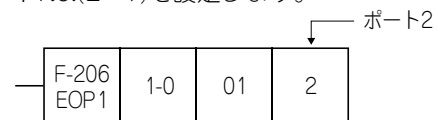
	JW30H	JW50H/70H/100H
自局	JW-22CM	—
中継局1	JW-22CM	JW-20CM
中継局2		
相手局		

- JW30H(A)からJW30H(B)を経由してJW30H(C)から4バイトのデータを読み出します。



注1 F-205実行後のフラグの状態は1階層通信の場合と同じです。

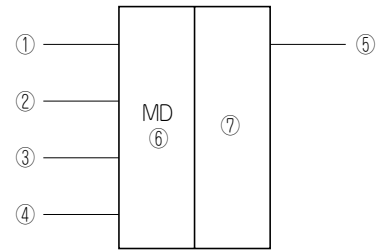
注2 中継局がJW50H/70H/100Hの場合 F-206には、



## 3-5 MD(メンテナンスディスプレイ)命令(F-20)

- 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をオンラインで監視するとともに、万一、被制御機器に故障が発生した場合、故障原因の発見のヒントとなる情報を、外部に出力し、故障原因の追求を容易にするものです。

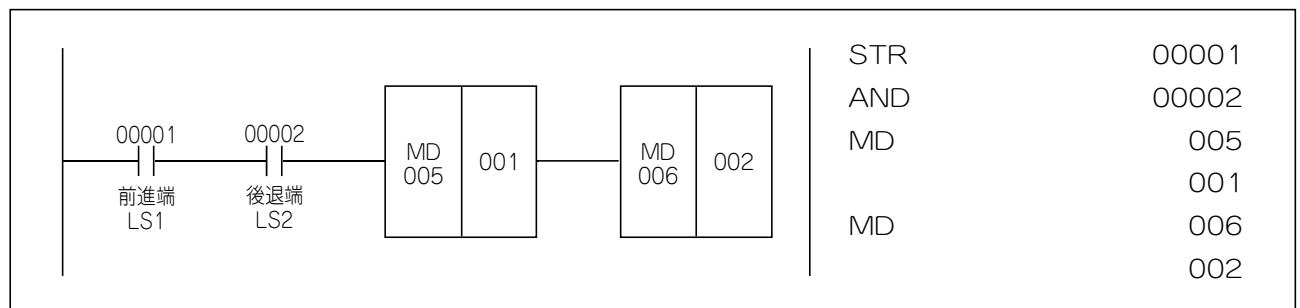
適応PC JW20H  
 JW30H  
 JW50H/70H/100H



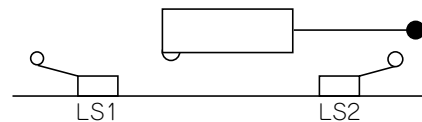
- ① } ⑥MD番号(000~777)
- ② } 入力情報 ⑦MDデータ(000~999)
- ③ } (00000~01576)
- ④ 出力指示
- ⑤ 拡張出力

### [1] 間欠故障のMD表示

(例1) 排他的入力の異常検知

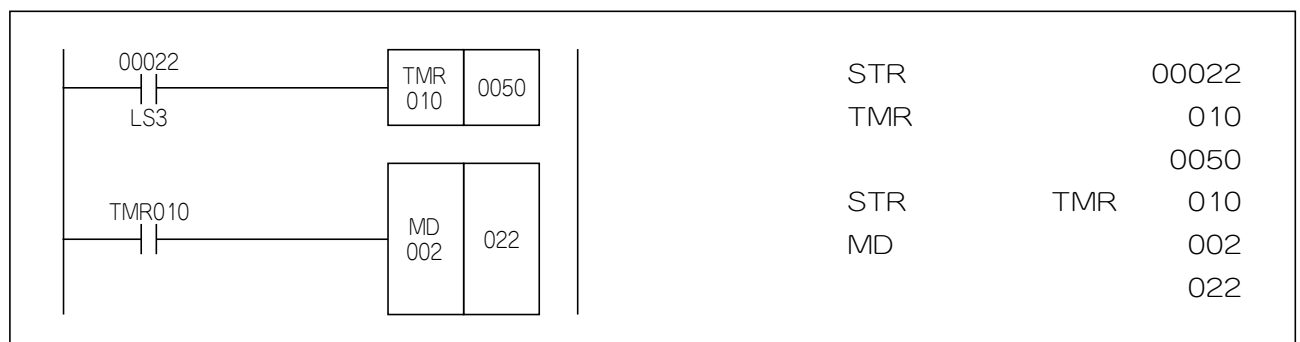


- 前進端リミットスイッチLS1(00001)と後退端リミットスイッチLS2(00002)が同時にONしない条件で、どちらかが溶着等の事故で同時にONした場合に記憶表示します。
- このプログラムでは、MDを2段に拡張して使用していますが、これは出力指示端子00001を出力データ001に、00002を002にそれぞれ対応させ、該当するリ



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、入力情報①, ②, ③は使用していません。

(例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(00022)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、入力情報①, ②, ③, 拡張出力⑤は使用していません。



## [2] サイクル番号のMD表示

- サイクル番号とサイクルの状態(動作指令、動作開始確認、動作終了確認)をMD表示することにより、被制御機器の異常時(シーケンスの停止)に、異常原因の推定が容易になり、装置のダウンタイムを大巾に減少させることができます。

### (例1) 直列動作のMD表示

- 図1のサイクル線図を例にとり、並行動作のないシーケンスにおけるシフトレジスタ、MD表示のプログラムについて説明します。

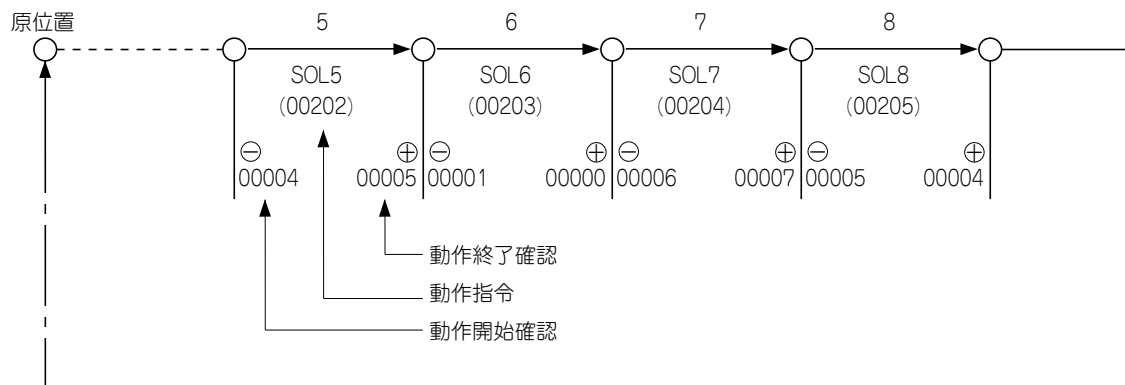
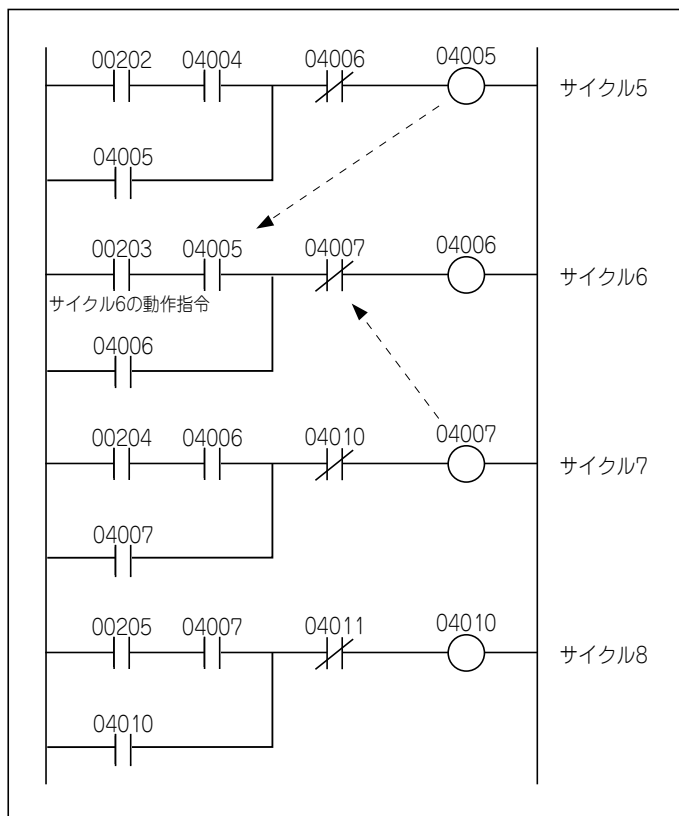


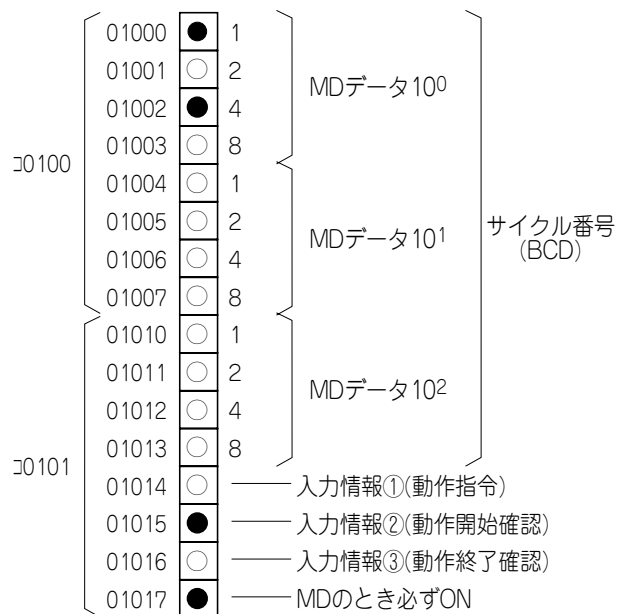
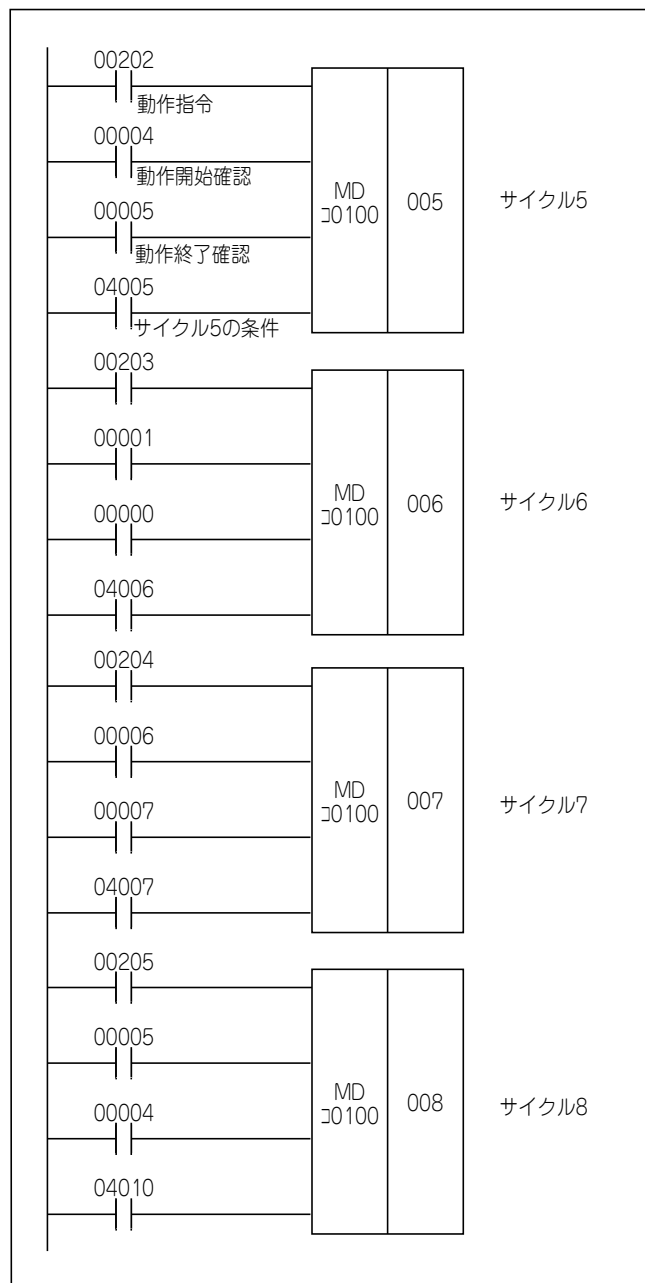
図1 サイクル線図(直列動作)

### (シフトレジスタのプログラム)



- サイクル6へ進む条件としては、サイクル5が実行されて(補助リレー04005がON)、サイクル6の動作指令(00203)がONになることです。
- サイクル6へ進む結果として、サイクル6の条件(補助リレー04006)がONになり、次の演算サイクルで、前のサイクル5の条件(04005)がリセットされます。
- 従って、シーケンスがサイクル5→6→7→8と進むに伴って、シフトレジスタが動作します。

(MD表示のプログラム)



サイクル	-----	5	6	7	8	-----
シフトレジスタ	-----	04005	04006	04007	04010	-----
MDデータ(BCD)	-----	005	006	007	008	-----
動作指令	-----	⊕00202	⊕00203	⊕00204	⊕00205	-----
動作開始確認	-----	⊖00004	⊖00001	⊖00006	⊖00005	-----
動作終了確認	-----	⊕00005	⊕00000	⊕00007	⊕00004	-----

- 正常動作では、サイクルの動き(→5→6→7→8→)に従って、シフトレジスタのON状態が移動し(→04005→04006→04007→04010→)、MDの表示内容も上表のように変化します。

(故障診断例)

MD表示 リレー番号	状態No. 表示内容	1					2					3					4					5								
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5				
01000 ┆ 01013	サイクル番号 (BCD)	005					006					007					008													
01014	動作指令	○	●	●	●	○	○	○	●	●	●	○	○	○	●	●	●	○	○	○	●	●	●	○	○	○	●	●	●	○
01015	動作開始確認	●	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
01016	動作終了確認	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
01017	MD	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

○ 消灯 ● 点灯

- 正常動作では、状態No.1→5のようにサイクル番号と、入力情報の表示を行い、次々と表示内容が変わっていきます。
- 被制御装置に異常が生じ、シーケンスの正常動作を行わない場合、例えば、サイクル番号005を表示し、入力情報の表示がNo.3の状態であったとすれば、故障の原因としては、次の事項が推定できます。

- 1) 動作指令がSOL5(00202)に届いていない。 → SOL5への配線系統不良、ヒューズ断 他
- 2) SOL5の動作不良
- 3) SOL5が正常に動作したが、動作終了確認ができない。 → LS5(00005)の不良、LS5からの配線系統不良 他

(例2) 並行動作のMD表示

- 図2に示す並行動作のMD表示には、仮想サイクルをソフトレジスタにおいて作成する必要があります。仮想サイクルを設定しない場合、故障の箇所によっては、正しい情報が得られなくなりますので注意してください。

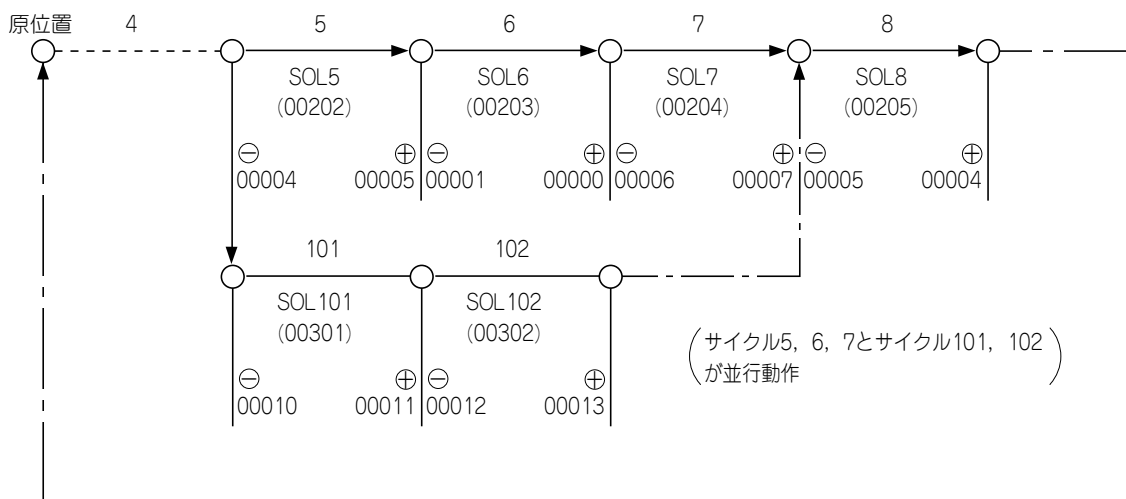
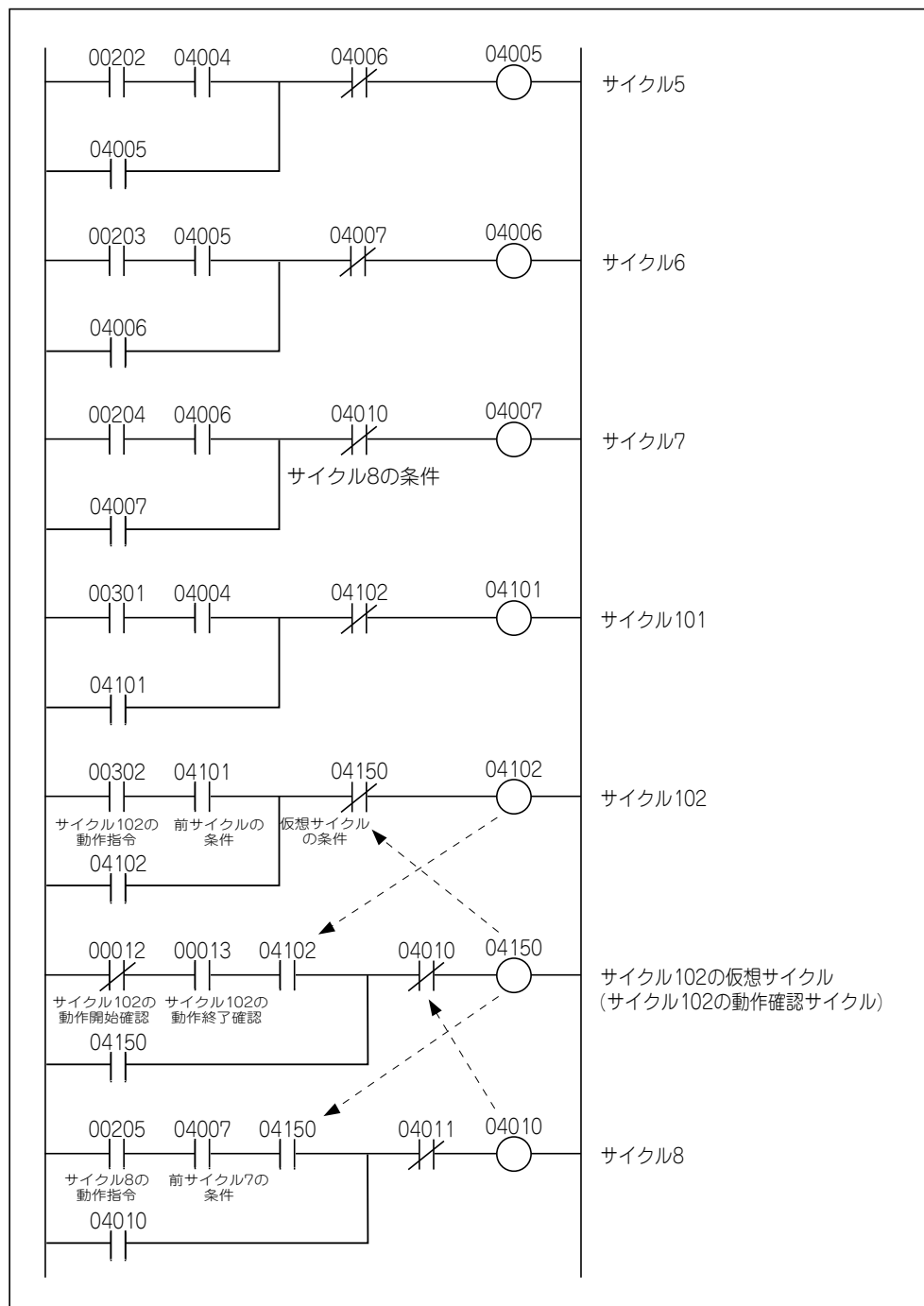


図2 サイクル線図(並行動作)

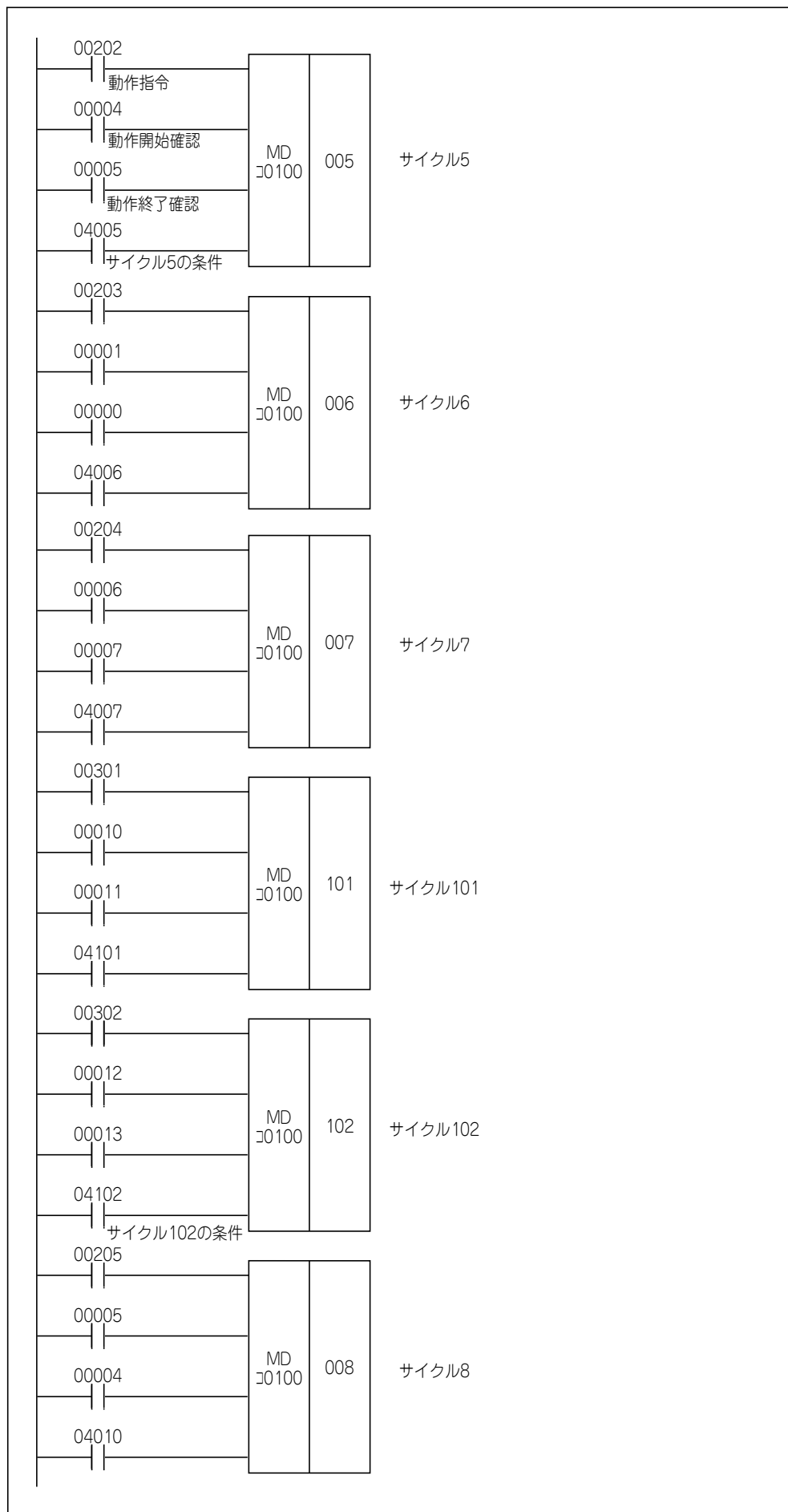
(シフトレジスタのプログラム)



- 仮想サイクルとは、並行動作の最終サイクルにおいて必要となるもので、並行動作の一連のサイクルの終了を確認するためのものです。
- サイクル102とサイクル102の仮想サイクルのラダー図に示すように、サイクル102へ進む(動作指令00302がON、前のサイクル101の条件04101がON)と、シフトレジスタ04102がONとなり、次にサイクル102の動作が完了すると、仮想サイクルの条件04150がONとなり、その結果として、サイクル102の条件04102がリセットされます。従って、並行動作101、102の動作が完了した時点においては、101、102の条件をMD表示するための条件04101、04102はともにOFFとなります。

- 並行動作5、6、7と101、102は、それぞれ前のサイクル4の条件がONであれば、独立に実行されます。従って、サイクル6で異常が発生し、サイクル7へ進まなくても、並行動作の101、102については、正常動作を行います。
- 仮想サイクルを設定しなかった場合、サイクル5、6、7のシフトレジスタは異常サイクル6でストップしますが、並行動作の101、102のシフトレジスタの中でサイクル102の条件04102もONとなります(MDの出力指示で04006と04102がON)。このとき、MD表示はプログラムの最終のほうが無効となるため、異常サイクルの6ではなく、正常サイクルの102を表示します。
- 仮想サイクルを設定すると、異常がサイクル6で起きても、サイクル102が完了した時点で、条件04102がリセットされ、MD表示には、異常サイクル6の条件04006のみONとなり、正しい情報を表示します。

(MD表示のプログラム)



# 第4章 応用プログラム事例

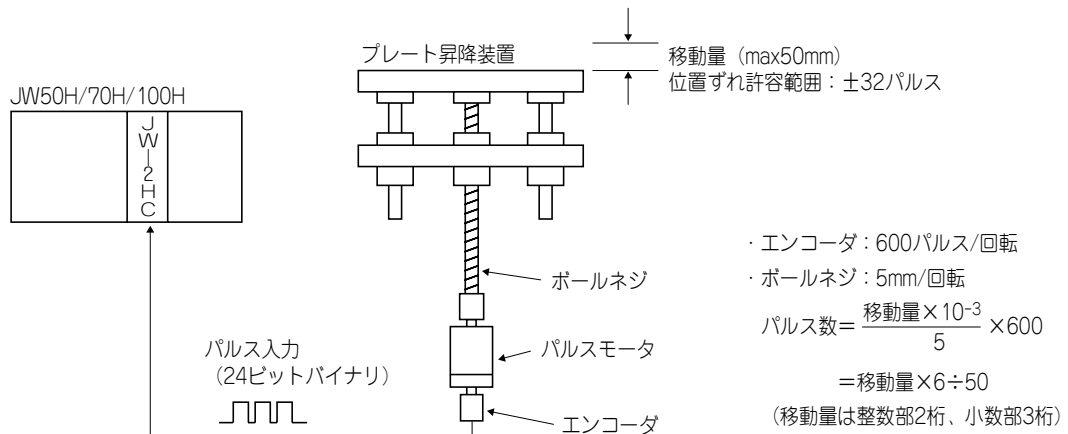
---

この章では、  
具体的な応用プログラム事例について紹介します。

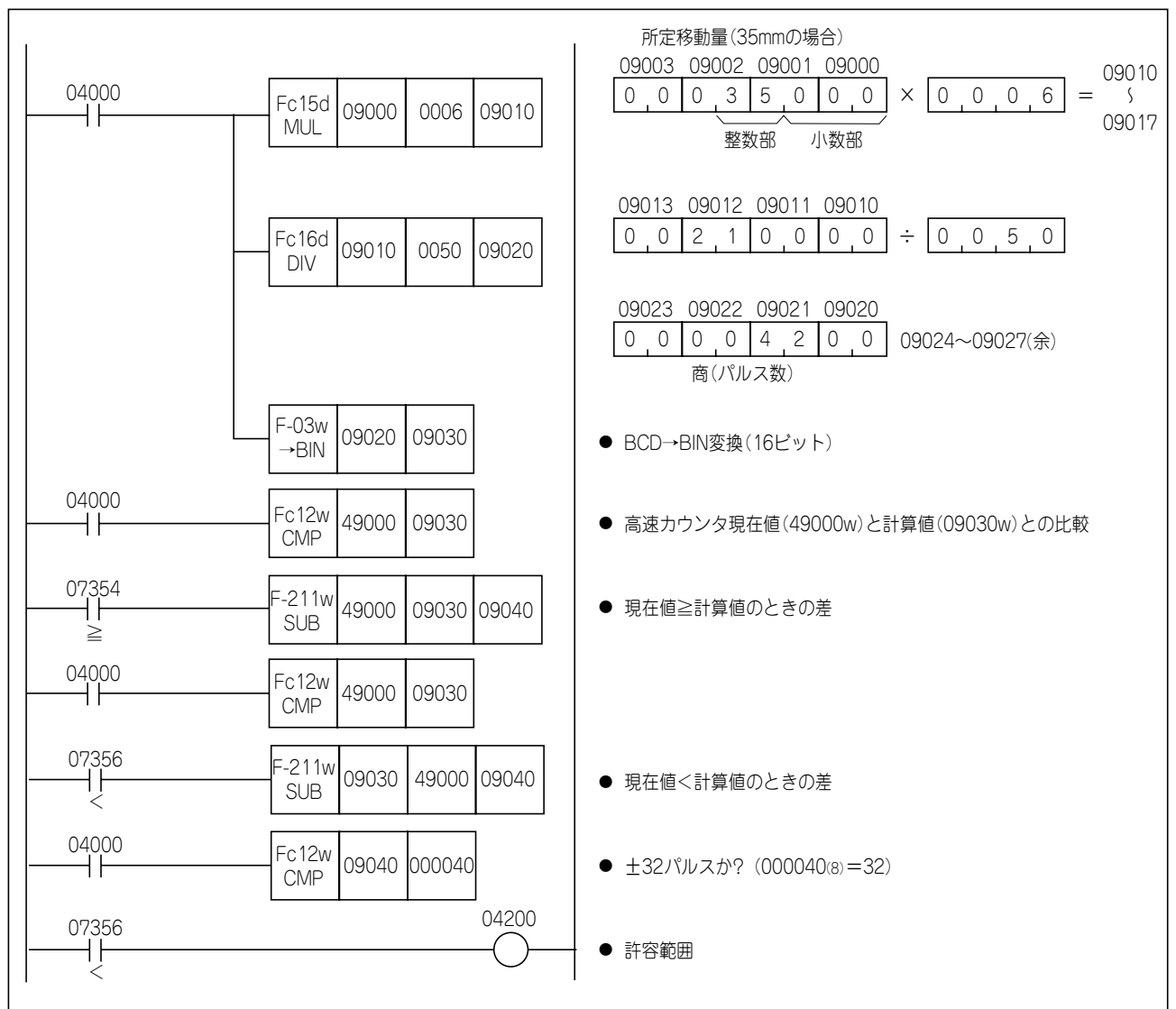
4-1	位置ずれ検出	4-1
4-2	鉄板の折り曲げ長さを求める	4-2
4-3	基板カセットの検索・削除処理	4-3
4-4	自動倉庫の入庫・出庫管理	4-5
4-5	アナログ出力電圧設定	4-8
4-6	データ照合	4-10
4-7	帳票用データの作成	4-14
4-8	スケール変換	4-17
4-9	昼夜判定処理	4-24
4-10	運転時間の切替	4-26
4-11	コンピュータリンクによるPC間通信	4-29

## 4 - 1 位置ずれ検出

- プレート昇降装置の所定移動量をパルス数に変換します。
- エンコーダからのパルスを高速カウンタユニット (JW-2HC) で取り込み、位置ずれ許容範囲に入っているかどうかを検出します。

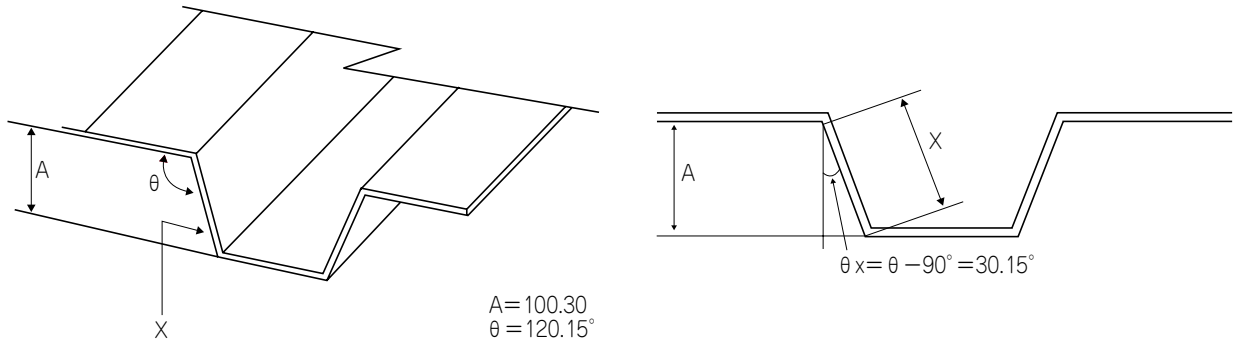


適応PC JW50H/70H/100H



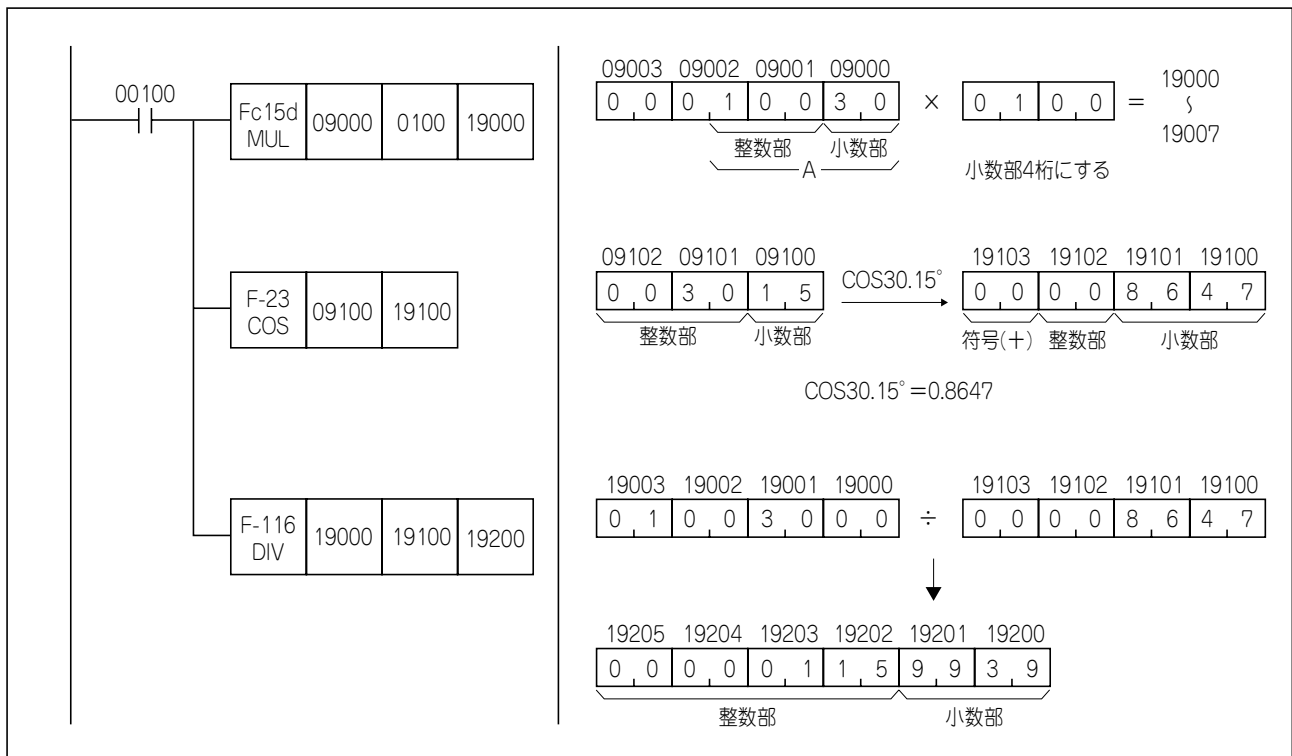
## 4 - 2 鉄板の折り曲げ長さを求める〔F-23(COS)命令使用〕

- 鉄板を折り曲げ、溝を作る場合、溝の深さAと折り曲げ角度θより、長さXを求めます。



$$A = 100.30 \quad \theta = 120.15^\circ \quad \text{より} \quad X = \frac{A}{\cos \theta_x} = \frac{100.30}{\cos 30.15^\circ} = \frac{100.30}{0.8647} = 115.99$$

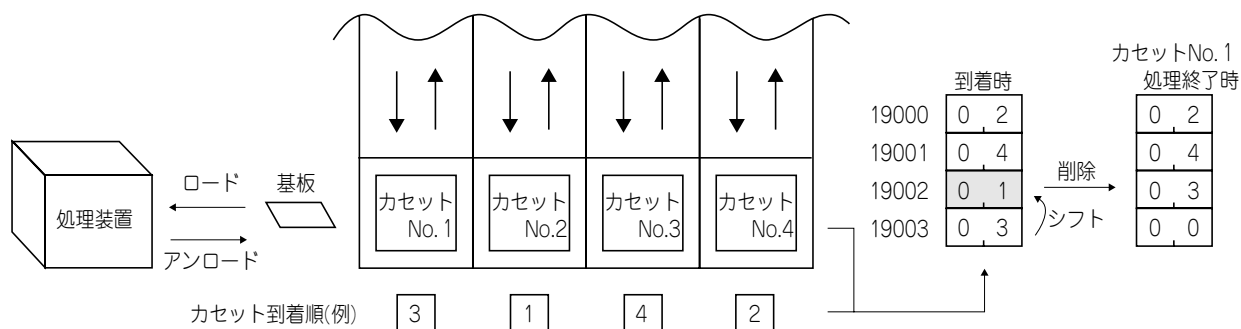
**適応PC** JW30H(JW-32CUH/H1, JW-33CUH/H1/H2/H3)  
JW50H/70H/100H





## 4 - 3 基板カセットの検索・削除処理

- 基板をカセットから処理装置にロード、アンロードする場所が4ヶ所あり、それぞれの場所にカセットが到着したとき、レジスタ19000～19003に到着順に1～4のカセットNo.を転送しておきます。
- いずれかのカセット処理が終了し、カセットを取り去るとき、レジスタ19000～19003から取り去るカセットNo.を削除します。



**適応PC** JW30H(JW-32CUH/H1, JW-33CUH/H1/H2/H3)  
JW50H/70H/100H

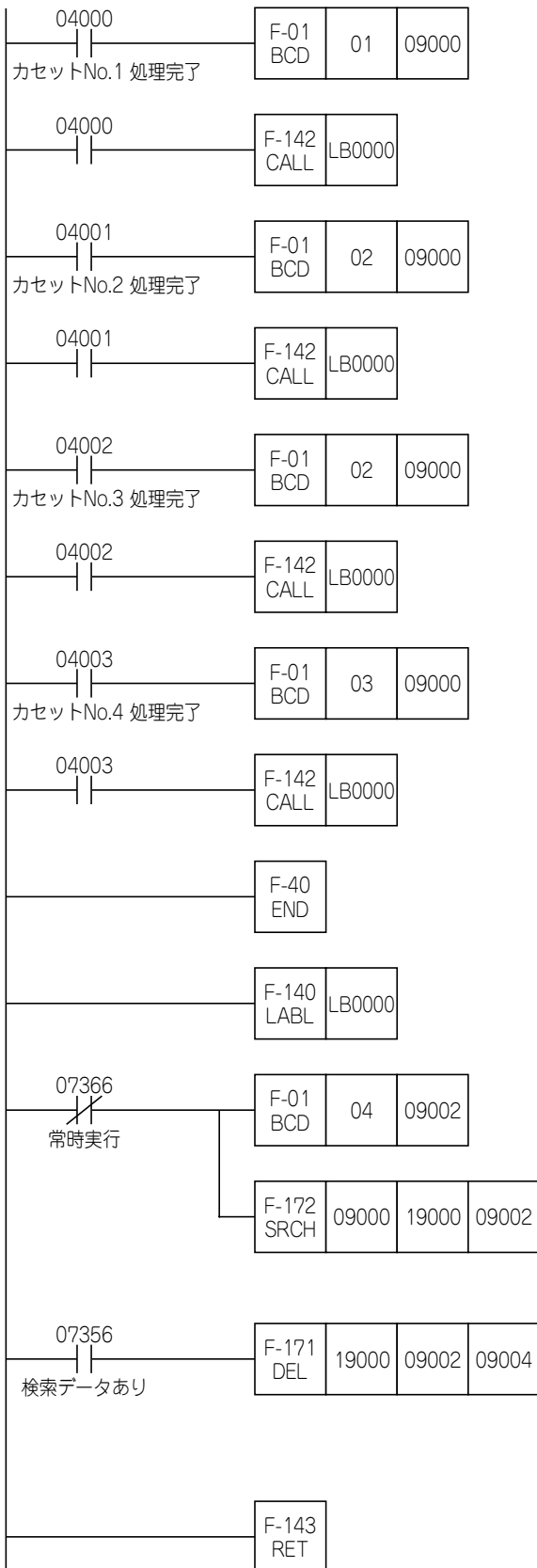
### 処理内容

1. F-172(データ検索)命令を使用して取り去る(処理完了)カセットNo.を検索します。
2. 検索データがあれば、F-171(データ削除)命令を使用して該当カセットNo.を削除します。

### データメモリ

リレー	内 容
04000	カセットNo.1 処理完了
04001	カセットNo.2 〃
04002	カセットNo.3 〃
04003	カセットNo.4 〃

レジスタ	内 容
09000	検索データ (01～04)
09001	
09002	検索バイト数 (4バイト)
09003	検索した個数
09004	検索したアドレス (000～003)
19000	カセットNo.格納用 (1番目)
19001	〃 (2番目)
19002	〃 (3番目)
19003	〃 (4番目)

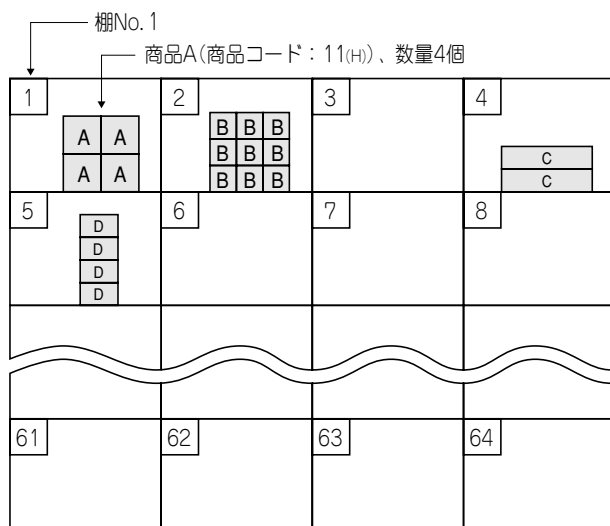


- 検索データの設定(01)
- 検索・削除処理サブルーチンをコール
- 検索データの設定(02)
- 検索・削除処理サブルーチンをコール
- 検索データの設定(03)
- 検索・削除処理サブルーチンをコール
- 検索データの設定(04)
- 検索・削除処理サブルーチンをコール
- プログラムの終了
- 検索・削除処理サブルーチンのラベル
- 検索バイト数の設定(4バイト)
- 19000~19003の4バイトで09000のデータを検索  
→ 09003 検索した個数  
09004 検索したアドレス(000~003)
- 19000~19003の4バイトで09004のアドレスのデータを削除  
(例) カセットNo.1処理完了時(04000がOFF→ON)



## 4 - 4 自動倉庫の入庫・出庫管理

- 自動倉庫の入庫・出庫処理をF-172(検索)、F-05(分配)、F-06(抽出)命令等を用いて行います。
- 商品棚は64個あり、各棚は商品コードと商品の数量で管理されています。



棚No.	商品コード	数 量
1	09000	19000
2	09001	19001
3	09002	19002
4	09003	19003
5	09004	19004
6	09005	19005
⋮	⋮	⋮
63	09076	19076
64	09077	19077

**適応PC** JW30H(JW-32CUH/H1, JW-33CUH/H1/H2/H3)  
JW50H/70H/100H

### データメモリ

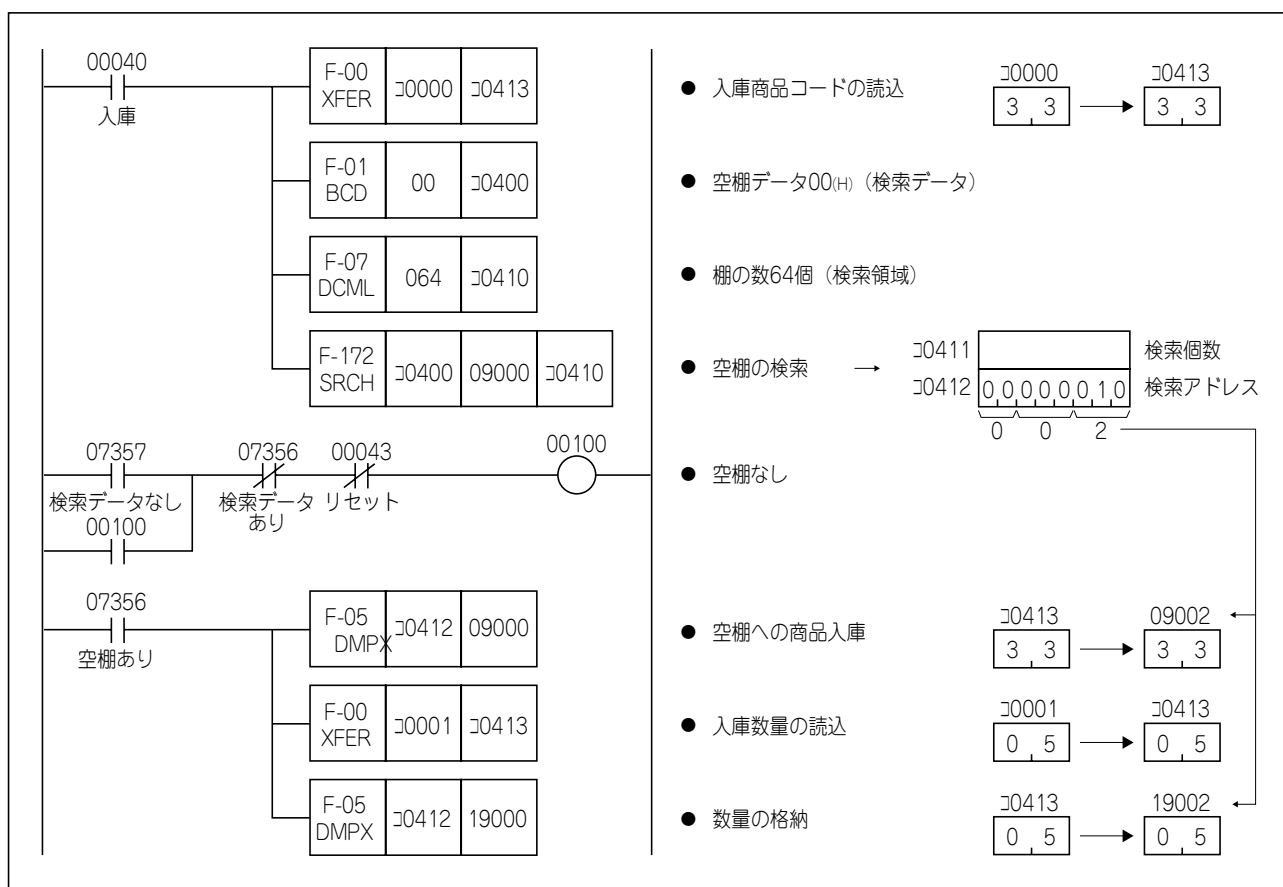
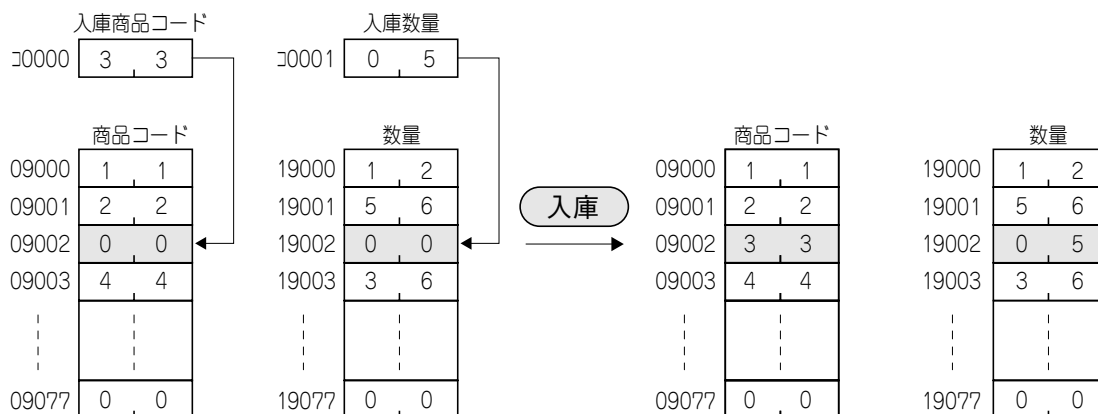
リレー	内 容
00040	入 庫
00041	数量確認
00042	出 庫
00043	リセット
00100	空棚なし
00101	出庫商品なし

レジスタ	内 容
J0000	入庫商品コード
J0001	入庫数量
J0002	出庫商品コード
J0003	出庫数量
J0400	検索データ
J0410	棚の個数
J0411	検索個数
J0412	検索アドレス(000~077)
J0413	分配・抽出データ

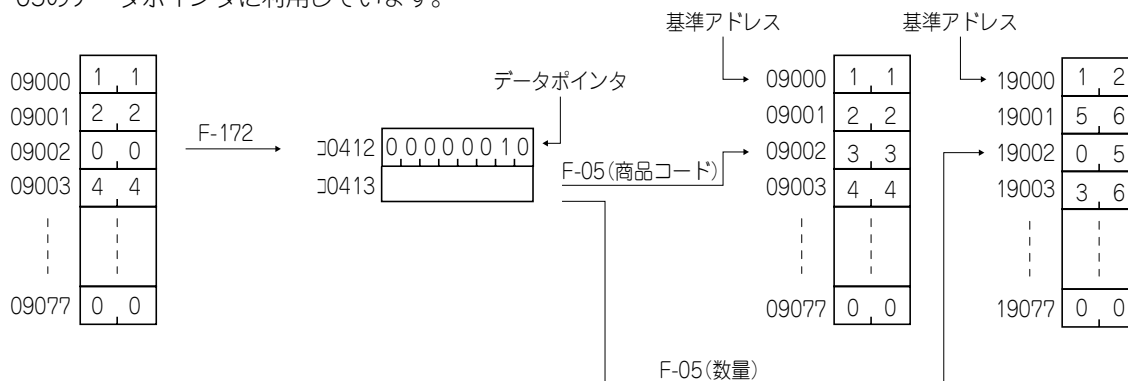
レジスタ	内 容
09000	商品コード (64個)
09077	
19000	商品数量
19077	

(1) 入庫処理

- 空の棚に0000で設定した商品コードの商品を0001で設定した数量を格納します。

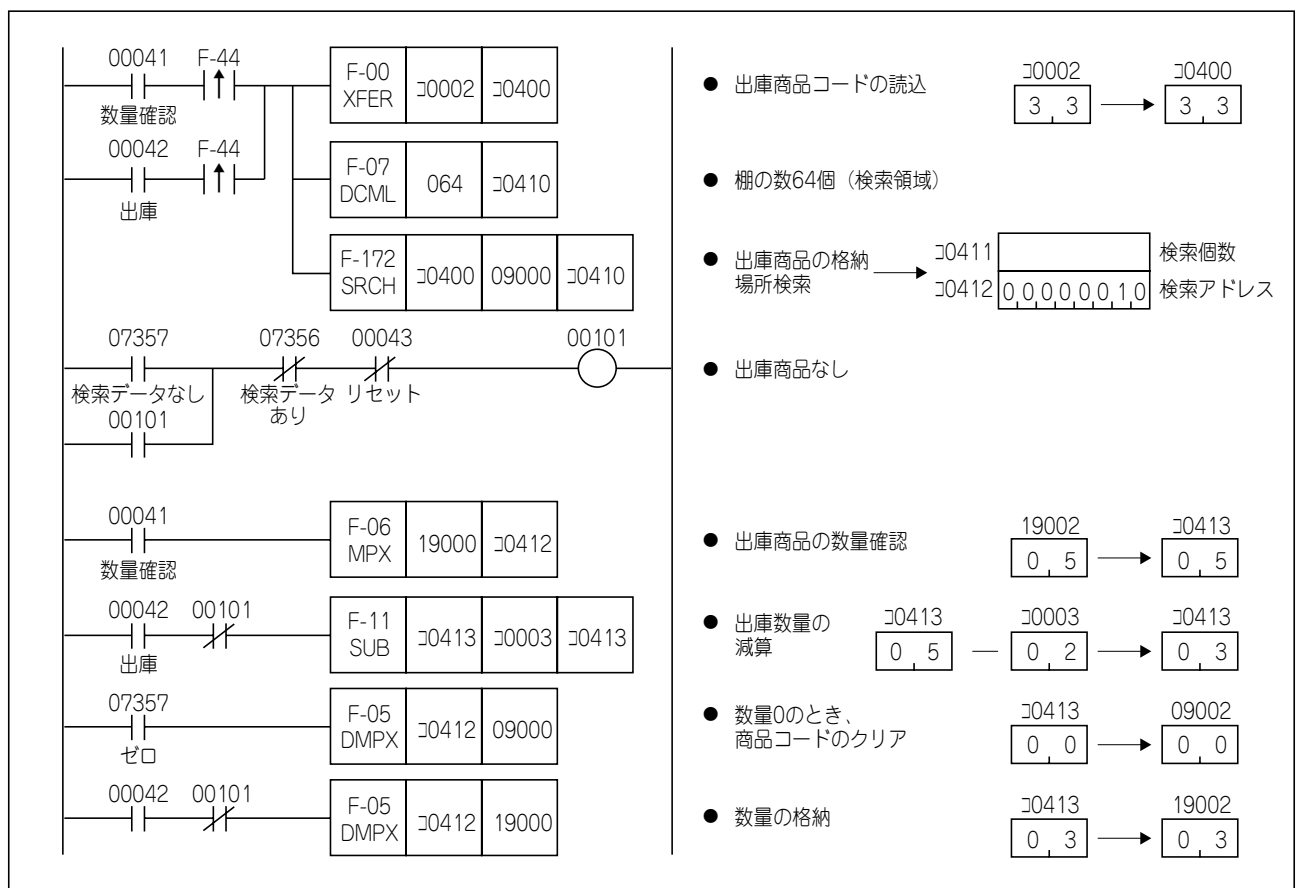
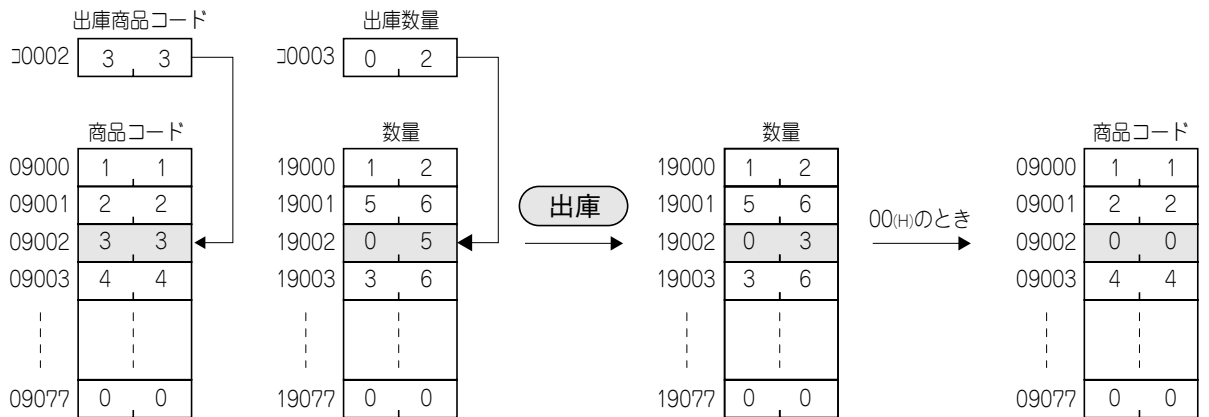


- 上記プログラムでは、F-172で求めた検索アドレスをF-05のデータポインタに利用しています。

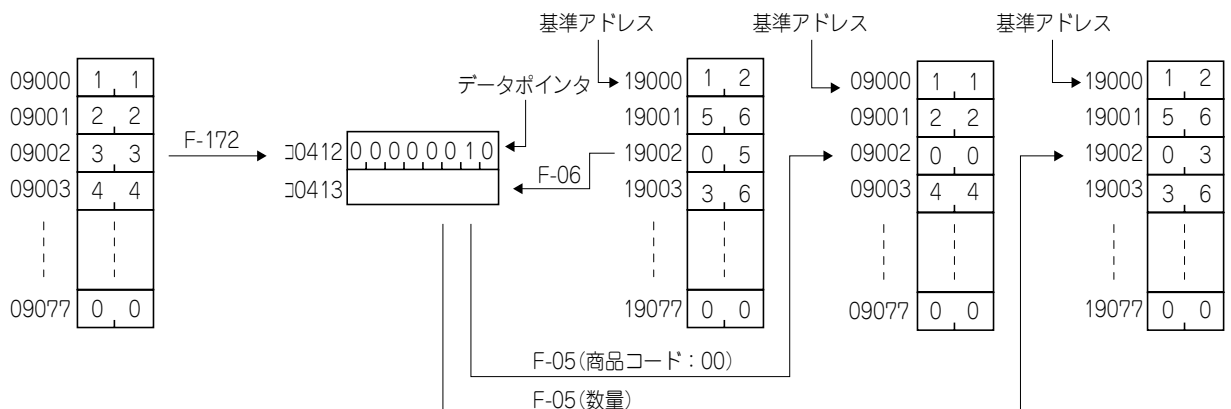


## (2) 出庫処理

- J0002で設定した商品コードの商品の個数を確認します。
- 確認後、J0003で設定した数量を取り出します。



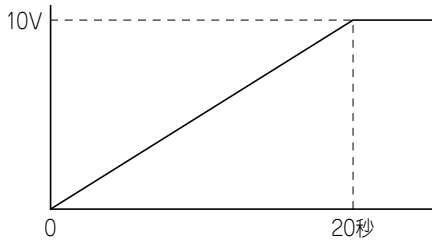
- 上記プログラムでは、F-172で求めた検索アドレスをF-06、F-05のデータポイントに利用しています。



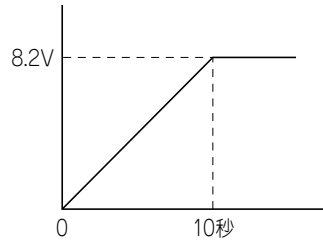
## 4-5 アナログ出力電圧設定

- アナログ出力ユニット(JW-2DA)を使用します。
- 設定時間後に設定電圧を出力します。
- 設定時間に到るまでは、電圧はリニアに出力します。  
(分解能：0.1秒単位)

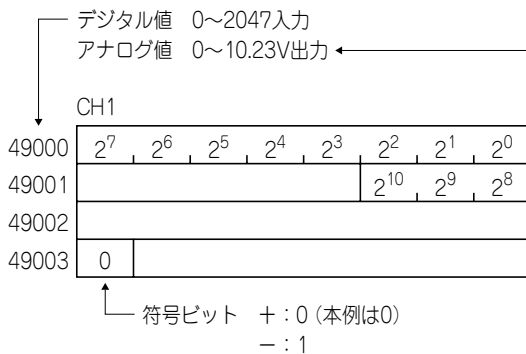
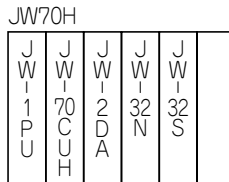
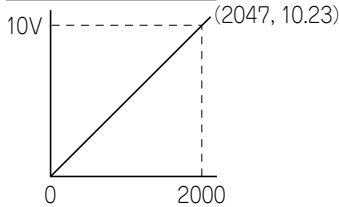
(例1) 時間設定 20秒  
電圧設定 10.0V



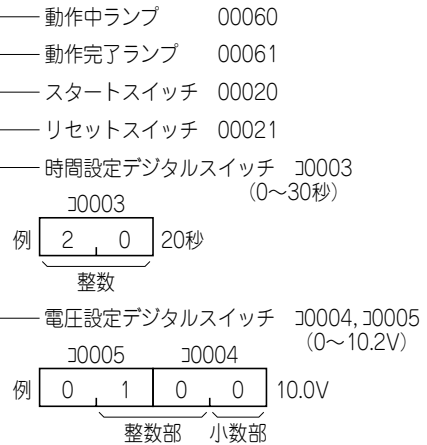
(例2) 時間設定 10秒  
電圧設定 8.2V



JW-2DA 電圧出力特性



JW-2DAは、上記レジスタにデジタル値を設定するだけで、電圧出力します。



適応PC JW50H/70H/100H

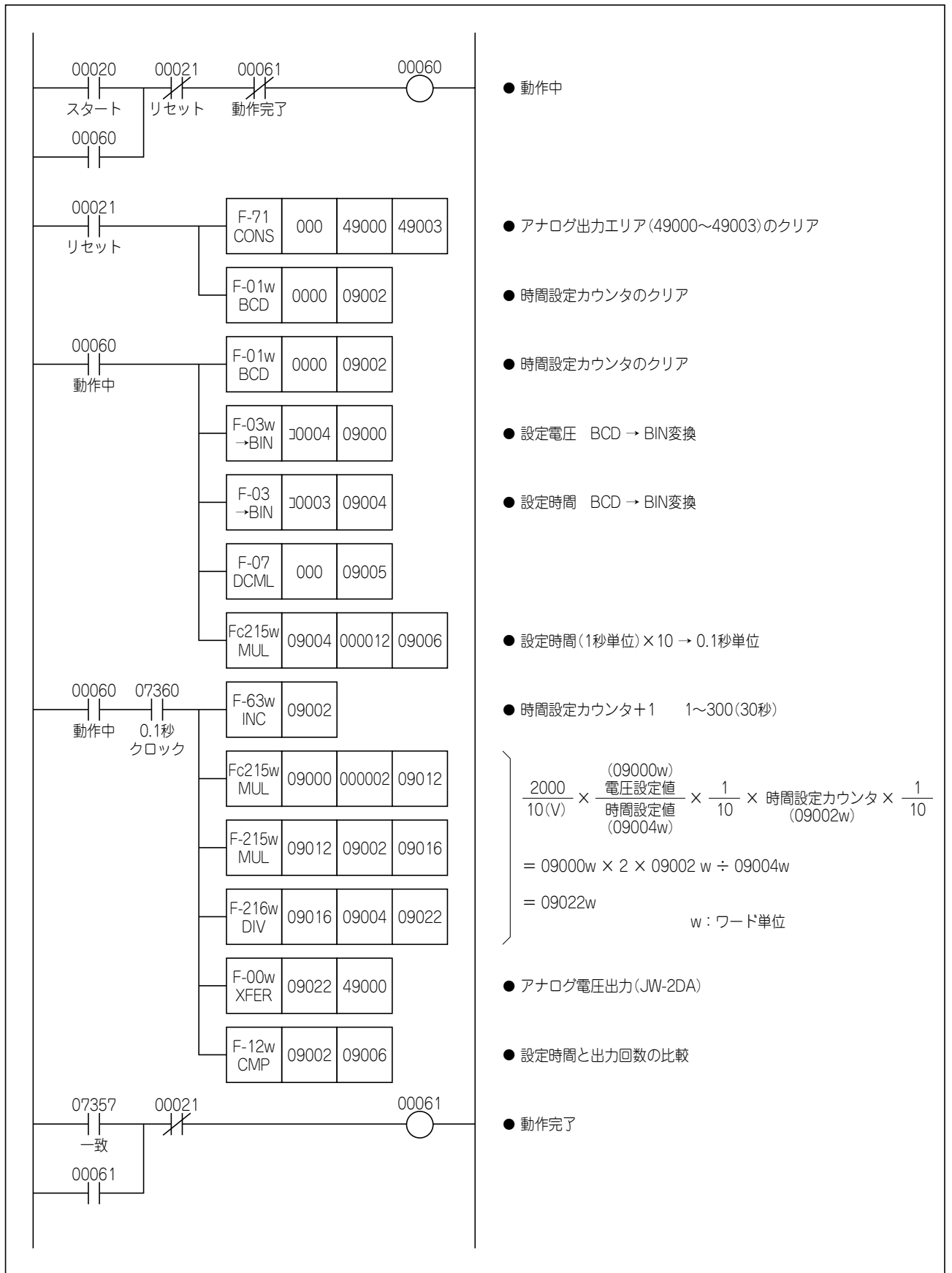
データメモリ

リレー	内 容
00020	スタートスイッチ
00021	リセットスイッチ
00060	動作中
00061	動作完了

レジスタ	内 容
30003	時間設定値
30004	電圧設定値
30005	〃

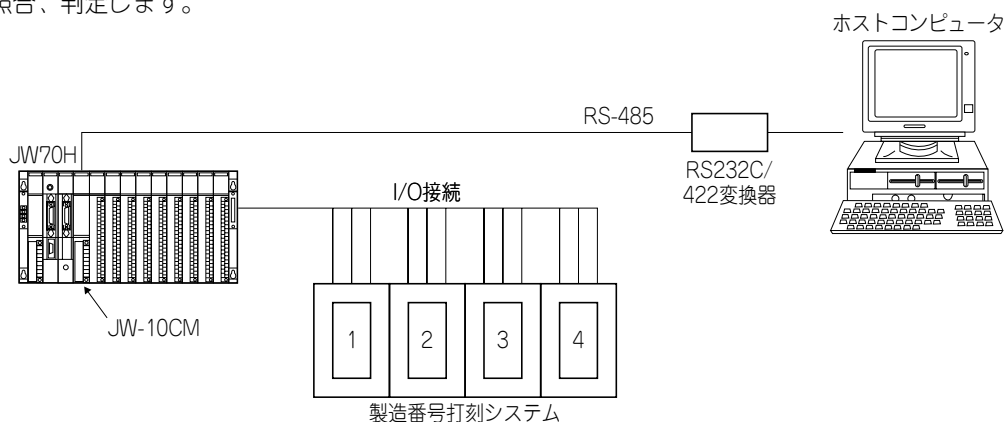
レジスタ	内 容
09000	設定電圧(バイナリ)
09001	〃
09002	時間設定カウンタ
09003	〃
09004	設定時間(バイナリ)
09005	〃
09006	出力回数
09007	〃
09022	アナログ出力値(バイナリ)
09023	〃

レジスタ	内 容
09010	演算用ワーキングレジスタ
〃	
09021	
49000	アナログ出力エリア
49001	〃
49002	〃
49003	〃



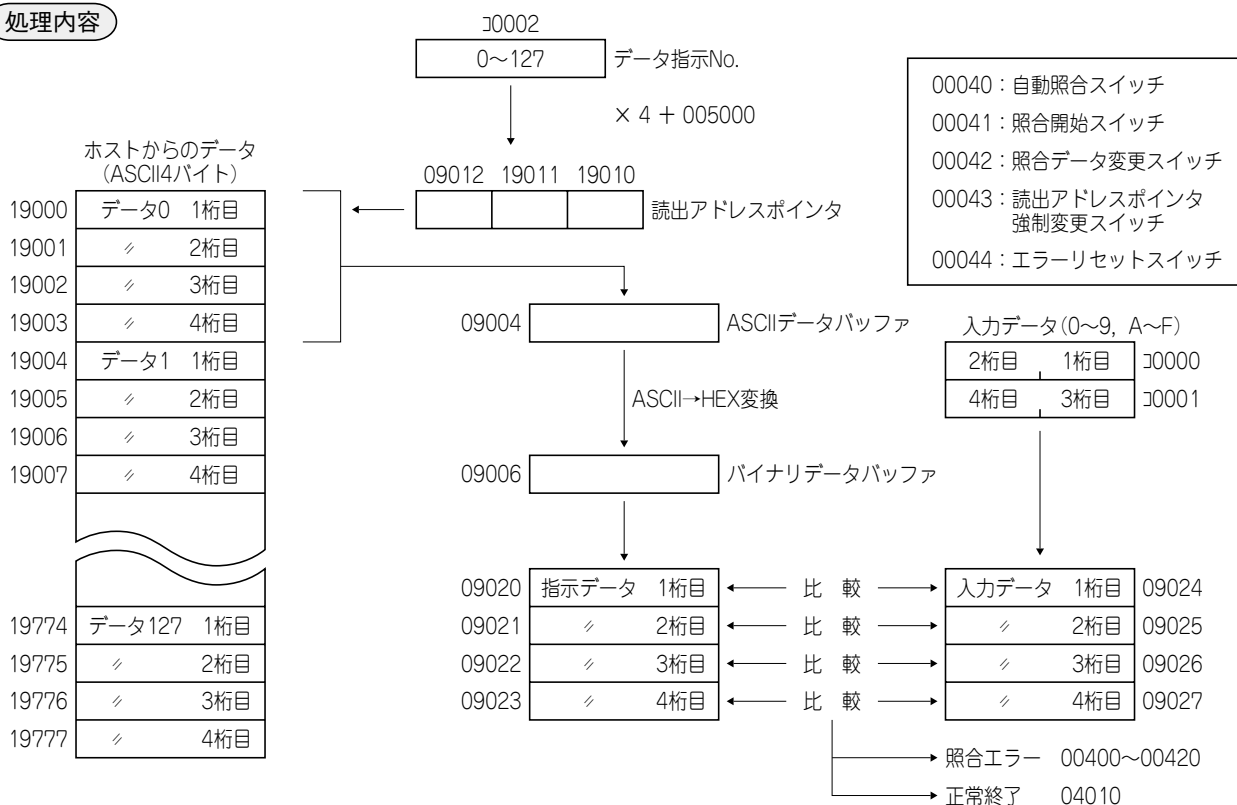
## 4-6 データ照合

- 製造番号の打刻システムで、現在入力されているデータとホストコンピュータからの指示データが一致しているかどうか照合、判定します。



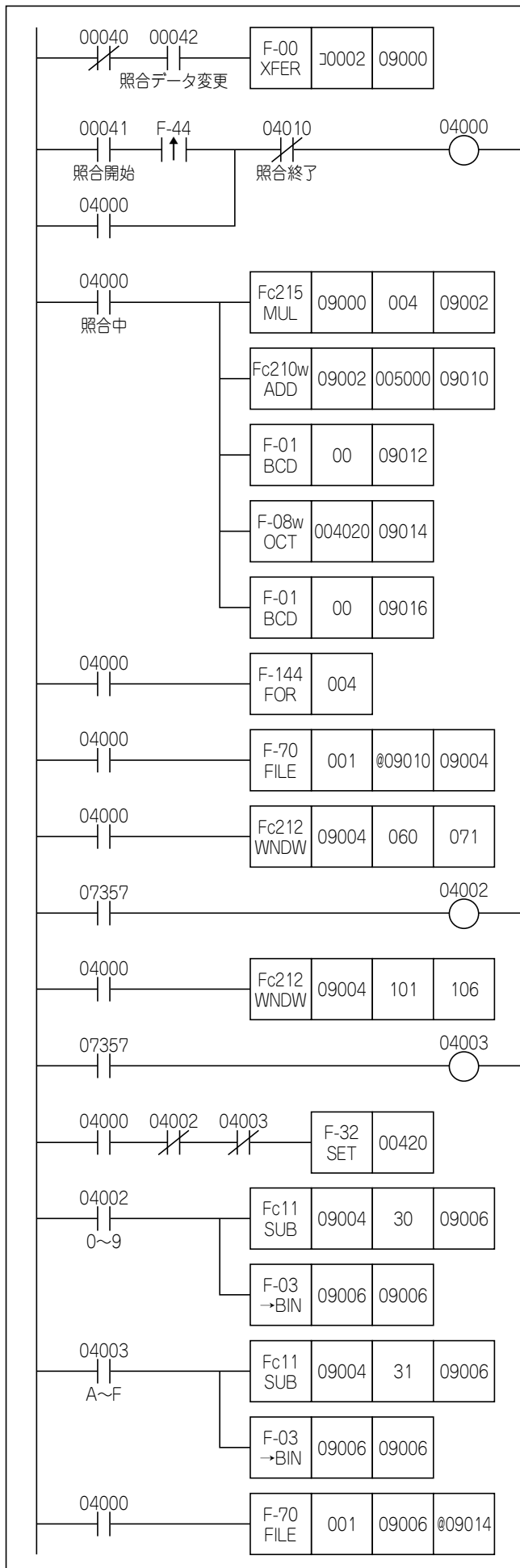
適応PC 全JWシリーズ

処理内容



- ホストコンピュータからの指示データは、19000~19777に格納しておきます。
- 照合を開始するデータの先頭アドレスは、00002に設定します。
- ホストコンピュータのデータがASCIIでないとき、エラーリレー(00420)をONします。
- ホストコンピュータのデータと入力データ(00000~00001で設定)を照合し、照合エラー時は00410~00413をONします。
- 照合が正常終了すると、読出アドレスポインタは次のデータアドレスに移りますが、エラーとなった場合は、エラーリセットスイッチ(00044)ON後、読出アドレスポインタ強制変更スイッチのONで次のデータアドレスに移ります。

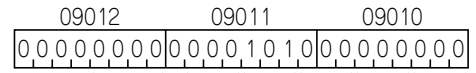




● データ指示No.(0~127)の読み

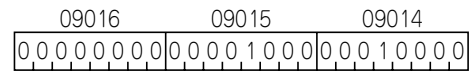
● 照合中

読みアドレスポインタ(19000~19777)の設定



= 19000

指示データ比較用レジスタ(09020~09023)の設定



= 09020

● 指示データの読出

● ASCII(0~9) ?  $30^{(H)} \leq 09004 \leq 39^{(H)}$   
(060<sup>(B)</sup>) (071<sup>(B)</sup>)

● ASCII(0~9)

● ASCII(A~F) ?  $41^{(H)} \leq 09004 \leq 46^{(H)}$   
(101<sup>(B)</sup>) (106<sup>(B)</sup>)

● ASCII(A~F)

● ASCII(0~9, A~F)でないとき、00420 ON  
(指示データ NG)

● 30~39 → 00~09 (BCD)

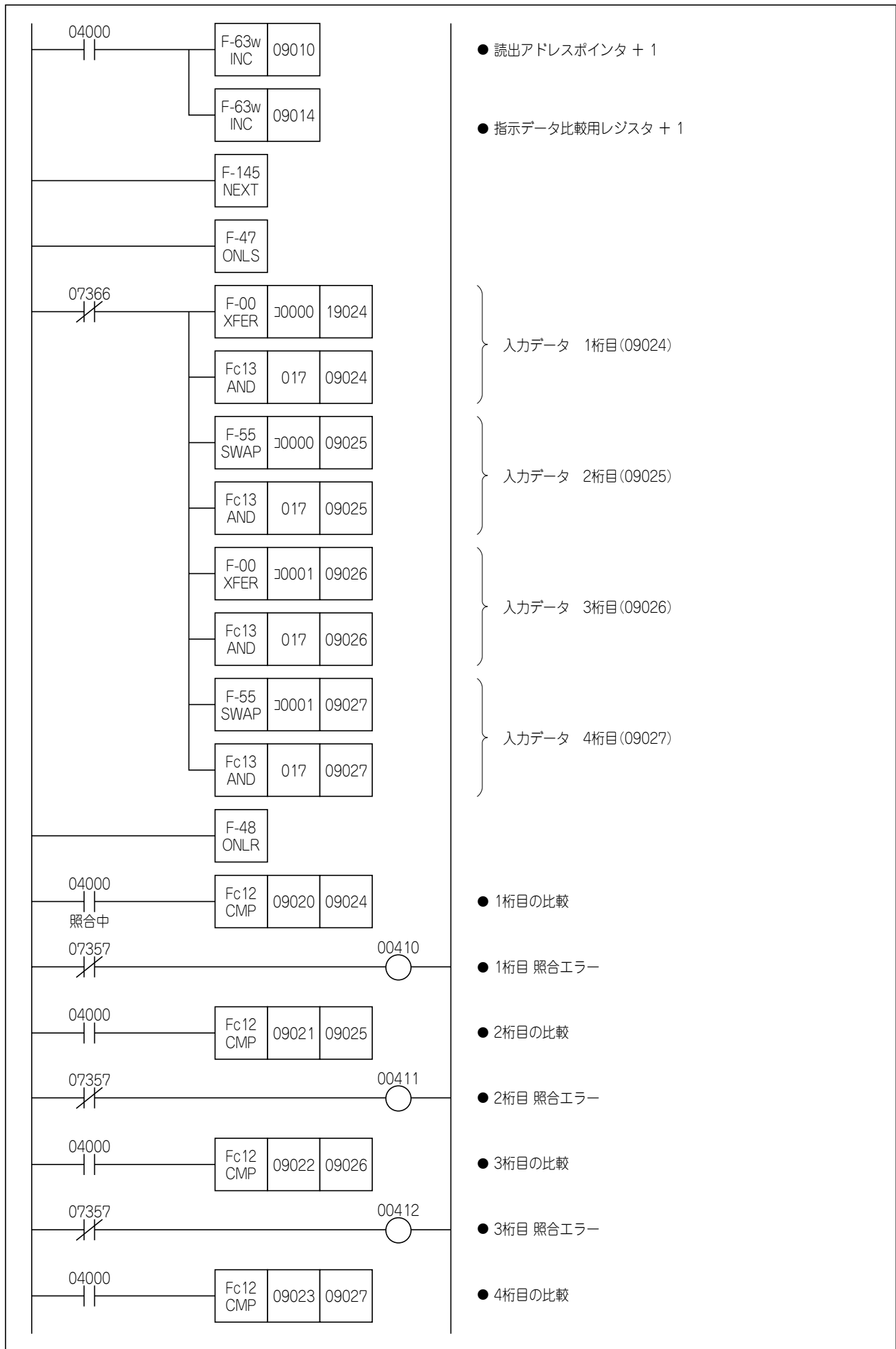
● 00~09 (BCD) → 0~9 (HEX)

● 41~46 → 10~15 (BCD)

● 10~15 (BCD) → A~F (HEX)

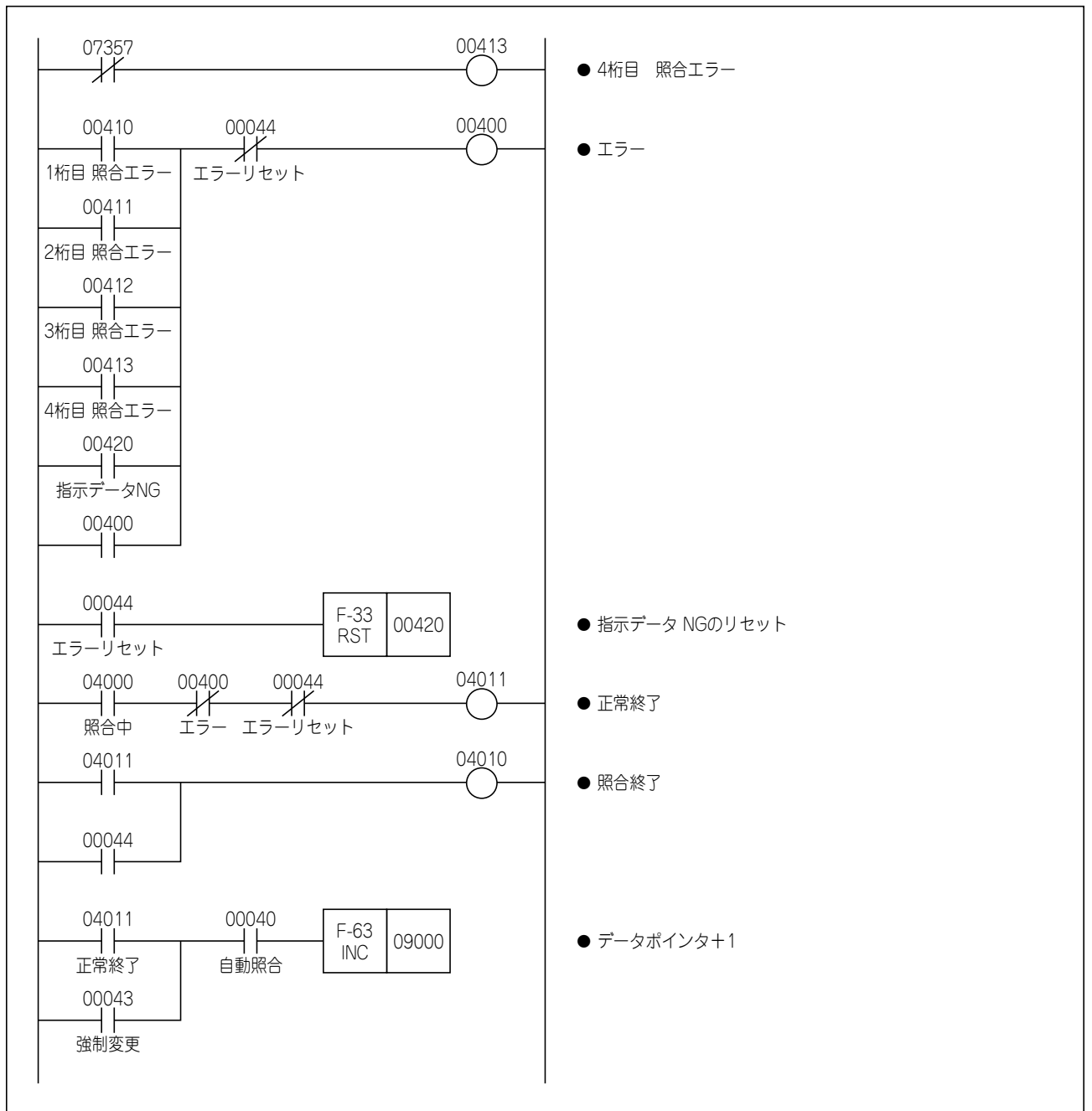
● 指示データ比較用レジスタ(09020~09023)への転送

(次ページへ続く)



- 読出アドレスポインタ + 1
- 指示データ比較用レジスタ + 1
- 入力データ 1桁目(09024)
- 入力データ 2桁目(09025)
- 入力データ 3桁目(09026)
- 入力データ 4桁目(09027)
- 1桁目の比較
- 1桁目 照合エラー
- 2桁目の比較
- 2桁目 照合エラー
- 3桁目の比較
- 3桁目 照合エラー
- 4桁目の比較

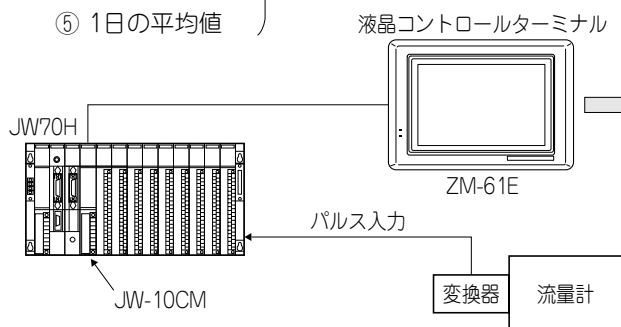
(次ページへ続く)



## 4 - 7 帳票用データの作成

- パルス入力(流体流量, 受電電力, 生産実績等)を積算カウンタし、下記をZM-61E等の表示器に表示します。

- ① 1時間毎の累計
  - ② 1日の合計
  - ③ 1日の最小値
  - ④ 1日の最大値
  - ⑤ 1日の平均値
- } 当日と前日の2日分のデータ蓄積



画面例

流量積算一覧		前画面	次画面
0:00~	100m <sup>3</sup>	本日合計 970m <sup>3</sup> 最小 42m <sup>3</sup> 最大 140m <sup>3</sup> 平均 94m <sup>3</sup>	
1:00~	92m <sup>3</sup>		
2:00~	105m <sup>3</sup>		
3:00~	79m <sup>3</sup>		
4:00~	130m <sup>3</sup>		
5:00~	140m <sup>3</sup>		

**適応PC** JW10 (JW-1424K/1442K/1624K/1642K)  
 JW20H (JW-22CU), J-board (Z-312J)  
 JW30H (JW-32CUH/H1  
 JW-33CUH/H1/H2/H3)  
 JW50H/70H/100H

**注1** 本プログラムには、表示器にデータを表示するためのプログラムは含んでいません。

### 処理内容

- 00000を入力パルスとし、09000wに総累計を格納します。

#### 1) 常時実行する内容

- ・正時より増加した量を時累計として09400~に格納します。
- ・当日の時累計を加算したものを合計値として09464wに格納します。

#### 2) 正時毎に実行する内容(時更新時処理)

- ・時累計のうち最小のものを最小値として09460wに格納します。
- ・時累計のうち最大のものを最大値として09462wに格納します。
- ・合計を時累計のサンプル回数で割ったものを平均値として09470wに格納します。

#### 3) 日付が変わったときに実行する内容(日更新時処理)

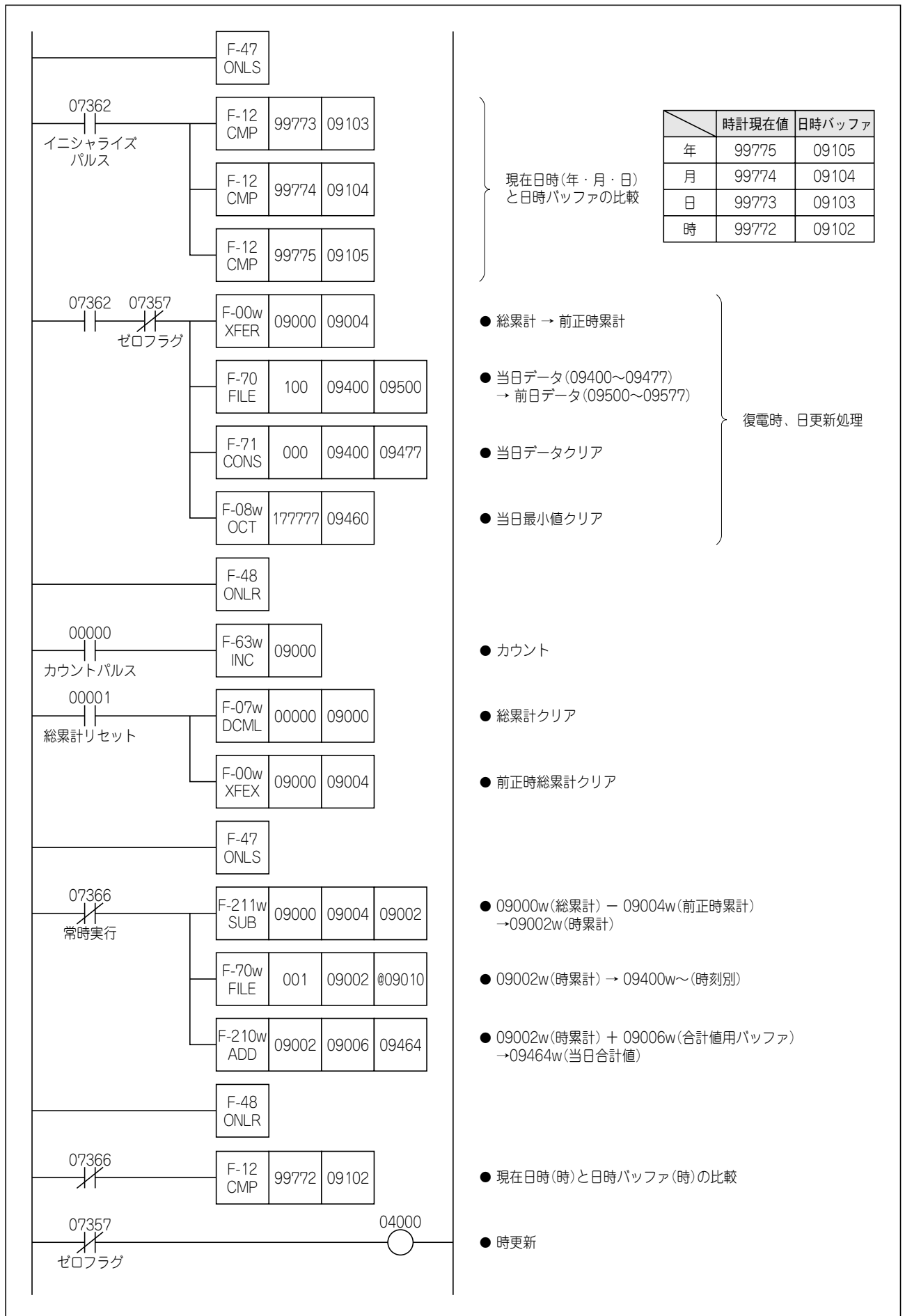
- ・1), 2)のデータ(09400~)を前日データ領域(09500~)に転送します。
- ・当日データ領域(09400~)は0クリアします。  
(最小値領域はFF(H)クリアします)

09000w	総累計	
09002w	時累計	(総累計-前正時)
09004w	前正時累計	

当日データ		前日データ	
09400w	00 ~ 01時	09500w	00 ~ 01時
09402w	01 ~ 02時	09502w	01 ~ 02時
09404w	02 ~ 03時	09504w	02 ~ 03時
09406w	03 ~ 04時	09506w	03 ~ 04時
09410w	04 ~ 05時	09510w	04 ~ 05時
⋮	⋮	⋮	⋮
09454w	22 ~ 23時	09554w	22 ~ 23時
09456w	23 ~ 24時	09556w	23 ~ 24時
09460w	最小値	09560w	最小値
09462w	最大値	09562w	最大値
09464w	合計値	09564w	合計値
09470w	平均値	09570w	平均値
09476w	サンプル回数	09576w	サンプル回数

**注2** JW10と他のPCでは時計のレジスタ番号が異なります。本プログラムではJW10以外のPCの番号で記しています。

	JW10	他のPC
年	11575	99775
月	11574	99774
日	11573	99773
時	11572	99772



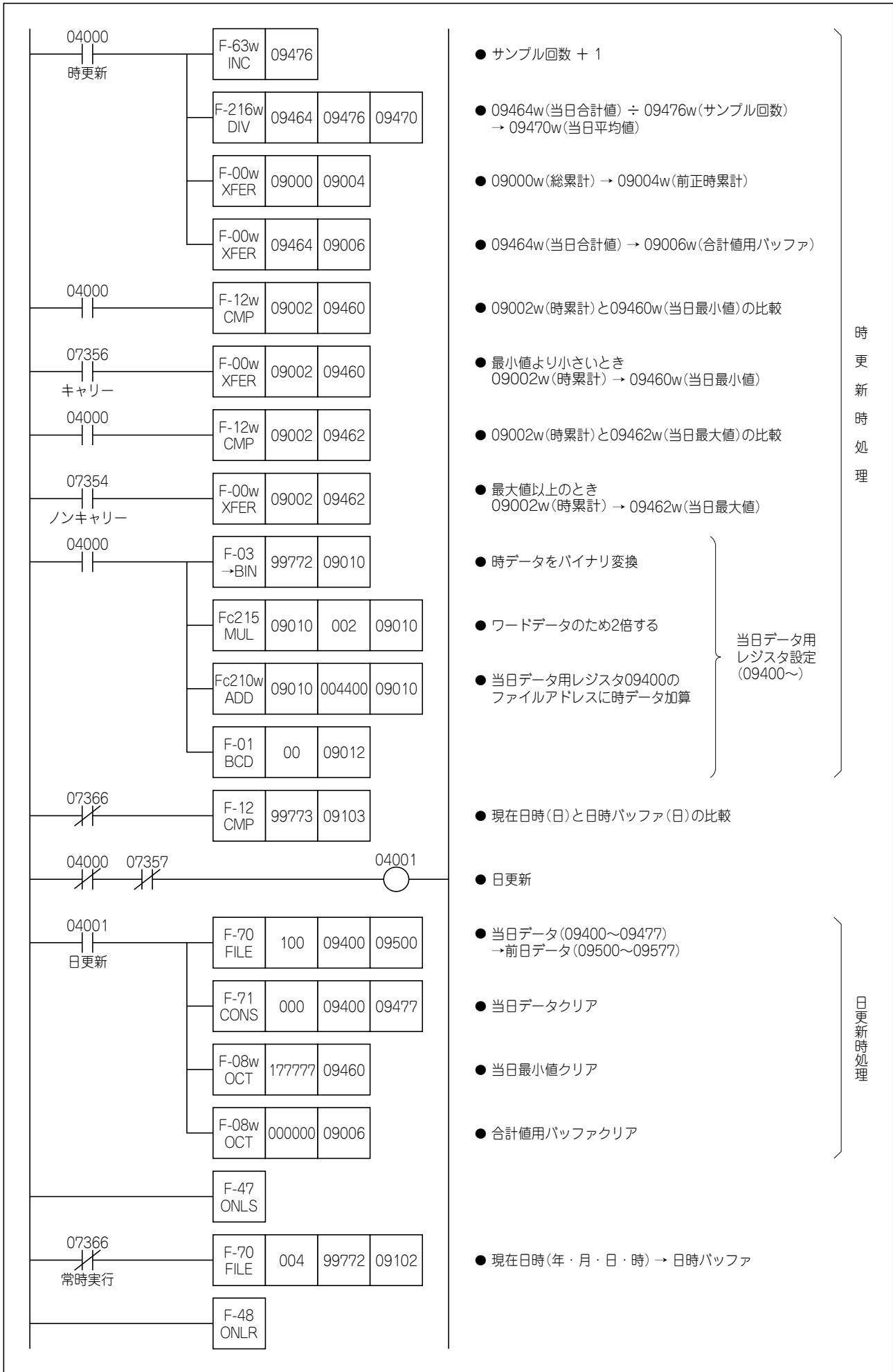
	時計現在値	日時バッファ
年	99775	09105
月	99774	09104
日	99773	09103
時	99772	09102

現在日時(年・月・日)  
と日時バッファの比較

復電時、日更新処理

- 総累計 → 前正時累計
- 当日データ(09400~09477)  
→ 前日データ(09500~09577)
- 当日データクリア
- 当日最小値クリア
- カウント
- 総累計クリア
- 前正時総累計クリア
- 09000w(総累計) - 09004w(前正時累計)  
→ 09002w(時累計)
- 09002w(時累計) → 09400w~(時刻別)
- 09002w(時累計) + 09006w(合計値用バッファ)  
→ 09464w(当日合計値)
- 現在日時(時)と日時バッファ(時)の比較
- 時更新

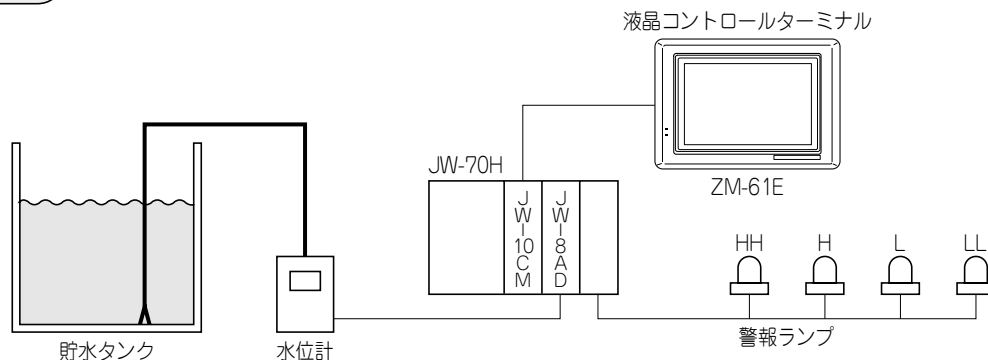
(次ページへ続く)



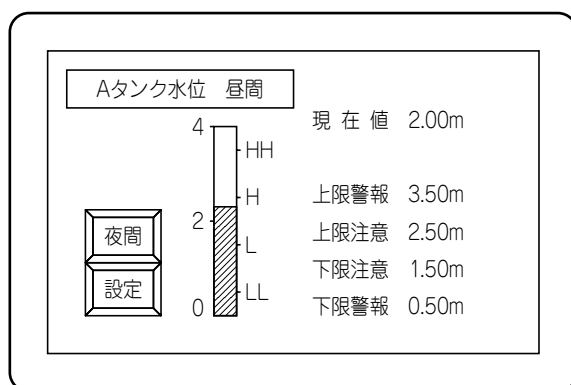
## 4 - 8 スケール変換

- 工場内の貯水タンクなどの水位をアナログ入力し、液晶コントロールターミナルなどへの表示用にデータを変換します。
- 液晶コントロールターミナルからの設定で、上上限、上限、下限、下限の4点を警報として出力します。

### システム例



### ZM-61E 画面例



水位計が4～20mA出力仕様の場合

- 4mA出力するときの値をベーススケールといいます。(本例の場合は0mで0000を設定)
- 20mA出力するときの値をフルスケールといいます。(本例の場合は4mで4000を設定)
- 警報出力のチャタリングを防ぐために復帰時に設ける余裕幅をヒステリシス幅といいます。(本例の場合、ヒステリシス幅2%とすると0.08mとなり、0080を設定)

- バーグラフに現在の水位を表示します。(0～100データを使用)
- 現在値にデジタル値を表示します。(スケール変換データを使用)
- 設定スイッチを押すとテンキーが表示され、警報設定値を変更できます。(本プログラムには含まれません)
- 夜間スイッチを押すと夜間設定の画面に切替ります。

### 適応PC 全JWシリーズ

- ただし、警報設定値を昼夜で区別する場合、時計を使用するため、時計のないPC(JW-1324K/1342K, JW-21CU, JW31CUH/H1, Z-311J)は使用できません。

データメモリ

リレー	内 容	
00400	データNG(BCD以外)	
00410	警報出力	HH
00411	〃	H
00412	〃	L
00413	〃	LL
04000	自己発振パルス	
04020	データ>設定値	HH
04021	〃 > 〃	H
04022	〃 < 〃	L
04023	〃 < 〃	LL
04024	データ<警報復帰値	HH
04025	〃 < 〃	H
04026	〃 > 〃	L
04027	〃 > 〃	LL
04100	データ1桁目	1
04101	〃	2
04102	〃	4
04103	〃	8
04104	データ2桁目	1
04105	〃	2
04106	〃	4
04107	〃	8
04110	データ3桁目	1
04111	〃	2
04112	〃	4
04113	〃	8
04114	データ4桁目	1
04115	〃	2
04116	〃	4
04117	〃	8

レジスタ	内 容	
09000	データ(200~1000)	
09001	〃	
09002	データ(0~100)	
09003	〃	
09004	データ(スケール)	
09005	〃	
09006		
09007		
09010	ペーススケール設定値	
09011	〃	
09012	フルスケール設定値	
09013	〃	
09014	ヒステリシス設定値	
09015	〃	
09016		
09017		
09020	警報設定値(昼間)	HH
09021	〃	HH
09022	〃	H
09023	〃	H
09024	〃	L
09025	〃	L
09026	〃	LL
09027	〃	LL
09030	警報設定値(夜間)	HH
09031	〃	HH
09032	〃	H
09033	〃	H
09034	〃	L
09035	〃	L
09036	〃	LL
09037	〃	LL

レジスタ	内 容	
09100	データ処理用バッファ	
09101	〃	
09102	〃	
09103	〃	
09104	〃	
09105	〃	
09106	〃	
09107	〃	
09110	〃	
09111	〃	
09112	〃	
09113	〃	
09114	〃	
09115	〃	
09116	現在時刻比較バッファ	
09117	〃	
09120	設定値バッファ	HH
09121	〃	HH
09122	〃	H
09123	〃	H
09124	〃	L
09125	〃	L
09126	〃	LL
09127	〃	LL
09130	警報復帰値	HH
09131	〃	HH
09132	〃	H
09133	〃	H
09134	〃	L
09135	〃	L
09136	〃	LL
09137	〃	LL

レジスタ	内 容	
19000	ソースデータ(200~1000)	
19001	〃	

プログラム

プログラムは全てワード単位で扱います。(レジスタ09000wのように記します)

- データは19000wに格納されているものとします。(200~1000)
- データ変換は2種類行い、0~100はバーグラフ用、09010w、09012wに設定された任意スケールで変換されたものはデジタル値用に使用します。
- 警報値の入力は任意のスケールで行い、ヒステリシス幅は4点共通とします。
- 警報設定値は、09:00~17:00までを昼間、17:00~09:00までを夜間とし、別々の設定を行うことができます。(昼夜の区別により使用する水量が極端に違う場合があるため)

警報出力	内 容	
上限警報 (HH)	設定値を越えるとON	設定値-ヒステリシス幅を下回るとOFF
上限注意 (H)	設定値を越えるとON	設定値-ヒステリシス幅を下回るとOFF
下限注意 (L)	設定値を下回るとON	設定値+ヒステリシス幅を越えるとOFF
下限警報 (LL)	設定値を下回るとON	設定値+ヒステリシス幅を越えるとOFF

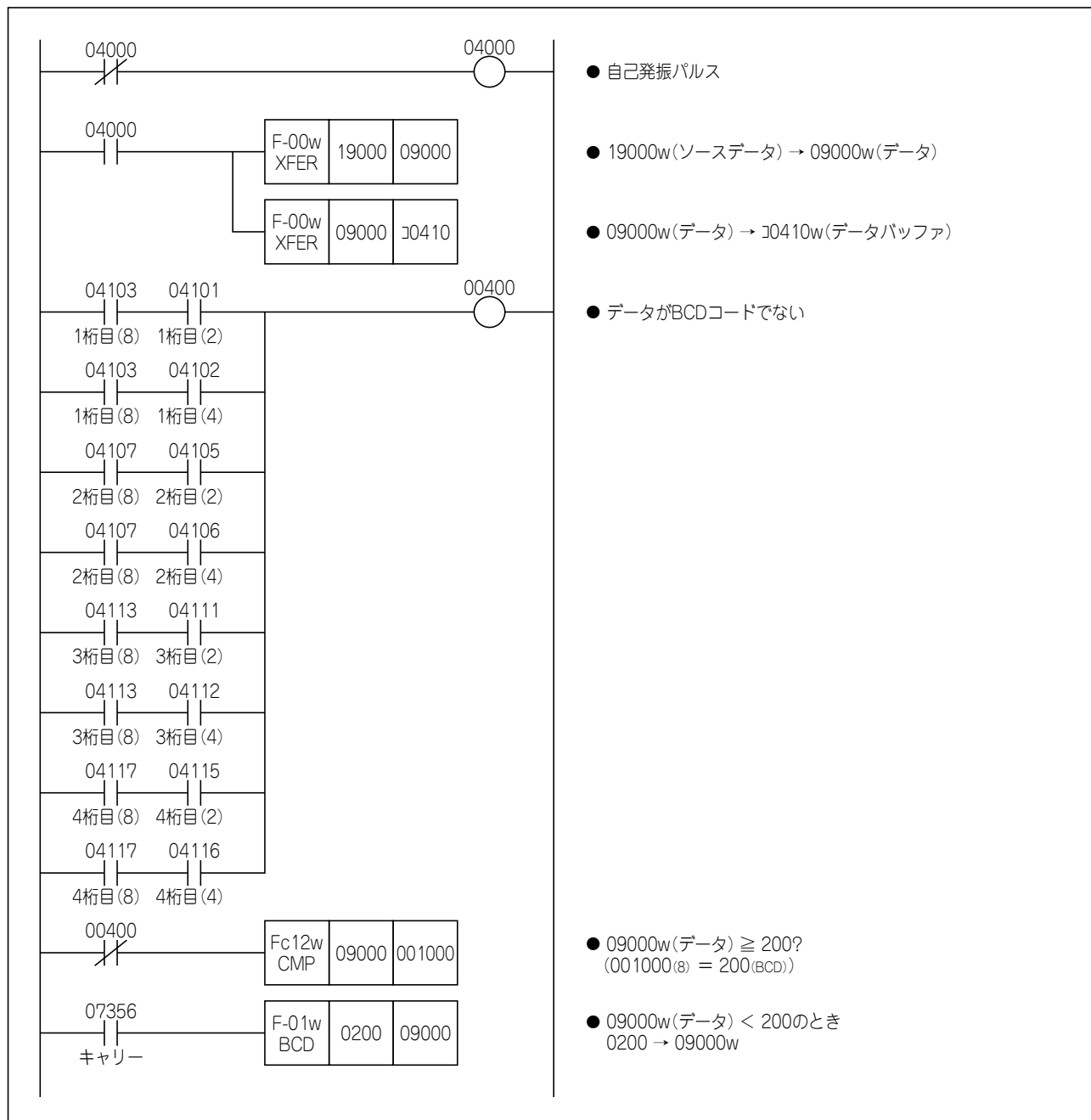


●プログラムは次の4つの部分より構成されます。

- (1) 初期処理
- (2) データ変換1
- (3) データ変換2
- (4) 警報設定との比較

(1) 初期処理

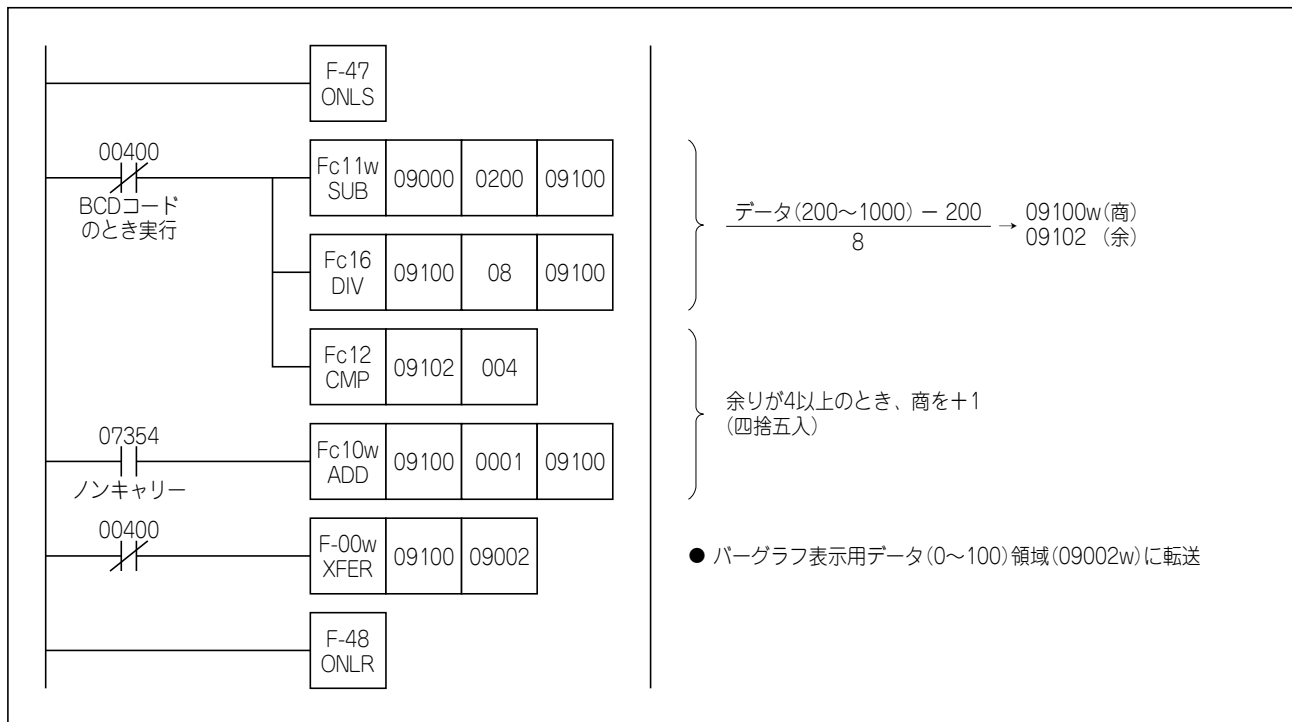
- 取込みデータがBCDコードかどうか判定します。NG時は変換処理を行いません。
- データが200未満でないことを判定します。200未満であれば200に設定します。



(2) データ変換1(バーグラフ表示用に変換)

- 200~1000のデータを0~100(バーグラフ表示用)に変換します。

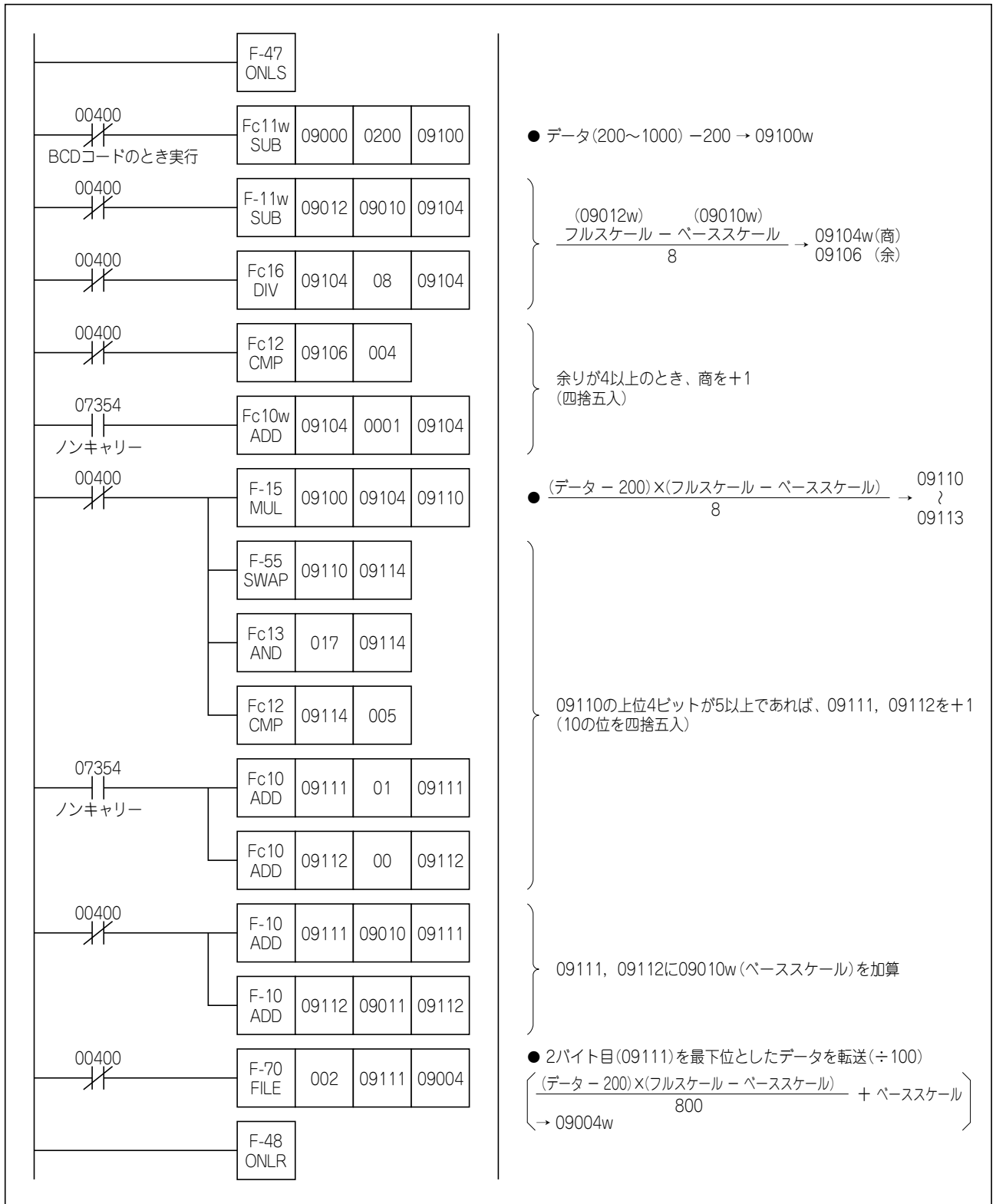
$$\text{変換値} = \frac{(\text{データ} - 200)}{8}$$



(3) データ変換2(任意スケールに変換)

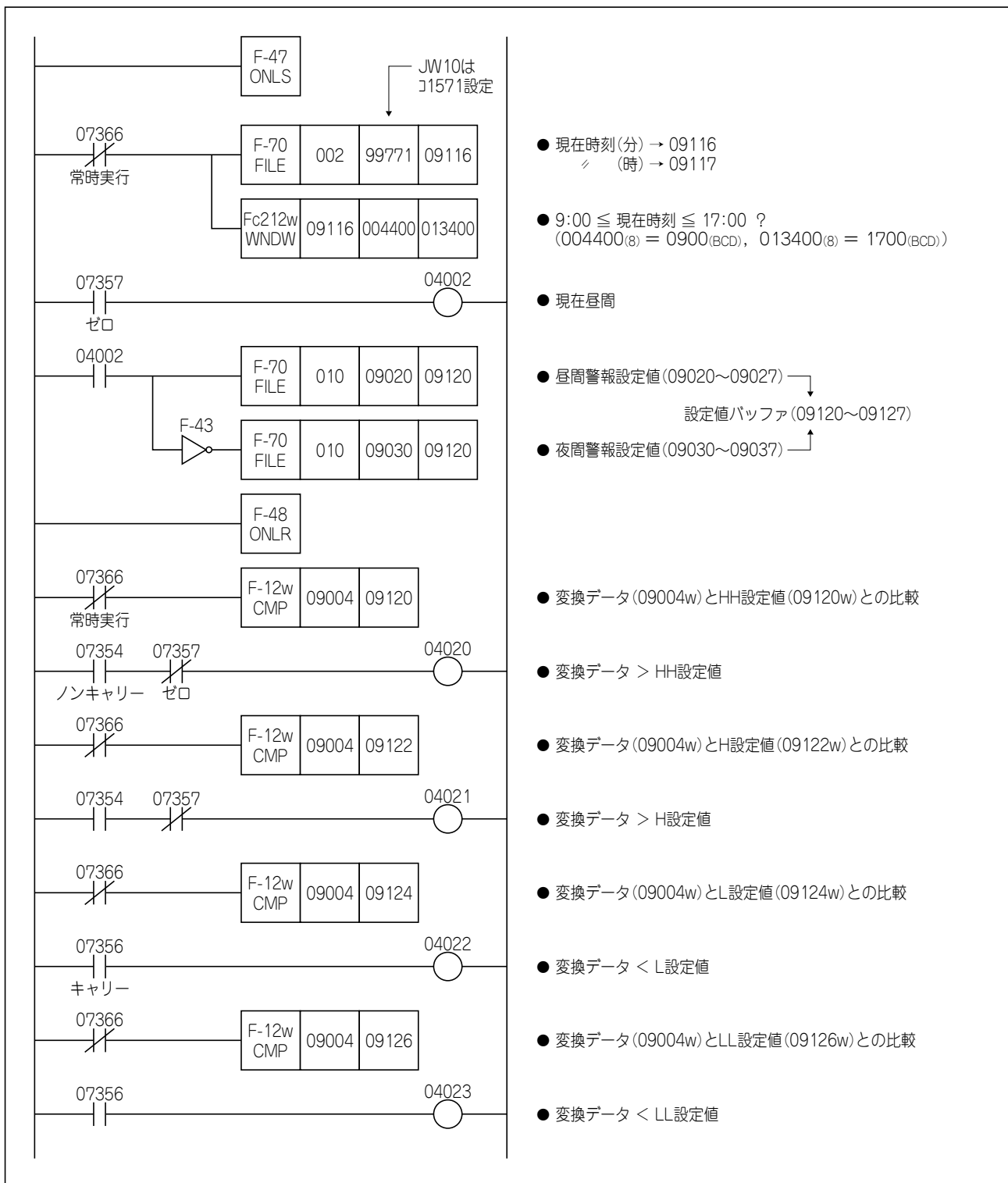
- 200~1000のデータを設定により任意のスケールに変換します。

$$\text{変換値} = \frac{(\text{データ} - 200) \times (\text{フルスケール} - \text{ベーススケール})}{800} + \text{ベーススケール}$$



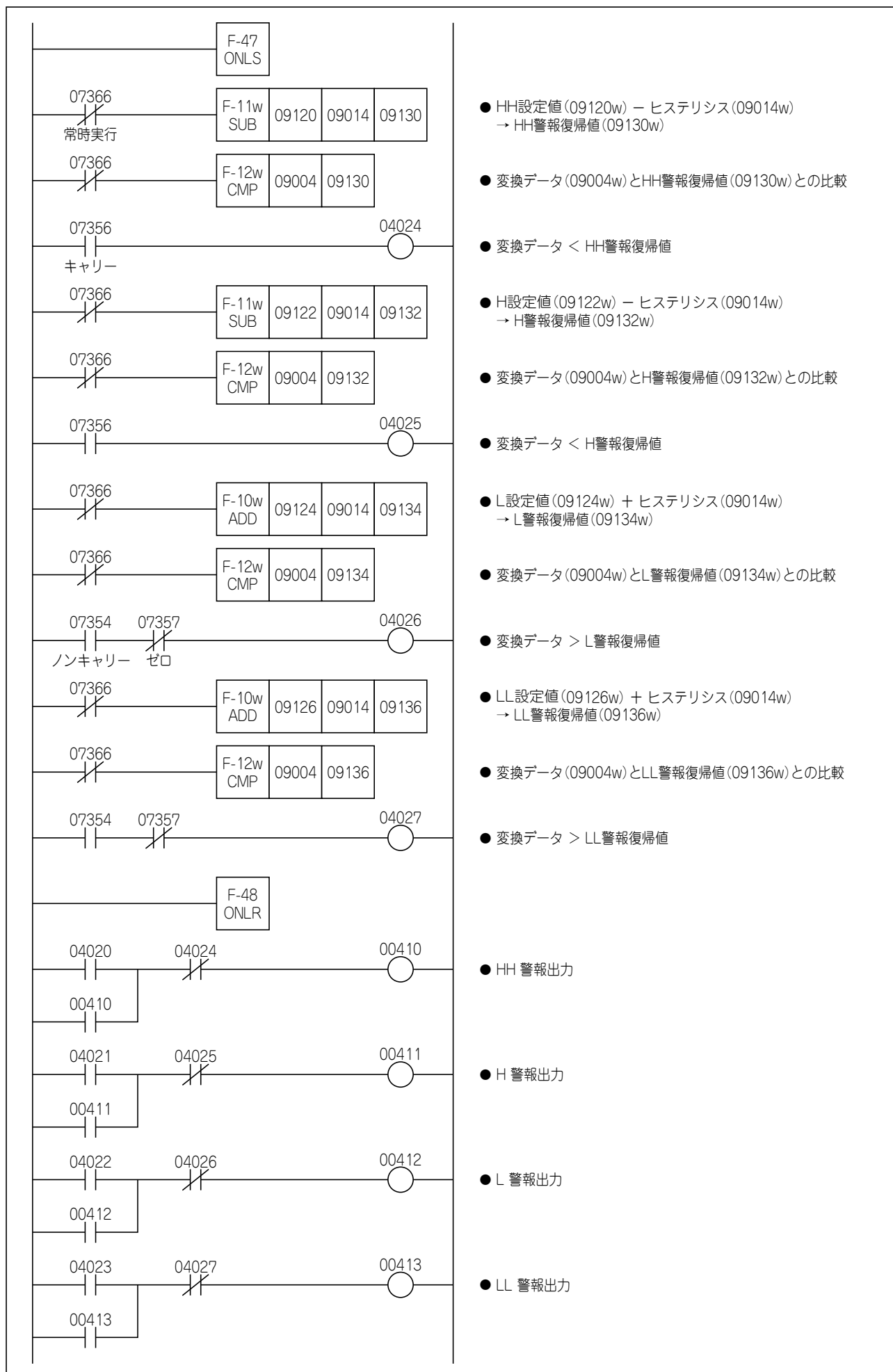
#### (4) 警報設定との比較

- 現在、昼間であれば昼間用の設定値、昼間でなければ夜間用の設定値を比較エリアに設定します。
- 変換データと設定値(HH, H, L, LL)を比較します。
- 変換データとヒステリシス幅より求めた警報復帰値(HH, H, L, LL)を比較します。



- 現在時刻(分) → 09116  
   / (時) → 09117
- 9:00 ≤ 現在時刻 ≤ 17:00 ?  
   (004400(8) = 0900(BCD), 013400(8) = 1700(BCD))
- 現在昼間
- 昼間警報設定値(09020~09027) →  
   設定値バッファ(09120~09127)
- 夜間警報設定値(09030~09037) →
- 変換データ(09004w)とHH設定値(09120w)との比較
- 変換データ > HH設定値
- 変換データ(09004w)とH設定値(09122w)との比較
- 変換データ > H設定値
- 変換データ(09004w)とL設定値(09124w)との比較
- 変換データ < L設定値
- 変換データ(09004w)とLL設定値(09126w)との比較
- 変換データ < LL設定値

(次ページへ続く)



## 4 - 9 昼夜判定処理

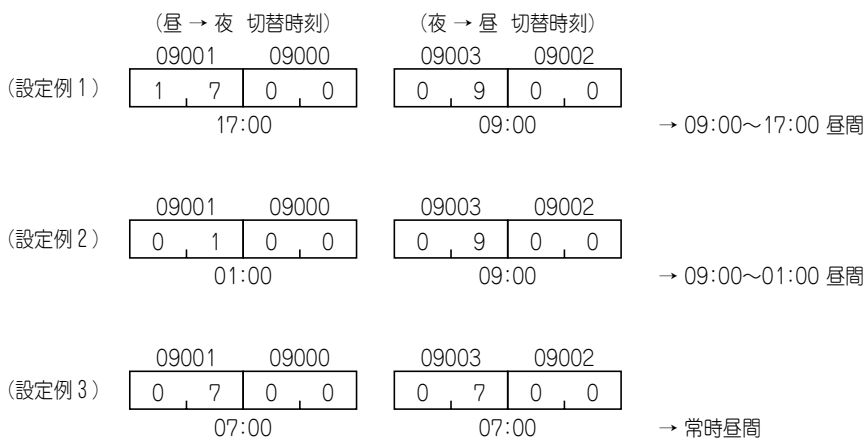
- プログラム例(4-8 スケール変換)等のシステムにおいて、時刻設定により動作を変更する必要がある場合に、その設定値と現在時刻との判定を行います。
- 本プログラム例では、昼夜の判定を行っています。

### 適応PC

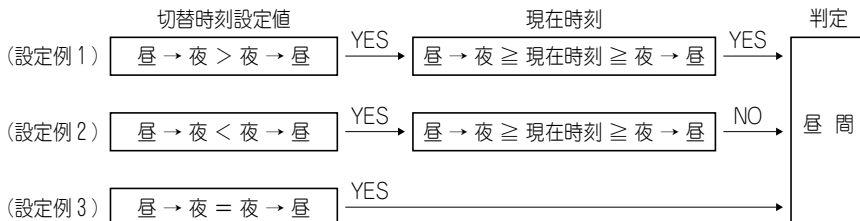
JW10(JW-1424K/1442K/1624K/1642K)  
 JW20H(JW-22CU)  
 JW30H(JW-32CUH/H1, JW-33CUH/H1/H2/H3)  
 JW50H/70H/100H  
 J-board(Z-312J)

} 時計機能のある PC

### 処理内容



- 昼夜の判定は上記3通りの設定を考慮しています。

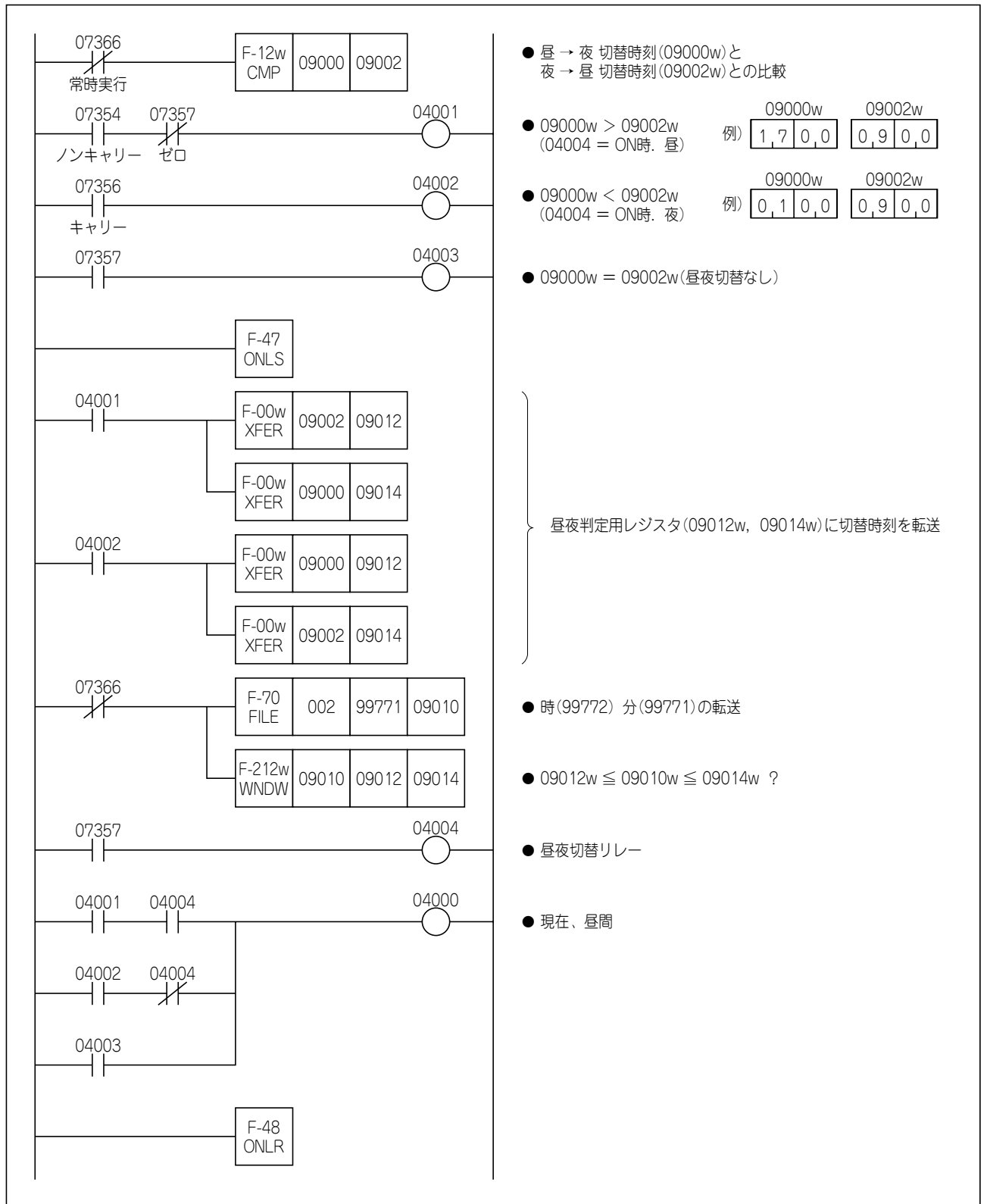


- 上記処理を行うと、現在時刻とまたがる設定をしたときも昼夜の判定が行えます。  
 (例) 現在時刻が16:30のとき、昼→夜設定を17:00から16:00に変更した場合も、その時点で「夜」と判定できます。

### データメモリ

リレー	内 容
04000	現在昼間
04001	04004=ON時. 昼
04002	04004=ON時. 夜
04003	切替時間設定同一
04004	昼夜切替リレー

レジスタ	内 容
09000	昼→夜 切替時刻設定値
09001	〃
09002	夜→昼 切替時刻設定値
09003	〃
09010	現在時刻データバッファ
09011	〃
09012	時刻比較用バッファ
09013	〃
09014	〃
09015	〃

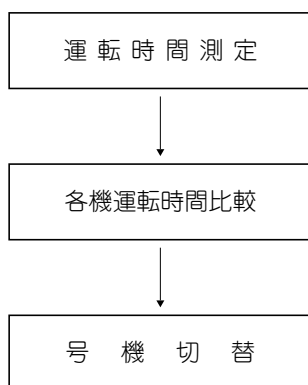


## 4 - 10 運転時間の切替

- バックアップ用1台を含め、2台のモーターが接続されており、常時運転は1台とする場合に、運転の切替を運転時間が均等になるように行います。

適応PC 全JWシリーズ

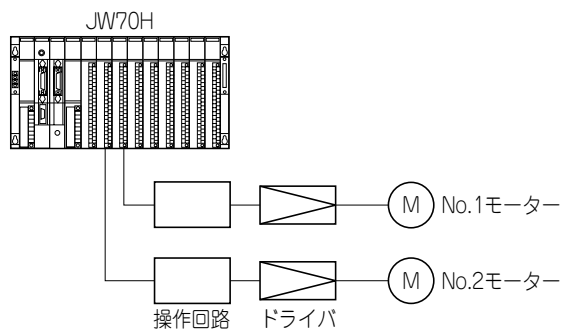
処理内容



- ・ No.1運転出力と1秒クロックで No.1運転時間を測定
- ・ No.2運転出力と1秒クロックで No.2運転時間を測定

- ・ No.1運転時間とNo.2運転時間を比較し、大きい方から小さい方の運転時間を減算する。

- ・ 2台の運転時間の差と許容設定値を比較し、許容設定値よりも差の方が大きければ、他機に切替える。
- ・ 自動時は運転時間差により出力し、自動以外は手動運転指示入力により運転出力する。

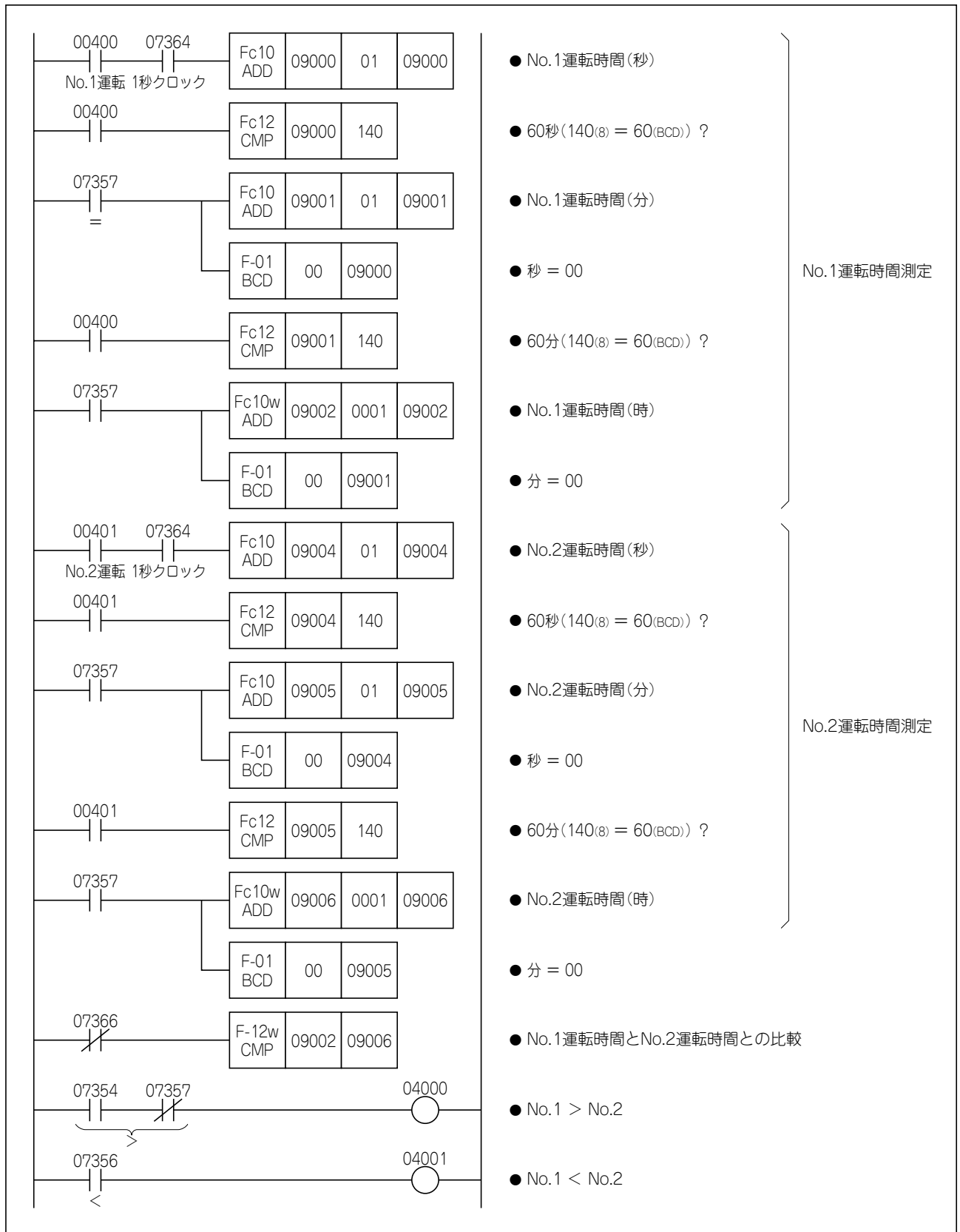


データメモリ

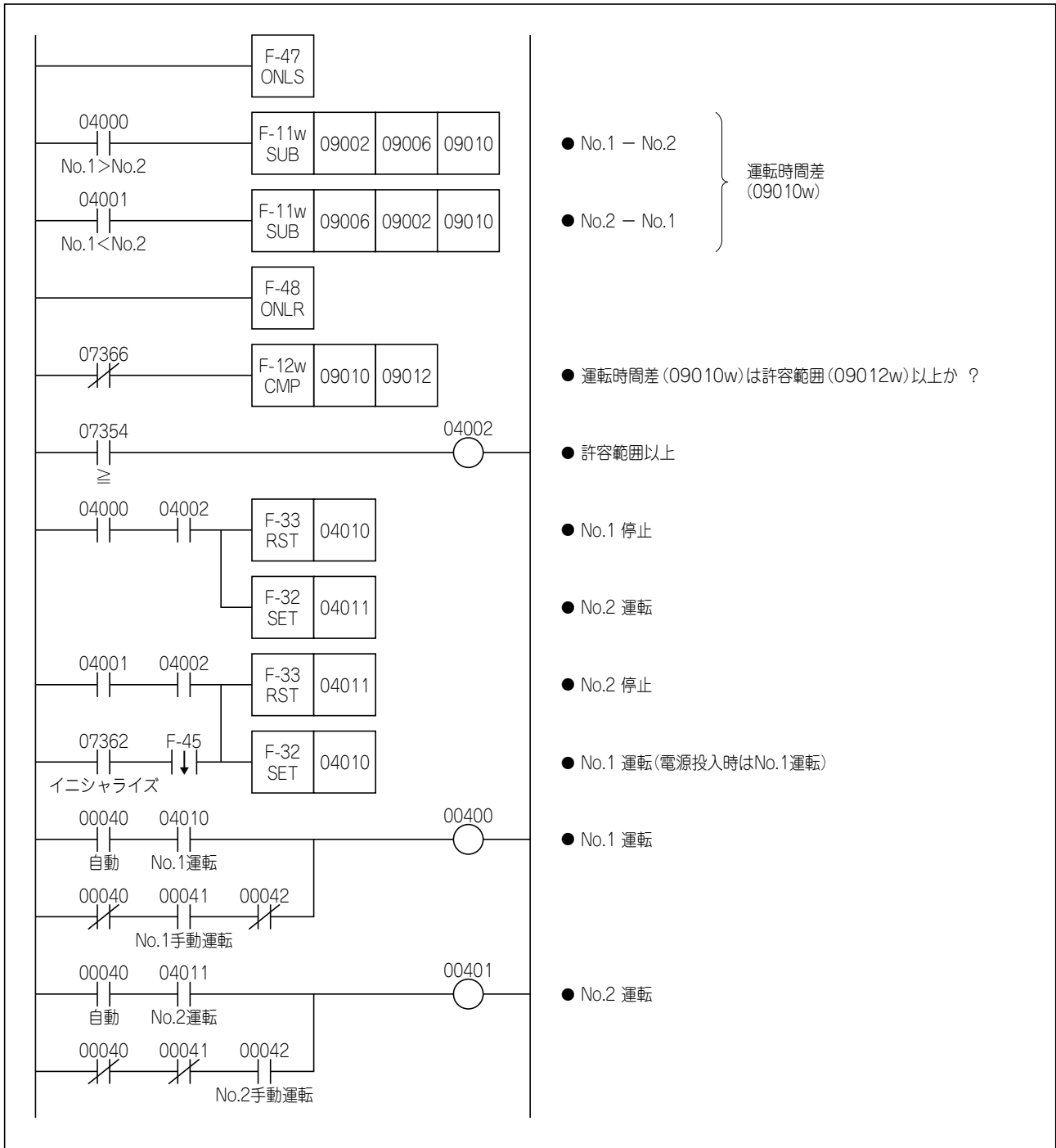
リレー	内 容
00040	自動
00041	No.1手動運転
00042	No.2手動運転
00400	No.1運転
00401	No.2運転
04000	No.1>No.2(運転時間差)
04001	No.1<No.2( 〃 )
04002	運転時間差>許容設定値
04010	No.1選択中
04011	No.2選択中

レジスタ	内 容
09000	No.1運転時間 (秒)
09001	〃 (分)
09002	〃 (時)
09003	〃 (時)
09004	No.2運転時間 (秒)
09005	〃 (分)
09006	〃 (時)
09007	〃 (時)
09010	運転時間の差
09011	〃
09012	運転時間差許容設定値
09013	〃





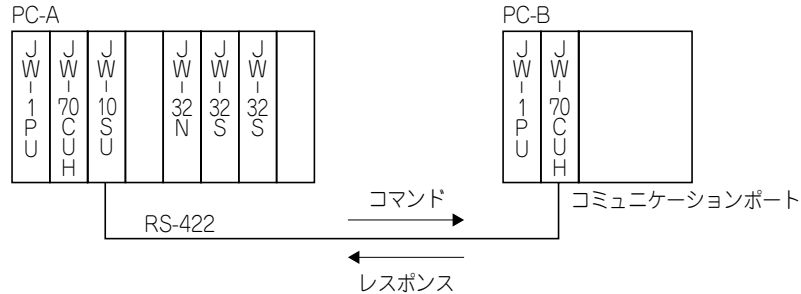
(次ページへ続く)



## 4-11 コンピュータリンクによるPC間通信

- シリアル/FユニットJW-10SUとJW70H/100Hのコミュニケーションポート間で、コンピュータリンクによるPC間のデータ通信を行います。

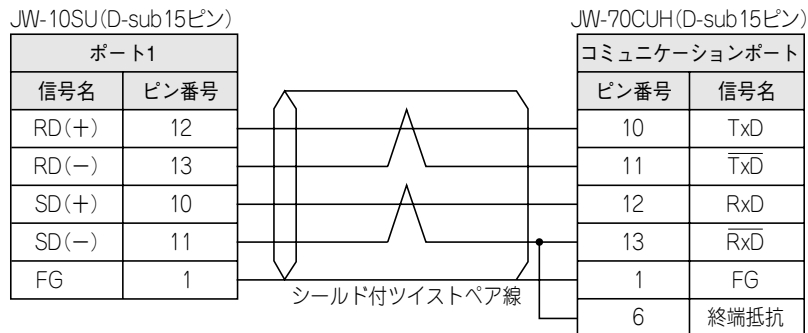
**注1** コンピュータリンクに関する詳細は、リンクユニットJW-10CM, JW-21CMの取扱説明書をご参照ください。



### 適応PC

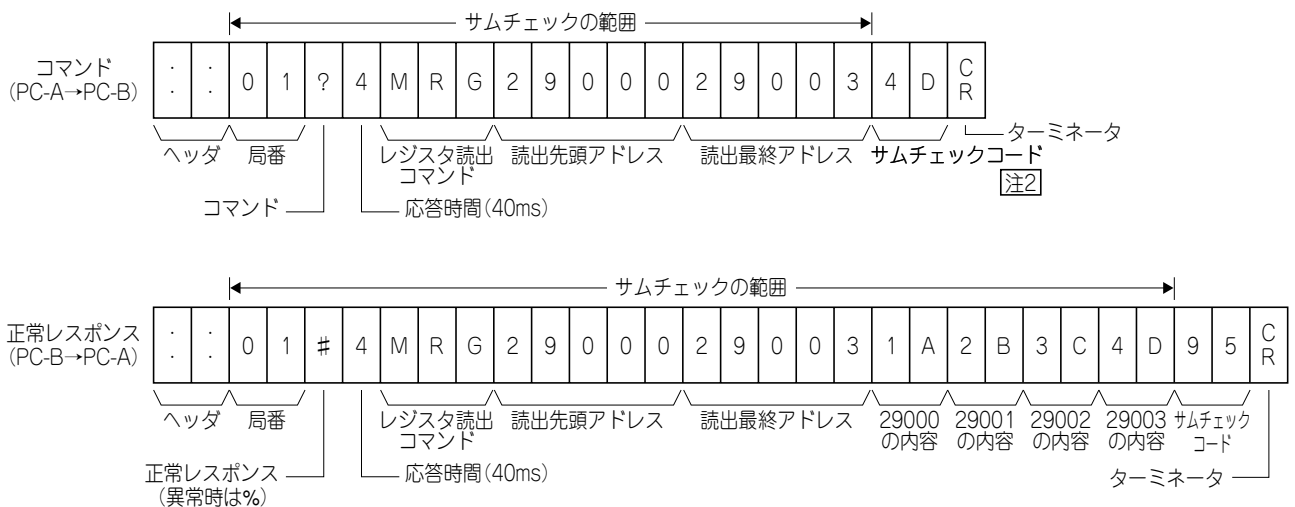
PC-A : JW50H/70H/100H  
 PC-B : JW50H/70H/100H  
 JW30H(JW-32CUH/H1, JW-33CUH/H1/H2/H3)  
 JW20H(JW-22CU)

### 配線方法



### 通信内容

レジスタ29000~29003の4バイトデータをモニタ



注2 サムチェックコードは局番の先頭からサムチェックコードの直前までをASCIIコードのまま加え、2の補数をとったものです。

$$\begin{aligned}
 & 30 + 31 + 3F + 34 + 4D + 52 + 47 + 32 + 39 + 30 + 30 + 30 + 32 + 39 + 30 + 30 + 33 \\
 & (0) (1) (?) (4) (M) (R) (G) (2) (9) (0) (0) (0) (2) (9) (0) (0) (3) \\
 & = 3B3 \rightarrow B3 \xrightarrow{2\text{の補数}} 4D
 \end{aligned}$$

- 2の補数とは2進数で表わしたデータのすべてのビットを反転(0→1, 1→0)し、1を加えた値です。
- プログラムでは、サムチェックコードの計算をF-77, F-78で行っています。

$$\begin{aligned}
 B3^{(H)} & \rightarrow 10110011 \\
 & \downarrow \text{ビット反転} \\
 01001100 & \\
 & \downarrow +1 \\
 01001101 & \rightarrow 4D^{(H)}
 \end{aligned}$$

**JW-10SU スイッチ設定**

スイッチ	内 容	設 定	
SW1	表示切替	0 (ユニット動作表示)	
SW2	4	ポート0終端抵抗	—
	3	ポート1終端抵抗	ON
	2	未使用	—
	1	モード切替	パラメータ設定時：OFF 通信時：ON

**JW-10SU パラメータ設定**

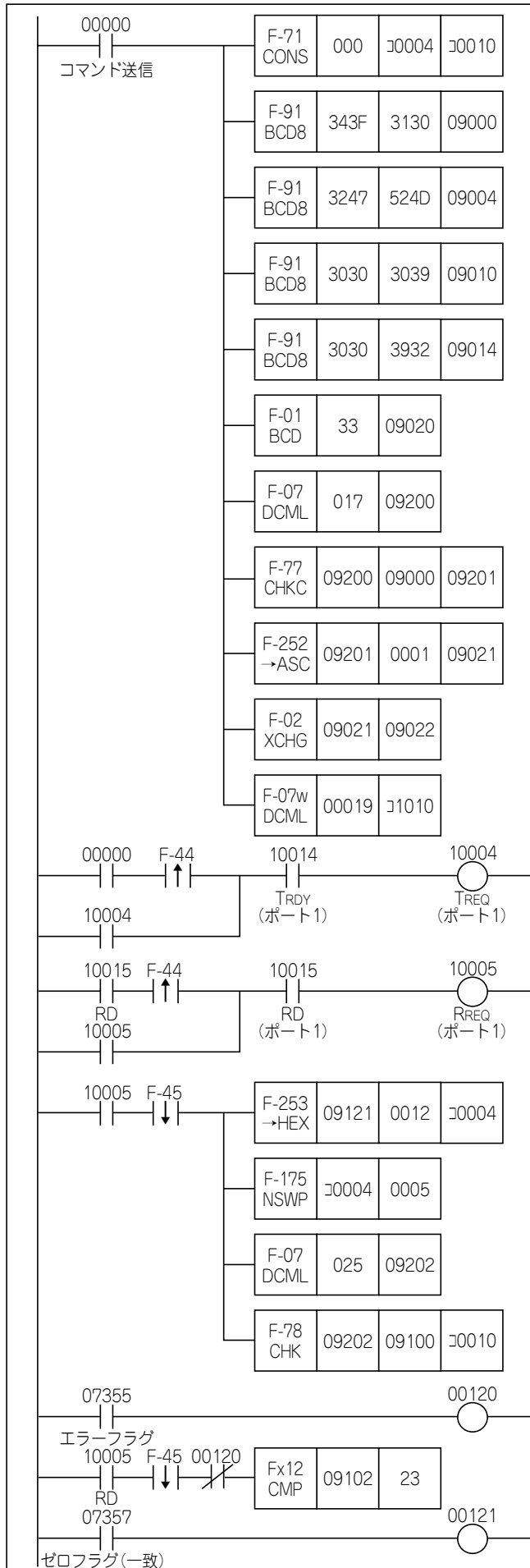
アドレス(8)	内 容	設定値(H)	設定内容
000000	} フラグ先頭アドレス	00	} 31000~
000001		02	
000002		00	
000100	ポート1 伝送手順	01	無手順
000101	ポート1 伝送速度	01	19200ビット/s
000102	ポート1 伝送方式	02	全二重
000103	ポート1 パリティ	03	偶数
000104	ポート1 通信回線	03	RS-422 4線式
000105	ポート1 データ長	01	7ビット
000106	ポート1 ストップビット	02	2ビット
000107	ポート1 伝送コード変換	02	無変換
000110	ポート1 制御信号	03	なし
000111	ポート1 制御キャラクタ	01	EXP1

アドレス(8)	内 容	設定値(H)	設定内容
000112	} EXP1 ヘッダ	3A	:
000113		3A	:
000114		00	NUL
000115		00	NUL
000116	} EXP1 ターミネータ	0D	CR
000117		00	NUL
000126	} 最大テキスト長	40	64バイト
000127		00	
000130	} 送信データ先頭アドレス	00	09000~
000131		08	(ファイルアドレス
000132		00	004000)
000133	} 受信データ先頭アドレス	40	09100~
000134		08	(ファイルアドレス
000135		00	004100)
003777	スタート準備スイッチ	81→01	通信動作可能

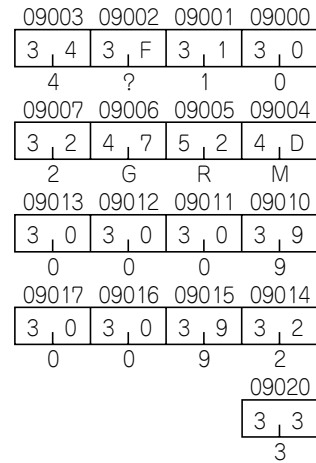
**PC-B システムメモリ設定**

アドレス(8)	内 容	設定値(H)	設 定 内 容
#236	コミュニケーションポート伝送仕様	30	19200ビット/s, 偶数パリティ, ストップビット2ビット
#237	コミュニケーションポート局番	01	局番01

PC-A プログラム



● データ表示用レジスタのクリア



送信データの設定

● サムチェックコード計算データ数の設定

● 09000~09020の17バイトのサムチェックコードを09201に格納



サムチェックコードを送信エリアに設定

● 送信バイト数の設定

● データ送信

● データ受信

● 09121~09132の10バイト(8進で0012)の内容(ASCII)を16進に変換し、00004~00010に格納

● 00004~00010の5バイトの上下ビットの交換

00004	1 A	29000の内容
00005	2 B	29001の内容
00006	3 C	29002の内容
00007	4 D	29003の内容
00010	9 5	サムチェックコード

● 09100~09130の25バイトのサムチェックコードを計算し、受信したサムチェックコード00010と比較

● 一致しないとき、サムチェックエラー

● 受信応答チェック(#か)

● 正常応答

# 付録 命令語一覧

命令語	シンボル	機能	P C機種名 (J-boardはJW20Hに含む)			
			JW10	JW20H	JW30H	JW50H/70H/100H
STR		a 接点で論理を開始。中間結果の記憶	○	○	○	○
STR NOT		b 接点で論理を開始。中間結果の記憶	○	○	○	○
AND		論理積	○	○	○	○
AND NOT		論理積否定	○	○	○	○
OR		論理和	○	○	○	○
OR NOT		論理和否定	○	○	○	○
AND STR		中間結果との論理積	○	○	○	○
OR STR		中間結果との論理和	○	○	○	○
OUT		演算結果の出力	○	○	○	○
TMR		タイマ(減算式) ① スタート入力(ONで計数) ② TMR番号 ③ 設定値 (0.1~199.9秒) (0.01~19.99秒)	○※1	○	○	○
DTMR (BCD)		タイマ(減算式) ① スタート入力(ONで計数) ② TMR番号(000~777) ③ 設定値(0.1~799.9秒)	×	○	○	○
DTMR (BIN)		タイマ(減算式) ① スタート入力(ONで計数) ② TMR番号(000~777) ③ 設定値(0.1~3276.7秒)	×	○	○	○
UTMR (BCD)		タイマ(加算式) ① スタート入力(ONで計数) ② TMR番号(000~777) ③ 設定値(0.1~799.9秒)	×	○	○	○
UTMR (BIN)		タイマ(加算式) ① スタート入力(ONで計数) ② TMR番号(000~777) ③ 設定値(0.1~3276.7秒)	×	○	○	○
CNT		カウンタ(減算式) ① 計数入力 ③ CNT番号 ② リセット入力 ④ 設定値(1~1999)	○※1	○	○	○
DCNT (BCD)		カウンタ(減算式) ① 計数入力 ③ CNT番号(000~777) ② リセット入力 ④ 設定値(1~7999)	×	○	○	○
DCNT (BIN)		カウンタ(減算式) ① 計数入力 ③ CNT番号(000~777) ② リセット入力 ④ 設定値(1~32767)	×	○	○	○
UCNT (BCD)		カウンタ(加算式) ① 計数入力 ③ CNT番号(000~777) ② リセット入力 ④ 設定値(1~7999)	×	○	○	○
UCNT (BIN)		カウンタ(加算式) ① 計数入力 ③ CNT番号(000~777) ② リセット入力 ④ 設定値(1~32767)	×	○	○	○
MD		メンテナンスディスプレイ ①, ②, ③ 入力情報 ⑥ MD番号(000~777) ④ 出力指示端子 ⑦ MDデータ(000~999) ⑤ 拡張出力	×	○	○	○

※1 TMR設定値③、CNT設定値④をレジスタ指定できます。

命令語	シンボル	機能	P C機種名 (J-boardはJW20Hを含む)							
			JW10	JW20H	JW30H	JW50H/70H/100H				
F-00	<table border="1"><tr><td>F-00 XFER</td><td>S</td><td>D</td></tr></table>	F-00 XFER	S	D	データレジスタ間の1バイト転送	○	○	○	○	
F-00 XFER	S	D								
F-00w	<table border="1"><tr><td>F-00w XFER</td><td>S</td><td>D</td></tr></table>	F-00w XFER	S	D	データレジスタ間の1ワード転送	○	○	○	○	
F-00w XFER	S	D								
F-00d	<table border="1"><tr><td>F-00d XFER</td><td>S</td><td>D</td></tr></table>	F-00d XFER	S	D	データレジスタ間の2ワード転送	×	○	○	○	
F-00d XFER	S	D								
F-01	<table border="1"><tr><td>F-01 BCD</td><td>n</td><td>D</td></tr></table>	F-01 BCD	n	D	BCD定数(2桁)の転送	○	○	○	○	
F-01 BCD	n	D								
F-01w	<table border="1"><tr><td>F-01w BCD</td><td>n</td><td>D</td></tr></table>	F-01w BCD	n	D	BCD定数(4桁)の転送	○	○	○	○	
F-01w BCD	n	D								
F-02	<table border="1"><tr><td>F-02 XCHG</td><td>D1</td><td>D2</td></tr></table>	F-02 XCHG	D1	D2	レジスタ間(1バイト)のデータ交換	○	○	○	○	
F-02 XCHG	D1	D2								
F-02w	<table border="1"><tr><td>F-02w XCHG</td><td>D1</td><td>D2</td></tr></table>	F-02w XCHG	D1	D2	レジスタ間(1ワード)のデータ交換	○	○	○	○	
F-02w XCHG	D1	D2								
F-02d	<table border="1"><tr><td>F-02d XCHG</td><td>D1</td><td>D2</td></tr></table>	F-02d XCHG	D1	D2	レジスタ間(2ワード)のデータ交換	×	○	○	○	
F-02d XCHG	D1	D2								
F-03	<table border="1"><tr><td>F-03 →BIN</td><td>S</td><td>D</td></tr></table>	F-03 →BIN	S	D	BCD(2桁)→BIN(8ビット)変換	○	○	○	○	
F-03 →BIN	S	D								
F-03w	<table border="1"><tr><td>F-03w →BIN</td><td>S</td><td>D</td></tr></table>	F-03w →BIN	S	D	BCD(4桁)→BIN(16ビット)変換	○	○	○	○	
F-03w →BIN	S	D								
F-04	<table border="1"><tr><td>F-04 →BCD</td><td>S</td><td>D</td></tr></table>	F-04 →BCD	S	D	BIN(8ビット)→BCD(2桁)変換	○	○	○	○	
F-04 →BCD	S	D								
F-04w	<table border="1"><tr><td>F-04w →BCD</td><td>S</td><td>D</td></tr></table>	F-04w →BCD	S	D	BIN(16ビット)→BCD(6桁)変換	○	○	○	○	
F-04w →BCD	S	D								
F-05	<table border="1"><tr><td>F-05 DMPX</td><td>S</td><td>D</td></tr></table>	F-05 DMPX	S	D	1バイトデータの分配	×	○	○	○	
F-05 DMPX	S	D								
F-05w	<table border="1"><tr><td>F-05w DMPX</td><td>S</td><td>D</td></tr></table>	F-05w DMPX	S	D	1ワードデータの分配	×	○	○	○	
F-05w DMPX	S	D								
F-06	<table border="1"><tr><td>F-06 MPX</td><td>S</td><td>D</td></tr></table>	F-06 MPX	S	D	1バイトデータの抽出	×	○	○	○	
F-06 MPX	S	D								
F-06w	<table border="1"><tr><td>F-06w MPX</td><td>S</td><td>D</td></tr></table>	F-06w MPX	S	D	1ワードデータの抽出	×	○	○	○	
F-06w MPX	S	D								
F-07	<table border="1"><tr><td>F-07 DCML</td><td>n</td><td>D</td></tr></table>	F-07 DCML	n	D	10進定数(1バイト)の転送	○	○	○	○	
F-07 DCML	n	D								
F-07w	<table border="1"><tr><td>F-07w DCML</td><td>n</td><td>D</td></tr></table>	F-07w DCML	n	D	10進定数(1ワード)の転送	○	○	○	○	
F-07w DCML	n	D								
F-08	<table border="1"><tr><td>F-08 OCT</td><td>n</td><td>D</td></tr></table>	F-08 OCT	n	D	8進定数(1バイト)の転送	○	○	○	○	
F-08 OCT	n	D								
F-08w	<table border="1"><tr><td>F-08w OCT</td><td>n</td><td>D</td></tr></table>	F-08w OCT	n	D	8進定数(1ワード)の転送	○	○	○	○	
F-08w OCT	n	D								
F-09	<table border="1"><tr><td>F-09 INV</td><td>S</td><td>D</td></tr></table>	F-09 INV	S	D	8ビットデータの反転	○	○	○	○	
F-09 INV	S	D								
F-09w	<table border="1"><tr><td>F-09w INV</td><td>S</td><td>D</td></tr></table>	F-09w INV	S	D	16ビットデータの反転	○	○	○	○	
F-09w INV	S	D								
F-09d	<table border="1"><tr><td>F-09d INV</td><td>S</td><td>D</td></tr></table>	F-09d INV	S	D	32ビットデータの反転	×	○	○	○	
F-09d INV	S	D								
F-10	<table border="1"><tr><td>F-10 ADD</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-10 ADD	S1	S2	D	レジスタ間(BCD2桁)の加算	○	○	○	○
F-10 ADD	S1	S2	D							
F-10w	<table border="1"><tr><td>F-10w ADD</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-10w ADD	S1	S2	D	レジスタ間(BCD4桁)の加算	○	○	○	○
F-10w ADD	S1	S2	D							
F-10d	<table border="1"><tr><td>F-10d ADD</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-10d ADD	S1	S2	D	レジスタ間(BCD8桁)の加算	×	○	○	○
F-10d ADD	S1	S2	D							
Fc10	<table border="1"><tr><td>Fc10 ADD</td><td>S1</td><td>n</td><td>D</td></tr></table>	Fc10 ADD	S1	n	D	レジスタ(BCD2桁)と定数(2桁)の加算	○	○	○	○
Fc10 ADD	S1	n	D							
Fc10w	<table border="1"><tr><td>Fc10w ADD</td><td>S1</td><td>n</td><td>D</td></tr></table>	Fc10w ADD	S1	n	D	レジスタ(BCD4桁)と定数(4桁)の加算	○	○	○	○
Fc10w ADD	S1	n	D							
Fc10d	<table border="1"><tr><td>Fc10d ADD</td><td>S1</td><td>n</td><td>D</td></tr></table>	Fc10d ADD	S1	n	D	レジスタ(BCD8桁)と定数(4桁)の加算	×	○	○	○
Fc10d ADD	S1	n	D							
F-11	<table border="1"><tr><td>F-11 SUB</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-11 SUB	S1	S2	D	レジスタ間(BCD2桁)の減算	○	○	○	○
F-11 SUB	S1	S2	D							
F-11w	<table border="1"><tr><td>F-11w SUB</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-11w SUB	S1	S2	D	レジスタ間(BCD4桁)の減算	○	○	○	○
F-11w SUB	S1	S2	D							
F-11d	<table border="1"><tr><td>F-11d SUB</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-11d SUB	S1	S2	D	レジスタ間(BCD8桁)の減算	×	○	○	○
F-11d SUB	S1	S2	D							

命令語	シンボル	機能	P C機種名 (J-boardはJW20Hを含む)							
			JW10	JW20H	JW30H	JW50H/70H/100H				
Fc11	<table border="1"><tr><td>Fc11 SUB</td><td>S1</td><td>n</td><td>D</td></tr></table>	Fc11 SUB	S1	n	D	レジスタ(BCD2桁)と定数(2桁)の減算	○	○	○	○
Fc11 SUB	S1	n	D							
Fc11w	<table border="1"><tr><td>Fc11w SUB</td><td>S1</td><td>n</td><td>D</td></tr></table>	Fc11w SUB	S1	n	D	レジスタ(BCD4桁)と定数(4桁)の減算	○	○	○	○
Fc11w SUB	S1	n	D							
Fc11d	<table border="1"><tr><td>Fc11d SUB</td><td>S1</td><td>n</td><td>D</td></tr></table>	Fc11d SUB	S1	n	D	レジスタ(BCD8桁)と定数(4桁)の減算	×	○	○	○
Fc11d SUB	S1	n	D							
F-12	<table border="1"><tr><td>F-12 CMP</td><td>S1</td><td>S2</td></tr></table>	F-12 CMP	S1	S2	レジスタ間(1バイト)の比較	○	○	○	○	
F-12 CMP	S1	S2								
F-12w	<table border="1"><tr><td>F-12w CMP</td><td>S1</td><td>S2</td></tr></table>	F-12w CMP	S1	S2	レジスタ間(1ワード)の比較	○	○	○	○	
F-12w CMP	S1	S2								
F-12d	<table border="1"><tr><td>F-12d CMP</td><td>S1</td><td>S2</td></tr></table>	F-12d CMP	S1	S2	レジスタ間(2ワード)の比較	×	○	○	○	
F-12d CMP	S1	S2								
Fc12	<table border="1"><tr><td>Fc12 CMP</td><td>S1</td><td>n</td></tr></table>	Fc12 CMP	S1	n	レジスタと8進定数(1バイト)の比較	○	○	○	○	
Fc12 CMP	S1	n								
Fc12w	<table border="1"><tr><td>Fc12w CMP</td><td>S1</td><td>n</td></tr></table>	Fc12w CMP	S1	n	レジスタと8進定数(1ワード)の比較	○	○	○	○	
Fc12w CMP	S1	n								
Fx12	<table border="1"><tr><td>Fx12 CMP</td><td>S1</td><td>n</td></tr></table>	Fx12 CMP	S1	n	レジスタと16進定数(1バイト)の比較	×	×	○	○	
Fx12 CMP	S1	n								
Fx12w	<table border="1"><tr><td>Fx12w CMP</td><td>S1</td><td>n</td></tr></table>	Fx12w CMP	S1	n	レジスタと16進定数(1ワード)の比較	×	×	○	○	
Fx12w CMP	S1	n								
F-13	<table border="1"><tr><td>F-13 AND</td><td>S</td><td>D</td></tr></table>	F-13 AND	S	D	レジスタ間(1バイト)の論理積	○	○	○	○	
F-13 AND	S	D								
F-13w	<table border="1"><tr><td>F-13w AND</td><td>S</td><td>D</td></tr></table>	F-13w AND	S	D	レジスタ間(1ワード)の論理積	○	○	○	○	
F-13w AND	S	D								
F-13d	<table border="1"><tr><td>F-13d AND</td><td>S</td><td>D</td></tr></table>	F-13d AND	S	D	レジスタ間(2ワード)の論理積	×	○	○	○	
F-13d AND	S	D								
Fc13	<table border="1"><tr><td>Fc13 AND</td><td>n</td><td>D</td></tr></table>	Fc13 AND	n	D	レジスタと8進定数(1バイト)の論理積	○	○	○	○	
Fc13 AND	n	D								
Fc13w	<table border="1"><tr><td>Fc13w AND</td><td>n</td><td>D</td></tr></table>	Fc13w AND	n	D	レジスタと8進定数(1ワード)の論理積	○	○	○	○	
Fc13w AND	n	D								
Fx13	<table border="1"><tr><td>Fx13 AND</td><td>n</td><td>D</td></tr></table>	Fx13 AND	n	D	レジスタと16進定数(1バイト)の論理積	×	×	○	○	
Fx13 AND	n	D								
Fx13w	<table border="1"><tr><td>Fx13w AND</td><td>n</td><td>D</td></tr></table>	Fx13w AND	n	D	レジスタと16進定数(1ワード)の論理積	×	×	○	○	
Fx13w AND	n	D								
F-14	<table border="1"><tr><td>F-14 OR</td><td>S</td><td>D</td></tr></table>	F-14 OR	S	D	レジスタ間(1バイト)の論理和	○	○	○	○	
F-14 OR	S	D								
F-14w	<table border="1"><tr><td>F-14w OR</td><td>S</td><td>D</td></tr></table>	F-14w OR	S	D	レジスタ間(1ワード)の論理和	○	○	○	○	
F-14w OR	S	D								
F-14d	<table border="1"><tr><td>F-14d OR</td><td>S</td><td>D</td></tr></table>	F-14d OR	S	D	レジスタ間(2ワード)の論理和	×	○	○	○	
F-14d OR	S	D								
Fc14	<table border="1"><tr><td>Fc14 OR</td><td>n</td><td>D</td></tr></table>	Fc14 OR	n	D	レジスタと8進定数(1バイト)の論理和	○	○	○	○	
Fc14 OR	n	D								
Fc14w	<table border="1"><tr><td>Fc14w OR</td><td>n</td><td>D</td></tr></table>	Fc14w OR	n	D	レジスタと8進定数(1ワード)の論理和	○	○	○	○	
Fc14w OR	n	D								
Fx14	<table border="1"><tr><td>Fx14 OR</td><td>n</td><td>D</td></tr></table>	Fx14 OR	n	D	レジスタと16進定数(1バイト)の論理和	×	×	○	○	
Fx14 OR	n	D								
Fx14w	<table border="1"><tr><td>Fx14w OR</td><td>n</td><td>D</td></tr></table>	Fx14w OR	n	D	レジスタと16進定数(1ワード)の論理和	×	×	○	○	
Fx14w OR	n	D								
F-15	<table border="1"><tr><td>F-15 MUL</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-15 MUL	S1	S2	D	レジスタ間(BCD4桁)の乗算	○	○	○	○
F-15 MUL	S1	S2	D							
F-15d	<table border="1"><tr><td>F-15d MUL</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-15d MUL	S1	S2	D	レジスタ間(BCD8桁)の乗算	×	○	○	○
F-15d MUL	S1	S2	D							
Fc15	<table border="1"><tr><td>Fc15 MUL</td><td>S1</td><td>n</td><td>D</td></tr></table>	Fc15 MUL	S1	n	D	レジスタ間(BCD4桁)とBCD定数(3桁)の乗算	○	○	○	○
Fc15 MUL	S1	n	D							
Fc15d	<table border="1"><tr><td>Fc15d MUL</td><td>S1</td><td>n</td><td>D</td></tr></table>	Fc15d MUL	S1	n	D	レジスタ間(BCD8桁)とBCD定数(4桁)の乗算	×	○	○	○
Fc15d MUL	S1	n	D							
F-16	<table border="1"><tr><td>F-16 DIV</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-16 DIV	S1	S2	D	レジスタ(BCD4桁)とレジスタ(BCD2桁)の除算	○	○	○	○
F-16 DIV	S1	S2	D							
F-16d	<table border="1"><tr><td>F-16d DIV</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-16d DIV	S1	S2	D	レジスタ(BCD8桁)とレジスタ(BCD8桁)の除算	×	○	○	○
F-16d DIV	S1	S2	D							
Fc16	<table border="1"><tr><td>Fc16 DIV</td><td>S1</td><td>n</td><td>D</td></tr></table>	Fc16 DIV	S1	n	D	レジスタ(BCD4桁)とBCD定数(2桁)の除算	○	○	○	○
Fc16 DIV	S1	n	D							
Fc16d	<table border="1"><tr><td>Fc16d DIV</td><td>S1</td><td>n</td><td>D</td></tr></table>	Fc16d DIV	S1	n	D	レジスタ(BCD8桁)とBCD定数(4桁)の除算	×	○	○	○
Fc16d DIV	S1	n	D							



命令語	シンボル	機能	P C機種名 (J-boardはJW20Hを含む)			
			JW10	JW20H	JW30H	JW50H/70H/100H
F-17		レジスタ間(1バイト)の一致	×	○	○	○
F-17w		レジスタ間(1ワード)の一致	×	○	○	○
F-17d		レジスタ間(2ワード)の一致	×	○	○	○
Fc17		レジスタと8進定数(1バイト)の一致	×	○	○	○
Fc17w		レジスタと8進定数(1ワード)の一致	×	○	○	○
Fx17		レジスタと16進定数(1バイト)の一致	×	×	○	○
Fx17w		レジスタと16進定数(1ワード)の一致	×	×	○	○
F-18		レジスタ間(1バイト)の排他的論理和	○	○	○	○
F-18w		レジスタ間(1ワード)の排他的論理和	○	○	○	○
F-18d		レジスタ間(2ワード)の排他的論理和	×	○	○	○
Fc18		レジスタと8進定数(1バイト)の排他的論理和	○	○	○	○
Fc18w		レジスタと8進定数(1ワード)の排他的論理和	○	○	○	○
Fx18		レジスタと16進定数(1バイト)の排他的論理和	×	×	○	○
Fx18w		レジスタと16進定数(1ワード)の排他的論理和	×	×	○	○
F-20		メンテナンスディスプレイ ①, ②, ③ 入力情報 ⑥ MD番号 (000~777) ④ 出力指示端子 ⑦ MDデータ (000~999) ⑤ 拡張出力	×	○	○	○
F-21		レジスタ(BCD8桁)の平方根	×	○	○ <sup>※3</sup>	○
F-22		三角関数(SIN)の演算	×	×	○ <sup>※3</sup>	○
F-23		三角関数(COS)の演算	×	×	○ <sup>※3</sup>	○
F-24		三角関数(TAN)の演算	×	×	○ <sup>※3</sup>	○
F-25		三角関数(SIN <sup>-1</sup> )の演算	×	×	○ <sup>※3</sup>	○
F-26		三角関数(COS <sup>-1</sup> )の演算	×	×	○ <sup>※3</sup>	○
F-27		三角関数(TAN <sup>-1</sup> )の演算	×	×	○ <sup>※3</sup>	○
F-28		直交座標系(X,Y)データの極座標系(γ,θ)への変換	×	×	○ <sup>※3</sup>	○
F-29		極座標系(γ,θ)データの直交座標系(X,Y)への変換	×	×	○ <sup>※3</sup>	○
F-30		マスターコントロールのセット	○	○	○	○
F-31		マスターコントロールのリセット	○	○	○	○
F-32		セットコイル	○	○	○	○
F-33		リセットコイル	○	○	○	○
F-34		時計の現在値との比較 (指定リレーのセット)	○ <sup>※1</sup>	○ <sup>※2</sup>	○ <sup>※3</sup>	○
F-35		時計の現在値との比較 (指定リレーのリセット)	○ <sup>※1</sup>	○ <sup>※2</sup>	○ <sup>※3</sup>	○

※1 JW-1424K/1442K/1624K/1642Kでプログラムできます。JW-1324K/1342Kではプログラムできません。

※2 JW-22CU、Z-312Jでプログラムできます。JW-21CU、Z-311Jではプログラムできません。

※3 JW-32CUH/H1、JW-33CUH/H1/H2/H3でプログラムできます。JW-31CUH/H1ではプログラムできません。

命令語	シンボル	機能	P C機種名 (J-boardはJW20Hを含む)			
			JW10	JW20H	JW30H	JW50H/70H/100H
F-36		時計の加算	×	○ <sup>※2</sup>	○ <sup>※3</sup>	○
F-37		時計の減算	×	○ <sup>※2</sup>	○ <sup>※3</sup>	○
F-38		時計現在値の転送	×	○ <sup>※2</sup>	○ <sup>※3</sup>	○
F-40		END命令	○	○	○	○
F-41		ジャンプコントロールのセット	○	○	○	○
F-42		ジャンプコントロールのリセット	○	○	○	○
F-43		ビット反転(ACCの内容を反転)	○	○	○	○
F-44		ON時微分	○	○	○	○
F-45		OFF時微分	○	○	○	○
F-47		レベル演算条件セット	○	○	○	○
F-48		レベル演算条件リセット	○	○	○	○
F-49		条件END	×	○	○	○
F-50		4→16デコーダ	○	○	○	○
F-51		16→4エンコーダ	○	○	○	○
F-52		7SEGデコーダ	○	○	○	○
F-53		BCD(4桁)→BIN(16ビット)変換	○	○	○	○
F-54		BIN(16ビット)→BCD(6桁)変換	○	○	○	○
F-55		上位4ビットと下位4ビットの交換	○	○	○	○
F-56		1バイトデータの10の補数	×	○	○	○
F-56w		1ワードデータの10の補数	×	○	○	○
F-56d		2ワードデータの10の補数	×	○	○	○
F-57		1バイトデータの2の補数	×	○	○	○
F-57w		1ワードデータの2の補数	×	○	○	○
F-57d		2ワードデータの2の補数	×	○	○	○
F-58		ONビット数の合計	○	○	○	○
F-60		両方向シフトレジスタ(1バイト) ① シフト方向指示入力 ③ シフト入力 ② データ入力 ④ リセット入力	○	○	○	○
F-60w		両方向シフトレジスタ(1ワード) ① シフト方向指示入力 ③ シフト入力 ② データ入力 ④ リセット入力	○	○	○	○
F-60d		両方向シフトレジスタ(2ワード) ① シフト方向指示入力 ③ シフト入力 ② データ入力 ④ リセット入力	×	○	○	○

※2 JW-22CU、Z-312Jでプログラムできます。JW-21CU、Z-311Jではプログラムできません。

※3 JW-32CUH/H1、JW-33CUH/H1/H2/H3でプログラムできます。JW-31CUH/H1ではプログラムできません。

命令語	シンボル	機能	P C機種名 (J-boardはJW20Hを含む)											
			JW10	JW20H	JW30H	JW50H/70H/100H								
F-61	①— <table border="1"><tr><td>F-61</td><td>D</td></tr><tr><td>ASFR</td><td></td></tr></table>	F-61	D	ASFR		非同期シフトレジスタ(1バイト) ①シフト方向指示入力 ②シフト入力	×	○	○	○				
F-61	D													
ASFR														
F-61w	①— <table border="1"><tr><td>F-61w</td><td>D</td></tr><tr><td>ASFR</td><td></td></tr></table>	F-61w	D	ASFR		非同期シフトレジスタ(1ワード) ①シフト方向指示入力 ②シフト入力	×	○	○	○				
F-61w	D													
ASFR														
F-61d	①— <table border="1"><tr><td>F-61d</td><td>D</td></tr><tr><td>ASFR</td><td></td></tr></table>	F-61d	D	ASFR		非同期シフトレジスタ(2ワード) ①シフト方向指示入力 ②シフト入力	×	○	○	○				
F-61d	D													
ASFR														
F-62	①— <table border="1"><tr><td>F-62</td><td>D</td></tr><tr><td>U/DC</td><td></td></tr></table>	F-62	D	U/DC		BCD2桁のアップ・ダウンカウンタ 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	○	○	○	○				
F-62	D													
U/DC														
F-62w	①— <table border="1"><tr><td>F-62w</td><td>D</td></tr><tr><td>U/DC</td><td></td></tr></table>	F-62w	D	U/DC		BCD4桁のアップ・ダウンカウンタ 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	○	○	○	○				
F-62w	D													
U/DC														
F-62d	①— <table border="1"><tr><td>F-62d</td><td>D</td></tr><tr><td>U/DC</td><td></td></tr></table>	F-62d	D	U/DC		BCD8桁のアップ・ダウンカウンタ 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	×	○	○	○				
F-62d	D													
U/DC														
F-63	— <table border="1"><tr><td>F-63</td><td>D</td></tr><tr><td>INC</td><td></td></tr></table>	F-63	D	INC		バイナリ加算カウンタ(1バイト)	○	○	○	○				
F-63	D													
INC														
F-63w	— <table border="1"><tr><td>F-63w</td><td>D</td></tr><tr><td>INC</td><td></td></tr></table>	F-63w	D	INC		バイナリ加算カウンタ(1ワード)	○	○	○	○				
F-63w	D													
INC														
F-64	— <table border="1"><tr><td>F-64</td><td>D</td></tr><tr><td>DEC</td><td></td></tr></table>	F-64	D	DEC		バイナリ減算カウンタ(1バイト)	○	○	○	○				
F-64	D													
DEC														
F-64w	— <table border="1"><tr><td>F-64w</td><td>D</td></tr><tr><td>DEC</td><td></td></tr></table>	F-64w	D	DEC		バイナリ減算カウンタ(1ワード)	○	○	○	○				
F-64w	D													
DEC														
F-65	— <table border="1"><tr><td>F-65</td><td>D</td></tr><tr><td>BCDI</td><td></td></tr></table>	F-65	D	BCDI		BCD加算カウンタ(1バイト)	×	○	○	○				
F-65	D													
BCDI														
F-65w	— <table border="1"><tr><td>F-65w</td><td>D</td></tr><tr><td>BCDI</td><td></td></tr></table>	F-65w	D	BCDI		BCD加算カウンタ(1ワード)	×	○	○	○				
F-65w	D													
BCDI														
F-66	— <table border="1"><tr><td>F-66</td><td>D</td></tr><tr><td>BCDD</td><td></td></tr></table>	F-66	D	BCDD		BCD減算カウンタ(1バイト)	×	○	○	○				
F-66	D													
BCDD														
F-66w	— <table border="1"><tr><td>F-66w</td><td>D</td></tr><tr><td>BCDD</td><td></td></tr></table>	F-66w	D	BCDD		BCD減算カウンタ(1ワード)	×	○	○	○				
F-66w	D													
BCDD														
F-67	— <table border="1"><tr><td>F-67</td><td>n</td><td>D</td></tr><tr><td>NSFH</td><td></td><td></td></tr></table>	F-67	n	D	NSFH			桁シフト(上位シフト)	×	○	○	○		
F-67	n	D												
NSFH														
F-68	— <table border="1"><tr><td>F-68</td><td>n</td><td>D</td></tr><tr><td>NSFL</td><td></td><td></td></tr></table>	F-68	n	D	NSFL			桁シフト(下位シフト)	×	○	○	○		
F-68	n	D												
NSFL														
F-69	— <table border="1"><tr><td>F-69</td><td>S</td><td>D</td></tr><tr><td>NXFR</td><td></td><td></td></tr></table>	F-69	S	D	NXFR			桁転送(下位桁)	×	○	○	○		
F-69	S	D												
NXFR														
F-70	— <table border="1"><tr><td>F-70</td><td>n</td><td>S</td><td>D</td></tr><tr><td>FILE</td><td></td><td></td><td></td></tr></table>	F-70	n	S	D	FILE				nバイト一括転送	○	○	○	○
F-70	n	S	D											
FILE														
F-70w	— <table border="1"><tr><td>F-70w</td><td>n</td><td>S</td><td>D</td></tr><tr><td>FILE</td><td></td><td></td><td></td></tr></table>	F-70w	n	S	D	FILE				nワード一括転送	○	○	○	○
F-70w	n	S	D											
FILE														
F-71	— <table border="1"><tr><td>F-71</td><td>n</td><td>D1</td><td>D2</td></tr><tr><td>CONS</td><td></td><td></td><td></td></tr></table>	F-71	n	D1	D2	CONS				8進定数(1バイト)一括転送	○	○	○	○
F-71	n	D1	D2											
CONS														
F-71w	— <table border="1"><tr><td>F-71w</td><td>n</td><td>D1</td><td>D2</td></tr><tr><td>CONS</td><td></td><td></td><td></td></tr></table>	F-71w	n	D1	D2	CONS				8進定数(1ワード)一括転送	○	○	○	○
F-71w	n	D1	D2											
CONS														
F-72	— <table border="1"><tr><td>F-72</td><td>n</td><td>S</td><td>D</td></tr><tr><td>DMPX</td><td></td><td></td><td></td></tr></table>	F-72	n	S	D	DMPX				ファイル1のレジスタへのnバイト分配	×	×	○ <sup>※3</sup>	○
F-72	n	S	D											
DMPX														
F-72w	— <table border="1"><tr><td>F-72w</td><td>n</td><td>S</td><td>D</td></tr><tr><td>DMPX</td><td></td><td></td><td></td></tr></table>	F-72w	n	S	D	DMPX				ファイル1のレジスタへのnワード分配	×	×	○ <sup>※3</sup>	○
F-72w	n	S	D											
DMPX														
F-73	— <table border="1"><tr><td>F-73</td><td>n</td><td>S</td><td>D</td></tr><tr><td>MPX</td><td></td><td></td><td></td></tr></table>	F-73	n	S	D	MPX				ファイル1のレジスタからのnバイト抽出	×	×	○ <sup>※3</sup>	○
F-73	n	S	D											
MPX														
F-73w	— <table border="1"><tr><td>F-73w</td><td>n</td><td>S</td><td>D</td></tr><tr><td>MPX</td><td></td><td></td><td></td></tr></table>	F-73w	n	S	D	MPX				ファイル1のレジスタからのnワード抽出	×	×	○ <sup>※3</sup>	○
F-73w	n	S	D											
MPX														
F-74	— <table border="1"><tr><td>F-74</td><td>n</td><td>S</td><td>D</td></tr><tr><td>nXFR</td><td></td><td></td><td></td></tr></table>	F-74	n	S	D	nXFR				nバイト転送	○	○	○	○
F-74	n	S	D											
nXFR														
F-74w	— <table border="1"><tr><td>F-74w</td><td>n</td><td>S</td><td>D</td></tr><tr><td>nXFR</td><td></td><td></td><td></td></tr></table>	F-74w	n	S	D	nXFR				nワード転送	○	○	○	○
F-74w	n	S	D											
nXFR														
F-76	— <table border="1"><tr><td>F-76</td><td>S1</td><td>S2</td><td>D</td></tr><tr><td>FILR</td><td></td><td></td><td></td></tr></table>	F-76	S1	S2	D	FILR				nバイト(レジスタS1で指定)一括転送	×	×	○	○
F-76	S1	S2	D											
FILR														
F-76w	— <table border="1"><tr><td>F-76w</td><td>S1</td><td>S2</td><td>D</td></tr><tr><td>FILR</td><td></td><td></td><td></td></tr></table>	F-76w	S1	S2	D	FILR				nワード(レジスタS1で指定)一括転送	×	×	○	○
F-76w	S1	S2	D											
FILR														
F-77	— <table border="1"><tr><td>F-77</td><td>S1</td><td>S2</td><td>D</td></tr><tr><td>CHKC</td><td></td><td></td><td></td></tr></table>	F-77	S1	S2	D	CHKC				サムチェックコード生成	×	×	○	○
F-77	S1	S2	D											
CHKC														
F-78	— <table border="1"><tr><td>F-78</td><td>S1</td><td>S2</td><td>S3</td></tr><tr><td>CHK</td><td></td><td></td><td></td></tr></table>	F-78	S1	S2	S3	CHK				データのチェック	×	×	○	○
F-78	S1	S2	S3											
CHK														

※3 JW-32CUH/H1, JW-33CUH/H1/H2/H3でプログラムできます。JW-31CUH/H1ではプログラムできません。

命令語	シンボル	機能	P C機種名 (J-boardはJW20Hを含む)							
			JW10	JW20H	JW30H	JW50H/70H/100H				
F-79	<table border="1"><tr><td>F-79 SORT</td><td>S1</td><td>n1</td><td>n2</td></tr></table>	F-79 SORT	S1	n1	n2	1バイトデータの並べかえ	×	×	○ <sup>※3</sup>	○
F-79 SORT	S1	n1	n2							
F-79w	<table border="1"><tr><td>F-79w SORT</td><td>S1</td><td>n1</td><td>n2</td></tr></table>	F-79w SORT	S1	n1	n2	1ワードデータの並べかえ	×	×	○ <sup>※3</sup>	○
F-79w SORT	S1	n1	n2							
F-80	<table border="1"><tr><td>F-80 IORF</td><td>D</td></tr></table>	F-80 IORF	D	1/O リフレッシュ (1バイト)	○	×	×	×		
	F-80 IORF	D								
	<table border="1"><tr><td>F-80 IORF</td><td>R-S</td></tr></table>	F-80 IORF	R-S	1/O リフレッシュ (1ユニット)	×	○	○	×		
F-80 IORF	R-S									
<table border="1"><tr><td>F-80 IORF</td><td>R-S-B</td></tr></table>	F-80 IORF	R-S-B	1/O リフレッシュ (1バイト)	×	×	×	○			
F-80 IORF	R-S-B									
F-81	<table border="1"><tr><td>F-81 IORF</td><td>n</td><td>D</td></tr></table>	F-81 IORF	n	D	1/O リフレッシュ (1ビット)	○	×	×	×	
	F-81 IORF	n	D							
<table border="1"><tr><td>F-81 DTRF</td><td>R-S</td></tr></table>	F-81 DTRF	R-S	特殊1/Oデータリフレッシュ	×	×	×	○			
F-81 DTRF	R-S									
F-82	<table border="1"><tr><td>F-82 IORF</td><td>SW</td></tr></table>	F-82 IORF	SW	特殊1/Oデータリフレッシュ	×	○	○	×		
F-82 IORF	SW									
F-85	<table border="1"><tr><td>F-85 PRRD</td><td>n1</td><td>SW-n2</td><td>D</td></tr></table>	F-85 PRRD	n1	SW-n2	D	特殊1/Oからの読出	×	○	○	×
F-85 PRRD	n1	SW-n2	D							
F-86	<table border="1"><tr><td>F-86 PRWR</td><td>n1</td><td>D</td><td>SW-n2</td></tr></table>	F-86 PRWR	n1	D	SW-n2	特殊1/Oへの書込	×	○	○	×
F-86 PRWR	n1	D	SW-n2							
F-90	<table border="1"><tr><td>F-90 REM</td><td>n</td></tr></table>	F-90 REM	n	リマーク n = 0000~3777	○	○	○	○		
F-90 REM	n									
F-91	<table border="1"><tr><td>F-91 BCD8</td><td>n1</td><td>n2</td><td>D</td></tr></table>	F-91 BCD8	n1	n2	D	BCD定数(8桁)の転送	×	×	○	○
F-91 BCD8	n1	n2	D							
F-97	<table border="1"><tr><td>F-97 DML8</td><td>n1</td><td>n2</td><td>D</td></tr></table>	F-97 DML8	n1	n2	D	10進定数(8桁)の転送	×	×	○	○
F-97 DML8	n1	n2	D							
F-100	<table border="1"><tr><td>F-100 ADRS</td><td>S</td><td>D</td></tr></table>	F-100 ADRS	S	D	間接アドレスの設定	×	×	○	○	
F-100 ADRS	S	D								
F-101	<table border="1"><tr><td>F-101 SEGM</td><td>n</td><td>file N</td><td>D</td></tr></table>	F-101 SEGM	n	file N	D	間接アドレスの設定	×	×	○	○
F-101 SEGM	n	file N	D							
F-102	<table border="1"><tr><td>F-102 MRD</td><td>n</td><td>file N</td><td>D</td></tr></table>	F-102 MRD	n	file N	D	直接指定アドレスのレジスタからの読出 (1バイト)	×	×	○ <sup>※3</sup>	○
F-102 MRD	n	file N	D							
F-102w	<table border="1"><tr><td>F-102w MRD</td><td>n</td><td>file N</td><td>D</td></tr></table>	F-102w MRD	n	file N	D	直接指定アドレスのレジスタからの読出 (1ワード)	×	×	○ <sup>※3</sup>	○
F-102w MRD	n	file N	D							
F-103	<table border="1"><tr><td>F-103 MWR</td><td>S</td><td>n</td><td>file N</td></tr></table>	F-103 MWR	S	n	file N	直接指定アドレスのレジスタへの書込 (1バイト)	×	×	○ <sup>※3</sup>	○
F-103 MWR	S	n	file N							
F-103w	<table border="1"><tr><td>F-103w MWR</td><td>S</td><td>n</td><td>file N</td></tr></table>	F-103w MWR	S	n	file N	直接指定アドレスのレジスタへの書込 (1ワード)	×	×	○ <sup>※3</sup>	○
F-103w MWR	S	n	file N							
F-112	<table border="1"><tr><td>F-112 NQMP</td><td>S1</td><td>S2</td><td>S3</td></tr></table>	F-112 NQMP	S1	S2	S3	nバイト一括比較	×	×	○	○
F-112 NQMP	S1	S2	S3							
F-112w	<table border="1"><tr><td>F-112w NQMP</td><td>S1</td><td>S2</td><td>S3</td></tr></table>	F-112w NQMP	S1	S2	S3	nワード一括比較	×	×	○	○
F-112w NQMP	S1	S2	S3							
F-116	<table border="1"><tr><td>F-116 DIV</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-116 DIV	S1	S2	D	レジスタ(BCD8桁)とレジスタ(BCD8桁)の除算 (商は整数部8桁、小数部4桁)	×	○	○	○
F-116 DIV	S1	S2	D							
F-130	<table border="1"><tr><td>F-130 BIT →</td><td>S1</td><td>S2</td></tr></table>	F-130 BIT →	S1	S2	ビット抽出(間接指定)	×	○	○	○	
F-130 BIT →	S1	S2								
F-131	<table border="1"><tr><td>F-131 BIT →</td><td>n</td><td>S</td></tr></table>	F-131 BIT →	n	S	ビット抽出(直接指定)	×	○	○	○	
F-131 BIT →	n	S								
F-132	① <table border="1"><tr><td>F-132 S/R</td><td>S</td><td>D</td></tr></table> ②	F-132 S/R	S	D	ビットセット/リセット(間接指定) ① セット/リセット指示入力 ② 実行入力	×	○	○	○	
F-132 S/R	S	D								
F-133	① <table border="1"><tr><td>F-133 S/R</td><td>n</td><td>D</td></tr></table> ②	F-133 S/R	n	D	ビットセット/リセット(直接指定) ① セット/リセット指示入力 ② 実行入力	×	○	○	○	
F-133 S/R	n	D								
F-140	<table border="1"><tr><td>F-140 LABL</td><td>LBn</td></tr></table>	F-140 LABL	LBn	ラベルの設定	○	○	○	○		
F-140 LABL	LBn									
F-141	<table border="1"><tr><td>F-141 JMP</td><td>LBn</td></tr></table>	F-141 JMP	LBn	ラベルヘジャンプ	○	○	○	○		
F-141 JMP	LBn									
F-142	<table border="1"><tr><td>F-142 CALL</td><td>LBn</td></tr></table>	F-142 CALL	LBn	ラベルをサブルーチンコール	○	○	○	○		
F-142 CALL	LBn									
F-143	<table border="1"><tr><td>F-143 RET</td></tr></table>	F-143 RET	サブルーチンからのリターン	○	○	○	○			
F-143 RET										
F-144	<table border="1"><tr><td>F-144 FOR</td><td>n</td></tr></table>	F-144 FOR	n	ループ回数設定	○	○	○	○		
F-144 FOR	n									
F-145	<table border="1"><tr><td>F-145 NEXT</td></tr></table>	F-145 NEXT	ループの終了	○	○	○	○			
F-145 NEXT										

※3 JW-32CUH/H1, JW-33CUH/H1/H2/H3でプログラムできます。JW-31CUH/H1ではプログラムできません。

命令語	シンボル	機能	P C機種名 (J-boardはJW20Hに含む)			
			JW10	JW20H	JW30H	JW50H/70H/100H
F-146		ループ回数のレジスタ設定	×	×	○	○
F-147		ループの条件終了	×	×	○	○
F-148		レジスタ設定ラベルをサブルーチンコール	×	×	○	○
F-149		サブルーチンからの条件終了	×	×	○	○
F-151		レジスタ設定ラベルへジャンプ	×	×	○	○
F-153		BCD(8桁)→BIN(32ビット)変換	×	○	○	○
F-154		BIN(32ビット)→BCD(10桁)変換	×	○	○	○
F-155		時(BCD4桁), 分(BCD2桁), 秒(BCD2桁)→ 秒(BCD8桁)変換	×	×	○ <sup>※3</sup>	○
F-156		秒(BCD8桁)→時(BCD4桁), 分(BCD2桁), 秒(BCD2桁)変換	×	×	○ <sup>※3</sup>	○
F-160		両方向シフトレジスタ(nビット) ① シフト方向指示入力 ③ シフト入力 ② データ入力 ④ リセット入力 (シフトは1ビット) (S1)=0~256 (S2)=0~7	×	×	○	○
Fc160		両方向シフトレジスタ(nビット) ① シフト方向指示入力 ③ シフト入力 ② データ入力 ④ リセット入力 (シフトは1ビット) n1=0~377 n2=0~7	×	×	○	○
F-161		非同期シフトレジスタ(nバイト) ① シフト方向指示入力 ② シフト入力	×	×	○	○
F-161w		非同期シフトレジスタ(nワード) ① シフト方向指示入力 ② シフト入力	×	×	○	○
F-163		バイナリ加算(+2)カウンタ(1バイト)	×	○	○	○
F-163w		バイナリ加算(+2)カウンタ(1ワード)	×	○	○	○
F-164		バイナリ減算(-2)カウンタ(1バイト)	×	○	○	○
F-164w		バイナリ減算(-2)カウンタ(1ワード)	×	○	○	○
F-170		データ挿入(1バイト)	×	×	○ <sup>※3</sup>	○
F-170w		データ挿入(1ワード)	×	×	○ <sup>※3</sup>	○
F-171		データ削除(1バイト)	×	×	○ <sup>※3</sup>	○
F-171w		データ削除(1ワード)	×	×	○ <sup>※3</sup>	○
F-172		データ検索(1バイト)	×	×	○ <sup>※3</sup>	○
F-172w		データ検索(1ワード)	×	×	○ <sup>※3</sup>	○
F-173		データチェンジ(1バイト) ① モード指定入力 ② 実行入力	×	×	○ <sup>※3</sup>	○
F-173w		データチェンジ(1ワード) ① モード指定入力 ② 実行入力	×	×	○ <sup>※3</sup>	○
F-174		レジスタ間(1バイト)データ交換(nバイト)	×	×	○	○
F-175		上位4ビットと下位4ビットの交換(nバイト)	×	×	○	○
F-176		直接指定アドレスのレジスタからの読出 (256バイト)	×	×	○ <sup>※3</sup>	○
F-177		直接指定アドレスのレジスタへの書込 (256バイト)	×	×	○ <sup>※3</sup>	○

※3 JW-32CUH/H1, JW-33CUH/H1/H2/H3でプログラムできます。JW-31CUH/H1ではプログラムできません。

命令語	シンボル	機能	P C機種名 (J-boardはJW20Hを含む)			
			JW10	JW20H	JW30H	JW50H/70H/100H
F-180	$\overline{\text{F-180}} \begin{array}{ c c c c } \hline \text{CP>} & \text{S1} & \text{S2} & \text{BIT} \\ \hline \end{array}$	レジスタ間(1バイト)の比較(>,リレー出力付)	×	×	○	×
F-180w	$\overline{\text{F-180w}} \begin{array}{ c c c c } \hline \text{CP>} & \text{S1} & \text{S2} & \text{BIT} \\ \hline \end{array}$	レジスタ間(1ワード)の比較(>,リレー出力付)	×	×	○	×
Fc180	$\overline{\text{Fc180}} \begin{array}{ c c c c } \hline \text{CP>} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	レジスタと定数(1バイト)の比較(>,リレー出力付)	×	×	○	×
Fc180w	$\overline{\text{Fc180w}} \begin{array}{ c c c c } \hline \text{CP>} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	レジスタと定数(1ワード)の比較(>,リレー出力付)	×	×	○	×
F-181	$\overline{\text{F-181}} \begin{array}{ c c c c } \hline \text{CP<} & \text{S1} & \text{S2} & \text{BIT} \\ \hline \end{array}$	レジスタ間(1バイト)の比較(<,リレー出力付)	×	×	○	×
F-181w	$\overline{\text{F-181w}} \begin{array}{ c c c c } \hline \text{CP<} & \text{S1} & \text{S2} & \text{BIT} \\ \hline \end{array}$	レジスタ間(1ワード)の比較(<,リレー出力付)	×	×	○	×
Fc181	$\overline{\text{Fc181}} \begin{array}{ c c c c } \hline \text{CP<} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	レジスタと定数(1バイト)の比較(<,リレー出力付)	×	×	○	×
Fc181w	$\overline{\text{Fc181w}} \begin{array}{ c c c c } \hline \text{CP<} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	レジスタと定数(1ワード)の比較(<,リレー出力付)	×	×	○	×
F-182	$\overline{\text{F-182}} \begin{array}{ c c c c } \hline \text{CP=} & \text{S1} & \text{S2} & \text{BIT} \\ \hline \end{array}$	レジスタ間(1バイト)の比較(=,リレー出力付)	×	×	○	×
F-182w	$\overline{\text{F-182w}} \begin{array}{ c c c c } \hline \text{CP=} & \text{S1} & \text{S2} & \text{BIT} \\ \hline \end{array}$	レジスタ間(1ワード)の比較(=,リレー出力付)	×	×	○	×
Fc182	$\overline{\text{Fc182}} \begin{array}{ c c c c } \hline \text{CP=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	レジスタと定数(1バイト)の比較(=,リレー出力付)	×	×	○	×
Fc182w	$\overline{\text{Fc182w}} \begin{array}{ c c c c } \hline \text{CP=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	レジスタと定数(1ワード)の比較(=,リレー出力付)	×	×	○	×
F-183	$\overline{\text{F-183}} \begin{array}{ c c c c } \hline \text{CP>=} & \text{S1} & \text{S2} & \text{BIT} \\ \hline \end{array}$	レジスタ間(1バイト)の比較(≥,リレー出力付)	×	×	○	×
F-183w	$\overline{\text{F-183w}} \begin{array}{ c c c c } \hline \text{CP>=} & \text{S1} & \text{S2} & \text{BIT} \\ \hline \end{array}$	レジスタ間(1ワード)の比較(≥,リレー出力付)	×	×	○	×
Fc183	$\overline{\text{Fc183}} \begin{array}{ c c c c } \hline \text{CP>=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	レジスタと定数(1バイト)の比較(≥,リレー出力付)	×	×	○	×
Fc183w	$\overline{\text{Fc183w}} \begin{array}{ c c c c } \hline \text{CP>=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	レジスタと定数(1ワード)の比較(≥,リレー出力付)	×	×	○	×
F-184	$\overline{\text{F-184}} \begin{array}{ c c c c } \hline \text{CP<=} & \text{S1} & \text{S2} & \text{BIT} \\ \hline \end{array}$	レジスタ間(1バイト)の比較(≤,リレー出力付)	×	×	○	×
F-184w	$\overline{\text{F-184w}} \begin{array}{ c c c c } \hline \text{CP<=} & \text{S1} & \text{S2} & \text{BIT} \\ \hline \end{array}$	レジスタ間(1ワード)の比較(≤,リレー出力付)	×	×	○	×
Fc184	$\overline{\text{Fc184}} \begin{array}{ c c c c } \hline \text{CP<=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	レジスタと定数(1バイト)の比較(≤,リレー出力付)	×	×	○	×
Fc184w	$\overline{\text{Fc184w}} \begin{array}{ c c c c } \hline \text{CP<=} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	レジスタと定数(1ワード)の比較(≤,リレー出力付)	×	×	○	×
F-185	$\overline{\text{F-185}} \begin{array}{ c c c c } \hline \text{CP<>} & \text{S1} & \text{S2} & \text{BIT} \\ \hline \end{array}$	レジスタ間(1バイト)の比較(>,リレー出力付)	×	×	○	×
F-185w	$\overline{\text{F-185w}} \begin{array}{ c c c c } \hline \text{CP<>} & \text{S1} & \text{S2} & \text{BIT} \\ \hline \end{array}$	レジスタ間(1ワード)の比較(>,リレー出力付)	×	×	○	×
Fc185	$\overline{\text{Fc185}} \begin{array}{ c c c c } \hline \text{CP<>} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	レジスタと定数(1バイト)の比較(>,リレー出力付)	×	×	○	×
Fc185w	$\overline{\text{Fc185w}} \begin{array}{ c c c c } \hline \text{CP<>} & \text{S} & \text{n} & \text{BIT} \\ \hline \end{array}$	レジスタと定数(1ワード)の比較(>,リレー出力付)	×	×	○	×
F-200	$\overline{\text{F-200}} \begin{array}{ c c c c } \hline \text{→POR} & \text{TASK n} & \text{@ S} & \text{PORT n} \\ \hline \end{array}$	ポートへの書込	×	×	×	○
F-201	$\overline{\text{F-201}} \begin{array}{ c c c c } \hline \text{POR→} & \text{TASK n} & \text{PORT n} & \text{@ D} \\ \hline \end{array}$	ポートからの読出	×	×	×	○
F-202	$\overline{\text{F-202}} \begin{array}{ c c c c } \hline \text{OPCH} & \text{UN-CH-ST} & \text{file N} & \text{n} \\ \hline \end{array}$	オープンチャンネル(局番8進定数設定)	×	×	○	×
	$\overline{\text{F-202}} \begin{array}{ c c c c } \hline \text{OPCH} & \text{PORT-CH-ST} & \text{file N} & \text{n} \\ \hline \end{array}$	オープンチャンネル(局番8進定数設定)	×	×	×	○
F-203	$\overline{\text{F-203}} \begin{array}{ c c c c } \hline \text{OPCH} & \text{UN-CH-ST} & \text{file N} & \text{n} \\ \hline \end{array}$	オープンチャンネル(局番16進定数設定)	×	×	○	×
	$\overline{\text{F-203}} \begin{array}{ c c c c } \hline \text{OPCH} & \text{PORT-CH-ST} & \text{file N} & \text{n} \\ \hline \end{array}$	オープンチャンネル(局番16進定数設定)	×	×	×	○

命令語	シンボル	機能	P C機種名 (J-boardはJW20Hに含む)			
			JW10	JW20H	JW30H	JW50H/70H/100H
F-204	F-204 SEND n S	送信命令	×	×	○	○
F-205	F-205 RCV n D	受信命令	×	×	○	○
F-206	F-206 EOP1 UN1-CH ST1 UN2	オープンチャンネル1(階層通信設定)	×	×	○	×
F-207	F-207 EOP2 ST2 file N n	オープンチャンネル2(階層通信設定)	×	×	○	×
F-210	F-210 ADD S1 S2 D	レジスタ間のバイナリ加算 (8ビット+8ビット)	○	○	○	○
F-210w	F-210w ADD S1 S2 D	レジスタ間のバイナリ加算 (16ビット+16ビット)	○	○	○	○
F-210d	F-210d ADD S1 S2 D	レジスタ間のバイナリ加算 (32ビット+32ビット)	×	○	○	○
Fc210	Fc210 ADD S1 n D	レジスタと定数のバイナリ加算 (8ビット+8ビット)	○	○	○	○
Fc210w	Fc210w ADD S1 n D	レジスタと定数のバイナリ加算 (16ビット+16ビット)	○	○	○	○
Fc210d	Fc210d ADD S1 n D	レジスタと定数のバイナリ加算 (32ビット+16ビット)	×	○	○	○
F-211	F-211 SUB S1 S2 D	レジスタ間のバイナリ減算 (8ビット-8ビット)	○	○	○	○
F-211w	F-211w SUB S1 S2 D	レジスタ間のバイナリ減算 (16ビット-16ビット)	○	○	○	○
F-211d	F-211d SUB S1 S2 D	レジスタ間のバイナリ減算 (32ビット-32ビット)	×	○	○	○
Fc211	Fc211 SUB S1 n D	レジスタと定数のバイナリ減算 (8ビット-8ビット)	○	○	○	○
Fc211w	Fc211w SUB S1 n D	レジスタと定数のバイナリ減算 (16ビット-16ビット)	○	○	○	○
Fc211d	Fc211d SUB S1 n D	レジスタと定数のバイナリ減算 (32ビット-16ビット)	×	○	○	○
F-212	F-212 WNDW S1 S2 S3	ウィンドウコンパレータ (1バイトレジスタ間)	○	○	○	○
F-212w	F-212w WNDW S1 S2 S3	ウィンドウコンパレータ (1ワードレジスタ間)	○	○	○	○
F-212d	F-212d WNDW S1 S2 S3	ウィンドウコンパレータ (2ワードレジスタ間)	×	○	○	○
Fc212	Fc212 WNDW S1 n1 n2	ウィンドウコンパレータ (1バイト8進定数間)	○	○	○	○
Fc212w	Fc212w WNDW S1 n1 n2	ウィンドウコンパレータ (1ワード8進定数間)	○	○	○	○
Fx212	Fx212 WNDW S1 n1 n2	ウィンドウコンパレータ (1バイト16進定数間)	×	×	○	○
Fx212w	Fx212w WNDW S1 n1 n2	ウィンドウコンパレータ (1ワード16進定数間)	×	×	○	○
F-215	F-215 MUL S1 S2 D	レジスタ間のバイナリ乗算 (8ビット×8ビット)	○	○	○	○
F-215w	F-215w MUL S1 S2 D	レジスタ間のバイナリ乗算 (16ビット×16ビット)	○	○	○	○
F-215d	F-215d MUL S1 S2 D	レジスタ間のバイナリ乗算 (32ビット×32ビット)	×	○	○	○
Fc215	Fc215 MUL S1 n D	レジスタと定数のバイナリ乗算 (8ビット×8ビット)	○	○	○	○
Fc215w	Fc215w MUL S1 n D	レジスタと定数のバイナリ乗算 (16ビット×16ビット)	○	○	○	○
Fc215d	Fc215d MUL S1 n D	レジスタと定数のバイナリ乗算 (32ビット×16ビット)	×	○	○	○

命令語	シンボル	機能	P C機種名 (J-boardはJW20Hを含む)							
			JW10	JW20H	JW30H	JW50H/70H/100H				
F-216	<table border="1"><tr><td>F-216 DIV</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-216 DIV	S1	S2	D	レジスタ間のバイナリ除算 (8ビット÷8ビット)	○	○	○	○
F-216 DIV	S1	S2	D							
F-216w	<table border="1"><tr><td>F-216w DIV</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-216w DIV	S1	S2	D	レジスタ間のバイナリ除算 (15ビット÷15ビット)	○	○	○	○
F-216w DIV	S1	S2	D							
F-216d	<table border="1"><tr><td>F-216d DIV</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-216d DIV	S1	S2	D	レジスタ間のバイナリ除算 (31ビット÷31ビット)	○	○	○	○
F-216d DIV	S1	S2	D							
Fc216	<table border="1"><tr><td>Fc216 DIV</td><td>S1</td><td>n</td><td>D</td></tr></table>	Fc216 DIV	S1	n	D	レジスタと定数のバイナリ除算 (8ビット÷8ビット)	○	○	○	○
Fc216 DIV	S1	n	D							
Fc216w	<table border="1"><tr><td>Fc216w DIV</td><td>S1</td><td>n</td><td>D</td></tr></table>	Fc216w DIV	S1	n	D	レジスタと定数のバイナリ除算 (15ビット÷15ビット)	○	○	○	○
Fc216w DIV	S1	n	D							
Fc216d	<table border="1"><tr><td>Fc216d DIV</td><td>S1</td><td>n</td><td>D</td></tr></table>	Fc216d DIV	S1	n	D	レジスタと定数のバイナリ除算 (31ビット÷15ビット)	×	○	○	○
Fc216d DIV	S1	n	D							
F-231	<table border="1"><tr><td>F-231 MCRN</td></tr></table>	F-231 MCRN	マスターコントロール ネスティング リセット	×	×	○	×			
F-231 MCRN										
F-242	<table border="1"><tr><td>F-242 JCRN</td></tr></table>	F-242 JCRN	ジャンプコントロール ネスティング リセット	×	×	○	×			
F-242 JCRN										
F-252	<table border="1"><tr><td>F-252 →ASC</td><td>S</td><td>n</td><td>D</td></tr></table>	F-252 →ASC	S	n	D	HEX→ASCII変換	×	○	○	○
F-252 →ASC	S	n	D							
F-253	<table border="1"><tr><td>F-253 →HEX</td><td>S</td><td>n</td><td>D</td></tr></table>	F-253 →HEX	S	n	D	ASCII→HEX変換	×	○	○	○
F-253 →HEX	S	n	D							
F-260	<table border="1"><tr><td>F-260 RTMR</td><td>S</td><td>D</td><td>BIT</td></tr></table>	F-260 RTMR	S	D	BIT	現在値と設定値が、レジスタ指定可能な タイマ命令	×	×	○	○
F-260 RTMR	S	D	BIT							
Fc260	<table border="1"><tr><td>Fc260 RTMR</td><td>n</td><td>D</td><td>BIT</td></tr></table>	Fc260 RTMR	n	D	BIT	現在値がレジスタ指定可能なタイマ命令 (設定値はBCD定数)	×	×	○	○
Fc260 RTMR	n	D	BIT							
F-261	<table border="1"><tr><td>① F-261 ② RCNT</td><td>S</td><td>D</td><td>BIT</td></tr></table>	① F-261 ② RCNT	S	D	BIT	現在値と設定値が、レジスタ指定可能な カウンタ命令 ① 計数入力 ② リセット入力	×	×	○	○
① F-261 ② RCNT	S	D	BIT							
Fc261	<table border="1"><tr><td>① Fc261 ② RCNT</td><td>n</td><td>D</td><td>BIT</td></tr></table>	① Fc261 ② RCNT	n	D	BIT	現在値がレジスタ指定可能なカウンタ命令 (設定値はBCD定数)① 計数入力 ② リセット入力	×	×	○	○
① Fc261 ② RCNT	n	D	BIT							
F-263	<table border="1"><tr><td>F-263 INC4</td><td>D</td></tr></table>	F-263 INC4	D	バイナリ加算(+4)カウンタ(1バイト)	×	×	○	○		
F-263 INC4	D									
F-263w	<table border="1"><tr><td>F-263w INC4</td><td>D</td></tr></table>	F-263w INC4	D	バイナリ加算(+4)カウンタ(1ワード)	×	×	○	○		
F-263w INC4	D									
F-264	<table border="1"><tr><td>F-264 DEC4</td><td>D</td></tr></table>	F-264 DEC4	D	バイナリ減算(-4)カウンタ(1バイト)	×	×	○	○		
F-264 DEC4	D									
F-264w	<table border="1"><tr><td>F-264w DEC4</td><td>D</td></tr></table>	F-264w DEC4	D	バイナリ減算(-4)カウンタ(1ワード)	×	×	○	○		
F-264w DEC4	D									
F-310	<table border="1"><tr><td>F-310 SADD</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-310 SADD	S1	S2	D	レジスタ間の符号付バイナリ加算 (31ビット+31ビット)	×	×	○	×
F-310 SADD	S1	S2	D							
F-311	<table border="1"><tr><td>F-311 SSUB</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-311 SSUB	S1	S2	D	レジスタ間の符号付バイナリ減算 (31ビット-31ビット)	×	×	○	×
F-311 SSUB	S1	S2	D							
F-315	<table border="1"><tr><td>F-315 SMUL</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-315 SMUL	S1	S2	D	レジスタ間の符号付バイナリ乗算 (31ビット×31ビット)	×	×	○	×
F-315 SMUL	S1	S2	D							
F-316	<table border="1"><tr><td>F-316 SDIV</td><td>S1</td><td>S2</td><td>D</td></tr></table>	F-316 SDIV	S1	S2	D	レジスタ間の符号付バイナリ除算 (31ビット÷31ビット)	×	×	○	×
F-316 SDIV	S1	S2	D							
NOP		無効命令	○	○	○	○				

(注) JW20Hのステップフロー命令(F-380~F-396)は省略しています。



## 改訂履歴

版、作成年月は表紙の右上に記載しております。

版	作成年月	改訂内容
初版	1996年11月	—————
改訂1.1版	1997年3月	・ JW10基本ユニット (JW-1342K/1442K/1642K) の追加 ・ JW30Hコントロールユニット (JW-31CUH1, JW-32CUH1, JW-33CUH1/H2/H3) の追加 ・ 説明改善 (2・73ページ)
改訂1.2版	1997年9月	・ 増刷改訂
改訂1.3版	2001年4月	・ 増刷改訂

# シャープマニファクチャリングシステム株式会社

本 社 〒581-8581 大阪府八尾市跡部本町4丁目1番33号

● インターネットホームページによるシャープ制御機器の情報サービス  
<http://www.sharp.co.jp/sms/>