

**HITACHI**

ソフトウェアマニュアル

プログラミング

**HI-FLOW For Windows<sup>®</sup>**

---

**SIOV**

Programmable Controller

SVJ-3-122 (G)

ソフトウェアマニュアル

プログラミング

**HI-FLOW For Windows<sup>®</sup>**

---

**SIOV**  
Programmable Controller

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

2002年10月	(第1版)	SVJ-3-122	(A)	(廃版)
2004年 3月	(第2版)	SVJ-3-122	(B)	(廃版)
2005年 3月	(第3版)	SVJ-3-122	(C)	(廃版)
2006年 6月	(第4版)	SVJ-3-122	(D)	(廃版)
2006年11月	(第5版)	SVJ-3-122	(D)	(廃版)
2008年11月	(第6版)	SVJ-3-122	(E)	(廃版)
2012年 8月	(第7版)	SVJ-3-122	(F)	(廃版)
2017年 7月	(第8版)	SVJ-3-122	(G)	

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

## 安全上のご注意

- システムの構築やプログラムの作成などは、このマニュアルの記載内容をよく読み、書かれている指示や注意を十分理解してから行ってください。誤操作により、システムが故障することがあります。
- このマニュアルは、必要なときすぐに参照できるよう、手近なところに保管してください。
- このマニュアルの記載内容について疑問点または不明点がございましたら、最寄りの弊社営業またはSEまでお知らせください。
- お客様の誤操作に起因する事故発生や損害については、弊社は責任を負いかねますのでご了承ください。
- 弊社提供ソフトウェアを改変して使用した場合に発生した事故や損害については、弊社は責任を負いかねますのでご了承ください。
- 弊社提供以外のソフトウェアを使用した場合の信頼性については、弊社は責任を負いかねますのでご了承ください。
- ファイルのバックアップ作業を日常業務に組み入れてください。ファイル装置の障害、ファイルアクセス中の停電、誤操作、その他何らかの原因によりファイルの内容を消失することがあります。このような事態に備え、計画的にファイルのバックアップを取っておいてください。
- 弊社製品が故障や誤動作したりプログラムに欠陥があった場合でも、使用されるシステムの安全が十分に確保されるよう、保護・安全回路は外部に設け、人身事故や重大な災害に対する安全対策が十分確保できるようなシステム設計としてください。
- 非常停止回路、インターロック回路などはPLCの外部で構成してください。PLCの故障により、機械の破損や事故の恐れがあります。
- 運転中のプログラム変更、強制出力、RUN、STOPなどは十分安全を確認してから行ってください。誤操作により、機械の破損や事故の恐れがあります。
- このマニュアルでは、安全上の注意事項のランクを潜在危険の重大度によって、「危険」、「警告」、「注意」、「通知」と区分しています。

### 警告表示の定義



：この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こす危険の存在を示す。



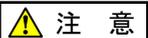
：この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こすおそれのある危険の存在を示す。



：この表示を無視して誤った取り扱いをすると、軽度の傷害または中程度の傷害を引き起こすおそれのある危険の存在を示す。



：この表示を無視して誤った取り扱いをすると、人身傷害とは関係のない損害を引き起こすおそれのある危険の存在を示す。

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

「重大な傷害」、「軽度の傷害または中程度の傷害」、「人身傷害とは関係のない損害」について、具体的な内容を以下に示します。

**重大な傷害**

失明、けが、やけど（高温、低温）、感電傷害、骨折、中毒などで、後遺症が残るものおよび治療のために入院、長期の通院を要するもの

**軽度の傷害または中程度の傷害**

治療のために入院や長期の通院を必要としないけが、やけど、感電傷害など

**人身傷害とは関係のない損害**

周囲の財物の損傷、弊社製品の故障や破損、データの損失など、人身傷害以外の損害

安全上の注意事項は、安全性を確保するための原則に基づいた、弊社製品における各種対策を補完する重要なものです。弊社製品やマニュアルに表示されている安全上の注意事項は、十分に検討されたものですが、それでも、予測を超えた事態が起こることが考えられます。操作するときは指示に従うだけでなく、常に自分自身でも注意するようにしてください。また、弊社製品の安全な運転および保守のために、各種規格、基準に従って安全施策を確立してください。

1.  このマニュアル内の警告表示

このマニュアルの中に書かれている警告とその記載箇所を、以下にまとめて示します。

1. 1 「通知」と表示されているもの

(6章、6-11ページ)

### 通 知

- NX/ToolsからNX/HOSTシステムファイルをPCsに送信した場合、NX/Tools以外で登録されたタスクは削除されます。そのため、NX/HOSTシステムファイルを送信した場合は、HI-FLOWプログラムを送信し、通信タスクを再送信してください。
- CPMSデバッガからタスク環境初期化コマンドを実行した場合、PCsに登録されているタスクがすべて削除されます。そのため、タスク環境初期化コマンドを実行した場合は、HI-FLOWプログラムを送信し、通信タスクを再送信してください。
- 通信タスクは以下の内容で送信されますので、CPMSデバッガで使用しないでください。  
タスク番号：206 ~ 208  
タスク格納エリア：/300E0000 ~ /300FFFFFF (\$TASK)  
/50800000 ~ /509FFFFFF (\$GLBRW)
- RPDPを使用している場合、モーション機能を使用できません。モーション機能を使用する場合は、HI-FLOWプログラム送信時に表示される以下のメッセージで  ボタンをクリックして、タスク環境を初期化してください。なお、タスク環境を初期化すると、PCsに登録されているRPDPのタスクはすべて削除されます。



このページは白紙です。

このマニュアルは、以下のプログラムプロダクトの説明をしたものです。

<プログラムプロダクト>

S-7895-03 「S10V HI-FLOWシステム」 (02-11)

## 来歴一覧表

改訂No.	来歴（改訂内容および改訂理由）	発行年月	備考
D	イーサネット通信にOPTETモジュール追加記載	2006.6	「S10V HI-FLOWシステム」（01-07）以降
D	モーション制御命令サポート	2006.6	「S10V HI-FLOWシステム」（02-01）以降
D	MP2300Hシステム構成変更手順を記載	2006.6	「S10V HI-FLOWシステム」（02-02）以降
D	誤記訂正	2006.11	
E	非同期プロセスエンド命令サポート	2008.11	「S10V HI-FLOWシステム」（02-04）以降
	モーション通信タスク自動送信機能サポート		「S10V HI-FLOWシステム」（02-05）以降
F	サポートOSにWindows® 7（32bit）を追加 警告表示全面見直し	2012.8	
G	サポートOSにWindows® 10（32bit）を追加	2017.7	「S10V HI-FLOWシステム」（02-11）以降

上記追加変更の他に、記述不明瞭な部分、単なる誤字・脱字などについては、お断りなく訂正しました。

# はじめに

日立プログラマブルコントローラ用の新しいプログラミング言語HI-FLOWは、フロー図形式の誰にでも簡単にプログラムできる言語です。

このマニュアルはHI-FLOWでプログラムを作るための命令語について説明しています。

なお、ラダープログラムについては、下記マニュアルを参照してください。

モーション制御命令はWindows® 7 (32bit) 、Windows® 10 (32bit) 非サポートです。

## <関連マニュアル>

- ・ソフトウェアマニュアル プログラミング S10V ラダー図 For Windows®  
(マニュアル番号 SVJ-3-121)
- ・ユーザーズマニュアル 基本モジュール (マニュアル番号 SVJ-1-100)

## <商標について>

- ・Microsoft®, Windows®は、米国Microsoft Corporationの米国およびその他の国における登録商標または商標です。
- ・Ethernet®は米国Xerox Corp.の登録商標です。

## <記憶容量の計算値についての注意>

- 2<sup>n</sup>計算値の場合 (メモリ容量・所要量、ファイル容量・所要量など)
  - 1KB (キロバイト) =1,024バイトの計算値です。
  - 1MB (メガバイト) =1,048,576バイトの計算値です。
  - 1GB (ギガバイト) =1,073,741,824バイトの計算値です。
- 10<sup>n</sup>計算値の場合 (ディスク容量など)
  - 1KB (キロバイト) =1,000バイトの計算値です。
  - 1MB (メガバイト) =1,000<sup>2</sup>バイトの計算値です。
  - 1GB (ギガバイト) =1,000<sup>3</sup>バイトの計算値です。

# 目 次

1	HI-FLOWプログラムの構成 .....	1-1
2	このマニュアルの使い方 .....	2-1
2.1	概 説 .....	2-2
2.2	構文説明 .....	2-3
2.3	応用命令説明 .....	2-4
2.4	モーション制御命令説明 .....	2-7
3	プロセス .....	3-1
3.1	プロセスとは? .....	3-2
3.2	プログラム .....	3-7
3.3	プロセス情報 .....	3-22
4	構文説明 .....	4-1
4.1	プロセススタート、プロセスエンド .....	4-2
4.2	ルートスタート、ルートエンド .....	4-6
4.3	ウェイト .....	4-7
4.4	ボックス .....	4-9
4.5	コントロールボックス .....	4-14
4.6	リピートスタート、リピートエンド .....	4-18
4.7	イフ .....	4-19
4.8	ジャンプ .....	4-21
4.9	エスケープ .....	4-22
4.10	パラスタート、パラエンド .....	4-23
4.11	セレクト、セルウェイト、セレクトエンド .....	4-24
4.12	マルチエントリー .....	4-25
4.13	コール .....	4-26
4.14	ファンクション .....	4-27
4.15	前条件付きウェイト .....	4-27
4.16	モーション .....	4-27
4.17	非同期プロセスエンド .....	4-28

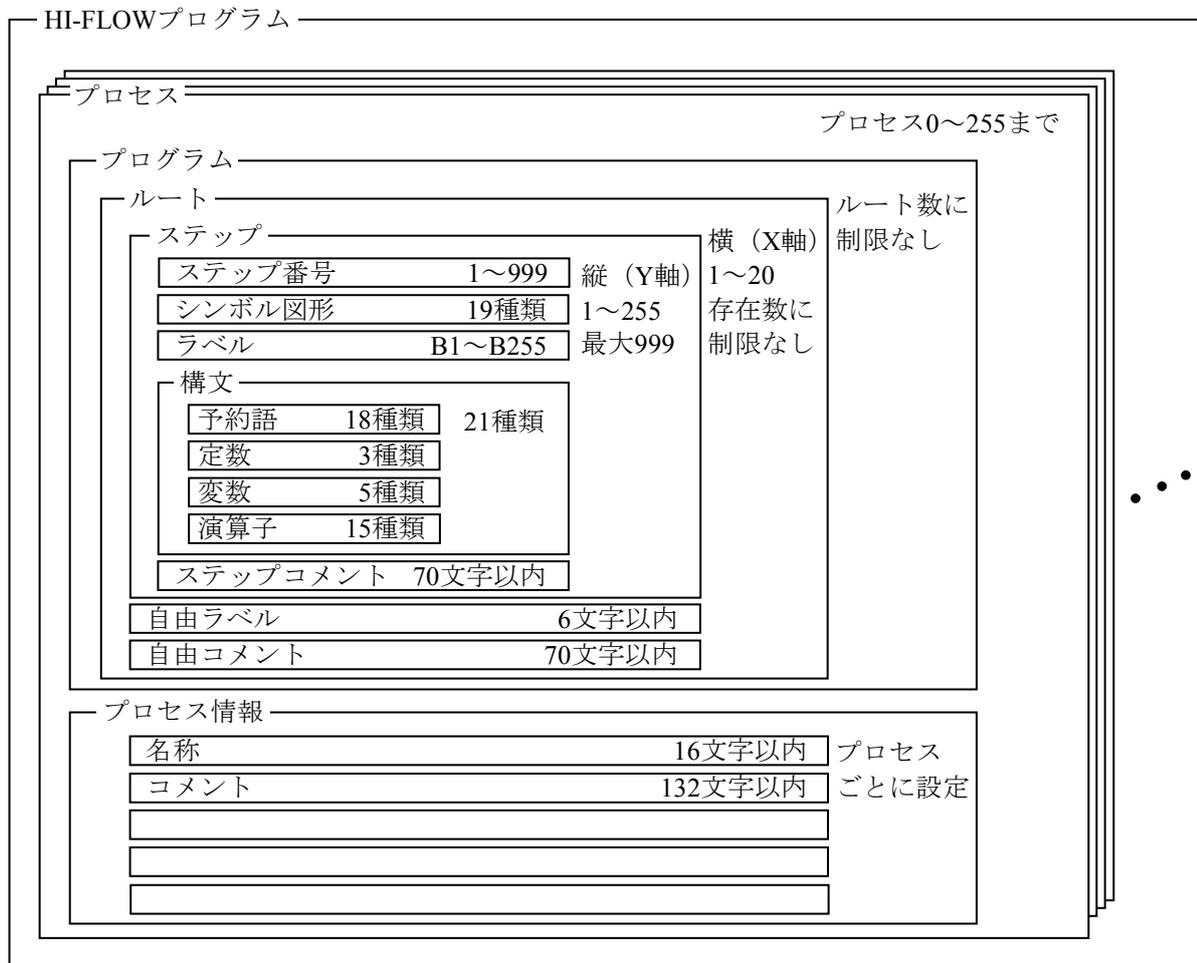
5	応用命令	5-1
5.1	概説	5-2
5.2	使用方法	5-2
5.3	パラメーター	5-2
5.4	演算時の型変換	5-4
5.5	システムエラーフラグ	5-5
5.6	機能説明	5-6
5.7	イーサネット通信応用命令	5-57
5.7.1	機能概要	5-57
5.7.2	使用方法	5-60
5.7.3	機能説明	5-71
5.7.4	サンプルプログラム	5-82
6	モーション制御命令	6-1
6.1	用途	6-2
6.2	仕様	6-3
6.2.1	システム構成	6-3
6.2.2	S10VとMP2300H間通信インタフェース	6-3
6.3	使用方法	6-4
6.4	モーション状態フラグ	6-12
6.5	機能説明	6-22
6.6	サンプルプログラム	6-72
補足	補足	Z-1
補足1	HI-FLOWプログラムの流れ	Z-2
補足2	PCsメモリー	Z-3
補足3	オンラインモード	Z-4
補足4	進行チェック	Z-8
補足5	HI-FLOWプログラムとCPU負荷	Z-10
補足6	MP2300Hシステム構成変更手順	Z-13

このページは白紙です。

# 1 HI-FLOW プログラムの構成

# 1 HI-FLOWプログラムの構成

このマニュアルは、新しいHI-FLOW言語の規格とその内容を説明しています。  
 実際にプログラムを考えると、必要に応じて参照してください。  
 ユーザーが作成するHI-FLOWプログラムは、次の要素から成り立っています。



## 2 このマニュアルの使い方

## 2 このマニュアルの使い方

### 2.1 概 説

このマニュアルは第1章で表した構成に従って作られています。

各項目に対応した章とページを示します。

項目	対応した章	ページ
プロセス	第3章	3-2
プログラム	3.2節	3-7
・ルート		3-7
・ステップ		3-12
・ステップ番号		3-13
・シンボル図形		3-13
・ラベル		3-17
・構文		3-17
・予約語		3-18
・定数		3-18
・変数		3-18
・演算子		3-20
・ステップコメント		3-20
・自由ラベル		3-21
・自由コメント		3-21
プロセス情報		3.3節
・名称	3-22	
・コメント	3-22	

## 2.2 構文説明

このマニュアルは、概要の次に構文の詳細を機能ごとに説明しています。  
各機能に対応した章とページを示します。

項目	図形	対応した章	ページ
構文説明		第4章	4-2
プロセススタート、 プロセスエンド	● ●	4. 1 節	4-2
・STP			4-3
・RST			4-4
・CLR			4-4
・ACT			4-4
ルートスタート、 ルートエンド	⊥ ⊥	4. 2 節	4-6
ウェイト		4. 3 節	4-7
・条件式	+		4-7
・タイマー			4-7
・出力ビット			4-7
・ウェイトタイマー			4-7
ボックス		4. 4 節	4-9
・代入式			4-9
・ON文			4-10
・OFF文			4-11
・パラレルタイマー			4-11
・TUP			4-12
・TRS			4-13
コントロールボックス		4. 5 節	4-14
・ACT			4-14
・RST			4-15
・STP			4-15
・CLR			4-16
リピートスタート、 リピートエンド	⊥ ⊥	4. 6 節	4-18
イフ	◇	4. 7 節	4-19
ジャンプ	↳	4. 8 節	4-21
エスケープ	✕	4. 9 節	4-22
パラスタート、 パラエンド	⊥ ⊥	4. 10 節	4-23
セレクト、 セルウェイト、 セレクトエンド	⊥ ≡≡≡	4. 11 節	4-24
マルチエントリー	⊥	4. 12 節	4-25
コール	⊥	4. 13 節	4-26
ファンクション	○	4. 14 節	4-27
前条件付きウェイト	+*	4. 15 節	4-27
モーション	⊙	4. 16 節	4-27
非同期プロセスエンド	↓	4. 17 節	4-28

## 2 このマニュアルの使い方

### 2.3 応用命令説明

HI-FLOWではラダー図と同様機能の応用命令をサポートしています。

応用命令の各機能に対応した項目の一覧を示します。

大別	種類	シンボル	機能概要	ページ
算術演算命令	加算	ADD	$S+D \rightarrow R$	5-7
	減算	SUB	$S-D \rightarrow R$	5-8
	+1	INC	$S+1 \rightarrow S$	5-9
	-1	DEC	$S-1 \rightarrow S$	5-10
	乗算	MUL	$S*D \rightarrow R$	5-11
	除算	DIV	$S/D \rightarrow R$	5-12
	剰余	MOD	S/Dの余り $\rightarrow R$	5-13
スケール変換	SCL	$S*D1/D2 \rightarrow R$	5-14	
論理演算命令	論理積	AND	$S \text{ AND } D \rightarrow R$	5-15
	論理和	OR	$S \text{ OR } D \rightarrow R$	5-16
	排他的論理和	EOR	$S \text{ EOR } D \rightarrow R$	5-17
	否定	NOT	$\text{NOT } S \rightarrow R$	5-18
比較演算命令	=	EQU	$S=D$ の真偽 $\rightarrow R$	5-19
	<>	NEQ	$S \neq D$ の真偽 $\rightarrow R$	5-20
	>	GT	$S>D$ の真偽 $\rightarrow R$	5-21
	>=	GE	$S \geq D$ の真偽 $\rightarrow R$	5-22
	<	LT	$S<D$ の真偽 $\rightarrow R$	5-23
	<=	LE	$S \leq D$ の真偽 $\rightarrow R$	5-24
	テスト	TST	Sの符号 $\rightarrow R$	5-25
データ転送命令	転送	MOV	$S \rightarrow D$	5-26
	一括転送	MOM	$S \sim S_n \rightarrow D \sim D_n$	5-27
	交換	EXC	$S \leftrightarrow D$	5-28
	FIFO書き込み	PSH	$S \rightarrow D$ (FIFOテーブル)	5-29
	FIFO読み出し	POP	$S$ (FIFOテーブル) $\rightarrow D$	5-30
	アドレスセット	AST	Sアドレス $\rightarrow D$	5-31
	サーチ	SCH	$S = D (n) \rightarrow n$ をRにセット	5-32
データ変換命令	BIN-BCD	BTD	$\text{BIN} \rightarrow \text{BCD}$ $S \text{ -----} \rightarrow R$	5-33
	BCD-BIN	DTB	$\text{BCD} \rightarrow \text{BIN}$ $S \text{ -----} \rightarrow R$	5-34
	BIN-7SEG	SEG	$\text{BIN} \rightarrow 7\text{セグメント}$ $S \text{ -----} \rightarrow R$	5-35

大別	種類	シンボル	機能概要	ページ
データ変換命令	BIN-ASC	ASP	BIN → ASCII (パック、アンパック)	5-36
		ASU	S ----→ (R, R+1) 、 (R, R+1, R+2, R+3)	5-37
	ASC-BIN	APB	ASCII (パック、アンパック) → BIN	5-38
		AUB	(S, S+1) 、 (S, S+1, S+2, S+3) ----→ R	5-39
	絶対値	ABS	S  → R	5-40
	+/-	NEG	-S → R	5-41
	デコード	DCD	S の $2^{11} \sim 2^{15}$ → R の $2^n$ ビットON	5-42
エンコード	ECD	S の最初のONビット番号 → R の $2^{11} \sim 2^{15}$	5-43	
シフト命令	論理右シフト	LSR	S 論理右シフト D → R	5-44
	論理左シフト	LSL	S 論理左シフト D → R	5-45
	算術右シフト	ASR	S 算術右シフト D → R	5-46
	算術左シフト	ASL	S 算術左シフト D → R	5-47
命回令転	右回転	ROR	S 右回転 R	5-48
	左回転	ROL	S 左回転 R	5-49
関数処理命令	リミッタ	LIM		5-50
	デッドバンド	BND		5-51
	デッドゾーン	ZON		5-52
	平方根	ROT		5-53
	最大値	MAX		5-54
	最小値	MIN		5-55
特殊命令	クリアー	XCLR YCLR GCLR RCLR KCLR TCLR UCLR CCLR VCLR ECLR FCLR JCLR QCLR HHCLR		5-56

## 2 このマニュアルの使い方

---

大別	種類	シンボル	機能概要	ページ
イーサネット通信命令	TCP通信	TOP	TCP接続のオープン	5-71
		TPOP	TCP接続のオープン	5-72
		TCLO	TCP接続のクローズ	5-73
		TRCV	TCP受信	5-74
		TSND	TCP送信	5-76
	UDP通信	UOP	UDPのオープン	5-77
		UCLO	UDPのクローズ	5-78
		URCV	UDP受信	5-79
		USND	UDP送信	5-81

## 2.4 モーション制御命令説明

HI-FLOWからモーション制御したい場合は、モーション制御命令を使用すると簡単にプログラムできます。  
モーション制御命令の各機能に対応した項目の一覧を示します。

モーション制御命令はWindows® 7 (32bit) 、Windows® 10 (32bit) 非サポートです。

大別	種類	シンボル	機能概要	ページ
モーション制御命令	サーボON	SVON	サーボ始動	6-24
	サーボOFF	SVOFF	サーボ停止	6-26
	位置決め	POS	目標位置と速度を指定して位置決め	6-28
	外部位置決め	EXPOS	移動中に外部位置決め信号をわで位置決め	6-32
	原点復帰	ZRET	原点復帰	6-36
	定速送り	FEED	移動方向と速度を指定して移動開始	6-41
	コマンド中断	ABORT	実行中コマンドの中断	6-45
	コマンド一時停止	HOLDS	実行中コマンドの一時停止	6-45
	コマンド一時停止解除	HOLDE	実行中コマンドの一時停止解除	6-45
	速度変更	CHGV	速度の変更	6-47
	速度・位置変更	CHGVP	速度・位置の変更	6-50
	トルク指令	TRQ	トルク値の設定・変更	6-53
	速度オーバーライド	CHGO	速度に対する%値の変更	6-56
	トルク制限値変更	CHGTL	トルク制限値の変更	6-59
	速度ループゲイン変更	KVS	速度ループゲインの変更	6-61
	位置ループゲイン変更	KPS	位置ループゲインの変更	6-63
	単位設定	CHGU	位置、速度の単位やフィルタタイプなどを変更	6-65
	アラームクリア	ALMCLR	指定した軸のアラームクリア	6-69
	コマンドなし	NOP	モーション制御命令情報をクリア	6-71

このページは白紙です。

## 3 プロセス

### 3 プロセス

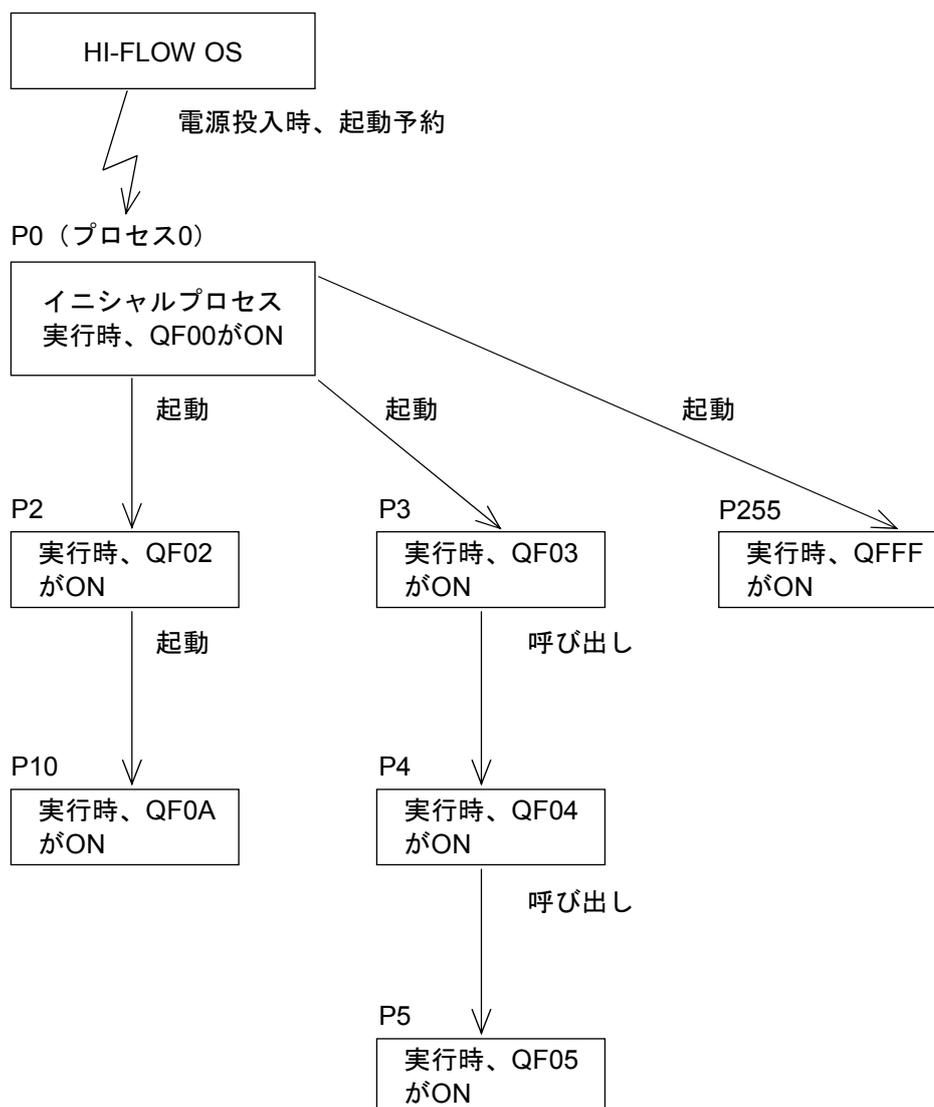
#### 3.1 プロセスとは？

プロセススタート（●）とプロセスエンド（●）または非同期プロセスエンド（▼）で括られたものはプロセスと呼ばれ、HI-FLOWプログラムの最大構成単位になります。プロセスは最低1つのルートから構成されたプログラムとプロセスに付属する情報のプロセス情報から構成されます。ユーザーは目的、機能別に1個または複数のプロセスを作成し、対象の設備を制御してください。

プロセスはP+プロセス番号（10進数）で認識します（P0～P255）。

P0はイニシャルプロセスと呼ばれ、PCsの電源投入時、HI-FLOW実行管理部（HI-FLOW OS）から起動予約されます。イニシャルプロセスからの起動がきっかけとなってP1～P255のプロセス間の制御ができます。

プロセスが実行中の場合、指定したPI/OレジスターがONし、その状態を監視できます（標準QF00～QFFF、「ソフトウェアマニュアル オペレーション HI-FLOW For Windows®（マニュアル番号 SVJ-3-132）」「4.7 HI-FLOWプロセスシートユーティリティ機能」のシステムビット割り付けコマンド参照）。



## プロセスの状態

PCs上のプロセスは9つの状態が考えられます。

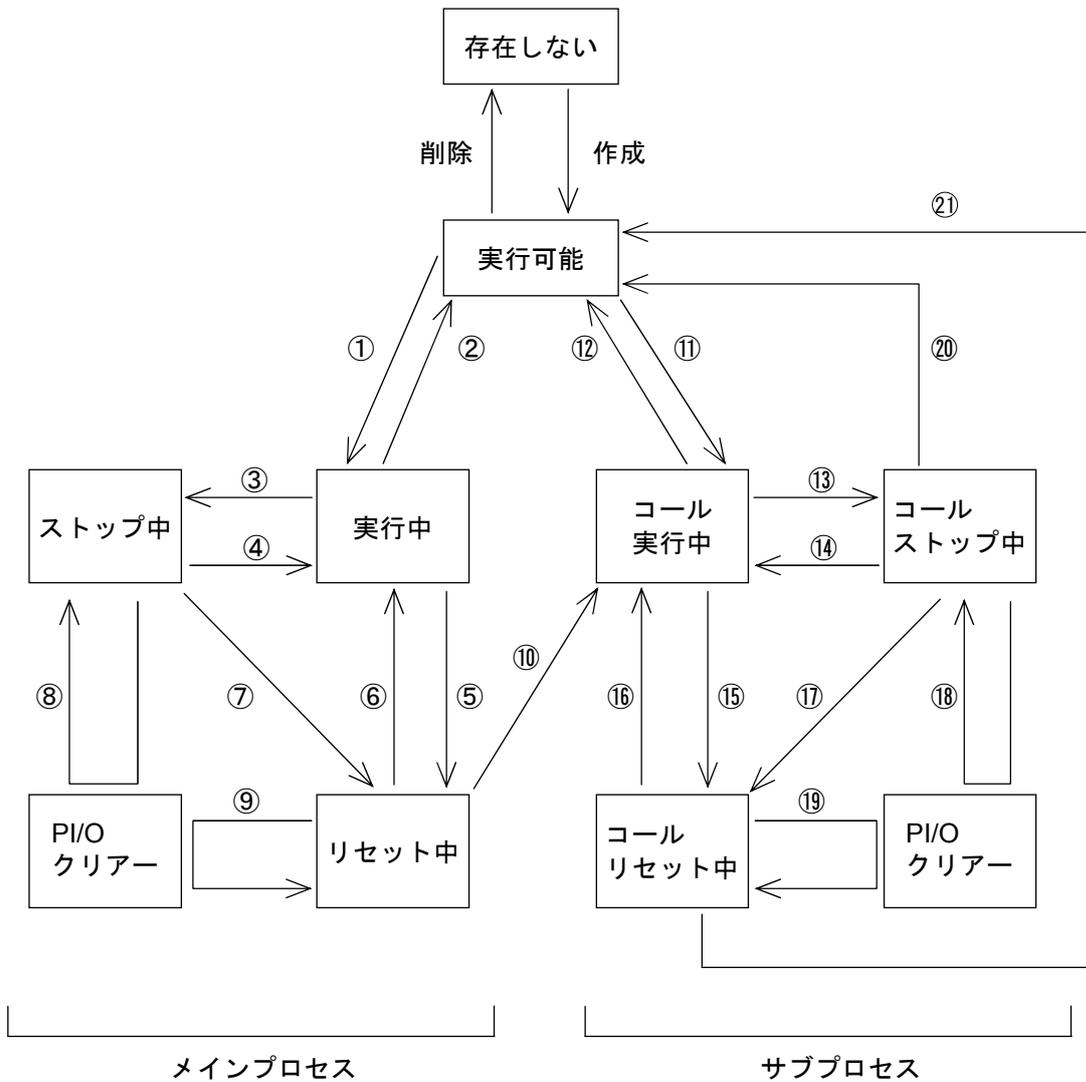
状態	内容
存在しない	HI-FLOWプロセスが存在しない。
実行可能	HI-FLOWプロセスが存在し、起動されると動作可能な状態。
実行中	HI-FLOWプロセスが他のプロセスからACT起動されて実行している状態。
ストップ中	HI-FLOWプロセスが何らかの条件成立で、プロセスのある地点で実行を停止している状態。プロセスの情報、PI/O値は保持。 タイマー経過値は保持と計測続行を指定。
リセット中	HI-FLOWプロセスが何らかの条件成立で、実行を打ち切りプロセススタートで停止している状態。プロセスの情報は初期化。 PI/O値は保持。タイマー経過値はアップとリセットを指定。
クリアー	HI-FLOWプロセスがストップ中／リセット中／コールストップ中／コールリセット中に何らかの条件成立でプロセスで使用しているビット型PI/O（ON文、パラレルタイマー）を0クリアーする。
コール実行中	HI-FLOWプロセスが他のプロセスからサブルーチンコールされて実行している状態。
コールストップ中	HI-FLOWプロセスがコール実行中に何らかの条件成立で、プロセスのある地点で実行を停止している状態。プロセスの情報、PI/O値は保持。タイマー経過値は保持と計測続行を指定。
コールリセット中	HI-FLOWプロセスがコール実行中何らかの条件成立で、実行を打ち切りプロセススタートで停止している状態。プロセスの情報は初期化。 PI/O値は保持。タイマー経過値はアップとリセットを指定。

ストップ中、リセット中は、一度の条件成立で状態が遷移し、その条件が不成立になっても状態はそのままですが、クリアーは条件が成立するたびに毎回行われます。

### 3 プロセス

#### プロセスの状態遷移

プロセスには9つの状態が存在しますが、その状態が何によって（図中の数字）、どのように（図中の矢印）遷移するか示します。



状態遷移関係図

- ① コントロールボックスACT ( ■ )
- ② エスケープ ( ✕ )
- ③ プロセススタートSTP ( ● )、コントロールボックスSTP ( ■ )
- ④ プロセススタートACT ( ● )、コントロールボックスACT ( ■ )
- ⑤ プロセススタートRST ( ● )、コントロールボックスRST ( ■ )
- ⑥ プロセススタートACT ( ● )、コントロールボックスACT ( ■ )
- ⑦ プロセススタートRST ( ● )、コントロールボックスRST ( ■ )
- ⑧ プロセススタートCLR ( ● )、コントロールボックスCLR ( ■ )
- ⑨ プロセススタートCLR ( ● )、コントロールボックスCLR ( ■ )
- ⑩ プロセスコール ( □ )
- ⑪ プロセスコール ( □ )
- ⑫ プロセスエンド ( ● )、エスケープ ( ✕ )  
コール元プロセスへのコントロールボックスRST ( ■ )  
コール元プロセスのプロセススタートRST ( ● )
- ⑬ プロセススタートSTP ( ● )  
コール元プロセスへのコントロールボックスSTP ( ■ )  
コール元プロセスのプロセススタートSTP ( ● )
- ⑭ プロセススタートACT ( ● )  
コール元プロセスへのコントロールボックスACT ( ■ )  
コール元プロセスのプロセススタートACT ( ● )
- ⑮ プロセススタートRST ( ● )
- ⑯ プロセススタートACT ( ● )
- ⑰ プロセススタートRST ( ● )
- ⑱ プロセススタートCLR ( ● )  
コール元プロセスへのコントロールボックスCLR ( ■ )  
コール元プロセスのプロセススタートCLR ( ● )
- ⑲ プロセススタートCLR ( ● )
- ⑳ コール元プロセスへのコントロールボックスRST ( ■ )  
コール元プロセスのプロセススタートRST ( ● )
- ㉑ コール元プロセスへのコントロールボックスRST ( ■ )  
コール元プロセスのプロセススタートRST ( ● )

プロセスが実行中またはコール実行中へ遷移するとき、マスターリセット／ゾーンの2種類の起動指定があります。無指定の場合はゾーン起動となります。

プロセスエンド ( ● )、エスケープ ( ✕ )、または実行可能状態に遷移するとき、PI/O値の選択 (保持または0クリア)、タイマー経過値の選択 (アップ/リセット/計測続行) は起動のされかたに従います。

### 3 プロセス

#### PCsキースイッチとプロセスの状態

PCs上のプロセスの状態がPCsキースイッチ、PCs停復電によってどうなるかを示します。

HI-FLOWではPCs状態がRUNとSIM RUNは区別されず、PCs本体の動作に従います。

[ a ] PCs停復電 (PCsキースイッチリセット)

PCs停復電が発生するとPCs上に存在するすべてのプロセスは初期化されます。

初期化の内容

- ・プロセスの状態を実行可能にする
- ・タイマーを停止する
- ・PI/OをOFFする (DW, FW, K, KWは保持します)

また、プロセス0 (イニシャルプロセス) は起動予約されます。起動予約とは次にPCsキースイッチがRUNになったときにプロセスが実行中になることです。

[ b ] PCsキースイッチSTOP中

PCsキースイッチがSTOP中のときPCsのPI/O、タイマーの状態が変化してもプロセスの状態はそのままです。

[ c ] PCsキースイッチRUN (SIM RUN) 中

PCsキースイッチがRUN (SIM RUN) 中のときPCsのPI/O、タイマーの状態が変化するとプロセスの状態はそれに対応します。

[ d ] PCsキースイッチSTOP→RUN (SIM RUN)

PCsキースイッチがSTOPからRUN (SIM RUN) になると、[ b ] から [ c ] になります。その際、PCs停復電直後だとプロセス0が実行中になります。

PCs停復電直後だけでなく指定によってPCs停復電と同じ効果 (ただし、HI-FLOW関連だけ) の後 [ c ] になることもできます (「ソフトウェアマニュアル オペレーション HI-FLOW For Windows® (マニュアル番号 SVJ-3-132)」 「4. 7 HI-FLOWプロセスシートユーティリティ機能」のシステムエディションコマンド参照)。

[ e ] PCsキースイッチRUN (SIM RUN) →STOP

PCsキースイッチがRUN (SIM RUN) からSTOPになると、[ c ] から [ b ] になります。その際、計測中のタイマー (WT, PT) も計測停止します。

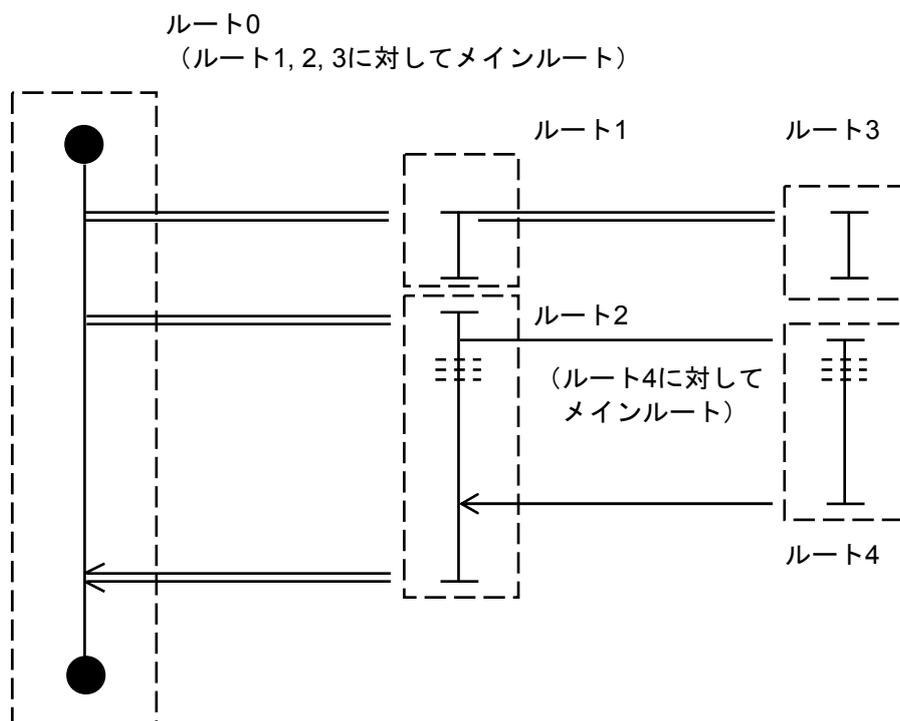
## 3.2 プログラム

プロセスはプログラムとプロセス情報で構成されていますが、プログラムは実際に設備を制御する部分で1つまたは複数のルートから構成されています。

### ルート

プロセススタート（●）とプロセスエンド（●）、プロセススタート（●）と非同期プロセスエンド（▼）またはルートスタート（⊥）とルートエンド（⊥）で括られた縦の流れはルートと呼ばれ、プロセスプログラムの構成単位となります。プロセスは複数のルートによって同期、選択処理ができます。分岐が発生するルートをメインルート、分岐したルートをサブルートと呼びます。サブルートはパラスタート（⊥）またはセレクト（⊥）によって分岐し、パラエンド（⊥）またはセレクトエンド（⊥）によって合流します。

ルートに対してユーザーは特に番号で認識する必要がないため、ルート番号はシステムでだけ管理しています。



同期ルートは必ずしも合流する必要はありません。その場合、分岐元ルートはルートに起動をかけただけとなります。

選択ルートはたとえ無条件分岐していても他のどこかのルートに合流する必要があります。

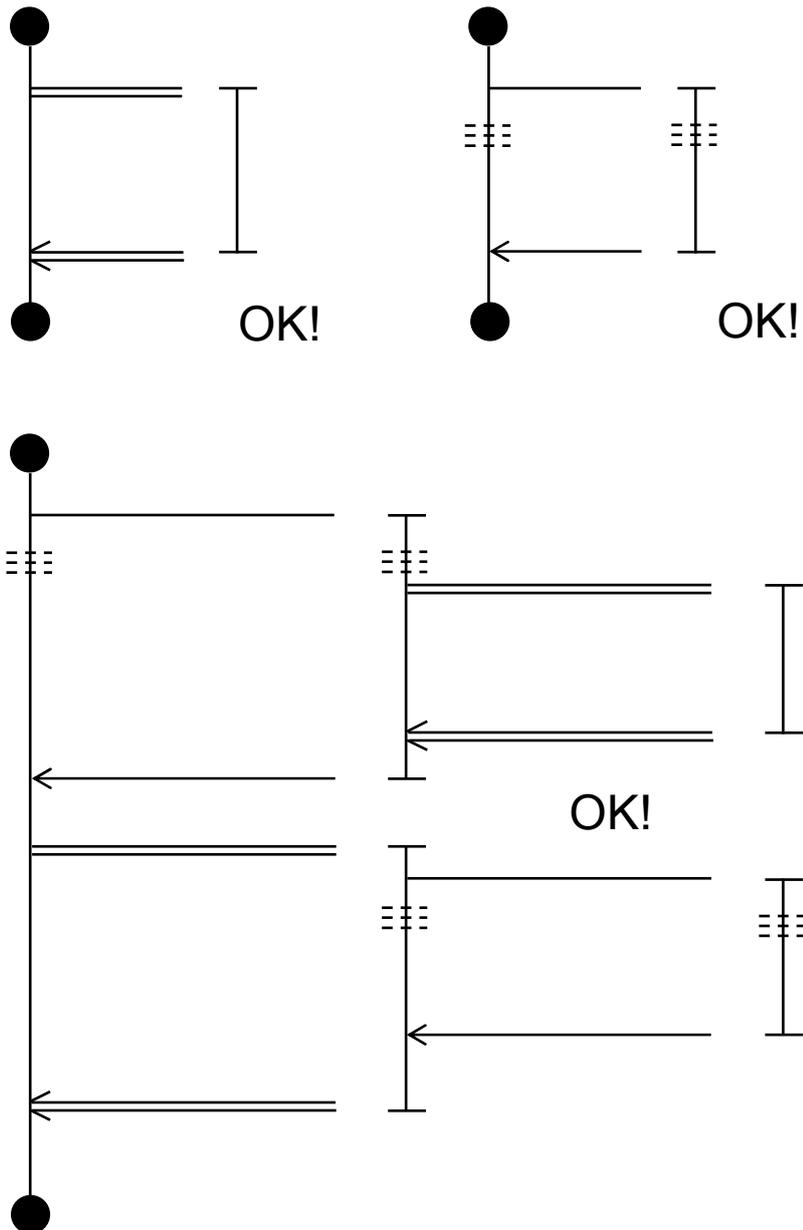
### 3 プロセス

#### (1) 同期構文と選択構文の混合

同期構文と選択構文がそれぞれ閉じた形でプログラミングされた場合は問題ありませんが、混合して作成される場合は注意が必要です。

##### (a) 分岐開始ルートと分岐合流ルートが同じ

同期、選択共にあらゆるパターンが可能です。

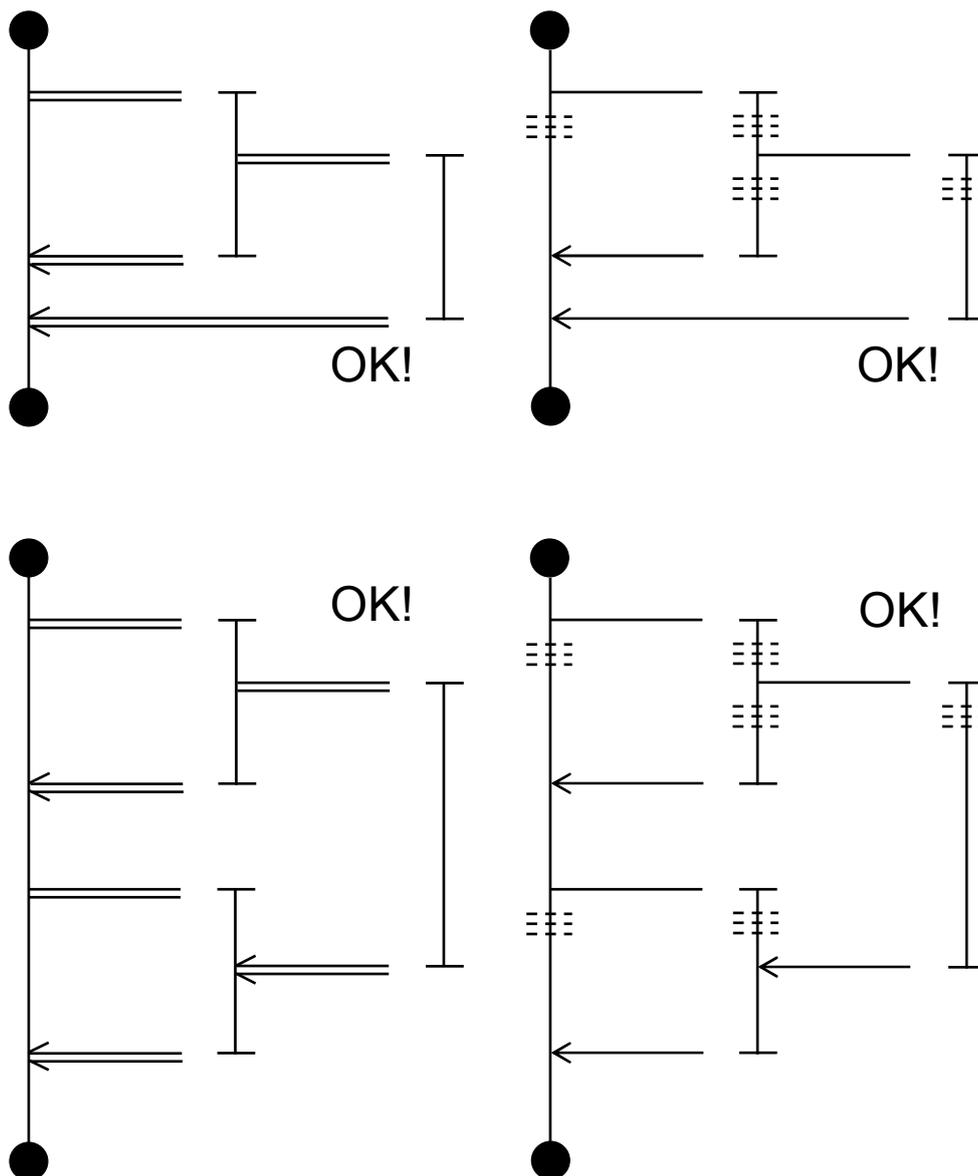


(b) 分岐開始ルートと分岐合流ルートが異なる

同期、選択構文で閉じている場合は可能ですが、それ以外では正しく動作しません。

これはプログラムとしては作成できますが、実際はうまく動作しないということです。

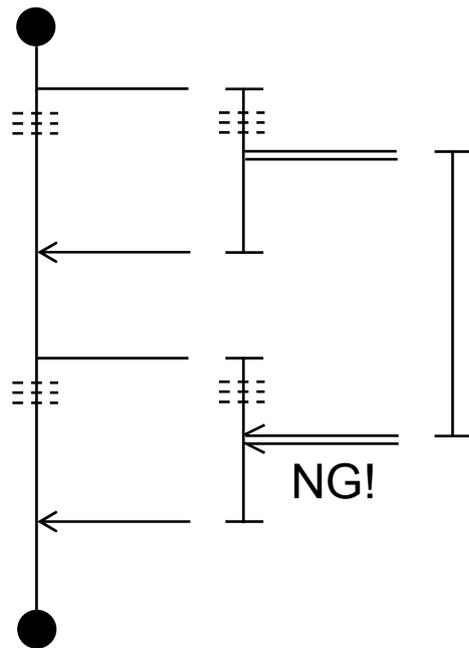
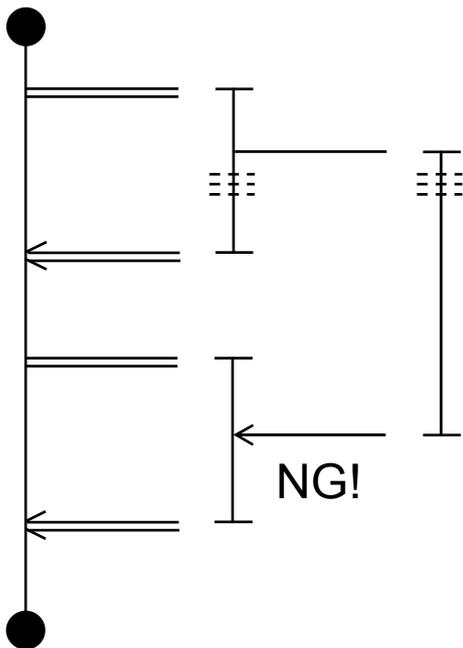
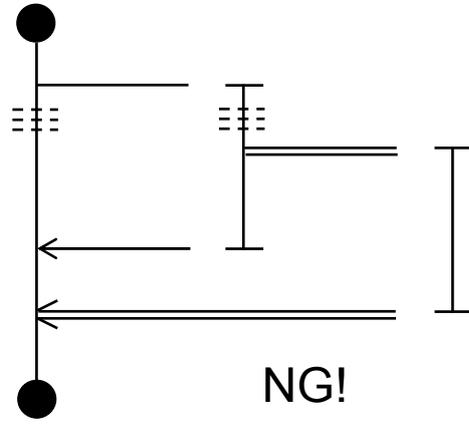
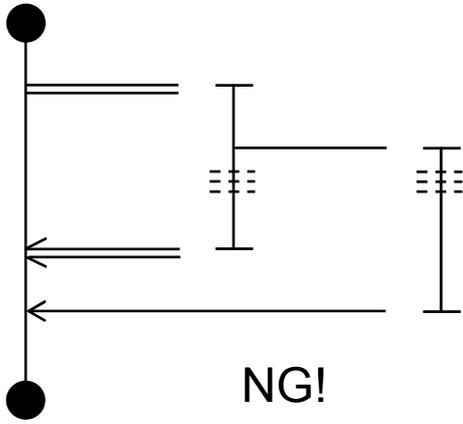
[正常動作]



### 3 プロセス

---

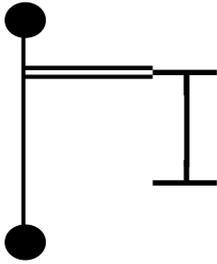
[正常動作しない場合]



## (2) 非同期ルートの構文

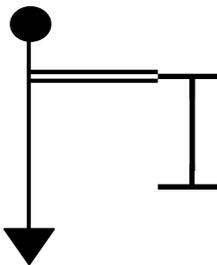
非同期ルートの構文は、非同期プロセスエンド使用時の合流なし分岐ルートを示します。同期ルートとの違いは下記となります。

## ① 同期ルート（プロセスエンド使用時）の場合



- ・プロセスエンド到達時に合流していない全分岐ルートの終了を待ってからプロセスを終了する。

## ② 非同期ルート（非同期プロセスエンド使用時）の場合



- ・非同期プロセスエンド到達時に合流していない分岐ルートが実行中でも、左図ではメインルートを終了する。
- ・分岐ルートの実行を継続したまま、次スキャンでメインルートを起動する。

詳細は、「4.17 非同期プロセスエンド」を参照してください。

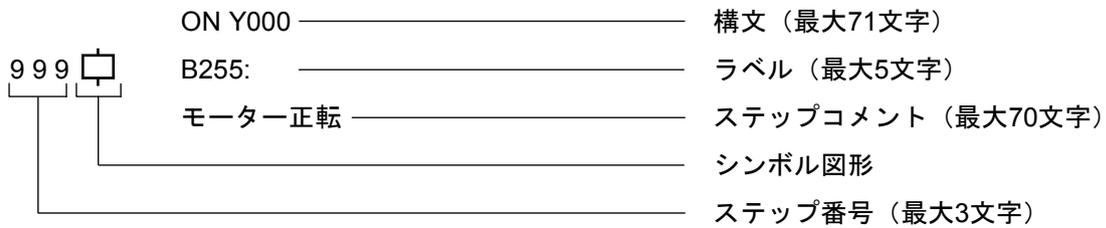
全分岐ルートが合流している場合、非同期プロセスエンドはプロセスエンドと同じ動作となります。

### 3 プロセス

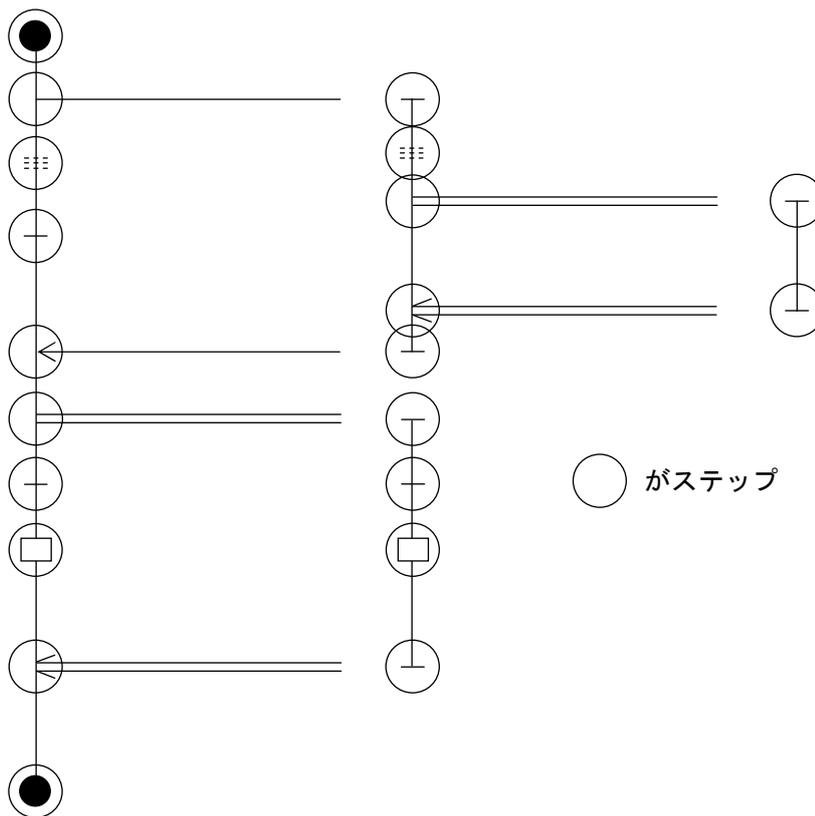
#### ステップ

ステップは自由ラベル、自由コメントと共にルートの構成単位となります。

ステップはステップ番号、シンボル図形、ラベル、構文、ステップコメントからなります。



※構文、ラベル、ステップコメントを組み合わせる場合、入力できるのは合計で70文字までです。また、構文中の論理演算子は、編集上1文字ですが、カウント上2文字で計算されます。



## ステップ番号

プロセス内のステップの一貫番号です。システムがプログラム作成時に自動割り付けします（1～999まで、つまり1プロセスは999ステップまで作成できます）。



## シンボル図形

条件、分岐、制御などの概略を意味します。ステップを作成するときは必ずシンボル図形が必要です。

シンボル図形だけでステップが完結するものと構文が必要なものがあります。



シンボル図形は19種類あり、図形そのものの形に意味があります。

次に図形の一覧を示します。

※構文、ラベル、ステップコメントを組み合わせる場合、入力できるのは合計で70文字までです。また、構文中の論理演算子は、編集上1文字ですが、カウント上2文字で計算されます。

### 3 プロセス

#### HI-FLOWで使用できる図形一覧

(1/3)

番号	図形	名称	機能	構文	備考
1		プロセス スタート	プロセスの開始	あり	
2		プロセス エンド	プロセスの終了	なし	
3		非同期 プロセス エンド	プロセスの終了	なし	非同期ルートはプロセスエ ンドで同期待ちしない
4		ルート スタート	サブルートの 開始	なし	
5		ルート エンド	サブルートの 終了	なし	
6		リピート スタート	繰り返し処理の 開始	あり	終了判定は>=で行う
7		リピート エンド	繰り返し処理の 終了	なし	
8		イフ	条件による処理 の分岐	あり	他ルート分岐可能
9		ジャンプ	無条件分岐	なし	他ルート分岐可能
10		エスケープ	自プロセスの 強制終了	なし	サブプロセスの場合、同一 スキャンでメインへ戻る
11		パラ スタート	同期サブルート への分岐	なし	
12		パラ エンド	同期サブルート の同期待ち	なし	同期待ちが成立したら同一 スキャンで次ステップへ
13		セレクト	選択サブルート への分岐	なし	
14		セル ウェイト	選択分岐時の ルート選択条件	あり	ルートスタート、セレクト と対で使用

(2/3)

番号	図形	名称	機能	構文	備考
15		セレクト エンド	選択サブルート の合流	なし	分岐元ルートに合流しなく てもよい スキャン遅れなしで次ス テップへ
		マルチ エントリー	設定した条件が 成立すると、こ のステップから 再実行	あり	
16		ウェイト	移行条件成立 待ち	あり	
			指定時間経過 待ち	あり	PI/O連続成立監視可能
17		ボックス	PI/O出力	あり	インターロック付きY出力 あり
			代入式	あり	
			PI/O波形出力	あり	
			タイマー リセット	あり	7個までの制限なし
			タイマー アップ	あり	
18		コント ロール ボックス	他プロセスに 対する状態制御	あり	マスターリセット機能あり
			タスク制御	あり	
19		コール	他プロセスの サブコール	あり	マスターリセット機能あり
20		ファンク ション	応用命令	あり	

### 3 プロセス

---

(3/3)

番号	図形	名称	機能	構文	備考
21	+*	前状態 クリアー 付き条件	条件移行時 PI/Oクリアー	あり	ウェイトと対で使用
		前状態 クリアー 付き ウェイト タイマー	タイマーアップ 時PI/Oクリアー	あり	
22		モーション	モーション制御 命令	あり	

## ラベル

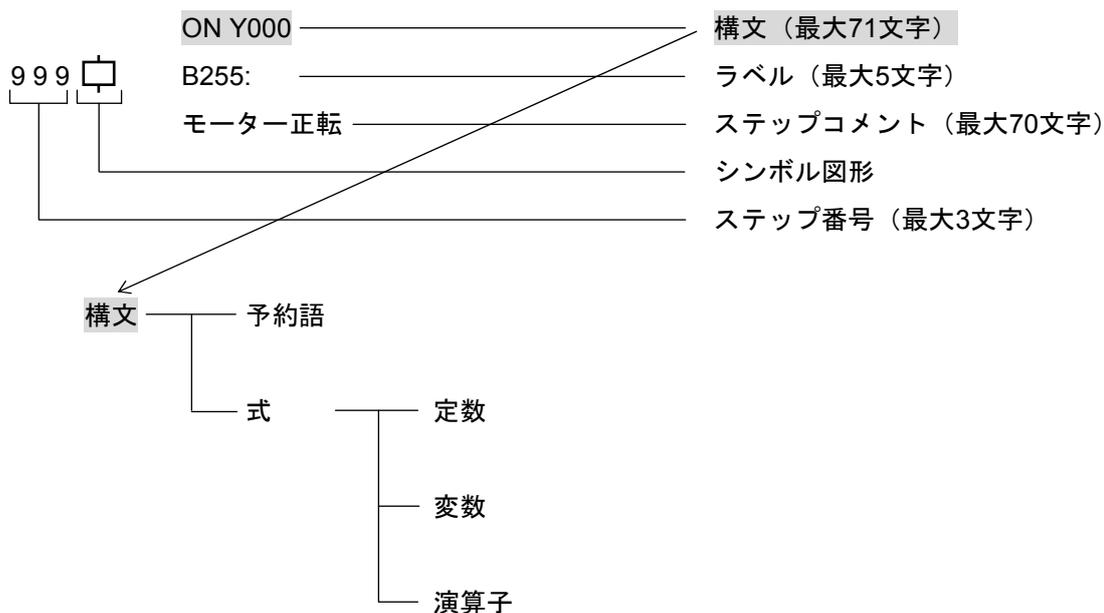
B1～B255（プロセスごとに作成可能、他プロセスへの分岐指定はできません）と：（コロン）で表し、分岐図形からの飛び先を示します。ステップにだけ付加できます。



## 構文

条件式、代入式、制御文などで図形の補助をし、内容を明らかにします。構文の必要のないシンボル図形もあります。

構文は予約語と定数、変数、演算子からなる式で構成されています。



※構文、ラベル、ステップコメントを組み合わせる場合、入力できるのは合計で70文字までです。また、構文中の論理演算子は、編集上1文字ですが、カウント上2文字で計算されます。

### 3 プロセス

#### 予約語

システムで特別な意味を持たせているため、ユーザーはシンボル名称として利用できませんので注意してください。

予約語一覧	
ACT, CLR, MRST, ON, OFF, RST, STP, TASK, TUP, TRS, TCNT, CNxxx, PTxxx, WTxxx, Bxxx, Pxxx, H????????	
応用命令名称 (第5章を参照してください)	
xxx	----- 10進定数を意味します
????????	----- 16進定数を意味します

#### 定数

HI-FLOWではロング長の定数が指定できます。

定数	ビット型: 0, 1
	ワード型: 10進数 -32768~32767 16進数 H0~HFFFF
	ロング長: ただし、応用命令内でだけ使用できます。 10進数 [-2147483648] ~ [2147483647] 16進数 [H0] ~ [HFFFFFFFF]

#### 変数

HI-FLOWでは実PI/Oレジスター (X, Yなど) が使用できます。

応用命令では@をPI/Oの前に付けることによる変数のインダイレクト指定、 [ ] による変数のロング長扱いができます。

HI-FLOWで使用可能な実PI/Oレジスター一覧を示します。

変数	ビット型	単純
		一次元配列 例: X000 (5)
	ワード型	単純
一次元配列 例: XW000 (FW000)		
ロング型	単純	ただし、応用命令内でだけ使用できます。 [FW000] でFW000とFW001をロング長で扱います。

ワード型およびロング型の値の範囲は以下のように扱います。

ワード型: -32768~32767

ロング型: -2147483648~2147483647

## PI/Oレジスタ一覧

項目	シンボル	範囲	型	備考	
レジスタ	外部入力	X	000 ~ FFF	ビット	
		XW	000 ~ FF0	ワード	
	外部出力	Y	000 ~ FFF	ビット	
		YW	000 ~ FF0	ワード	
	通信リンク レジスタ	G	000 ~ FFF	ビット	
		GW	000 ~ FF0	ワード	
		A	000 ~ FFF	ビット	
		AW	000 ~ FF0	ワード	
	内部レジスタ	R	000 ~ FFF	ビット	
		RW	000 ~ FF0	ワード	
		K	000 ~ FFF	ビット	
		KW	000 ~ FF0	ワード	
		M	000 ~ FFF	ビット	
		MW	000 ~ FF0	ワード	
		E	000 ~ FFF	ビット	
		EW	000 ~ FF0	ワード	
		Z	000 ~ 3FF	ビット	
		ZW	000 ~ 3F0	ワード	
		S	000 ~ BFF	ビット	
		SW	000 ~ BF0	ワード	
	その他 レジスタ	J	000 ~ FFF	ビット	ラダーとのリンクのため
		JW	000 ~ FF0	ワード	
		Q	000 ~ FFF	ビット	
		QW	000 ~ FF0	ワード	
		HH	000 ~ 1FF	ビット	他プロセスとのリンクのため
		DW	000 ~ FFF	ワード	
FW		000 ~ BFF	ワード		
S10V 拡張レジスタ	LB	0000 ~ FFFF	ビット		
	LBW	0000 ~ FFF0	ワード		
	LWW	0000 ~ FFFF	ワード		
	LXW	0000 ~ 3FFF	ワード		
タイマー	WT	000 ~ 255		10進表記	
	PT	000 ~ 255			
カウンター	CN	000 ~ 127		10進表記	
ラベル	B	001 ~ 255 ユーザー設定ラベルあり (6文字以内)		10進表記、プロセスごと	

### 3 プロセス

#### 演算子

演算子には論理／四則／関係／カッコの4種類があります。また四則演算は符号なしとして扱います。

項目	内容	優先順位
演算子	論理 & (アンド)   (オア) ~ (ノット) ^ (イオア)	5
	四則 * / + -	2
		3
	関係 =、<>、<、>、>=、<=	4
	カッコ 7重まで	1

#### ステップコメント

英字／数字／カナ／平仮名／漢字／特殊記号を使用して記述します。  
一行に入るだけ記述できます。必ずしも作成する必要はありません。



※構文、ラベル、ステップコメントを組み合わせる場合、入力できるのは合計で70文字までです。また、構文中の論理演算子は、編集上1文字ですが、カウント上2文字で計算されます。

## 自由ラベル

HI-FLOWではステップ以外に飛び先ラベルを作成できます（省略可能）。これを自由ラベルといい、英字から始まる6文字以内で予約語以外の名称をユーザーが自由に付けることができます。最後に「[コロン]」が必要です。

自由ラベルはステップ以外にだけ付加できます。分岐図形の飛び先となります。

LABEL: \_\_\_\_\_ 自由ラベル（6文字以内）  
合流地点 \_\_\_\_\_ 自由コメント（70文字以内）

## 自由コメント

HI-FLOWではステップ以外の箇所にコメントを作成できます（省略可能）。これを自由コメントといい、英字／数字／カナ／平仮名／漢字／特殊記号を使用して一行に入るだけ記述できます。これによってわかりやすい箇所にコメントが付加できます。

自由コメントはステップ以外にだけ付加できます。

LABEL: \_\_\_\_\_ 自由ラベル（6文字以内）  
合流地点 \_\_\_\_\_ 自由コメント（70文字以内）

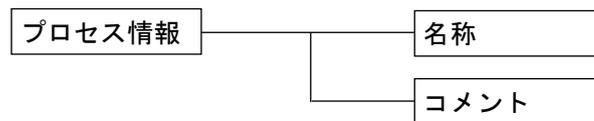
※自由ラベル、自由コメントを組み合わせる場合、入力できるのは合計で70文字（自由ラベルの「[コロン]」を含む）までとなります。

## 3 プロセス

### 3.3 プロセス情報

プロセスはプログラムとプロセス情報で構成されていますが、プロセス情報でプロセスに関する付属的な情報の定義付けを行います。これによりユーザーにとってより親しみやすいプロセスを作成できます。

プロセス情報は2つの要素で構成されており、プロセス情報コマンドによってユーザーが任意に変更できます。



#### 名称

プロセス情報のなかの名称でユーザーはそのプロセスに半角16文字、全角（漢字）8文字以内でユニークな名前を付けることができます。

#### コメント

プロセス情報のなかのコメントでユーザーはそのプロセスに半角132文字、全角（漢字）66文字以内でコメントを付けることができます。

## 4 構文説明

## 4 構文説明

この章では図形および飛び先ラベルと組み合わせた構文種類とその内容を示します。また代表的な例をいくつか示します。

なお、[ ] は省略可能、{ } は選択、～は繰り返しを意味します。

### 4.1 プロセススタート、プロセスエンド

プロセスの開始、終了を意味します。図形は自動的に付加されますので入力する必要はありません。

プロセススタートはそのプロセスのストップ/リセット/リスタート/PI/Oイニシャライズを行う条件を設定できます (STP, RST, CLR, ACT参照)。

プロセスエンドは自ルート以外のルートがすべて終了していると処理を行います。終了していないと終了するまで待ち状態となります。ただし、非同期プロセスエンドを使用する非同期ルートは例外です (詳しくは「4.17 非同期プロセスエンド」を参照してください)。起動した際、マスターリセット指定されていると自プロセスでONするビット型PI/Oを0クリアーします (ON文、パラレルタイマー)。

また、自プロセスで使用中のタイマーの扱いは起動のされかたに従います。TUPオプション指定で起動されていると、使用中のタイマーをアップし、TRSオプション指定で起動されているとタイマーを打ち切ります。無指定の場合はタイマーの計測を続行します。

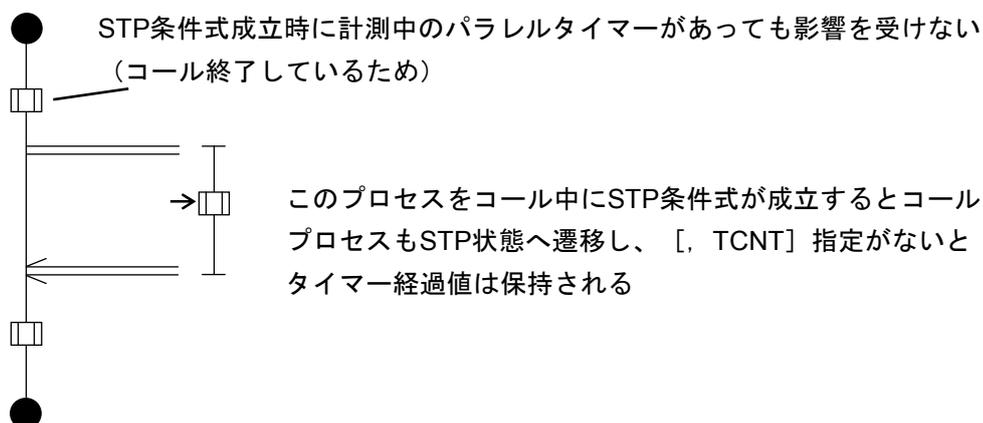
[構文]

● [ { STP 条件式 [, TCNT] [ { ON PI/Oビット群 [:OFF PI/Oビット群] }  
[ OFF PI/Oビット群 [:ON PI/Oビット群] } ] } ]  
[ { RST 条件式 [, TUP] [ { ON PI/Oビット群 [:OFF PI/Oビット群] }  
[ OFF PI/Oビット群 [:ON PI/Oビット群] } ] } ]  
[ { CLR 条件式 } ]  
[ { ACT 条件式 } ]  
\*PI/Oビット群 - PI/Oビット式 [, PI/Oビット式] ~

● または ▼ どちらも構文なし

## STP

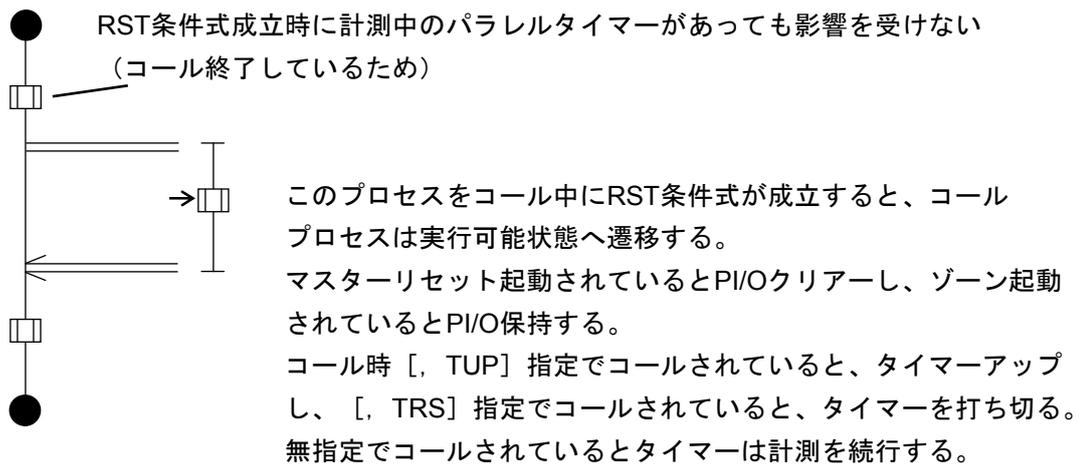
- プロセスの状態が実行中のとき、条件式が成立すると現在実行中の位置で自プロセスの実行を停止します（ストップ状態へ遷移）。
- STP成立時タイマー経過値、自プロセスでONするビット型PI/O値は保持します（ON文、パラレルタイマー）。ただし、他のプロセスなどによるON/OFFは防げませんので注意してください。
- STP条件成立時、オプション指定のPI/Oビット群を指定どおりON/OFFします（条件が不成立のときは指定の逆に毎スキャンON/OFFします）。
- [, TCNT] オプションを指定するとストップ状態遷移時タイマーは計測続行し、無指定の場合経過値を保持します。
- STP条件成立時コールされていたプロセスもコール元プロセスと同様にストップ状態へ遷移しますが、コール終了または未コールプロセスは何も影響を受けません。



## 4 構文説明

### RST

- ・プロセスの状態が実行中またはストップ中のとき、条件式が成立すると自プロセスの実行を停止し、プロセススタートで待ち状態になります（リセット状態へ遷移）。
- ・RST成立時、自プロセスでON/OFFするビット型PI/O値は保持します（ON文、OFF文、パラレルタイマー）。ただし、他のプロセスなどによるON/OFFは防げませんので注意してください。
- ・RST条件成立時、オプション指定のPI/Oビット群を指定どおりON/OFFします（条件が不成立のときは指定の逆に毎スキャンON/OFFします）。
- ・タイマーは [ , TUP ] オプションを指定すると経過値を設定値にしタイマーアップします。無指定の場合経過値を0クリアし、計測を打ち切ります。
- ・RST条件成立時コールされていたプロセスは実行可能状態へ遷移しますが、その際、PI/O、タイマーの扱いは起動のされかたに従います。  
また、コール終了または未コールプロセスは何も影響を受けません。



### CLR

ストップ状態またはリセット状態時、条件が成立すると自プロセスでONするビット型PI/Oを0クリアします（ON文、パラレルタイマーで使用しているビット型PI/O）。

### ACT

ストップ状態またはリセット状態で、かつSTP条件またはRST条件が不成立の場合、条件式が成立するとプロセスの実行を再開します（実行中へ遷移）。

[プロセススタート (●) のプログラム例] ～は同じ行の続き

1. ● STP X000, RST X001, CLR X002, ACT X003

- ・ X000がONのときストップ状態へ (タイマー経過値保持)
- ・ X001がONのときリセット状態へ (タイマー打ち切り)
- ・ ストップ/リセット状態でX002がONのときこのプロセスで使用しているON文、パラレルタイマー内ビット型PI/Oを0クリアー
- ・ ストップ/リセット状態時、X000とX001がOFFでX003がONのときプロセス実行中へ

2. ● STP G000&X020, TCNT [ON J000:OFF J001] ～, RST Q000, TUP

- ・ G000, X020が共にONのときストップ状態へ (タイマー計測続行)
- ・ ストップ状態遷移時J000をONし、J001をOFFする
- ・ Q000がONのときリセット状態へ (タイマーアップ)
- ・ プロセス実行中、毎スキャンJ000をOFFし、J001をONする

3. ● RST FW000<DW000 [OFF G100] , ACT FW001=0

- ・ FW000がDW000より小さくなったときリセット状態へ (タイマー打ち切り) リセット状態遷移時G100をOFFする
- ・ ストップ/リセット状態時、FW000がDW000以上でFW001が0のときプロセス実行中へ
- ・ プロセス実行中、毎スキャンG100をONする

4. ● RST Q001, TUP [ON J001, G200] , CLR X200

- ・ Q001がONのときリセット状態へ (タイマー打ち切り)
- ・ リセット状態遷移時J001, G200をONする
- ・ ストップ/リセット状態でX200がONのときこのプロセスで使用しているON文、パラレルタイマー内ビット型PI/Oを0クリアー
- ・ プロセス実行中、毎スキャンJ001とG200をOFFする

プロセススタート内のSTP, RST, CLR, ACTはどういう順番であってもかまいません。

## 4 構文説明

### 4.2 ルートスタート、ルートエンド

┘ 構文なし

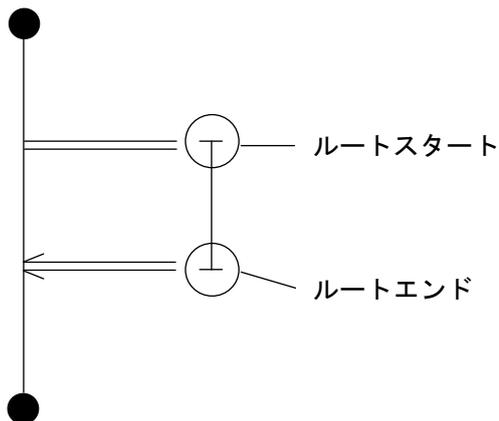
└ 構文なし

ルートスタートはサブルートの開始、ルートエンドはサブルートの終了を意味し、必ず対で使用してください。

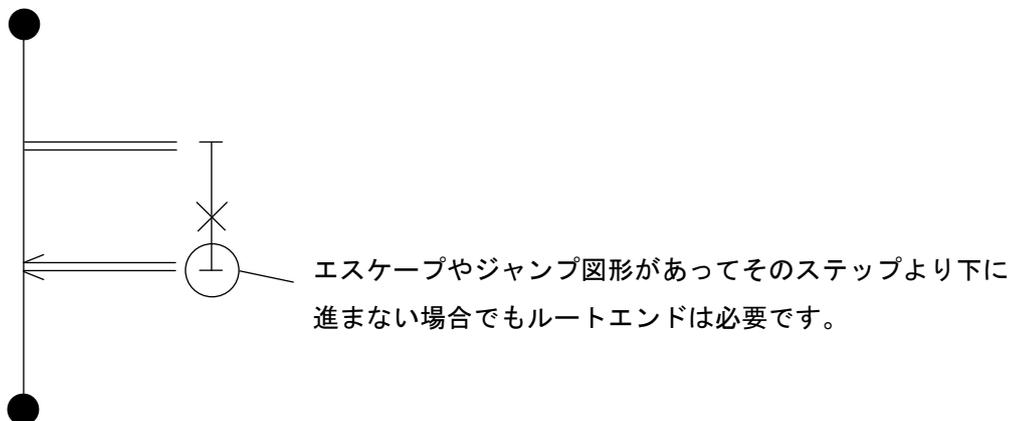
サブルートを作成することによって同期構文や選択分岐構文が構築できます。

[ルートスタート (┘), ルートエンド (└) のプログラム例]

1.



2.



### 4.3 ウェイト

次ステップへの移行条件が成立するまでこのステップで待ち状態になります。  
移行条件が条件式の場合と、ウェイトタイマー（指定時間経過待ち）の場合があります。

[構文]

```

+ { 条件式 [, タイマー, 出力ビット] }
  { WTxxx (式 [, 条件式]) }

```

#### 条件式

- ・ビット型またはワード型の数と演算子で構成されます。

#### タイマー

- ・条件式成立までの監視タイマーです。単位は100msです。
- ・10進数の定数で入力してください。
- ・設定範囲は、0～32767です。なお、-32768～-1を設定すると、32768～65535が設定されたものとして動作します。
- ・同時に監視可能なタイマー数は64個までです。同時に監視状態になるタイマー数が64個を超えないようにしてください。

#### 出力ビット

- ・上記タイマーで指定した時間が経過しても条件式が成立しない場合に、このビットがONします。
- ・出力ビットに指定できるレジスターは下記ビット型レジスターです。  
Y, G, A, R, K, M, E, Z, S, J, Q
- ・監視開始時にこのビットは無条件にOFFされます。
- ・タイマーで指定した時間が経過しても条件が成立しない限り、次ステップへは移行しません。
- ・出力ビットをONした後に条件が成立しても、出力ビットはOFFしません。

#### ウェイトタイマー

- ・ウェイトタイマーを使用すると任意のステップで、指定時間の間進行を遅延させることができます。  
WT000～WT255（番号は10進）を使用でき、遅延指定時間は100ms刻みで0～32767（10進）の範囲で設定できます。なお、-32768～-1を設定すると、32768～65535が設定されたものとして動作します。
- ・複数箇所と同じ番号のウェイトタイマーが指定時間経過待ちになった場合、タイマーを先に占有したステップがタイマー開放するまで他のステップは指定PI/O（標準HH1FA）をONし、タイマー開放待ちになります。したがって、他のタイマー遅延時間が延びる結果になります。
- ・ウェイトタイマーに条件式を設定できます。その場合、条件式が指定時間分連続成立するまで待ち状態になります。

## 4 構文説明

---

[ウェイト（ $\dagger$ ）のプログラム例]

1.  $\dagger$  X000

X000がONのとき次ステップに進む

2.  $\dagger$  GW000<H2000

GW000がH2000より小さくなったとき次ステップに進む

3.  $\dagger$  X001 (FW000)

条件チェック時（毎回異なる可能性あり）のFW000の値を添え字値としたXレジスターがONしたとき次ステップへ進む。

4.  $\dagger$  WT000 (100)

最初にこのステップに到着してから10秒経過後、次ステップへ進む。

5.  $\dagger$  WT255 (10, X01F)

このステップに到着後、X01Fが連続1秒ONし続けたら次ステップへ進む。

6.  $\dagger$  GW000>H2000, 100, Y000

GW000の値がH2000より大きくなったとき次ステップへ進む。

10秒以内にGW000の値がH2000より大きくならなかった場合はY000をONする。

Y000がONした後に、GW000の値がH2000より大きくなってもY000をOFFしない。

## 4.4 ボックス

PI/Oの出力、データ処理、タイマー制御を行います。

ボックスは：(コロン) で区切ることによって複文構成にできます。

[構文]

```

□ {ON PI/Oビット式 [, PI/Oビット式] ~}
  {OFF PI/Oビット式 [, PI/Oビット式] ~}
  {代入式}
  {PT番号 (t1 [, t2] ) , {ON ビットPI/O [, ビットPI/O] ~ [:OFF ビットPI/O [, ビットPI/O] ~] } ) }
                                {OFF ビットPI/O [, ビットPI/O] ~ }
  [ { [TUP]   {WT番号} [ { [, WT番号} ] }
    {TRS}   {PT番号} [ { [, PT番号] ~ } ]
    {CN番号} [ { [, CN番号} ] } ] ]

```

```

[ { :ON PI/Oビット式 [, PI/Oビット式~繰り返し] }
  { :OFF PI/Oビット式 [, PI/Oビット式~繰り返し] }
  { :代入式 }
  { :PT番号 (t1 [, t2] ) , {ON ビットPI/O [, ビットPI/O] ~ [:OFF ビットPI/O [, ビットPI/O] ~] } ) } } ~
                                {OFF ビットPI/O [, ビットPI/O] ~ }
  [ { [TUP]   {WT番号} [ { [, WT番号} ] }
    {TRS}   {PT番号} [ { [, PT番号] ~ } ]
    {CN番号} [ { [, CN番号} ] } ] ]

```

## 代入式

論理、四則演算の結果を変数へ代入します。式は1次元配列が可能です。配列添字値はワード型に限ります。使用可能な変数と演算子を以下に示します。

(ビット型変数)

Y, G, A R, K, M E, Z, J Q, HH, LB	=	Y, G, A R, K, M E, Z, J Q, HH, LB X, S 0, 1	( ) &   ~ ^	Y, G, A R, K, M E, Z, J Q, HH, LB X, S 0, 1
--	---	--	-------------------------	--

(ワード型変数)

YW, GW AW, RW KW, MW EW, ZW JW, QW DW, FW LBW LWW LXW	=	YW, GW AW, RW KW, MW EW, ZW JW, QW DW, FW XW, SW LBW LWW LXW 10進定数 16進定数	( ) &   ~ ^ * / + -	YW, GW AW, RW KW, MW EW, ZW JW, QW DW, FW XW, SW LBW LWW LXW 10進定数 16進定数
---	---	---	---	---

## 4 構文説明

演算項目、結果は符号なしとみなします。

乗算は乗数、被乗数とも1ワードで、1ワードで表せない部分は切り捨てられて結果も1ワードとなります。

除算も除数、被除数とも1ワードで、1ワードで表せない部分は切り捨てられて結果も1ワードとなります。

また、0除算されると答えは変化しません。

演算結果の状態（正常終了、オーバーフロー発生など）のアンサーバックはありません。アンサーバックが必要な場合は応用命令を使用してください。

### [代入文（□）のプログラム例]

1. □ FW000=FW001+FW002

その時点でのFW001とFW002の値を加算してFW000に代入し、次ステップへ進む。

2. □ YW000 (DW001) =HFFFF

その時点でのDW001を添え字値としたYW000の配列内容へ/FFFFを代入する。

## ON文

指定されたPI/O出力ビット（Y, G, A, R, K, M, E, Z, J, Q, HH）をONします。

、（カンマ）で区切ることによって複数のPI/Oの出力ができます。PI/O出力ビットは1次元配列が可能です。配列添字値はワード型に限ります。

### [ON文（□）のプログラム例]

1. □ ON Y000, Y00F:OFF Y001

Y000とY00FをONし、Y001をOFFして次ステップへ進む。

2. □ ON G000 (GW010)

G000からその時点でのGW010の値分離れたビットをONして次ステップへ進む。

## OFF文

指定されたPI/O出力ビット（Y, G, A, R, K, M, E, Z, J, Q, HH）をOFFします。

、（カンマ）で区切ることによって複数のPI/Oを出力できます。PI/O出力ビットは1次元配列が可能ですが、配列添字値はワード型に限ります。

### [OFF文（□）のプログラム例]

1. □ OFF Y000, Y001

Y000とY001をOFFして次ステップへ進む。

2. □ OFF G000 (GW010)

G000からその時点でのGW010の値分離れたビットをOFFして次ステップへ進む。

## パラレルタイマー

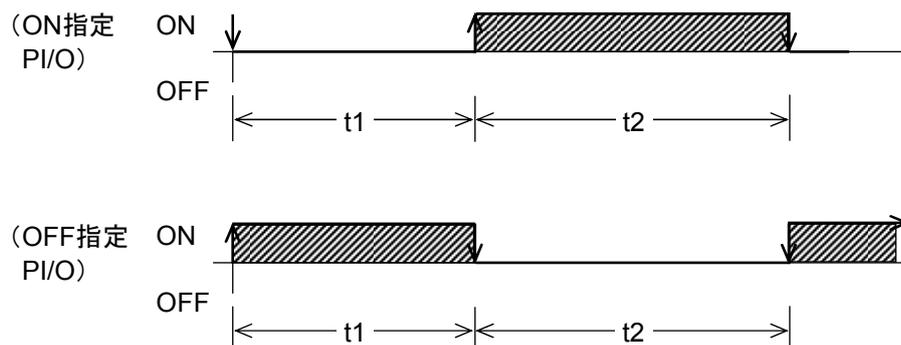
任意のPI/Oに波形を出力します。t1は立ち上がり時間、t2は立ち下がり時間を示します。

t1が0のとき、ON指定PI/Oはt2時間経過後の立ち下がりだけとなり、OFF指定PI/Oはt2時間経過後の立ち上がりだけとなります。t2が0または省略時、ON指定PI/Oはt1時間経過後の立ち上がりだけとなり、OFF指定PI/Oはt2時間経過後の立ち下がりだけとなります。また、波形出力指令後はすぐに次ステップへ進みます。

PT000～PT255の範囲で使用でき、指定時間は100ms単位、0～32767の範囲でt1, t2それぞれに設定します。なお、-32768～-1を設定すると、32768～65535が設定されたものとして動作します。

タイマー起動時、指定タイマーがすでに使用中の場合は指定PI/O（標準HH1F9）をONし、タイマーが開放されるまで待ち状態となります。

使用できるビットPI/Oは、（カンマ）による複数記述や：（コロン）による複文、1次元配列ができます。使用可能なビットPI/O種類はY, G, A, R, K, M, E, Z, J, Q, HHです。



## 4 構文説明

[パラレルタイマー (□) のプログラム例]

1. □ PT000 (10, 10, ON Y000:OFF Y001)

	このステップ通過時 (すぐ次ステップへ)	1秒経過後	2秒経過後
Y000	?→OFF	→ON	→OFF
Y001	?→ON	→OFF	→ON

2. □ PT010 (20, ON G000:OFF G001)

	このステップ通過時 (すぐ次ステップへ)	2秒経過後	
G000	?→OFF	→ON	→
G001	?→ON	→OFF	→

3. □ PT255 (0, 30, ON J100:OFF J101)

	このステップ通過時 (すぐ次ステップへ)	3秒経過後	
J100	?→ON	→OFF	→
J101	?→OFF	→ON	→

### TUP (タイマーアップ)

計測中のタイマーをアップ状態にします。

ウェイトタイマーの場合、計測中のタイマーの経過値を設定値にします。この結果待ち状態が解除されタイマーアップ待ちのステップは次ステップへ進みます。

パラレルタイマーの場合、タイマー経過値を $t_2$  ( $t_2$ 省略時は $t_1$ ) にします。この結果パラレルタイマーは指定時間より早くPI/O出力します。

ループカウンタの場合、カウンタの経過値を終了値にします。その結果次のループチェックで脱出します。

[タイマーアップ (□) のプログラム例]

1. □ TUP WT001, WT002, PT001, CN001

ウェイトタイマー1、2、パラレルタイマー1、カウンタ1をアップします。

## TRS (タイマーリセット)

計測中のタイマーをリセットします。

ウェイトタイマー／ループカウンタの場合、タイマーアップと同じ処理を行います。パラレルタイマーの場合、計測中のタイマーのt1, t2経過値をリセットします。指定PI/Oの状態はタイマーリセット発行時の保持となります。

### [タイマーリセット (□) のプログラム例]

1. □ TRS WT001, WT002, PT001, CN001

ウェイトタイマー1、2、パラレルタイマー1、カウンタ1をリセットします。

## 4 構文説明

### 4.5 コントロールボックス

他のプロセスに対して起動（再起動）、ストップ、リセット、PI/Oクリアーを行います。

[構文]

```

■ { ACT Pxxx { [-Pxxx] [, ステップ番号] [, MRST] [ {, TUP} ] } }
                                     {, TRS}
                                     {, TASK, ファクター番号 }
{ RST Pxxx { [-Pxxx] [, TUP] } }
  { [, TASK] }
{ STP Pxxx [-Pxxx] [, TCNT] }
{ CLR Pxxx [-Pxxx] }
    
```

ACT

	項目	内容
1	機能概略	P0～P255で指定されたプロセスに起動をかける。 - (ハイフン) でプロセスの範囲指定可能。ステップ番号指定がなければステップ1から実行する。指定されたステップはメインルートでなくてもよい。起動後はすぐ次ステップへ進む。
2	起動されたプロセスの動き	起動されたプロセスは一度きりの実行でなく、プロセスエンドが実行終了すると次スキャンで再びプロセススタートから実行する（ステップ指定されていても同様）。
3	実行中プロセス起動	コントロールボックス結果表示ビットのACTビットをONし、次ステップへ進む（標準HH1FF）。
4	存在しないプロセス起動	コントロールボックス結果表示ビットのACTビットをONし、次ステップへ進む（標準HH1FF）。
5	ストップ状態プロセス起動	ストップ状態のプロセスに起動がかかり、実行が再開する。
6	リセット状態プロセス起動	リセット状態のプロセスに起動がかかり、プロセススタートから実行が再開する。
7	タイマー状態指示	， TUPオプションの指定でプロセスエンド、エスケープ実行時および実行可能状態移行時に自プロセスで占有しているパラレルタイマーをアップする。 ， TRSオプションの指定でプロセスエンド、エスケープ実行時および実行可能状態移行時に自プロセスで占有しているパラレルタイマーをリセットする。
8	マスターリセット指定起動	， MRSTオプションの指定で、プロセスエンド、エスケープ実行時および実行可能状態移行時に自プロセスでONしているビット型PI/Oを0クリアーする（ON文、パラレルタイマー）。
9	CPMSタスク起動	， TASK, ファクタ番号オプションでPxxxをCPMSタスク（1～127）としてRLEAS, QUEUEマクロを発行する。

## RST

	項目	内容
1	機能概略	P0～P255で指定されたプロセスにリセットをかける。 - (ハイフン) で範囲指定可能。発行後はすぐ次ステップへ進む。
2	リセットされたプロセスの動き	指定されたプロセスの実行を打ち切りリセット状態へ遷移し、プロセススタートで再実行待ち（他プロセスからのACT起動か、自プロセススタートのACT条件成立で再実行）。
3	タイマー状態指示	， TUPオプションの指定でプロセスが占有しているパラレルタイマーをアップする。無指定の場合はタイマーをリセットする。このオプションが有効なのは指定プロセスだけでコールプロセスには影響を与えない。
4	被リセットプロセスのPI/O	マスターリセット起動されていると自プロセスでON/OFFしているビットPI/Oを0クリアする。
5	ストップ状態プロセスへ発行	指定されたプロセスの実行を打ち切りリセット状態へ遷移し、プロセススタートで再実行待ち（他プロセスからのACT起動か、自プロセススタートのACT条件成立で再実行）。
6	存在しないプロセスへ発行	コントロールボックス結果表示ビットのRSTビットをONし、次ステップへ進む（標準HH1FD）。
7	自プロセスへのリセット発行	自プロセス番号をパラメーターで指定する。
8	CPMSタスク停止	， TASKオプションでPxxxをCPMSタスクとしてABORTマクロ発行。

## STP

	項目	内容
1	機能概略	P0～P255で指定されたプロセスにストップをかける。 - (ハイフン) で範囲指定可能。発行後はすぐ次ステップへ進む。
2	ストップされたプロセスの動き	指定されたプロセスの実行を停止、ストップ状態へ遷移する。 現在実行位置で再実行待ち。
3	再実行条件	他プロセスからのACT起動および自プロセススタートのACT条件成立で再実行。
4	タイマー状態指示	， TCNTオプションの指定でプロセスが占有しているパラレルタイマーの計測を続行する。無指定の場合はタイマー計測を停止する。 このオプションは指定プロセスがコールでひもづけているすべてのプロセスに有効。
5	被ストッププロセスのPI/O	マスターリセット起動されていると自プロセスでON/OFFしているビットPI/Oを0クリアする。
6	存在しないプロセスへ発行	コントロールボックス結果表示ビットのSTPビットをONし、次ステップへ進む（標準HH1FE）。
7	リセット状態プロセスへ発行	コントロールボックス結果表示ビットのSTPビットをONし、次ステップへ進む（標準HH1FE）。
8	自プロセスへのストップ発行	自プロセス番号をパラメーターで指定する。

## 4 構文説明

### CLR

	項目	内容
1	機能概略	P0～P255で指定されたプロセスでON/OFFしているビットPI/Oを0クリアする。発行後はすぐ次ステップへ進む。指定プロセスの状態がストップまたはリセット状態だけ許可。他プロセスでのPI/O使用状況はチェックされずクリアするので注意する必要あり。- (ハイフン) で範囲指定可能。
2	存在しないプロセスへ発行	コントロールボックス結果表示ビットのCLRビットをONし、次ステップへ進む (標準HH1FC)。
3	実行中プロセスへ発行	コントロールボックス結果表示ビットのCLRビットをONし、次ステップへ進む (標準HH1FC)。
4	未起動プロセスへ発行	コントロールボックス結果表示ビットのCLRビットをONし、次ステップへ進む (標準HH1FC)。

#### [コントロールボックス ( ■ ) のプログラム例]

##### 1. ■ ACT P1-P5, MRST

プロセス1からプロセス5をステップ1からマスターリセット起動し、次ステップへ進みます。  
起動されたプロセスはプロセスエンド、エスケープ実行時およびプロセス実行可能状態へ遷移時  
パラレルタイマーは計測を続行します。

##### 2. ■ ACT P100, 5, TUP

プロセス100をステップ5からゾーン起動し、次ステップへ進みます。  
起動されたプロセスはプロセスエンド、エスケープ実行時およびプロセス実行可能状態へ遷移時  
パラレルタイマーはアップします。

##### 3. ■ ACT P80, TASK, 3

CPMSタスク80に対してRLEASマクロを発行し、ファクター3でQUEUEマクロを発行した後、次  
ステップへ進みます。

## 4. ■ RST P10

プロセス10をリセット状態にして次ステップへ進みます。  
RST発行状態時、計測していたパラレルタイマーはリセットされます。

## 5. ■ RST P11, TUP

プロセス11をリセット状態にして次ステップへ進みます。  
RST発行状態時、計測していたパラレルタイマーはアップされます。

## 6. ■ RST P12, TASK

CPMSタスク12に対してABORTマクロを発行した後、次ステップへ進みます。

## 7. ■ STP P50

プロセス50をストップ状態にして次ステップへ進みます。  
STP発行状態時、計測していたパラレル/ウェイトタイマーは停止されます。

## 8. ■ STP P51, TCNT

プロセス51をストップ状態にして次ステップへ進みます。  
STP発行状態時、計測していたパラレル/ウェイトタイマーは停止されず、計測続行されます。

## 9. ■ CLR P40

プロセス40で使用しているビット型PI/Oを0クリアーして次ステップへ進みます。

## 4 構文説明

### 4.6 リピートスタート、リピートエンド

リピートスタートとリピートエンドの間を繰り返し実行します。

同一ルート内でリピートスタートとリピートエンドの数が同じでないと文法エラーとなります。

繰り返すたびに増分を初期値に加算し、その値が終了値より大きくなるまで繰り返します。

初期値が終了値より大きい場合、リピートスタートとリピートエンドの間は一度も実行されずに次のステップへ進みます。また増分を省略すると1を増分とし、増分が0の場合無限ループとなります。

初期値、終了値、および増分の設定範囲は0～32767です。なお、-32768～-1を設定すると、32768～65535が設定されたものとして動作します。

[構文]

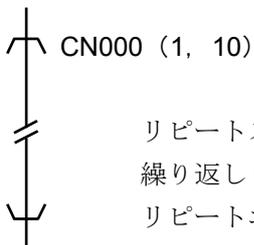
 CNxxx (初期値, 終了値 [, 増分])

(xxxは000～127の10進数)

 構文なし

[リピートスタート () , リピートエンド () のプログラム例]

1.



リピートスタートとリピートエンドの間を10回

繰り返してリピートエンドの次のステップへ進みます。

リピートエンドを実行後、すぐにリピートスタートを実行します。

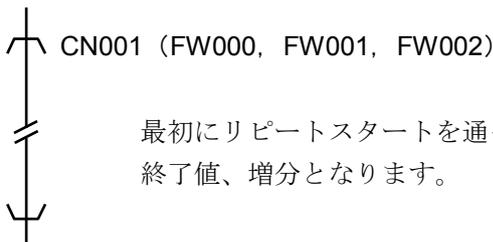
2.



リピートスタートとリピートエンドの間を3回

繰り返してリピートエンドの次のステップへ進みます。

3.



最初にリピートスタートを通ったときのFW000～FW002の値が初期値、

終了値、増分となります。

## 4.7 イフ

条件式の真偽を判定して、すぐにそれぞれに対応した処理を行います。

条件式が成立した場合（真）、,（カンマ）と ;（セミコロン）までを実行します。

条件が不成立の場合（偽）、;（セミコロン）以降を実行します。

;（セミコロン）以降を省略した場合で、条件式が不成立のときは次ステップへ進みます。

,（カンマ）、;（セミコロン）の後にラベルを指定するとそのラベルへ分岐します。

## [構文]

```

◇ 条件式, {          飛び先ラベル (Bxxx)          }
           {          自由ラベル                  }
           {ON/OFF文}  {[:ON/OFF文]}
           {代入式}   {[:代入式]}
           {ACT文}    {[:ACT文]}
           {STP文}    {[:STP文]}
           {RST文}    {[:RST文]}
           {CLR文}    {[:CLR文]}
           {TUP文}    {[:TUP文]}
           {TRS文}    {[:TRS文]}
           {PT文}     {[:PT文]}
           ~
           [ {          ;飛び先ラベル (Bxxx)          }
             {          ;自由ラベル                  }
             {;ON/OFF文} {[:ON/OFF文]}
             {;代入式}  {[:代入式]}
             {;ACT文}   {[:ACT文]}
             {;STP文}   {[:STP文]}
             {;RST文}   {[:RST文]}
             {;CLR文}   {[:CLR文]}
             {;TUP文}   {[:TUP文]}
             {;TRS文}   {[:TRS文]}
             {;PT文}    {[:PT文]}
           ]

```

(xxxは1~255の10進数)

## &lt; 留意事項 &gt;

他プロセスへの分岐は不可ですが、他ルートへの分岐は許可します。ただし、実際に実行する際、以下の場合は正しく動作しない場合がありますので注意してください。

- ・ループスタートからループエンドの内部への分岐
- ・並列処理内部からの分岐
- ・並列処理内部への分岐
- ・すでに実行中のルートへの分岐

## 4 構文説明

---

[イフ (◇) のプログラム例]

1. ◇ X000, B1;LABEL

X000がONのときはB1ラベルの存在するステップへジャンプし、OFFのときはLABELラベルの存在する箇所の次のステップへジャンプする。

2. ◇ H0<> (YW000&H3000) , ON Q005

YW000の内容とH3000の論理積が0以外ならばQ005をONし、0ならば何もしないで次ステップへ進む。

3. ◇ Q000, FW100=FW100+1;ACT P10

Q000がONのときはFW000に1を加算して次ステップへ進み、OFFのときはプロセス10にACT起動をかけ次ステップへ進む。

4. ◇ GW000=4, STP P6;RST P7;EW000=8;ON J000

GW000が4のときプロセス6をストップ状態へ、プロセス7をリセット状態にし次ステップへ進む。

GW000が4以外のときEW000を8にし、J000をONして次ステップへ進む。

5. ◇ X010, ON J000, J001, J002, J003;ERRLB

X010がONのときJ000, J001, J002, J003をONし次ステップへ進む。

X010がOFFのときERRLBラベルの存在する箇所の次のステップへジャンプする。

## 4.8 ジャンプ

プロセス内の指定したラベルへ無条件で分岐します。

1プロセスにつきラベルはB1～B255まで指定できます。

HI-FLOWでは自由ラベル（6文字以内、ユーザーが名称を自由に設定可能、ステップ以外にだけ付加可能）を指定できます。

### 【構文】

```
↳ { 飛び先ラベル (Bxxx) }
   { 自由ラベル          }
```

### <留意事項>

他プロセスへの分岐はできませんが、他ルートへの分岐は許可します。ただし、実際に実行する際、以下の場合は正しく動作しないことがありますので注意してください。

- ・ループスタートからループエンドの内部への分岐
- ・並列処理内部からの分岐
- ・並列処理内部への分岐
- ・すでに実行中のルートへの分岐

### [ジャンプ（↳）のプログラム例]

#### 1. ↳ B1

B1ラベルの存在するステップへジャンプし、そのステップからすぐ実行する。

#### 2. ↳ ERRBLK

LABELラベルの存在する箇所の次のステップへジャンプし、そのステップからすぐ実行する。

## 4 構文説明

### 4.9 エスケープ

自プロセスを強制終了させます。

メインプロセスの場合、全ルートを強制終了させ実行可能状態へ遷移します。その際、コール中のプロセスがあるとすべてエスケープさせます。自プロセスで使用中のタイマーの扱いは起動のされかたに従います（TUP, TRSオプション）。

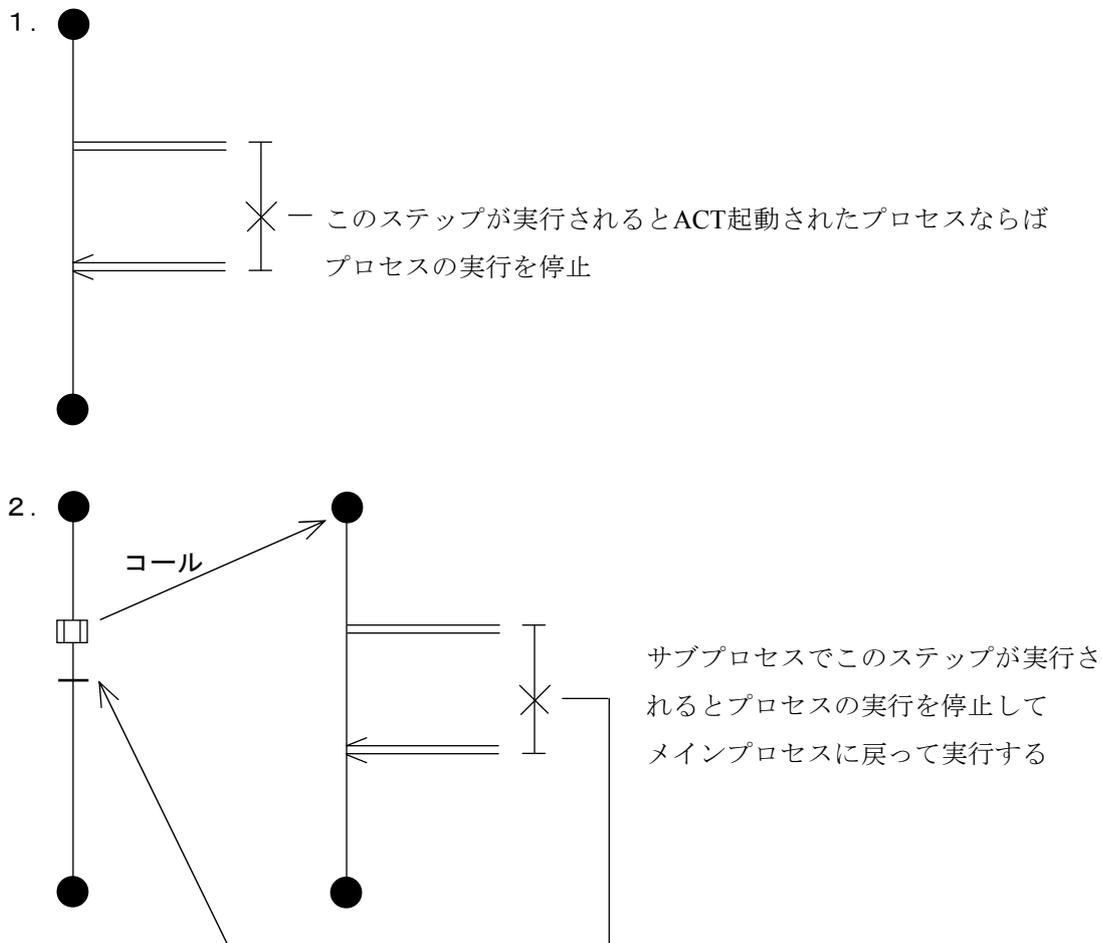
サブプロセスの場合も基本的にはメインプロセスと同じですが、実行箇所を同一スキャンでメインプロセスに戻します。

マスターリセット起動されていると、自プロセスでONするビット型PI/Oを0クリアーします（ON文、パラレルタイマー）。

[構文]

✕ 構文なし

[エスケープ（✕）のプログラム例]



## 4.10 パラスタート、パラエンド

パラスタート、パラエンド対で同期処理部分を表します。

パラスタートは同期しているサブルートに起動をかけた後、自ルートの次ステップへ進みます。

パラエンドは合流しているすべてのルートが終了後、自ルートの次ステップを実行することを示します。

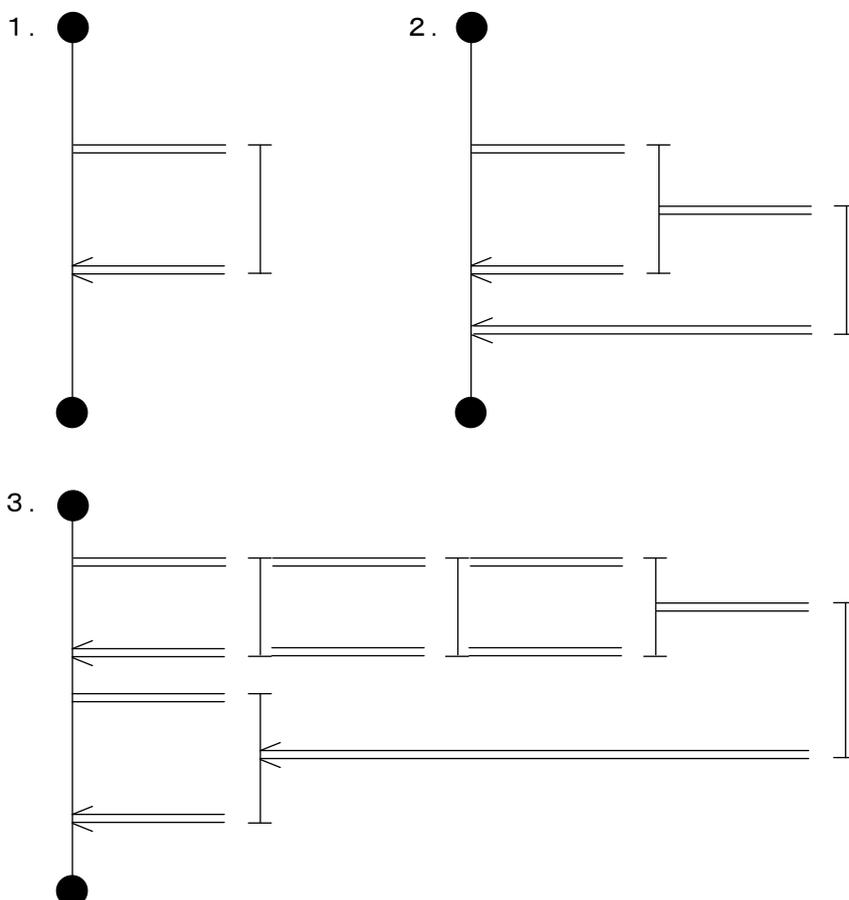
従来はパラエンドが合流しているサブルートの終了を監視（つまりメインルートは実行中）していたため次ステップ実行が1スキャン遅れました。今回はパラエンド、ルートエンドともに自分が最後に合流したかをチェックし、最後ならばメインルート合流部の次ステップ実行を、最後でなかったら自ルートの実行を終了（メインルートが常に実行中とは限らない）するような変更によって、スキャン遅れは起こりません。

【構文】

┆ 構文なし

≧ 構文なし

【パラスタート（┆）、パラエンド（≧）のプログラム例】



## 4 構文説明

### 4.11 セレクト、セルウェイト、セレクトエンド

セレクト、セルウェイト、セレクトエンド一組で選択分岐処理部分を表します。

セレクトは選択分岐ルートに起動をかけた後、自ルートのセルウェイトに進みます（セレクト、セルウェイトまたはルートスタート、セルウェイトは必ず連続していなければなりません）。

セルウェイトは自ルートの条件式が成立すると他ルートの実行を終了させて、自ルートの次ステップへ進みます。今回はサブルートが選択された場合はメインルートを終了します（選択されたルートだけ実行）。

画面左側のルートから条件式をチェックするので、同スキャンで複数の条件が成立すると一番左のルートが選択されます。

従来はセレクトエンドが合流しているサブルートの終了を監視（つまりメインルートは実行中）していたため次ステップ実行が1スキャン遅れましたが、今回はサブルートが選択されるとそのルートのルートエンドがメインルートに起動をかけ合流部の次ステップを実行するため1スキャン遅れの解消がなされています。

また、従来はセレクトエンドとセレクトは同じルートに存在しなければなりませんでした。今回は同一ルートに存在しなくても（分岐元ルートに合流しなくても）構いません。

#### <留意事項>

セルウェイトはセレクトの次ステップになければいけません。

#### [構文]

┆ 構文なし

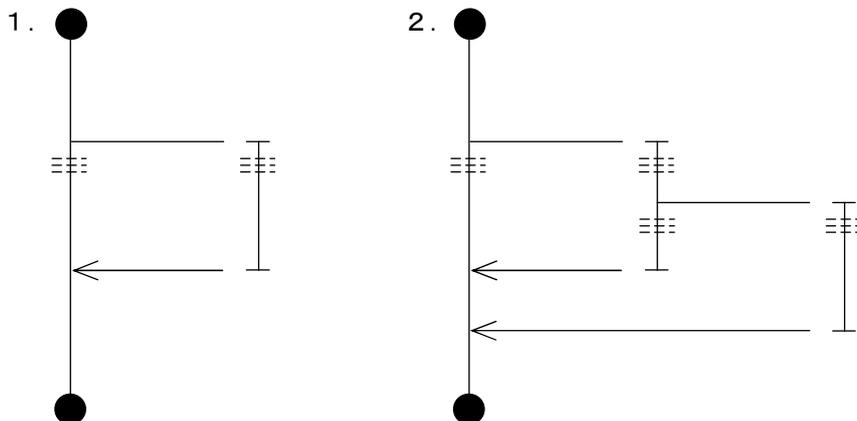
≡≡≡ 条件式 [, タイマー, 出力ビット]

≡≡≡ 構文なし

- ・タイマー
- ・出力ビット

※タイマーと出力ビットについては、「4.3 ウェイト」を参照してください。

#### [セレクト (┆), セルウェイト (≡≡≡), セレクトエンド (≡≡≡) のプログラム例]



## 4.12 マルチエントリー

セレクトエンドと同じ図形に条件式を設定するとマルチエントリーとみなします。

プロセスが実行中のとき、条件式が成立するとそのマルチエントリーの存在するステップから再実行します（プロセスの最初の実行時も条件式が成立するとそこから実行します）。条件式のチェックはスキヤンの最初に行うので、最大1スキヤンの遅れが発生する場合があります。

ステップの小さいものから条件をチェックするので、同スキヤンで複数の条件が成立するとステップ番号の一番小さいステップから再実行します。

マルチエントリーはサブルートにも設定できます。

条件が成立して実行するとき、タイマー（PT, WT）、カウンター（CN）、コール中プロセス、そのマルチエントリーが存在するルート以外のルートはすべてイニシャライズしますが、PI/O値は保持します。

### [構文]

┌ 条件式

### <留意事項>

- ・ループスタートからループエンド内部にマルチエントリーを設定すると、正しく動作しない場合がありますので注意してください。
- ・同期構文のサブルートにマルチエントリーは設定できません。

### [マルチエントリー（┌）のプログラム例]

#### 1. ┌ X000

X000がONのときこのステップから再実行する。

#### 2. ┌ GW000<H2000

GW000がH2000より小さくなったときこのステップから再実行する。

## 4 構文説明

### 4.13 コール

P0～P255で指定されたプロセスをサブルーチンコールします。[, ステップ番号] オプションで指定ステップから実行を開始します（省略するとプロセススタートから実行します）。

指定プロセスが存在しない、指定ステップが存在しない、または自プロセスをコールするとコントロールボックス結果表示ビットのCALLビットをONして次ステップへ進みます。

指定プロセスがすでに実行中の場合、そのプロセスがコールできる（実行可能状態に遷移する）まで待ち続けますが、ACT起動されて、リセット中のプロセスはコールできます。

サブプロセスからさらに他のプロセスがコール可能で最大16までネスティングができます。

[, MRST] オプションでマスターリセットコールができます。マスターリセットコールされるとコールプロセス終了時、エスケープ実行時、および実行可能状態移行時に自プロセスでONしているビットPI/Oを0クリアします。

[, TUP] オプションの指定でプロセスエンド、エスケープ実行時および実行指示可能状態移行時に自プロセスで占有しているパラレルタイマーをアップします。

[, TRS] オプションの指定でプロセスエンド、エスケープ実行時および実行可能状態移行時に自プロセスで占有しているパラレルタイマーをリセットします。無指定の場合はプロセス終了後もパラレルタイマーの計測を続行します。

#### [構文]

```
□ Pxxx [, ステップ番号] [, MRST] { [, TUP] }  
{ [, TRS] }
```

#### [コール ( □ ) のプログラム例]

##### 1. □ P1

プロセス1をステップ1からゾーンコールします。

コールされたプロセスはプロセスエンド、エスケープ実行時およびプロセス実行可能状態へ遷移時パラレルタイマーは計測を続行します。

##### 2. □ P2, 5, MRST

プロセス2をステップ5からマスターリセットコールします。

コールされたプロセスはプロセスエンド、エスケープ実行時およびプロセス実行可能状態へ遷移時パラレルタイマーは計測を続行します。

##### 3. □ P3, TUP

プロセス3をステップ1からゾーンコールします。

コールされたプロセスはプロセスエンド、エスケープ実行時およびプロセス実行可能状態へ遷移時パラレルタイマーはアップします。

## 4.14 ファンクション

ボックスでサポートされている演算、データ処理機能を補うための機能です。詳細内容は第5章を参照してください。

[構文]

 応用命令名称 パラメーター [, パラメーター] ~

## 4.15 前条件付きウェイト

移行条件が成立するまではウェイトは同じですが、条件成立後、次ステップへ進む前に前ステップがON文またはプロセスコールの場合、そのPI/OをOFFしてから進みます。前ステップがON文、プロセスコール以外の場合は何もしないで進みます（ウェイトと同じ）。

分岐によってこのステップから実行する場合、分岐元の前条件はクリアしないので注意してください。SFC規格に対応するための機能です。

[構文]

$\dagger^*$  { 条件式 }  
 { WTxxx (式 [, SB] [, 条件式] ) }

## 4.16 モーション

HI-FLOWからモーションを制御するための機能です。詳細内容は第6章を参照してください。

[構文]

 応用命令名称  
 共通パラメーター  
 軸番号 [, 軸番号] ~  
 軸別パラメーター [, 軸別パラメーター] ~

## 4 構文説明

### 4.17 非同期プロセスエンド

プロセススタートと対で使用し、非同期ルートと同期させずにプロセスエンドを実行します。

非同期ルートをプロセスエンドと非同期で動作させるには、分岐するサブルートがルートエンドで合流しないことが条件となります。

プロセスがACT起動の場合、非同期ルートは初回にサブルートを起動して以降、メインルートが自処理を進めて行きプロセスエンドに到達した際に非同期ルートが実行中であっても終了を待たずにプロセスエンドを実行します。再度、プロセススタートから処理を進めて行き非同期ルートのルートスタートに到達した場合は非同期ルートの実行中チェックを行い、実行中であれば起動をかけません。

プロセスがコール起動の場合、プロセスエンドと同様に非同期プロセスエンドで非同期ルートの終了を待つて終了します。

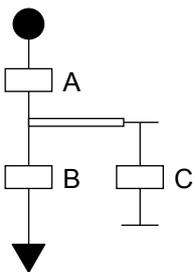
[構文]



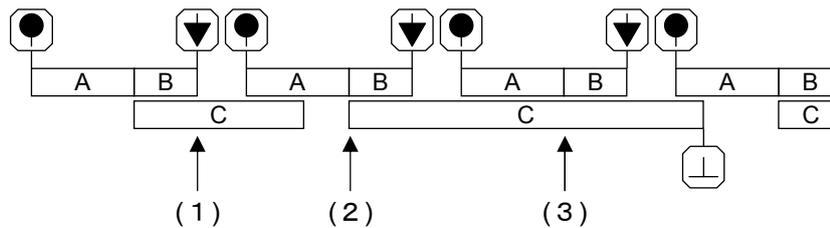
[非同期のプログラム例]

#### 1. 非同期ルートが1つの場合

<回路例>



<タイムチャート>

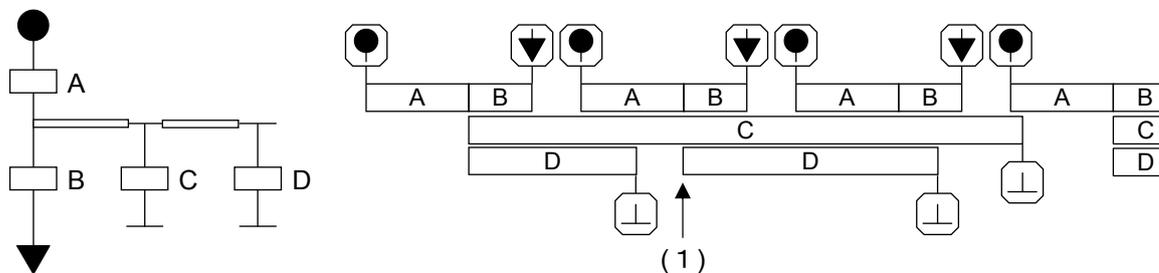


- (1) 非同期プロセスエンド到達時に非同期ルートが動作中の場合、非同期ルートの処理を継続したままプロセスを終了します。
- (2) 非同期ルートのルートスタート到達前に非同期ルートが動作完了した場合、非同期ルートを起動します。
- (3) 非同期ルートのルートスタート到達時に非同期ルートが動作中の場合、動作中の非同期ルートの処理を継続します。

## 2. 非同期ルートが複数の場合

&lt;回路例&gt;

&lt;タイムチャート&gt;

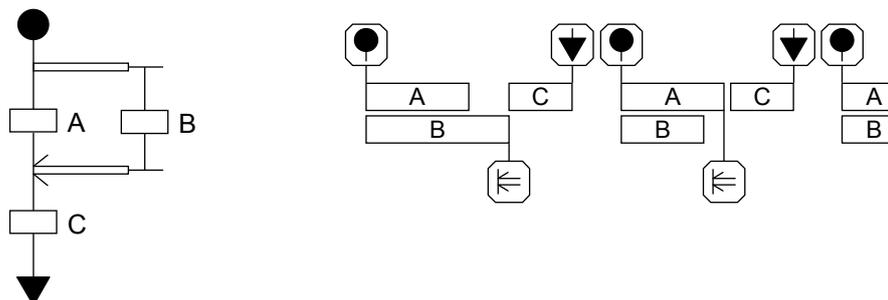


(1) 非同期ルートのルートスタート到達時、動作が完了した非同期ルートは起動され、動作中の非同期ルートは処理を継続します。

## 3. 非同期プロセスエンドでルートエンドが合流している場合

&lt;回路例&gt;

&lt;タイムチャート&gt;

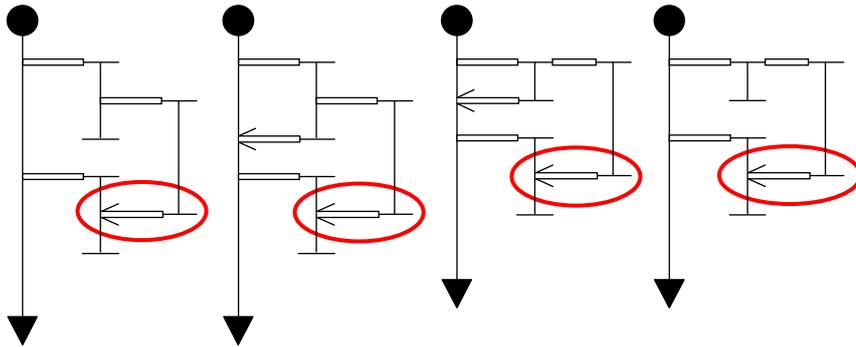


プロセスエンドを指定した場合と同じ動作となり、パラエンドでメインルートとサブルートの全処理が終了後、パラエンド以降のメインルートの処理を継続します。

## 4 構文説明

(注)

- (1) 下図のパターンのように、非同期プロセスで分岐と合流が別であり合流ルートが非同期ルートであるパターンはサポートしておりません。実行した場合、メインルートを新たに起動してもサブルートを実行中と判断して起動がかからない状態になりますので注意してください。



- (2) 非同期プロセスエンドを使用しているプロセスを、非同期プロセスエンドに対応していないHI-FLOWシステム（Ver-Rev 02-03以前）で受信した場合、プロセスの受信は成功しますが、受信したプロセスを表示すると非同期プロセスエンドが表示されません。このプロセスは非同期プロセスエンドをプロセスエンドに置き換えることで送信が可能となりますが、非同期ルートの動作が変わりますので注意してください。
- (3) 起動の際にマスターリセットを指定した場合、非同期ルートの実行中にビット型PI/Oがクリアされる可能性がありますので注意してください。

## 5 応用命令

## 5 応用命令

### 5.1 概 説

HI-FLOWの構文にサポートされている演算機能、データ処理機能は四則演算、論理演算、代入だけ（ワード長だけ）です。そこでHI-FLOWではラダー図と同様機能の応用命令をサポートしています。

### 5.2 使用方法

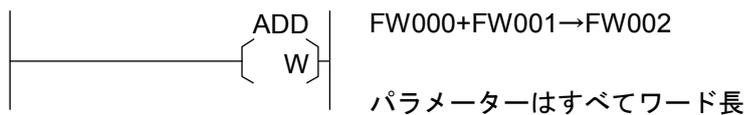
応用命令は次に示すような方法でプログラミングします。

○ 応用命令名称 パラメーター [, パラメーター] ~

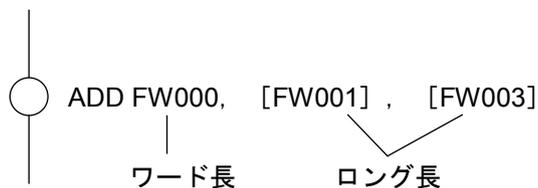
### 5.3 パラメーター

HI-FLOWの応用命令はラダーの演算ファンクションと異なり、各応用命令とそれに使用できるパラメーターの型が一致しなくてもかまいません。

(ラダー)



(HI-FLOW)



例えば FW000

1	0001	→ 応用命令実行後 →	0001
2	0000		0000
3	0002		0002
4	1111		0000

1	0001
2	0000
3	0002
4	0003

パラメーターは通常ソース、デスティネーション、リザルトの3種類があります。それぞれS, D, Rで表します。

パラメーターはビット型PI/O、ワード型PI/O、定数の3種類があります。

また、HI-FLOWの応用命令ではパラメーターに対してアドレッシングモードを指定できます。アドレッシングモードは以下に示す4種類があります。

1. ダイレクトワード長指定・・・パラメーターそのままの記述です。
2. ダイレクトロング長指定・・・ [ ] (大カッコ) でパラメーターを囲みます。
3. インダイレクトワード長指定・・・1. の記述の前に@を付けます。
4. インダイレクトロング長指定・・・2. の記述の前に@を付けます。

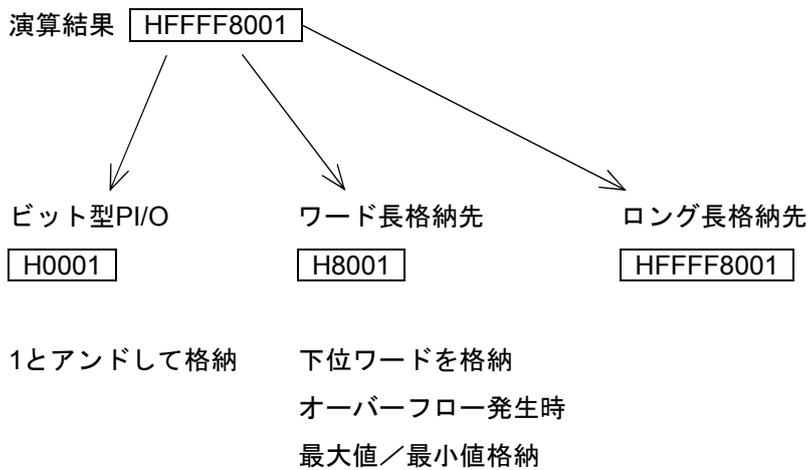
アドレッシングモード	パラメーター																
	ビット型PI/O	ワード型PI/O	定数														
	X000 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>データ1</td></tr><tr><td>データ2</td></tr></table> X001 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>データ2</td></tr></table> データ1,2 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>データa</td></tr><tr><td>データb</td></tr></table>	データ1	データ2	データ2	データa	データb	FW000 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>データ3</td></tr><tr><td>データ4</td></tr></table> FW001 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>データ4</td></tr></table> データ3,4 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>データc</td></tr><tr><td>データd</td></tr></table>	データ3	データ4	データ4	データc	データd	XXXX YYYYYYYY XXXX <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>データe</td></tr><tr><td>データf</td></tr></table> YYYYYYYY <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>データg</td></tr><tr><td>データh</td></tr></table>	データe	データf	データg	データh
データ1																	
データ2																	
データ2																	
データa																	
データb																	
データ3																	
データ4																	
データ4																	
データc																	
データd																	
データe																	
データf																	
データg																	
データh																	
1. ダイレクトワード長	データ1と1の アンド結果	データ3	XXXX ただし ロング長YYYYYYYYは下 位ワードだけ														
例	X000	FW000	1230 H20000000														
2. ダイレクトロング長	データ1,2と1の アンド結果	データ3,4	XXXX, YYYYYYYY ただし、XXXXはロング長 として扱います。														
例	[X000]	[FW000]	[H1234] [H20000000]														
3. インダイレクトワード長	パラメーターエラー	データc ただしデータ3,4が奇数 の場合エラー	XXXXの場合 データe YYYYYYYYの場合 データg XXXX, YYYYYYYYが 奇数の場合エラー														
例		@FW000	@HFFF0 @H180000														
4. インダイレクトロング長	パラメーターエラー	データc, d ただしデータ3,4が奇数 の場合エラー	XXXXの場合 データe, f YYYYYYYYの場合 データg, h XXXX, YYYYYYYYが 奇数の場合エラー														
例		@[FW000]	@[HFFF0] @[H180000]														

## 5.4 演算時の型変換

演算をするためパラメータ値を取り込む際、すべてロング長に符号拡張します。

FW000 8001 ————— 演算中はHFFFF8001と扱われます。

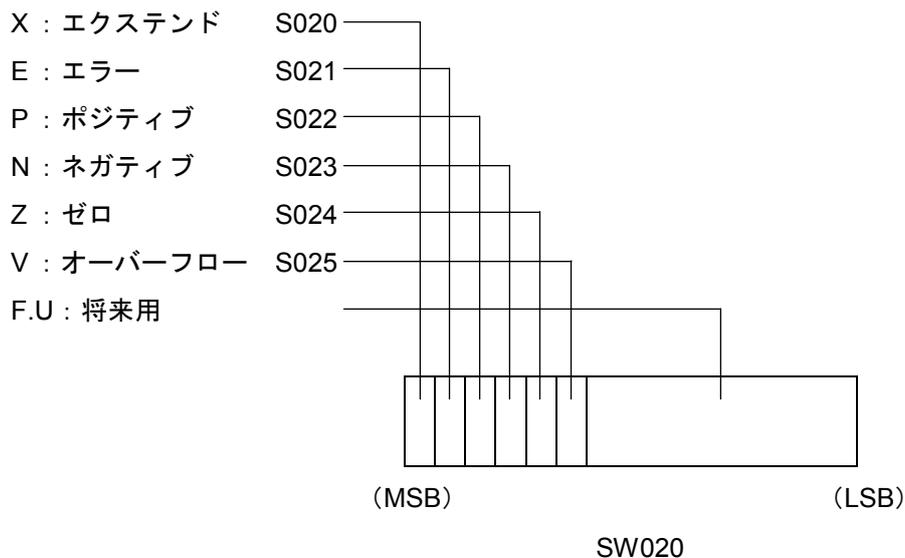
演算結果を格納する際は格納先にあわせて結果を型変換します。



## 5.5 システムエラーフラグ

HI-FLOW応用命令の実行結果に従い各種のフラグがSW020に設定されます。

### フラグ種類



各フラグは応用命令ごとのフラグの設定条件によってそれぞれ設定されます。しかし、次に示す条件が成立すると全応用命令に共通で下記フラグが設定されます。

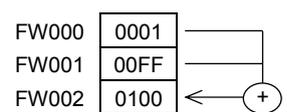
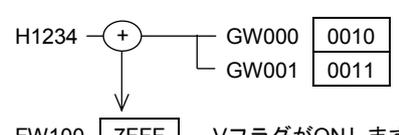
- ・エラーフラグ・・・使用している応用命令のパラメーター数が異なる場合。  
CPUにメモリープロテクトがかかっているとき、リザルト (R) で指定したアドレス、PI/Oがプロテクトエリア内を指していたとき。  
指定されたPI/Oに異常があるとき（使用できないなど）。
- ・オーバーフローフラグ・・・演算結果がリザルト (R) で指定された範囲（ワード、ロング）を超える数値のとき。演算結果には各サイズの限界値が設定されます。  

ワード長	正のオーバーフロー/7FFF
	負のオーバーフロー/8000
ロング長	正のオーバーフロー/7FFFFFFF
	負のオーバーフロー/80000000

## 5 応用命令

### 5.6 機能説明

この節では各応用命令の詳細について述べます。以下に示す形式で説明します。

応用命令名称	機能名称																					
ADD	加算																					
機能説明	ソースとデスティネーションの内容を加算してリザルトに格納します。																					
パラメーターと処理内容	 <p>○ ADD S, D, R</p> <p>S : ソース D : デスティネーション R : リザルト</p>	<p>S+D → R</p>																				
フラグの設定	E, Vが変化、他は保持																					
備考																						
使用例	<p>○ ADD FW000, FW001, FW002</p>  <p>FW000 0001 FW001 00FF FW002 0100</p> <p>○ ADD H1234, [GW000], FW100</p>  <p>H1234 + GW000 0010 GW001 0011 FW100 7FFF VフラグがONします。</p>																					
有効パラメーター	<table border="1"> <thead> <tr> <th>S, D, R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>		S, D, R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D, R	ビット型 PI/O	ワード型 PI/O	定数																			
ダイレクト ワード長	○	○	○																			
ダイレクト ロング長	○	○	○																			
インダイレクト ワード長	×	△	△																			
インダイレクト ロング長	×	△	△																			
△はアドレス値が奇数時はパラメーターエラー																						

応用命令の処理概要を説明します。

処理内容を図示します。

パラメーターの並びを示します。

命令後変更するフラグを示します。

注意点を示します。

主な使用例を表します。

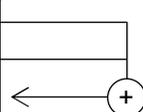
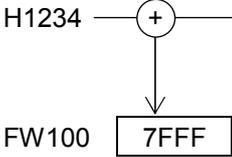
S (ソース)、D (デスティネーション)、R (リザルト) に指定有効なパラメーター種類を示します。

指定不可

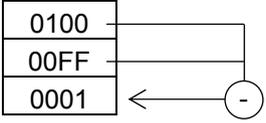
条件付き指定有効

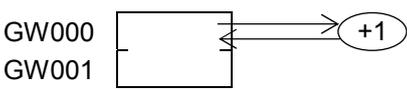
指定有効

対象のパラメーター種類 (ソース、デスティネーション、リザルト)

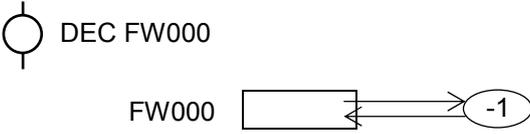
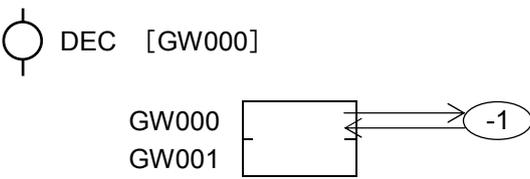
ADD	加算																																								
機能説明	ソースとデスティネーションの内容を加算してリザルトに格納します。																																								
パラメーターと処理内容	 ADD S, D, R <div style="border: 1px solid black; padding: 5px; display: inline-block; margin-left: 20px;">S+D → R</div> <p>S : ソース D : デスティネーション R : リザルト</p>																																								
フラグの設定	E, Vが変化、他はOFF																																								
備考																																									
使用例	 ADD FW000, FW001, FW002 <div style="margin-left: 20px;"> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>FW000</td><td>0001</td></tr> <tr><td>FW001</td><td>00FF</td></tr> <tr><td>FW002</td><td>0100</td></tr> </table>  </div>  ADD H1234, [GW000], FW100 <div style="margin-left: 20px;"> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>H1234</td><td></td></tr> <tr><td>GW000</td><td>0010</td></tr> <tr><td>GW001</td><td>0011</td></tr> </table>  <p>FW100 <span style="border: 1px solid black; padding: 2px;">7FFF</span> VフラグがONします。</p> </div>	FW000	0001	FW001	00FF	FW002	0100	H1234		GW000	0010	GW001	0011																												
FW000	0001																																								
FW001	00FF																																								
FW002	0100																																								
H1234																																									
GW000	0010																																								
GW001	0011																																								
有効パラメーター	<table border="1" style="display: inline-table; vertical-align: top; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table; vertical-align: top;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	○	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	×																																						
ダイレクトロング長	○	○	×																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									

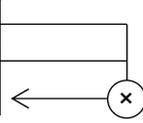
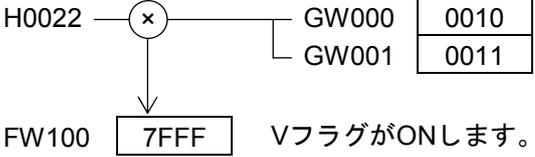
## 5 応用命令

SUB	減算																																									
機能説明	ソースからデスティネーションの内容を減算してリザルトに格納します。																																									
パラメーターと処理内容	 SUB S, D, R  S : ソース D : デスティネーション R : リザルト	S-D → R																																								
フラグの設定	E, Vが変化、他はOFF																																									
備考																																										
使用例	 SUB FW000, FW001, FW002  <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>FW000</td><td>0100</td></tr> <tr><td>FW001</td><td>00FF</td></tr> <tr><td>FW002</td><td>0001</td></tr> </table>    SUB H1234, [GW000], FW100  <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>H1234</td><td>-</td></tr> </table>  VフラグがONします。			FW000	0100	FW001	00FF	FW002	0001	H1234	-																															
FW000	0100																																									
FW001	00FF																																									
FW002	0001																																									
H1234	-																																									
有効パラメーター	<table border="1"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	<table border="1"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクトワード長	○	○	○																																							
ダイレクトロング長	○	○	○																																							
インダイレクトワード長	×	△	△																																							
インダイレクトロング長	×	△	△																																							
R	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクトワード長	○	○	×																																							
ダイレクトロング長	○	○	×																																							
インダイレクトワード長	×	△	△																																							
インダイレクトロング長	×	△	△																																							
△はアドレス値が奇数時はパラメーターエラー																																										

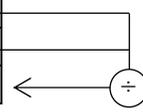
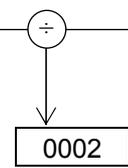
INC	+1 (インクリメント)																						
機能説明	ソースの内容に1を加算します。																						
パラメーターと処理内容	 INC S  S : ソース	$S+1 \rightarrow S$																					
フラグの設定	E, Vが変化、他はOFF																						
備考																							
使用例	 INC FW000     INC [GW000]    GW000, GW001をロング変数と見なしてインクリメントします。																						
有効パラメーター	<table border="1"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>			S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																				
ダイレクト ワード長	○	○	×																				
ダイレクト ロング長	○	○	×																				
インダイレクト ワード長	×	△	△																				
インダイレクト ロング長	×	△	△																				
△はアドレス値が奇数時はパラメーターエラー																							

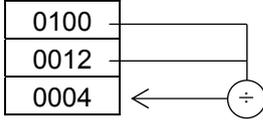
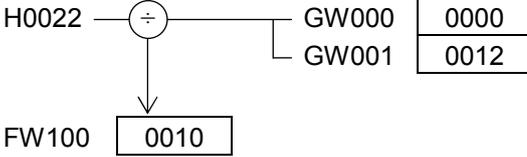
## 5 応用命令

DEC	-1 (デクリメント)																						
機能説明	ソースの内容から1を減算します。																						
パラメーターと処理内容	 DEC S  S : ソース	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> S-1 → S </div>																					
フラグの設定	E, Vが変化、他はOFF																						
備考																							
使用例	  GW000, GW001をロング変数と見なして デクリメントします。																						
有効 パラメーター	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 20%;">S</th> <th style="width: 15%;">ビット型 PI/O</th> <th style="width: 15%;">ワード型 PI/O</th> <th style="width: 50%;">定 数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>			S	ビット型 PI/O	ワード型 PI/O	定 数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定 数																				
ダイレクト ワード長	○	○	×																				
ダイレクト ロング長	○	○	×																				
インダイレクト ワード長	×	△	△																				
インダイレクト ロング長	×	△	△																				
△はアドレス 値が奇数時は パラメーター エラー																							

MUL	乗算																																								
機能説明	ソースとデスティネーションの内容を乗算してリザルトに格納します。																																								
パラメーターと処理内容	 MUL S, D, R <div style="border: 1px solid black; padding: 5px; display: inline-block; margin-left: 20px;">S×D → R</div> <p>S : ソース D : デスティネーション R : リザルト</p>																																								
フラグの設定	E, Vが変化、他はOFF																																								
備考																																									
使用例	 MUL FW000, FW001, FW002 <div style="margin-left: 20px;"> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>FW000</td><td>0100</td></tr> <tr><td>FW001</td><td>00FF</td></tr> <tr><td>FW002</td><td>FF00</td></tr> </table>  </div>  MUL H22, [GW000], FW100 <div style="margin-left: 20px;"> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>H0022</td><td>×</td></tr> </table>  <p>VフラグがONします。</p> </div>	FW000	0100	FW001	00FF	FW002	FF00	H0022	×																																
FW000	0100																																								
FW001	00FF																																								
FW002	FF00																																								
H0022	×																																								
有効パラメーター	<table border="1" style="display: inline-table; vertical-align: top; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table; vertical-align: top;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	○	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	×																																						
ダイレクトロング長	○	○	×																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									

## 5 応用命令

DIV	除算																																								
機能説明	ソースをデスティネーションの内容で除算して商をリザルトに格納します。																																								
パラメーターと処理内容	<div style="display: flex; align-items: center;"> <div style="margin-right: 20px;">  <p>DIV S, D, R</p> <p>S : ソース D : デスティネーション R : リザルト</p> </div> <div style="border: 1px solid black; padding: 5px;"> <math>S \div D \rightarrow R</math> </div> </div>																																								
フラグの設定	E, Vが変化、他はOFF																																								
備考	D=0のとき、EフラグをONして何もしません。																																								
使用例	<div style="margin-bottom: 20px;">  <p>DIV FW000, FW001, FW002</p> <div style="display: flex; align-items: center; margin-top: 10px;"> <table border="1" style="margin-right: 10px;"> <tr><td>FW000</td><td>0100</td></tr> <tr><td>FW001</td><td>0010</td></tr> <tr><td>FW002</td><td>0010</td></tr> </table>  </div> </div> <div>  <p>DIV H22, [GW000], FW100</p> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 20px;"> <p>H0022</p>  </div> <div style="margin-right: 20px;"> <table border="1"> <tr><td>GW000</td><td>0000</td></tr> <tr><td>GW001</td><td>0011</td></tr> </table> </div> </div> </div>	FW000	0100	FW001	0010	FW002	0010	GW000	0000	GW001	0011																														
FW000	0100																																								
FW001	0010																																								
FW002	0010																																								
GW000	0000																																								
GW001	0011																																								
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	○	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	×																																						
ダイレクトロング長	○	○	×																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									

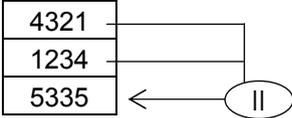
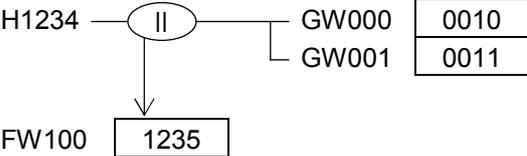
MOD	剰余																																											
機能説明	ソースをデスティネーションの内容で除算して剰余をリザルトに格納します。																																											
パラメーターと処理内容	 MOD S, D, R  S : ソース D : デスティネーション R : リザルト	$S \div D$ の剰余 $\rightarrow R$																																										
フラグの設定	E, Vが変化、他はOFF																																											
備考	D=0のとき、EフラグをONして何もしません。オーバーフロー時、R=0。																																											
使用例	 MOD FW000, FW001, FW002  <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>FW000</td><td>0100</td></tr> <tr><td>FW001</td><td>0012</td></tr> <tr><td>FW002</td><td>0004</td></tr> </table>    MOD H22, [GW000], FW100  <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>H0022</td><td></td></tr> </table> 				FW000	0100	FW001	0012	FW002	0004	H0022																																	
FW000	0100																																											
FW001	0012																																											
FW002	0004																																											
H0022																																												
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクトワード長	○	○	○																																									
ダイレクトロング長	○	○	○																																									
インダイレクトワード長	×	△	△																																									
インダイレクトロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクトワード長	○	○	×																																									
ダイレクトロング長	○	○	×																																									
インダイレクトワード長	×	△	△																																									
インダイレクトロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												

## 5 応用命令

SCL	スケール変換																																								
機能説明	ソースをデスティネーションの内容でスケール変換してリザルトに格納します。																																								
パラメーターと処理内容	<div style="display: flex; align-items: center;"> <div style="margin-right: 20px;"> <math>\bigcirc</math> SCL S, D1, D2, R         </div> <div style="border: 1px solid black; padding: 5px;"> <math>S \times D1 \div D2 \rightarrow R</math> </div> </div> <p>S : ソース D1 : デスティネーション1 D2 : デスティネーション2 R : リザルト</p>																																								
フラグの設定	E, Vが変化、他はOFF																																								
備考	乗算オーバーフロー発生時リザルトにオーバーフロー値を書き込み終了します。 D2=0のとき、EフラグをONして何もしません。オーバーフロー時、R=0。																																								
使用例	<div style="margin-bottom: 20px;"> <math>\bigcirc</math> SCL FW000, FW001, FW002, FW003           <table style="margin-left: 20px; border-collapse: collapse;"> <tr> <td style="padding-right: 10px;">FW000</td> <td style="border: 1px solid black; padding: 2px 10px;">3320</td> <td rowspan="4" style="border: none; padding: 0 10px;">┌───┐ ├───┤ ├───┤ └───┘</td> <td rowspan="4" style="border: none; padding: 0 10px;">┌───┐ ├───┤ ├───┤ └───┘</td> </tr> <tr> <td>FW001</td> <td style="border: 1px solid black; padding: 2px 10px;">0010</td> </tr> <tr> <td>FW002</td> <td style="border: 1px solid black; padding: 2px 10px;">0066</td> </tr> <tr> <td>FW003</td> <td style="border: 1px solid black; padding: 2px 10px;">0805</td> </tr> </table> <div style="margin-left: 100px; border: 1px solid black; border-radius: 15px; padding: 2px 10px; display: inline-block;">/3320×/10÷/66</div> </div> <div> <math>\bigcirc</math> SCL GW000, GW001, H1110, FW100           <table style="margin-left: 20px; border-collapse: collapse;"> <tr> <td style="padding-right: 10px;">GW000</td> <td style="border: 1px solid black; padding: 2px 10px;">2222</td> <td rowspan="2" style="border: none; padding: 0 10px;">┌───┐ ├───┤</td> </tr> <tr> <td>GW001</td> <td style="border: 1px solid black; padding: 2px 10px;">0012</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td style="padding: 0 10px;">H1110</td> <td></td> </tr> <tr> <td>FW100</td> <td style="border: 1px solid black; padding: 2px 10px;">0024</td> <td colspan="4" style="border: none; padding: 0 10px;">└───┘</td> <td style="border: 1px solid black; border-radius: 15px; padding: 2px 10px;">/2222×/12÷/1110</td> </tr> </table> </div>	FW000	3320	┌───┐ ├───┤ ├───┤ └───┘	┌───┐ ├───┤ ├───┤ └───┘	FW001	0010	FW002	0066	FW003	0805	GW000	2222	┌───┐ ├───┤	┌───┐ ├───┤	┌───┐ ├───┤	┌───┐ ├───┤	GW001	0012					H1110		FW100	0024	└───┘				/2222×/12÷/1110									
FW000	3320	┌───┐ ├───┤ ├───┤ └───┘	┌───┐ ├───┤ ├───┤ └───┘																																						
FW001	0010																																								
FW002	0066																																								
FW003	0805																																								
GW000	2222	┌───┐ ├───┤	┌───┐ ├───┤	┌───┐ ├───┤	┌───┐ ├───┤																																				
GW001	0012																																								
				H1110																																					
FW100	0024	└───┘				/2222×/12÷/1110																																			
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, D1, D2</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>ダイレクトロング長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>インダイレクトワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>ダイレクトロング長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>インダイレクトワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table>	S, D1, D2	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D1, D2	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	○	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	×																																						
ダイレクトロング長	○	○	×																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									

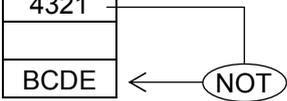
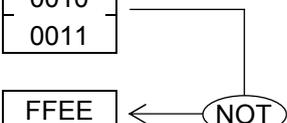
AND	論理積																																								
機能説明	ソースとデスティネーションの内容の論理積をリザルトに格納します。																																								
パラメーターと処理内容	 AND S, D, R <div style="border: 1px solid black; padding: 5px; display: inline-block; margin-left: 400px;">S &amp;&amp; D → R</div>  S : ソース D : デスティネーション R : リザルト																																								
フラグの設定	Eが変化、他はOFF																																								
備考	Rがワード長の場合は演算結果の下位ワードが書き込まれます。																																								
使用例	 AND FW000, FW001, FW002  <table style="border-collapse: collapse; margin-left: 100px;"> <tr><td style="border: none;">FW000</td><td style="border: 1px solid black; padding: 2px;">0001</td><td style="border: none;">└──┘</td></tr> <tr><td style="border: none;">FW001</td><td style="border: 1px solid black; padding: 2px;">00FF</td><td style="border: none;">└──┘</td></tr> <tr><td style="border: none;">FW002</td><td style="border: 1px solid black; padding: 2px;">0001</td><td style="border: none;">←──┘</td></tr> </table> <div style="margin-left: 400px; border: 1px solid black; border-radius: 50%; padding: 2px 5px;">&amp;&amp;</div>   AND H1234, [GW000], FW100  <table style="border-collapse: collapse; margin-left: 100px;"> <tr><td style="border: none;">H1234</td><td style="border: none;">──┘</td><td style="border: none;">└──┘</td><td style="border: none;">GW000</td><td style="border: 1px solid black; padding: 2px;">0010</td></tr> <tr><td style="border: none;"></td><td style="border: none;">└──┘</td><td style="border: none;">└──┘</td><td style="border: none;">GW001</td><td style="border: 1px solid black; padding: 2px;">0011</td></tr> </table> <div style="margin-left: 100px; border: 1px solid black; border-radius: 50%; padding: 2px 5px;">&amp;&amp;</div> <div style="margin-left: 100px; border: 1px solid black; padding: 2px;">FW100 0010</div>	FW000	0001	└──┘	FW001	00FF	└──┘	FW002	0001	←──┘	H1234	──┘	└──┘	GW000	0010		└──┘	└──┘	GW001	0011																					
FW000	0001	└──┘																																							
FW001	00FF	└──┘																																							
FW002	0001	←──┘																																							
H1234	──┘	└──┘	GW000	0010																																					
	└──┘	└──┘	GW001	0011																																					
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>ダイレクトロング長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>インダイレクトワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>ダイレクトロング長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>インダイレクトワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	○	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	×																																						
ダイレクトロング長	○	○	×																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									

## 5 応用命令

OR	論理和																																								
機能説明	ソースとデスティネーションの内容の論理和をリザルトに格納します。																																								
パラメーターと処理内容	 OR S, D, R <div style="border: 1px solid black; padding: 5px; display: inline-block; margin-left: 20px;">S    D → R</div> <p>S : ソース D : デスティネーション R : リザルト</p>																																								
フラグの設定	Eが変化、他はOFF																																								
備考	Rがワード長の場合は演算結果の下位ワードが書き込まれます。																																								
使用例	 OR FW000, FW001, FW002 <div style="margin-top: 10px;"> <table border="1" style="display: inline-table; border-collapse: collapse;"> <tr><td>FW000</td><td>4321</td></tr> <tr><td>FW001</td><td>1234</td></tr> <tr><td>FW002</td><td>5335</td></tr> </table>  </div>  OR H1234, [GW000], FW100 <div style="margin-top: 10px;">  </div>	FW000	4321	FW001	1234	FW002	5335																																		
FW000	4321																																								
FW001	1234																																								
FW002	5335																																								
有効パラメーター	<table border="1" style="display: inline-table; border-collapse: collapse; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table; border-collapse: collapse;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	○	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	×																																						
ダイレクトロング長	○	○	×																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									

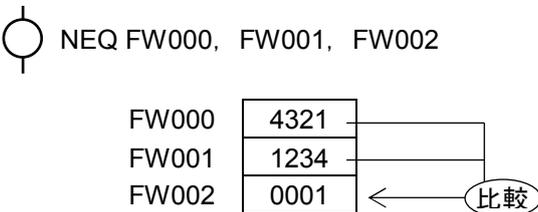
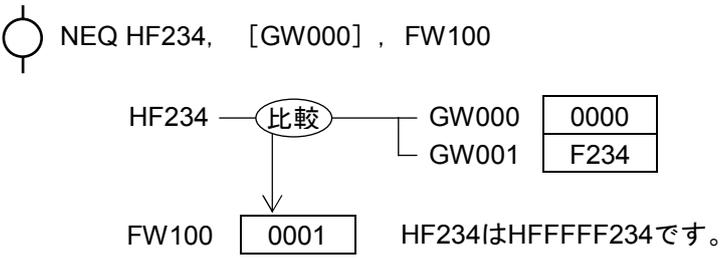
EOR	排他的論理和																																											
機能説明	ソースとデスティネーションの内容の排他的論理和をリザルトに格納します。																																											
パラメーターと処理内容	 EOR S, D, R		$S \wedge D \rightarrow R$																																									
	S : ソース D : デスティネーション R : リザルト																																											
フラグの設定	Eが変化、他はOFF																																											
備考	Rがワード長の場合は演算結果の下位ワードが書き込まれます。																																											
使用例	 EOR FW000, FW001, FW002 <table border="1" style="margin-left: 40px;"> <tr><td>FW000</td><td>4321</td></tr> <tr><td>FW001</td><td>1234</td></tr> <tr><td>FW002</td><td>5115</td></tr> </table>  EOR H1234, [GW000], FW100 <table border="1" style="margin-left: 40px;"> <tr><td>H1234</td><td>^^</td></tr> <tr><td>FW100</td><td>1225</td></tr> </table> <table border="1" style="margin-left: 100px;"> <tr><td>GW000</td><td>0010</td></tr> <tr><td>GW001</td><td>0011</td></tr> </table>				FW000	4321	FW001	1234	FW002	5115	H1234	^^	FW100	1225	GW000	0010	GW001	0011																										
FW000	4321																																											
FW001	1234																																											
FW002	5115																																											
H1234	^^																																											
FW100	1225																																											
GW000	0010																																											
GW001	0011																																											
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクトワード長	○	○	○																																									
ダイレクトロング長	○	○	○																																									
インダイレクトワード長	×	△	△																																									
インダイレクトロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクトワード長	○	○	×																																									
ダイレクトロング長	○	○	×																																									
インダイレクトワード長	×	△	△																																									
インダイレクトロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												

## 5 応用命令

NOT	否定																																								
機能説明	ソースの内容の否定（ビット反転）をリザルトに格納します。																																								
パラメーターと処理内容	<p>  NOT S, R         </p> <div style="border: 1px solid black; padding: 5px; display: inline-block; margin-left: 400px;">           S（ビット反転） → R         </div> <p>           S : ソース            R : リザルト         </p>																																								
フラグの設定	Eが変化、他はOFF																																								
備考																																									
使用例	<p>  NOT FW000, FW002         </p> <div style="margin-left: 100px;"> <table border="1" style="border-collapse: collapse;"> <tr><td style="padding: 2px;">FW000</td><td style="padding: 2px;">4321</td></tr> <tr><td style="padding: 2px;">FW002</td><td style="padding: 2px;">BCDE</td></tr> </table>  </div> <p>  NOT [GW000], FW100         </p> <div style="margin-left: 100px;"> <table border="1" style="border-collapse: collapse;"> <tr><td style="padding: 2px;">GW000</td><td style="padding: 2px;">0010</td></tr> <tr><td style="padding: 2px;">GW001</td><td style="padding: 2px;">0011</td></tr> <tr><td style="padding: 2px;">GW100</td><td style="padding: 2px;">FFEE</td></tr> </table>  </div>	FW000	4321	FW002	BCDE	GW000	0010	GW001	0011	GW100	FFEE																														
FW000	4321																																								
FW002	BCDE																																								
GW000	0010																																								
GW001	0011																																								
GW100	FFEE																																								
有効パラメーター	<table border="1" style="border-collapse: collapse; width: 50%; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr><td>ダイレクトワード長</td><td style="text-align: center;">○</td><td style="text-align: center;">○</td><td style="text-align: center;">○</td></tr> <tr><td>ダイレクトロング長</td><td style="text-align: center;">○</td><td style="text-align: center;">○</td><td style="text-align: center;">○</td></tr> <tr><td>インダイレクトワード長</td><td style="text-align: center;">×</td><td style="text-align: center;">△</td><td style="text-align: center;">△</td></tr> <tr><td>インダイレクトロング長</td><td style="text-align: center;">×</td><td style="text-align: center;">△</td><td style="text-align: center;">△</td></tr> </tbody> </table> <table border="1" style="border-collapse: collapse; width: 50%;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr><td>ダイレクトワード長</td><td style="text-align: center;">○</td><td style="text-align: center;">○</td><td style="text-align: center;">×</td></tr> <tr><td>ダイレクトロング長</td><td style="text-align: center;">○</td><td style="text-align: center;">○</td><td style="text-align: center;">×</td></tr> <tr><td>インダイレクトワード長</td><td style="text-align: center;">×</td><td style="text-align: center;">△</td><td style="text-align: center;">△</td></tr> <tr><td>インダイレクトロング長</td><td style="text-align: center;">×</td><td style="text-align: center;">△</td><td style="text-align: center;">△</td></tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	○	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	×																																						
ダイレクトロング長	○	○	×																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									

EQU	等しいか比較																																								
機能説明	ソースとデスティネーションの内容を比較し、等しければ1、それ以外は0をリザルトに格納します。																																								
パラメーターと処理内容	<p>○ EQU S, D, R</p> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> <p>S = D 1 → R</p> <p>S ≠ D 0 → R</p> </div> <p>S : ソース D : デスティネーション R : リザルト</p>																																								
フラグの設定	Eが変化、他はOFF																																								
備考	ワード長データはロング長に符号拡張して比較します。																																								
使用例	<p>○ EQU FW000, FW001, FW002</p> <div style="display: flex; align-items: center;"> <table border="1" style="margin-right: 20px;"> <tr><td>FW000</td><td>4321</td></tr> <tr><td>FW001</td><td>1234</td></tr> <tr><td>FW002</td><td>0000</td></tr> </table> </div> <p>○ EQU HF234, [GW000], FW100</p> <div style="display: flex; align-items: center;"> <table border="1" style="margin-right: 20px;"> <tr><td>HF234</td><td>比較</td></tr> </table> <table border="1" style="margin-right: 20px;"> <tr><td>GW000</td><td>0000</td></tr> <tr><td>GW001</td><td>F234</td></tr> </table> </div> <p>FW100 0000      HF234はHFFFFFF234です。</p>	FW000	4321	FW001	1234	FW002	0000	HF234	比較	GW000	0000	GW001	F234																												
FW000	4321																																								
FW001	1234																																								
FW002	0000																																								
HF234	比較																																								
GW000	0000																																								
GW001	F234																																								
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクト ワード長	○	○	○																																						
ダイレクト ロング長	○	○	○																																						
インダイレクト ワード長	×	△	△																																						
インダイレクト ロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクト ワード長	○	○	×																																						
ダイレクト ロング長	○	○	×																																						
インダイレクト ワード長	×	△	△																																						
インダイレクト ロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									

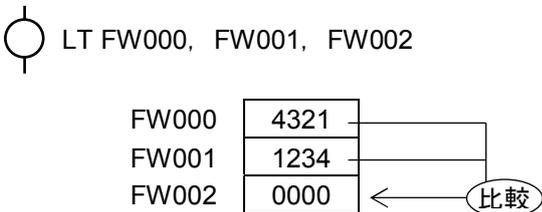
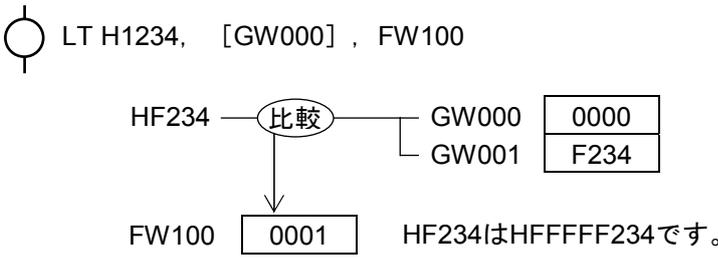
## 5 応用命令

NEQ	等しくないか比較																																									
機能説明	ソースとデスティネーションの内容を比較し、等しくなければ1、それ以外は0をリザルトに格納します。																																									
パラメーターと処理内容	 NEQ S, D, R  S : ソース D : デスティネーション R : リザルト	$S \neq D \quad 1 \rightarrow R$ $S = D \quad 0 \rightarrow R$																																								
フラグの設定	Eが変化、他はOFF																																									
備考	ワード長データはロング長に符号拡張して比較します。																																									
使用例	 																																									
有効パラメーター	<table border="1"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	<table border="1"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	○																																							
ダイレクト ロング長	○	○	○																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
R	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	×																																							
ダイレクト ロング長	○	○	×																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
△はアドレス値が奇数時はパラメーターエラー																																										

GT	より大きい比較																																									
機能説明	ソースとデスティネーションの内容を比較し、ソースが大きければ1、それ以外は0をリザルトに格納します。																																									
パラメーターと処理内容	<p>○ GT S, D, R</p> <p>S : ソース D : デスティネーション R : リザルト</p>	<table border="1"> <tr> <td>S &gt; D</td> <td>1</td> <td>→ R</td> </tr> <tr> <td>S ≤ D</td> <td>0</td> <td>→ R</td> </tr> </table>	S > D	1	→ R	S ≤ D	0	→ R																																		
S > D	1	→ R																																								
S ≤ D	0	→ R																																								
フラグの設定	Eが変化、他はOFF																																									
備考	ワード長データはロング長に符号拡張して比較します。																																									
使用例	<p>○ GT FW000, FW001, FW002</p> <p>○ GT H1234, [GW000], FW100</p> <p>FW100 0000 HF234はHFFFFFF234です。</p>																																									
有効パラメーター	<table border="1"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	<table border="1"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	○																																							
ダイレクト ロング長	○	○	○																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
R	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	×																																							
ダイレクト ロング長	○	○	×																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
△はアドレス値が奇数時はパラメーターエラー																																										

## 5 応用命令

GE	より以上か比較																																																																					
機能説明	ソースとデスティネーションの内容を比較し、ソースが等しいか大きければ1、それ以外は0をリザルトに格納します。																																																																					
パラメーターと処理内容	<div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <p>○ GE S, D, R</p> <p>S : ソース D : デスティネーション R : リザルト</p> </div> <div style="width: 35%; border: 1px solid black; padding: 5px;"> <math>S \geq D \quad 1 \rightarrow R</math>  <math>S &lt; D \quad 0 \rightarrow R</math> </div> </div>																																																																					
フラグの設定	Eが変化、他はOFF																																																																					
備考	ワード長データはロング長に符号拡張して比較します。																																																																					
使用例	<p>○ GE FW000, FW001, FW002</p> <div style="display: flex; align-items: center; margin: 10px 0;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>FW000</td><td>4321</td></tr> <tr><td>FW001</td><td>1234</td></tr> <tr><td>FW002</td><td>0001</td></tr> </table> <div style="margin-left: 10px;"> </div> </div> <p>○ GE H1234, [GW000], FW100</p> <div style="display: flex; align-items: center; margin: 10px 0;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>HF234</td></tr> </table> <div style="margin-left: 10px;"> </div> <table border="1" style="border-collapse: collapse; text-align: center; margin-left: 10px;"> <tr><td>GW000</td><td>0000</td></tr> <tr><td>GW001</td><td>F234</td></tr> </table> </div> <p>FW100 <span style="border: 1px solid black; padding: 2px 5px;">0000</span>    HF234はFFFFFF234です。</p>	FW000	4321	FW001	1234	FW002	0001	HF234	GW000	0000	GW001	F234																																																										
FW000	4321																																																																					
FW001	1234																																																																					
FW002	0001																																																																					
HF234																																																																						
GW000	0000																																																																					
GW001	F234																																																																					
有効パラメーター	<table border="1" style="border-collapse: collapse; width: 100%; text-align: center;"> <thead> <tr> <th rowspan="2">S, D</th> <th colspan="2">ビット型</th> <th colspan="2">ワード型</th> <th rowspan="2">定数</th> </tr> <tr> <th>PI/O</th> <th>PI/O</th> <th>PI/O</th> <th>PI/O</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td></td> <td>△</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td></td> <td>△</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="border-collapse: collapse; width: 100%; text-align: center; margin-top: 10px;"> <thead> <tr> <th>R</th> <th colspan="2">ビット型</th> <th colspan="2">ワード型</th> <th rowspan="2">定数</th> </tr> <tr> <th></th> <th>PI/O</th> <th>PI/O</th> <th>PI/O</th> <th>PI/O</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td></td> <td>△</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td></td> <td>△</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型		ワード型		定数	PI/O	PI/O	PI/O	PI/O	ダイレクトワード長	○	○	○	○	○	ダイレクトロング長	○	○	○	○	○	インダイレクトワード長	×		△	△	△	インダイレクトロング長	×		△	△	△	R	ビット型		ワード型		定数		PI/O	PI/O	PI/O	PI/O	ダイレクトワード長	○	○	○	○	×	ダイレクトロング長	○	○	○	○	×	インダイレクトワード長	×		△	△	△	インダイレクトロング長	×		△	△	△
S, D	ビット型		ワード型		定数																																																																	
	PI/O	PI/O	PI/O	PI/O																																																																		
ダイレクトワード長	○	○	○	○	○																																																																	
ダイレクトロング長	○	○	○	○	○																																																																	
インダイレクトワード長	×		△	△	△																																																																	
インダイレクトロング長	×		△	△	△																																																																	
R	ビット型		ワード型		定数																																																																	
	PI/O	PI/O	PI/O	PI/O																																																																		
ダイレクトワード長	○	○	○	○	×																																																																	
ダイレクトロング長	○	○	○	○	×																																																																	
インダイレクトワード長	×		△	△	△																																																																	
インダイレクトロング長	×		△	△	△																																																																	
△はアドレス値が奇数時はパラメーターエラー																																																																						

LT	より小さいか比較																																										
機能説明	ソースとデスティネーションの内容を比較し、ソースが小さければ1、それ以外は0をリザルトに格納します。																																										
パラメーターと処理内容	 LT S, D, R S : ソース D : デスティネーション R : リザルト	$S < D \quad 1 \rightarrow R$ $S \geq D \quad 0 \rightarrow R$																																									
フラグの設定	Eが変化、他はOFF																																										
備考	ワード長データはロング長に符号拡張して比較します。																																										
使用例	 																																										
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>			S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクト ワード長	○	○	○																																								
ダイレクト ロング長	○	○	○																																								
インダイレクト ワード長	×	△	△																																								
インダイレクト ロング長	×	△	△																																								
R	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクト ワード長	○	○	×																																								
ダイレクト ロング長	○	○	×																																								
インダイレクト ワード長	×	△	△																																								
インダイレクト ロング長	×	△	△																																								
△はアドレス値が奇数時はパラメーターエラー																																											

## 5 応用命令

LE	より以下か比較																																								
機能説明	ソースとデスティネーションの内容を比較し、ソースが等しいか小さければ1、それ以外は0をリザルトに格納します。																																								
パラメーターと処理内容	<div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <p>○ LE S, D, R</p> <p>S : ソース D : デスティネーション R : リザルト</p> </div> <div style="width: 35%; border: 1px solid black; padding: 5px;"> <p><math>S \leq D</math>    1 → R</p> <p><math>S &gt; D</math>    0 → R</p> </div> </div>																																								
フラグの設定	Eが変化、他はOFF																																								
備考	ワード長データはロング長に符号拡張して比較します。																																								
使用例	<p>○ LE FW000, FW001, FW002</p> <div style="display: flex; align-items: center; margin-top: 10px;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>FW000</td><td>4321</td></tr> <tr><td>FW001</td><td>1234</td></tr> <tr><td>FW002</td><td>0000</td></tr> </table> <div style="margin-left: 10px;"> <p>→</p> <p>← (比較)</p> </div> </div> <p>○ LE H1234, [GW000], FW100</p> <div style="display: flex; align-items: center; margin-top: 10px;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>HF234</td></tr> </table> <div style="margin-left: 10px;"> <p>(比較)</p> <p>↓</p> </div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>FW100</td><td>0001</td></tr> </table> <div style="margin-left: 20px;"> <p>GW000 0000</p> <p>GW001 F234</p> </div> </div> <p>HF234はFFFFFF234です。</p>	FW000	4321	FW001	1234	FW002	0000	HF234	FW100	0001																															
FW000	4321																																								
FW001	1234																																								
FW002	0000																																								
HF234																																									
FW100	0001																																								
有効パラメーター	<table border="1" style="border-collapse: collapse; text-align: center; width: 50%;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr><td>ダイレクトワード長</td><td>○</td><td>○</td><td>○</td></tr> <tr><td>ダイレクトロング長</td><td>○</td><td>○</td><td>○</td></tr> <tr><td>インダイレクトワード長</td><td>×</td><td>△</td><td>△</td></tr> <tr><td>インダイレクトロング長</td><td>×</td><td>△</td><td>△</td></tr> </tbody> </table> <table border="1" style="border-collapse: collapse; text-align: center; width: 50%; margin-left: 20px;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr><td>ダイレクトワード長</td><td>○</td><td>○</td><td>×</td></tr> <tr><td>ダイレクトロング長</td><td>○</td><td>○</td><td>×</td></tr> <tr><td>インダイレクトワード長</td><td>×</td><td>△</td><td>△</td></tr> <tr><td>インダイレクトロング長</td><td>×</td><td>△</td><td>△</td></tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	○	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	×																																						
ダイレクトロング長	○	○	×																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									

TST	符号テスト																				
機能説明	ソースの内容をテストし、P, Z, Nのフラグを設定します。																				
パラメーターと処理内容	 TST S S : ソース <div style="border: 1px solid black; padding: 5px; margin-left: 400px;"> S &gt; 0 : P=1, Z=0, N=0  S = 0 : P=0, Z=1, N=0  S &lt; 0 : P=0, Z=0, N=1 </div>																				
フラグの設定	E, P, Z, Nが変化、他はOFF																				
備考	ワード長データはロング長に符号拡張してテストします。																				
使用例	 TST FW000 <div style="margin-left: 100px;"> FW000 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 40px;">4321</td></tr><tr><td> </td></tr><tr><td> </td></tr></table> <span style="margin-left: 20px;">←</span> <span style="border: 1px solid black; border-radius: 50%; padding: 2px 5px;">テスト</span> </div>  TST [GW000] <div style="margin-left: 100px;"> GW000 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 40px;">FFFF</td></tr><tr><td>F234</td></tr></table> <span style="margin-left: 20px;">←</span> <span style="border: 1px solid black; border-radius: 50%; padding: 2px 5px;">テスト</span> </div> GW020 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td style="width: 40px;">1000</td></tr></table> <span style="margin-left: 20px;">←</span> <span style="border: 1px solid black; border-radius: 50%; padding: 2px 5px;">テスト</span>	4321			FFFF	F234	1000														
4321																					
FFFF																					
F234																					
1000																					
有効パラメーター	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 15%;">S</th> <th style="width: 15%;">ビット型 PI/O</th> <th style="width: 15%;">ワード型 PI/O</th> <th style="width: 15%;">定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																		
ダイレクト ワード長	○	○	○																		
ダイレクト ロング長	○	○	○																		
インダイレクト ワード長	×	△	△																		
インダイレクト ロング長	×	△	△																		
△はアドレス値が奇数時はパラメーターエラー																					

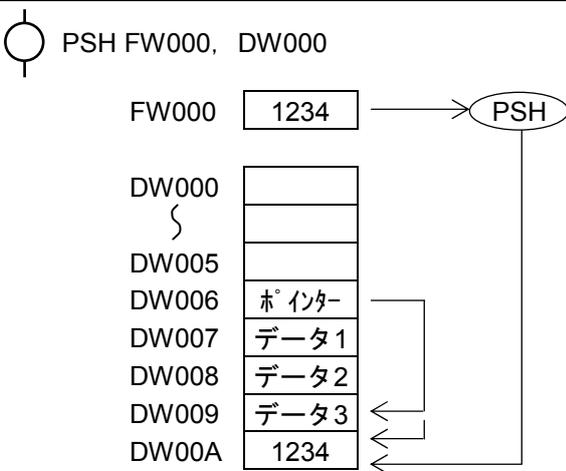
## 5 応用命令

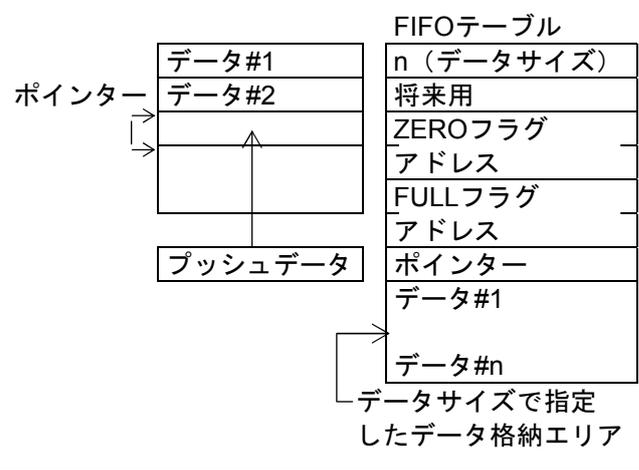
MOV	転送																																								
機能説明	ソースの内容をデスティネーションに転送します。																																								
パラメーターと処理内容	 MOV S, D <div style="border: 1px solid black; display: inline-block; padding: 2px; margin-left: 10px;">S → D</div> <p>S : ソース D : デスティネーション</p>																																								
フラグの設定	Eが変化、他はOFF																																								
備考	転送するサイズが異なる場合は型変換します。																																								
使用例	 MOV FW000, FW002 <div style="margin-left: 20px;"> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>FW000</td><td>4321</td></tr> <tr><td>FW002</td><td>4321</td></tr> </table>  </div>  MOV HF234, @ [H180000] <div style="margin-left: 20px;"> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>HF234</td><td>→</td><td>H180000</td><td><table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>FFFF</td></tr><tr><td>F234</td></tr></table></td></tr> <tr><td></td><td></td><td>2</td><td></td></tr> </table> <p style="text-align: center;">HF234はHFFFFFF234です。</p> </div>	FW000	4321	FW002	4321	HF234	→	H180000	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>FFFF</td></tr><tr><td>F234</td></tr></table>	FFFF	F234			2																											
FW000	4321																																								
FW002	4321																																								
HF234	→	H180000	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>FFFF</td></tr><tr><td>F234</td></tr></table>	FFFF	F234																																				
FFFF																																									
F234																																									
		2																																							
有効パラメーター	<table border="1" style="display: inline-table; vertical-align: top; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table; vertical-align: top;"> <thead> <tr> <th>D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	○	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
D	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	×																																						
ダイレクトロング長	○	○	×																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									

MOM	一括転送																																																
機能説明	ソースの内容をn要素分（ワード、ロング）一括でデスティネーションに転送します。																																																
パラメーターと処理内容	 MOM S, n, D S : ソース D : デスティネーション n : 転送要素数	$S_1 \rightarrow D_1$ $\vdots$ $S_n \rightarrow D_n$																																															
フラグの設定	E, Vが変化、他はOFF																																																
備考	n ≤ 0、n > 256のとき未処理。Sが定数の場合は定数値をDの型に変換して設定します。S, Dの型が異なった場合は型変換して設定します。																																																
使用例	 MOM FW000, 1, FW002 <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">FW000</div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="width: 40px;">4321</td></tr> <tr><td style="width: 40px;"> </td></tr> <tr><td style="width: 40px;"> </td></tr> </table> <div style="margin: 0 10px;">←</div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="width: 40px;">FW002</td></tr> <tr><td style="width: 40px;">4321</td></tr> </table> </div>  MOM FW000, 2, @ [H180000] <div style="display: flex; align-items: center; margin-top: 10px;"> <table border="1" style="border-collapse: collapse; text-align: center; margin-right: 20px;"> <tr><td style="width: 40px;">FW000</td><td style="width: 40px;">F234</td></tr> <tr><td style="width: 40px;">FW001</td><td style="width: 40px;">0001</td></tr> <tr><td style="width: 40px;">FW002</td><td style="width: 40px;">0000</td></tr> <tr><td style="width: 40px;">FW003</td><td style="width: 40px;">FFFF</td></tr> </table> <div style="margin: 0 10px;">→</div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="width: 40px;">H180000</td><td style="width: 40px;">FFFF</td></tr> <tr><td style="width: 40px;">2</td><td style="width: 40px;">F234</td></tr> <tr><td style="width: 40px;">4</td><td style="width: 40px;">0000</td></tr> <tr><td style="width: 40px;">6</td><td style="width: 40px;">0001</td></tr> </table> </div> <p style="text-align: center; margin-top: 10px;">HF234はHFFFFFF234、H0001はH00000001です。</p>				4321			FW002	4321	FW000	F234	FW001	0001	FW002	0000	FW003	FFFF	H180000	FFFF	2	F234	4	0000	6	0001																								
4321																																																	
FW002																																																	
4321																																																	
FW000	F234																																																
FW001	0001																																																
FW002	0000																																																
FW003	FFFF																																																
H180000	FFFF																																																
2	F234																																																
4	0000																																																
6	0001																																																
有効パラメーター	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 25%;">S</th> <th style="width: 15%;">ビット型 PI/O</th> <th style="width: 15%;">ワード型 PI/O</th> <th style="width: 15%;">定数</th> <th style="width: 20%;"></th> <th style="width: 20%;">D</th> <th style="width: 15%;">ビット型 PI/O</th> <th style="width: 15%;">ワード型 PI/O</th> <th style="width: 15%;">定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> <td></td> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> <td></td> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> <td></td> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S	ビット型 PI/O	ワード型 PI/O	定数		D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○		ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	○		ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△		インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△		インダイレクトロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数		D	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクトワード長	○	○	○		ダイレクトワード長	○	○	×																																									
ダイレクトロング長	○	○	○		ダイレクトロング長	○	○	×																																									
インダイレクトワード長	×	△	△		インダイレクトワード長	×	△	△																																									
インダイレクトロング長	×	△	△		インダイレクトロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																																	

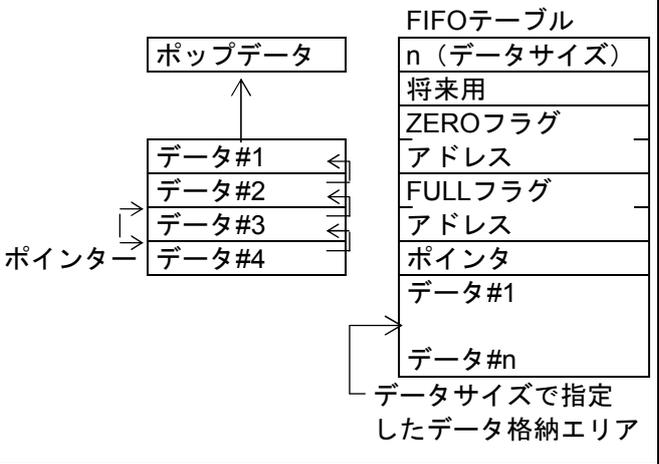
## 5 応用命令

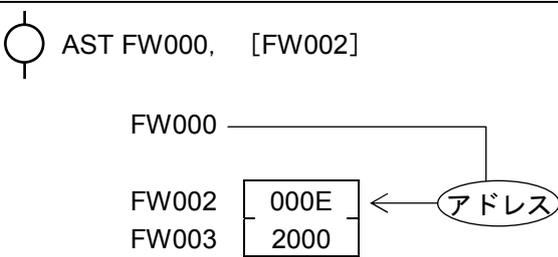
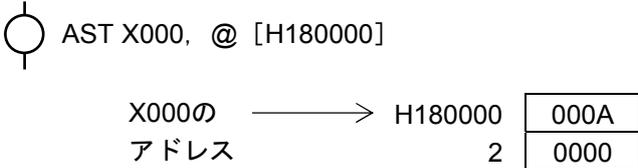
EXC	交換																										
機能説明	ソースの内容とデスティネーションを交換します。																										
パラメーターと処理内容	 EXC S, D <div style="border: 1px solid black; padding: 5px; display: inline-block; margin-left: 20px;">S ↔ D</div> <p>S : ソース D : デスティネーション</p>																										
フラグの設定	Eが変化、他はOFF																										
備考	転送するサイズが異なる場合、型変換して交換します。																										
使用例	 EXC FW000, FW002 <div style="margin-left: 20px;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="padding: 2px;">FW000</td><td style="padding: 2px;">1234</td><td style="padding: 2px;">←</td></tr> <tr><td style="padding: 2px;">FW002</td><td style="padding: 2px;">4321</td><td style="padding: 2px;">←</td></tr> </table> </div>  EXC @H170000, @[H180000] <div style="margin-left: 20px;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="padding: 2px;">H170000</td><td style="padding: 2px;">F234</td><td style="padding: 2px;">↔</td><td style="padding: 2px;">H180000</td><td style="padding: 2px;">0010</td></tr> <tr><td></td><td></td><td></td><td style="padding: 2px;">2</td><td style="padding: 2px;">0001</td></tr> </table> <p>交換後</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="padding: 2px;">H170000</td><td style="padding: 2px;">7FFF</td><td style="padding: 2px;"></td><td style="padding: 2px;">H180000</td><td style="padding: 2px;">FFFF</td></tr> <tr><td></td><td></td><td></td><td style="padding: 2px;">2</td><td style="padding: 2px;">F234</td></tr> </table> </div>	FW000	1234	←	FW002	4321	←	H170000	F234	↔	H180000	0010				2	0001	H170000	7FFF		H180000	FFFF				2	F234
FW000	1234	←																									
FW002	4321	←																									
H170000	F234	↔	H180000	0010																							
			2	0001																							
H170000	7FFF		H180000	FFFF																							
			2	F234																							
有効パラメーター	<table border="1" style="border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="padding: 5px;">S, D</th> <th style="padding: 5px;">ビット型 PI/O</th> <th style="padding: 5px;">ワード型 PI/O</th> <th style="padding: 5px;">定数</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;">△はアドレス 値が奇数時は パラメーター エラー</td> <td style="padding: 5px;">○</td> <td style="padding: 5px;">○</td> <td style="padding: 5px;">×</td> </tr> <tr> <td style="padding: 5px;">ダイレクト ワード長</td> <td style="padding: 5px;">○</td> <td style="padding: 5px;">○</td> <td style="padding: 5px;">×</td> </tr> <tr> <td style="padding: 5px;">ダイレクト ロング長</td> <td style="padding: 5px;">○</td> <td style="padding: 5px;">○</td> <td style="padding: 5px;">×</td> </tr> <tr> <td style="padding: 5px;">インダイレクト ワード長</td> <td style="padding: 5px;">×</td> <td style="padding: 5px;">△</td> <td style="padding: 5px;">△</td> </tr> <tr> <td style="padding: 5px;">インダイレクト ロング長</td> <td style="padding: 5px;">×</td> <td style="padding: 5px;">△</td> <td style="padding: 5px;">△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	△はアドレス 値が奇数時は パラメーター エラー	○	○	×	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△		
S, D	ビット型 PI/O	ワード型 PI/O	定数																								
△はアドレス 値が奇数時は パラメーター エラー	○	○	×																								
ダイレクト ワード長	○	○	×																								
ダイレクト ロング長	○	○	×																								
インダイレクト ワード長	×	△	△																								
インダイレクト ロング長	×	△	△																								

PSH	FIFO書き込み																																								
機能説明	ソースの内容をFIFOテーブルへプッシュします。FIFOテーブルのデータ長はワードだけです。																																								
パラメーターと処理内容	 PSH S, TB S : ソース TB : FIFOテーブル先頭アドレス																																								
フラグの設定	Eが変化、他はOFF																																								
備考	<p><math>n \leq 0</math>、<math>n &gt; 256</math>のとき未処理。                  ポインタ&lt;0またはデータサイズ&lt;ポインタのとき未処理。ポインタ=データサイズのときFULLフラグONし未処理。プッシュ後、ポインタをインクリメントしてnになった場合FULLフラグON。それ以外はZEROフラグOFF、FULLフラグOFF。TBが定数の場合はそれをテーブルアドレスと見なします。</p>																																								
使用例																																									
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>TB</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>×</td> <td>○</td> <td>△</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>△</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	TB	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	×	○	△	ダイレクトロング長	×	○	△	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	○	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
TB	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	×	○	△																																						
ダイレクトロング長	×	○	△																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						



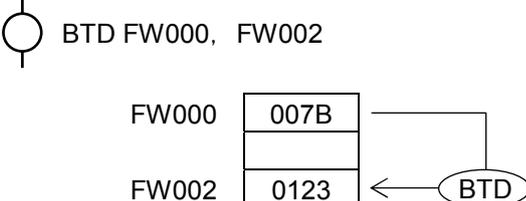
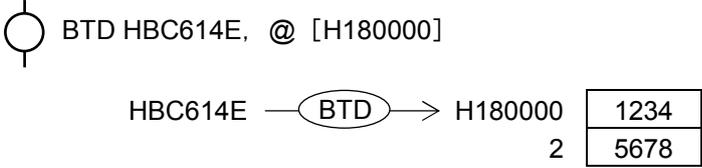
## 5 応用命令

POP	FIFO読み出し																																								
機能説明	FIFOテーブルのポップ処理を行い、ポップデータをデスティネーションに格納します。 FIFOテーブルのデータ長はワードだけです。																																								
パラメーターと処理内容	 POP TB, D D : デスティネーション TB : FIFOテーブル先頭アドレス																																								
フラグの設定	Eが変化、他はOFF																																								
備考	<p><math>n \leq 0</math>、<math>n &gt; 256</math>のとき未処理。  ポインタ<math>&lt; 0</math>またはデータサイズ<math>&lt;</math>  ポインタのとき未処理。ポイン  ター<math>= 0</math>のときZEROフラグONし未処  理。ポップ後、ポインタをデクリ  メントして0になった場合ZEROフラグON。それ以外はZEROフラグOFF、FULLフラグ  OFF。TBが定数の場合はそれをテーブルアドレスと見なします。</p> 																																								
使用例	 POP DW000, FW000 FW000 1234 ← POP DW000 DW005 DW006 ホイター DW007 1234 DW008 データ2 DW009 データ3 DW00A データ4																																								
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>TB</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>×</td> <td>○</td> <td>△</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>△</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	TB	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	×	○	△	ダイレクト ロング長	×	○	△	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
TB	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクト ワード長	×	○	△																																						
ダイレクト ロング長	×	○	△																																						
インダイレクト ワード長	×	△	△																																						
インダイレクト ロング長	×	△	△																																						
D	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクト ワード長	○	○	×																																						
ダイレクト ロング長	○	○	×																																						
インダイレクト ワード長	×	△	△																																						
インダイレクト ロング長	×	△	△																																						
△はアドレス 値が奇数時は パラメーター エラー																																									

AST	アドレスセット																																											
機能説明	ソースのアドレスデータをデスティネーションに転送します。 PI/Oだけ有効です。																																											
パラメーターと処理内容	 AST S, D	Sのアドレス → D																																										
	S : ソース D : デスティネーション																																											
フラグの設定	Eが変化、他はOFF																																											
備考	Dがワード長指定だとアドレス値がワード長に型変換されますので注意してください。																																											
使用例	 																																											
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>×</td> <td>×</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>×</td> <td>×</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>×</td> <td>×</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	×	×	インダイレクト ロング長	×	×	×	D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	×	×	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	×	×	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	×																																									
ダイレクト ロング長	○	○	×																																									
インダイレクト ワード長	×	×	×																																									
インダイレクト ロング長	×	×	×																																									
D	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	×	×	×																																									
ダイレクト ロング長	○	○	×																																									
インダイレクト ワード長	×	×	×																																									
インダイレクト ロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												

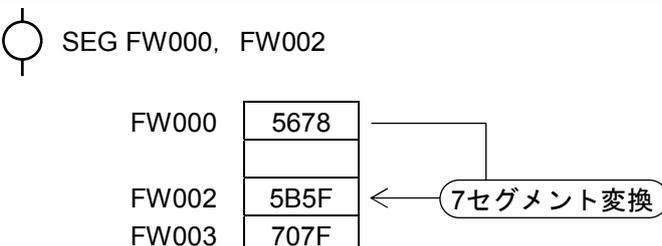
## 5 応用命令

SCH	サーチ																																								
機能説明	ソースの内容がデスティネーションから指定エリア (m) 内に存在するか探し、デスティネーションからのステップ数 (n) をリザルトに格納します。																																								
パラメーターと処理内容	<div style="display: flex; align-items: flex-start;"> <div style="flex: 1;"> <p>○ SCH S, D, m, R</p> <p>S : ソース</p> <p>D : デスティネーション</p> <p>m : サーチステップ数</p> <p>R : リザルト</p> </div> <div style="flex: 1; text-align: center;"> <p>S</p> <p>データ</p> <p>サーチテーブル</p> <p>D (0)</p> <p>(1)</p> <p>(n) データ</p> <p>(m-1)</p> <p>R</p> <p>n</p> <p>サーチ範囲</p> </div> </div>																																								
フラグの設定	Eが変化、他はOFF																																								
備考	<p><math>m \leq 0</math>、<math>m &gt; 256</math> のとき未処理。</p> <p>一致データは最初に見つけたもの。</p> <p>サーチ範囲に一致データがないときリザルトには-1が設定されます。</p> <p>サーチデータの型 (ロング、ワード) が異なる場合はエラーになります。</p> <p>nは0から始まります。</p>																																								
使用例	<p>○ SCH DW000, FW000, 5, FW005</p> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 20px;"> <p>DW000 1234</p> <p>FW000 0000</p> <p>FW001 1234</p> <p>FW002 0000</p> <p>FW003 1234</p> <p>FW004 0000</p> <p>FW005 0001</p> </div> <div style="margin-right: 20px;"> <p>1番目</p> </div> <div style="text-align: center;"> <p>→ サーチ</p> </div> </div>																																								
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, m</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>ダイレクトロング長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>インダイレクトワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>D, R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>ダイレクトロング長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>インダイレクトワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table>	S, m	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	D, R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, m	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	○	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
D, R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	×																																						
ダイレクトロング長	○	○	×																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									

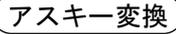
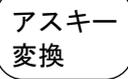
BTD	バイナリー→BCD変換																																											
機能説明	ソースの内容をバイナリーからBCDに変換し、リザルトに格納します。																																											
パラメーターと処理内容	 BTD S, R	S (バイナリー) → R (BCD)																																										
	S : ソース R : リザルト																																											
フラグの設定	E, Vが変化、他はOFF																																											
備考	S<0のときEフラグON、VフラグOFFし、未処理。 オーバーフロー時H9999、またはH99999999を設定。																																											
使用例	 																																											
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	○																																									
ダイレクト ロング長	○	○	○																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	×																																									
ダイレクト ロング長	○	○	×																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												

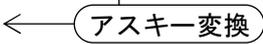
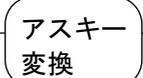
## 5 応用命令

DTB	BCD→バイナリー変換																																									
機能説明	ソースの内容をBCDからバイナリーに変換し、リザルトに格納します。																																									
パラメーターと処理内容	 DTB S, R  S : ソース R : リザルト	S (BCD) → R (バイナリー)																																								
フラグの設定	E, Vが変化、他はOFF																																									
備考	SのなかでA~Fが使用されているとEフラグONして未処理。																																									
使用例	 DTB FW000, FW002  <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>FW000</td><td>1234</td></tr> <tr><td>FW002</td><td>04D2</td></tr> </table>    DTB H99999999, @ [H180000]  <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>H99999999</td><td>DTB</td><td>H180000</td></tr> </table> <table border="1" style="display: inline-table; margin-left: 20px;"> <tr><td>05F5</td></tr> <tr><td>E0FF</td></tr> </table>		FW000	1234	FW002	04D2	H99999999	DTB	H180000	05F5	E0FF																															
FW000	1234																																									
FW002	04D2																																									
H99999999	DTB	H180000																																								
05F5																																										
E0FF																																										
有効パラメーター	<table border="1"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	<table border="1"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクトワード長	○	○	○																																							
ダイレクトロング長	○	○	○																																							
インダイレクトワード長	×	△	△																																							
インダイレクトロング長	×	△	△																																							
R	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクトワード長	○	○	×																																							
ダイレクトロング長	○	○	×																																							
インダイレクトワード長	×	△	△																																							
インダイレクトロング長	×	△	△																																							
△はアドレス値が奇数時はパラメーターエラー																																										

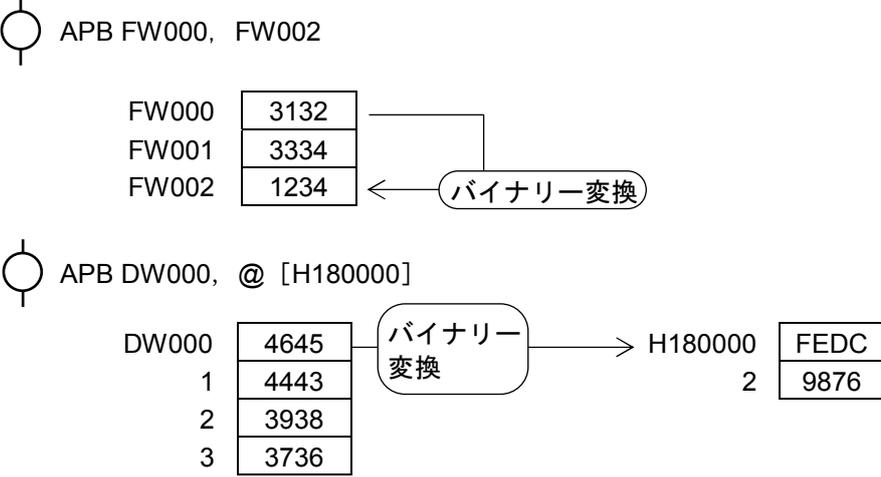
SEG	バイナリー→7セグメント変換																																									
機能説明	ソースの内容をバイナリーから7セグメントデータに変換し、リザルトに格納します。																																									
パラメーターと処理内容	 SEG S, R  S : ソース R : リザルト	S (バイナリー) → R (7セグメントデータ)																																								
フラグの設定	Eが変化、他はOFF																																									
備考	Sのサイズ×2がRに書き込まれます。																																									
使用例	  <p>[7セグメント対応表]</p> <table border="1"> <thead> <tr> <th>No.</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> <th>8</th> <th>9</th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>データ</td> <td>7E</td> <td>30</td> <td>6D</td> <td>79</td> <td>33</td> <td>5B</td> <td>5F</td> <td>70</td> <td>7F</td> <td>7B</td> <td>77</td> <td>1F</td> <td>4E</td> <td>3D</td> <td>4F</td> <td>47</td> </tr> </tbody> </table>		No.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	データ	7E	30	6D	79	33	5B	5F	70	7F	7B	77	1F	4E	3D	4F	47						
No.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																										
データ	7E	30	6D	79	33	5B	5F	70	7F	7B	77	1F	4E	3D	4F	47																										
有効パラメーター	<table border="1"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	<table border="1"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	○																																							
ダイレクト ロング長	○	○	○																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
R	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	×																																							
ダイレクト ロング長	○	○	×																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
△はアドレス値が奇数時はパラメーターエラー																																										

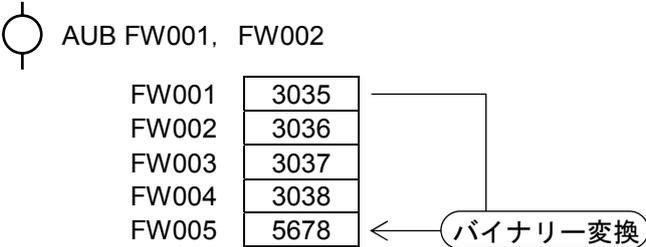
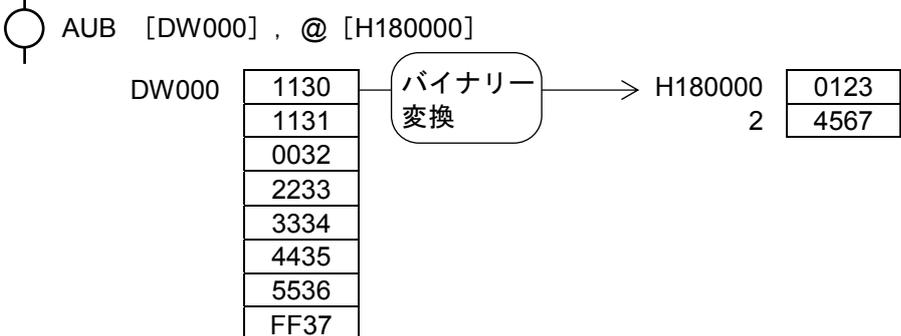
## 5 応用命令

ASP	バイナリー→アスキー変換パックモード																																																			
機能説明	ソースの内容をバイナリーからアスキーデータに変換し、パックモードでリザルトに格納します。																																																			
パラメーターと処理内容	 ASP S, R  S : ソース R : リザルト	S (バイナリー) → R (アスキーパック)																																																		
フラグの設定	Eが変化、他はOFF																																																			
備考	Sのサイズ×2がRに書き込まれます。																																																			
使用例	 ASP FW000, FW002  <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>FW000</td><td>5678</td></tr> <tr><td>FW002</td><td>3536</td></tr> <tr><td>FW003</td><td>3738</td></tr> </table>    ASP HDEF01234, @ [H180000]  <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>HDEF01234</td><td>アスキー変換</td></tr> </table>  <table border="1" style="display: inline-table; margin-left: 20px;"> <tr><td>H180000</td><td>4445</td></tr> <tr><td>2</td><td>4630</td></tr> <tr><td>4</td><td>3132</td></tr> <tr><td>6</td><td>3334</td></tr> </table>  <p>[アスキー、バイナリー対応表]</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <th>バイナリー</th> <th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th> </tr> <tr> <th>アスキー</th> <td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td> </tr> </table>		FW000	5678	FW002	3536	FW003	3738	HDEF01234	アスキー変換	H180000	4445	2	4630	4	3132	6	3334	バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46
FW000	5678																																																			
FW002	3536																																																			
FW003	3738																																																			
HDEF01234	アスキー変換																																																			
H180000	4445																																																			
2	4630																																																			
4	3132																																																			
6	3334																																																			
バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																																				
アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46																																				
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr><td>ダイレクトワード長</td><td>○</td><td>○</td><td>○</td></tr> <tr><td>ダイレクトロング長</td><td>○</td><td>○</td><td>○</td></tr> <tr><td>インダイレクトワード長</td><td>×</td><td>△</td><td>△</td></tr> <tr><td>インダイレクトロング長</td><td>×</td><td>△</td><td>△</td></tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr><td>ダイレクトワード長</td><td>○</td><td>○</td><td>×</td></tr> <tr><td>ダイレクトロング長</td><td>○</td><td>○</td><td>×</td></tr> <tr><td>インダイレクトワード長</td><td>×</td><td>△</td><td>△</td></tr> <tr><td>インダイレクトロング長</td><td>×</td><td>△</td><td>△</td></tr> </tbody> </table>		S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△										
S	ビット型 PI/O	ワード型 PI/O	定数																																																	
ダイレクトワード長	○	○	○																																																	
ダイレクトロング長	○	○	○																																																	
インダイレクトワード長	×	△	△																																																	
インダイレクトロング長	×	△	△																																																	
R	ビット型 PI/O	ワード型 PI/O	定数																																																	
ダイレクトワード長	○	○	×																																																	
ダイレクトロング長	○	○	×																																																	
インダイレクトワード長	×	△	△																																																	
インダイレクトロング長	×	△	△																																																	
△はアドレス値が奇数時はパラメーターエラー																																																				

ASU	バイナリー→アスキー変換アンパックモード																																																							
機能説明	ソースの内容をバイナリーからアスキーデータに変換し、アンパックモードでリザルトに格納します。																																																							
パラメーターと処理内容	 ASU S, R S : ソース R : リザルト	S (バイナリー) → R (アスキーアンパック)																																																						
フラグの設定	Eが変化、他はOFF																																																							
備考	Sのサイズ×4がRに書き込まれます。																																																							
使用例	 ASU FW001, FW002 <table border="1" data-bbox="502 806 734 974"> <tr><td>FW001</td><td>5678</td></tr> <tr><td>FW002</td><td>3035</td></tr> <tr><td>FW003</td><td>3036</td></tr> <tr><td>FW004</td><td>3037</td></tr> <tr><td>FW005</td><td>3038</td></tr> </table>   ASU HDEF01234, @ [H180000] HDEF01234  H180000 <table border="1" data-bbox="1093 1086 1212 1355"> <tr><td>3044</td></tr> <tr><td>2 3045</td></tr> <tr><td>4 3046</td></tr> <tr><td>6 3030</td></tr> <tr><td>8 3031</td></tr> <tr><td>A 3032</td></tr> <tr><td>C 3033</td></tr> <tr><td>E 3034</td></tr> </table> [アスキー、バイナリー対応表] <table border="1" data-bbox="383 1433 1452 1500"> <tr><th>バイナリー</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th></tr> <tr><th>アスキー</th><td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td></tr> </table>				FW001	5678	FW002	3035	FW003	3036	FW004	3037	FW005	3038	3044	2 3045	4 3046	6 3030	8 3031	A 3032	C 3033	E 3034	バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46
FW001	5678																																																							
FW002	3035																																																							
FW003	3036																																																							
FW004	3037																																																							
FW005	3038																																																							
3044																																																								
2 3045																																																								
4 3046																																																								
6 3030																																																								
8 3031																																																								
A 3032																																																								
C 3033																																																								
E 3034																																																								
バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																																								
アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46																																								
有効パラメーター	<table border="1" data-bbox="375 1579 893 1948"> <thead> <tr> <th>S</th> <th>ビット型 P/I/O</th> <th>ワード型 P/I/O</th> <th>定数</th> </tr> </thead> <tbody> <tr><td>ダイレクトワード長</td><td>○</td><td>○</td><td>○</td></tr> <tr><td>ダイレクトロング長</td><td>○</td><td>○</td><td>○</td></tr> <tr><td>インダイレクトワード長</td><td>×</td><td>△</td><td>△</td></tr> <tr><td>インダイレクトロング長</td><td>×</td><td>△</td><td>△</td></tr> </tbody> </table>				S	ビット型 P/I/O	ワード型 P/I/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	<table border="1" data-bbox="941 1579 1460 1948"> <thead> <tr> <th>R</th> <th>ビット型 P/I/O</th> <th>ワード型 P/I/O</th> <th>定数</th> </tr> </thead> <tbody> <tr><td>ダイレクトワード長</td><td>○</td><td>○</td><td>×</td></tr> <tr><td>ダイレクトロング長</td><td>○</td><td>○</td><td>×</td></tr> <tr><td>インダイレクトワード長</td><td>×</td><td>△</td><td>△</td></tr> <tr><td>インダイレクトロング長</td><td>×</td><td>△</td><td>△</td></tr> </tbody> </table>				R	ビット型 P/I/O	ワード型 P/I/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△								
S	ビット型 P/I/O	ワード型 P/I/O	定数																																																					
ダイレクトワード長	○	○	○																																																					
ダイレクトロング長	○	○	○																																																					
インダイレクトワード長	×	△	△																																																					
インダイレクトロング長	×	△	△																																																					
R	ビット型 P/I/O	ワード型 P/I/O	定数																																																					
ダイレクトワード長	○	○	×																																																					
ダイレクトロング長	○	○	×																																																					
インダイレクトワード長	×	△	△																																																					
インダイレクトロング長	×	△	△																																																					
△はアドレス値が奇数時はパラメーターエラー																																																								

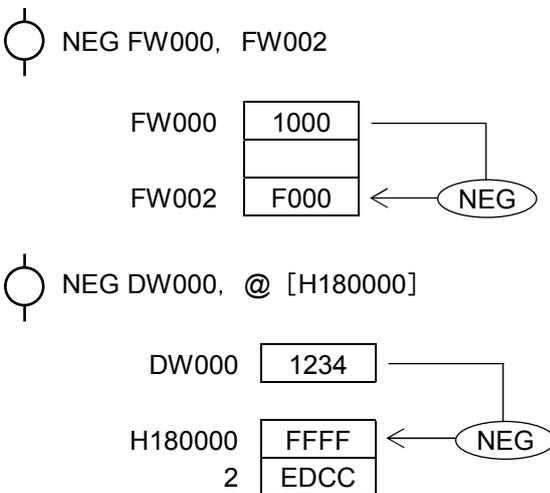
## 5 応用命令

APB	アスキー→バイナリー変換パックモード																																									
機能説明	ソースの内容をアスキーデータ（パックモード）からバイナリーに変換し、リザルトに格納します。																																									
パラメーターと処理内容	 APB S, R  S : ソース R : リザルト	S (アスキーパック) → R (バイナリー)																																								
フラグの設定	Eが変化、他はOFF																																									
備考	Rのサイズ×2がSから取り込まれ変換されます。 SにH30～39、H41～46のデータがあるとEフラグONし、未処理。																																									
使用例	 <p>[アスキー、バイナリー対応表]</p> <table border="1"> <tr> <th>バイナリー</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> <th>8</th> <th>9</th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> </tr> <tr> <th>アスキー</th> <td>30</td> <td>31</td> <td>32</td> <td>33</td> <td>34</td> <td>35</td> <td>36</td> <td>37</td> <td>38</td> <td>39</td> <td>41</td> <td>42</td> <td>43</td> <td>44</td> <td>45</td> <td>46</td> </tr> </table>		バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46						
バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																										
アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46																										
有効パラメーター	<table border="1"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	×	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	<table border="1"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	×	○	×																																							
ダイレクト ロング長	×	○	×																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
R	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	×																																							
ダイレクト ロング長	○	○	×																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
△はアドレス値が奇数時はパラメーターエラー																																										

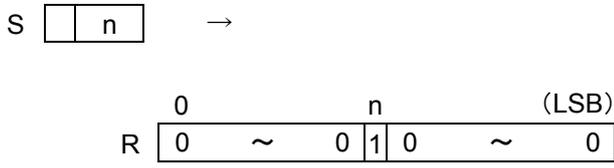
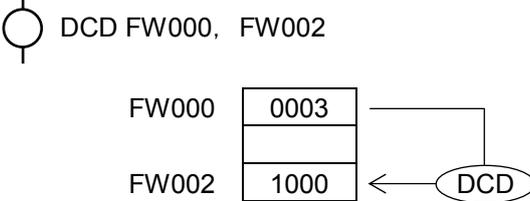
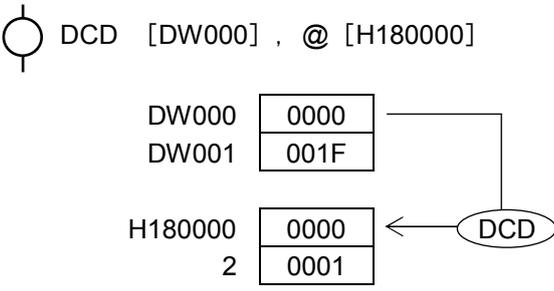
AUB	アスキー→バイナリー変換アンパックモード																																											
機能説明	ソースの内容をアスキーデータ（アンパックモード）からバイナリーに変換し、リザルトに格納します。																																											
パラメーターと処理内容	 AUB S, R S : ソース R : リザルト		S (アスキー → R (バイナリー) アンパック)																																									
フラグの設定	Eが変化、他はOFF																																											
備考	Rのサイズ×4がSから取り込まれ変換されます。 SにH30～39、H41～46のデータがあるとEフラグONし、未処理。																																											
使用例	  [アスキー、バイナリー対応表] <table border="1" data-bbox="383 1444 1452 1512"> <tr> <td>バイナリー</td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td> </tr> <tr> <td>アスキー</td> <td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td> </tr> </table>				バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46						
バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																												
アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46																												
有効パラメーター	<table border="1" data-bbox="375 1579 893 1948"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>		S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	×	○	×	ダイレクトロング長	×	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	<table border="1" data-bbox="941 1579 1460 1948"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>		R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクトワード長	×	○	×																																									
ダイレクトロング長	×	○	×																																									
インダイレクトワード長	×	△	△																																									
インダイレクトロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクトワード長	○	○	×																																									
ダイレクトロング長	○	○	×																																									
インダイレクトワード長	×	△	△																																									
インダイレクトロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												

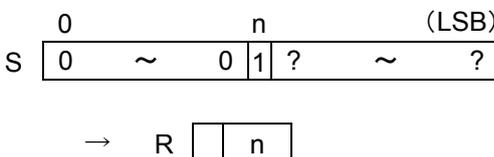
## 5 応用命令

ABS	絶対値																																								
機能説明	ソースの内容の絶対値をリザルトに格納します。																																								
パラメーターと処理内容	 ABS S, R <table border="1" style="float: right; margin-left: 20px;"> <tr> <td> S </td> <td>→</td> <td>R</td> </tr> </table> <p>S : ソース R : リザルト</p>	S	→	R																																					
S	→	R																																							
フラグの設定	E, Vが変化、他はOFF																																								
備考	オーバーフロー発生時、リザルトにはH7FFFFFFFが設定されます。																																								
使用例	 ABS FW000, FW002 <table style="margin-left: 20px;"> <tr> <td>FW000</td> <td><table border="1"><tr><td>FF9C</td></tr></table></td> <td rowspan="2">←</td> <td rowspan="2"></td> </tr> <tr> <td>FW002</td> <td><table border="1"><tr><td>0064</td></tr></table></td> </tr> </table>  ABS DW000, @ [H180000] <table style="margin-left: 20px;"> <tr> <td>DW000</td> <td><table border="1"><tr><td>FFFB</td></tr></table></td> <td rowspan="2">←</td> <td rowspan="2"></td> </tr> <tr> <td>H180000 2</td> <td><table border="1"><tr><td>0000</td></tr><tr><td>0005</td></tr></table></td> </tr> </table>	FW000	<table border="1"><tr><td>FF9C</td></tr></table>	FF9C	←		FW002	<table border="1"><tr><td>0064</td></tr></table>	0064	DW000	<table border="1"><tr><td>FFFB</td></tr></table>	FFFB	←		H180000 2	<table border="1"><tr><td>0000</td></tr><tr><td>0005</td></tr></table>	0000	0005																							
FW000	<table border="1"><tr><td>FF9C</td></tr></table>	FF9C	←																																						
FF9C																																									
FW002	<table border="1"><tr><td>0064</td></tr></table>	0064																																							
0064																																									
DW000	<table border="1"><tr><td>FFFB</td></tr></table>	FFFB	←																																						
FFFB																																									
H180000 2	<table border="1"><tr><td>0000</td></tr><tr><td>0005</td></tr></table>	0000	0005																																						
0000																																									
0005																																									
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクト ワード長	○	○	○																																						
ダイレクト ロング長	○	○	○																																						
インダイレクト ワード長	×	△	△																																						
インダイレクト ロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクト ワード長	○	○	×																																						
ダイレクト ロング長	○	○	×																																						
インダイレクト ワード長	×	△	△																																						
インダイレクト ロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									

NEG	符号変換																																											
機能説明	ソースの内容の符号変換処理を行いリザルトに格納します。																																											
パラメーターと処理内容	 <b>NEG S, R</b>		-S → R																																									
	S : ソース R : リザルト																																											
フラグの設定	E, Vが変化、他はOFF																																											
備考	オーバーフロー発生時、リザルトにはH7FFF, H7FFFFFFFが設定されます。																																											
使用例																																												
有効パラメーター	<table border="1"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>		S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	<table border="1"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>		R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	○																																									
ダイレクト ロング長	○	○	○																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	×																																									
ダイレクト ロング長	○	○	×																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												

## 5 応用命令

DCD	デコード																																									
機能説明	ソースの内容をデコードして結果をリザルトに格納します。																																									
パラメーターと処理内容	 DCD S, R S : ソース R : リザルト	 <p>Sで指定されたnを使用して、RのMSB側から数えて(0から数えます)ビット番号nのビットをONします。</p>																																								
フラグの設定	Eが変化、他はOFF																																									
備考	Sで有効なビットはRのサイズ指定で決まります。ワード指定時：下位4ビット、ロング指定時：下位5ビット																																									
使用例	 <p>DCD [DW000], @ [H180000]</p> 																																									
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>		S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクトワード長	○	○	○																																							
ダイレクトロング長	○	○	○																																							
インダイレクトワード長	×	△	△																																							
インダイレクトロング長	×	△	△																																							
R	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクトワード長	○	○	×																																							
ダイレクトロング長	○	○	×																																							
インダイレクトワード長	×	△	△																																							
インダイレクトロング長	×	△	△																																							
△はアドレス値が奇数時はパラメーターエラー																																										

ECD	エンコード																																											
機能説明	ソースの内容をエンコードして結果をリザルトに格納します。																																											
パラメーターと処理内容	 ECD S, R S : ソース R : リザルト	 <p>SのMSB側から数えて（0から数えます）最初に1が検出されたnをRに格納します。</p>																																										
フラグの設定	Eが変化、他はOFF																																											
備考	S=0のとき未処理。エンコードするビットはMSBから最初に1が検出されたビットについてだけ有効。																																											
使用例	 ECD FW000, FW002 FW000 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0456</td></tr></table> FW002 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0005</td></tr></table> ←  ECD				0456	0005																																						
0456																																												
0005																																												
	 ECD [DW000] , @ [H180000] DW000 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0000</td></tr></table> DW001 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0080</td></tr></table> H180000 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0000</td></tr><tr><td>0018</td></tr></table> ←  ECD				0000	0080	0000	0018																																				
0000																																												
0080																																												
0000																																												
0018																																												
有効パラメーター	<table border="1" style="display: inline-table; vertical-align: middle;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table; vertical-align: middle; margin-left: 20px;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	○																																									
ダイレクト ロング長	○	○	○																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	×																																									
ダイレクト ロング長	○	○	×																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												

## 5 応用命令

LSR	論理右シフト																																									
機能説明	ソースの内容をデスティネーションの内容で右へシフトし、結果をリザルトに格納します。																																									
パラメーターと処理内容	<p>⊙ LSR S, D, R</p> <p>S : ソース R : リザルト D : デスティネーション</p>	<p>RSはワード ロングによる</p>																																								
フラグの設定	Eが変化、他はOFF																																									
備考	Dの有効ビットはSがワード長のとき下位4ビット、ロング長のとき下位5ビット。																																									
使用例	<p>⊙ LSR FW000, FW001, FW002</p> <table border="1"> <tr><td>FW000</td><td>0456</td></tr> <tr><td>FW001</td><td>0004</td></tr> <tr><td>FW002</td><td>0045</td></tr> </table> <p>⊙ LSR [DW000], 2, @[H180000]</p> <table border="1"> <tr><td>DW000</td><td>8765</td></tr> <tr><td>DW001</td><td>4321</td></tr> <tr><td>H180000</td><td>21D9</td></tr> <tr><td>2</td><td>50C8</td></tr> </table>			FW000	0456	FW001	0004	FW002	0045	DW000	8765	DW001	4321	H180000	21D9	2	50C8																									
FW000	0456																																									
FW001	0004																																									
FW002	0045																																									
DW000	8765																																									
DW001	4321																																									
H180000	21D9																																									
2	50C8																																									
有効パラメーター	<table border="1"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	<table border="1"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	○																																							
ダイレクト ロング長	○	○	○																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
R	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	×																																							
ダイレクト ロング長	○	○	×																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
△はアドレス値が奇数時はパラメーターエラー																																										

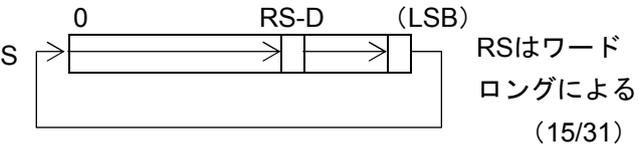
LSL	論理左シフト																																										
機能説明	ソースの内容をデスティネーションの内容で左へシフトし、結果をリザルトに格納します。																																										
パラメーターと処理内容	<p>⊙ LSL S, D, R</p> <p>S : ソース R : リザルト D : デスティネーション</p>																																										
フラグの設定	Eが変化、他はOFF																																										
備考	Dの有効ビットはSがワード長のとき下位4ビット、ロング長のとき下位5ビット。																																										
使用例	<p>⊙ LSL FW000, FW001, FW002</p> <table border="1"> <tr><td>FW000</td><td>0456</td><td rowspan="3">→</td></tr> <tr><td>FW001</td><td>0004</td></tr> <tr><td>FW002</td><td>4560</td><td>← (LSL)</td></tr> </table> <p>⊙ LSL [DW000], 2, @[H180000]</p> <table border="1"> <tr><td>DW000</td><td>8765</td><td rowspan="2">→ 2</td></tr> <tr><td>DW001</td><td>4321</td></tr> <tr><td>H180000</td><td>1D95</td><td rowspan="2">← (LSL)</td></tr> <tr><td>2</td><td>0C84</td></tr> </table>			FW000	0456	→	FW001	0004	FW002	4560	← (LSL)	DW000	8765	→ 2	DW001	4321	H180000	1D95	← (LSL)	2	0C84																						
FW000	0456	→																																									
FW001	0004																																										
FW002	4560		← (LSL)																																								
DW000	8765	→ 2																																									
DW001	4321																																										
H180000	1D95	← (LSL)																																									
2	0C84																																										
有効パラメーター	<table border="1"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>			S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクト ワード長	○	○	○																																								
ダイレクト ロング長	○	○	○																																								
インダイレクト ワード長	×	△	△																																								
インダイレクト ロング長	×	△	△																																								
R	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクト ワード長	○	○	×																																								
ダイレクト ロング長	○	○	×																																								
インダイレクト ワード長	×	△	△																																								
インダイレクト ロング長	×	△	△																																								
△はアドレス値が奇数時はパラメーターエラー																																											

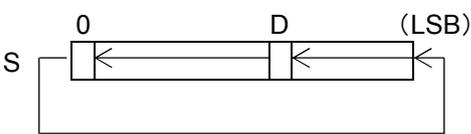
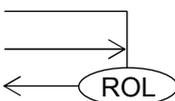
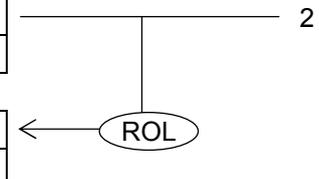
## 5 応用命令

ASR	算術右シフト																																										
機能説明	ソースの内容をデスティネーションの内容で右へシフト（符号ビット保持）し、結果をリザルトに格納します。																																										
パラメーターと処理内容	<p>⊙ ASR S, D, R</p> <p>S : ソース R : リザルト D : デスティネーション</p>																																										
フラグの設定	E, Vが変化、他はOFF																																										
備考	Rがワード長のとき下位ワードが設定されます。Dの有効ビットはSがワード長のとき下位4ビット、ロング長のとき下位5ビット。																																										
使用例	<p>⊙ ASR FW000, FW001, FW002</p> <p>⊙ ASR [DW000] , 2, @ [H180000]</p>																																										
有効パラメーター	<table border="1"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>			S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクトワード長	○	○	○																																								
ダイレクトロング長	○	○	○																																								
インダイレクトワード長	×	△	△																																								
インダイレクトロング長	×	△	△																																								
R	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクトワード長	○	○	×																																								
ダイレクトロング長	○	○	×																																								
インダイレクトワード長	×	△	△																																								
インダイレクトロング長	×	△	△																																								
△はアドレス値が奇数時はパラメーターエラー																																											

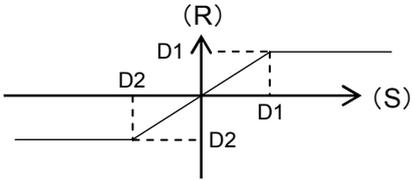
ASL	算術左シフト																																									
機能説明	ソースの内容をデスティネーションの内容で左へシフトし、結果をリザルトに格納します。オーバーフロー発生時フルスケールを設定します。																																									
パラメーターと処理内容	<p>⊙ ASL S, D, R</p> <p>S : ソース R : リザルト D : デスティネーション</p>	<p>LSBはワードロングによる (15/31)</p>																																								
フラグの設定	E, Vが変化、他はOFF																																									
備考	Dの有効ビットはSがワード長るとき下位4ビット、ロング長るとき下位5ビット。																																									
使用例	<p>⊙ ASL FW000, FW001, FW002</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>FW000</td><td>0456</td></tr> <tr><td>FW001</td><td>0004</td></tr> <tr><td>FW002</td><td>4560</td></tr> </table> <p>ASL</p> <p>⊙ ASL [DW000], 2, @[H180000]</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>DW000</td><td>4765</td></tr> <tr><td>DW001</td><td>4321</td></tr> </table> <p>2</p> <table border="1" style="display: inline-table;"> <tr><td>H180000</td><td>7FFF</td></tr> <tr><td>2</td><td>FFFF</td></tr> </table> <p>ASL</p> <p>オーバーフロー発生 (VフラグON)</p>				FW000	0456	FW001	0004	FW002	4560	DW000	4765	DW001	4321	H180000	7FFF	2	FFFF																								
FW000	0456																																									
FW001	0004																																									
FW002	4560																																									
DW000	4765																																									
DW001	4321																																									
H180000	7FFF																																									
2	FFFF																																									
有効パラメーター	<table border="1"> <thead> <tr> <th>S, D</th> <th>ビット型 P/I/O</th> <th>ワード型 P/I/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 P/I/O	ワード型 P/I/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	<table border="1"> <thead> <tr> <th>R</th> <th>ビット型 P/I/O</th> <th>ワード型 P/I/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	R	ビット型 P/I/O	ワード型 P/I/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 P/I/O	ワード型 P/I/O	定数																																							
ダイレクト ワード長	○	○	○																																							
ダイレクト ロング長	○	○	○																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
R	ビット型 P/I/O	ワード型 P/I/O	定数																																							
ダイレクト ワード長	○	○	×																																							
ダイレクト ロング長	○	○	×																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
△はアドレス値が奇数時はパラメーターエラー																																										

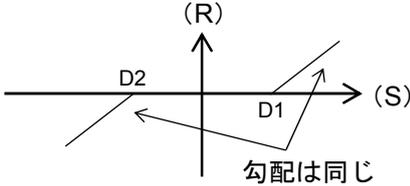
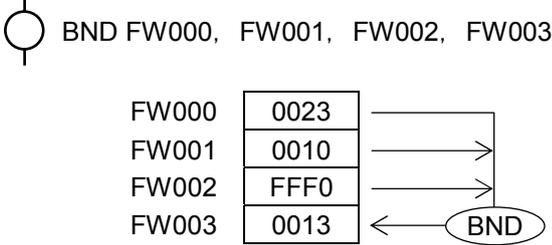
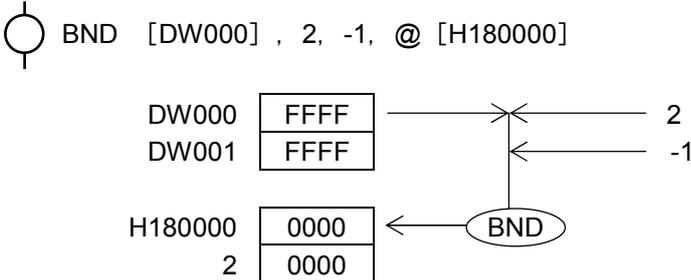
## 5 応用命令

ROR	右回転																																									
機能説明	ソースの内容をデスティネーションの内容で右へ回転し、結果をリザルトに格納します。																																									
パラメーターと処理内容	 ROR S, D, R S : ソース R : リザルト D : デスティネーション		RSはワード ロングによる (15/31)																																							
フラグの設定	Eが変化、他はOFF																																									
備考	Dの有効ビットはSがワード長のとき下位4ビット、ロング長のとき下位5ビット。																																									
使用例	 ROR FW000, FW001, FW002 <table border="1" data-bbox="454 840 874 958"> <tr><td>FW000</td><td>8456</td><td rowspan="3">→</td></tr> <tr><td>FW001</td><td>0004</td></tr> <tr><td>FW002</td><td>6845</td></tr> </table> <p style="text-align: center;">← ROR</p>  ROR [DW000], 2, @[H180000] <table border="1" data-bbox="454 1086 1013 1272"> <tr><td>DW000</td><td>8765</td><td rowspan="2">— 2</td></tr> <tr><td>DW001</td><td>4321</td></tr> <tr><td>H180000</td><td>61D9</td><td rowspan="2">← ROR</td></tr> <tr><td>2</td><td>50C8</td></tr> </table>			FW000	8456	→	FW001	0004	FW002	6845	DW000	8765	— 2	DW001	4321	H180000	61D9	← ROR	2	50C8																						
FW000	8456	→																																								
FW001	0004																																									
FW002	6845																																									
DW000	8765	— 2																																								
DW001	4321																																									
H180000	61D9	← ROR																																								
2	50C8																																									
有効 パラメーター	<table border="1" data-bbox="331 1368 847 1776"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定 数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定 数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	<table border="1" data-bbox="898 1368 1414 1776"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定 数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	R	ビット型 PI/O	ワード型 PI/O	定 数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定 数																																							
ダイレクト ワード長	○	○	○																																							
ダイレクト ロング長	○	○	○																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
R	ビット型 PI/O	ワード型 PI/O	定 数																																							
ダイレクト ワード長	○	○	×																																							
ダイレクト ロング長	○	○	×																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
△はアドレス 値が奇数時は パラメーター エラー																																										

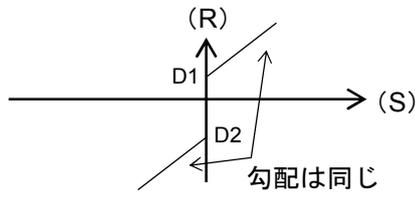
ROL	左回転																																										
機能説明	ソースの内容をデスティネーションの内容で左へ回転し、結果をリザルトに格納します。																																										
パラメーターと処理内容	 ROL S, D, R S : ソース R : リザルト D : デスティネーション	 RSはワード ロングによる (15/31)																																									
フラグの設定	Eが変化、他はOFF																																										
備考	Dの有効ビットはSがワード長るとき下位4ビット、ロング長るとき下位5ビット。																																										
使用例	 ROL FW000, FW001, FW002 <table border="1" data-bbox="502 840 917 952"> <tr><td>FW000</td><td>8456</td><td rowspan="3">→</td></tr> <tr><td>FW001</td><td>0004</td></tr> <tr><td>FW002</td><td>4568</td></tr> </table>   ROL [DW000], 2, @[H180000] <table border="1" data-bbox="502 1086 1061 1265"> <tr><td>DW000</td><td>8765</td><td rowspan="2">—</td><td rowspan="2">2</td></tr> <tr><td>DW001</td><td>4321</td></tr> </table> <table border="1" data-bbox="502 1198 941 1265"> <tr><td>H180000</td><td>1D95</td><td rowspan="2">←</td></tr> <tr><td>2</td><td>0C86</td></tr> </table> 			FW000	8456	→	FW001	0004	FW002	4568	DW000	8765	—	2	DW001	4321	H180000	1D95	←	2	0C86																						
FW000	8456	→																																									
FW001	0004																																										
FW002	4568																																										
DW000	8765	—	2																																								
DW001	4321																																										
H180000	1D95	←																																									
2	0C86																																										
有効パラメーター	<table border="1" data-bbox="375 1366 893 1769"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" data-bbox="941 1366 1460 1769"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>			S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクト ワード長	○	○	○																																								
ダイレクト ロング長	○	○	○																																								
インダイレクト ワード長	×	△	△																																								
インダイレクト ロング長	×	△	△																																								
R	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクト ワード長	○	○	×																																								
ダイレクト ロング長	○	○	×																																								
インダイレクト ワード長	×	△	△																																								
インダイレクト ロング長	×	△	△																																								
△はアドレス値が奇数時はパラメーターエラー																																											

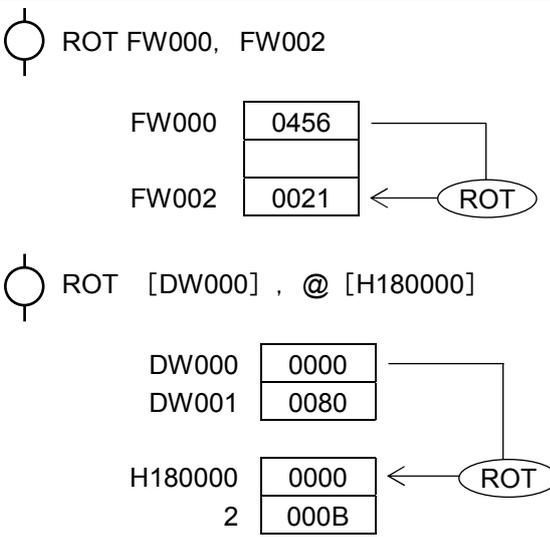
## 5 応用命令

LIM	リミッタ																																								
機能説明	ソースの内容を境界値（デスティネーションD1, D2）の内容と比較し、結果をリザルトに格納します。																																								
パラメーターと処理内容	<p>  LIM S, D1, D2, R         </p> <p>           S : ソース            R : リザルト            D1, D2 : デスティネーション         </p> 																																								
フラグの設定	E, Vが変化、他はOFF																																								
備考	D1<D2のときEフラグON																																								
使用例	<p>  LIM FW000, FW001, FW002, FW003         </p> <table border="1" data-bbox="454 884 874 1037"> <tr> <td>FW000</td> <td>0023</td> <td rowspan="4">→</td> </tr> <tr> <td>FW001</td> <td>0010</td> </tr> <tr> <td>FW002</td> <td>FFF0</td> </tr> <tr> <td>FW003</td> <td>0010</td> </tr> <tr> <td></td> <td></td> <td>←</td> <td>LIM</td> </tr> </table> <p>  LIM [DW000] , 2, -1, @ [H180000]         </p> <table border="1" data-bbox="454 1167 1023 1355"> <tr> <td>DW000</td> <td>FFFF</td> <td rowspan="2">→</td> <td rowspan="2">2</td> </tr> <tr> <td>DW001</td> <td>FFFF</td> </tr> <tr> <td>H180000</td> <td>FFFF</td> <td rowspan="2">←</td> <td rowspan="2">LIM</td> </tr> <tr> <td>2</td> <td>FFFF</td> </tr> </table>	FW000	0023	→	FW001	0010	FW002	FFF0	FW003	0010			←	LIM	DW000	FFFF	→	2	DW001	FFFF	H180000	FFFF	←	LIM	2	FFFF															
FW000	0023	→																																							
FW001	0010																																								
FW002	FFF0																																								
FW003	0010																																								
		←	LIM																																						
DW000	FFFF	→	2																																						
DW001	FFFF																																								
H180000	FFFF	←	LIM																																						
2	FFFF																																								
有効パラメーター	<table border="1" data-bbox="331 1451 847 1854"> <thead> <tr> <th>S, D1, D2</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" data-bbox="898 1451 1414 1854"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <p>△はアドレス値が奇数時はパラメーターエラー</p>	S, D1, D2	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D1, D2	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	○	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	×																																						
ダイレクトロング長	○	○	×																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						

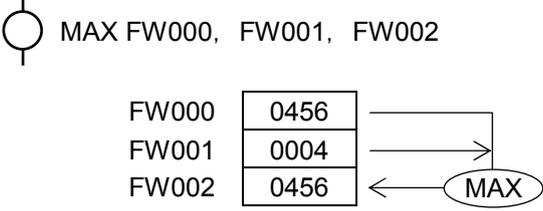
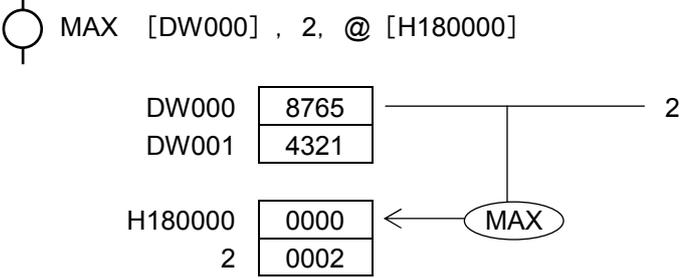
BND	デッドバンド																																											
機能説明	ソースの内容を境界値（デスティネーションD1, D2）の内容と比較し、境界範囲内を不感帯（データ0）としてリザルトに格納します。																																											
パラメーターと処理内容	 BND S, D1, D2, R S : ソース R : リザルト D1, D2 : デスティネーション																																											
フラグの設定	E, Vが変化、他はOFF																																											
備考	D1<D2のときEフラグON																																											
使用例	 																																											
有効パラメーター	<table border="1" data-bbox="375 1447 895 1854"> <thead> <tr> <th>S, D1, D2</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" data-bbox="943 1447 1463 1854"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <p>△はアドレス値が奇数時はパラメーターエラー</p>				S, D1, D2	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D1, D2	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	○																																									
ダイレクト ロング長	○	○	○																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	×																																									
ダイレクト ロング長	○	○	×																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									

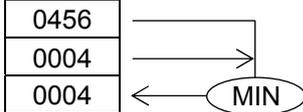
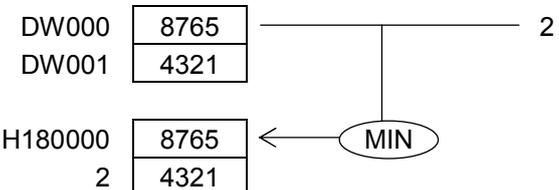
## 5 応用命令

ZON	デッドゾーン																																										
機能説明	ソースの内容を正負によってそれぞれバイアス（デスティネーションD1, D2）を加算し、結果をリザルトに格納します。																																										
パラメーターと処理内容	 ZON S, D1, D2, R S : ソース R : リザルト D1, D2 : デスティネーション																																										
フラグの設定	E, Vが変化、他はOFF																																										
備考	D1<D2のときEフラグON																																										
使用例	 ZON FW000, FW001, FW002, FW003 <table border="1" data-bbox="446 884 877 1052"> <tr><td>FW000</td><td>0023</td><td>→</td></tr> <tr><td>FW001</td><td>0010</td><td>→</td></tr> <tr><td>FW002</td><td>FFF0</td><td>→</td></tr> <tr><td>FW003</td><td>0033</td><td>←</td></tr> </table>  ZON [DW000] , 2, -1, @ [H180000] <table border="1" data-bbox="446 1164 1037 1366"> <tr><td>DW000</td><td>FFFF</td><td>←</td><td>2</td></tr> <tr><td>DW001</td><td>FFFF</td><td>←</td><td>-1</td></tr> <tr><td>H180000</td><td>FFFF</td><td>←</td><td></td></tr> <tr><td>2</td><td>FFFF</td><td></td><td></td></tr> </table>			FW000	0023	→	FW001	0010	→	FW002	FFF0	→	FW003	0033	←	DW000	FFFF	←	2	DW001	FFFF	←	-1	H180000	FFFF	←		2	FFFF														
FW000	0023	→																																									
FW001	0010	→																																									
FW002	FFF0	→																																									
FW003	0033	←																																									
DW000	FFFF	←	2																																								
DW001	FFFF	←	-1																																								
H180000	FFFF	←																																									
2	FFFF																																										
有効パラメーター	<table border="1" data-bbox="327 1456 845 1870"> <thead> <tr> <th>S, D1, D2</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr><td>ダイレクトワード長</td><td>○</td><td>○</td><td>○</td></tr> <tr><td>ダイレクトロング長</td><td>○</td><td>○</td><td>○</td></tr> <tr><td>インダイレクトワード長</td><td>×</td><td>△</td><td>△</td></tr> <tr><td>インダイレクトロング長</td><td>×</td><td>△</td><td>△</td></tr> </tbody> </table> <table border="1" data-bbox="893 1456 1412 1870"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr><td>ダイレクトワード長</td><td>○</td><td>○</td><td>×</td></tr> <tr><td>ダイレクトロング長</td><td>○</td><td>○</td><td>×</td></tr> <tr><td>インダイレクトワード長</td><td>×</td><td>△</td><td>△</td></tr> <tr><td>インダイレクトロング長</td><td>×</td><td>△</td><td>△</td></tr> </tbody> </table>			S, D1, D2	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	○	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	○	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D1, D2	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクトワード長	○	○	○																																								
ダイレクトロング長	○	○	○																																								
インダイレクトワード長	×	△	△																																								
インダイレクトロング長	×	△	△																																								
R	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクトワード長	○	○	×																																								
ダイレクトロング長	○	○	×																																								
インダイレクトワード長	×	△	△																																								
インダイレクトロング長	×	△	△																																								
△はアドレス値が奇数時はパラメーターエラー																																											

ROT	平方根																																											
機能説明	ソースの内容の平方根（整数部だけ）をリザルトに格納します。																																											
パラメーターと処理内容	 ROT S, R  S : ソース R : リザルト	$S \geq 0$ のとき Sの平方根（ルート） → R $S < 0$ のとき 0 → R																																										
フラグの設定	E, Vが変化、他はOFF																																											
備考																																												
使用例																																												
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	○																																									
ダイレクト ロング長	○	○	○																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	×																																									
ダイレクト ロング長	○	○	×																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												

## 5 応用命令

MAX	最大値																																											
機能説明	ソースの内容とデスティネーションの内容の大小比較を行い、大きい値をリザルトに格納します。																																											
パラメーターと処理内容	 MAX S, D, R  S : ソース R : リザルト D : デスティネーション	$S \geq D$ のとき S → R $S < D$ のとき D → R																																										
フラグの設定	E, Vが変化、他はOFF																																											
備考																																												
使用例	 <p>MAX FW000, FW001, FW002</p> <p>FW000 0456 FW001 0004 FW002 0456</p> <p>MAX</p>  <p>MAX [DW000], 2, @ [H180000]</p> <p>DW000 8765 DW001 4321</p> <p>H180000 0000 2 0002</p> <p>MAX</p>																																											
有効パラメーター	<table border="1"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S, D	ビット型 PI/O	ワード型 PI/O	定数	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	○	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数	R	ビット型 PI/O	ワード型 PI/O	定数																																					
ダイレクト ワード長	○	○	○	ダイレクト ワード長	○	○	×																																					
ダイレクト ロング長	○	○	○	ダイレクト ロング長	○	○	×																																					
インダイレクト ワード長	×	△	△	インダイレクト ワード長	×	△	△																																					
インダイレクト ロング長	×	△	△	インダイレクト ロング長	×	△	△																																					
△はアドレス値が奇数時はパラメーターエラー																																												

MIN	最小値																																											
機能説明	ソースの内容とデスティネーションの内容の大小比較を行い、小さい値をリザルトに格納します。																																											
パラメーターと処理内容	 MIN S, D, R  S : ソース R : リザルト D : デスティネーション	$S \leq D$ のとき S → R $S > D$ のとき D → R																																										
フラグの設定	E, Vが変化、他はOFF																																											
備考																																												
使用例	 MIN FW000, FW001, FW002  <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>FW000</td><td>0456</td></tr> <tr><td>FW001</td><td>0004</td></tr> <tr><td>FW002</td><td>0004</td></tr> </table>    MIN [DW000], 2, @ [H180000]  <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>DW000</td><td>8765</td></tr> <tr><td>DW001</td><td>4321</td></tr> </table> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>H180000</td><td>8765</td></tr> <tr><td>2</td><td>4321</td></tr> </table> 				FW000	0456	FW001	0004	FW002	0004	DW000	8765	DW001	4321	H180000	8765	2	4321																										
FW000	0456																																											
FW001	0004																																											
FW002	0004																																											
DW000	8765																																											
DW001	4321																																											
H180000	8765																																											
2	4321																																											
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	○	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	○	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	○																																									
ダイレクト ロング長	○	○	○																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	×																																									
ダイレクト ロング長	○	○	×																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												

## 5 応用命令

CLR	クリアー																														
機能説明	指定のI/Oエリアをクリアーします。TCLR, UCLR, CCLRは計数値エリアもクリアーします。																														
パラメーターと処理内容	 名称 S  名称：各CLR命令名称 S：ソース（クリアーするI/Oエリアの先頭）																														
フラグの設定	すべてのフラグに0を設定																														
備考																															
説明	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>名称</th> <th>機能</th> </tr> </thead> <tbody> <tr> <td>XCLR</td> <td>X000～XFFFをクリアー</td> </tr> <tr> <td>YCLR</td> <td>Y000～YFFFをクリアー</td> </tr> <tr> <td>GCLR</td> <td>G000～GFFFをクリアー</td> </tr> <tr> <td>RCLR</td> <td>R000～RFFFをクリアー</td> </tr> <tr> <td>KCLR</td> <td>K000～KFFFをクリアー</td> </tr> <tr> <td>TCLR</td> <td>T000～T3FFをクリアー Tの計測値クリアー</td> </tr> <tr> <td>UCLR</td> <td>U000～U3FFをクリアー Uの計測値クリアー</td> </tr> <tr> <td>CCLR</td> <td>C000～C3FFをクリアー Cの計測値クリアー</td> </tr> <tr> <td>VCLR</td> <td>V000～VFFFをクリアー</td> </tr> <tr> <td>ECLR</td> <td>E000～EFFFをクリアー</td> </tr> <tr> <td>FCLR</td> <td>S020～S027をクリアー</td> </tr> <tr> <td>JCLR</td> <td>J000～JFFFをクリアー</td> </tr> <tr> <td>QCLR</td> <td>Q000～QFFFをクリアー</td> </tr> <tr> <td>HHCLR</td> <td>HH000～HH1FFをクリアー</td> </tr> </tbody> </table>	名称	機能	XCLR	X000～XFFFをクリアー	YCLR	Y000～YFFFをクリアー	GCLR	G000～GFFFをクリアー	RCLR	R000～RFFFをクリアー	KCLR	K000～KFFFをクリアー	TCLR	T000～T3FFをクリアー Tの計測値クリアー	UCLR	U000～U3FFをクリアー Uの計測値クリアー	CCLR	C000～C3FFをクリアー Cの計測値クリアー	VCLR	V000～VFFFをクリアー	ECLR	E000～EFFFをクリアー	FCLR	S020～S027をクリアー	JCLR	J000～JFFFをクリアー	QCLR	Q000～QFFFをクリアー	HHCLR	HH000～HH1FFをクリアー
名称	機能																														
XCLR	X000～XFFFをクリアー																														
YCLR	Y000～YFFFをクリアー																														
GCLR	G000～GFFFをクリアー																														
RCLR	R000～RFFFをクリアー																														
KCLR	K000～KFFFをクリアー																														
TCLR	T000～T3FFをクリアー Tの計測値クリアー																														
UCLR	U000～U3FFをクリアー Uの計測値クリアー																														
CCLR	C000～C3FFをクリアー Cの計測値クリアー																														
VCLR	V000～VFFFをクリアー																														
ECLR	E000～EFFFをクリアー																														
FCLR	S020～S027をクリアー																														
JCLR	J000～JFFFをクリアー																														
QCLR	Q000～QFFFをクリアー																														
HHCLR	HH000～HH1FFをクリアー																														
使用例	 XCLR X000   HHCLR HH000																														

## 5.7 イーサネット通信応用命令

### 5.7.1 機能概要

HI-FLOWプログラムでTCP通信、UDP通信を行う場合、イーサネット通信応用命令を使用してください。

HI-FLOWシステムのイーサネット通信応用命令として、以下のインタフェースをサポートします。

命令	機能
TOP	TCPコネクションのオープン (クライアント)
TPOP	TCPコネクションのオープン (サーバー)
TCLO	TCPコネクションのクローズ
TRCV	TCP受信
TSND	TCP送信
UOP	UDPのオープン
UCLO	UDPのクローズ
URCV	UDP受信
USND	UDP送信

システム拡張演算ファンクションに従った通信の仕様を以下に示します。

項目	仕様	備考
同時使用可能ソケット数	CMU : 16	TCPの送受信、UDPの送受信の合計
	ET.NET (メイン) : 16	
	ET.NET (サブ) : 16	
	OPTET (Module 0~3) 共用 : 32	
送受信データサイズ	TCP通信 : 0~4,096バイト	
	UDP通信 : 0~1,472バイト	
ポート番号	1~65535	10000~59999の使用を推奨、60000以降はシステム予約

## 5 応用命令

---

イーサネット通信演算ファンクションを使用するには、LPUユニットの構成によって、以下に示すバージョン以降のモジュールを使用する必要があります。

構成	前提モジュール
CMUだけ	CMU (LQP520) : Ver-Rev 03-01以降
CMU+ET.NET	LPU (LQP510) : Ver-Rev 02-02以降
	CMU (LQP520) : Ver-Rev 04-00以降
	ET.NET (LQE720) : Ver-Rev 01-00以降
CMU+OPTET	LPU (LQP510) : Ver-Rev 02-02以降
	CMU (LQP520) : Ver-Rev 06-00以降
	OPTET (LQE710) : Ver-Rev 01-00以降

なお、上記Ver-Revは、S10V基本システムの「Module List」で表示される各モジュールのマイクロプログラムのVer-Revです。

イーサネット通信応用命令を実行すると、各管理番号に実行結果がシステムレジスターS9C0～S9FF, S690～S6AFに設定されます。

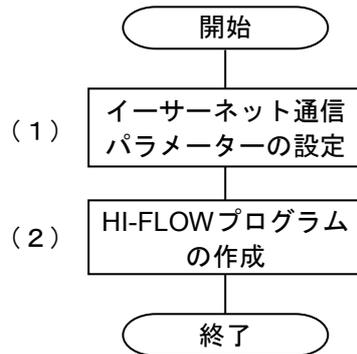
正常終了時は0、異常終了時は1が管理番号に該当するシステムレジスターに設定されます。管理番号とは、使用できるソケットに対応付けている番号のことです。

レジスター		管理番号	備考
ワード	ビット		
SW9C0	S9C0	1	CMU イーサネット通信用
	S9C1	2	
	⎵	⎵	
	S9CE	15	
	S9CF	16	
SW9D0	S9D0	17	ET.NET (メイン) イーサネット通信用
	S9D1	18	
	⎵	⎵	
	S9DE	31	
	S9DF	32	
SW9E0	S9E0	33	ET.NET (サブ) イーサネット通信用
	S9E1	34	
	⎵	⎵	
	S9EE	47	
	S9EF	48	
SW9F0	S9F0	将来用	
	⎵		
	S9FF		
SW690	S690	49	OPTET イーサネット通信用
	S691	50	
	⎵	⎵	
	S69E	63	
	S69F	64	
SW6A0	S6A0	65	OPTET イーサネット通信用
	S6A1	66	
	⎵	⎵	
	S6AE	79	
	S6AF	80	

## 5 応用命令

### 5.7.2 使用方法

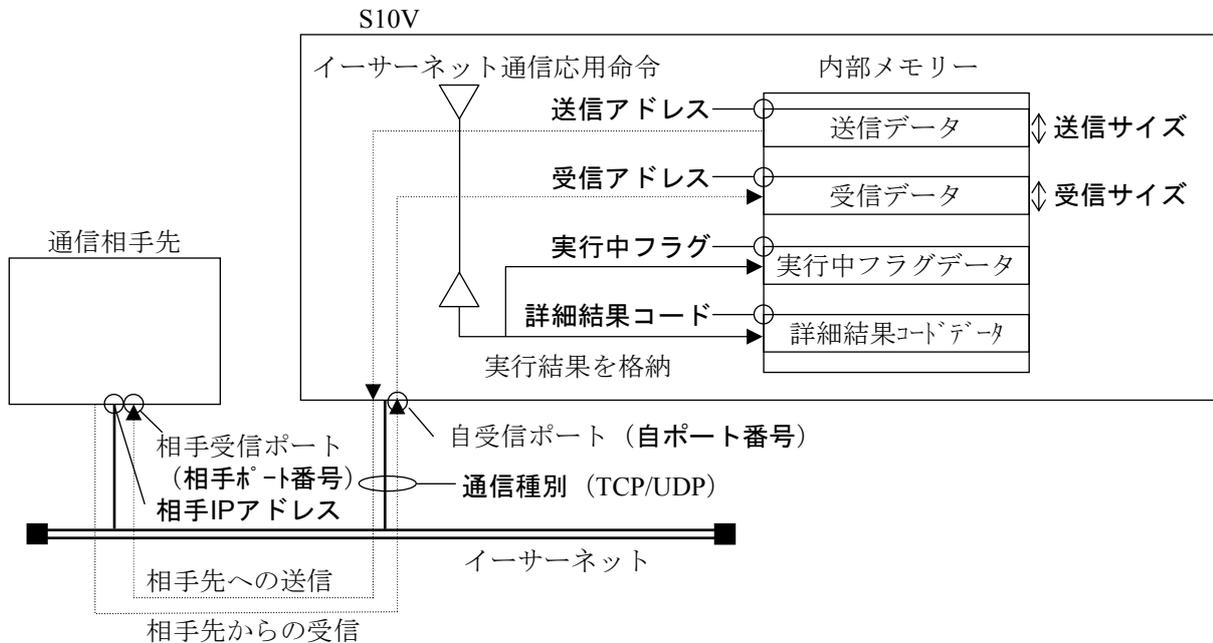
イーサネット通信応用命令は、HI-FLOWシステムの[イーサネット通信設定]画面で設定するパラメーター情報をもとに動作します。そのため、次のフローに示すように、HI-FLOWプログラム作成前に[イーサネット通信設定]画面でパラメーター情報を設定しておく必要があります。



パラメーター情報は、以下の図を参考にして設定してください。

図の中の太字となっている項目が、[イーサネット通信設定]画面で設定する情報になります。

[イーサネット通信設定]画面の設定情報の詳細については、「(1) イーサネット通信パラメーターの設定」を参照してください。

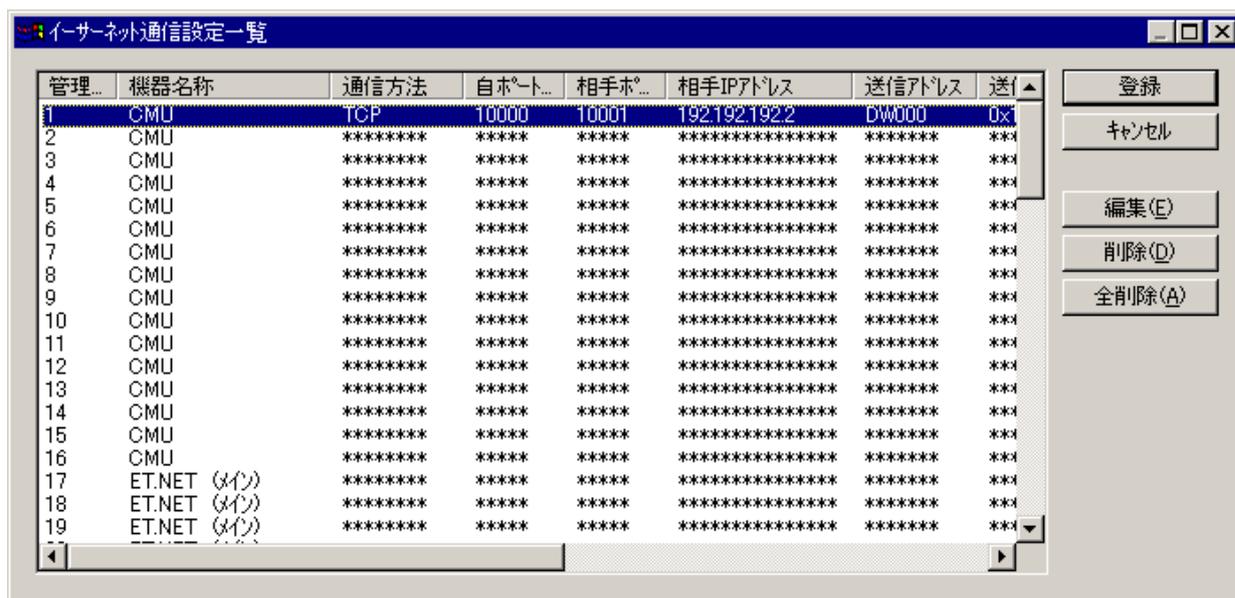


## (1) イーサネット通信パラメーターの設定

「イーサネット通信設定」画面でパラメーター情報を設定する場合、HI-FLOWプロセスシート／HI-FLOWシートから「ユーティリティ」－「イーサネット通信設定」を選択してください。

なお、「イーサネット通信設定」画面の設定は、オンライン時だけ設定できます。

以下の「イーサネット通信設定一覧」画面が表示されます。



「イーサネット通信設定一覧」画面

「イーサネット通信設定一覧」画面でパラメーター情報の行を指定し、**編集** ボタンをクリックまたは「イーサネット通信設定一覧」画面でパラメーター情報の行をダブルクリックすると、指定した行の「イーサネット通信設定」画面が表示されます。

なお、設定の詳細は「ソフトウェアマニュアル オペレーション HI-FLOW For Windows® (マニュアル番号 SVJ-3-132)」を参照してください。

## 5 応用命令

[イーサネット通信設定] 画面

以下に [イーサネット通信設定] 画面のパラメーター情報について説明します。

管理番号：

[イーサネット通信設定一覧] 画面で指定された管理番号を表示します。

機器名称：

[イーサネット通信設定一覧] 画面で指定された通信を行うモジュールを表示します。

機器名称は管理番号に対して固定で、以下のモジュールが表示されます。

管理番号49～80のOPTETを指定する場合、使用するOPTETのModule番号を指定してください。

管理番号	機器名称
1～16	CMU
17～32	ET.NET (メイン)
33～48	ET.NET (サブ)
49～80	OPTET (Module 0～3)

## 通信方法：

コンボボックスから「TCP」または「UDP」を選択してください。デフォルトは、「TCP」です。

## 自ポート番号：

通信を行うポート番号を10進数で指定してください（指定範囲は1～65535）。デフォルトは空白です（10000～59999の使用を推奨。60000以降はシステム予約）。

## 相手ポート番号：

通信相手先のポート番号を10進数で指定してください（指定範囲は1～65535）。デフォルトは空白です（10000～59999の使用を推奨。60000以降はシステム予約）。

## 相手IPアドレス：

通信相手先のIPアドレスを指定してください。デフォルトは空白です。UDP送信でブロードキャスト送信を行う場合は、255.255.255.255のようにノードアドレスを255で指定してください。

## 送信アドレス：

送信データの先頭アドレスをPI/Oのワード形式（ロングワード、フロート専用レジスタはロングワード、フロート形式）で指定してください。ビット型レジスタ、PI/Oとして割り当たっていないエリア、およびレジスタをまたがった指定はできません。デフォルトは空白です。また、送信アドレスと送信サイズから送信データの終了アドレスを計算して表示します。

## 送信サイズ：

データの送信サイズを16進数で指定してください。デフォルトは空白です。単位はバイトで、通信種別ごとに以下のサイズが指定できます。

TCP : 0x0～0x1000 (0～4,096)

UDP : 0x0～0x5C0 (0～1,472)

## 受信アドレス：

受信データを格納するエリアの先頭アドレスをPI/Oのワード形式（ロングワード、フロート専用レジスタはロングワード、フロート形式）で指定してください。ビット型レジスタ、PI/Oとして割り当たっていないエリア、およびレジスタをまたがった指定はできません。デフォルトは空白です。

また、受信アドレスと受信サイズから受信データの終了アドレスを計算して表示します。

## 5 応用命令

### 受信サイズ：

データの受信サイズを16進数で指定してください。デフォルトは空白です。単位はバイトで通信種別ごとに以下のサイズが指定できます。

TCP：0x0～0x1000（0～4,096）

UDP：0x0～0x5C0（0～1,472）

### 受信タイムアウト時間：

受信命令発行時に受信データが受信できない場合の受信データ到着待ち時間を設定してください。設定範囲は、0～100（0～10秒）で100ms単位で指定します（0は、タイムアウトなし）。デフォルトは10（1秒）です。タイムアウト時間を設定して、受信命令で受信タイムアウトが発生した場合、受信命令が受信データなし（EWOULDBLOCK）のエラーとなります。

### 実行中フラグ：

イーサネット通信応用命令の処理が実行中かどうかを示すエリアをビット型レジスターで指定してください。デフォルトは空白です。

### 詳細結果コード：

イーサネット通信応用命令の実行結果の詳細な結果コードを格納するエリアを、ロング型レジスターで指定してください。デフォルトは空白です。

### ソケット切断方法：

通信方法が「TCP」の場合だけ設定できます。コンボボックスから「未送信データ送信待ち」または「未送信データ破棄」を選択してください。デフォルトは、「未送信データ送信待ち」です。選択項目と意味を以下に示します。

未送信データ送信待ち・・・送信が未完了のデータがある場合、データが流れるまで待ち状態になります。読み取られていない受信データは破棄されます。

未送信データ破棄・・・送信が未完了のデータがある場合、データが流れるのを待たずに通信路の切断とソケットの解放を行います。この場合、相手ホストのTCPにはRSTが送信されます。通常の切断方法と異なるため相手ホストでRSTを受信した場合の動作（UPへのRST受信時の報告方法）は注意してください。読み取られていない受信データは破棄されます。

以下に [イーサネット通信設定] 画面で指定できるレジスターを示します。

<設定レジスター一覧>

(1/2)

No.	項目	シンボル	送信アドレス	受信アドレス	実行中フラグ	詳細結果コード
1	外部入力	X	○	○	○	○
2	外部出力	Y	○	○	○	○
3	内部レジスター	R	○	○	○	○
4	キープリレー	K	○	○	○	○
5	オンディレイタイマー	T	○	○	○	○
6	ワンショットタイマー	U	○	○	○	○
7	アップダウンカウンタ	C	○	○	○	○
8	グローバルリンク レジスター	G	○	○	○	○
9	ネステイングコイル	N	○	○	○	○
10	プロセスレジスター	P	○	○	○	○
11	イベントレジスター	E	○	○	○	○
12	エッジ接点	V	○	○	○	○
13	ゼットレジスター	Z	○	○	○	○
14	システムレジスター	S	○	○	○	○
15	データレジスター	DW	○	○	—	○
16	ワークレジスター	FW	○	○	—	○
17	内部レジスター	M	○	○	○	○
18	内部レジスター (ロングワード)	BD	—	—	—	—
19	高速リモートI/O入力用	I	○	○	—	○
20	高速リモートI/O出力用	O	○	○	—	○
21	HI-FLOW—ラダー間	J	○	○	○	○
22	共有データレジスター	Q	○	○	○	○
23	ワークレジスター	LB	○	○	○	○
24	ワード専用ワーク レジスター	LW	○	○	—	○

○：指定できます。

—：指定できません。

## 5 応用命令

(2/2)

No.	項目	シンボル	送信アドレス	受信アドレス	実行中フラグ	詳細結果コード
25	ロングワード専用 ワークレジスター	LL	○	○	—	○
26	単精度浮動小数点専用 ワークレジスター	LF	○	○	—	○
27	ワード専用ワーク レジスター (停電保持)	LX	○	○	—	○
28	ロングワード専用ワーク レジスター (停電保持)	LM	○	○	—	○
29	単精度浮動小数点専用ワー クレジスター (停電保持)	LG	○	○	—	○
30	ラダーコンバーター専用 ワークレジスター	LR	○	○	○	○
31	ラダーコンバーター専用 ワークレジスター (エッジ接点用)	LV	○	○	○	○

○：指定できます。

—：指定できません。

以下にイーサネット通信応用命令の詳細結果コードを示します。

< 詳細結果コード一覧 >

(1/2)

値	意味	対処方法
0	正常 (TOP, TPOP, TCLO, UOP, UCLO)	—
0~4,096	正常 (送受信データサイズ) (TRCV, TSND, URCV, USND)	—
0x80000005 (EIO)	アダプター (デバイス) に重障害発生	エラーログ情報 (*1) に記載の対処方法を参照してください。
0x8000000D (EACCES)	宛先IPアドレスにブロードキャストアドレス指定	イーサネット通信設定が異常です。設定を見直してください。
0x80000016 (EINVAL)	切断されたソケットを指定または受信バッファ長が負の値	イーサネット通信設定が異常です。設定を見直してください。
0x800000DA (EMSGSIZE)	送信データ長範囲外	イーサネット通信設定が異常です。設定を見直してください。
0x800000E2 (EADDRINUSE)	ポート番号が別ソケットで使用中	使用ポート番号を見直してください。
0x800000E3 (EADDRNOTAVAIL)	ポート番号、IPアドレス指定誤り	イーサネット通信設定が異常です。設定を見直してください。
0x800000E4 (ENETDOWN)	デバイス未初期化または停止された	エラーログ情報 (*1) に記載の対処方法を参照してください。
0x800000E5 (ENETUNREACH)	宛先IPアドレスの経路情報なし	CMUモジュール/ET.NETモジュールの経路情報設定を見直してください。 (*2)
0x800000E7 (ECONNABORTED)	コネクションの接続が打ち切られた	<ul style="list-style-type: none"> <li>・ ケーブルの配線を見直してください。</li> <li>・ 接続先ホストのプログラムを見直してください。</li> </ul>
0x800000E8 (ECONNRESET)	接続先ホストのTCPからコネクションがリセット	接続先ホストのプログラムを見直してください。
0x800000E9 (ENOBUFS)	メモリー確保失敗	エラーログ情報 (*1) に記載の対処方法を参照してください。
0x800000EB (ENOTCONN)	コネクションが確立していないソケットに送信	TOP/TPOPが実行に失敗しています。プログラムを見直してください。

(\*1) エラーログ情報を参照する方法については、「ユーザズマニュアル 基本モジュール (マニュアル番号 SVJ-1-100)」を参照してください。

(\*2) 経路情報の設定は、各モジュールの設定ツールで行ってください。

## 5 応用命令

(2/2)

値	意味	対処方法
0x800000EC (ESHUTDOWN)	他タスクからソケットが解放された	ラダー／HI-FLOWプログラムで同じ管理番号を使用していないか確認してください。
0x800000EE (ETIMEDOUT)	コネクション要求タイムアウト	・ ケーブルの配線を見直してください。 ・ 接続先ホストのプログラムを見直してください。
0x800000EF (ECONNREFUSED)	接続先ソケットが存在しない（サーバータスクが未bind）	接続先ホストのプログラムを見直してください。
0x800000F6 (EWOULDBLOCK)	受信データなし、TCPの送信ウィンドウが満杯で送信できない	プログラムを見直してください。
0x800000F9 (ENSOCK)	オープンできるソケット数オーバー	同時に使用するソケットが1モジュールあたり16以内になるよう、プログラムを見直してください。
0x80000516 (EBADF)	ソケット未オープン	TOP/TPOP/UOPが実行されていないか、実行に失敗しています。プログラムを見直してください。
0xFFFFFFFFB	イーサネットモジュールがダウン中	LPUを一度リセットしてET.NETモジュールを再立ち上げしてください。再立ち上げしても同じエラーが返る場合は、ET.NETモジュールが故障している可能性があります。モジュールを交換してください。
0xFFFFFFFFC	イーサネットモジュールが未実装	CMUモジュール／ET.NETモジュールの実装を確認してください。
0xFFFFFFFFD	タスク起動失敗	CMUモジュールがイーサネット通信応用命令に対応しているバージョンであるか確認してください。
0xFFFFFFFFE	通信識別子異常（管理番号が使用中）	同じ管理番号のイーサネット通信設定を使用しているラダー／HI-FLOWがないか確認してください。
0xFFFFFFFFF	使用種別不整合（パラメーター情報の送信方法、通信種別と異なる）	イーサネット通信設定の通信方法と、ラダー／HI-FLOWプログラムの通信種別が同一であるか確認してください。

### ・ エラー種別

0x8XXXXXXXX : CPMSソケットマクロエラー（実際のCPMSソケットマクロエラーに0x80000000を付加）

0xFXXXXXXXX : システムプログラム、タスクでのエラー

以下にイーサネット通信時のエラーログについて説明します。

管理番号ごとに、8ケースのエラーのトレースログを収集しており、基本システムの [イーサ通信エラーログ情報 (ラダー、HI-FLOW)] 画面から参照できます。 [イーサ通信エラーログ情報 (ラダー、HI-FLOW)] 画面からエラーログ情報を参照する方法については「ユーザズマニュアル 基本モジュール (マニュアル番号 SVJ-1-100)」を参照してください。

< [イーサ通信エラーログ情報 (ラダー、HI-FLOW)] 画面 >

ID	モジュール	トレース	エラーコード	内容	時刻
40	ET.NET (サフ)	RECV	0x800000E8	Connection reset by peer	07/14 15:25:47.000
40	ET.NET (サフ)	BIND	0x800000E2	Address already in use	07/14 00:43:10.000
40	ET.NET (サフ)	BIND	0x800000E2	Address already in use	07/14 00:43:10.000
40	ET.NET (サフ)	BIND	0x800000E2	Address already in use	07/14 00:43:10.000
40	ET.NET (サフ)	BIND	0x800000E2	Address already in use	07/14 00:43:10.000
40	ET.NET (サフ)	BIND	0x800000E2	Address already in use	07/14 00:43:09.000
40	ET.NET (サフ)	BIND	0x800000E2	Address already in use	07/14 00:43:09.000
40	ET.NET (サフ)	BIND	0x800000E2	Address already in use	07/14 00:43:09.000
48	ET.NET (サフ)	CONNECT	0x800000EE	Connection timed out	07/14 13:43:39.000
48	ET.NET (サフ)	CONNECT	0x800000EF	Connection refused	07/14 13:39:02.000
48	ET.NET (サフ)	CONNECT	0x800000EF	Connection refused	07/14 13:38:52.000
48	ET.NET (サフ)	CONNECT	0x800000EF	Connection refused	07/14 13:38:42.000
48	ET.NET (サフ)	CONNECT	0x800000EF	Connection refused	07/14 13:38:32.000
48	ET.NET (サフ)	SEND	0x800000E8	Connection reset by peer	07/14 13:38:22.000

閉じる

更新(R)

エラーログ削除(D)

全エラーログ削除(A)

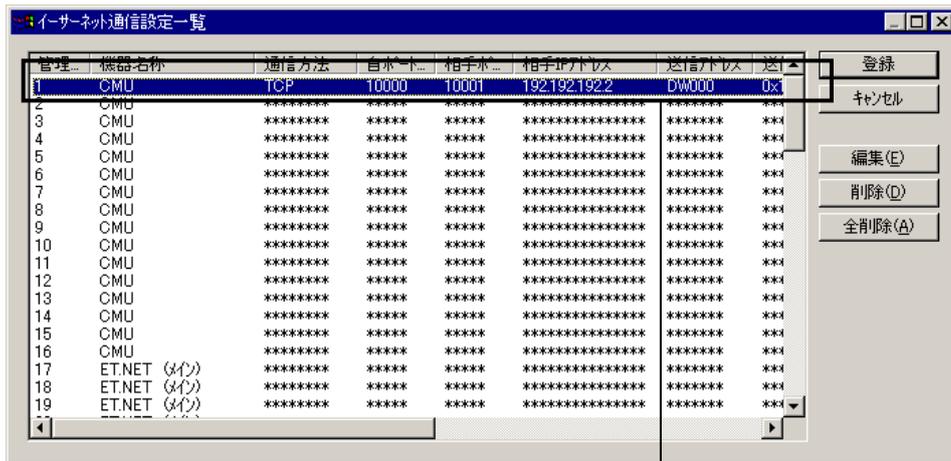
保存(V)

## 5 応用命令

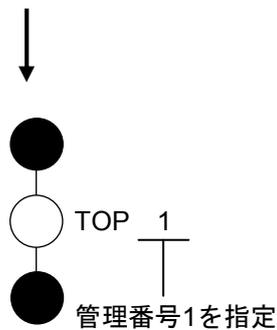
### (2) HI-FLOWプログラムの作成

イーサネット通信応用命令のパラメーターには、[イーサネット通信設定]画面で設定した管理番号だけを指定して、HI-FLOWプログラムを作成します。イーサネット通信応用命令は、管理番号から[イーサネット通信設定]画面で設定した情報に従って動作します。

<作成例>



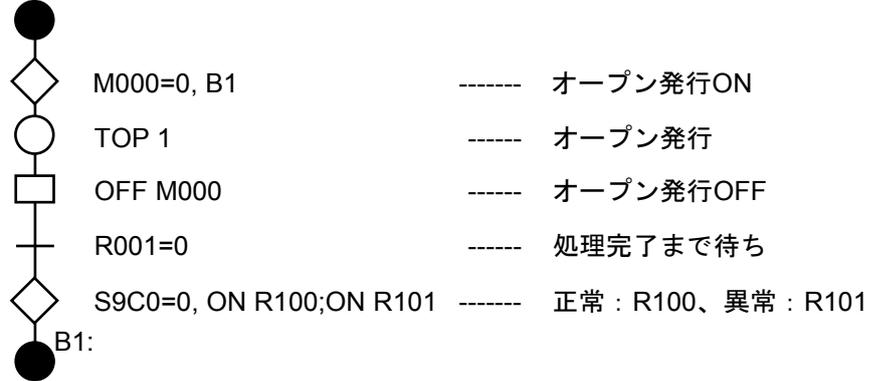
[イーサネット通信設定]画面で管理番号1の情報を設定



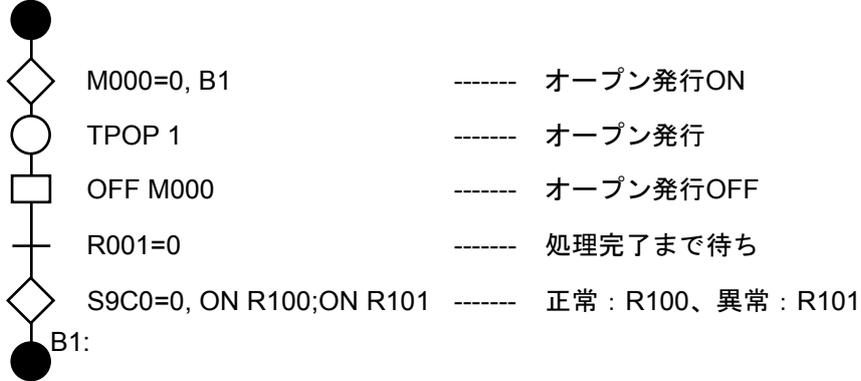
## 5.7.3 機能説明

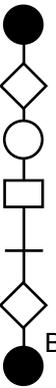
「5.6 機能説明」と同様の形式で説明します。

説明文中のシステムレジスタは、イーサネット通信応用命令の実行結果を格納するシステムレジスタ（S9C0～S9FF）のことで。

TOP	TCPコネクションのオープン（クライアント）																				
機能説明	TCPコネクションをクライアントとしてオープンします。																				
パラメーターと処理内容	 <p>S：ソース（通信識別子） モジュールによって以下に示す管理番号を指定します。</p> <table border="1" data-bbox="667 705 1173 896"> <thead> <tr> <th>管理番号</th> <th>機器名称</th> </tr> </thead> <tbody> <tr> <td>1～16</td> <td>CMU</td> </tr> <tr> <td>17～32</td> <td>ET.NET（メイン）</td> </tr> <tr> <td>33～48</td> <td>ET.NET（サブ）</td> </tr> <tr> <td>49～80</td> <td>OPTET（Module 0～3）</td> </tr> </tbody> </table> <p>TCP, UDPあわせて、1つのモジュールにつき最大16までの通信が可能です。 OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。</p>	管理番号	機器名称	1～16	CMU	17～32	ET.NET（メイン）	33～48	ET.NET（サブ）	49～80	OPTET（Module 0～3）										
管理番号	機器名称																				
1～16	CMU																				
17～32	ET.NET（メイン）																				
33～48	ET.NET（サブ）																				
49～80	OPTET（Module 0～3）																				
フラグの設定	パラメーター数、データタイプ異常が発生したとき、Eが変化。その他はOFF。																				
備考	TCPコネクションのオープン（クライアント）が成功すると、システムレジスタに0が設定され、詳細結果コードに0が設定されます。失敗した場合は、システムレジスタに1が設定され、詳細結果コードにエラーNo.（0以外の値）が設定されます。 処理が成功したか失敗したかはシステムレジスタで判定してください。 失敗時のエラー要因を知りたい場合は、詳細結果コードを参照してください。																				
使用例	<p>管理番号=1、実行中フラグ=R001でオープン</p>  <p>----- オープン発行ON ----- オープン発行 ----- オープン発行OFF ----- 処理完了まで待ち ----- 正常：R100、異常：R101</p>																				
有効パラメーター	<table border="1" data-bbox="375 1668 893 2016"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> </tbody> </table> <p>△はアドレス値が奇数時はパラメーターエラー</p>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	×	○	○	ダイレクト ロング長	×	×	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	×	×
S	ビット型 PI/O	ワード型 PI/O	定数																		
ダイレクト ワード長	×	○	○																		
ダイレクト ロング長	×	×	×																		
インダイレクト ワード長	×	△	△																		
インダイレクト ロング長	×	×	×																		

## 5 応用命令

TPOP	TCPコネクションのオープン（サーバー）																				
機能説明	TCPコネクションをクライアントとしてオープンします。																				
パラメーターと処理内容	 TPOP S <p>S：ソース（通信識別子） モジュールによって以下に示す管理番号を指定します。</p> <table border="1" data-bbox="619 546 1125 748"> <thead> <tr> <th>管理番号</th> <th>機器名称</th> </tr> </thead> <tbody> <tr> <td>1～16</td> <td>CMU</td> </tr> <tr> <td>17～32</td> <td>ET.NET（メイン）</td> </tr> <tr> <td>33～48</td> <td>ET.NET（サブ）</td> </tr> <tr> <td>49～80</td> <td>OPTET（Module 0～3）</td> </tr> </tbody> </table> <p>TCP, UDPあわせて、1つのモジュールにつき最大16までの通信が可能です。 OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。</p>	管理番号	機器名称	1～16	CMU	17～32	ET.NET（メイン）	33～48	ET.NET（サブ）	49～80	OPTET（Module 0～3）										
管理番号	機器名称																				
1～16	CMU																				
17～32	ET.NET（メイン）																				
33～48	ET.NET（サブ）																				
49～80	OPTET（Module 0～3）																				
フラグの設定	パラメーター数、データタイプ異常が発生したとき、Eが変化。その他はOFF。																				
備考	TCPコネクションのオープン（サーバー）が成功すると、システムレジスターに0が設定され、詳細結果コードに0が設定されます。失敗した場合は、システムレジスターに1が設定され、詳細結果コードにエラーNo.（0以外の値）が設定されます。 処理が成功したか失敗したかはシステムレジスターで判定してください。 失敗時のエラー要因を知りたい場合は、詳細結果コードを参照してください。																				
使用例	<p>管理番号=1、実行中フラグ=R001でオープン</p>  <p>  M000=0, B1 ----- オープン発行ON   TPOP 1 ----- オープン発行   OFF M000 ----- オープン発行OFF   R001=0 ----- 処理完了まで待ち   S9C0=0, ON R100; ON R101 ----- 正常：R100、異常：R101   B1: </p>																				
有効パラメーター	<table border="1" data-bbox="331 1585 847 1957"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> </tbody> </table> <p>△はアドレス値が奇数時はパラメーターエラー</p>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	×	○	○	ダイレクト ロング長	×	×	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	×	×
S	ビット型 PI/O	ワード型 PI/O	定数																		
ダイレクト ワード長	×	○	○																		
ダイレクト ロング長	×	×	×																		
インダイレクト ワード長	×	△	△																		
インダイレクト ロング長	×	×	×																		

TCLO	TCPコネクションのクローズ																				
機能説明	TCPコネクションをクローズします。																				
パラメーターと処理内容	 TCLO S S: ソース (通信識別子) モジュールによって以下に示す管理番号を指定します。 <table border="1" data-bbox="667 546 1166 748"> <thead> <tr> <th>管理番号</th> <th>機器名称</th> </tr> </thead> <tbody> <tr> <td>1~16</td> <td>CMU</td> </tr> <tr> <td>17~32</td> <td>ET.NET (メイン)</td> </tr> <tr> <td>33~48</td> <td>ET.NET (サブ)</td> </tr> <tr> <td>49~80</td> <td>OPTET (Module 0~3)</td> </tr> </tbody> </table> TCP, UDPあわせて、1つのモジュールにつき最大16までの通信が可能です。 OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。	管理番号	機器名称	1~16	CMU	17~32	ET.NET (メイン)	33~48	ET.NET (サブ)	49~80	OPTET (Module 0~3)										
管理番号	機器名称																				
1~16	CMU																				
17~32	ET.NET (メイン)																				
33~48	ET.NET (サブ)																				
49~80	OPTET (Module 0~3)																				
フラグの設定	パラメーター数、データタイプ異常が発生したとき、Eが変化。その他はOFF。																				
備考	TCPコネクションのクローズが成功すると、システムレジスターに0が設定され、詳細結果コードに0が設定されます。失敗した場合は、システムレジスターに1が設定され、詳細結果コードにエラーNo. (0以外の値) が設定されます。 処理が成功したか失敗したかはシステムレジスターで判定してください。 失敗時のエラー要因を知りたい場合は、詳細結果コードを参照してください。																				
使用例	<b>管理番号=1、実行中フラグ=R001でオープン</b>  <table border="0" data-bbox="494 1209 1276 1478"> <tr> <td>M000=0, B1</td> <td>-----</td> <td>クローズ発行ON</td> </tr> <tr> <td>TCLO 1</td> <td>-----</td> <td>クローズ発行</td> </tr> <tr> <td>OFF M000</td> <td>-----</td> <td>クローズ発行OFF</td> </tr> <tr> <td>R001=0</td> <td>-----</td> <td>処理完了まで待ち</td> </tr> <tr> <td>S9C0=0, ON R100; ON R101</td> <td>-----</td> <td>正常 : R100、異常 : R101</td> </tr> </table>	M000=0, B1	-----	クローズ発行ON	TCLO 1	-----	クローズ発行	OFF M000	-----	クローズ発行OFF	R001=0	-----	処理完了まで待ち	S9C0=0, ON R100; ON R101	-----	正常 : R100、異常 : R101					
M000=0, B1	-----	クローズ発行ON																			
TCLO 1	-----	クローズ発行																			
OFF M000	-----	クローズ発行OFF																			
R001=0	-----	処理完了まで待ち																			
S9C0=0, ON R100; ON R101	-----	正常 : R100、異常 : R101																			
有効パラメーター	<table border="1" data-bbox="375 1585 893 1960"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> </tbody> </table> △はアドレス値が奇数時はパラメーターエラー	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	×	○	○	ダイレクト ロング長	×	×	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	×	×
S	ビット型 PI/O	ワード型 PI/O	定数																		
ダイレクト ワード長	×	○	○																		
ダイレクト ロング長	×	×	×																		
インダイレクト ワード長	×	△	△																		
インダイレクト ロング長	×	×	×																		

## 5 応用命令

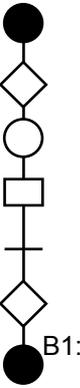
(1/2)

TRCV	TCP受信										
機能説明	イーサネット通信設定をもとにデータを受信します。										
パラメーターと処理内容	<p>  TRCV S         </p> <p>S : ソース (通信識別子)</p> <p>モジュールによって以下に示す管理番号を指定します。</p> <table border="1" data-bbox="620 636 1121 848"> <thead> <tr> <th>管理番号</th> <th>機器名称</th> </tr> </thead> <tbody> <tr> <td>1~16</td> <td>CMU</td> </tr> <tr> <td>17~32</td> <td>ET.NET (メイン)</td> </tr> <tr> <td>33~48</td> <td>ET.NET (サブ)</td> </tr> <tr> <td>49~80</td> <td>OPTET (Module 0~3)</td> </tr> </tbody> </table> <p>TCP, UDPあわせて、1つのモジュールにつき最大16までの通信が可能です。 OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。</p>	管理番号	機器名称	1~16	CMU	17~32	ET.NET (メイン)	33~48	ET.NET (サブ)	49~80	OPTET (Module 0~3)
管理番号	機器名称										
1~16	CMU										
17~32	ET.NET (メイン)										
33~48	ET.NET (サブ)										
49~80	OPTET (Module 0~3)										
フラグの設定	パラメーター数、データタイプ異常が発生したとき、Eが変化。その他はOFF。										
備考	<p>受信に成功すると、システムレジスターに0が設定され、詳細結果コードに受信データサイズが設定されます。受信に失敗した場合、システムレジスターに1が設定され、詳細結果コードにエラーNo. (負の値) が設定されます。処理が成功したか失敗したかはシステムレジスターで判定してください。</p> <p>失敗時のエラー要因を知りたい場合は、詳細結果コードを参照してください。なお、詳細結果コードがEWOULDBLOCKの場合、TRCVの再発行が可能です。詳細結果コードがEWOULDBLOCKで、受信を継続したい場合は、TRCVを再発行してください。</p>										

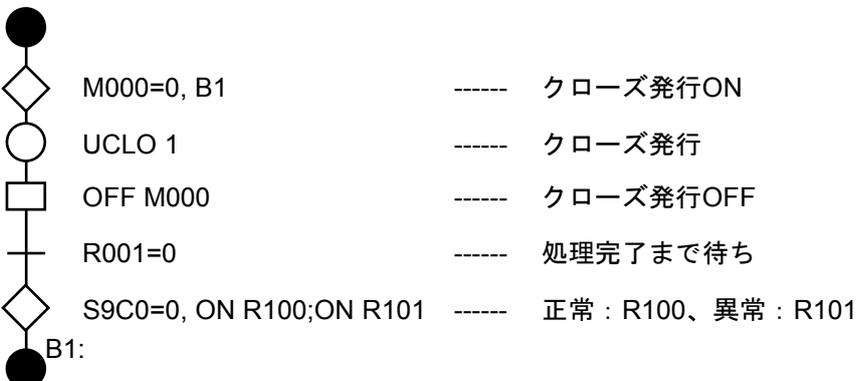
<p>使用例</p>	<p>管理番号=1、処理完了フラグ=R001、詳細結果コード=LWL0000、受信サイズ1,024バイトで受信。 受信データがないときは受信リトライ。</p> <p>●</p> <p>◇ M000=0, B1 ----- 受信発行ON</p> <p>∧ CN000 (0, 1, 0) ----- 無限ループ</p> <p>○ TRCV 1 ----- 受信発行</p> <p>— R001=0 ----- 処理完了まで待ち</p> <p>◇ S9C0, RERR ----- 異常終了（詳細結果コード判定へジャンプ）</p> <p>○ EQU [LWW0000], [1024], R104 ----- 正常終了（受信サイズ=受信データ）</p> <p>○ NEQ [LWW0000], [1024], R106 ----- 正常終了（受信サイズ&gt;受信データ）</p> <p>◇ R104   R106, OFF M000 ----- 受信発行OFF</p> <p>→ REND ----- 受信終了判定へジャンプ</p> <p>— RERR: ----- 詳細結果コード判定処理ラベル</p> <p>○ EQU [LWW0000], H800000F6, R003 ----- EWOULDBLOCKか判定</p> <p>◇ R003=0, OFF M000:ON R103 ----- 異常終了時受信終了、EWOULDBLOCK時受信リトライ</p> <p>— REND: ----- 受信終了判定処理ラベル</p> <p>◇ M000=0, B1 ----- 受信発行OFF時受信処理終了</p> <p>● B1:</p>																				
<p>有効 パラメーター</p> <p>△はアドレス 値が奇数時は パラメーター エラー</p>	<table border="1"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定 数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定 数	ダイレクト ワード長	×	○	○	ダイレクト ロング長	×	×	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	×	×
S	ビット型 PI/O	ワード型 PI/O	定 数																		
ダイレクト ワード長	×	○	○																		
ダイレクト ロング長	×	×	×																		
インダイレクト ワード長	×	△	△																		
インダイレクト ロング長	×	×	×																		

## 5 応用命令

TSND	TCP送信																								
機能説明	イーサネット通信設定をもとにデータを送信します。																								
パラメーターと処理内容	 TSND S S : ソース (通信識別子) モジュールによって以下に示す管理番号を指定します。 <table border="1" data-bbox="620 546 1121 748" style="margin-left: 40px;"> <thead> <tr> <th>管理番号</th> <th>機器名称</th> </tr> </thead> <tbody> <tr> <td>1~16</td> <td>CMU</td> </tr> <tr> <td>17~32</td> <td>ET.NET (メイン)</td> </tr> <tr> <td>33~48</td> <td>ET.NET (サブ)</td> </tr> <tr> <td>49~80</td> <td>OPTET (Module 0~3)</td> </tr> </tbody> </table> TCP, UDPあわせて、1つのモジュールにつき最大16までの通信が可能です。 OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。	管理番号	機器名称	1~16	CMU	17~32	ET.NET (メイン)	33~48	ET.NET (サブ)	49~80	OPTET (Module 0~3)														
管理番号	機器名称																								
1~16	CMU																								
17~32	ET.NET (メイン)																								
33~48	ET.NET (サブ)																								
49~80	OPTET (Module 0~3)																								
フラグの設定	パラメーター数、データタイプ異常が発生したとき、Eが変化。その他はOFF。																								
備考	送信に成功すると、システムレジスターに0が設定され、詳細結果コードに0が設定されます。失敗した場合は、システムレジスターに1が設定され、詳細結果コードにエラーNo. (0以外の値) が設定されます。 処理が成功したか失敗したかはシステムレジスターで判定してください。 失敗時のエラー要因を知りたい場合は、詳細結果コードを参照してください。																								
使用例	<b>管理番号=1、実行中フラグ=R001でオープン</b> 																								
有効パラメーター	<table border="1" data-bbox="331 1585 847 1955"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>△はアドレス値が奇数時はパラメーターエラー</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトワード長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	△はアドレス値が奇数時はパラメーターエラー	×	○	○	ダイレクトワード長	×	○	○	ダイレクトロング長	×	×	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	×	×
S	ビット型 PI/O	ワード型 PI/O	定数																						
△はアドレス値が奇数時はパラメーターエラー	×	○	○																						
ダイレクトワード長	×	○	○																						
ダイレクトロング長	×	×	×																						
インダイレクトワード長	×	△	△																						
インダイレクトロング長	×	×	×																						

UOP	UDPのオープン																				
機能説明	UDPをオープンします。																				
パラメーターと処理内容	 UOP S S : ソース (通信識別子) モジュールによって以下に示す管理番号を指定します。 <table border="1" data-bbox="667 546 1166 748"> <thead> <tr> <th>管理番号</th> <th>機器名称</th> </tr> </thead> <tbody> <tr> <td>1~16</td> <td>CMU</td> </tr> <tr> <td>17~32</td> <td>ET.NET (メイン)</td> </tr> <tr> <td>33~48</td> <td>ET.NET (サブ)</td> </tr> <tr> <td>49~80</td> <td>OPTET (Module 0~3)</td> </tr> </tbody> </table> TCP, UDPあわせて、1つのモジュールにつき最大16までの通信が可能です。 OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。	管理番号	機器名称	1~16	CMU	17~32	ET.NET (メイン)	33~48	ET.NET (サブ)	49~80	OPTET (Module 0~3)										
管理番号	機器名称																				
1~16	CMU																				
17~32	ET.NET (メイン)																				
33~48	ET.NET (サブ)																				
49~80	OPTET (Module 0~3)																				
フラグの設定	パラメーター数、データタイプ異常が発生したとき、Eが変化。その他はOFF。																				
備考	UDPのオープンが成功すると、システムレジスターに0が設定され、詳細結果コードに0が設定されます。失敗した場合は、システムレジスターに1が設定され、詳細結果コードにエラーNo. (0以外の値) が設定されます。 処理が成功したか失敗したかはシステムレジスターで判定してください。 失敗時のエラー要因を知りたい場合は、詳細結果コードを参照してください。																				
使用例	<b>管理番号=1、実行中フラグ=R001でオープン</b>  <table border="0" data-bbox="502 1209 1276 1467"> <tr> <td>M000=0, B1</td> <td>-----</td> <td>オープン発行ON</td> </tr> <tr> <td>UOP 1</td> <td>-----</td> <td>オープン発行</td> </tr> <tr> <td>OFF M000</td> <td>-----</td> <td>オープン発行OFF</td> </tr> <tr> <td>R001=0</td> <td>-----</td> <td>処理完了まで待ち</td> </tr> <tr> <td>S9C0=0, ON R100; ON R101</td> <td>-----</td> <td>正常 : R100、異常 : R101</td> </tr> </table>	M000=0, B1	-----	オープン発行ON	UOP 1	-----	オープン発行	OFF M000	-----	オープン発行OFF	R001=0	-----	処理完了まで待ち	S9C0=0, ON R100; ON R101	-----	正常 : R100、異常 : R101					
M000=0, B1	-----	オープン発行ON																			
UOP 1	-----	オープン発行																			
OFF M000	-----	オープン発行OFF																			
R001=0	-----	処理完了まで待ち																			
S9C0=0, ON R100; ON R101	-----	正常 : R100、異常 : R101																			
有効パラメーター	<table border="1" data-bbox="375 1585 893 1960"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> </tbody> </table> △はアドレス値が奇数時はパラメーターエラー	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	×	○	○	ダイレクト ロング長	×	×	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	×	×
S	ビット型 PI/O	ワード型 PI/O	定数																		
ダイレクト ワード長	×	○	○																		
ダイレクト ロング長	×	×	×																		
インダイレクト ワード長	×	△	△																		
インダイレクト ロング長	×	×	×																		

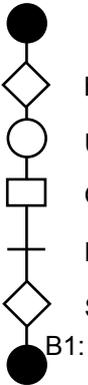
## 5 応用命令

UCLO	UDPのクローズ																				
機能説明	UDPをクローズします。																				
パラメーターと処理内容	 UCLO S S : ソース (通信識別子) モジュールによって以下に示す管理番号を指定します。 <table border="1" data-bbox="620 546 1121 757"> <thead> <tr> <th>管理番号</th> <th>機器名称</th> </tr> </thead> <tbody> <tr> <td>1~16</td> <td>CMU</td> </tr> <tr> <td>17~32</td> <td>ET.NET (メイン)</td> </tr> <tr> <td>33~48</td> <td>ET.NET (サブ)</td> </tr> <tr> <td>49~80</td> <td>OPTET (Module 0~3)</td> </tr> </tbody> </table> TCP, UDPあわせて、1つのモジュールにつき最大16までの通信が可能です。 OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。	管理番号	機器名称	1~16	CMU	17~32	ET.NET (メイン)	33~48	ET.NET (サブ)	49~80	OPTET (Module 0~3)										
管理番号	機器名称																				
1~16	CMU																				
17~32	ET.NET (メイン)																				
33~48	ET.NET (サブ)																				
49~80	OPTET (Module 0~3)																				
フラグの設定	パラメーター数、データタイプ異常が発生したとき、Eが変化。その他はOFF。																				
備考	UDPのクローズが成功すると、システムレジスターに0が設定され、詳細結果コードに0が設定されます。失敗した場合は、システムレジスターに1が設定され、詳細結果コードにエラーNo. (0以外の値) が設定されます。 処理が成功したか失敗したかはシステムレジスターで判定してください。 失敗時のエラー要因を知りたい場合は、詳細結果コードを参照してください。																				
使用例	<b>管理番号=1、実行中フラグ=R001でオープン</b> 																				
有効パラメーター	<table border="1" data-bbox="331 1592 847 1966"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> </tbody> </table> △はアドレス値が奇数時はパラメーターエラー	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	×	○	○	ダイレクト ロング長	×	×	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	×	×
S	ビット型 PI/O	ワード型 PI/O	定数																		
ダイレクト ワード長	×	○	○																		
ダイレクト ロング長	×	×	×																		
インダイレクト ワード長	×	△	△																		
インダイレクト ロング長	×	×	×																		

(1/2)

URCV	UDP受信										
機能説明	イーサネット通信設定をもとにデータを受信します。										
パラメーターと処理内容	<p>○ URCV S</p> <p>S : ソース (通信識別子)</p> <p>モジュールによって以下に示す管理番号を指定します。</p> <table border="1"> <thead> <tr> <th>管理番号</th> <th>機器名称</th> </tr> </thead> <tbody> <tr> <td>1~16</td> <td>CMU</td> </tr> <tr> <td>17~32</td> <td>ET.NET (メイン)</td> </tr> <tr> <td>33~48</td> <td>ET.NET (サブ)</td> </tr> <tr> <td>49~80</td> <td>OPTET (Module 0~3)</td> </tr> </tbody> </table> <p>TCP, UDPあわせて、1つのモジュールにつき最大16までの通信が可能です。 OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。</p>	管理番号	機器名称	1~16	CMU	17~32	ET.NET (メイン)	33~48	ET.NET (サブ)	49~80	OPTET (Module 0~3)
管理番号	機器名称										
1~16	CMU										
17~32	ET.NET (メイン)										
33~48	ET.NET (サブ)										
49~80	OPTET (Module 0~3)										
フラグの設定	パラメーター数、データタイプ異常が発生したとき、Eが変化。その他はOFF。										
備考	<p>受信に成功すると、システムレジスターに0が設定され、詳細結果コードに受信データサイズが設定されます。受信に失敗した場合、システムレジスターに1が設定され、詳細結果コードにエラーNo. (負の値) が設定されます。処理が成功したか失敗したかはシステムレジスターで判定してください。</p> <p>失敗時のエラー要因を知りたい場合は、詳細結果コードを参照してください。なお、詳細結果コードがEWOULDBLOCKの場合、URCVを再発行できません。詳細結果コードがEWOULDBLOCKで、受信を継続したい場合は、URCVを再発行してください。</p>										

<p>使用例</p>	<p>管理番号=1、処理完了フラグ=R001、詳細結果コード=LWL0000、受信サイズ1,024バイトで受信。 受信データがないときは受信リトライ。</p> <p>●</p> <p>◇ M000=0, B1 ----- 受信発行ON</p> <p>∩ CN000 (0, 1, 0) ----- 無限ループ</p> <p>○ URCV 1 ----- 受信発行</p> <p>— R001=0 ----- 処理完了まで待ち</p> <p>◇ S9C0, RERR ----- 異常終了（詳細結果コード判定へジャンプ）</p> <p>○ EQU [LWW0000], [1024], R104 ----- 正常終了（受信サイズ=受信データ）</p> <p>○ NEQ [LWW0000], [1024], R106 ----- 正常終了（受信サイズ&gt;受信データ）</p> <p>◇ R104   R106, OFF M000 ----- 受信発行OFF</p> <p>→ REND ----- 受信終了判定へジャンプ</p> <p>RERR: ----- 詳細結果コード判定処理ラベル</p> <p>○ EQU [LWW0000], H800000F6, R003 ----- EWOULDBLOCKか判定</p> <p>◇ R003=0, OFF M000:ON R103 ----- 異常終了時受信終了、EWOULDBLOCK時受信リトライ</p> <p>REND: ----- 受信終了判定処理ラベル</p> <p>◇ M000=0, B1 ----- 受信発行OFF時受信処理終了</p> <p>∩ B1: ●</p>																				
<p>有効 パラメーター</p> <p>△はアドレス 値が奇数時は パラメーター エラー</p>	<table border="1"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	×	○	○	ダイレクト ロング長	×	×	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	×	×
S	ビット型 PI/O	ワード型 PI/O	定数																		
ダイレクト ワード長	×	○	○																		
ダイレクト ロング長	×	×	×																		
インダイレクト ワード長	×	△	△																		
インダイレクト ロング長	×	×	×																		

USND	UDP送信																				
機能説明	イーサネット通信設定をもとにデータを送信します。																				
パラメーターと処理内容	 USND S S: ソース (通信識別子) モジュールによって以下に示す管理番号を指定します。 <table border="1" data-bbox="667 546 1166 748"> <thead> <tr> <th>管理番号</th> <th>機器名称</th> </tr> </thead> <tbody> <tr> <td>1~16</td> <td>CMU</td> </tr> <tr> <td>17~32</td> <td>ET.NET (メイン)</td> </tr> <tr> <td>33~48</td> <td>ET.NET (サブ)</td> </tr> <tr> <td>49~80</td> <td>OPTET (Module 0~3)</td> </tr> </tbody> </table> TCP, UDPあわせて、1つのモジュールにつき最大16までの通信が可能です。 OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。	管理番号	機器名称	1~16	CMU	17~32	ET.NET (メイン)	33~48	ET.NET (サブ)	49~80	OPTET (Module 0~3)										
管理番号	機器名称																				
1~16	CMU																				
17~32	ET.NET (メイン)																				
33~48	ET.NET (サブ)																				
49~80	OPTET (Module 0~3)																				
フラグの設定	パラメーター数、データタイプ異常が発生したとき、Eが変化。その他はOFF。																				
備考	送信に成功すると、システムレジスターに0が設定され、詳細結果コードに0が設定されます。失敗した場合は、システムレジスターに1が設定され、詳細結果コードにエラーNo. (0以外の値) が設定されます。 処理が成功したか失敗したかはシステムレジスターで判定してください。 失敗時のエラー要因を知りたい場合は、詳細結果コードを参照してください。																				
使用例	<b>管理番号=1、実行中フラグ=R001でオープン</b>  <table border="0" data-bbox="502 1220 1276 1473"> <tr> <td>M000=0, B1</td> <td>-----</td> <td>送信発行ON</td> </tr> <tr> <td>USND 1</td> <td>-----</td> <td>送信発行</td> </tr> <tr> <td>OFF M000</td> <td>-----</td> <td>送信発行OFF</td> </tr> <tr> <td>R001=0</td> <td>-----</td> <td>処理完了まで待ち</td> </tr> <tr> <td>S9C0=0, ON R100; ON R101</td> <td>-----</td> <td>正常 : R100、異常 : R101</td> </tr> </table>	M000=0, B1	-----	送信発行ON	USND 1	-----	送信発行	OFF M000	-----	送信発行OFF	R001=0	-----	処理完了まで待ち	S9C0=0, ON R100; ON R101	-----	正常 : R100、異常 : R101					
M000=0, B1	-----	送信発行ON																			
USND 1	-----	送信発行																			
OFF M000	-----	送信発行OFF																			
R001=0	-----	処理完了まで待ち																			
S9C0=0, ON R100; ON R101	-----	正常 : R100、異常 : R101																			
有効パラメーター	<table border="1" data-bbox="376 1585 893 1957"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>×</td> <td>×</td> </tr> </tbody> </table> △はアドレス値が奇数時はパラメーターエラー	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	×	○	○	ダイレクト ロング長	×	×	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	×	×
S	ビット型 PI/O	ワード型 PI/O	定数																		
ダイレクト ワード長	×	○	○																		
ダイレクト ロング長	×	×	×																		
インダイレクト ワード長	×	△	△																		
インダイレクト ロング長	×	×	×																		

## 5 応用命令

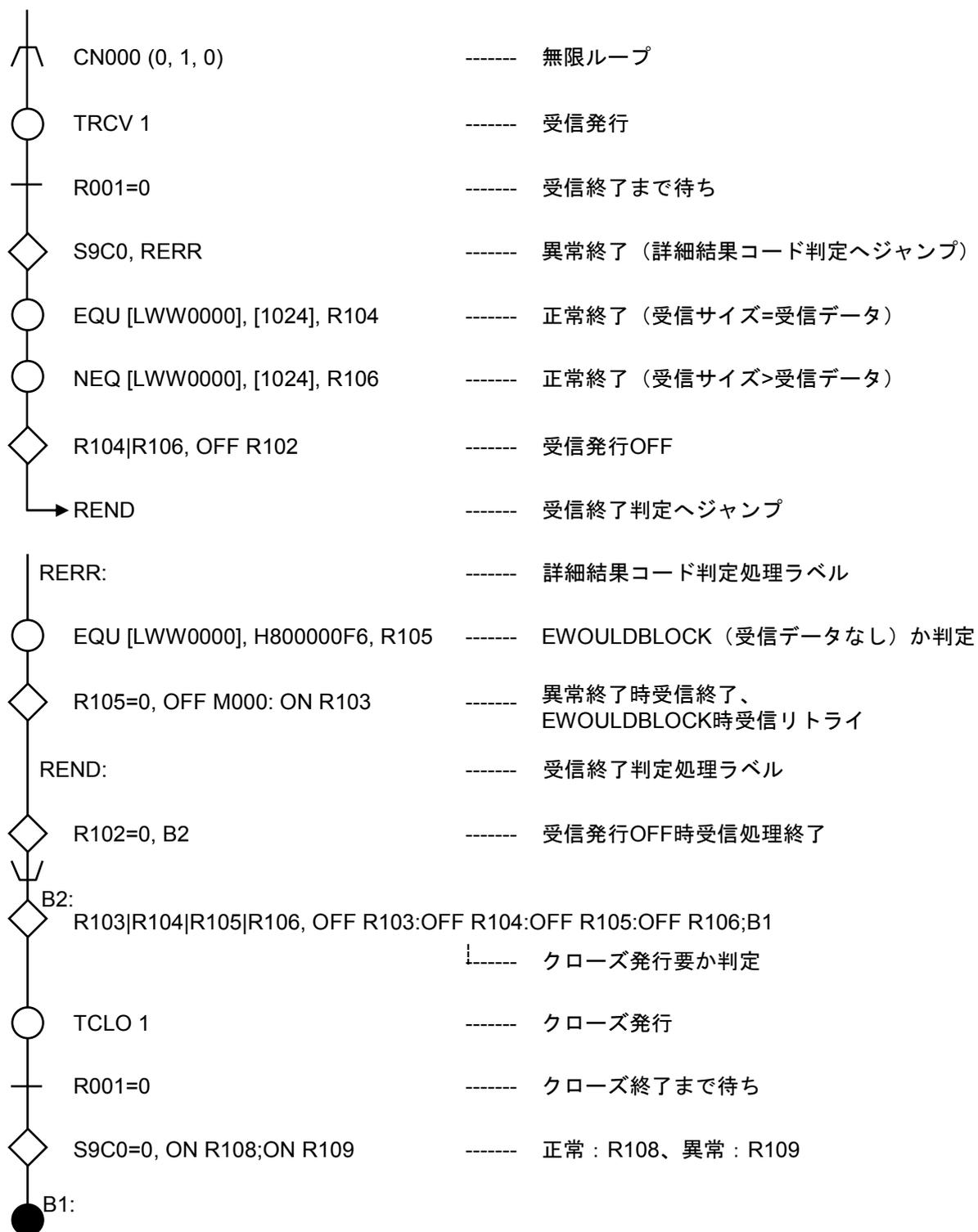
### 5.7.4 サンプルプログラム

イーサネット通信応用命令を使用して、ソケットのオープン、送受信、ソケットがクローズするまでのサンプルプログラムを記載します。

このサンプルプログラムは、パラメーター設定情報に管理番号=1、実行中フラグ=R001、詳細結果コード=LWL0000を設定したものです。

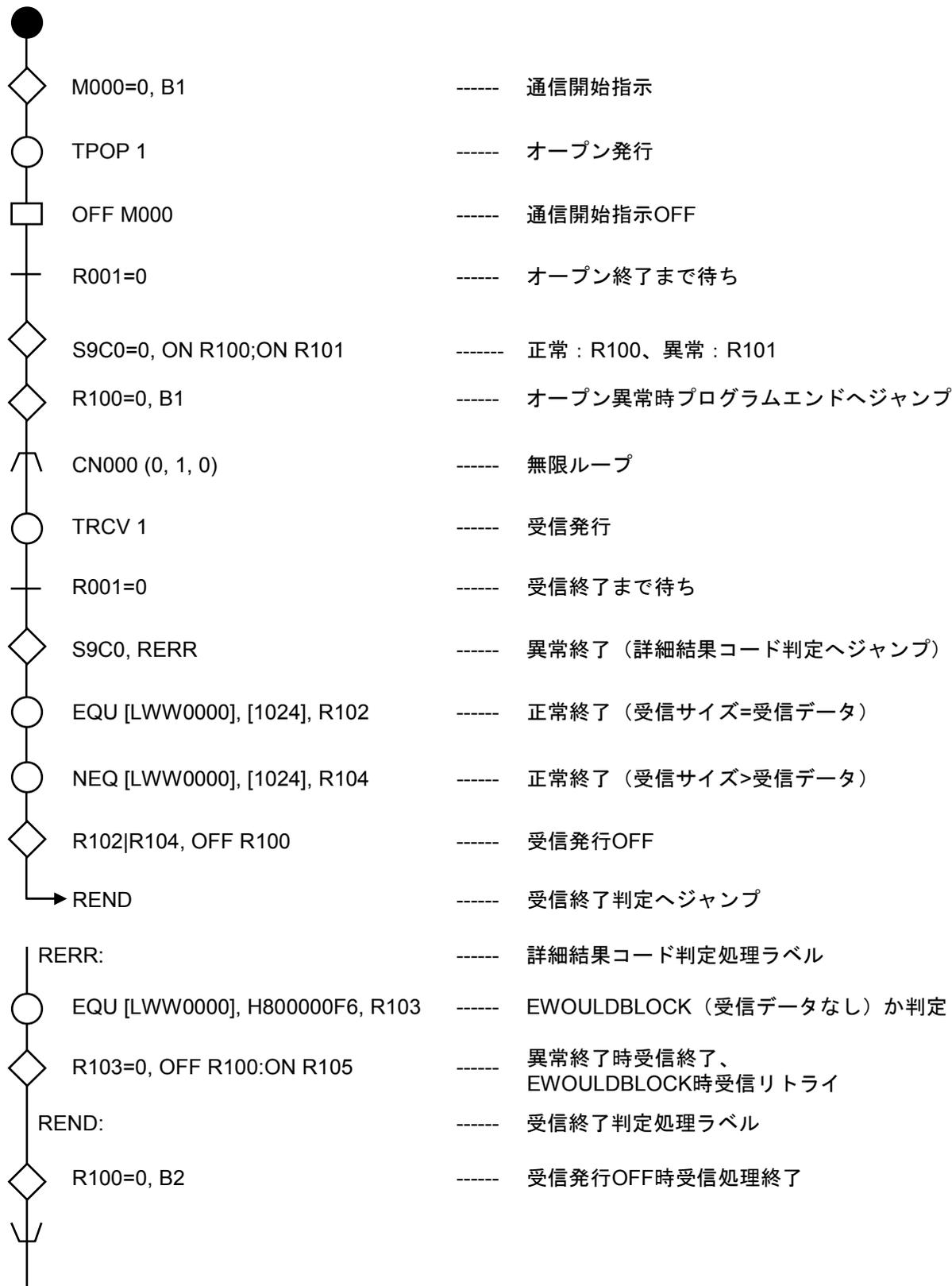
#### (1) TCPクライアント

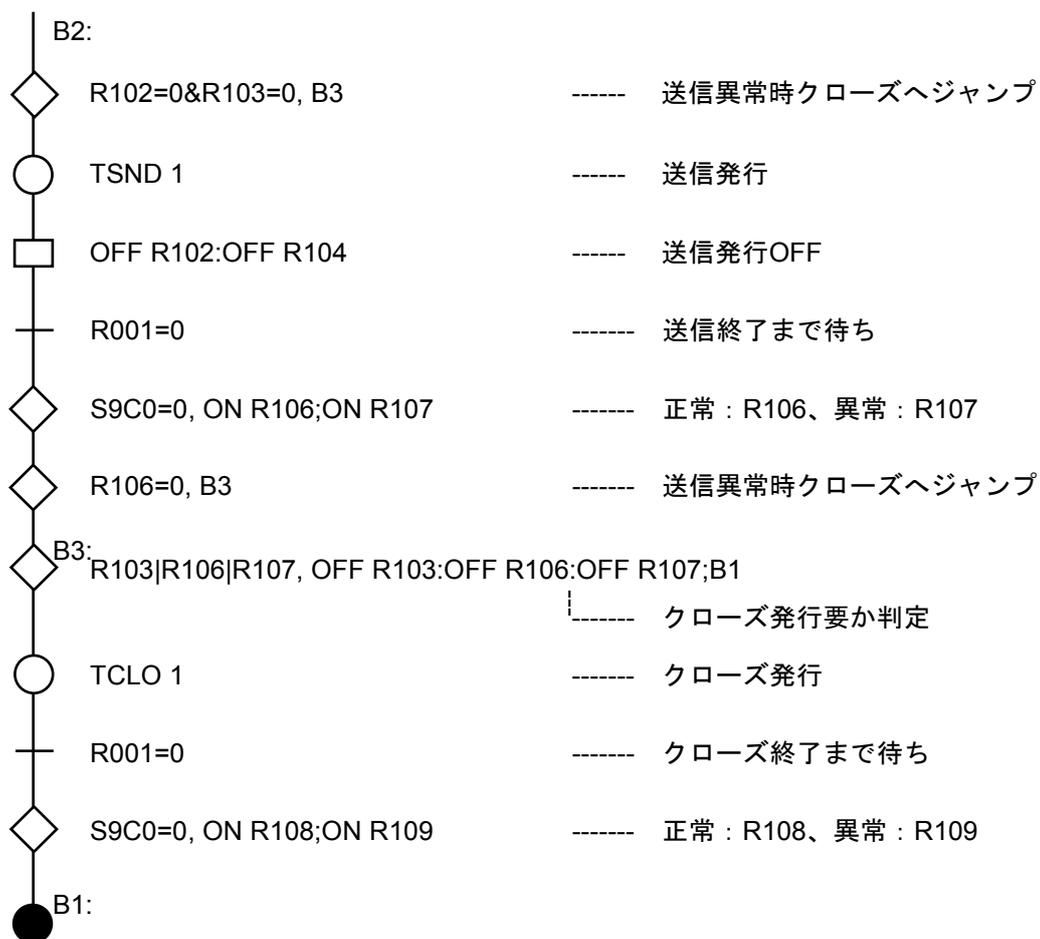




## 5 応用命令

### (2) TCPサーバー





このページは白紙です。

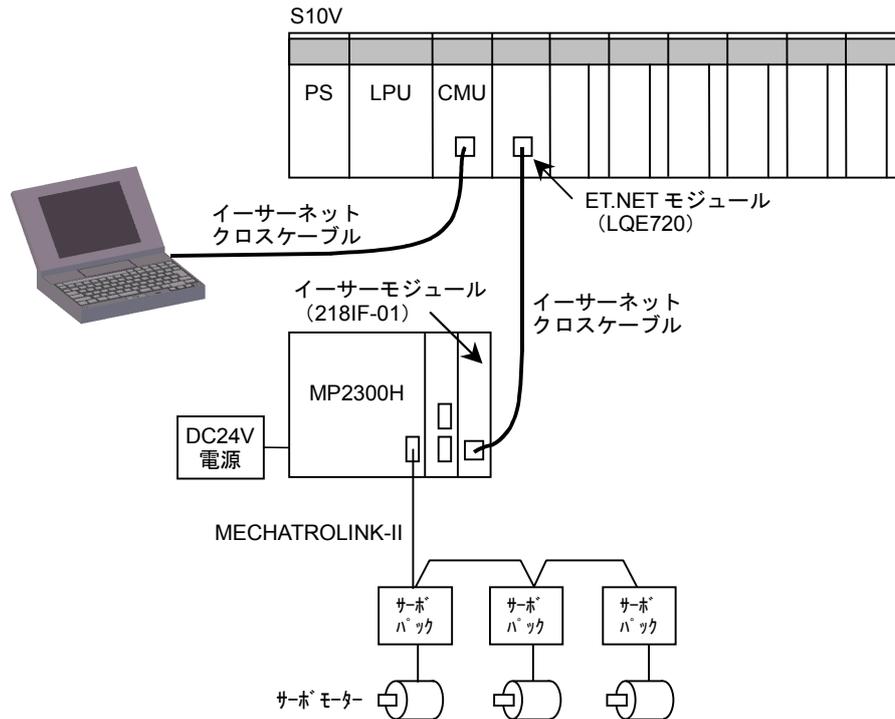
## 6 モーション制御命令

## 6 モーション制御命令

### 6.1 用途

モーション制御命令は、S10Vと株式会社安川電機製モーションコントローラーMP2300H（以降MP2300Hと略します）を接続し、HI-FLOWから直接サーボモーターを制御する機能です。

モーション制御命令はWindows® 7（32bit）、Windows® 10（32bit）非サポートです。



HI-FLOWから制御できる最大軸数は32軸です。

## 6.2 仕様

### 6.2.1 システム構成

モーション制御命令を使用するには、以下に示すバージョン以降のモジュールを使用する必要があります。

モジュール名称	型式	Ver-Rev
LPU	LQP510	02-02以降
CMU	LQP520	04-00以降
ET.NET	LQE720	01-00以降

なお、上記Ver-Revは、S10V基本システムの「Module List」で表示される各モジュールのマイクロプログラムのVer-Revです。

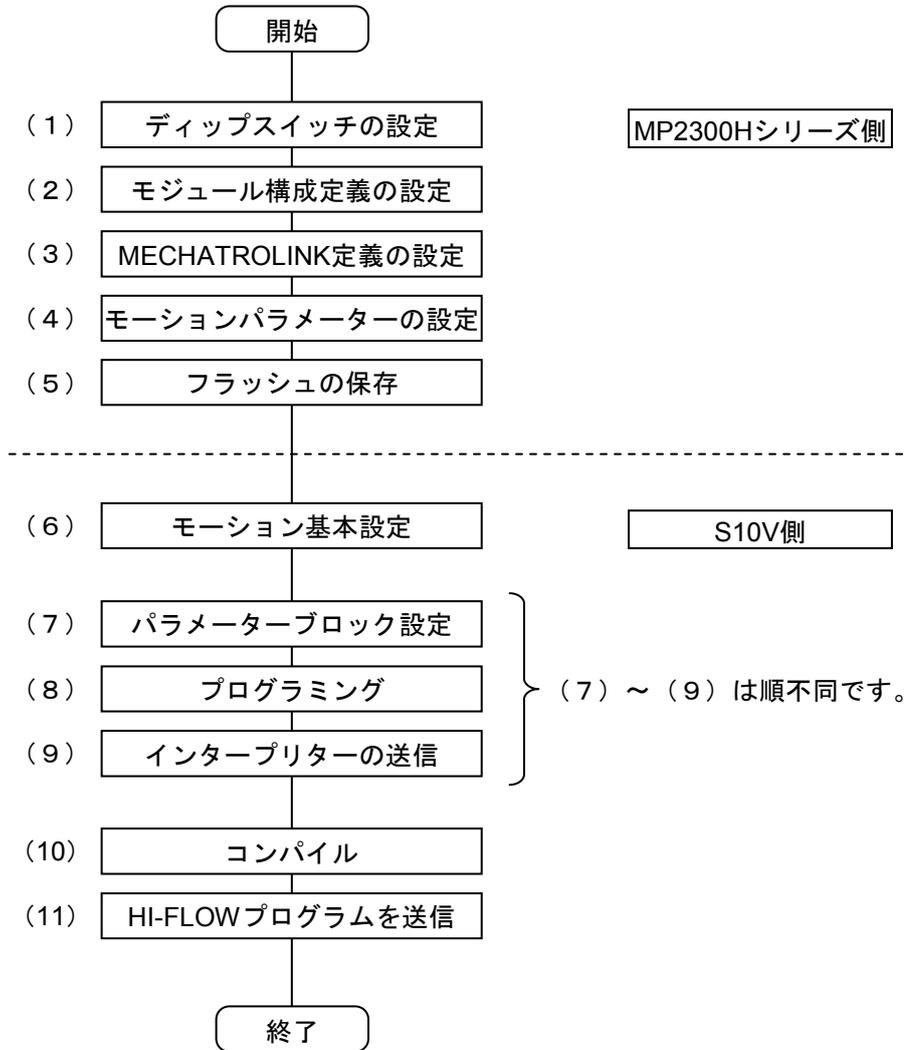
### 6.2.2 S10VとMP2300H間通信インタフェース

項目	仕様
通信形態	1対1通信 (S10VとMP2300H間)
(株)安川電機対象モーションコントローラー	Ethernet®インタフェースユニットを搭載したモーションコントローラー (MP2300H)
使用ポート	34200, 34201, 34205 (10進数) (S10V, MP2300H共)
使用IPアドレス	任意。ただし、S10VとMP2300H間でネットワークアドレスは一致させてください。
通信プロトコル	TCP, UDP

S10VとMP2300H間の物理回線は、モーション制御専用の回線とし、S10VおよびMP2300H以外は接続しないでください。

### 6.3 使用方法

モーション制御命令を使用するには、次に示すような方法でプログラミングします。



**■ MP2300H側のセットアップ**

モーションモジュールを使用するには、HI-FLOWで使用する前に、以下の設定が必要となります。

モーションモジュールのセットアップは、MPシリーズ用エンジニアリングツール「MPE720」から行います。ツールの使用方法については、MP2300Hシリーズのマニュアルおよび「補足6 MP2300Hシステム構成変更手順」を参照してください。

**(1) ディップスイッチの設定**

モーションモジュールの動作条件をディップスイッチから設定します。

**(2) モジュール構成定義の設定**

モーションモジュールのモジュール構成定義を設定します。

**(3) MECHATROLINK定義の設定**

MECHATROLINK伝送システムを使用するうえで必要なパラメーターを設定します。

**(4) モーションパラメーターの設定**

各軸ごとにモーションをコントロールするうえで必要なパラメーターを設定します。

**(5) フラッシュの保存**

設定した情報をMP2300Hシリーズのフラッシュに保存します。

**■ S10V側のセットアップ**

S10V側のセットアップはHI-FLOWシステムから行います。ツールの使用方法については、「ソフトウェアマニュアル オペレーション HI-FLOW For Windows® (マニュアル番号 SVJ-3-132)」を参照してください。

**(6) モーション基本設定**

モーション制御命令を実行する通信設定、軸の定義、モーション状態フラグの格納先、軸ステータスのLPUメモリー転写先を設定します。

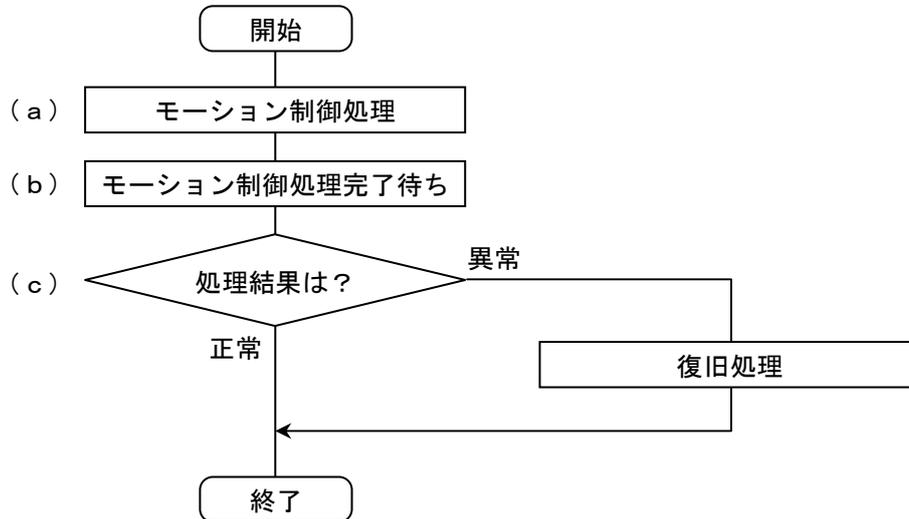
**(7) パラメーターブロック設定**

パラメーターブロック値を設定します。この設定は必要に応じて行ってください。基本的には初回だけ設定します。

## 6 モーション制御命令

### (8) プログラミング

モーション制御命令を使用するには、次のフローに示すように処理の後にモーションモジュールの状態をチェックする必要があります。



#### (a) モーション制御処理

モーション制御命令は、以下の19種類があります。

No.	命令	説明
1	サーボ ON	サーボ 始動
2	サーボ OFF	サーボ 停止
3	位置決め	目標位置と速度を指定して位置決め
4	外部位置決め	移動中に外部位置決め信号をわで位置決め
5	原点復帰	原点復帰
6	定速送り	移動方向と速度を指定して移動開始
7	コマンド 中断	実行中コマンド の中断
8	コマンド 一時停止	実行中コマンド の一時停止
9	コマンド 一時停止解除	実行中コマンド の一時停止解除
10	速度変更	速度の変更
11	速度・位置変更	速度・位置の変更
12	トルク指令	トルク値の設定・変更
13	速度オーバーライド	速度に対する%値の変更
14	トルク制限値変更	トルク制限値の変更
15	速度ループゲイン変更	速度ループゲインの変更
16	位置ループゲイン変更	位置ループゲインの変更
17	単位設定	位置、速度の単位やフィルタタイプ などを変更
18	アラームクリア	指定した軸のアラームクリア
19	NOP	コマンド なし

モーション制御命令は、HI-FLOWシートに貼り付けたモーション制御シンボルから [モーション命令編集] 画面を使用して命令を指定します。

## (b) モーション制御処理完了待ち

モーション制御命令は、処理が完了しなくても命令が終了します。そのためモーション制御命令は、コマンド発行後、処理の完了を待ってから次のコマンドを発行することを推奨します。ただし、速度・位置変更などの制御命令実行中に動作パラメーターを変更するコマンドはその限りではありません（詳細は、各制御命令の機能説明を参照してください）。

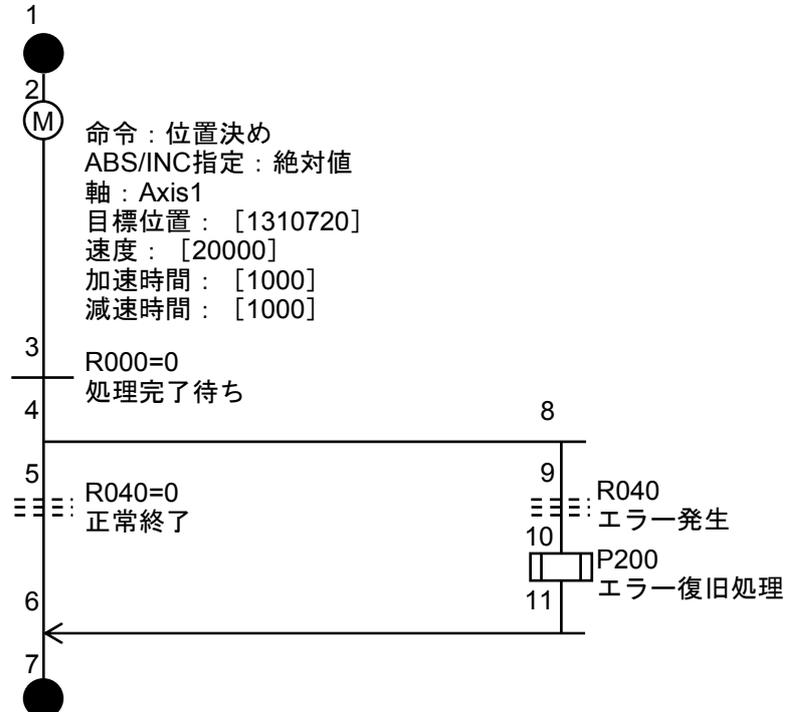
処理完了の確認は、モーション基本設定画面で設定した「動作フラグ」がすべての制御対象の軸でOFFになるのを監視することで行います。

## (c) 処理結果の判定

動作フラグは異常終了した場合もOFFになります。そのため処理が成功したかを判定するには「エラーフラグ」からコマンドの成否を判定してください。エラーの詳細はコマンドエラーに格納されます。なお、モーション側でイベント的に発生するエラーはコマンドエラーだけでは詳細を判定できない場合もあります。これらは、軸ステータスのアラーム、ワーニング、モーションコマンドステータスのBit3（コマンド異常終了状態）で判定可能です。

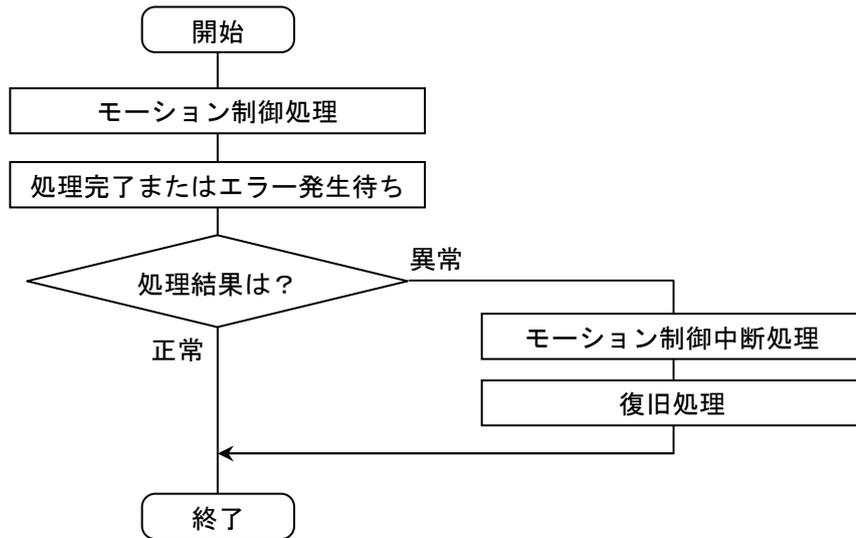
プログラム例を以下に示します。

(R000 : 動作フラグ、R040 : エラーフラグ)



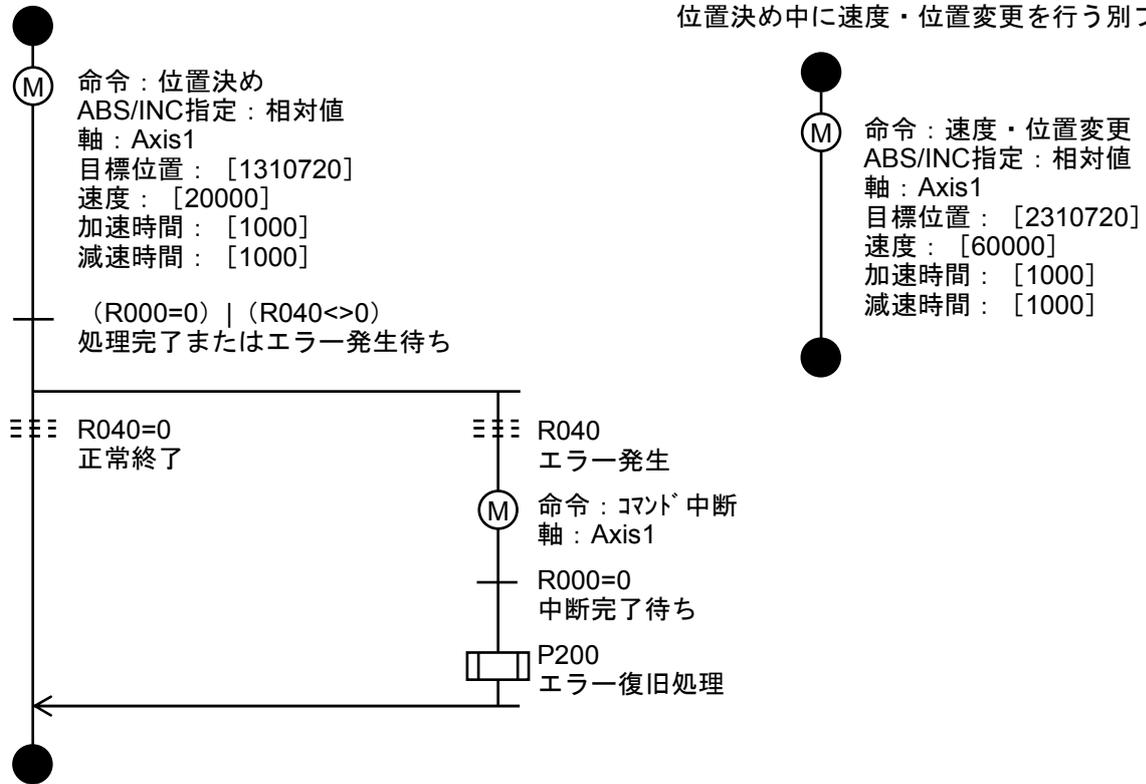
## 6 モーション制御命令

位置決め途中での速度・位置変更など、モーション制御命令実行中に動作パラメーターを変更するようなプログラムの場合、処理完了前に動作パラメーター変更によるエラーが発生する可能性があります。そのような場合、次のフローに示すようにモーション制御処理（例：位置決め）の完了は処理完了だけでなく、エラー発生も同時に待つようにしてください。また、エラーが発生した場合、処理が完了していない可能性がありますので、処理を中断してください。



プログラム例を以下に示します。

(R000 : 動作フラグ、R040 : エラーフラグ)



## 6 モーション制御命令

### (9) インタープリターの送信

HI-FLOWのシステム入れ替えまたは組み込みを行います。この手順は、1台の実機（S10V）に対し、初回だけ行います。

<手順>

HI-FLOWプロセスシートから [ユーティリティ] - [PCs] - [PCsプロセス削除] を選択してください。

[PCsプロセス削除] 画面が表示されます。



この画面から、**全プロセス及びシステム交換** ボタンを選択して **OK** ボタンをクリックすることで、HI-FLOWのシステムが入れ替わります。

### (10) コンパイル

HI-FLOWプロセスシートから [ビルド] - [ビルド] を選択し、作成したHI-FLOWプログラムをコンパイルします。HI-FLOWプログラムにエラーが発生した場合は、アウトプットバーに表示されるエラー内容を参考にHI-FLOWプログラムを修正し、コンパイルを再実行してください。

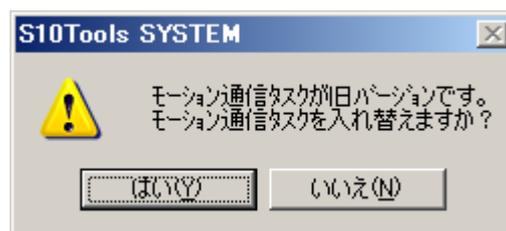
### (11) HI-FLOWプログラム送信

HI-FLOWプロセスシートから [モード] - [オンライン] - [送信] - [全プロセス] を選択し、HI-FLOWプログラムをPCsに送信します。

PCs側に通信タスク環境が構築されていない場合は、PCsリセット後に、自動的にタスク環境が初期化されます。

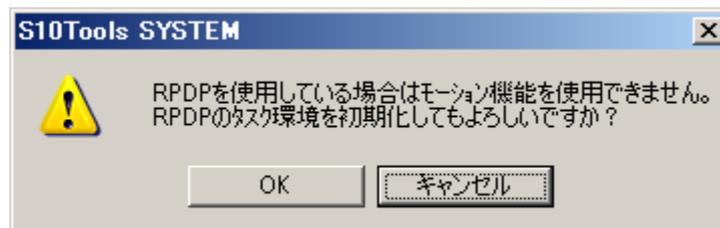
PCs側に通信タスクが送信されていない場合は、HI-FLOWプログラム送信時に通信タスクが自動的に送信されます。通信タスクの送信に成功すると、HI-FLOWプログラム送信後にPCsがリセットされます。

通信タスク送信時に以下のメッセージが表示された場合は、一部の追加機能を有効にするため、**はい** ボタンをクリックして、PCs側の通信タスクを最新版に入れ替えることを推奨します。



## 通 知

- NX/ToolsからNX/HOSTシステムファイルをPCsに送信した場合、NX/Tools以外で登録されたタスクは削除されます。そのため、NX/HOSTシステムファイルを送信した場合は、HI-FLOWプログラムを送信し、通信タスクを再送信してください。
- CPMSデバッガからタスク環境初期化コマンドを実行した場合、PCsに登録されているタスクがすべて削除されます。そのため、タスク環境初期化コマンドを実行した場合は、HI-FLOWプログラムを送信し、通信タスクを再送信してください。
- 通信タスクは以下の内容で送信されますので、CPMSデバッガで使用しないでください。  
タスク番号 : 206 ~ 208  
タスク格納エリア : /300E0000 ~ /300FFFFFF (\$TASK)  
/50800000 ~ /509FFFFFF (\$GLBRW)
- RPDPを使用している場合、モーション機能を使用できません。モーション機能を使用する場合は、HI-FLOWプログラム送信時に表示される以下のメッセージで  ボタンをクリックして、タスク環境を初期化してください。なお、タスク環境を初期化すると、PCsに登録されているRPDPのタスクはすべて削除されます。



## 6 モーション制御命令

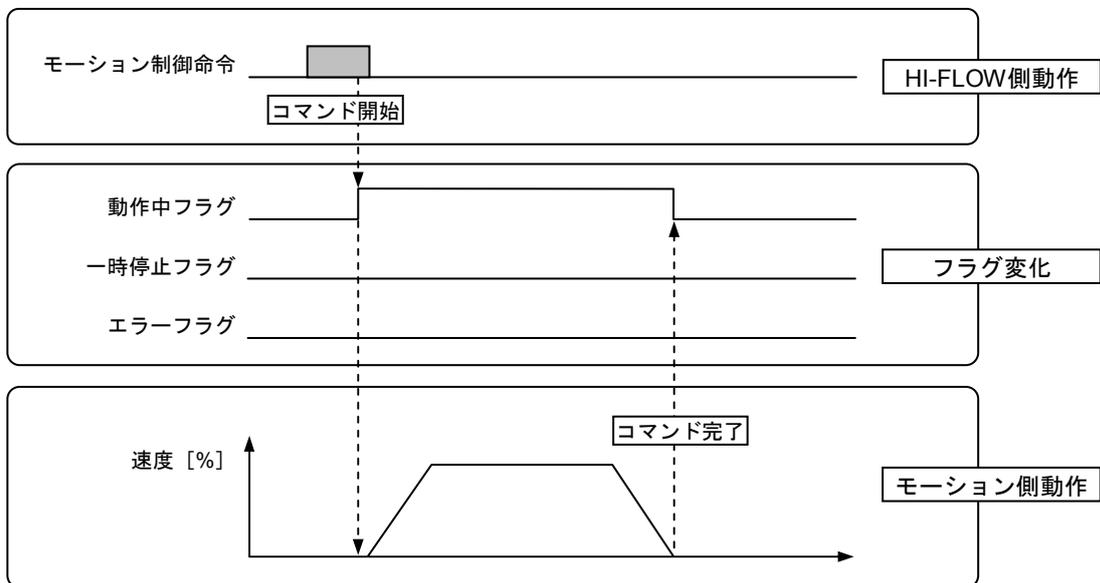
### 6.4 モーション状態フラグ

モーション制御命令の実行結果やモーションの状態が格納されるフラグで、モーション制御命令の実行条件チェック、モーション制御処理完了待ち、および処理結果のチェックに使用します。

各フラグの内容を以下に示します。

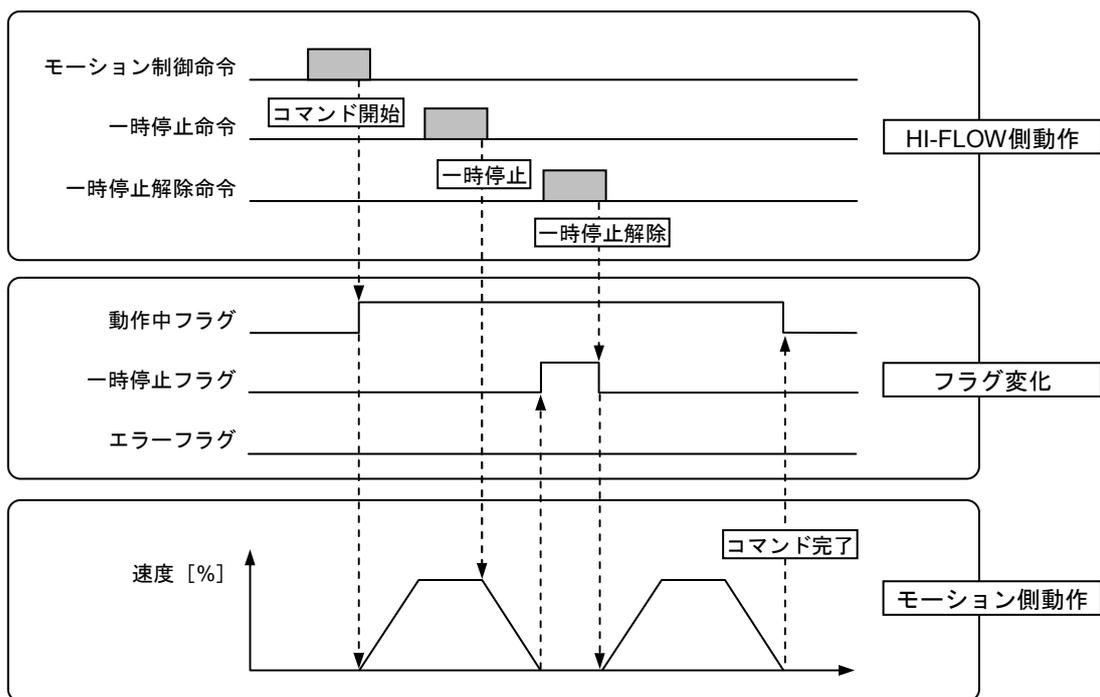
#### (1) 動作フラグ

動作フラグはコマンドの実行状態を管理するためのフラグで、コマンド実行開始で1、コマンド停止で0がセットされます。このフラグを監視することでモーション制御命令の終了を判定することができます。



## (2) 一時停止フラグ

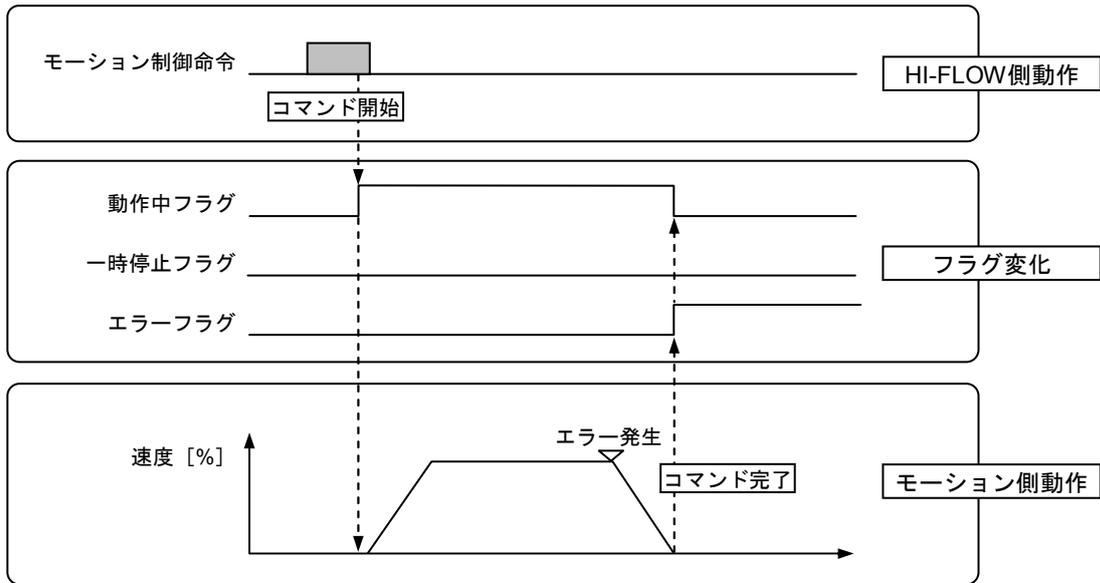
一時停止フラグはコマンドの一時停止状態を管理するためのフラグで、コマンド一時停止で1、コマンド一時停止解除で0がセットされます。このフラグを監視することでモーション制御命令が一時停止中であるか判定することができます。



## 6 モーション制御命令

### (3) エラーフラグ

エラーフラグはコマンドの成否を管理するためのフラグで、コマンド正常終了時は0、コマンド異常終了時は1がセットされます。このフラグを監視することでモーション制御命令に異常が発生したか判定することができます。



## (4) コマンドエラー

コマンドエラーは、HI-FLOW側で発生したエラーコードを格納するためのエリアで、エラーフラグがONしたときにHI-FLOW側で発生したエラー内容を判定するために使用します（モーション側で発生したエラーは、軸ステータスのアラームまたはワーニングを参照してください）。

このコマンドエラーには、以下のエラーコードが格納されます。

(1/2)

エラーコード	エラー項目	エラー内容	エラー時の対処
0x2002	パラメーター異常応答	コマンドのパラメーター異常	コマンドパラメーターを見直してください。
0x2003	軸アラーム発生中	コマンド発行対象軸でアラーム発生中	アラームクリアーコマンドを発行して、アラームを除去してください。
0x2004	モーションコマンド実行中	サーボON発行時、MP2300HのモーションコマンドがNOPでなかった。	アラームクリアーコマンドを発行して、モーションコマンドをNOPにしてください。
0x2005	アラームクリアー実行中	アラームクリアーを実行しているため、コマンドを実行できません。	アラームクリアー終了後に、再度コマンドを発行してください。
0x2006	軸使用中	コマンド発行対象軸が、上書き不可コマンドを実行中	コマンドが終了するまで待つて、次のコマンドを発行してください。
0x2007	サーボOFF状態でのコマンド実行	コマンド発行対象軸がサーボOFF状態です。	サーボON実施後、コマンドを発行してください。
0x2008	運転準備未完	サーボON発行時にモーションモジュールが運転準備を完了していません。	MECHATROLINKケーブルが外れていないか確認してください。
0x2009	未接続モジュール指定応答	未接続のモジュールの軸に対してコマンドを発行しました。	未接続のモジュールに対してコマンドを発行していないか、Axis定義を見直してください。Axis定義に問題がない場合は、MECHATROLINKケーブルが外れていないか確認してください。
0x200A	不正な加算方式	「位置決め」または「外部位置決め」コマンドに対する「速度・位置変更」コマンドの使用法誤り。	MP2300Hの位置指令タイプの設定が増分値加算方式の場合、「位置決め」または「外部位置決め」コマンドを絶対値指定で発行した後に「速度・位置変更」コマンドでの変更はできません。プログラムを見直してください。
0x2100	不明コマンド受信応答	サポート外コマンドコードを受信しました。	通信タスクが壊れている可能性があります。通信タスクを再ローディングしてください。
0x2101	通信タイムアウト中応答	MP2300Hでタイムアウト検出中	リトライで回復しない場合、MP2300Hを停復電してください。
0x2102	通信開始準備未完応答	MP2300Hで通信準備ができていません。	通信準備が完了するまで待つて、再度コマンドを実行してください。
0x2103	コマンドサイズ不足応答	通信ヘッダー中の伝送語数より実際の伝送語数が小さい。	通信タスクが壊れている可能性があります。通信タスクを再ローディングしてください。

## 6 モーション制御命令

(2/2)

エラーコード	エラー項目	エラー内容	エラー時の対処
0x2104	コマンドサイズ超過応答	通信ヘッダー中の伝送語数より実際の伝送語数が大きい。	通信タスクが壊れている可能性があります。通信タスクを再ローディングしてください。
0x2105	コマンドエントリーサイズ不整合	通信データ中のエントリーサイズと実際のエントリー情報が合致していない。	
0x4001	コマンド通信リトライオーバー	コマンド通信でモーションモジュールからの応答が検出できませんでした。	MP2300Hが立ち上っているか、接続ケーブルが外れていないか確認してください。
0x8001	ET.NET異常	ET.NETハンドラーでエラーを検出しました。	ET.NETがエラー停止している可能性があります。ET.NETを交換してください。
0x8100	モーション側タイムアウト時間設定異常	モーション側タイムアウト時間の設定値が壊れています。	ツールで [モーション基本設定] の [通信設定] のモーション側タイムアウト時間の設定値を修正してください。

## (5) 軸ステータス割り付け

モーションの状態を格納するためのエリアで、軸ごとのエラー情報や処理状態を確認するために使用します。

軸ステータスは、軸ごとに以下のように割り付けられます。

相対アドレス (ワード)

00	管理情報
01	将来用
02	運転ステータス
03	モーションコマンドレスポンスコード
04	モーションコマンドステータス
05	位置管理ステータス
06	ワーニング
07	
08	アラーム
09	
0A	軸ステータス
0B	将来用
0C	
0D	
0E	
0F	

## ■ 管理情報

モーションの運転ステータスが報告されます。

Bit	運転ステータス内容
0	無効フラグ (情報が無効時ON)
1~15	未使用

## 6 モーション制御命令

### ■ 運転ステータス

モーションの運転ステータスが報告されます。

Bit	運転ステータス内容
0	モーションコントローラ運転準備完了
1	運転中（サーボON）
2	システムBUSY
3	サーボREADY
4～15	未使用

### ■ モーションコマンドレスポンスコード

実行中のコマンド番号が報告されます。

値	コマンド内容
0	コマンドなし
1	位置決め
2	外部位置決め
3	原点復帰
4	直線補間、等速制御
7	定速送り
23	速度変更、速度・位置変更
24	トルク指令

### ■ モーションコマンドステータス

モーションコマンドの実行ステータスが報告されます。

Bit	モーションコマンドステータス内容
0	コマンド実行中フラグ（BUSY）
1	コマンド一時停止完了（HOLD）
2	未使用
3	コマンド異常終了状態（FAIL）
4～6	未使用
7	ABSエンコーダーリセット完了
8	コマンド実行完了（COMPLETE）
9～15	未使用

■ 位置管理ステータス

Bit	コマンド内容
0	払い出し完了
1	位置決め完了
2	ラッチ完了
3	位置決め近傍
4	原点位置
5	原点復帰完了
6	マシンロック中
7	未使用
8	ABSシステム無限長位置管理情報LOAD完了
9	POSMAXターン数プリセット完了
10~15	未使用

■ ワーニング

軸単位で発生したワーニング内容が報告されます。

Bit	ワーニング内容
0	偏差異常
1	設定パラメーター設定異常
2	固定パラメーター設定異常
3	サーボドライバ異常
4	モーションコマンド設定異常
5	未使用
6	正方向オーバートラベル
7	負方向オーバートラベル
8	サーボオン未完
9	サーボドライバ通信警告
10~31	未使用

## 6 モーション制御命令

### ■ アラーム

軸単位で発生したアラーム内容が報告されます。

Bit	アラーム内容
0	サーボドライバー異常
1	正方向オーバートラベル
2	負方向オーバートラベル
3	正方向ソフトリミット
4	負方向ソフトリミット
5	サーボオフ
6	位置決めタイムオーバー
7	位置決め移動量過大
8	速度過大
9	偏差異常
10	フィルタータイプ変更エラー
11	フィルター時定数変更エラー
12	未使用
13	原点未設定
14~15	システム用
16	サーボドライバー同期通信エラー
17	サーボドライバー通信エラー
18	サーボドライバーコマンドタイムアウトエラー
19	ABSエンコーダー回転量オーバー
20	PG断線エラー
20~29	未使用
30	サーボパック設定モーター種別不一致
31	サーボパック接続エンコーダー種別不一致

### ■ 軸ステータス

モーションの軸ステータスが報告されます。

Bit	運転ステータス内容
0	システム用
1	アラームクリアー実行中
2	軸指定異常
3~15	未使用

「軸指定異常」は、軸が実装ありでOFF、実装なし（未接続）でONします。



## 6 モーション制御命令

### 6.5 機能説明

この節では、各モーション制御命令の詳細について述べます。以下に示す形式で説明します。

POS	位置決め	モーション制御命令名称						
機能説明	指定した目標位置と速度で位置決めします。	モーション制御命令の処理内容を説明します。						
パラメーターと処理内容	<p>Ⓜ 命令：位置決め  ABS/INC指定：①参照  軸：位置決めを実行する軸番号  パラメーターブロック：②～⑤を省略時に使用されるパラメーターグループの番号  目標位置：②参照  速度：③参照  加速時間：④参照  減速時間：⑤参照</p> <p>① ABS/INC指定  目標位置の指令方式を切り替えてください。指令方式には、以下の2種類があります。</p> <table border="1"> <thead> <tr> <th>選択項目</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>相対値</td> <td>目標位置を現在位置からの相対値として設定</td> </tr> <tr> <td>絶対値</td> <td>目標位置を原点からの絶対値として設定</td> </tr> </tbody> </table> <p>② 目標位置  位置決め目標位置を指定してください。  「① ABS/INC指定」の指定によって数値の意味が異なります。</p> <p>③ 速度  位置決め動作時の速度を指定してください。</p> <p>④ 加速時間  位置決め加速度を指定してください。</p> <p>⑤ 減速時間  位置決め減速度を指定してください。</p>	選択項目	内容	相対値	目標位置を現在位置からの相対値として設定	絶対値	目標位置を原点からの絶対値として設定	パラメーターの内容を説明します。
選択項目	内容							
相対値	目標位置を現在位置からの相対値として設定							
絶対値	目標位置を原点からの絶対値として設定							
フラグの設定	<p>動作フラグ：位置決め開始でON、終了でOFFします。  位置決め失敗時にOFFします。  一時停止フラグ：一時停止命令が完了でON、一時停止解除命令が完了でOFFします。  エラーフラグ：位置決め失敗時にONします。  コマンドエラー：エラーフラグがONしたときに、エラー内容に対応したエラーコードが格納されます（エラーコードの内容は「6.4 モーション状態フラグ」を参照してください）。</p> <p>&lt;通常&gt;</p> <p>&lt;中断/コマンド上書き&gt;</p>	命令後変更するフラグの説明およびタイムチャートを示します。						

(前ページの続き)  
フラグの設定

<一時停止/一時停止解除>

HI-FLOW 側動作

位置決め命令 一時停止命令 一時停止解除命令

モーション制御命令

フラグ変化

動作中フラグ  
一時停止フラグ  
エラーフラグ

モーション側動作

速度

時間

<異常終了>

HI-FLOW 側動作

位置決め命令

モーション制御命令

フラグ変化

動作中フラグ  
一時停止フラグ  
エラーフラグ

モーション側動作

速度

時間

エラー発生

備考

速度には0~327.67%のオーバーライドをかけることができます。

使用例

● M 命令：位置決め  
ABS/INC指定：相対値  
軸：Axis1  
パラメータロック：PB1  
目標位置：[10000000]  
速度：[500000]  
加速時間：[10000]  
減速時間：[10000]

R000=0

----- 位置決め開始

----- 位置決め完了待ち

速度

50000

(速度)

10000000 (目標位置)

10000 (加速時間)

10000 (減速時間)

時間

有効パラメーター

No.	パラメーター名	PI/O種別	設定範囲
①	ABS/INC指定	ワード	0：相対値 1：絶対値
②	目標位置	ロング	-2147483648~2147483647
③	速度	ロング	-2147483648~2147483647
④	加速時間	ロング	0~2147483647
⑤	減速時間	ロング	0~2147483647

(PI/Oをインダイレクト指定した場合、奇数アドレス値指定でパラメーターエラー)

(前ページの続き)

注意点を示します。

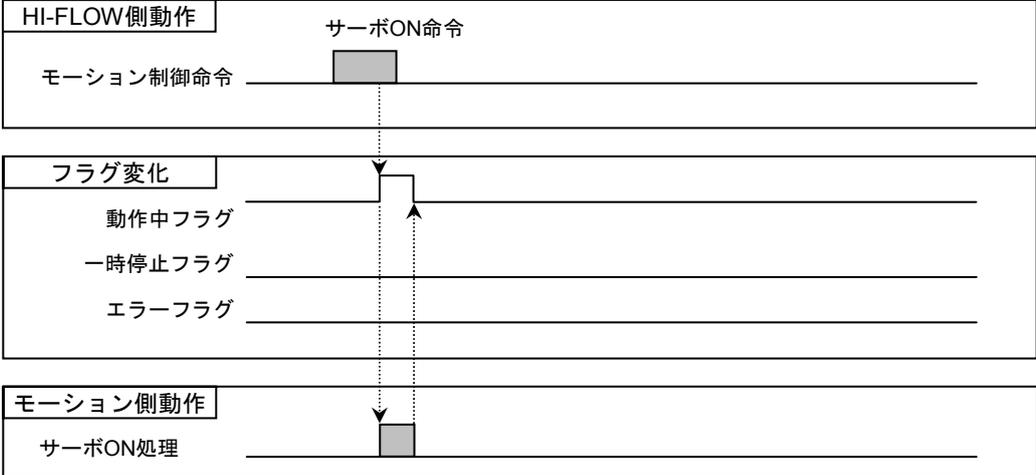
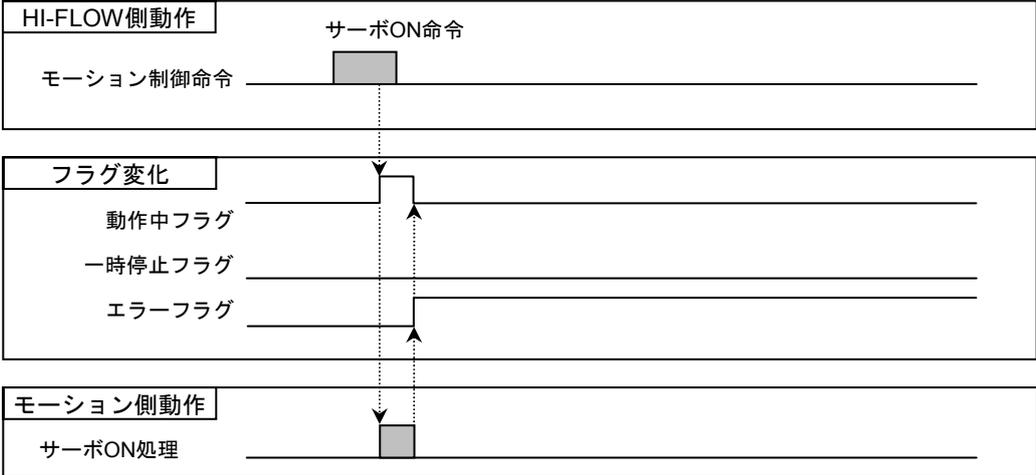
主な使用例を表します。

主な使用例の動作結果を表します。

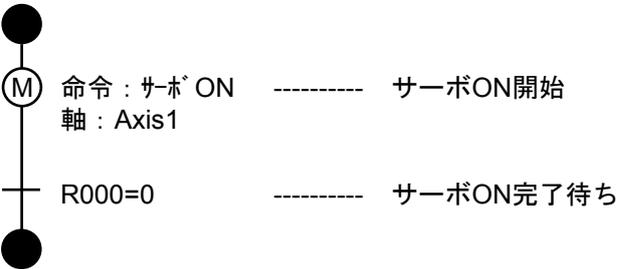
指定可能なパラメーターを示します。モーション制御命令では、ワード型PI/O、ロング型PI/O、定数が指定できます。またアドレッシングモードで指定することもできます。

## 6 モーション制御命令

(1/2)

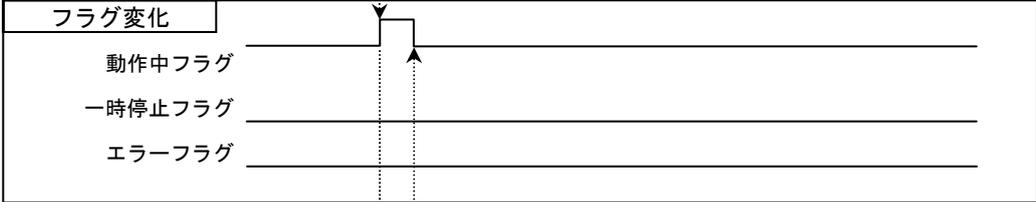
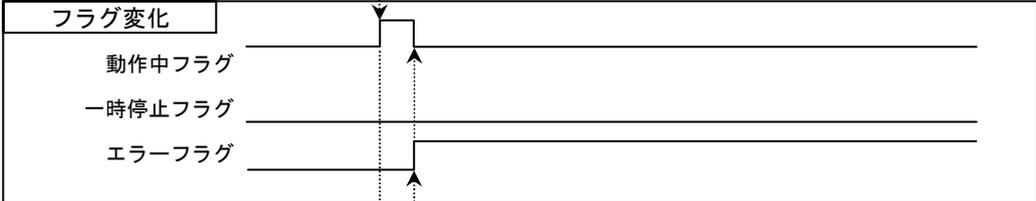
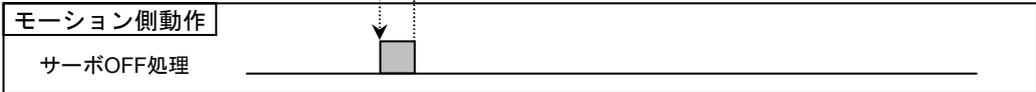
SVON	サーボON
機能説明	サーボモーターを通電状態に切り替えます。
パラメーターと処理内容	 命令：サーボON 軸：サーボONを実行する軸番号（指定範囲は1～32）
フラグの設定	<p>動作フラグ：サーボON開始でON、終了でOFFします。          サーボONに失敗したときにOFFします。</p> <p>一時停止フラグ：常時OFFです。</p> <p>エラーフラグ：サーボONに失敗したときにONします。</p> <p>&lt;通常&gt;</p>  <p>&lt;異常終了&gt;</p> 
備考	アラーム発生中はサーボONできません。サーボONの前にアラームクリアー命令を実行してください。

(2/2)

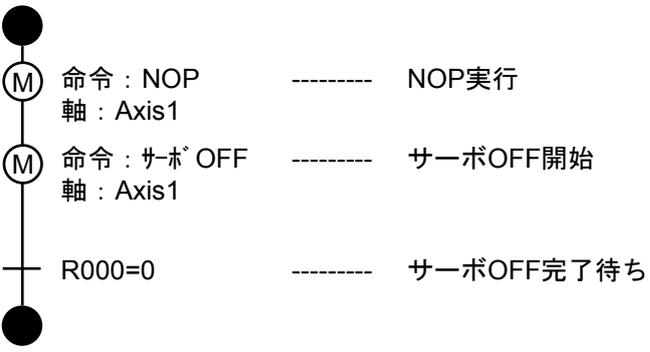
<p>使用例</p>	
<p>有効 パラメーター</p>	

## 6 モーション制御命令

(1/2)

SVOFF	サーボOFF
機能説明	サーボモータを非通電状態に切り替えます。
パラメーターと処理内容	<p>  命令：サーボOFF              軸：サーボOFFを実行する軸番号（指定範囲は1～32）         </p> <p>             &lt;注意事項&gt;              サーボOFF実行前には、必ずNOPを実行し状態をクリアーしてください。         </p>
フラグの設定	<p>             動作フラグ：サーボOFF開始でON、終了でOFFします。              サーボOFFに失敗したときにOFFします。         </p> <p>             一時停止フラグ：常時OFFです。         </p> <p>             エラーフラグ：サーボOFFに失敗したときにONします。         </p> <p>             &lt;通常&gt;         </p> <div data-bbox="354 913 1390 1043"> <p>HI-FLOW側動作</p>  </div> <div data-bbox="354 1070 1390 1272"> <p>フラグ変化</p>  </div> <div data-bbox="354 1299 1390 1391"> <p>モーション側動作</p>  </div> <p>             &lt;異常終了&gt;         </p> <div data-bbox="354 1469 1390 1599"> <p>HI-FLOW側動作</p>  </div> <div data-bbox="354 1626 1390 1827"> <p>フラグ変化</p>  </div> <div data-bbox="354 1854 1390 1946"> <p>モーション側動作</p>  </div>
備考	

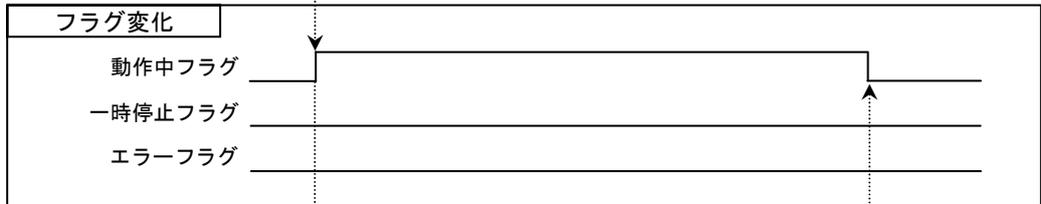
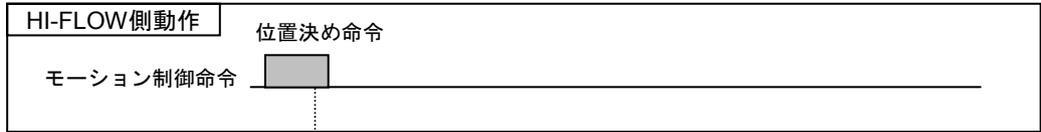
(2/2)

使用例	 <p>命令 : NOP      -----      NOP実行 軸 : Axis1</p> <p>命令 : サーボOFF      -----      サーボOFF開始 軸 : Axis1</p> <p>R000=0      -----      サーボOFF完了待ち</p>
有効 パラメーター	

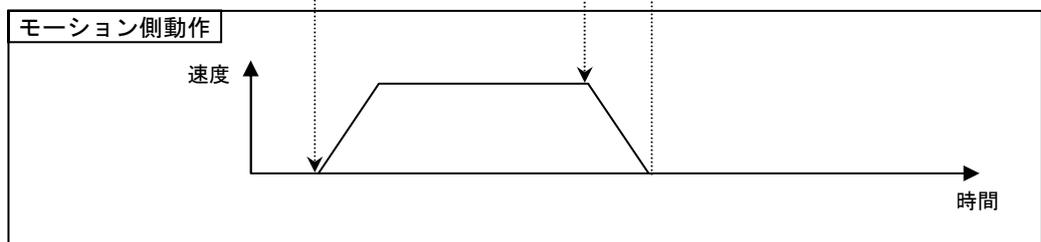
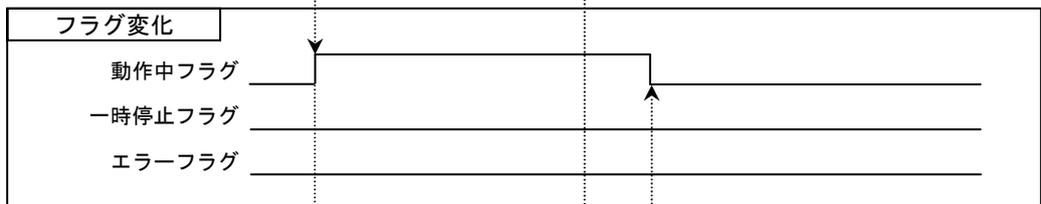
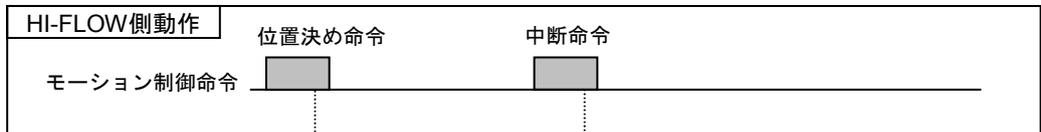
POS	位置決め						
機能説明	指定した目標位置と速度で位置決めします。						
パラメーターと処理内容	<p>  命令：位置決め            ABS/INC指定：①参照            軸：位置決めを実行する軸番号            パラメーターブロック：②～⑤を省略時に使用されるパラメーターグループの番号            目標位置：②参照            速度：③参照            加速時間：④参照            減速時間：⑤参照         </p> <p>           ① ABS/INC指定            目標位置の指令方式を切り替えてください。指令方式には、以下の2種類があります。         </p> <table border="1" data-bbox="464 947 1279 1070"> <thead> <tr> <th>選択項目</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>相対値</td> <td>目標位置を現在位置からの相対値として設定</td> </tr> <tr> <td>絶対値</td> <td>目標位置を原点からの絶対値として設定</td> </tr> </tbody> </table> <p>           ② 目標位置            位置決め目標位置を指定してください。            「① ABS/INC指定」の指定によって数値の意味が異なります。         </p> <p>           ③ 速度            位置決め動作時の速度を指定してください。         </p> <p>           ④ 加速時間            位置決め動作時の加速時間／加速度を指定してください。         </p> <p>           ⑤ 減速時間            位置決め動作時の減速時間／減速度を指定してください。         </p>	選択項目	内容	相対値	目標位置を現在位置からの相対値として設定	絶対値	目標位置を原点からの絶対値として設定
選択項目	内容						
相対値	目標位置を現在位置からの相対値として設定						
絶対値	目標位置を原点からの絶対値として設定						
フラグの設定	<p>動作フラグ：位置決め開始でON、終了でOFFします。</p> <p>位置決めに失敗したときにOFFします。</p> <p>一時停止フラグ：一時停止命令が完了でON、一時停止解除命令が完了でOFFします。</p> <p>エラーフラグ：位置決めに失敗したときにONします。</p> <p>コマンドエラー：エラーフラグがONしたときに、エラー内容に対応したエラーコードが格納されます（エラーコードの内容は「6.4 モーション状態フラグ」を参照してください）。</p>						

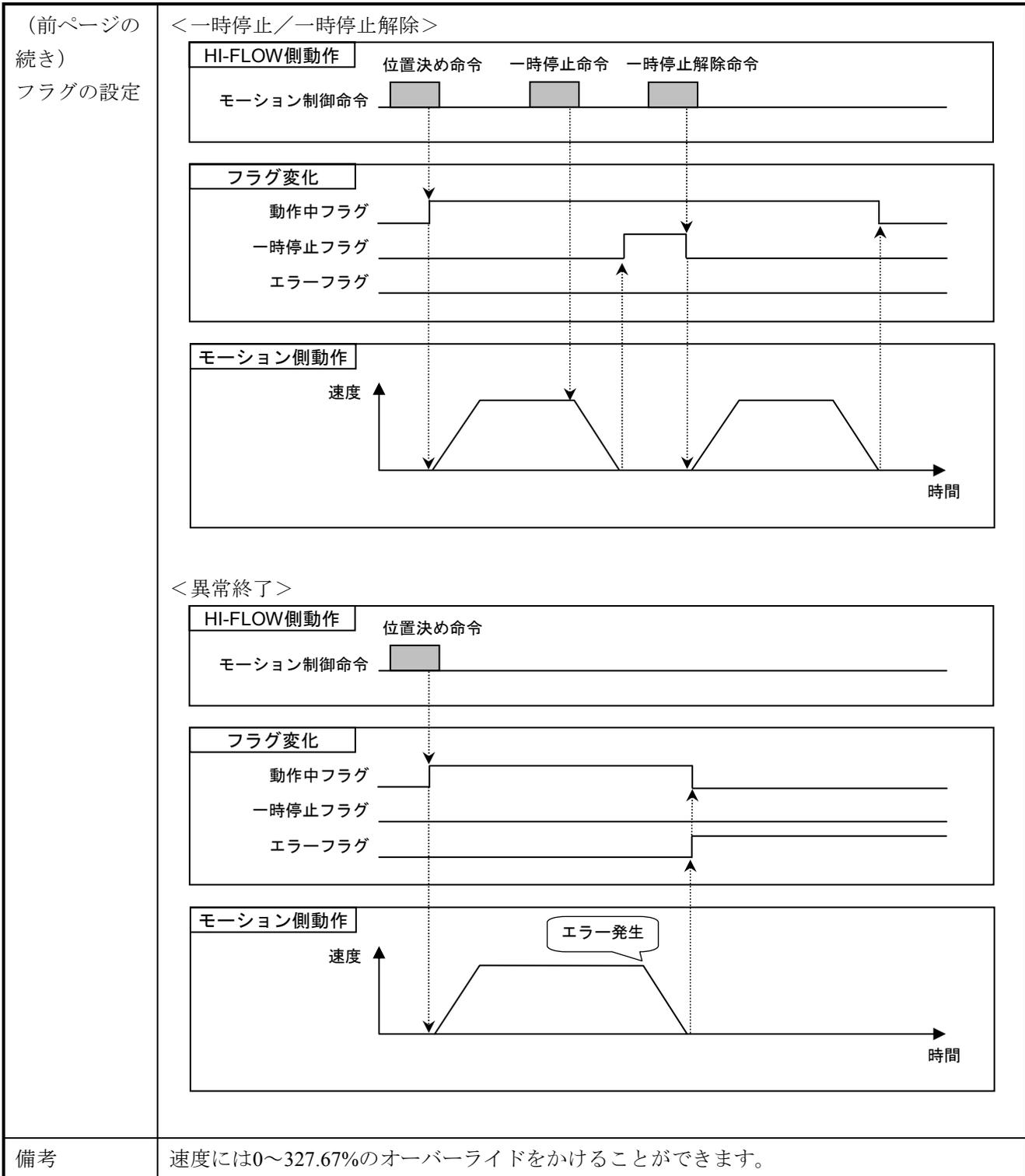
(前ページの  
続き)  
フラグの設定

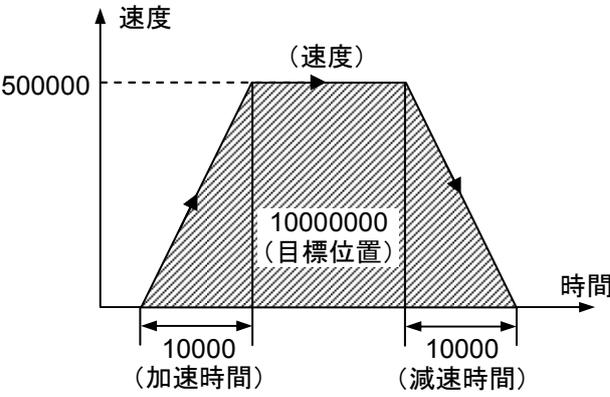
<通常>



<中断/コマンド上書き>



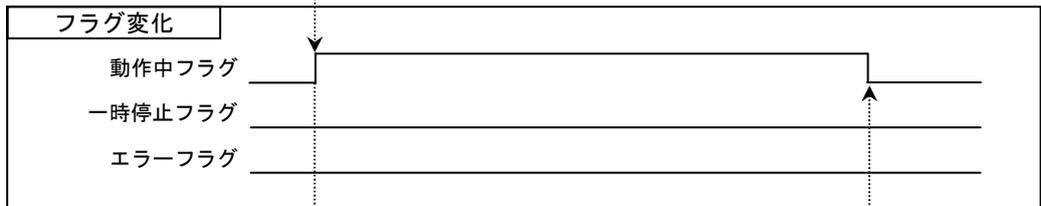
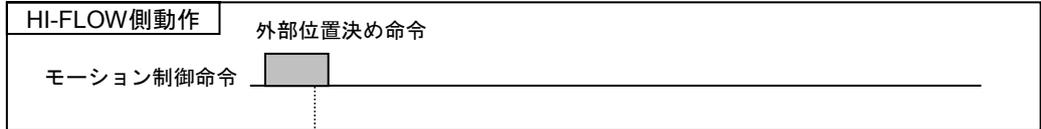


使用例	<div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="text-align: center; margin-right: 10px;">  </div> <div> <p>命令：位置決め</p> <p>ABS/INC指定：相対値</p> <p>軸：Axis1</p> <p>パラメータブロック：PB1</p> <p>目標位置：[10000000]</p> <p>速度：[500000]</p> <p>加速時間：[10000]</p> <p>減速時間：[10000]</p> </div> <div style="margin-left: 20px;"> <p>----- 位置決め開始</p> </div> </div> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="text-align: center; margin-right: 10px;">  </div> <div> <p>R000=0</p> </div> <div style="margin-left: 20px;"> <p>----- 位置決め完了待ち</p> </div> </div> <div style="text-align: center;">  <p>速度</p> <p>500000</p> <p>(速度)</p> <p>10000000 (目標位置)</p> <p>時間</p> <p>10000 (加速時間)</p> <p>10000 (減速時間)</p> </div>																								
有効 パラメーター	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 5%;">No.</th> <th style="width: 30%;">パラメーター名</th> <th style="width: 15%;">PI/O種別</th> <th style="width: 50%;">設定範囲</th> </tr> </thead> <tbody> <tr> <td>①</td> <td>ABS/INC指定</td> <td>ワード</td> <td>0：相対値 1：絶対値</td> </tr> <tr> <td>②</td> <td>目標位置</td> <td>ロング</td> <td>-2147483648～2147483647</td> </tr> <tr> <td>③</td> <td>速度</td> <td>ロング</td> <td>-2147483648～2147483647</td> </tr> <tr> <td>④</td> <td>加速時間</td> <td>ロング</td> <td>0～2147483647</td> </tr> <tr> <td>⑤</td> <td>減速時間</td> <td>ロング</td> <td>0～2147483647</td> </tr> </tbody> </table> <p style="text-align: center; margin-top: 5px;">(PI/Oをインダイレクト指定した場合、奇数アドレス値指定でパラメーターエラー)</p>	No.	パラメーター名	PI/O種別	設定範囲	①	ABS/INC指定	ワード	0：相対値 1：絶対値	②	目標位置	ロング	-2147483648～2147483647	③	速度	ロング	-2147483648～2147483647	④	加速時間	ロング	0～2147483647	⑤	減速時間	ロング	0～2147483647
No.	パラメーター名	PI/O種別	設定範囲																						
①	ABS/INC指定	ワード	0：相対値 1：絶対値																						
②	目標位置	ロング	-2147483648～2147483647																						
③	速度	ロング	-2147483648～2147483647																						
④	加速時間	ロング	0～2147483647																						
⑤	減速時間	ロング	0～2147483647																						

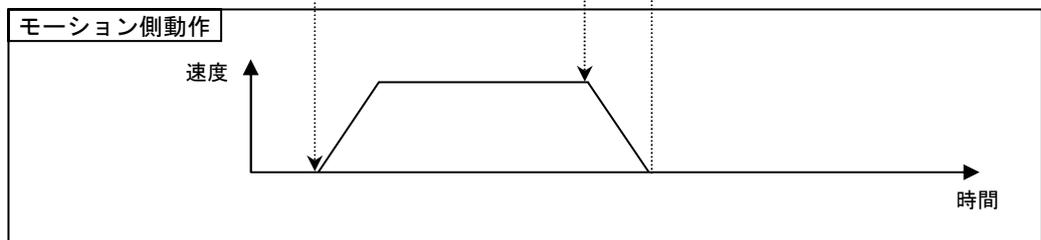
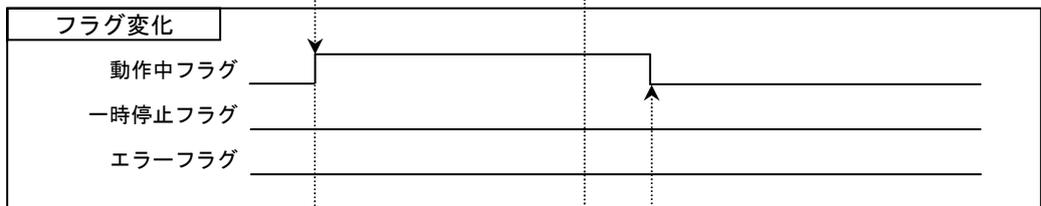
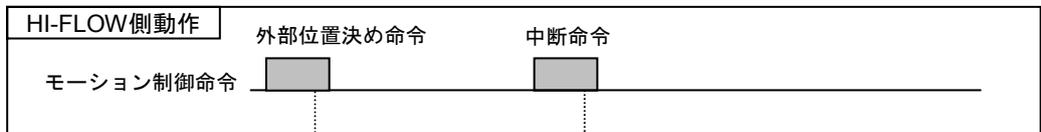
EXPOS	外部位置決め						
機能説明	<p>指定した目標位置と速度で位置決めします。</p> <p>移動中に外部信号がONされると、指定した外部位置決め最終走行距離だけ進んだ位置に位置決めします。外部信号がONされない場合は、目標位置に位置決めします。</p>						
パラメータと処理内容	<p> 命令：外部位置決め</p> <p>ABS/INC指定：①参照</p> <p>軸：外部位置決めを実行する軸番号</p> <p>パラメータブロック：②～⑥省略時に使用されるパラメータグループの番号</p> <p>目標位置：②参照</p> <p>速度：③参照</p> <p>加速時間：④参照</p> <p>減速時間：⑤参照</p> <p>外部位置決め最終走行距離：⑥参照</p> <p>① ABS/INC指定</p> <p>目標位置の指令方式を切り替えてください。指令方式には、以下の2種類があります。</p> <table border="1" data-bbox="464 1077 1278 1200"> <thead> <tr> <th>選択項目</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>相対値</td> <td>目標位置を現在位置からの相対値として設定</td> </tr> <tr> <td>絶対値</td> <td>目標位置を原点からの絶対値として設定</td> </tr> </tbody> </table> <p>② 目標位置</p> <p>位置決め目標位置を指定してください。</p> <p>「① ABS/INC指定」の指定によって数値の意味が異なります。</p> <p>③ 速度</p> <p>位置決め動作時の速度を指定してください。</p> <p>④ 加速時間</p> <p>位置決め動作時の加速時間／加速度を指定してください。</p> <p>⑤ 減速時間</p> <p>位置決め動作時の減速時間／減速度を指定してください。</p> <p>⑥ 外部位置決め最終走行距離</p> <p>外部位置決め信号入力後の移動量を指定してください。</p> <p>&lt;注意事項&gt;</p> <p>外部位置決め完了後は、必ずNOPを実行してください。</p>	選択項目	内容	相対値	目標位置を現在位置からの相対値として設定	絶対値	目標位置を原点からの絶対値として設定
選択項目	内容						
相対値	目標位置を現在位置からの相対値として設定						
絶対値	目標位置を原点からの絶対値として設定						
フラグの設定	<p>動作フラグ：外部位置決め開始でON、終了でOFFします。</p> <p>外部位置決めに失敗したときにOFFします。</p> <p>一時停止フラグ：一時停止命令が完了でON、一時停止解除命令が完了でOFFします。</p> <p>エラーフラグ：外部位置決めに失敗したときにONします。</p>						

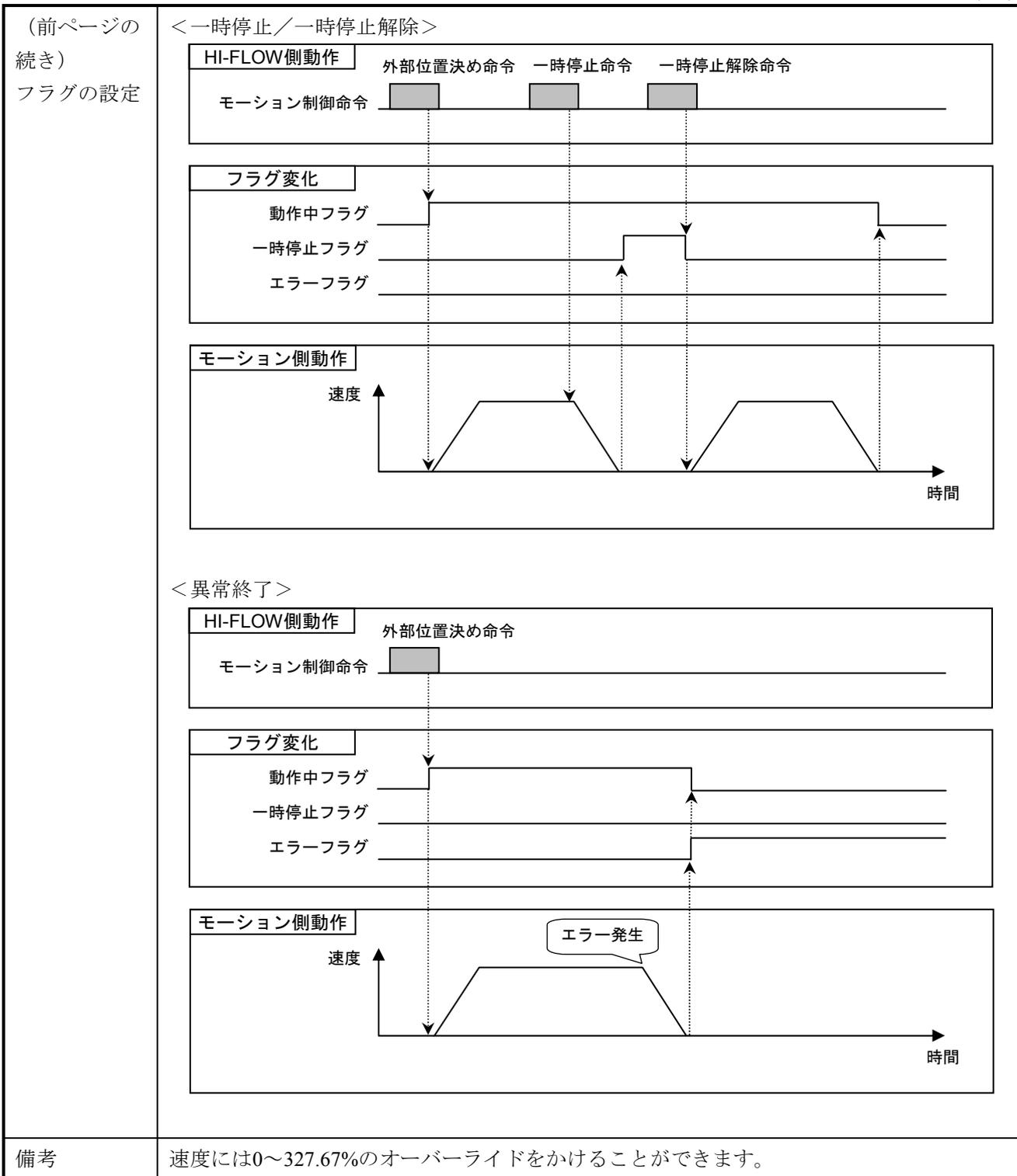
(前ページの  
続き)  
フラグの設定

<通常>



<中断/コマンド上書き>





使用例

命令：外部位置決め  
 ABS/INC指定：相対値  
 軸：Axis1  
 パラメータブロック：PB1  
 目標位置：[1000000]  
 速度：[500000]  
 加速時間：[10000]  
 減速時間：[10000]  
 外部位置決め最終走行距離：[200000]

R000=0

命令：NOP  
 軸：Axis1

----- 外部位置決め開始  
 ----- 外部位置決め完了待ち  
 ----- NOP実行

速度

500000

(速度)

200000

外部位置決め最終走行距離

時間

10000 (加速時間)

10000 (減速時間)

外部位置決め信号

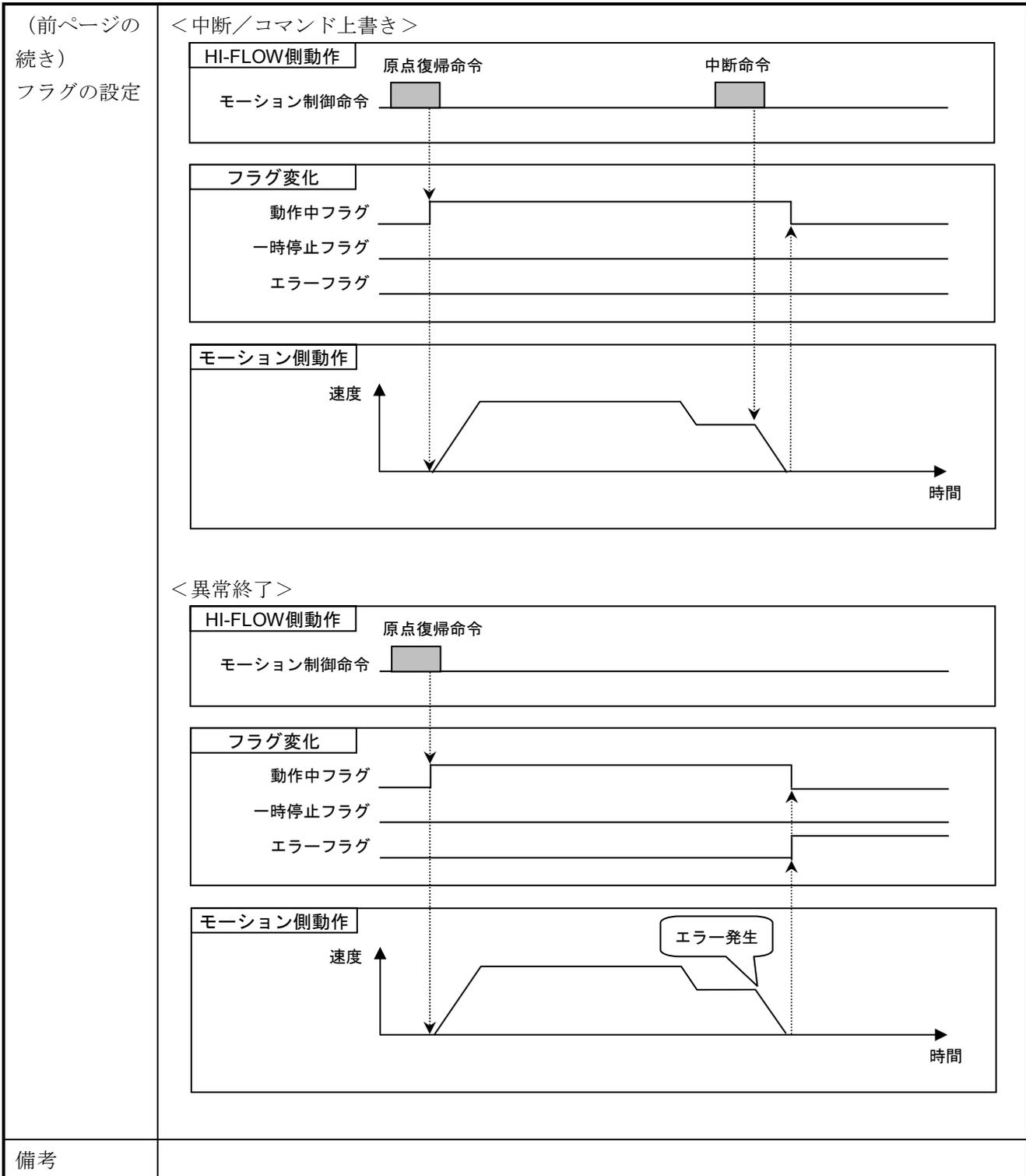
有効パラメーター

No.	パラメーター名	PI/O種別	設定範囲
①	ABS/INC指定	ワード	0：相対値 1：絶対値
②	目標位置	ロング	-2147483648～2147483647
③	速度	ロング	-2147483648～2147483647
④	加速時間	ロング	0～2147483647
⑤	減速時間	ロング	0～2147483647
⑥	外部位置決め最終走行距離	ロング	-2147483648～2147483647

(PI/Oをインダイレクト指定した場合、奇数アドレス値指定でパラメーターエラー)

ZRET	原点復帰																																				
機能説明	機械座標系の原点に復帰します。																																				
パラメーターと処理内容	<p> 命令：原点復帰          原点復帰方式：①参照          軸：原点復帰を実行する軸番号          パラメーターブロック：②～⑧省略時に使用されるパラメーターグループの番号          速度：②参照          加速時間：③参照          減速時間：④参照          原点復帰方向：⑤参照          アプローチ速度：⑥参照          クリープ速度：⑦参照          最終走行距離：⑧参照</p> <p>① 原点復帰方式          原点復帰を切り替えます。原点復帰方式には、以下の17種類があります。</p> <table border="1"> <thead> <tr> <th>方式</th> <th>内容</th> </tr> </thead> <tbody> <tr> <td>DEC1+C相パルス方式</td> <td>減速LSとC相パルスによる3段減速方式</td> </tr> <tr> <td>ZERO信号方式</td> <td>ZERO信号による原点復帰方式</td> </tr> <tr> <td>DEC1+ZERO信号方式</td> <td>減速LSとZERO信号による3段減速方式</td> </tr> <tr> <td>C相パルス方式</td> <td>C相パルスによる原点復帰方式</td> </tr> <tr> <td>DEC2+ZERO信号</td> <td>減速LSを領域信号とし、ZERO信号を原点信号とする原点復帰方式</td> </tr> <tr> <td>DEC1+LMT+ZERO信号</td> <td>減速LSと2つの原点復帰用リミット信号（LMT）を領域信号とし、ZERO信号を原点復帰信号とする原点復帰方式</td> </tr> <tr> <td>DEC2+C相パルス</td> <td>減速LSを領域信号とし、C相信号を原点信号とする原点復帰方式</td> </tr> <tr> <td>DEC1+LMT+C相パルス</td> <td>減速LSと2つの原点復帰用リミット信号（LMT）を領域信号とし、C相信号を原点復帰信号とする原点復帰方式</td> </tr> <tr> <td>C相パルス</td> <td>C相パルスだけによる方式</td> </tr> <tr> <td>POT&amp;C相パルス</td> <td>正側OT信号とC相パルスによる方式</td> </tr> <tr> <td>POT</td> <td>正側OT信号だけによる方式</td> </tr> <tr> <td>HOMELS&amp;C相パルス</td> <td>HOME信号とC相パルスによる方式</td> </tr> <tr> <td>HOMELS</td> <td>HOME信号だけによる方式</td> </tr> <tr> <td>NOT&amp;C相パルス</td> <td>逆側OT信号とC相パルスによる方式</td> </tr> <tr> <td>NOT</td> <td>逆側OT信号だけによる方式</td> </tr> <tr> <td>INPUT&amp;C相パルス</td> <td>入力信号とC相パルスによる方式</td> </tr> <tr> <td>INPUT</td> <td>入力信号だけによる方式</td> </tr> </tbody> </table>	方式	内容	DEC1+C相パルス方式	減速LSとC相パルスによる3段減速方式	ZERO信号方式	ZERO信号による原点復帰方式	DEC1+ZERO信号方式	減速LSとZERO信号による3段減速方式	C相パルス方式	C相パルスによる原点復帰方式	DEC2+ZERO信号	減速LSを領域信号とし、ZERO信号を原点信号とする原点復帰方式	DEC1+LMT+ZERO信号	減速LSと2つの原点復帰用リミット信号（LMT）を領域信号とし、ZERO信号を原点復帰信号とする原点復帰方式	DEC2+C相パルス	減速LSを領域信号とし、C相信号を原点信号とする原点復帰方式	DEC1+LMT+C相パルス	減速LSと2つの原点復帰用リミット信号（LMT）を領域信号とし、C相信号を原点復帰信号とする原点復帰方式	C相パルス	C相パルスだけによる方式	POT&C相パルス	正側OT信号とC相パルスによる方式	POT	正側OT信号だけによる方式	HOMELS&C相パルス	HOME信号とC相パルスによる方式	HOMELS	HOME信号だけによる方式	NOT&C相パルス	逆側OT信号とC相パルスによる方式	NOT	逆側OT信号だけによる方式	INPUT&C相パルス	入力信号とC相パルスによる方式	INPUT	入力信号だけによる方式
方式	内容																																				
DEC1+C相パルス方式	減速LSとC相パルスによる3段減速方式																																				
ZERO信号方式	ZERO信号による原点復帰方式																																				
DEC1+ZERO信号方式	減速LSとZERO信号による3段減速方式																																				
C相パルス方式	C相パルスによる原点復帰方式																																				
DEC2+ZERO信号	減速LSを領域信号とし、ZERO信号を原点信号とする原点復帰方式																																				
DEC1+LMT+ZERO信号	減速LSと2つの原点復帰用リミット信号（LMT）を領域信号とし、ZERO信号を原点復帰信号とする原点復帰方式																																				
DEC2+C相パルス	減速LSを領域信号とし、C相信号を原点信号とする原点復帰方式																																				
DEC1+LMT+C相パルス	減速LSと2つの原点復帰用リミット信号（LMT）を領域信号とし、C相信号を原点復帰信号とする原点復帰方式																																				
C相パルス	C相パルスだけによる方式																																				
POT&C相パルス	正側OT信号とC相パルスによる方式																																				
POT	正側OT信号だけによる方式																																				
HOMELS&C相パルス	HOME信号とC相パルスによる方式																																				
HOMELS	HOME信号だけによる方式																																				
NOT&C相パルス	逆側OT信号とC相パルスによる方式																																				
NOT	逆側OT信号だけによる方式																																				
INPUT&C相パルス	入力信号とC相パルスによる方式																																				
INPUT	入力信号だけによる方式																																				

<p>(前ページの続き) パラメーターと処理内容</p>	<p>② 速度 原点復帰動作時の速度を指定してください。</p> <p>③ 加速時間 原点復帰の加速時間／加速度を指定してください。</p> <p>④ 減速時間 原点復帰の減速時間／減速度を指定してください。</p> <p>⑤ 原点復帰方向 原点復帰動作時の移動方向を指定してください。移動方向は、以下の2種類があります。</p> <table border="1" data-bbox="625 660 1209 788"> <thead> <tr> <th>選択項目</th> <th>備考</th> </tr> </thead> <tbody> <tr> <td>逆転</td> <td>初期値</td> </tr> <tr> <td>正転</td> <td></td> </tr> </tbody> </table> <p>⑥ アプローチ速度 原点復帰動作で、減速LSを踏んでからの移動速度を指定してください。</p> <p>⑦ クリープ速度 原点復帰動作で、原点信号を検出してから原点位置に移動するときの速度を指定してください。</p> <p>⑧ 最終走行距離 原点復帰動作で、原点信号を検出してから移動量を指定してください。移動完了した位置が機械座標系の原点となります。</p> <p>&lt;注意事項&gt; 原点復帰完了後は、必ずNOPを実行してください。</p>	選択項目	備考	逆転	初期値	正転	
選択項目	備考						
逆転	初期値						
正転							
<p>フラグの設定</p>	<p>動作フラグ：原点復帰開始でON、終了でOFFします。 原点復帰に失敗したときにOFFします。</p> <p>一時停止フラグ：常時OFFです。</p> <p>エラーフラグ：原点復帰に失敗したときにONします。</p> <p>&lt;通常&gt;</p> <div data-bbox="399 1339 1436 1966"> </div>						



使用例



命令：原点復帰  
 原点復帰方法：DEC1+C相パルス  
 軸：Axis1  
 パラメータブロック：PB1  
 速度：[500000]  
 加速時間：[10000]  
 減速時間：[10000]  
 原点復帰方向：逆方向  
 アプローチ速度：[100000]  
 クリープ速度：[50000]  
 最終走行距離：[20000]

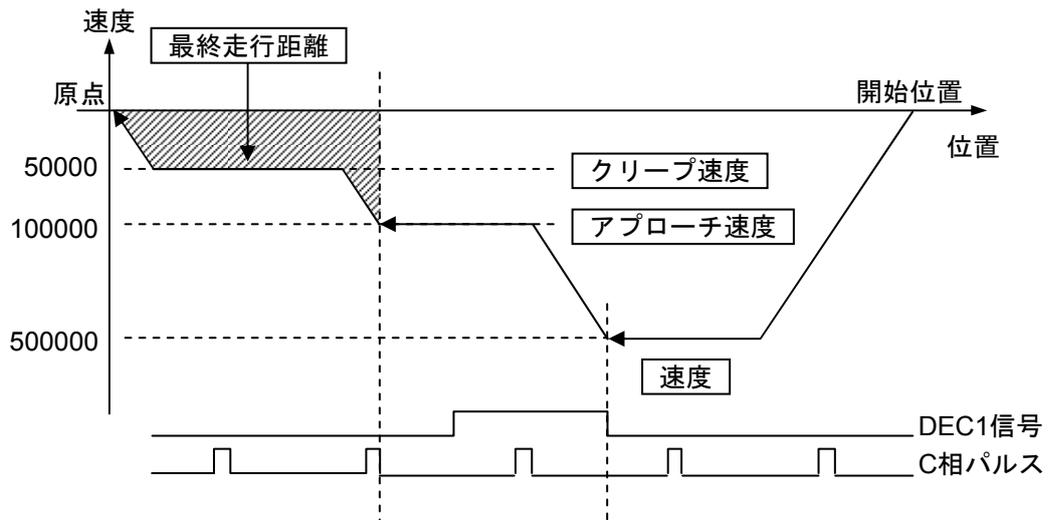
----- 原点復帰開始

R000=0

----- 原点復帰完了待ち

命令：NOP  
 軸：Axis1

----- NOP実行



## 6 モーション制御命令

(5/5)

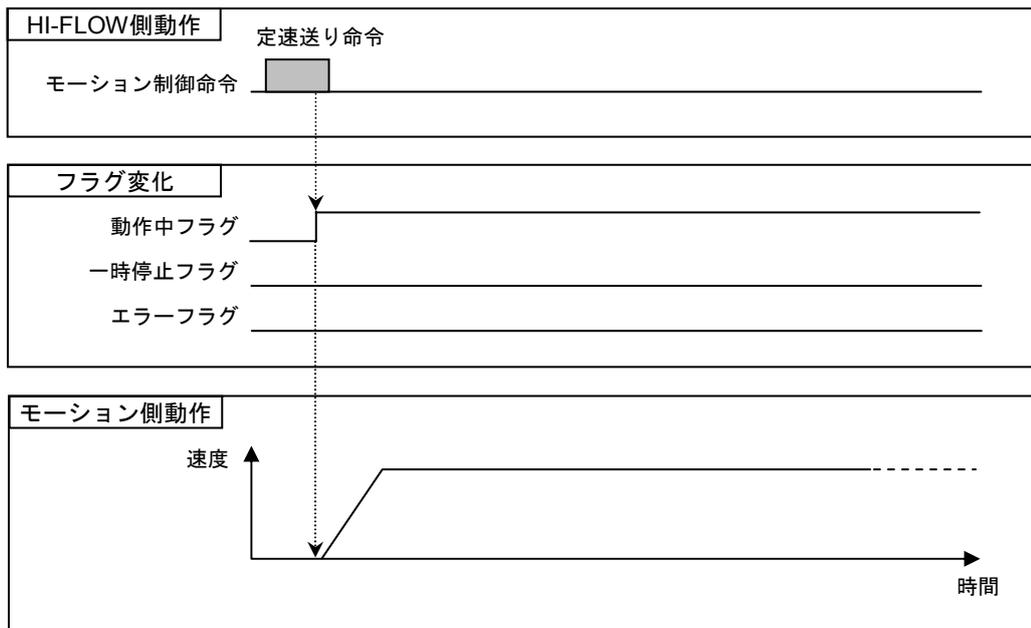
有効 パラメーター	No.	パラメーター名	PI/O種別	設定範囲
	①	原点復帰方法	ワード	0 : DEC1+C相パルス方式 1 : ZERO信号方式 2 : DEC1+ZERO信号方式 3 : C相パルス方式 4 : DEC2+ZERO信号 5 : DEC1+LMT+ZERO信号 6 : DEC2+C相パルス 7 : DEC1+LMT+C相パルス 11 : C相パルス 12 : POT&C相パルス 13 : POT 14 : HOME LS&C相パルス 15 : HOME LS 16 : NOT&C相パルス 17 : NOT 18 : INPUT&C相パルス 19 : INPUT
	②	速度	ロング	-2147483648～2147483647
	③	加速時間	ロング	0～2147483647
	④	減速時間	ロング	0～2147483647
	⑤	原点復帰方向	ワード	0 : 逆転 1 : 正転
	⑥	アプローチ速度	ロング	-2147483648～2147483647
	⑦	クリーブ速度	ロング	-2147483648～2147483647
	⑧	最終走行距離	ロング	0～2147483647
	(PI/Oをインダイレクト指定した場合、奇数アドレス値指定でパラメーターエラー)			

FEED	定速送り						
機能説明	<p>指定した方向と速度で移動します。            コマンド中断命令が実行されるまで移動を続けます。</p>						
パラメータと処理内容	<p> 命令：定速送り            軸：定速送りを実行する軸番号            パラメータブロック：①～④を省略時に使用されるパラメータグループの番号            速度：①参照            加速時間：②参照            減速時間：③参照            移動方向：④参照</p> <p>① 速度            定速送り動作時の速度を指定してください。</p> <p>② 加速時間            定速送り開始時の加速時間／加速度を指定してください。</p> <p>③ 減速時間            定速送り停止時の減速時間／減速度を指定してください。</p> <p>④ 移動方向            定速送りの移動方向を指定してください。移動方向には、以下の2種類があります。</p> <table border="1" data-bbox="624 1200 1211 1323"> <thead> <tr> <th>選択項目</th> <th>備考</th> </tr> </thead> <tbody> <tr> <td>正転</td> <td>初期値</td> </tr> <tr> <td>逆転</td> <td></td> </tr> </tbody> </table>	選択項目	備考	正転	初期値	逆転	
選択項目	備考						
正転	初期値						
逆転							
フラグの設定	<p>動作フラグ：定速送り開始でON、終了でOFFします。            定速送りに失敗したときにOFFします。</p> <p>一時停止フラグ：常時OFFです。</p> <p>エラーフラグ：定速送りに失敗したときにONします。</p>						

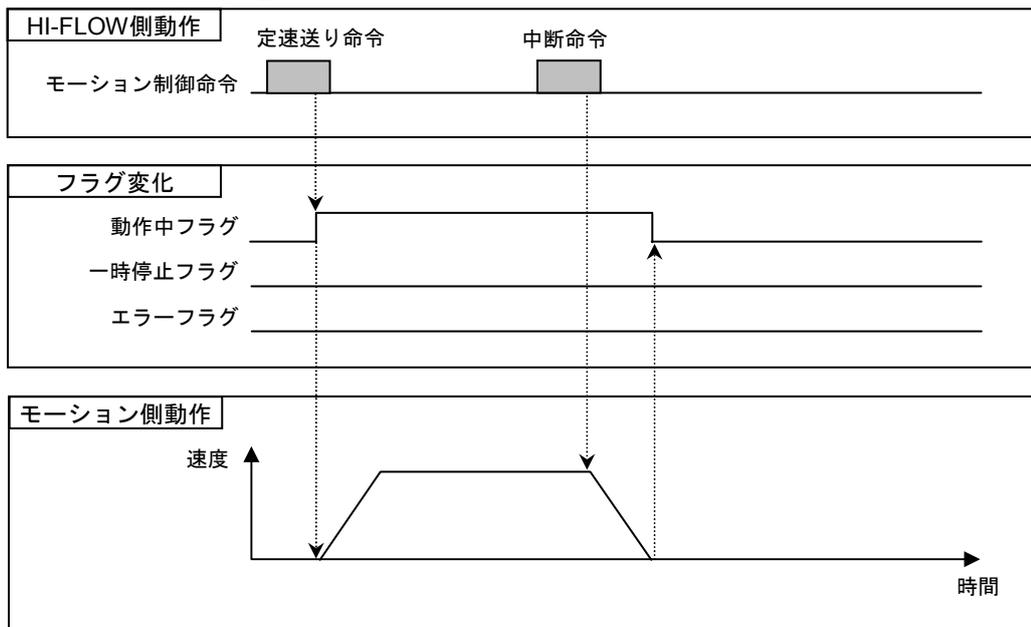
(前ページの  
続き)  
フラグの設定

動作フラグ：定速送り開始でON、終了でOFFします。  
定速送りに失敗したときにOFFします。  
一時停止フラグ：常時OFFです。  
エラーフラグ：定速送りに失敗したときにONします。

<通常>



<中断/コマンド上書き>



<p>(前ページの 続き) フラグの設定</p>	<p>&lt;異常終了&gt;</p> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <p>HI-FLOW側動作</p> <p>定速送り命令</p> <p>モーション制御命令</p> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <p>フラグ変化</p> <p>動作中フラグ</p> <p>一時停止フラグ</p> <p>エラーフラグ</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p>モーション側動作</p> <p>速度</p> <p>エラー発生</p> <p>時間</p> </div>
<p>備考</p>	
<p>使用例</p>	<p>●</p> <p>(M) 命令 : 定速送り                    ----- 定速送り開始          軸 : Axis1          パラメータブロック : PB1          速度 : [500000]          加速時間 : [10000]          減速時間 : [10000]          移動方向 : 正転</p> <p>— WT000 (100)                    ----- 10秒ウェイト</p> <p>(M) 命令 : コマンド 中断                ----- 定速送り中断          軸 : Axis1</p> <p>— R000=0                    ----- 定速送り完了待ち</p> <p>●</p>

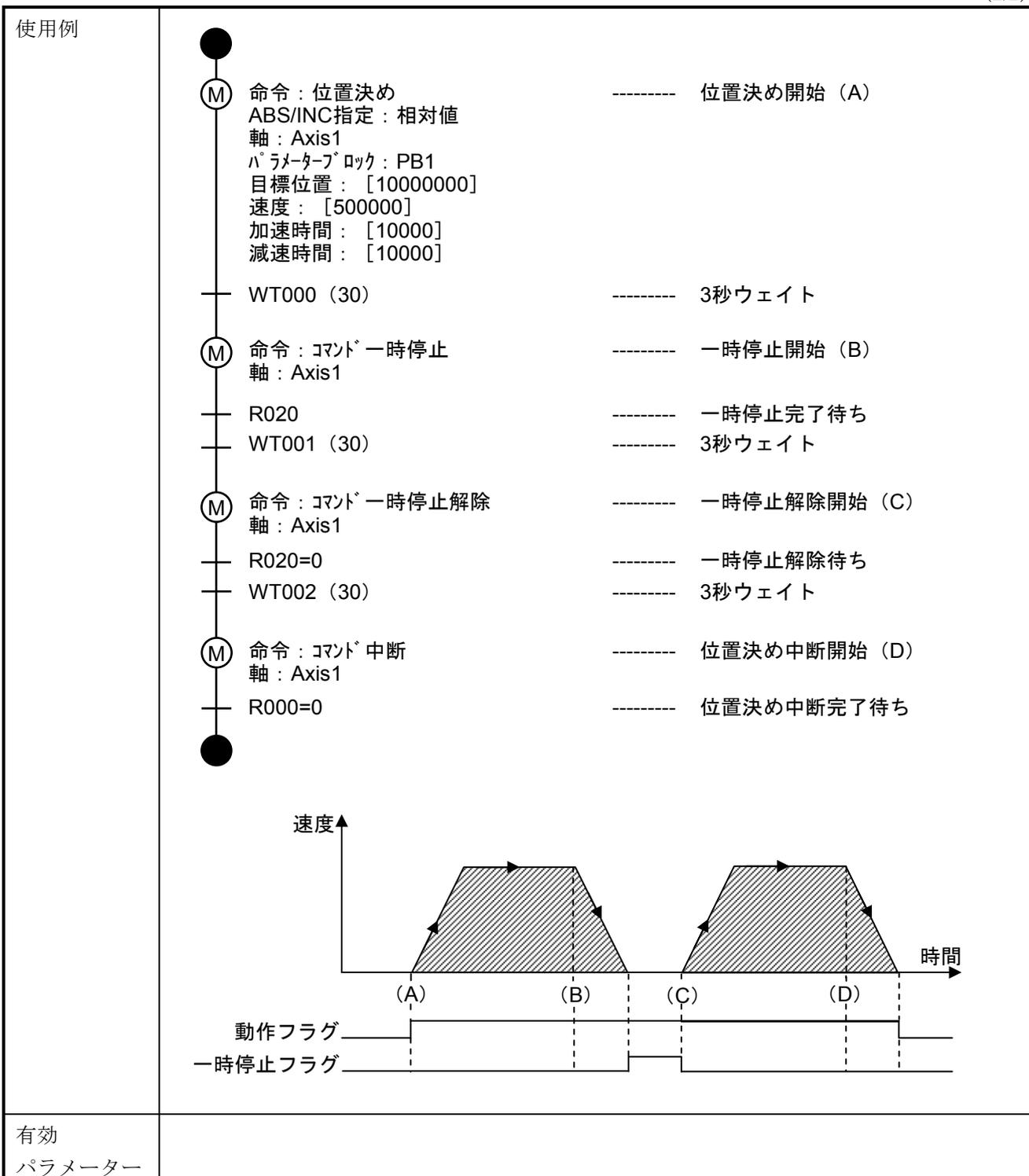
## 6 モーション制御命令

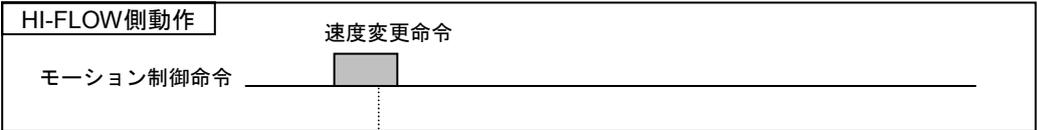
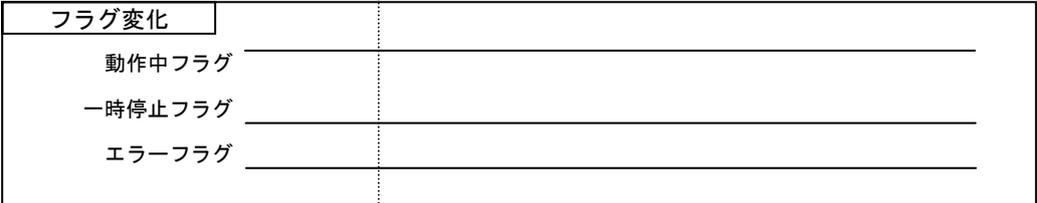
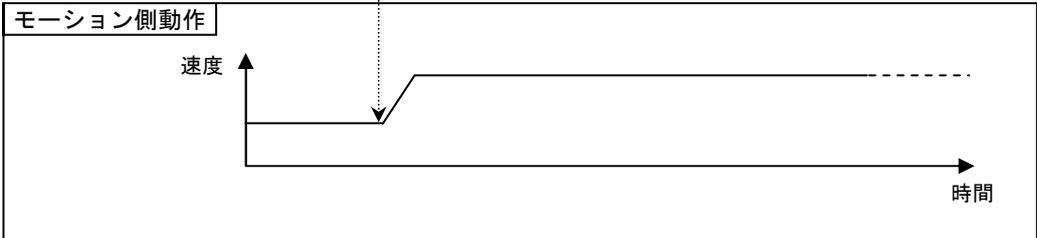
(4/4)

<p>(前ページの 続き) 使用例</p>																					
<p>有効 パラメーター</p>	<table border="1"> <thead> <tr> <th>No.</th> <th>パラメーター名</th> <th>PI/O種別</th> <th>設定範囲</th> </tr> </thead> <tbody> <tr> <td>①</td> <td>速度</td> <td>ロング</td> <td>-2147483648～2147483647</td> </tr> <tr> <td>②</td> <td>加速時間</td> <td>ロング</td> <td>0～2147483647</td> </tr> <tr> <td>③</td> <td>減速時間</td> <td>ロング</td> <td>0～2147483647</td> </tr> <tr> <td>④</td> <td>移動方向</td> <td>ワード</td> <td>0：正転 1：逆転</td> </tr> </tbody> </table> <p>(PI/Oをインダイレクト指定した場合、奇数アドレス値指定でパラメーターエラー)</p>	No.	パラメーター名	PI/O種別	設定範囲	①	速度	ロング	-2147483648～2147483647	②	加速時間	ロング	0～2147483647	③	減速時間	ロング	0～2147483647	④	移動方向	ワード	0：正転 1：逆転
No.	パラメーター名	PI/O種別	設定範囲																		
①	速度	ロング	-2147483648～2147483647																		
②	加速時間	ロング	0～2147483647																		
③	減速時間	ロング	0～2147483647																		
④	移動方向	ワード	0：正転 1：逆転																		

(1/2)

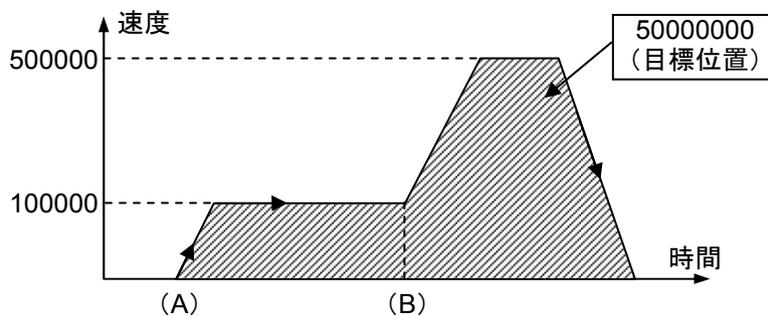
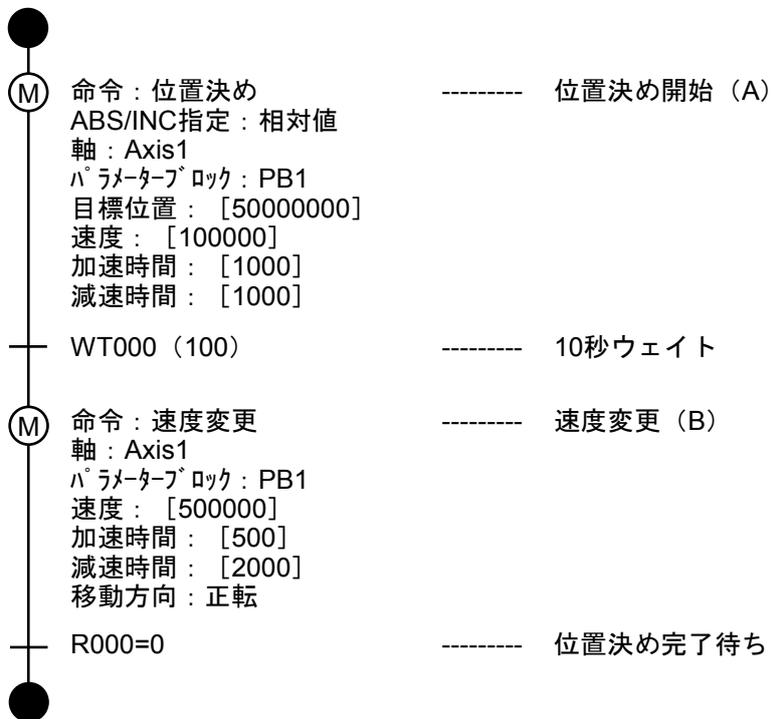
ABORT HOLDS HOLDE	コマンド中断 コマンド一時停止 コマンド一時停止解除																		
機能説明	実行中のモーション制御命令を中断／一時停止／一時停止解除します。																		
パラメータ と処理内容	 命令：コマンド中断／一時停止／一時停止解除 軸：コマンドの動作を変更する軸番号（指定範囲は1～32）																		
フラグの設定	動作フラグ：コマンド中断が完了でOFFします。 一時停止フラグ：コマンド一時停止開始でONします。 コマンド一時停止解除が完了でOFFします。 エラーフラグ：コマンド中断／一時停止／一時停止解除が失敗したときにONします。																		
備考	コマンドを中断／一時停止／一時停止解除コマンドは、以下のコマンドで実行できます。 <table border="1" data-bbox="497 922 1337 1160"> <thead> <tr> <th>コマンド</th> <th>中断</th> <th>一時停止／解除</th> </tr> </thead> <tbody> <tr> <td>位置決め</td> <td>○</td> <td>○</td> </tr> <tr> <td>外部位置決め</td> <td>○</td> <td>○</td> </tr> <tr> <td>原点復帰</td> <td>○</td> <td>×</td> </tr> <tr> <td>定速送り</td> <td>○</td> <td>×</td> </tr> <tr> <td>トルク指令</td> <td>○</td> <td>×</td> </tr> </tbody> </table> <p>○：実行できます。          ×：実行できません。</p>	コマンド	中断	一時停止／解除	位置決め	○	○	外部位置決め	○	○	原点復帰	○	×	定速送り	○	×	トルク指令	○	×
コマンド	中断	一時停止／解除																	
位置決め	○	○																	
外部位置決め	○	○																	
原点復帰	○	×																	
定速送り	○	×																	
トルク指令	○	×																	



CHGV	速度変更						
機能説明	移動中の速度を変更します。						
パラメータと処理内容	<p>  命令：速度変更              軸：速度を変更する軸番号              パラメータブロック：①～④を省略時に使用されるパラメータグループの番号              速度：①参照              加速時間：②参照              減速時間：③参照              移動方向：④参照         </p> <p>             ① 速度              変更後の速度を指定してください。         </p> <p>             ② 加速時間              変更後の加速時間／加速度を指定してください。         </p> <p>             ③ 減速時間              変更後の減速時間／減速度を指定してください。         </p> <p>             ④ 移動方向              変更後の移動方向を指定してください。移動方向には、以下の2種類があります。         </p> <table border="1" data-bbox="624 1099 1211 1225"> <thead> <tr> <th>選択項目</th> <th>備考</th> </tr> </thead> <tbody> <tr> <td>正転</td> <td>初期値</td> </tr> <tr> <td>逆転</td> <td></td> </tr> </tbody> </table>	選択項目	備考	正転	初期値	逆転	
選択項目	備考						
正転	初期値						
逆転							
フラグの設定	<p>             動作フラグ：変化しません。              一時停止フラグ：常時OFFです。              エラーフラグ：速度変更に失敗したときにONします。              &lt;通常&gt;         </p> <div data-bbox="400 1402 1437 1532"> <p>HI-FLOW側動作</p>  </div> <div data-bbox="400 1559 1437 1762"> <p>フラグ変化</p>  </div> <div data-bbox="400 1789 1437 2027"> <p>モーション側動作</p>  </div>						

<p>(前ページの続き) フラグの設定</p>	<p>&lt;異常終了&gt;</p> <p>The diagram consists of three vertically stacked panels. The top panel, titled 'HI-FLOW側動作', shows a 'モーション制御命令' (Motion Control Command) as a rectangular pulse. A '速度変更命令' (Velocity Change Command) is indicated by a vertical dashed line at the end of the pulse. The middle panel, titled 'フラグ変化' (Flag Change), shows three flags: '動作中フラグ' (Operation in Progress Flag), '一時停止フラグ' (Temporary Stop Flag), and 'エラーフラグ' (Error Flag). The error flag transitions from low to high at the end of the motion control pulse. The bottom panel, titled 'モーション側動作' (Motion Side Action), shows a velocity profile with '速度' (Velocity) on the vertical axis and '時間' (Time) on the horizontal axis. A dashed line indicates a step increase in velocity at the end of the motion control pulse.</p>																														
<p>備考</p>	<p>実行中コマンドに対する速度変更の各パラメーターの有効／無効の一覧を示します。</p> <table border="1" data-bbox="443 1153 1300 1400"> <thead> <tr> <th>コマンド</th> <th>移動方向</th> <th>速度</th> <th>加速時間</th> <th>減速時間</th> </tr> </thead> <tbody> <tr> <td>位置決め</td> <td>×</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>外部位置決め</td> <td>×</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>原点復帰</td> <td>×</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>定速送り</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>トルク指令</td> <td>×</td> <td>○</td> <td>×</td> <td>×</td> </tr> </tbody> </table> <p>○：有効 ×：無効</p>	コマンド	移動方向	速度	加速時間	減速時間	位置決め	×	○	○	○	外部位置決め	×	○	○	○	原点復帰	×	○	○	○	定速送り	○	○	○	○	トルク指令	×	○	×	×
コマンド	移動方向	速度	加速時間	減速時間																											
位置決め	×	○	○	○																											
外部位置決め	×	○	○	○																											
原点復帰	×	○	○	○																											
定速送り	○	○	○	○																											
トルク指令	×	○	×	×																											

(前ページの  
続き)  
使用例



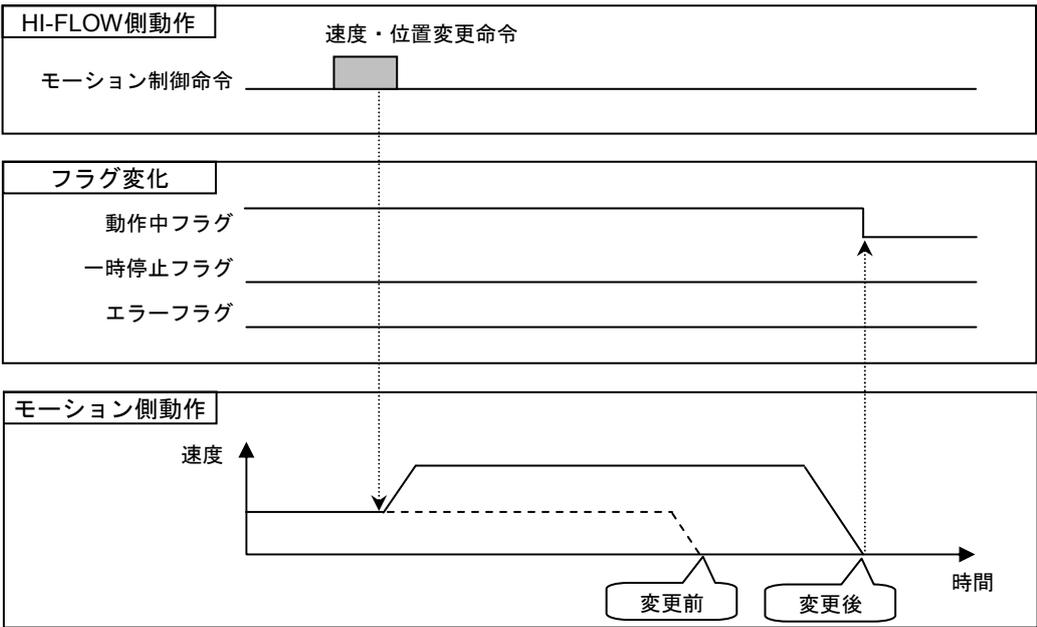
有効  
パラメーター

No.	パラメーター名	PI/O種別	設定範囲
①	速度	ロング	-2147483648～2147483647
②	加速時間	ロング	0～2147483647
③	減速時間	ロング	0～2147483647
④	移動方向	ワード	0：正転 1：逆転

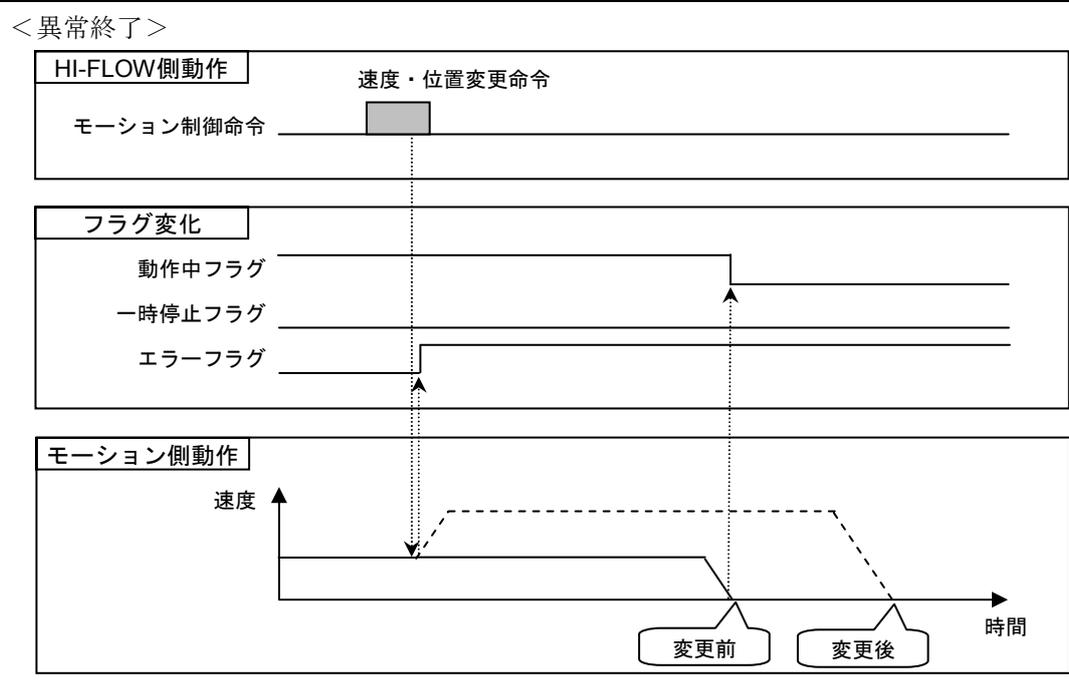
(PI/Oをインダイレクト指定した場合、奇数アドレス値指定でパラメーターエラー)

## 6 モーション制御命令

(1/3)

CHGVP	速度・位置変更
機能説明	動作中の目標位置と位置決め速度を変更します。
パラメーターと処理内容	<p>  命令：速度・位置変更              軸：速度・位置を変更する軸番号              パラメーターブロック：①～④を省略時に使用されるパラメーターグループの番号              目標位置：①参照              速度：②参照              加速時間：③参照              減速時間：④参照         </p> <p>             ① 目標位置              変更後の目標位置を指定してください。         </p> <p>             ② 速度              変更後の速度を指定してください。         </p> <p>             ③ 加速時間              変更後の加速時間／加速度を指定してください。         </p> <p>             ④ 減速時間              変更後の減速時間／減速度を指定してください。         </p>
フラグの設定	<p>             動作フラグ：変化しません。              一時停止フラグ：常時OFFです。              エラーフラグ：速度・位置変更に失敗したときにONします。         </p> <p>&lt;通常&gt;</p> 

(前ページの  
続き)  
フラグの設定



備考

実行中コマンドに対する速度・位置変更の各パラメーターの有効／無効の一覧を示します。

コマンド	目標位置	速度	加速時間	減速時間
位置決め	○	○	○	○
外部位置決め	×	○	○	○
原点復帰	×	○	○	○
定速送り	×	○	○	○
トルク指令	×	○	×	×

○：有効  
×：無効

MP2300Hの位置指令タイプの設定が増分値加算方式の場合、「位置決め」または「外部位置決め」コマンドを絶対値指定で発行した後に「速度・位置変更」コマンドでの変更はできません。「速度・位置変更」コマンドを発行した場合、不正な加算方式（エラーコード=0x200A）のエラーが発生します。

使用例

●

(M) 命令：位置決め  
 ABS/INC指定：相対値  
 軸：Axis1  
 パラメータブロック：PB1  
 目標位置：[50000000]  
 速度：[100000]  
 加速時間：[1000]  
 減速時間：[1000]

----- 位置決め開始 (A)

WT000 (100) ----- 10秒ウェイト

(M) 命令：速度・位置変更  
 軸：Axis1  
 パラメータブロック：PB1  
 目標位置：[100000000]  
 速度：[500000]  
 加速時間：[500]  
 減速時間：[2000]

----- 速度・位置変更 (B)

R000=0 ----- 位置決め完了待ち

●

速度

500000

100000

100000000 (目標位置)

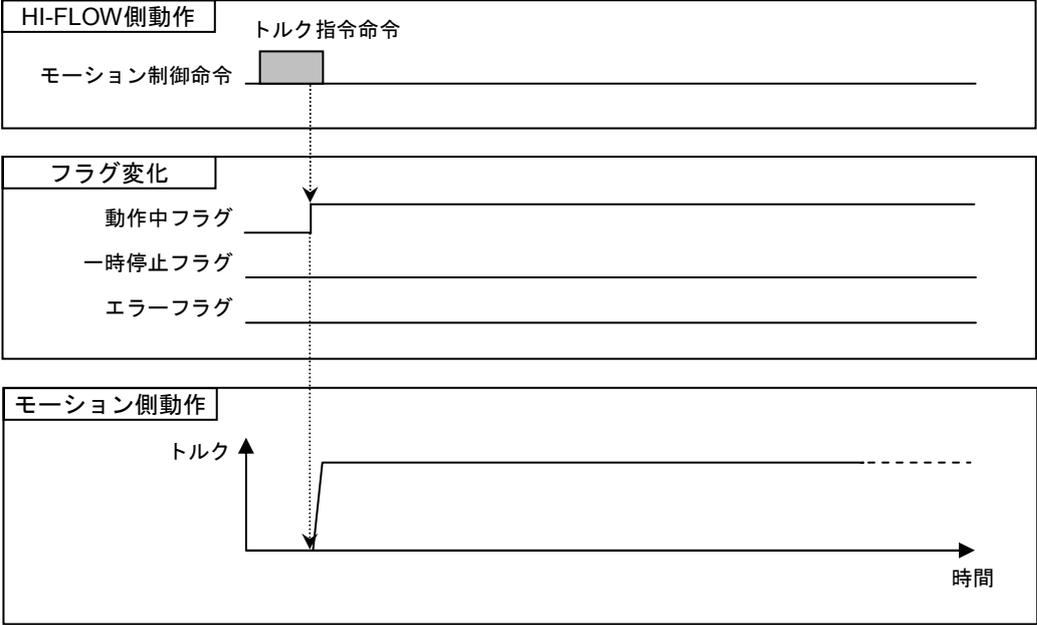
時間

(A) (B)

有効  
パラメーター

No.	パラメーター名	PI/O種別	設定範囲
①	目標位置	ロング	-2147483648～2147483647
②	速度	ロング	-2147483648～2147483647
③	加速時間	ロング	0～2147483647
④	減速時間	ロング	0～2147483647

(PI/Oをインダイレクト指定した場合、奇数アドレス値指定でパラメーターエラー)

TRQ	トルク指令
機能説明	トルク制御モードで運転します。
パラメータと処理内容	<p>Ⓜ 命令：トルク制御          軸：トルクを制御する軸番号          パラメータブロック：①～②を省略時に使用されるパラメータグループの番号          トルク値：①参照          速度制限値：②参照</p> <p>① トルク値          変更後のトルク値を指定してください。</p> <p>② 速度制限値          変更後の速度を指定してください。</p>
フラグの設定	<p>動作フラグ：トルク制御に失敗したときにOFFします。          一時停止フラグ：常時OFFです。          エラーフラグ：トルク制御に失敗したときにONします。</p> <p>&lt;通常&gt;</p>  <p>The diagram illustrates the timing of the TRQ command and its effects. It is divided into three sections:</p> <ul style="list-style-type: none"> <li><b>HI-FLOW側動作 (HI-FLOW Side Action):</b> Shows the 'トルク指令命令' (Torque Command) as a pulse. The 'モーション制御命令' (Motion Control Command) is active during this pulse.</li> <li><b>フラグ変化 (Flag Change):</b> Shows the state of three flags: '動作中フラグ' (Operation Flag), '一時停止フラグ' (Stop Flag), and 'エラーフラグ' (Error Flag). The '動作中フラグ' transitions from OFF to ON at the start of the torque command and back to OFF at the end. The '一時停止フラグ' remains OFF. The 'エラーフラグ' transitions from OFF to ON at the end of the torque command and back to OFF at the start of the next motion command.</li> <li><b>モーション側動作 (Motion Side Action):</b> Shows the 'トルク' (Torque) over '時間' (Time). The torque starts at zero, rises to a constant value during the motion command, and then returns to zero.</li> </ul>

<p>(前ページの 続き) フラグの設定</p>	<p>&lt; 中断/コマンド上書き &gt;</p> <div data-bbox="354 383 1390 517"> <p>HI-FLOW側動作</p> </div> <div data-bbox="354 539 1390 745"> <p>フラグ変化</p> </div> <div data-bbox="354 768 1390 1010"> <p>モーション側動作</p> </div> <p>&lt; 異常終了 &gt;</p> <div data-bbox="354 1115 1390 1249"> <p>HI-FLOW側動作</p> </div> <div data-bbox="354 1272 1390 1478"> <p>フラグ変化</p> </div> <div data-bbox="354 1500 1390 1742"> <p>モーション側動作</p> </div>
<p>備考</p>	<ul style="list-style-type: none"> <li>・サーボOFF中でもコマンドを実行できます。</li> <li>・トルク制御を終了する場合は、コマンド中断コマンドを実行してください。</li> </ul>

使用例	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="margin-bottom: 20px;"> <p>●</p> <p>(M) 命令：トルク指令 軸：Axis1 パラメータブロック：PB1 トルク値：[1000] 速度制限値：1000</p> <p>----- トルク指令開始 (A)</p> <p>WT000 (100) ----- 10秒ウェイト</p> </div> <div style="margin-bottom: 20px;"> <p>(M) 命令：トルク指令 軸：Axis1 パラメータブロック：PB1 トルク値：[3000] 速度制限値：3000</p> <p>----- トルク指令変更 (B)</p> <p>WT001 (100) ----- 10秒ウェイト</p> </div> <div> <p>(M) 命令：コマンド中断 軸：Axis1</p> <p>----- トルク指令終了 (C)</p> <p>R000=0 ----- トルク指令終了待ち</p> <p>●</p> </div> </div> <div style="margin-top: 20px;"> <p>トルク</p> <p>3000</p> <p>1000</p> <p>時間</p> <p>(A) (B) (C)</p> </div>												
有効 パラメーター	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 5%;">No.</th> <th style="width: 35%;">パラメーター名</th> <th style="width: 15%;">PI/O種別</th> <th style="width: 45%;">設定範囲</th> </tr> </thead> <tbody> <tr> <td>①</td> <td>トルク値</td> <td>ロング</td> <td>-2147483648～2147483647</td> </tr> <tr> <td>②</td> <td>速度制限値</td> <td>ワード</td> <td>-32768～32767</td> </tr> </tbody> </table> <p>(PI/Oをインダイレクト指定した場合、奇数アドレス値指定でパラメーターエラー)</p>	No.	パラメーター名	PI/O種別	設定範囲	①	トルク値	ロング	-2147483648～2147483647	②	速度制限値	ワード	-32768～32767
No.	パラメーター名	PI/O種別	設定範囲										
①	トルク値	ロング	-2147483648～2147483647										
②	速度制限値	ワード	-32768～32767										

## 6 モーション制御命令

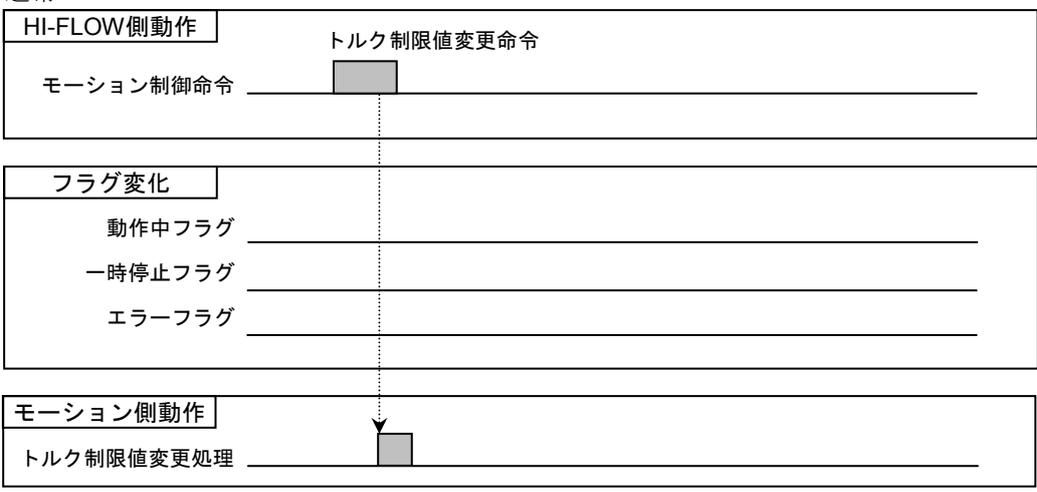
(1/3)

CHGO	速度オーバーライド
機能説明	速度設定値に対する出力割合を0.01%単位で設定します。 位置決め動作中に変更可能で、設定値によって即時に加減速処理されます。
パラメータと処理内容	<p>Ⓜ 命令：速度オーバーライド          軸：速度オーバーライドを変更する軸番号          パラメータブロック：①を省略時に使用されるパラメータグループの番号          速度オーバーライド：①参照</p> <p>① 速度オーバーライド          速度設定値に対する出力割合を0.01%で指定してください。</p>
フラグの設定	<p>動作フラグ：変化しません。          一時停止フラグ：常時OFFです。          エラーフラグ：速度オーバーライドの変更に失敗したときにONします。</p> <p>&lt;通常&gt;</p> <div data-bbox="354 1003 1390 1626" style="border: 1px solid black; padding: 5px;"> <p><b>HI-FLOW側動作</b></p> <p>速度オーバーライド命令      速度オーバーライド命令</p> <p>モーション制御命令</p> <p><b>フラグ変化</b></p> <p>動作中フラグ</p> <p>一時停止フラグ</p> <p>エラーフラグ</p> <p><b>モーション側動作</b></p> <p>速度 ↑</p> <p>時間 →</p> </div>

(2/3)

<p>(前ページの 続き) フラグの設定</p>	<p>&lt;異常終了&gt;</p>
<p>備考</p>	<ul style="list-style-type: none"> <li>速度オーバーライドは常に有効です。速度オーバーライド機能を使用しない場合は、10000 (=100.00%) 固定としてください。</li> <li>速度オーバーライドが0の場合、出力速度が0となり、モーターは動作しません。</li> </ul>

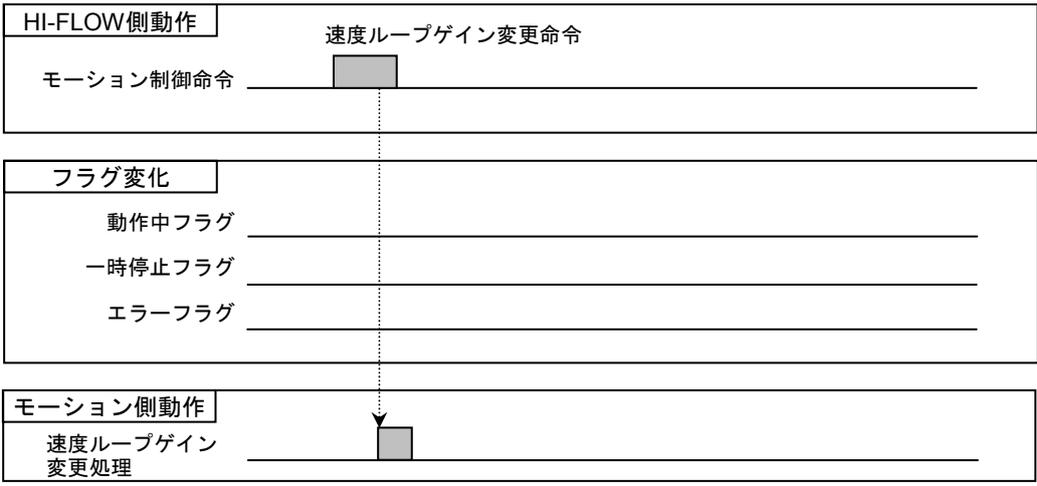


CHGTL	トルク制限値変更
機能説明	機械側の保護として、トルク指令動作時にサーボモーター回転速度の制限をかける機能です。
パラメーターと処理内容	<p>  命令：トルク制限値変更              軸：トルク制限値を変更する軸番号              パラメータブロック：①～②を省略時に使用されるパラメータグループの番号              正転側トルク制限値：①参照              逆転側トルク制限値：②参照         </p> <p>             ① 正転側トルク制限値              トルク指令動作時の正転方向の最高速度を指定してください。         </p> <p>             ② 逆転側トルク制限値              トルク指令動作時の逆転方向の最高速度を指定してください。         </p>
フラグの設定	<p>             動作フラグ：変化しません。              一時停止フラグ：常時OFFです。              エラーフラグ：トルク制限値の変更に失敗したときにONします。         </p> <p>&lt;通常&gt;</p> <div style="border: 1px solid black; padding: 5px;"> <p>HI-FLOW側動作</p> <p style="text-align: center;">トルク制限値変更命令</p> <p>モーション制御命令 <span style="display: inline-block; width: 100px; height: 10px; background-color: gray; vertical-align: middle;"></span></p> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>フラグ変化</p> <p>動作中フラグ _____</p> <p>一時停止フラグ _____</p> <p>エラーフラグ _____</p> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>モーション側動作</p> <p>トルク制限値変更処理 <span style="display: inline-block; width: 100px; height: 10px; background-color: gray; vertical-align: middle;"></span></p> </div> 

## 6 モーション制御命令

(2/2)

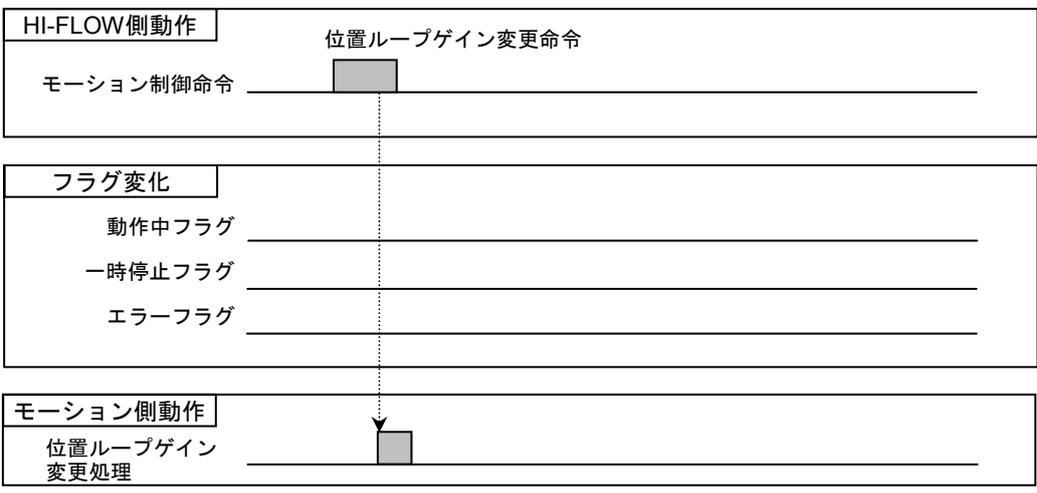
<p>(前ページの続き) フラグの設定</p>	<p>&lt;異常終了&gt;</p>												
<p>備考</p>													
<p>使用例</p>	<p>命令 : トルク制限値変更 軸 : Axis1 パラメータブロック : PB1 正転側トルク制限値 : [800] 逆転側トルク制限値 : [800]</p> <p>R000=0</p>												
<p>有効 パラメーター</p>	<table border="1"> <thead> <tr> <th>No.</th> <th>パラメーター名</th> <th>PI/O種別</th> <th>設定範囲</th> </tr> </thead> <tbody> <tr> <td>①</td> <td>正転側トルク制限値</td> <td>ロング</td> <td>0～800</td> </tr> <tr> <td>②</td> <td>逆転側トルク制限値</td> <td>ロング</td> <td>0～800</td> </tr> </tbody> </table> <p>(PI/Oをインダイレクト指定した場合、奇数アドレス値指定でパラメーターエラー)</p>	No.	パラメーター名	PI/O種別	設定範囲	①	正転側トルク制限値	ロング	0～800	②	逆転側トルク制限値	ロング	0～800
No.	パラメーター名	PI/O種別	設定範囲										
①	正転側トルク制限値	ロング	0～800										
②	逆転側トルク制限値	ロング	0～800										

KVS	速度ループゲイン変更
機能説明	サーボモーターの速度ループゲインを設定します。
パラメーターと処理内容	<p>              命令：速度ループゲイン変更              軸：速度ループゲインを変更する軸番号              パラメータブロック：①～③を省略時に使用されるパラメータグループの番号              速度ループゲイン：①参照              速度フィードフォワード：②参照              速度積分時定数：③参照         </p> <p>             ① 速度ループゲイン              サーボパックの速度ループの応答性を設定します。機械系が振動しない範囲で、できるだけ高めに設定したほうがサーボ系が安定します。単位はHzです。         </p> <p>             ② 速度フィードフォワード              フィードフォワード補償をかけることによって、位置決め時間を短縮できます。単位は0.01%です。         </p> <p>             ③ 速度積分時定数              微小入力に対する応答性を設定します。ただしサーボ系では遅れの要素になるため、時定数を大きくすると応答性が悪くなります。単位は0.01msです。         </p>
フラグの設定	<p>             動作フラグ：変化しません。              一時停止フラグ：常時OFFです。              エラーフラグ：速度ループゲインの変更に失敗したときにONします。         </p> <p>&lt;通常&gt;</p> <div style="border: 1px solid black; padding: 5px;"> <p>HI-FLOW側動作</p> <p style="text-align: center;">速度ループゲイン変更命令</p> <p>モーション制御命令 <span style="display: inline-block; width: 100px; height: 10px; background-color: gray; vertical-align: middle;"></span></p> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>フラグ変化</p> <p>動作中フラグ _____</p> <p>一時停止フラグ _____</p> <p>エラーフラグ _____</p> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>モーション側動作</p> <p>速度ループゲイン変更処理 <span style="display: inline-block; width: 50px; height: 10px; background-color: gray; vertical-align: middle;"></span></p> </div> 

## 6 モーション制御命令

(2/2)

<p>(前ページの続き) フラグの設定</p>	<p>&lt;異常終了&gt;</p>																
<p>備考</p>																	
<p>使用例</p>	<p>命令：速度ループゲイン変更 軸：Axis1 パラメータブロック：PB1 速度ループゲイン：500 速度フィードフォワード：500 速度積分時定数：2000</p> <p>R000=0</p>																
<p>有効 パラメーター</p>	<table border="1"> <thead> <tr> <th>No.</th> <th>パラメーター名</th> <th>PI/O種別</th> <th>設定範囲</th> </tr> </thead> <tbody> <tr> <td>①</td> <td>速度ループゲイン</td> <td>ワード</td> <td>0~2000</td> </tr> <tr> <td>②</td> <td>速度フィードフォワード</td> <td>ワード</td> <td>0~32767</td> </tr> <tr> <td>③</td> <td>速度積分時定数</td> <td>ワード</td> <td>15~65535</td> </tr> </tbody> </table> <p>(PI/Oをインダイレクト指定した場合、奇数アドレス値指定でパラメーターエラー)</p>	No.	パラメーター名	PI/O種別	設定範囲	①	速度ループゲイン	ワード	0~2000	②	速度フィードフォワード	ワード	0~32767	③	速度積分時定数	ワード	15~65535
No.	パラメーター名	PI/O種別	設定範囲														
①	速度ループゲイン	ワード	0~2000														
②	速度フィードフォワード	ワード	0~32767														
③	速度積分時定数	ワード	15~65535														

KPS	位置ループゲイン変更
機能説明	サーボモーターの位置ループゲインを設定します。
パラメーターと処理内容	<p>  命令：位置ループゲイン変更              軸：位置ループゲインを変更する軸番号              パラメーターブロック：①～②を省略時に使用されるパラメーターグループの番号              位置ループゲイン：①参照              位置積分時定数：②参照         </p> <p>             ① 位置ループゲイン              サーボパックの位置ループの応答性を設定します。位置ループゲインを高く設定すれば応答性が高くなり、位置決め時間が短くなります。機械の剛性、イナーシャおよびサーボモータの種類に応じて最適値を設定してください。単位は0.1/sです。         </p> <p>             ② 位置積分時定数              電子カム、電子シャフトなどで追従精度を向上させるときに使用します。単位はmsです。         </p>
フラグの設定	<p>             動作フラグ：変化しません。              一時停止フラグ：常時OFFです。              エラーフラグ：位置ループゲインの変更に失敗したときにONします。         </p> <p>&lt;通常&gt;</p> <div style="border: 1px solid black; padding: 5px;"> <p>HI-FLOW側動作</p> <p style="text-align: center;">位置ループゲイン変更命令</p> <p>モーション制御命令 <span style="display: inline-block; width: 100px; height: 10px; background-color: gray; vertical-align: middle;"></span></p> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>フラグ変化</p> <p>動作中フラグ _____</p> <p>一時停止フラグ _____</p> <p>エラーフラグ _____</p> </div> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>モーション側動作</p> <p>位置ループゲイン変更処理 <span style="display: inline-block; width: 100px; height: 10px; background-color: gray; vertical-align: middle;"></span></p> </div> 

## 6 モーション制御命令

(2/2)

<p>(前ページの続き) フラグの設定</p>	<p>&lt;異常終了&gt;</p>												
<p>備考</p>													
<p>使用例</p>	<p>●</p> <p>(M) 命令：位置ループゲイン変更 軸：Axis1 パラメータブロック：PB1 位置ループゲイン：1000 位置積分時定数：0</p> <p>— R000=0</p> <p>----- 位置ループゲイン変更開始</p> <p>----- 位置ループゲイン変更完了待ち</p> <p>●</p>												
<p>有効 パラメーター</p>	<table border="1"> <thead> <tr> <th>No.</th> <th>パラメーター名</th> <th>PI/O種別</th> <th>設定範囲</th> </tr> </thead> <tbody> <tr> <td>①</td> <td>位置ループゲイン</td> <td>ワード</td> <td>0～32767</td> </tr> <tr> <td>②</td> <td>位置積分時定数</td> <td>ワード</td> <td>0～32767</td> </tr> </tbody> </table> <p>(PI/Oをインダイレクト指定した場合、奇数アドレス値指定でパラメーターエラー)</p>	No.	パラメーター名	PI/O種別	設定範囲	①	位置ループゲイン	ワード	0～32767	②	位置積分時定数	ワード	0～32767
No.	パラメーター名	PI/O種別	設定範囲										
①	位置ループゲイン	ワード	0～32767										
②	位置積分時定数	ワード	0～32767										

CHGU	単位設定																														
機能説明	各種パラメーターの制御単位を設定します。																														
パラメーターと処理内容	<p>Ⓜ 命令：単位設定            軸：単位を設定する軸番号            パラメーターブロック：①～⑩省略時に使用されるパラメーターグループの番号            速度単位：①参照            加減速度単位：②参照            フィルタータイプ：③参照            トルク単位：④参照            外部位置決め信号設定：⑤参照            位置決め完了幅：⑥参照            位置決め近傍検出幅：⑦参照            加速時間：⑧参照            減速時間：⑨参照            フィルター時定数：⑩参照</p> <p>① 速度単位            速度単位を指定してください。速度単位には、以下の4種類があります。</p> <table border="1"> <thead> <tr> <th>選択項目</th> <th>備考</th> </tr> </thead> <tbody> <tr> <td>指令単位/sec</td> <td></td> </tr> <tr> <td>10**n指令単位/min</td> <td>初期値</td> </tr> <tr> <td>0.01%</td> <td></td> </tr> <tr> <td>0.0001%</td> <td></td> </tr> </tbody> </table> <p>② 加減速度単位            加減速度単位を指定してください。速度単位には、以下の2種類があります。</p> <table border="1"> <thead> <tr> <th>選択項目</th> <th>備考</th> </tr> </thead> <tbody> <tr> <td>指令単位/sec**2</td> <td></td> </tr> <tr> <td>ms</td> <td>初期値</td> </tr> </tbody> </table> <p>③ フィルタータイプ            加減速フィルターのタイプを指定してください。フィルタータイプには、以下の3種類があります。</p> <table border="1"> <thead> <tr> <th>選択項目</th> <th>備考</th> </tr> </thead> <tbody> <tr> <td>フィルターなし</td> <td>初期値</td> </tr> <tr> <td>指数関数加減速フィルター</td> <td></td> </tr> <tr> <td>移動平均フィルター</td> <td></td> </tr> </tbody> </table> <p>④ トルク単位            トルク指令の単位を指定してください。トルク単位には、以下の2種類があります。</p> <table border="1"> <thead> <tr> <th>選択項目</th> <th>備考</th> </tr> </thead> <tbody> <tr> <td>0.01%</td> <td>初期値</td> </tr> <tr> <td>0.0001%</td> <td></td> </tr> </tbody> </table>	選択項目	備考	指令単位/sec		10**n指令単位/min	初期値	0.01%		0.0001%		選択項目	備考	指令単位/sec**2		ms	初期値	選択項目	備考	フィルターなし	初期値	指数関数加減速フィルター		移動平均フィルター		選択項目	備考	0.01%	初期値	0.0001%	
選択項目	備考																														
指令単位/sec																															
10**n指令単位/min	初期値																														
0.01%																															
0.0001%																															
選択項目	備考																														
指令単位/sec**2																															
ms	初期値																														
選択項目	備考																														
フィルターなし	初期値																														
指数関数加減速フィルター																															
移動平均フィルター																															
選択項目	備考																														
0.01%	初期値																														
0.0001%																															

<p>(前ページの続き) パラメーターと処理内容</p>	<p>⑤ 外部位置決め信号設定 外部位置決めで使用する外部信号を指定してください。外部位置決め信号は、以下の4種類があります。</p> <table border="1" data-bbox="580 454 1163 674"> <thead> <tr> <th>選択項目</th> <th>備考</th> </tr> </thead> <tbody> <tr> <td>C相パルス入力信号</td> <td></td> </tr> <tr> <td>/EXT1</td> <td>初期値</td> </tr> <tr> <td>/EXT2</td> <td></td> </tr> <tr> <td>/EXT3</td> <td></td> </tr> </tbody> </table> <p>⑥ 位置決め完了幅 位置決め時、位置指令払い出し完了後に位置決め完了信号がONする範囲を設定します。システムの機械諸元に適した値を設定してください。小さすぎると位置決め完了に時間がかかります。</p> <p>⑦ 位置決め近傍検出幅 目標位置とフィードバック位置との差の絶対値が、この設定範囲内であれば、位置決め近傍がONします。</p> <p>⑧ 加速時間 位置決め時の加速時間/加速度を指定してください。単位は「② 加減速度単位」によります。</p> <p>⑨ 減速時間 位置決め時の減速時間/減速度を指定してください。単位は「② 加減速度単位」によります。</p> <p>⑩ フィルター時定数 加減速フィルター時定数を指定してください。単位は0.1msです。</p>	選択項目	備考	C相パルス入力信号		/EXT1	初期値	/EXT2		/EXT3	
選択項目	備考										
C相パルス入力信号											
/EXT1	初期値										
/EXT2											
/EXT3											
<p>フラグの設定</p>	<p>動作フラグ：単位設定開始でON、終了でOFFします。 単位の変更に失敗したときにOFFします。</p> <p>一時停止フラグ：常時OFFです。</p> <p>エラーフラグ：単位の変更に失敗したときにONします。</p> <p>&lt;通常&gt;</p> <div data-bbox="354 1458 1390 1935"> <p>The diagram illustrates the timing of various signals during a unit setting process. It is divided into three main sections:</p> <ul style="list-style-type: none"> <li><b>HI-FLOW側動作 (HI-FLOW Side Action):</b> Shows a pulse for 'モーション制御命令' (Motion Control Command) and a longer pulse for '単位設定命令' (Unit Setting Command).</li> <li><b>フラグ変化 (Flag Change):</b> Shows the state of three flags: '動作中フラグ' (Action in Progress Flag), '一時停止フラグ' (Temporary Stop Flag), and 'エラーフラグ' (Error Flag). The '動作中フラグ' transitions from OFF to ON at the start of the '単位設定命令' pulse and returns to OFF at its end. The other two flags remain OFF throughout.</li> <li><b>モーション側動作 (Motion Side Action):</b> Shows a pulse for '単位設定処理' (Unit Setting Processing) that occurs during the '単位設定命令' pulse.</li> </ul> </div>										

<p>(前ページの 続き) フラグの設定</p>	<p>&lt;異常終了&gt;</p>
<p>備考</p>	<p>加速時間および減速時間は、デフォルト設定または加減速度単位を「ms」に設定した場合、設定範囲は0～32767 (ms) となります。加速時間および減速時間に32767以上の値を指定して外部位置決めを実行すると、設定が32767にクランプして外部位置決めされ、ワーニングの「設定パラメーター設定異常」フラグがONします。</p>
<p>使用例</p>	<p>命令：単位設定 軸：Axis1 パラメータブロック：PB1 速度単位：指令単位/sec 加減速度単位：指令単位/sec**2</p> <p>----- 単位設定開始</p> <p>----- 単位設定完了待ち</p>

## 6 モーション制御命令

(4/4)

有効 パラメーター	No.	パラメーター名	PI/O種別	設定範囲
	①	速度単位	ワード	0 : 指令単位/sec 1 : 10**n指令単位/min 2 : 0.01% 3 : 0.0001%
	②	加減速度単位	ワード	0 : 指令単位/sec**2 1 : ms
	③	フィルタータイプ	ワード	0 : フィルターなし 1 : 指数関数加減速フィルター 2 : 移動平均フィルター
	④	トルク単位	ワード	0 : 0.01% 1 : 0.0001%
	⑤	外部位置決め 信号設定	ワード	2 : C相パルス入力信号 3 : /EXT1 4 : /EXT2 5 : /EXT3
	⑥	位置決め完了幅	ロング	0~65535
	⑦	位置決め近傍検出幅	ロング	0~65535
	⑧	加速時間	ロング	0~2147483647
	⑨	減速時間	ロング	0~2147483647
⑩	フィルター時定数	ワード	0~65535	

(PI/Oをインダイレクト指定した場合、奇数アドレス値指定でパラメーターエラー)

ALMCLR	アラームクリアー
機能説明	指定した軸のアラームをクリアーします。
パラメーターと処理内容	<p>Ⓜ 命令：アラームクリアー          軸：アラームをクリアーする軸番号（指定範囲は1～32）</p> <p>&lt;注意事項&gt;          アラームクリアー完了後は、必ず100ms以上ウェイトしてください。</p>
フラグの設定	<p>動作フラグ：アラームクリアー開始でON、終了でOFFします。          アラームクリアーに失敗したときにOFFします。</p> <p>一時停止フラグ：常時OFFです。</p> <p>エラーフラグ：アラームクリアーに失敗したときにONします。</p> <p>&lt;通常&gt;</p> <div data-bbox="400 916 1437 1046"> <p>HI-FLOW側動作</p> </div> <div data-bbox="400 1070 1437 1274"> <p>フラグ変化</p> </div> <div data-bbox="400 1299 1437 1393"> <p>モーション側動作</p> </div> <p>&lt;異常終了&gt;</p> <div data-bbox="400 1485 1437 1615"> <p>HI-FLOW側動作</p> </div> <div data-bbox="400 1639 1437 1843"> <p>フラグ変化</p> </div> <div data-bbox="400 1868 1437 1962"> <p>モーション側動作</p> </div>

## 6 モーション制御命令

(2/2)

備考	
使用例	<p>命令 : アラームクリアー 軸 : Axis1</p> <p>----- アラームクリアー開始</p> <p>----- アラームクリアー完了待ち</p> <p>----- 100msウェイト</p>
有効 パラメーター	

(1/1)

NOP	コマンドなし
機能説明	モーションコントローラー側に記憶しているモーション制御命令情報をクリアします。モーション制御命令実行中の場合は、実行を中断してクリアします（コマンド中断（ABORT）と同等です）。
パラメーターと処理内容	 命令：NOP 軸：軸番号（指定範囲は1～32）
フラグの設定	動作フラグ：コマンド実行中でない場合は変化しません。 コマンド実行中の場合は、コマンド中断が完了でOFFします。 エラーフラグ：制御命令情報のクリアが失敗したときにONします。
備考	外部位置決めコマンドまたは原点復帰コマンド完了後に、必ず発行してください。
使用例	外部位置決めコマンドまたは原点復帰コマンドを参照してください。
有効パラメーター	

## 6 モーション制御命令

### 6.6 サンプルプログラム

モーション制御命令を使用して2軸を位置決めするサンプルプログラムを記載します。

#### ■ 動作内容

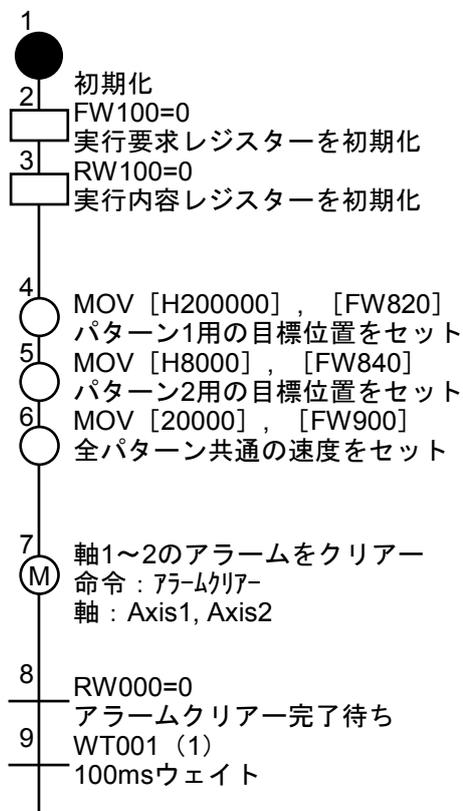
- ① FW100に1を指定すると、正転方向および逆転方向に2軸を同時に位置決めします。
- ② FW100に2を指定すると、正転方向および逆転方向に90度ずつ1回転するまで位置決めします。
- ③ FW100に0x1234を指定すると、①と②を実行しつづけます。
- ④ FW100に0を指定すると、実行中のモーション制御を強制停止します。

#### ■ 設定内容

動作フラグ : R000～R01F  
一時停止フラグ : R020～R03F  
エラーフラグ : R040～R05F

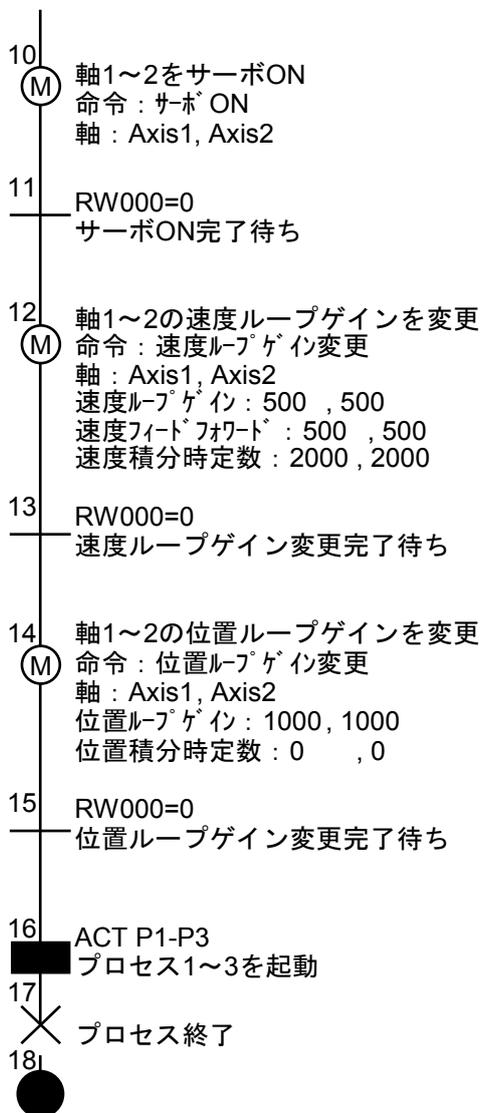
#### ■ サンプルプログラム

(1) プロセス0：初期化処理



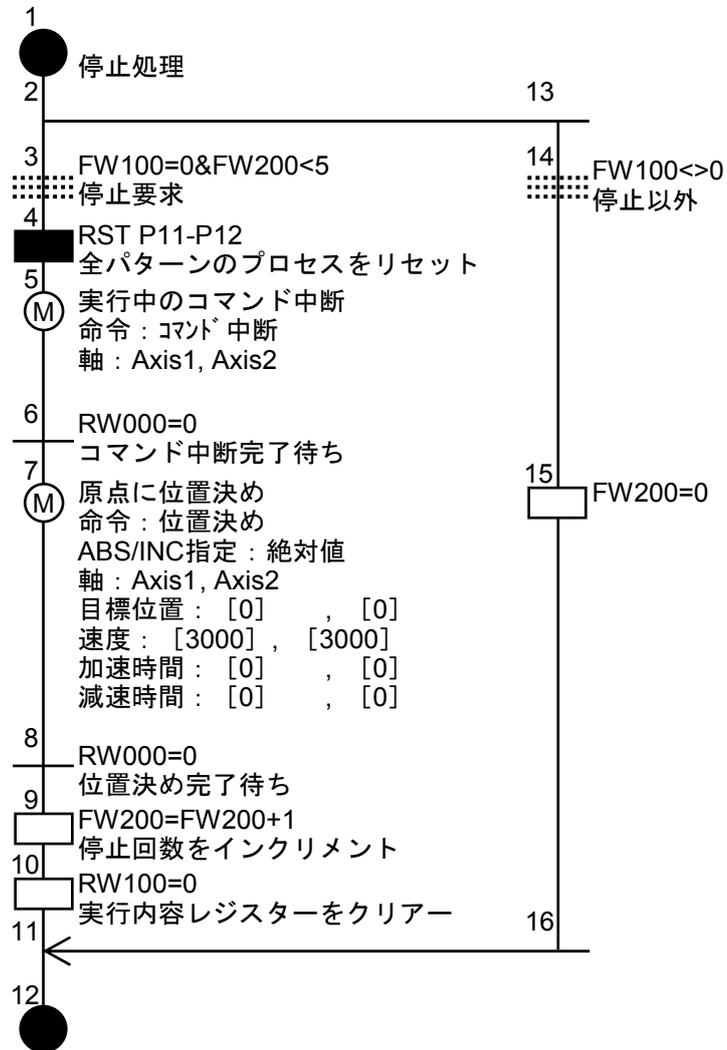
<次ページに続く>

<前ページの続き>



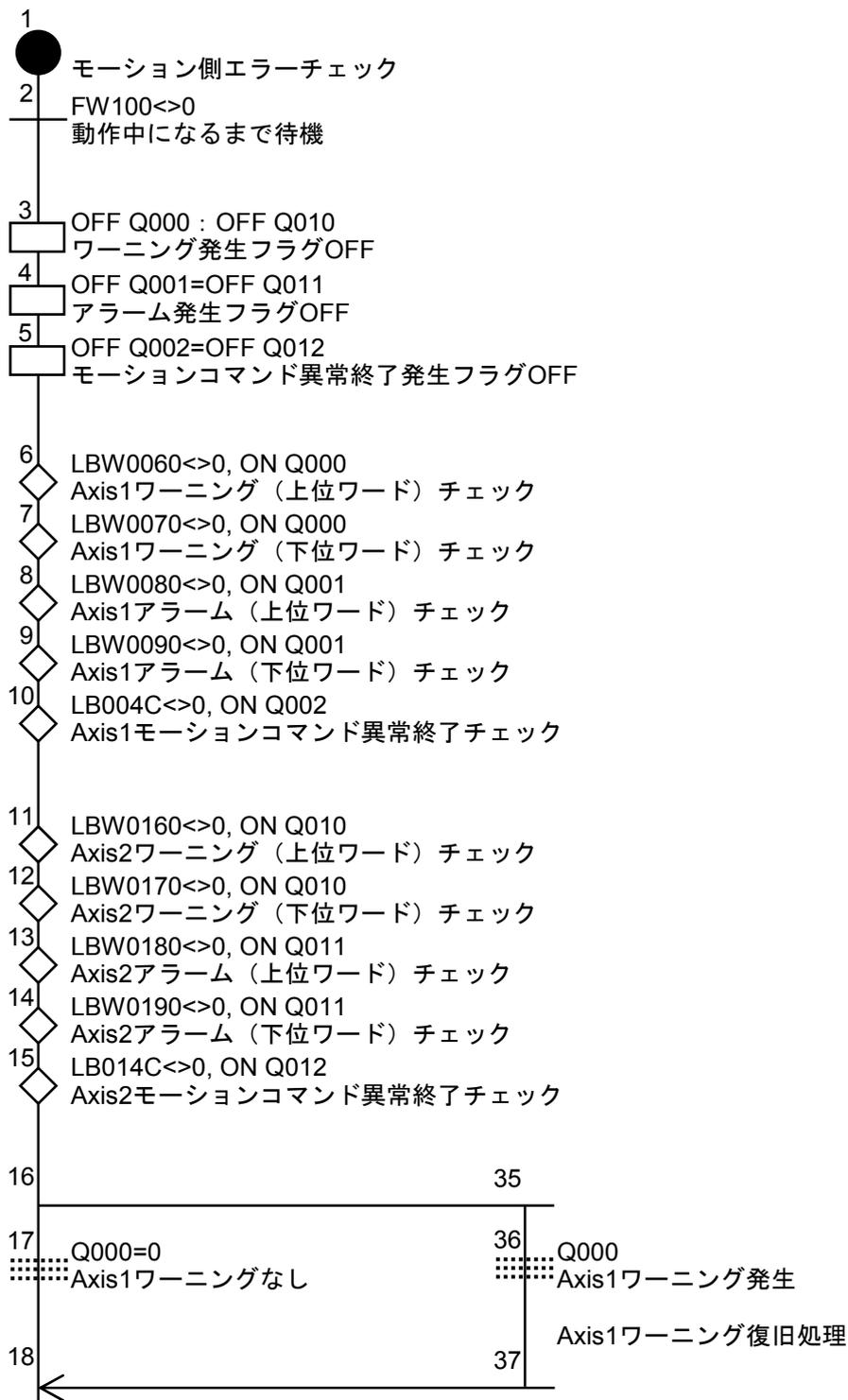


(3) プロセス2：停止処理



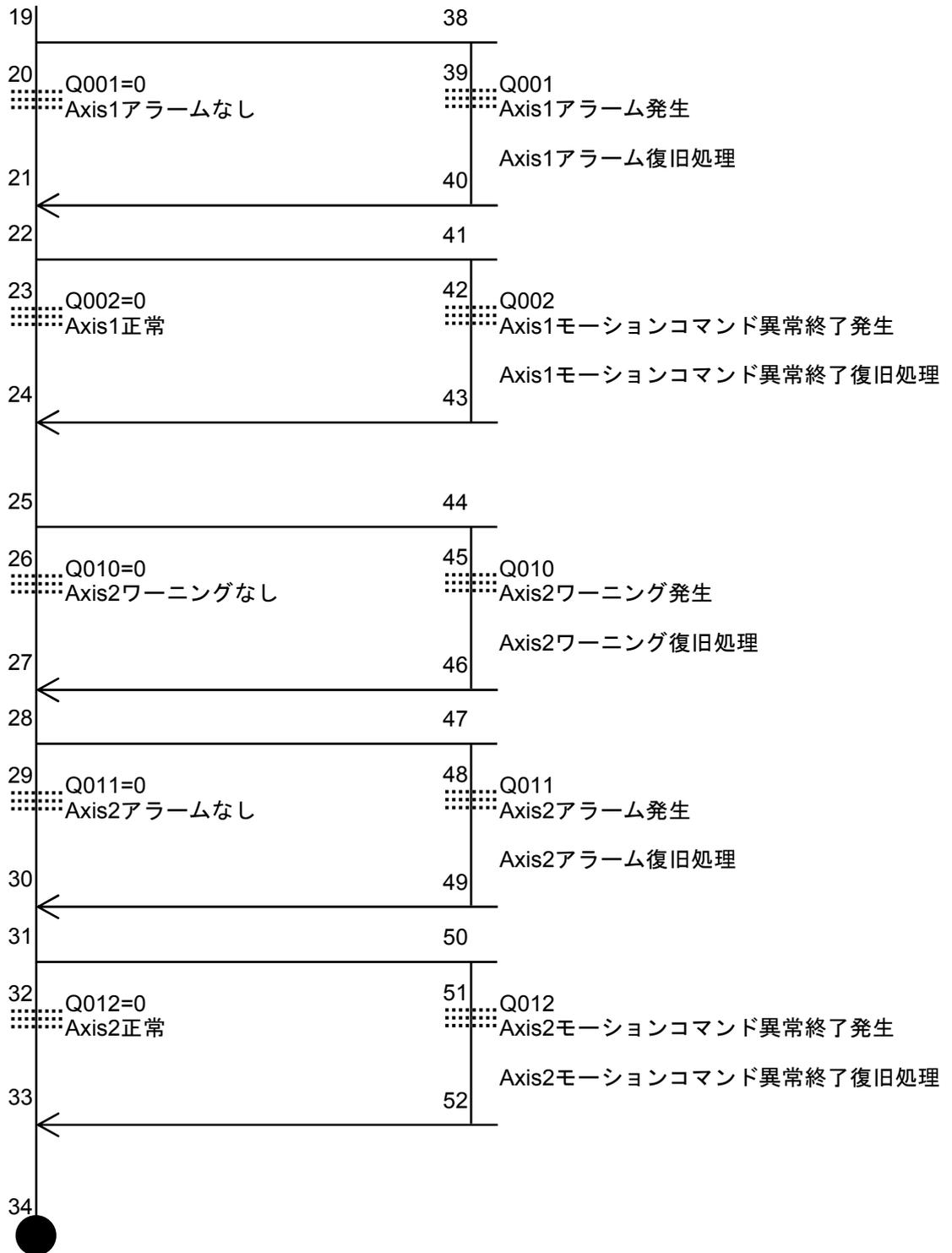
## 6 モーション制御命令

### (4) プロセス3：モーション側エラーチェック処理



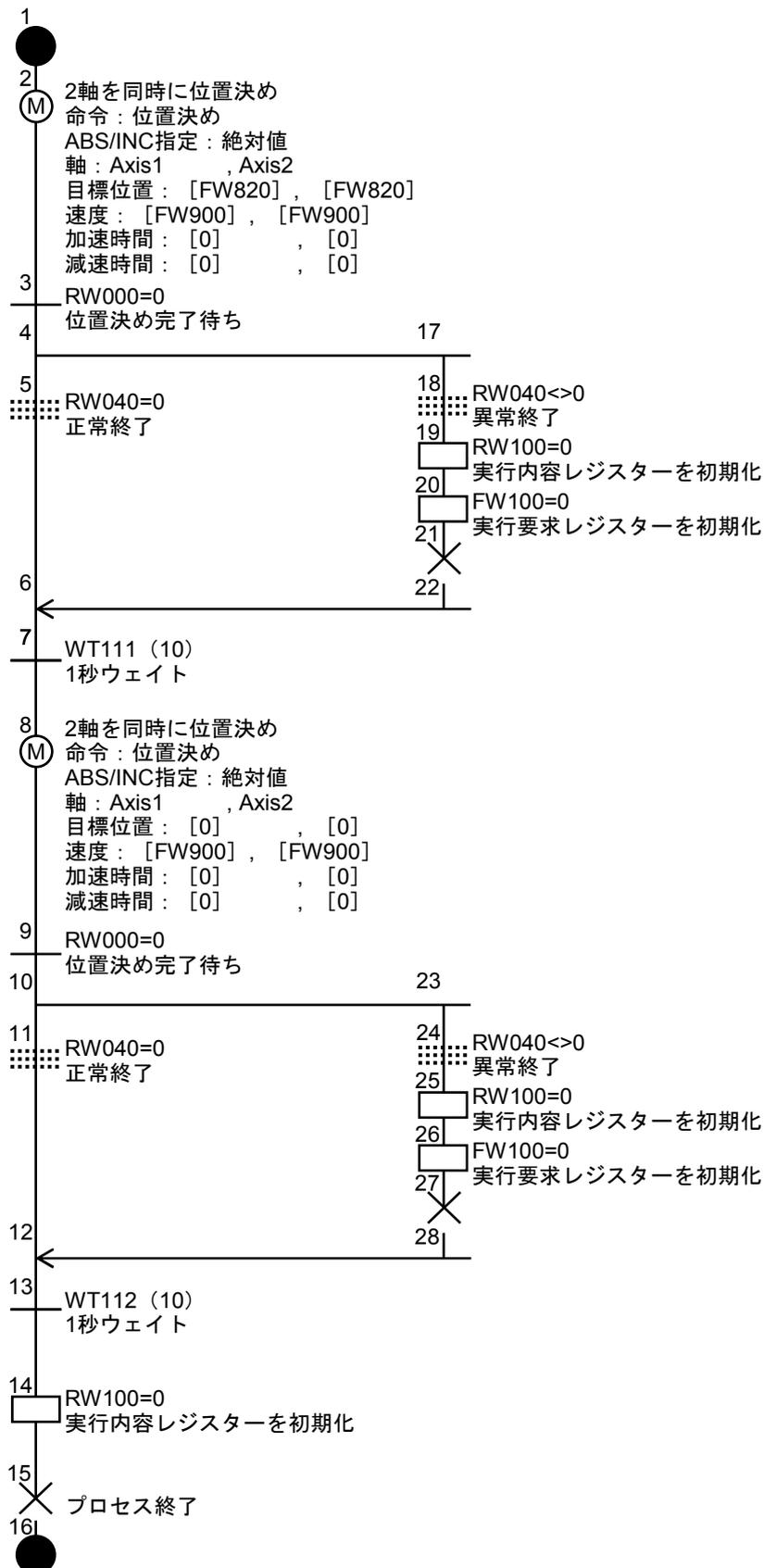
<次ページに続く>

<前ページの続き>

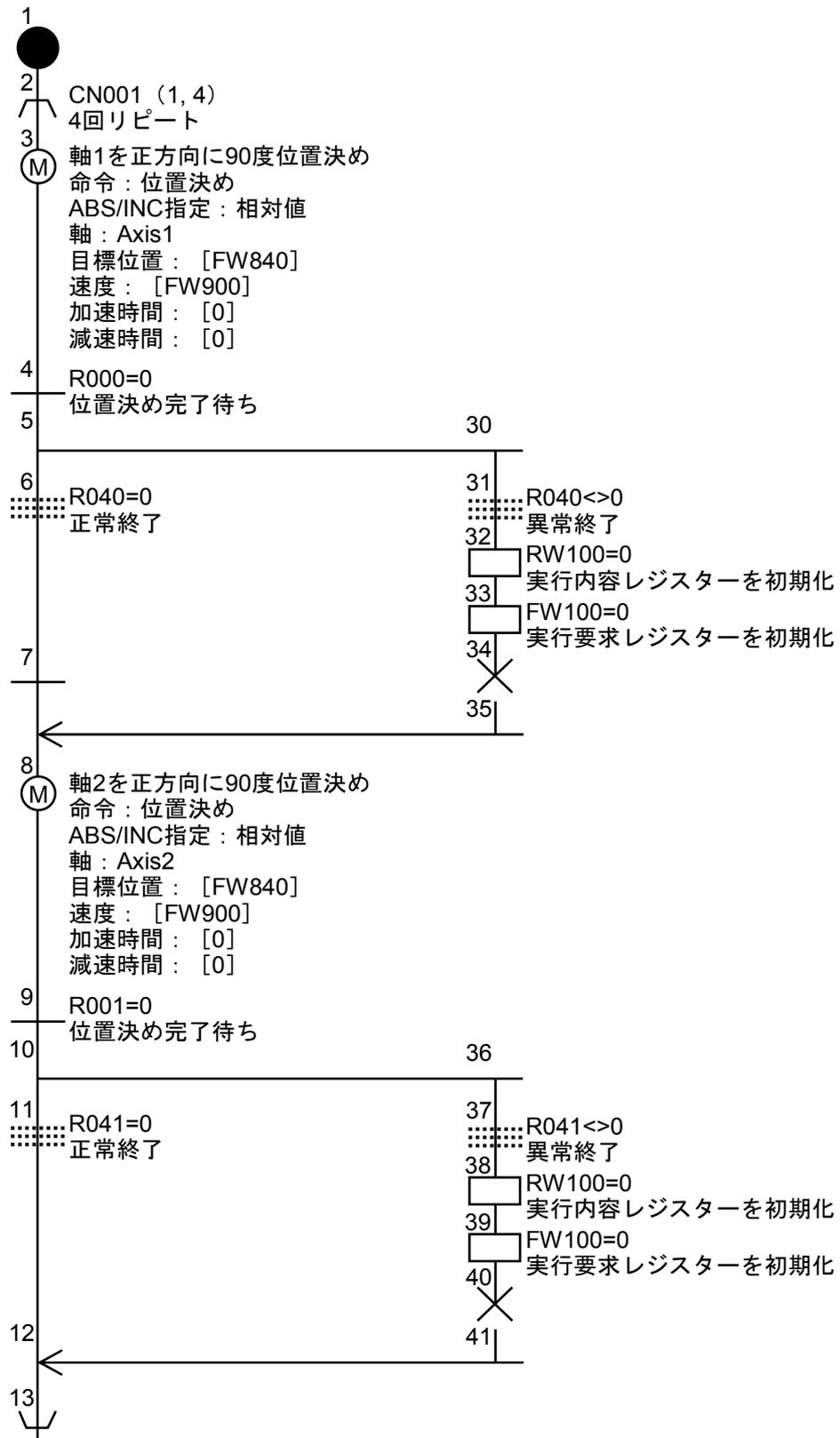


## 6 モーション制御命令

### (5) プロセス11：2軸同時制御



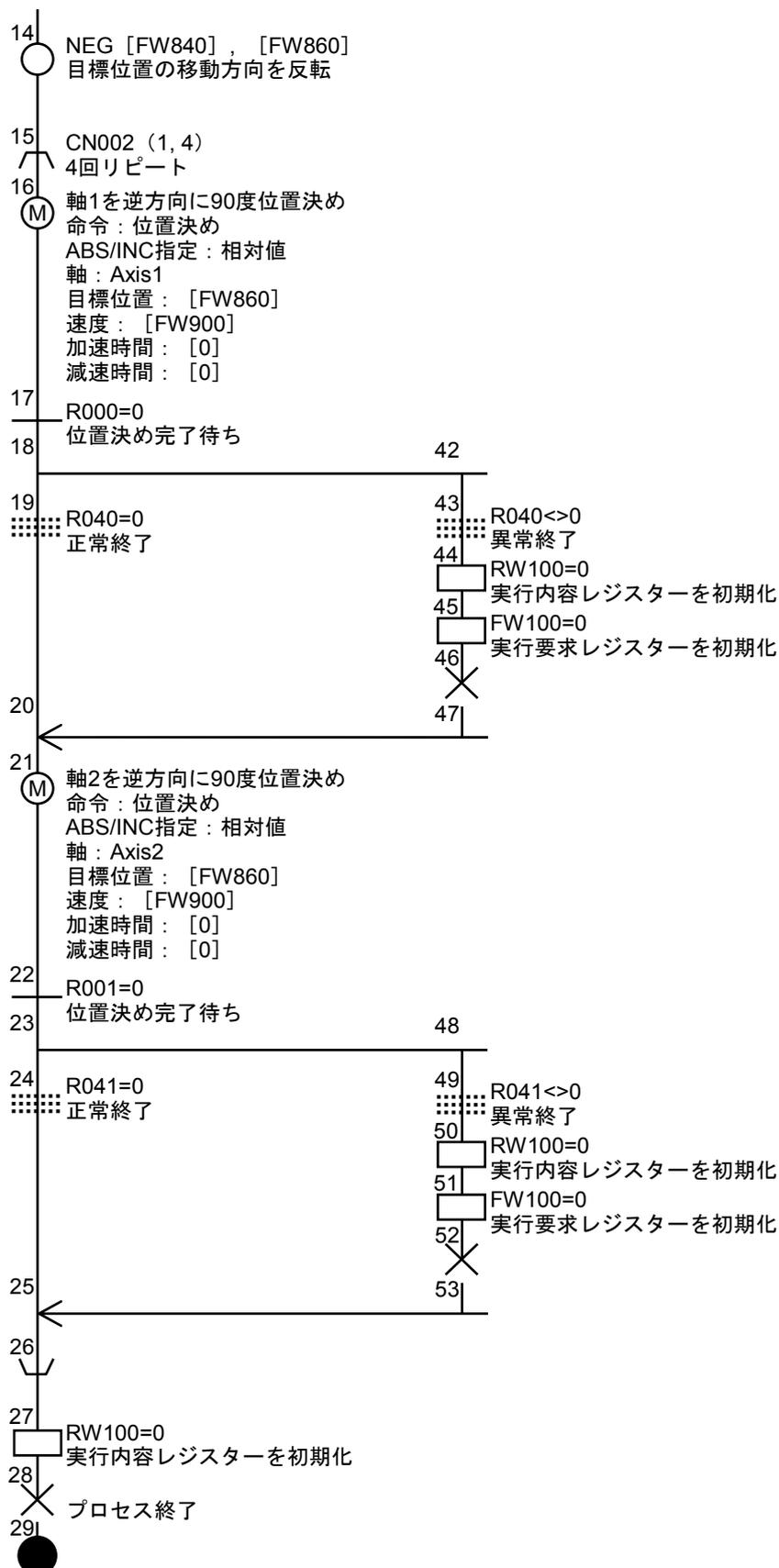
## (6) プロセス12：軸を90度回転



&lt;次ページに続く&gt;

## 6 モーション制御命令

<前ページの続き>

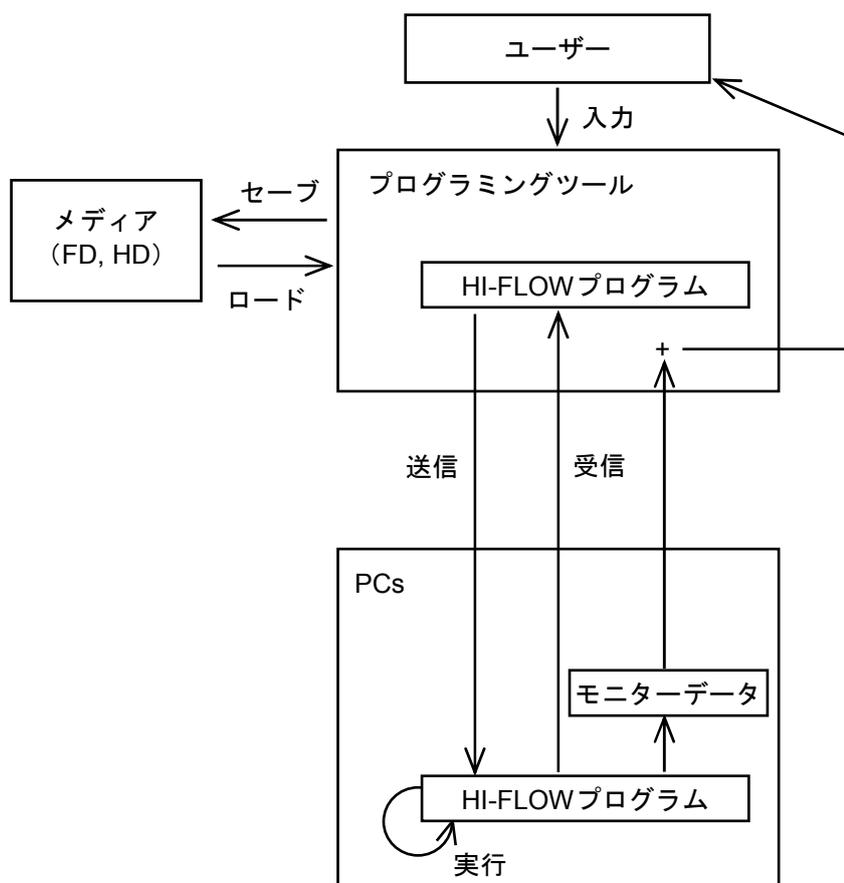


補 足

## 補足 1 HI-FLOWプログラムの流れ

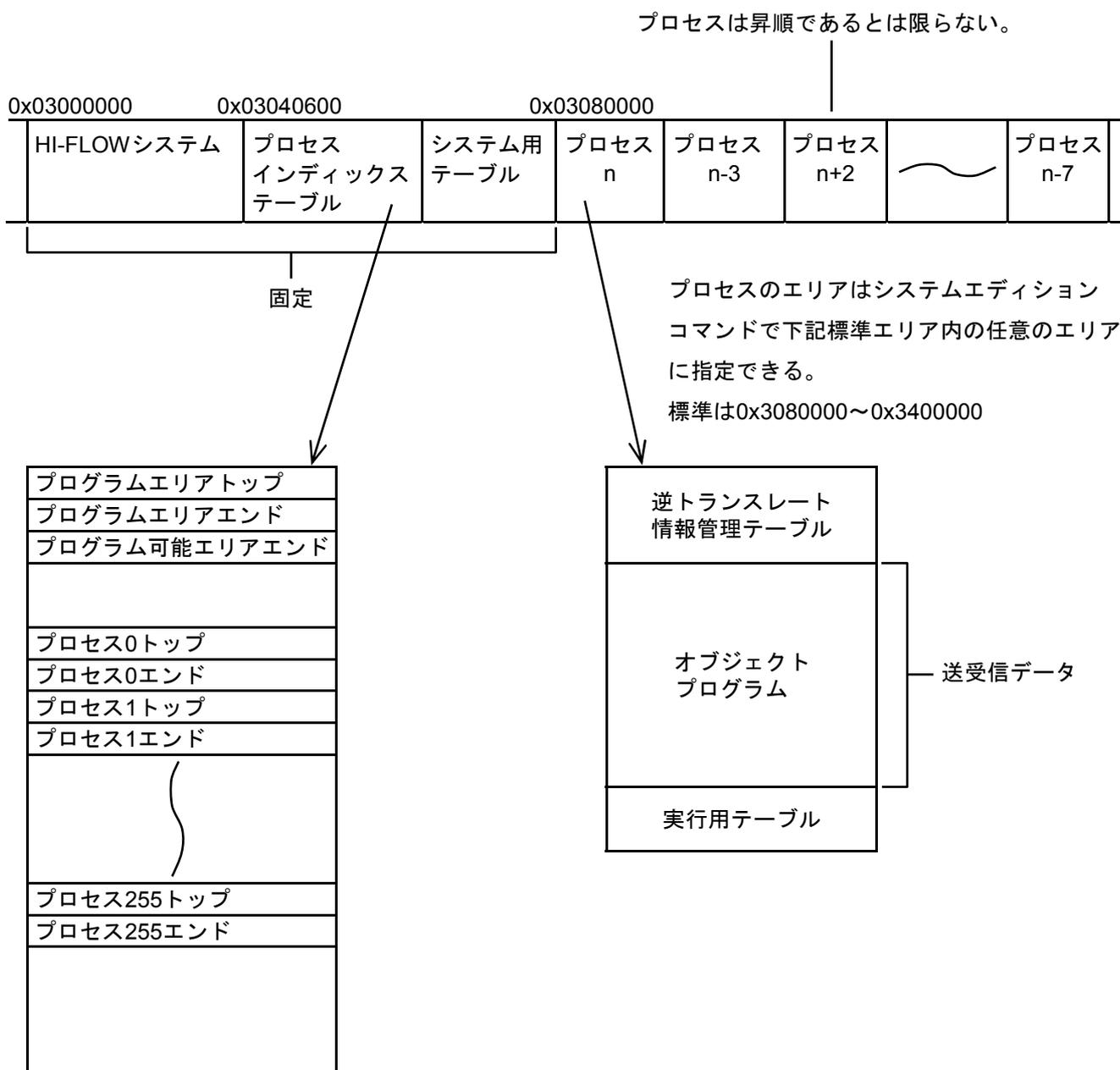
HI-FLOWプログラムは、プログラミングツールで作成されPCsで実行します。実行結果などをモニターするときは必要最小限のデータをPCsから受け取り、ツールにあるプログラムと合成して出力します。これは通信量を少なくしてモニター速度を上げるためです。

また、作成したプログラムをセーブ/ロードするため各メディア（FD, HD）とのやり取りもあります。



補足2 PCsメモリー

PCs上で実行されるHI-FLOWプログラムは、CMUモジュール内の以下に示すエリアに存在します。PCs上では実際にメモリーに配置されますのでメモリーマップイメージを示します。



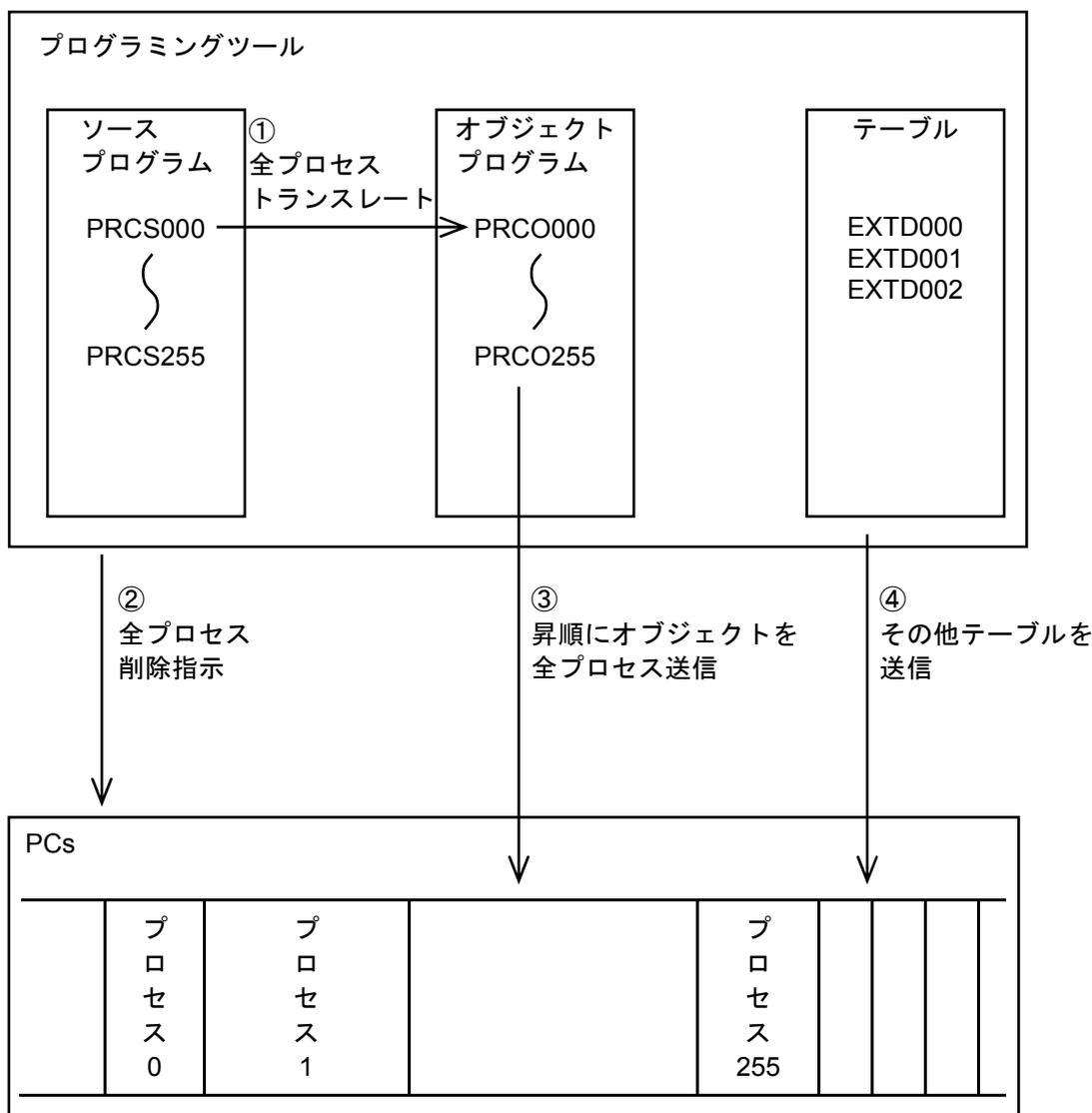
### 補足3 オンラインモード

オフラインとはPCsのメモリー内容とは関係なく、編集対象をプログラミングツールのプログラムにすることです。

オンラインとは編集／メモリー対象をPCsのプログラムにすることです。ところが対象をPCsにしたとき、モニター時に必要なデータをすべてPCsから読み込んでいないため（通信時間がかかる）ツール側のプログラムとPCsのプログラムが一致している必要があります。一致させる手段として送信／受信があります。または、HI-FLOWのプログラムはプロセスで完結しているため1プロセスだけ一致していればそのプロセスは編集／モニターができます。時間的な節約の意味も含めて全プロセス／1プロセスの送受信があります。

#### (1) 全プロセス送信

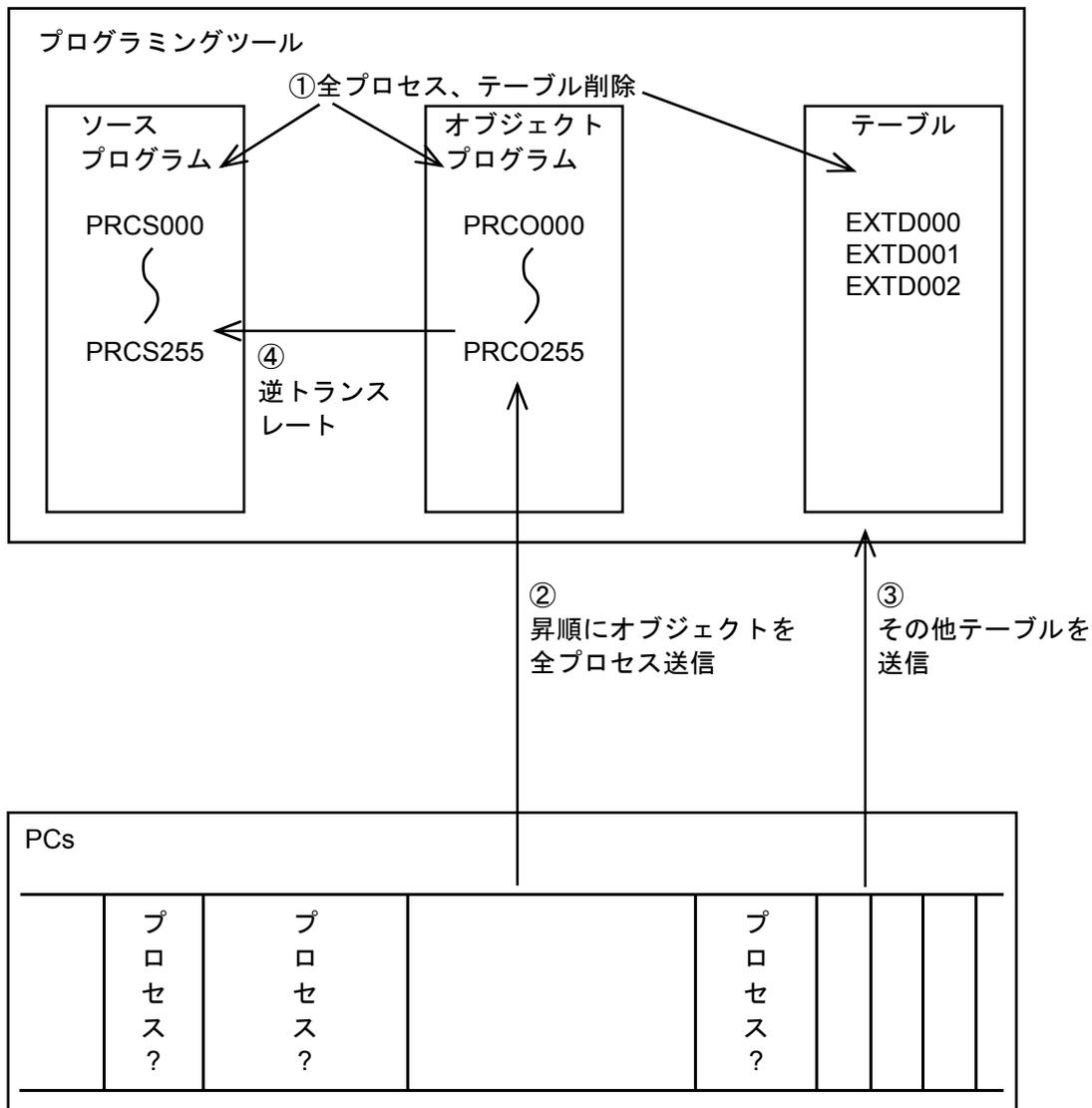
ツール上に存在する全HI-FLOWプログラムをPCsへ送信する際のデータの流れを示します。



全プロセス送信後、メモリー上のプロセス、テーブルは昇順となる

(2) 全プロセス受信

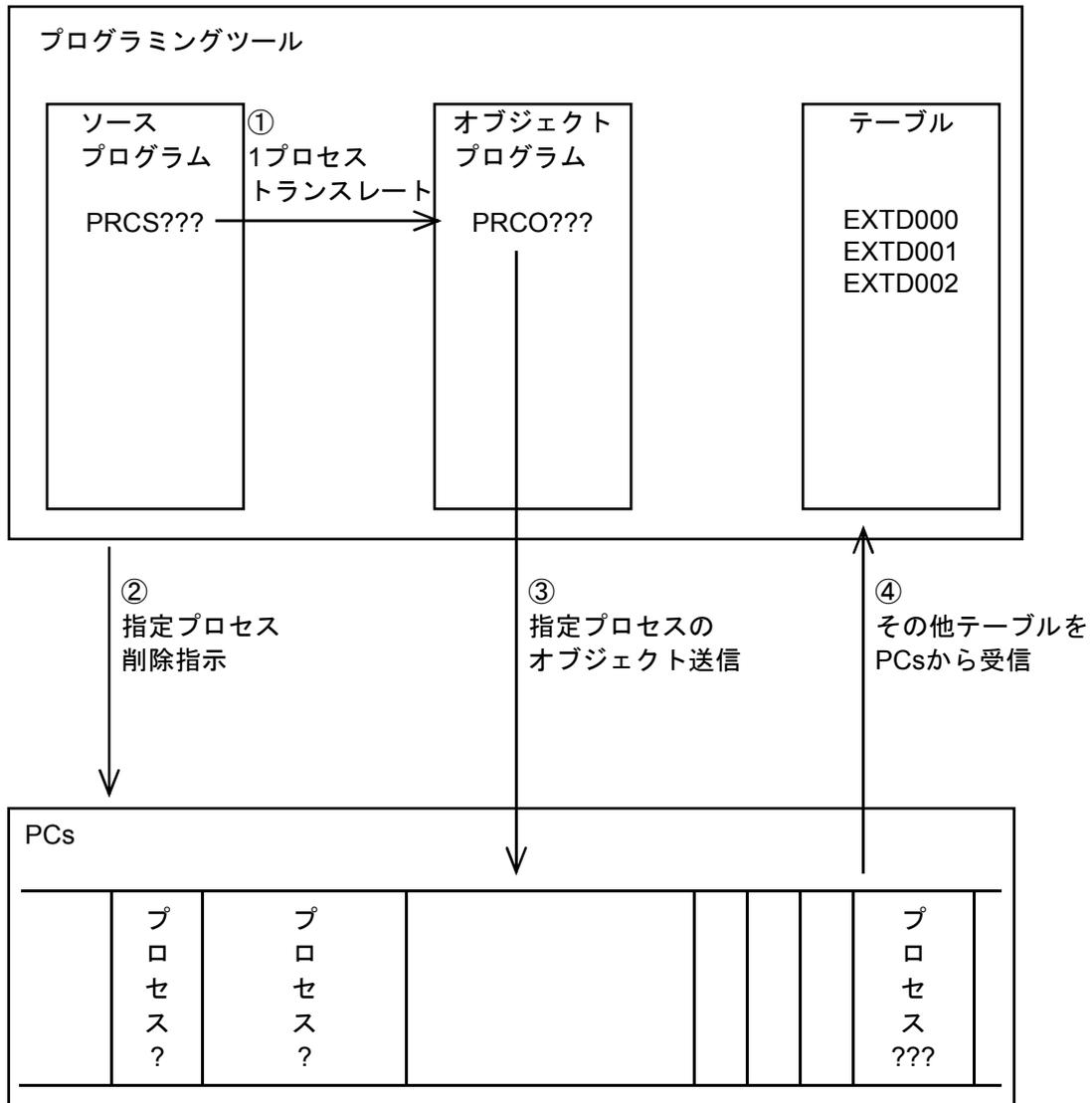
PCs上に存在する全HI-FLOWプログラムをツールへ受信する際のデータの流れを示します。



受信時にメモリー上のプロセス、テーブルが昇順である保証はない

(3) 1プロセス送信

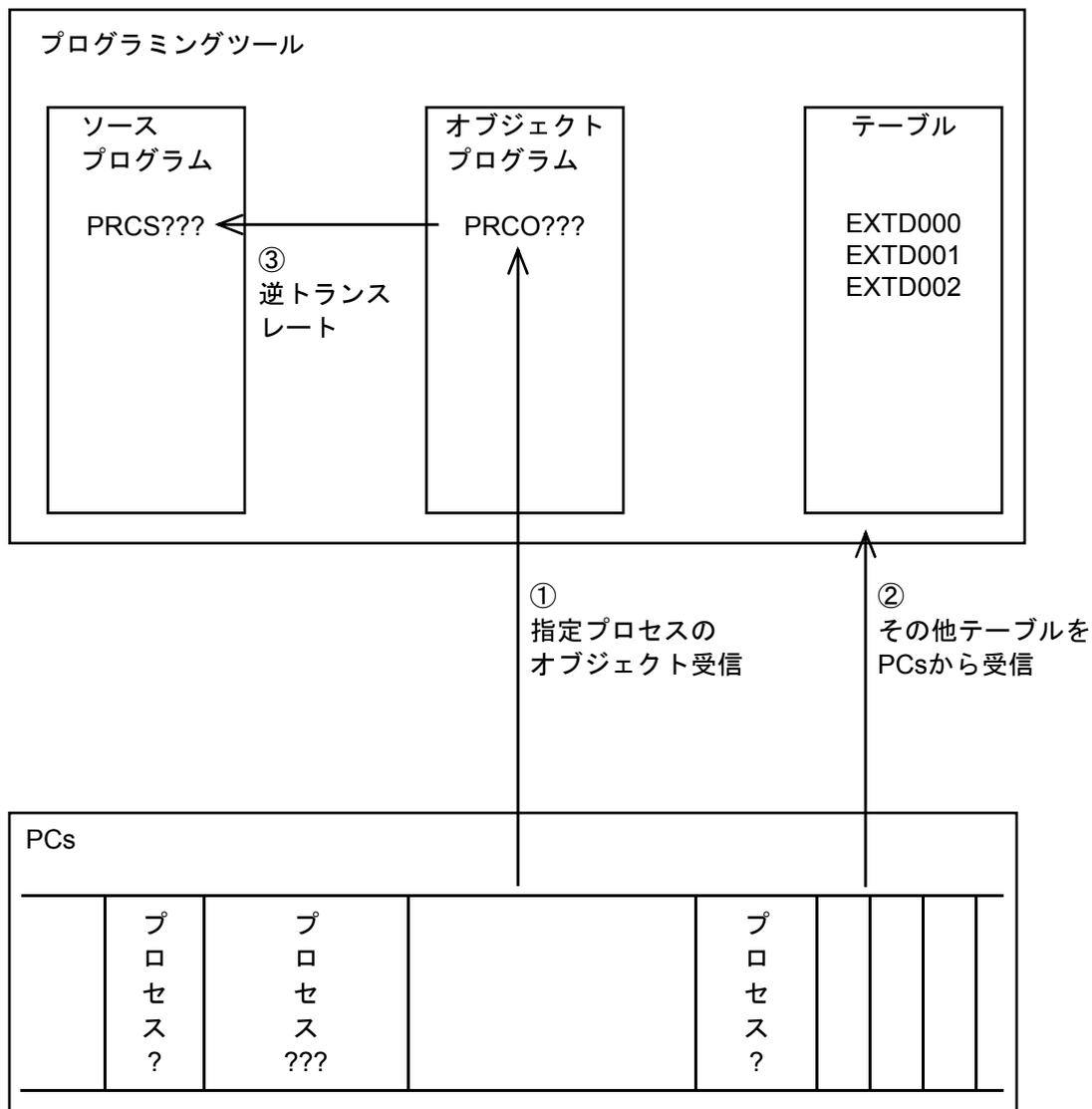
ツール上に存在する特定のHI-FLOWプログラムを1つだけPCsへ送信する際のデータの流れを示します。



送信後、指定プロセスはメモリー上最後になる

(4) 1プロセス受信

PCs上に存在する特定のHI-FLOWプロセスを1つだけツールに受信する際のデータの流れを示します。



受信時にメモリー上のプロセス、テーブルが昇順である保証はない

## 補足4 進行チェック

HI-FLOWは、ユーザープログラム上の進行位置をモニターカーソルが示します。PCs上のHI-FLOWシステムは、現在の進行位置の進行管理を行っています。

この章では、PCsに転送されたユーザープログラムがPCs上でどのように進行チェックされるかを示します。

項目	内容
基本則	<p>PCsのスキャン時間ごとに進行チェックする。</p> <p>ACT起動をかけられたプロセスはプロセス番号の小さいものから大きいものへと進行チェックする。</p> <p>同一プロセス内のルート番号の小さいものから大きいものへと進行チェックする（ルート番号は画面上、左上&lt;左下&lt;右上&lt;右下と大きくなる）。</p> <p>同一ルート内のステップ番号の小さいものから大きいものへと進行する。</p> <p>ステップ実行が終了すると次ステップへ進行するが、進行できないときは、次番号のルートの進行チェックをする。次スキャン時、このプロセス、ルートの進行チェックはこのステップから行う。</p>
コールプロセス	<p>コールされたプロセスはコール元プロセス、コール元ルートの次に進行チェックされる。コールされたプロセスの進行チェック終了時はプロセス実行が終了していないとコール元プロセス、コール元ルートの次のルートを進捗チェックする。終了した場合は、コール元ルートの次ステップを進行チェックする。</p>
プロセス制御	<p>ACTによって起動をかけられたプロセスはその時点で進行チェックしているACT起動されたプロセスよりもプロセス番号が小さければ次スキャン時、大きければ同一スキャンで進行チェックされる。</p> <p>RST, STP, CLRはコントロールボックス、プロセススタートとも、その時点で処理される。</p>
常時監視	<p>プロセススタートの条件（ACT, STP, RST, CLR）、マルチエントリー条件はそのプロセス進行チェックの前にチェックし、条件成立時はプロセス進行チェック前に処理を行う。</p> <p>インターロック付きY出力条件はスキャン時間ごと、最初のプロセス進行チェック前にチェックし、Y出力をON/OFFする。</p>
分岐	<p>分岐ステップ（イフ、ジャンプ）実行後は飛び先のステップを進行チェックする。したがって、実行ルートが1スキャン進行チェックしなかったり、2度進行チェックしたりする場合が考えられる。また、進行条件なしの閉ループは無限実行となる場合が考えられるので注意が必要。</p>
繰り返し	<p>リピートエンドは次にリピートスタートを進行チェックする。したがって、進行条件なしの繰り返しは無限実行となる場合が考えられるので注意が必要。</p>

項目	内容
強制終了	エスケープを実行すると、次プロセスの進行チェックを行う。コールプロセスの場合、コール元プロセス、コール元ルートの次ステップの進行チェックをする。
同期	パラスタート実行後は次ステップの進行チェックをする。 パラエンド、ルートエンドは同期ルートがすべて終了していたら、合流ルートのパラエンドの次のステップの進行チェックをする。すべて終了していなければ、自ルートを停止し次ルートの進行チェックをする。
選択	セレクト実行後は次ステップの進行チェックをする。 セルウエイト条件成立時は、他の選択ルートを停止し、次ステップの進行チェックをする。不成立時は次ルートの進行チェックをする。 セレクトエンド、ルートエンドは、合流ルートのセレクトエンドの次のステップの進行チェックをする。その際、合流ルートが停止していたら起動をかける。
条件待ち	ウエイトは条件が成立していれば次ステップへ進み進行チェックをする。条件が不成立ならば、次番号のルートの進行チェックをする。次スキャン時、このプロセス、ルートの進行チェックはこのステップから行う。 前状態クリアー付きの場合は、次ステップへ進む前に前ステップがON文であれば、それを0クリアーする。
遅延なし 図形	どんな場合も遅延なく次ステップへ進む図形名称を示す。 プロセススタート、ルートスタート、パラスタート、セレクト、マルチエンタリー、ボックス、コントロールボックス、ファンクション、コールプロセス終了時のプロセスエンド、同期終了時のルートエンド、パラエンド、選択合流時のルートエンド、セレクトエンド、および分岐関係（リピートスタート、リピートエンド、イフ、ジャンプ）はスキャン遅れなく進むべきステップへ進む。
非同期合流	パラスタート実行後は次ステップの進行チェックをする。 非同期プロセスエンド到達時に非同期ルート（非同期分岐ルート以外）が終了していれば、次スキャンでプロセススタートを実行し次ステップの進行チェックをする。 再度、パラスタートに到達した際に非同期ルートが実行を継続していれば、ルートスタートを実行しないで次ルートの進行チェックをする。

## 補足5 HI-FLOWプログラムとCPU負荷

HI-FLOWプログラムはPCs上でCMUモジュールのOSの一部として動作します。したがって、HI-FLOWプログラムが増えるとPCsのCMUモジュールのOS負荷が上がります。

負荷が上がり過ぎるとシーケンスサイクルが止まるなど、システム全体として正しく動作しなくなります。そこで、HI-FLOWプログラムの上手な作り方と負荷判断の目安を示します。

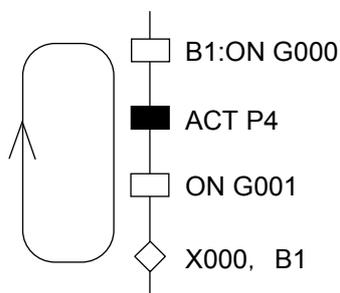
### <HI-FLOWプログラムの上手な作り方>

1. HI-FLOWプログラムの負荷大小は実行しているルート数によります。

HI-FLOWプログラムの縦（ルート）の長さには関係ありません。したがって、あまりたくさんのプロセス、ルートに分割しすべてを実行させるようなプログラムは負荷が重くなります。

2. 余分なループに注意する。

必要がなく、かつ停止部分がないループに注意してください。



左記のプログラムはX000がOFFするまでループ内のステップを実行し続け、負荷が増大します。

3. タイマー番号は小さい順に詰めて使う。

タイマー（ウェイトタイマー、パラレルタイマー、カウンタ）はタイマー番号が小さいほど負荷が軽くなっています。

4. ウェイトタイマーは同一ルートでは同一番号を使う。

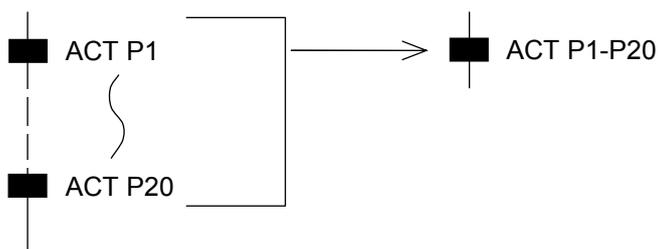
同一ルートのウェイトタイマーは、絶対に同時には実行されません。したがって、同一タイマー番号を付けて、なるべく後ろのタイマー番号を使用しないようにします。

5. コールプロセスは必要最小限にする。

プログラムをサブルーチン化するとわかりやすいプログラムになりますが、実行時はコールプロセス化されていない場合よりも負荷が重くなります。プログラムの構造化はよく検討して行ってください。

6. コントロールボックスの連続はなるべく避けるようにする。

コントロールボックスの実行負荷はかなり重いので連続使用はなるべく避けてください。やむをえず使用する場合は、プロセスの連続指定を有効に利用してください。



7. システムコントロールビットの設定は必要最小限にする。

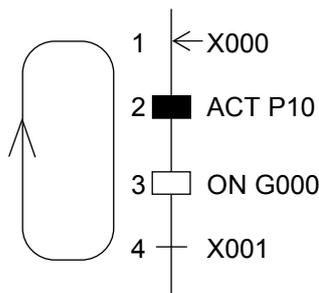
システムコントロールビットはシーケンスサイクルごと、ステップ実行ごとにチェックが必要となり負荷がかなりかかります。必要最小限の設定をしてください。

8. マルチエントリーの使用は必要最小限にする。

マルチエントリーステップはシーケンスごとにチェックが必要となり、使用ステップが増えると負荷が重くなります。必要最小限の使用としてください。

9. マルチエントリーのインループに注意する。

マルチエントリーはシーケンスサイクルごとに条件式をチェックし、成立していればそのステップから実行します。ところが成立条件がエッジ成立でなく連続成立しているとインループしてしまいます。マルチエントリー条件はエッジトリガにしてください。



左記プログラムでX001が不成立でX000がONし続けるとシーケンスごとに1から4まで実行し続けます。防ぐためにはX000をエッジ条件にします。

10. プロセススタートのSTP, RSTの使用は必要最小限にする。

プロセススタートのSTP, RSTはシーケンスサイクルごとに条件をチェックするため、負荷がかなりあります。設定は必要最小限にします。

11. プロセススタートのCLR設定に注意する。

プロセススタートのCLRは条件が成立するたびにPI/Oのクリアを行うため負荷が重くなります（RST, STP, ACTは一度条件が成立すると条件チェックしません）。CLRのチェック条件は注意して作成してください。

12. 応用命令の連続使用はなるべく避けるようにする。

応用命令は停止なしに演算を行うため、連続して記述するとそのシーケンスサイクルが延びる可能性があります。十分注意して作成してください。

13. 複雑な条件式はなるべく避けるようにする。

HI-FLOWで条件式を複雑にすると式解析にラダーからかなりの時間がかかります。あまり複雑な条件はラダーで一度受けてからHI-FLOWに渡したほうが負荷が軽くなります。

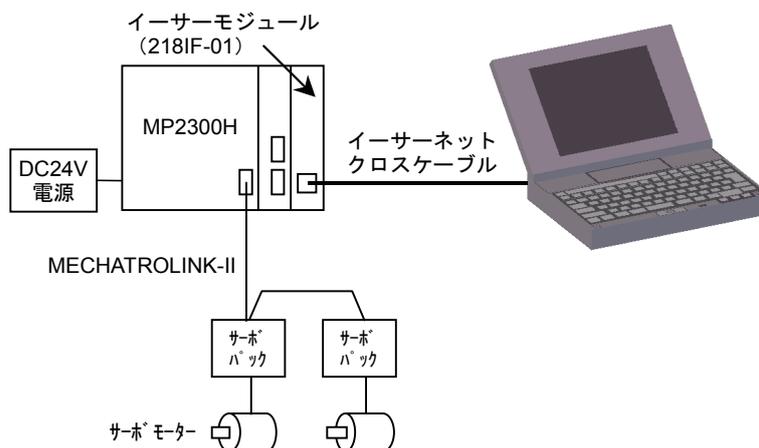
## 補足 6 MP2300Hシステム構成変更手順

モーションコントローラーMP2300Hにオプションモジュールやサーボパックを追加・交換する場合、セルフコンフィグレーション機能を使用してMP2300Hのシステム構成情報を更新する必要があります。

セルフコンフィグレーションは、モジュールを追加・交換したときにシステム構成情報を更新するモジュールセルフコンフィグレーションと、MP2300Hを新規構築時にシステム構成情報を作成する全モジュールセルフコンフィグレーションがあります。以下にそれぞれのセルフコンフィグレーションの手順を示します。

### 1. モジュールセルフコンフィグレーション手順

- (1) MP2300Hの電源がOFFであることを確認し、モーションモジュール、サーボパック、およびサーボモーターを接続します。
- (2) パソコンとMP2300Hをイーサネットクロスケーブルで以下の図に従い配線します。



- (3) イーサナーモジュール（型式：218IF-01）のスイッチを以下の内容に設定し、MP2300Hの電源をONにします。

#### ■ イーサナーモジュール

スイッチ名称	設定	動作モード
INIT	OFF→ON	デフォルトIPアドレス（192.168.1.1）で起動します。
TEST	OFF	

- (4) Windows®のコントロールパネルの [ネットワークとダイヤルアップ接続] から、パソコンのIPアドレスとサブネットマスクを以下の内容に設定します。

項目	設定	備考
IPアドレス	192.168.1.xx	xxは2～254の範囲で設定してください。
サブネットマスク	255.255.255.0	

- (5) Windows®の“Ping機能” から、パソコンとMP2300H間の接続が有効になっているか確認します。以下に方法を示します。

■ Windows®の“Ping機能” による接続確認方法

Windows®のスタートメニューからコマンドプロンプトを立ち上げます。このコマンドプロンプトから“Ping” コマンドを入力し、パソコンとイーサーモジュール間の基本的な通信テストを実行します。Pingコマンドは以下の内容で入力します。

```
C:¥WINDOWS> ping 192.168.1.1
```

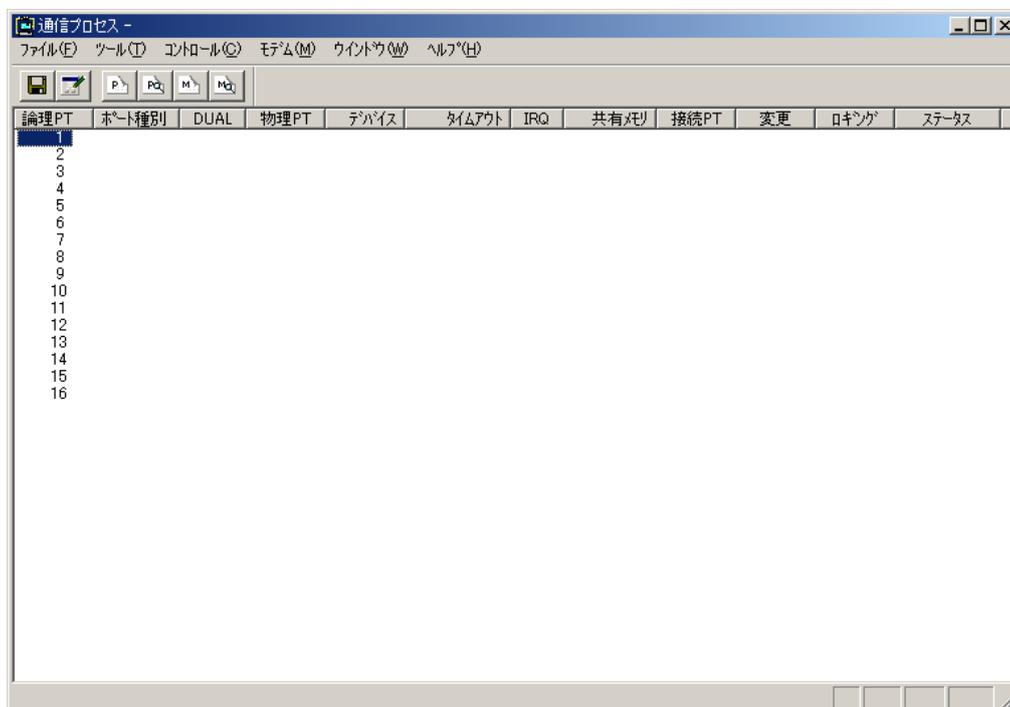
対象のイーサーモジュールが正しく接続されている場合は、以下のメッセージが表示されます。

```
Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=4ms TTL=254
Reply from 192.168.1.1: bytes=32 time=3ms TTL=254
Reply from 192.168.1.1: bytes=32 time=3ms TTL=254
Reply from 192.168.1.1: bytes=32 time=3ms TTL=254
C:¥WINDOWS>
```

対象のイーサーモジュールが正しく接続されていない場合は、以下のメッセージ (タイムアウト) が表示されます。

```
Pinging 192.168.1.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
C:¥WINDOWS>
```

- (6) Windows®のスタートメニューから [YE\_Applications] - [Communication Manager] をクリックし、通信プロセスを立ち上げます。



- (7) リストから「論理PT」の「1」をダブルクリックして「論理ポート設定」画面を表示します。



- (8) 「ポート種別」に「CP-218」を指定し、**詳細** ボタンをクリックして [CP-218ポート設定] 画面を表示します。

CP-218ポート設定

IPアドレス(第1ポート) 192.168.1.201

IPアドレス(第2ポート)

デフォルト指定  する  しない

エンジニアリングポート 10000 (256 - 65535)

メッセージ受信設定

TCPポート数 10 (0 - 16)

TCP先頭ポート 20000 (20000 - 65535)

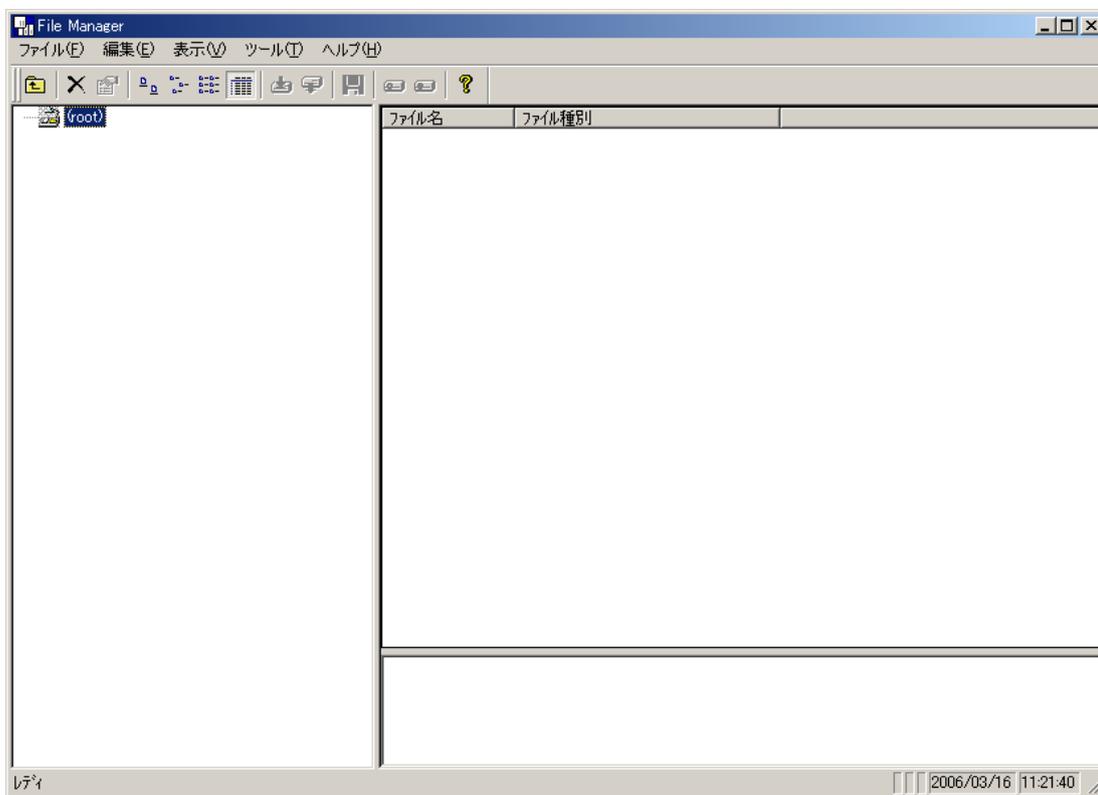
UDPポート数 10 (0 - 16)

UDP先頭ポート 20000 (20000 - 65535)

OK キャンセル

- (9) 「IPアドレス (第1ポート)」にパソコンのIPアドレスを指定し、**OK** ボタンをクリックします。
- (10) [論理ポート設定] 画面に戻るので **OK** ボタンをクリックします。
- (11) 通信プロセスのメニューから [ファイル] - [アプリケーションの終了] を選択して通信プロセスを終了します。

- (12) Windows®のスタートメニューから [YE\_Applications] - [MPE720] をクリックし、File Managerを立ち上げます。



- (13) File Managerのツリー表示部から (root) を右クリックして表示されるメニューから、[新規作成] - [グループフォルダ] をクリックし、[新規作成] 画面を表示します。



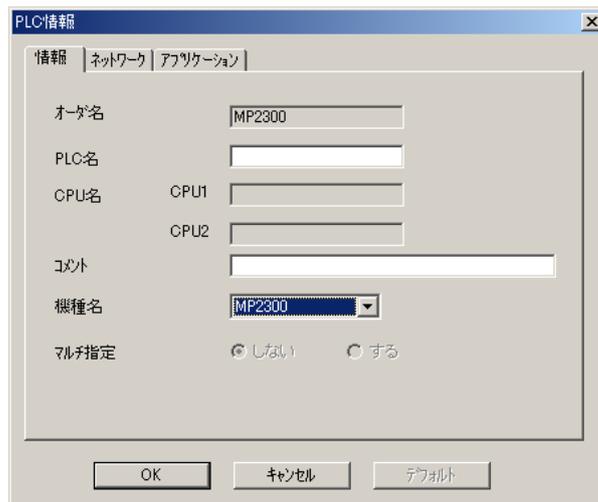
[新規作成] 画面から、新規作成するグループ名を指定し、**OK** ボタンをクリックします (例 : PLC)。

- (14) File Managerのツリー表示部に指定したグループが作成されますので、作成されたグループ名を右クリックして表示されるメニューから、[新規作成] - [オーダフォルダ] をクリックし、[新規作成] 画面を表示します。



[新規作成] 画面から、新規作成するオーダ名を指定し、**OK** ボタンをクリックします  
(例：MP2300H)。

- (15) File Managerのツリー表示部に指定したオーダフォルダが作成されますので、作成されたオーダ名を右クリックして表示されるメニューから、[新規作成] - [PLCフォルダ] をクリックし、[PLC情報] 画面を表示します。



[PLC情報] 画面から、以下の項目を指定し、**OK** ボタンをクリックします。

■ 情報

PLC名：任意（例：sample）

機種名：MP2300

■ ネットワーク

オンライン指定：する

論理ポート番号：1：CP-218

IPアドレス：192.168.1.1（MP2300HのIPアドレス）

- (16) File Managerのツリー表示部に指定したPLCフォルダが作成されますので、作成されたPLCフォルダを右クリックして表示されるメニューから、「オンライン」にチェックマークが入っていることを確認し、ログオンをクリックし、[CPUログオン] 画面を表示します。



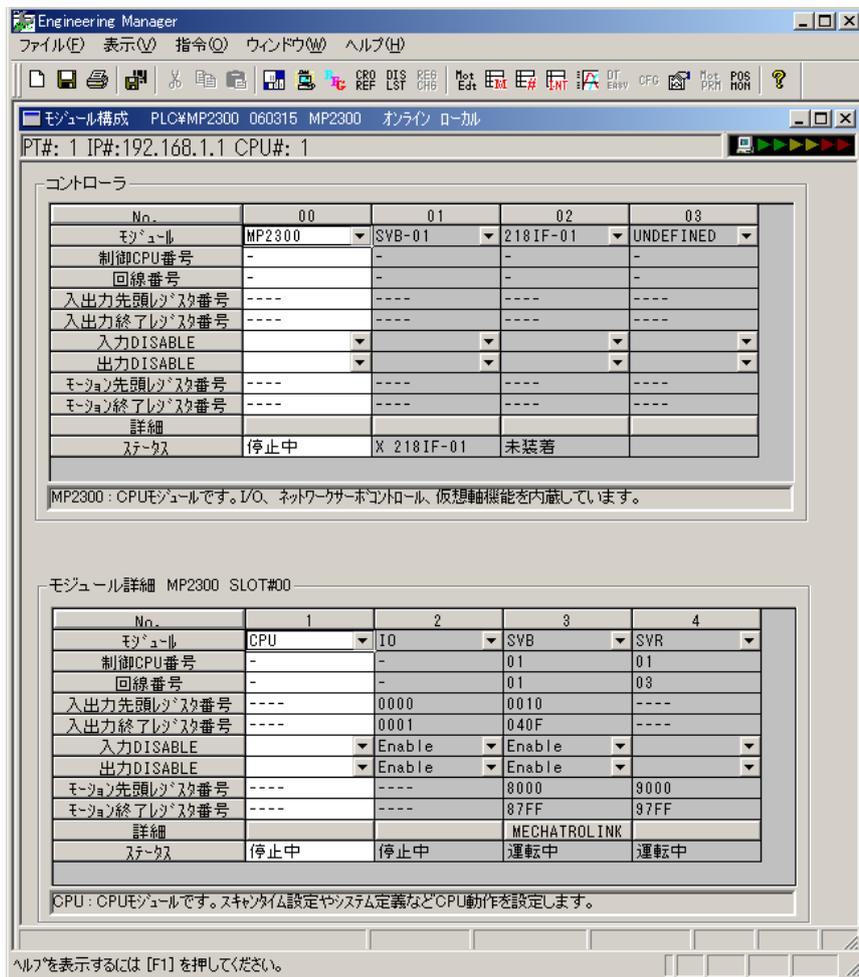
- (17) [CPUログオン] 画面から、ユーザ名およびパスワードを指定し、**OK** ボタンをクリックします。

- MP2300Hは、ユーザ名およびパスワードにデフォルトで「USER-A」が設定されています。
- ログイン時に以下のメッセージが表示される場合があります。



この警告メッセージは、後の作業で修正されるため問題ありません。

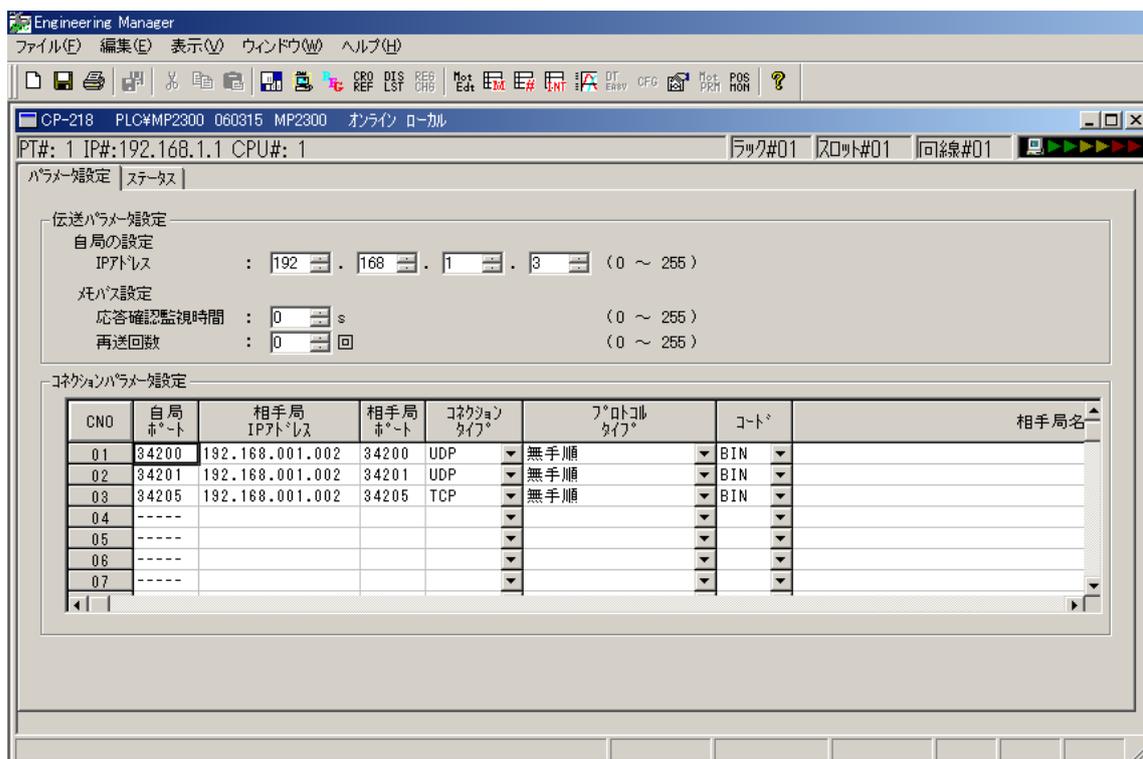
- (18) File Managerのツリー表示部から [定義フォルダ] - [モジュール構成] をダブルクリックし、Engineering Managerの [モジュール構成] 画面を表示します。



- (19) [モジュール構成] 画面の「コントローラ」グループで、システム構成を変更したモジュールのNo.を右クリックして表示されるメニューから、「モジュールセルフコンフィグレーション」をクリックし、セルフコンフィグレーションを実行します。

- イーサーモジュールのIPアドレスを設定または変更したい場合は、(20) ~ (23) で設定してください。
- イーサーモジュールをセルフコンフィグレーションした場合、イーサーモジュールの通信設定が初期化されますので、(20) ~ (23) で再設定してください。
- 上記以外の場合は、(24) に進んでください。

- (20) [モジュール構成] 画面の「コントローラ」グループから、モジュールが「218IF-01」となっているNo.をクリックします。
- (21) 「モジュール詳細」グループの表示が切り替わりますので、モジュールが「218IF」となっているNo.をダブルクリックし、[CP-218] 画面を表示します。

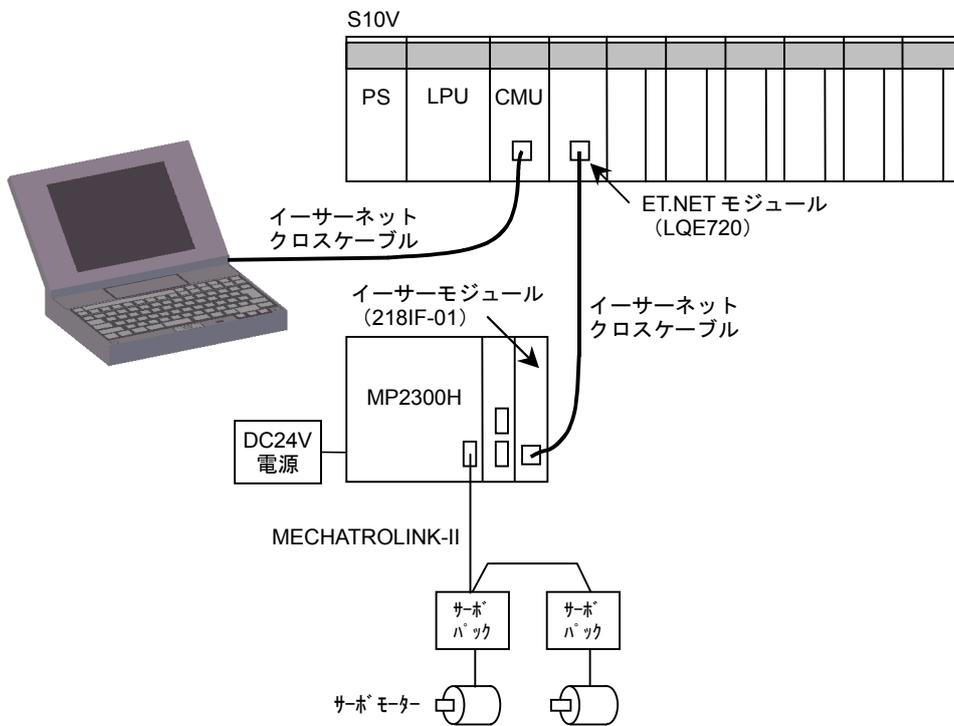


- (22) [CP-218] 画面のパラメータ設定から、以下の項目を指定し、Engineering Managerのメニューから[ファイル] - [保存] をクリックします。
- 伝送パラメータ設定 (自局の設定)  
IPアドレス：イーターモジュールのIPアドレス (例：192.168.1.3)
  - コネクションパラメータ設定

CNO	自局ポート	相手局IPアドレス	相手局ポート	コネクションタイプ	プロトコルタイプ	コード
01	34200	(ET.NETのIPアドレス)	34200	UDP	無手順	BIN
02	34201	(ET.NETのIPアドレス)	34201	UDP	無手順	BIN
03	34205	(ET.NETのIPアドレス)	34205	TCP	無手順	BIN

- (23) Engineering Managerのメニューから、[ウインドウ] - [モジュール構成] をクリックし、[モジュール構成] 画面を表示します。

- (24) セルフコンフィグレーション実行後、Engineering Managerのメニューから、[ファイル] – [保存 & FLASH保存] をクリックします。クリック後、フラッシュ書き込み確認メッセージが表示されるので  ボタンをクリックします。次に、CPUのSTOP確認メッセージが表示されるので  ボタンをクリックしてフラッシュ保存を実行します。
- (25) フラッシュ保存が完了すると、CPUをRUN切り替え確認メッセージが表示されるので  ボタンをクリックします。
- (26) Engineering Managerを閉じてFile Managerに戻り、(15) で作成したPLCフォルダを右クリックして表示されるメニューから、[ログオフ] をクリックします。
- (27) MP2300Hを電源OFF後、MP2300HおよびイーサーモジュールのスイッチをすべてOFFに設定し、パソコン、S10V、MP2300Hをイーサーネットクロスケーブルで以下の図に従い配線します。

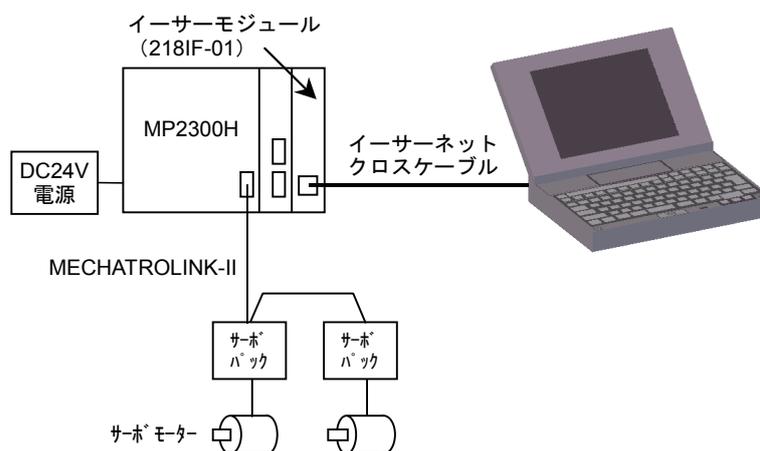


配線後、MP2300H、S10Vの順番で電源をONにします。

以上でMP2300Hのシステム構成の変更は完了です。

## 2. 全モジュールセルフコンフィグレーション手順

- (1) MP2300Hの電源がOFFであることを確認し、モーションモジュール、サーボパック、およびサーボモーターを接続します。
- (2) パソコンとMP2300Hをイーサネットクロスケーブルで以下の図に従い配線します。



- (3) MP2300Hおよびイーサネットモジュール（型式：218IF-01）のスイッチを以下の内容に設定し、MP2300Hの電源をONにします。

### ■ MP2300H

スイッチ名称	設定	動作モード
STOP	OFF	
SUP	OFF	
INIT	OFF→ON	デフォルトの設定で起動します。
CNFG	OFF→ON	接続された機器のセルフコンフィグレーションを実行します。
MON	OFF	
TEST	OFF	

### ■ イーサネットモジュール

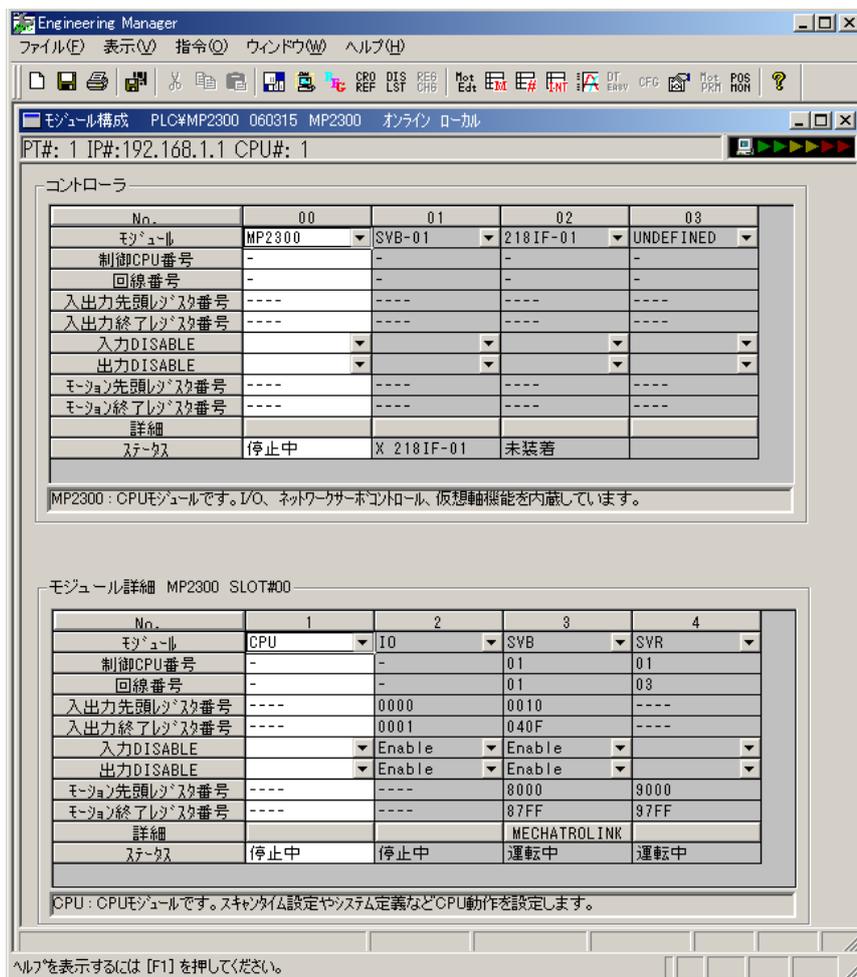
スイッチ名称	設定	動作モード
INIT	OFF→ON	デフォルトIPアドレス（192.168.1.1）で起動します。
TEST	OFF	

- (4) MP2300Hを電源ON後、全モジュールセルフコンフィグレーションが実行されます。セルフコンフィグレーション実行中は「RUN」LEDが点滅しますので、「RUN」LEDの点滅が終了し「RDY」LEDが点灯するまで待ってください。

- (5) Windows®のコントロールパネルの「ネットワークとダイヤルアップ接続」から、パソコンのIPアドレスとサブネットマスクを以下の内容に設定します。

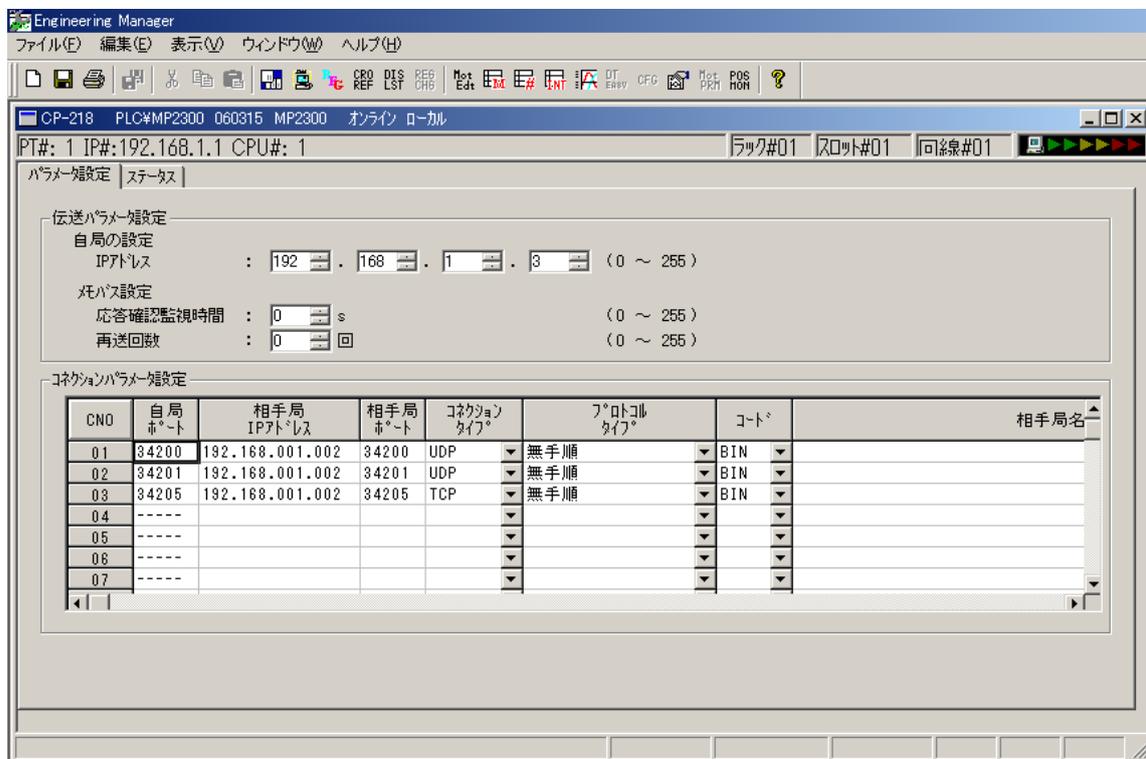
項目	設定	備考
IPアドレス	192.168.1.xx	xxは2～254の範囲で設定してください。
サブネットマスク	255.255.255.0	

- (6) Windows®の「Ping機能」から、パソコンとMP2300H間の接続が有効になっているか確認します。確認方法は「1. モジュールセルフコンフィグレーション手順」の(5)を参照してください。
- (7) 「1. モジュールセルフコンフィグレーション手順」の(6)～(18)を参考に、Engineering Managerの「モジュール構成」画面を表示します。



- (8) 「モジュール構成」画面の「コントローラ」グループから、モジュールが「218IF-01」となっているNo.をクリックします。

- (9) 「モジュール詳細」グループの表示が切り替わりますので、モジュールが「218IF」となっているNo.をダブルクリックし、[CP-218]画面を表示します。

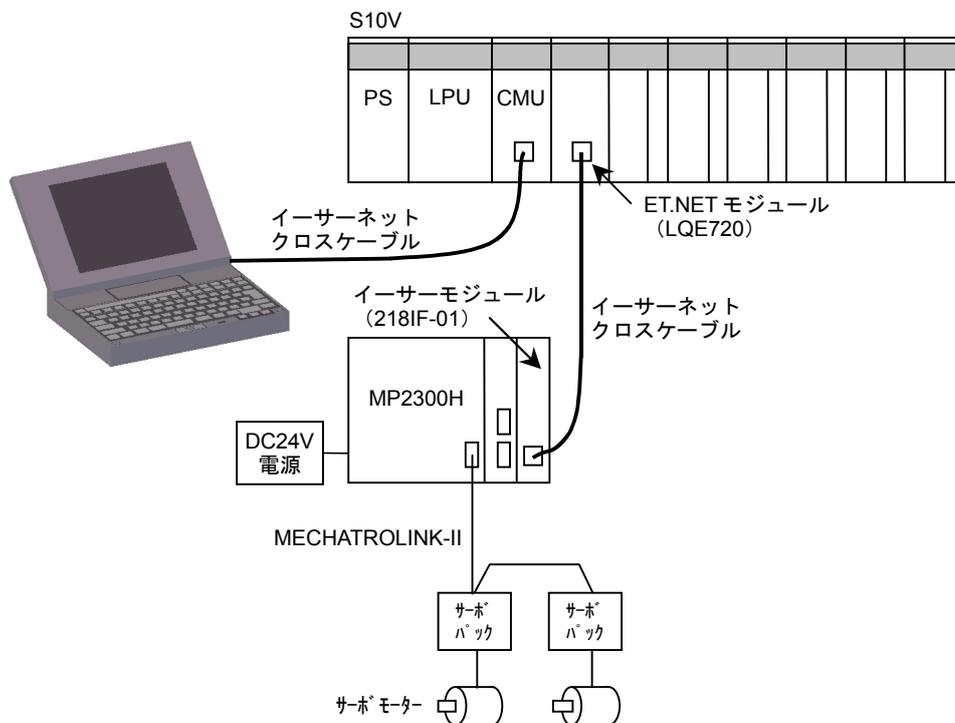


- (10) [CP-218]画面のパラメータ設定から、以下の項目を指定し、Engineering Managerのメニューから[ファイル] - [保存]をクリックします。
- 伝送パラメータ設定 (自局の設定)  
IPアドレス：イーサーモジュールのIPアドレス (例：192.168.1.3)
  - コネクションパラメータ設定

CNO	自局ポート	相手局IPアドレス	相手局ポート	コネクションタイプ	プロトコルタイプ	コード
01	34200	(ET.NETのIPアドレス)	34200	UDP	無手順	BIN
02	34201	(ET.NETのIPアドレス)	34201	UDP	無手順	BIN
03	34205	(ET.NETのIPアドレス)	34205	TCP	無手順	BIN

- (11) Engineering Managerのメニューから、[ウインドウ] - [モジュール構成]をクリックし、[モジュール構成]画面を表示します。

- (12) セルフコンフィグレーション実行後、Engineering Managerのメニューから、[ファイル] - [保存 & FLASH保存] をクリックします。クリック後、フラッシュ書き込み確認メッセージが表示されるので  ボタンをクリックします。次に、CPUのSTOP確認メッセージが表示されるので  ボタンをクリックしてフラッシュ保存を実行します。
- (13) フラッシュ保存が完了すると、CPUをRUN切り替え確認メッセージが表示されるので  ボタンをクリックします。
- (14) Engineering Managerを閉じてFile Managerに戻り、「1. モジュールセルフコンフィグレーション手順」の(15)で作成したPLCフォルダを右クリックして表示されるメニューから、[ログオフ] をクリックします。
- (15) MP2300Hを電源OFF後、MP2300HおよびイーサーモジュールのスイッチをすべてOFFに設定し、パソコン、S10V、MP2300Hをイーサーネットクロスケーブルで以下の図に従い配線します。



配線後、MP2300H、S10Vの順番で電源をONにします。

以上でMP2300Hのシステム構成の変更は完了です。