

ソフトウェアマニュアル
プログラミング

HIDIC
S10 シリーズ

HI-FLOW For Windows®

2α
S10mini
シリーズ

対象機種

HIDIC-S10/2	NESP-S25E
HIDIC-S10/2 E	NESP-2 E
HIDIC-S10/2 H	NESP-2 H
HIDIC-S10/2 Hf	NESP-2 Hf
S10mini モデルS	
S10mini モデルH	
S10mini モデルF	
S10mini モデルD	

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

1997年 8月 (第1版) SAJ-3-122(A) (廃版)
1999年11月 (第2版) SAJ-3-122(B) (廃版)
2000年10月 (第3版) SAJ-3-122(C) (廃版)
2003年 8月 (第4版) SAJ-3-122(D)

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写することは、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、最寄りの当社営業または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、当社は責任を負いかねますのでご了承ください。

当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、当社は責任を負いかねますのでご了承ください。

当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、当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。

当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。

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、フロー図形式で、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。

<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)

NESP (Nissan Electronic Sequence Processor) シリーズは、下記の対応を参照のうえ使用してください。

【HIDIC-S10αシリーズ】		【NESPシリーズ】
HIDIC-S10/2α	NESP-S25E
HIDIC-S10/2αE	NESP-2αE
HIDIC-S10/2αH	NESP-2αH
HIDIC-S10/2αHf	NESP-2αHf

<商標について>

・ Microsoft® Windows® operating system, Microsoft® Windows® 95 operating system, Microsoft® Windows® 98 operating system, Microsoft® Windows® 2000 operating system, Microsoft® Windows® XP operating systemは、米国Microsoft Corporationの米国およびその他の国における登録商標です。

・ Ethernetは米国Xerox Corp.の登録商標です。

その他、記載されている会社名、製品名は、各社の商標または登録商標です。

目 次

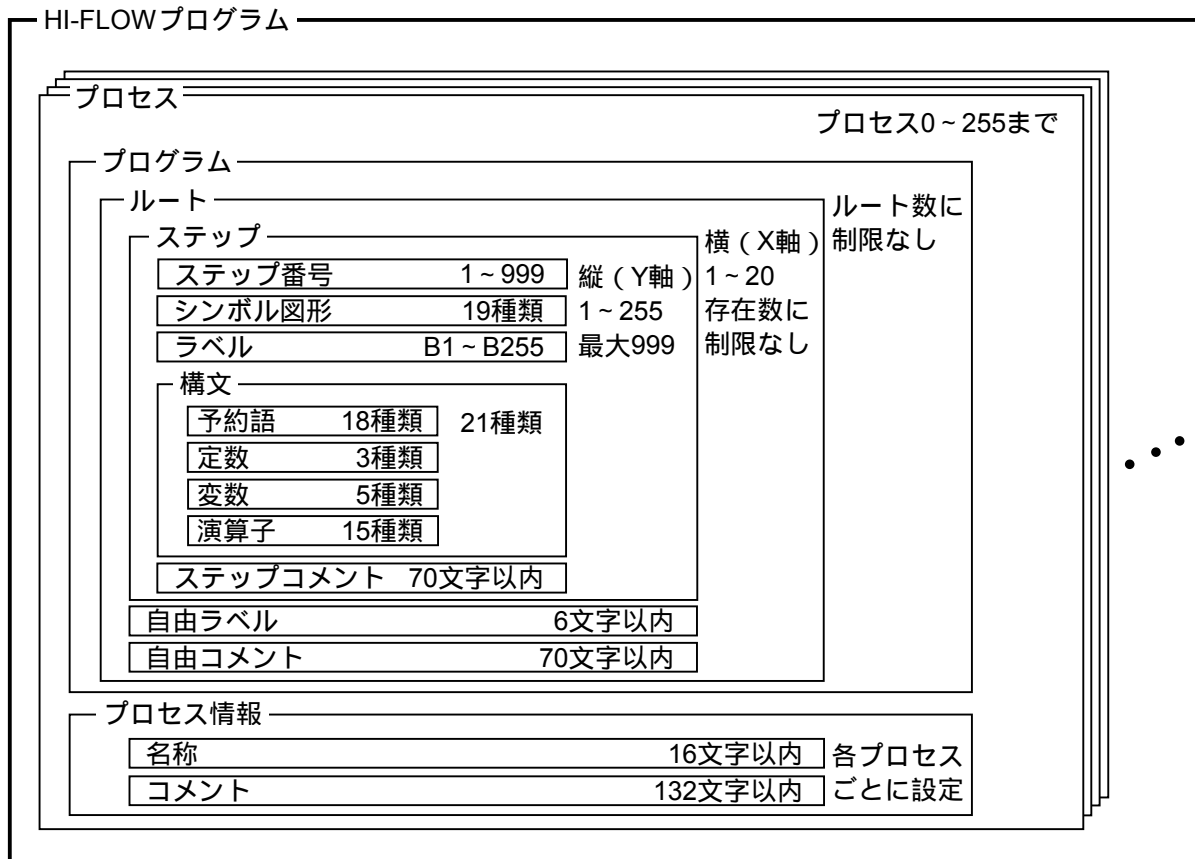
1	HI-FLOWプログラムの構成	1
2	このマニュアルの使い方	3
2.1	概 説	4
2.2	構文説明	5
2.3	応用命令説明	6
3	プロセス	9
3.1	プロセスとは?	10
3.2	プログラム	15
3.3	プロセス情報	28
4	構文説明	29
4.1	プロセススタート、プロセスエンド	30
4.2	ルートスタート、ルートエンド	34
4.3	ウェイト	35
4.4	ボックス	37
4.5	コントロールボックス	42
4.6	リピートスタート、リピートエンド	46
4.7	イフ	47
4.8	ジャンプ	49
4.9	エスケープ	50
4.10	パラスタート、パラエンド	51
4.11	セレクト、セルウェイト、セレクトエンド	52
4.12	マルチエントリ	53
4.13	コール	54
4.14	ファンクション	55
4.15	前条件付きウェイト	55
5	応用命令	57
5.1	概 説	58
5.2	使用方法	58
5.3	パラメータ	58
5.4	演算時の型変換	60

5.5	システムエラーフラグ	61
5.6	機能説明	62
補	足	113
補足1	HI-FLOWプログラムの流れ	114
補足2	PCsメモリ	115
補足3	オンラインモード	116
補足4	進行チェック	120
補足5	HI-FLOWプログラムとCPU負荷	122
索	引	127

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・節とページを示します。

項 目	対応した章・節	ページ
プロセス	第3章	9
・プログラム	3.2節	15
・ルート		15
・ステップ		19
・ステップ番号		20
・シンボル図形		20
・ラベル		23
・構文		23
・予約語		24
・定数		24
・変数		24
・演算子		26
・ステップコメント		26
・自由ラベル		27
・自由コメント		27
・プロセス情報	3.3節	28
・名称		28
・コメント		28

2.2 構文説明

このマニュアルは、概説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・節とページを示します。

項目	図形	対応した章・節	ページ
構文説明		第4章	29
・プロセススタート、 プロセスエンド	●	4.1節	30
・STP	●		30
・RST			31
・CLR			32
・ACT			32
・ルートスタート、 ルートエンド	⌋ ⌌	4.2節	34
・ウェイト		4.3節	35
・条件式	+		35
・タイマ			35
・出力ビット			35
・ウェイトタイマ			35
・ボックス		4.4節	37
・代入式			37
・ON文			38
・OFF文	□		39
・パラレルタイマ			39
・TUP			40
・TRS			41
・コントロールボックス		4.5節	42
・ACT	■		42
・RST			43
・STP			43
・CLR			44
・リピートスタート、 リピートエンド	⌋ ⌌	4.6節	46
・イフ	◇	4.7節	47
・ジャンプ	↳	4.8節	49
・エスケープ	✖	4.9節	50
・パラスタート、 パラエンド	⌋ ⌌	4.10節	51
・セレクト、 セルウェイト、 セレクトエンド	⌋ ≡≡≡	4.11節	52
・マルチエントリ	⌋	4.12節	53
・コール	□	4.13節	54
・ファンクション	○	4.14節	55
・前条件付きウェイト	+*	4.15節	55

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 R	63
	減算	SUB	S-D R	64
	+1	INC	S+1 S	65
	-1	DEC	S-1 S	66
	乗算	MUL	S*D R	67
	除算	DIV	S/D R	68
	剰余	MOD	S/Dの余り R	69
	スケール変換	SCL	S*D1/D2 R	70
論理演算命令	論理積	AND	S AND D R	71
	論理和	OR	S OR D R	72
	排他的論理和	EOR	S EOR D R	73
	否定	NOT	NOT S R	74
比較演算命令	=	EQU	S=Dの真偽 R	75
	<>	NEQ	S<>Dの真偽 R	76
	>	GT	S>Dの真偽 R	77
	>=	GE	S>=Dの真偽 R	78
	<	LT	S<Dの真偽 R	79
	<=	LE	S<=Dの真偽 R	80
	テスト	TST	Sの符号 R	81
データ転送命令	転送	MOV	S D	82
	一括転送	MOM	S ~ Sn D ~ Dn	83
	交換	EXC	S D	84
	FIFO書き込み	PSH	S D (FIFOテーブル)	85
	FIFO読み出し	POP	S (FIFOテーブル) D	86
	アドレスセット	AST	Sアドレス D	87
	サーチ	SCH	S =D (n) nをRにセット	88
データ変換命令	BIN-BCD	BTD	BIN BCD S -----> R	89
	BCD-BIN	DTB	BCD BIN S -----> R	90
	BIN-7SEG	SEG	BIN 7セグメント S -----> R	91

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)	92
		ASU	S -----> (R, R+1)、(R, R+1, R+2, R+3)	93
	ASC-BIN	APB	ASCII (パック、アンパック) BIN	94
		AUB	(S, S+1)、(S, S+1, S+2, S+3) -----> R	95
	絶対値	ABS	S R	96
	+/-	NEG	-S R	97
	デコード	DCD	Sの $2^{11} \sim 2^{15}$ Rの 2^n ビットON	98
エンコード	ECD	Sの最初のONビット番号 Rの $2^{11} \sim 2^{15}$	99	
シフト命令	論理右シフト	LSR	S 論理右シフト D R	100
	論理左シフト	LSL	S 論理左シフト D R	101
	算術右シフト	ASR	S 算術右シフト D R	102
	算術左シフト	ASL	S 算術左シフト D R	103
回転命令	右回転	ROR	S 右回転 R	104
	左回転	ROL	S 左回転 R	105
関数処理命令	リミッタ	LIM		106
	デッドバンド	BND		107
	デッドゾーン	ZON		108
	平方根	ROT		109
	最大値	MAX		110
	最小値	MIN		111
特殊命令	クリア	XCLR YCLR GCLR RCLR KCLR TCLR UCLR CCLR VCLR ECLR FCLR JCLR QCLR HHCLR		112

3 プロセス

3 プロセス

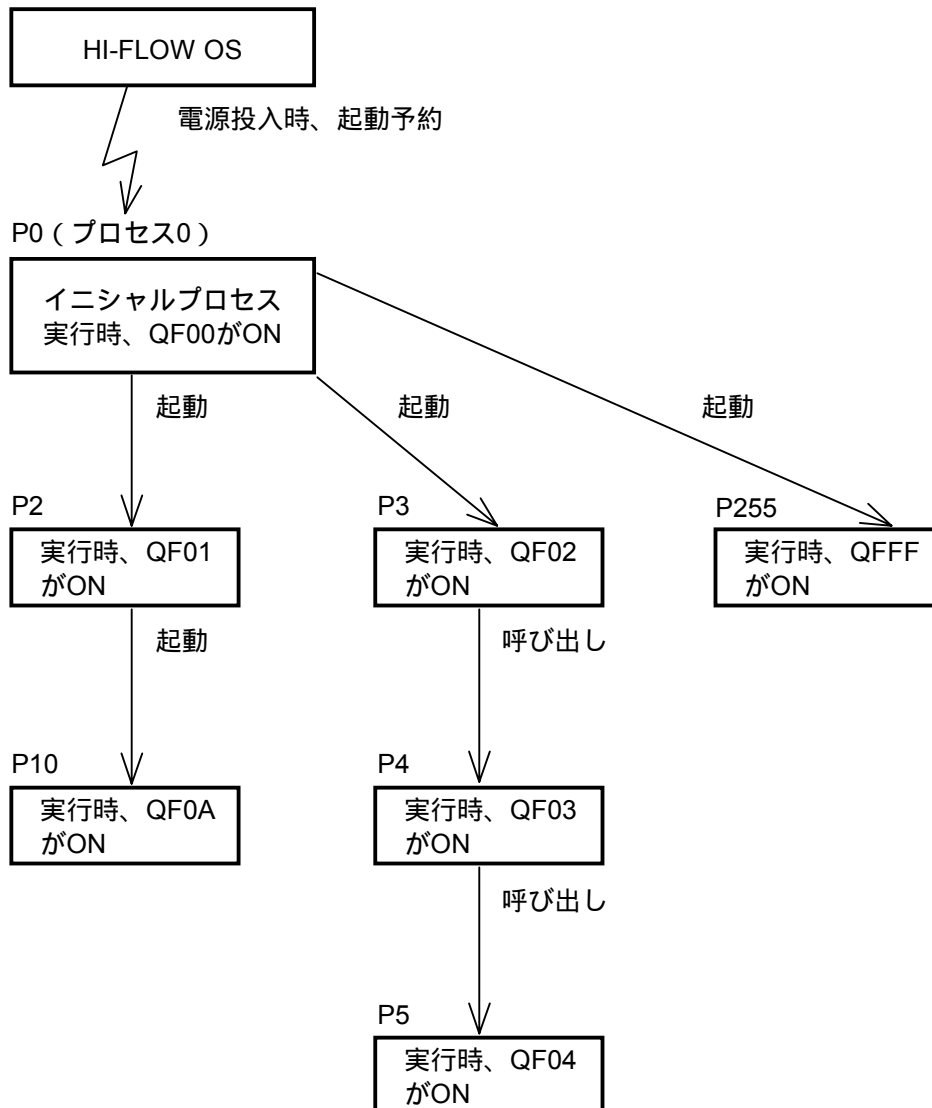
3.1 プロセスとは？

プロセススタート (●) とプロセスエンド (●) 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となります。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レジスタがONし、その状態を監視できます (標準QF00 ~ QFFF、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より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。プロセスの情報はイニシャル。PI/O値は保持。タイマ経過値はアップとリセットを指定。
クリア	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）を0クリアする。
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。プロセスの情報はイニシャル。PI/O値は保持。タイマ経過値はアップとリセットを指定。

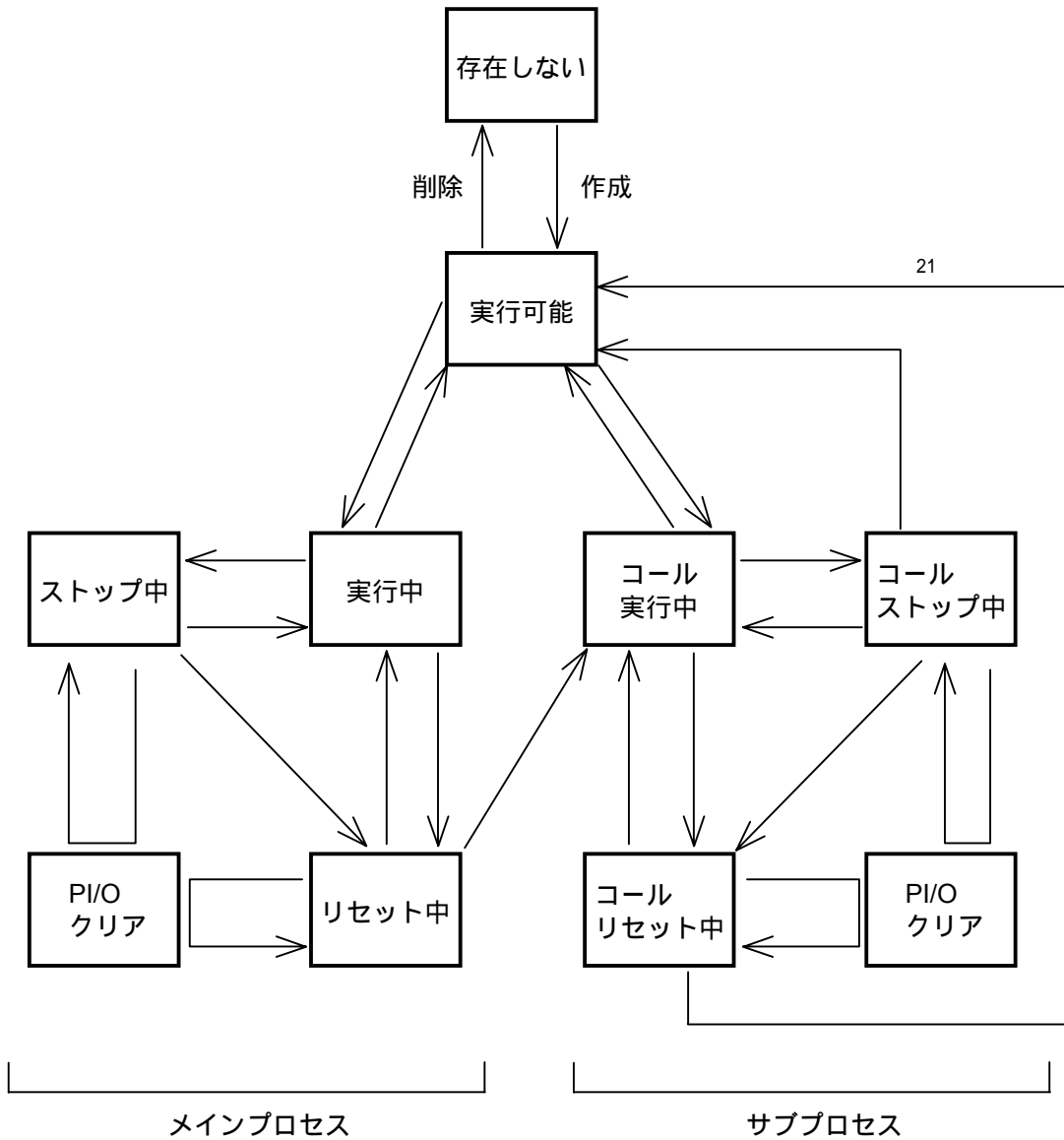
ストップ中、リセット中は、一度の条件成立で状態が遷移し、その条件が不成立になっても状態はそのままですが、クリアは条件が成立するたびに毎行われます。

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]



コントロールボックスACT (■)

エスケープ (×)

プロセススタートSTP (●)、コントロールボックスSTP (■)

プロセススタートACT (●)、コントロールボックスACT (■)

プロセススタートRST (●)、コントロールボックスRST (■)

プロセススタートACT (●)、コントロールボックスACT (■)

プロセススタートRST (●)、コントロールボックスRST (■)

プロセススタートCLR (●)、コントロールボックスCLR (■)

プロセススタートCLR (●)、コントロールボックスCLR (■)

プロセスコール (☐)

プロセスコール (☐)

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 (●)

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 (●)

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 (●)

プロセススタートRST (●)

プロセススタートACT (●)

プロセススタートRST (●)

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 (●)

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)、タイマ経過値の選択 (アップ/リセット/計測続行) は起動方法に従います。

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キースイッチRUNとSIM RUNは区別せず、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りPCs停復電と同じ効果 (ただし、HI-FLOW関連のみ) の後(c)になることもできます (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 (WT、PT) も計測停止します。

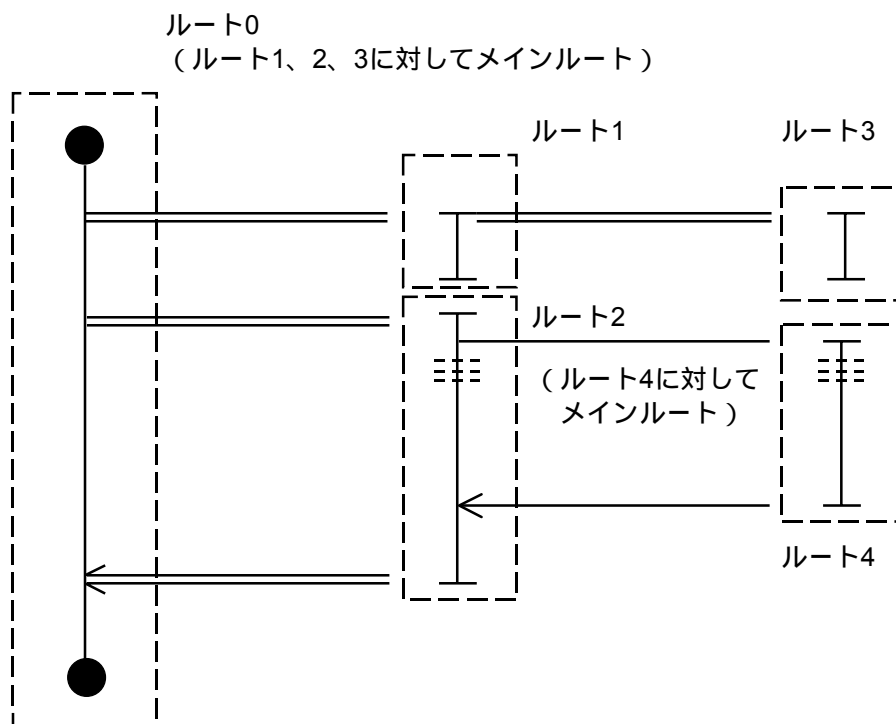
3.2 プログラム

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（●）またはルートスタート（T）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り同期、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（⊢）により分岐し、パラエンド（≡）またはセレクトエンド（⊣）により合流します。

ルートに対してユーザ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のみ管理しています。



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ただけとなります。

選択ルートは、例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。

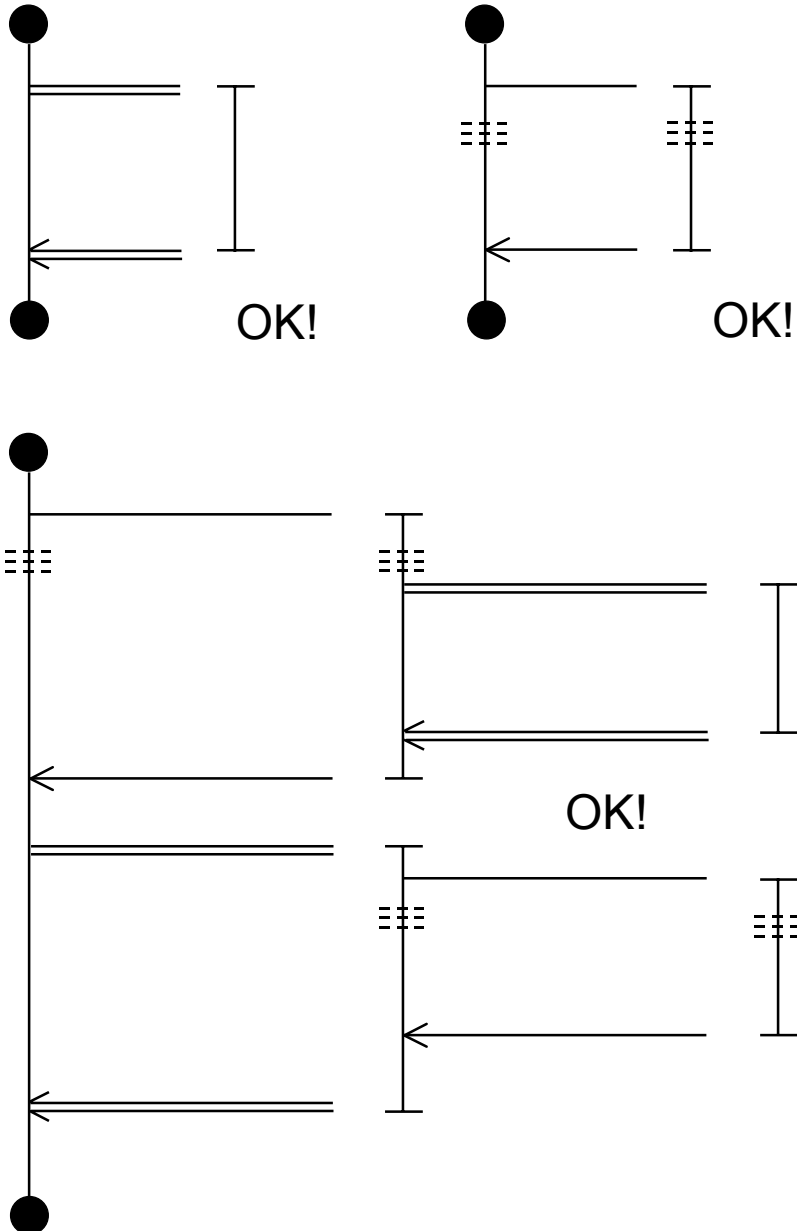
3 プロセス

(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。

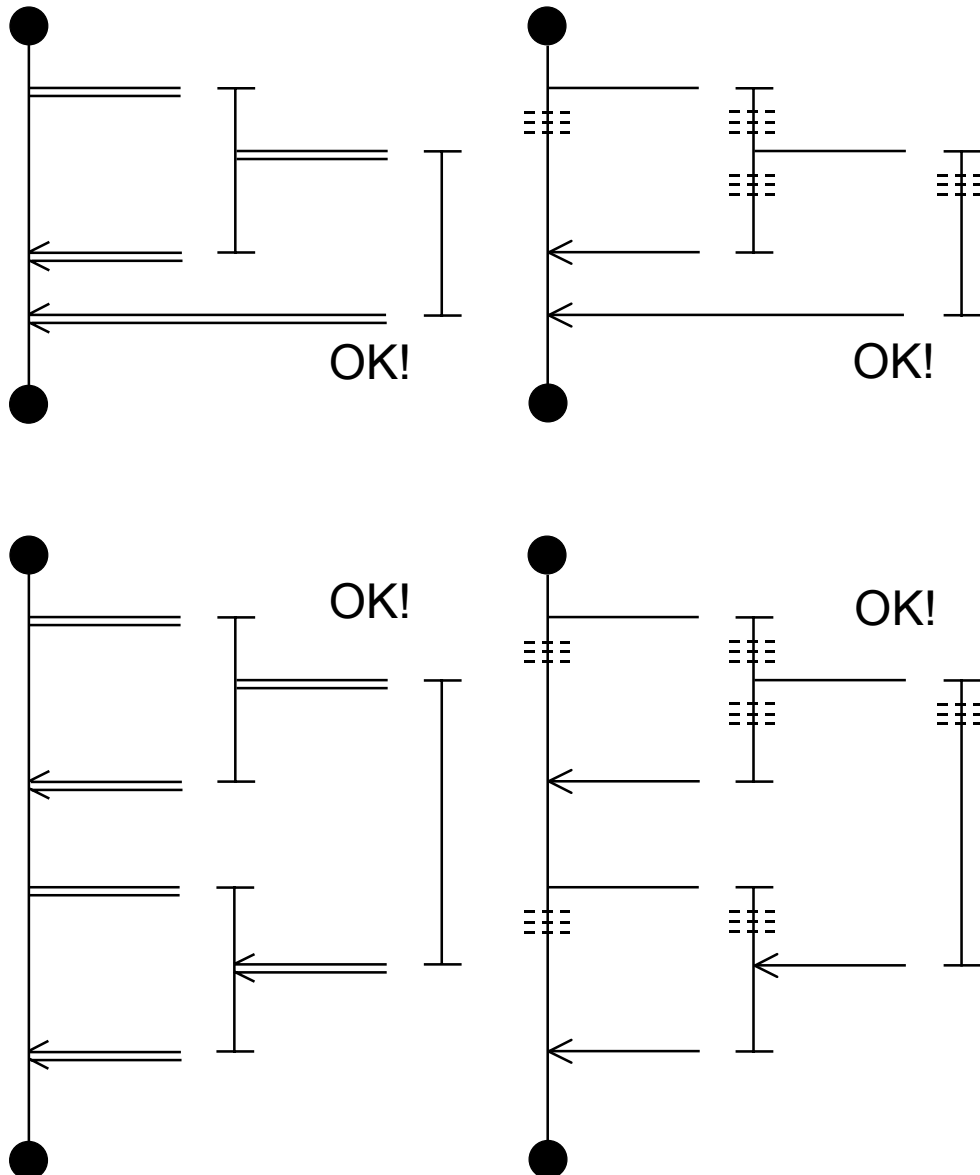


(b) 分岐開始ルートと分岐合流ルートが異なる

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。

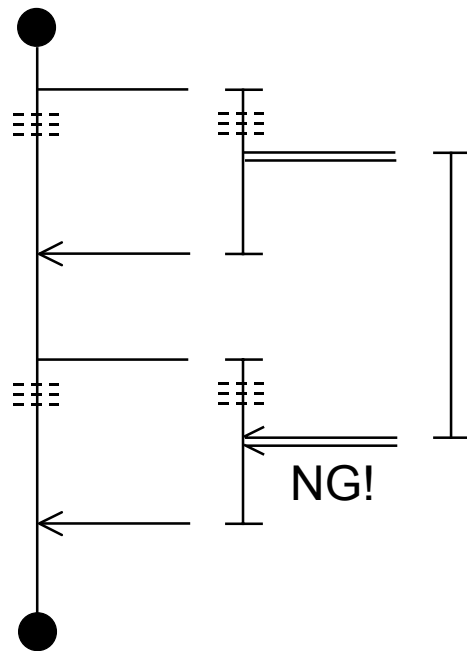
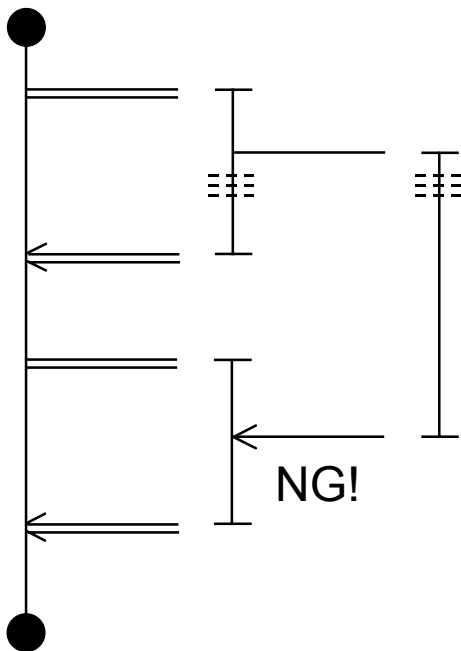
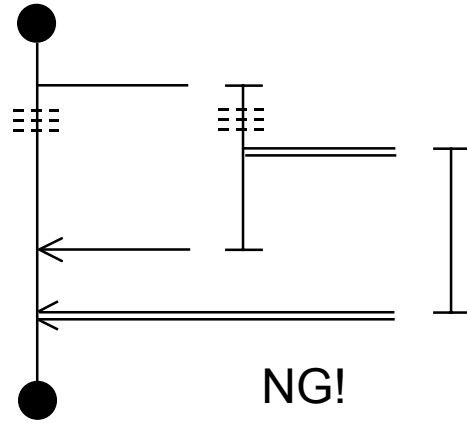
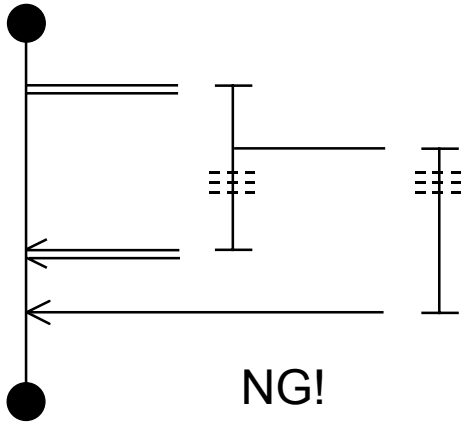
これはプログラムとしては作成できますが、実際の動作はうまく動作しないということです。

[正常動作]



3 プロセス

[正常動作しない場合]



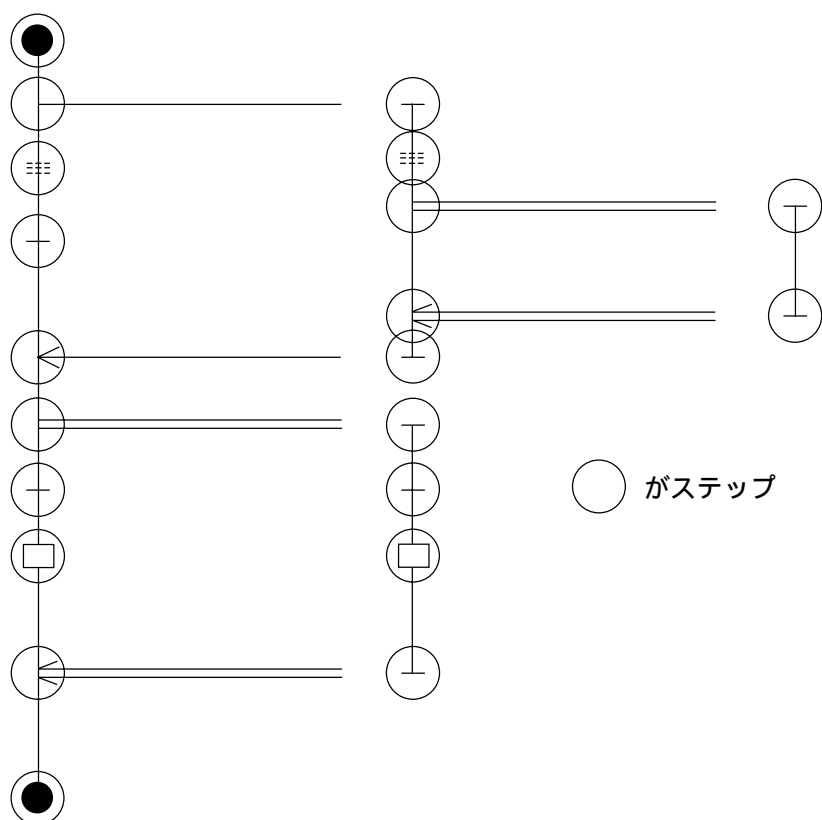
ステップ

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になります。

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、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

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、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。

[HI-FLOWで使用できる図形一覧]

(1/2)

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		ルート スタート	サブルートの開 始	なし	
4		ルート エンド	サブルートの終 了	なし	
5		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
6		リピート エンド	繰り返し処理の 終了	なし	
7		イフ	条件による処理 の分岐	あり	他ルート分岐可能
8		ジャンプ	無条件分岐	なし	他ルート分岐可能
9		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、同じ スキャンでメインへ戻る。
10		パラ スタート	同期サブルート への分岐	なし	
11		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じ スキャンで次ステップへ
12		セレクト	選択サブルート への分岐	なし	
13		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
14		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い でもよい スキャン遅れなしで次ス テップへ
		マルチ エントリ	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	

3 プロセス

(2/2)

番号	図形	名称	機能	構文	備考
15	+	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
16	□	ボックス	PI/O出力	あり	インタロック付きY出力あり
			代入式	あり	
			PI/O波形出力	あり	
			タイマリセット	あり	7個までの制限なし
			タイマアップ	あり	
17	■	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	マスタリセット機能あり
			タスク制御	あり	
18	□	コール	他プロセスのサブコール	あり	マスタリセット機能あり
19	○	ファンクション	応用命令	あり	
20	+*	前状態クリア付条件	条件移行時PI/Oクリア	あり	ウェイトと対で使用
		前状態クリア付ウェイトタイマ	タイマアップ時PI/Oクリア	あり	

ラベル

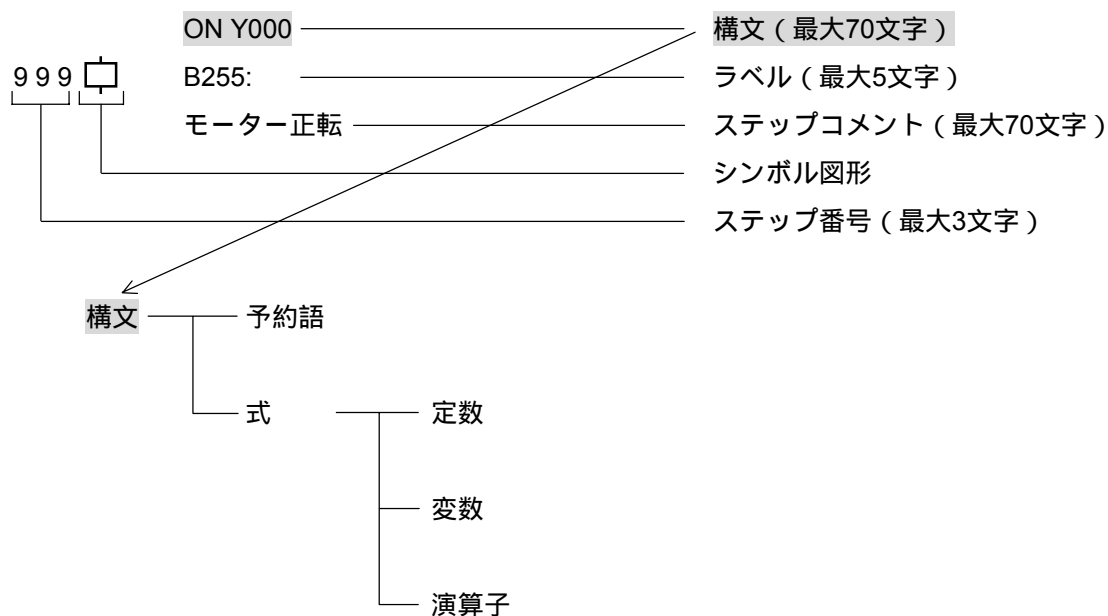
B1～B255（プロセスごとに作成可能、他プロセスへの分岐指定はできません）と:（コロン）で表し、分岐図形からの飛び先を示します。ステップにのみ付加できます。



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、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

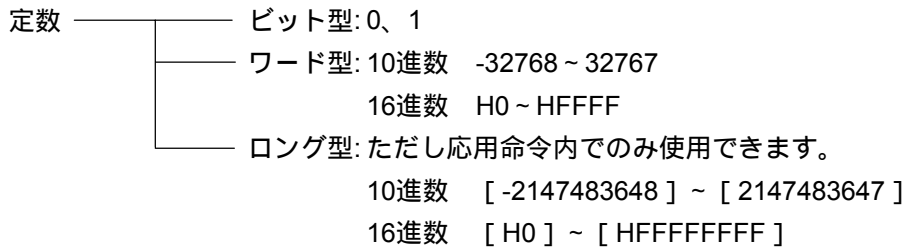
予約語

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（「5 応用命令」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。

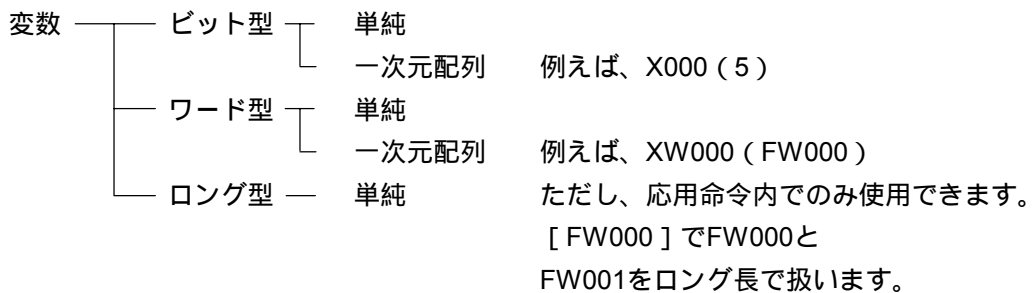


変数

HI-FLOWでは実PI/Oレジスタ（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レジスタ一覧を示します。



[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	
	タイマ	WT	000 ~ 255		10進表記
		PT	000 ~ 255		
	カウンタ	CN	000 ~ 127		10進表記
ラベル	B	001 ~ 255 ユーザ設定ラベルあり (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、カッコ / 算術 / 関係 / 論理の4種類があります。また、算術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、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のみ付加できます。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まで記述できます。これにより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のみ付加できます。

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）

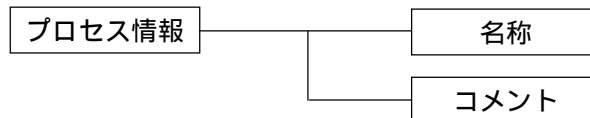
自由ラベル、自由コメントを組み合わせる場合、入力できるのは合計で70文字（自由ラベルの: [コロン] を含む）までとなります。

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随的な情報の定義付けを行います。これによりユーザ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いて、プロセス情報コマンドによってユーザが任意に変更できます。



名称

プロセス情報の中の名称で、ユーザ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

プロセス情報の中のコメントで、ユーザ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イニシャル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たら処理を行います。終了していないと終了するまで待ち状態となります。起動された際、マスタリセット指定されていたら自プロセスで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)。

また、自プロセスで使用中のタイマの扱いは起動方法に従います。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を打ち切ります。無指定の場合はタイマの計測を続行します。

[構文]

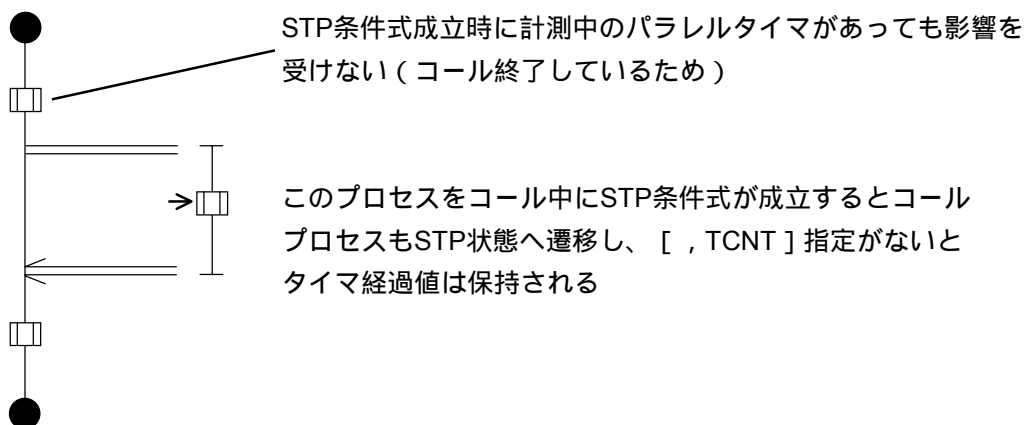
```
● [ { STP 条件式 [ , TCNT ] [ { ON PI/Oビット群 [ :OFF PI/Oビット群 ] } } ] }
    [ { OFF PI/Oビット群 [ :ON PI/Oビット群 ] } } ]
    { , RST 条件式 [ , TUP ] [ { ON PI/Oビット群 [ :OFF PI/Oビット群 ] } } ] }
    [ { OFF PI/Oビット群 [ :ON PI/Oビット群 ] } } ]
    { , CLR 条件式 }
    { , ACT 条件式 } ]
    *PI/Oビット群 - PI/Oビット式 [ , PI/Oビッ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

● 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)。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は計測続行し、無指定だと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。



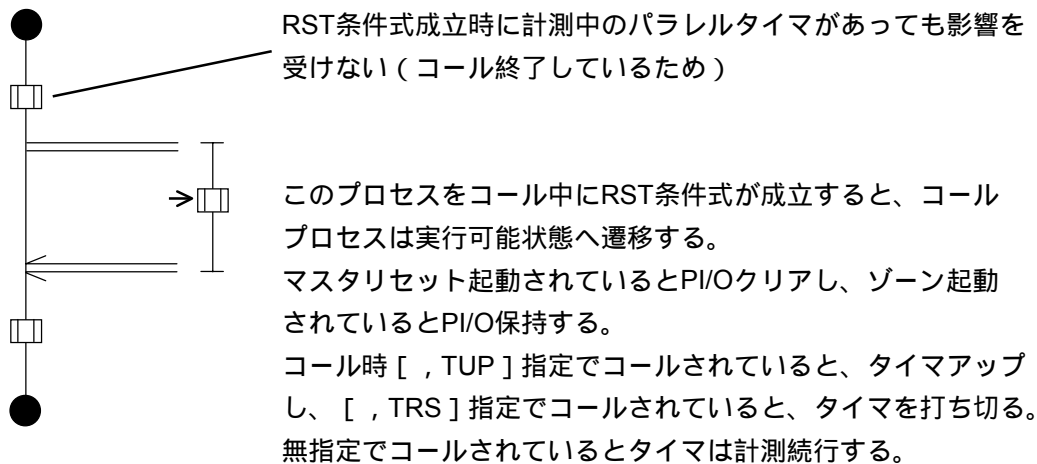
RST

- ・ 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)。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、タイマアップします。無指定だと経過値を0クリアし、計測を打ち切ります。

4 構文説明

- ・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、タイマの扱いは起動方法に従います。

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 (タイマ経過値保持)
- ・ X001がONのときリセット状態へ (タイマ打ち切り)
- ・ 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、パラレルタイマ内ビット型PI/Oを0クリア
- ・ 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、パラレルタイマ内ビット型PI/Oを0クリア
- ・ 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

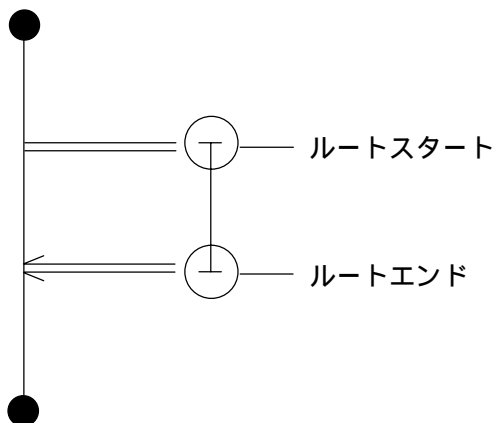
┌ 構文なし

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。

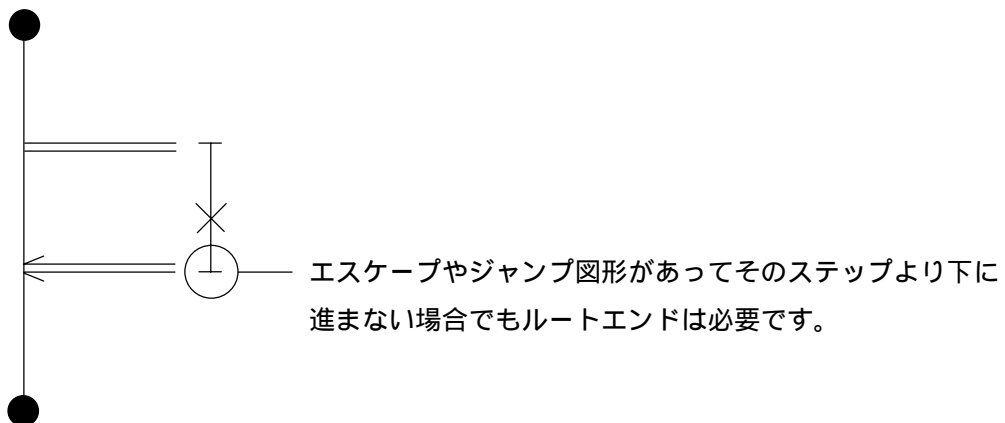
サブルートを作成することにより同期構文や選択分岐構文が構築できます。

[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（指定時間経過待ち）の場合があります。

[構文]

```

+ { 条件式 [ , タイマ , 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（バージョン07-00以降のHI-FLOWシステムでのみ使用可能）

- ・条件式成立までの監視タイマです。単位は100ミリ秒です。
- ・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数は64個までです。同時に監視状態になるタイマ数が64個を超えないようにしてください。

出力ビット（バージョン07-00以降のHI-FLOWシステムでのみ使用可能）

- ・上記タイマ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。
- ・出力ビットに指定可能なレジスタは下記ビット型レジスタです。
(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

- ・ウェイトタイマ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100ミリ秒刻みで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。
- ・複数箇所同じ番号のウェイトタイマが指定時間経過待ちになった場合、タイマを先に占有したステップがタイマ開放するまで他のステップは指定PI/O（標準HH1FA）をONしタイマ開放待ちになります。したがって、他のタイマ遅延時間が延びる結果になります。
- ・ウェイトタイマ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 (+) のプログラム例]

1 . + X000

X000がONのとき次ステップに進む。

2 . + GW000<H2000

GW000がH2000より小さくなったとき次ステップに進む。

3 . + 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レジスタがONしたとき次ステップへ進む。

4 . + 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 . + 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 . + 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ができます。

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

(ワード型変数)

YW、GW AW、RW KW、MW EW、ZW JW、QW DW、FW	=	YW、GW AW、RW KW、MW EW、ZW JW、QW DW、FW XW、SW 10進定数 16進定数	() & ~ ^ * / + -	YW、GW AW、RW KW、MW EW、ZW JW、QW DW、FW XW、SW 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ワードで表せられ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ワードで表せられ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が、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。

[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

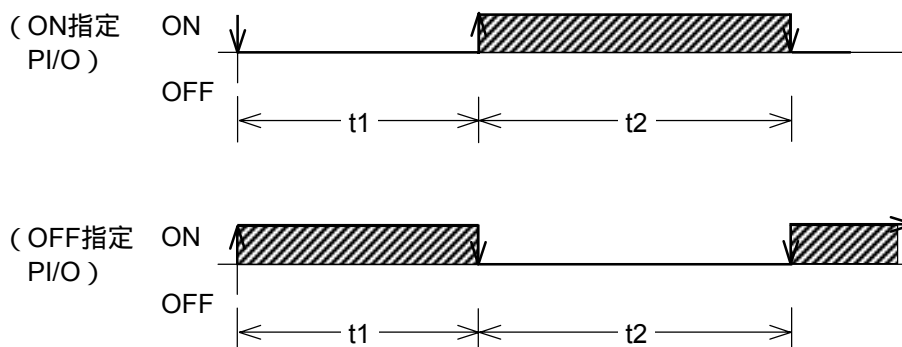
任意のPI/Oに波形を出力します。t1は立ち上がり時間、t2は立ち下がり時間を示します。

t1が0のとき、ON指定PI/Oはt2時間経過後の立ち下がりのみとなり、OFF指定PI/Oはt2時間経過後の立ち上がりのみとなります。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のみとなり、OFF指定PI/Oはt2時間経過後の立ち下がりのみとなります。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100ミリ秒単位、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。


タイマ起動時、指定タイマがすでに使用中の場合は指定PI/O（標準HH1F9）をONしタイマ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 () 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 (タイマアップ)


計測中のタイマをアップ状態にします。

ウェイトタイマの場合、計測中のタイマ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。

パラレルタイマの場合、タイマ経過値を 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。

[タイマアップ () のプログラム例]

1 .  TUP WT001 , WT002 , PT001 , CN001


ウェイトタイマ1,2、パラレルタイマ1、カウンタ1をアップします。

TRS (タイマリセット)

計測中のタイマをリセットします。

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

[タイマリセット () のプログラム例]

1 .  TRS WT001 , WT002 , PT001 , CN001

ウェイトタイマ1, 2、パラレルタイマ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クリアを行います。

[構文]

```

■ { ACT Pxxx { [-Pxxx] [ , ステップ番号 ] [ , MRST ] [ { , TUP } ] } }
                                     { , TRS }
                                     { , TASK , ファクタ番号 }
{ 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	
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をリセットする。
8	マスタリセット指定起動	,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クリアする（ON文、パラレルタイマ）。
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をアップする。無指定の場合はタイマをリセットする。このオプションが有効なのは指定プロセスのみ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起動か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の計測を続行する。無指定の場合はタイマ計測を停止する。 このオプションは指定プロセスがコールでリンクし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	
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のみ許可。他プロセスでのPI/O使用状況はチェックせ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(■)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クリアして次ステップへ進みます。

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。

同じ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せ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)

\sphericalangle 構文なし

[リピートスタート(∇) , リピートエンド(\sphericalangle) のプログラム例]

1 .

∇ CN000 (1 , 10)

\sphericalangle

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
リピートエンドの次のステップへ進みます。

∇

リピートエンドは実行後はすぐにリピートスタートを実行します。

2 .

∇ CN127 (1 , 5 , 2)

\sphericalangle

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
リピートエンドの次のステップへ進みます。

∇

3 .

∇ CN001 (FW000 , FW001 , FW002)

\sphericalangle

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)以下を省略した場合で、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)

< 留意事項 >

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 (\diamond) のプログラム例]

1 . \diamond 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 . \diamond 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 . \diamond 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 . \diamond 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 . \diamond 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。

パソコンHI-FLOWでは自由ラベル（6文字以内、ユーザが名称を自由に設定可能、ステップ以外にのみ付加可能）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

[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、そのステップよりすぐ実行する。

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、そのステップよりすぐ実行する。

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の扱いは起動方法に従います（TUP、TRS オプション）。

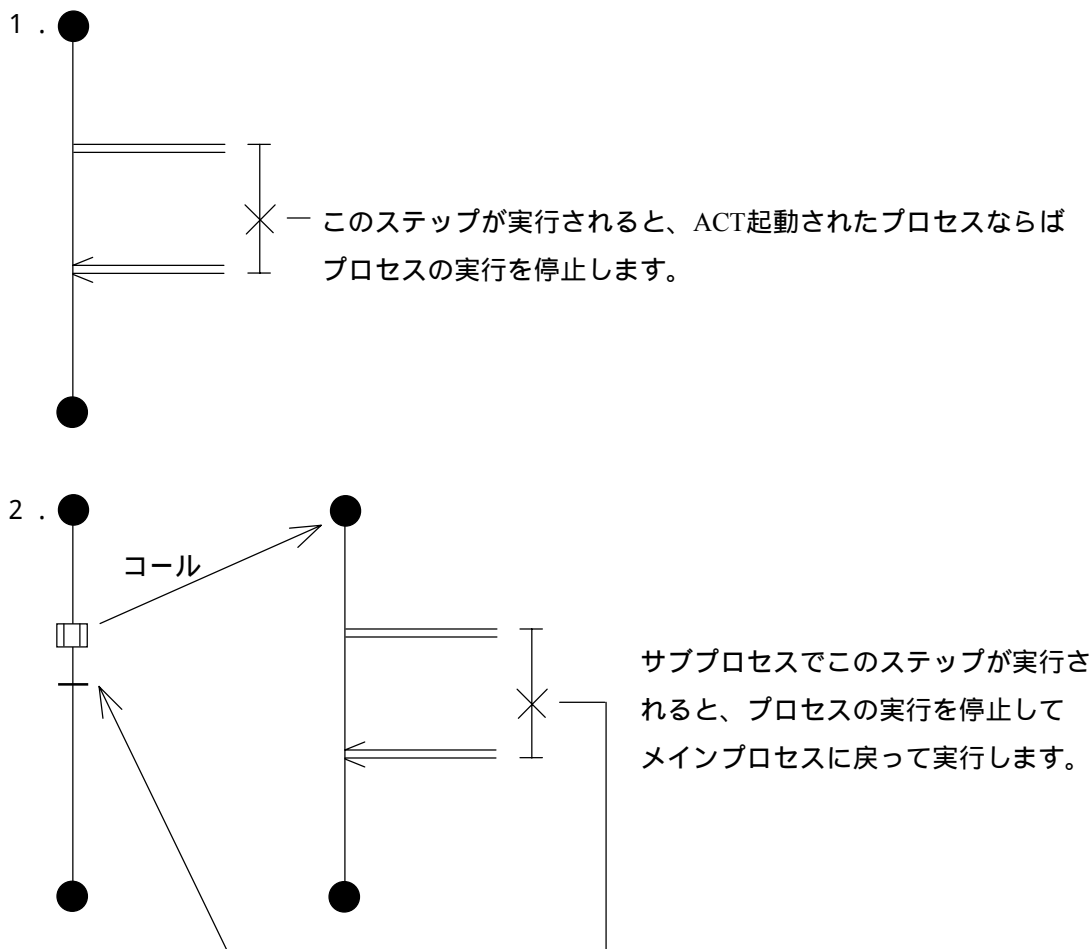
サブプロセスの場合も基本的にはメインプロセスと同じですが、実行箇所を同じスキャンでメインプロセスに戻します。

マスタ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クリアします（ON文、パラレルタイマ）。

[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。

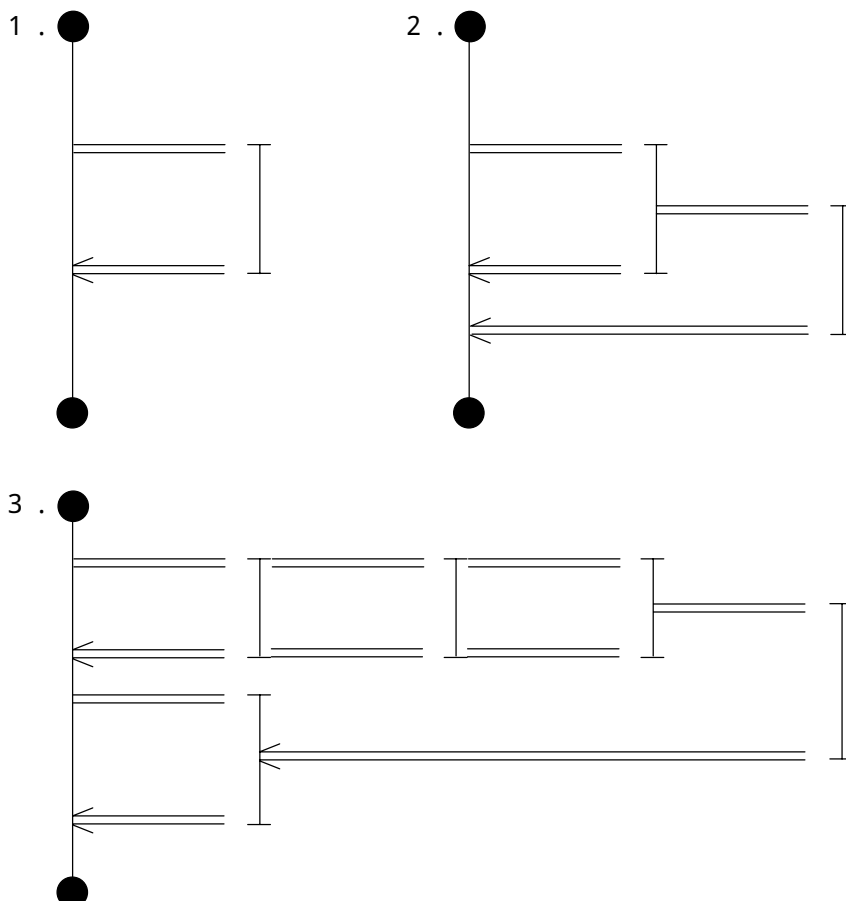
従来はパラエンドが合流しているサブルートの終了を監視（つまりメインルートは実行中）していたため次ステップ実行が1スキャン遅れました。今回はパラエンド、ルートエンドともに自分が最後に合流したかをチェックし、最後ならばメインルート合流部の次ステップ実行を、最後でなかったら自ルートの実行を終了（メインルートが常に実行中とは限らない）するような変更により、スキャン遅れは起こりません。

[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]



4.11 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（選択されたルートのみ実行）。

画面左側のルートから条件式をチェックするので、同じ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。今回は同じ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]

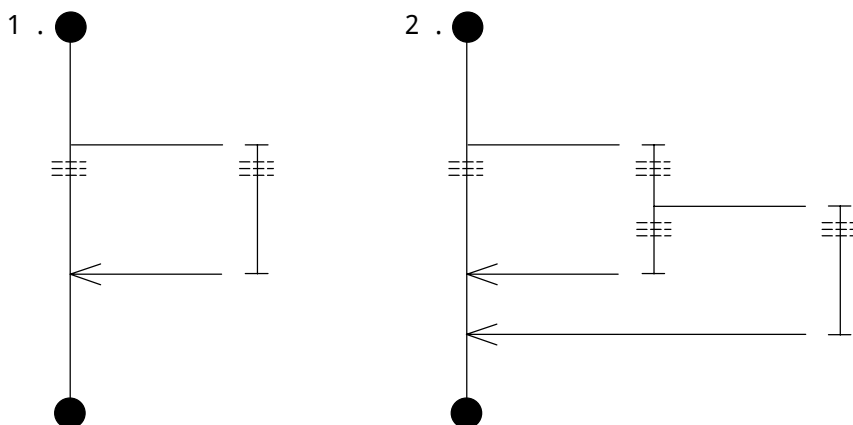
┆ 構文なし

≡≡≡ 条件式 [, タイマ , 出力ビット]

┆ 構文なし

- ・ タイマ（バージョン07-00以降のHI-FLOWシステムでのみ使用可能）
 - ・ 出力ビット（バージョン07-00以降のHI-FLOWシステムでのみ使用可能）
- タイマ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

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、同じスキヤンで複数の条件が成立すると一番ステップ番号の小さいステップから再実行します。

マルチエントリ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（PT、WT）、カウンタ（CN）、コール中プロセス、そのマルチエントリ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は設定できません。

[マルチエントリ (⌘) 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。

4.13 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までネスタリングができます。

[, MRST] オプションでマスタリセットコールができます。マスタ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「5 応用命令」を参照してください。

[構文]

 応用命令名称 パラメータ [, パラメータ] ~

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]) }

5 应用命令

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、代入のみ（ワード長のみ）です。そこでパソコンHI-FLOWではラダー図と同じ機能の応用命令をサポートしています。

5.2 使用方法

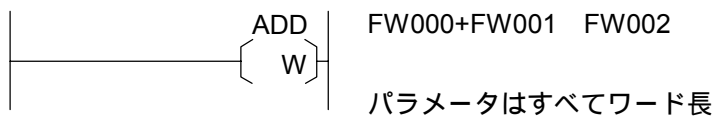
応用命令は次に示すような方法でプログラミングします。

○ 応用命令名称 パラメータ [, パラメータ] ~

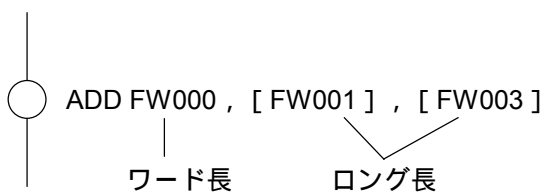
5.3 パラメータ

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	0001		0001
1	0000		0000
2	0002	→ 応用命令実行後 →	0002
3	1111		0000
4	1111		0003

パラメータは通常ソース、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。

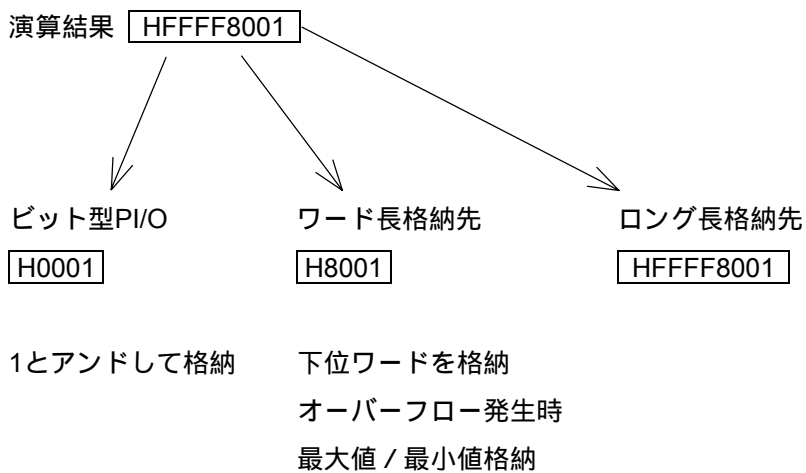
アドレッシングモード	パラメータ																		
	ビット型PI/O	ワード型PI/O	定数																
	X000 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>データ1</td></tr><tr><td>データ2</td></tr></table> X001 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>データ2</td></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><tr><td>データg</td></tr><tr><td>データh</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																			
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 位ワードのみ																
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。

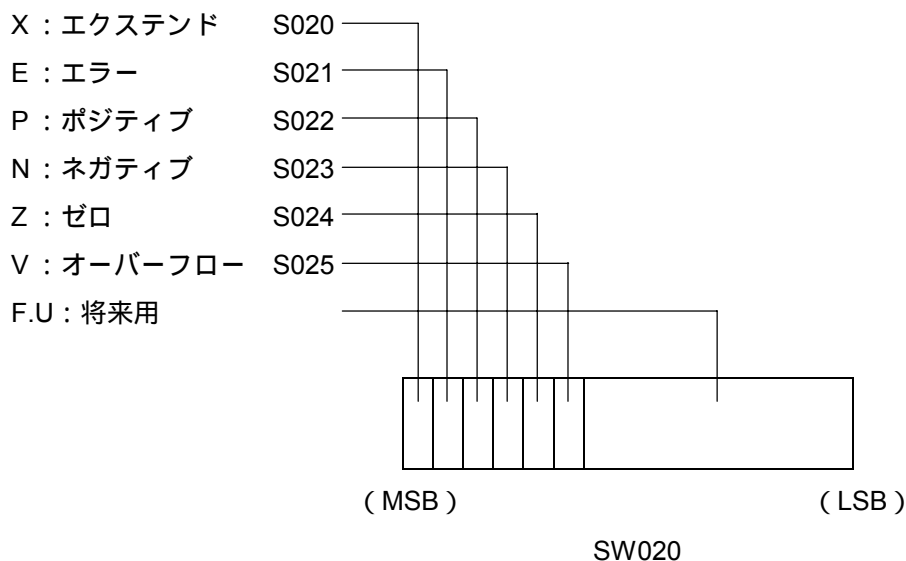
演算結果を格納する際は格納先にあわせて結果を型変換します。



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数が異なる場合。

CPUにメモリプロテクトがかかっているときにリザルト (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)。



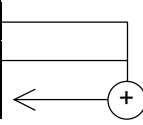

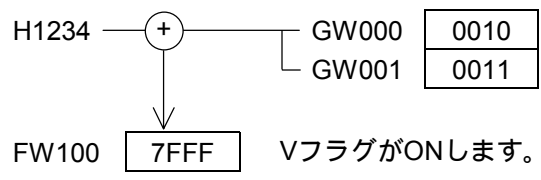
オーバーフローフラグ：演算結果がリザルト (R) で指定された範囲 (ワード、ロング) を超える数値のとき。演算結果には各サイズの限界値が設定されます。

ワード長 正のオーバーフロー /7FFF


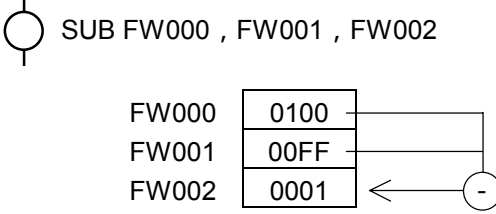
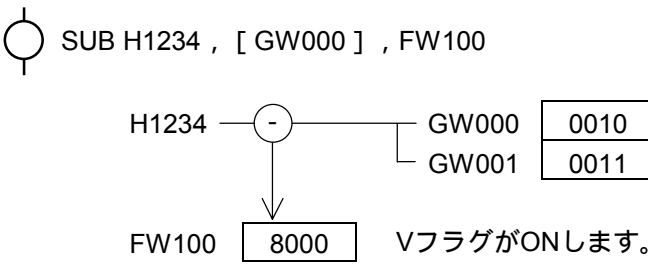
負のオーバーフロー /8000


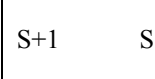
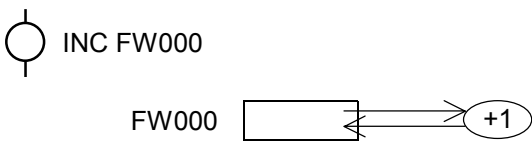
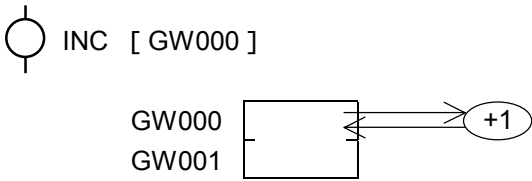
ロング長 正のオーバーフロー /7FFFFFFF

負のオーバーフロー /80000000


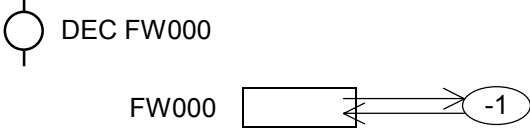
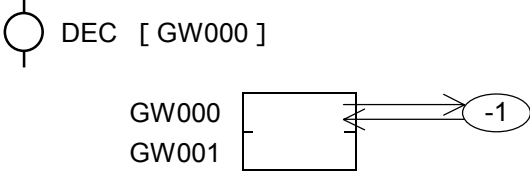
ADD	加算																																								
機能説明	ソースとデスティネーションの内容を加算してリザルトに格納します。																																								
パラメータと 処理内容	 ADD S, D, R <div style="float: right; border: 1px solid black; padding: 5px; margin-left: 20px;"> S+D R </div> <p>S : ソース D : デスティネーション R : リザルト</p>																																								
フラグの設定	E、Vが変化、他はOFF。																																								
処理時間	0.41ミリ秒																																								
備考																																									
使用例	 ADD FW000, FW001, FW002 <div style="margin-left: 40px;"> <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: 40px;"> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>H1234</td><td>+</td></tr> </table>  <p style="margin-left: 100px;">VフラグがONします。</p> </div>	FW000	0001	FW001	00FF	FW002	0100	H1234	+																																
FW000	0001																																								
FW001	00FF																																								
FW002	0100																																								
H1234	+																																								
有効 パラメータ	<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 エラー																																									


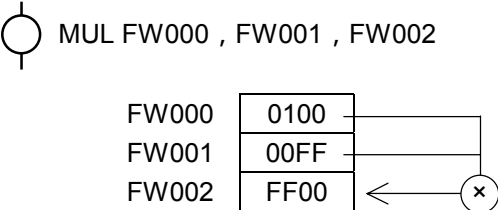
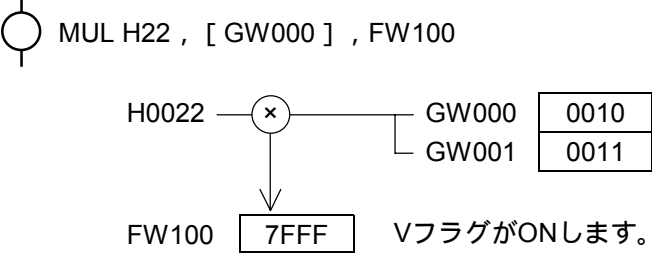
5 応用命令

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。																																										
パラメータと 処理内容	 SUB S, D, R	S-D	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、Vが変化、他はOFF。																																										
処理時間	0.41ミリ秒																																										
備考																																											
使用例	 																																										
有効 パラメータ	<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 エラー																																											



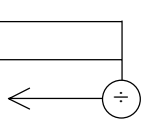

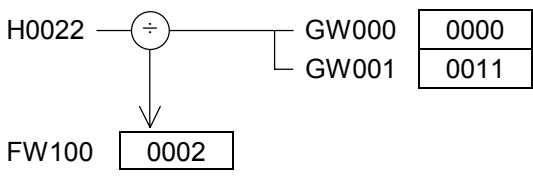
INC	+1 (インクリメント)																						
機能説明	ソースの内容に1を加算します。																						
パラメータと 処理内容	 INC S																						
	S : ソース																						
フラグの設定	E、Vが変化、他はOFF。																						
処理時間	0.29ミリ秒																						
備考																							
使用例	  GW000、GW001をロング変数と見なして インクリメントします。																						
有効 パラメータ	<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 エラー																							



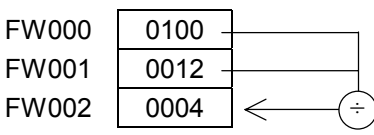

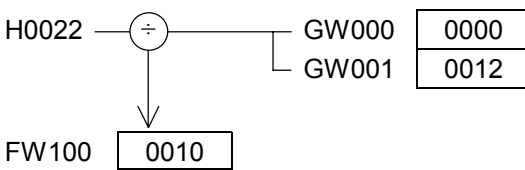
5 応用命令

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。																						
パラメータと 処理内容	 DEC S	S-1	S																				
	S : ソース																						
フラグの設定	E、Vが変化、他はOFF。																						
処理時間	0.29ミリ秒																						
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	<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 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。																						
パラメータと 処理内容	 MUL S, D, R	S×D	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、Vが変化、他はOFF。																						
処理時間	0.49ミリ秒 (S、Dともワード長) 1.35ミリ秒 (S、Dどちらかがワード長)																						
備考																							
使用例	 																						
有効 パラメータ	<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	×		
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	×																						
はアドレス 値が奇数時は パラメータ エラー	<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	×		
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	×																						



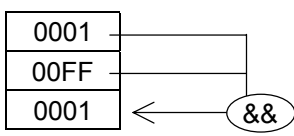

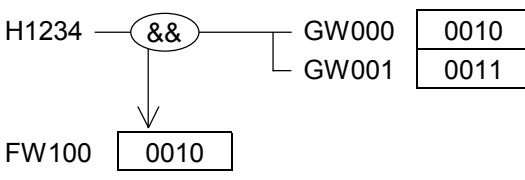
5 応用命令

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。																																								
パラメータと 処理内容	 DIV S, D, R <div style="float: right; margin-left: 20px;"> $S \div D \quad R$ </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。																																								
処理時間	0.49ミリ秒 (S、Dともワード長) 1.35ミリ秒 (S、Dどちらかがロング長)																																								
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 FW000, FW001, FW002 <div style="margin-left: 40px;"> <table border="1" style="display: inline-table; vertical-align: middle;"> <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 H22, [GW000], FW100 <div style="margin-left: 40px;">  </div>	FW000	0100	FW001	0010	FW002	0010																																		
FW000	0100																																								
FW001	0010																																								
FW002	0010																																								
有効 パラメータ	<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 エラー																																									



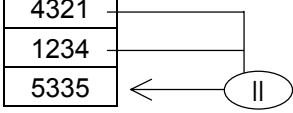

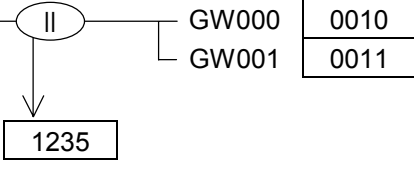
MOD	剰余																						
機能説明	ソースをデスティネーションの内容で除算して剰余をリザルトに格納します。																						
パラメータと 処理内容	 MOD S, D, R	$S \div D$ の剰余 R																					
	S : ソース D : デスティネーション R : リザルト																						
フラグの設定	E、Vが変化、他はOFF。																						
処理時間	0.50ミリ秒 (S、Dともワード長) 1.40ミリ秒 (S、Dどちらかがロング長)																						
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   MOD H22, [GW000], FW100 																						
有効 パラメータ	<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	×		
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	×																						
はアドレス 値が奇数時は パラメータ エラー	<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	×		
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	×																						



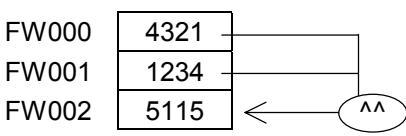

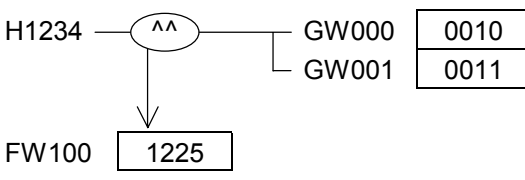
5 応用命令

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。																																										
パラメータと 処理内容	 SCL S, D1, D2, R	$S \times D1 \div D2 \quad R$																																									
	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																																										
フラグの設定	E、Vが変化、他はOFF。																																										
処理時間	1.38ミリ秒																																										
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する。 D2=0のとき、EフラグをONして何もしない。オーバーフロー時、R=0																																										
使用例	 SCL FW000, FW001, FW002, FW003 <table border="1" style="margin-left: 20px;"> <tr><td>FW000</td><td>3320</td><td rowspan="4">←</td><td rowspan="4">/3320×10÷66</td></tr> <tr><td>FW001</td><td>0010</td></tr> <tr><td>FW002</td><td>0066</td></tr> <tr><td>FW003</td><td>0805</td></tr> </table>  SCL GW000, GW001, H1110, FW100 <table border="1" style="margin-left: 20px;"> <tr><td>GW000</td><td>2222</td><td rowspan="2">←</td><td rowspan="2">/2222×12÷1110</td></tr> <tr><td>GW001</td><td>0012</td></tr> <tr><td>FW100</td><td>0024</td><td>←</td><td></td></tr> </table>			FW000	3320	←	/3320×10÷66	FW001	0010	FW002	0066	FW003	0805	GW000	2222	←	/2222×12÷1110	GW001	0012	FW100	0024	←																					
FW000	3320	←	/3320×10÷66																																								
FW001	0010																																										
FW002	0066																																										
FW003	0805																																										
GW000	2222	←	/2222×12÷1110																																								
GW001	0012																																										
FW100	0024	←																																									
有効 パラメータ	<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></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, 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 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。																																								
パラメータと 処理内容	 AND S, D, R <div style="float: right; border: 1px solid black; padding: 2px;">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。																																								
処理時間	0.36ミリ秒																																								
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 <div style="margin-top: 10px;"> <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>0001</td></tr> </table>  </div>  AND H1234, [GW000], FW100 <div style="margin-top: 10px;"> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>H1234</td><td>&&</td></tr> </table>  </div>	FW000	0001	FW001	00FF	FW002	0001	H1234	&&																																
FW000	0001																																								
FW001	00FF																																								
FW002	0001																																								
H1234	&&																																								
有効 パラメータ	<table border="1" style="display: inline-table; vertical-align: middle;"> <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: 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, 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 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。																																								
パラメータと 処理内容	 OR S, D, R <div style="float: right; border: 1px solid black; padding: 5px; 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。																																								
処理時間	0.36ミリ秒																																								
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; margin-right: 20px;"> <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;"> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>H1234</td><td>1234</td></tr> </table>  </div>	FW000	4321	FW001	1234	FW002	5335	H1234	1234																																
FW000	4321																																								
FW001	1234																																								
FW002	5335																																								
H1234	1234																																								
有効 パラメータ	<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 エラー																																									


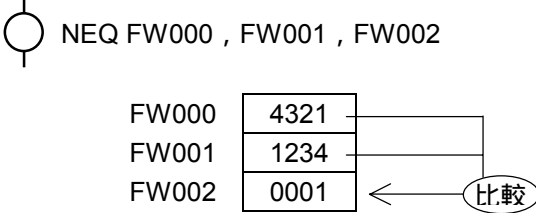
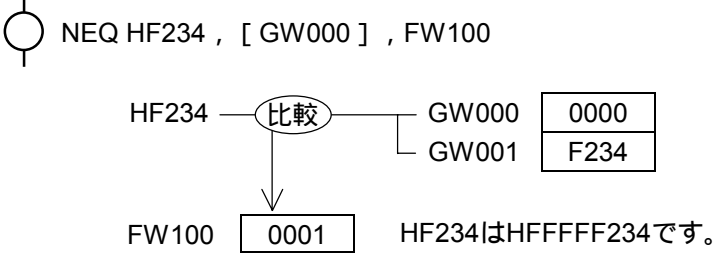
EOR	排他的論理和																																											
機能説明	ソースとデスティネーションの内容の排他的論理和をリザルトに格納します。																																											
パラメータと 処理内容	 EOR S, D, R	$S \wedge D \quad R$																																										
	S : ソース D : デスティネーション R : リザルト																																											
フラグの設定	Eが変化、他はOFF。																																											
処理時間	0.36ミリ秒																																											
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   EOR H1234, [GW000], FW100 																																											
有効 パラメータ	<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 エラー																																												


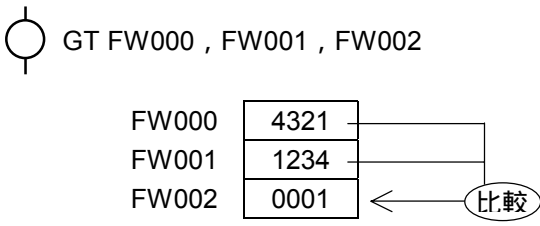
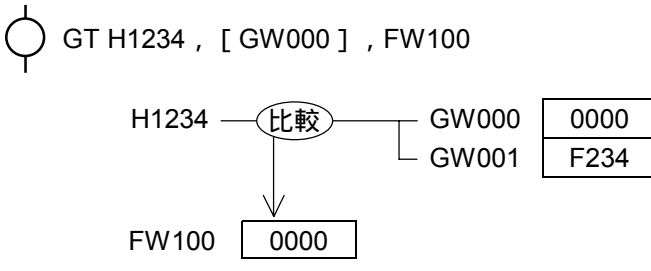
5 応用命令

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。																																								
パラメータと 処理内容	 NOT S, R <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid black; padding: 5px;">S (ビット反転)</div> <div style="border: 1px solid black; padding: 5px;">R</div> </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。																																								
処理時間	0.36ミリ秒																																								
備考																																									
使用例	 NOT FW000, FW002 <div style="margin-top: 10px;"> <table style="border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px;">FW000</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">4321</td> <td style="border: none; padding: 0 5px;">←</td> <td style="border: none;"></td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">FW002</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">BCDE</td> <td style="border: none; padding: 0 5px;">←</td> <td style="border: 1px solid black; border-radius: 50%; padding: 2px; text-align: center;">NOT</td> </tr> </table> </div>  NOT [GW000], FW100 <div style="margin-top: 10px;"> <table style="border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px;">GW000</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">0010</td> <td style="border: none; padding: 0 5px;">←</td> <td style="border: none;"></td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">GW001</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">0011</td> <td style="border: none; padding: 0 5px;">←</td> <td style="border: none;"></td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">GW100</td> <td style="border: 1px solid black; padding: 2px; text-align: center;">FFEE</td> <td style="border: none; padding: 0 5px;">←</td> <td style="border: 1px solid black; border-radius: 50%; padding: 2px; text-align: center;">NOT</td> </tr> </table> </div>	FW000	4321	←		FW002	BCDE	←	NOT	GW000	0010	←		GW001	0011	←		GW100	FFEE	←	NOT																				
FW000	4321	←																																							
FW002	BCDE	←	NOT																																						
GW000	0010	←																																							
GW001	0011	←																																							
GW100	FFEE	←	NOT																																						
有効 パラメータ	<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 style="text-align: center;">×</td> <td></td> <td></td> </tr> <tr> <td>インダイレクト ロング長</td> <td style="text-align: center;">×</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 style="text-align: center;">×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td></td> <td></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></td> <td></td> </tr> <tr> <td>インダイレクト ロング長</td> <td style="text-align: center;">×</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 エラー																																									


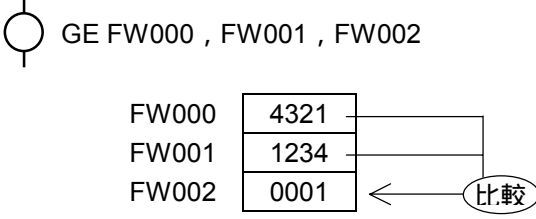
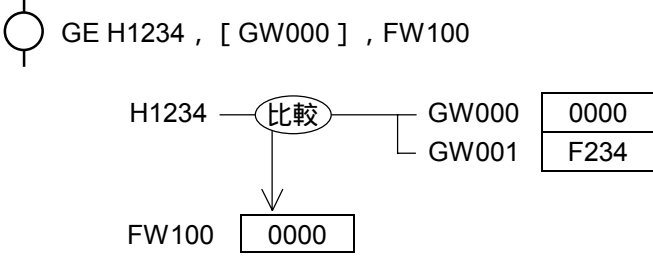
EQU	等しいか比較																																										
機能説明	ソースとデスティネーションの内容を比較し、等しければ1、それ以外は0をリザルトに格納します。																																										
パラメータと処理内容	 EQU S, D, R S : ソース D : デスティネーション R : リザルト	<table border="1"> <tr> <td>S = D</td> <td>1</td> <td>R</td> </tr> <tr> <td>S ≠ D</td> <td>0</td> <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。																																										
処理時間	0.40ミリ秒																																										
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	 EQU FW000, FW001, FW002 <table border="1"> <tr> <td>FW000</td> <td>4321</td> <td rowspan="3">← 比較</td> </tr> <tr> <td>FW001</td> <td>1234</td> </tr> <tr> <td>FW002</td> <td>0000</td> </tr> </table>  EQU HF234, [GW000], FW100 <table border="1"> <tr> <td>HF234</td> <td rowspan="2">← 比較</td> <td>GW000</td> <td>0000</td> </tr> <tr> <td></td> <td>GW001</td> <td>F234</td> </tr> </table> FW100 <table border="1"><tr><td>0000</td></tr></table> HF234はHFFFFFF234です。			FW000	4321	← 比較	FW001	1234	FW002	0000	HF234	← 比較	GW000	0000		GW001	F234	0000																									
FW000	4321	← 比較																																									
FW001	1234																																										
FW002	0000																																										
HF234	← 比較	GW000	0000																																								
		GW001	F234																																								
0000																																											
有効パラメータ	<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>直接ワード長</td> <td></td> <td></td> <td></td> </tr> <tr> <td>直接ロング長</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>直接ワード長</td> <td></td> <td></td> <td>×</td> </tr> <tr> <td>直接ロング長</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	直接ワード長				直接ロング長				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	直接ワード長			×	直接ロング長			×	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																																								
直接ワード長																																											
直接ロング長																																											
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																																								
直接ワード長			×																																								
直接ロング長			×																																								
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																																											


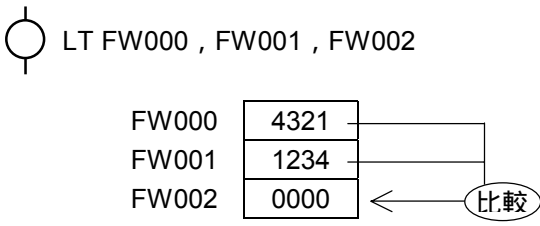
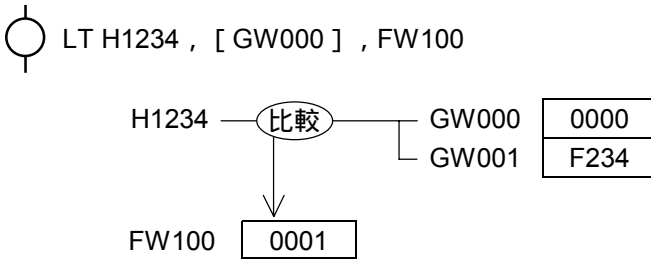
5 応用命令

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	<table border="1"> <tr> <td>S ≠ 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。																						
処理時間	0.40ミリ秒																						
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>NEQ FW000, FW001, FW002</p> <p>FW000: 4321 FW001: 1234 FW002: 0001</p> <p>比較結果: 0001</p>  <p>NEQ HF234, [GW000], FW100</p> <p>HF234: HFFFFFF234 GW000: 0000 GW001: F234 FW100: 0001</p> <p>比較結果: 0001</p> <p>HF234はHFFFFFF234です。</p>																						
有効パラメータ	<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	×		
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	×																						
はアドレス値が奇数時はパラメータエラー	<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	×		
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	×																						


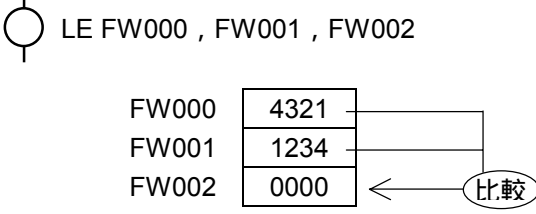
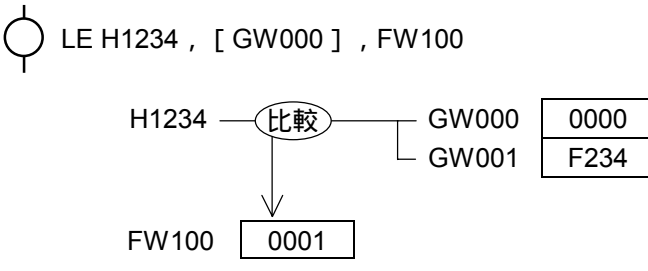
GT	より大きい比較																						
機能説明	ソースとデスティネーションの内容を比較し、ソースが大きければ1、それ以外は0をリザルトに格納します。																						
パラメータと処理内容	 GT S, D, R S : ソース D : デスティネーション R : リザルト	<table border="1"> <tr> <td>S > D</td> <td>1</td> <td>R</td> </tr> <tr> <td>S ≤ D</td> <td>0</td> <td>R</td> </tr> </table>		S > D	1	R	S ≤ D	0	R														
S > D	1	R																					
S ≤ D	0	R																					
フラグの設定	Eが変化、他はOFF。																						
処理時間	0.40ミリ秒																						
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	<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>直接ワード長</td> <td></td> <td></td> <td>×</td> </tr> <tr> <td>直接ロング長</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	直接ワード長			×	直接ロング長			×	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																				
直接ワード長			×																				
直接ロング長			×																				
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エラー	<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>直接ワード長</td> <td></td> <td></td> <td>×</td> </tr> <tr> <td>直接ロング長</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	直接ワード長			×	直接ロング長			×	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																				
直接ワード長			×																				
直接ロング長			×																				
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

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。																																											
パラメータと処理内容	 GE S, D, R S : ソース D : デスティネーション R : リザルト	<table border="1"> <tr> <td>S</td> <td>D</td> <td>1</td> <td>R</td> </tr> <tr> <td>S < D</td> <td>0</td> <td></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。																																											
処理時間	0.40ミリ秒																																											
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	<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>直接ワード長</td> <td></td> <td></td> <td></td> <td>直接ワード長</td> <td></td> <td></td> <td>×</td> </tr> <tr> <td>直接ロング長</td> <td></td> <td></td> <td></td> <td>直接ロング長</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	直接ワード長				直接ワード長			×	直接ロング長				直接ロング長			×	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																																					
直接ワード長				直接ワード長			×																																					
直接ロング長				直接ロング長			×																																					
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																																												


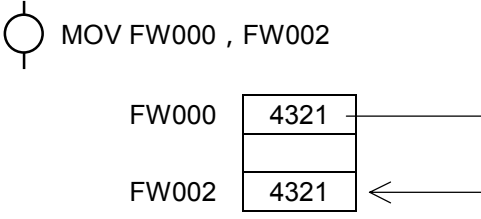
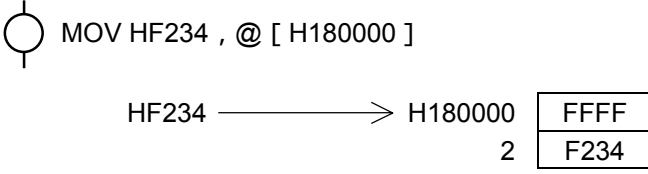
LT	より小さいか比較																						
機能説明	ソースとデスティネーションの内容を比較し、ソースが小さければ1、それ以外は0をリザルトに格納します。																						
パラメータと処理内容	 LT S, D, R S : ソース D : デスティネーション R : リザルト	<table border="1"> <tr> <td>S < D</td> <td>1</td> <td>R</td> </tr> <tr> <td>S ≥ D</td> <td>0</td> <td>R</td> </tr> </table>		S < D	1	R	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。																						
処理時間	0.40ミリ秒																						
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	<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>直接ワード長</td> <td></td> <td></td> <td>×</td> </tr> <tr> <td>直接ロング長</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	直接ワード長			×	直接ロング長			×	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																				
直接ワード長			×																				
直接ロング長			×																				
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エラー	<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>直接ワード長</td> <td></td> <td></td> <td>×</td> </tr> <tr> <td>直接ロング長</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	直接ワード長			×	直接ロング長			×	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																				
直接ワード長			×																				
直接ロング長			×																				
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

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。																						
パラメータと処理内容	 LE S, D, R S : ソース D : デスティネーション R : リザルト	<table border="1"> <tr> <td>S</td> <td>D</td> <td>1</td> <td>R</td> </tr> <tr> <td>S > D</td> <td>0</td> <td></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。																						
処理時間	0.40ミリ秒																						
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	<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	×		
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	×																						
はアドレス 値が奇数時は パラメータ エラー	<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	×		
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	×																						




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。																				
パラメータと 処理内容	 TST S S : ソース <div style="border: 1px solid black; padding: 5px; margin-left: 200px;"> S > 0 : P=1、Z=0、N=0 S = 0 : P=0、Z=1、N=0 S < 0 : P=0、Z=0、N=1 </div>																				
フラグの設定	E、P、Z、Nが変化、他はOFF。																				
処理時間	0.17ミリ秒																				
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>4321</td></tr></table> FW002 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>2000</td></tr></table> ← テスト </div>  TST [GW000] <div style="margin-left: 100px;"> GW000 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>FFFF</td></tr></table> GW001 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>F234</td></tr></table> GW020 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1000</td></tr></table> ← テスト </div>	4321	2000	FFFF	F234	1000															
4321																					
2000																					
FFFF																					
F234																					
1000																					
有効 パラメータ	<table border="1" style="width: 100%; border-collapse: collapse;"> <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>直接 ワード長</td> <td></td> <td></td> <td></td> </tr> <tr> <td>直接 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 style="text-align: center;">×</td> <td></td> <td></td> </tr> <tr> <td>インダイレクト ロング長</td> <td style="text-align: center;">×</td> <td></td> <td></td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	直接 ワード長				直接 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																		
直接 ワード長																					
直接 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 エラー																					


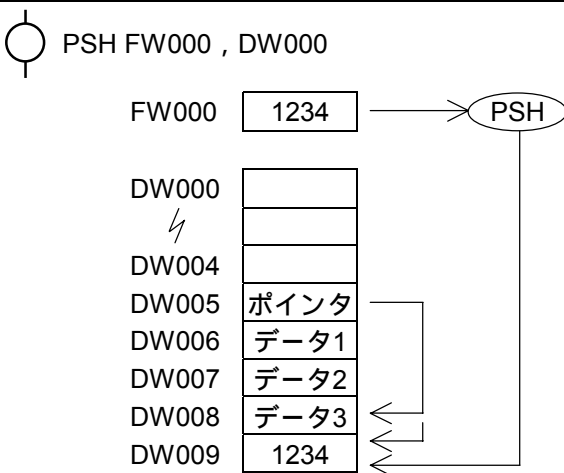
5 応用命令

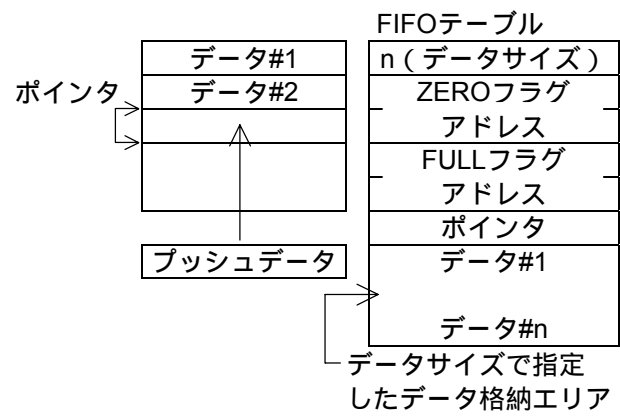
MOV	転送																																										
機能説明	ソースの内容をデスティネーションに転送します。																																										
パラメータと 処理内容	 MOV S , D	S	D																																								
	S : ソース D : デスティネーション																																										
フラグの設定	Eが変化、他はOFF。																																										
処理時間	0.28ミリ秒																																										
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	  HF234はHFFFFFF234です。																																										
有効 パラメータ	<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 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	<table border="1"> <tr> <td>S1</td> <td>D1</td> </tr> <tr> <td>⚡</td> <td></td> </tr> <tr> <td>Sn</td> <td>Dn</td> </tr> </table>		S1	D1	⚡		Sn	Dn																																		
S1	D1																																										
⚡																																											
Sn	Dn																																										
フラグの設定	E、Vが変化、他はOFF。																																										
処理時間	0.39+0.02×nミリ秒																																										
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 <table border="1"> <tr> <td>FW000</td> <td>4321</td> <td rowspan="2">→</td> </tr> <tr> <td>FW002</td> <td>4321</td> </tr> </table>  MOM FW000 , 2 , @ [H180000] <table border="1"> <tr> <td>FW000</td> <td>F234</td> <td rowspan="4">→</td> <td>H180000</td> <td>FFFF</td> </tr> <tr> <td>FW001</td> <td>0001</td> <td>2</td> <td>F234</td> </tr> <tr> <td>FW002</td> <td>0000</td> <td>4</td> <td>0000</td> </tr> <tr> <td>FW003</td> <td>FFFF</td> <td>6</td> <td>0001</td> </tr> </table> HF234はHFFFFFF234、H0001はH00000001です。			FW000	4321	→	FW002	4321	FW000	F234	→	H180000	FFFF	FW001	0001	2	F234	FW002	0000	4	0000	FW003	FFFF	6	0001																		
FW000	4321	→																																									
FW002	4321																																										
FW000	F234	→	H180000	FFFF																																							
FW001	0001		2	F234																																							
FW002	0000		4	0000																																							
FW003	FFFF		6	0001																																							
有効 パラメータ	<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> <table border="1"> <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 エラー																																											


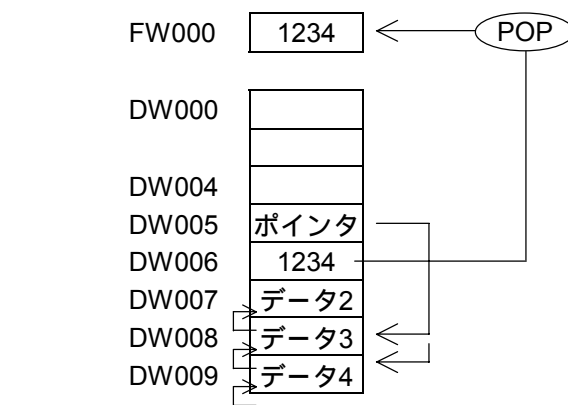
5 応用命令


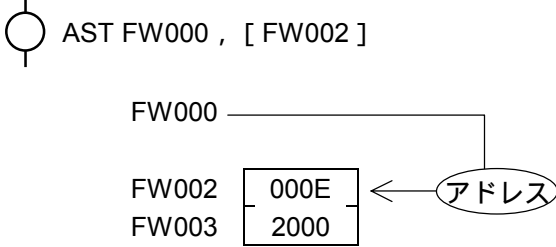
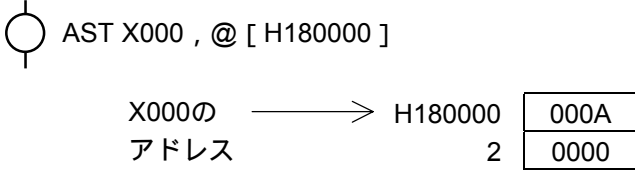
EXC	交換																						
機能説明	ソースの内容とデスティネーションを交換します。																						
パラメータと 処理内容	 EXC S, D	S	D																				
	S : ソース D : デスティネーション																						
フラグの設定	Eが変化、他はOFF。																						
処理時間	0.31ミリ秒																						
備考	転送するサイズが異なる場合、型変換して交換します。																						
使用例	 EXC FW000, FW002 <div style="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>1234</td></tr> <tr><td> </td></tr> </table> <div style="margin: 0 10px;">←</div> <div style="border: 1px solid black; width: 20px; height: 20px; margin: 0 5px;"></div> <div style="margin: 0 10px;">←</div> <div style="margin-right: 10px;">FW002</div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>4321</td></tr> <tr><td> </td></tr> </table> </div>  EXC @H170000, @[H180000] <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">H170000</div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>F234</td></tr> </table> <div style="margin: 0 10px;">↔</div> <div style="margin-right: 10px;">H180000</div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>0010</td></tr> <tr><td>0001</td></tr> </table> </div> <p style="margin-left: 20px;">交換後</p> <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;">H170000</div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>7FFF</td></tr> </table> <div style="margin: 0 10px;">↔</div> <div style="margin-right: 10px;">H180000</div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>FFFF</td></tr> <tr><td>F234</td></tr> </table> </div>			1234		4321		F234	0010	0001	7FFF	FFFF	F234										
1234																							
4321																							
F234																							
0010																							
0001																							
7FFF																							
FFFF																							
F234																							
有効 パラメータ	<table border="1" style="border-collapse: collapse; text-align: center;"> <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	×		
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	×																						
はアドレス 値が奇数時は パラメータ エラー																							

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のみです。																																								
パラメータと処理内容	 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。																																								
処理時間	0.31ミリ秒																																								
備考	n 0、n>256のとき未処理。 ポインタ<0、またはデータサイズ<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。																																								
使用例																																									
有効パラメータ	<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	×																																								






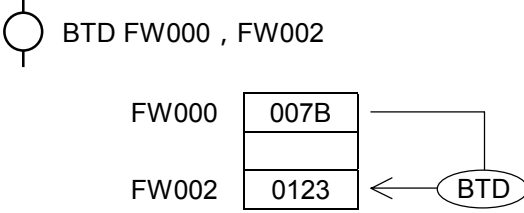
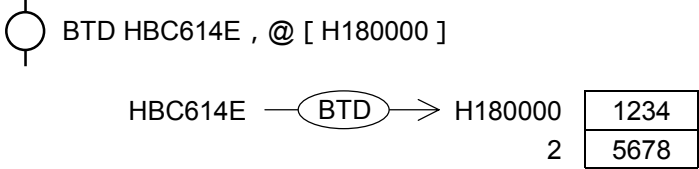
5 応用命令

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のみです。																																								
パラメータと 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。																																								
処理時間	0.32+0.01×nミリ秒																																								
備考	<p>n = 0、n>256のとき未処理。 ポインタ<0、またはデータサイズ<ポインタのとき未処理。ポインタ=0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																																									
有効 パラメータ	<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	×																																								



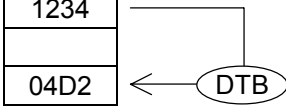

AST	アドレスセット																																										
機能説明	ソースのアドレスデータをデスティネーションに転送します。 PI/Oのみ有効です。																																										
パラメータと 処理内容	 AST S, D	<table border="1" style="width: 100%;"> <tr> <td style="width: 50%;">Sのアドレス</td> <td style="width: 50%;">D</td> </tr> </table>		Sのアドレス	D																																						
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。																																										
処理時間	0.25ミリ秒																																										
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	<table border="1" style="width: 100%;"> <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> <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> <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	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 エラー																																											


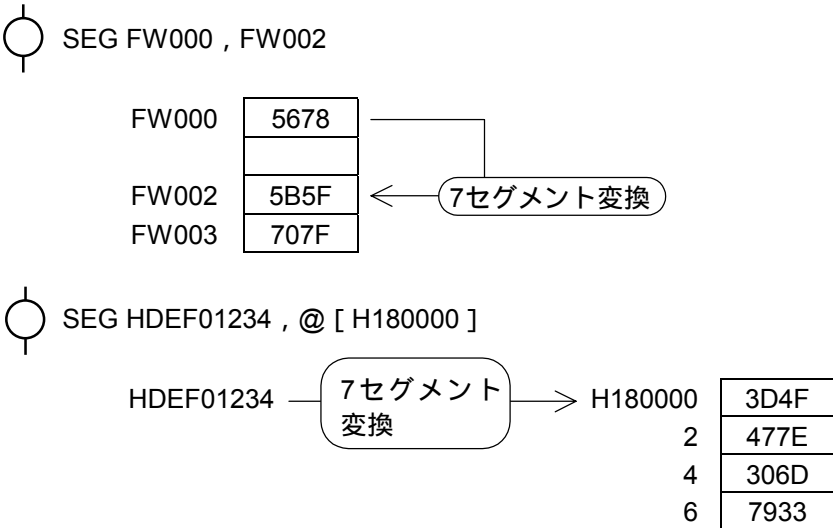
5 応用命令

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。																																								
パラメータと処理内容	<p>  SCH S, D, m, R S : ソース D : デスティネーション m : サーチステップ数 R : リザルト </p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>S</p> <p>データ</p> </div> <div style="text-align: center;"> <p>R</p> <p>n</p> </div> </div> <p style="text-align: center;">サーチテーブル</p> <div style="display: flex; align-items: center;"> <div style="text-align: center;"> <p>D(0)</p> <p>(1)</p> <p>(n)</p> <p>(m-1)</p> </div> <div style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px; margin: 0 5px;"> <p>データ</p> </div> <div style="margin-left: 10px;"> <p>— サーチ範囲</p> </div> </div>																																								
フラグの設定	Eが変化、他はOFF。																																								
処理時間	0.45+0.01×nミリ秒																																								
備考	<p>m 0、m>256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; flex-direction: column; align-items: center;"> <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="margin-right: 10px;">DW000</div> <table border="1" style="border-collapse: collapse;"> <tr><td style="width: 40px; text-align: center;">1234</td></tr> </table> <div style="margin: 0 10px;">→</div> <div style="border: 1px solid black; border-radius: 50%; padding: 2px 5px;">サーチ</div> </div> <div style="display: flex; align-items: center;"> <div style="margin-right: 10px;">FW000</div> <table border="1" style="border-collapse: collapse;"> <tr><td style="width: 40px; text-align: center;">0000</td></tr> <tr><td style="width: 40px; text-align: center;">1234</td></tr> <tr><td style="width: 40px; text-align: center;">0000</td></tr> <tr><td style="width: 40px; text-align: center;">1234</td></tr> <tr><td style="width: 40px; text-align: center;">0000</td></tr> <tr><td style="width: 40px; text-align: center;">0001</td></tr> </table> <div style="margin: 0 10px;">←</div> <div style="margin-right: 10px;">1番目</div> <div style="border-left: 1px solid black; border-right: 1px solid black; padding: 0 5px; margin: 0 5px;"> <p>データ</p> </div> </div> </div>	1234	0000	1234	0000	1234	0000	0001																																	
1234																																									
0000																																									
1234																																									
0000																																									
1234																																									
0000																																									
0001																																									
有効パラメータ	<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></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 style="text-align: center;">×</td> <td></td> <td></td> </tr> <tr> <td>インダイレクトロング長</td> <td style="text-align: center;">×</td> <td></td> <td></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></td> <td></td> <td style="text-align: center;">×</td> </tr> <tr> <td>ダイレクトロング長</td> <td></td> <td></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></td> <td></td> </tr> <tr> <td>インダイレクトロング長</td> <td style="text-align: center;">×</td> <td></td> <td></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エラー																																									




BT D	バイナリ BCD変換																																									
機能説明	ソースの内容をバイナリからBCDに変換し、リザルトに格納します。																																									
パラメータと 処理内容	 BT D S, R S : ソース R : リザルト	S (バイナリ) R (BCD)																																								
フラグの設定	E, Vが変化、他はOFF。																																									
処理時間	1.03ミリ秒																																									
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	 <p>BT D FW000, FW002</p> <p>FW000: 007B FW002: 0123</p> <p>BT D symbol with arrow pointing to FW002</p>  <p>BT D HBC614E, @ [H180000]</p> <p>HBC614E → BT D symbol → H180000</p> <p>2</p> <table border="1" data-bbox="957 1220 1077 1310"> <tr><td>1234</td></tr> <tr><td>5678</td></tr> </table>		1234	5678																																						
1234																																										
5678																																										
有効 パラメータ	<table border="1" data-bbox="375 1400 893 1780"> <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 1400 1460 1780"> <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 エラー																																										




5 応用命令

DTB	BCD バイナリ変換																																									
機能説明	ソースの内容をBCDからバイナリに変換し、リザルトに格納します。																																									
パラメータと 処理内容	 DTB S, R S : ソース R : リザルト	S (BCD) R (バイナリ)																																								
フラグの設定	E, Vが変化、他はOFF。																																									
処理時間	0.46ミリ秒																																									
備考	Sの中で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> <tr><td></td><td></td><td>2</td></tr> </table> <table border="1" style="display: inline-table;"> <tr><td>05F5</td></tr> <tr><td>E0FF</td></tr> </table>		FW000	1234	FW002	04D2	H99999999	DTB	H180000			2	05F5	E0FF																												
FW000	1234																																									
FW002	04D2																																									
H99999999	DTB	H180000																																								
		2																																								
05F5																																										
E0FF																																										
有効 パラメータ	<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 エラー																																										


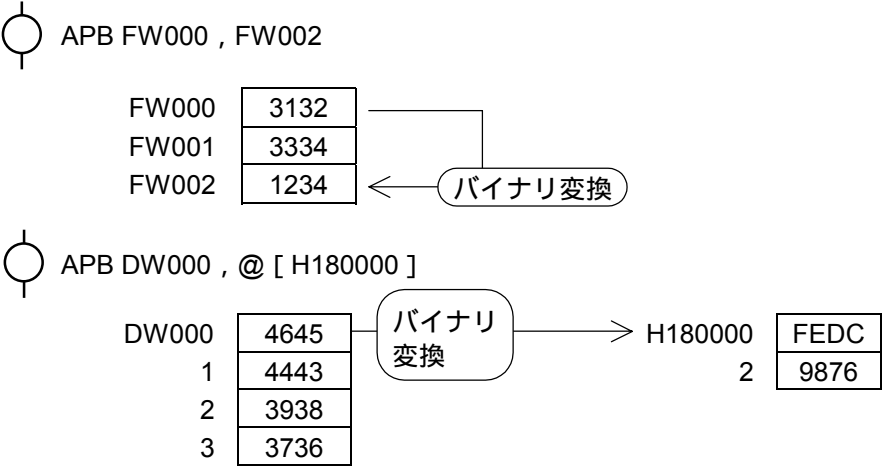


SEG	バイナリ 7セグメント変換																																									
機能説明	ソースの内容をバイナリから7セグメントデータに変換し、リザルトに格納します。																																									
パラメータと 処理内容	 SEG S, R	S (バイナリ) R (7セグメントデータ)																																								
	S : ソース R : リザルト																																									
フラグの設定	Eが変化、他はOFF。																																									
処理時間	0.35ミリ秒																																									
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" data-bbox="411 1451 1428 1527"> <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	<table border="1" data-bbox="375 1624 893 1993"> <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 1624 1460 1993"> <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 エラー																																										


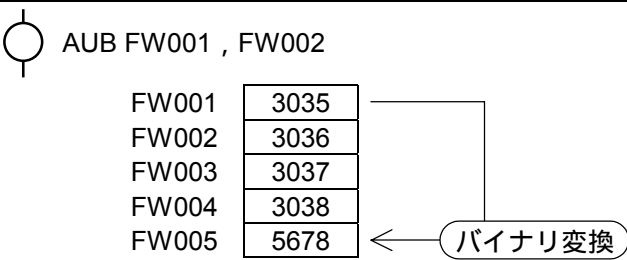
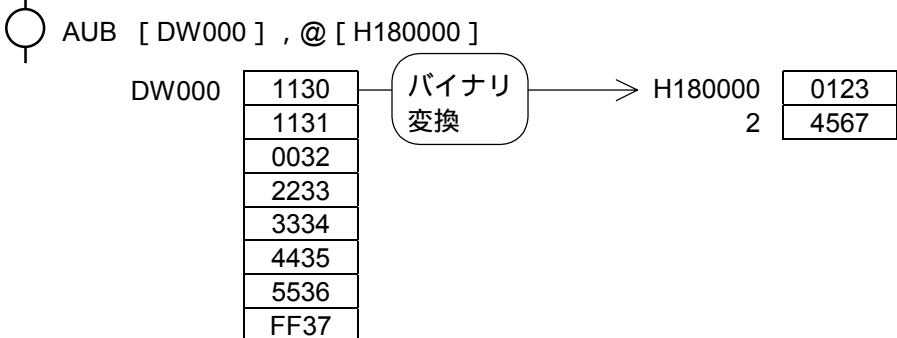
5 応用命令

ASP	バイナリ 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。																																																											
パラメータと処理内容	 ASP S, R	S (バイナリ) R (アスキー、パックモード)																																																										
	S : ソース R : リザルト																																																											
フラグの設定	Eが変化、他はOFF。																																																											
処理時間	0.47ミリ秒																																																											
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="margin-left: 40px;"> <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="margin-left: 40px;"> <tr><td>HDEF01234</td><td>アスキー変換</td><td>H180000</td><td>4445</td></tr> <tr><td></td><td></td><td>2</td><td>4630</td></tr> <tr><td></td><td></td><td>4</td><td>3132</td></tr> <tr><td></td><td></td><td>6</td><td>3334</td></tr> </table> [アスキー、バイナリ対応表] <table border="1" style="margin-left: 40px;"> <tr><th>バイナリ</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	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	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	<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エラー																																																												


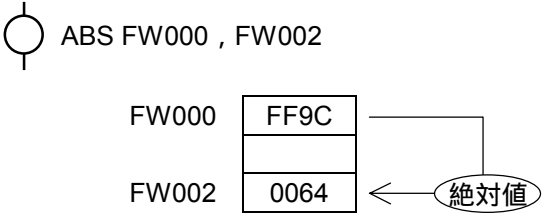
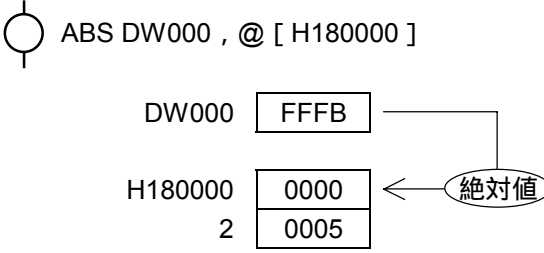
ASU	バイナリ アスキー変換アンパックモード																																																																													
機能説明	ソースの内容をバイナリからアスキーデータに変換し、アンパックモードでリザルトに格納します。																																																																													
パラメータと処理内容	 ASU S, R S : ソース R : リザルト	S (バイナリ) R (アスキー、アンパックモード)																																																																												
フラグの設定	Eが変化、他はOFF。																																																																													
処理時間	0.49ミリ秒																																																																													
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="502 896 734 1075"> <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] <table border="1" data-bbox="502 1153 1212 1433"> <tr><td>HDEF01234</td><td></td><td>H180000</td><td>3044</td></tr> <tr><td></td><td></td><td>2</td><td>3045</td></tr> <tr><td></td><td></td><td>4</td><td>3046</td></tr> <tr><td></td><td></td><td>6</td><td>3030</td></tr> <tr><td></td><td></td><td>8</td><td>3031</td></tr> <tr><td></td><td></td><td>A</td><td>3032</td></tr> <tr><td></td><td></td><td>C</td><td>3033</td></tr> <tr><td></td><td></td><td>E</td><td>3034</td></tr> </table> <p>[アスキー、バイナリ対応表]</p> <table border="1" data-bbox="391 1489 1444 1579"> <tr><td>バイナリ</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>		FW001	5678	FW002	3035	FW003	3036	FW004	3037	FW005	3038	HDEF01234		H180000	3044			2	3045			4	3046			6	3030			8	3031			A	3032			C	3033			E	3034	バイナリ	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																																																																													
HDEF01234		H180000	3044																																																																											
		2	3045																																																																											
		4	3046																																																																											
		6	3030																																																																											
		8	3031																																																																											
		A	3032																																																																											
		C	3033																																																																											
		E	3034																																																																											
バイナリ	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	<table border="1" data-bbox="375 1635 893 2004"> <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 1635 1460 2004"> <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エラー																																																																														


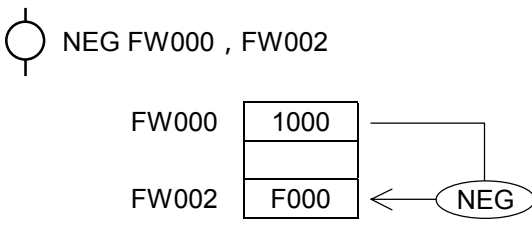
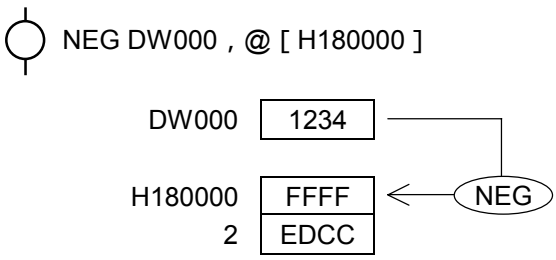
5 応用命令

APB	アスキー バイナリ変換パックモード																																									
機能説明	ソースの内容をアスキーデータ（パックモード）からバイナリに変換し、リザルトに格納します。																																									
パラメータと処理内容	 APB S, R S : ソース R : リザルト	S (アスキー、パックモード) R (バイナリ)																																								
フラグの設定	Eが変化、他はOFF。																																									
処理時間	0.57ミリ秒																																									
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>  APB FW000, FW002  APB DW000, @ [H180000] </p> <p>[アスキー、バイナリ対応表]</p> <table border="1" data-bbox="352 1458 1394 1536"> <thead> <tr> <th>バイナリ</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> <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> </tbody> </table>		バイナリ	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	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	<table border="1" data-bbox="331 1626 847 1998"> <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>直接ワード長</td> <td>×</td> <td></td> <td>×</td> </tr> <tr> <td>直接ロング長</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	直接ワード長	×		×	直接ロング長	×		×	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 1626 1414 1998"> <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>直接ワード長</td> <td></td> <td></td> <td>×</td> </tr> <tr> <td>直接ロング長</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	直接ワード長			×	直接ロング長			×	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																																							
直接ワード長	×		×																																							
直接ロング長	×		×																																							
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																																							
直接ワード長			×																																							
直接ロング長			×																																							
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エラー																																										


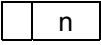
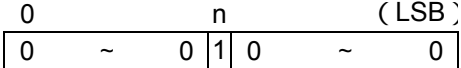
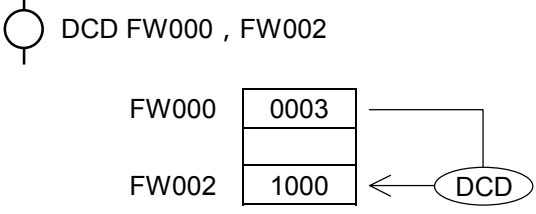
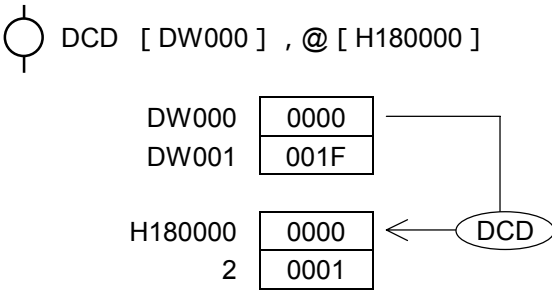
AUB	アスキー バイナリ変換アンパックモード																																																															
機能説明	ソースの内容をアスキーデータ（アンパックモード）からバイナリに変換し、リザルトに格納します。																																																															
パラメータと処理内容	 AUB S, R S : ソース R : リザルト	S (アスキー、アンパックモード) R (バイナリ)																																																														
フラグの設定	Eが変化、他はOFF。																																																															
処理時間	0.57ミリ秒																																																															
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	 <p>AUB FW001, FW002</p> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>FW001</td><td>3035</td></tr> <tr><td>FW002</td><td>3036</td></tr> <tr><td>FW003</td><td>3037</td></tr> <tr><td>FW004</td><td>3038</td></tr> <tr><td>FW005</td><td>5678</td></tr> </table> <p style="margin-left: 100px;">← バイナリ変換</p>  <p>AUB [DW000], @[H180000]</p> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>DW000</td><td>1130</td></tr> <tr><td></td><td>1131</td></tr> <tr><td></td><td>0032</td></tr> <tr><td></td><td>2233</td></tr> <tr><td></td><td>3334</td></tr> <tr><td></td><td>4435</td></tr> <tr><td></td><td>5536</td></tr> <tr><td></td><td>FF37</td></tr> </table> <p style="margin-left: 20px;">→ H180000</p> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0123</td></tr> <tr><td>2 4567</td></tr> </table> <p style="margin-left: 10px;">(バイナリ変換)</p> <p>[アスキー、バイナリ対応表]</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td>バイナリ</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>		FW001	3035	FW002	3036	FW003	3037	FW004	3038	FW005	5678	DW000	1130		1131		0032		2233		3334		4435		5536		FF37	0123	2 4567	バイナリ	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	3035																																																															
FW002	3036																																																															
FW003	3037																																																															
FW004	3038																																																															
FW005	5678																																																															
DW000	1130																																																															
	1131																																																															
	0032																																																															
	2233																																																															
	3334																																																															
	4435																																																															
	5536																																																															
	FF37																																																															
0123																																																																
2 4567																																																																
バイナリ	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	<table border="1" style="width: 100%; text-align: center;"> <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" style="width: 100%; text-align: center;"> <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																																																																


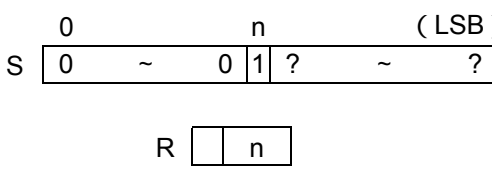
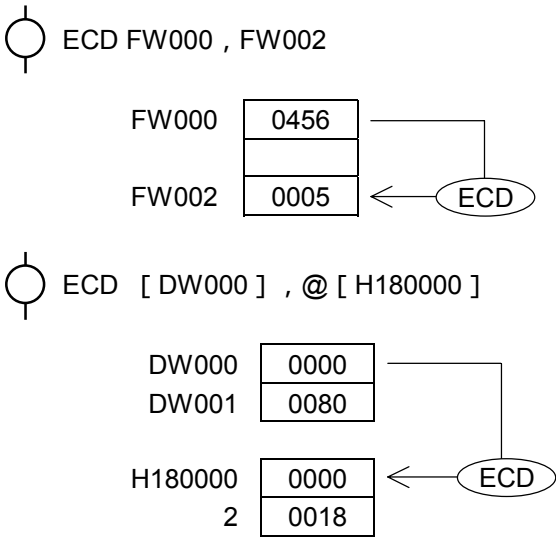


5 応用命令

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。																																											
パラメータと 処理内容	 ABS S, R S : ソース R : リザルト	S R																																										
フラグの設定	E, Vが変化、他はOFF。																																											
処理時間	0.32ミリ秒																																											
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	 <p>ABS FW000, FW002</p> <p>FW000: FF9C FW002: 0064 ← 絶対値</p>  <p>ABS DW000, @ [H180000]</p> <p>DW000: FFFB H180000: 0000 2: 0005 ← 絶対値</p>																																											
有効 パラメータ	<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 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。																																														
パラメータと 処理内容	 NEG S , R			-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。																																														
処理時間	0.32ミリ秒																																														
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	<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 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。																						
パラメータと 処理内容	 DCD S, R S : ソース R : リザルト	 S [n]	 R [0 ~ 0 1 0 ~ 0] (LSB)																				
フラグの設定	Eが変化、他はOFF。																						
処理時間	0.38ミリ秒																						
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 FW000, FW002</p> <p>FW000 [0003] FW002 [1000] ← DCD</p>  <p>DCD [DW000], @ [H180000]</p> <p>DW000 [0000] DW001 [001F] H180000 [0000] 2 [0001] ← DCD</p>																						
有効 パラメータ	<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>直接 ワード長</td> <td></td> <td></td> <td></td> </tr> <tr> <td>直接 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	直接 ワード長				直接 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																				
直接 ワード長																							
直接 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 エラー	<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>直接 ワード長</td> <td></td> <td></td> <td>×</td> </tr> <tr> <td>直接 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	直接 ワード長			×	直接 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																				
直接 ワード長			×																				
直接 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	×																						

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。																																										
パラメータと 処理内容	 ECD S, R S : ソース R : リザルト	 <p style="text-align: center;"> $S \quad \begin{array}{ c c c c c } \hline 0 & \sim & 0 & 1 & ? \sim ? \\ \hline \end{array}$ $R \quad \begin{array}{ c } \hline n \\ \hline \end{array}$ </p>																																									
フラグの設定	Eが変化、他はOFF。																																										
処理時間	0.38+0.01×nミリ秒																																										
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のみ有効。																																										
使用例	 <p>  ECD FW000, FW002 FW000 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0456</td></tr><tr><td> </td></tr></table> ← FW002 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0005</td></tr></table> ← (ECD) </p> <p>  ECD [DW000], @ [H180000] DW000 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0000</td></tr><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) </p>			0456		0005	0000	0080	0000	0018																																	
0456																																											
0005																																											
0000																																											
0080																																											
0000																																											
0018																																											
有効 パラメータ	<table border="1" style="width: 100%; border-collapse: collapse;"> <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> <th style="width: 15%;">R</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>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	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	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 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。																																										
処理時間	0.37ミリ秒																																										
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" 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>0045</td></tr> </table> <p>○ LSR [DW000], 2, @[H180000]</p> <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;"> <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	<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 エラー																																											


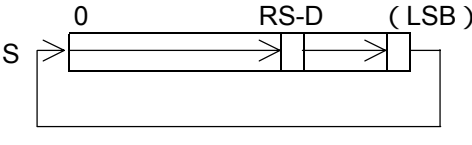
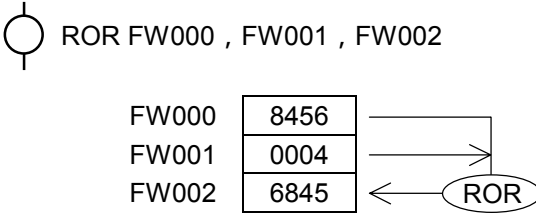
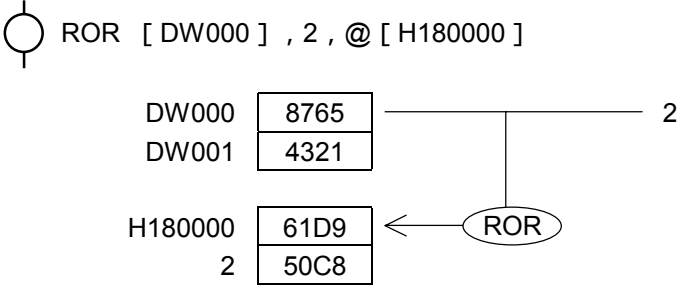
LSL	論理左シフト																																											
機能説明	ソースの内容をデスティネーションの内容で左へシフトし、結果をリザルトに格納します。																																											
パラメータと処理内容	\odot LSL S, D, R S : ソース R : リザルト D : デスティネーション																																											
フラグの設定	Eが変化、他はOFF。																																											
処理時間	0.37ミリ秒																																											
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	\odot LSL FW000, FW001, FW002 <table border="1" style="margin-left: 20px;"> <tr><td>FW000</td><td>0456</td><td rowspan="3" style="text-align: center;">→</td></tr> <tr><td>FW001</td><td>0004</td></tr> <tr><td>FW002</td><td>4560</td><td style="text-align: center;">← (LSL)</td></tr> </table> \odot LSL [DW000], 2, @ [H180000] <table border="1" style="margin-left: 20px;"> <tr><td>DW000</td><td>8765</td><td rowspan="2" style="text-align: center;">— 2</td></tr> <tr><td>DW001</td><td>4321</td></tr> <tr><td>H180000</td><td>1D95</td><td rowspan="2" style="text-align: center;">← (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	<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																																												


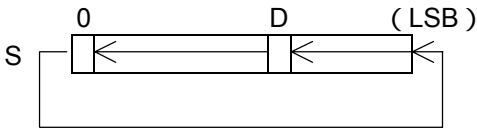
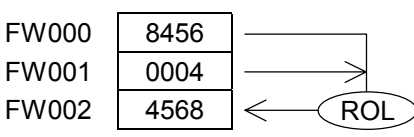
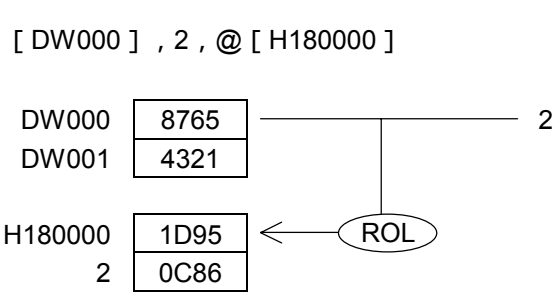
5 応用命令

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>	<p>RSはワード ロングによる (LSB) (15/31)</p>																																								
フラグの設定	E, Vが変化、他はOFF。																																									
処理時間	0.43ミリ秒																																									
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> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>FW000</td><td>8456</td></tr> <tr><td>FW001</td><td>0004</td></tr> <tr><td>FW002</td><td>F845</td></tr> </table> <p style="text-align: center;">ASR</p> <p>○ ASR [DW000], 2, @[H180000]</p> <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> <p style="text-align: center;">2</p> <table border="1" style="display: inline-table;"> <tr><td>H180000</td><td>E1D9</td></tr> <tr><td>2</td><td>0246</td></tr> </table> <p style="text-align: center;">ASR</p>			FW000	8456	FW001	0004	FW002	F845	DW000	8765	DW001	4321	H180000	E1D9	2	0246																									
FW000	8456																																									
FW001	0004																																									
FW002	F845																																									
DW000	8765																																									
DW001	4321																																									
H180000	E1D9																																									
2	0246																																									
有効パラメータ	<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>直接ワード長</td> <td></td> <td></td> <td></td> </tr> <tr> <td>直接ロング長</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	直接ワード長				直接ロング長				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>直接ワード長</td> <td></td> <td></td> <td>×</td> </tr> <tr> <td>直接ロング長</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	直接ワード長			×	直接ロング長			×	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																																							
直接ワード長																																										
直接ロング長																																										
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																																							
直接ワード長			×																																							
直接ロング長			×																																							
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。																																										
処理時間	0.41ミリ秒																																										
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>○ ASL 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></tr> </table> <p>← ASL</p> <p>○ ASL [DW000], 2, @[H180000]</p> <table border="1"> <tr><td>DW000</td><td>4765</td><td rowspan="2">→</td><td rowspan="2">2</td></tr> <tr><td>DW001</td><td>4321</td></tr> </table> <table border="1"> <tr><td>H180000</td><td>7FFF</td><td rowspan="2">←</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	→	2	DW001	4321	H180000	7FFF	←	2	FFFF																						
FW000	0456	→																																									
FW001	0004																																										
FW002	4560																																										
DW000	4765	→	2																																								
DW001	4321																																										
H180000	7FFF	←																																									
2	FFFF																																										
有効パラメータ	<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																																											


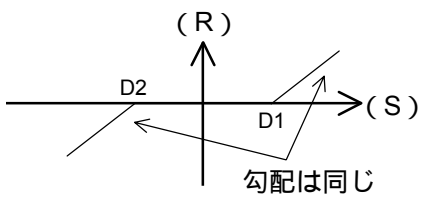


5 応用命令

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。																																									
処理時間	0.37ミリ秒																																									
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	 																																									
有効 パラメータ	<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 エラー																																										


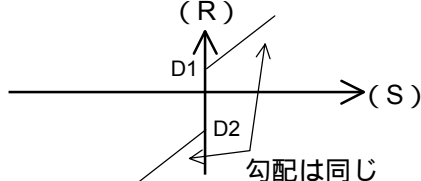


ROL	左回転																																										
機能説明	ソースの内容をデスティネーションの内容で左へ回転し、結果をリザルトに格納します。																																										
パラメータと 処理内容	 ROL S, D, R S : ソース R : リザルト D : デスティネーション																																										
フラグの設定	Eが変化、他はOFF。																																										
処理時間	0.37ミリ秒																																										
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	 																																										
有効 パラメータ	<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 エラー																																											


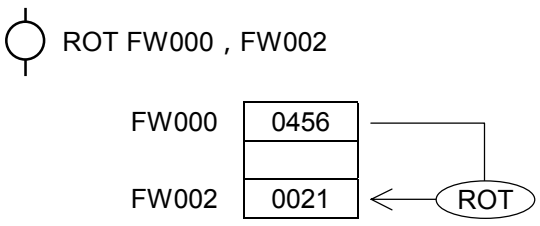
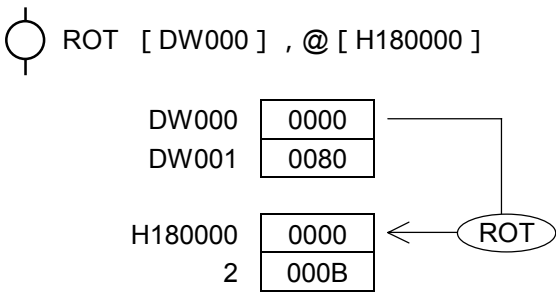
5 応用命令

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。																																								
処理時間	0.52ミリ秒																																								
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" style="display: inline-table; margin-right: 20px;"> <tr><td>FW000</td><td>0023</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> </table> <p>○ LIM [DW000], 2, -1, @ [H180000]</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>DW000</td><td>FFFF</td></tr> <tr><td>DW001</td><td>FFFF</td></tr> </table> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>H180000</td><td>FFFF</td></tr> <tr><td>2</td><td>FFFF</td></tr> </table>	FW000	0023	FW001	0010	FW002	FFF0	FW003	0010	DW000	FFFF	DW001	FFFF	H180000	FFFF	2	FFFF																								
FW000	0023																																								
FW001	0010																																								
FW002	FFF0																																								
FW003	0010																																								
DW000	FFFF																																								
DW001	FFFF																																								
H180000	FFFF																																								
2	FFFF																																								
有効パラメータ	<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></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, 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エラー																																									

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。																																									
処理時間	0.52ミリ秒																																									
備考	D1<D2のときEフラグON																																									
使用例	 BND FW000, FW001, FW002, FW003 <table border="1" data-bbox="502 974 917 1131"> <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>0013</td><td>←</td></tr> </table>  BND [DW000], 2, -1, @ [H180000] <table border="1" data-bbox="502 1254 1069 1456"> <tr><td>DW000</td><td>FFFF</td><td>→</td></tr> <tr><td>DW001</td><td>FFFF</td><td>→</td></tr> <tr><td>H180000</td><td>0000</td><td>←</td></tr> <tr><td>2</td><td>0000</td><td>←</td></tr> </table>			FW000	0023	→	FW001	0010	→	FW002	FFF0	→	FW003	0013	←	DW000	FFFF	→	DW001	FFFF	→	H180000	0000	←	2	0000	←															
FW000	0023	→																																								
FW001	0010	→																																								
FW002	FFF0	→																																								
FW003	0013	←																																								
DW000	FFFF	→																																								
DW001	FFFF	→																																								
H180000	0000	←																																								
2	0000	←																																								
有効パラメータ	<table border="1" data-bbox="375 1545 893 1960"> <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>	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	×			<table border="1" data-bbox="941 1545 1460 1960"> <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, 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エラー																																										


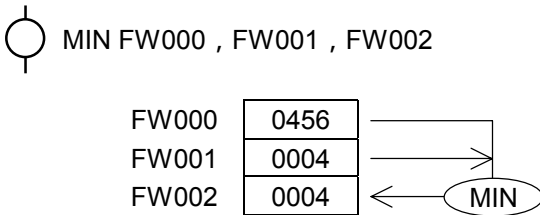
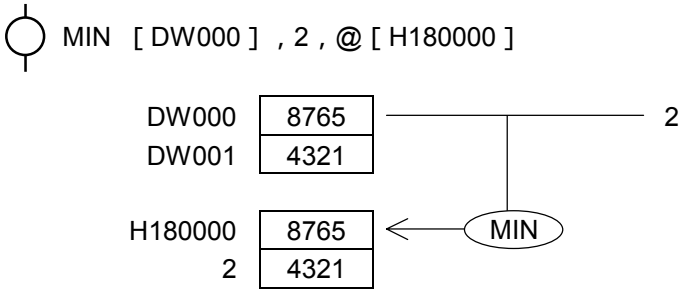
5 応用命令

ZON	デッドゾーン																																										
機能説明	ソースの内容を正負によりそれぞれバイアス（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。																																										
処理時間	0.52ミリ秒																																										
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="454 996 869 1153"> <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>0033</td></tr> </table> <p style="text-align: right;">← (ZON)</p>  ZON [DW000], 2, -1, @ [H180000] <table border="1" data-bbox="454 1276 1021 1467"> <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">(ZON)</td></tr> <tr><td>2</td><td>FFFF</td></tr> </table>			FW000	0023	→	FW001	0010	FW002	FFF0	FW003	0033	DW000	FFFF	→	2	DW001	FFFF	H180000	FFFF	←	(ZON)	2	FFFF																			
FW000	0023	→																																									
FW001	0010																																										
FW002	FFF0																																										
FW003	0033																																										
DW000	FFFF	→	2																																								
DW001	FFFF																																										
H180000	FFFF	←	(ZON)																																								
2	FFFF																																										
有効パラメータ	<table border="1" data-bbox="331 1556 845 1971"> <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>x</td><td></td><td></td></tr> <tr><td>インダイレクトロング長</td><td>x</td><td></td><td></td></tr> </tbody> </table> <table border="1" data-bbox="893 1556 1412 1971"> <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>x</td></tr> <tr><td>ダイレクトロング長</td><td></td><td></td><td>x</td></tr> <tr><td>インダイレクトワード長</td><td>x</td><td></td><td></td></tr> <tr><td>インダイレクトロング長</td><td>x</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	x			インダイレクトロング長	x			R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長			x	ダイレクトロング長			x	インダイレクトワード長	x			インダイレクトロング長	x		
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	x																																										
インダイレクトロング長	x																																										
R	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクトワード長			x																																								
ダイレクトロング長			x																																								
インダイレクトワード長	x																																										
インダイレクトロング長	x																																										
はアドレス値が奇数時はパラメータエラー																																											


ROT	平方根																																										
機能説明	ソースの内容の平方根（整数部のみ）をリザルトに格納します。																																										
パラメータと 処理内容	 ROT S, R S : ソース R : リザルト	S ≥ 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。																																										
処理時間	0.77ミリ秒																																										
備考																																											
使用例	 																																										
有効 パラメータ	<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 エラー																																											

5 応用命令

MAX	最大																																								
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を比較し、大きい値をリザルトに格納します。																																								
パラメータと処理内容	<p>○ MAX 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>S D のとき S R S < D のとき D R</p>																																								
フラグの設定	E, Vが変化、他はOFF。																																								
処理時間	0.42ミリ秒																																								
備考																																									
使用例	<p>○ MAX FW000, FW001, FW002</p> <p>FW000 0456 FW001 0004 FW002 0456</p> <p>○ MAX [DW000], 2, @ [H180000]</p> <p>DW000 8765 DW001 4321 H180000 0000 2 0002</p>																																								
有効パラメータ	<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	×																																								

MIN	最小																																											
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を比較し、小さい値をリザルトに格納します。																																											
パラメータと処理内容	 MIN S, D, R S : ソース R : リザルト D : デスティネーション	S 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。																																											
処理時間	0.42ミリ秒																																											
備考																																												
使用例	 																																											
有効パラメータ	<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エラー																																												

5 応用命令

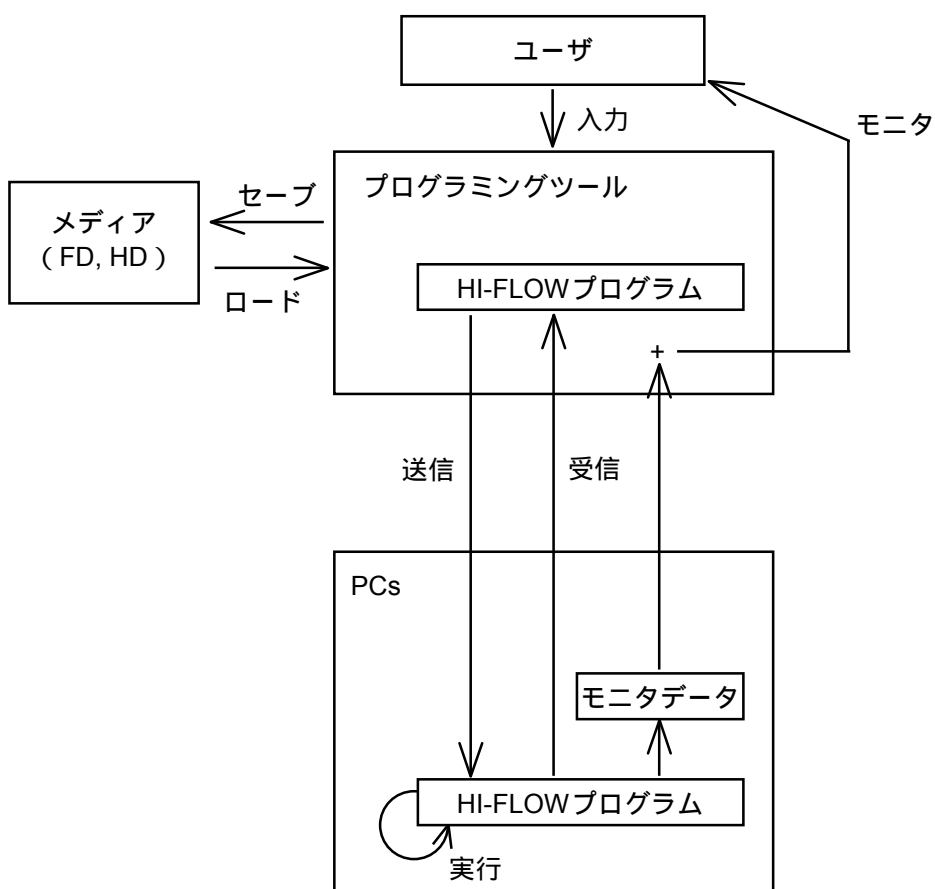
CLR	クリア																																													
機能説明	指定のI/Oエリアをクリアします。TCLR, UCLR, CCLRは計数値エリアもクリアします。																																													
パラメータと 処理内容	 名称 S 名称 : 各CLR命令名称 S : ソース (クリアするI/Oエリアの先頭)																																													
フラグの設定	すべてのフラグに0を設定。																																													
処理時間	下記参照																																													
備考																																														
説明	<table border="1" data-bbox="485 882 1262 1794"> <thead> <tr> <th>名称</th> <th>機能</th> <th>処理時間</th> </tr> </thead> <tbody> <tr> <td>XCLR</td> <td>X000 ~ XFFFをクリア</td> <td>0.92ミリ秒</td> </tr> <tr> <td>YCLR</td> <td>Y000 ~ YFFFをクリア</td> <td>0.92ミリ秒</td> </tr> <tr> <td>GCLR</td> <td>G000 ~ GFFFをクリア</td> <td>0.92ミリ秒</td> </tr> <tr> <td>RCLR</td> <td>R000 ~ RFFFをクリア</td> <td>0.92ミリ秒</td> </tr> <tr> <td>KCLR</td> <td>K000 ~ KFFFをクリア</td> <td>0.92ミリ秒</td> </tr> <tr> <td>TCLR</td> <td>T000 ~ T3FFをクリア Tの計測値クリア</td> <td>2.46ミリ秒</td> </tr> <tr> <td>UCLR</td> <td>U000 ~ U3FFをクリア Uの計測値クリア</td> <td>1.70ミリ秒</td> </tr> <tr> <td>CCLR</td> <td>C000 ~ C3FFをクリア Cの計測値クリア</td> <td>1.70ミリ秒</td> </tr> <tr> <td>VCLR</td> <td>V000 ~ VFFFをクリア</td> <td>0.92ミリ秒</td> </tr> <tr> <td>ECLR</td> <td>E000 ~ EFFFをクリア</td> <td>0.92ミリ秒</td> </tr> <tr> <td>FCLR</td> <td>S020 ~ S027をクリア</td> <td>0.15ミリ秒</td> </tr> <tr> <td>JCLR</td> <td>J000 ~ JFFFをクリア</td> <td>0.92ミリ秒</td> </tr> <tr> <td>QCLR</td> <td>Q000 ~ QFFFをクリア</td> <td>0.92ミリ秒</td> </tr> <tr> <td>HHCLR</td> <td>HH000 ~ HH1FFをクリア</td> <td>1.70ミリ秒</td> </tr> </tbody> </table>	名称	機能	処理時間	XCLR	X000 ~ XFFFをクリア	0.92ミリ秒	YCLR	Y000 ~ YFFFをクリア	0.92ミリ秒	GCLR	G000 ~ GFFFをクリア	0.92ミリ秒	RCLR	R000 ~ RFFFをクリア	0.92ミリ秒	KCLR	K000 ~ KFFFをクリア	0.92ミリ秒	TCLR	T000 ~ T3FFをクリア Tの計測値クリア	2.46ミリ秒	UCLR	U000 ~ U3FFをクリア Uの計測値クリア	1.70ミリ秒	CCLR	C000 ~ C3FFをクリア Cの計測値クリア	1.70ミリ秒	VCLR	V000 ~ VFFFをクリア	0.92ミリ秒	ECLR	E000 ~ EFFFをクリア	0.92ミリ秒	FCLR	S020 ~ S027をクリア	0.15ミリ秒	JCLR	J000 ~ JFFFをクリア	0.92ミリ秒	QCLR	Q000 ~ QFFFをクリア	0.92ミリ秒	HHCLR	HH000 ~ HH1FFをクリア	1.70ミリ秒
名称	機能	処理時間																																												
XCLR	X000 ~ XFFFをクリア	0.92ミリ秒																																												
YCLR	Y000 ~ YFFFをクリア	0.92ミリ秒																																												
GCLR	G000 ~ GFFFをクリア	0.92ミリ秒																																												
RCLR	R000 ~ RFFFをクリア	0.92ミリ秒																																												
KCLR	K000 ~ KFFFをクリア	0.92ミリ秒																																												
TCLR	T000 ~ T3FFをクリア Tの計測値クリア	2.46ミリ秒																																												
UCLR	U000 ~ U3FFをクリア Uの計測値クリア	1.70ミリ秒																																												
CCLR	C000 ~ C3FFをクリア Cの計測値クリア	1.70ミリ秒																																												
VCLR	V000 ~ VFFFをクリア	0.92ミリ秒																																												
ECLR	E000 ~ EFFFをクリア	0.92ミリ秒																																												
FCLR	S020 ~ S027をクリア	0.15ミリ秒																																												
JCLR	J000 ~ JFFFをクリア	0.92ミリ秒																																												
QCLR	Q000 ~ QFFFをクリア	0.92ミリ秒																																												
HHCLR	HH000 ~ HH1FFをクリア	1.70ミリ秒																																												
使用例	 XCLR X000  HHCLR HH000																																													

補 足

補足1 HI-FLOWプログラムの流れ

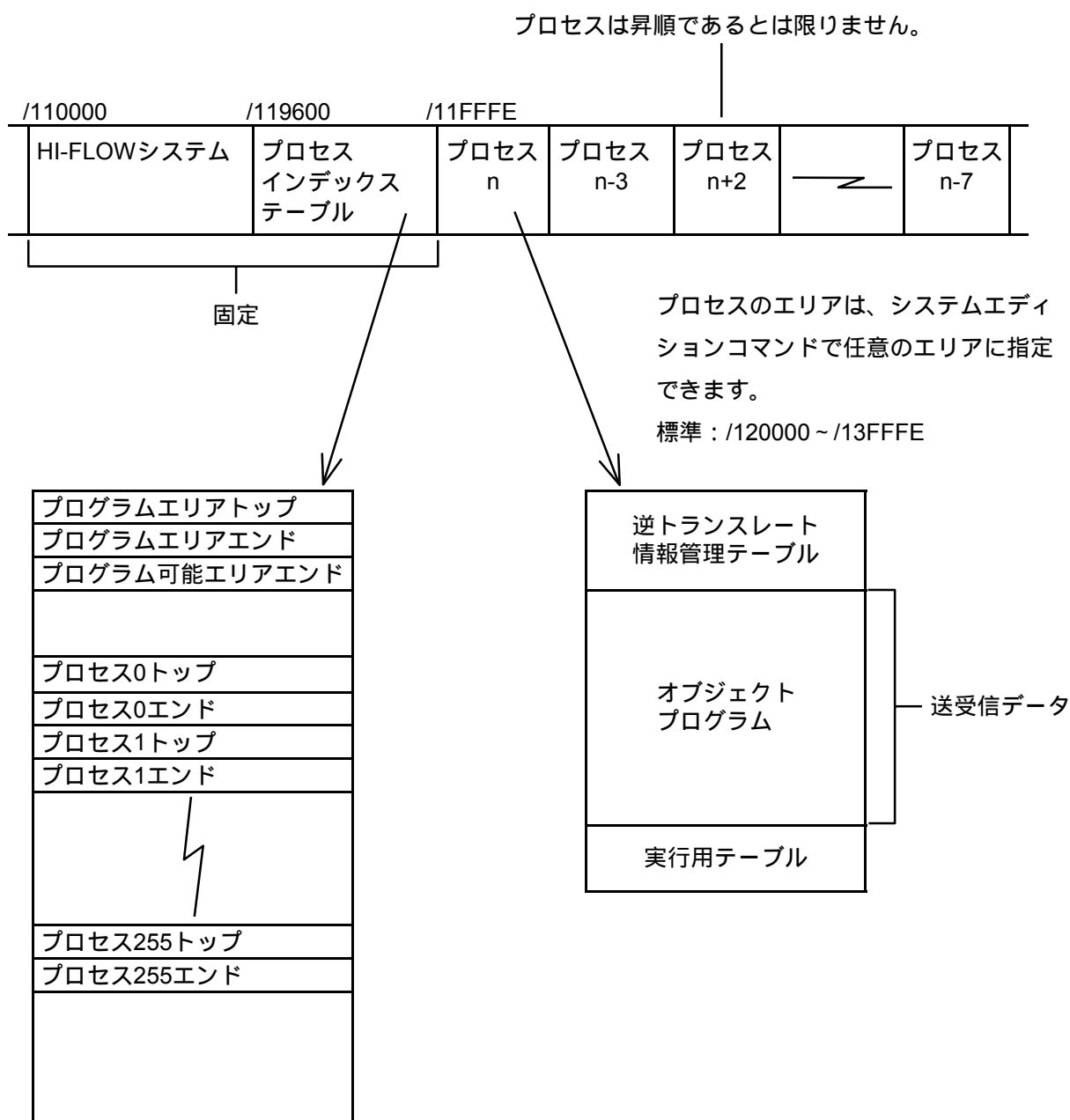
HI-FLOWプログラムはプログラミングツールで作成されPCsで実行します。実行結果などをモニタするときには必要最小限のデータをPCsから受け取り、ツールにあるプログラムと合成して出力します。これは通信量を少なくしてモニタ速度を上げるためです。

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

PCs上で実行されるHI-FLOWプログラムは以下に示すエリアに存在します。PCs上では実際にメモリに配置されますのでメモリマップイメージを示します。



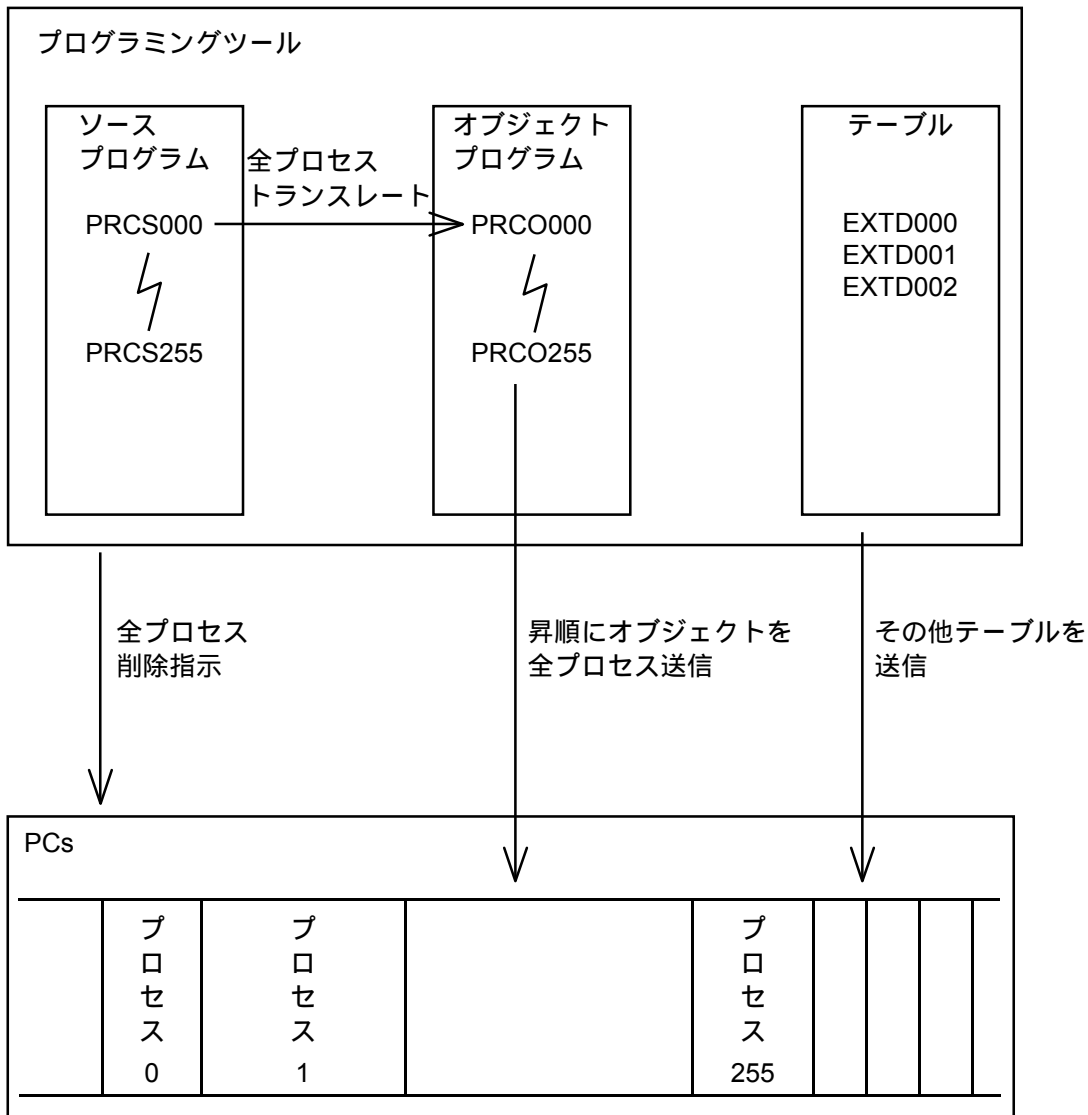
補足3 オンラインモード

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を意味します。

オンラインとは編集/メモリ対象をPCsのプログラムにすることを意味します。ところが対象をPCsにしたとき、モニタ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プロセスのみ一致していればそのプロセスは編集/モニタができます。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

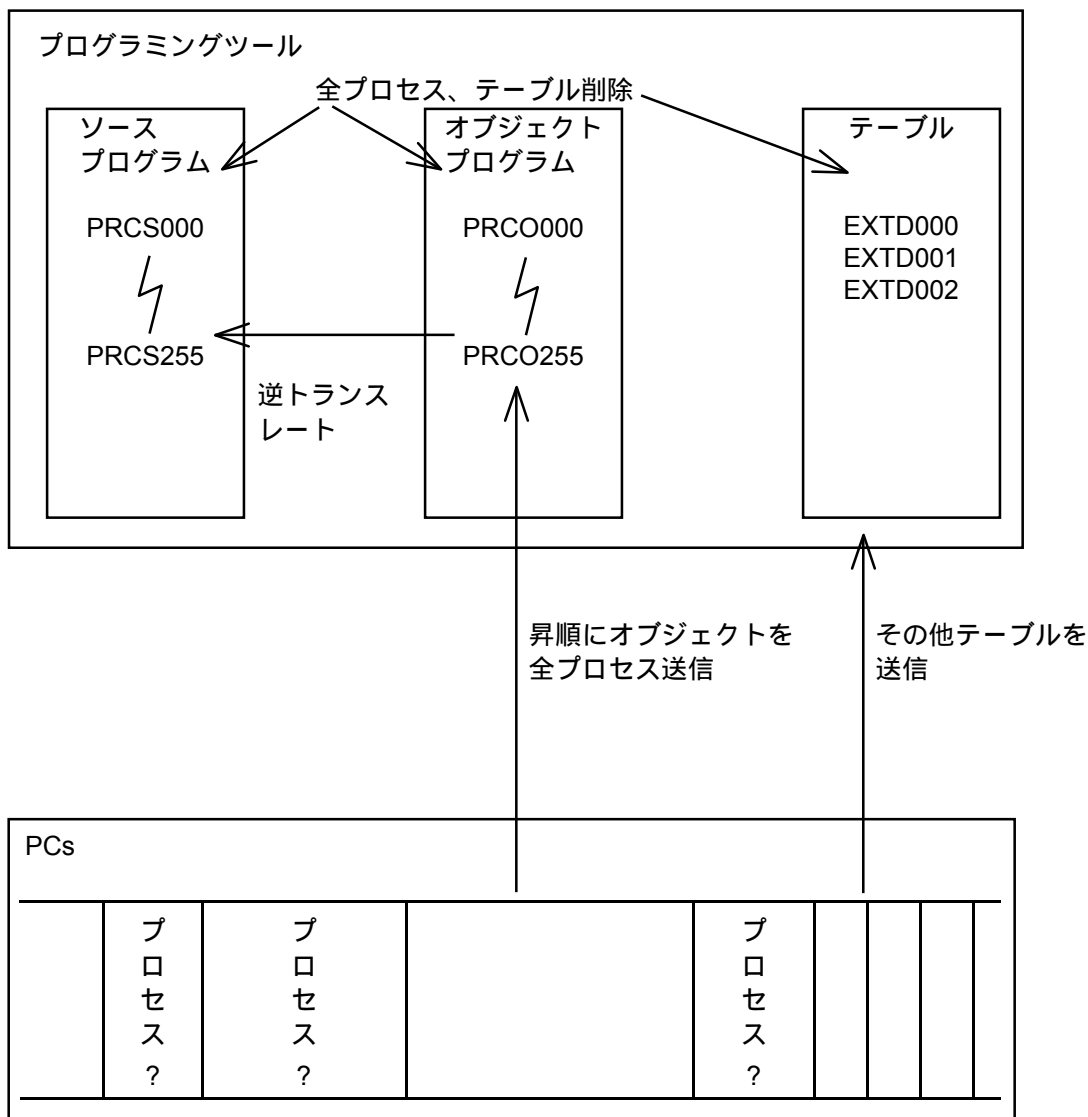
ツール上に存在する全HI-FLOWプログラムをPCsへ送信する際のデータの流れを示します。



全プロセス送信後、メモリ上のプロセス、テーブルは昇順となります。

(2) 全プロセス受信

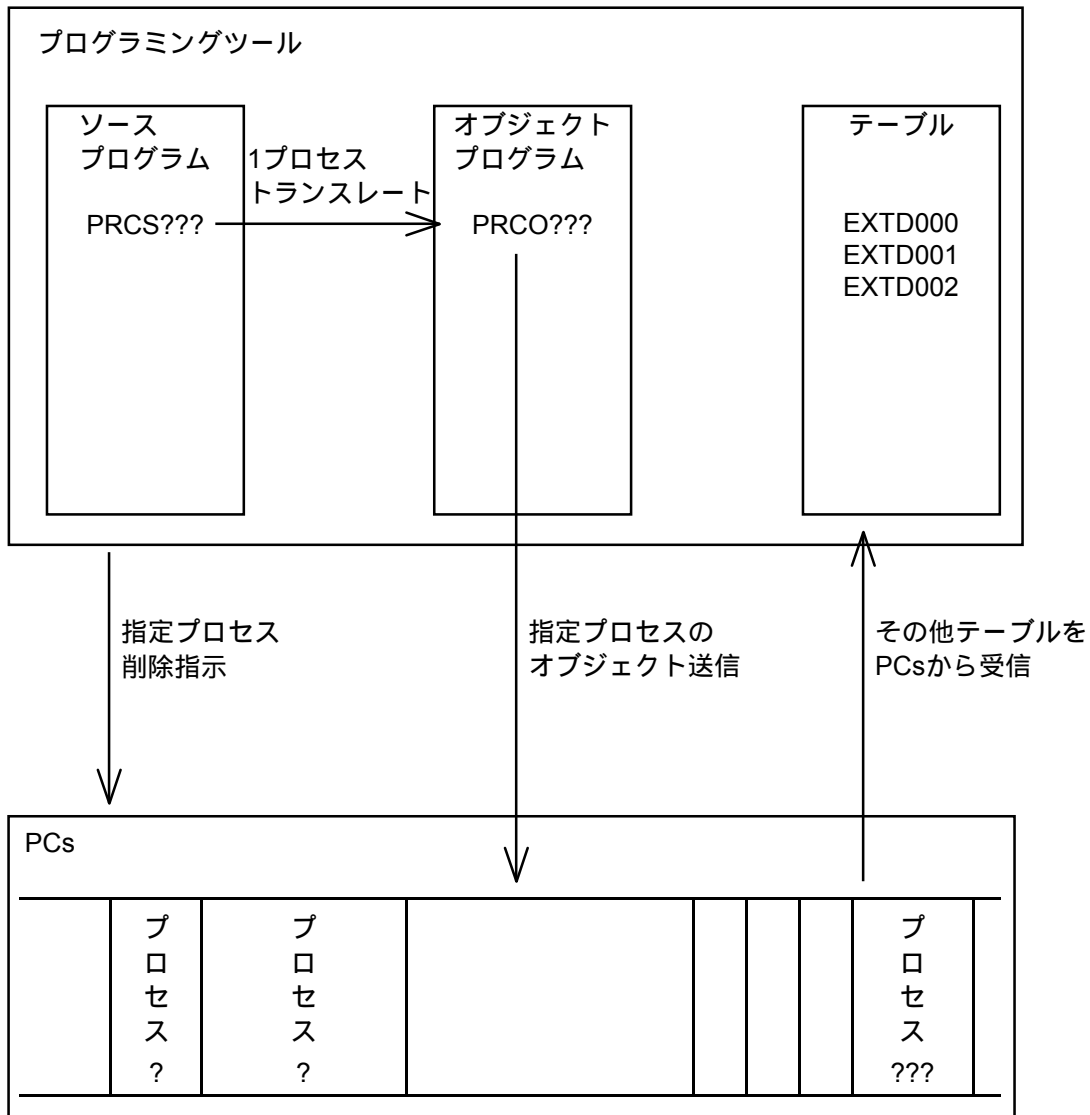
PCs上に存在する全HI-FLOWプログラムをツールへ受信する際のデータの流れを示します。



受信時にメモリ上のプロセス、テーブルが昇順である保証はありません。

(3) 1プロセス送信

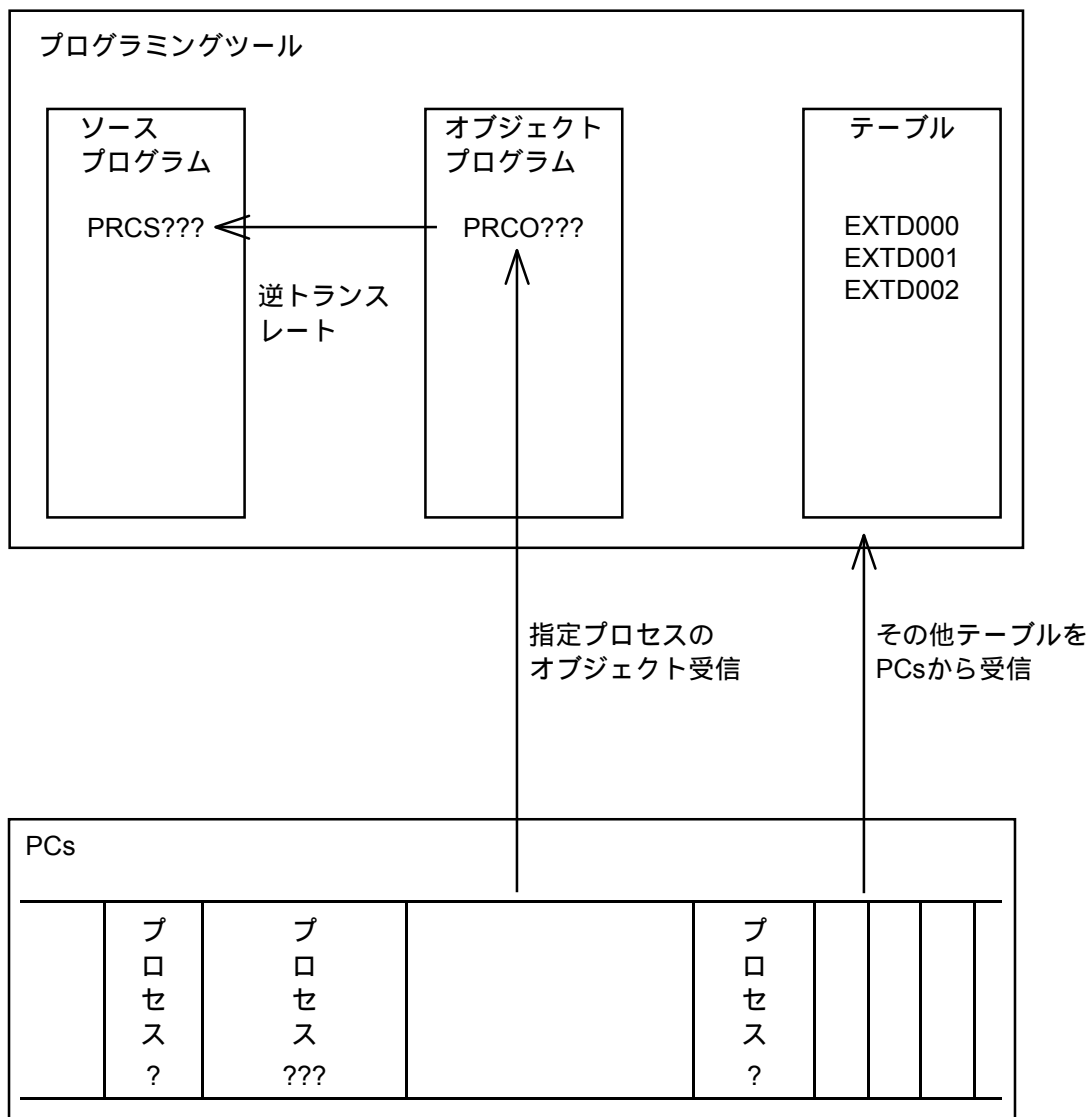
ツール上に存在する特定のHI-FLOWプログラムを1つだけPCsへ送信する際のデータの流れを示します。



送信後、指定プロセスはメモリ上で最後になります。

(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。



受信時にメモリ上のプロセス、テーブルが昇順である保証はありません。

補足4 進行チェック

HI-FLOWはユーザプログラム上の進行位置をモニタ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。

PCsに転送されたユーザプログラムが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	PCsのスキャン時間ごとに進行チェックする。 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。 同じ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<左下<右上<右下と大きくなる）。 同じ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。 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。
コールプロセス	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が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行チェックする。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。
プロセス制御	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プロセス番号が小さければ次スキャン時、大きければ同じスキャンで進行チェックされる。 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。
常時監視	プロセススタートの条件（ACT, STP, RST, CLR）、マルチエントリ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。 インタロック付き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。
分岐	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。
繰り返し	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。

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。 セルウェイト条件成立時は、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なら起動をかける。
条件待ち	ウェ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、マルチエントリ、ボックス、コントロールボックス、ファンクションまた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。

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OS負荷が上がります。OSの負荷が上がると次のような現象が現われます。

- ・シーケンスサイクル時間が設定よりも延びる。
- ・PCsのLED表示が止まる。
- ・PCsリセットするとLED表示がされない。

さらに負荷が上が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。

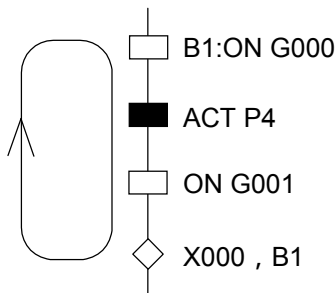
<HI-FLOWプログラムの上手な作り方>

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. タイマ番号は小さい順に詰めて使う。

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

4. ウェイトタイムは同じルートでは同じ番号を使う。

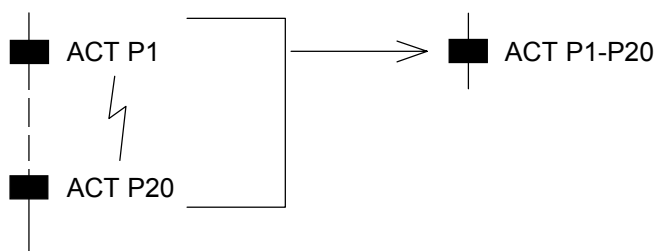
同じルートのウェイトタイムは同時には絶対実行されません。したがって、同じタイム番号を付けてなるべく後ろのタイム番号を使用しないようにします。

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。やむなく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。

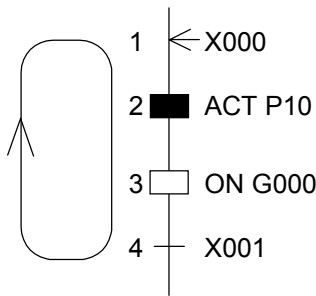
システムコントロールビットはシーケンスサイクルごと、ステップ実行ごとにチェックが必要となり負荷がかなりかかります。必要最小限の設定をしてください。

8. マルチエントリの使用は必要最小限にする。

マルチエントリ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. マルチエントリのインループに注意する。

マルチエントリ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。

プロセススタートに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。

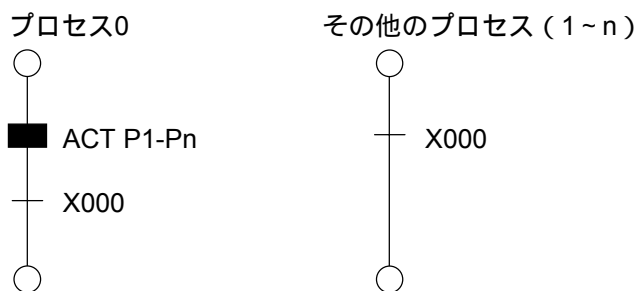
HI-FLOWで条件式を複雑にすると式解析にラダーよりかなりの時間がかかります。あまり複雑な条件はラダーで一度受けてからHI-FLOWに渡したほうが負荷が軽くなります。

HI-FLOWプログラムとCPU負荷の相関図

S10/2 α 、2 α E、および2 α H (f) に関して、プロセスの実行数とそのプロセスが実行可能なシーケンスサイクル時間の目安を示します。

条件を以下に示します。

- (1) プロセスは下記とし、条件は不成立。またラダ - など他のプログラムはなし。



- (2) システムコントロールビットはすべて無効。
 (3) PCs復電 (CPUキースイッチRUN) で実シーケンスサイクル時間を測定する (SW140シーケンスサイクル積算カウンタを一定時間測定する)。

[S10/2 α] ■ : 設定時間内に実行できます。

▨ : 設定するとシーケンスサイクルをストップします。

シーケンスサイクル 設定時間 (単位はミリ秒)	実行プロセス数 (n)						
	10	30	50	100	150	200	256
10		▨	▨	▨	▨	▨	▨
20	■			▨	▨	▨	▨
30	■	■		▨	▨	▨	▨
50	■		■			▨	▨
75	■			■			
100	■			■	■		
125	■			■	■		
150	■			■	■	■	
175	■			■	■	■	■
200	■			■	■	■	■

[S10/2αE] ■■■：設定時間内に実行できます。

シーケンスサイクル 設定時間 (単位はミリ秒)	実行プロセス数 (n)						
	10	30	50	100	150	200	256
10	■■■	■■■					
20	■■■	■■■	■■■	■■■			
30	■■■	■■■	■■■	■■■	■■■		
50	■■■	■■■	■■■	■■■	■■■	■■■	■■■
75	■■■	■■■	■■■	■■■	■■■	■■■	■■■
100	■■■	■■■	■■■	■■■	■■■	■■■	■■■
125	■■■	■■■	■■■	■■■	■■■	■■■	■■■
150	■■■	■■■	■■■	■■■	■■■	■■■	■■■
175	■■■	■■■	■■■	■■■	■■■	■■■	■■■
200	■■■	■■■	■■■	■■■	■■■	■■■	■■■

[S10/2αH (f)] ■■■：設定時間内に実行できます。

シーケンスサイクル 設定時間 (単位はミリ秒)	実行プロセス数 (n)						
	10	30	50	100	150	200	256
10	■■■	■■■	■■■				
20	■■■	■■■	■■■	■■■	■■■		
30	■■■	■■■	■■■	■■■	■■■	■■■	
50	■■■	■■■	■■■	■■■	■■■	■■■	■■■
75	■■■	■■■	■■■	■■■	■■■	■■■	■■■
100	■■■	■■■	■■■	■■■	■■■	■■■	■■■
125	■■■	■■■	■■■	■■■	■■■	■■■	■■■
150	■■■	■■■	■■■	■■■	■■■	■■■	■■■
175	■■■	■■■	■■■	■■■	■■■	■■■	■■■
200	■■■	■■■	■■■	■■■	■■■	■■■	■■■

実際にシステム設計する場合は上記表を参照し、安全率をかけてシーケンスサイクルタイムを設定してください。ラダ - の演算ファンクション、Cモードプログラムが存在することを考慮すると動作可能なシーケンスサイクル時間の半分程度が目安となります。

索 引

	前条件付きウェイト	55			
ア行			タ行		
イフ	47		代入式	37	
ウェイト	35		タイマ	35	
ウェイトタイマ	35		定数	24	
エスケープ	50		同期構文と選択構文の混合	16	
演算子	26				
応用命令	58		ハ行		
応用命令 演算時の型変換	60		パラスタート、パラエンド	51	
応用命令 概説	58		パラレルタイマ	39	
応用命令 機能説明	62		ファンクション	55	
応用命令 システムエラーフラグ	61		プログラム	15	
応用命令 使用方法	58		プロセス	10	
応用命令説明	6		プロセス情報	28	
応用命令 パラメータ	58		プロセススタート、プロセスエンド	30	
			プロセスとは?	10	
カ行			プロセスの状態	11	
概説	4		プロセスの状態遷移	12	
コール	54		分岐開始ルートと分岐合流ルートが同じ	16	
構文	23		分岐開始ルートと分岐合流ルートが異なる	17	
構文説明	5, 30		変数	24	
このマニュアルの使い方	4		ボックス	37	
コメント	28				
コントロールボックス	42		マ行		
			マルチエントリ	53	
サ行			名称	28	
ジャンプ	49				
自由コメント	27		ヤ行		
自由ラベル	27		予約語	24	
出力ビット	35				
条件式	35		ラ行		
シンボル図形	20		ラベル	23	
ステップ	19				
ステップコメント	26				
ステップ番号	20				
セレクト、セルウェイト、セレクトエンド	52				

リピータスタート、リピータエンド	46	JCLR	112
ルート	15	KCLR	112
ルートスタート、ルートエンド	34	LE	80
		LIM	106
		LSL	101
		LSR	100
		LT	79
		MAX	110
		MIN	111
		MOD	69
		MOM	83
		MOV	82
		MUL	67
		NEG	97
		NEQ	76
		NOT	74
		OFF文	39
		ON文	38
		OR	72
		PCsキースイッチとプロセスの状態	14
		POP	86
		PSH	85
		QCLR	112
		RCLR	112
		ROL	105
		ROR	104
		ROT	109
		RST	31, 43
		SCH	88
		SCL	70
		SEG	91
		STP	30, 43
		SUB	64
		TCLR	112
		TRS (タイマリセット)	41
		TST	81
ABS	96		
ACT	32, 42		
ADD	63		
AND	71		
APB	94		
ASL	103		
ASP	92		
ASR	102		
AST	87		
ASU	93		
AUB	95		
BND	107		
BTD	89		
CCLR	112		
CLR	32, 44, 112		
DCD	98		
DEC	66		
DIV	68		
DTB	90		
ECD	99		
ECLR	112		
EOR	73		
EQU	75		
EXC	84		
FCLR	112		
GCLR	112		
GE	78		
GT	77		
HHCLR	112		
HI-FLOWプログラムの構成	2		
INC	65		

索引

TUP (タイマアップ)	40
UCLR	112
VCLR	112
XCLR	112
YCLR	112
ZON	108

ご利用者各位

〒101-8010

東京都千代田区神田駿河台4丁目6番地
株式会社日立製作所

お 願 い

各位にはますますご清栄のことと存じます。

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

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