

HITACHI

ソフトウェアマニュアル

プログラミング

HI-FLOW For Windows[®]

S10VE

SEJ-3-122(A)

ソフトウェアマニュアル

プログラミング

HI-FLOW For Windows®

SIOVE

この製品を輸出される場合には、『外国為替及び外国貿易法』の規制ならびに『米国輸出管理規則』など外国の輸出関連法規をご確認のうえ、必要な手続きをお取りください。
なお、ご不明な点がございましたら、当社担当営業にお問い合わせください。

2018年 8月 (第1版) SEJ-3-122 (A)

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

安全上のご注意

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

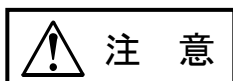
警告表示の定義



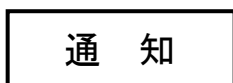
: この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こす危険の存在を示す。



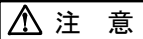

: この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こすおそれのある危険の存在を示す。



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



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

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

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

重大な傷害

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

軽度の傷害または中程度の傷害

治療のために入院や長期の通院を必要としないけが、やけど、感電傷害など

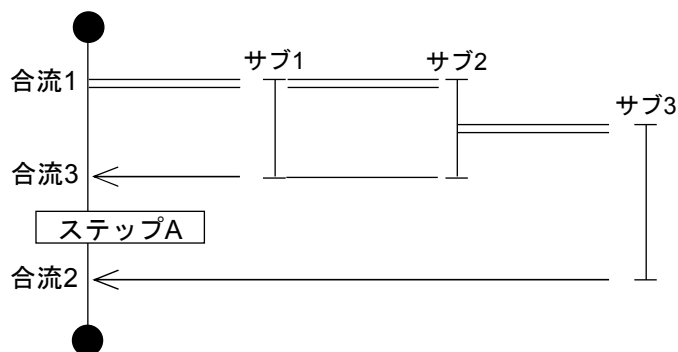
人身傷害とは関係のない損害

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

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

通 知

- 下図のパターンのように、パラスタートしてセレクトエンドで合流する構文において、サブルートの中で再度パラスタートし、サブルートではなくメインルートにセレクトエンドで合流する構文は記述しないでください。合流地点に達した場合に、強制終了されるのは合流地点が同一のルートだけであるため、合流していないサブルートが実行中のまま残ってしまいます。

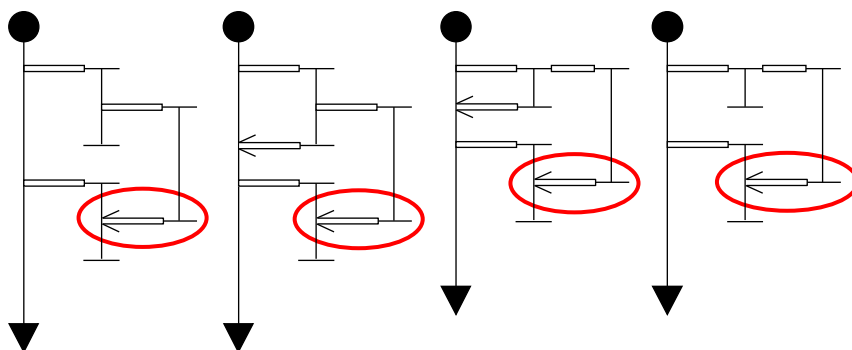


[動作説明]

- ・サブ1～3が実行中の状態で、サブ1、2が合流3に到達する前にサブ3が合流2に到達した場合、サブ1とサブ2の実行点は強制終了されず実行中のまま残ってしまいます。

(4-28ページ)

- 下図のパターンのように、非同期プロセスの分岐と合流が異なり、合流ルートが非同期ルートである構文は記述しないでください。実行した場合、メインルートを新たに起動してもサブルートを実行中と判断して起動がかからない状態になります。



- 起動の際にマスターリセットを指定した場合、非同期ルートの実行中にビット型PI/Oがクリアされる可能性がありますので注意してください。

(4-35ページ)

来歴一覧表

改訂No.	来歴（改訂内容）	発行年月	備考
A	新規作成	2018.8	

はじめに

日立プログラマブルコントローラ用のプログラミング言語HI-FLOWは、フロー図形式で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。

<商標について>

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

<記憶容量の計算値についての注意>

- 2ⁿ計算値の場合（メモリ容量・所要量、ファイル容量・所要量など）
 - 1KB（キロバイト）＝1,024バイトの計算値です。
 - 1MB（メガバイト）＝1,048,576バイトの計算値です。
 - 1GB（ギガバイト）＝1,073,741,824バイトの計算値です。
 - 1TB（テラバイト）＝1,099,511,627,776バイトの計算値です。
- 10ⁿ計算値の場合（ディスク容量など）
 - 1KB（キロバイト）＝1,000バイトの計算値です。
 - 1MB（メガバイト）＝1,000²バイトの計算値です。
 - 1GB（ギガバイト）＝1,000³バイトの計算値です。
 - 1TB（テラバイト）＝1,000⁴バイトの計算値です。

目次

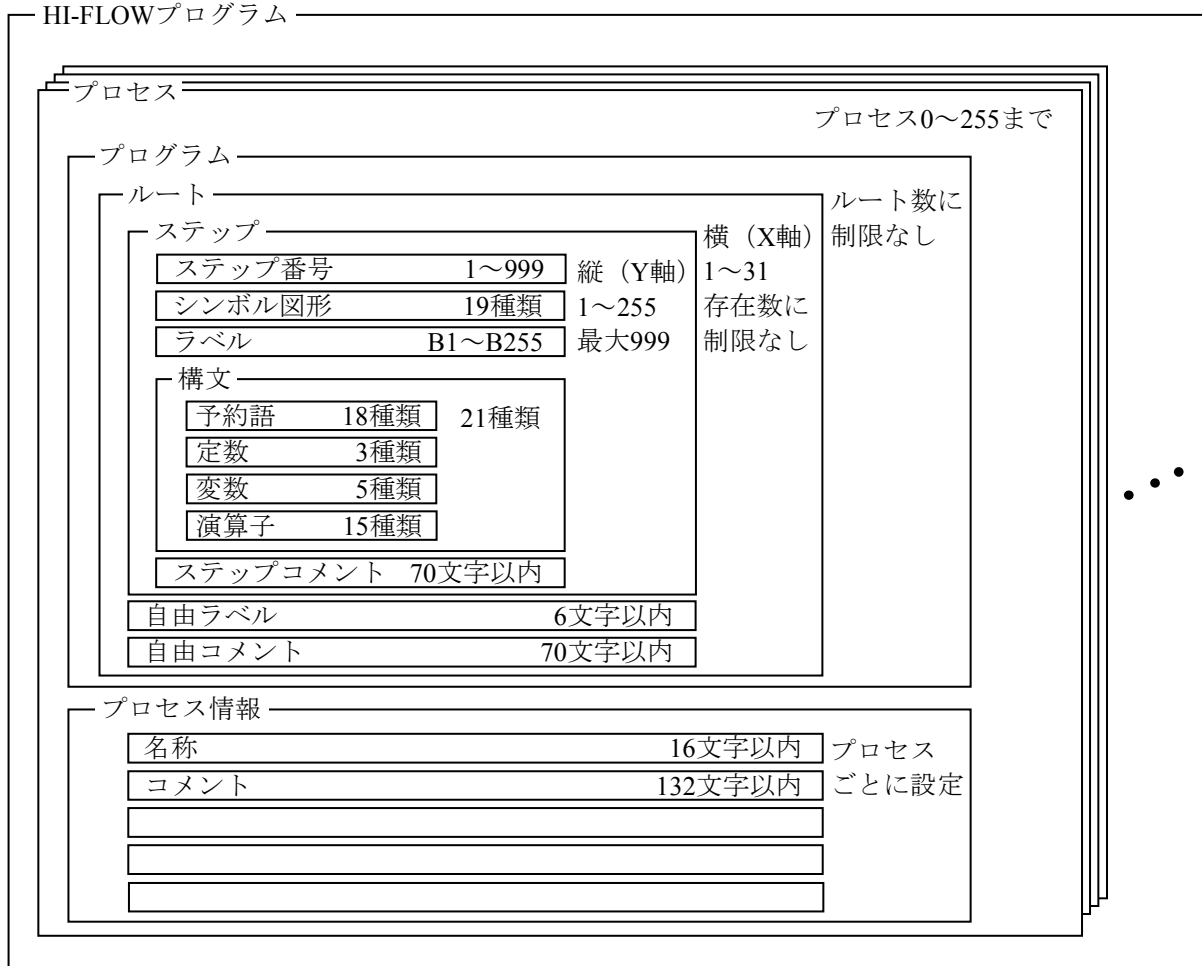
第1章 HI-FLOWプログラムの構成	1-1
第2章 このマニュアルの使い方	2-1
2.1 概説	2-1
2.2 構文説明	2-2
2.3 応用命令説明	2-4
第3章 プロセス	3-1
3.1 プロセスとは?	3-1
3.2 プログラム	3-6
3.3 プロセス情報	3-21
第4章 構文説明	4-1
4.1 プロセススタート、プロセスエンド	4-1
4.2 ルートスタート、ルートエンド	4-5
4.3 ウェイト	4-6
4.4 ボックス	4-8
4.5 コントロールボックス	4-15
4.6 リpeatスタート、リpeatエンド	4-19
4.7 イフ	4-20
4.8 ジャンプ	4-22
4.9 エスケープ	4-23
4.10 パラスタート、パラエンド	4-24
4.11 パラスタート、セレクトエンド	4-25
4.12 セレクト、セルウェイト、セレクトエンド	4-29
4.13 マルチエントリー	4-30
4.14 コール	4-31
4.15 ファンクション	4-32
4.16 前条件付きウェイト	4-32
4.17 非同期プロセスエンド	4-33
第5章 応用命令	5-1
5.1 概説	5-1
5.2 使用方法	5-1
5.3 パラメーター	5-1
5.4 演算時の型変換	5-4

5. 5	システムエラーフラグ	5-5
5. 6	機能説明	5-6
補足		Z-1
補足A	HI-FLOWプログラムの流れ	Z-1
補足B	PCsメモリー	Z-2
補足C	オンラインモード	Z-3
補足D	進行チェック	Z-7
補足E	HI-FLOWプログラムとCPU負荷	Z-9

このページは白紙です。

第1章 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きに、必要に応じて参照してください。
 ユーザーが作成するHI-FLOWプログラムは、次の要素から成り立っています。



このページは白紙です。

第2章 このマニュアルの使い方

2. 1 概説

このマニュアルは第1章で表した構成に従って作られています。
各項目に対応した章とページを示します。

項目	対応した章	ページ
プロセス	第3章	3-1
プログラム	3. 2節	3-6
・ ルート		3-6
・ ステップ		3-11
・ ステップ番号		3-12
・ シンボル図形		3-12
・ ラベル		3-16
・ 構文		3-16
・ 予約語		3-17
・ 定数		3-17
・ 変数		3-17
・ 演算子		3-19
・ ステップコメント		3-19
・ 自由ラベル		3-20
・ 自由コメント		3-20
プロセス情報	3. 3節	3-21
・ 名称		3-21
・ コメント		3-21

第2章 このマニュアルの使い方

2. 2 構文説明

このマニュアルは、概要の次に構文の詳細を機能ごとに説明しています。

各機能に対応した章とページを示します。

項目	図形	対応した章	ページ
構文説明		第4章	4-1
プロセススタート、 プロセスエンド	● ●	4. 1 節	4-1
・STP			4-2
・RST			4-3
・CLR			4-3
・ACT			4-3
ルートスタート、 ルートエンド	⌋ ⌋	4. 2 節	4-5
ウェイト	+	4. 3 節	4-6
・条件式			4-6
・タイマー			4-6
・出力ビット			4-6
・ウェイトタイマー			4-6
ボックス	□	4. 4 節	4-8
・代入式			4-8
・特殊代入式			4-9
・ON文			4-12
・OFF文			4-12
・パラレルタイマー			4-13
・TUP			4-14
・TRS			4-14
コントロールボックス	■	4. 5 節	4-15
・ACT			4-15
・RST			4-16
・STP			4-16
・CLR			4-17
リピートスタート、 リピートエンド	⌋ ⌋	4. 6 節	4-19
イフ	◇	4. 7 節	4-20
ジャンプ	↳	4. 8 節	4-22
エスケープ	✕	4. 9 節	4-23
パラスタート、 パラエンド	⌋ ⌋	4. 10 節	4-24
パラスタート、 セレクトエンド	⌋ ⌋	4. 11 節	4-25
セレクト、 セルウェイト、 セレクトエンド	⌋ ≡≡≡	4. 12 節	4-29
マルチエントリー	⌋	4. 13 節	4-30

項目	図形	対応した章	ページ
コール	Ⓜ	4. 14節	4-31
ファンクション	⓪	4. 15節	4-32
前条件付きウェイト	⊕*	4. 16節	4-32
非同期プロセスエンド	↓	4. 17節	4-33

2.3 応用命令説明

HI-FLOWではラダー図と同様機能の応用命令をサポートしています。

応用命令の各機能に対応した項目の一覧を示します。

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

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

このページは白紙です。

第3章 プロセス

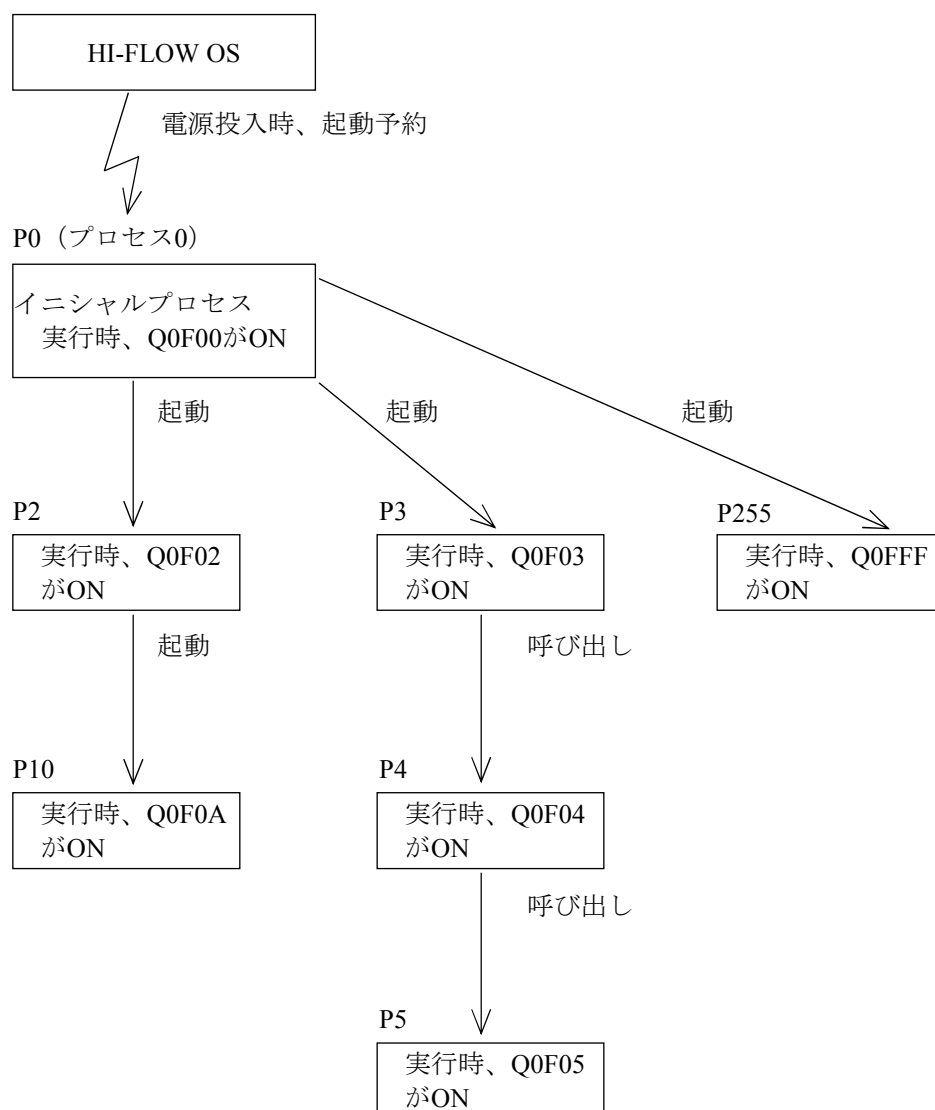
3.1 プロセスとは？

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

プロセスはP+プロセス番号（10進数）で認識します（P0～P255）。

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

プロセスが実行中の場合、指定したPI/OレジスターがONし、その状態を監視できます（標準Q0F00～Q0FFF、「S10VE ソフトウェアマニュアル オペレーション HI-FLOW For Windows®（マニュアル番号 SEJ-3-132）」の「4.7 HI-FLOWプロセスシートユーティリティ機能」の「システムビット割付」を参照）。



プロセスの状態

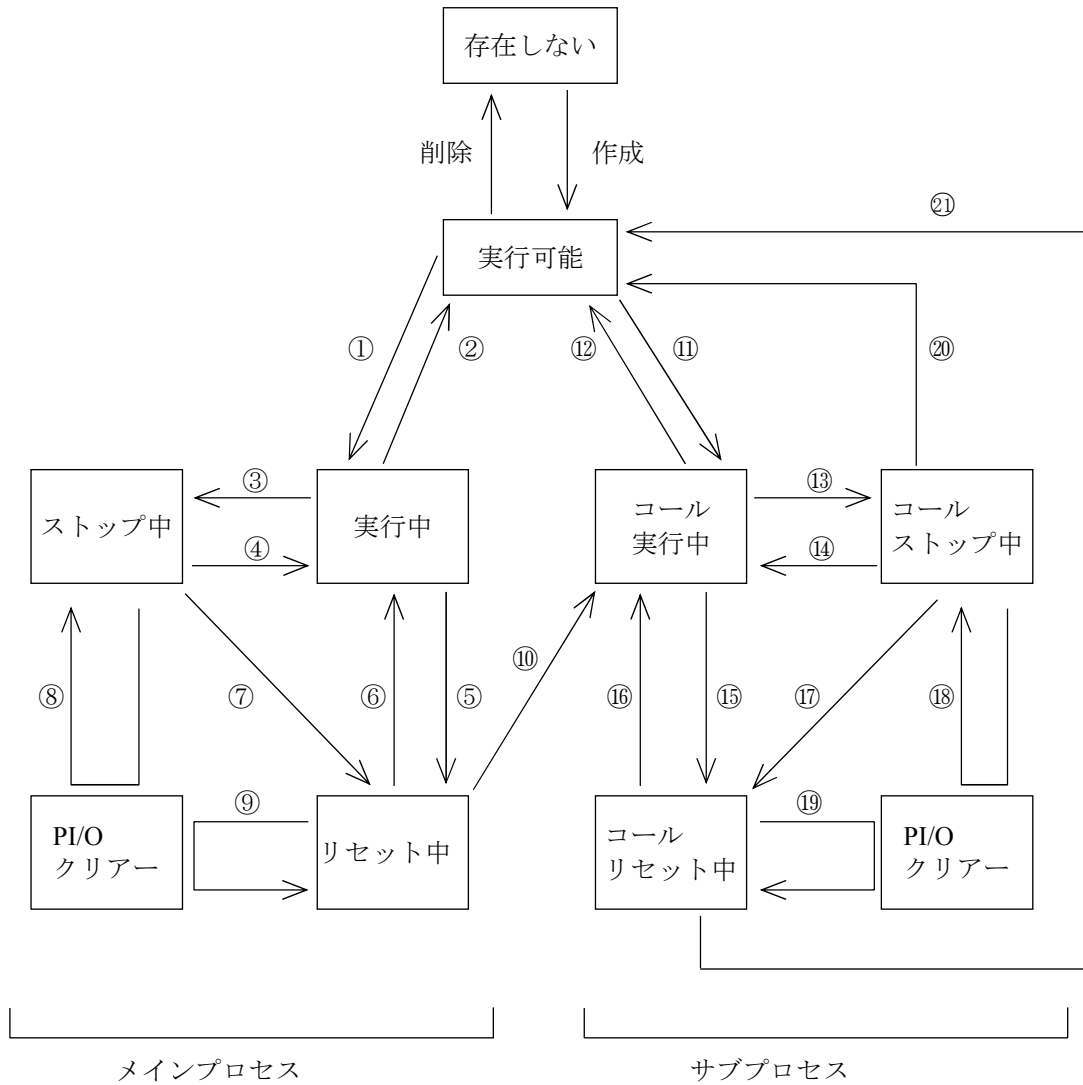
PCs上のプロセスは9つの状態が考えられます。

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

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

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

第3章 プロセス

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

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

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

PCsキースイッチとプロセスの状態

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

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

[a] PCs停復電 (PCsキースイッチリセット)

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

初期化の内容

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

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

[b] PCsキースイッチSTOP中

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

[c] PCsキースイッチRUN (SIM RUN) 中

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

[d] PCsキースイッチSTOP→RUN (SIM RUN)

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

PCs停復電直後だけでなく指定によってPCs停復電と同じ効果 (ただし、HI-FLOW関連だけ) の後 [c] になることもできます。

[e] PCsキースイッチRUN (SIM RUN) →STOP

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

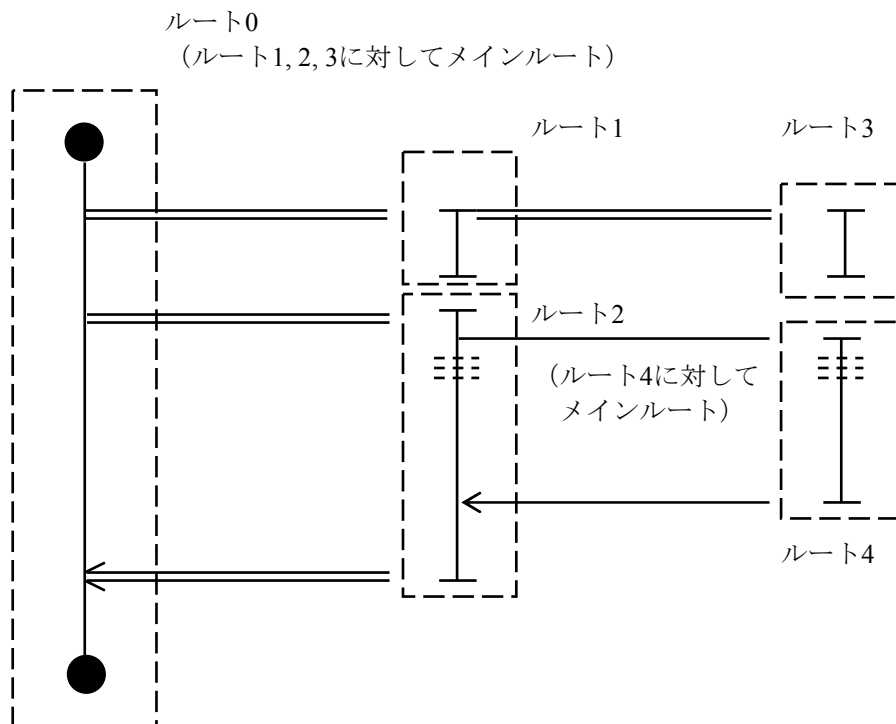
3. 2 プログラム

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

ルート

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

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



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

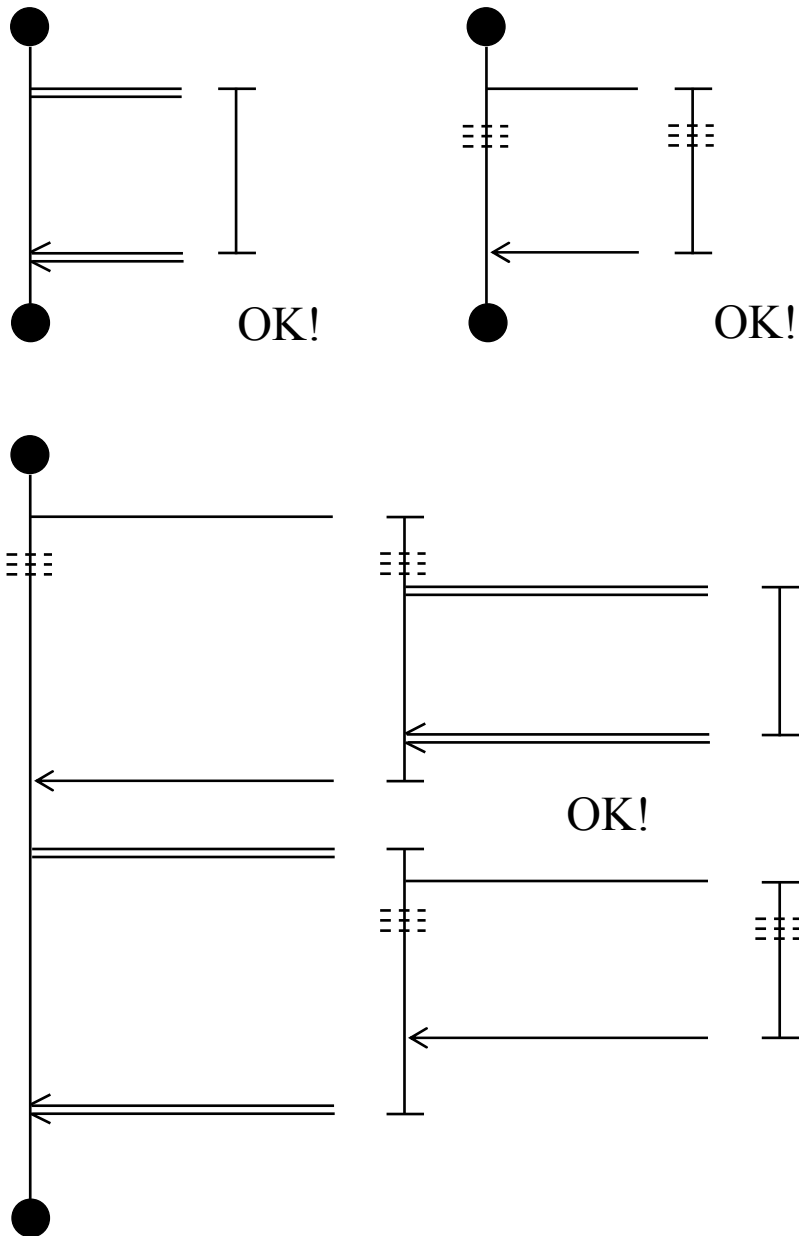
選択ルートはたとえ無条件分岐していても他のどこかのルートに合流する必要があります。

(1) 同期構文と選択構文の混合

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

(a) 分岐開始ルートと分岐合流ルートが同じ

同期、選択共にあらゆるパターンが可能です。

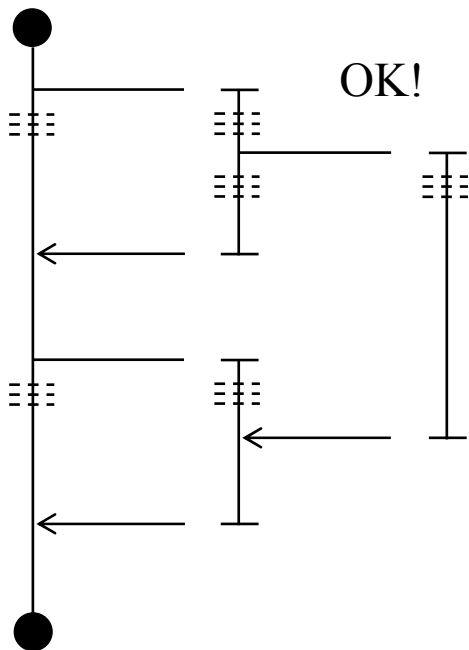
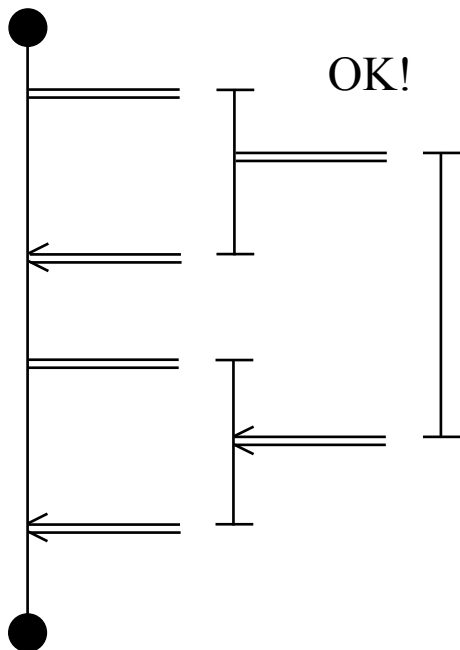
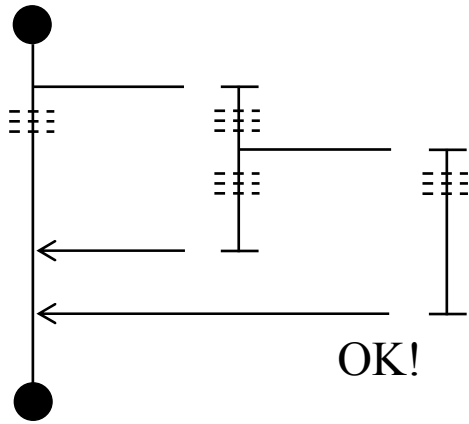
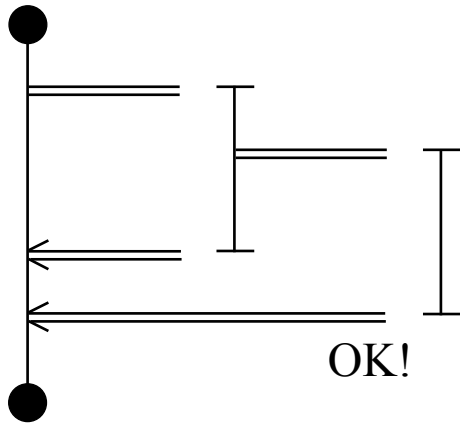


(b) 分岐開始ルートと分岐合流ルートが異なる

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

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

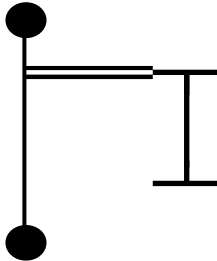
[正常動作]



(2) 非同期ルートの構文

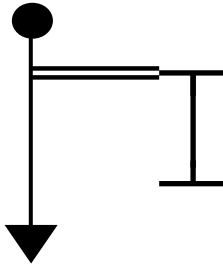
非同期ルートの構文は、非同期プロセスエンド使用時の合流なし分岐ルートを示します。同期ルートとの違いは下記となります。

① 同期ルート（プロセスエンド使用時）の場合



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

② 非同期ルート（非同期プロセスエンド使用時）の場合



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

詳細は、「4. 17 非同期プロセスエンド」を参照してください。

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

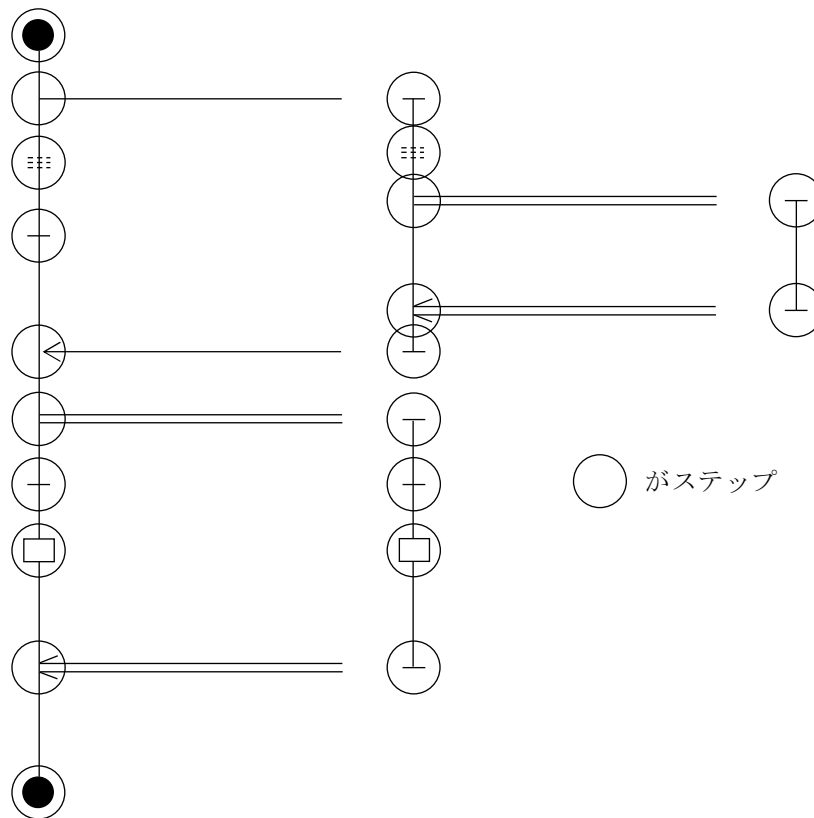
ステップ

ステップは自由ラベル、自由コメントと共にルートの構成単位となります。

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



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



ステップ番号

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



シンボル図形

条件、分岐、制御などの概略を意味します。ステップを作成するときは必ずシンボル図形が必要です。シンボル図形だけでステップが完結するものと構文が必要なものがあります。



シンボル図形は19種類あり、図形そのものの形に意味があります。次に図形の一覧を示します。

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

HI-FLOWで使用できる図形一覧

(1/3)

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

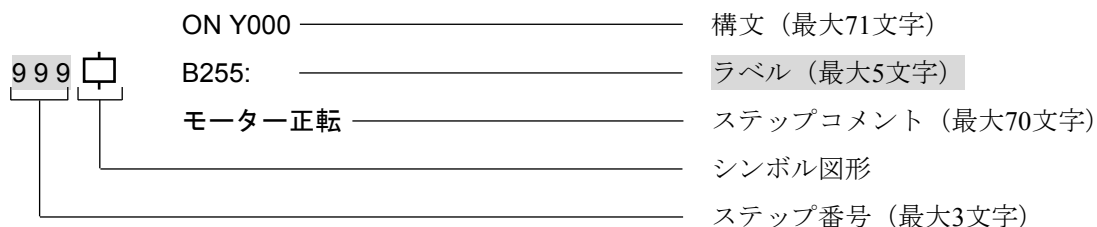
番号	図形	名称	機能	構文	備考
15		セレクト エンド	選択サブルート の合流	なし	分岐元ルートに合流しなく てもよい スキャン遅れなしで次ス テップへ
			同期サブルート の合流	なし	分岐元ルートに合流する 必要あり 同期サブルートは終了
		マルチ エントリー	設定した条件が 成立すると、こ のステップから 再実行	あり	
16		ウェイト	移行条件成立 待ち	あり	
			指定時間経過 待ち	あり	PI/O連続成立監視可能
17		ボックス	PI/O出力	あり	インターロック付きY出力あ り
			代入式	あり	
			PI/O波形出力	あり	
			タイマー リセット	あり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

(3/3)

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

ラベル

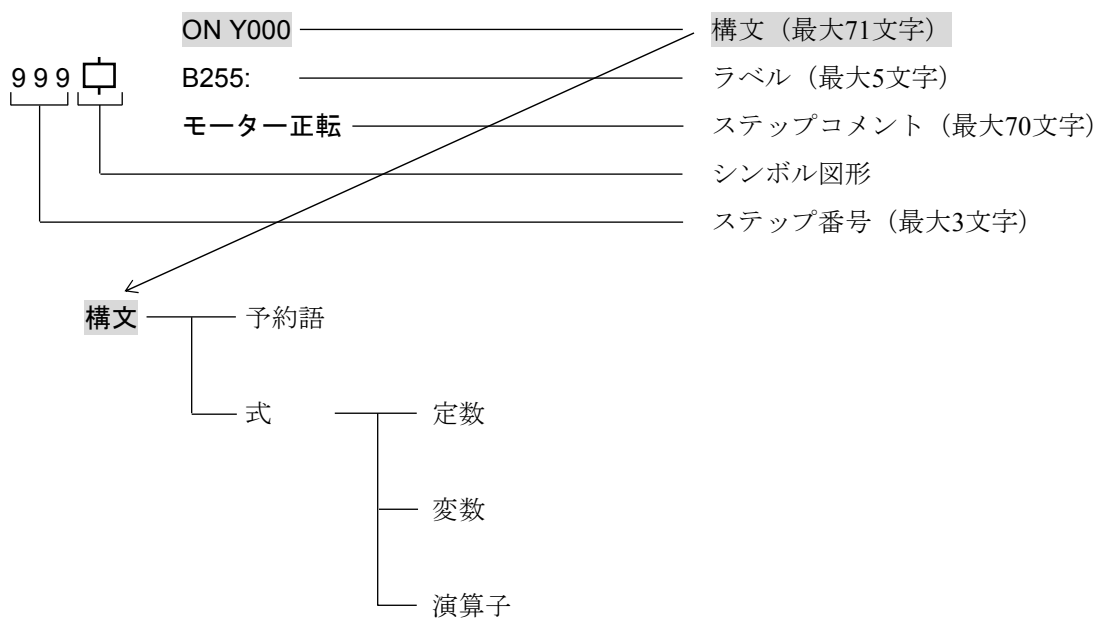
B1～B259（プロセスごとに作成可能、他プロセスへの分岐指定はできません）と：（コロン）で表し、分岐図形からの飛び先を示します。ステップにだけ付加できます。



構文

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

構文は予約語と定数、変数、演算子からなる式で構成されています。



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

予約語

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

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

定数

HI-FLOWではロング長の定数が指定できます。

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

変数

HI-FLOWでは実PI/Oレジスター (X, Yなど) が使用できます。

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

HI-FLOWで使用可能な実PI/Oレジスター一覧を示します。

変数	ビット型	単純	例えば、X0000 (5)。ただし、応用命令内では使用不可。
		一次元配列	
	ワード型	単純	例えば、XW0000 (FW000)。ただし、応用命令内では使用不可。
		一次元配列	
	ロング型	単純	ただし、応用命令内でだけ使用できます。 [FW000] でFW000とFW001をロング長で扱います。

ワード型およびロング型の値の範囲は以下のように扱います。

ワード型: -32768~32767

ロング型: -2147483648~2147483647

第3章 プロセス

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	0000 ~ FFFF	ビット	
		XW	0000 ~ FFF0	ワード	
	外部出力	Y	0000 ~ FFFF	ビット	
		YW	0000 ~ FFF0	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	0000 ~ FFFF	ビット	
		MW	0000 ~ FFF0	ワード	
		E	0000 ~ FFFF	ビット	
		EW	0000 ~ FFF0	ワード	
		Z	000 ~ 3FF	ビット	
		ZW	000 ~ 3F0	ワード	
		S	0000 ~ BFFF	ビット	
		SW	0000 ~ BFF0	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	0000 ~ FFFF	ビット	
		QW	0000 ~ FFF0	ワード	
		HH	000 ~ 1FF	ビット	他プロセスとのリンクのため
		DW	000 ~ FFF	ワード	
		FW	000 ~ BFF	ワード	
	S10V 拡張レジスタ	LB	0000 ~ FFFF	ビット	
		LBW	0000 ~ FFF0	ワード	
		LWW	0000 ~ FFFF	ワード	
LXW		0000 ~ 3FFF	ワード		
タイマー	WT	000 ~ 255		10進表記	
	WTS	000 ~ 255			
	PT	000 ~ 255			
カウンター	CN	000 ~ 127		10進表記	
	CNE	000 ~ 127			
ラベル	B	001 ~ 255 ユーザー設定ラベルあり (6文字以内)		10進表記、プロセスごと	

演算子

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

項目		内容	優先順位
演算子	論理	& (アンド) (オア) ~ (ノット) ^ (イオア)	5
	四則	*	2
		/	
	関係	+	3
		-	
関係	=、<>、<、>、>=、<=	4	
カッコ	6重まで	1	

ステップコメント

英字／数字／カナ／平仮名／漢字／特殊記号を使用して記述します。

一行に入るだけ記述できます。必ずしも作成する必要はありません。



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

自由ラベル

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

自由ラベルはステップ以外にだけ付加できます。分岐図形の飛び先となります。

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

自由コメント

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

自由コメントはステップ以外にだけ付加できます。

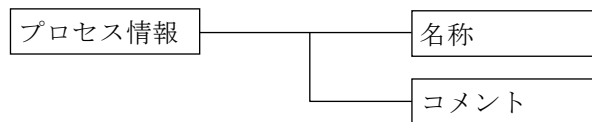
LABEL: _____ 自由ラベル（6文字以内）
合流地点 _____ 自由コメント（70文字以内）

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

3.3 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。これによりユーザーにとってより親しみやすいプロセスを作成できます。

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



名称

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

コメント

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

このページは白紙です。

第4章 構文説明

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

なお、[] は省略可能、{ } は選択、～は繰り返しを意味します。

4. 1 プロセススタート、プロセスエンド

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

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

プロセスエンドは自ルート以外のルートがすべて終了していると処理を行います。終了していないと終了するまで待ち状態となります。ただし、非同期プロセスエンドを使用する非同期ルートは例外です (詳しくは「4. 17 非同期プロセスエンド」を参照してください)。起動した際、マスターリセット指定されていると自プロセスでONするビット型PI/Oを0クリアーします (ON文、パラレルタイマー)。

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

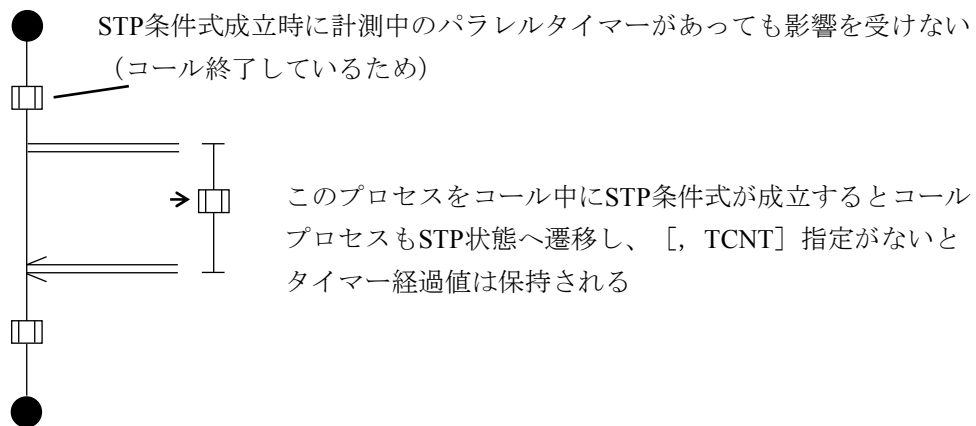
[構文]

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

● または ▼ どちらも構文なし

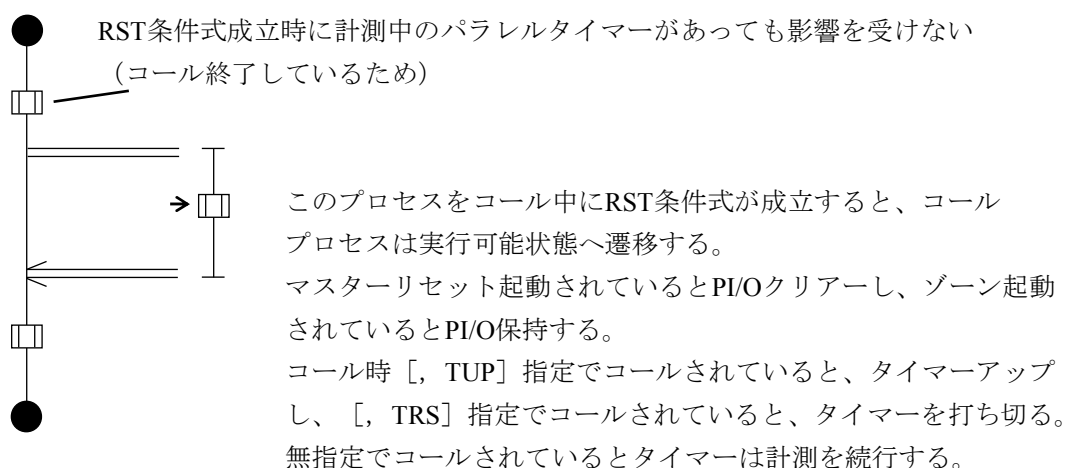
STP

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



RST

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



CLR

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

ACT

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

[プロセススタート (●) のプログラム例] ～は同じ行の続き

1. ● STP X0000, RST X001, CLR X0002, ACT X0003

- ・ X0000がONのときストップ状態へ (タイマー経過値保持)
- ・ X0001がONのときリセット状態へ (タイマー打ち切り)
- ・ ストップ/リセット状態でX0002がONのときこのプロセスで使用しているON文、パラレルタイマー内ビット型PI/Oを0クリアー
- ・ ストップ/リセット状態時、X0000とX0001がOFFでX0003がONのときプロセス実行中へ

2. ● STP G000&X0020, TCNT [ON J000:OFF J001] ～, RST Q0000, TUP

- ・ G000, X0020が共にONのときストップ状態へ (タイマー計測続行)
- ・ ストップ状態遷移時J000をONし、J001をOFFする
- ・ Q0000がONのときリセット状態へ (タイマーアップ)
- ・ プロセス実行中、毎スキャンJ000をOFFし、J001をONする

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

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

4. ● RST Q0001, TUP [ON J001, G200] , CLR X0200

- ・ Q0001がONのときリセット状態へ (タイマー打ち切り)
- ・ リセット状態遷移時J001, G200をONする
- ・ ストップ/リセット状態でX0200がONのときこのプロセスで使用しているON文、パラレルタイマー内ビット型PI/Oを0クリアー
- ・ プロセス実行中、毎スキャンJ001とG200をOFFする

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

4. 2 ルートスタート、ルートエンド

┘ 構文なし

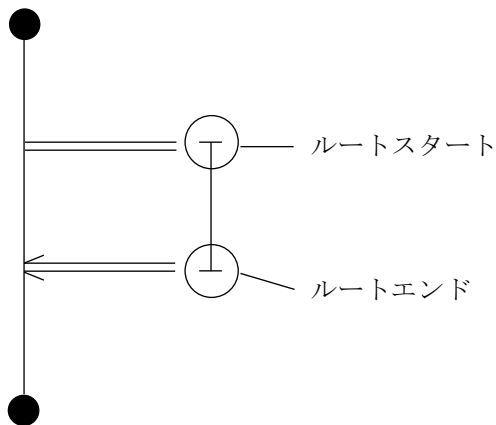
└ 構文なし

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

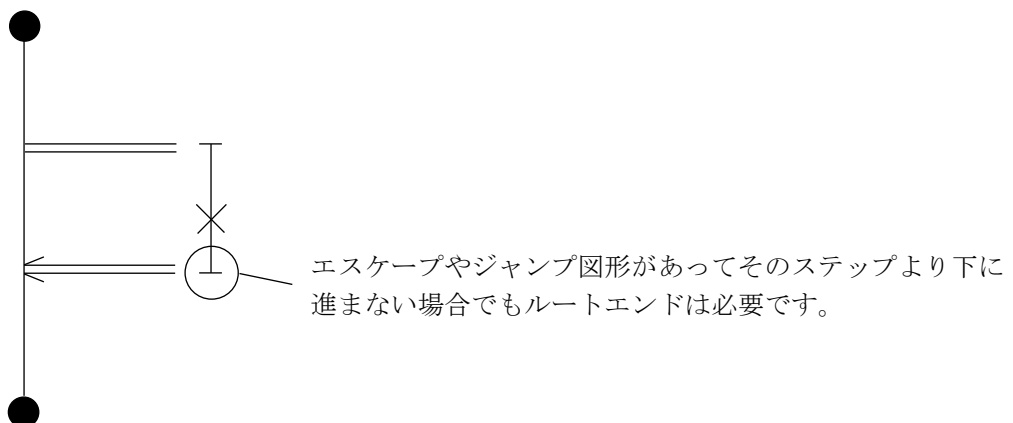
サブルートを作成することによって同期構文や選択分岐構文が構築できます。

[ルートスタート (┘), ルートエンド (└) のプログラム例]

1.



2.



4.3 ウェイト

次ステップへの移行条件が成立するまでこのステップで待ち状態になります。

移行条件が条件式の場合と、ウェイトタイマー（指定時間経過待ち）の場合があります。

[構文]

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

条件式

- ・ビット型またはワード型の数と演算子で構成されます。

タイマー

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

出力ビット

- ・上記タイマーで指定した時間が経過しても条件式が成立しない場合に、このビットがONします。
- ・出力ビットに指定できるレジスタ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。
- ・タイマーで指定した時間が経過しても条件が成立しない限り、次ステップへは移行しません。
- ・出力ビットをONした後に条件が成立しても、出力ビットはOFFしません。

ウェイトタイマー

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

[ウェイト（ \dagger ）のプログラム例]

1. \dagger X0000

X0000がONのとき次ステップに進む

2. \dagger GW000<H2000

GW000がH2000より小さくなったとき次ステップに進む

3. \dagger X0001 (FW000)

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

4. \dagger WT000 (100)

最初にこのステップに到着してから10秒経過後、次ステップへ進む。

5. \dagger WT255 (10, X001F)

このステップに到着後、X001Fが連続1秒ONし続けたら次ステップへ進む。

6. \dagger GW000>H2000, 100, Y0000

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Y0000をONする。

Y0000がONした後に、GW000の値がH2000より大きくなってもY0000をOFFしない。

7. \dagger WT000 (FW000, X001F)

このステップに到着後、X001FがFW000の値の時間（100ms単位）ONし続けたら次ステップへ進む。ウェイト中にFW000の値を変更した場合、開始時の値で動作し、次回実行時に反映されます。

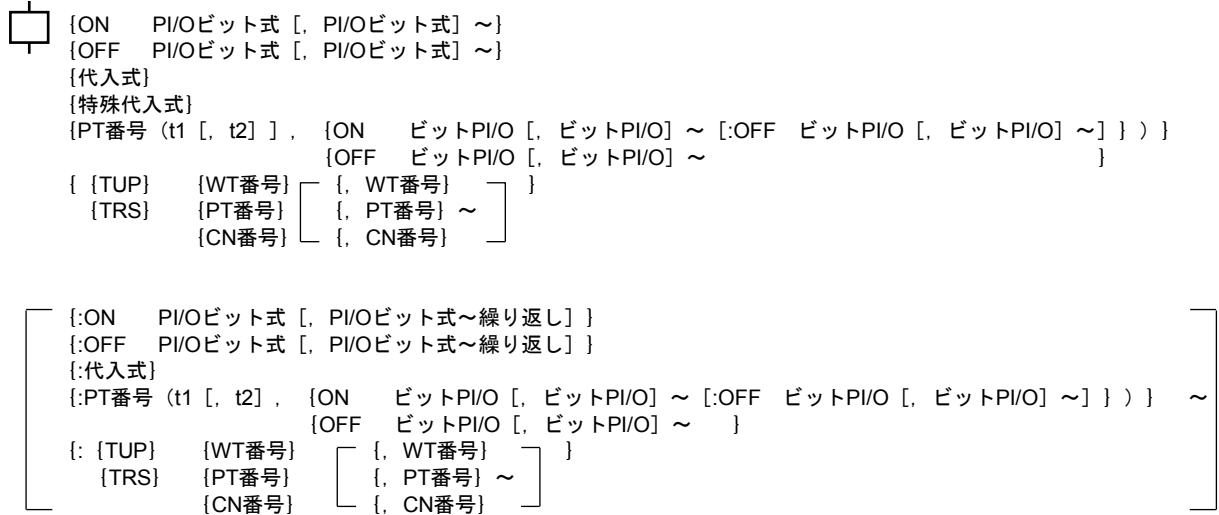
第4章 構文説明

4. 4 ボックス

PI/Oの出力、データ処理、タイマー制御を行います。

ボックスは：（コロン）で区切ることによって複文構成にできます。

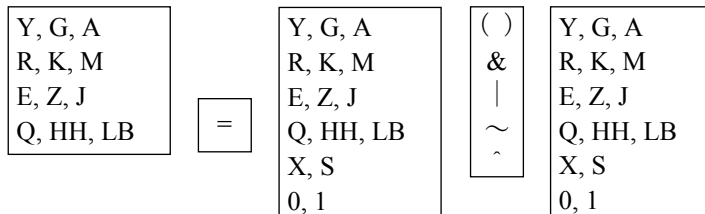
[構文]



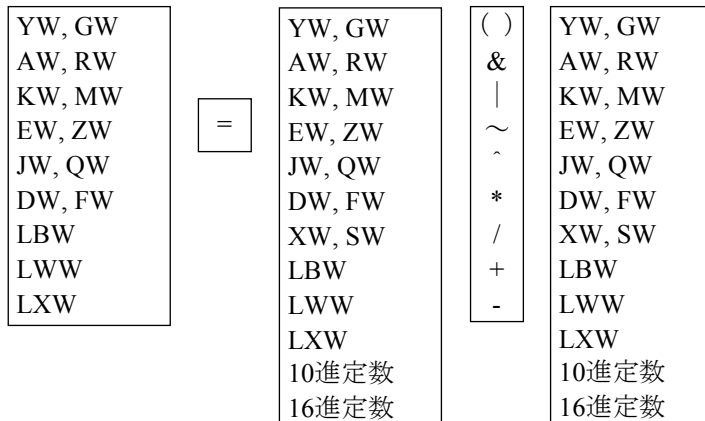
代入式

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が、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)



(ワード型変数)



演算項目、結果は符号なしとみなします。

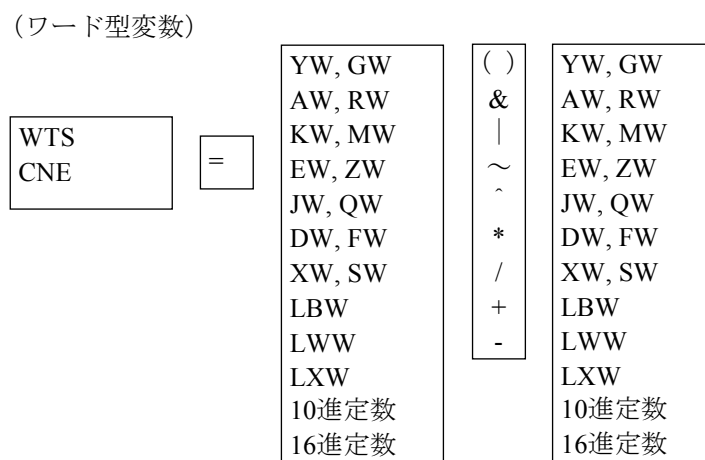
乗算は乗数、被乗数とも1ワードで、1ワードで表せない部分は切り捨てられて結果も1ワードとなります。

除算も除数、被除数とも1ワードで、1ワードで表せない部分は切り捨てられて結果も1ワードとなります。また、0除算されると答えは変化しません。

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

特殊代入式


タイマー（WT）とカウンタ（CN）については、特殊な代入式をサポートします。




WTSはタイマー設定値であり、計測中タイマーの設定値を変更できます。計測中のタイマーを、計測開始した時より早くタイムアップさせたい時に使用できます。計測開始していないタイマーやタイムアップ済のタイマーに対して設定値を変更しても意味を持ちません。


CNEはカウンター終了値であり、リピートスタート／リピートエンド構文内で使用した場合に意味を持ちます。リピート中に終了値を変更して、開始した時の終了値より早く終了させたり、遅く終了させたり出来ます。リピートスタート／リピートエンド構文以外で使用しても意味を持ちません。終了値の変更を誤ると、無限ループとなりシステムダウンを引き起こす可能性もありますので、注意して使用してください。

第4章 構文説明

[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.  YW0000 (DW001) =HFFFF

その時点でのDW001を添え字値としたYW0000の配列内容へ/FFFFを代入する。

3.  WTS001=50

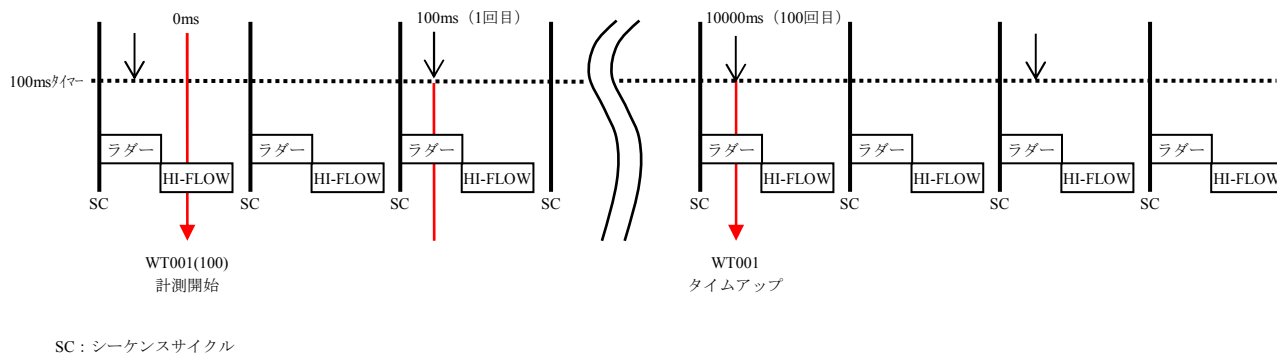
WT001の設定値を50に変更。

4.  CNE001=DW000

CN001の終了値をDW000の内容に変更。

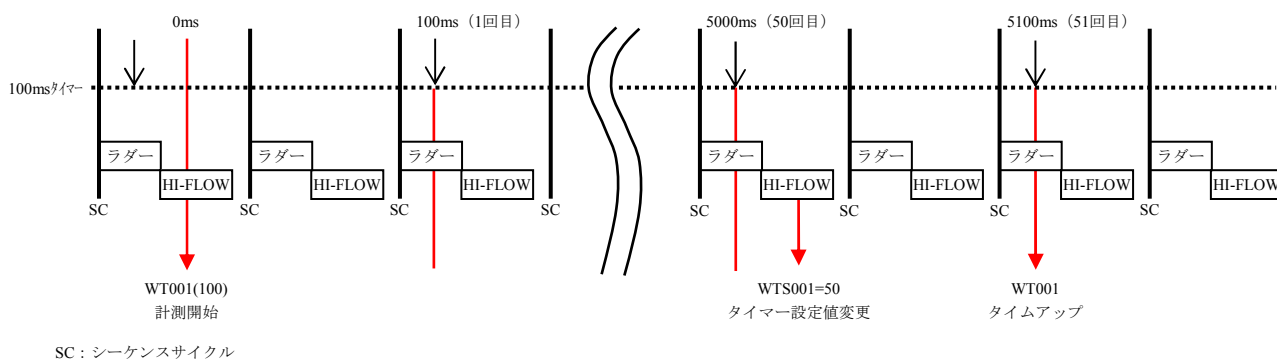
タイマー（WT）の動作を以下に示します。

<通常のタイムアップ>



WT001に100を設定して計測開始した場合、計測開始後100回目のタイマー割込みにてタイムアップします。

<タイマー設定値変更時のタイムアップ>



WT001に100を設定して計測開始し、5000ms後に設定値を50に変更した場合、計測開始後51回目以降のタイマー割込みにてタイムアップします。すなわち、タイマー設定値変更は、変更した瞬間ではなく変更後の100ms間隔のタイマー割込みがあった時点でタイムアップします。

ON文

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

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

[ON文（□）のプログラム例]

1. □ ON Y0000, Y000F:OFF Y0001

Y0000とY000FをONし、Y0001をOFFして次ステップへ進む。

2. □ ON G000 (GW010)

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

OFF文

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

，（カンマ）で区切ることによって複数のPI/Oを出力できます。PI/O出力ビットは1次元配列が可能です。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 Y0000, Y0001

Y0000とY0001をOFFして次ステップへ進む。

2. □ OFF G000 (GW010)

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

パラレルタイマー

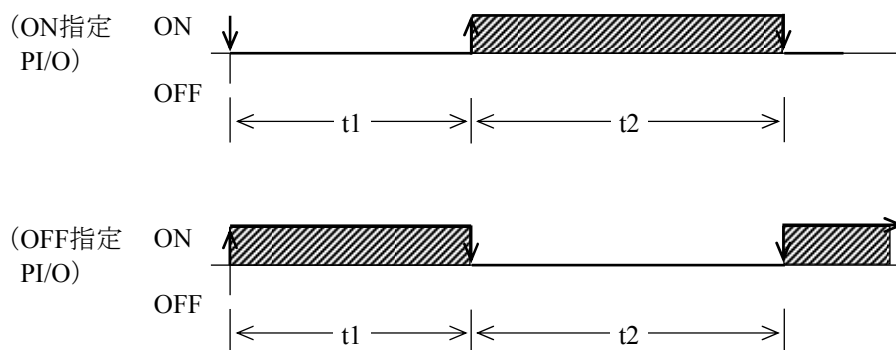
任意のPI/Oに波形を出力します。t1は立ち上がり時間、t2は立ち下がり時間を示します。

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

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

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

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



[パラレルタイマー（□）のプログラム例]

- PT000 (10, 10, ON Y0000:OFF Y0001)


	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
Y0000	? → OFF	→ ON	→ OFF
Y0001	? → ON	→ OFF	→ ON

? : ONまたはOFF

- 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	→

? : ONまたは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	→

? : ONまたはOFF

TUP (タイマーアップ)


計測中のタイマーをアップ状態にします。

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

パラレルタイマーの場合、タイマー経過値をt2 (t2省略時はt1) にします。この結果パラレルタイマーは指定時間より早くPI/O出力します。

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

[タイマーアップ () のプログラム例]


1.  TUP WT001, WT002, PT001, CN001


ウェイトタイマー1、2、パラレルタイマー1、カウンター1をアップします。

TRS (タイマーリセット)

計測中のタイマーをリセットします。

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

[タイマーリセット () のプログラム例]

1.  TRS WT001, WT002, PT001, CN001

ウェイトタイマー1、2、パラレルタイマー1、カウンター1をリセットします。

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, ファクター番号}
{ RST Pxxx { [-Pxxx] [, TUP] } }
  { [, TASK] }
{ STP Pxxx [-Pxxx] [, TCNT] }
{ CLR Pxxx [-Pxxx] }

```

ACT

	項目	内容
1	機能概略	P0～P255で指定されたプロセスに起動をかける。 - (ハイフン) でプロセスの範囲指定可能。ステップ番号指定がなければステップ1から実行する。指定されたステップはメインルートでなくてもよい。起動後はすぐ次ステップへ進む。
2	起動されたプロセスの動き	起動されたプロセスは一度きりの実行でなく、プロセスエンドが実行終了すると次スキャンで再びプロセススタートから実行する（ステップ指定されていても同様）。
3	実行中プロセス起動	コントロールボックス結果表示ビットのACTビットをONし、次ステップへ進む（標準HH1FF）。
4	存在しないプロセス起動	コントロールボックス結果表示ビットのACTビットをONし、次ステップへ進む（標準HH1FF）。
5	ストップ状態プロセス起動	ストップ状態のプロセスに起動がかかり、実行が再開する。
6	リセット状態プロセス起動	リセット状態のプロセスに起動がかかり、プロセススタートから実行が再開する。
7	タイマー状態指示	, TUPオプションの指定でプロセスエンド、エスケープ実行時および実行可能状態移行時に自プロセスで占有しているパラレルタイマーをアップする。 , TRSオプションの指定でプロセスエンド、エスケープ実行時および実行可能状態移行時に自プロセスで占有しているパラレルタイマーをリセットする。
8	マスターリセット指定起動	, MRSTオプションの指定で、プロセスエンド、エスケープ実行時および実行可能状態移行時に自プロセスでONしているビット型PI/Oを0クリアする（ON文、パラレルタイマー）。
9	CPMSタスク起動	, TASK, ファクタ番号オプションでPxxxをCP上のCPMSタスク（1～255）としてRLEAS, QUEUEマクロを発行する。

第4章 構文説明

RST

	項目	内容
1	機能概略	P0～P255で指定されたプロセスにリセットをかける。 - (ハイフン) で範囲指定可能。発行後はすぐ次ステップへ進む。
2	リセットされたプロセスの動き	指定されたプロセスの実行を打ち切りリセット状態へ遷移し、プロセススタートで再実行待ち（他プロセスからのACT起動か、自プロセススタートのACT条件成立で再実行）。
3	タイマー状態指示	， TUPオプションの指定でプロセスが占有しているパラレルタイマーをアップする。無指定の場合はタイマーをリセットする。このオプションが有効なのは指定プロセスだけでコールプロセスには影響を与えない。
4	被リセットプロセスのPI/O	マスターリセット起動されていると自プロセスでON/OFFしているビットPI/Oを0クリアーする。
5	ストップ状態プロセスへ発行	指定されたプロセスの実行を打ち切りリセット状態へ遷移し、プロセススタートで再実行待ち（他プロセスからのACT起動か、自プロセススタートのACT条件成立で再実行）。
6	存在しないプロセスへ発行	コントロールボックス結果表示ビットのRSTビットをONし、次ステップへ進む（標準HH1FD）。
7	自プロセスへのリセット発行	自プロセス番号をパラメーターで指定する。
8	CPMSタスク停止	， TASKオプションでPxxxをCP上のCPMSタスク（1～255）としてABORTマクロ発行（本機能は非サポートです）。

STP

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

CLR

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

[コントロールボックス (■) のプログラム例]

1. ■ ACT P1-P5, MRST

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

2. ■ ACT P100, 5, TUP

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

3. ■ ACT P80, TASK, 3

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

4. ■ RST P10

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

5. ■ RST P11, TUP

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

6. ■ RST P12, TASK

CPMSタスク12に対してABORTマクロを発行した後、次ステップへ進みます（本機能は非サポートです）。

7. ■ STP P50

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

8. ■ STP P51, TCNT

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

9. ■ CLR P40

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

4. 6 リピートスタート、リピートエンド

リピートスタートとリピートエンドの間を繰り返し実行します。


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

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


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



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

[構文]

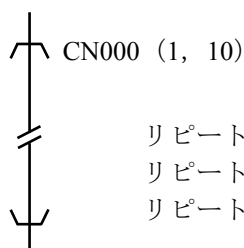
 CNxxx (初期値, 終了値 {, 増分})

(xxxは000～127の10進数)

 構文なし

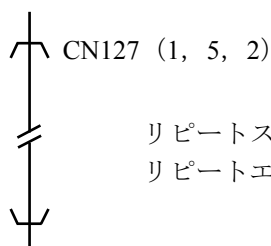
[リピートスタート (), リピートエンド () のプログラム例]

1.



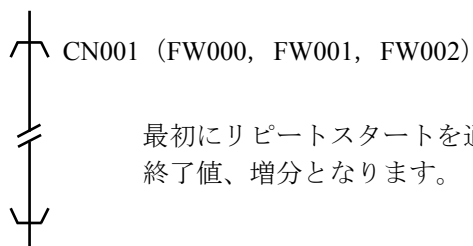
リピートスタートとリピートエンドの間を10回繰り返し
リピートエンドの次のステップへ進みます。
リピートエンドを実行後、すぐにリピートスタートを実行します。

2.



リピートスタートとリピートエンドの間を3回繰り返し
リピートエンドの次のステップへ進みます。

3.

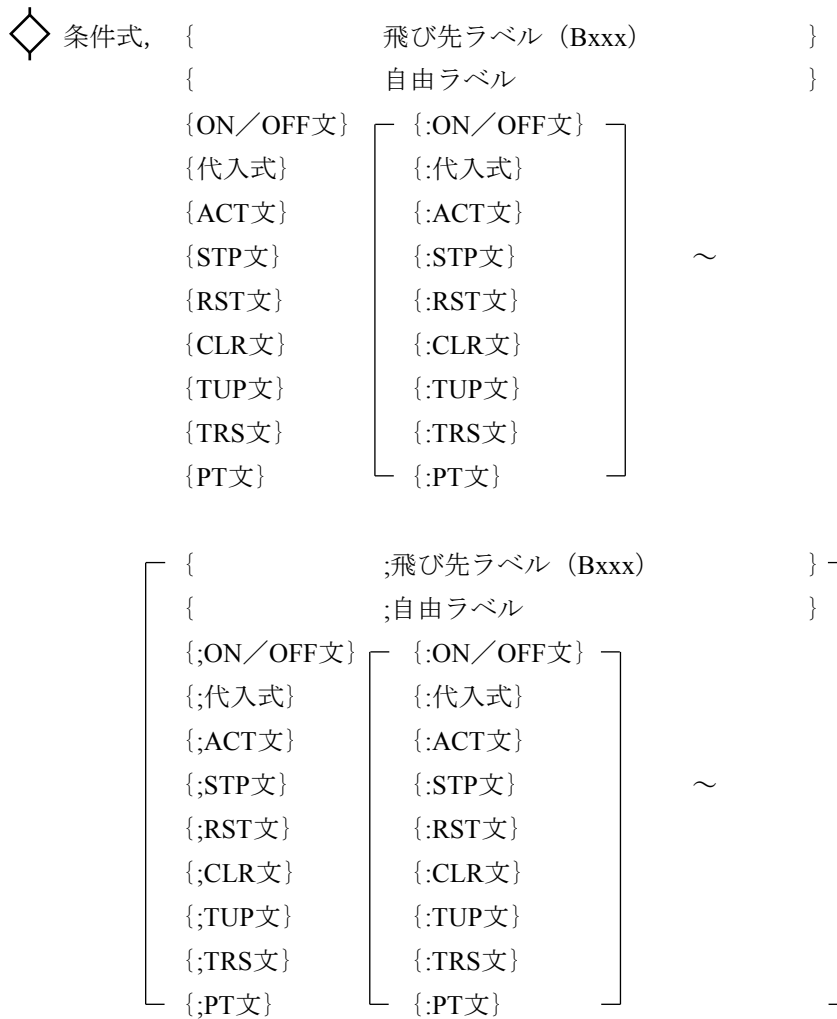


最初にリピートスタートを通ったときのFW000～FW002の値が初期値、
終了値、増分となります。

4.7 イフ

条件式の真偽を判定して、すぐにそれぞれに対応した処理を行います。
 条件式が成立した場合（真）、,（カンマ）と ;（セミコロン）までを実行します。
 条件が不成立の場合（偽）、;（セミコロン）以降を実行します。
 ;（セミコロン）以降を省略した場合で、条件式が不成立のときは次ステップへ進みます。
 ,（カンマ）、;（セミコロン）の後にラベルを指定するとそのラベルへ分岐します。

[構文]



(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

[イフ (◇) のプログラム例]

1. ◇ X0000, B1;LABEL

X0000がONのときはB1ラベルの存在するステップへジャンプし、OFFのときはLABELラベルの存在する箇所の次のステップへジャンプする。

2. ◇ H0<> (YW0000&H3000) , ON Q0005

YW0000の内容とH3000の論理積が0以外ならばQ0005をONし、0ならば何もしないで次ステップへ進む。

3. ◇ Q0000, FW100=FW100+1;ACT P10

Q0000がONのときはFW000に1を加算して次ステップへ進み、OFFのときはプロセス10にACT起動をかけ次ステップへ進む。

4. ◇ GW000=4, STP P6;RST P7;EW0000=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EW0000を8にし、J000をONして次ステップへ進む。

5. ◇ X0010, ON J000, J001, J002, J003;ERRLB

X0010がONのときJ000, J001, J002, J003をONし次ステップへ進む。

X0010がOFFのときERRLBラベルの存在する箇所の次のステップへジャンプする。

4. 8 ジャンプ

プロセス内の指定したラベルへ無条件で分岐します。

1プロセスにつきラベルはB1～B255まで指定できます。

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

[構文]

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

<留意事項>

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

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

[ジャンプ (↳) のプログラム例]

1. ↳ B1

B1ラベルの存在するステップへジャンプし、そのステップからすぐ実行する。

2. ↳ ERRBLK

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

4.9 エスケープ

自プロセスを強制終了させます。

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

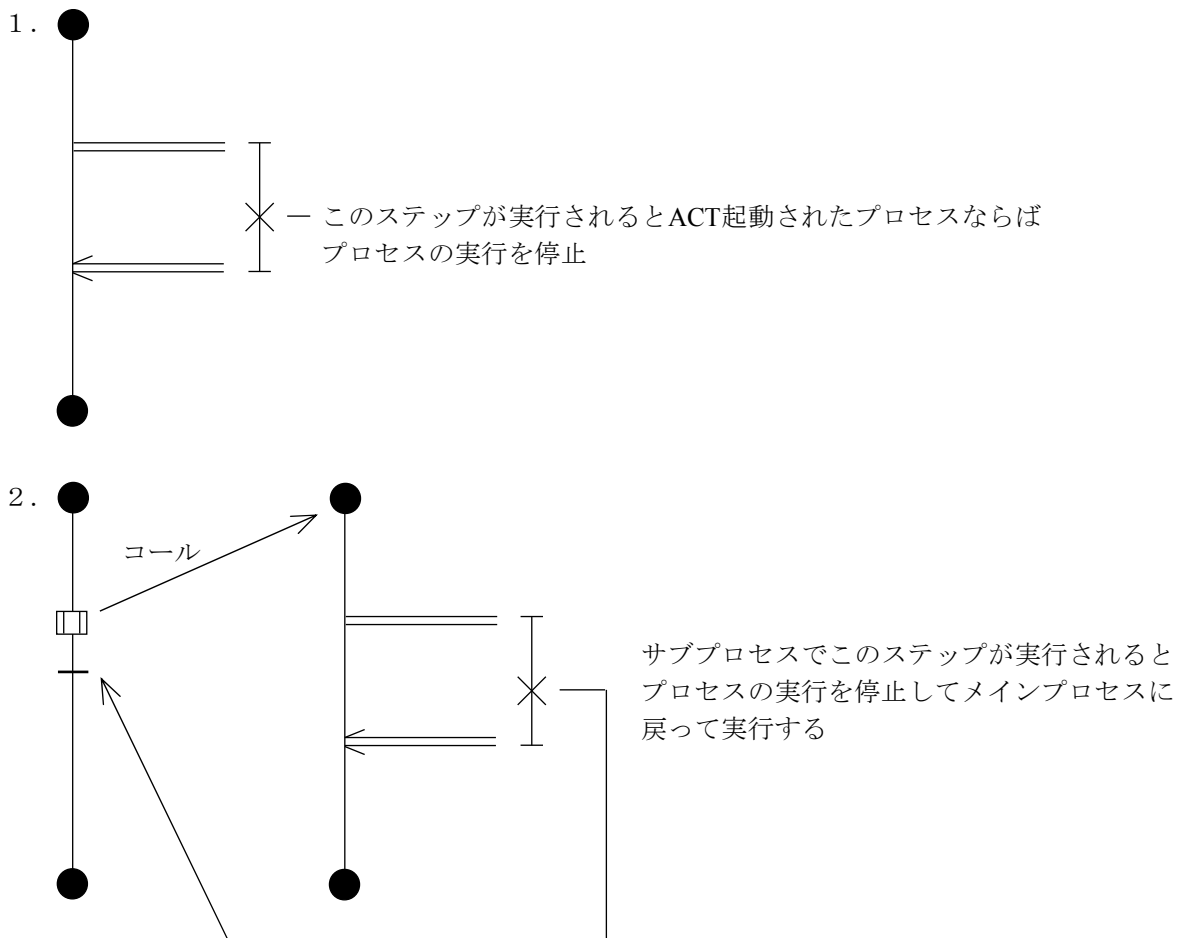
サブプロセスの場合も基本的にはメインプロセスと同じですが、実行箇所を同一スキャンでメインプロセスに戻します。

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

[構文]

✕ 構文なし

[エスケープ (✕) のプログラム例]



4. 10 パラスタート、パラエンド

パラスタート、パラエンド対で同期処理部分を表します。

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

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

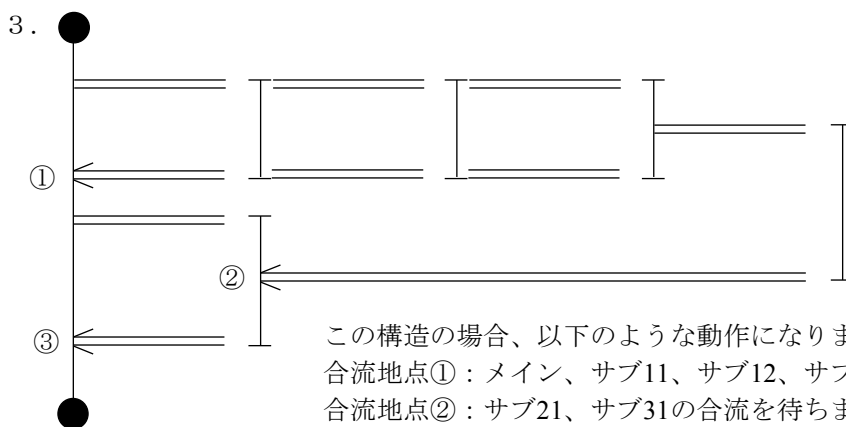
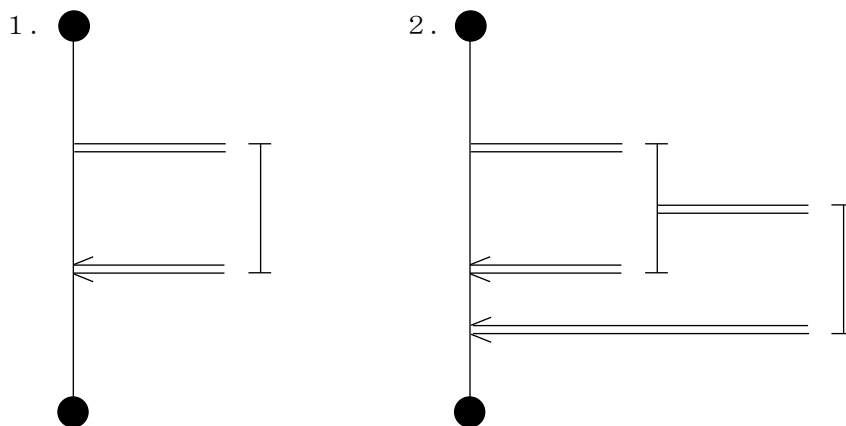
パラエンド、ルートエンドともに自分が最後に合流したかをチェックし、最後ならばメインルート合流部の次ステップ実行を、最後でなかったら自ルートの実行を終了（メインルートが常に実行中とは限らない）するため、スキャン遅れは起こりません。

[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]



この構造の場合、以下のような動作になります。
 合流地点①：メイン、サブ11、サブ12、サブ12の合流を待ちます。
 合流地点②：サブ21、サブ31の合流を待ちます。
 合流地点③：メイン、サブ21の合流を待ちます。

4. 11 パラスタート、セレクトエンド

パラスタート、セレクトエンド対で並列処理部分を表します。

パラスタートは複数の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。

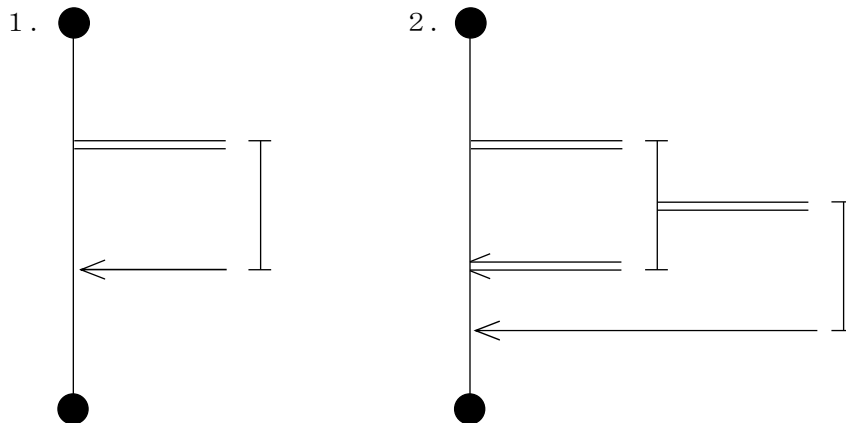
セレクトエンドは合流しているすべてのルートの内のいずれか1つのルートが終了すると、他の終了していないルートを強制終了させ、自ルートの次ステップを実行します。

[構文]

⊢ 構文なし

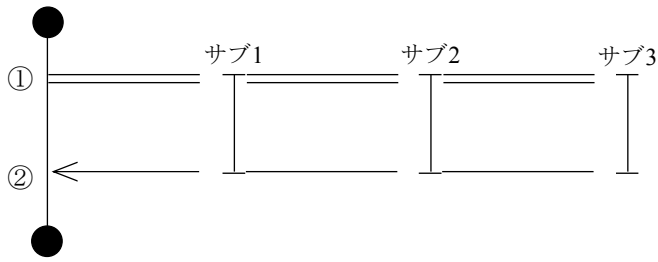
⊣ 構文なし

[パラスタート (⊢), セレクトエンド (⊣) のプログラム例]



[動作説明1]

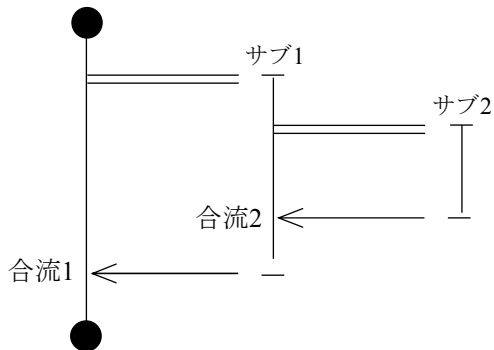
パラスタート+セレクトエンドのみの場合



- ・メインルートの①のステップを実行すると、メインルート/サブ1/サブ2/サブ3の4つのルートを並列実行します。
- ・メインルート/サブ1/サブ2/サブ3のいずれかのルート1つが終了（合流）すると、他のルートは強制終了し、メインルートの次のステップを実行します。

[動作説明2]

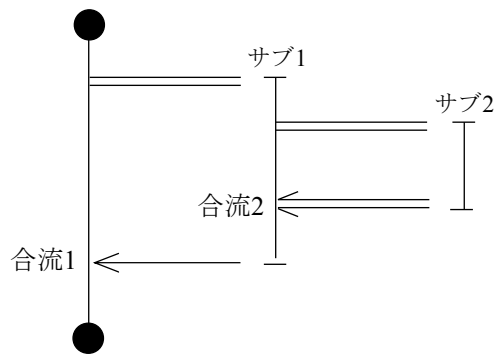
パラスタート+セレクトエンドの中にパラスタート+セレクトエンドがある場合



- ・合流1が先に到達（この場合メインルートが先に到達）した場合、サブ1とサブ2は強制終了します。
- ・合流2が先に到達した場合は、サブ1とサブ2の通常のパラスタート+セレクトエンドの動作となります。

[動作説明3]

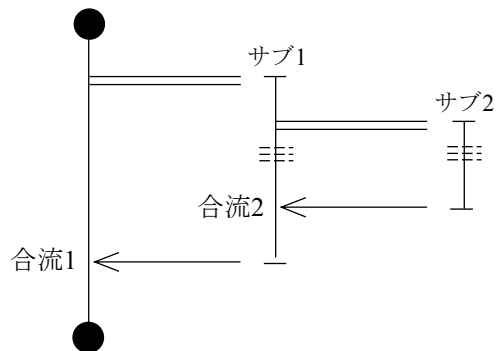
パラスタート+セレクトエンドの中にパラスタート+パラエンドがある場合



- ・合流1が先に到達（この場合メインルートが先に到達）した場合、サブ1とサブ2は強制終了します。
- ・合流2が先に到達した場合は、サブ1とサブ2の通常のパラスタート+パラエンドの動作となります。

[動作説明4]

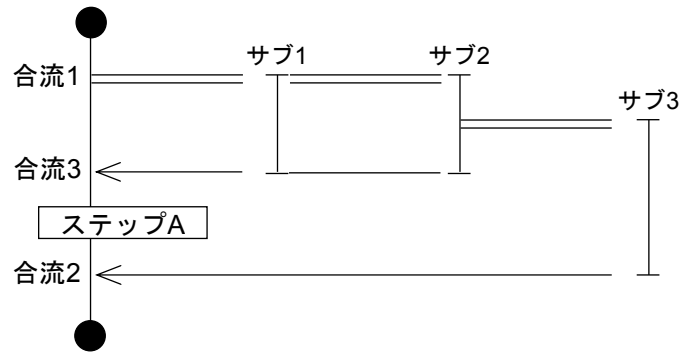
パラスタート+セレクトエンドの中にセレクト+セレクトエンドがある場合



- ・合流1が先に到達（この場合メインルートが先に到達）した場合、サブ1とサブ2は強制終了します。
- ・合流2が先に到達した場合は、サブ1とサブ2の通常のリセレクト+リセレクトエンドの動作となります。

通 知

下図のパターンのように、パラスタートしてセレクトエンドで合流する構文において、サブルートの中で再度パラスタートし、サブルートではなくメインルートにセレクトエンドで合流する構文は記述しないでください。合流地点に達した場合に、強制終了されるのは合流地点が同一のルートだけであるため、合流していないサブルートが実行中のまま残ってしまいます。



[動作説明]

- ・サブ1～3が実行中の状態で、サブ1、2が合流3に到達する前にサブ3が合流2に到達した場合、サブ1とサブ2の実行点は強制終了されず実行中のまま残ってしまいます。

4. 12 セレクト、セルウェイト、セレクトエンド

セレクト、セルウェイト、セレクトエンド一組で選択分岐処理部分を表します。

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

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

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

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、スキャン遅れは起こりません。

また、セレクトエンドとセレクトは同一ルートに存在しなくても（分岐元ルートに合流しなくても）構いません。

<留意事項>

セルウェイトはセレクトの次ステップになければいけません。

[構文]

┆ 構文なし

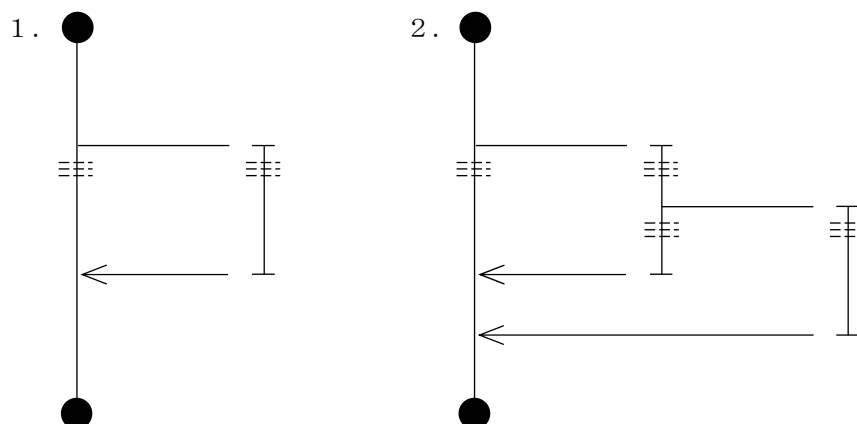
≡≡≡ 条件式 [, タイマー, 出力ビット]

⌋ 構文なし

- ・タイマー
- ・出力ビット

※タイマーと出力ビットについては、「4. 3 ウェイト」を参照してください。

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



4. 13 マルチエントリー

セレクトエンドと同じ図形に条件式を設定するとマルチエントリーとみなします。

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

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

マルチエントリーはサブルートにも設定できます。

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

[構文]

┌ 条件式

<留意事項>

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

[マルチエントリー（┌）のプログラム例]

1. ┌ X0000

X0000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. 14 コール

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

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

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

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

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

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

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

[構文]

```

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

```

[コール（□□）のプログラム例]

1. □□ P1

プロセス1をステップ1からゾーンコールします。

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

2. □□ P2, 5, MRST

プロセス2をステップ5からマスターリセットコールします。

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

3. □□ P3, TUP

プロセス3をステップ1からゾーンコールします。

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

4. 15 ファンクション

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

[構文]

\bigcirc 応用命令名称 パラメーター [, パラメーター] ~

4. 16 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 [, 条件式]) }

4. 17 非同期プロセスエンド

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

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

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

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

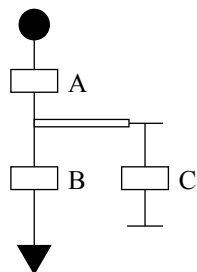
[構文]



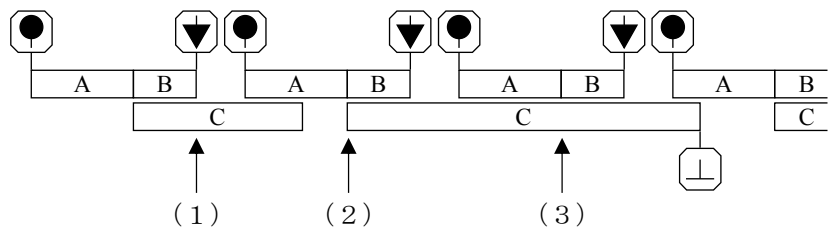
[非同期のプログラム例]

1. 非同期ルートが1つの場合

<回路例>



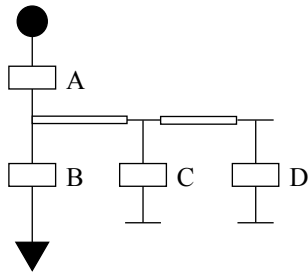
<タイムチャート>



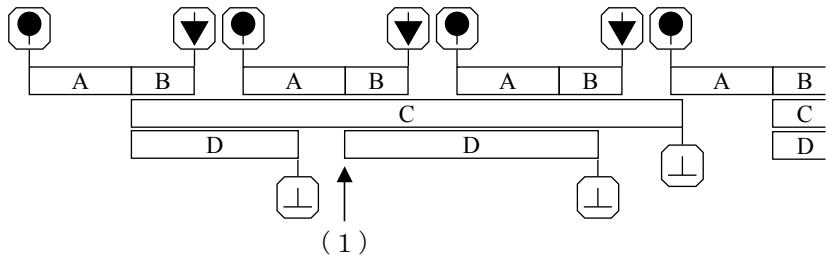
- (1) 非同期プロセスエンド到達時に非同期ルートが動作中の場合、非同期ルートの処理を継続したままプロセスを終了します。
- (2) 非同期ルートのルートスタート到達前に非同期ルートが動作完了した場合、非同期ルートを起動します。
- (3) 非同期ルートのルートスタート到達時に非同期ルートが動作中の場合、動作中の非同期ルートの処理を継続します。

2. 非同期ルートが複数の場合

<回路例>



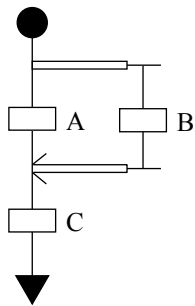
<タイムチャート>



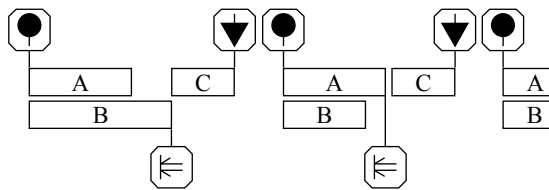
(1) 非同期ルートのルートスタート到達時、動作が完了した非同期ルートは起動され、動作中の非同期ルートは処理を継続します。

3. 非同期プロセスエンドでルートエンドが合流している場合

<回路例>



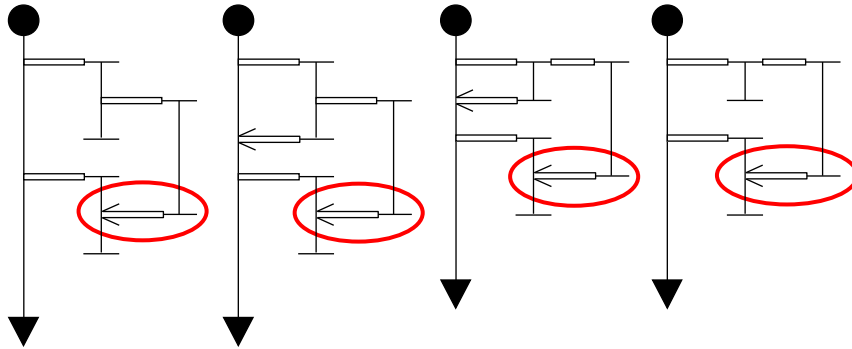
<タイムチャート>



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

通 知

- 下図のパターンのように、非同期プロセスの分岐と合流が異なり、合流ルートが非同期ルートである構文は記述しないでください。実行した場合、メインルートを新たに起動してもサブルートを実行中と判断して起動がかからない状態になります。



- 起動の際にマスターリセットを指定した場合、非同期ルートの実行中にビット型PI/Oがクリアされる可能性がありますので注意してください。

このページは白紙です。

第5章 応用命令

5.1 概説

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

5.2 使用方法

応用命令は次に示すような方法でプログラミングします。

○ 応用命令名称 パラメーター [, パラメーター] ~

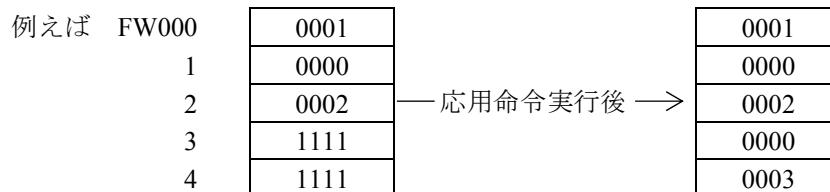
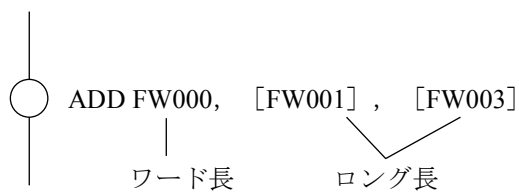
5.3 パラメーター

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

(ラダー)



(HI-FLOW)



第5章 応用命令

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

パラメーターはビット型PI/O、ワード型PI/O、定数の3種類があります。

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

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

アドレッシングモード	パラメーター																
	ビット型PI/O	ワード型PI/O	定数														
	X0000 <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> X0001 <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 YYYYYYYYYY XXXX <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>データe</td></tr><tr><td>データf</td></tr></table> YYYYYYYYYY <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>データg</td></tr><tr><td>データh</td></tr></table>	データe	データf	データg	データh
データ1																	
データ2																	
データ2																	
データa																	
データb																	
データ3																	
データ4																	
データ4																	
データc																	
データd																	
データe																	
データf																	
データg																	
データh																	
1. ダイレクトワード長	データ1と1の アンド結果	データ3	XXXX ただし ロング長YYYYYYYYYYは下 位ワードだけ														
例	X0000	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, YYYYYYYYYY ただし、XXXXはロング長 として扱います。														
例	[X0000]	[FW000]	[H1234] [H20000000]														
3. インダイレクトワード長	パラメーターエラー	データc ただしデータ3,4が奇数の 場合エラー	XXXXの場合 データe YYYYYYYYYYの場合 データg XXXX, YYYYYYYYYYが奇数 の場合エラー														
例		@FW000	@HFFF0 @H180000														
4. インダイレクトロング長	パラメーターエラー	データc, d ただしデータ3,4が奇数の 場合エラー	XXXXの場合 データe, f YYYYYYYYYYの場合 データg, h XXXX, YYYYYYYYYYが奇数 の場合エラー														
例		@[FW000]	@[HFFF0] @[H180000]														

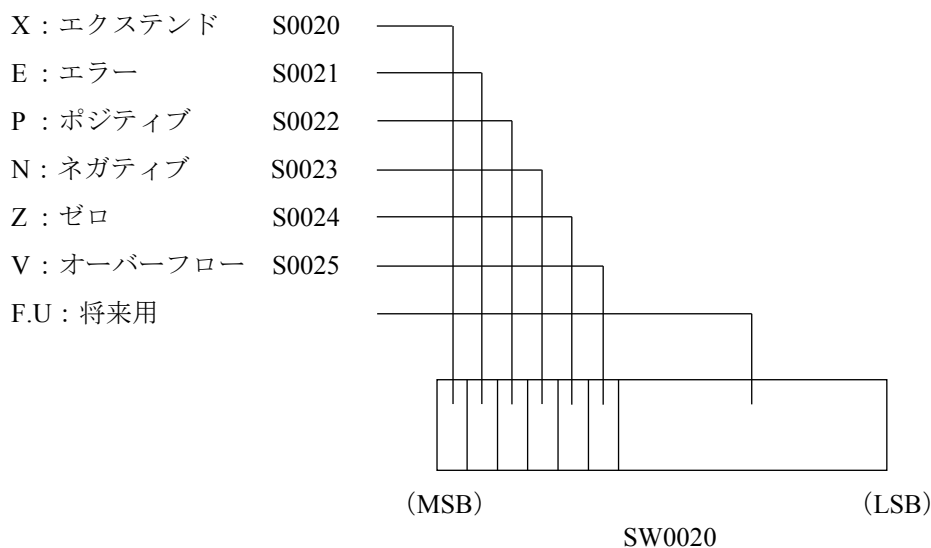
<留意事項>

インダイレクトロング長でPI/Oビットエリア（H0220 0000～H023F 0000, H0270 0000～H027F 0000）の範囲は、ロングワードでのアクセスが不可のため指定しないでください。

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SW0020に設定されます。

フラグ種類



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

- ・エラーフラグ・・・使用している応用命令のパラメーター数が異なる場合。

CPUにメモリープロテクトがかかっているとき、リザルト (R) で指定したアドレス、PI/Oがプロテクトエリア内を指していたとき。

指定されたPI/Oに異常があるとき (使用できないなど)。

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

ワード長 正のオーバーフロー H7FFF

負のオーバーフロー H8000

ワード長は、ビット型PI/Oが指定できますが、LSBのみ有効です。




LSB以外に設定された場合は、オーバーフローフラグが設定されません。

ロング長 正のオーバーフロー H7FFFFFFF



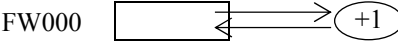

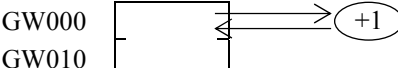
負のオーバーフロー H80000000



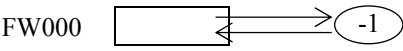

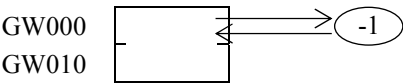
5.6 機能説明

この節では各応用命令の詳細について述べます。以下に示す形式で説明します。



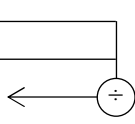

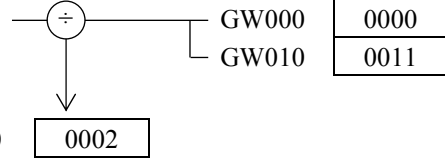
応用命令名称	機能名称																																									
ADD	加算	応用命令の処理概要を説明します。																																								
機能説明	ソースとデスティネーションの内容を加算してリザルトに格納します。																																									
パラメーターと処理内容	 ADD S, D, R S: ソース D: デスティネーション R: リザルト	S+D → R 処理内容を図示します。 パラメーターの並びを示します。																																								
フラグの設定	E, Vが変化、他はOFF																																									
備考																																										
使用例	 ADD FW000, FW001, FW002 FW000: 0001 FW001: 00FF FW002: 0100	注意点を示します。 主な使用例を表します。																																								
	 ADD H1234, [GW000], FW100 H1234: 0010 GW000: 0010 GW010: 0011 FW100: 7FFF	VフラグがONします。																																								
有効パラメーター	<table border="1"> <thead> <tr> <th>S, D</th> <th>ビット型 P/O</th> <th>ワード型 P/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 P/O	ワード型 P/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	<table border="1"> <thead> <tr> <th>R</th> <th>ビット型 P/O</th> <th>ワード型 P/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/O	ワード型 P/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 P/O	ワード型 P/O	定数																																							
ダイレクト ワード長	○	○	○																																							
ダイレクト ロング長	×	○	○																																							
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/O	ワード型 P/O	定数																																							
ダイレクト ワード長	○	○	×																																							
ダイレクト ロング長	×	○	×																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
△はアドレス値が奇数時はパラメーターエラー	指定不可 条件付き指定有効	指定有効																																								
対象のパラメーター種類 (ソース、デスティネーション、リザルト)																																										


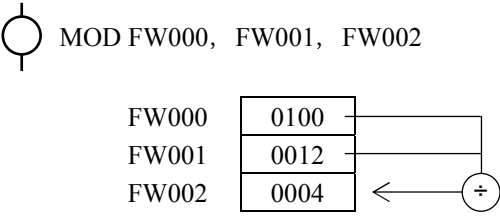
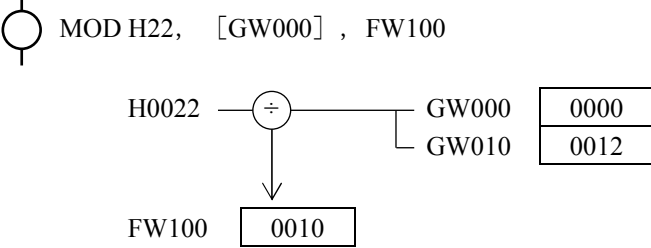
ADD	加算																																													
機能説明	ソースとデスティネーションの内容を加算してリザルトに格納します。																																													
パラメーターと処理内容	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: left;"> <p>○ ADD S, D, R</p> <p>S : ソース D : デスティネーション R : リザルト</p> </div> <div style="border: 1px solid black; padding: 5px; width: 150px;"> $S+D \rightarrow R$ </div> </div>																																													
フラグの設定	E, Vが変化、他はOFF																																													
備考																																														
使用例	<p>○ ADD FW000, FW001, FW002</p> <div style="display: flex; align-items: center; margin-left: 40px;"> <table border="1" style="border-collapse: collapse; text-align: center;"> <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 style="margin: 0 10px;"> </div> </div> <p>○ ADD H1234, [GW000], FW100</p> <div style="display: flex; align-items: center; margin-left: 40px;"> <div style="margin-right: 20px;"> <p>H1234</p> </div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>GW000</td><td>0010</td></tr> <tr><td>GW010</td><td>0011</td></tr> </table> </div> <div style="margin-left: 20px;"> <p>FW100</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>7FFF</td></tr> </table> <p>VフラグがONします。</p> </div>	FW000	0001	FW001	00FF	FW002	0100	GW000	0010	GW010	0011	7FFF																																		
FW000	0001																																													
FW001	00FF																																													
FW002	0100																																													
GW000	0010																																													
GW010	0011																																													
7FFF																																														
有効パラメーター	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th rowspan="2">S, D</th> <th>ビット型</th> <th>ワード型</th> <th rowspan="2">定数</th> <th rowspan="2">R</th> <th>ビット型</th> <th>ワード型</th> <th rowspan="2">定数</th> </tr> <tr> <th>PI/O</th> <th>PI/O</th> <th>PI/O</th> <th>PI/O</th> </tr> </thead> <tbody> <tr> <td rowspan="5">△はアドレス値が奇数時はパラメーターエラー</td> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>○</td> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型	ワード型	定数	R	ビット型	ワード型	定数	PI/O	PI/O	PI/O	PI/O	△はアドレス値が奇数時はパラメーターエラー	ダイレクトワード長	○	○	○	ダイレクトワード長	○	○	×	ダイレクトロング長	×	○	○	ダイレクトロング長	×	○	×	インダイレクトワード長	×	△	△	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型		ワード型	定数			R	ビット型		ワード型	定数																																			
	PI/O	PI/O	PI/O		PI/O																																									
△はアドレス値が奇数時はパラメーターエラー	ダイレクトワード長	○	○	○	ダイレクトワード長	○	○	×																																						
	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	×	△	△																																						

INC	+1 (インクリメント)																						
機能説明	ソースの内容に1を加算します。																						
パラメーターと処理内容	 INC S S : ソース	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> $S+1 \rightarrow S$ </div>																					
フラグの設定	E, Vが変化、他はOFF																						
備考																							
使用例	 INC FW000   INC [GW000]  GW000, GW010をロング変数と見なしてインクリメントします。																						
有効パラメーター	<table border="1" 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	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																				
ダイレクト ワード長	○	○	×																				
ダイレクト ロング長	×	○	×																				
インダイレクト ワード長	×	△	△																				
インダイレクト ロング長	×	△	△																				
△はアドレス値が奇数時はパラメーターエラー																							

DEC	-1 (デクリメント)																						
機能説明	ソースの内容から1を減算します。																						
パラメーターと処理内容	 DEC S S : ソース	<div style="border: 1px solid black; padding: 5px; display: inline-block;">S-1 → S</div>																					
フラグの設定	E, Vが変化、他はOFF																						
備考																							
使用例	 DEC FW000   DEC [GW000]  GW000, GW010をロング変数と見なしてデクリメントします。																						
有効パラメーター	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 25%;">S</th> <th style="width: 25%;">ビット型 PI/O</th> <th style="width: 25%;">ワード型 PI/O</th> <th style="width: 25%;">定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>			S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	×	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																				
ダイレクトワード長	○	○	×																				
ダイレクトロング長	×	○	×																				
インダイレクトワード長	×	△	△																				
インダイレクトロング長	×	△	△																				
△はアドレス値が奇数時はパラメーターエラー																							

MUL	乗算																																								
機能説明	ソースとデスティネーションの内容を乗算してリザルトに格納します。																																								
パラメーターと処理内容	<p>○ MUL S, D, R</p> <p style="text-align: right;">S×D → R</p> <p>S : ソース D : デスティネーション R : リザルト</p>																																								
フラグの設定	E, Vが変化、他はOFF																																								
備考																																									
使用例	<p>○ MUL FW000, FW001, FW002</p> <div style="display: flex; align-items: center;"> <table border="1" style="margin-right: 20px;"> <tr><td>FW000</td><td>0100</td></tr> <tr><td>FW001</td><td>00FF</td></tr> <tr><td>FW002</td><td>FF00</td></tr> </table> </div> <p>○ MUL H22, [GW000], FW100</p> <div style="display: flex; align-items: center;"> <table border="1" style="margin-right: 20px;"> <tr><td>H0022</td><td>7FFF</td></tr> </table> <table border="1" style="margin-left: 20px;"> <tr><td>GW000</td><td>0010</td></tr> <tr><td>GW010</td><td>0011</td></tr> </table> </div> <p>FW100 7FFF VフラグがONします。</p>	FW000	0100	FW001	00FF	FW002	FF00	H0022	7FFF	GW000	0010	GW010	0011																												
FW000	0100																																								
FW001	00FF																																								
FW002	FF00																																								
H0022	7FFF																																								
GW000	0010																																								
GW010	0011																																								
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	×	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	×	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	×	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	×																																						
ダイレクトロング長	×	○	×																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									


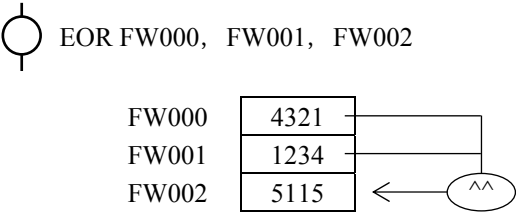
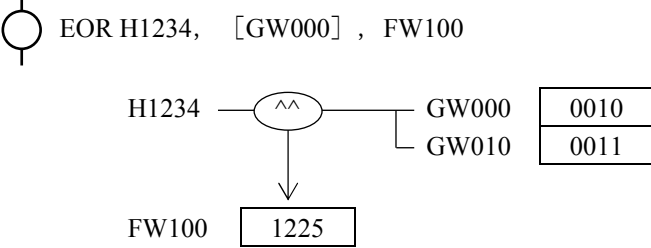
DIV	除算																																								
機能説明	ソースをデスティネーションの内容で除算して商をリザルトに格納します。																																								
パラメーターと処理内容	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;">  <p>DIV S, D, R</p> </div> <div style="border: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



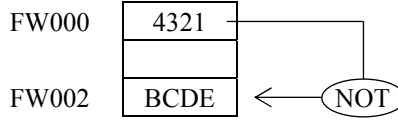

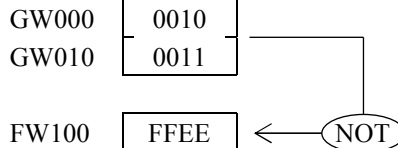
MOD	剰余																						
機能説明	ソースをデスティネーションの内容で除算して剰余をリザルトに格納します。																						
パラメーターと処理内容	 MOD S, D, R	$S \div D$ の剰余 → R																					
	S : ソース D : デスティネーション R : リザルト																						
フラグの設定	E, Vが変化、他はOFF																						
備考	D=0のとき、EフラグをONして何もしません。																						
使用例	 																						
有効パラメーター	<table border="1"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>			S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	×	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
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ーエラー	<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	×	△	△																				


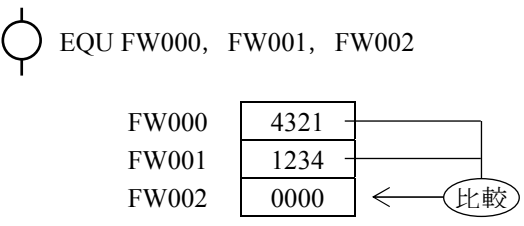
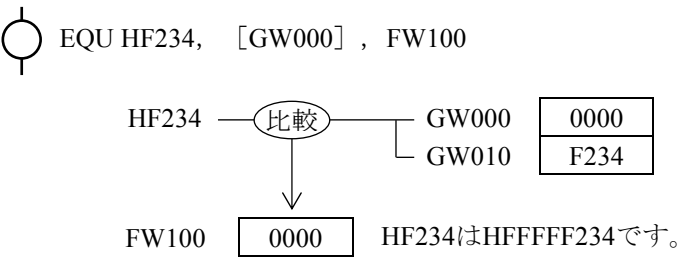
SCL	スケール変換																																								
機能説明	ソースをデスティネーションの内容でスケール変換してリザルトに格納します。																																								
パラメーターと処理内容	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: left;"> <p>○ SCL S, D1, D2, R</p> <p>S : ソース D1 : デスティネーション1 D2 : デスティネーション2 R : リザルト</p> </div> <div style="border: 1px solid black; padding: 5px; width: 150px;"> $S \times D1 \div D2 \rightarrow R$ </div> </div>																																								
フラグの設定	E, Vが変化、他はOFF																																								
備考	乗算オーバーフロー発生時リザルトにオーバーフロー値を書き込み終了します。 D2=0のとき、EフラグをONして何もしません。																																								
使用例	<p>○ SCL FW000, FW001, FW002, FW003</p> <div style="display: flex; align-items: center; margin-left: 40px;"> <table border="1" style="border-collapse: collapse;"> <tr><td>FW000</td><td>3320</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> <div style="margin-left: 10px;"> </div> </div> <p>○ SCL GW000, GW010, H1110, FW100</p> <div style="display: flex; align-items: center; margin-left: 40px;"> <table border="1" style="border-collapse: collapse;"> <tr><td>GW000</td><td>2222</td></tr> <tr><td>GW010</td><td>0012</td></tr> <tr><td>FW100</td><td>0024</td></tr> </table> <div style="margin-left: 10px;"> </div> </div>	FW000	3320	FW001	0010	FW002	0066	FW003	0805	GW000	2222	GW010	0012	FW100	0024																										
FW000	3320																																								
FW001	0010																																								
FW002	0066																																								
FW003	0805																																								
GW000	2222																																								
GW010	0012																																								
FW100	0024																																								
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, D1, D2</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</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ーエラー																																									


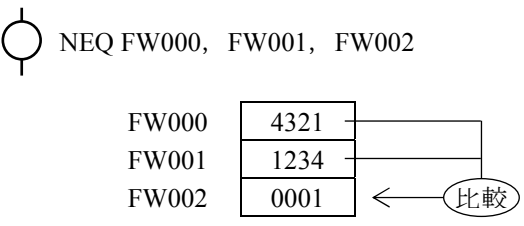
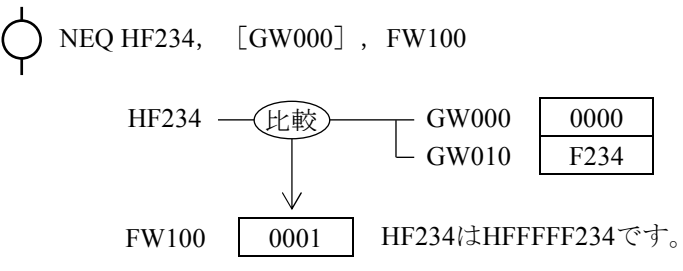
AND	論理積																																								
機能説明	ソースとデスティネーションの内容の論理積をリザルトに格納します。																																								
パラメーターと処理内容	<p>○ AND S, D, R</p> <p>S && D → R</p> <p>S : ソース D : デスティネーション R : リザルト</p>																																								
フラグの設定	Eが変化、他はOFF																												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


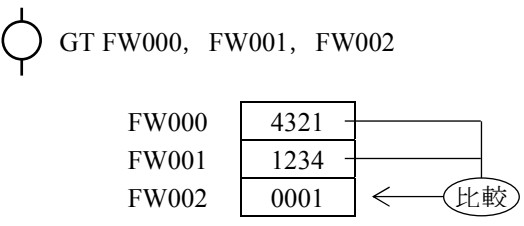
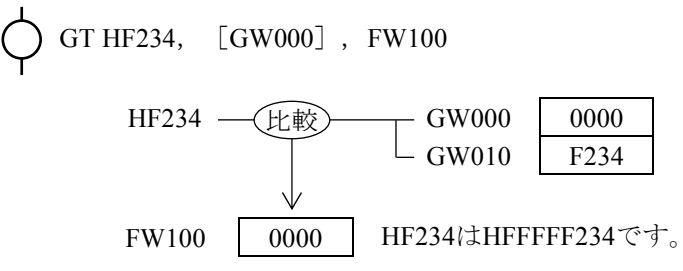
OR	論理和																																								
機能説明	ソースとデスティネーションの内容の論理和をリザルトに格納します。																																								
パラメーターと処理内容	<div style="display: flex; justify-content: space-between;"> <div style="width: 60%;"> <p>○ OR S, D, R</p> <p>S : ソース D : デスティネーション R : リザルト</p> </div> <div style="width: 35%; text-align: center;"> <p>S D → R</p> </div> </div>																																								
フラグの設定	Eが変化、他はOFF																																								
備考	Rがワード長の場合は演算結果の下位ワードが書き込まれます。																																								
使用例	<p>○ OR FW000, FW001, FW002</p> <div style="display: flex; align-items: center; margin-left: 40px;"> <table border="1" style="margin-right: 10px;"> <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> <p>○ OR H1234, [GW000], FW100</p> <div style="display: flex; align-items: center; margin-left: 40px;"> <table border="1" style="margin-right: 10px;"> <tr><td>H1234</td><td>1234</td></tr> </table> <table border="1" style="margin-left: 20px; margin-top: 10px;"> <tr><td>GW000</td><td>0010</td></tr> <tr><td>GW010</td><td>0011</td></tr> </table> </div> <div style="display: flex; align-items: center; margin-left: 40px;"> <table border="1" style="margin-right: 10px;"> <tr><td>FW100</td><td>1235</td></tr> </table> </div>	FW000	4321	FW001	1234	FW002	5335	H1234	1234	GW000	0010	GW010	0011	FW100	1235																										
FW000	4321																																								
FW001	1234																																								
FW002	5335																																								
H1234	1234																																								
GW000	0010																																								
GW010	0011																																								
FW100	1235																																								
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>ダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>インダイレクトワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>ダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>インダイレクトワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	×	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	×	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	×	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	×																																						
ダイレクトロング長	×	○	×																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									


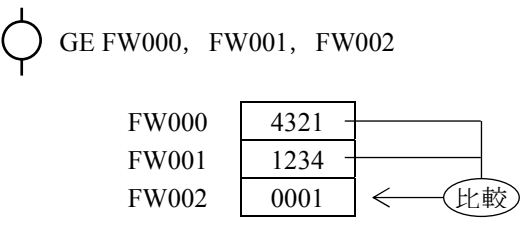
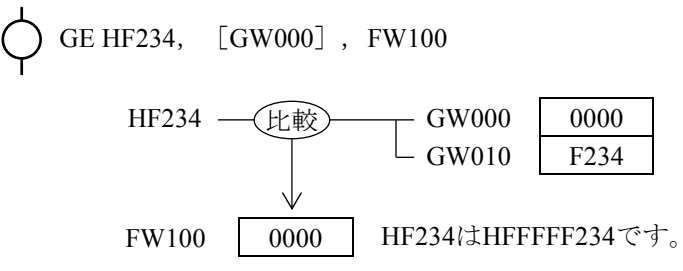
EOR	排他的論理和																																								
機能説明	ソースとデスティネーションの内容の排他的論理和をリザルトに格納します。																																								
パラメーターと処理内容	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;">  <p>EOR S, D, R</p> <p>S : ソース D : デスティネーション R : リザルト</p> </div> <div style="border: 1px solid black; padding: 5px; width: 150px;"> $S \wedge D \rightarrow R$ </div> </div>																																								
フラグの設定	Eが変化、他はOFF																																								
備考	Rがワード長の場合は演算結果の下位ワードが書き込まれます。																																								
使用例	<div style="margin-bottom: 20px;">  <p>FW000: 4321 FW001: 1234 FW002: 5115</p> </div> <div>  <p>H1234: 1234 GW000: 0010 GW010: 0011 FW100: 1225</p> </div>																																								
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>ダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>インダイレクトワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>ダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>インダイレクトワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	×	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	×	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	○																																						
ダイレクトロング長	×	○	○																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクトワード長	○	○	×																																						
ダイレクトロング長	×	○	×																																						
インダイレクトワード長	×	△	△																																						
インダイレクトロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									


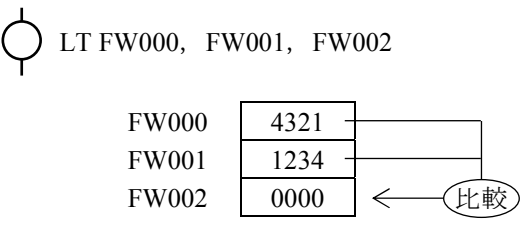
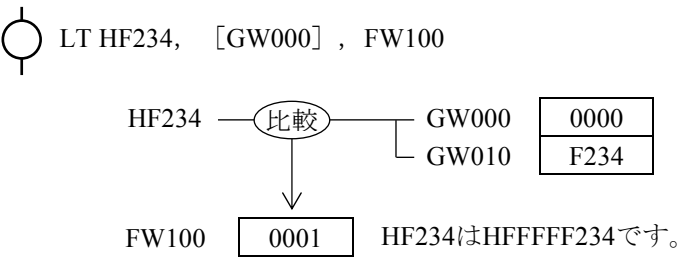
NOT	否定																																								
機能説明	ソースの内容の否定（ビット反転）をリザルトに格納します。																																								
パラメーターと処理内容	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="text-align: center;">  NOT S, R </div> <div style="border: 1px solid black; padding: 5px;"> S（ビット反転） → 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																																								
備考																																									
使用例	<div style="margin-bottom: 20px;">  NOT FW000, FW002  </div> <div>  NOT [GW000], FW100  </div>																																								
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクト ワード長	○	○	○																																						
ダイレクト ロング長	×	○	○																																						
インダイレクト ワード長	×	△	△																																						
インダイレクト ロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクト ワード長	○	○	×																																						
ダイレクト ロング長	×	○	×																																						
インダイレクト ワード長	×	△	△																																						
インダイレクト ロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									


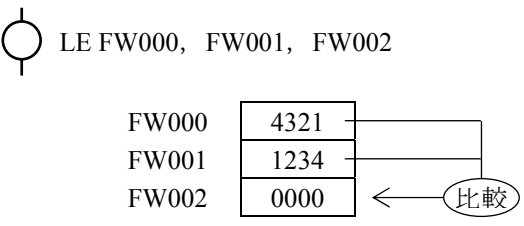
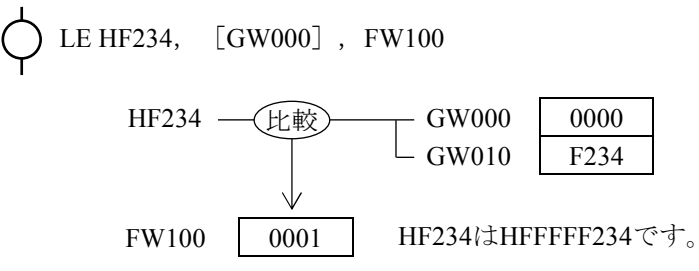
EQU	等しいか比較																																										
機能説明	ソースとデスティネーションの内容を比較し、等しければ1、それ以外は0をリザルトに格納します。																																										
パラメーターと処理内容	 EQU S, D, R S : ソース D : デスティネーション R : リザルト	<table border="1"> <tr> <td>S = D</td> <td>1</td> <td>→</td> <td>R</td> </tr> <tr> <td>S ≠ D</td>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



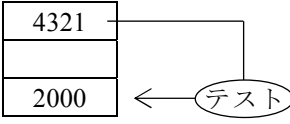

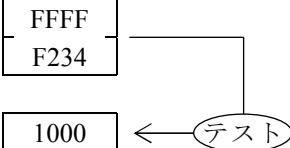
NEQ	等しくないか比較																																										
機能説明	ソースとデスティネーションの内容を比較し、等しくなければ1、それ以外は0をリザルトに格納します。																																										
パラメーターと処理内容	 NEQ S, D, R S : ソース D : デスティネーション R : リザルト	$S \neq D \quad 1 \rightarrow R$ $S = D \quad 0 \rightarrow R$																															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



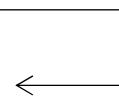

GT	より大きい比較																																									
機能説明	ソースとデスティネーションの内容を比較し、ソースが大きければ1、それ以外は0をリザルトに格納します。																																									
パラメーターと処理内容	 GT S, D, R S : ソース D : デスティネーション R : リザルト	<table border="1"> <tr> <td>$S > D$</td> <td>1</td> <td>→</td> <td>R</td> </tr> <tr> <td>$S \leq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



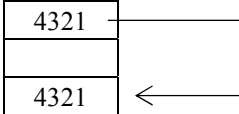

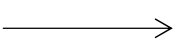
GE	より以上か比較																																										
機能説明	ソースとデスティネーションの内容を比較し、ソースが等しいか大きければ1、それ以外は0をリザルトに格納します。																																										
パラメーターと処理内容	 GE S, D, R S : ソース D : デスティネーション R : リザルト	<table border="1"> <tr> <td>$S \geq D$</td> <td>1</td> <td>→</td> <td>R</td> </tr> <tr> <td>$S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




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


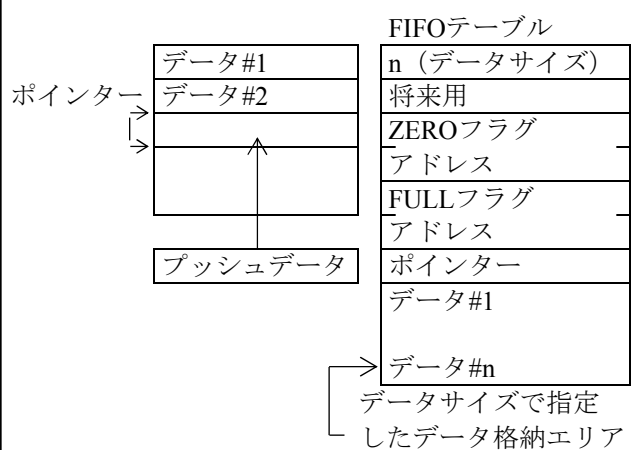

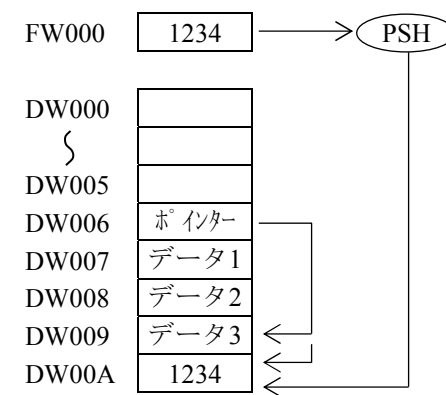
LE	より以下か比較																																									
機能説明	ソースとデスティネーションの内容を比較し、ソースが等しいか小さければ1、それ以外は0をリザルトに格納します。																																									
パラメーターと処理内容	 LE S, D, R S : ソース D : デスティネーション R : リザルト	<table border="1"> <tr> <td>$S \leq D$</td> <td>1</td> <td>→ R</td> </tr> <tr> <td>$S > D$</td>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


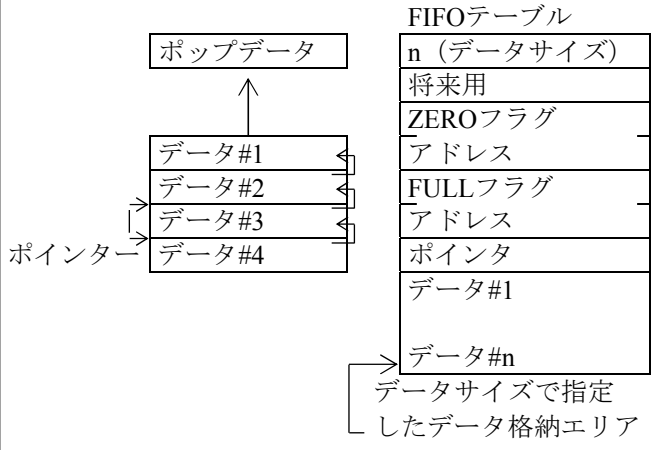
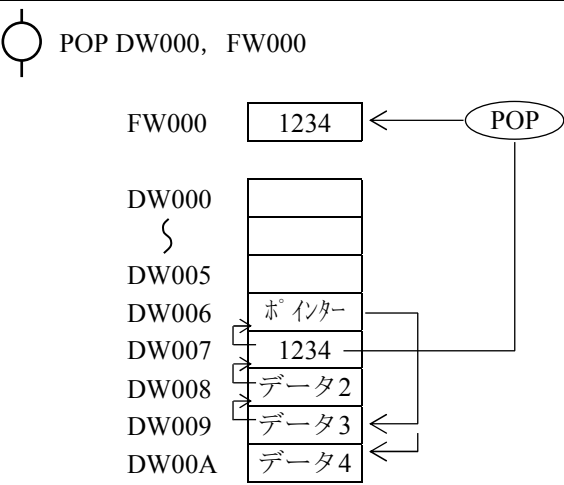
TST	符号テスト																				
機能説明	ソースの内容をテストし、P, Z, Nのフラグを設定します。																				
パラメーターと処理内容	 TST S S : ソース <div style="border: 1px solid black; padding: 5px; display: inline-block; margin-left: 200px;"> S > 0 : P=1, Z=0, N=0 S = 0 : P=0, Z=1, N=0 S < 0 : P=0, Z=0, N=1 </div>																				
フラグの設定	E, P, Z, Nが変化、他はOFF																				
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;"> <table border="1" style="border-collapse: collapse;"> <tr><td style="padding: 2px;">FW000</td><td style="padding: 2px;">4321</td></tr> <tr><td style="padding: 2px;">SW0020</td><td style="padding: 2px;">2000</td></tr> </table> <div style="margin-left: 20px;">  </div> </div>  TST [GW000] <div style="margin-left: 100px;"> <table border="1" style="border-collapse: collapse;"> <tr><td style="padding: 2px;">GW000</td><td style="padding: 2px;">FFFF</td></tr> <tr><td style="padding: 2px;">GW010</td><td style="padding: 2px;">F234</td></tr> </table> <div style="margin-left: 20px;">  </div> </div>	FW000	4321	SW0020	2000	GW000	FFFF	GW010	F234												
FW000	4321																				
SW0020	2000																				
GW000	FFFF																				
GW010	F234																				
有効パラメーター	<table border="1" style="border-collapse: collapse; width: 100%;"> <thead> <tr> <th style="width: 20%;">S</th> <th style="width: 20%;">ビット型 PI/O</th> <th style="width: 20%;">ワード型 PI/O</th> <th style="width: 40%;">定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table> <p>△はアドレス値が奇数時はパラメーターエラー</p>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																		
ダイレクト ワード長	○	○	○																		
ダイレクト ロング長	×	○	○																		
インダイレクト ワード長	×	△	△																		
インダイレクト ロング長	×	△	△																		



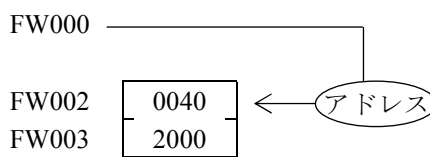

MOV	転送																																								
機能説明	ソースの内容をデスティネーションに転送します。																																								
パラメーターと処理内容	 MOV S, D <table border="1" style="float: right; margin-left: 20px;"> <tr> <td>S → D</td> </tr> </table> <p>S : ソース D : デスティネーション</p>	S → D																																							
S → D																																									
フラグの設定	Eが変化、他はOFF																																								
備考	転送するサイズが異なる場合は型変換します。																																								
使用例	 MOV FW000, FW002 <div style="margin-left: 40px;"> <table border="1" style="display: inline-table; margin-right: 10px;"> <tr><td>FW000</td><td>4321</td></tr> <tr><td> </td><td> </td></tr> <tr><td>FW002</td><td>4321</td></tr> </table>  </div>  MOV HF234, @ [H400A00] <div style="margin-left: 40px;"> <table border="1" style="display: inline-table; margin-right: 10px;"> <tr><td>HF234</td><td>→</td><td>H400A00</td><td><table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>FFFF</td></tr> <tr><td>F234</td></tr> </table></td></tr> <tr><td> </td><td> </td><td>2</td><td> </td></tr> </table> </div> <p style="text-align: center;">HF234はHFFFFFF234です。</p>	FW000	4321			FW002	4321	HF234	→	H400A00	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>FFFF</td></tr> <tr><td>F234</td></tr> </table>	FFFF	F234			2																									
FW000	4321																																								
FW002	4321																																								
HF234	→	H400A00	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>FFFF</td></tr> <tr><td>F234</td></tr> </table>	FFFF	F234																																				
FFFF																																									
F234																																									
		2																																							
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクト ワード長	○	○	○																																						
ダイレクト ロング長	×	○	○																																						
インダイレクト ワード長	×	△	△																																						
インダイレクト ロング長	×	△	△																																						
D	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクト ワード長	○	○	×																																						
ダイレクト ロング長	×	○	×																																						
インダイレクト ワード長	×	△	△																																						
インダイレクト ロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									


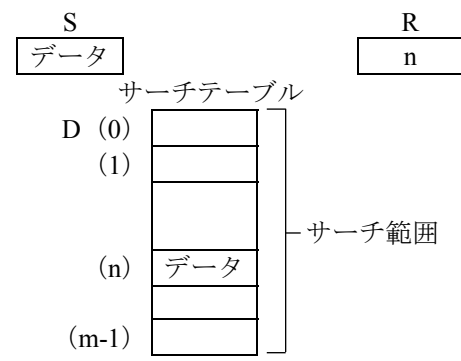
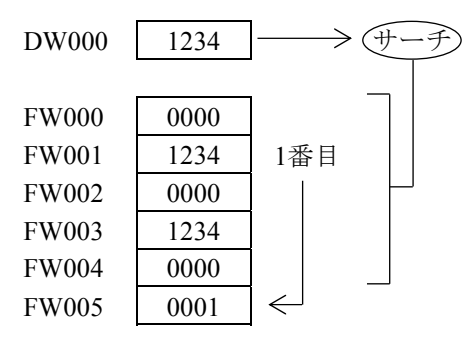
MOM	一括転送																																											
機能説明	ソースの内容をn要素分（ワード、ロング）一括でデスティネーションに転送します。																																											
パラメーターと処理内容	 MOM S, n, D S : ソース D : デスティネーション n : 転送要素数		S1 → D1 Sn → Dn																																									
フラグの設定	E, Vが変化、他はOFF																																											
備考	n ≤ 0、n > 256のとき未処理となります。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" style="display: inline-table; margin-right: 20px;"> <tr><td>FW000</td><td>4321</td></tr> <tr><td>FW002</td><td>4321</td></tr> </table>   MOM FW000, 2, @ [H400A00] <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>FW000</td><td>F234</td></tr> <tr><td>FW001</td><td>0001</td></tr> <tr><td>FW002</td><td>0000</td></tr> <tr><td>FW003</td><td>FFFF</td></tr> </table> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>H400A00</td><td>FFFF</td></tr> <tr><td>2</td><td>F234</td></tr> <tr><td>4</td><td>0000</td></tr> <tr><td>6</td><td>0001</td></tr> </table>  HF234はHFFFFFF234、H0001はH00000001です。				FW000	4321	FW002	4321	FW000	F234	FW001	0001	FW002	0000	FW003	FFFF	H400A00	FFFF	2	F234	4	0000	6	0001																				
FW000	4321																																											
FW002	4321																																											
FW000	F234																																											
FW001	0001																																											
FW002	0000																																											
FW003	FFFF																																											
H400A00	FFFF																																											
2	F234																																											
4	0000																																											
6	0001																																											
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	×	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	×	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクトワード長	○	○	○																																									
ダイレクトロング長	×	○	○																																									
インダイレクトワード長	×	△	△																																									
インダイレクトロング長	×	△	△																																									
D	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクトワード長	○	○	×																																									
ダイレクトロング長	×	○	×																																									
インダイレクトワード長	×	△	△																																									
インダイレクトロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												


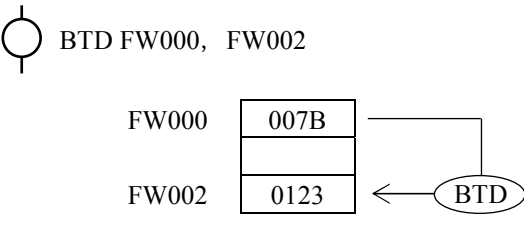
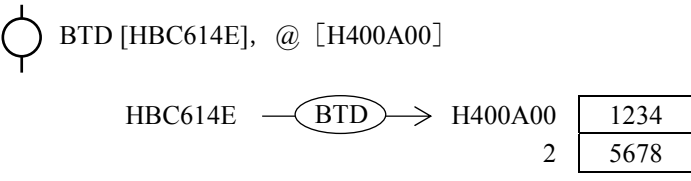
EXC	交換																															
機能説明	ソースの内容とデスティネーションを交換します。																															
パラメーターと処理内容	 EXC S, D <div style="border: 1px solid black; padding: 5px; display: inline-block; margin-left: 20px;">S ↔ D</div> <p>S : ソース D : デスティネーション</p>																															
フラグの設定	E, Vが変化、他はOFF																															
備考	転送するサイズが異なる場合、型変換して交換します。																															
使用例	 EXC FW000, FW002 <div style="margin-left: 20px;"> <table style="border-collapse: collapse;"> <tr> <td style="padding-right: 10px;">FW000</td> <td style="border: 1px solid black; padding: 2px 10px;">1234</td> <td style="padding: 0 5px;">←</td> <td rowspan="2" style="border: 1px solid black; width: 20px; height: 20px;"></td> </tr> <tr> <td></td> <td style="border: 1px solid black; padding: 2px 10px;"></td> <td></td> </tr> <tr> <td style="padding-right: 10px;">FW002</td> <td style="border: 1px solid black; padding: 2px 10px;">4321</td> <td style="padding: 0 5px;">←</td> <td></td> </tr> </table> </div>  EXC @H400B00, @[H400A00] <div style="margin-left: 20px;"> <table style="border-collapse: collapse; margin-bottom: 10px;"> <tr> <td style="padding-right: 10px;">H400B00</td> <td style="border: 1px solid black; padding: 2px 10px;">F234</td> <td style="padding: 0 10px;">↔</td> <td style="padding-right: 10px;">H400A00</td> <td style="border: 1px solid black; padding: 2px 10px;">0010</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="text-align: center;">2</td> <td style="border: 1px solid black; padding: 2px 10px;">0001</td> </tr> </table> <p>交換後</p> <table style="border-collapse: collapse;"> <tr> <td style="padding-right: 10px;">H400B00</td> <td style="border: 1px solid black; padding: 2px 10px;">7FFF</td> <td style="padding: 0 20px;"></td> <td style="padding-right: 10px;">H400A00</td> <td style="border: 1px solid black; padding: 2px 10px;">FFFF</td> </tr> <tr> <td></td> <td></td> <td></td> <td style="text-align: center;">2</td> <td style="border: 1px solid black; padding: 2px 10px;">F234</td> </tr> </table> </div>	FW000	1234	←					FW002	4321	←		H400B00	F234	↔	H400A00	0010				2	0001	H400B00	7FFF		H400A00	FFFF				2	F234
FW000	1234	←																														
FW002	4321	←																														
H400B00	F234	↔	H400A00	0010																												
			2	0001																												
H400B00	7FFF		H400A00	FFFF																												
			2	F234																												
有効パラメーター	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 20%;">S, D</th> <th style="width: 15%;">ビット型 PI/O</th> <th style="width: 15%;">ワード型 PI/O</th> <th style="width: 10%;">定数</th> </tr> </thead> <tbody> <tr> <td>△はアドレス 値が奇数時は パラメーター エラー</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	△はアドレス 値が奇数時は パラメーター エラー	○	○	×	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△							
S, D	ビット型 PI/O	ワード型 PI/O	定数																													
△はアドレス 値が奇数時は パラメーター エラー	○	○	×																													
ダイレクト ワード長	○	○	×																													
ダイレクト ロング長	×	○	×																													
インダイレクト ワード長	×	△	△																													
インダイレクト ロング長	×	△	△																													






PSH	FIFO書き込み																																								
機能説明	ソースの内容をFIFOテーブルへプッシュします。FIFOテーブルのデータ長はワードだけです。																																								
パラメーターと処理内容	 PSH S, TB S : ソース TB : FIFOテーブル先頭アドレス																																								
フラグの設定	E, Vが変化、他はOFF																																								
備考	<p>$n \leq 0$、$n > 256$のとき未処理となります。ポインター< 0またはデータサイズ$<$ポインターのとき未処理となります。ポインター$=$データサイズのときFULLフラグONし未処理となります。プッシュ後、ポインターをインクリメントしてnになった場合、FULLフラグONになります。それ以外はZEROフラグOFF、FULLフラグOFFになります。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>  <p>FIFOテーブル</p> <ul style="list-style-type: none"> n (データサイズ) 将来用 ZEROフラグ アドレス FULLフラグ アドレス ポインター データ#1 ... データ#n <p>データ格納エリアはデータサイズで指定した領域です。</p>																																								
使用例	 PSH FW000, DW000  <p>FW000: 1234 → PSH</p> <p>DW000: []</p> <p>DW005: []</p> <p>DW006: ポインター</p> <p>DW007: データ1</p> <p>DW008: データ2</p> <p>DW009: データ3</p> <p>DW00A: 1234</p> <p>動作: PSHはFW000の値(1234)をポインター(DW006)の指す位置にプッシュし、ポインターはデータ1(DW007)の位置に移動し、データ2(DW008)とデータ3(DW009)は既に存在する。</p>																																								
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>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	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									


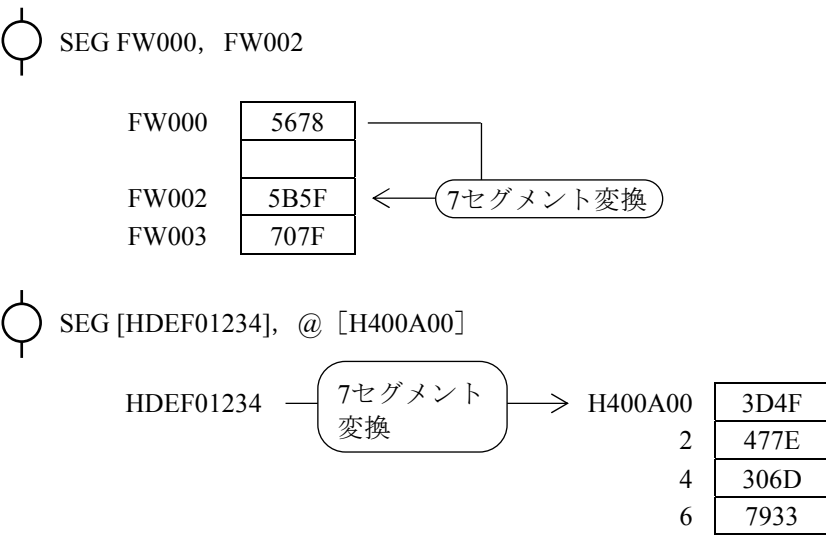

POP	FIFO読み出し																																									
機能説明	FIFOテーブルのポップ処理を行い、ポップデータをデスティネーションに格納します。 FIFOテーブルのデータ長はワードだけです。																																									
パラメーターと処理内容	 POP TB, D D : デスティネーション TB : FIFOテーブル先頭アドレス																																									
フラグの設定	Eが変化、他はOFF																																									
備考	<p>$n \leq 0$、$n > 256$のとき未処理となります。ポインタ< 0またはデータサイズ$<$ポインタのとき未処理となります。ポインタ$= 0$のときZEROフラグONし未処理となります。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になります。それ以外はZEROフラグOFF、FULLフラグOFFになります。TBが定数の場合はそれをテーブルアドレスと見なします。</p>																																									
使用例																																										
有効パラメーター	<table border="1"> <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>	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	×	△	△	<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>	D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	×	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
TB	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクトワード長	×	○	△																																							
ダイレクトロング長	×	○	△																																							
インダイレクトワード長	×	△	△																																							
インダイレクトロング長	×	△	△																																							
D	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクトワード長	○	○	×																																							
ダイレクトロング長	×	○	×																																							
インダイレクトワード長	×	△	△																																							
インダイレクトロング長	×	△	△																																							
△はアドレス値が奇数時はパラメーターエラー																																										



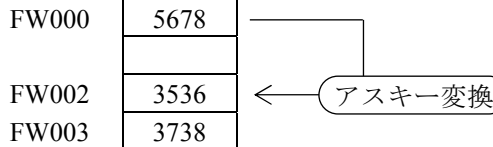

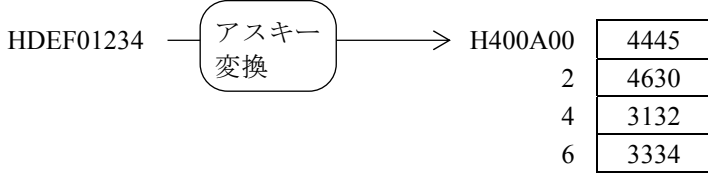
AST	アドレスセット																																											
機能説明	ソースのアドレスデータをデスティネーションに転送します。 PI/Oだけ有効です。																																											
パラメーターと処理内容	 AST S, D	<div style="border: 1px solid black; padding: 5px; display: inline-block;">Sのアドレス → D</div>																																										
	S : ソース D : デスティネーション																																											
フラグの設定	Eが変化、他はOFF																																											
備考																																												
使用例	 AST FW000, [FW002] <div style="margin-left: 40px;">  </div>  AST X0000, @ [H400A00] <div style="margin-left: 40px;"> X0000の → H400A00 <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>0070</td></tr> <tr><td>0000</td></tr> </table> アドレス 2 </div>				0070	0000																																						
0070																																												
0000																																												
有効パラメーター	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th style="width: 15%;">S</th> <th style="width: 15%;">ビット型 PI/O</th> <th style="width: 15%;">ワード型 PI/O</th> <th style="width: 15%;">定数</th> <th style="width: 15%;">D</th> <th style="width: 15%;">ビット型 PI/O</th> <th style="width: 15%;">ワード型 PI/O</th> <th style="width: 15%;">定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> <td>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ーエラー																																												




SCH	サーチ																																										
機能説明	ソースの内容がデスティネーションから指定エリア (m) 内に存在するか探し、デスティネーションからのステップ数 (n) をリザルトに格納します。																																										
パラメーターと処理内容	 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																																										
フラグの設定	Eが変化、他はOFF																																										
備考	m ≤ 0、m > 256のとき未処理となります。一致データは最初にみつけたものとなります。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。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。nは0から始まります。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																										
使用例																																											
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, 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>×</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, 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, m	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	×	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	D, R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	×	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, m	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクトワード長	○	○	○																																								
ダイレクトロング長	×	○	○																																								
インダイレクトワード長	×	△	△																																								
インダイレクトロング長	×	△	△																																								
D, R	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクトワード長	○	○	×																																								
ダイレクトロング長	×	○	×																																								
インダイレクトワード長	×	△	△																																								
インダイレクトロング長	×	△	△																																								
△はアドレス値が奇数時はパラメーターエラー																																											


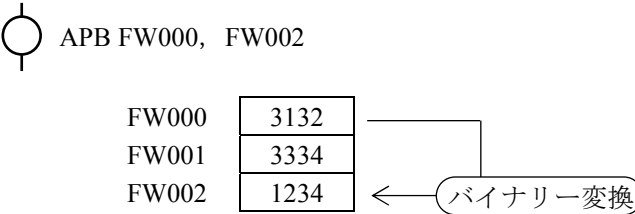
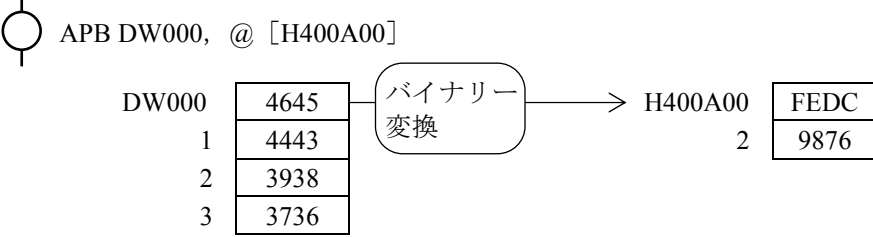
BTD	バイナリー→BCD変換																																											
機能説明	ソースの内容をバイナリーからBCDに変換し、リザルトに格納します。																																											
パラメーターと処理内容	 BTD S, R		S (バイナリー) → R (BCD)																																									
	S : ソース R : リザルト																																											
フラグの設定	E, Vが変化、他はOFF																																											
備考	S<0のときEフラグON、VフラグOFFし、未処理となります。 オーバーフロー時H9999、またはH99999999を設定します。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																											
使用例	 <p>FW000 007B</p> <p>FW002 0123</p> <p>← (BTD)</p>																																											
	 <p>BTD [HBC614E], @[H400A00]</p> <p>HBC614E → (BTD) → H400A00</p> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>1234</td></tr> <tr><td>5678</td></tr> </table>				1234	5678																																						
1234																																												
5678																																												
有効パラメーター	<table border="1" style="display: inline-table; vertical-align: top; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table; vertical-align: top;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	○																																									
ダイレクト ロング長	×	○	○																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	×																																									
ダイレクト ロング長	×	○	×																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												


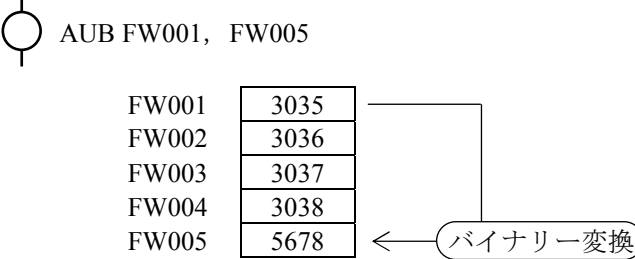
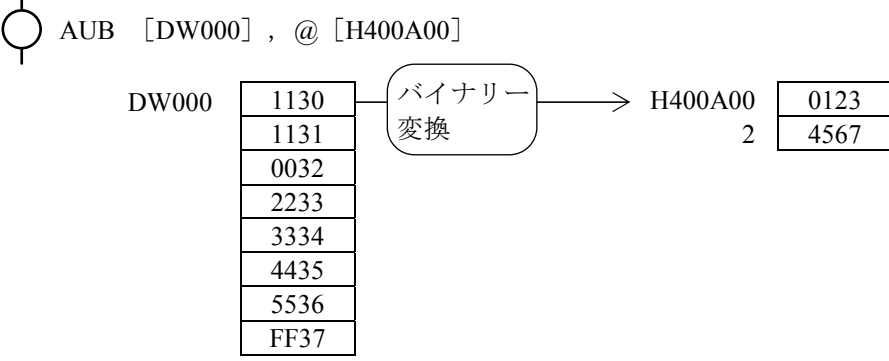
DTB	BCD→バイナリー変換																																											
機能説明	ソースの内容をBCDからバイナリーに変換し、リザルトに格納します。																																											
パラメーターと処理内容	 DTB S, R		S (BCD) → R (バイナリー)																																									
	S : ソース R : リザルト																																											
フラグの設定	E, Vが変化、他はOFF																																											
備考	SのなかでA~Fが使用されているとEフラグONして未処理となります。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																											
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>  ←				FW000	1234	FW002	04D2																																				
FW000	1234																																											
FW002	04D2																																											
	 DTB [H99999999], @ [H400A00] <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>H99999999</td><td>→</td><td>H400A00</td></tr> </table>  <table border="1" style="display: inline-table; margin-left: 20px;"> <tr><td>05F5</td></tr> <tr><td>E0FF</td></tr> </table>				H99999999	→	H400A00	05F5	E0FF																																			
H99999999	→	H400A00																																										
05F5																																												
E0FF																																												
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	○																																									
ダイレクト ロング長	×	○	○																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	×																																									
ダイレクト ロング長	×	○	×																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												



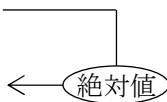


SEG	バイナリー→7セグメント変換																																											
機能説明	ソースの内容をバイナリーから7セグメントデータに変換し、リザルトに格納します。																																											
パラメーターと処理内容	 SEG S, R S : ソース R : リザルト		S (バイナリー) → R (7セグメントデータ)																																									
フラグの設定	Eが変化、他はOFF																																											
備考	Sのサイズ×2がRに書き込まれます。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																											
使用例	 <p>  SEG [HDEF01234], @ [H400A00] </p> <p>[7セグメント対応表]</p> <table border="1"> <thead> <tr> <th>No.</th> <th>0</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> <th>7</th> <th>8</th> <th>9</th> <th>A</th> <th>B</th> <th>C</th> <th>D</th> <th>E</th> <th>F</th> </tr> </thead> <tbody> <tr> <td>データ</td> <td>7E</td> <td>30</td> <td>6D</td> <td>79</td> <td>33</td> <td>5B</td> <td>5F</td> <td>70</td> <td>7F</td> <td>7B</td> <td>77</td> <td>1F</td> <td>4E</td> <td>3D</td> <td>4F</td> <td>47</td> </tr> </tbody> </table>				No.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	データ	7E	30	6D	79	33	5B	5F	70	7F	7B	77	1F	4E	3D	4F	47						
No.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																												
データ	7E	30	6D	79	33	5B	5F	70	7F	7B	77	1F	4E	3D	4F	47																												
有効パラメーター	<table border="1"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>○</td> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S	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ーエラー																																												


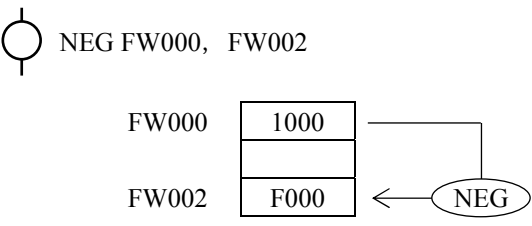
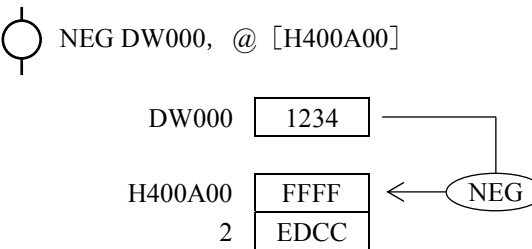
ASP	バイナリー→アスキー変換パックモード																																																					
機能説明	ソースの内容をバイナリーからアスキーデータに変換し、パックモードでリザルトに格納します。																																																					
パラメーターと処理内容	 ASP S, R S : ソース R : リザルト	S (バイナリー) → R (アスキーパック)																																																				
フラグの設定	Eが変化、他はOFF																																																					
備考	Sのサイズ×2がRに書き込まれます。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																																					
使用例	 ASP FW000, FW002 <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>FW000</td><td>5678</td></tr> <tr><td>FW002</td><td>3536</td></tr> <tr><td>FW003</td><td>3738</td></tr> </table>   ASP [HDEF01234], @ [H400A00] <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>HDEF01234</td><td></td></tr> </table>  <table border="1" style="display: inline-table; margin-left: 20px;"> <tr><td>H400A00</td><td>4445</td></tr> <tr><td>2</td><td>4630</td></tr> <tr><td>4</td><td>3132</td></tr> <tr><td>6</td><td>3334</td></tr> </table> [アスキー、バイナリー対応表] <table border="1" style="width: 100%; text-align: center;"> <tr> <td>バイナリー</td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td> </tr> <tr> <td>アスキー</td> <td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td> </tr> </table>				FW000	5678	FW002	3536	FW003	3738	HDEF01234		H400A00	4445	2	4630	4	3132	6	3334	バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46
FW000	5678																																																					
FW002	3536																																																					
FW003	3738																																																					
HDEF01234																																																						
H400A00	4445																																																					
2	4630																																																					
4	3132																																																					
6	3334																																																					
バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																																						
アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46																																						
有効パラメーター	<table border="1" style="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> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>○</td> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S	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ーエラー																																																						


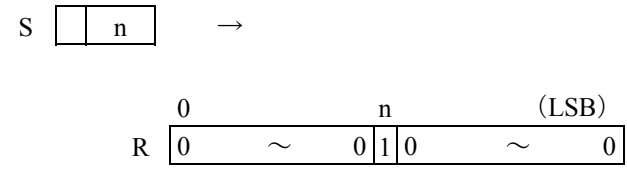
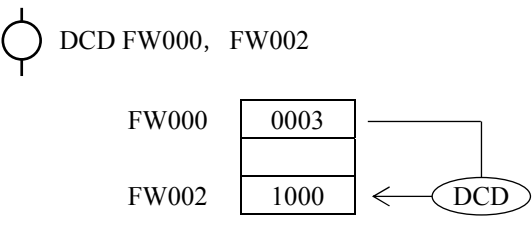
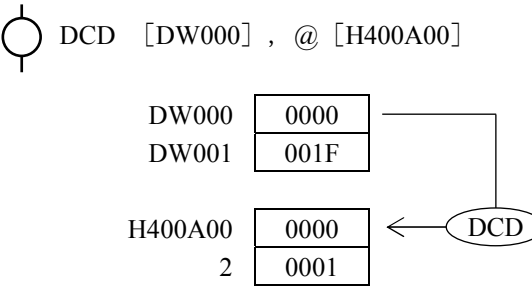
ASU	バイナリー→アスキー変換アンパックモード																																																																															
機能説明	ソースの内容をバイナリーからアスキーデータに変換し、アンパックモードでリザルトに格納します。																																																																															
パラメーターと処理内容	 ASU S, R S: ソース R: リザルト	S (バイナリー) → R (アスキーアンパック)																																																																														
フラグの設定	Eが変化、他はOFF																																																																															
備考	Sのサイズ×4がRに書き込まれます。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																																																															
使用例	 ASU FW001, FW002 <table border="1" style="margin-left: 20px;"> <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> <p style="margin-left: 20px;">← アスキー変換</p>  ASU [HDEF01234], @ [H400A00] <table border="1" style="margin-left: 20px;"> <tr><td>HDEF01234</td><td>アスキー変換</td><td>H400A00</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、バイナリー対応表]</p> <table border="1" style="margin-left: 20px;"> <tr><td>バイナリー</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td></tr> <tr><td>アスキー</td><td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td></tr> </table>				FW001	5678	FW002	3035	FW003	3036	FW004	3037	FW005	3038	HDEF01234	アスキー変換	H400A00	3044			2	3045			4	3046			6	3030			8	3031			A	3032			C	3033			E	3034	バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46
FW001	5678																																																																															
FW002	3035																																																																															
FW003	3036																																																																															
FW004	3037																																																																															
FW005	3038																																																																															
HDEF01234	アスキー変換	H400A00	3044																																																																													
		2	3045																																																																													
		4	3046																																																																													
		6	3030																																																																													
		8	3031																																																																													
		A	3032																																																																													
		C	3033																																																																													
		E	3034																																																																													
バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																																																																
アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46																																																																
有効パラメーター	<table border="1" 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>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>○</td> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S	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ーエラー																																																																																

APB	アスキー→バイナリー変換パックモード																																					
機能説明	ソースの内容をアスキーデータ（パックモード）からバイナリーに変換し、リザルトに格納します。																																					
パラメーターと処理内容	 APB S, R		S（アスキーパック）→ R（バイナリー）																																			
	S：ソース R：リザルト																																					
フラグの設定	Eが変化、他はOFF																																					
備考	Rのサイズ×2がSから取り込まれ変換されます。 SにH30～39、H41～46のデータがあるとEフラグONし、未処理となります。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																					
使用例	 <p>FW000 3132 FW001 3334 FW002 1234 ← バイナリー変換</p>  <p>DW000 4645 1 4443 2 3938 3 3736</p> <p>バイナリー変換 → H400A00 2 FEDC 9876</p> <p>[アスキー、バイナリー対応表]</p> <table border="1" data-bbox="347 1339 1353 1415"> <tr> <td>バイナリー</td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td> </tr> <tr> <td>アスキー</td> <td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td> </tr> </table>				バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46
バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																						
アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46																						
有効パラメーター	<table border="1" data-bbox="347 1512 869 1881"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	×	○	×	ダイレクトロング長	×	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△														
S	ビット型 PI/O	ワード型 PI/O	定数																																			
ダイレクトワード長	×	○	×																																			
ダイレクトロング長	×	○	×																																			
インダイレクトワード長	×	△	△																																			
インダイレクトロング長	×	△	△																																			
△はアドレス値が奇数時はパラメーターエラー	<table border="1" data-bbox="906 1512 1428 1881"> <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	×	△	△																																			

AUB	アスキー→バイナリー変換アンパックモード																																											
機能説明	ソースの内容をアスキーデータ（アンパックモード）からバイナリーに変換し、リザルトに格納します。																																											
パラメーターと処理内容	 AUB S, R S：ソース R：リザルト		S（アスキー → R（バイナリー）アンパック）																																									
フラグの設定	Eが変化、他はOFF																																											
備考	Rのサイズ×4がSから取り込まれ変換されます。 SにH30～39、H41～46のデータがあるとEフラグONし、未処理となります。 Sが16ビット指定の場合、下位8ビットが有効値となります。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																											
使用例	 <p>AUB FW001, FW005</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <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="text-align: center;">← バイナリー変換</p>  <p>AUB [DW000], @[H400A00]</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <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="text-align: center;">バイナリー変換 →</p> <table border="1" style="display: inline-table; margin-left: 20px;"> <tr><td>H400A00</td><td>0123</td></tr> <tr><td>2</td><td>4567</td></tr> </table>				FW001	3035	FW002	3036	FW003	3037	FW004	3038	FW005	5678	DW000	1130		1131		0032		2233		3334		4435		5536		FF37	H400A00	0123	2	4567										
FW001	3035																																											
FW002	3036																																											
FW003	3037																																											
FW004	3038																																											
FW005	5678																																											
DW000	1130																																											
	1131																																											
	0032																																											
	2233																																											
	3334																																											
	4435																																											
	5536																																											
	FF37																																											
H400A00	0123																																											
2	4567																																											
	[アスキー、バイナリー対応表] <table border="1" style="width: 100%; text-align: center;"> <tr> <td>バイナリー</td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td> </tr> <tr> <td>アスキー</td> <td>30</td><td>31</td><td>32</td><td>33</td><td>34</td><td>35</td><td>36</td><td>37</td><td>38</td><td>39</td><td>41</td><td>42</td><td>43</td><td>44</td><td>45</td><td>46</td> </tr> </table>				バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46						
バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																												
アスキー	30	31	32	33	34	35	36	37	38	39	41	42	43	44	45	46																												
有効パラメーター	<table border="1" 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> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>×</td> <td>○</td> <td>×</td> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>×</td> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S	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ーエラー																																												

ABS	絶対値																																								
機能説明	ソースの内容の絶対値をリザルトに格納します。																																								
パラメーターと処理内容	<div style="display: flex; align-items: center;"> <div style="margin-right: 20px;">  ABS S, R </div> <div style="border: 1px solid black; padding: 5px;"> S → R</div> </div> <p>S : ソース R : リザルト</p>																																								
フラグの設定	E, Vが変化、他はOFF																																								
備考	オーバーフロー発生時、リザルトにはH7FFFFFFFが設定されます。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																								
使用例	<div style="margin-bottom: 20px;">  ABS FW000, FW002 </div> <div style="margin-bottom: 20px;"> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>FW000</td><td>FF9C</td></tr> <tr><td>FW002</td><td>0064</td></tr> </table>  </div> <div>  ABS DW000, @ [H400A00] </div> <div> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>DW000</td><td>FFFB</td></tr> <tr><td>H400A00</td><td>0000</td></tr> <tr><td>2</td><td>0005</td></tr> </table>  </div>	FW000	FF9C	FW002	0064	DW000	FFFB	H400A00	0000	2	0005																														
FW000	FF9C																																								
FW002	0064																																								
DW000	FFFB																																								
H400A00	0000																																								
2	0005																																								
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクト ワード長	○	○	○																																						
ダイレクト ロング長	×	○	○																																						
インダイレクト ワード長	×	△	△																																						
インダイレクト ロング長	×	△	△																																						
R	ビット型 PI/O	ワード型 PI/O	定数																																						
ダイレクト ワード長	○	○	×																																						
ダイレクト ロング長	×	○	×																																						
インダイレクト ワード長	×	△	△																																						
インダイレクト ロング長	×	△	△																																						
△はアドレス値が奇数時はパラメーターエラー																																									


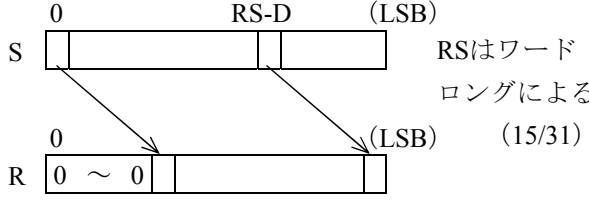


NEG	符号変換																																									
機能説明	ソースの内容の符号変換処理を行いリザルトに格納します。																																									
パラメーターと処理内容	 NEG S, R	$-S \rightarrow R$																																								
	S : ソース R : リザルト																																									
フラグの設定	E, Vが変化、他はOFF																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


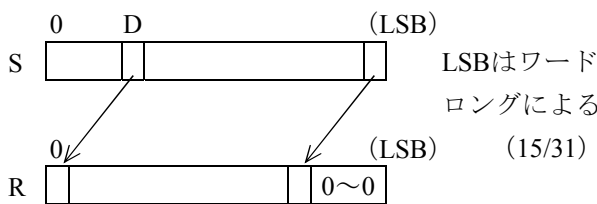
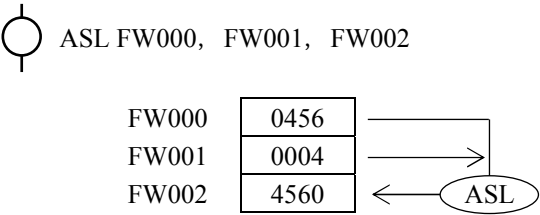
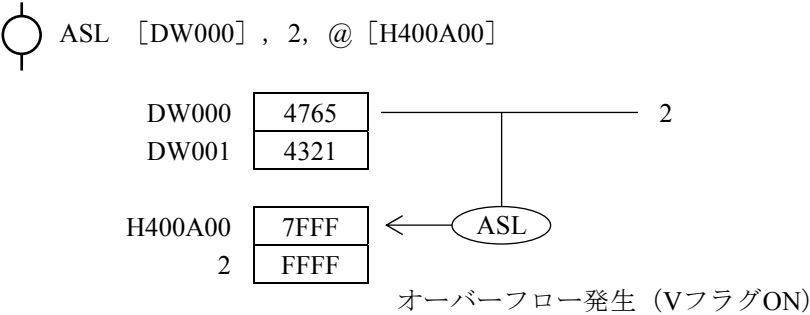
DCD	デコード																																										
機能説明	ソースの内容をデコードして結果をリザルトに格納します。																																										
パラメーターと処理内容	 DCD S, R S : ソース R : リザルト	 <p>S <table border="1" style="display: inline-table;"><tr><td> </td><td>n</td></tr></table> →</p> <p>R <table border="1" style="display: inline-table;"><tr><td>0</td><td> </td><td> </td><td>n</td><td> </td><td>(LSB)</td></tr><tr><td>0</td><td> </td><td> </td><td>1</td><td> </td><td>0</td></tr></table></p>		n	0			n		(LSB)	0			1		0	Sで指定されたnを使用して、RのMSB側から数えて（0から数えます）ビット番号nのビットをONします。																										
	n																																										
0			n		(LSB)																																						
0			1		0																																						
フラグの設定	Eが変化、他はOFF																																										
備考	Sの有効ビットはRがワード長るとき下位4ビット、ロング長るとき下位5ビットとなります。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																										
使用例	 <p>DCD FW000, FW002</p> <p>FW000 <table border="1" style="display: inline-table;"><tr><td>0003</td></tr></table></p> <p>FW002 <table border="1" style="display: inline-table;"><tr><td>1000</td></tr></table></p> <p>DCD</p>  <p>DCD [DW000], @[H400A00]</p> <p>DW000 <table border="1" style="display: inline-table;"><tr><td>0000</td></tr></table></p> <p>DW001 <table border="1" style="display: inline-table;"><tr><td>001F</td></tr></table></p> <p>H400A00 <table border="1" style="display: inline-table;"><tr><td>0000</td></tr><tr><td>0001</td></tr></table></p> <p>2</p> <p>DCD</p>			0003	1000	0000	001F	0000	0001																																		
0003																																											
1000																																											
0000																																											
001F																																											
0000																																											
0001																																											
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>			S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクト ワード長	○	○	○																																								
ダイレクト ロング長	×	○	○																																								
インダイレクト ワード長	×	△	△																																								
インダイレクト ロング長	×	△	△																																								
R	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクト ワード長	○	○	×																																								
ダイレクト ロング長	×	○	×																																								
インダイレクト ワード長	×	△	△																																								
インダイレクト ロング長	×	△	△																																								
△はアドレス値が奇数時はパラメーターエラー																																											

ECD	エンコード																																									
機能説明	ソースの内容をエンコードして結果をリザルトに格納します。																																									
パラメーターと処理内容	<p>⊙ ECD S, R</p> <p>S : ソース R : リザルト</p>	<p>S <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>~</td><td>0</td><td>1</td><td>?</td><td>~</td><td>?</td></tr></table> (LSB)</p> <p>→ R <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td> </td><td> </td><td> </td><td>n</td><td> </td></tr></table></p> <p>SのMSB側から数えて (0から数えます) 最初に1が検出されたnをRに格納します。</p>	0	~	0	1	?	~	?				n																													
0	~	0	1	?	~	?																																				
			n																																							
フラグの設定	Eが変化、他はOFF																																									
備考	S=0のとき未処理となります。エンコードするビットはMSBから最初に1が検出されたビットについてだけ有効となります。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																									
使用例	<p>⊙ ECD FW000, FW002</p> <table border="1" style="margin-left: 20px;"> <tr><td>FW000</td><td>0456</td></tr> <tr><td>FW002</td><td>0005</td></tr> </table> <p style="margin-left: 100px;">← ECD</p> <p>⊙ ECD [DW000] , @ [H400A00]</p> <table border="1" style="margin-left: 20px;"> <tr><td>DW000</td><td>0000</td></tr> <tr><td>DW001</td><td>0080</td></tr> </table> <table border="1" style="margin-left: 20px; margin-top: 10px;"> <tr><td>H400A00</td><td>0000</td></tr> <tr><td>2</td><td>0018</td></tr> </table> <p style="margin-left: 100px;">← ECD</p>			FW000	0456	FW002	0005	DW000	0000	DW001	0080	H400A00	0000	2	0018																											
FW000	0456																																									
FW002	0005																																									
DW000	0000																																									
DW001	0080																																									
H400A00	0000																																									
2	0018																																									
有効パラメーター	<table border="1"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	<table border="1"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	○																																							
ダイレクト ロング長	×	○	○																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
R	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	×																																							
ダイレクト ロング長	×	○	×																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
△はアドレス値が奇数時はパラメーターエラー																																										

LSR	論理右シフト																																									
機能説明	ソースの内容をデスティネーションの内容で右へシフトし、結果をリザルトに格納します。																																									
パラメーターと処理内容	<p>○ LSR S, D, R</p> <p>S : ソース R : リザルト D : デスティネーション</p>																																									
フラグの設定	Eが変化、他はOFF																																									
備考	Dの有効ビットはSがワード長のとき下位4ビット、ロング長のとき下位5ビットとなります。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																									
使用例	<p>○ LSR FW000, FW001, FW002</p> <table border="1"> <tr><td>FW000</td><td>0456</td></tr> <tr><td>FW001</td><td>0004</td></tr> <tr><td>FW002</td><td>0045</td></tr> </table> <p>○ LSR [DW000], 2, @ [H400A00]</p> <table border="1"> <tr><td>DW000</td><td>8765</td></tr> <tr><td>DW001</td><td>4321</td></tr> <tr><td>H400A00</td><td>21D9</td></tr> <tr><td>2</td><td>50C8</td></tr> </table>			FW000	0456	FW001	0004	FW002	0045	DW000	8765	DW001	4321	H400A00	21D9	2	50C8																									
FW000	0456																																									
FW001	0004																																									
FW002	0045																																									
DW000	8765																																									
DW001	4321																																									
H400A00	21D9																																									
2	50C8																																									
有効パラメーター	<table border="1"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	<table border="1"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	○																																							
ダイレクト ロング長	×	○	○																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
R	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	×																																							
ダイレクト ロング長	×	○	×																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
△はアドレス値が奇数時はパラメーターエラー																																										


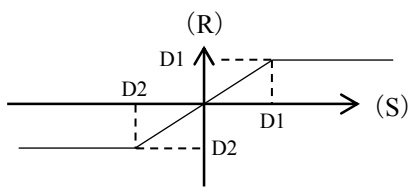
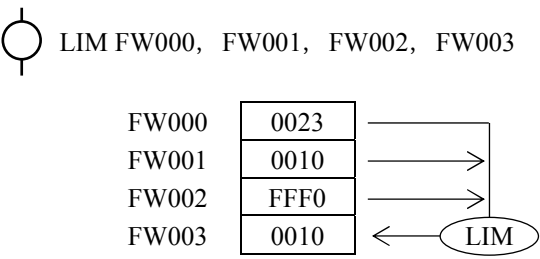
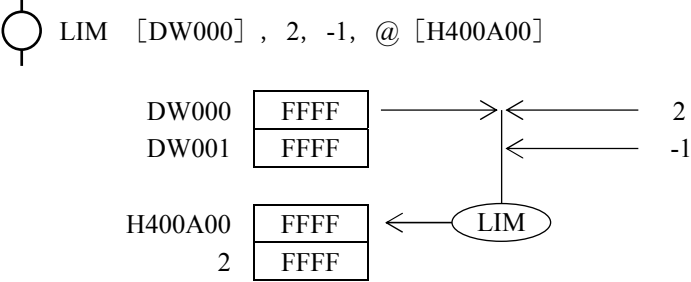
LSL	論理左シフト																																									
機能説明	ソースの内容をデスティネーションの内容で左へシフトし、結果をリザルトに格納します。																																									
パラメーターと処理内容	<p>○ LSL S, D, R</p> <p>S : ソース R : リザルト D : デスティネーション</p>																																									
フラグの設定	Eが変化、他はOFF																																									
備考	Dの有効ビットはSがワード長るとき下位4ビット、ロング長るとき下位5ビットとなります。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																									
使用例	<p>○ LSL FW000, FW001, FW002</p> <table border="1"> <tr><td>FW000</td><td>0456</td><td rowspan="3">→</td></tr> <tr><td>FW001</td><td>0004</td></tr> <tr><td>FW002</td><td>4560</td></tr> </table> <p style="text-align: right;">← ○ LSL</p> <p>○ LSL [DW000] , 2, @ [H400A00]</p> <table border="1"> <tr><td>DW000</td><td>8765</td><td rowspan="2">——— 2</td></tr> <tr><td>DW001</td><td>4321</td></tr> </table> <table border="1"> <tr><td>H400A00</td><td>1D95</td><td rowspan="2">← ○ LSL</td></tr> <tr><td>2</td><td>0C84</td></tr> </table>				FW000	0456	→	FW001	0004	FW002	4560	DW000	8765	——— 2	DW001	4321	H400A00	1D95	← ○ LSL	2	0C84																					
FW000	0456	→																																								
FW001	0004																																									
FW002	4560																																									
DW000	8765	——— 2																																								
DW001	4321																																									
H400A00	1D95	← ○ LSL																																								
2	0C84																																									
有効パラメーター	<table border="1"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	<table border="1"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	○																																							
ダイレクト ロング長	×	○	○																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
R	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	×																																							
ダイレクト ロング長	×	○	×																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
△はアドレス値が奇数時はパラメーターエラー																																										


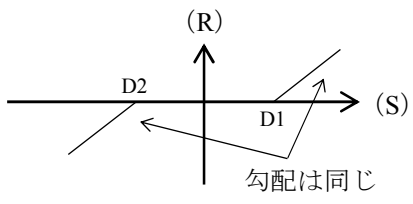
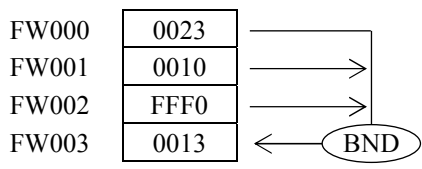
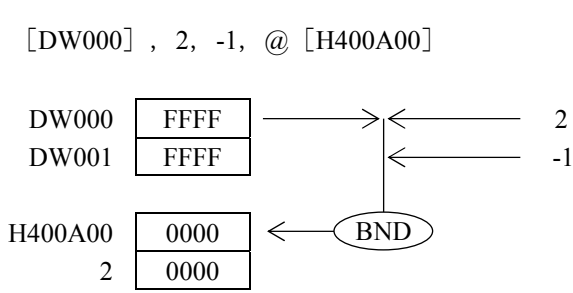
ASR	算術右シフト																																											
機能説明	ソースの内容をデスティネーションの内容で右へシフト（符号ビット保持）し、結果をリザルトに格納します。																																											
パラメーターと処理内容	 ASR S, D, R S : ソース R : リザルト D : デスティネーション																																											
フラグの設定	E, Vが変化、他はOFF																																											
備考	Dの有効ビットはSがワード長るとき下位4ビット、ロング長るとき下位5ビットとなります。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																											
使用例	 ASR FW000, FW001, FW002 <table border="1" data-bbox="462 862 885 985"> <tr><td>FW000</td><td>8456</td><td rowspan="3">→</td></tr> <tr><td>FW001</td><td>0004</td></tr> <tr><td>FW002</td><td>F845</td><td>← (ASR)</td></tr> </table>  ASR [DW000] , 2, @ [H400A00] <table border="1" data-bbox="462 1108 1021 1310"> <tr><td>DW000</td><td>8765</td><td rowspan="2">— 2</td></tr> <tr><td>DW001</td><td>4321</td></tr> <tr><td>H400A00</td><td>E1D9</td><td rowspan="2">← (ASR)</td></tr> <tr><td>2</td><td>0246</td></tr> </table>				FW000	8456	→	FW001	0004	FW002	F845	← (ASR)	DW000	8765	— 2	DW001	4321	H400A00	E1D9	← (ASR)	2	0246																						
FW000	8456	→																																										
FW001	0004																																											
FW002	F845		← (ASR)																																									
DW000	8765	— 2																																										
DW001	4321																																											
H400A00	E1D9	← (ASR)																																										
2	0246																																											
有効パラメーター	<table border="1" data-bbox="343 1400 869 1769"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr><td>ダイレクトワード長</td><td>○</td><td>○</td><td>○</td></tr> <tr><td>ダイレクトロング長</td><td>×</td><td>○</td><td>○</td></tr> <tr><td>インダイレクトワード長</td><td>×</td><td>△</td><td>△</td></tr> <tr><td>インダイレクトロング長</td><td>×</td><td>△</td><td>△</td></tr> </tbody> </table> <table border="1" data-bbox="901 1400 1428 1769"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr><td>ダイレクトワード長</td><td>○</td><td>○</td><td>×</td></tr> <tr><td>ダイレクトロング長</td><td>×</td><td>○</td><td>×</td></tr> <tr><td>インダイレクトワード長</td><td>×</td><td>△</td><td>△</td></tr> <tr><td>インダイレクトロング長</td><td>×</td><td>△</td><td>△</td></tr> </tbody> </table>				S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	○	ダイレクトロング長	×	○	○	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクトワード長	○	○	×	ダイレクトロング長	×	○	×	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクトワード長	○	○	○																																									
ダイレクトロング長	×	○	○																																									
インダイレクトワード長	×	△	△																																									
インダイレクトロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクトワード長	○	○	×																																									
ダイレクトロング長	×	○	×																																									
インダイレクトワード長	×	△	△																																									
インダイレクトロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												

ASL	算術左シフト																																											
機能説明	ソースの内容をデスティネーションの内容で左へシフトし、結果をリザルトに格納します。オーバーフロー発生時フルスケールを設定します。																																											
パラメーターと処理内容	 ASL 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, Vが変化、他はOFF																																											
備考	Dの有効ビットはSがワード長のとき下位4ビット、ロング長のとき下位5ビットとなります。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																											
使用例	 																																											
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	○																																									
ダイレクト ロング長	×	○	○																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	×																																									
ダイレクト ロング長	×	○	×																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												



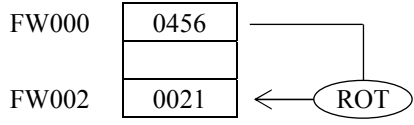

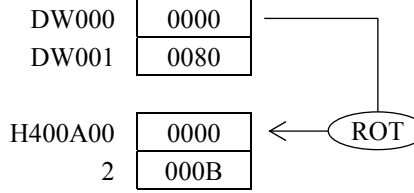
ROR	右回転																																									
機能説明	ソースの内容をデスティネーションの内容で右へ回転し、結果をリザルトに格納します。																																									
パラメーターと処理内容	<p>○ ROR S, D, R</p> <p>S : ソース R : リザルト D : デスティネーション</p>	<p>RSはワード ロングによる (15/31)</p>																																								
フラグの設定	Eが変化、他はOFF																																									
備考	Dの有効ビットはSがワード長のとき下位4ビット、ロング長のとき下位5ビットとなります。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																									
使用例	<p>○ ROR FW000, FW001, FW002</p> <table border="1"> <tr><td>FW000</td><td>8456</td><td rowspan="3">→</td></tr> <tr><td>FW001</td><td>0004</td></tr> <tr><td>FW002</td><td>6845</td></tr> </table> <p style="text-align: center;">← ROR</p> <p>○ ROR [DW000], 2, @ [H400A00]</p> <table border="1"> <tr><td>DW000</td><td>8765</td><td rowspan="2">→</td></tr> <tr><td>DW001</td><td>4321</td></tr> </table> <p style="text-align: center;">2</p> <table border="1"> <tr><td>H400A00</td><td>61D9</td><td rowspan="2">←</td></tr> <tr><td>2</td><td>50C8</td></tr> </table> <p style="text-align: center;">← ROR</p>			FW000	8456	→	FW001	0004	FW002	6845	DW000	8765	→	DW001	4321	H400A00	61D9	←	2	50C8																						
FW000	8456	→																																								
FW001	0004																																									
FW002	6845																																									
DW000	8765	→																																								
DW001	4321																																									
H400A00	61D9	←																																								
2	50C8																																									
有効 パラメーター	<table border="1"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	<table border="1"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	○																																							
ダイレクト ロング長	×	○	○																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
R	ビット型 PI/O	ワード型 PI/O	定数																																							
ダイレクト ワード長	○	○	×																																							
ダイレクト ロング長	×	○	×																																							
インダイレクト ワード長	×	△	△																																							
インダイレクト ロング長	×	△	△																																							
△はアドレス 値が奇数時は パラメーター エラー																																										



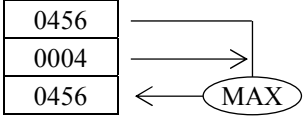

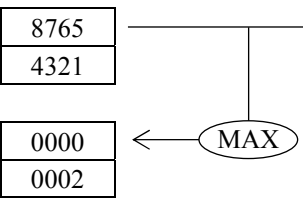
ROL	左回転																																										
機能説明	ソースの内容をデスティネーションの内容で左へ回転し、結果をリザルトに格納します。																																										
パラメーターと処理内容	<p>⊙ ROL S, D, R</p> <p>S : ソース R : リザルト D : デスティネーション</p>	<p>RSはワード ロングによる (15/31)</p>																																									
フラグの設定	Eが変化、他はOFF																																										
備考	Dの有効ビットはSがワード長のとき下位4ビット、ロング長のとき下位5ビットとなります。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																										
使用例	<p>⊙ ROL FW000, FW001, FW002</p> <table border="1" style="margin-left: 20px;"> <tr><td>FW000</td><td>8456</td><td rowspan="3" style="text-align: center;">→</td></tr> <tr><td>FW001</td><td>0004</td></tr> <tr><td>FW002</td><td>4568</td></tr> </table> <p style="text-align: center;">← (ROL)</p> <p>⊙ ROL [DW000] , 2, @ [H400A00]</p> <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> </table> <table border="1" style="margin-left: 20px;"> <tr><td>H400A00</td><td>1D95</td><td rowspan="2" style="text-align: center;">← (ROL)</td></tr> <tr><td>2</td><td>0C86</td></tr> </table>			FW000	8456	→	FW001	0004	FW002	4568	DW000	8765	— 2	DW001	4321	H400A00	1D95	← (ROL)	2	0C86																							
FW000	8456	→																																									
FW001	0004																																										
FW002	4568																																										
DW000	8765	— 2																																									
DW001	4321																																										
H400A00	1D95	← (ROL)																																									
2	0C86																																										
有効 パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, D</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table>			S, D	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクト ワード長	○	○	○																																								
ダイレクト ロング長	×	○	○																																								
インダイレクト ワード長	×	△	△																																								
インダイレクト ロング長	×	△	△																																								
R	ビット型 PI/O	ワード型 PI/O	定数																																								
ダイレクト ワード長	○	○	×																																								
ダイレクト ロング長	×	○	×																																								
インダイレクト ワード長	×	△	△																																								
インダイレクト ロング長	×	△	△																																								
△はアドレス 値が奇数時は パラメーター エラー																																											



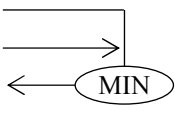

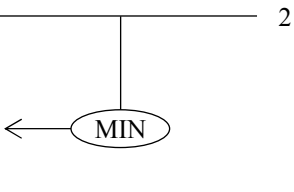
LIM	リミッタ																																											
機能説明	ソースの内容を境界値（デスティネーションD1, D2）の内容と比較し、結果をリザルトに格納します。																																											
パラメーターと処理内容	 LIM S, D1, D2, R S : ソース R : リザルト D1, D2 : デスティネーション																																											
フラグの設定	E, Vが変化、他はOFF																																											
備考	D1<D2のときEフラグONになります。																																											
使用例	 																																											
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S, 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	×	△	△																																									

BND	デッドバンド																																											
機能説明	ソースの内容を境界値（デスティネーションD1, D2）の内容と比較し、境界範囲内を不感帯（データ0）としてリザルトに格納します。																																											
パラメーターと処理内容	 BND S, D1, D2, R S : ソース R : リザルト D1, D2 : デスティネーション																																											
フラグの設定	E, Vが変化、他はOFF																																											
備考	D1<D2のときEフラグONになります。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																											
使用例	 																																											
有効パラメーター	<table border="1" data-bbox="347 1388 869 1758"> <thead> <tr> <th>S, D1, D2</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" data-bbox="901 1388 1428 1758"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <p>△はアドレス値が奇数時はパラメーターエラー</p>				S, D1, D2	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S, D1, D2	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	○																																									
ダイレクト ロング長	×	○	○																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	×																																									
ダイレクト ロング長	×	○	×																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									

ZON	デッドゾーン																																										
機能説明	ソースの内容を正負によってそれぞれバイアス（デスティネーションD1, D2）を加算し、結果をリザルトに格納します。																																										
パラメーターと処理内容	<p>○ ZON S, D1, D2, R</p> <p>S : ソース R : リザルト D1, D2 : デスティネーション</p>																																										
フラグの設定	E, Vが変化、他はOFF																																										
備考	D1<D2のときEフラグONになります。 Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																										
使用例	<p>○ ZON FW000, FW001, FW002, FW003</p> <table border="1" style="margin-left: 20px;"> <tr><td>FW000</td><td>0023</td><td rowspan="4" style="text-align: center;">→</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="margin-left: 100px;">← ○ ZON</p> <p>○ ZON [DW000] , 2, -1, @ [H400A00]</p> <table border="1" style="margin-left: 20px;"> <tr><td>DW000</td><td>FFFF</td><td rowspan="2" style="text-align: center;">←</td><td>2</td></tr> <tr><td>DW001</td><td>FFFF</td></tr> </table> <table border="1" style="margin-left: 20px; margin-top: 10px;"> <tr><td>H400A00</td><td>FFFF</td><td rowspan="2" style="text-align: center;">←</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	H400A00	FFFF	←	○ ZON	2	FFFF																			
FW000	0023	→																																									
FW001	0010																																										
FW002	FFF0																																										
FW003	0033																																										
DW000	FFFF	←	2																																								
DW001	FFFF																																										
H400A00	FFFF	←	○ ZON																																								
2	FFFF																																										
有効パラメーター	<table border="1" style="width: 100%; border-collapse: collapse;"> <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> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td>ダイレクトワード長</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>ダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">○</td> <td style="text-align: center;">○</td> <td>ダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">○</td> <td style="text-align: center;">×</td> </tr> <tr> <td>インダイレクトワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> <td>インダイレクトワード長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> <tr> <td>インダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> <td>インダイレクトロング長</td> <td style="text-align: center;">×</td> <td style="text-align: center;">△</td> <td style="text-align: center;">△</td> </tr> </tbody> </table>			S, D1, D2	ビット型 PI/O	ワード型 PI/O	定数	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, D1, D2	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	×	△	△																																				

ROT	平方根																																											
機能説明	ソースの内容の平方根（整数部だけ）をリザルトに格納します。																																											
パラメーターと処理内容	 ROT S, R S : ソース R : リザルト	$S \geq 0$ のとき Sの平方根（ルート） → R $S < 0$ のとき 0 → R																																										
フラグの設定	E, Vが変化、他はOFF																																											
備考	Rにビット型PI/Oを指定すると結果が正しく設定されないため使用しないでください。																																											
使用例	 ROT FW000, FW002 <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>FW000</td><td>0456</td></tr> <tr><td>FW002</td><td>0021</td></tr> </table>   ROT [DW000] , @ [H400A00] <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>DW000</td><td>0000</td></tr> <tr><td>DW001</td><td>0080</td></tr> </table> <table border="1" style="display: inline-table;"> <tr><td>H400A00</td><td>0000</td></tr> <tr><td>2</td><td>000B</td></tr> </table> 				FW000	0456	FW002	0021	DW000	0000	DW001	0080	H400A00	0000	2	000B																												
FW000	0456																																											
FW002	0021																																											
DW000	0000																																											
DW001	0080																																											
H400A00	0000																																											
2	000B																																											
有効パラメーター	<table border="1" style="display: inline-table; margin-right: 20px;"> <thead> <tr> <th>S</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table> <table border="1" style="display: inline-table;"> <thead> <tr> <th>R</th> <th>ビット型 PI/O</th> <th>ワード型 PI/O</th> <th>定数</th> </tr> </thead> <tbody> <tr> <td>ダイレクト ワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクト ロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクト ワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクト ロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	○	ダイレクト ロング長	×	○	○	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△	R	ビット型 PI/O	ワード型 PI/O	定数	ダイレクト ワード長	○	○	×	ダイレクト ロング長	×	○	×	インダイレクト ワード長	×	△	△	インダイレクト ロング長	×	△	△
S	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	○																																									
ダイレクト ロング長	×	○	○																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
R	ビット型 PI/O	ワード型 PI/O	定数																																									
ダイレクト ワード長	○	○	×																																									
ダイレクト ロング長	×	○	×																																									
インダイレクト ワード長	×	△	△																																									
インダイレクト ロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																												

MAX	最大値																																															
機能説明	ソースの内容とデスティネーションの内容の大小比較を行い、大きい値をリザルトに格納します。																																															
パラメーターと処理内容	 MAX S, D, R S : ソース R : リザルト D : デスティネーション		$S \geq D$ のとき S → R $S < D$ のとき D → R																																													
フラグの設定	E, Vが変化、他はOFF																																															
備考																																																
使用例	 MAX FW000, FW001, FW002 <div style="display: flex; align-items: center; margin-top: 10px;"> <table border="1" style="margin-right: 10px;"> <tr><td>FW000</td><td>0456</td></tr> <tr><td>FW001</td><td>0004</td></tr> <tr><td>FW002</td><td>0456</td></tr> </table>  </div>  MAX [DW000] , 2, @ [H400A00] <div style="display: flex; align-items: center; margin-top: 10px;"> <table border="1" style="margin-right: 10px;"> <tr><td>DW000</td><td>8765</td></tr> <tr><td>DW001</td><td>4321</td></tr> </table> <div style="margin-right: 10px;"> <p>————— 2</p> <p>↓</p> </div> <table border="1" style="margin-right: 10px;"> <tr><td>H400A00</td><td>0000</td></tr> <tr><td>2</td><td>0002</td></tr> </table>  </div>				FW000	0456	FW001	0004	FW002	0456	DW000	8765	DW001	4321	H400A00	0000	2	0002																														
FW000	0456																																															
FW001	0004																																															
FW002	0456																																															
DW000	8765																																															
DW001	4321																																															
H400A00	0000																																															
2	0002																																															
有効パラメーター	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th rowspan="2">S, D</th> <th>ビット型</th> <th>ワード型</th> <th rowspan="2">定数</th> <th rowspan="2">R</th> <th>ビット型</th> <th>ワード型</th> <th rowspan="2">定数</th> </tr> <tr> <th>PI/O</th> <th>PI/O</th> <th>PI/O</th> <th>PI/O</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>○</td> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S, D	ビット型	ワード型	定数	R	ビット型	ワード型	定数	PI/O	PI/O	PI/O	PI/O	ダイレクトワード長	○	○	○	ダイレクトワード長	○	○	×	ダイレクトロング長	×	○	○	ダイレクトロング長	×	○	×	インダイレクトワード長	×	△	△	インダイレクトワード長	×	△	△	インダイレクトロング長	×	△	△	インダイレクトロング長	×	△	△
S, D	ビット型	ワード型	定数	R		ビット型	ワード型			定数																																						
	PI/O	PI/O			PI/O	PI/O																																										
ダイレクトワード長	○	○	○	ダイレクトワード長	○	○	×																																									
ダイレクトロング長	×	○	○	ダイレクトロング長	×	○	×																																									
インダイレクトワード長	×	△	△	インダイレクトワード長	×	△	△																																									
インダイレクトロング長	×	△	△	インダイレクトロング長	×	△	△																																									
△はアドレス値が奇数時はパラメーターエラー																																																

MIN	最小値																																					
機能説明	ソースの内容とデスティネーションの内容の大小比較を行い、小さい値をリザルトに格納します。																																					
パラメーターと処理内容	 MIN S, D, R S : ソース R : リザルト D : デスティネーション		$S \leq D$ のとき S → R $S > D$ のとき D → R																																			
フラグの設定	E, Vが変化、他はOFF																																					
備考																																						
使用例	 MIN FW000, FW001, FW002 <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;"> FW000 FW001 FW002 </div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="width: 40px;">0456</td></tr> <tr><td style="width: 40px;">0004</td></tr> <tr><td style="width: 40px;">0004</td></tr> </table> <div style="margin-left: 10px;">  </div> </div>  MIN [DW000] , 2, @ [H400A00] <div style="display: flex; align-items: center; margin-top: 10px;"> <div style="margin-right: 10px;"> DW000 DW001 H400A00 2 </div> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td style="width: 40px;">8765</td></tr> <tr><td style="width: 40px;">4321</td></tr> <tr><td style="width: 40px;">8765</td></tr> <tr><td style="width: 40px;">4321</td></tr> </table> <div style="margin-left: 10px;">  </div> </div>				0456	0004	0004	8765	4321	8765	4321																											
0456																																						
0004																																						
0004																																						
8765																																						
4321																																						
8765																																						
4321																																						
有効パラメーター	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th rowspan="2">S, D</th> <th colspan="2">ビット型</th> <th colspan="2">ワード型</th> <th rowspan="2">定数</th> </tr> <tr> <th>PI/O</th> <th>PI/O</th> <th>PI/O</th> <th>PI/O</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>○</td> <td>○</td> <td>○</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				S, D	ビット型		ワード型		定数	PI/O	PI/O	PI/O	PI/O	ダイレクトワード長	○	○	○	○	○	ダイレクトロング長	×	○	○	○	○	インダイレクトワード長	×	△	△	△	△	インダイレクトロング長	×	△	△	△	△
S, D	ビット型		ワード型			定数																																
	PI/O	PI/O	PI/O	PI/O																																		
ダイレクトワード長	○	○	○	○	○																																	
ダイレクトロング長	×	○	○	○	○																																	
インダイレクトワード長	×	△	△	△	△																																	
インダイレクトロング長	×	△	△	△	△																																	
△はアドレス値が奇数時はパラメーターエラー	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th rowspan="2">R</th> <th colspan="2">ビット型</th> <th colspan="2">ワード型</th> <th rowspan="2">定数</th> </tr> <tr> <th>PI/O</th> <th>PI/O</th> <th>PI/O</th> <th>PI/O</th> </tr> </thead> <tbody> <tr> <td>ダイレクトワード長</td> <td>○</td> <td>○</td> <td>○</td> <td>×</td> <td>×</td> </tr> <tr> <td>ダイレクトロング長</td> <td>×</td> <td>○</td> <td>○</td> <td>×</td> <td>×</td> </tr> <tr> <td>インダイレクトワード長</td> <td>×</td> <td>△</td> <td>△</td> <td>△</td> <td>△</td> </tr> <tr> <td>インダイレクトロング長</td> <td>×</td> <td>△</td> <td>△</td> <td>△</td> <td>△</td> </tr> </tbody> </table>				R	ビット型		ワード型		定数	PI/O	PI/O	PI/O	PI/O	ダイレクトワード長	○	○	○	×	×	ダイレクトロング長	×	○	○	×	×	インダイレクトワード長	×	△	△	△	△	インダイレクトロング長	×	△	△	△	△
R	ビット型		ワード型			定数																																
	PI/O	PI/O	PI/O	PI/O																																		
ダイレクトワード長	○	○	○	×	×																																	
ダイレクトロング長	×	○	○	×	×																																	
インダイレクトワード長	×	△	△	△	△																																	
インダイレクトロング長	×	△	△	△	△																																	

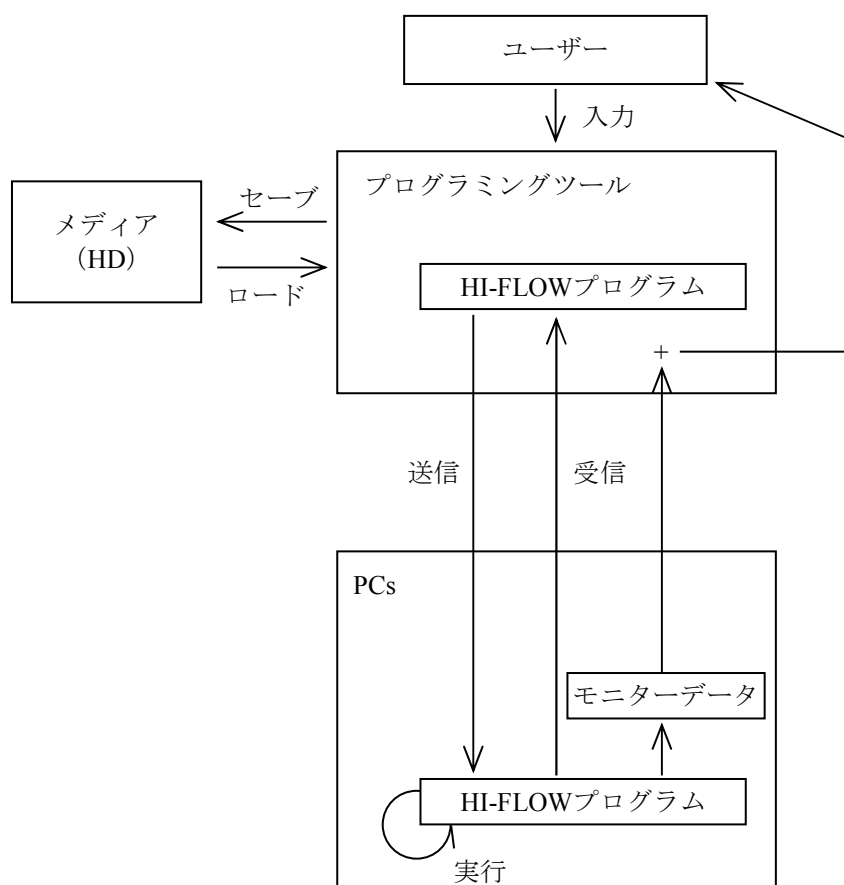
CLR	クリアー																														
機能説明	指定のI/Oエリアをクリアーします。TCLR, UCLR, CCLRは計数値エリアもクリアーします。																														
パラメーターと処理内容	 名称 S 名称：各CLR命令名称 S：ソース（未使用です。0を指定してください。）																														
フラグの設定	すべてのフラグをOFF																														
備考																															
説明	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>名称</th> <th>機能</th> </tr> </thead> <tbody> <tr> <td>XCLR</td> <td>X0000～XFFFFをクリアー</td> </tr> <tr> <td>YCLR</td> <td>Y0000～YFFFFをクリアー</td> </tr> <tr> <td>GCLR</td> <td>G000～GFFFをクリアー</td> </tr> <tr> <td>RCLR</td> <td>R000～RFFFをクリアー</td> </tr> <tr> <td>KCLR</td> <td>K000～KFFFをクリアー</td> </tr> <tr> <td>TCLR</td> <td>T000～T3FFをクリアー Tの計測値クリアー</td> </tr> <tr> <td>UCLR</td> <td>U000～U3FFをクリアー Uの計測値クリアー</td> </tr> <tr> <td>CCLR</td> <td>C000～C3FFをクリアー Cの計測値クリアー</td> </tr> <tr> <td>VCLR</td> <td>V000～VFFFをクリアー</td> </tr> <tr> <td>ECLR</td> <td>E0000～EFFFFをクリアー</td> </tr> <tr> <td>FCLR</td> <td>S0020～S002Fをクリアー</td> </tr> <tr> <td>JCLR</td> <td>J000～JFFFをクリアー</td> </tr> <tr> <td>QCLR</td> <td>Q0000～QFFFFをクリアー</td> </tr> <tr> <td>HHCLR</td> <td>HH000～HH1FFをクリアー</td> </tr> </tbody> </table>	名称	機能	XCLR	X0000～XFFFFをクリアー	YCLR	Y0000～YFFFFをクリアー	GCLR	G000～GFFFをクリアー	RCLR	R000～RFFFをクリアー	KCLR	K000～KFFFをクリアー	TCLR	T000～T3FFをクリアー Tの計測値クリアー	UCLR	U000～U3FFをクリアー Uの計測値クリアー	CCLR	C000～C3FFをクリアー Cの計測値クリアー	VCLR	V000～VFFFをクリアー	ECLR	E0000～EFFFFをクリアー	FCLR	S0020～S002Fをクリアー	JCLR	J000～JFFFをクリアー	QCLR	Q0000～QFFFFをクリアー	HHCLR	HH000～HH1FFをクリアー
名称	機能																														
XCLR	X0000～XFFFFをクリアー																														
YCLR	Y0000～YFFFFをクリアー																														
GCLR	G000～GFFFをクリアー																														
RCLR	R000～RFFFをクリアー																														
KCLR	K000～KFFFをクリアー																														
TCLR	T000～T3FFをクリアー Tの計測値クリアー																														
UCLR	U000～U3FFをクリアー Uの計測値クリアー																														
CCLR	C000～C3FFをクリアー Cの計測値クリアー																														
VCLR	V000～VFFFをクリアー																														
ECLR	E0000～EFFFFをクリアー																														
FCLR	S0020～S002Fをクリアー																														
JCLR	J000～JFFFをクリアー																														
QCLR	Q0000～QFFFFをクリアー																														
HHCLR	HH000～HH1FFをクリアー																														
使用例	 XCLR 0  HHCLR 0																														

補足

補足A HI-FLOWプログラムの流れ

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

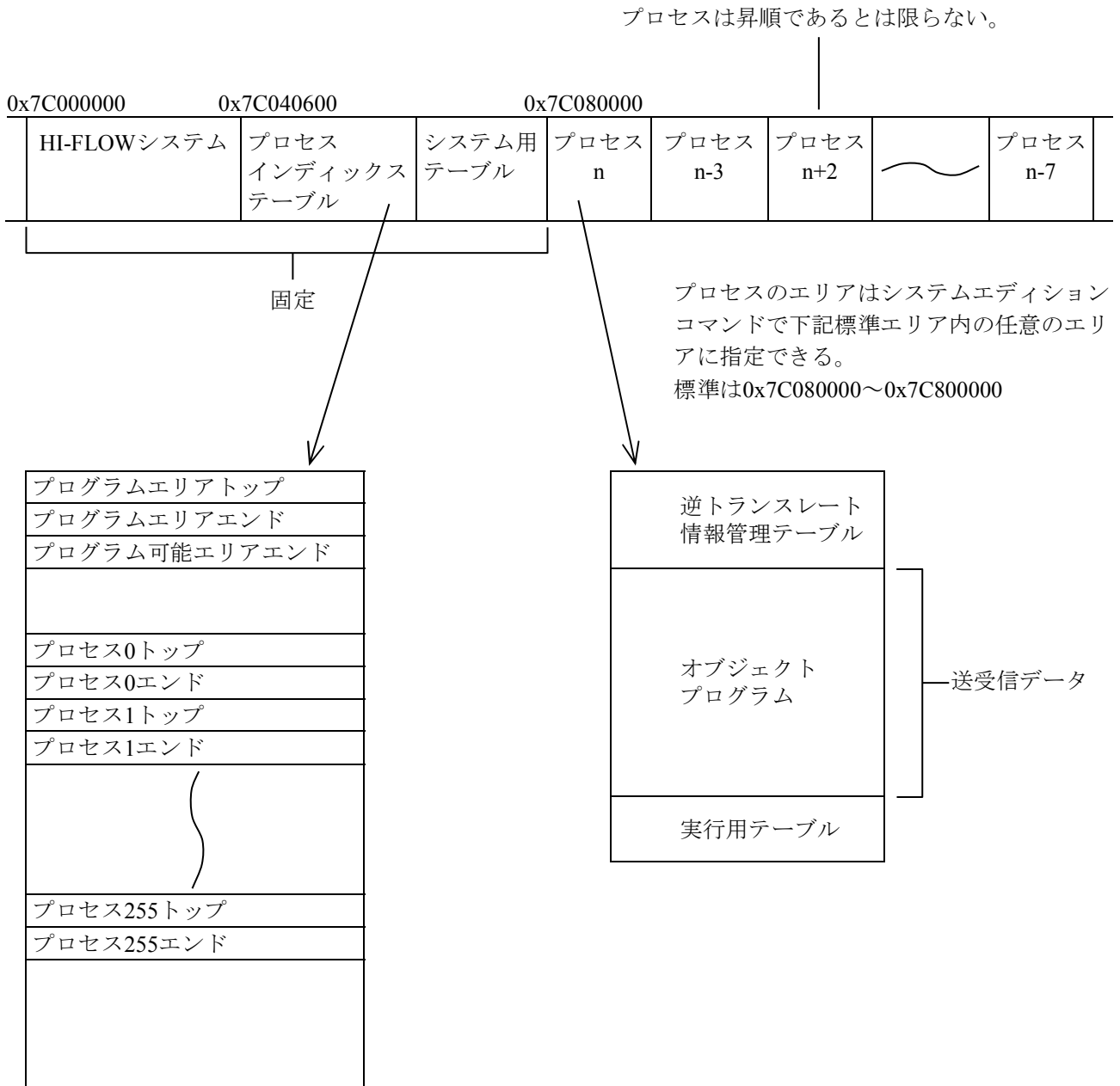
また、作成したプログラムをセーブ/ロードするため各メディア（HD）とのやり取りもあります。



補足

補足B PCsメモリー

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



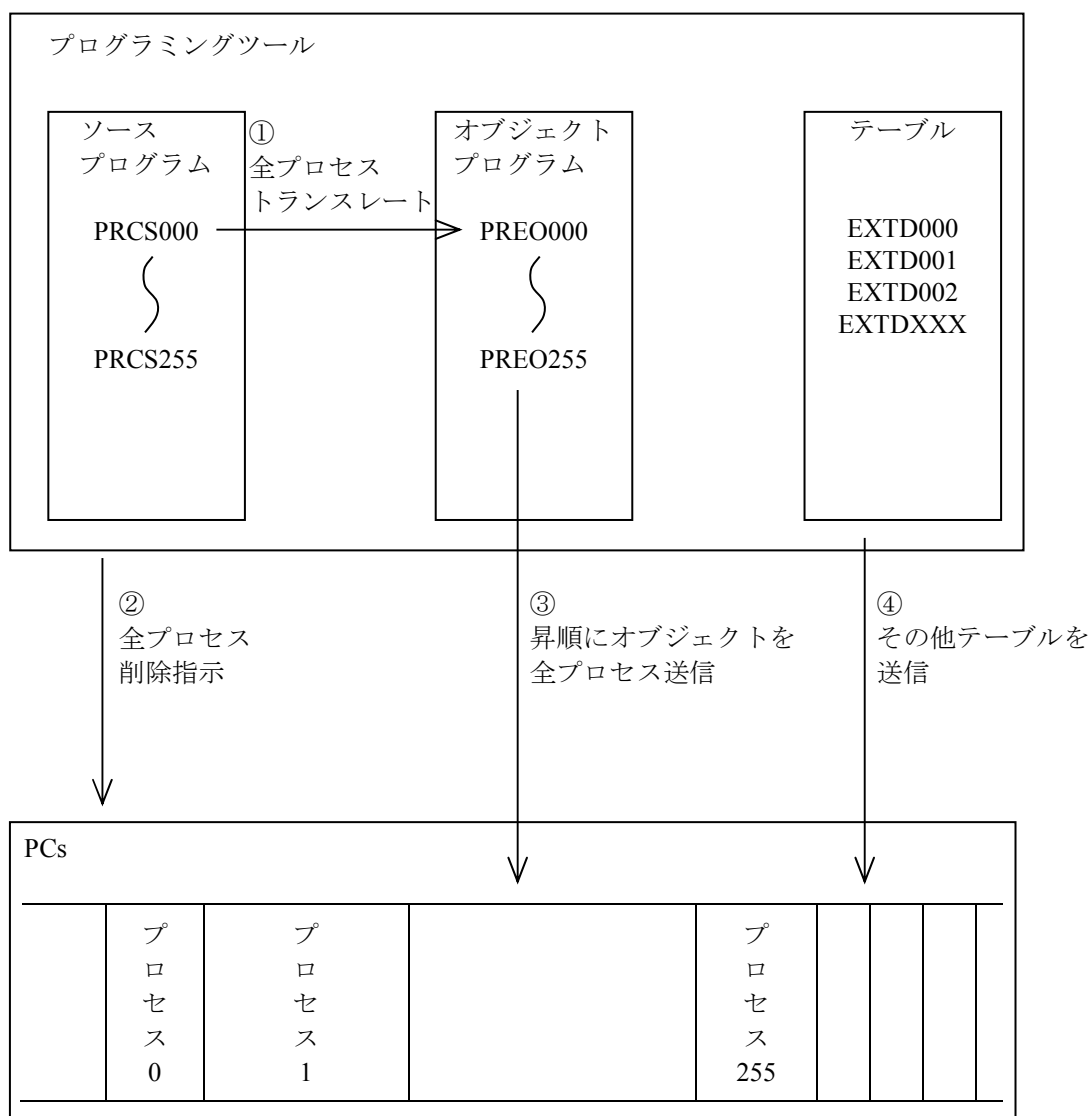
補足C オンラインモード

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

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

(1) 全プロセス送信

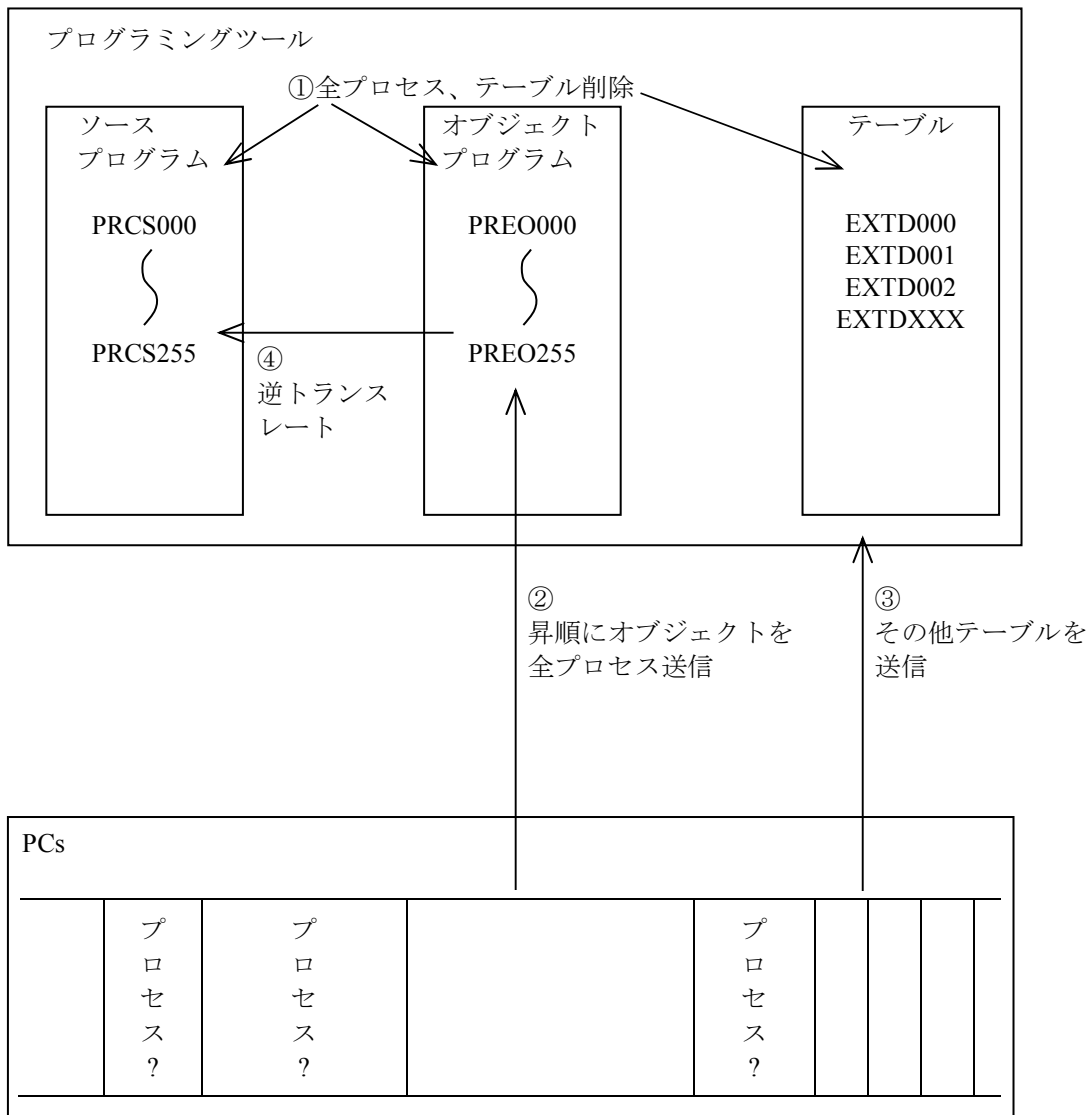
ツール上に存在する全HI-FLOWプログラムをPCsへ送信する際のデータの流れを示します。



全プロセス送信後、メモリー上のプロセス、テーブルは昇順となる

(2) 全プロセス受信

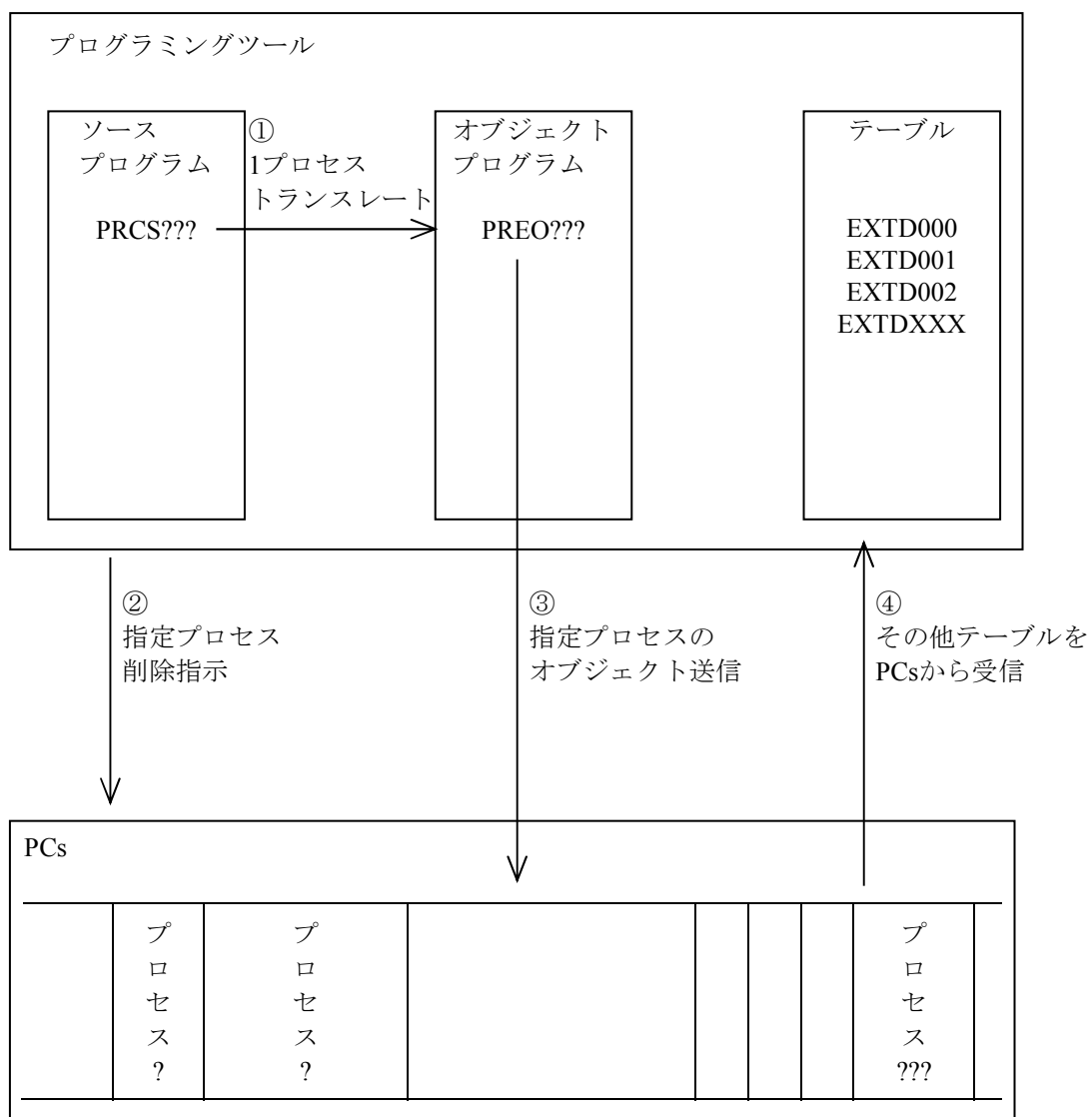
PCs上に存在する全HI-FLOWプログラムをツールへ受信する際のデータの流れを示します。



受信時にメモリー上のプロセス、テーブルが昇順である保証はない

(3) 1プロセス送信

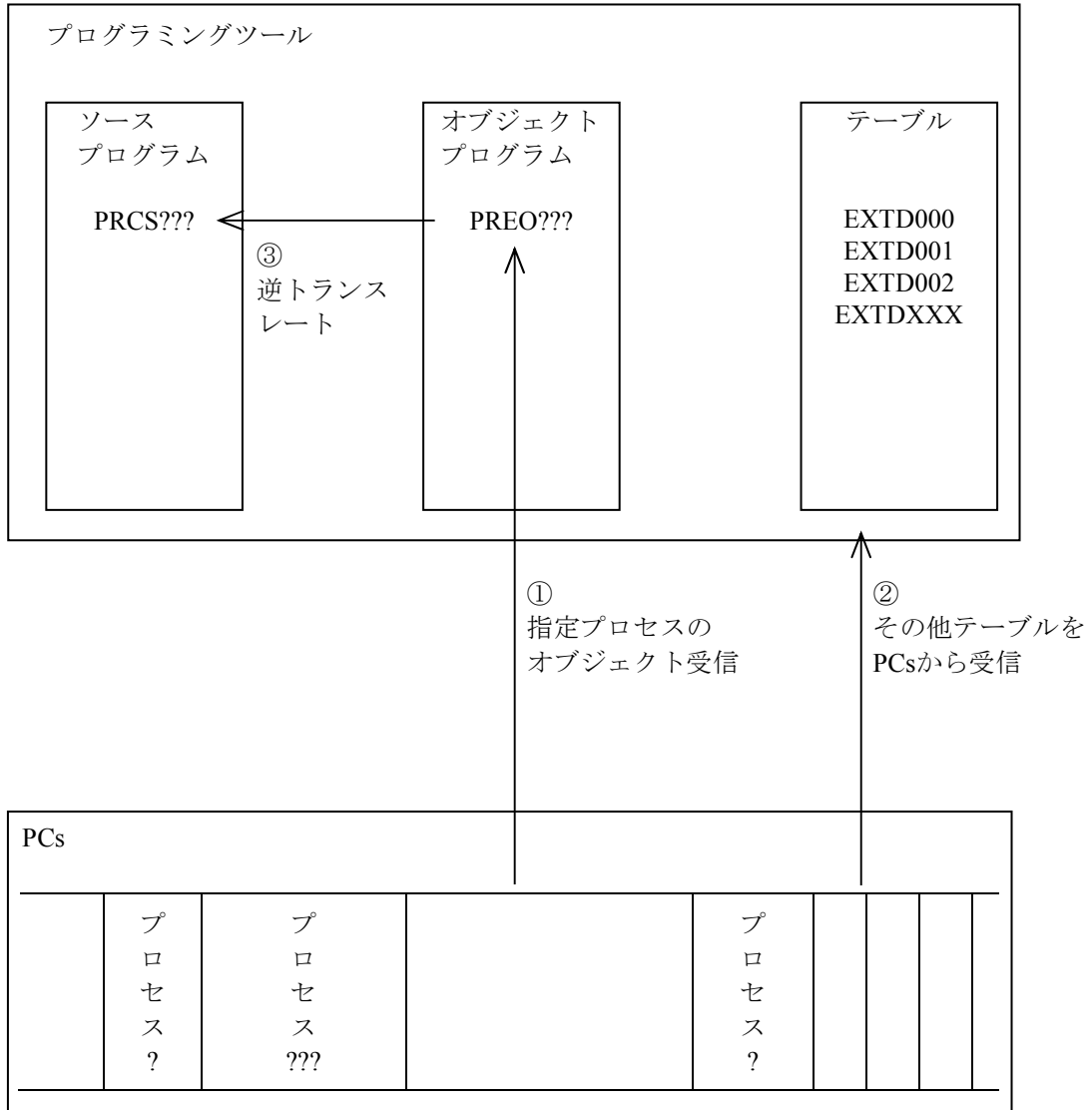
ツール上に存在する特定のHI-FLOWプログラムを1つだけPCsへ送信する際のデータの流れを示します。



送信後、指定プロセスはメモリー上最後になる

(4) 1プロセス受信

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



受信時にメモリー上のプロセス、テーブルが昇順である保証はない

補足D 進行チェック

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

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

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

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

補足E HI-FLOWプログラムとCPU負荷

HI-FLOWプログラムはPCs上でCPUのHP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CPUのHP負荷が上がります。

負荷が上がり過ぎるとシーケンスサイクルが止まるなど、システム全体として正しく動作しなくなります。

そこで、HI-FLOWプログラムの上手な作り方と負荷判断の目安を示します。

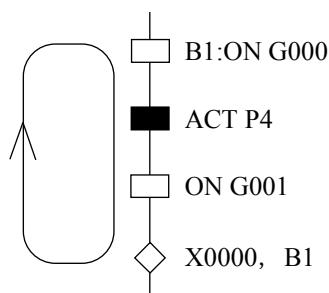
<HI-FLOWプログラムの上手な作り方>

1. HI-FLOWプログラムの負荷大小は実行しているルート数によります。

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。HI-FLOWプログラムの縦（ルート）の長さには関係ありません。

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X0000がOFFするまでループ内のステップを実行し続け、負荷が増大します。

3. タイマー番号は小さい順に詰めて使う。

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

4. ウェイトタイマーは同一ルートでは同一番号を使う。

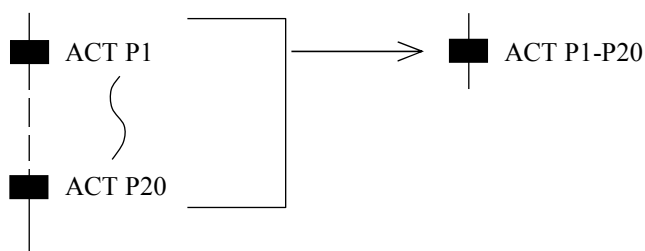
同一ルートのウェイトタイマーは、絶対に同時には実行されません。したがって、同一タイマー番号を付けて、なるべく後ろのタイマー番号を使用しないようにします。

5. コールプロセスは必要最小限にする。

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

6. コントロールボックスの連続使用はなるべく避けるようにする。

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



7. システムコントロールビットの設定は必要最小限にする。

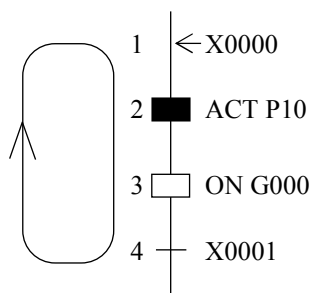
システムコントロールビットはシーケンスサイクルごと、ステップ実行ごとにチェックが必要となり負荷が重くなります。必要最小限の設定をしてください。

8. マルチエントリーの使用は必要最小限にする。

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

9. マルチエントリーのインループに注意する。

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



左記プログラムでX0001が不成立でX0000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X0000をエッジ条件（X0000がOFFからONに変化したときのみ実行）にします。

10. プロセススタートのSTP, RSTの使用は必要最小限にする。

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

11. プロセススタートのCLR設定に注意する。

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

12. 応用命令の連続使用はなるべく避けるようにする。

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

13. 複雑な条件式はなるべく避けるようにする。

HI-FLOWで条件式を複雑にすると式解析にラダーと比較してかなりの時間がかかります。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。

このページは白紙です。