

# HITACHI

ソフトウェアマニュアル  
プログラミング

## S10V ラダー図 For Windows<sup>®</sup>

---

# S10V

Programmable Controller

ソフトウェアマニュアル  
プログラミング  
S10V ラダー図  
For Windows®

---

**S10V**  
Programmable Controller

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

2002年10月	(第1版)	SVJ-3-121	(A)	(廃版)
2003年6月	(第2版)	SVJ-3-121	(B)	(廃版)
2004年4月	(第3版)	SVJ-3-121	(C)	(廃版)
2005年3月	(第4版)	SVJ-3-121	(D)	(廃版)
2005年8月	(第5版)	SVJ-3-121	(E)	(廃版)
2006年6月	(第6版)	SVJ-3-121	(F)	(廃版)
2007年8月	(第7版)	SVJ-3-121	(G)	(廃版)
2012年10月	(第8版)	SVJ-3-121	(H)	(廃版)
2017年8月	(第9版)	SVJ-3-121	(I)	

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

### 警告表示の定義



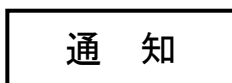
：この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こす危険の存在を示す。



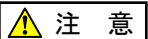
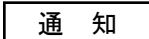
：この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こすおそれのある危険の存在を示す。



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



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

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

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

**重大な傷害**

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


**軽度の傷害または中程度の傷害**

治療のために入院や長期の通院を必要としないけが、やけど、感電傷害など

**人身傷害とは関係のない損害**

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

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

1.  このマニュアル内の警告表示

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

1. 1 「通知」と表示されているもの

(2章、2-17ページ)

通 知
Ver.-Rev. : 01-16以降のラダー図システムで作成した、奇数番号のロングワードレジスターまたはPSHO/POPOを含むラダープログラムを、Ver.-Rev. : 01-15以前のラダー図システムまたは一括ロードにてモジュールレビジョンL (Ver.-Rev. : 02-05) 以前のLPUに送信すると、LPUが“不当命令検出”にて停止します。

(補足、Z-5ページ)

通 知
Ver.-Rev. : 01-16以降のラダー図システムで作成した、奇数番号のロングワードレジスターまたはPSHO/POPOを含むラダープログラムを、Ver.-Rev. : 01-15以前のラダー図システムまたは一括ロードにてモジュールレビジョンL (Ver.-Rev. : 02-05) 以前のLPUに送信すると、LPUが“不当命令検出”にて停止します。

このページは白紙です。

このマニュアルは、以下のプログラムプロダクトの説明をしたものです。

<プログラムプロダクト>

S-7895-02 「S10V ラダー図システム」 01-38



## 来歴一覧表

改訂No.	来歴（改訂内容および改訂理由）	発行年月	備考
E	オンディレイタイマーの最大点数を512点から2048点に変更	2005.8	「S10V ラダー図システム」01-09以降
F	イーサネット通信にOPTETモジュール追加	2006.6	「S10V ラダー図システム」01-12以降
G	コンバート機能拡張に伴う追加	2007.8	「S10V ラダー図システム」01-16以降
H	サポートOSにWindows® 7（32bit）を追加	2012.10	「S10V ラダー図システム」01-34以降
	警告表示全面見直し		
I	サポートOSにWindows® 10（32bit）を追加	2017.8	「S10V ラダー図システム」01-38以降

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

# はじめに

このマニュアルは、ラダープログラムを作成するときの命令語について説明しています。

命令語には、大きく分けてラダー命令と演算ファンクション命令があります。

ラダー命令は、リレー回路の動作を行うものです。

演算ファンクション命令は、加減乗除などの演算処理を行うものです。

S10Vの製品には、標準仕様品と耐環境仕様品があります。耐環境仕様品は、標準仕様品と比べ部品のメッキ厚、コーティングが強化されています。

耐環境仕様品型式は、標準仕様品型式の後に「-Z」が付いています。

(例) 標準仕様品 : LQP520

耐環境仕様品 : LQP520-Z

このマニュアルは、標準仕様品と耐環境仕様品とで共通の内容となっています。このマニュアルには、標準仕様品のモジュール型式のみを記載していますが、耐環境仕様品をご使用の場合も、このマニュアルに従って、正しくご使用いただくようお願いいたします。

プロセスレジスター（レジスター名：P）によるタスク起動機能は、LPUモジュールとCMUモジュールのモジュールレビジョンが下記の場合のみ使用できます。下記以外の組み合わせの場合、プロセスレジスターを使用してもタスクが実行されません。

モジュール名称	モジュール型式	モジュールレビジョン
LPU（基本モジュール）	LQP510	D以降
CMU	LQP520	B以降

## <関連マニュアル>

ソフトウェアマニュアル オペレーション S10V ラダー図 For Windows®

(マニュアル番号 SVJ-3-131)

## <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。

# 目 次

1	ラダー命令	1-1
1.1	ラダープログラム	1-2
1.2	ラダープログラムの動作順序	1-5
1.3	ラダープログラム命令一覧	1-6
1.3.1	ラダープログラム命令一覧	1-6
1.3.2	a接点	1-9
1.3.3	b接点	1-10
1.3.4	立ち上がり／立ち下がりエッジ接点	1-11
1.3.5	演算結果push, read, pop	1-12
1.3.6	演算結果push+a接点, read+a接点, pop+a接点	1-12
1.3.7	演算結果push+b接点, read+b接点, pop+b接点	1-13
1.3.8	ブロック結合（並列接続）（ORB）	1-13
1.3.9	NOT	1-14
1.3.10	コイル	1-15
1.3.11	セットコイル／リセットコイル	1-15
1.3.12	比較命令	1-16
1.3.13	ラダー命令のインデックス指定方式	1-19
1.3.14	回路とステップ	1-20
1.4	リセット、復電時とSTOP⇔RUN遷移時のレジスター状態	1-21
1.5	レジスター一覧	1-22
1.5.1	ラダー命令と使用できるレジスター	1-22
1.5.2	レジスターナンバー一覧	1-24
1.6	ラダーウォッチドッグタイマー	1-62
1.6.1	ラダーウォッチドッグタイマー動作概要	1-62
1.6.2	ラダーウォッチドッグタイマー監視時間設定可能範囲	1-63
1.6.3	ラダーWDTエラー発生時のエラー情報	1-63
2	演算ファンクション	2-1
2.1	機能概要	2-2
2.2	機能仕様	2-4
2.3	演算ファンクションで使用するレジスター	2-8
2.3.1	演算ファンクションで使用できるレジスター	2-8
2.3.2	ビットレジスターの扱い	2-11
2.3.3	ビットレジスターとワード／ロングワードレジスターの関係	2-12

2.4	演算ファンクション入力 .....	2-13
2.5	演算ファンクション一覧表 .....	2-20
2.6	命令の詳細 .....	2-25
2.7	イーサネット通信命令 .....	2-180
2.7.1	機能概要 .....	2-180
2.7.2	使用方法 .....	2-183
2.7.3	命令の詳細 .....	2-196
2.7.4	サンプルプログラム .....	2-220
補 足	.....	Z-1
補足A	平均スキャンタイムの確認 .....	Z-2
A.1	S10V ラダー図システムによる確認 .....	Z-2
A.2	ラダープログラムによる確認 .....	Z-3
補足B	注意事項 .....	Z-4
B.1	ラダープログラムコンバート時の注意事項 .....	Z-4

## 図 目 次

図 A-1 スキャンタイム表示プログラム .....	Z-3
----------------------------	-----

## 表 目 次

表 1-1 ラダー命令（基本命令）一覧 .....	1-6
表 1-2 ラダー命令（比較命令）一覧 .....	1-8
表 1-3 演算ファンクション命令一覧 .....	1-8
表 1-4 使用できるレジスター一覧 .....	1-22
表 1-5 レジスターナンバー一覧 .....	1-24
表 1-6 システムレジスター一覧 .....	1-50
表 2-1 演算ファンクションで使用できるレジスター一覧 .....	2-8

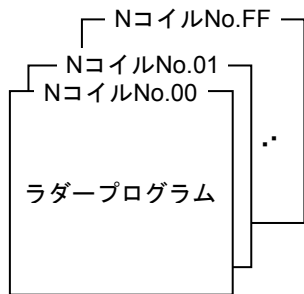
# 1 ラダー命令

# 1 ラダー命令

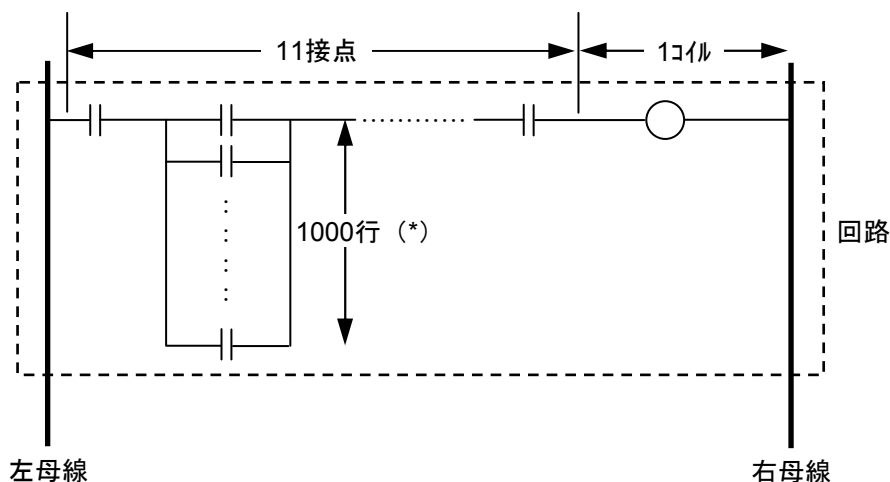
## 1.1 ラダープログラム

ラダープログラムとは、ラダー命令（a接点やb接点などの命令）と演算ファンクション命令（加減算などの演算処理を行う命令）を組み合わせて作成するプログラムです。

どのラダープログラムも、ネスティングコイルと呼ばれる1つ以上のプログラミング単位で構築します。1つのラダープログラム内では、Nコイルは00～FF（計256個）作成できます。NコイルNo.00はマスターコイルと呼ばれ、ラダープログラム実行時にシーケンスサイクルごとに必ず主ルーチンとして実行されます。NコイルNo.01～FFはサブNコイルと呼ばれ、マスターNコイルから、または他のサブNコイルからサブルーチンとして起動されます。

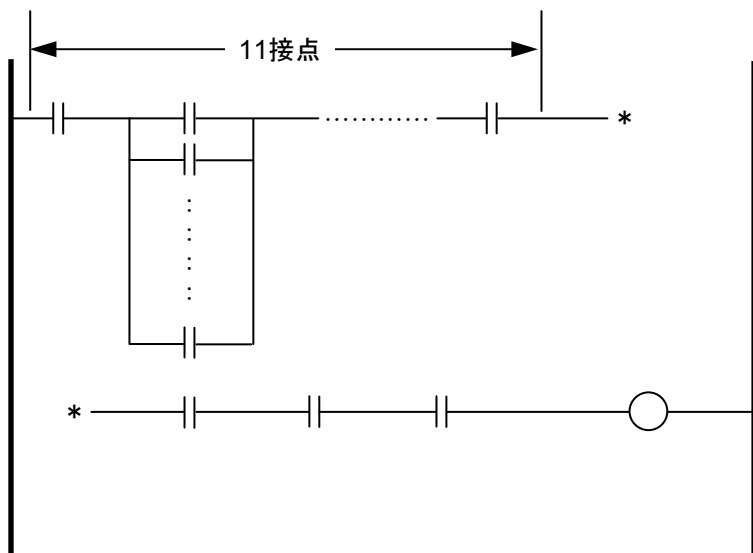


ラダープログラムは、ラダー命令と演算ファンクション命令を組み合わせることにより動作します。動作方向は、左母線から始まり右母線に接続されることにより終了します。右母線に直接接続できるラダー命令は、出力命令（コイルと演算ファンクション命令）のみです。ラダープログラムとして動作する最小のプログラミング単位を回路と呼び、1回路の最大の大きさは、1000行（\*）×12列（11接点 + 1出力）になります。



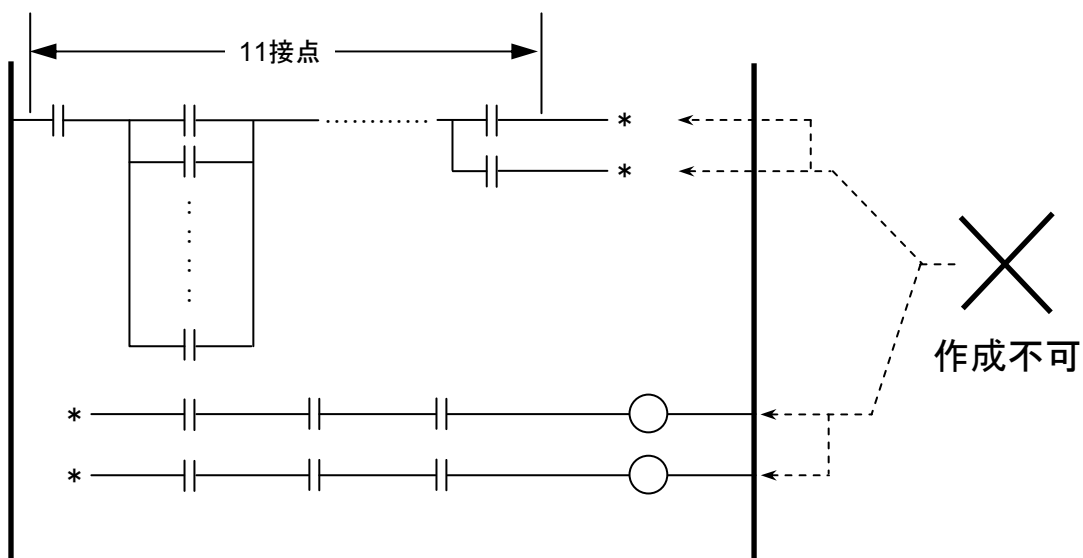
(\*) 1回路の最大行数は、ラダー図システムのVer.-Rev.番号が01-16以降から1000行をサポートしています。それ以前のVer.-Rev.番号のラダー図システムの場合、最大16行です。

11接点以上をAND接続する場合、図に示すような折り返し回路が作成できます。ただし、下記の制限事項があります。



<制限1> 並列のAND接続不可

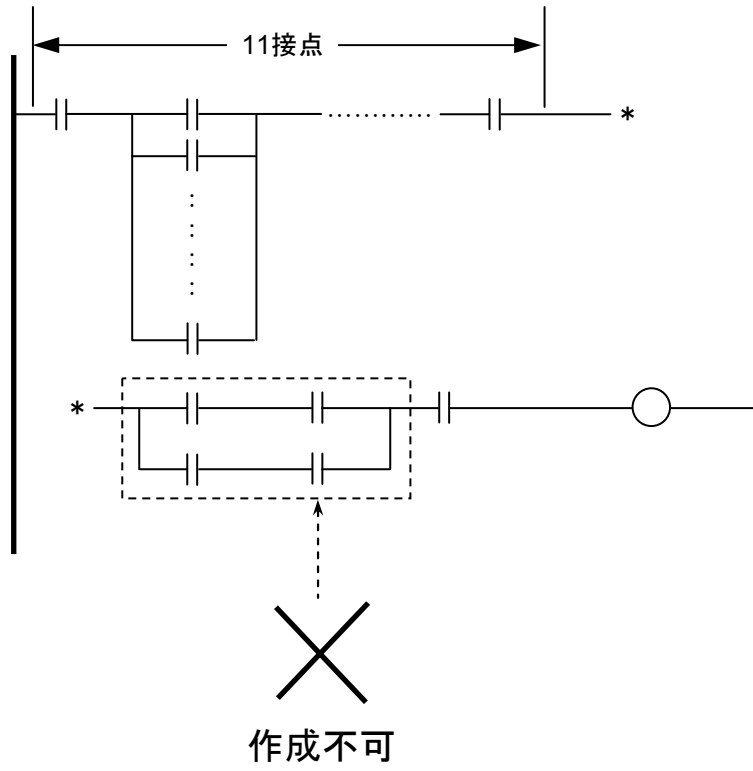
\*印の前に分岐回路は作成できません。





# 1 ラダー命令

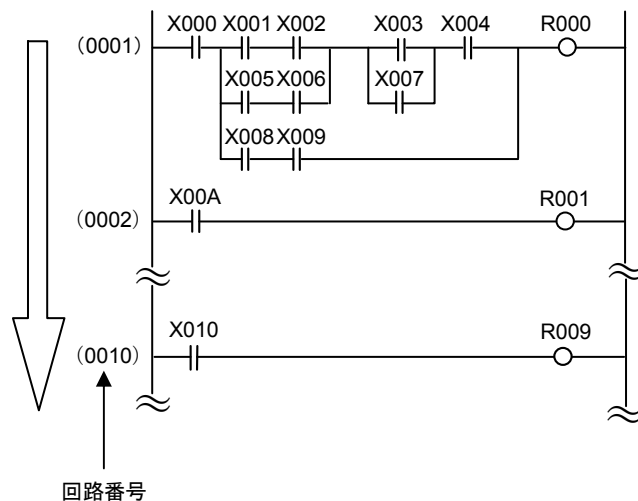
<制限2> \*印の後ろに分岐回路は作成できません。



## 1. 2 ラダープログラムの動作順序

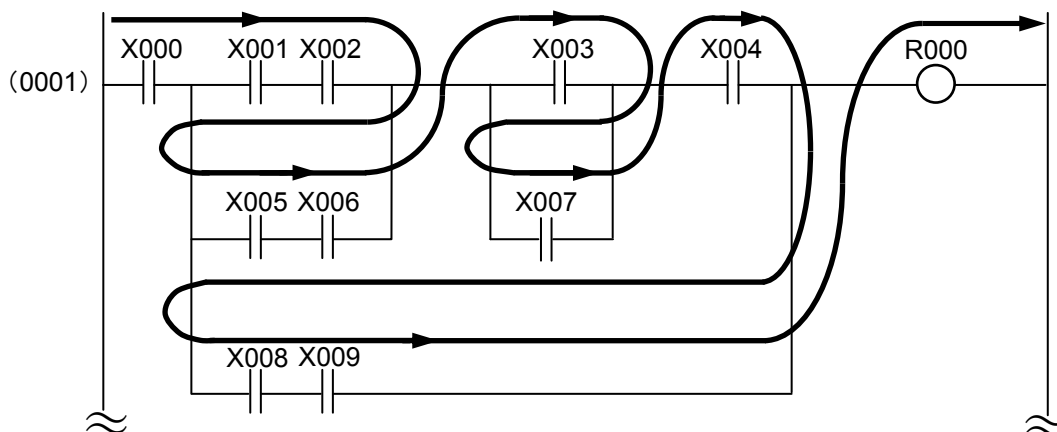
ラダープログラムは、回路番号の順番（昇順）に従って動作します。

【ラダープログラムの例】



1回路内での動作順序を以下に示します。動作は、左上から右向きに行われます。並列回路がある場合、並列回路を処理した後、右の回路を処理します。

下図は、回路の動作順序例を示したものです。下図の回路の場合、太線上の矢印順に動作します。



## 1 ラダー命令

### 1.3 ラダープログラム命令一覧

#### 1.3.1 ラダープログラム命令一覧

ラダープログラムで使用する命令の一覧を表1-1に示します。

##### (1) ラダー命令（基本命令）

基本命令は1命令 = 1ステップとなります。ただし、レジスター名にインデックス指定をした場合は、1命令 = 2ステップになります。

表1-1 ラダー命令（基本命令）一覧

(1/2)

命令名称	記号	オペコード	機能
a接点開始		LD	a接点の開始を意味します。 (a接点：指定したレジスター値が1のときON)
a接点 直列接続		AND	a接点と直前の命令との直列接続を意味します。
b接点開始		LDN	b接点の開始を意味します。 (b接点：指定したレジスター値が0のときON)
b接点 直列接続		ANDN	b接点と直前の命令との直列接続を意味します。
立ち上がり エッジ接点		EGP	入力の立ち上がりエッジを検出した1シーケンスサイクル だけONします。
立ち下がり エッジ接点		EGF	入力の立ち下がりエッジを検出した1シーケンスサイクル だけONします。
演算結果push		SPS	直前の演算結果を記憶します。
演算結果read		SRD	演算結果pushで記憶した演算結果を読み出します。
演算結果pop		SPP	演算結果pushで記憶した演算結果を読み出し後、記憶した 演算結果をリセット（クリア）します。
演算結果push +a接点		SPSAND	演算結果を記憶し、a接点を実行します。
演算結果read +a接点		SRDAND	演算結果pushで記憶した演算結果を読み出し、a接点を実 行します。
演算結果pop +a接点		SPPAND	演算結果pushで記憶した演算結果を読み出し後、記憶した 演算結果をリセットし、a接点を実行します。

表 1-1 ラダー命令（基本命令）一覧

(2/2)

命令名称	記号	オペコード	機能
演算結果push +b接点		SPSANDN	演算結果を記憶し、b接点を実行します。
演算結果read +b接点		SRDANDN	演算結果pushで記憶した演算結果を読み出し、b接点を実行します。
演算結果pop +b接点		SPPANDN	演算結果pushで記憶した演算結果を読み出し後、記憶した演算結果をリセットし、b接点を実行します。
ブロック結合 (並列接続)		ORB	2つの論理ブロックを並列接続します。
NOT		NOT	入力を反転して出力します。
コイル		OUT	指定したレジスタに出力します。 指定したレジスタにより機能が異なります。 T：オンデレイタイマー、U：ワンショットタイマー、C：アップダウンカウンタ、 N：ネスタインクコイル、P：プロセス起動コイル
セットコイル		OUTS	セットコイルがONすると、リセットコイルがONするまで キープリレーのON状態を維持します。
リセットコイ ル		OUTR	セットコイル、リセットコイルには、キープリレー（レジ スター名：K）のみ指定することができます。

## 1 ラダー命令

### (2) ラダー命令（比較命令）

比較命令は1命令 = 3ステップになります。ただし、レジスター名にインデックス指定をした場合は、1命令 = 4～5ステップになります。

表 1-2 ラダー命令（比較命令）一覧


命令名称	記号	オペコード	機能
比較 (EQU)		LEQU	ワードデータを比較し、条件成立でON、非成立でOFFを出力します。 ・比較データに定数を指定できます。 ・定数、変数（レジスター内容）とも最上位ビットをサインビットとみなして比較します。
比較 (NEQ)		LNEQ	
比較 (GT)		LGT	
比較 (GE)		LGE	
比較 (LT)		LLT	
比較 (LE)		LLE	

### (3) 演算ファンクション命令

演算ファンクション命令は1命令 = 1～10ステップになります。

演算ファンクション命令の詳細は、「2 演算ファンクション」を参照してください。

表 1-3 演算ファンクション命令一覧

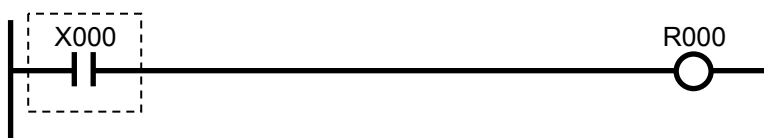
命令名称	記号	オペコード	機能
演算ファンクション		—	ワード/ロングワード/フローティング指定したレジスター/定数を用いて各種演算命令を実行します。

### 1.3.2 a接点

a接点は、指定したレジスタ値がONのときにONを出力します。

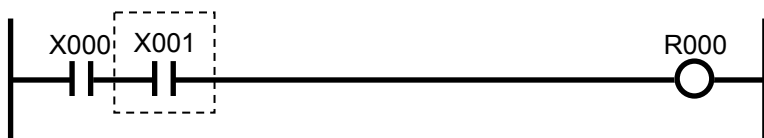
#### (1) a接点開始 (LD)

a接点開始では、指定したレジスタ値がONのときにONを出力します。下記回路例では、X000がONのときにR000がONします。



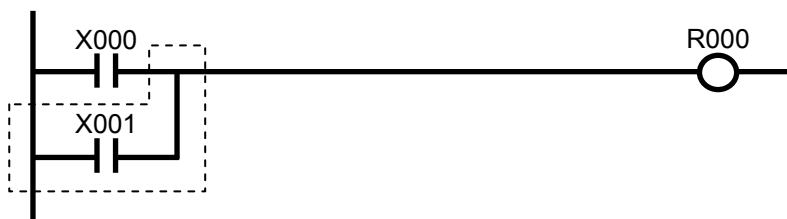
#### (2) a接点直列接続 (AND)

a接点直列接続では、指定したレジスタ値と直前の演算結果をAND演算し、結果がONのときにONを出力します。下記回路例では、X000がON、かつX001がONのときにR000がONします。



#### (3) a接点並列接続 (LD+ORB)

a接点並列接続では、指定したレジスタ値と直前の演算結果をOR演算し、結果がONのときにONを出力します。下記回路例では、X000またはX001がONのときにR000がONします。



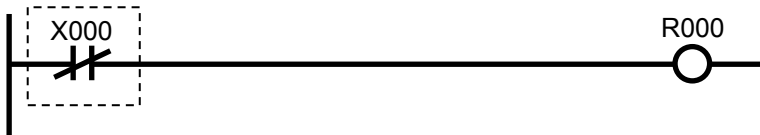
## 1 ラダー命令

### 1.3.3 b接点

b接点は、指定したレジスタ値がOFFのときにONを出力します。

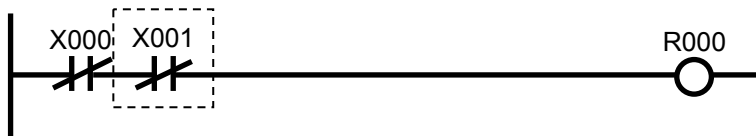
#### (1) b接点開始 (LDN)

b接点開始では、指定したレジスタ値がOFFのときにONを出力します。下記回路例では、X000がOFFのときにR000がONします。



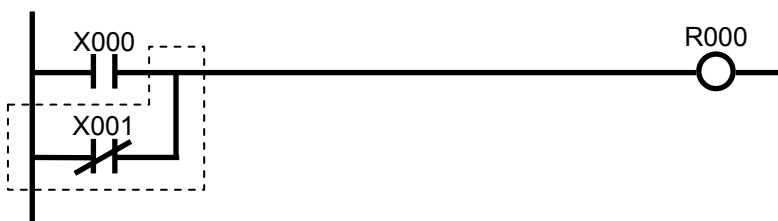
#### (2) b接点直列接続 (ANDN)

b接点直列接続では、指定したレジスタ値を反転した値とそれまでの演算結果とのAND演算を行い、結果がONのときにONを出力します。下記回路例では、X000がOFF、かつX001がOFFのときにR000がONします。



#### (3) b接点並列接続 (LDN+ORB)

b接点並列接続では、指定したレジスタ値を反転した値とそれまでの演算結果とのOR演算を行い、結果がONのときにONを出力します。下記回路例では、X000がONまたはX001がOFFのときにR000がONします。



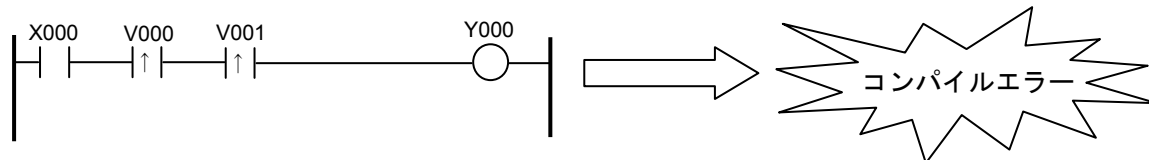
### 1.3.4 立ち上がり／立ち下がりエッジ接点

立ち上がりエッジ接点 (↑) と立ち下がりエッジ接点 (↓) は、直前の演算結果の立ち上がり／立ち下がりエッジを検出した1シーケンスサイクルだけONします。

詳細は、「1.5 レジスタ一覧」の「V エッジ接点」を参照してください。

(注) エッジ接点の直前にエッジ接点を組み合わせた回路は、コンパイル時にエラーとなり作成できません (以下参照)。

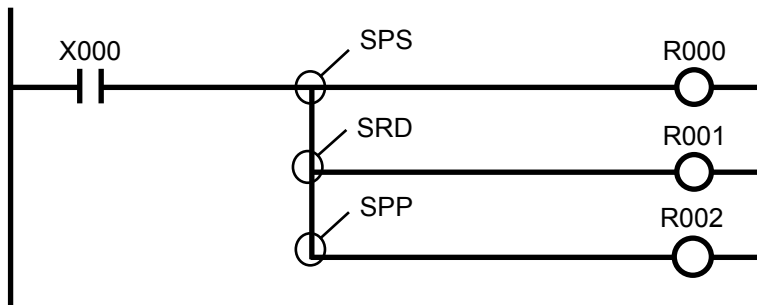
<エッジ接点を組み合わせた場合>





## 1 ラダー命令

### 1.3.5 演算結果push, read, pop



(1) 演算結果push (SPS)

直前の演算結果を記憶します。

(2) 演算結果read (SRD)

SPS, SPSAND、またはSPSANDNで記憶した演算結果を読み出します。

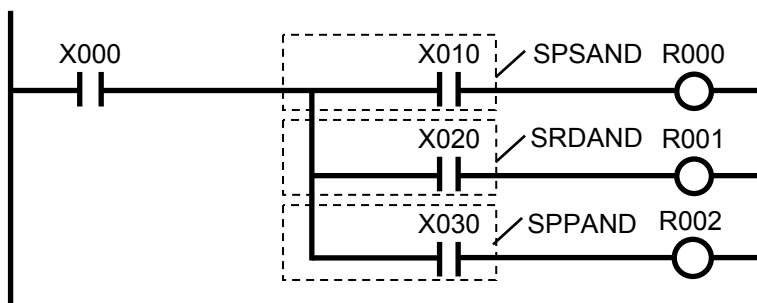
(3) 演算結果pop (SPP)

・ SPS, SPSAND、またはSPSANDNで記憶した演算結果を読み出します。

・ SPS, SPSAND、またはSPSANDNで記憶した演算結果をクリアします。

上記回路例では、X000がONのときにR000, R001、およびR002がONします。

### 1.3.6 演算結果push+a接点, read+a接点, pop+a接点



(1) 演算結果push+a接点 (SPSAND)

直前の演算結果を記憶し、その演算結果で次のa接点を演算します。

(2) 演算結果read+a接点 (SRDAND)

SPS, SPSAND、またはSPSANDNで記憶した演算結果を読み出し、読み出した演算結果で次のa接点を演算します。

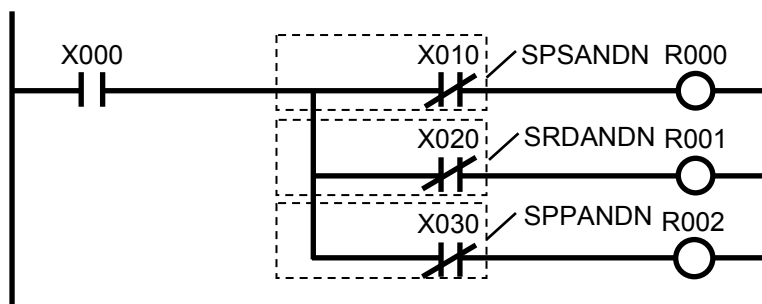
(3) 演算結果pop+a接点 (SPPAND)

・ SPS, SPSAND、またはSPSANDNで記憶した演算結果を読み出し、読み出した演算結果で次のa接点を演算します。

・ SPS, SPSAND、またはSPSANDNで記憶した演算結果をクリアします。

上記回路例では、X000がON、かつX010がONのときにR000がONします。X000がON、かつX020がONのときにR001がONします。X000がON、かつX030がONのときにR002がONします。

## 1. 3. 7 演算結果push+b接点, read+b接点, pop+b接点



## (1) 演算結果push+b接点 (SPSANDN)

直前の演算結果を記憶し、その演算結果で次のb接点を演算します。

## (2) 演算結果read+b接点 (SRDANDN)

SPS, SPSAND、またはSPSANDNで記憶した演算結果を読み出し、読み出した演算結果で次のb接点を演算します。

## (3) 演算結果pop+b接点 (SPPANDN)

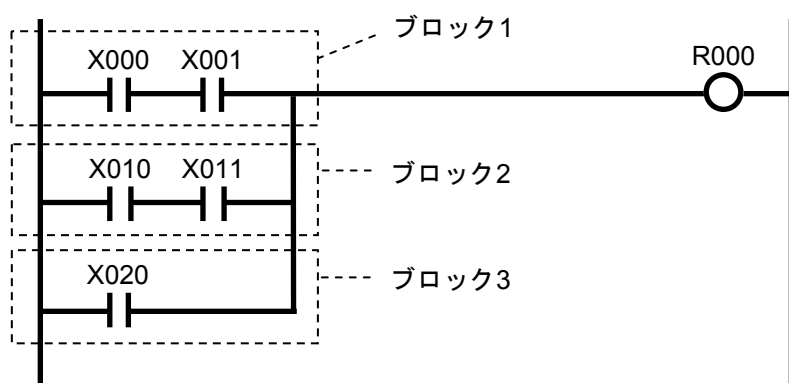
- ・ SPS, SPSAND、またはSPSANDNで記憶した演算結果を読み出し、読み出した演算結果で次のb接点を演算します。

- ・ SPS, SPSAND、またはSPSANDNで記憶した演算結果をクリアします。

上記回路例では、X000がON、かつX010がOFFのときにR000がONします。X000がON、かつX020がOFFのときにR001がONします。X000がON、かつX030がOFFのときにR002がONします。

## 1. 3. 8 ブロック結合 (並列接続) (ORB)

ブロック結合は、ブロックが複数ある回路のブロック間のOR演算を行います。下記回路例では、ブロック1~ブロック3のいずれかのブロックの演算結果がONの場合、R000がONします。

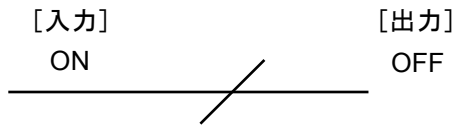


# 1 ラダー命令

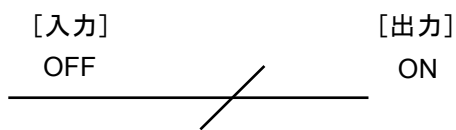
## 1.3.9 NOT

入力を反転して出力します。

<入力がONの場合>

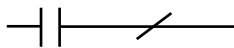


<入力がOFFの場合>

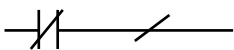


入りに指定できる命令は、a接点、b接点、エッジ接点、比較、並列接続のいずれか、または入力シンボルなしとなります。

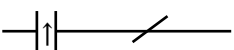
<a接点の結果を反転する場合>



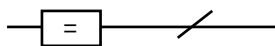
<b接点の結果を反転する場合>



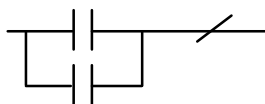
<エッジ接点の結果を反転する場合>



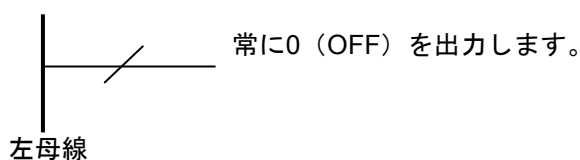
<比較の結果を反転する場合>



<並列接続の結果を反転する場合>



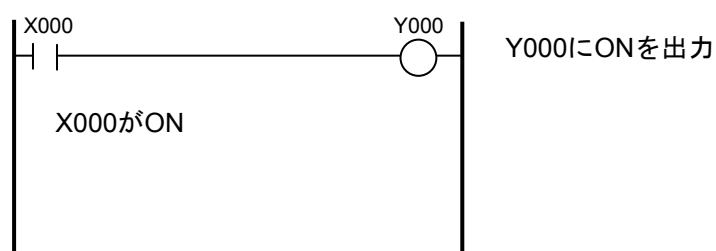
<入力シンボルがない場合>



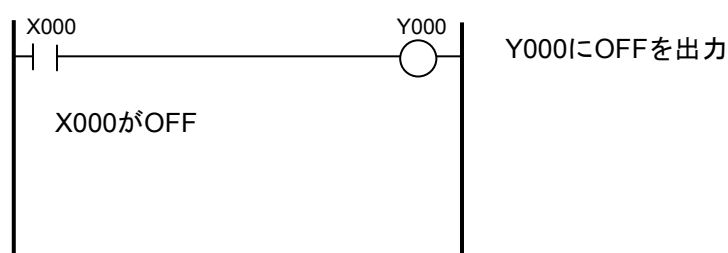
### 1. 3. 10 コイル

指定したレジスタに、それまでの演算結果（ONまたはOFF）を出力します。コイルにタイマー（Tレジスタ）、ワンショット（Uレジスタ）、カウンター（Cレジスタ）を指定した場合の動作は、「1. 5 レジスタ一覧」の「T オンディレイタイマー」、「U ワンショットタイマー」、「C アップダウンカウンター」を参照してください。

<コイルの直前までの条件がONしている場合>



<コイルの直前までの条件がOFFの場合>



### 1. 3. 11 セットコイル／リセットコイル

セットコイルは、それまでの演算結果がONのときキープリレーをONします。その後、演算結果がOFFに変化してもキープリレーはON状態を保存します（OFFしません）。リセットコイルは、セットコイルでONしたキープリレーをOFFします。

詳細は、「1. 5 レジスタ一覧」の「K キープリレー」を参照してください。

## 1 ラダー命令

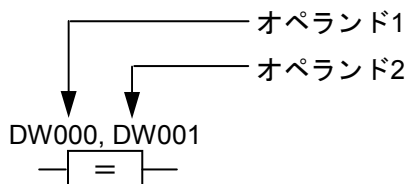
### 1. 3. 12 比較命令

比較命令には、以下の6種類があります。

- 比較 (EQU)
- 比較 (NEQ)
- 比較 (GT)
- 比較 (GE)
- 比較 (LT)
- 比較 (LE)

#### (1) 比較 (EQU)

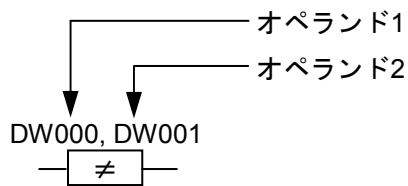
オペランド1とオペランド2の値が一致した場合、出力がONになります。不一致の場合は、OFFになります。



オペランドに指定できるのは、定数ワードおよびワードレジスタのみです。 (\*)

#### (2) 比較 (NEQ)

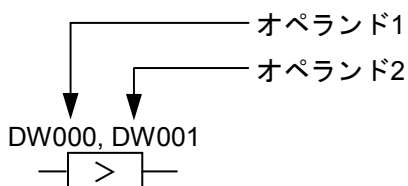
オペランド1とオペランド2の値が不一致の場合、出力がONになります。等しい場合は、OFFになります。



オペランドに指定できるのは、定数ワードおよびワードレジスタのみです。 (\*)

#### (3) 比較 (GT)

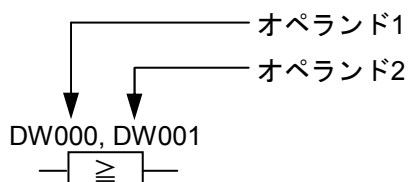
オペランド1がオペランド2より大きい場合、出力がONになります。それ以外の場合は、OFFになります。



オペランドに指定できるのは、定数ワードおよびワードレジスタのみです。 (\*)

## (4) 比較 (GE)

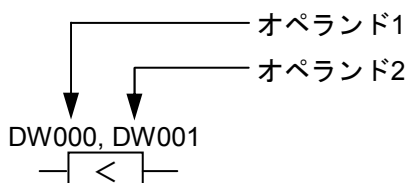
オペランド1がオペランド2より大きいまたは等しい場合、出力がONになります。それ以外の場合は、OFFになります。



オペランドに指定できるのは、定数ワードおよびワードレジスターのみです。 (\*)

## (5) 比較 (LT)

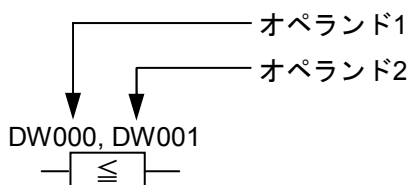
オペランド1がオペランド2より小さい場合、出力がONになります。それ以外の場合は、OFFになります。



オペランドに指定できるのは、定数ワードおよびワードレジスターのみです。 (\*)

## (6) 比較 (LE)

オペランド1がオペランド2より小さいまたは等しい場合、出力がONになります。それ以外の場合は、OFFになります。



オペランドに指定できるのは、定数ワードおよびワードレジスターのみです。 (\*)

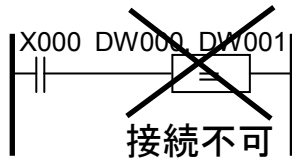
- (\*) 定数はオペランド2のみ指定できます。定数（整数）の範囲は-32768～32767になります。  
 ワード型以外のレジスターは使用できません。  
 定数、レジスターの内容とも符号付きの数とみなして比較します。

## 1 ラダー命令

---

(注) 比較命令は、右母線の出力に直接接続することはできません。

<右母線の出力に直接接続した場合>



### 1. 3. 13 ラダー命令のインデックス指定方式

ラダー命令のうち、a接点、b接点、立ち上がり／立ち下がりエッジ接点、コイル、比較命令は、インデックス指定できます。

- ベースレジスタ（インデックスレジスタ）方式

実行レジスタアドレス = ベースレジスタの番号 + インデックスレジスタの内容（単位：ワード）

この方式は、ベースレジスタの番号 + インデックスレジスタの内容分、離れたエリアを実行アドレスとしてアクセスします。

ベースレジスタに指定できるレジスタの型はA接点、B接点、エッジ接点、コイルの場合はビット型のみ、比較命令の場合はワード型のみです。

インデックスレジスタに指定できるレジスタの型はワード型のみです。

（例）X020（FW000）

X020（FW000）のとき、FW000の内容をH0020としたとき、X020 + H0020 → X040を示します。

（注1）FW000の内容が、H0FF0やH1200などX020に加算してXFFF（Xの最大値）を超えるとき、他のレジスタをアクセスする場合がありますため、動作は保証しません。

（注2）比較命令以外の場合、下記になります。

ベースレジスタの番号 + インデックスレジスタの内容 = 実行レジスタのアドレス

比較命令の場合は、下記になります。

実行レジスタのアドレス = ベースレジスタの番号 + インデックスレジスタの内容 × H0010（16進数）

（例）XW000（FW001）

FW001がH0040の場合、下記になります。

000（ベースレジスタの番号） + H0040（インデックスレジスタの内容） × H0010 = XW400

#### [制限事項]

コイルに下記レジスタ名を使用した場合は、入力時エラーになりインデックス指定はできません。

機能名称	レジスタ名
オンディレイタイマー	T
ワンショットタイマー	U
アップダウンカウンター	C
ネスティング	N
プロセスレジスタ	P

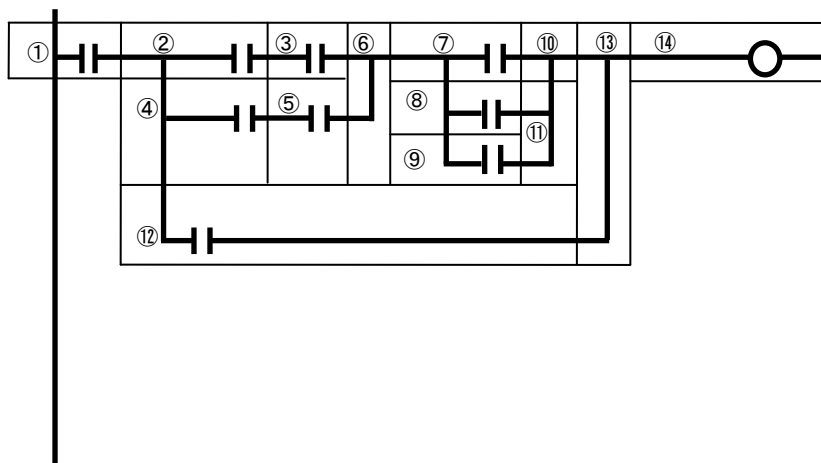


## 1 ラダー命令

---

### 1. 3. 14 回路とステップ

下図の回路は、命令語では14ステップになりますが、ツール上でコンパイル時、1回路の先頭に自動的にスタート命令が付加されるため、15ステップになります。また、プログラムは番号で示した順番で格納、実行されます。□で囲まれた単位がステップになります。



## 1. 4 リセット、復電時とSTOP⇔RUN遷移時のレジスター状態

	レジスター名	リセット、復電時	STOP⇔RUN遷移時
ビット レジスター	T, U接点、コイル	クリア	保持
	C接点、Cコイル	保持	保持
	K	保持	保持
	S	初期値	保持
	X, Y, R, M, A, N, P, E, V, Z, J, Q, LB, LR, LV	クリア	保持
ワードおよび ロングワード レジスター	T, U設定値	保持	保持
	T, U計数值	クリア	保持
	C設定値および計数值	保持	保持
	FW, DW, BD, LX, LM, LG	保持	保持
	LW, LL, LF, IW, OW, HH	クリア	保持

# 1 ラダー命令

## 1.5 レジスタ一覧

### 1.5.1 ラダー命令と使用できるレジスタ

ラダー命令で使用できるレジスタ一覧を表1-4に示します。

表1-4 使用できるレジスタ一覧

(1/2)

	機能名称	レジスタ名	ラダー記号						リセット、 復電後の 状態	
										
入出力	外部入力	X	●	—	—	—	●	●	クリア	
	外部出力	Y	●	—	●	—	●	●	クリア	
内部補助機能	内部レジスタ	R	●	—	●	—	●	●	クリア	
	拡張内部レジスタ	M, A	●	—	●	—	●	●	クリア	
	キープリレー	K	●	—	—	●	●	●	保持	
	オンディレイ タイマー	接点、コイル	T	●	—	●	—	●	●	クリア
		設定値	TS	—	—	—	—	●	●	保持
		計数值	TC	—	—	—	—	●	●	クリア
	ワンショット タイマー	接点、コイル	U	●	—	●	—	●	●	クリア
		設定値	US	—	—	—	—	●	●	保持
		計数值	UC	—	—	—	—	●	●	クリア
	アップダウン カウンター	接点、コイル	CU	—	—	●	—	—	—	保持
			CD	—	—	●	—	—	—	保持
			CR	—	—	●	—	—	—	保持
			C0	●	—	—	—	●	●	保持
		設定値	CS	—	—	—	—	●	●	保持
		計数值	CC	—	—	—	—	●	●	保持
	グローバルリンクレジスタ	G	●	—	●	—	●	●	クリア	
	ネスティングコイル	NM	—	—	●	—	—	—	クリア	
		NZ	—	—	●	—	—	—	クリア	
		N0	●	—	—	—	●	●	クリア	
	プロセスレジスタ	P	●	—	●	—	●	●	クリア	
	イベントレジスタ	E	●	—	●	—	●	●	クリア	
エッジ接点	V	—	●	—	—	●	●	クリア		
ゼットレジスタ	Z	●	—	●	—	●	●	クリア		
システムレジスタ	S	●	—	—	—	●	●	初期値		
HI-FLOWとラダー間 共有データレジスタ	J	●	—	—	—	●	●	クリア		
	Q	●	—	●	—	●	●	クリア		

● : 使用できるレジスタ

— : 使用できないレジスタ

表 1-4 使用できるレジスタ一覧

(2/2)

	機能名称	レジスタ名	ラダー記号						リセット、 復電後の 状態
内部補助機能	HI-FLOWプロセス間レジスタ	HH	—	—	—	—	—	—	クリア
	拡張内部レジスタ	LB	●	—	●	—	●	●	クリア
	コンバーター専用内部レジスタ	LR	●	—	●	—	●	●	クリア
	コンバーター専用エッジ接点レジスタ	LV	—	●	—	—	●	●	クリア
	I/O入力レジスタ (将来用)	IW	—	—	—	—	●	●	クリア
	I/O出力レジスタ (将来用)	OW	—	—	—	—	●	●	クリア
	内部レジスタ	BD	—	—	—	—	—	●	保持
		BW (*)	—	—	—	—	—	●	エリア依存
	ファンクションデータレジスタ	DW	—	—	—	—	●	●	保持
	ファンクションワークレジスタ	FW	—	—	—	—	●	●	保持
	拡張ファンクションワークレジスタ	LW	—	—	—	—	●	●	クリア
	ロングワードワークレジスタ	LL	—	—	—	—	—	●	クリア
	単精度浮動小数点ワークレジスタ	LF	—	—	—	—	—	●	クリア
	バックアップ用ワードワークレジスタ	LX	—	—	—	—	●	●	保持
バックアップ用ロングワードワークレジスタ	LM	—	—	—	—	—	●	保持	
バックアップ用単精度浮動小数点ワークレジスタ	LG	—	—	—	—	—	●	保持	

● : 使用できるレジスタ

— : 使用できないレジスタ

(\*) 間接アドレス方式でアクセスします。

## 1 ラダー命令

### 1.5.2 レジスタナンバー一覧

ラダープログラムで使用できるレジスタ一覧を表1-5に示します。レジスタ名やアクセスする型によって、レジスタナンバーの範囲が異なります。

表1-5 レジスタナンバー一覧

(1/2)

No.	レジスタ名	アクセス			
		ビット	ワード	ロングワード	単精度浮動小数点
1	X	X000~XFFF	XW000~XWFF0	XL000~XLFE0	—
2	Y	Y000~YFFF	YW000~YWFF0	YL000~YLFE0	—
3	R	R000~RFFF	RW000~RWFF0	RL000~RLFE0	—
4	M	M000~MFFF	MW000~MWFF0	ML000~MLFE0	—
5	A	A000~AFFF	AW000~AWFF0	AL000~ALFE0	—
6	K	K000~KFFF	KW000~KWFF0	KL000~KLFE0	—
7	T	T000~T7FF	TW000~TW7F0	TL000~TL7E0	—
8	TS	—	TS000~TS1FF	—	—
9	TC	—	TC000~TC1FF	—	—
10	U	U000~U0FF	UW000~UW0F0	UL000~UL0E0	—
11	US	—	US000~US0FF	—	—
12	UC	—	UC000~UC0FF	—	—
13	CU	CU00~CUFF	—	—	—
14	CD	CD00~CDFF	—	—	—
15	CR	CR00~CRFF	—	—	—
16	C0	C000~C0FF	CW000~CW0F0	CL000~CL0E0	—
17	CS	—	CS000~CS0FF	—	—
18	CC	—	CC000~CC0FF	—	—
19	G	G000~GFFF	GW000~GWFF0	GL000~GLFE0	—
20	NM	NM01~NMFF	—	—	—
21	NZ	NZ01~NZFF	—	—	—
22	N0	N001~N0FF	NW000~NW0F0	NL000~NL0E0	—
23	P	P001~P080	PW000~PW080	PL000~PL060	—
24	E	E000~EFFF	EW000~EWFF0	EL000~ELFE0	—
25	V	V000~VFFF	VW000~VWFF0	VL000~VLFE0	—
26	Z	Z000~Z3FF	ZW000~ZW3F0	ZL000~ZL3E0	—
27	S	S000~SBFF	SW000~SWBF0	SL000~SLBE0	—

— : アクセスできません。

表 1-5 レジスタナンバー一覧

(2/2)

No.	レジスタ名	アクセス			
		ビット	ワード	ロングワード	単精度浮動小数点
28	J	J000~JFFF	JW000~JWFF0	JL000~JLFE0	—
29	Q	Q000~QFFF	QW000~QWFF0	QL000~QLFE0	—
30	LB	LB0000~LBFFFF	LBW0000~LBWFFFF0	LBL0000~LBLFFFE0	—
31	LR	LR0000~LR0FFF	LRW0000~LRW0FF0	LRL0000~LRL0FE0	—
32	LV	LV0000~LV0FFF	LVW0000~LVW0FF0	LVL0000~LVL0FE0	—
33	IW	—	IW000~IWFFF	IL000~ILFFE	—
34	OW	—	OW000~OWFFF	OL000~OLFFE	—
35	BD	—	—	BD000~BD1FE	—
36	BW (*)	—	BW000~BW1FE	BL000~BL1FE	—
37	DW	—	DW000~DWFFF	DL000~DLFFE	—
38	FW	—	FW000~FWBFF	FL000~FLBFE	—
39	LW	—	LWW0000~LWWFFFFF	LWL0000~LWLFFFE	—
40	LL	—	—	LLL0000~LLL1FFF	—
41	LF	—	—	—	LF0000~LF1FFF
42	LX	—	LXW0000~LXW3FFF	LXL0000~LXL3FFE	—
43	LM	—	—	LML0000~LML1FFF	—
44	LG	—	—	—	LG0000~LG1FFF

— : アクセスできません。

(\*) 間接アドレス方式でアクセスします。

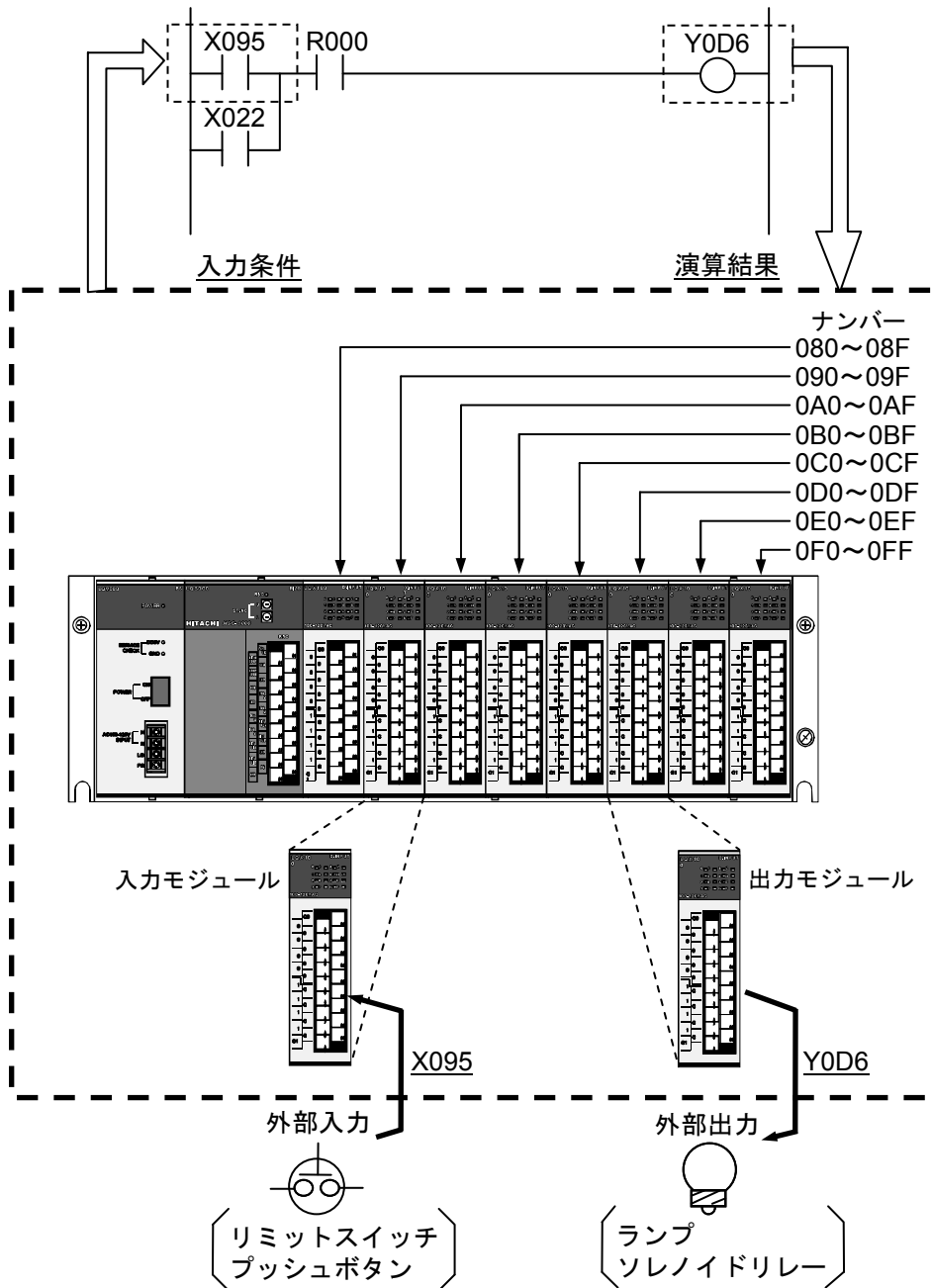
# X, Y 外部入出力

ナンバーの範囲	000~FFF
リモートI/Oの入出力範囲	000~7FF

PCsに接続された外部入出力モジュール経由で信号の入出力をします。  
 X：入力モジュール経由で外部の入力信号を取り込みます。  
 Y：ラダープログラムの演算結果を出力モジュールから外部へ出力します。

● 回路例

入力モジュールのX095がONのとき、出力モジュールのY0D6に対して信号を出力します。



このページは白紙です。

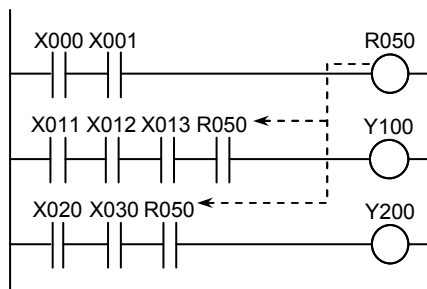


## R, M, A, LB 内部レジスタ

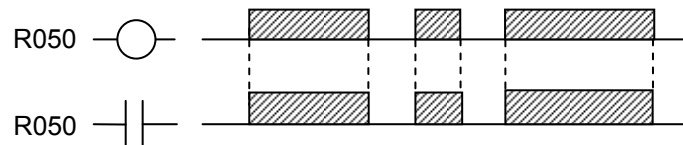
レジスタ名	R, M, A	LB
ナンバーの範囲	000~FFF	0000~FFFF

ラダー命令の演算結果を中継するための内部レジスタです。コイルがONすると同時に接点がONし、コイルがOFFすると同時に接点もOFFします。

### ● 回路例



### ● タイムチャート



R, M, A, LBの各レジスタで機能的な違いはありません。

このページは白紙です。

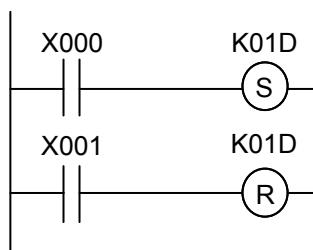
## K キープリレー

ナンバーの範囲	000~FFF
セットリング パルス幅	最小1シーケンスサイクル
セット、リセット 同時入力時	プログラムの下方にある方が優先されます。

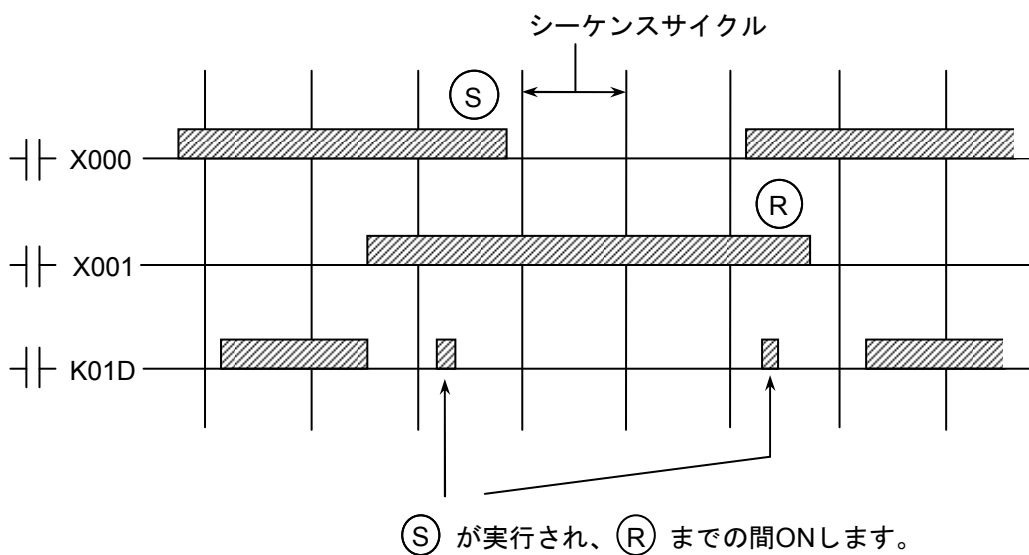
キープリレーは、セットコイルがONすると、リセットコイルがONするまで接点がONし、接点の状態は停電保持されます。また、セットコイルとリセットコイルが同時にONした場合、プログラムの下方にある方が優先されます。

### (1) リセット優先回路

#### ● 回路例

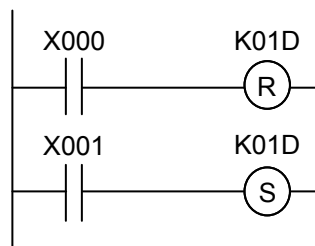


#### ● タイムチャート

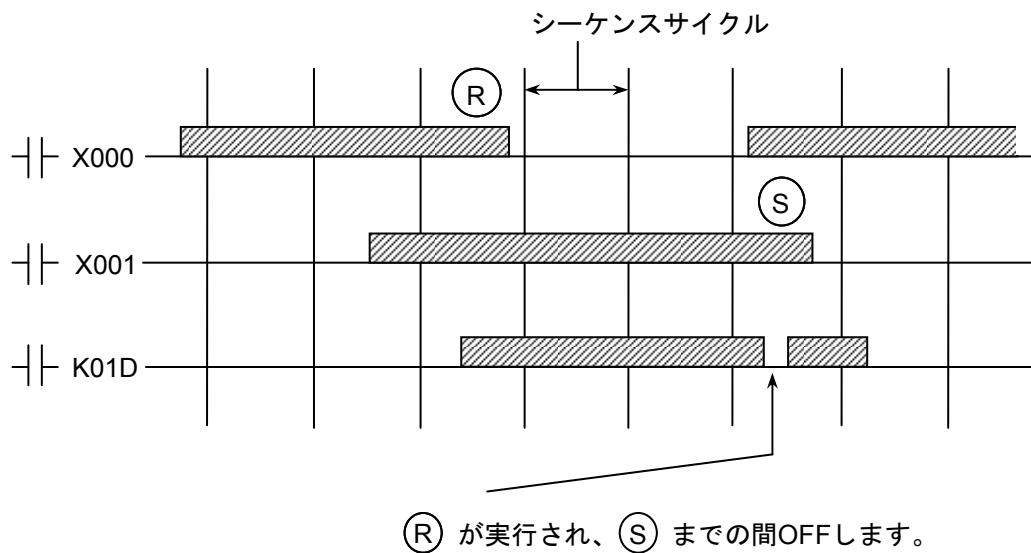


(2) セット優先回路

● 回路例



● タイムチャート



## T オンディレイタイマー

	100msタイマー	10msタイマー (設定による)
ナンバーの範囲 (*1)	000~1FFまたは000~7FF	000~00F
設定値	0~9999 (0.0~999.9秒)	0~9999 (0.0~99.99秒)
誤差	最小100ms + 1シーケンスサイクル	最小10ms + 1シーケンスサイクル
セットリングパルス幅 (*2)	最小100ms	最小10ms

(\*1) 100msタイマーのナンバーの範囲はLPUモジュールのモジュールレビジョンにより異なります。

使用可能なナンバーの範囲とLPUのモジュールレビジョンとの関係を以下に示します。

LPUモジュール型式	モジュール レビジョン	ナンバーの範囲	備考
LQP510	[A] ~ [H]	000~1FF	—
LQP510-Z	[A] ~ [H]		
LQP710-Z	[A] ~ [E]		
LQP510	[I] 以降	000~7FF	ナンバー200~7FFまでを使用するためには S10Vラダー図システム (型式: S-7895- 02) のバージョンが01-09以降である必要が あります。
LQP510-Z	[I] 以降		
LQP710-Z	[F] 以降		

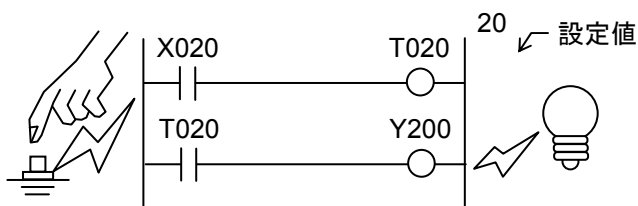
(\*2) セットリングパルス幅とは、オンディレイタイマーのコイルをONする接点がONしていなければならない最小時間のことを意味します。

オンディレイタイマーのコイルがONしてから、設定値の時間だけ遅れて接点がONします。設定値の範囲は、0.0~999.9秒で0.1秒単位に設定できます。

先頭から16点 (T000~T00F) は、設定により10msタイマーとして使用できます。

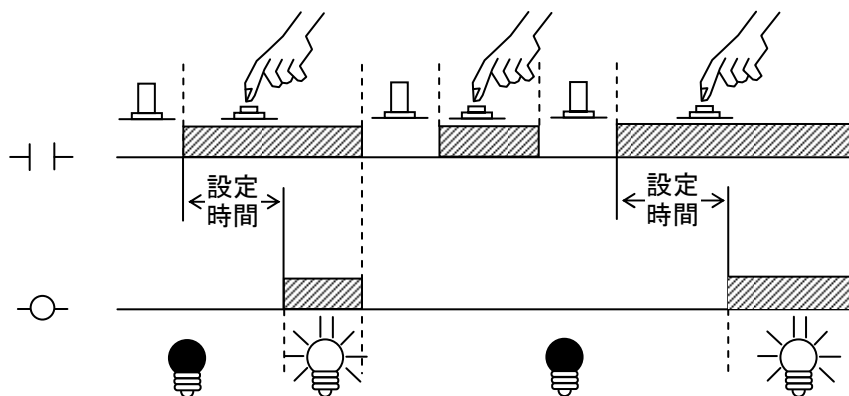
設定は、S10V用プログラミングソフトウェア S10Vラダー図システム (型式: S-7895-02) の [ユーティリティ] - [PCsエディション] - [容量変更] で行います。操作方法は、「ソフトウェアマニュアル オペレーション S10V ラダー図 For Windows® (マニュアル番号 SVJ-3-131)」を参照してください。

### ● 回路例



図に示した回路の場合、押しボタン (X020) を押してから、設定時間後 (この場合2秒後) にランプ (Y200) が点灯し、手を離すと同時にランプが消灯します。

## ● タイムチャート



(注)

- ・ 設定時間に達するまでにコイルがOFFした場合、接点はONしません。再度、コイルがONした場合、タイマーは0からカウントアップします。
- ・ 計数値は0～65535まで更新し、オーバーフロー時は0からカウントを繰り返します。
- ・ 100msタイマーで使用する場合、オンディレイタイマーのコイルのON/OFF検出は、ラダー回路の実行周期（シーケンスサイクル）とは非同期に、100msごとに行われます（10msタイマーの場合は、10msごとに行われます）。コイルがONしている時間が100msより短い場合、コイルのONが検出されず、オンディレイタイマーが動作しない場合があります。オンディレイタイマーを確実に動作させるには、コイルのON時間が100ms以上になるようにラダー回路を作成してください。

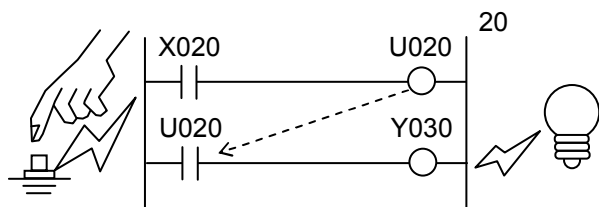
## U ワンショットタイマー

ナンバーの範囲	000～0FF
設定値	0～9999 (0.0～999.9秒)
誤差	最小100ms + 1シーケンスサイクル
セットリングパルス幅 (*)	最小100ms

ワンショットタイマーのコイルがONしてから設定値の時間だけ接点がONします。設定値の範囲は、0.0～999.9秒で0.1秒単位に設定できます（S10V 基本システムからの設定は、0～9999となります）。

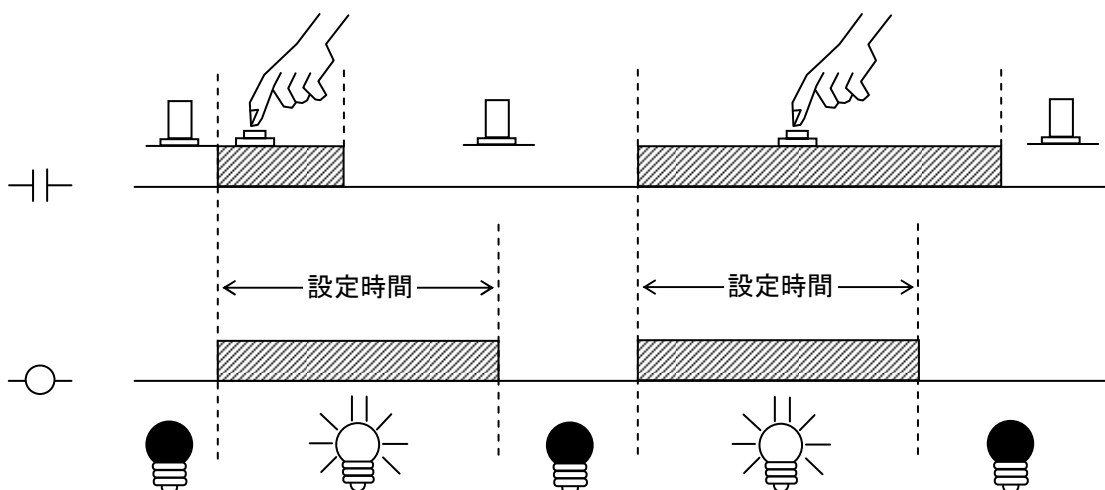
(\*) セットリングパルス幅とは、ワンショットタイマーのコイルをONする接点がONしていない最小時間のことを意味します。

### ● 回路例



図に示した回路の場合、プッシュボタンを押すと、ランプ（Y030）が設定時間（この場合2秒）だけ点灯します。

### ● タイムチャート





(注)

- ワンショットタイマーのコイルのON/OFF検出は、ラダー回路の実行周期（シーケンスサイクル）とは非同期に、100msごとに行われます。コイルがONしている時間が100msより短い場合、コイルのONが検出されず、ワンショットタイマーが動作しない場合があります。ワンショットタイマーを確実に動作させるには、コイルのON時間が100ms以上になるようラダー回路を作成してください。
- ワンショットタイマーのコイルが設定時間経過前にOFF状態に変化しても、設定時間が経過するまではワンショットタイマーの接点（Uレジスター）はON状態になります。つまり、ワンショットタイマーのコイルはカウント開始後から設定時間経過するまでの間はワンショットタイマーのコイルの状態（ON/OFF）にかかわらず、カウントアップします。



## C アップダウンカウンター

ナンバーの範囲		CU	00~FF
		CD	
		CR	
設定値	0~9999カウント		
セットリングパルス幅 (*)	最小1シーケンスサイクル		
セット、リセット同時入力時	リセット優先		
停電時	不揮発		

アップダウンカウンターは、アップカウンター (CU) とダウンカウンター (CD) から構成され、計数値はアップカウンターのコイル立ち上がり時にカウントアップ (+1) し、ダウンカウンターのコイル立ち上がり時にカウントダウン (-1) します。

カウンター接点 (C0) は、計数値が設定値を超えたときONします。リセットコイル (CR) は計数値を0クリアし、カウンター接点をOFFします。

CU : アップカウンター

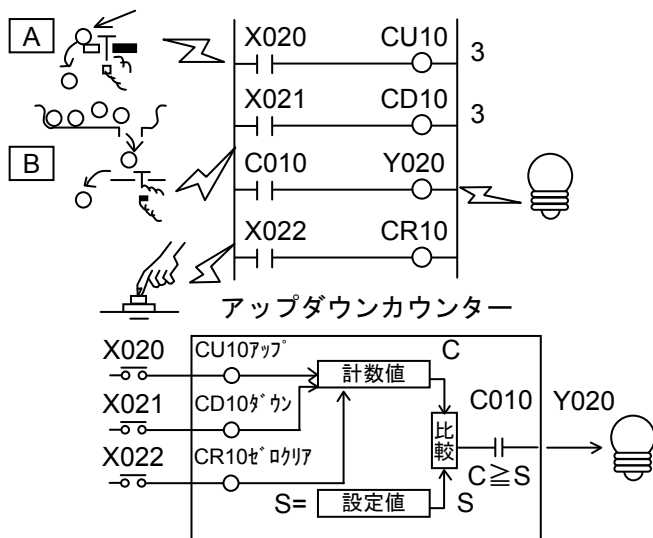
CD : ダウンカウンター

CR : リセット

C0 : カウンター接点

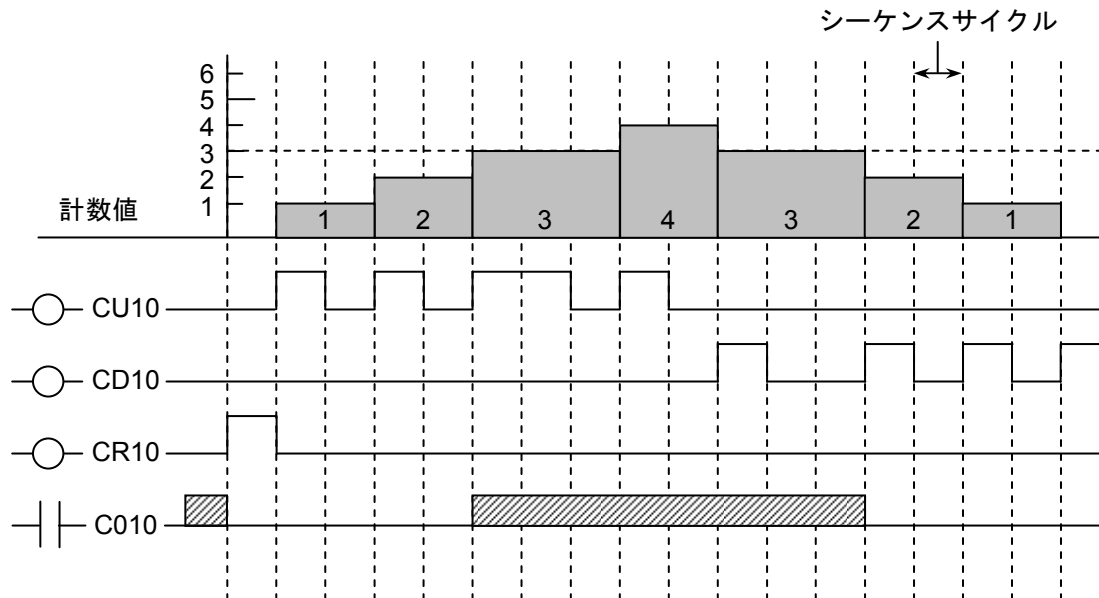
(\*) セットリングパルス幅とは、アップカウンター、ダウンカウンター、リセットのコイルをONする接点がONしていなければならぬ最小時間のことを意味します。

### ● 回路例



- ・ 図に示した回路の場合、上方からバスケットに入ったボールの数をスイッチ **A** (X020) でカウントし、下方へ落ちたボールの数をスイッチ **B** (X021) でカウントし、現在バスケットに入っているボールの数を計数します。
- ・ バスケットに入っているボールの数が3個以上になるとランプ (Y020) が点灯します。また、押しボタン (X022) を押し、計数値が0クリアされ、ランプが消灯します。

## ● タイムチャート



(注)

- ・アップカウンタの場合、計数值は設定値を超えてもカウントされ続けます。計数值がオーバーフロー（0xFFFFを超えた場合）すると、0から再度カウントします。そのとき、ON状態のカウンター接点はOFF状態に変わります。
- ・ダウンカウンタの場合、計数值が0になるとカウントダウンは停止します。
- ・カウンターコイルが立ち上がる（OFF→ON）シーケンスサイクルの途中でPLCの電源を切→入した場合、コイルの状態はONしますが、計数が1カウントされない場合があります。

この現象を回避するには、下記事項を守ってください。

- ① ユーザーは、コイルの状態をOFF→ONに変化させないように安定させてからPLCの電源を切ってください。
- ② 不意の電源断に対してはUPSを採用してください。

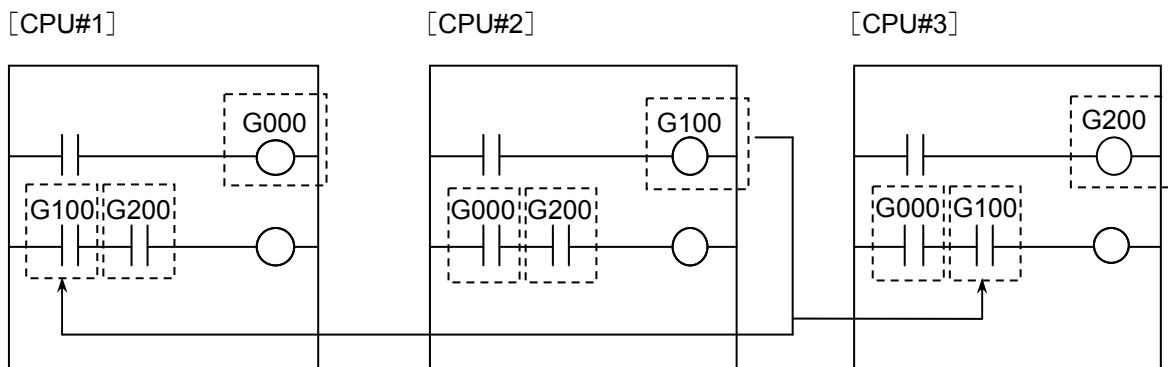
## G グローバル リンクレジスタ

ナンバーの 範囲	000~FFF
-------------	---------

グローバルリンクレジスタ（Gレジスタ）は、CPU間リンクモジュール（オプション）などを実装時に、CPU間のインターロック情報などを交換するためのレジスタです。

動作はコイルのON（またはOFF）で、他CPUの同じナンバーの接点がON（またはOFF）します。

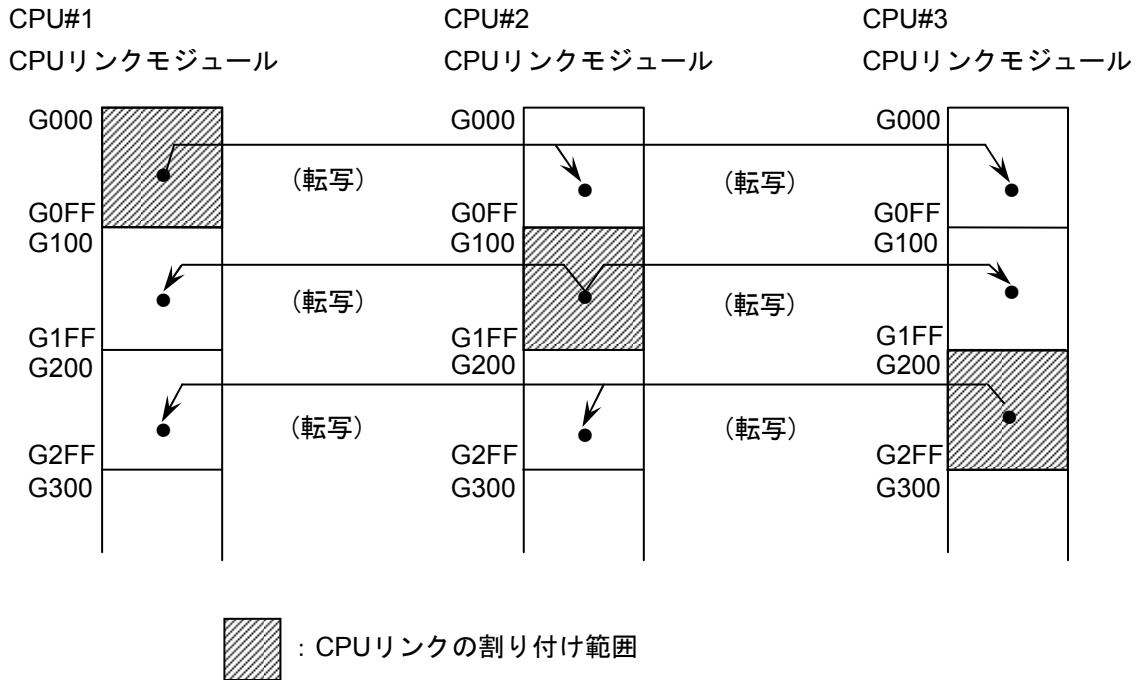
### ● ラダープログラム回路例（CPU間リンクモジュール使用時）



#### [動作]

- CPU#1のG000のコイルがON（OFF）した場合、CPU#2およびCPU#3のG000のa接点がON（OFF）します。
- CPU#2のG100のコイルがON（OFF）した場合、CPU#1およびCPU#3のG100のa接点がON（OFF）します。
- CPU#3のG200のコイルがON（OFF）した場合、CPU#1およびCPU#2のG200のa接点がON（OFF）します。

● CPUリンクモジュールの動作



- CPU#1のGレジスタエリアのG000～G0FFの内容をCPU#2およびCPU#3の同レジスタエリアに転写します。
- CPU#2のGレジスタエリアのG100～G1FFの内容をCPU#1およびCPU#3の同レジスタエリアに転写します。
- CPU#3のGレジスタエリアのG200～G2FFの内容をCPU#1およびCPU#2の同レジスタエリアに転写します。

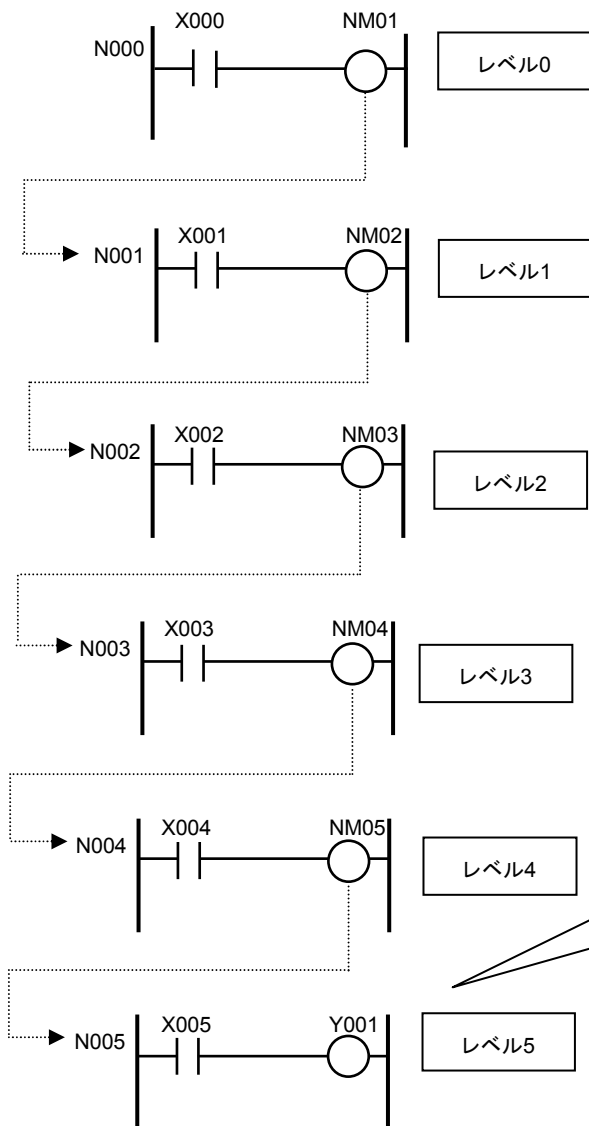
## N ネスティングコイル

ナンバーの 範囲	000～OFF
ネスティング 最大レベル	4レベル

シーケンスプログラムを制御対象プラントごとに分割し、モジュール化するための機能を持ちます。ネスティングコイルのうち、N000をマスターNコイル、N001～N0FFをサブNコイルと呼びます。マスターNコイルはLPU RUNの場合、シーケンスサイクルごとに起動され、サブNコイルは、マスターNコイルまたはサブNコイルから呼び出されます。

ネスティングコイルは、Nコイルの立ち下がり（ON→OFF）時に使用されているコイルをOFFするマスターコントロール（NM）と、前回の状態を保持するゾーンコントロール（NZ）の2種類から選択します。また、N（ネスティングコイル）は、最大4レベルまでネスティングできます。4レベルを超えたネスティングは、ラダープログラム実行時にエラーになります。

<最大ネスティングを超えたマスターコントロールの例>



レベル4を超えたネスティングでラダープログラムを実行した場合、LPUモジュールの「ERR」LED（赤色）が点灯し、ラダープログラムが停止します。

（注）マスターNコイル（NコイルNo.00）は、接点またはコイルに指定できません（入力エラーになります）。

(1) マスターコントロール (NM)

マスターコントロールには、＜マスターコントロールの動作モード＞に示したように2つの動作モードがあります。モードは、S10V ラダー図システム（型式：S-7895-02）の [ユーティリティ] - [PCs エディション] - [容量変更] で切り替えます。操作方法は、「ソフトウェアマニュアル オペレーション S10V ラダー図 For Windows®（マニュアル番号 SVJ-3-131）」を参照してください。

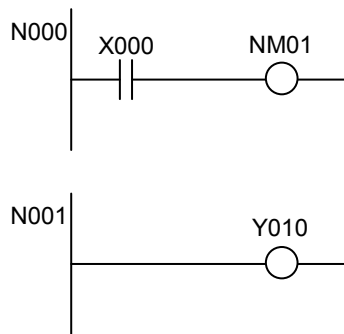
＜マスターコントロールの動作モード＞

通常モード：マスターコントロールコイル立ち下がり（ON→OFF）のタイミングで、立ち下がったマスターコントロールコイル以下のネスティングコイルで使用されている立ち上がり（立ち下がり）エッジ接点、ノーマルコイルがOFFになります。

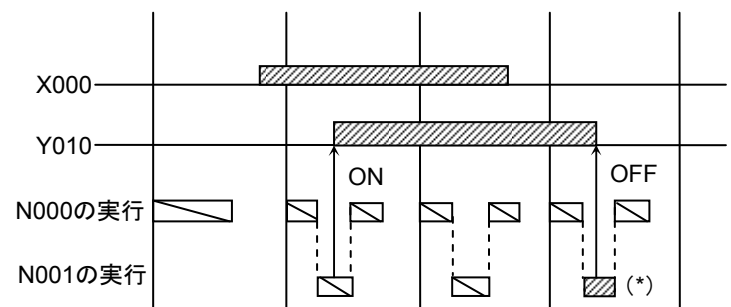
0出力モード：通常モード動作に加えてセットコイル、リセットコイルもOFFになります。

● 回路例

N000からN001に対してマスターコントロールをします。



● タイムチャート

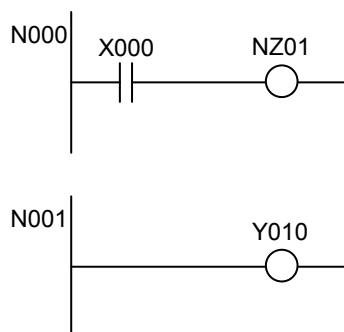


(\*) NM01の立ち下がりのとき、N001で使用されているコイルをOFFします。マスターコントロールの動作モードでOFFされるコイルは、設定したマスターコントロールの動作モードにより決まります。

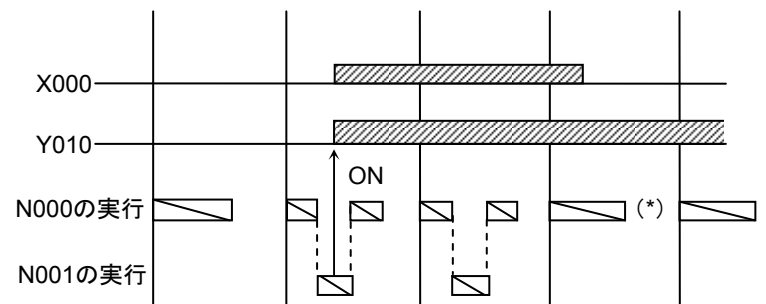
(2) ゾーンコントロール (NZ)

● 回路例

N000からN001に対してゾーンコントロールをします。




● タイムチャート



(\*) NZ01の立ち下がりのとき、マスターコントロールの場合と異なりN001で使用されているコイルは前回の状態を保持します。

## P プロセスレジスター

ナンバーの 範囲	001~080
起動方法	レベル起動 

ユーザーが、C言語やアセンブラ言語で作成したコンピューター言語によるプログラム（以下、タスクと呼びます。）をラダープログラムから起動するためのレジスターです。

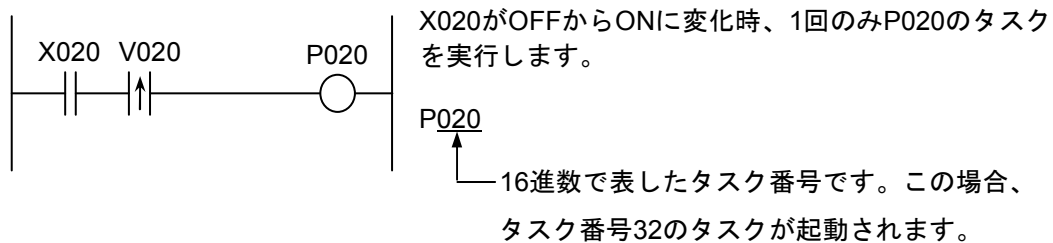
このコイルをONすることにより、ナンバー（タスク番号）に対応したタスクが起動されます。タスクの実行は、CMUモジュール側で行われます。

＜プロセスレジスターの割り付け＞

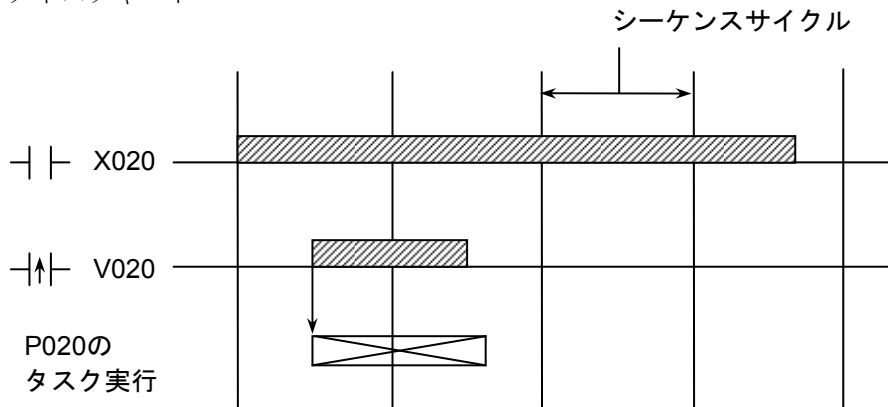
区 分	ナンバー	名 称	内 容
ユーザー 作成	P001	イニシャル タスク	LPUモジュールをリセットまたは停復電したときに必ず起動されるタスクです。システムをイニシャライズするプログラムを割り付けてください。
	P002 ~ P080	ユーザー タスク	ユーザーが作成したプログラムを割り付けてください。

### (1) ON時1回実行

#### ● 回路例

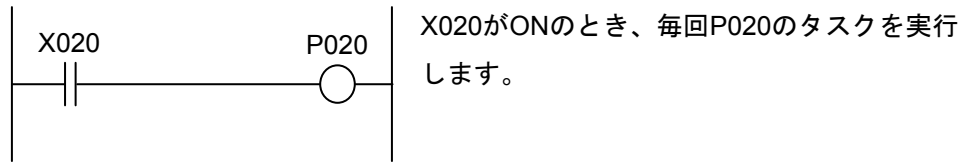


#### ● タイムチャート

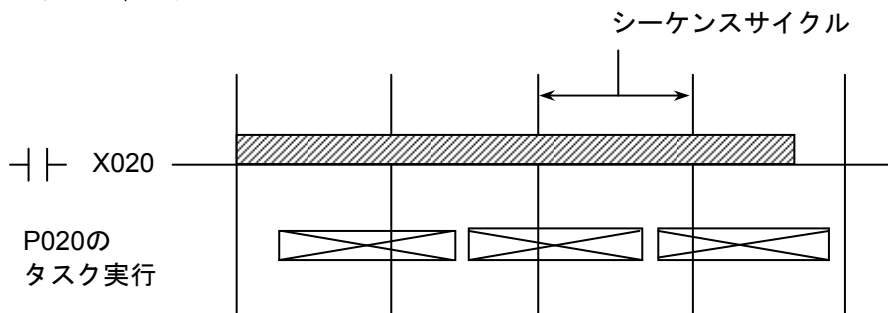


(2) ON時毎回実行

● 回路例



● タイムチャート



(注)

- ・タスクを登録していないコイルをONした場合、何も実行されません。
- ・このコイルで起動できるタスクは、タスク番号1～128（プロセスレジスターナンバーでは001～080）までです。タスク番号129～255は、このコイルから起動できません。
- ・タスクの実行はCMUモジュールで行われます。CMUモジュールを実装していない場合は、コイルがON/OFFするのみでタスクは起動されません。
- ・タスクの実行時間より短い間隔で同じナンバーのPコイルをONした場合のタスク動作は下記になります。

指定タスクがCPU実行待ち状態の場合

（起動されているが実行されていない状態）：1回のみCPMSが多重起動を記憶し、タスク実行終了後、再度タスクが起動されます。

指定タスクがCPU実行待ち状態以外の場合：2回まで多重起動を記憶します。

- ・LPUモジュールはCMUモジュールへタスク起動要求を発行後、CMUモジュールからの応答を監視します。一定時間監視し、応答がない場合はラダープログラムの実行を停止します。その場合の監視時間は、ラダーウォッチドッグタイマー監視時間の設定値（50～10000ms、デフォルト2000ms）と10000msを比較し小さい方の値になります。
- ・プロセスレジスター（レジスター名：P）によるタスク起動機能は、LPUモジュールとCMUモジュールのモジュールレビジョンが下記の場合のみ使用できます。下記以外の組み合わせの場合、プロセスレジスターを使用してもタスクが実行されません。

モジュール名称	モジュール型式	モジュールレビジョン
LPU（基本モジュール）	LQP510	D以降
CMU	LQP520	B以降



## E イベントレジスター

ナンバーの範囲	000～FFF
USER LED点灯範囲	000～1FF
4チャンネルアナログ、 パルスカウンター 入出力範囲	400～FFF

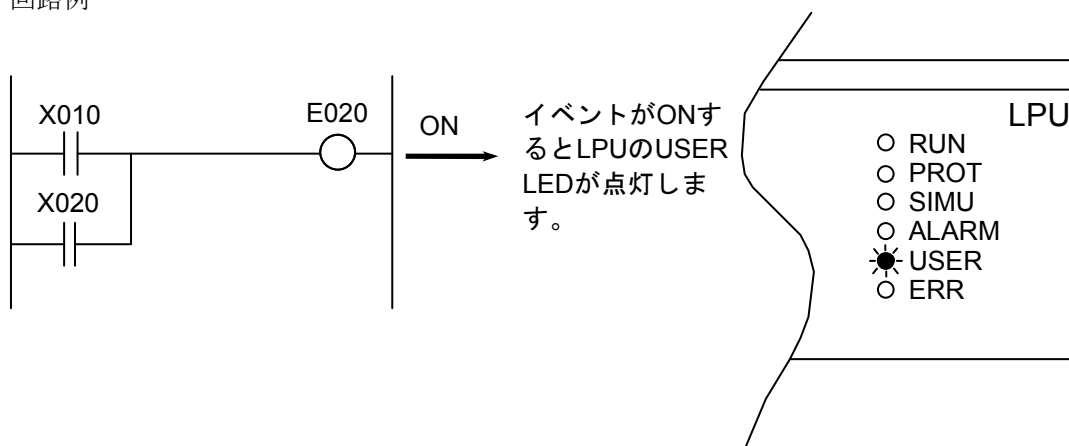
ユーザーエラーなどのイベント情報を出力します。出力したイベント情報は、S10V 基本システムのイベントレジスターモニター機能により監視できます。また、E000～E01FいずれかがONした場合、LPUモジュールのUSER LEDが点灯します。E000～E01FがすべてOFFした場合、USER LEDが消灯します。

ただし、S10V 基本システムのイベントレジスターモニター機能で表示される範囲およびLPUモジュールのUSER LEDが点灯する範囲は、E000～E1FFまでとなります。

また、ナンバー400～FFFは、リモートI/Oに接続されたアナログモジュールおよびパルスカウンターモジュールの入出力用として使用します。使用するためには、S10V ラダー図システム（型式：S-7895-02）の [ユーティリティ] - [PCsエディション] - [アナログカウンタ] で設定してください。操作方法は、「ソフトウェアマニュアル オペレーション S10V ラダー図 For Windows®（マニュアル番号 SVJ-3-131）」を参照してください。

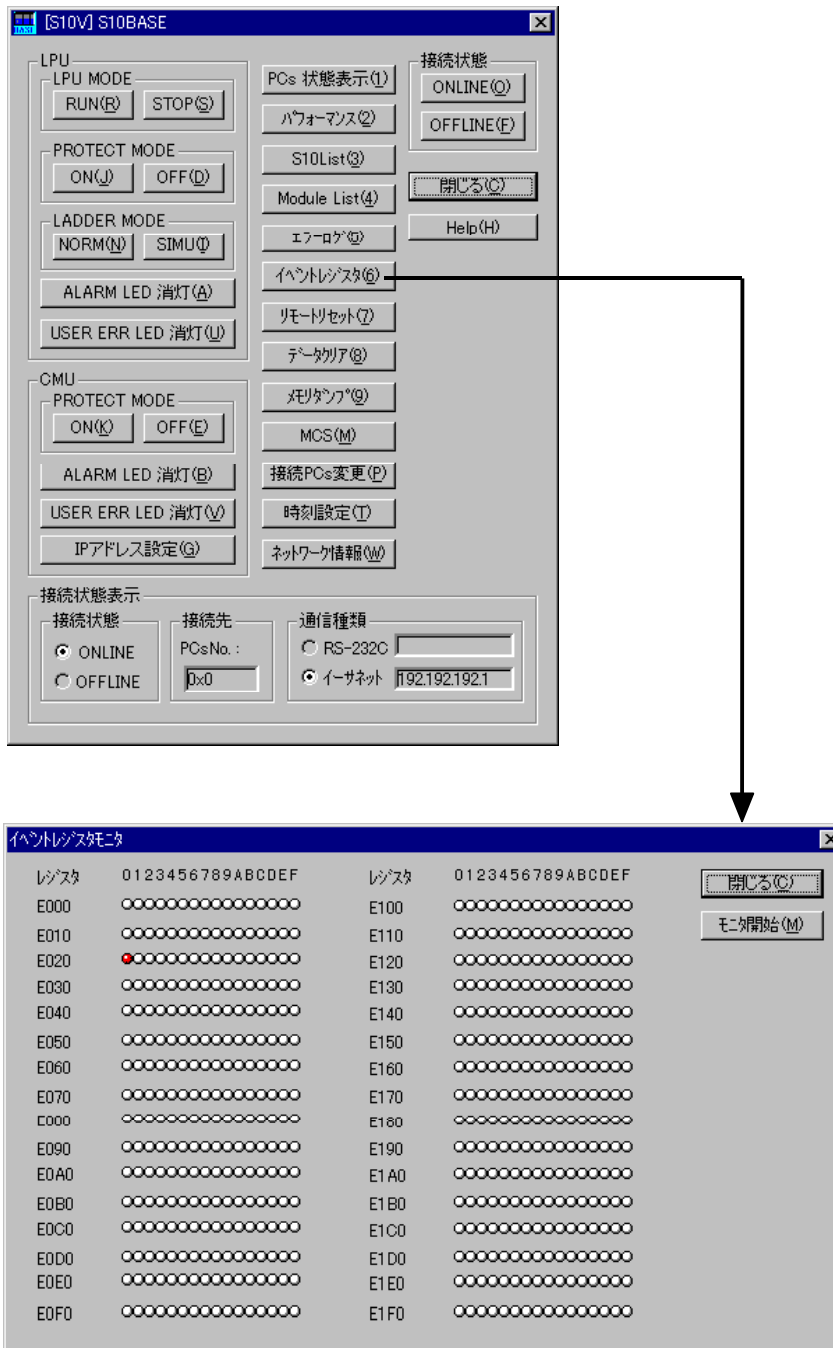
（注）全種類のアナログモジュールが、イベントレジスターを使用するわけではありません。イベントレジスターを使用するかどうかは、アナログモジュールの取扱説明書を参照してください。

### ● 回路例



イベント情報の確認は、S10V 基本システムのイベント情報表示機能で確認します（下図参照）。

### S10V 基本システムによるイベントレジスターモニター



## V エッジ接点

ナンバーの 範囲	000~FFF
-------------	---------

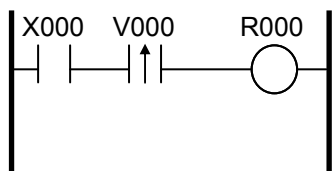
エッジ接点には立ち上がりエッジ検出接点（↑）と立ち下がりエッジ検出接点（↓）があり、それぞれのエッジを検出した1シーケンスサイクルだけONします。

なお、同じナンバーは立ち上がりエッジ接点および立ち下がりエッジ接点として使用できません。

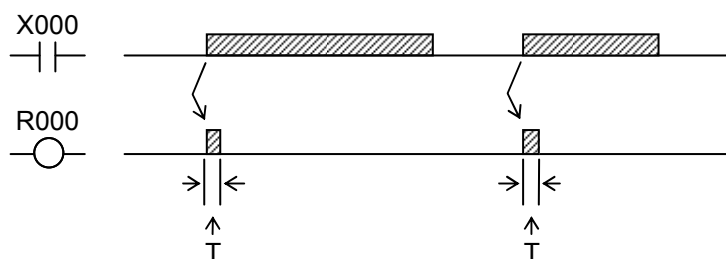
### (1) 立ち上がりエッジ接点

立ち上がり（OFF→ON）を検出した場合に、1シーケンスサイクルのみONします。

<回路例>



<タイムチャート>

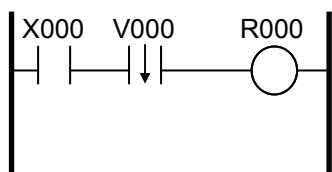


T: 1シーケンスサイクル

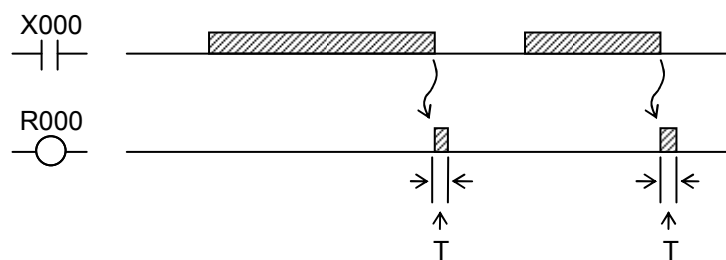
### (2) 立ち下がりエッジ接点

立ち下がり（ON→OFF）を検出した場合に、1シーケンスサイクルのみONします。

<回路例>



<タイムチャート>



T: 1シーケンスサイクル

(注) 同じナンバーの立ち上がり／立ち下がりエッジ接点を、同一プログラム内の複数個所で使用しないでください。プログラムが正常に動作しません。

このページは白紙です。

## Z ゼットレジスター

ナンバーの範囲	000～3FF
上位割り込み発生用	204
RS-422モジュール 使用時の上位 割り込み	200～203

Z204は、LPUモジュールのUP LINKに接続された上位計算機へH-7338方式の割り込みを発生させるレジスターです。Z200～Z203はRS-422モジュール（オプション）から上位計算機へ割り込みを発生させるレジスターです。Z200～Z203の詳細は、「ユーザーズマニュアル オプション RS-232C/422 (LQE560/565) (マニュアル番号 SVJ-1-121)」を参照してください。Z200～Z204以外のナンバーは使用できません。

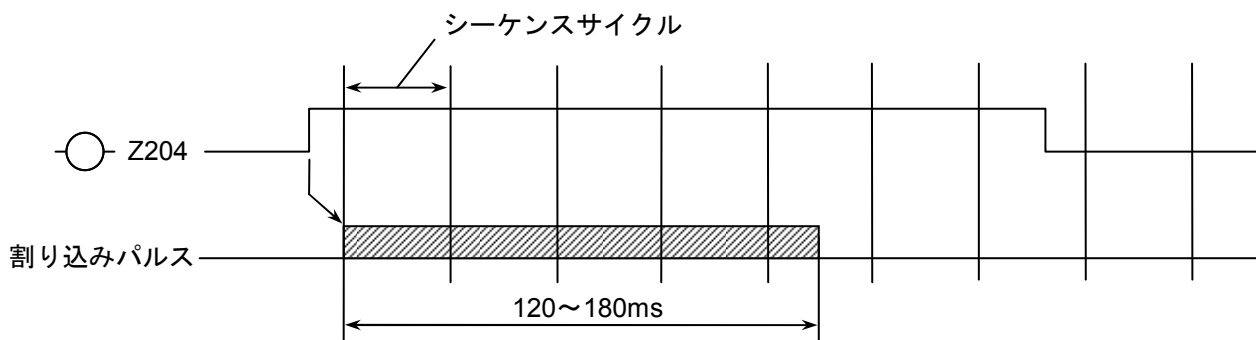
セットリングパルス幅：最小1シーケンスサイクル

<上位計算機への割り込みを発生させる回路例>



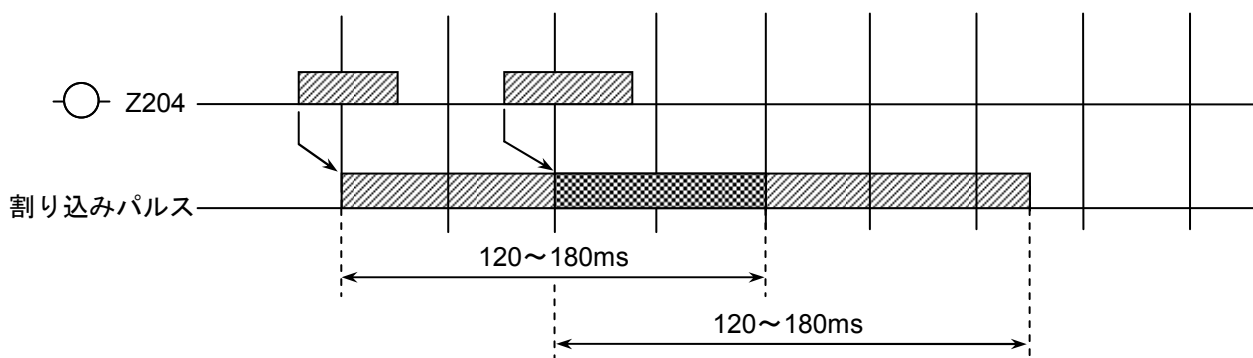
上位計算機への割り込み発生用コイルZ204の立ち上がり（OFF→ON）を検出すると、割り込みパルスを120～180ms間発生します。この処理はシーケンスサイクルと同期して行われます。割り込み発生中に再度Z204の立ち上がりを検出した場合は、再度120～180ms間割り込みパルスが発生します。

<タイムチャート>



<割り込みパルスが伸びる場合>

上記回路で、Z204のON⇔OFFが120～180ms以内で2回以上ON→OFFした場合、多重割り込みが発生しパルス幅が長くなります。



このようなことを防ぐには、割り込みを受け付ける上位計算機との間でインターロックを取ってください。

このページは白紙です。

## S システムレジスター

ナンバーの 範囲	000～BFF
-------------	---------

システムの動作などを反映した読み込み専用のレジスターです。

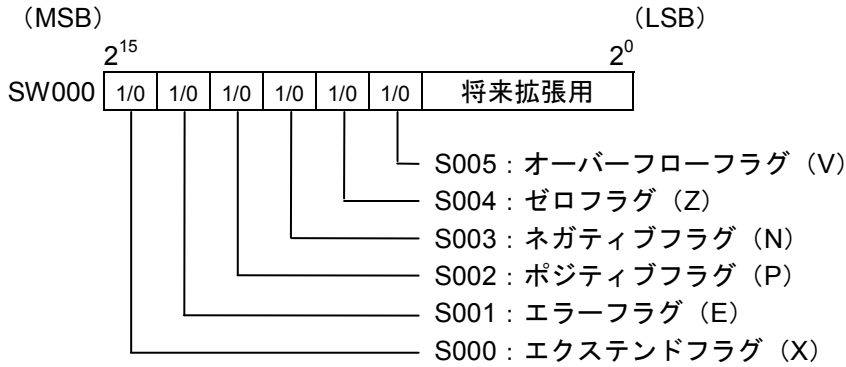
システムレジスター一覧を表1-6に示します。

表1-6 システムレジスター一覧

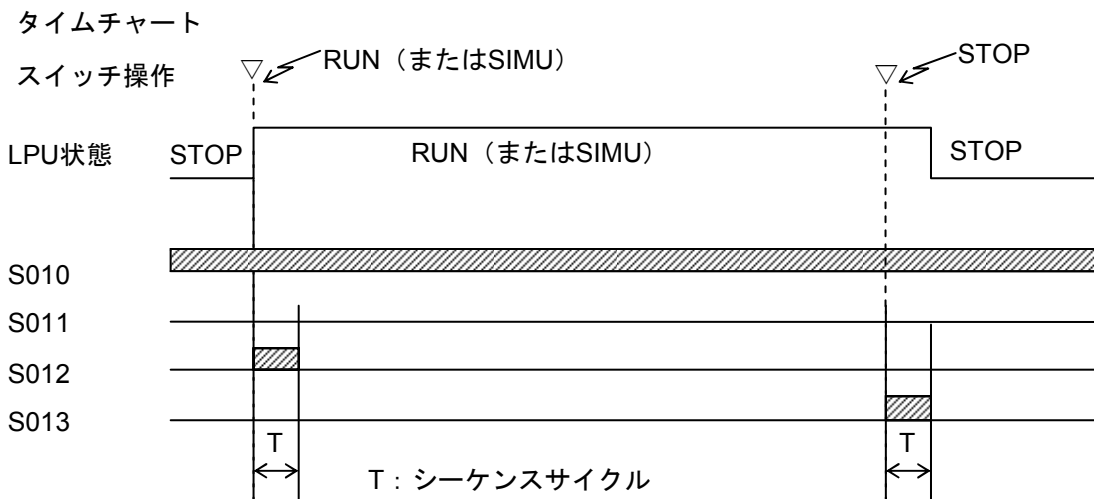
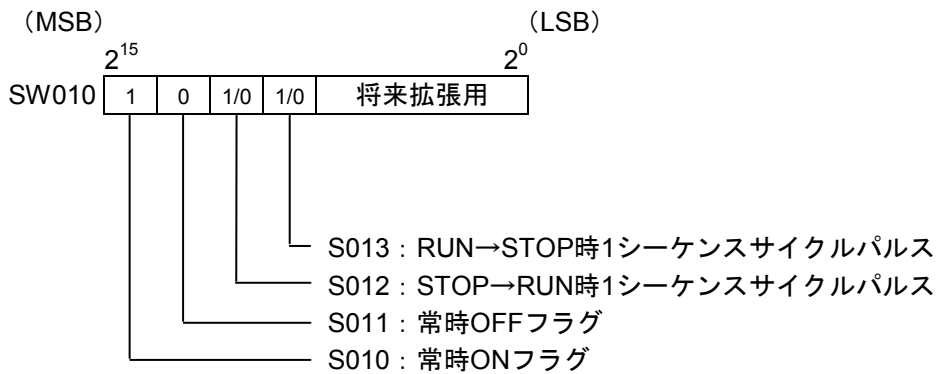
No.	レジスター ナンバー	概要
1	S000～S00F	演算ファンクションフラグレジスター
2	S010～S01F	ラダープログラム制御レジスター
3	S020～S02F	HI-FLOW応用命令実行結果フラグレジスター
4	S030～S0FF	システム予約
5	S100～S15F	ラダープログラム制御カウンター
6	S160～S1FF	システム予約
7	S200～S20F	時刻制御レジスター
8	S210～S27F	システム予約
9	S280～S2EF	時刻格納レジスター
10	S2F0～S2FF	システム予約
11	S300～S47F	リモートI/O状態レジスター
12	S480～S4FF	システム予約
13	S500～S68F	オプションモジュール状態レジスター
14	S690～S6AF	イーサネット通信実行結果フラグレジスター
15	S6B0～S8FF	システム予約
16	S900～S93F	シーケンスサイクルスキャンタイム
17	S940～S97F	ラダー実行時間
18	S980～S9BF	オプションモジュール状態レジスター (D.NET)
19	S9C0～S9FF	イーサネット通信実行結果フラグレジスター
20	SA00～SA8F	オプションモジュール状態レジスター (J.NET/IR.LINK)
21	SA90～SAFF	システム予約
22	SB00～SB1F	LPUユニットI/O情報
23	SB20～SBEF	システム予約
24	SBF0～SBFF	LPUステータスレジスター

(1) 演算ファンクションフラグレジスター

システム演算ファンクション命令実行後のフラグの状態を示したレジスターです。ラダー回路のモニター、MCS機能からは参照できません（常にOFFと表示されます）。



(2) ラダープログラム制御レジスター



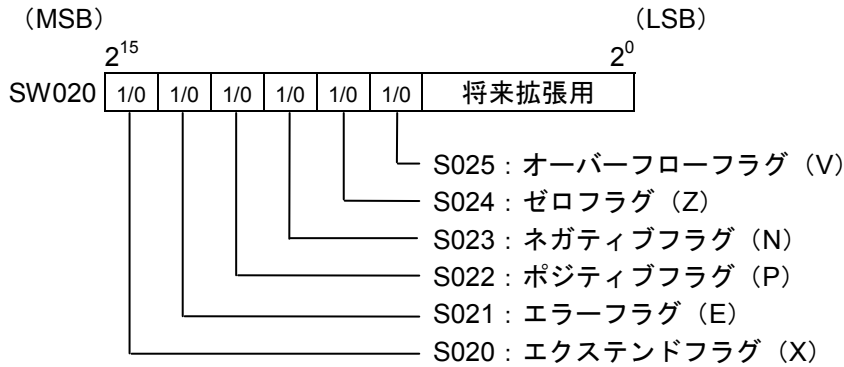
(注) 停電した場合は、S010, S011, S012, S013はONしません。



## S システムレジスタ

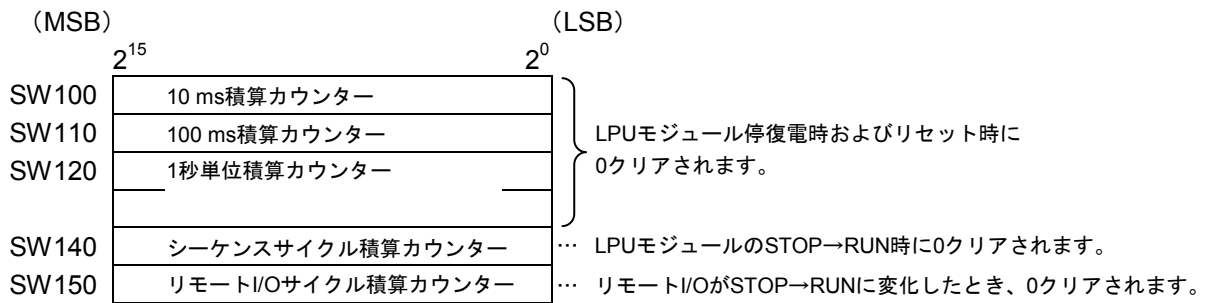
### (3) HI-FLOW応用命令実行結果フラグレジスタ

HI-FLOW応用命令実行後のフラグの状態を示したレジスタです。



### (4) ラダープログラム制御カウンター

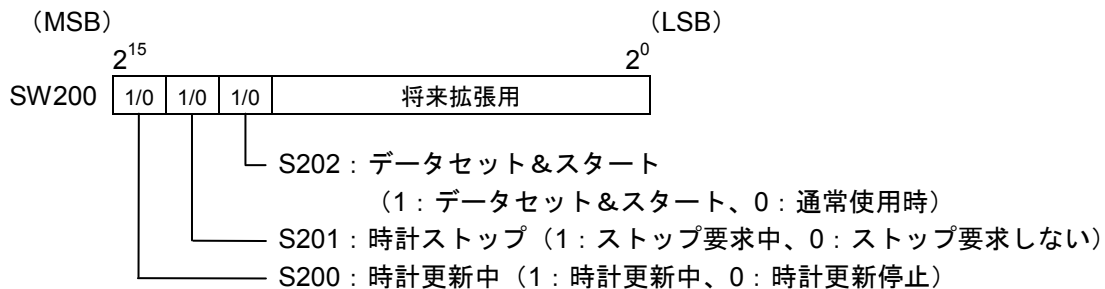
シーケンス制御で使用できる積算カウンターです。



- 全カウンターとも、オーバーフロー時は「0」から再カウントします。
- カウンターの精度はOSの割り込みによって処理しているため、±10%程度の誤差が生じます。

### (5) 時刻制御レジスタ

LPUモジュールの現在時刻の設定を制御するレジスタです。LPUモジュールの現在時刻を設定するときに使用します。



## (6) 時刻格納レジスター

年, 月, 日, 時, 分, 秒, 曜日を格納するレジスターです。LPUモジュールの時刻を設定するときは、このレジスターに時刻情報を設定します。データはバイナリータイプで格納します。

	(MSB) 2 <sup>15</sup>	2 <sup>8</sup> 2 <sup>7</sup>	(LSB) 2 <sup>0</sup>
SW280	未使用		秒
SW290	未使用		分
SW2A0	未使用		時
SW2B0	未使用		日
SW2C0	未使用		月
SW2D0	年 (西暦)		
SW2E0	未使用		曜日

秒 : 0~59で設定

分 : 0~59で設定

時 : 0~23で設定

日 : 1~31で設定

月 : 1~12で設定

年 : 1970~2069で設定

曜日 : 1~7で設定

(1=日, 2=月, 3=火, 4=水, 5=木, 6=金, 7=土)

## S システムレジスタ

### (7) リモートI/O状態レジスタ

リモートI/Oの登録状態、タイムアウト、FUSE断ステーション情報を示すレジスタです。

<レジスタの割り付け>

S300	登録 ステーション	・ 現在回線に接続されていて、1度でも正常に応答があったステーションに対応したレジスタに「1」が設定されます。 (*)
S380	タイムアウト ステーション	・ 登録されているステーションで、タイムアウトエラーが発生しているステーションに対応したレジスタに「1」が設定されます。 (*)
S400	FUSE断 ステーション	・ 登録されているステーションで、FUSE断（DOモジュールのヒューズ切れ）が発生しているステーションに対応したレジスタに「1」が設定されます。 (*)
S47F		(*) 各ステーションとビットの対応

No.	XまたはYの ナンバー	登録ステーション	タイムアウトステーション	FUSE断ステーション
0	000~00F	S300	S380	S400
1	010~01F	S301	S381	S401
2	020~02F	S302	S382	S402
3	030~03F	S303	S383	S403
4	040~04F	S304	S384	S404
5	050~05F	S305	S385	S405
6	060~06F	S306	S386	S406
7	070~07F	S307	S387	S407
8	080~08F	S308	S388	S408
9	090~09F	S309	S389	S409
10	0A0~0AF	S310	S390	S410
11	0B0~0BF	S311	S391	S411
12	0C0~0CF	S312	S392	S412
13	0D0~0DF	S313	S393	S413
14	0E0~0EF	S314	S394	S414
15	0F0~0FF	S315	S395	S415
124	7C0~7CF	S37C	S3FC	S47C
125	7D0~7DF	S37D	S3FD	S47D
126	7E0~7EF	S37E	S3FE	S47E
127	7F0~7FF	S37F	S3FF	S47F

## (8) オプションモジュール状態レジスター

CPU間リンク、RS-232C/RS-422などのエラー情報が格納されるレジスターです。各モジュールの割り付けは以下ようになります。なお、このレジスターへのデータ設定は、各モジュールのシステムプログラムで行うため、詳細なビット構成は、各オプションモジュールのユーザーズマニュアルを参照してください。

<オプションモジュールレジスターの割り付け>

S500	システム 予約
S580	CPU間リンク モジュール情報
S5C0	RS-232C/RS-422 モジュール
S640	高速リモートI/O モジュール
S680	システム 予約
S6FF	

上記レジスターは、S10Vユニットの停復電またはLPUモジュールのリセットのタイミングで0クリアされます。

## S システムレジスター

### (9) シーケンスサイクル時間

シーケンスサイクルの測定結果を格納します。

	(MSB)		(LSB)
	$2^{15}$		$2^0$
SW900	最新測定データ (ms)		} LPUモジュールをSTOP状態からRUN状態に切り替えたタイミングで0クリアされます。
SW910	最大測定データ (ms)		
SW920	最小測定データ (ms)		
SW930	最近16回の平均データ (ms) (*)		

(\*) 平均データは、16回計測されるまで格納されません。

### (10) ラダー実行時間

ラダー実行時間の測定結果を格納します。HI-FLOWを使用している場合には、ラダー実行時間とHI-FLOW実行時間の合計を格納します。

	(MSB)		(LSB)
	$2^{15}$		$2^0$
SW940	最新測定データ (ms)		} LPUモジュールをSTOP状態からRUN状態に切り替えたタイミングで0クリアされます。
SW950	最大測定データ (ms)		
SW960	最小測定データ (ms)		
SW970	最近16回の平均データ (ms) (*)		

(\*) 平均データは、16回計測されるまで格納されません。

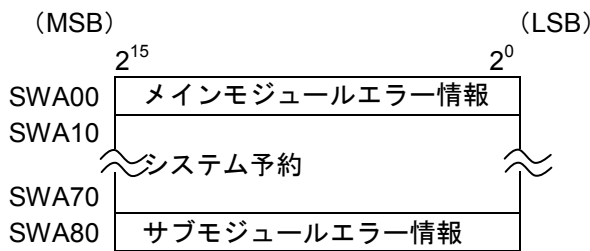
### (11) オプションモジュール状態レジスター (D.NET)

各モジュール (チャンネル0~3) に発生したエラー情報を格納するレジスターです。詳細は、「ユーザーズマニュアル オプション D.NET (LQE570/575) (マニュアル番号 SVJ-1-106)」を参照してください。

	(MSB)		(LSB)
	$2^{15}$		$2^0$
SW980	チャンネル0モジュールエラー情報		
SW990	チャンネル1モジュールエラー情報		
SW9A0	チャンネル2モジュールエラー情報		
SW9B0	チャンネル3モジュールエラー情報		

(12) オプションモジュール状態レジスター (J.NET/IR.LINK)

各モジュール（メインまたはサブモジュール）に発生したエラー情報を格納するレジスターです。詳細は、「ユーザーズマニュアル オプション J.NET (LQE540) (マニュアル番号 SVJ-1-104)」、「ユーザーズマニュアル オプション J.NET-INT (LQE545) (マニュアル番号 SVJ-1-107)」、および「ユーザーズマニュアル オプション IR.LINK (LQE546) (マニュアル番号 SVJ-1-117)」を参照してください。



(13) LPUユニットI/O情報

LPUモジュールのI/Oユニット情報を格納するレジスターです。各情報の1ビットが、1スロットに対応します。

LPUユニットI/O情報のビット構成



(\* ) PCT : パルスカウンター

スロットNo.	ビットレジスター		
	エラー情報	実装情報	I/O種別
0	SB00	SB08	SB10
1	SB01	SB09	SB11
2	SB02	SB0A	SB12
3	SB03	SB0B	SB13
4	SB04	SB0C	SB14
5	SB05	SB0D	SB15
6	SB06	SB0E	SB16
7	SB07	SB0F	SB17

## S システムレジスター

### (14) LPUステータスレジスター

現在のLPU状態を示すレジスターです。

<LPUステータスのビット構成>

	(MSB)	$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	(LSB)
SWBF0		1/0	1/0	*	1/0	1/0	*	1/0	*	1/0	1/0	1/0	1/0	*	1/0	1/0	0	
ビットNo.		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

\*は将来拡張用です。

ビット No.	ビット レジスター No.	各ビットの内容	
		ON (=1)	OFF (=0)
0	SBF0	STOP中	RUN中
1	SBF1	シミュレーション中	通常RUN中
2	SBF2	将来拡張用	
3	SBF3	プロテクトSW ON状態	プロテクトSW OFF状態
4	SBF4	リモートI/O動作中	リモートI/O停止中
5	SBF5	将来拡張用	
6	SBF6	ラダー書き換え中	ラダー書き換え完了
7	SBF7	将来拡張用	
8	SBF8	CELLエラー (*1) 警告	CELL正常
9	SBF9	タイムアウトエラーステーションあり	タイムアウトエラーステーションなし
A	SBFA	FUSE断ステーションあり	FUSE断ステーションなし
B	SBFB	オプションモジュールエラー (*2) あり	オプションモジュールエラー (*2) なし
C	SBFC	将来拡張用	
D	SBFD	GR (復電) またはリセット時に0クリア	
E	SBFE	LPUダウン中	LPU正常運転中
F	SBFF	—	LPU OS実行中

(\*1) CELLエラー：LPU内メモリアップ用バッテリー（電池）の容量が低下したことを意味します。

(\*2) オプションモジュールエラー：LPUからオプションモジュール内メモリアクセス時に、パリティエラーが発生したことを意味します。

## (15) イーサネット通信実行結果フラグレジスター

イーサネット通信命令の実行結果フラグを格納するレジスターです。

管理番号ごとに実行結果がシステムレジスターS9C0～S9EF、S690～S6AFに設定されます。

正常終了時は0、異常終了時は1が管理番号に該当するシステムレジスターに設定されます。

管理番号とは、使用できるソケットに対応付けている番号のことです。

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



# LR, LV ラダーコンバーター専用ワークレジスタ

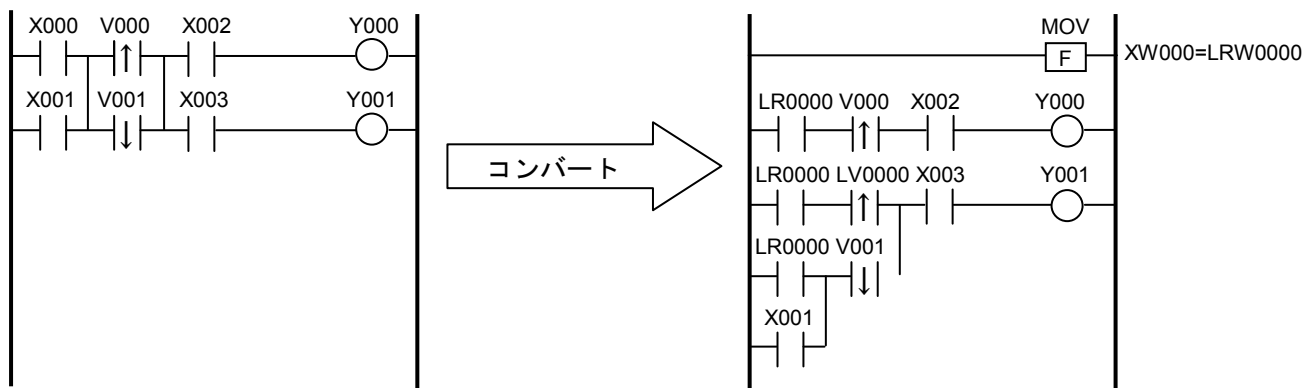
ナンバーの 範囲	0000~0FFF
-------------	-----------

S10/2αシリーズまたはS10miniシリーズで作成した右下がりラダープログラムを水平ラダープログラムに変換するとき、コンバーターが使用する専用レジスタです。ユーザーによる使用は控えてください。

LR：接点またはコイルに使用します。

LV：エッジ接点に使用します。

## ● 回路例



このページは白紙です。

## 1 ラダー命令

### 1.6 ラダーウォッチドッグタイマー

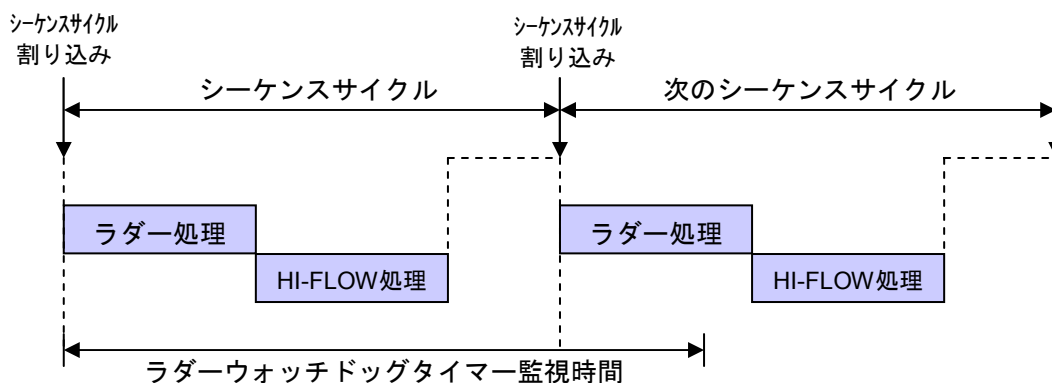
ラダーウォッチドッグタイマーは、ラダー処理とHI-FLOW処理が設定した監視時間内に処理が終了するかどうかを監視するためのものです。設定した監視時間内に処理が終了しなかった場合、ラダープログラムウォッチドッグタイマータイムアウトエラー（以降、ラダーWDTエラーと略します。）になり、次の動作をします。

- LPUのERR LED点灯、エラー詳細情報を記録し、LPUを停止します。
- ラダー、HI-FLOW、Cモードタスク、リモートI/O通信など、すべての動作が停止します（S10V 基本システムとの通信用RS-232C処理は動作します）。
- PCs OK信号がOFFします。

#### 1.6.1 ラダーウォッチドッグタイマー動作概要

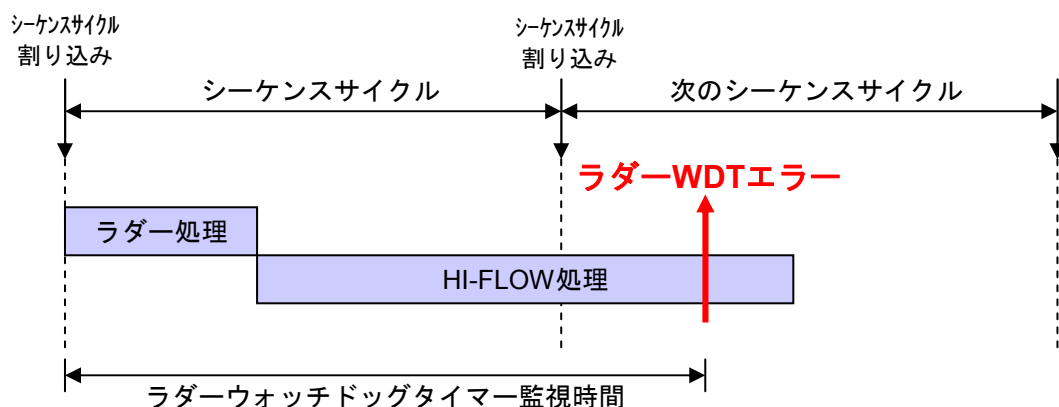
<通常動作>

下記のようにラダー処理とHI-FLOW処理（HI-FLOWを使用していない場合はラダー処理のみ）が監視時間内に終了する場合、シーケンスサイクル割り込みのタイミングでラダーウォッチドッグタイマーは再設定（リセット）されるためラダーWDTエラーは発生しません。



<タイムアウト検出時の動作>

ラダー処理とHI-FLOW処理（HI-FLOWを使用していない場合はラダー処理のみ）が無限ループなどの理由により監視時間内に終了しなかった場合、ラダーウォッチドッグタイマーがタイムアウトし、ラダーWDTエラーを検出して、ラダー処理とHI-FLOW処理を停止します。



### 1. 6. 2 ラダーウォッチドッグタイマー監視時間設定可能範囲

ラダーウォッチドッグタイマーの監視時間の設定可能範囲を以下に示します。また、出荷時のデフォルトは2000 [ms] に設定されています。

設定可能範囲：50～10000 [ms]

(注)

- ・監視時間の設定を変更する場合、ラダーおよびHI-FLOWの処理時間を考慮し、実際に要する時間よりも50ms以上大きい値を設定してください。
- ・ラダーウォッチドッグタイマーの再設定（リセット）はシーケンスサイクルのタイミングで行われます。したがって、監視時間をシーケンスサイクル設定値より短く設定した場合、ラダーが正常動作していてもラダーWDTエラーが発生してしまいます。これを防止するために、監視時間をシーケンスサイクル以下に設定した場合、監視時間はシーケンスサイクル+10msとして動作します。

### 1. 6. 3 ラダーWDTエラー発生時のエラー情報

ラダーWDTエラー発生時は、LPUモジュールのERR LEDが点灯します。発生しているエラーが、ラダーWDTエラーかそれ以外のエラーかは、S10V 基本システムのエラーログ表示機能で切り分けてください。ラダーWDTエラー発生時は、「エラーコード：0x1206、内容：ラダープログラムウォッチドッグタイマタイムアウトエラー」がS10V 基本システムに表示されます。S10V 基本システムの操作方法および説明は、「ユーザーズマニュアル 基本モジュール（マニュアル番号 SVJ-1-100）」を参照してください。

このページは白紙です。

## 2 演算ファンクション

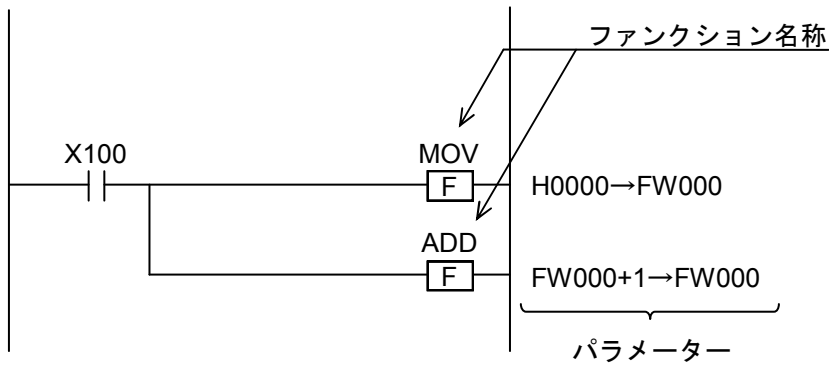
## 2 演算ファンクション

### 2.1 機能概要

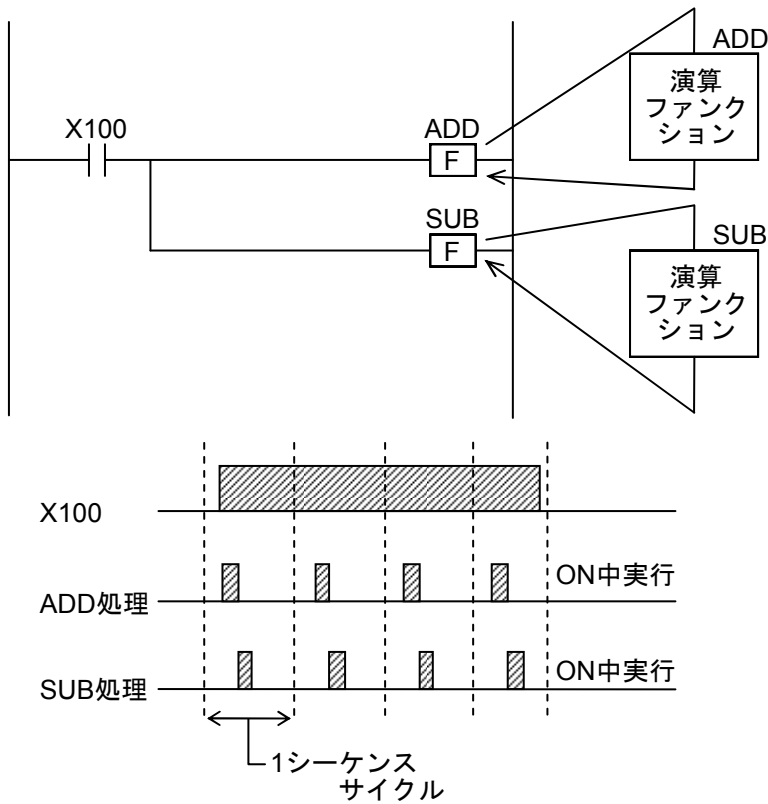
ラダープログラムで算術演算をしたい場合は、演算ファンクションを使用すると簡単にプログラムできます。

#### ● 演算ファンクションの動作

##### 【回路例】



##### 【動作】



入力条件が成立している間、シーケンスサイクルごとに毎回演算をします。

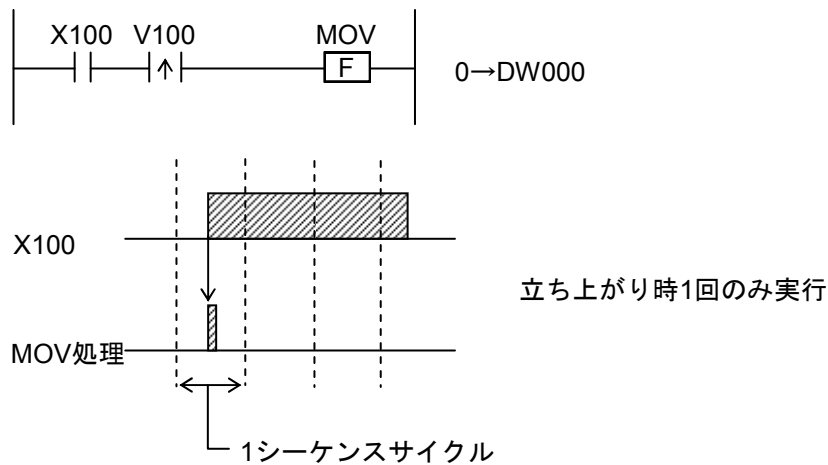
(1) パラメーター

演算ファンクションは、処理内容に対応してファンクション名称を割り付けてあります。それぞれのファンクションに対してパラメーターがあります。パラメーターにはレジスターおよび定数が指定できます。

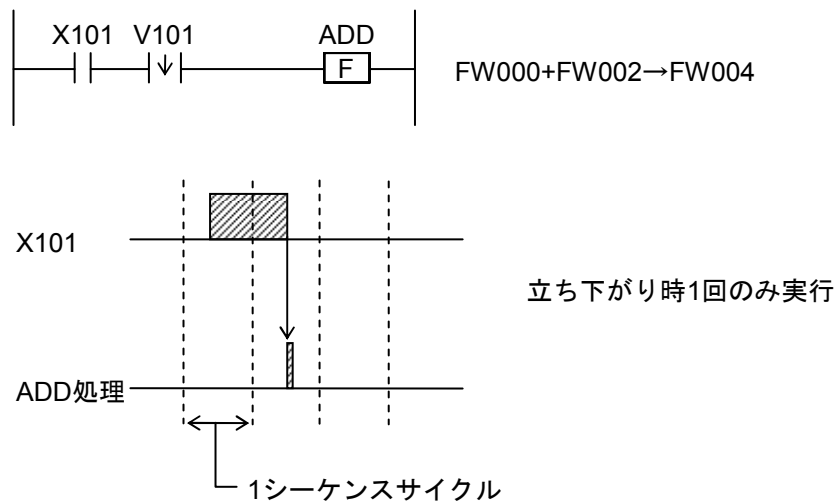
(2) 動作

演算ファンクションはコイルのON中、毎回起動されます。コイルONの立ち上がり時または立ち下がり時に1回のみ起動させたい場合は、立ち上がりエッジ接点または立ち下がりエッジ接点と組み合わせます。

(例1) 立ち上がりエッジ接点との組み合わせ



(例2) 立ち下がりエッジ接点との組み合わせ

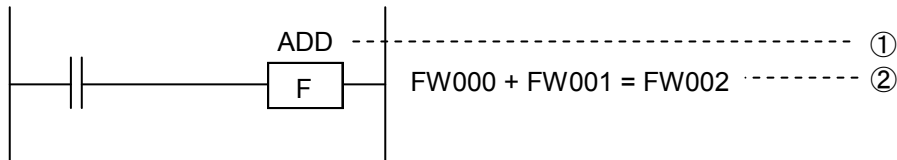




## 2 演算ファンクション

### 2.2 機能仕様

#### (1) 演算ファンクション構成



- ① ファンクション名称：演算ファンクションの機能名称を示します。
- ② パラメーター：演算の対象になるレジスターや定数データを示します。

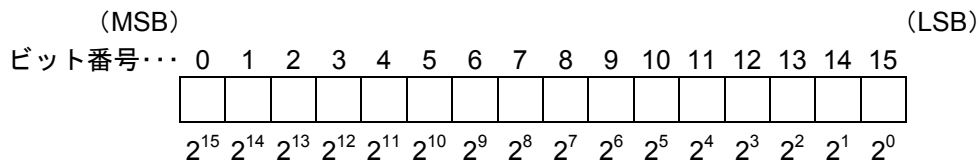
#### (2) データフォーマット

演算ファンクションで使用するデータにはワード、ロングワード、フローティングデータがあります。

##### ● ワード

ワードデータは、符号付き16ビット単精度整数です。

ワードデータは、以下に示すように、各ビットにビット番号が付けられています。



ワードデータで扱うデータの範囲は次のとおりです。

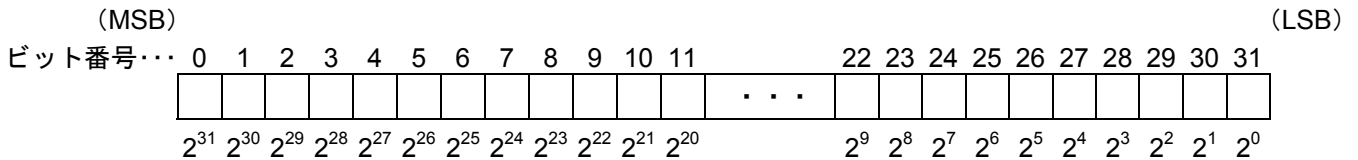
10進数：-32768～+32767

16進数：H8000～H7FFF（16進数はHで表します。）

(注1) ビットレジスター（X000, R123など）はワードデータとして扱います。その場合、データとしてはLSBのみ有効となります。詳細は「2.3.2 ビットレジスターの扱い」を参照してください。

(注2) オンディレイタイマー（T）、ワンショットタイマー（U）、およびアップダウンカウンタ（C）の計数值（TC\*\*\*, UC\*\*\*, CC\*\*\*：\*\*\*=ナンバー）、および設定値（TS\*\*\*, US\*\*\*, CS\*\*\*：\*\*\*=ナンバー）は、ワードデータとして扱います。

- ロングワード  
 ロングワードデータは、符号付き32ビット倍精度整数です。  
 ロングワードデータは、以下に示すように、各ビットにビット番号が付けられています。

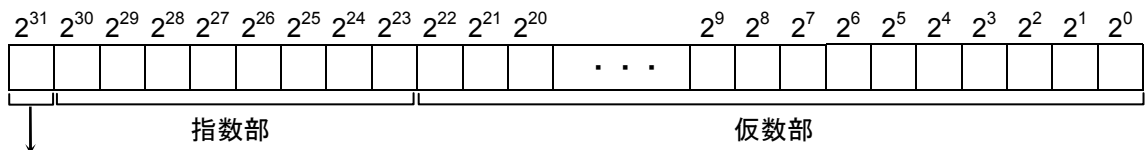


ロングワードデータで扱うデータの範囲は次のとおりです。

10進数 : -2147483648~+2147483647

16進数 : H80000000~H7FFFFFFF

- フローティング  
 フローティングデータは、32ビットの単精度浮動小数点データです。  
 フローティングデータは、以下に示すようなビット構成をしています。



仮数部符号  
 (0 : 正、1 : 負)

フローティングデータで扱うデータの範囲は次のとおりです。

$$0, \pm 2^{-126} \sim \pm 2^{128}$$

フローティング演算で異常発生時には以下に示す結果を返します。

無効演算 : 演算結果フラグのEビットがONします。結果を格納するレジスターの内容は変化しません。

0除算 : 演算結果フラグのEビットがONします。結果を格納するレジスターの内容は変化しません。

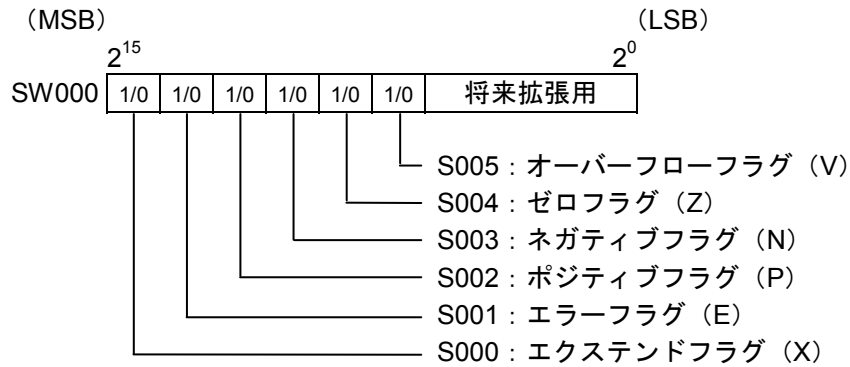
オーバーフロー : 絶対値が表現できる最大値になる有限数 ( $\pm 3.402823E38$ ) を返します。

アンダーフロー : 正しい符号を持った0を返します。

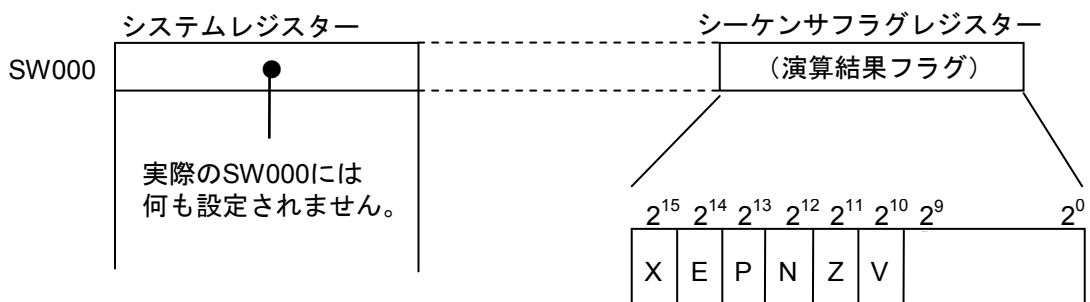
## 2 演算ファンクション

### (3) フラグの設定

演算ファンクションは、演算結果に従い各種フラグを設定します。以下にフラグの種類とフラグが設定されるエリア、およびフラグが設定される条件を示します。



(注1) S000～S00Fは、システム演算ファンクション命令実行後のフラグの状態を反映した読み出し専用のレジスターです。S000～S00Fはラダープログラムから参照できますが、ラダーツールのMCSおよびラダー回路モニターによる参照はできません。



(注2) フローティング演算についても、演算結果をこのフラグに反映します。

<フラグの設定条件>

No.	種別	フラグ					フラグ設定条件		
		X	E	P	N	Z	ワード時	ロングワード時	フローティング
1	ADD	-	-	-	-	●	V: 演算結果が-32768~32767のとき0、 以外のとき1	V: 演算結果が-2147483648~2147483647の とき0、以外のとき1	
2	ADD (フローティング)	-	●	-	-	-			E: 演算結果が異常 (*) のとき1、 以外のとき0
3	SUB	-	-	-	-	●	V: 演算結果が-32768~32767のとき0、 以外のとき1	V: 演算結果が-2147483648~2147483647の とき0、以外のとき1	
4	SUB (フローティング)	-	●	-	-	-			E: 演算結果が異常 (*) のとき1、 以外のとき0
5	INC	-	-	-	-	●	V: 演算結果が-32768~32767のとき0、 以外のとき1	V: 演算結果が-2147483648~2147483647の とき0、以外のとき1	
6	DEC	-	-	-	-	●	V: 演算結果が-32768~32767のとき0、 以外のとき1	V: 演算結果が-2147483648~2147483647の とき0、以外のとき1	
7	MUL	-	-	-	-	●	V: 演算結果が-32768~32767のとき0、 以外のとき1	V: 演算結果が-2147483648~2147483647の とき0、以外のとき1	
8	MUL (フローティング)	-	●	-	-	-			E: 演算結果が異常 (*) のとき1、 以外のとき0
9	DIV	-	●	-	-	●	E: 除数=0のとき1、以外のとき0 V: 商=32768のとき1、以外のとき0	E: 除数=0のとき1、以外のとき0 V: 商=2147483648 のとき1、以外のとき0	
10	DIV (フローティング)	-	●	-	-	-			E: 除数=0のとき1、以外のとき0 演算結果が異常 (*) のとき1、 以外のとき0
11	MOD	-	●	-	-	●	E: 除数=0のとき1、以外のとき0 V: 商=32768のとき1、以外のとき0	E: 除数=0のとき1、以外のとき0 V: 商=2147483648のとき1、以外のとき0	
12	SCL	-	●	-	-	●	E: 除数=0のとき1、以外のとき0 V: 演算結果が-32768~32767のとき0、 以外のとき1		
13	TST	-	-	●	●	●	P: データ>0のとき1、以外のとき0 N: データ<0のとき1、以外のとき0 Z: データ=0のとき1、以外のとき0		
14	BTD	-	●	-	-	●	E: データ<0のとき1、以外のとき0 V: データ>9999のとき1、以外のとき0	E: データ<0のとき1、以外のとき0 V: データ>99999999のとき1、以外のとき0	
15	DTB	-	●	-	-	-	E: あるディジット (4ビット) でHA~HFを検出時1、以外のとき0		
16	APB	-	●	-	-	-	E: H30~H39, H41~H46以外のデータ 検出時1、以外のとき0		
17	AUB	-	●	-	-	-			
18	DTS	-	-	-	-	●		V: 演算結果が-32768~32767のとき0、 以外のとき1	
19	ABS	-	-	-	-	●	V: データ=-32768のとき1、以外のとき0	V: データ=-2147483648のとき1、 以外のとき0	
20	NEG	-	-	-	-	●	V: データ=0のとき1、以外のとき0		
21	ECD	-	●	-	-	-	E: データ=0のとき1、以外のとき0		
22	ASL	-	-	-	-	●	V: シフト操作中に符号ビットが1回でも変化すれば1、それ以外0		
23	LIM	-	●	-	-	-	E: 上限値<下限値のとき1、以外のとき0		
24	LIM (フローティング)	-	●	-	-	-			E: 上限値<下限値のとき1、以外のとき0 演算結果が異常 (*) のとき1、 以外のとき0
25	BND	-	●	-	-	●	E: 上限値<下限値のとき1、以外のとき0 V: 演算結果が-32768~32767のとき0、 以外のとき1	E: 上限値<下限値のとき1、以外のとき0 V: 演算結果が-2147483648~2147483647 のとき0、以外のとき1	
26	BND (フローティング)	-	●	-	-	-			E: 上限値<下限値のとき1、以外のとき0 演算結果が異常 (*) のとき1、 以外のとき0
27	ZON	-	●	-	-	●	E: 上限値<下限値のとき1、以外のとき0 V: 演算結果が-32768~32767のとき0、 以外のとき1	E: 上限値<下限値のとき1、以外のとき0 V: 演算結果が-2147483648~2147483647 のとき0、以外のとき1	
28	ZON (フローティング)	-	●	-	-	-			E: 上限値<下限値のとき1、以外のとき0 演算結果が異常 (*) のとき1、 以外のとき0
29	TAN	-	●	-	-	-			E: 演算結果が異常 (*) のとき1、 以外のとき0
30	ASIN	-	●	-	-	-			E: データが-1.0~1.0の範囲外のとき1、 以外のとき0
31	ACOS	-	●	-	-	-			
32	EXP	-	●	-	-	-			E: 演算結果が異常 (*) のとき1、 以外のとき0
33	LOG	-	●	-	-	-			E: 指定値<0 演算結果が異常 (*) のとき1、 以外のとき0
34	上記以外	-	-	-	-	-	どのフラグも実行直前の値を保持している。		

— : 実行直前の値を保持

● : フラグ設定条件を参照

(\*) フローティングの演算結果が下記のとき異常

演算結果が0でなく、かつ $\pm 2^{-126} \sim \pm 2^{128}$ 範囲外のとき

## 2 演算ファンクション

### 2.3 演算ファンクションで使用するレジスター

「2.2 機能仕様」に示したように、演算ファンクションのパラメーターにはレジスターを指定できます。ここでは、演算ファンクションで使用するレジスターについて説明します。

#### 2.3.1 演算ファンクションで使用できるレジスター

表2-1に演算ファンクションで使用できるレジスター一覧を示します。各レジスターにはそれぞれ専用の用途とレジスター名を割り付けてあります。プログラミングや保守の効率化のために、各レジスターは割り付けた用途で使用して頂くことを推奨しますが、別の用途で使用することもできます。

表2-1 演算ファンクションで使用できるレジスター一覧

(1/3)

名称	レジスター名 (サイズ)	ナンバー	用途	リセット、 復電後の状態
外部入力	X (ビット)	000~FFF	リモートI/Oに接続	クリア
	XW (ワード)	000~FF0	された入力モジュール	
	XL (ロングワード)	000~FE0	からのデータ入力用	
外部出力	Y (ビット)	000~FFF	リモートI/Oに接続	クリア
	YW (ワード)	000~FF0	された出力モジュール	
	YL (ロングワード)	000~FE0	へのデータ出力用	
内部レジスター	R, M, A (ビット)	000~FFF	演算結果中継用	クリア
	RW, MW, AW (ワード)	000~FF0		
	RL, ML, AL (ロングワード)	000~FE0		
キープリレー	K (ビット)	000~FFF	演算結果保持用	保持
	KW (ワード)	000~FF0		
	KL (ロングワード)	000~FE0		
オンディレイタイマー (接点、コイル)	T (ビット)	000~1FF	オンディレイタイマー	クリア
	TW (ワード)	000~1F0		
	TL (ロングワード)	000~1E0		
オンディレイタイマー (設定値)	TS (ワード)	000~1FF		保持
オンディレイタイマー (計数值)	TC (ワード)	000~1FF		クリア
ワンショットタイマー (接点、コイル)	U (ビット)	000~0FF	ワンショットタイマー	クリア
	UW (ワード)	000~0F0		
	UL (ロングワード)	000~0E0		
ワンショットタイマー (設定値)	US (ワード)	000~1FF		保持
ワンショットタイマー (計数值)	UC (ワード)	000~1FF		クリア

表 2-1 演算ファンクションで利用できるレジスタ一覧

(2/3)

名称	レジスタ名 (サイズ)	ナンバー	用途	リセット、 復電後の状態
アップダウンカウンタ (接点、コイル)	C (ビット)	000~0FF	条件成立後カウント用	保持
	CW (ワード)	000~0F0		
	CL (ロングワード)	000~0E0		
アップダウンカウンタ (設定値)	CS (ワード)	000~1FF		
アップダウンカウンタ (計数值)	CC (ワード)	000~1FF		
グローバルリンクレジスタ	G (ビット)	000~FFF	PLC間リンク用	クリア
	GW (ワード)	000~FF0		
	GL (ロングワード)	000~FE0		
ネステイングコイル	N (ビット)	000~0FF	サブラダープログラム 呼び出し用	クリア
	NW (ワード)	000~0F0		
	NL (ロングワード)	000~0E0		
プロセスレジスタ	P (ビット)	001~080	タスク起動用	クリア
	PW (ワード)	000~080		
	PL (ロングワード)	000~060		
イベントレジスタ	E (ビット)	000~FFF	イベント情報出力用 およびアナログ、 パルスカウンタ用	クリア
	EW (ワード)	000~FF0		
	EL (ロングワード)	000~FE0		
エッジ接点	V (ビット)	000~FFF	エッジ検出用	クリア
	VW (ワード)	000~FF0		
	VL (ロングワード)	000~FE0		
ゼットレジスタ	Z (ビット)	000~3FF	上位割り込み発生用	クリア
	ZW (ワード)	000~3F0		
	ZL (ロングワード)	000~3E0		
システムレジスタ	S (ビット)	000~BFF	システム状態表示用	初期値
	SW (ワード)	000~BF0		
	SL (ロングワード)	000~BE0		
HI-FLOWとラダー間 共有データレジスタ	J, Q (ビット)	000~FFF	HI-FLOWとラダー間 データ共有用	クリア
	JW, QW (ワード)	000~FF0		
	JL, QL (ロングワード)	000~FE0		

## 2 演算ファンクション

表 2-1 演算ファンクションで使用できるレジスタ一覧

(3/3)

名称	レジスタ名 (サイズ)	ナンバー	用途	リセット、 復電後の状態
内部拡張レジスタ	LB (ビット)	0000~FFFF	演算結果中継用	クリア
	LBW (ワード)	0000~FFF0		
	LBL (ロングワード)	0000~FFE0		
コンバーター 専用レジスタ	LR, LV (ビット)	0000~0FFF	コンバーター専用 (ユーザー使用不可)	クリア
	LRW, LVW (ワード)	0000~0FF0		
	LRL, LVL (ロングワード)	0000~0FE0		
I/O入出力用レジスタ (将来用)	IW, OW (ワード)	000~FFF	将来用	クリア
内部レジスタ	BD (ロングワード)	000~1FE	インダイレクト アクセス用	保持
	BW (ワード)	000~1FE		エリア依存
	BL (ロングワード)	000~1FE		
ファンクション データレジスタ	DW (ワード)	000~FFF	定数データエリア	保持
	DL (ロング)	000~FFE		
ファンクション ワークレジスタ	FW (ワード)	000~BFF	ワークエリア	保持
	FL (ロング)	000~BFE		
拡張ファンクション ワークレジスタ	LWW (ワード)	0000~FFFF	ワークエリア	クリア
	LWL (ロング)	0000~FFFE		
ロングワード ワークレジスタ	LLL	0000~1FFF	ワークエリア (ロングワード)	クリア
単精度浮動小数点 ワークレジスタ	LF	0000~1FFF	浮動小数点演算	クリア
バックアップ用 ワードワークレジスタ	LXW (ワード) (*)	0000~3FFF	リセット後保持が必要 なデータの格納用途	保持
	LXL (ロング)	0000~3FFE		
バックアップ用ロング ワードワークレジスタ	LML (*)	0000~1FFF	リセット後保持が必要 なデータの格納用途	保持
バックアップ用単精度浮動 小数点ワークレジスタ	LG (*)	0000~1FFF	リセット後保持が必要 なデータの格納用途	保持

(\*) バックアップ用であるLX, LM, LGレジスタは他のレジスタと比較し、アクセス時間が余計にかかります。したがって、初期値の保持用やエラー時のデータ退避用などに使用し、常時使用しないようにプログラミングしてください。

### 2.3.2 ビットレジスタの扱い

演算ファンクションでは、X000やRFF0などのビットレジスタ（「表2-1 演算ファンクションで利用できるレジスタ一覧」の中でビットと記載されているレジスタ）はワードデータとして扱います。データの内容はLSBのみ有効で、その他のビットは読み出し時はすべて0、書き込みは無効です。

以下にビットレジスタを演算ファンクションで使用する場合のデータフォーマットを示します。

	(MSB)														(LSB)	
ビットレジスタ	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0/1
	$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$

\* : 読み出し時は0、書き込みは無効です。

(例1) MOV HFFFF -> R000実行後MOV R000 -> FW000

転送命令「MOV」でビットレジスタR000にHFFFF（16進数定数）を転送後、再度「MOV」でR000の内容をワードレジスタFW000に転送すると、FW000にはH0001が格納されます。

(例2) MOV FW010 -> LB0000実行後MOV LB0000 -> DW000

FW010の内容がH1234の場合、「MOV」でワードレジスタの内容をビットレジスタLB0000に転送後、再度「MOV」でLB0000の内容をDW000に転送すると、DW000にはH0000が格納されます。

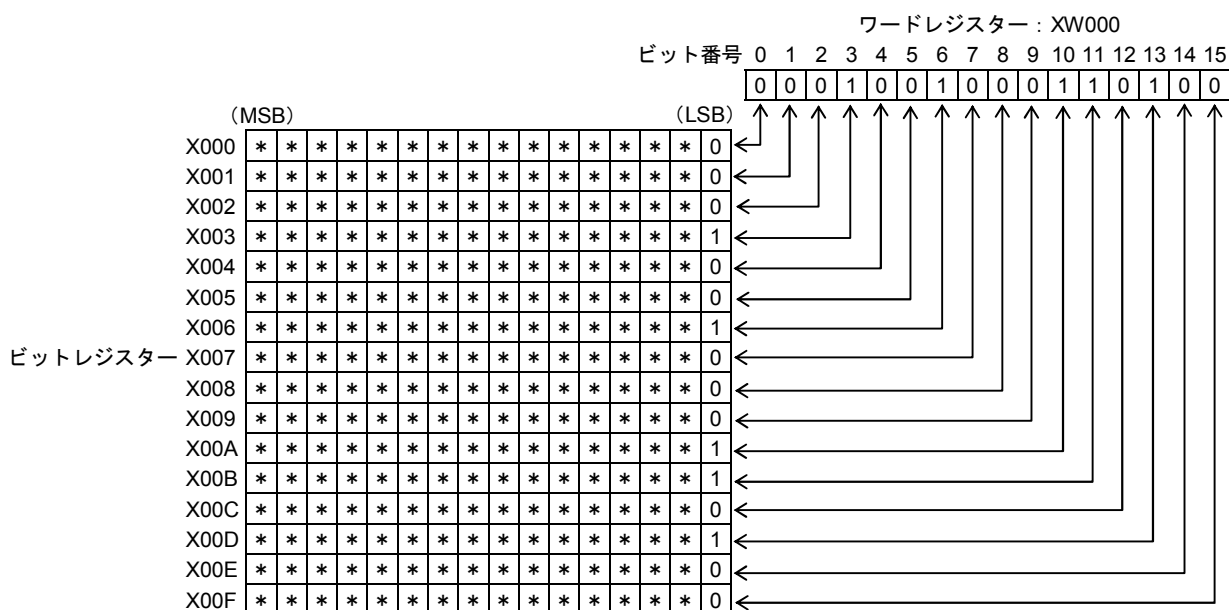


## 2 演算ファンクション

### 2.3.3 ビットレジスターとワード/ロングワードレジスターの関係

ビットレジスターとワードレジスターは下記のように対応しています。下記の例はビットレジスターX000～X00FがワードレジスターXW000に対応していることを示しています。以降、X010～X01FがXW010、X020～X02FがXW020に対応します。下記ではXレジスターを例に説明しましたが、他のビットレジスターも考え方は同様です。

ビットレジスターとワードレジスターは、実体としては同じため完全に同期されています。ワードレジスターにデータを書き込んだ直後にビットレジスターを読み出しても、直前の書き込み内容を正常に読み出すことができます。



\* : 読み出し時は0、書き込みは無効です。

ロングワードの場合も同様で、ビットレジスターX000～X01FがロングワードレジスターXL000に対応します。

## 2. 4 演算ファンクション入力

演算ファンクションは、[シンボル情報 (演算ファンクション)] 画面で入力します。

入力方法は、シンボルとパラメーター、およびパラメーター間をすべてスペース ( ) で区切ります。

パラメーターの数は、各ファンクションにより異なります。詳細は、「2. 6 命令の詳細」を参照してください。

ファンクション名称 [ ] パラメーター [ ] パラメーター [ ] パラメーター…… [Enter]

(入力例)

ADD [ ] RW000 [ ] FW000 [ ] FW000 [Enter]

(1) レジスター入力の場合

設定できるエリア	入力例	備考
I/Oエリア (ビット)	X000	演算ファンクションではワードデータとして扱います (LSBのみ有効)。
I/Oエリア (ワード)	YW000	Wはワードを示します。
I/Oエリア (ロングワード)	RL000	Lはロングワードを示します。
ファンクションワークレジスターエリア	FW025	ワークエリア
ファンクションデータレジスターエリア	DW050	定数データエリア
拡張ファンクションワークレジスターエリア	LWW0000	Wはワードを示します。
ロングワードワークレジスターエリア	LLL0000	Lはロングワードを示します。
単精度浮動小数点ワークレジスターエリア	LF0001	単精度浮動小数点演算用
バックアップ用ワークレジスターエリア (ワード、ロングワード、フローティング)	LXW0000	ワード、ロングワード、フローティングの3種類を指定できます。
T, U, C設定値エリア	TS003	Sは設定値を示します。
T, U, C計数値エリア	UC007	Cは計数値を示します。
高速I/O (ワード) エリア	IW000	将来用です。

- I/Oエリア：レジスター名がX, Y, R, M, A, K, T, U, C, G, N, P, E, V, Z, S, J, Q, LB, LVのレジスター
- ナンバーは、3桁または4桁で入力します。

## 2 演算ファンクション

### (2) 定数（イミディエート）入力の場合

#### (a) 10進数入力

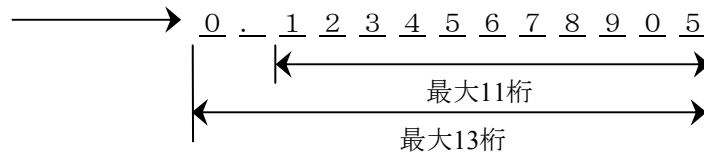
- ① 直接数値を入力（正の10進数）  $\longrightarrow$  1 2 5 3
- ② + / - を入力後、数値入力  $\longrightarrow$  + 3 2 1 0 5
- ③ 入力桁数は  $\longrightarrow$  - 1 2 5
- ・ワードの場合 最大5桁  $\longrightarrow$  (+ / -) 3 2 7 6 7
  - ・ロングワードの場合 最大10桁  $\longrightarrow$  (+ / -) 2 1 4 7 4 8 3 6 4 7

#### (b) 16進数入力

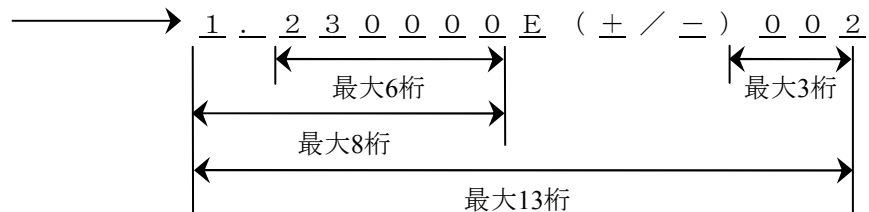
- ① 「H」を入力後、数値入力  $\longrightarrow$  H 0 5 F 3
- ② 入力桁数は  $\longrightarrow$  H 1 2 3 4 A B C D
- ・ワードの場合 最大4桁  $\longrightarrow$  H 0 5 F 3
  - ・ロングワードの場合 最大8桁  $\longrightarrow$  H 1 2 3 4 A B C D

#### (c) フローティング（単精度浮動小数点）入力

- ① 小数点形式の入力 最大13桁（小数点含）、小数点以下は最大11桁



- ② 指数形式の入力 仮数部最大8桁、仮数部小数点以下最大6桁、指数部最大3桁



(注) フローティングの定数を入力する場合、必ず小数点を入力してください。小数点を入力しない場合、フローティングデータとは認識されず入力エラーになります。

## (3) 演算ファンクション命令のインデックス指定方式

演算ファンクション命令のインデックス指定には、以下に示す2種類の方式があります。

## (a) ベースレジスター（インデックスレジスター）方式

実行レジスターアドレス = ベースレジスターの番号 + インデックスレジスターの内容（単位：ワード）

この方式は、ベースレジスターの番号 + インデックスレジスターの内容ワード分離れたエリアを実行アドレスとしてアクセスします。

インデックスレジスターに指定できるレジスターの型はワード型のみです。

（例）DW020（FW000）, R400（FW010）など

DW020（FW000）の場合、FW000の内容がH0020としたとき、DW020+H0020 → DW040を示します。

R400（FW010）の場合、FW010の内容がH0080としたとき、R400+H0080 → R480を示します。

（注1）FW000の内容がH0FF0やH1200など、DW020に加算してDWFFF（DWの最大値）を超えるとき、他のレジスターをアクセスする必要があるため動作の保証はしません。

（注2）ベースレジスターに指定するレジスターにより、ベースレジスターの番号 + インデックスレジスターの内容 = 実行レジスターのアドレスにはなりません。詳細は、「(4) 演算ファンクションインデックス指定時の注意事項」の (a) を参照してください。

ベースレジスターにナンバー先頭（000や0000）を指定する場合、ナンバーを省略できます。

（例）DW（FW000）, XW（DW000）など

DW（FW000）の場合、FW000の内容がH0020としたとき、DW020を示します。

## (b) 参照形式（インダイレクトレジスター）方式

実行レジスターアドレス = インダイレクトレジスターの内容（中身）

このインデックス指定のフォーマットは、

参照形式（ロングワード型レジスター）

という指定で使います。参照形式には、W（ワード）、L（ロングワード）、F（フローティング）が指定できます。

インダイレクトレジスターに指定できるレジスターの型はロングワードのみです。

（例）W（FL000）, L（DL000）など

W（FL000）の場合、FL000が示す内容をアドレス扱いして動作します。

例えば、FL000が示す内容を000A0000とした場合、アドレス000A0000の内容（中身）をワードデータとして扱うことを示します。

（注）インデックス指定でワード境界にロングワードアクセスした場合、演算ファンクションアドレスエラーを検出します。詳細は、「(4) 演算ファンクションインデックス指定時の注意事項」の (b) を参照してください。

## 2 演算ファンクション

### (4) 演算ファンクションインデックス指定時の注意事項

#### (a) ベースレジスタ（インデックスレジスタ）方式でのレジスタナンバー

下表のNo.2, 3, 6、および7に示すレジスタをベースレジスタに指定する場合は、ベースレジスタの番号 + インデックスレジスタの内容 = 実行レジスタのアドレスにはなりません。指定時は、下記を理解し、十分注意して使用してください。

No.	レジスタ種別	レジスタ名	実行レジスタアドレス
1	I/Oレジスタ (ビット)	X, Y, R, M, A, K, T, U, C, G, N, P, E, V, Z, S, J, Q, LB, LR, LV	ベースレジスタの番号 + インデックスレジスタの内容 (16進数)
2	I/Oレジスタ (ワード)	XW, YW, RW, MW, AW, KW, TW, UW, CW, GW, NW, PW, EW, VW, ZW, SW, JW, QW, LBW, LRW, LVW	ベースレジスタの番号 + インデックスレジスタの内容 (16進数) × H0010 (16進数)
3	I/Oレジスタ (ロングワード)	XL, YL, RL, ML, AL, KL, TL, UL, CL, GL, NL, PL, EL, VL, ZL, SL, JL, QL, LBL, LRL, LVL	
4	ワークレジスタ (ワード)	DW, FW, LWW, LXW	
5	ワークレジスタ (ロングワード)	DL, FL, LWL, LXL	ベースレジスタの番号 + インデックスレジスタの内容 (16進数) ÷ H0002 (16進数)
6	ロングワード専用 レジスタ	BD, LLL, LML	
7	浮動小数点専用 レジスタ	LF, LG	

#### (例) ・G000 (DW001)

DW001がH0010の場合、実行レジスタアドレスはG010になります。

#### ・RW020 (FW000)

FW000がH0030の場合、実行レジスタアドレスはRW320になります。

#### ・LLL0000 (FW000)

FW000がH0040の場合、実行レジスタアドレスはLLL0020になります。

(b) インデックス指定時のロングワード／フローティングアクセス

LPUモジュールレビジョンL (Ver.-Rev. : 02-05) 以前のS10Vでは、演算ファンクションでのロングワード／フローティングアクセスは、ロングワード境界のみ行えます。インデックス指定でワード境界をロングワード／フローティングアクセスした場合、LPUモジュールは「演算ファンクションアドレスエラー」を検出し、LPUモジュールがダウン (ERR LED点灯、ラダー停止、リモートI/O停止) し、ラダープログラムの誤りを報告します。

通常、ラダー図システムでは回路入力時にチェックし、ワード境界からロングワード／フローティングアクセスを検出した場合「ロングレジスタの番号がワード境界をまたがっています」と表示して入力を無効にします (ラダー図システムVer.-Rev. : 01-16以降では、「奇数番号のロングワードレジスタが指定されています」と表示)。しかし、インデックス指定時は実行するまでインデックスレジスタの内容が確定しないため、ラダー図システムではチェックできません。したがって、インデックス指定時は、実行レジスタアドレスがワード境界からロングワードアクセスしないよう十分注意して使用してください。演算ファンクションアドレスエラーは、ラダープログラミング時に注意することで回避できます。プログラミングでの回避策を以下に示します。

インデックス方式	回避策
ベースレジスタ (インデックスレジスタ) 方式	ベースレジスタにロングワード型を指定した場合、インデックスレジスタの内容を必ず偶数にします。
参照形式 (インダイレクトレジスタ) 方式	参照形式にLまたはFを指定した場合、インダイレクトレジスタの内容を、必ず下記のエラーが発生するナンバー以外にします。

ロングワードアクセス時に演算ファンクションアドレスエラーが発生するナンバーを以下に示します (LPUモジュールレビジョンM (Ver.-Rev. : 02-06) 以降は下記の指定をサポートしています)。

レジスタ種別	レジスタ名	エラーが発生するナンバー
I/Oレジスタ (ロングワード)	XL, YL, RL, ML, AL, KL, TL, UL, CL, GL, NL, PL, EL, VL, ZL, SL, JL, QL, LBL, LRL, LVL	XL0 <sup>3</sup> 0, LBL00 <sup>5</sup> 0のように、 <sup>□</sup> の部分が奇数の場合にエラー発生
ワークレジスタ (ロングワード)	DL, FL, LWL, LXL	DL10 <sup>7</sup> , LWLFFF <sup>9</sup> のように、 <sup>□</sup> の部分が奇数の場合にエラー発生

通 知

Ver.-Rev. : 01-16以降のラダー図システムで作成した、奇数番号のロングワードレジスタまたはPSHO/POPOを含むラダープログラムを、Ver.-Rev. : 01-15以前のラダー図システムまたは一括ロードにてモジュールレビジョンL (Ver.-Rev. : 02-05) 以前のLPUに送信すると、LPUが“不当命令検出”にて停止します。

## 2 演算ファンクション

(例1) ベースレジスタ (インデックスレジスタ) 方式

- XL000 (FW000)

FW000がH0003の場合、実行レジスタアドレス = XL030になりエラーが発生します。

FW000が偶数の場合、エラーは発生しません。

- LWL1000 (FW000)

FW000がH0005の場合、実行レジスタアドレス = LWL1005になりエラーが発生します。

FW000が偶数の場合、エラーは発生しません。

- LLL2000 (FW000)

FW000がH00FFの場合、実行レジスタアドレス = ワード境界になりエラーが発生しま

す。FW000が偶数の場合、エラーは発生しません。

- LF0000 (FW000)

FW000がH007Fの場合、実行レジスタアドレス = ワード境界になりエラーが発生しま

す。FW000が偶数の場合、エラーは発生しません。

(例2) 参照形式 (インダイレクトレジスタ) 方式

- L (RL000)

RL000の内容がYW010の場合、エラーが発生します。YW000やYW020ならばエラーは発生しません。

ASTでRL000にYWのアドレスセット時、YW\*△\*の△が偶数になるように指定します。

AST YW010 -> RL000 : L (RL000) アクセス時にエラーが発生します。

AST YW020 -> RL000 : L (RL000) アクセスは正常に行われます。

- L (DL004)

DL004の内容がLWW0001の場合、エラーが発生します。LWW0002やLWWFFFEならばエラーは発生しません。

ASTでDL004にLWWのアドレスセット時、LWW\*\*\*△の△が偶数になるように指定します。

AST LWW0001 -> DL004 : L (DL004) アクセス時にエラーが発生します。

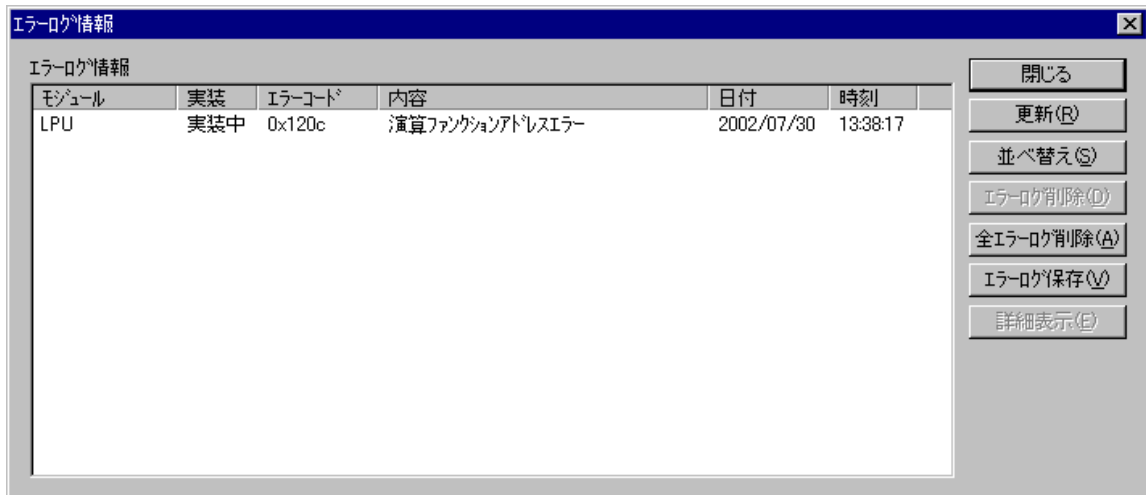
AST LWW0002 -> DL004 : L (DL004) アクセスは正常に行われます。

## 【補足】 S10V 基本システムでの演算ファンクションアドレスエラー発生確認方法

LPUモジュールのERR LED点灯時、発生したエラーが演算ファンクションアドレスエラーか、それ以外かはS10V 基本システムを使用して切り分けます。

S10V 基本システムでの切り分け手順を以下に示します。

- ① S10V 基本システムを起動します。
- ② エラーログ ボタンをクリックします。
- ③ [エラーログ情報] 画面が表示されるので、モジュール「LPU」の行にエラーコード「0x120c」、内容「演算ファンクションアドレスエラー」が表示されているか確認します。  
演算ファンクションアドレスエラーが発生した状態での [エラーログ情報] 画面の例を以下に示します。





## 2 演算ファンクション

### 2.5 演算ファンクション一覧表

(1/5)

大分類	分類	シンボル	処理単位	処理内容	フラグ						ページ
					X	E	P	N	Z	V	
算術演算命令	加算	ADD	ワード	$(S) + (D) \rightarrow (R)$	-	-	-	-	-	●	2-26
			ロング		-	-	-	-	-	●	
			フローティング		-	●	-	-	-	-	
	減算	SUB	ワード	$(S) - (D) \rightarrow (R)$	-	-	-	-	-	●	2-30
			ロング		-	-	-	-	-	●	
			フローティング		-	●	-	-	-	-	
	+ 1	INC	ワード	$(S) + 1 \rightarrow (S)$	-	-	-	-	-	●	2-34
			ロング		-	-	-	-	-	●	
	- 1	DEC	ワード	$(S) - 1 \rightarrow (S)$	-	-	-	-	-	●	2-36
			ロング		-	-	-	-	-	●	
乗算	MUL	ワード	$(S) \times (D) \rightarrow (R)$	-	-	-	-	-	●	2-38	
		ロング		-	-	-	-	-	●		
		フローティング		-	●	-	-	-	-		
除算	DIV	ワード	$(S) \div (D) \rightarrow (R)$ (商)	-	●	-	-	-	●	2-42	
		ロング		-	●	-	-	-	●		
		フローティング		-	●	-	-	-	-		
剰余	MOD	ワード	$(S) \div (D) \rightarrow (R)$ (剰余)	-	●	-	-	-	●	2-46	
		ロング		-	●	-	-	-	●		
スケール変換	SCL	ワード	$(S) \times (D1) \div (D2) \rightarrow (R)$	-	●	-	-	-	●	2-48	
論理演算命令	論理積	AND	ワード	$(S) \wedge (D) \rightarrow (R)$	-	-	-	-	-	-	2-50
			ロング		-	-	-	-	-	-	
	論理和	OR	ワード	$(S) \vee (D) \rightarrow (R)$	-	-	-	-	-	-	2-52
			ロング		-	-	-	-	-	-	
排他的論理和	EOR	ワード	$(S) \oplus (D) \rightarrow (R)$	-	-	-	-	-	-	2-54	
		ロング		-	-	-	-	-	-		
否定	NOT	ワード	$\overline{(S)} \rightarrow (R)$	-	-	-	-	-	-	2-56	
比較命令	=	EQU	ワード	$(S) = (D)$ のとき $1 \rightarrow (R)$	-	-	-	-	-	-	2-58
			ロング	$(S) \neq (D)$ のとき $0 \rightarrow (R)$	-	-	-	-	-	-	
			フローティング	$(S) \neq (D)$ のとき $1 \rightarrow (R)$	-	-	-	-	-	-	
	≠	NEQ	ワード	$(S) \neq (D)$ のとき $1 \rightarrow (R)$	-	-	-	-	-	-	2-60
			ロング	$(S) = (D)$ のとき $0 \rightarrow (R)$	-	-	-	-	-	-	
			フローティング	$(S) = (D)$ のとき $0 \rightarrow (R)$	-	-	-	-	-	-	
	>	GT	ワード	$(S) > (D)$ のとき $1 \rightarrow (R)$	-	-	-	-	-	-	2-62
			ロング	$(S) \leq (D)$ のとき $0 \rightarrow (R)$	-	-	-	-	-	-	
フローティング			$(S) \leq (D)$ のとき $0 \rightarrow (R)$	-	-	-	-	-	-		
<	LT	ワード	$(S) < (D)$ のとき $1 \rightarrow (R)$	-	-	-	-	-	-	2-64	
		ロング	$(S) \geq (D)$ のとき $0 \rightarrow (R)$	-	-	-	-	-	-		
		フローティング	$(S) \geq (D)$ のとき $0 \rightarrow (R)$	-	-	-	-	-	-		

●：演算結果により変化

-：演算実行直前の値を保持

S：ソース

D：デスティネーション

R：リザルト

(2/5)

大分類	分類	シンボル	処理単位	処理内容	フラグ						ページ
					X	E	P	N	Z	V	
比較命令	≥	GE	ワード	(S) ≥ (D) のとき 1 → (R)	-	-	-	-	-	-	2-66
			ロング	(S) < (D) のとき 0 → (R)							
			フローティング								
	≤	LE	ワード	(S) ≤ (D) のとき 1 → (R)	-	-	-	-	-	-	2-68
			ロング	(S) > (D) のとき 0 → (R)							
			フローティング								
テスト	TST	ワード ロング フローティング	(S) をテストし、P, N, Zフラグをセットします。	-	-	●	●	●	-	2-70	
転送命令	転送	MOV	ワード ロング フローティング	(S) → (D)	-	-	-	-	-	-	2-72
	一括転送	MOM	ワード ロング	(S, n) → (D)	-	-	-	-	-	-	2-74
	同一データ一括転送	INI	ワード ロング	(S) → (D, n)	-	-	-	-	-	-	2-76
	交換	EXC	ワード ロング	(S) ↔ (D)	-	-	-	-	-	-	2-78
	FIFO書き込み	PSH	ワード	(S) → FIFOテーブル	-	-	-	-	-	-	2-80
	FIFO読み出し	POP	ワード	FIFOテーブル → (D)	-	-	-	-	-	-	2-82
	FIFO書き込み	PSHO	ワード	(S) → FIFOテーブル	-	-	-	-	-	-	2-84
	FIFO読み出し	POPO	ワード	FIFOテーブル → (D)	-	-	-	-	-	-	2-86
	アドレスセット	AST	ロング	Sのアドレス値 → (D)							2-88
	サーチ	SCH	ワード ロング フローティング	(S) をDからm (サーチステップ数) サーチし、一致No. → (R)	-	-	-	-	-	-	2-90
変換命令	BIN→FLOAT	BTF	ワード ロング	BIN → FLOAT (S) → (R)	-	-	-	-	-	-	2-94
	FLOAT→BIN	FTB	ワード ロング	FLOAT → BIN (S) → (R)	-	-	-	-	-	-	2-96
	BIN→BCD	BTD	ワード ロング	BIN → BCD (S) → (R)	-	●	-	-	-	●	2-98
	BCD→BIN	DTB	ワード ロング	BCD → BIN (S) → (R)	-	●	-	-	-	-	2-100
	BIN→7SEG	SEG	ワード ロング	BIN → 7SEG (S) → (R)	-	-	-	-	-	-	2-102

- : 演算結果により変化
- : 演算実行直前の値を保持
- S : ソース
- D : デスティネーション
- R : リザルト
- n : ワード数
- m : サーチステップ数

## 2 演算ファンクション

(3/5)

大分類	分類	シンボル	処理単位	処理内容	フラグ						ページ
					X	E	P	N	Z	V	
変換命令	BIN→ASCII	ASP	ワード	BIN → ASCII (パックモード) (S) → (R)	-	-	-	-	-	-	2-104
		ASU	ワード	BIN → ASCII (アンパックモード) (S) → (R)	-	-	-	-	-	-	2-106
	ASCII→BIN	APB	ワード	ASCII → BIN (パックモード) (S) → (R)	-	●	-	-	-	-	2-108
		AUB	ワード	ASCII → BIN (アンパックモード) (S) → (R)	-	●	-	-	-	-	2-110
	SINGLE ↓ DOUBLE	STD	ワード	(S) W → (R) L	-	-	-	-	-	-	2-112
	DOUBLE ↓ SINGLE	DTS	ロング	(S) L → (R) W	-	-	-	-	-	●	2-114
	絶対値	ABS	ワード	(S)   → (R)	-	-	-	-	-	●	2-116
			ロング								
			フローティング								
	+/-	NEG	ワード	- (S) → (R)	-	-	-	-	-	●	2-118
ロング											
フローティング											
デコード	DCD	ワード	(S) の数値n ... 1 → (R) のnビット	-	-	-	-	-	-	2-120	
		ロング									
エンコード	ECD	ワード	(S) の1が立っているビットn ... n → (R)	-	●	-	-	-	-	2-122	
		ロング									
シフト命令	論理右	ワード	(S) 右 (D) ビットシフト → 0, (R)	-	-	-	-	-	-	2-124	
		ロング									
	論理左	ワード	(S) 左 (D) ビットシフト → (R) , 0	-	-	-	-	-	-	2-126	
		ロング									
算術右	ASR	ワード	(S) 右 (D) ビットシフト → MSB, (R)	-	-	-	-	-	-	2-128	
		ロング									
算術左	ASL	ワード	(S) 左 (D) ビットシフト → (R) , V	-	-	-	-	-	●	2-130	
		ロング									
ローテイト命令	右回転	ワード	(S) 右 (D) ビット回転 → (R)	-	-	-	-	-	-	2-132	
		ロング									
左回転	ROL	ワード	(S) 左 (D) ビット回転 → (R)	-	-	-	-	-	-	2-134	
		ロング									
関数処理命令	LIMITER	LIM	ワード	(D1) < (S) .. (D1) → (R)	-	●	-	-	-	-	2-136
			ロング	(D2) ≤ (S) ≤ D1 .. (S) → (R)							
			フローティング	(S) < (D2) .. (D2) → (R)							
	DEAD BAND	BND	ワード	(D1) < (S) .. (S) - (D1) → (R)	-	●	-	-	-	●	2-140
			ロング	(D2) ≤ (S) ≤ (D1) .. 0 → (R)	-	●	-	-	-	●	
			フローティング	(S) < (D2) .. (S) - (D2) → (R)	-	●	-	-	-	-	

● : 演算結果により変化

- : 演算実行直前の値を保持

S : ソース

D : デスティネーション

R : リザルト

(4/5)

大分類	分類	シンボル	処理単位	処理内容	フラグ						ページ
					X	E	P	N	Z	V	
関数処理命令	DEAD ZONE	ZON	ワート <sup>o</sup>	$(S) > 0 \dots (S) + (D1) \rightarrow (R)$	-	●	-	-	-	●	2-144
			ロンク <sup>o</sup>	$(S) = 0 \dots 0 \rightarrow (R)$	-	●	-	-	-	●	
			フローティング <sup>o</sup>	$(S) < 0 \dots (S) + (D2) \rightarrow (R)$	-	●	-	-	-	-	
	平方根	SQR	ワート <sup>o</sup>	$(S) \geq 0 \dots \text{SQR}(S) \rightarrow (R)$	-	-	-	-	-	-	2-148
			ロンク <sup>o</sup>	$(S) < 0 \dots 0 \rightarrow (R)$	-	-	-	-	-	-	
			フローティング <sup>o</sup>		-	-	-	-	-	-	
	サイン	SIN	フローティング <sup>o</sup>	$\text{SIN}(S) \rightarrow (R)$	-	-	-	-	-	-	2-152
	コサイン	COS	フローティング <sup>o</sup>	$\text{COS}(S) \rightarrow (R)$	-	-	-	-	-	-	2-154
	タンジェント	TAN	フローティング <sup>o</sup>	$\text{TAN}(S) \rightarrow (R)$	-	●	-	-	-	-	2-156
	アークサイン	ASIN	フローティング <sup>o</sup>	$\text{SIN}^{-1}(S) \rightarrow (R)$	-	●	-	-	-	-	2-158
	アークコサイン	ACOS	フローティング <sup>o</sup>	$\text{COS}^{-1}(S) \rightarrow (R)$	-	●	-	-	-	-	2-160
	アークタンジェント	ATAN	フローティング <sup>o</sup>	$\text{TAN}^{-1}(S) \rightarrow (R)$	-	-	-	-	-	-	2-162
	指数演算	EXP	フローティング <sup>o</sup>	$\text{EXP}(S) \rightarrow (R)$	-	●	-	-	-	-	2-164
	自然対数	LOG	フローティング <sup>o</sup>	$\text{LOG}(S) \rightarrow (R)$	-	●	-	-	-	-	2-166
最大値	MAX	ワート <sup>o</sup>	$(S) \geq (R) \dots (S) \rightarrow (R)$	-	-	-	-	-	-	2-168	
		ロンク <sup>o</sup>	$(S) < (R) \dots (D) \rightarrow (R)$	-	-	-	-	-	-		
		フローティング <sup>o</sup>		-	-	-	-	-	-		
最小値	MIN	ワート <sup>o</sup>	$(S) \leq (R) \dots (S) \rightarrow (R)$	-	-	-	-	-	-	2-170	
		ロンク <sup>o</sup>	$(S) > (R) \dots (D) \rightarrow (R)$	-	-	-	-	-	-		
		フローティング <sup>o</sup>		-	-	-	-	-	-		
特殊命令	クリア	XCLR	-	Xのエリアをクリアします。	-	-	-	-	-	-	2-172
		YCLR	-	Yのエリアをクリアします。	-	-	-	-	-	-	2-172
		GCLR	-	Gのエリアをクリアします。	-	-	-	-	-	-	2-172
		RCLR	-	Rのエリアをクリアします。	-	-	-	-	-	-	2-172
		KCLR	-	Kのエリアをクリアします。	-	-	-	-	-	-	2-172
		TCLR	-	Tのエリアおよび計数値をクリアします。	-	-	-	-	-	-	2-172
		UCLR	-	Uのエリアおよび計数値をクリアします。	-	-	-	-	-	-	2-172
		CCLR	-	Cのエリアおよび計数値をクリアします。	-	-	-	-	-	-	2-172
		VCLR	-	Vのエリアをクリアします。	-	-	-	-	-	-	2-172
		ECLR	-	Eのエリアをクリアします。	-	-	-	-	-	-	2-172
		FCLR	-	演算結果フラグをクリアします。	-	-	-	-	-	-	2-172

- ：演算結果により変化
- ：演算実行直前の値を保持
- S：ソース
- D：デスティネーション
- R：リザルト

## 2 演算ファンクション

(5/5)

大分類	分類	シンボル	処理単位	処理内容	フラグ						ページ
					X	E	P	N	Z	V	
ジャンプ命令	条件成立	JT	—	条件成立で指定したラベルへジャンプ	—	—	—	—	—	—	2-174
	無条件	JMP	—	無条件で指定したラベルへジャンプ	—	—	—	—	—	—	2-176
	条件成立 SEND	JSE	—	条件成立でSEND命令（シーケンス終了）へジャンプ	—	—	—	—	—	—	2-178
イーサネット通信命令	TCP通信	TOP	—	TCPコネクションのオープン	—	—	—	—	—	—	2-198
		TPOP	—	TCPコネクションのオープン	—	—	—	—	—	—	2-200
		TCLO	—	TCPコネクションのクローズ	—	—	—	—	—	—	2-202
		TRCV	—	TCP受信	—	—	—	—	—	—	2-204
		TSND	—	TCP送信	—	—	—	—	—	—	2-208
	UDP通信	UOP	—	UDPのオープン	—	—	—	—	—	—	2-210
		UCLO	—	UDPのクローズ	—	—	—	—	—	—	2-212
		URCV	—	UDP受信	—	—	—	—	—	—	2-214
		USND	—	UDP送信	—	—	—	—	—	—	2-218

—：演算実行直前の値を保持

## 2.6 命令の詳細

演算ファンクション命令の説明は、以下に示す形式になっています。

### (1) 入力形式

命令の入力形式を示します。

### (2) 機能

命令が処理する機能について説明しています。

### (3) データタイプ

パラメーターに指定できるデータタイプに○を付けています。

(例)

DW000などのレジスター  
やH0001などの定数が使用  
できるかを表しています。

LLL0000などのレジスターや  
H04231556などの定数が使用  
できるかを表しています。

LF0000などのレジスターや  
1.12E-002などの定数が使用  
できるかを表しています。

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○：指定できます。

—：指定できません。

レジスターが指定できる場  
合、インデックス指定ができ  
るかを表しています。

この例では、S（ソース）およびD（デスティネーション）にはワード、ロングワード、フローティングデータのアドレス（インデックス指定含む）、および定数を指定できます。R（リザルト）にはワード、ロングワード、フローティングデータのレジスター指定（インデックス指定含む）ができます。

(注) R000, Y1FFなどビットのI/Oエリアは、演算ファンクションではワードデータとして扱います。

この場合、LSBのみ有効で、他のビットは読み出し時は常に0、書き込みは無効です。詳細は「2.3.2 ビットレジスターの扱い」を参照してください。

### (4) プログラム例

簡単なラダープログラム例と処理の内容を示しています。

### (5) エラー処理

エラー発生時の条件と演算結果フラグに反映されるフラグを示しています。

# ADD 加算 : ADD

## (1) 入力形式

```
ADD S + D -> R
```

S : ソース格納レジスタまたは定数

D : デスティネーション格納レジスタまたは定数

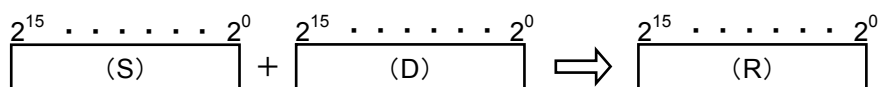
R : リザルト (演算結果) を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「+」および「->」は省略できます。

## (2) 機能

### ● ワードデータの加算

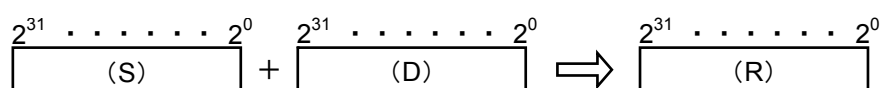
ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータを加算し、結果をリザルト (R) に格納します。



ソース (S)、デスティネーション (D)、リザルト (R) に指定できる値および格納される値の範囲は、-32768~32767です。

### ● ロングワードデータの加算

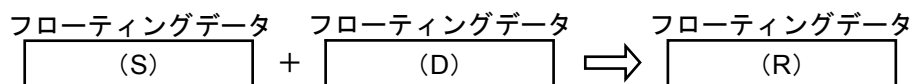
ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータを加算し、結果をリザルト (R) に格納します。



ソース (S)、デスティネーション (D)、リザルト (R) に指定できる値および格納される値の範囲は、-2147483648~2147483647です。

● フローティングデータの加算

ソース (S) で指定されたフローティングデータとデスティネーション (D) で指定されたフローティングデータを加算し、結果をリザルト (R) に格納します。



ソース (S)、デスティネーション (D)、リザルト (R) に指定できる値および格納される値の範囲は下記です。

$$0, \pm 2^{-126} \sim \pm 2^{128}$$

(3) データタイプ

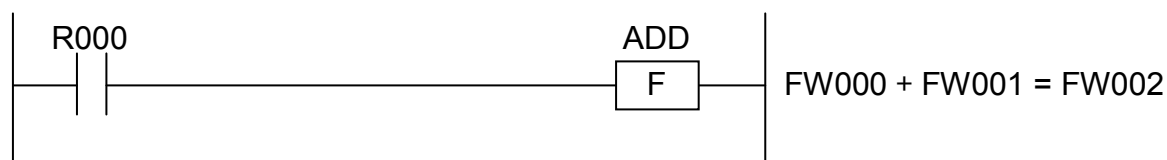
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○ : 指定できます。

— : 指定できません。

S, D、およびRの型 (ワード/ロングワード/フローティング) はすべて合わせてください。異なった型は指定できません (入力エラーになります)。

(4) プログラム例



入力条件R000がONのとき、FW000の内容とFW001の内容を加算し、FW002に格納します。



## ADD 加算 : ADD

### (5) エラー処理

#### ● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	↕

V : ワードデータ時

- ・リザルト (R) が-32768~32767のとき0、それ以外のとき1

ロングワードデータ時

- ・リザルト (R) が-2147483648~2147483647のとき0、それ以外のとき1

フローティングデータ時

- ・演算結果によりVは変化しません (保持します)。

E : ワードおよびロングワードデータ時

- ・演算結果によりEは変化しません (保持します)。

フローティングデータ時

- ・リザルト (R) が0ではなく、かつ下記の範囲外のとき1、それ以外のとき0  
 $\pm 2^{-126} \sim \pm 2^{128}$

V, E以外のフラグはすべて保持します。

#### ● オーバーフロー発生時、リザルト (R) には下記のフルスケール値が格納されます。

	正のオーバーフロー時	負のオーバーフロー時
ワード	H7FFF	H8000
ロングワード	H7FFFFFFF	H80000000
フローティング	+3.402823E38	-3.402823E38

フローティングでオーバーフロー発生時、VフラグはONしません (ワードおよびロングワード時はONします)。

#### ● EフラグON時は、無処理 (リザルト (R) は変化なし) です。

#### ● フローティングでアンダーフロー発生時、リザルト (R) には正しい符号を持った0が格納されま す。アンダーフロー発生時、演算結果フラグは変化しません。

このページは白紙です。

# SUB 減算 : SUB

## (1) 入力形式

```
SUB S - D -> R
```

S : ソース格納レジスタまたは定数

D : デスティネーション格納レジスタまたは定数

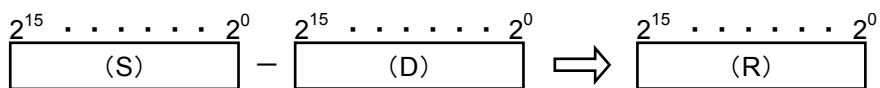
R : リザルト (演算結果) を格納するレジスタまたは定数

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「-」および「->」は省略できます。

## (2) 機能

### ● ワードデータの減算

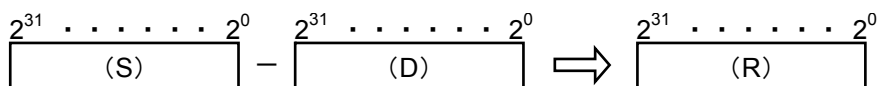
ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータを減算し、結果をリザルト (R) に格納します。



ソース (S)、デスティネーション (D)、リザルト (R) に指定できる値および格納される値の範囲は、-32768~32767です。

### ● ロングワードデータの減算

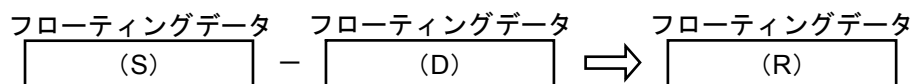
ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータを減算し、結果をリザルト (R) に格納します。



ソース (S)、デスティネーション (D)、リザルト (R) に指定できる値および格納される値の範囲は、-2147483648~2147483647です。

● フローティングデータの減算

ソース (S) で指定されたフローティングデータとデスティネーション (D) で指定されたフローティングデータを減算し、結果をリザルト (R) に格納します。



ソース (S)、デスティネーション (D)、リザルト (R) に指定できる値および格納される値の範囲は下記です。

$$0, \pm 2^{-126} \sim \pm 2^{128}$$

(3) データタイプ

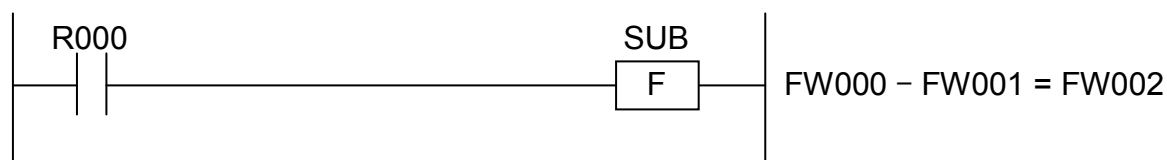
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○：指定できます。

—：指定できません。

S, D、およびRの型（ワード/ロングワード/フローティング）はすべて合わせてください。異なった型は指定できません（入力エラーになります）。

(4) プログラム例



入力条件R000がONのとき、FW000の内容とFW001の内容を減算し、FW002に格納します。

## SUB 減算 : SUB

### (5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
-	↕	-	-	-	↕

V : ワードデータ時

- ・リザルト (R) が-32768~32767のとき0、それ以外のとき1

ロングワードデータ時

- ・リザルト (R) が-2147483648~2147483647のとき0、それ以外のとき1

フローティングデータ時

- ・演算結果によりVは変化しません (保持します)。

E : ワードおよびロングワードデータ時

- ・演算結果によりEは変化しません (保持します)。

フローティングデータ時

- ・リザルト (R) が0ではなく、かつ下記の範囲外のとき1、それ以外のとき0  
 $\pm 2^{-126} \sim \pm 2^{128}$

V, E以外のフラグはすべて保持します。

- オーバーフロー発生時、リザルト (R) には下記のフルスケール値が格納されます。

	正のオーバーフロー時	負のオーバーフロー時
ワード	H7FFF	H8000
ロングワード	H7FFFFFFF	H80000000
フローティング	+3.402823E38	-3.402823E38

フローティングでオーバーフロー発生時、VフラグはONしません (ワードおよびロングワード時はONします)。

- EフラグON時は、無処理 (リザルト (R) は変化なし) です。

- フローティングでアンダーフロー発生時、リザルト (R) には正しい符号を持った0が格納されま  
 ず。アンダーフロー発生時、演算結果フラグは変化しません。

このページは白紙です。

# INC +1 : INCREMENT

## (1) 入力形式

```
INC S
```

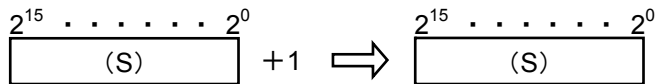
S : INC (+1) を行うデータの格納レジスター

(\*) ファンクション名称とパラメーターの間には、必ずスペースを入力してください。

## (2) 機能

- ワードデータのインクリメント

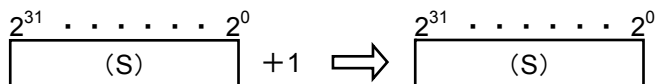
ソース (S) で指定された16ビットデータの内容を+1します。



ソース (S) に指定できる値および格納される値の範囲は、-32768~32767です。

- ロングワードデータのインクリメント

ソース (S) で指定された32ビットデータの内容を+1します。



ソース (S) に指定できる値および格納される値の範囲は、-2147483648~2147483647です。

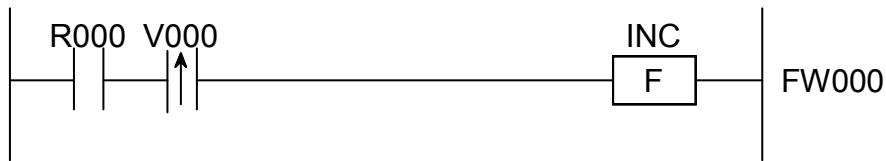
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	—	○	—	—	—	○

○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がOFF→ONに変化時、1回のみFW000の内容を+1します。

(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
-	-	-	-	-	↑↓

V : ワードデータ時

- ・リザルト (R) が-32768~32767のとき0、それ以外るとき1

ロングワードデータ時

- ・リザルト (R) が-2147483648~2147483647のとき0、それ以外るとき1

V以外のフラグはすべて保持します。

- オーバーフロー発生時、リザルト (R) には下記のフルスケール値が格納されます。

ワード	ロングワード
H7FFF	H7FFFFFFF



# DEC -1 : DECREMENT

## (1) 入力形式

```
DEC S
```

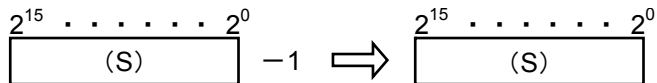
S : DEC (-1) を行うデータの格納レジスター

(\*) ファンクション名称とパラメーターの間には、必ずスペースを入力してください。

## (2) 機能

### ● ワードデータのデクリメント

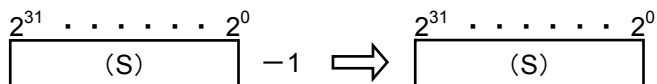
ソース (S) で指定された16ビットデータの内容を-1します。



ソース (S) に指定できる値および格納される値の範囲は、-32768~32767です。

### ● ロングワードのデクリメント

ソース (S) で指定された32ビットデータの内容を-1します。



ソース (S) に指定できる値および格納される値の範囲は、-2147483648~2147483647です。

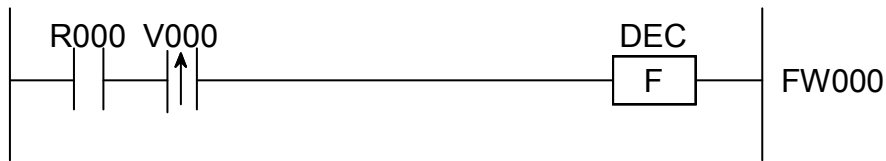
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	—	○	—	—	—	○

○ : 指定できます。

— : 指定できません。

## (4) プログラム例



入力条件R000がOFF→ONに変化時、1回のみFW000の内容を-1します。

## (5) エラー処理

## ● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	↕

V : ワードデータ時

- ・リザルト (R) が-32768～32767のとき0、それ以外るとき1

ロングワードデータ時

- ・リザルト (R) が-2147483648～2147483647のとき0、それ以外るとき1

V以外のフラグはすべて保持します。

## ● オーバーフロー発生時、リザルト (R) には下記のフルスケール値が格納されます。

ワード	ロングワード
H8000	H80000000

# MUL 乗算 : MULTIPLY

## (1) 入力形式

```
MUL S * D -> R
```

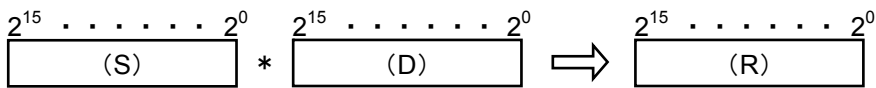
- S : ソース格納レジスタまたは定数
- D : デスティネーション格納レジスタまたは定数
- R : リザルト (演算結果) を格納するレジスタ

(\*) シンボル (命令語名称) とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「\*」および「->」は省略できます。

## (2) 機能

### ● ワードデータの乗算

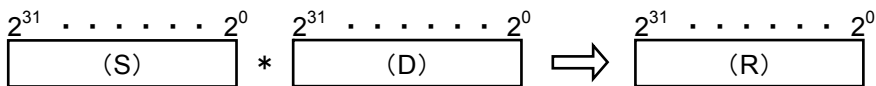
ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータを乗算し、結果をリザルト (R) に格納します。



ソース (S)、デスティネーション (D)、リザルト (R) に指定できる値および格納される値の範囲は、-32768~32767です。

### ● ロングワードデータの乗算

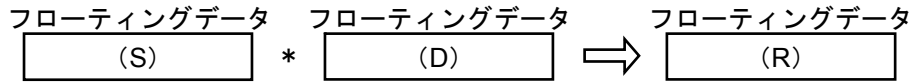
ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータを乗算し、結果をリザルト (R) に格納します。



ソース (S)、デスティネーション (D)、リザルト (R) に指定できる値および格納される値の範囲は、-2147483648~2147483647です。

● フローティングデータの乗算

ソース (S) で指定されたフローティングデータとデスティネーション (D) で指定されたフローティングデータを乗算し、結果をリザルト (R) に格納します。



ソース (S)、デスティネーション (D)、リザルト (R) に指定できる値および格納される値の範囲は下記です。

$$0, \pm 2^{-126} \sim \pm 2^{128}$$

(3) データタイプ

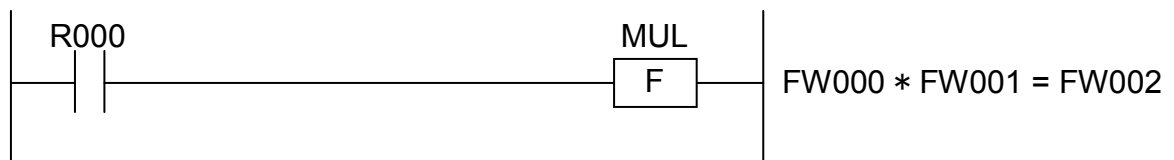
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○ : 指定できます。

— : 指定できません。

S, D、およびRの型 (ワード/ロングワード/フローティング) はすべて合わせてください。異なった型は指定できません (入力エラーになります)。

(4) プログラム例



入力条件R000がONのとき、FW000の内容とFW001の内容を乗算し、FW002へ格納します。

## MUL 乗算 : MULTIPLY

### (5) エラー処理

#### ● 演算結果フラグ

X	E	P	N	Z	V
-	↕	-	-	-	↕

V : ワードデータ時

- ・リザルト (R) が-32768~32767のとき0、それ以外のとき1

ロングワードデータ時

- ・リザルト (R) が-2147483648~2147483647のとき0、それ以外のとき1

フローティングデータ時

- ・演算結果によりVは変化しません (保持します)。

E : ワードおよびロングワードデータ時

- ・演算結果によりEは変化しません (保持します)。

フローティングデータ時

- ・リザルト (R) が0ではなく、かつ下記の範囲外のとき1、それ以外のとき0  
 $\pm 2^{-126} \sim \pm 2^{128}$

V, E以外のフラグはすべて保持します。

#### ● オーバーフロー発生時、リザルト (R) には下記のフルスケール値が格納されます。

	正のオーバーフロー時	負のオーバーフロー時
ワード	H7FFF	H8000
ロングワード	H7FFFFFFF	H80000000
フローティング	+3.402823E38	-3.402823E38

フローティングでオーバーフロー発生時、VフラグはONしません (ワードおよびロングワード時はONします)。

#### ● EフラグON時は、無処理 (リザルト (R) は変化なし) です。

#### ● フローティングでアンダーフロー発生時、リザルト (R) には正しい符号を持った0が格納されま す。アンダーフロー発生時、演算結果フラグは変化しません。

このページは白紙です。

# DIV 除算 : DIVIDE

## (1) 入力形式

```
DIV S / D -> R
```

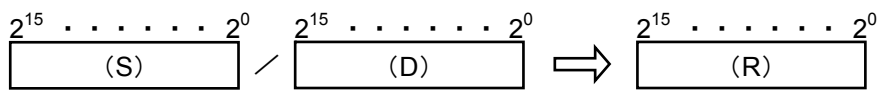
- S : ソース格納レジスタまたは定数
- D : デスティネーション格納レジスタまたは定数
- R : リザルト (演算結果) を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「/」および「->」は省略できます。

## (2) 機能

### ● ワードデータの除算

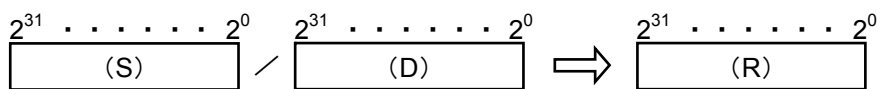
ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータを除算し、結果 (商) をリザルト (R) に格納します。



ソース (S)、デスティネーション (D)、リザルト (R) に指定できる値および格納される値の範囲は、-32768~32767です。

### ● ロングワードデータの除算

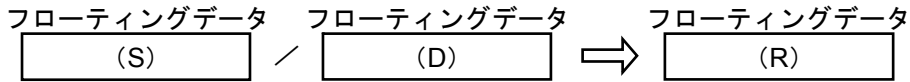
ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータを除算し、結果 (商) をリザルト (R) に格納します。



ソース (S)、デスティネーション (D)、リザルト (R) に指定できる値および格納される値の範囲は、-2147483648~2147483647です。

● フローティングデータの除算

ソース (S) で指定されたフローティングデータとデスティネーション (D) で指定されたフローティングデータを除算し、結果をリザルト (R) に格納します。



ソース (S)、デスティネーション (D)、リザルト (R) に指定できる値および格納される値の範囲は下記です。

$$0, \pm 2^{-126} \sim \pm 2^{128}$$

(3) データタイプ

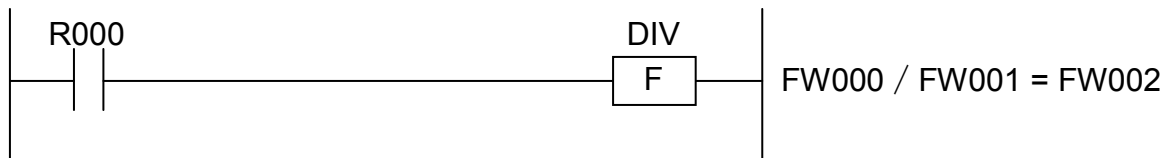
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○ : 指定できます。

— : 指定できません。

S, D、およびRの型 (ワード/ロングワード/フローティング) はすべて合わせてください。異なった型は指定できません (入力エラーになります)。

(4) プログラム例



入力条件R000がONのとき、FW000の内容とFW001の内容を除算し、結果 (商) をFW002へ格納します。



## DIV 除算 : DIVIDE

### (5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
-	↕	-	-	-	↕

V : ワードデータ時

- ・リザルト (R) = 32768 のとき1、それ以外るとき0

ロングワードデータ時

- ・リザルト (R) = 2147483648 のとき1、それ以外るとき0

フローティングデータ時

- ・演算結果によりVは変化しません (保持します)。

E : ワードおよびロングワードデータ時

- ・D = 0 のとき1、それ以外るとき0

フローティングデータ時

- ・リザルト (R) が0ではなく、かつ下記範囲外るとき1、それ以外るとき0

$$\pm 2^{-126} \sim \pm 2^{128}$$

V, E以外のフラグはすべて保持します。

- 0除算時、エラーフラグ (E) がONし (オーバーフローフラグ (V) OFF)、リザルト (R) は変化しません。
- オーバーフロー発生時、リザルト (R) には下記のフルスケール値が格納されます。

	正のオーバーフロー時	負のオーバーフロー時
ワード	H7FFF	H8000
ロングワード	H7FFFFFFF	H80000000
フローティング	+3.402823E38	-3.402823E38

フローティングでオーバーフロー発生時、VフラグはONしません (ワードおよびロングワード時はONします)。

- EフラグON時は、無処理 (リザルト (R) は変化なし) です。
- フローティングでアンダーフロー発生時、リザルト (R) には正しい符号を持った0が格納されま  
す。アンダーフロー発生時、演算結果フラグは変化しません。

このページは白紙です。

# MOD 剰余 : MOD

## (1) 入力形式

```
MOD S % D -> R
```

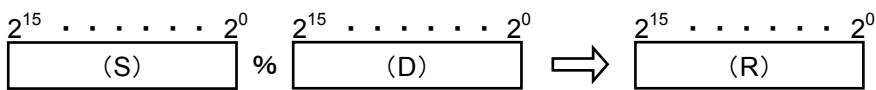
- S : ソース格納レジスタまたは定数
- D : デスティネーション格納レジスタまたは定数
- R : リザルト (演算結果) を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「%」および「->」は省略できます。

## (2) 機能

### ● ワードデータの剰余算

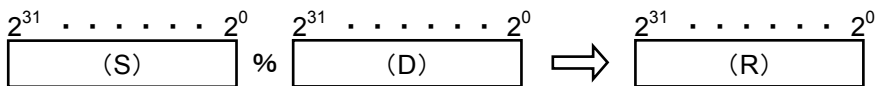
ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータを除算し、除算結果の剰余をリザルト (R) に格納します。



ソース (S)、デスティネーション (D)、リザルト (R) に指定できる値および格納される値の範囲は、-32768~32767です。

### ● ロングワードデータの剰余算

ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータを除算し、除算結果の剰余をリザルト (R) に格納します。



ソース (S)、デスティネーション (D)、リザルト (R) に指定できる値および格納される値の範囲は、-2147483648~2147483647です。

## (3) データタイプ

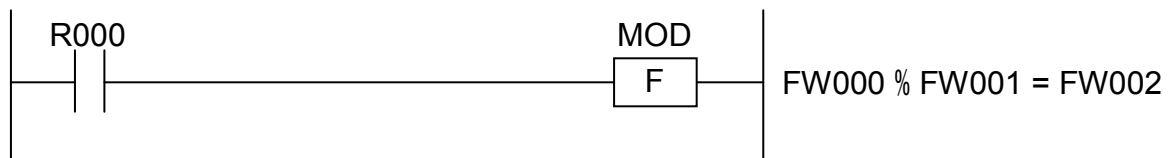
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
D	○	○	○	○	—	—	○
R	○	—	○	—	—	—	○

○ : 指定できます。

— : 指定できません。

S, D、およびRの型（ワード/ロングワード）はすべて合わせてください。異なった型は指定できません（入力エラーになります）。

## (4) プログラム例



入力条件R000がONのとき、FW000の内容とFW001の内容を除算し、除算結果の剰余をFW002へ格納します。

## (5) エラー処理

## ● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	↕

V : ワードデータ時

・商 = 32768のとき1、それ以外のとき0

ロングワードデータ時

・商 = 2147483648のとき1、それ以外のとき0

E : デスティネーション (D) = 0のとき1、それ以外のとき0

V, E以外のフラグはすべて保持します。

- 0除算時、エラーフラグ (E) がONし（オーバーフローフラグ (V) OFF）、リザルト (R) は変化しません。
- オーバーフロー発生時、リザルト (R) には0が格納されます。

# SCL スケール変換 : SCALE CHANGE

## (1) 入力形式

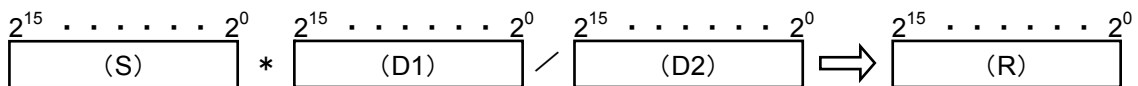
SCL S : D1 : D2 -> R

- S : ソース格納レジスタまたは定数
- D1, D2 : デスティネーション格納レジスタまたは定数
- R : リザルト (演算結果) を格納するレジスタ

(\* ) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

ソース (S) で指定された16ビットデータとデスティネーション (D1) / (D2) を乗算し、結果をリザルト (R) に格納します。



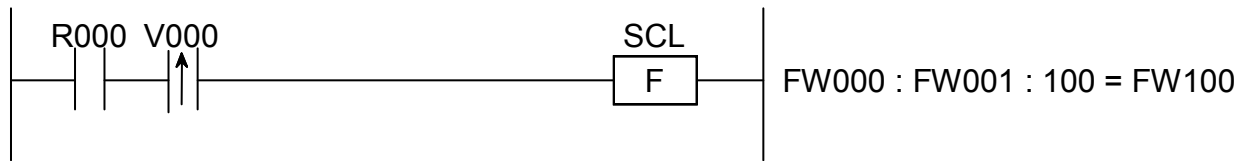
ソース (S) 、デスティネーション (D1) 、 (D2) 、リザルト (R) に指定できる値および格納される値の範囲は、-32768~32767です。

## (3) データタイプ

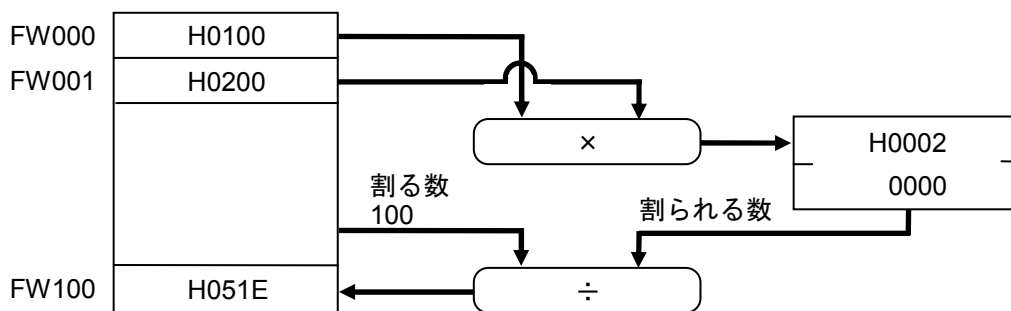
	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	—	—	—	—	○
D1	○	○	—	—	—	—	○
D2	○	○	—	—	—	—	○
R	○	—	—	—	—	—	○

- : 指定できます。
- : 指定できません。

(4) プログラム例



入力条件R000がOFF→ONに変化時、1回のみFW000の内容をスケール変換し、結果をFW100へ格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
-	↕	-	-	-	↕

V : リザルト (R) が-32768~32767のとき0、それ以外るとき1

E : デスティネーション (D2) = 0のとき1、それ以外るとき0

V, E以外のフラグはすべて保持します。

- 0除算時、エラーフラグ (E) がONし (オーバーフローフラグ (V) OFF) 、リザルト (R) は変化しません。

- オーバーフロー発生時、リザルト (R) には下記のフルスケール値が格納されます。

正のオーバーフロー時	H7FFF
負のオーバーフロー時	H8000

# AND 論理積 : AND

## (1) 入力形式

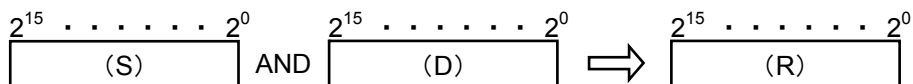
AND S : D -> R

- S : ソース格納レジスタまたは定数
- D : デスティネーション格納レジスタまたは定数
- R : リザルト (演算結果) を格納するレジスタ

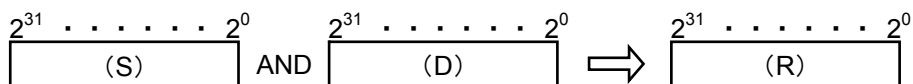
(\* ) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

- **ワードデータの論理積**  
 ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータの論理積をリザルト (R) に格納します。



- **ロングワードデータの論理積**  
 ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータの論理積をリザルト (R) に格納します。



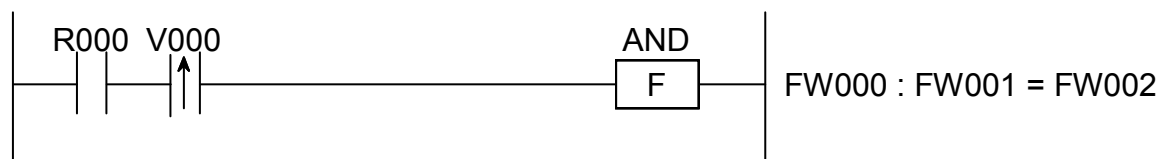
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	—	—	○
D	○	○	○	○	—	—	○
R	○	—	○	—	—	—	○

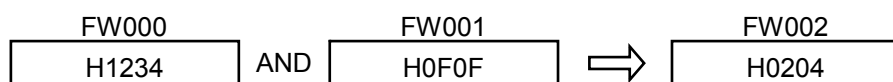
- : 指定できます。
- : 指定できません。

S, D、およびRの型 (ワード/ロングワード) はすべて合わせてください。異なった型は指定できません (入力エラーになります)。

## (4) プログラム例



入力条件R000がOFF→ONに変化時、1回のみFW000の内容とFW001の内容の論理積をFW002へ格納します。



## (5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。



# OR 論理和 : OR

## (1) 入力形式

OR S : D -> R

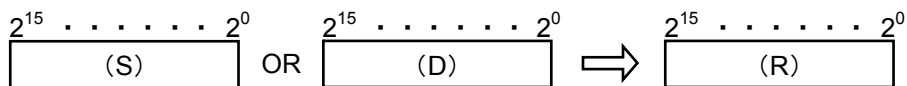
- S : ソース格納レジスタまたは定数
- D : デスティネーション格納レジスタまたは定数
- R : リザルト (演算結果) を格納するレジスタ

(\* ) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

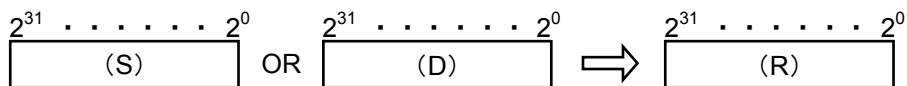
### ● ワードデータの論理和

ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータの論理和をリザルト (R) に格納します。



### ● ロングワードデータの論理和

ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータの論理和をリザルト (R) に格納します。



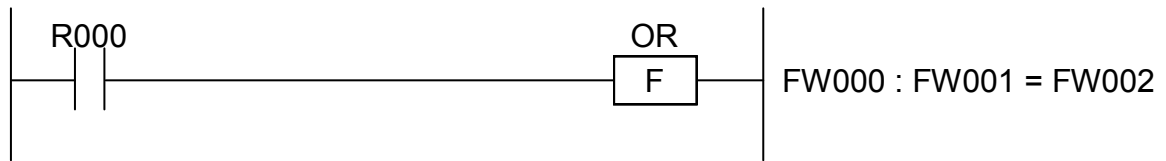
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	—	—	○
D	○	○	○	○	—	—	○
R	○	—	○	—	—	—	○

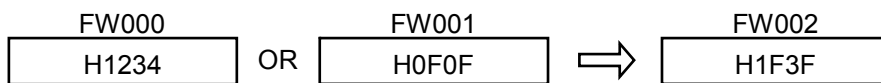
- : 指定できます。
- : 指定できません。

S, D、およびRの型 (ワード/ロングワード) はすべて合わせてください。異なった型は指定できません (入力エラーになります)。

## (4) プログラム例



入力条件R000がONのとき、FW000の内容とFW001の内容の論理和をFW002へ格納します。



## (5) エラー処理

## ● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# EOR 排他的論理和 : EOR

## (1) 入力形式

EOR S : D -> R

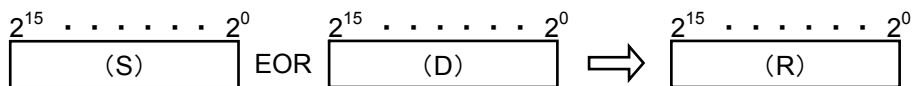
- S : ソース格納レジスタまたは定数
- D : デスティネーション格納レジスタまたは定数
- R : リザルト (演算結果) を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

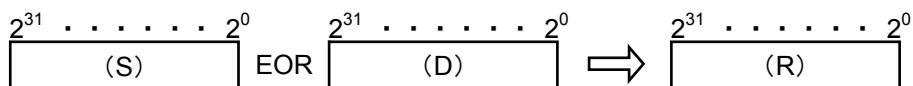
### ● ワードデータの排他的論理和

ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータの排他的論理和をリザルト (R) に格納します。



### ● ロングワードデータの排他的論理和

ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータの排他的論理和をリザルト (R) に格納します。



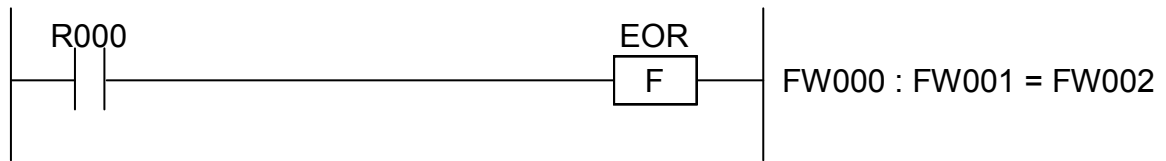
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	—	—	○
D	○	○	○	○	—	—	○
R	○	—	○	—	—	—	○

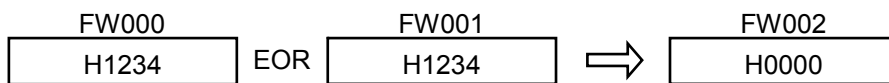
- : 指定できます。
- : 指定できません。

S, D、およびRの型 (ワード/ロングワード) はすべて合わせてください。異なった型は指定できません (入力エラーになります)。

## (4) プログラム例



入力条件R000がONのとき、FW000の内容とFW001の内容の排他的論理和をFW002へ格納します。



## (5) エラー処理

## ● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# NOT 否定 : NOT

## (1) 入力形式

NOT S -> R

S : ソース格納レジスター

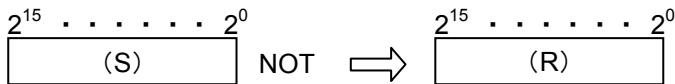
R : リザルト (演算結果) を格納するレジスター

(\* ) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

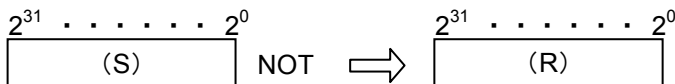
### ● ワードデータの否定

ソース (S) で指定された16ビットデータをビット反転し、リザルト (R) に格納します。



### ● ロングワードデータの否定

ソース (S) で指定された32ビットデータをビット反転し、リザルト (R) に格納します。



## (3) データタイプ

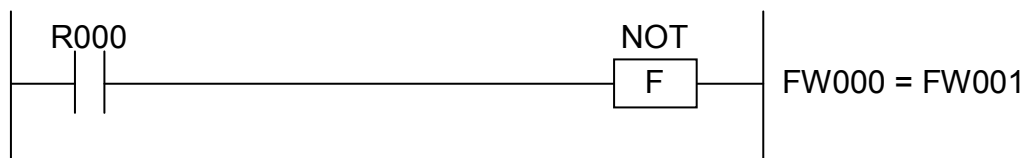
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	—	○	—	—	—	○
R	○	—	○	—	—	—	○

○ : 指定できます。

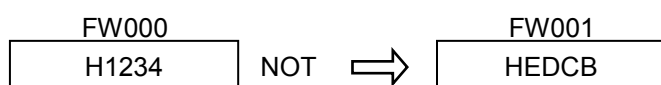
— : 指定できません。

SとRの型 (ワード/ロングワード) は合わせてください。異なった型は指定できません (入力エラーになります)。

## (4) プログラム例



入力条件R000がONのとき、FW000の内容をビット反転し、FW001へ格納します。



## (5) エラー処理

## ● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# EQU = : EQUAL

## (1) 入力形式

```
EQU S : D -> R
```

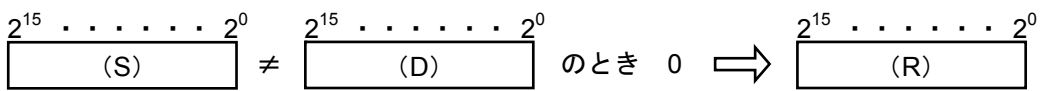
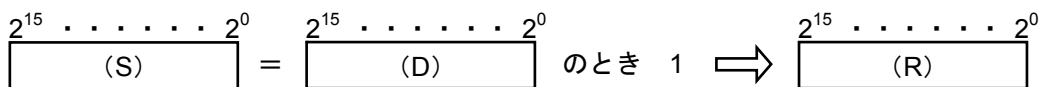
S : ソース格納レジスタまたは定数  
D : デスティネーション格納レジスタまたは定数  
R : リザルト (演算結果) を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

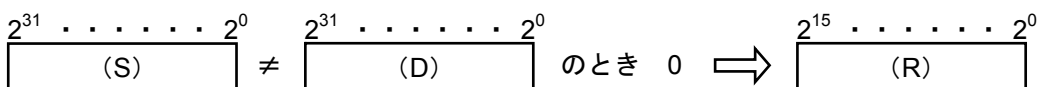
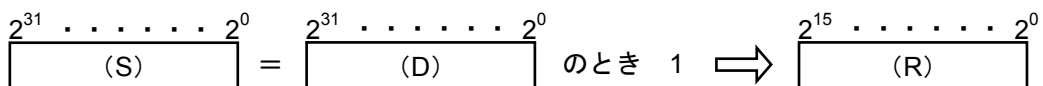
### ● ワードデータの比較

ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータの内容を比較し、等しければ1、等しくなければ0をリザルト (R) に格納します。



### ● ロングワードデータの比較

ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータの内容を比較し、等しければ1、等しくなければ0をリザルト (R) に格納します。



### ● フローティングデータの比較

ソース (S) で指定されたフローティングデータとデスティネーション (D) で指定されたフローティングデータの内容を比較し、等しければ1、等しくなければ0をリザルト (R) に格納します。



(注) フローティングデータの場合、誤差により等しくならないことがありますので注意してください。

(3) データタイプ

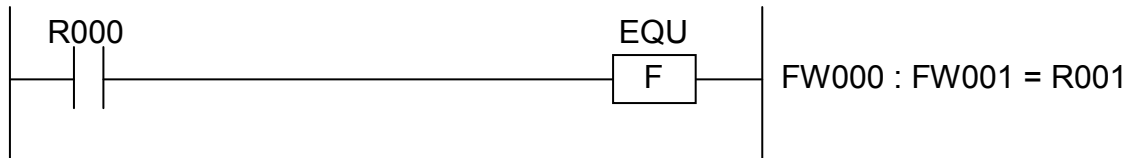
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	—	—	—	—	○

○ : 指定できます。

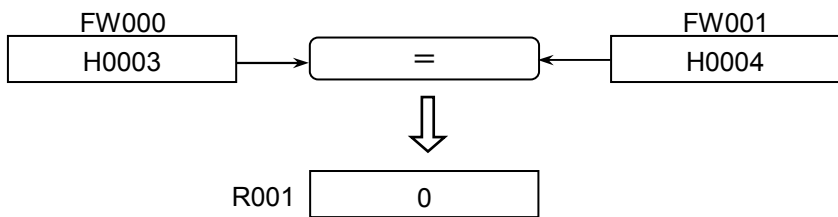
— : 指定できません。

SとDの型（ワード/ロングワード/フローティング）は合わせてください。異なった型は指定できません（入力エラーになります）。Rはワード型固定です。

(4) プログラム例



入力条件R000がONのとき、FW000とFW001の内容を比較し、比較結果をR001へ格納します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。



# NEQ $\neq$ : NOT EQUAL

## (1) 入力形式

```
NEQ S : D -> R
```

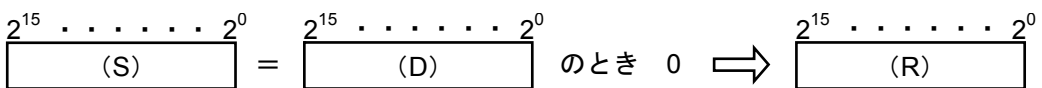
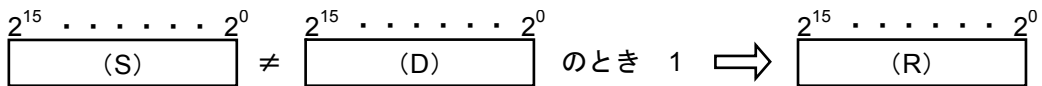
- S : ソース格納レジスタまたは定数
- D : デスティネーション格納レジスタまたは定数
- R : リザルト (演算結果) を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

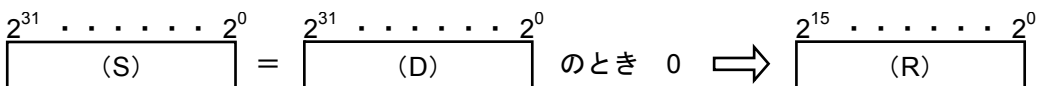
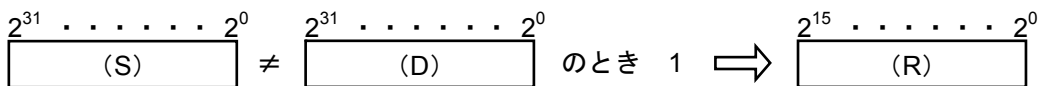
### ● ワードデータの比較

ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータの内容を比較し、等しくなければ1、等しければ0をリザルト (R) に格納します。



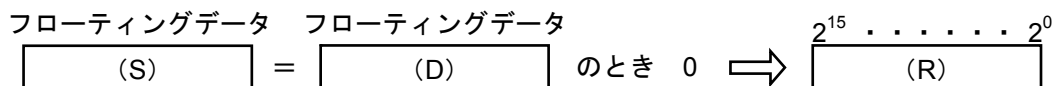
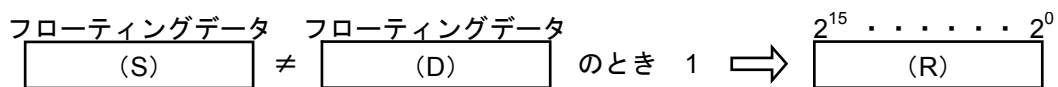
### ● ロングワードデータの比較

ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータの内容を比較し、等しくなければ1、等しければ0をリザルト (R) に格納します。



### ● フローティングデータの比較

ソース (S) で指定されたフローティングデータとデスティネーション (D) で指定されたフローティングデータの内容を比較し、等しくなければ1、等しければ0をリザルト (R) に格納します。



(3) データタイプ

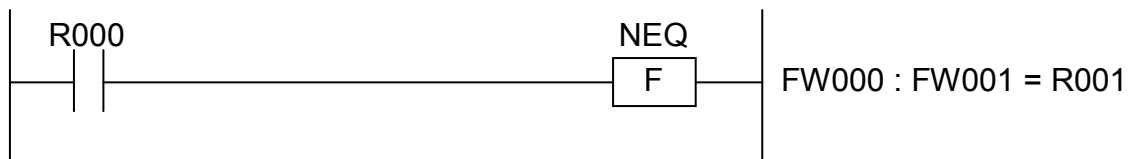
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	—	—	—	—	○

○ : 指定できます。

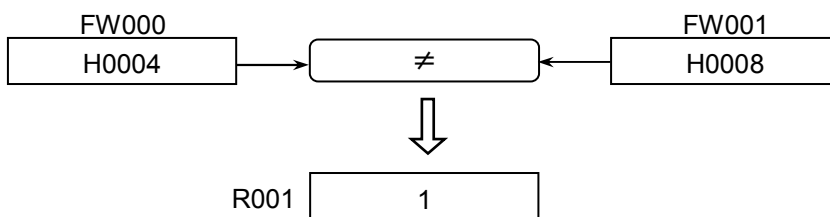
— : 指定できません。

SとDの型（ワード/ロングワード/フローティング）は合わせてください。異なった型は指定できません（入力エラーになります）。Rはワード型固定です。

(4) プログラム例



入力条件R000がONのとき、FW000とFW001の内容を比較し、比較結果をR001へ格納します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# GT > : GREATER THAN

## (1) 入力形式

GT S : D -> R

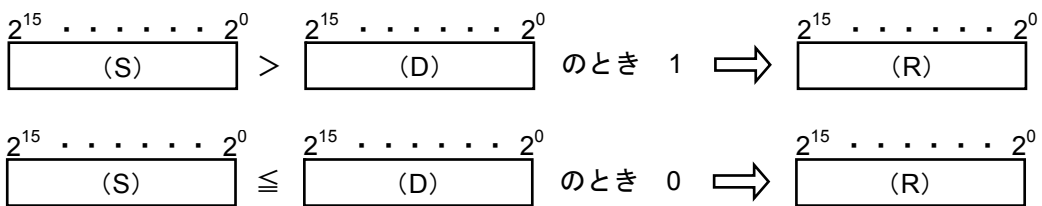
- S : ソース格納レジスタまたは定数
- D : デスティネーション格納レジスタまたは定数
- R : リザルト (演算結果) を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

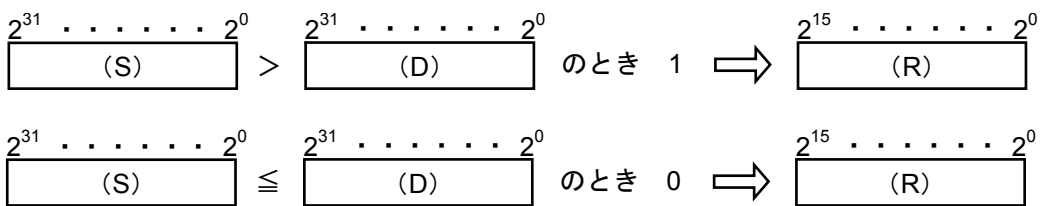
### ● ワードデータの比較

ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータを大小比較し、比較結果をリザルト (R) に格納します。



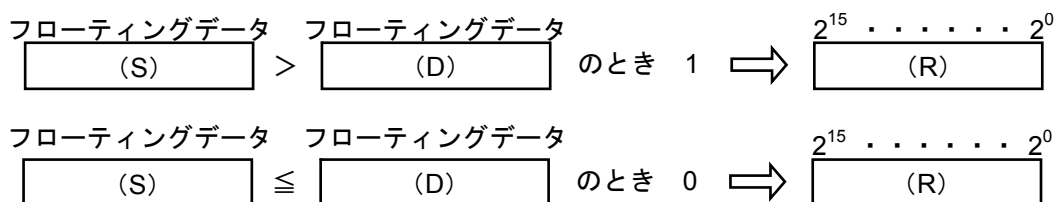
### ● ロングワードデータの比較

ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータを大小比較し、比較結果をリザルト (R) に格納します。



### ● フローティングデータの比較

ソース (S) で指定されたフローティングデータとデスティネーション (D) で指定されたフローティングデータを大小比較し、比較結果をリザルト (R) に格納します。



## (3) データタイプ

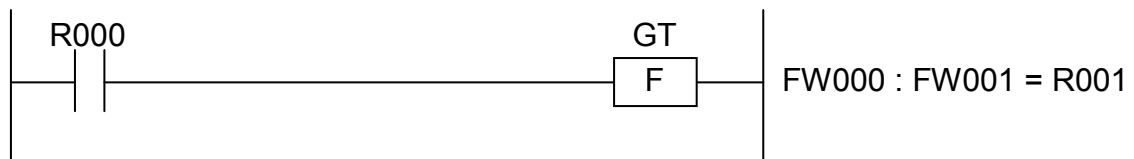
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	—	—	—	—	○

○：指定できます。

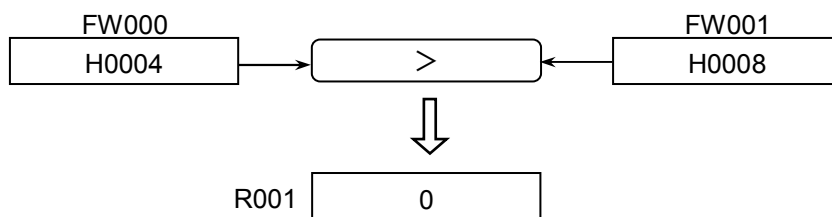
—：指定できません。

SとDの型（ワード/ロングワード/フローティング）は合わせてください。異なった型は指定できません（入力エラーになります）。Rはワード型固定です。

## (4) プログラム例



入力条件R000がONのとき、FW000とFW001の内容を大小比較し、比較結果をR001へ格納します。



## (5) エラー処理

## ● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# LT < : LESS THAN

## (1) 入力形式

LT S : D -> R

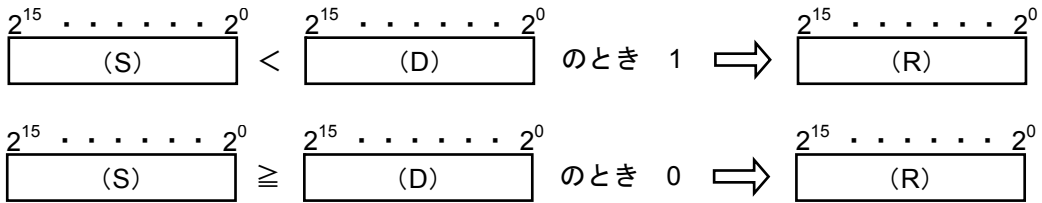
- S : ソース格納レジスタまたは定数
- D : デスティネーション格納レジスタまたは定数
- R : リザルト (演算結果) を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

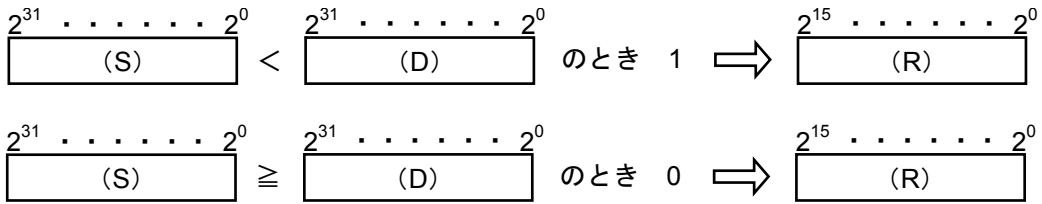
### ● ワードデータの比較

ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータを大小比較し、比較結果をリザルト (R) に格納します。



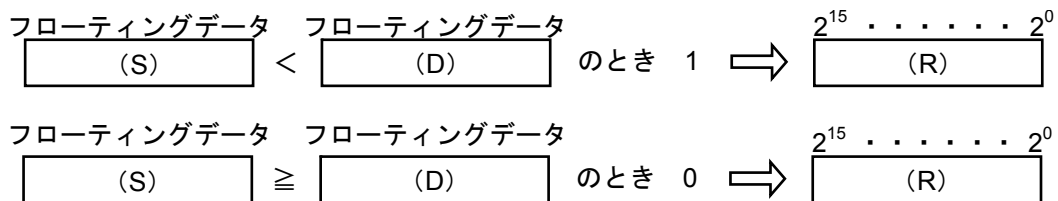
### ● ロングワードデータの比較

ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータを大小比較し、比較結果をリザルト (R) に格納します。



### ● フローティングデータの比較

ソース (S) で指定されたフローティングデータとデスティネーション (D) で指定されたフローティングデータを大小比較し、比較結果をリザルト (R) に格納します。



(3) データタイプ

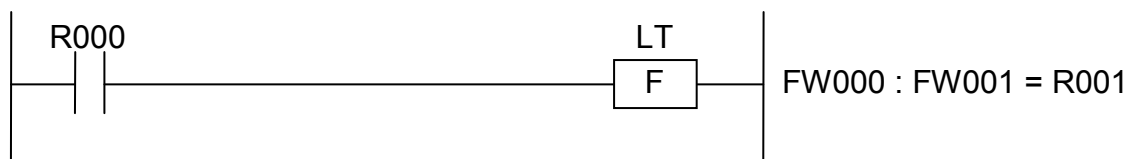
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	—	—	—	—	○

○ : 指定できます。

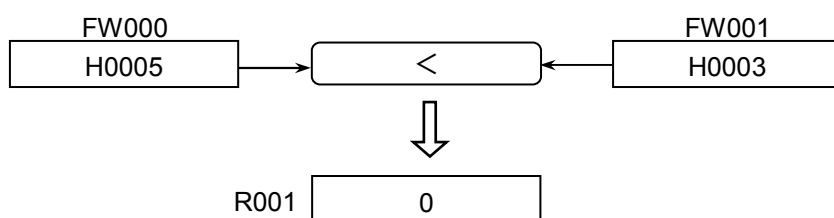
— : 指定できません。

SとDの型（ワード/ロングワード/フローティング）は合わせてください。異なった型は指定できません（入力エラーになります）。Rはワード型固定です。

(4) プログラム例



入力条件R000がONのとき、FW000とFW001の内容を大小比較し、比較結果をR001へ格納します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# GE $\geq$ : GREATER or EQUAL

## (1) 入力形式

```
GE S : D -> R
```

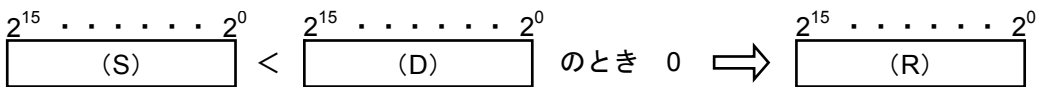
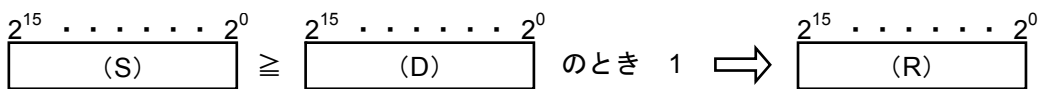
- S : ソース格納レジスタまたは定数
- D : デスティネーション格納レジスタまたは定数
- R : リザルト（演算結果）を格納するレジスタ

(\*) シンボル（命令語名称）とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

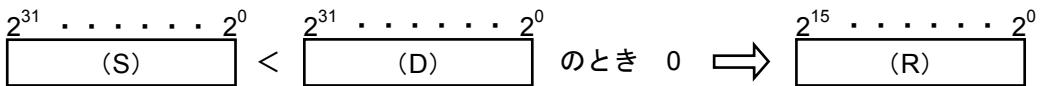
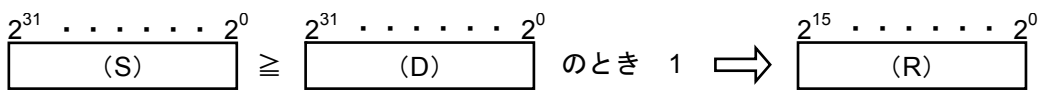
### ● ワードデータの比較

ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータを大小比較し、比較結果をリザルト (R) に格納します。



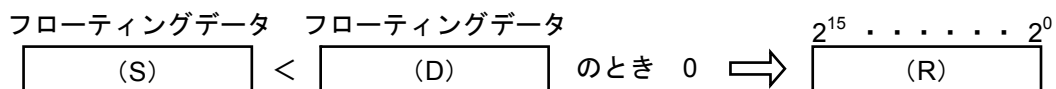
### ● ロングワードデータの比較

ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータを大小比較し、比較結果をリザルト (R) に格納します。



### ● フローティングデータの比較

ソース (S) で指定されたフローティングデータとデスティネーション (D) で指定されたフローティングデータを大小比較し、比較結果をリザルト (R) に格納します。



(3) データタイプ

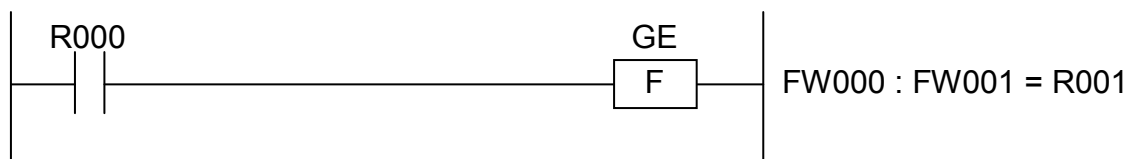
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	—	—	—	—	○

○ : 指定できます。

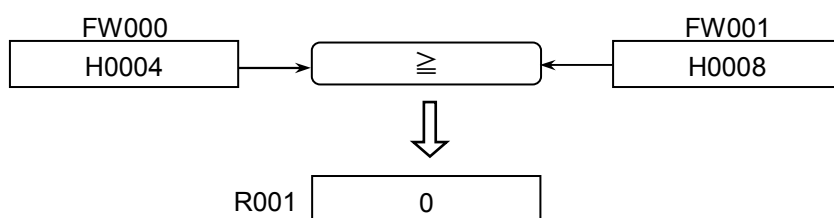
— : 指定できません。

SとDの型（ワード/ロングワード/フローティング）は合わせてください。異なった型は指定できません（入力エラーになります）。Rはワード型固定です。

(4) プログラム例



入力条件R000がONのとき、FW000とFW001の内容を大小比較し、比較結果をR001へ格納します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。



# LE $\leq$ : LESS or EQUAL

## (1) 入力形式

```
LE S : D -> R
```

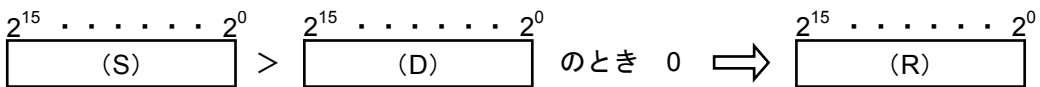
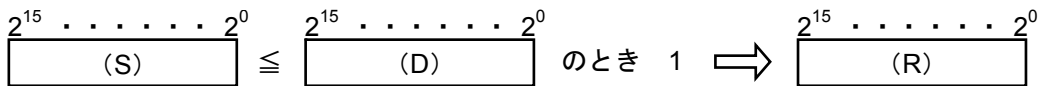
- S : ソース格納レジスタまたは定数
- D : デスティネーション格納レジスタまたは定数
- R : リザルト (演算結果) を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

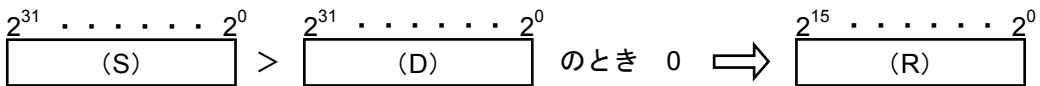
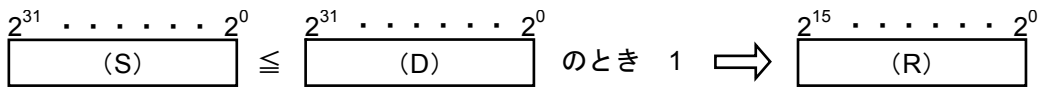
### ● ワードデータの比較

ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータを大小比較し、比較結果をリザルト (R) に格納します。



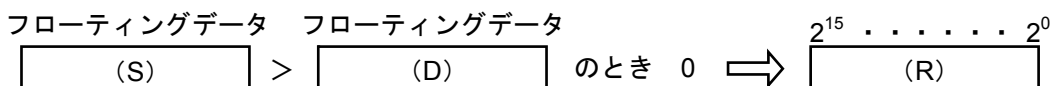
### ● ロングワードデータの比較

ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータを大小比較し、比較結果をリザルト (R) に格納します。



### ● フローティングデータの比較

ソース (S) で指定されたフローティングデータとデスティネーション (D) で指定されたフローティングデータを大小比較し、比較結果をリザルト (R) に格納します。



## (3) データタイプ

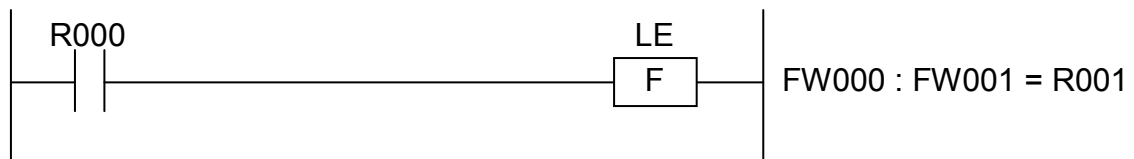
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	—	—	—	—	○

○ : 指定できます。

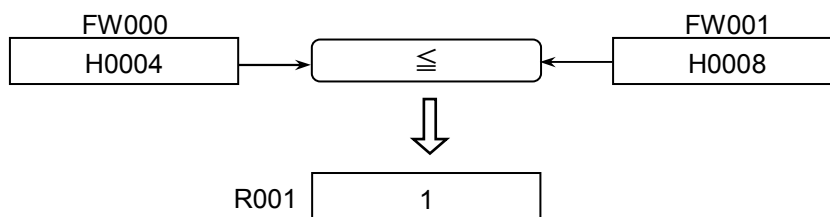
— : 指定できません。

SとDの型（ワード/ロングワード/フローティング）は合わせてください。異なった型は指定できません（入力エラーになります）。Rはワード型固定です。

## (4) プログラム例



入力条件R000がONのとき、FW000とFW001の内容を大小比較し、比較結果をR001へ格納します。



## (5) エラー処理

## ● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# TST テスト : TEST

## (1) 入力形式

TST S

S : ソース格納レジスター

(\* ファンクション名称とパラメーターの間には、必ずスペースを入力してください。)

## (2) 機能

ソース (S) の内容の極性をテストし、ポジティブ (P)、ネガティブ (N)、ゼロ (Z) の各フラグを設定します。P, N, Z以外のフラグは値を保持します。

<演算結果フラグ>

X	E	P	N	Z	V
-	-	↕	↕	↕	-

### ● ワードデータのテスト

$$\boxed{\begin{matrix} 2^{15} & \dots & 2^0 \\ (S) \end{matrix}} > 0 \quad : \text{P ON (N, Z OFF)}$$

$$\boxed{\begin{matrix} 2^{15} & \dots & 2^0 \\ (S) \end{matrix}} = 0 \quad : \text{Z ON (P, N OFF)}$$

$$\boxed{\begin{matrix} 2^{15} & \dots & 2^0 \\ (S) \end{matrix}} < 0 \quad : \text{N ON (P, Z OFF)}$$

### ● ロングワードデータのテスト

$$\boxed{\begin{matrix} 2^{31} & \dots & 2^0 \\ (S) \end{matrix}} > 0 \quad : \text{P ON (N, Z OFF)}$$

$$\boxed{\begin{matrix} 2^{31} & \dots & 2^0 \\ (S) \end{matrix}} = 0 \quad : \text{Z ON (P, N OFF)}$$

$$\boxed{\begin{matrix} 2^{31} & \dots & 2^0 \\ (S) \end{matrix}} < 0 \quad : \text{N ON (P, Z OFF)}$$

### ● フローティングデータのテスト

フローティングデータ

$$\boxed{\begin{matrix} (S) \end{matrix}} > 0 \quad : \text{P ON (N, Z OFF)}$$

フローティングデータ

$$\boxed{\begin{matrix} (S) \end{matrix}} = 0 \quad : \text{Z ON (P, N OFF)}$$

フローティングデータ

$$\boxed{\begin{matrix} (S) \end{matrix}} < 0 \quad : \text{N ON (P, Z OFF)}$$

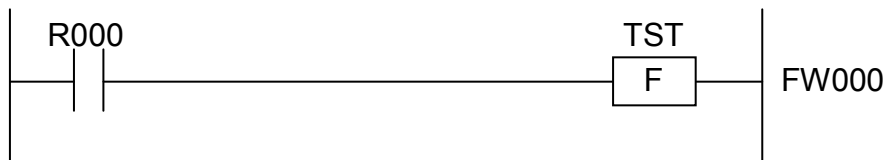
(3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	—	○	—	○	—	○

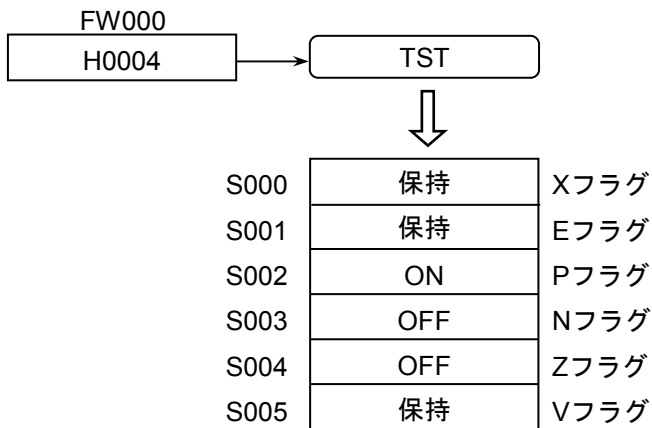
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がONのとき、FW000の内容の極性をテストし、フラグを設定します。



(5) エラー処理

- フローティング演算時に非数、無限大をソース (S) に指定した場合、演算結果フラグは下記になります。

ソース (S)	演算結果フラグ
非数	N ON (P, Z OFF)
+無限大	P ON (N, Z OFF)
-無限大	N ON (P, Z OFF)

# MOV 転送 : MOVE

## (1) 入力形式

MOV S -> D

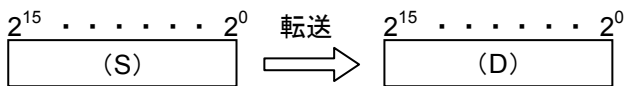
S : ソース格納レジスタまたは定数  
 D : デスティネーション格納レジスタ

(\* ) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) で指定されたデータの内容をデスティネーション (D) に転送します。

- ワードデータの転送



- ロングワードデータの転送



- フローティングデータの転送



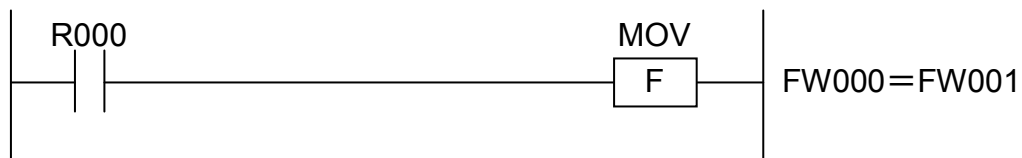
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	○	○	○
D	○	—	○	—	○	—	○

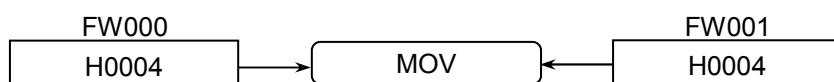
○ : 指定できます。  
 — : 指定できません。

SとDの型 (ワード/ロングワード/フローティング) は合わせてください。異なった型は指定できません (入力エラーになります)。

## (4) プログラム例



入力条件R000がONのとき、FW000の内容をFW001に転送します。



## (5) エラー処理

## ● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# MOM 一括転送 : MOVE MULTI

## (1) 入力形式

```
MOM S : n -> D
```

S : ソース格納レジスタ

n : 転送ワード数または転送ロングワード数 (定数)

D : デスティネーション格納レジスタ

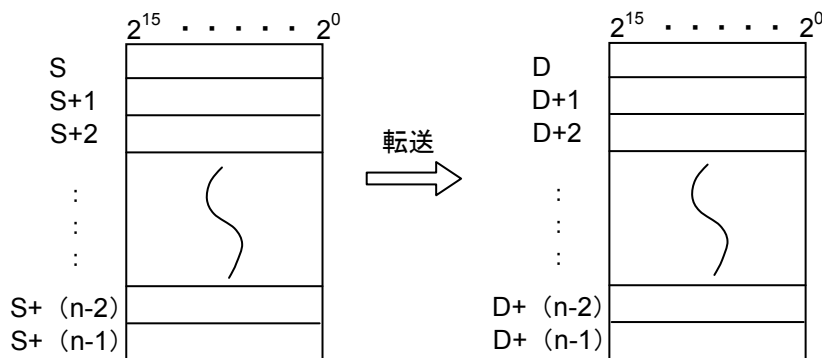
(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

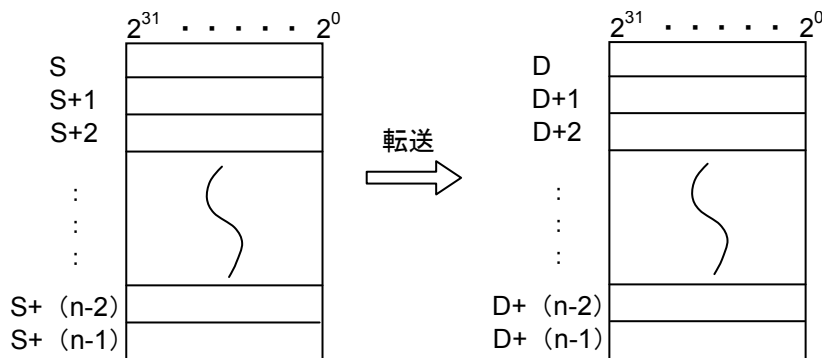
ソース (S) からnステップをデスティネーション (D) からnステップに転送します。

nには1~256ステップを指定できます (1~256以外は無処理です)。

- ワードデータの一括転送



- ロングワードデータの一括転送



(3) データタイプ

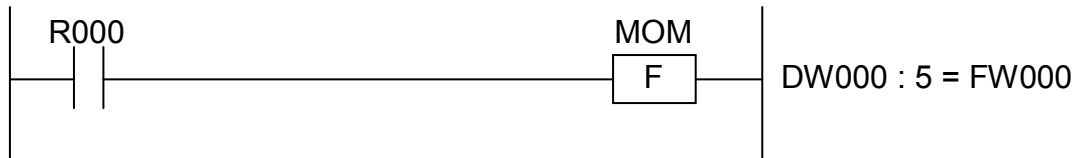
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	—	○	—	—	—	○
n	—	○	—	—	—	—	—
D	○	—	○	—	—	—	○

○ : 指定できます。

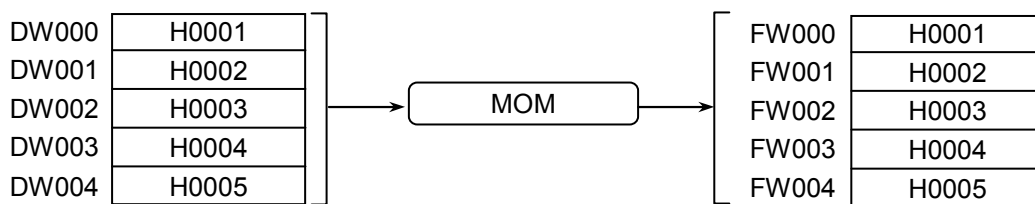
— : 指定できません。

SとDの型（ワード/ロングワード）は合わせてください。異なった型は指定できません（入力エラーになります）。nはワード型定数固定です。

(4) プログラム例



入力条件R000がONのとき、DW000から5ステップをFW000から5ステップに転送します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。



# INI 同一データ一括転送：INITIAL

## (1) 入力形式

INI S : n -> D

S : ソース格納レジスタまたは定数

n : 転送ワード数または転送ロングワード数 (定数)

D : デスティネーション格納レジスタ

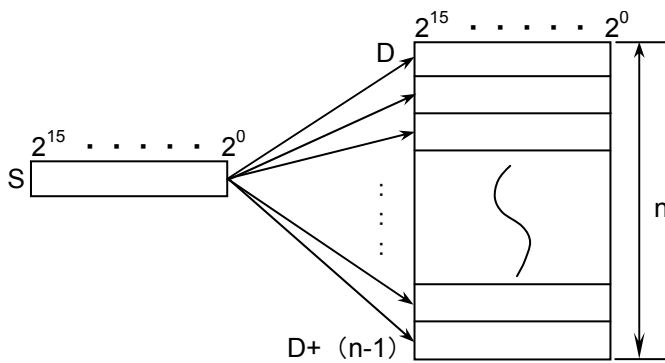
(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

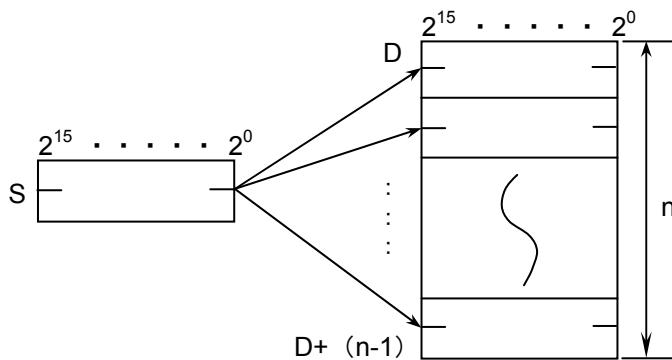
ソース (S) の内容をデスティネーション (D) から n ステップに転送します。

n には 1~256 ステップを指定できます (1~256 以外は無処理です)。

- 同じワードデータの一括転送



- 同じロングワードデータの一括転送



(3) データタイプ

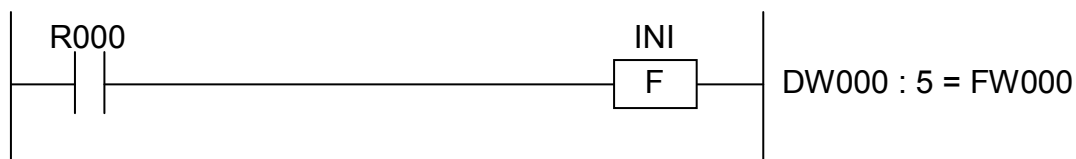
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
n	—	○	—	—	—	—	—
D	○	—	○	—	—	—	○

○ : 指定できます。

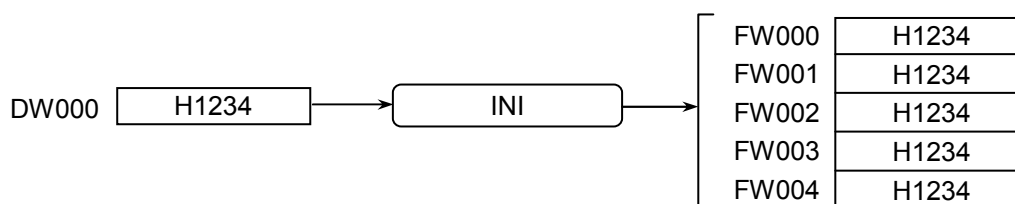
— : 指定できません。

SとDの型（ワード/ロングワード）は合わせてください。異なった型は指定できません（入力エラーになります）。nはワード型固定です。

(4) プログラム例



入力条件R000がONのとき、DW000の内容をFW000から5ステップに転送します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# EXC 交換 : EXCHANGE

## (1) 入力形式

EXC S : D

S : ソース格納レジスター

D : デスティネーション格納レジスター

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」は省略できます。

## (2) 機能

ソース (S) の内容とデスティネーション (D) の内容を交換 (データの入れ替え) します。

- ワードデータの交換



- ロングワードデータの交換



## (3) データタイプ

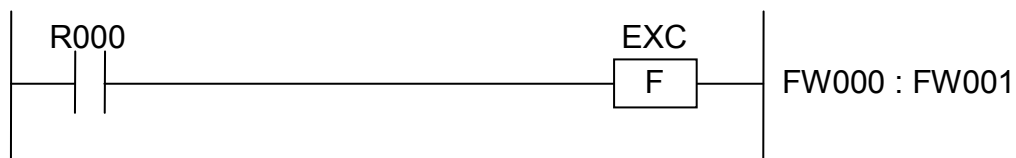
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	—	○	—	—	—	○
D	○	—	○	—	—	—	○

○ : 指定できます。

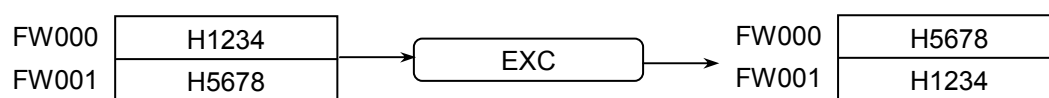
— : 指定できません。

SとDの型 (ワード/ロングワード) は合わせてください。異なった型は指定できません (入力エラーになります)。

## (4) プログラム例



入力条件R000がONのとき、FW000の内容とFW001の内容を交換します。



## (5) エラー処理

## ● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# PSH FIFO書き込み : FIFO PUSH

## (1) 入力形式

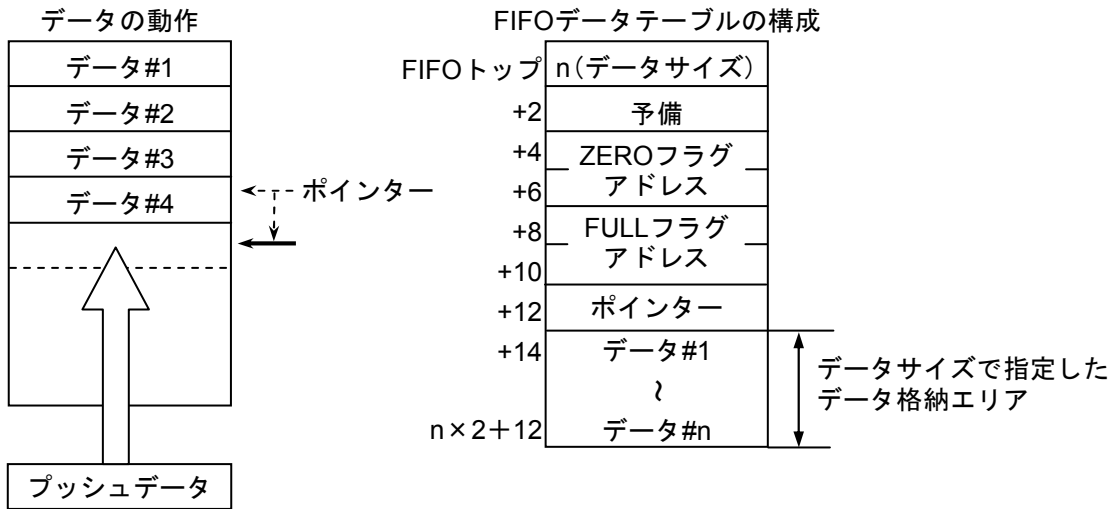
```
PSH S -> TB
```

S : ソース格納レジスター  
TB : FIFOテーブル先頭レジスター

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) の内容をFIFOテーブルへプッシュします。



### (注)

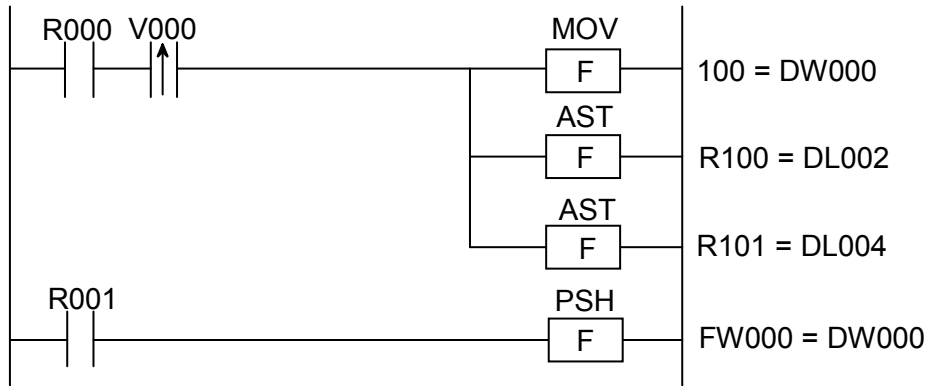
- ・プッシュ処理前にポインタがnであれば、FULLフラグをONし、プッシュ無処理とします（ただし、ZEROフラグはOFFします）。また、プッシュ処理後、ポインタをインクリメントしてnになった場合も、FULLフラグをONします。それ以外の場合は、FULLフラグをOFFします。
- ・演算実行後、ZEROフラグはOFFします（下記条件により無処理の場合を除きます）。
- ・データサイズ $n \leq 0$ または $256 < n$ の場合は、無処理です。
- ・ポインタ $< 0$ または $n < \text{ポインタ}$ の場合は、無処理です。

## (3) データタイプ

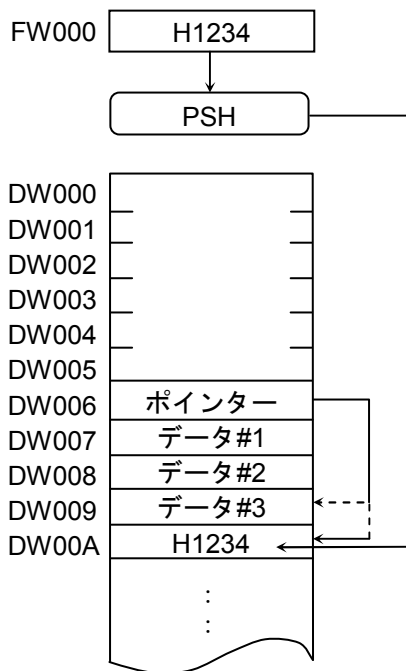
	ワード		ロングワード		フローティング		インデックス指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	—	—	—	—	—	○
TB	○	—	—	—	—	—	○

○ : 指定できます。  
— : 指定できません。

(4) プログラム例



入力条件R000がOFF→ONに変化時、1回のみデータサイズ（100）とZEROフラグ（R100）、FULLフラグ（R101）のアドレスを設定します。入力条件R001がONのとき、FW000の内容をDW000からFIFOデータテーブルへ設定します（データサイズnは、MOV（転送）によりイミディエートデータを設定します。ZEROフラグ、FULLフラグのアドレスはAST（アドレスセット）により設定します）。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
-	-	-	-	-	-

フラグはすべて保持します。

# POP FIFO読み出し : FIFO POP

## (1) 入力形式

```
POP TB -> D
```

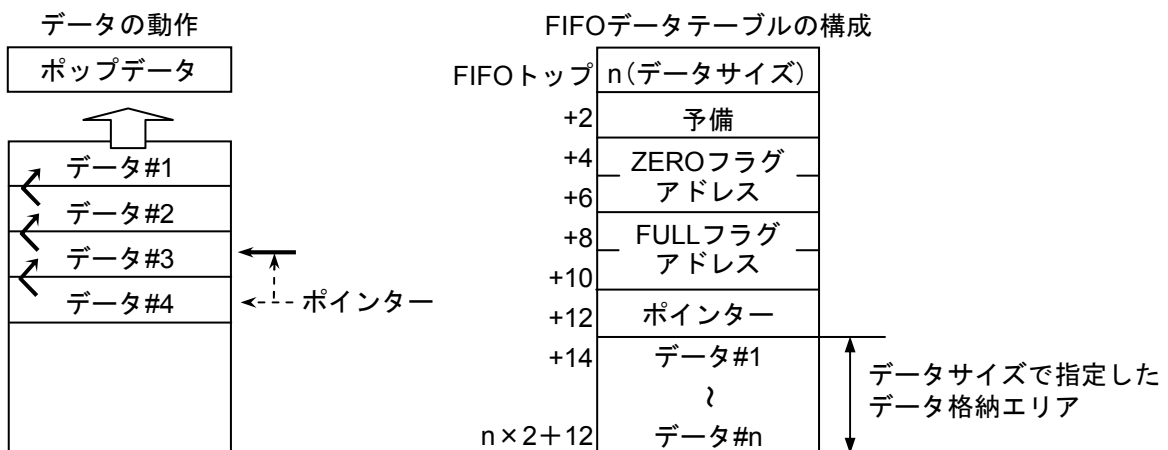
TB : FIFOテーブル先頭アドレス (レジスター)

D : デスティネーション格納レジスター

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

FIFOテーブルをポップ処理し、ポップデータをデスティネーション (D) へ格納します。



### (注)

- ・ポップ処理前にポインターが0であれば、ZEROフラグをONし、ポップ無処理とします (ただし、FULLフラグはOFFします)。また、ポップ処理後、ポインターをデクリメントして0になった場合も、ZEROフラグをONします。それ以外の場合は、ZEROフラグをOFFします。
- ・演算実行後、FULLフラグをOFFします (下記条件により無処理の場合を除きます)。
- ・データサイズ  $n \leq 0$  または  $256 < n$  の場合は、無処理です。
- ・ポインター  $< 0$  または  $n < \text{ポインター}$  の場合は、無処理です。

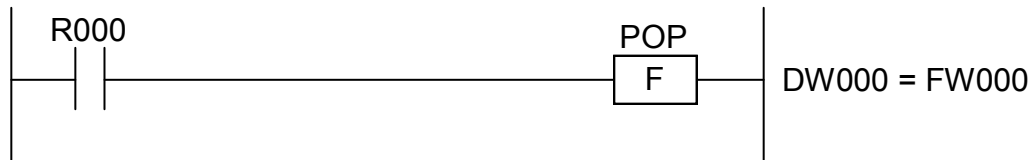
(3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
D	○	—	—	—	—	—	○
TB	○	—	—	—	—	—	○

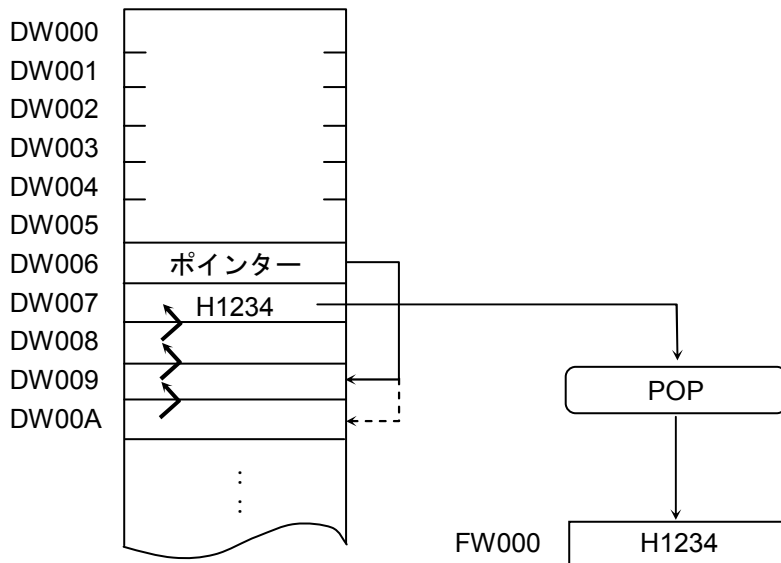
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がONのとき、DW000のFIFOデータテーブルからデータを取り込み、FW000に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。



# PSHO FIFO書き込み : FIFO PUSH

## (1) 入力形式

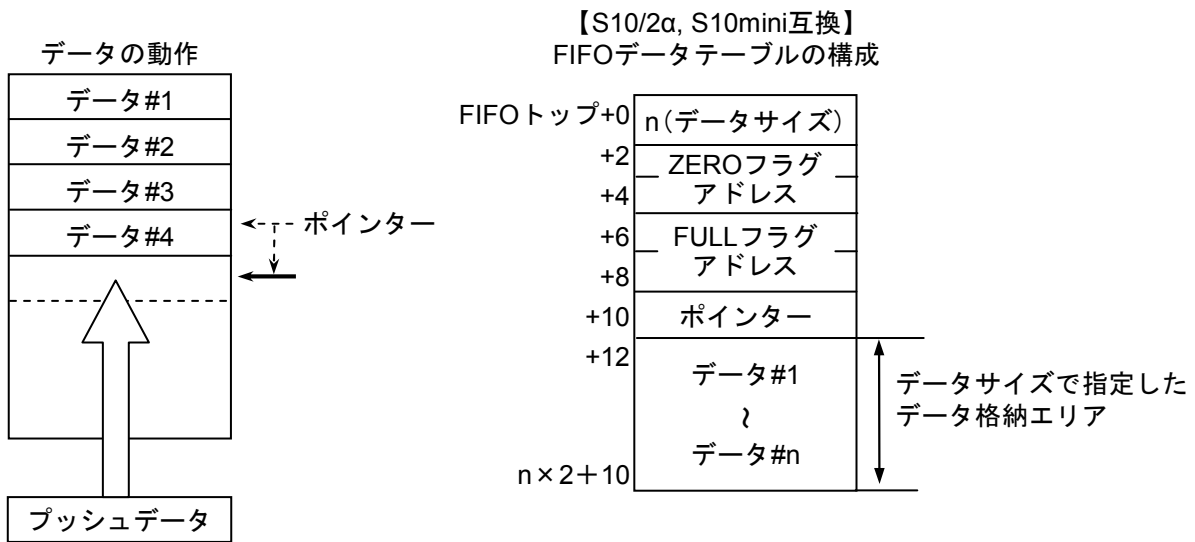
PSHO S -> TB

S : ソース格納レジスター  
 TB : FIFOテーブル先頭レジスター

(\* ) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) の内容をS10/2α, S10mini互換FIFOテーブルへプッシュします。



### (注)

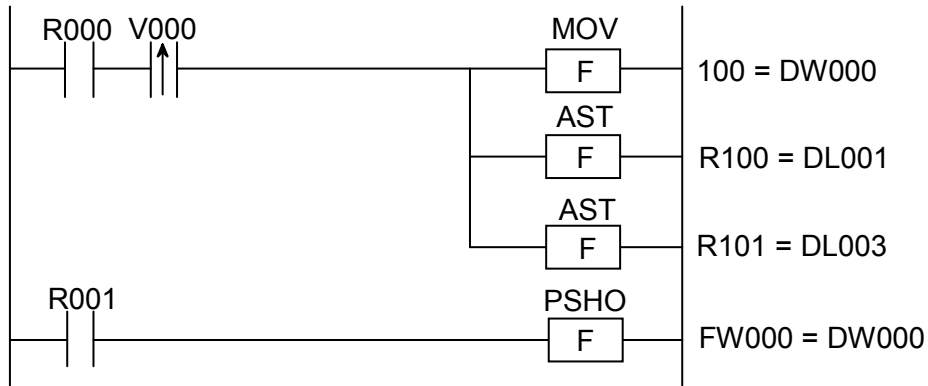
- ・プッシュ処理前にポインターがnであれば、FULLフラグをONし、プッシュ無処理とします（ただし、ZEROフラグはOFFします）。また、プッシュ処理後、ポインターをインクリメントしてnになった場合も、FULLフラグをONします。それ以外の場合は、FULLフラグをOFFします。
- ・演算実行後、ZEROフラグはOFFします（下記条件により無処理の場合を除きます）。
- ・データサイズn ≤ 0または256 < nの場合は、無処理です。
- ・ポインター < 0またはn < ポインターの場合は、無処理です。

## (3) データタイプ

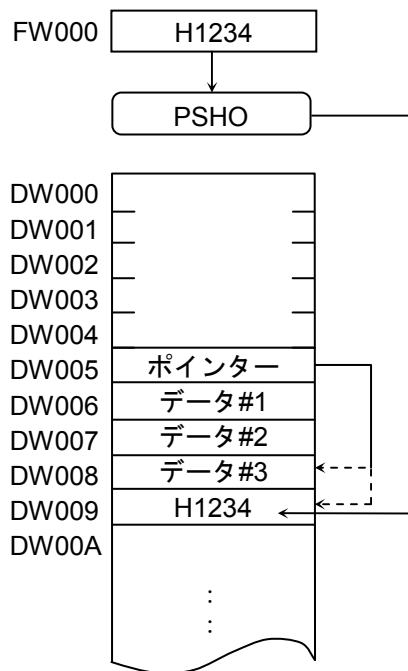
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	—	—	—	—	—	○
TB	○	—	—	—	—	—	○

○ : 指定できます。  
 — : 指定できません。

(4) プログラム例



入力条件R000がOFF→ONに変化時、1回のみデータサイズ（100）とZEROフラグ（R100）、FULLフラグ（R101）のアドレスを設定します。入力条件R001がONのとき、FW000の内容をDW000からFIFOデータテーブルへ設定します（データサイズnは、MOV（転送）によりイミディエートデータを設定します。ZEROフラグ、FULLフラグのアドレスはAST（アドレスセット）により設定します）。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
-	-	-	-	-	-

フラグはすべて保持します。

# POPO FIFO読み出し : FIFO POPO

## (1) 入力形式

POPO TB -> D

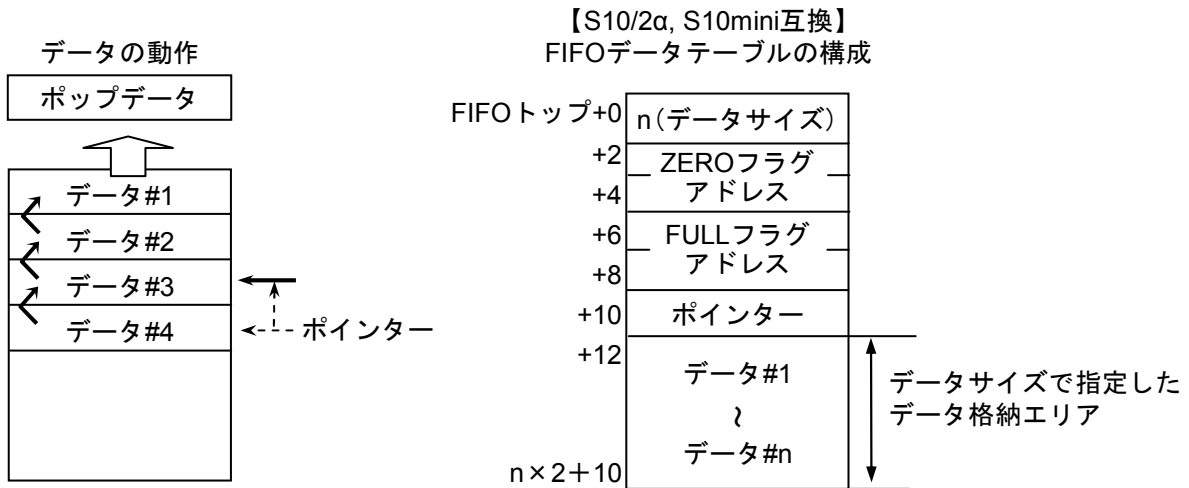
TB : FIFOテーブル先頭アドレス (レジスター)

D : デスティネーション格納レジスター

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

S10/2α, S10mini互換FIFOテーブルをポップ処理し、ポップデータをデスティネーション (D) へ格納します。



### (注)

- ポップ処理前にポインターが0であれば、ZEROフラグをONし、ポップ無処理とします (ただし、FULLフラグはOFFします)。また、ポップ処理後、ポインターをデクリメントして0になった場合も、ZEROフラグをONします。それ以外の場合は、ZEROフラグをOFFします。
- 演算実行後、FULLフラグをOFFします (下記条件により無処理の場合を除きます)。
- データサイズ $n \leq 0$ または $256 < n$ の場合は、無処理です。
- ポインター $< 0$ または $n < \text{ポインター}$ の場合は、無処理です。

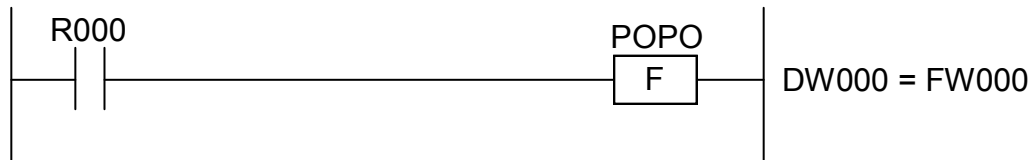
(3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
D	○	—	—	—	—	—	○
TB	○	—	—	—	—	—	○

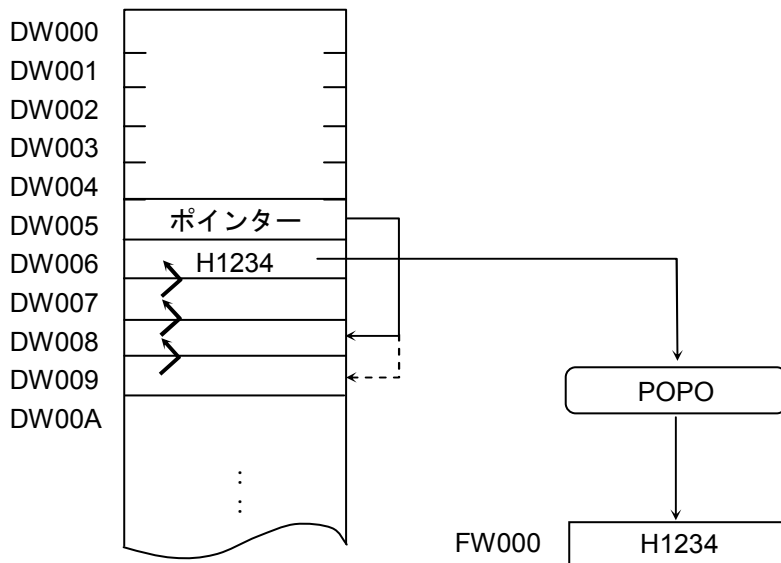
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がONのとき、DW000のFIFOデータテーブルからデータを取り込み、FW000に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# AST アドレスセット : ADDRESS SET

## (1) 入力形式

AST S -> D

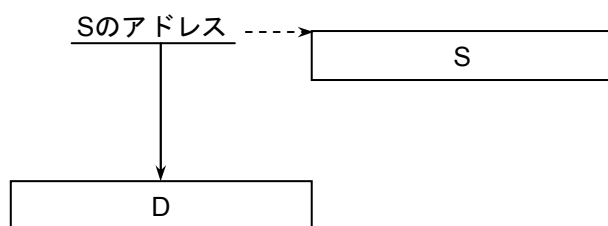
S : ソース格納レジスタ

D : デスティネーション格納レジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) のアドレスデータをデスティネーション (D) に格納します。



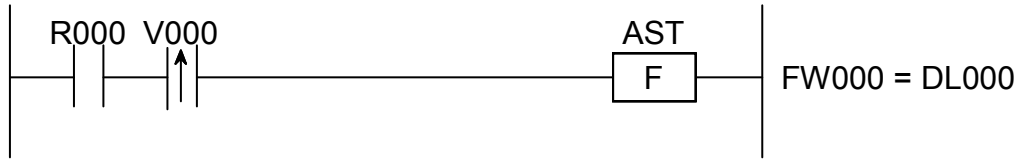
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	—	—	—	—	—	○
D	—	—	○	—	—	—	○

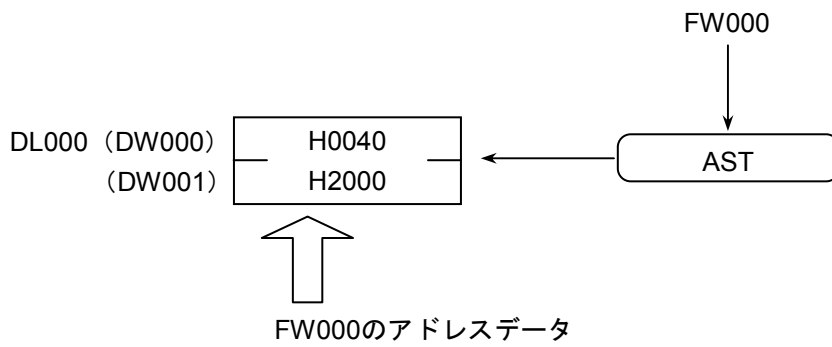
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がOFF→ONに変化時、FW000のアドレスデータをDL000に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# SCH サーチ : SEARCH

## (1) 入力形式

```
SCH S : D : m -> R
```

S : ソース格納レジスタまたは定数

D : デスティネーション格納レジスタ

m : サーチステップ数 (定数)

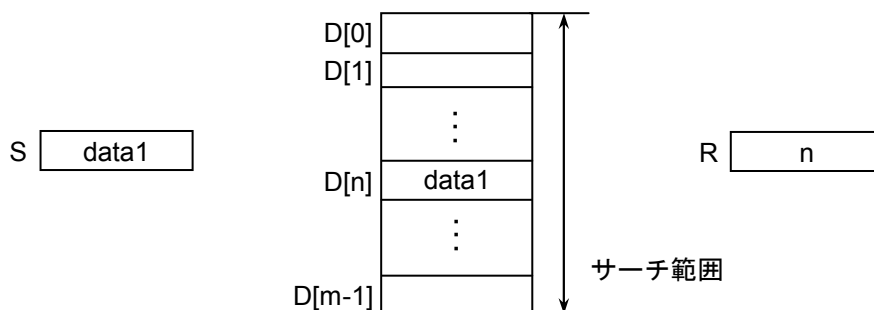
R : リザルト (演算結果) を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

### ● ワードデータのサーチ

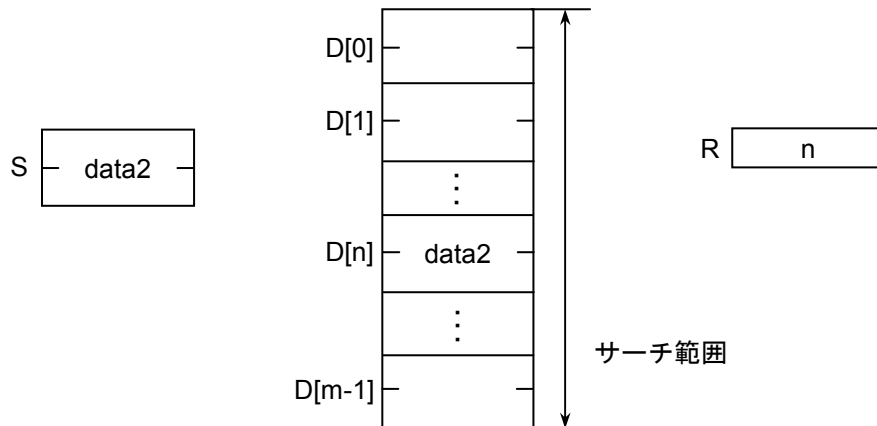
ソース (S) で指定されたワードデータの内容が、指定されたデスティネーション (D) から m ステップ数内に存在するかどうかをサーチし、ソース (S) で指定されたワードデータと一致するデータを探し出すまでのステップ数を指定されたリザルト (R) に格納します。



- 一致データは、サーチ範囲の先頭から最初に見つけたものが対象となります。
- サーチ範囲に一致データがない場合、リザルト (R) には -1 (HFFF) が格納されます。
- m (サーチステップ数) が、1~256の範囲外であった場合は無処理です。

● ログワードデータのサーチ

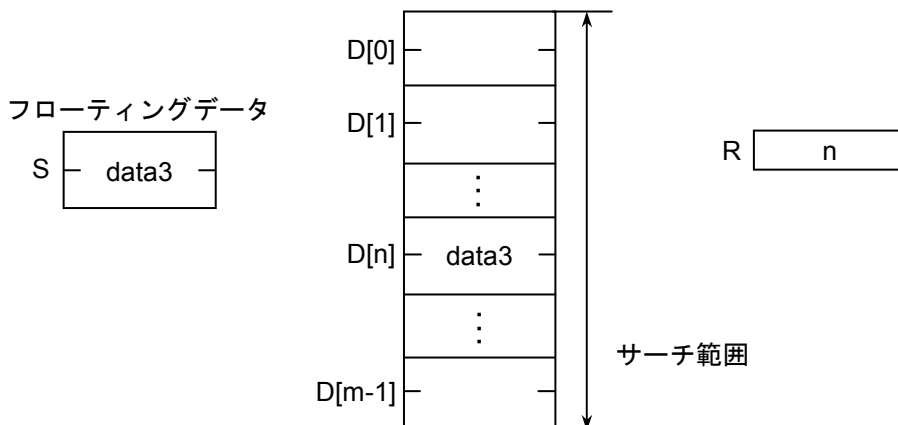
ソース (S) で指定されたログワードの内容が、指定されたデスティネーション (D) からmステップ数内に存在するかどうかサーチし、ソース (S) で指定されたログワードと一致するデータを探し出すまでのステップ数を指定されたリザルト (R) に格納します。



- 一致データは、サーチ範囲の先頭から最初に見つけたものが対象となります。
- サーチ範囲に一致データがない場合、リザルト (R) には-1 (HFFFF) が格納されます。
- m (サーチステップ数) が、1～256の範囲外であった場合は無処理です。

● フローティングデータのサーチ

ソース (S) で指定されたフローティングデータの内容が、指定されたデスティネーション (D) からmステップ数内に存在するかどうかサーチし、ソース (S) で指定されたフローティングデータと一致するデータを探し出すまでのステップ数を、指定されたリザルト (R) に格納します。



- 一致データは、サーチ範囲の先頭から最初に見つけたものが対象となります。
- サーチ範囲に一致データがない場合、リザルト (R) には-1 (HFFFF) が格納されます。
- m (サーチステップ数) が、1～256の範囲外であった場合は無処理です。

(注) フローティングデータの場合、誤差によりデータが一致しないことがあるので注意してください。



# SCH サーチ : SEARCH

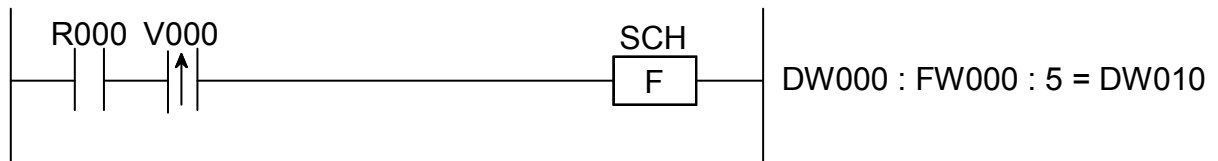
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	—	○	—	○	—	○
m	—	○	—	—	—	—	—
R	○	—	—	—	—	—	○

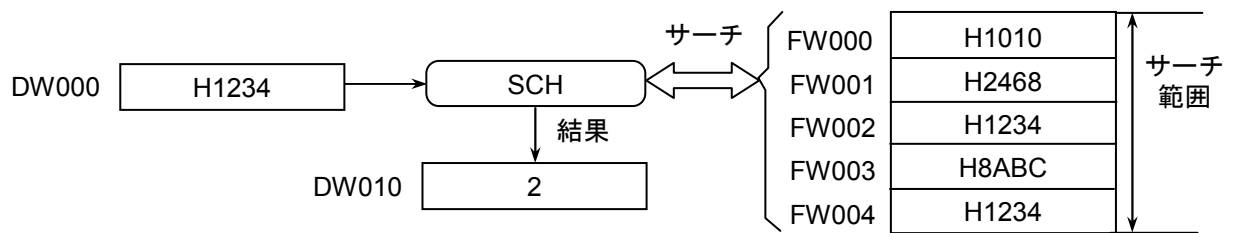
○ : 指定できます。  
 — : 指定できません。

SとDの型（ワード/ロングワード/フローティング）は合わせてください。異なった型は指定できません（入力エラーになります）。mはワード型定数、Rはワード型アドレス固定です。

## (4) プログラム例



入力条件R000がOFF→ONに変化時、1回のみDW000の内容と同じデータがFW000から5ステップ内に存在するかどうかサーチし、その結果をDW010に格納します。



## (5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

このページは白紙です。

# BTF BINARY→FLOAT変換

## (1) 入力形式

BTF S -> R

S : バイナリーデータ格納レジスタまたはバイナリー定数

R : 変換したフローティングデータを格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

### ● ワードデータのフローティング変換

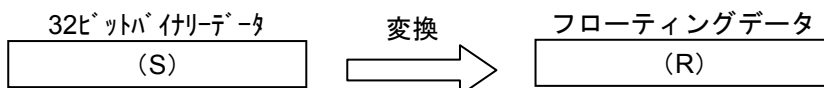
ソース (S) で指定された16ビットのバイナリーデータをフローティングデータに変換し、結果をリザルト (R) に格納します。



ソース (S) に指定できる値の範囲は、-32768～32767です。

### ● ロングワードデータのフローティング変換

ソース (S) で指定された32ビットのバイナリーデータをフローティングデータに変換し、結果をリザルト (R) に格納します。



ソース (S) に指定できる値の範囲は、-2147483648～2147483647です。

## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	—	—	○
R	—	—	—	—	○	—	○

○ : 指定できます。

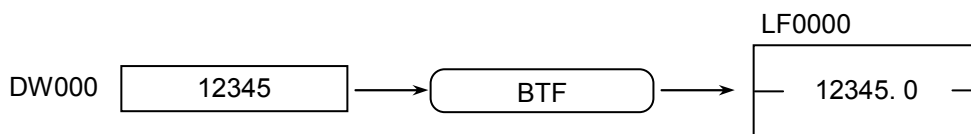
— : 指定できません。

Rはフローティング型固定です。

(4) プログラム例



入力条件R000がONのとき、DW000の内容をフローティングデータに変換し、LF0000に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

(注) フローティングデータは、32ビットの単精度で処理するため、有効桁数が2進数で表現したときは24ビット、10進数で表現したときは約7桁になります。

したがって、整数値が-16777216～16777215（24ビットBIN値）の範囲を超えた場合、変換した値に誤差が生じます。変換結果は、整数値の上位ビットから25ビット目を0捨1入し、26ビット目以降を切り捨てた値になります。

## (1) 入力形式

FTB S -> R

S : フローティングデータ格納レジスタまたはフローティング定数

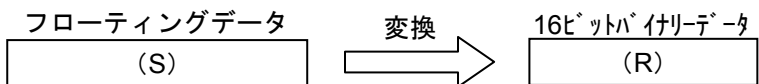
R : 変換したバイナリーデータを格納するレジスタ

(\* ) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

### ● フローティングデータのバイナリーワードデータ変換

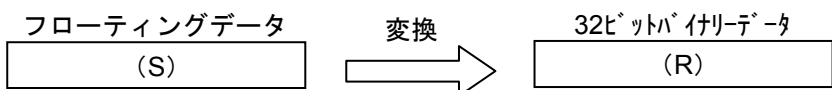
ソース (S) で指定されたフローティングデータを16ビットのバイナリーデータに変換し、結果をリザルト (R) に格納します。



- ・ ソース (S) に指定できる値の範囲は、-32768~32767です。
- ・ 変換後のデータは、フローティングデータの小数点以下1桁目を四捨五入した値になります。

### ● フローティングデータのバイナリーロングワードデータ変換

ソース (S) で指定されたフローティングデータを32ビットのバイナリーデータに変換し、結果をリザルト (R) に格納します。



- ・ ソース (S) に指定できる値の範囲は、-2147483648~2147483647です。
- ・ 変換後のデータは、フローティングデータの小数点以下1桁目を四捨五入した値になります。

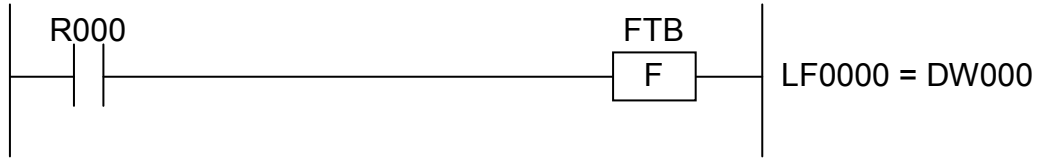
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	—	—	—	—	○	○	○
R	○	—	○	—	—	—	○

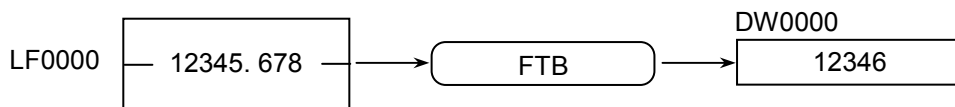
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がONのとき、LF0000の内容をバイナリーデータに変換し、DW0000に格納します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	↕

V：ワードデータ時

- ・  $-32768 \leq \text{リザルト (R)} \leq 32767$  のとき0、それ以外るとき1

ロングワードデータ時

- ・  $-2147483648 \leq \text{リザルト (R)} \leq 2147483647$  のとき0、それ以外るとき1

V以外のフラグはすべて保持します。

● オーバーフロー発生時、リザルト (R) には下記のフルスケール値が格納されます。

	正のオーバーフロー時	負のオーバーフロー時
ワード	32767	-32768
ロングワード	2147483647	-2147483648

# BTD BINARY→BCD変換

## (1) 入力形式

BTD S -> R

B：バイナリーデータ格納レジスタまたはバイナリー定数

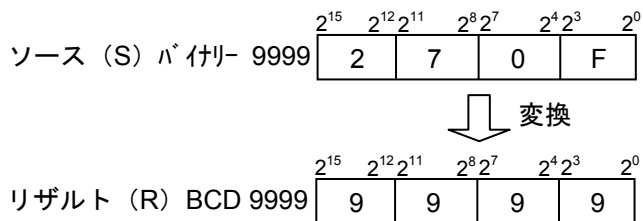
R：変換したBCDデータを格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

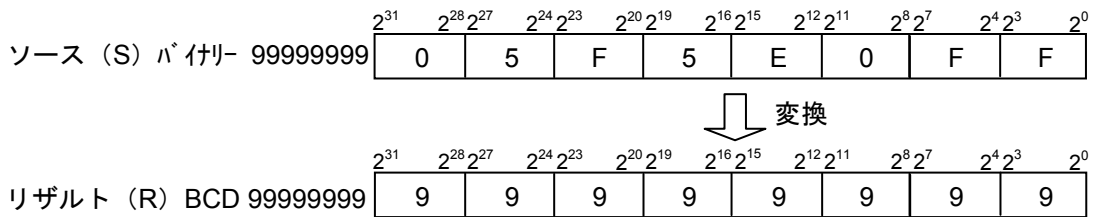
### ● ワードデータのBCD変換

ソース (S) で指定されたバイナリーデータ (0~9999) をBCDに変換し、結果をリザルト (R) に格納します。BCD：2進化10進数 (Binary Coded Decimal)



### ● ロングワードデータのBCD変換

ソース (S) で指定されたバイナリーデータ (0~99999999) をBCDに変換し、結果をリザルト (R) に格納します。



## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	—	—	○
R	○	—	○	—	—	—	○

○：指定できます。

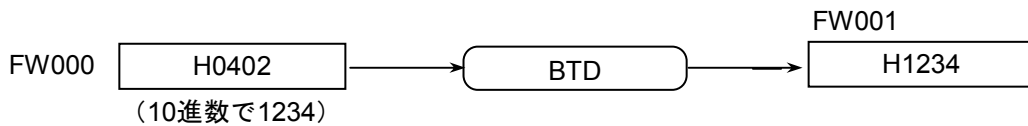
—：指定できません。

SとRの型 (ワード/ロングワード) は合わせてください。異なった型は指定できません (入力エラーになります)。

(4) プログラム例



入力条件R000がONのとき、FW000の内容をバイナリーデータからBCDに変換し、結果をFW001に格納します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	↕

E：ソース (S) <0のとき1、それ以外のとき0

V：ワードデータ時

・ソース (S) >9999のとき1、それ以外のとき0

ロングワードデータ時

・ソース (S) >99999999のとき1、それ以外のとき0

E, V以外のフラグはすべて保持します。

- ソース (S) <0のとき、演算結果フラグのEフラグがON (VフラグはOFF) し、無処理となりリザルト (R) (演算結果) の値は変化しません。
- オーバーフロー発生 (VフラグON) 時、リザルト (R) には下記のフルスケール値が格納されま  
す。

ワード	ロングワード
H9999	H99999999



## (1) 入力形式

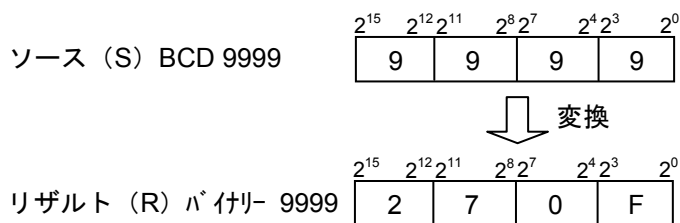
DTB S -> R

S : BCDデータ格納レジスターまたはBCD定数  
 R : 変換したバイナリーデータを格納するレジスター

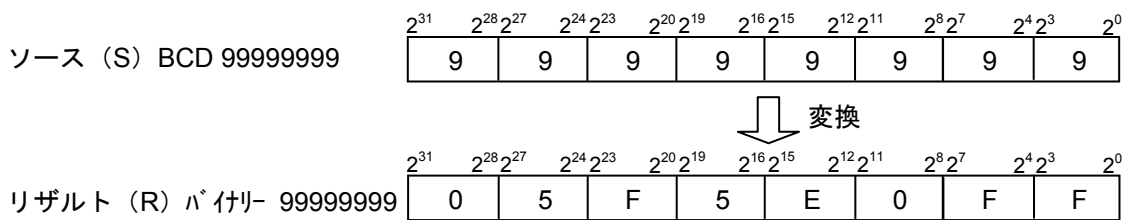
(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

- ワードBCDデータのバイナリー変換  
 ソース (S) で指定されたBCDデータ (0~9999) をバイナリーデータに変換し、結果をリザルト (R) に格納します。



- ロングワードBCDデータのバイナリー変換  
 ソース (S) で指定されたBCDデータ (0~99999999) をバイナリーデータに変換し、結果をリザルト (R) に格納します。



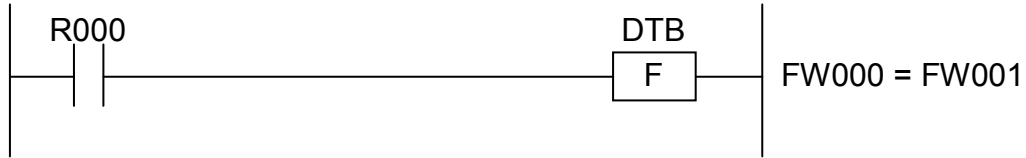
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
R	○	—	○	—	—	—	○

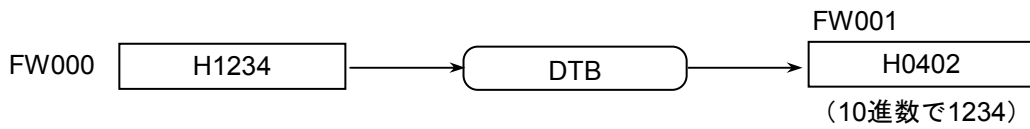
○ : 指定できます。  
 — : 指定できません。

SとRの型 (ワード/ロングワード) は合わせてください。異なった型は指定できません (入力エラーになります)。

(4) プログラム例



入力条件R000がONのとき、FW000の内容をBCDからバイナリーデータに変換し、結果をFW001に格納します。



(5) エラー処理

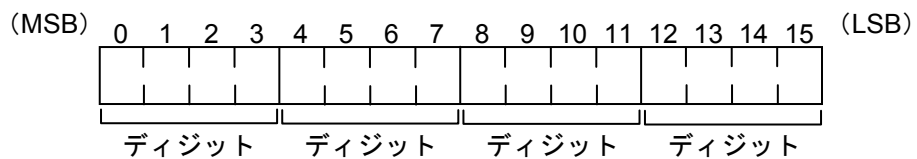
● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	—

E：ソース (S) で指定されたデータ内のディジット (4ビット) (\*) が0~9のとき0、上記以外  
のとき1

E以外のフラグはすべて保持します。

(\*) ディジット



# SEG BINARY→セグメント変換

## (1) 入力形式

SEG S -> R

S : バイナリーデータ格納レジスターまたはバイナリー定数

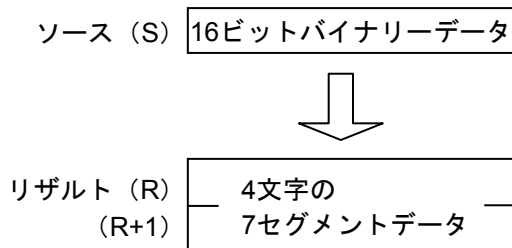
R : 変換した7セグメントデータを格納するレジスター

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

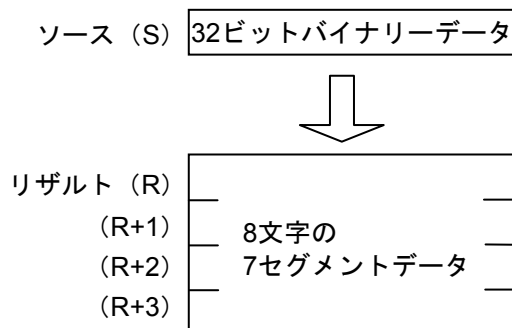
### ● ワードデータのセグメント変換

ソース (S) で指定された16ビットバイナリーデータを7セグメントデータに変換し、結果をリザルト (R) に格納します。

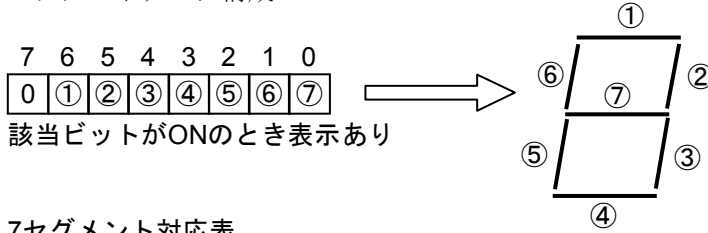


### ● ロングワードデータのセグメント変換

ソース (S) で指定された32ビットバイナリーデータを7セグメントデータに変換し、結果をリザルト (R) に格納します。



<セグメントデータ構成>



7セグメント対応表

No.	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
表示	0	1	2	3	4	5	6	7	8	9	A	b	c	d	E	F
データ	H7E	H30	H6D	H79	H33	H5B	H5F	H70	H7F	H7B	H77	H1F	H4E	H3D	H4F	H47

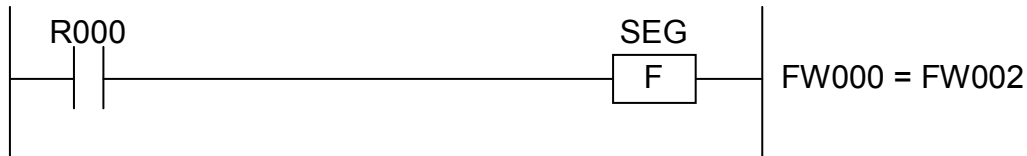
(3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
R	○	—	○	—	—	—	○

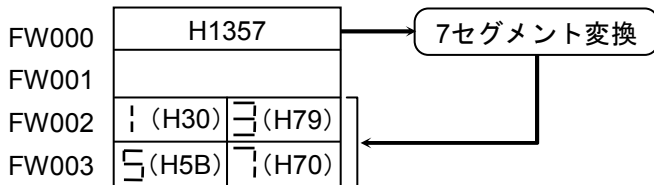
○：指定できます。  
—：指定できません。

SとRの型（ワード/ロングワード）は合わせてください。異なった型は指定できません（入力エラーになります）。

(4) プログラム例



入力条件R000がONのとき、FW000の内容をバイナリーデータから4文字の7セグメントデータに変換し、FW002に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# ASP BINARY→ASCII変換パックモード

## (1) 入力形式

ASP S -> R

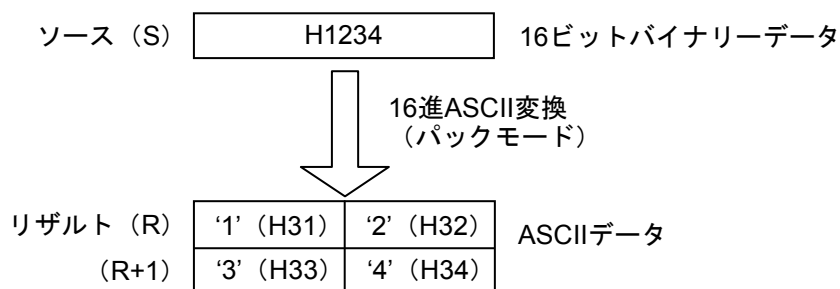
S : バイナリーデータ格納レジスターまたはバイナリー定数

R : 変換したASCIIデータを格納するレジスター

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) で指定された16ビットバイナリーデータをパックモードの16進ASCIIデータに変換し、結果をリザルト (R) に格納します。



<ASCII対応表>

バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ASCII	H30	H31	H32	H33	H34	H35	H36	H37	H38	H39	H41	H42	H43	H44	H45	H46

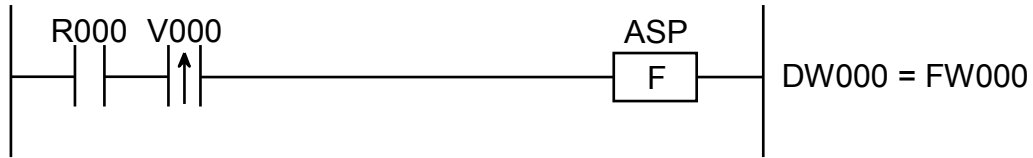
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	○
R	○	—	—	—	—	—	○

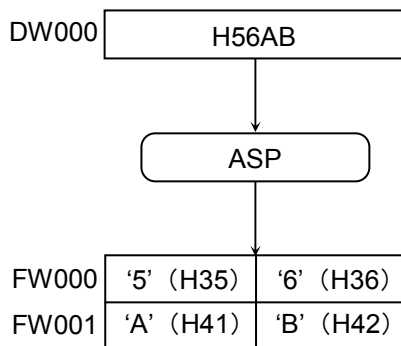
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がOFF→ONに変化時、1回のみDW000の内容をバイナリーデータからパックモードの16進ASCIIデータに変換してFW000に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
-	-	-	-	-	-

フラグはすべて保持します。

# ASU BINARY→ASCII変換アンパックモード

## (1) 入力形式

ASU S -> R

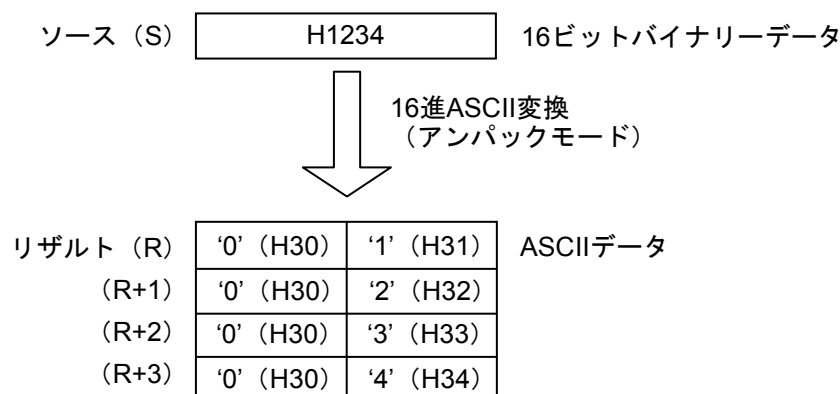
S : バイナリーデータ格納レジスターまたはバイナリー定数

R : 変換したASCIIデータを格納するレジスター

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) で指定された16ビットバイナリーデータをアンパックモードの16進ASCIIデータに変換し、結果をリザルト (R) に格納します。



結果は1バイトずつ上位データから (R) , (R+1) , (R+2) , (R+3) の下位バイトに格納します。  
(R) ~ (R+3) の上位バイトには '0' (H30) が格納されます。

<ASCII対応表>

バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ASCII	H30	H31	H32	H33	H34	H35	H36	H37	H38	H39	H41	H42	H43	H44	H45	H46

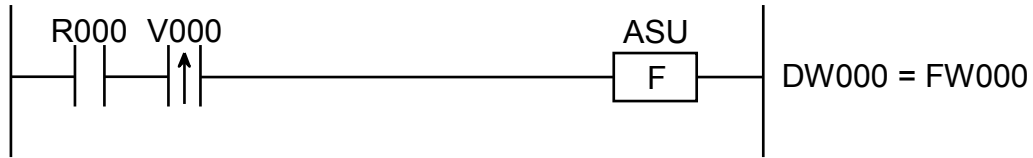
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	○
R	○	—	—	—	—	—	○

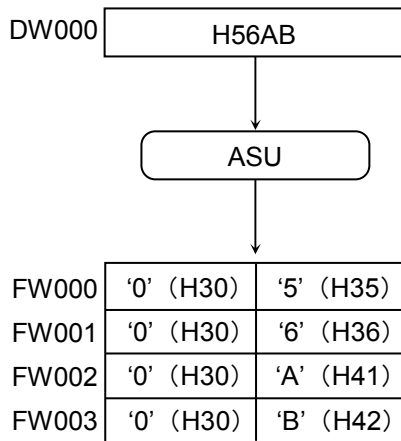
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がOFF→ONに変化時、1回のみDW000の内容をバイナリーデータからアンパックモードの16進ASCIIデータに変換してFW000に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。



# APB ASCII→BINARY変換パックモード

## (1) 入力形式

APB S -> R

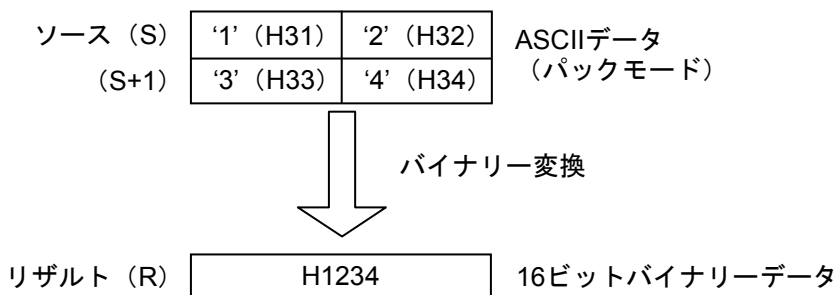
S : ASCIIデータ格納レジスター

R : 変換したバイナリーデータを格納するレジスター

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) で指定されたパックモードの16進ASCIIデータを16ビットバイナリーデータに変換し、結果をリザルト (R) に格納します。



<ASCII対応表>

バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ASCII	H30	H31	H32	H33	H34	H35	H36	H37	H38	H39	H41	H42	H43	H44	H45	H46

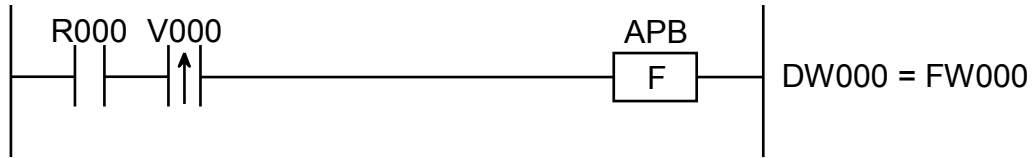
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	—	—	—	—	—	○
R	○	—	—	—	—	—	○

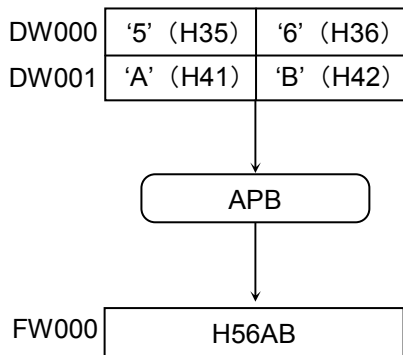
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がOFF→ONに変化時、1回のみDW000の内容をパックモードの16進ASCIIデータからバイナリーデータに変換してFW000に格納します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	—

E：ソース (S) が16進ASCII (H30～H39, H41～H46) 以外のデータ検出時1、それ以外するとき0  
E以外のフラグはすべて保持します。

● EフラグがONしたとき、リザルト (R) は変化しません。

# AUB ASCII→BINARY変換アンパックモード

## (1) 入力形式

```
AUB S -> R
```

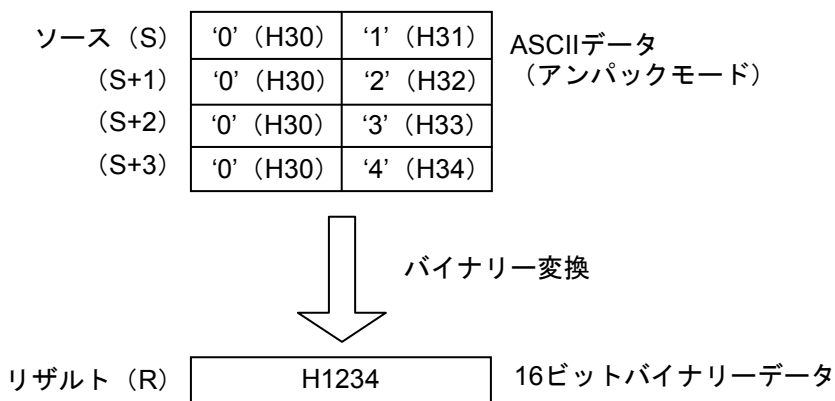
S : ASCIIデータ格納レジスター

R : 変換したバイナリーデータを格納するレジスター

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) で指定されたアンパックモードの16進ASCIIデータを16ビットバイナリーデータに変換し、結果をリザルト (R) に格納します。



ソース (S) から (S+3) の上位バイトは任意です。

### <ASCII対応表>

バイナリー	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ASCII	H30	H31	H32	H33	H34	H35	H36	H37	H38	H39	H41	H42	H43	H44	H45	H46

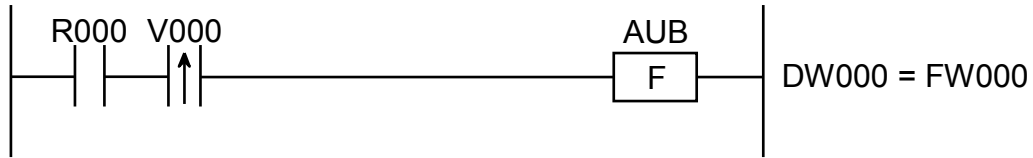
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	—	—	—	—	—	○
R	○	—	—	—	—	—	○

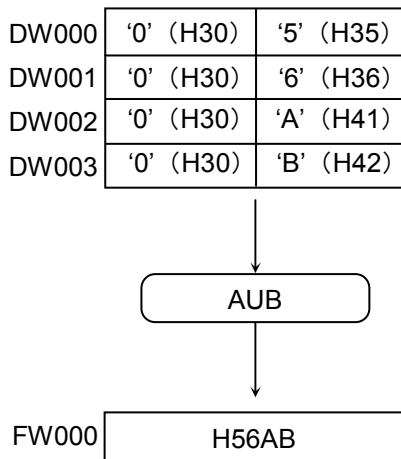
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がOFF→ONに変化時、1回のみDW000の内容をバイナリーデータからアンパックモードの16進ASCIIデータに変換してFW000に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	—

E：ソース (S) が16進ASCII (H30～H39, H41～H46) 以外のデータ検出時1、それ以外するとき0  
E以外のフラグはすべて保持します。

- EフラグがONしたとき、リザルト (R) は変化しません。

# STD SINGLE→DOUBLE変換

## (1) 入力形式

STD S -> R

S : 16ビットバイナリーデータ格納レジスターまたは16ビットバイナリー定数

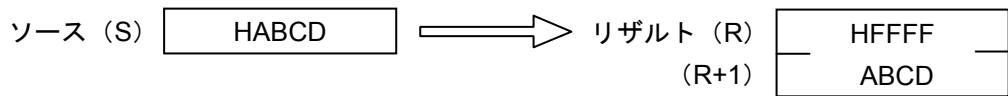
R : 変換した32ビットバイナリーデータを格納するレジスター

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

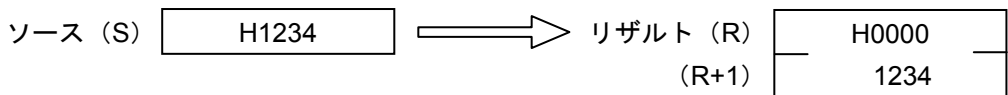
## (2) 機能

ソース (S) で指定された16ビットバイナリーデータを32ビットバイナリーデータに符号拡張し、結果をリザルト (R) に格納します。

### ● 符号ビットONの場合



### ● 符号ビットOFFの場合



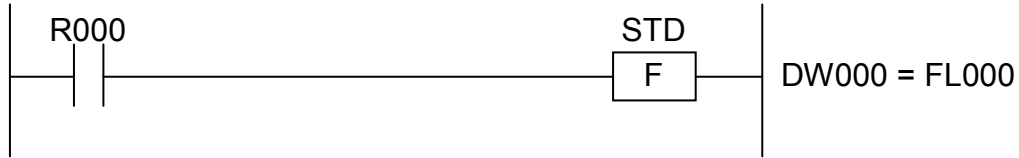
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	○
R	—	—	○	—	—	—	○

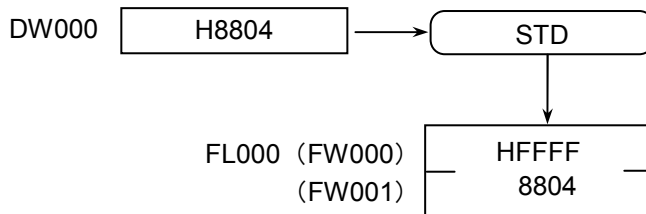
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がONのとき、DW000の内容を16ビットバイナリーデータから32ビットバイナリーデータに符号拡張し、結果をFL000に格納します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# DTS DOUBLE→SINGLE変換

## (1) 入力形式

DTS S -> R

S : 32ビットバイナリーデータ格納レジスターまたは32ビットバイナリー定数

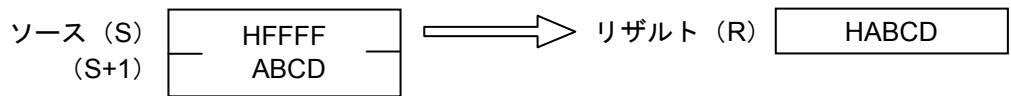
R : 変換した16ビットバイナリーデータを格納するレジスター

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

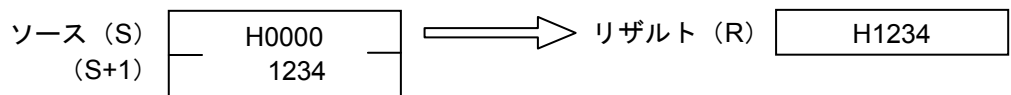
## (2) 機能

ソース (S) で指定された32ビットバイナリーデータを16ビットバイナリーデータに変換し、結果をリザルト (R) に格納します。

### ● 符号ビットONの場合



### ● 符号ビットOFFの場合



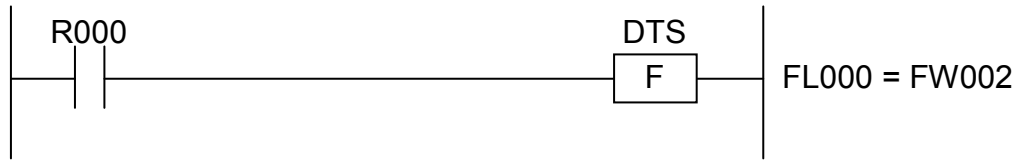
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	—	—	○	○	—	—	○
R	○	—	—	—	—	—	○

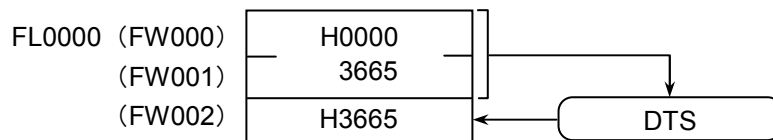
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がONのとき、FL000の内容を32ビットバイナリーデータから16ビットバイナリーデータに変換して、結果をFW002に格納します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	↕

V：ソース (S) が-32768～32767のとき0、それ以外のとき1  
V以外のフラグはすべて保持します。

- オーバーフロー発生 (VフラグON) 時、リザルト (R) には下記のフルスケール値が格納されます。

ソース (S) >32767のとき	H7FFF
ソース (S) <-32767のとき	H8000



# ABS 絶対値 : ABSOLUTE VALUE

## (1) 入力形式

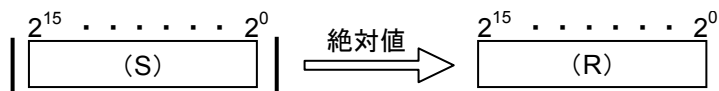
ABS S -> R

S : ソース格納レジスタまたは定数  
R : 絶対値を格納するレジスタ

(\* ) シンボル (命令語名称) とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

- ワードデータの絶対値  
ソース (S) で指定された16ビットデータの絶対値をリザルト (R) に格納します。



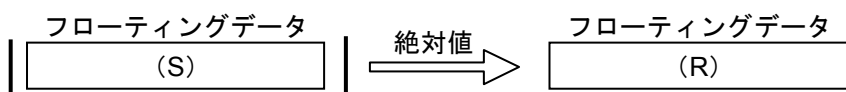
ソース (S) , リザルト (R) に指定できる値および格納される値の範囲は、-32768~32767です。

- ロングワードデータの絶対値  
ソース (S) で指定された32ビットデータの絶対値をリザルト (R) に格納します。



ソース (S) , リザルト (R) に指定できる値および格納される値の範囲は、-2147483648~2147483647です。

- フローティングデータの絶対値  
ソース (S) で指定されたフローティングデータの絶対値をリザルト (R) に格納します。



ソース (S) , リザルト (R) に指定できる値および格納される値の範囲は、下記です。  
 $0, \pm 2^{-126} \sim \pm 2^{128}$

## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

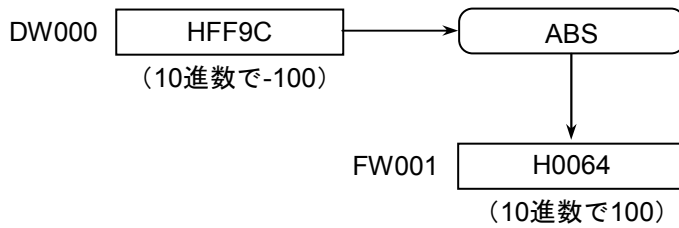
○ : 指定できます。  
— : 指定できません。

SとRの型 (ワード/ロングワード/フローティング) は合わせてください。異なった型は指定できません (入力エラーになります)。

(4) プログラム例



入力条件R000がONのとき、DW000の内容の絶対値をFW001に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
-	-	-	-	-	↕

V : ワードデータ時

- ・ソース (S) = -32768のとき1、それ以外るとき0

ロングワードデータ時

- ・ソース (S) = -2147483648のとき1、それ以外るとき0

フローティングデータ時

- ・演算結果によりVは変化しません (保持します)。

V以外のフラグはすべて保持します。

- オーバーフロー発生 (VフラグON) 時、リザルト (R) には下記のフルスケール値が格納されま  
す。

ワード	ロングワード
H7FFF	H7FFFFFFF

- フローティング演算時、非数、無限大をソース (S) に指定した場合は、リザルト (R) に下記の  
値が格納されます。この場合、EフラグはONしません。

ソース (S)	リザルト (R)
非数	非数
+無限大	+無限大
-無限大	+無限大

# NEG 符号変換 : SIGN CHANGE

## (1) 入力形式

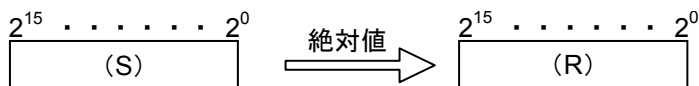
NEG S -> R

S : 符号変換するデータの格納レジスタまたは定数  
 R : 変換したデータを格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

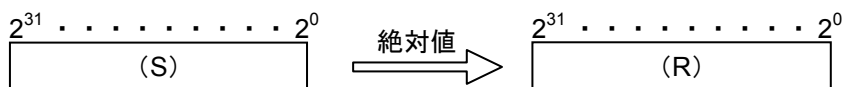
## (2) 機能

- ワードデータの符号変換  
 ソース (S) で指定された16ビットデータを符号変換し、リザルト (R) に格納します。



ソース (S) , リザルト (R) に指定できる値および格納される値の範囲は、-32768~32767です。

- ロングワードデータの符号変換  
 ソース (S) で指定された32ビットデータを符号変換し、リザルト (R) に格納します。



ソース (S) , リザルト (R) に指定できる値および格納される値の範囲は、-2147483648~2147483647です。

- フローティングデータの符号変換  
 ソース (S) で指定されたフローティングデータを符号変換し、リザルト (R) に格納します。



ソース (S) , リザルト (R) に指定できる値および格納される値の範囲は、下記です。  
 $0, \pm 2^{-126} \sim \pm 2^{128}$

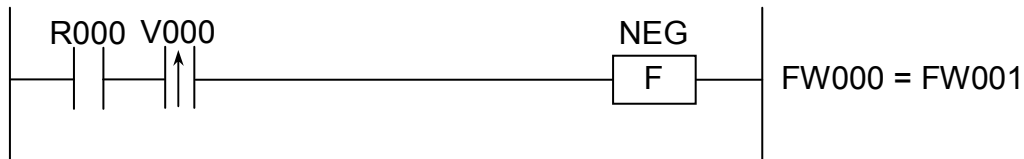
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

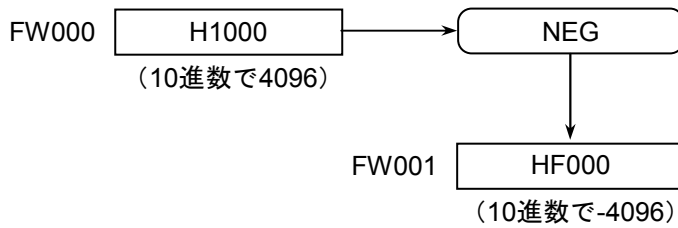
○ : 指定できます。  
 — : 指定できません。

SとRの型 (ワード/ロングワード/フローティング) は合わせてください。異なった型は指定できません (入力エラーになります)。

(4) プログラム例



入力条件R000がOFF→ONに変化時、1回のみFW000の内容を符号変換してFW001に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
-	-	-	-	-	↑↓

V : ワードデータ時

- ・ ソース (S) = -32768 のとき1、それ以外 のとき0

ロングワードデータ時

- ・ ソース (S) = -2147483648 のとき1、それ以外 のとき0

フローティングデータ時

- ・ 演算結果によりVは変化しません (保持します)。

V以外のフラグはすべて保持します。

- オーバーフロー発生 (VフラグON) 時、リザルト (R) には下記のフルスケール値が格納されま  
す。

ワード	ロングワード
H7FFF	H7FFFFFFF

- フローティング演算時、非数、無限大をソース (S) に指定した場合は、リザルト (R) に下記の  
値が格納されます。この場合、EフラグはONしません。

ソース (S)	リザルト (R)
非数	非数
+無限大	-無限大
-無限大	+無限大

# DCD デコード : DECODE

## (1) 入力形式

```
DCD S -> R
```

S : デコードするデータの格納レジスターまたは定数

R : デコードした結果を格納するレジスター

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

### ● ワードデータのデコード

- ・ソース (S) で指定されたデータの下位4ビットをデコードし、その結果を指定されたリザルト (R) 内のビットに格納します。
- ・ソース (S) で指定されたデータの下位4ビットが有効になります。
- ・ソース (S) には、0~15の定数が指定できます。

### ● ロングワードデータのデコード

- ・ソース (S) で指定されたデータの下位5ビットをデコードし、その結果を指定されたリザルト (R) 内のビットに格納します。
- ・ソース (S) で指定されたデータの下位5ビットが有効になります。
- ・ソース (S) には、0~31の定数が指定できます。

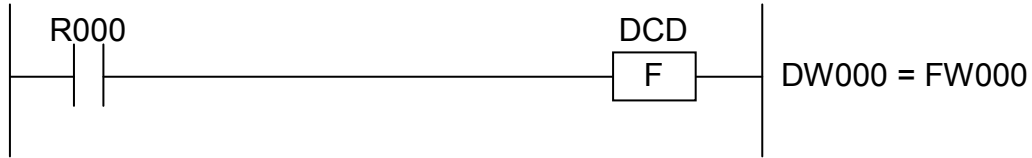
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	○
R	○	—	○	—	—	—	○

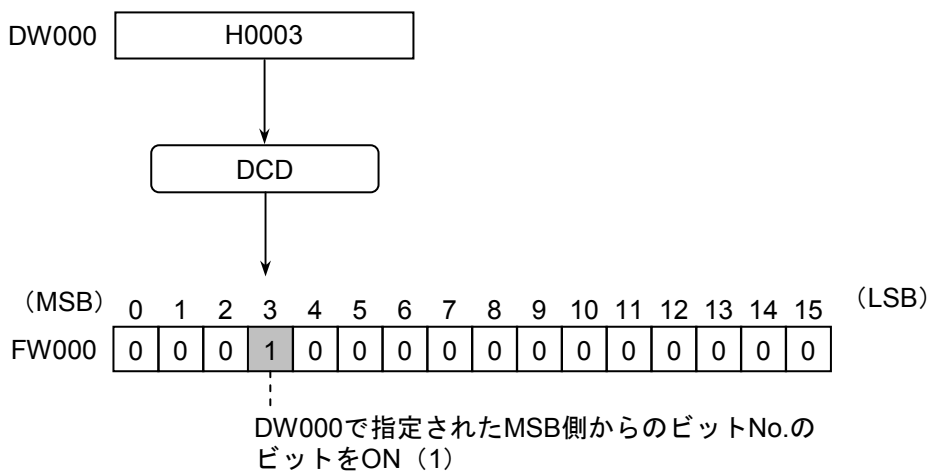
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がONのとき、DW000の内容をデコードして結果をFW000に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# ECD エンコード : ENCODE

## (1) 入力形式

ECD S -> R

S : エンコードするデータの格納レジスタまたは定数

R : エンコードした結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

- ・ソース (S) で指定されたデータをエンコードし、その結果をリザルト (R) に格納します。
- ・ソース (S) = 0 の場合は無処理です (リザルト (R) の内容は変化しません)。

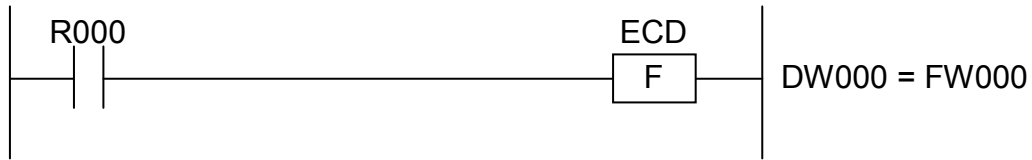
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	—	—	○
R	○	—	—	—	—	—	○

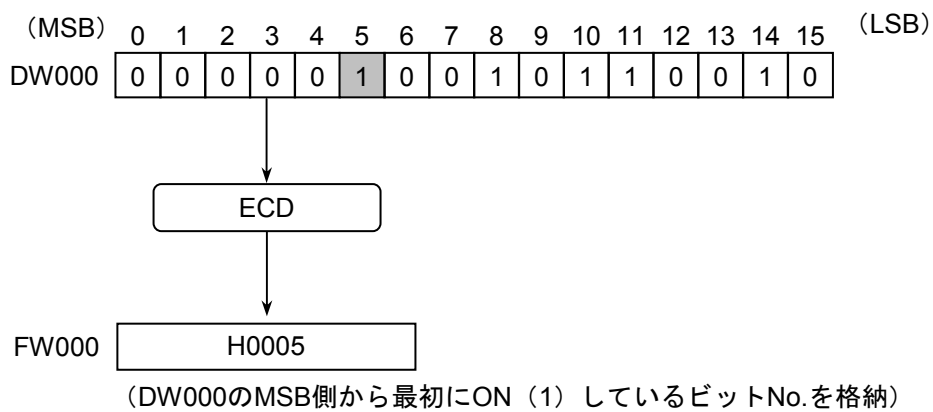
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がONのとき、DW000の内容をエンコードして結果をFW000に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	—

E : ソース (S) = 0のとき1、それ以外るとき0

E以外のフラグはすべて保持します。



# LSR 論理右シフト : LOGICAL SHIFT RIGHT

## (1) 入力形式

```
LSR S : D -> R
```

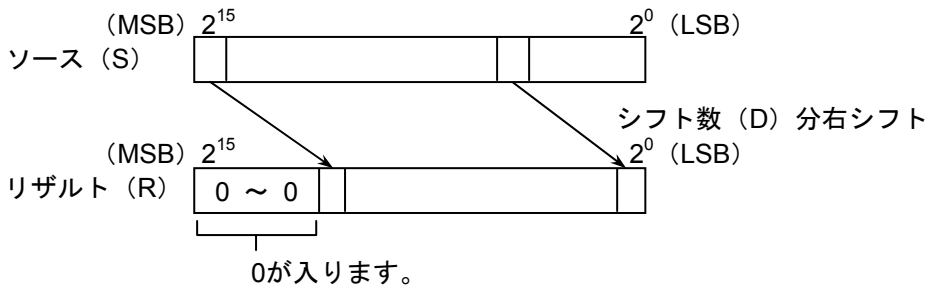
S : シフトするデータを格納したレジスタまたは定数  
D : シフトするビット数を格納したレジスタまたは定数  
R : シフトした結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

### ● ワードデータの右シフト

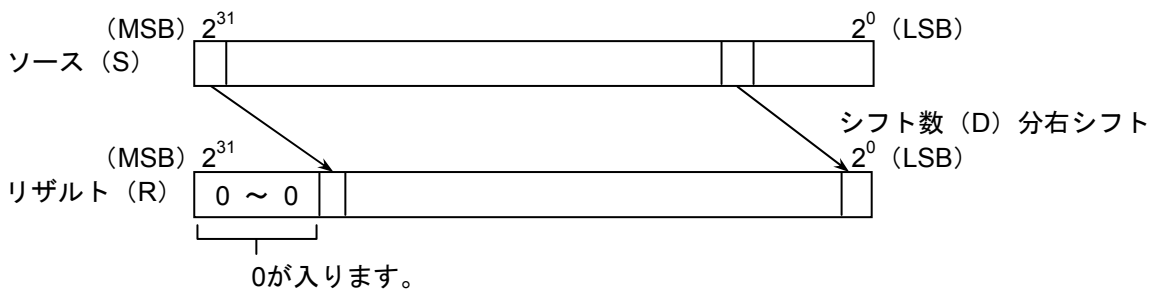
ソース (S) で指定された16ビットデータを、デスティネーション (D) で指定されたシフト数分右にシフトし、結果をリザルト (R) に格納します。



- ・シフト数は、デスティネーション (D) で指定されたデータの低位4ビットが有効になります。
- ・デスティネーション (D) には、0~15の定数が指定できます。

### ● ロングワードデータの右シフト

ソース (S) で指定された32ビットデータを、デスティネーション (D) で指定されたシフト数分右にシフトし、結果をリザルト (R) に格納します。



- ・シフト数は、デスティネーション (D) で指定されたデータの低位5ビットが有効になります。
- ・デスティネーション (D) には、0~31の定数が指定できます。

(3) データタイプ

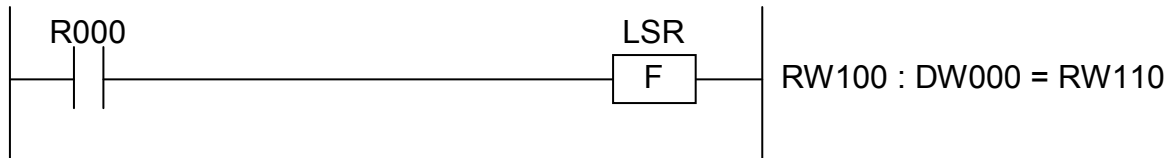
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
D	○	○	—	—	—	—	○
R	○	—	○	—	—	—	○

○ : 指定できます。

— : 指定できません。

SとRの型(ワード/ロングワード)は合わせてください。異なった型は指定できません(入力エラーになります)。Dはワード型固定です。

(4) プログラム例



入力条件R000がONのとき、RW100の内容をDW000で指定されたシフト数分右へシフトし、結果をRW110に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# LSL 論理左シフト : LOGICAL SHIFT LEFT

## (1) 入力形式

```
LSL S : D -> R
```

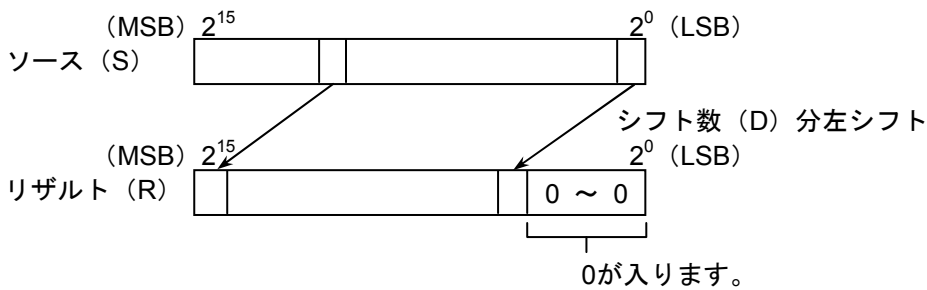
S : シフトするデータを格納したレジスタまたは定数  
D : シフトするビット数を格納したレジスタまたは定数  
R : シフトした結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

### ● ワードデータの左シフト

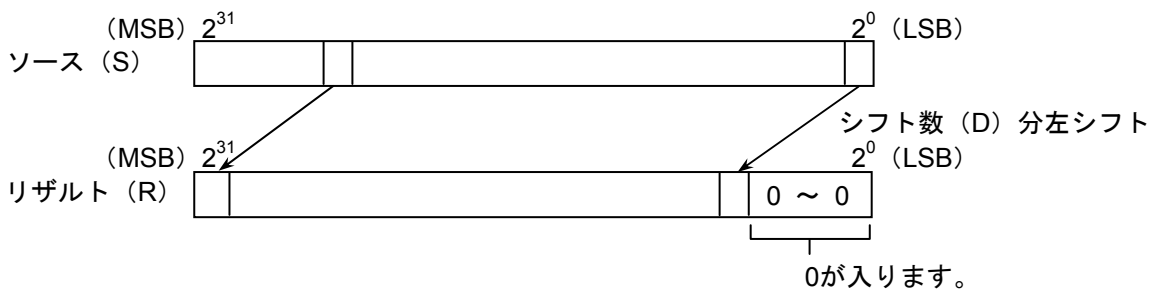
ソース (S) で指定された16ビットデータを、デスティネーション (D) で指定されたシフト数分左にシフトし、結果をリザルト (R) に格納します。



- ・シフト数は、デスティネーション (D) で指定されたデータの低位4ビットが有効になります。
- ・デスティネーション (D) には、0~15の定数が指定できます。

### ● ロングワードデータの左シフト

ソース (S) で指定された32ビットデータを、デスティネーション (D) で指定されたシフト数分左にシフトし、結果をリザルト (R) に格納します。



- ・シフト数は、デスティネーション (D) で指定されたデータの低位5ビットが有効になります。
- ・デスティネーション (D) には、0~31の定数が指定できます。

(3) データタイプ

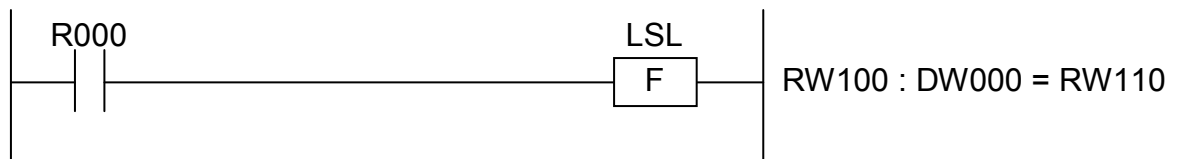
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
D	○	○	—	—	—	—	○
R	○	—	○	—	—	—	○

○ : 指定できます。

— : 指定できません。

SとRの型 (ワード/ロングワード) は合わせてください。異なった型は指定できません (入力エラーになります)。Dはワード型固定です。

(4) プログラム例



入力条件R000がONのとき、RW100の内容をDW000で指定されたシフト数分左へシフトし、結果をRW110に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# ASR 算術右シフト : ARITHMETIC SHIFT RIGHT

## (1) 入力形式

ASR S : D -> R

S : シフトするデータを格納したレジスタまたは定数

D : シフトするビット数を格納したレジスタまたは定数

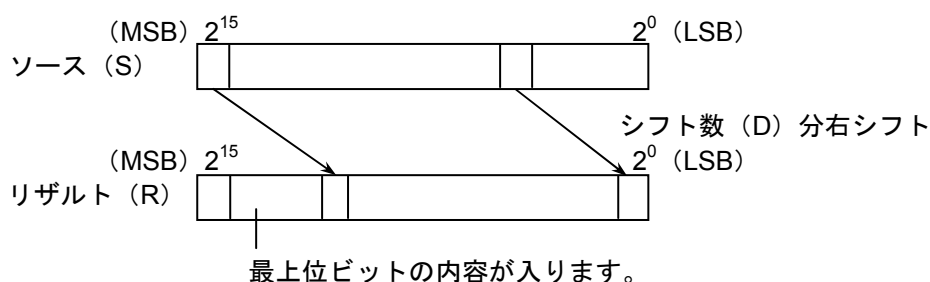
R : シフトした結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

### ● ワードデータの右シフト

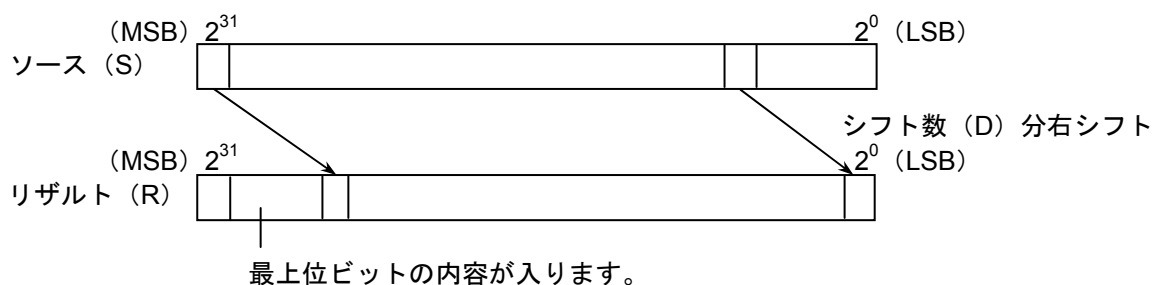
ソース (S) で指定された16ビットデータを、デスティネーション (D) で指定されたシフト数分右にシフト (符号ビット保持) し、結果をリザルト (R) に格納します。



- ・シフト数は、デスティネーション (D) で指定されたデータの低位4ビットが有効になります。
- ・デスティネーション (D) には、0~15の定数が指定できます。

### ● ロングワードデータの右シフト

ソース (S) で指定された32ビットデータを、デスティネーション (D) で指定されたシフト数分右にシフト (符号ビット保持) し、結果をリザルト (R) に格納します。



- ・シフト数は、デスティネーション (D) で指定されたデータの低位5ビットが有効になります。
- ・デスティネーション (D) には、0~31の定数が指定できます。

(3) データタイプ

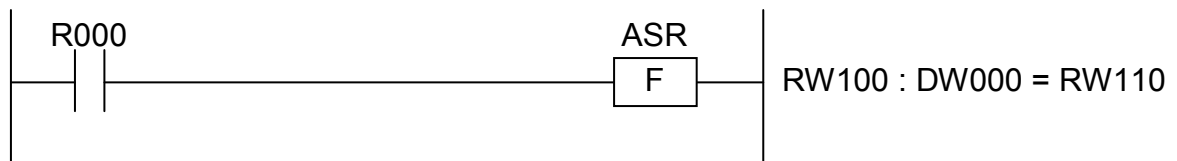
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
D	○	○	—	—	—	—	○
R	○	—	○	—	—	—	○

○ : 指定できます。

— : 指定できません。

SとRの型（ワード/ロングワード）は合わせてください。異なった型は指定できません（入力エラーになります）。Dはワード型固定です。

(4) プログラム例



入力条件R000がONのとき、RW100の内容をDW000で指定されたシフト数分右へシフトし、結果をRW110に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# ASL 算術左シフト : ARITHMETIC SHIFT LEFT

## (1) 入力形式

ASL S : D -> R

S : シフトするデータを格納したレジスタまたは定数

D : シフトするビット数を格納したレジスタまたは定数

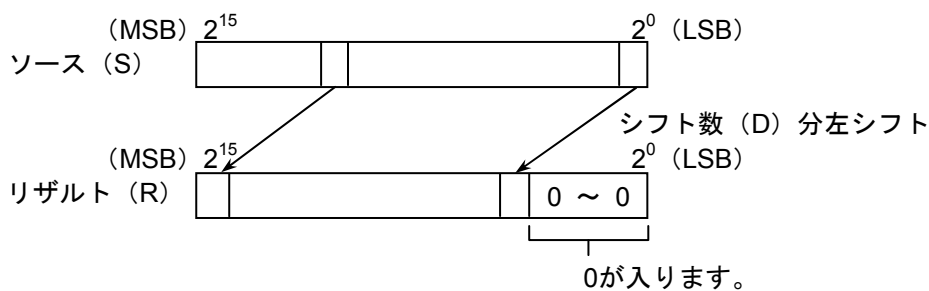
R : シフトした結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

### ● ワードデータの左シフト

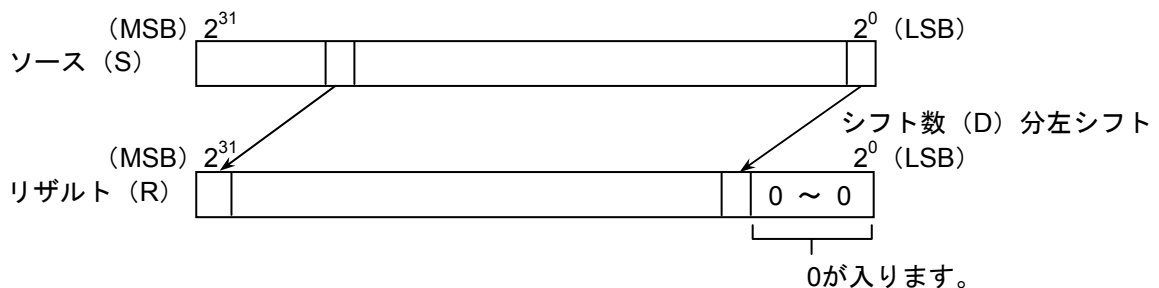
ソース (S) で指定された16ビットデータを、デスティネーション (D) で指定されたシフト数分左にシフトし、結果をリザルト (R) に格納します。



- ・シフト数は、デスティネーション (D) で指定されたデータの低位4ビットが有効になります。
- ・デスティネーション (D) には、0~15の定数が指定できます。

### ● ロングワードデータの左シフト

ソース (S) で指定された32ビットデータを、デスティネーション (D) で指定されたシフト数分左にシフトし、結果をリザルト (R) に格納します。



- ・シフト数は、デスティネーション (D) で指定されたデータの低位5ビットが有効になります。
- ・デスティネーション (D) には、0~31の定数が指定できます。

(3) データタイプ

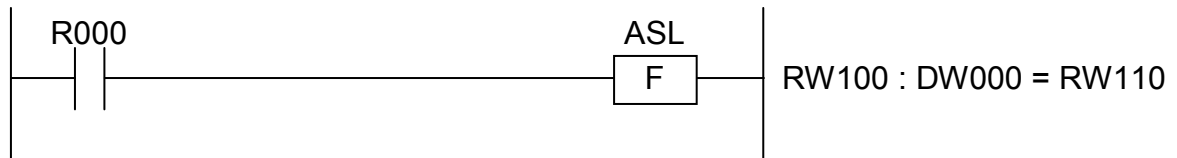
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
D	○	○	—	—	—	—	○
R	○	—	○	—	—	—	○

○ : 指定できます。

— : 指定できません。

SとRの型 (ワード/ロングワード) は合わせてください。異なった型は指定できません (入力エラーになります)。Dはワード型固定です。

(4) プログラム例



入力条件R000がONのとき、RW100の内容をDW000で指定されたシフト数分左へシフトし、結果をRW110に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	↕

V : シフト操作中に符号ビットが1回でも変化すれば1、それ以外るとき0  
V以外のフラグはすべて保持します。

- オーバーフロー発生 (VフラグON) 時、リザルト (R) に下記のフルスケール値が格納されま  
す。

	ワード	ロングワード
(S) >0のとき	H7FFF	H7FFFFFFF
(S) <0のとき	H8000	H80000000



# ROR 右回転 : ROTATE RIGHT

## (1) 入力形式

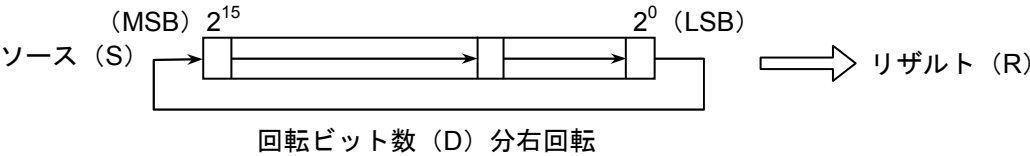
```
ROR S : D -> R
```

S : 回転するデータを格納したレジスタまたは定数  
D : 回転するビット数を格納したレジスタまたは定数  
R : 回転した結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

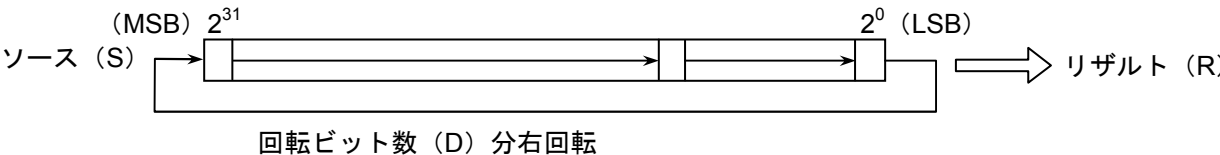
## (2) 機能

- ワードデータの右回転  
ソース (S) で指定された16ビットデータを、デスティネーション (D) で指定された回転ビット数分右に回転し、結果をリザルト (R) に格納します。



- ・ 回転ビット数は、デスティネーション (D) で指定されたデータの下位4ビットが有効になります。
- ・ デスティネーション (D) には、0~15の定数が指定できます。

- ロングワードデータの右回転  
ソース (S) で指定された32ビットデータを、デスティネーション (D) で指定された回転ビット数分右に回転し、結果をリザルト (R) に格納します。



- ・ 回転ビット数は、デスティネーション (D) で指定されたデータの下位5ビットが有効になります。
- ・ デスティネーション (D) には、0~31の定数が指定できます。

(3) データタイプ

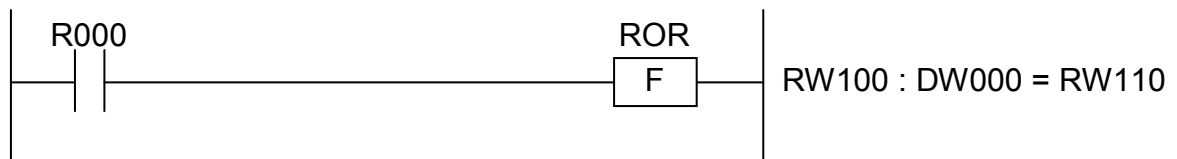
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
D	○	○	—	—	—	—	○
R	○	—	○	—	—	—	○

○ : 指定できます。

— : 指定できません。

SとRの型（ワード/ロングワード）は合わせてください。異なった型は指定できません（入力エラーになります）。Dはワード型固定です。

(4) プログラム例



入力条件R000がONのとき、RW100の内容をDW000で指定された回転ビット数分右へ回転し、結果をRW110に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# ROL 左回転 : ROTATE LEFT

## (1) 入力形式

```
ROL S : D -> R
```

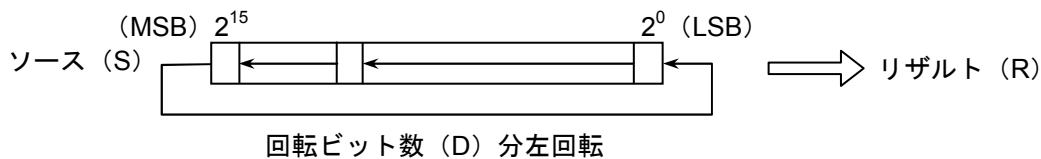
- S : 回転するデータを格納したレジスタまたは定数
- D : 回転するビット数を格納したレジスタまたは定数
- R : 回転した結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

### ● ワードデータの左回転

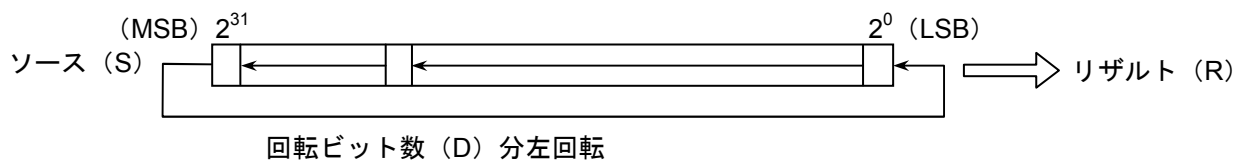
ソース (S) で指定された16ビットデータを、デスティネーション (D) で指定された回転ビット数分左に回転し、結果をリザルト (R) に格納します。



- ・回転ビット数は、デスティネーション (D) で指定されたデータの下位4ビットが有効になります。
- ・デスティネーション (D) には、0~15の定数が指定できます。

### ● ロングワードデータの左回転

ソース (S) で指定された32ビットデータを、デスティネーション (D) で指定された回転ビット数分左に回転し、結果をリザルト (R) に格納します。



- ・回転ビット数は、デスティネーション (D) で指定されたデータの下位5ビットが有効になります。
- ・デスティネーション (D) には、0~31の定数が指定できます。

(3) データタイプ

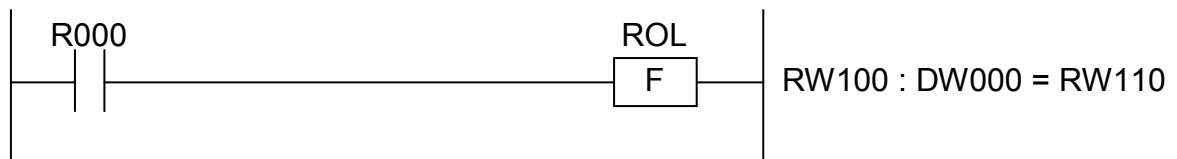
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
D	○	○	—	—	—	—	○
R	○	—	○	—	—	—	○

○ : 指定できます。

— : 指定できません。

SとRの型（ワード/ロングワード）は合わせてください。異なった型は指定できません（入力エラーになります）。Dはワード型固定です。

(4) プログラム例



入力条件R000がONのとき、RW100の内容をDW000で指定されたシフト数分左へ回転し、結果をRW110に格納します。



(5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# LIM リミッター：LIMITER

## (1) 入力形式

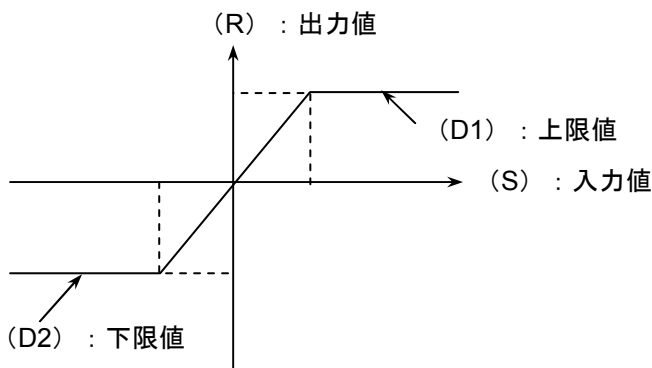
LIM S : D1 : D2 -> R

- S : 入力値を格納したレジスタまたは定数
- D1 : 上限値を格納したレジスタまたは定数
- D2 : 下限値を格納したレジスタまたは定数
- R : リミット制御により制御した出力値を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

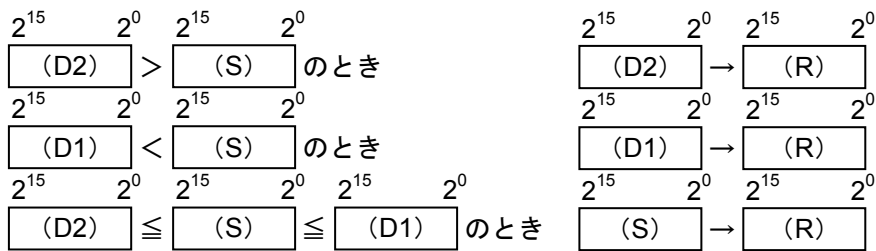
## (2) 機能

ソース (S) で指定した入力値が、デスティネーション (D1) , (D2) で指定した上下限リミット値の範囲内か比較し、リザルト (R) に格納する出力値を制御します。



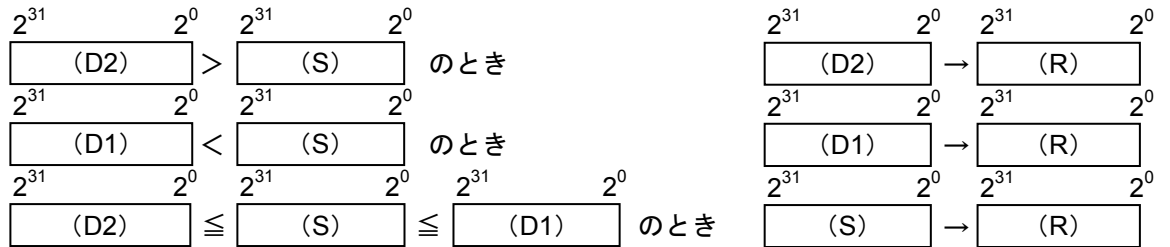
### ● ワードデータのリミッター

- ・下記の16ビットデータをリミッター制御します。



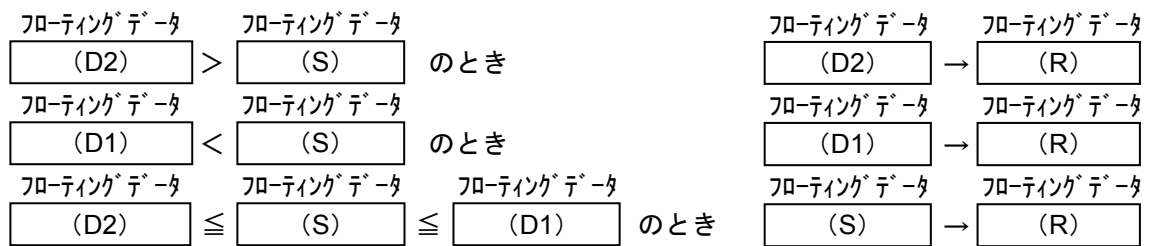
- ・ソース (S) , デスティネーション (D1) , (D2) に指定できる値の範囲は、-32768~32767です。

- ロングワードデータのリミッター
  - ・下記の32ビットデータをリミッター制御します。



- ・ソース (S) , デスティネーション (D1) , (D2) に指定できる値の範囲は、-2147483648～2147483647です。

- フローティングデータのリミッター
  - ・下記のフローティングデータをリミッター制御します。



- ・ソース (S) , デスティネーション (D1) , (D2) に指定できる値の範囲は下記です。  
 $0, \pm 2^{-126} \sim \pm 2^{128}$

(3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D1	○	○	○	○	○	○	○
D2	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

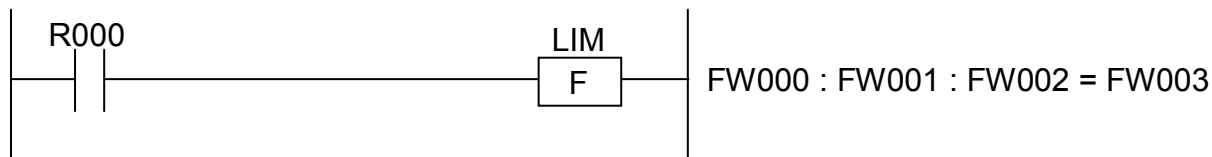
○ : 指定できます。

— : 指定できません。

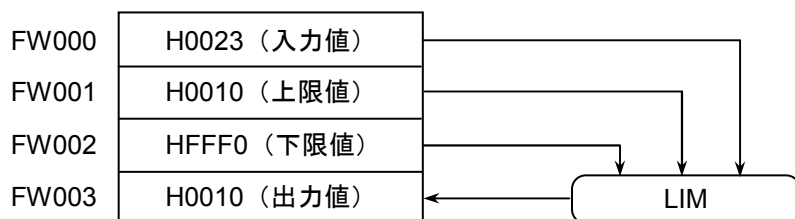
S, D1, D2, Rの型 (ワード/ロングワード/フローティング) はすべて合わせてください。異なった型は指定できません (入力エラーになります) 。

## LIM リミッター : LIMITER

### (4) プログラム例



入力条件R000がONのとき、FW000の内容とFW001およびFW002の内容と比較して、リミット制御の出力値をFW003に格納します。



### (5) エラー処理

#### ● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	—

E : デスティネーション (D1) < (D2) のとき1、それ以外のとき0  
E以外のフラグはすべて保持します。

#### ● EフラグON時は、(D1) または (D2) と比較しません。

このページは白紙です。



# BND デッドバンド : DEAD BAND

## (1) 入力形式

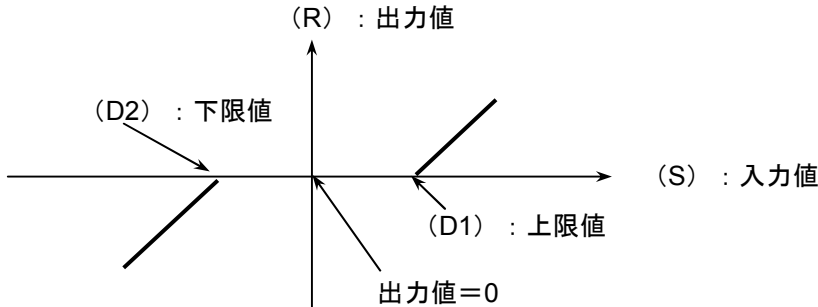
BND S : D1 : D2 -> R

- S : 不感帯の入力値を格納したレジスタまたは定数
- D1 : 不感帯の上限値を格納したレジスタまたは定数
- D2 : 不感帯の下限値を格納したレジスタまたは定数
- R : 不感帯制御により制御した出力値を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

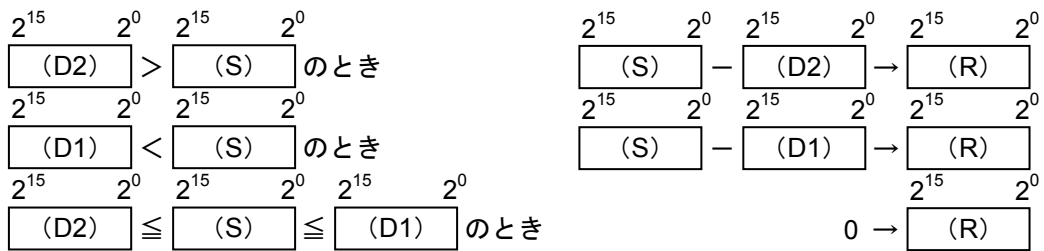
## (2) 機能

ソース (S) で指定した入力値がデスティネーション (D1) , (D2) で指定した不感帯の上下限値の範囲内か比較し、リザルト (R) に格納する出力値を制御します (境界範囲内をデータ0 (不感帯) としてリザルト (R) に格納します) 。



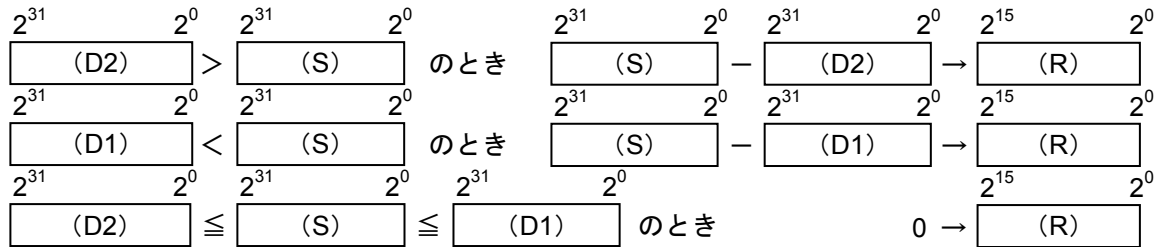
### ● ワードデータの不感帯制御

- ・下記の16ビットデータを不感帯制御します。



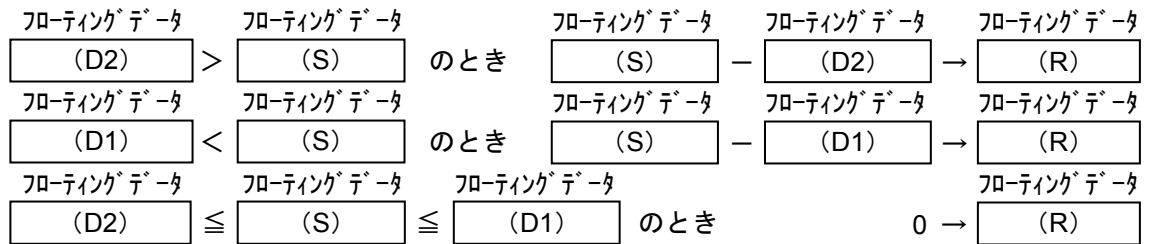
- ・ソース (S) , デスティネーション (D1) , (D2) に指定できる値の範囲は、-32768~32767です。

- ロングワードデータの不感帯制御
  - ・下記の32ビットデータを不感帯制御します。



- ・ソース (S) , デスティネーション (D1) , (D2) に指定できる値の範囲は、-2147483648～2147483647です。

- フローティングデータの不感帯制御
  - ・下記のフローティングデータを不感帯制御します。



- ・ソース (S) , デスティネーション (D1) , (D2) に指定できる値の範囲は下記です。  
 $0, \pm 2^{-126} \sim \pm 2^{128}$

(3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	○	○	○
D1	○	○	○	○	○	○	○
D2	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

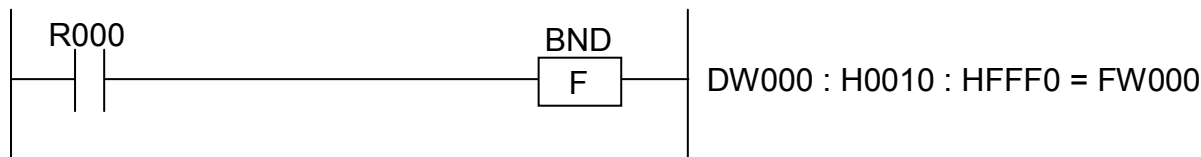
○ : 指定できます。

— : 指定できません。

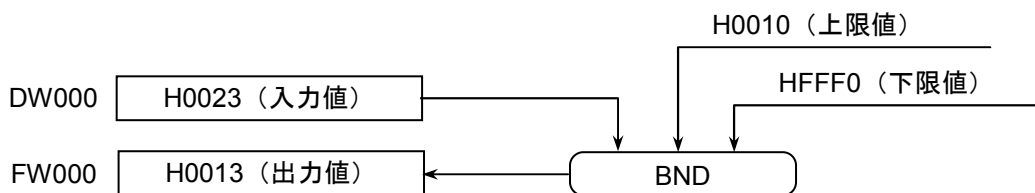
S, D1, D2, Rの型 (ワード/ロングワード/フローティング) はすべて合わせてください。異なった型は指定できません (入力エラーになります)。

## BND デッドバンド : DEAD BAND

### (4) プログラム例



入力条件R000がONのとき、DW000の内容と定数H0010, HFFF0の内容を比較して、不感帯制御の出力値をFW000に格納します。



### (5) エラー処理

#### ● 演算結果フラグ

X	E	P	N	Z	V
-	↕	-	-	-	↕

V : ワードデータ時

- ・リザルト (R) が-32768~32767 のとき0、それ以外るとき1

ロングワードデータ時

- ・リザルト (R) が-2147483648~2147483647のとき0、それ以外るとき1

フローティングデータ時

- ・演算結果によりVは変化しません (保持します)。

E : ワードおよびロングワードデータ時

- ・デスティネーション (D1) < (D2) のとき1
- ・デスティネーション (D1) ≥ (D2) のとき0

フローティングデータ時

- ・リザルト (R) が0ではなく、かつ下記範囲外るとき1、それ以外るとき0  
 $\pm 2^{-126} \sim \pm 2^{128}$

V, E以外のフラグはすべて保持します。

- (D1) < (D2) のとき、エラーフラグ (Eフラグ) がON (オーバーフローフラグ (Vフラグ) はOFF) し、リザルト (R) は変化しません。
- オーバーフロー発生時、リザルト (R) には下記のフルスケール値が格納されます。

	正のオーバーフロー時	負のオーバーフロー時
ワード	H7FFF	H8000
ロングワード	H7FFFFFFF	H80000000
フローティング	+3.402823E38	-3.402823E38

フローティングでオーバーフロー発生時、VフラグはONしません (ワードおよびロングワード時はONします)。

- フローティングでアンダーフロー発生時、リザルト (R) には正しい符号を持った0が格納されま  
す。演算結果フラグは変化しません。

# ZON デッドゾーン : DEAD ZONE

## (1) 入力形式

ZON S : D1 : D2 -> R

S : ゾーン制御するための入力値を格納したレジスタまたは定数

D1 : 入力値に加算する正のバイアス値を格納したレジスタまたは定数

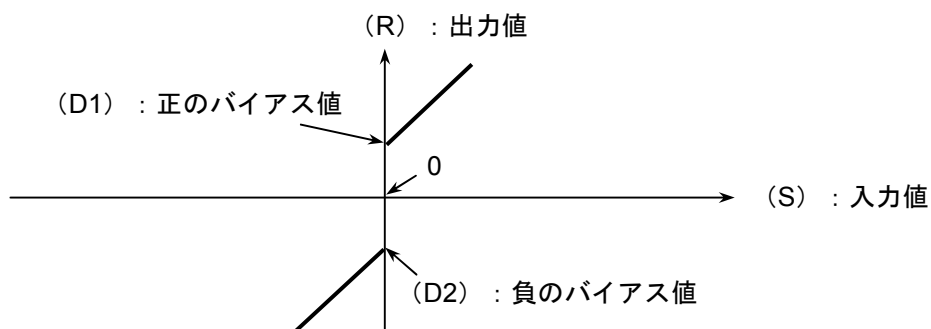
D2 : 入力値に加算する負のバイアス値を格納したレジスタまたは定数

R : ゾーン制御した出力値を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

ソース (S) で指定した入力値にデスティネーション (D1) , (D2) で指定したバイアス値を付加して、リザルト (R) に格納します。



### ● ワードデータのゾーン制御

- 下記の16ビットデータをゾーン制御します。

$$\begin{matrix} 2^{15} & 2^0 \\ \boxed{\text{(S)}} \end{matrix} > 0 \text{ のとき } \begin{matrix} 2^{15} & 2^0 \\ \boxed{\text{(S)}} \end{matrix} + \begin{matrix} 2^{15} & 2^0 \\ \boxed{\text{(D1)}} \end{matrix} \rightarrow \begin{matrix} 2^{15} & 2^0 \\ \boxed{\text{(R)}} \end{matrix}$$

$$\begin{matrix} 2^{15} & 2^0 \\ \boxed{\text{(S)}} \end{matrix} = 0 \text{ のとき } 0 \rightarrow \begin{matrix} 2^{15} & 2^0 \\ \boxed{\text{(R)}} \end{matrix}$$

$$\begin{matrix} 2^{15} & 2^0 \\ \boxed{\text{(S)}} \end{matrix} < 0 \text{ のとき } \begin{matrix} 2^{15} & 2^0 \\ \boxed{\text{(S)}} \end{matrix} + \begin{matrix} 2^{15} & 2^0 \\ \boxed{\text{(D2)}} \end{matrix} \rightarrow \begin{matrix} 2^{15} & 2^0 \\ \boxed{\text{(R)}} \end{matrix}$$

- ソース (S) , デスティネーション (D1) , (D2) に指定できる値の範囲は、-32768~32767です。

- ロングワードデータのゾーン制御
  - ・下記の32ビットデータをゾーン制御します。

$$\begin{array}{l}
 \begin{array}{|c|} \hline 2^{31} \quad 2^0 \\ \hline \boxed{\text{(S)}} \\ \hline \end{array} > 0 \text{ のとき } \begin{array}{|c|} \hline 2^{31} \quad 2^0 \\ \hline \boxed{\text{(S)}} \\ \hline \end{array} + \begin{array}{|c|} \hline 2^{31} \quad 2^0 \\ \hline \boxed{\text{(D1)}} \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline 2^{31} \quad 2^0 \\ \hline \boxed{\text{(R)}} \\ \hline \end{array} \\
 \begin{array}{|c|} \hline 2^{31} \quad 2^0 \\ \hline \boxed{\text{(S)}} \\ \hline \end{array} = 0 \text{ のとき } \begin{array}{|c|} \hline 2^{31} \quad 2^0 \\ \hline 0 \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline 2^{31} \quad 2^0 \\ \hline \boxed{\text{(R)}} \\ \hline \end{array} \\
 \begin{array}{|c|} \hline 2^{31} \quad 2^0 \\ \hline \boxed{\text{(S)}} \\ \hline \end{array} < 0 \text{ のとき } \begin{array}{|c|} \hline 2^{31} \quad 2^0 \\ \hline \boxed{\text{(S)}} \\ \hline \end{array} + \begin{array}{|c|} \hline 2^{31} \quad 2^0 \\ \hline \boxed{\text{(D2)}} \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline 2^{31} \quad 2^0 \\ \hline \boxed{\text{(R)}} \\ \hline \end{array}
 \end{array}$$

- ・ソース (S) , デスティネーション (D1) , (D2) に指定できる値の範囲は、-2147483648～2147483647です。

- フローティングデータのゾーン制御
  - ・下記のフローティングデータをゾーン制御します。

$$\begin{array}{l}
 \begin{array}{|c|} \hline \text{フローティングデータ} \\ \hline \boxed{\text{(S)}} \\ \hline \end{array} > 0 \text{ のとき } \begin{array}{|c|} \hline \text{フローティングデータ} \\ \hline \boxed{\text{(S)}} \\ \hline \end{array} + \begin{array}{|c|} \hline \text{フローティングデータ} \\ \hline \boxed{\text{(D1)}} \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline \text{フローティングデータ} \\ \hline \boxed{\text{(R)}} \\ \hline \end{array} \\
 \begin{array}{|c|} \hline \text{フローティングデータ} \\ \hline \boxed{\text{(S)}} \\ \hline \end{array} = 0 \text{ のとき } \begin{array}{|c|} \hline \text{フローティングデータ} \\ \hline 0 \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline \text{フローティングデータ} \\ \hline \boxed{\text{(R)}} \\ \hline \end{array} \\
 \begin{array}{|c|} \hline \text{フローティングデータ} \\ \hline \boxed{\text{(S)}} \\ \hline \end{array} < 0 \text{ のとき } \begin{array}{|c|} \hline \text{フローティングデータ} \\ \hline \boxed{\text{(S)}} \\ \hline \end{array} + \begin{array}{|c|} \hline \text{フローティングデータ} \\ \hline \boxed{\text{(D2)}} \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline \text{フローティングデータ} \\ \hline \boxed{\text{(R)}} \\ \hline \end{array}
 \end{array}$$

- ・ソース (S) , デスティネーション (D1) , (D2) に指定できる値の範囲は下記です。  
 $0, \pm 2^{-126} \sim \pm 2^{128}$

(3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D1	○	○	○	○	○	○	○
D2	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

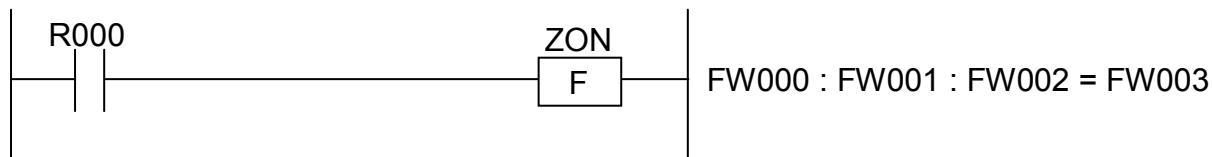
○ : 指定できます。

— : 指定できません。

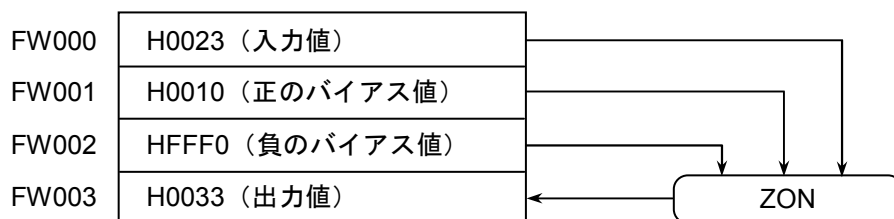
S, D1, D2, Rの型 (ワード/ロングワード/フローティング) はすべて合わせてください。異なった型は指定できません (入力エラーになります)。

## ZON デッドゾーン : DEAD ZONE

### (4) プログラム例



入力条件R000がONのとき、FW000の内容とFW001およびFW002の内容を加算して、ゾーン制御の出力値をFW003に格納します。



### (5) エラー処理

#### ● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	↕

V : ワードデータ時

- ・リザルト (R) が-32768~32767のとき0、それ以外のとき1

ロングワードデータ時

- ・リザルト (R) が-2147483648~2147483647のとき0、それ以外のとき1

フローティングデータ時

- ・演算結果によりVは変化しません (保持します)。

E : ワードおよびロングワードデータ時

- ・デスティネーション (D1) < (D2) のとき1
- ・デスティネーション (D1) ≥ (D2) のとき0

フローティングデータ時

- ・リザルト (R) が0ではなく、かつ下記範囲外のとき1、それ以外のとき0  
 $\pm 2^{-126} \sim \pm 2^{128}$

V, E以外のフラグはすべて保持します。

- (D1) < (D2) のとき、エラーフラグ (Eフラグ) がON (オーバーフローフラグ (Vフラグ) はOFF) し、リザルト (R) は変化しません。
- オーバーフロー発生時、リザルト (R) には下記のフルスケール値が格納されます。

	正のオーバーフロー時	負のオーバーフロー時
ワード	H7FFF	H8000
ロングワード	H7FFFFFFF	H80000000
フローティング	+3.402823E38	-3.402823E38

フローティングでオーバーフロー発生時、VフラグはONしません (ワードおよびロングワード時はONします)。

- フローティングでアンダーフロー発生時、リザルト (R) には正しい符号を持った0が格納されま  
す。演算結果フラグは変化しません。



# SQR 平方根 : ROOT

## (1) 入力形式

SQR S -> R

S : 平方根演算するデータの格納レジスタまたは定数

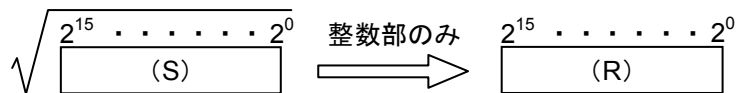
R : 演算結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

### ● ワードデータの平方根

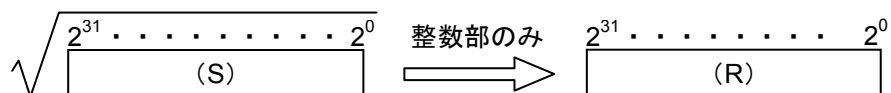
- ・ソース (S) で指定された16ビットデータの平方根を演算し、整数部のみリザルト (R) に格納します。



- ・ソース (S) <0の場合、リザルト (R) には0が格納されます。
- ・ソース (S) に指定できる値の範囲は、-32768~32767です。

### ● ロングワードデータの平方根

- ・ソース (S) で指定された32ビットデータの平方根を演算し、整数部のみリザルト (R) に格納します。



- ・ソース (S) <0の場合、リザルト (R) には0が格納されます。
- ・ソース (S) に指定できる値の範囲は、-2147483648~2147483647です。

### ● フローティングデータの平方根

- ・ソース (S) で指定されたフローティングデータの平方根を演算し、演算結果をリザルト (R) に格納します。



- ・ソース (S) <0の場合、リザルト (R) には0が格納されます。
- ・ソース (S) に指定できる値の範囲は下記です。

$$0, \pm 2^{-126} \sim \pm 2^{128}$$

(3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

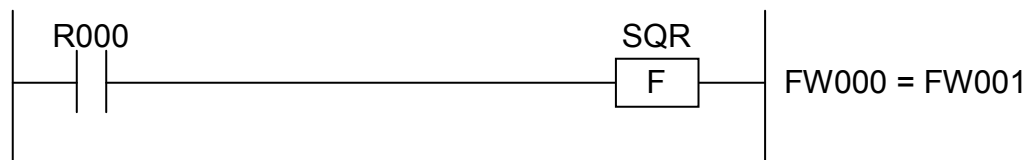
○ : 指定できます。

— : 指定できません。

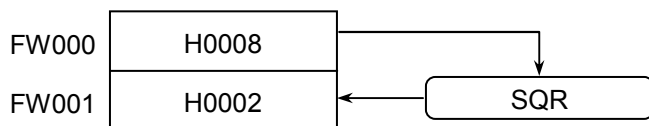
SとRの型（ワード/ロングワード/フローティング）は合わせてください。異なった型は指定できません（入力エラーになります）。

(4) プログラム例

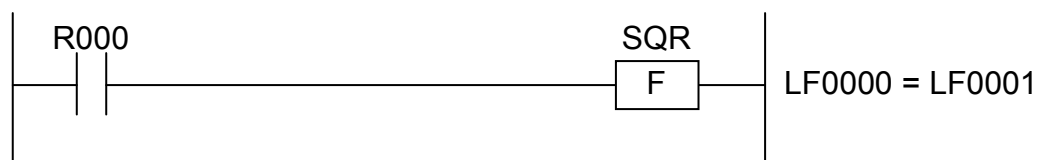
● ワードデータの平方根



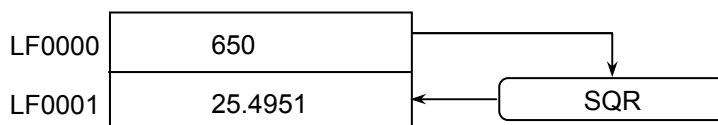
入力条件R000がONのとき、FW000の内容で平方根演算し、演算結果（整数部のみ）をFW001に格納します。



● フローティングデータの平方根



入力条件R000がONのとき、LF0000の内容で平方根演算し、演算結果をLF0001に格納します。



### (5) エラー処理

- 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	—

E : ワードデータおよびロングワードデータ時

- ・演算結果によりEは変化しません（保持します）。

フローティングデータ時

- ・リザルト (R) が0ではなく、かつ下記範囲外の時1、それ以外の時0  
 $\pm 2^{-126} \sim \pm 2^{128}$

E以外のフラグはすべて保持します。

- EフラグON時は、無処理（リザルト (R) は変化なし）です。

このページは白紙です。

# SIN 正弦 : SIN

## (1) 入力形式

SIN S -> R

S : SIN (正弦) 演算する角度データの格納レジスタまたは定数

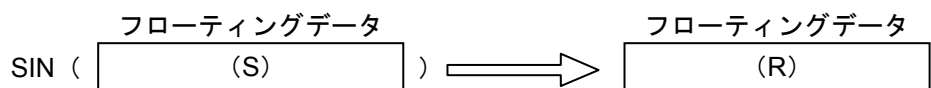
R : 演算結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) で指定した角度のSIN (正弦) 値を演算し、演算結果をリザルト (R) に格納します。

ソース (S) に指定する角度はラジアン値 (角度  $\times \pi / 180$ ) を設定します。



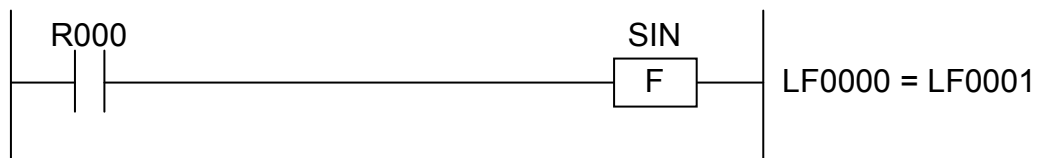
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	—	—	—	—	○	○	○
R	—	—	—	—	○	—	○

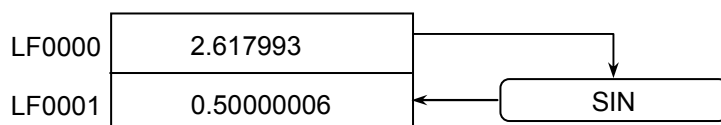
○ : 指定できます。

— : 指定できません。

## (4) プログラム例



入力条件R000がONのとき、LF0000の内容でSIN演算し、演算結果をLF0001に格納します。



## (5) エラー処理

## ● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	—

E : リザルト (R) が0ではなく、かつ下記範囲外るとき1、それ以外るとき0  
 $\pm 2^{-126} \sim \pm 2^{128}$

E以外のフラグはすべて保持します。

## ● EフラグON時は、無処理 (リザルト (R) は変化なし) です。

# COS 余弦 : COS

## (1) 入力形式

```
COS S -> R
```

S : COS (余弦) 演算する角度データの格納レジスターまたは定数

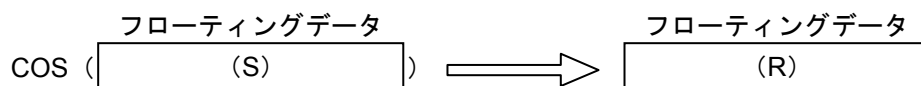
R : 演算結果を格納するレジスター

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) で指定した角度のCOS (余弦) 値を演算し、演算結果をリザルト (R) に格納します。

ソース (S) に指定する角度はラジアン値 (角度  $\times \pi / 180$ ) を設定します。



## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	-	-	-	-	○	○	○
R	-	-	-	-	○	-	○

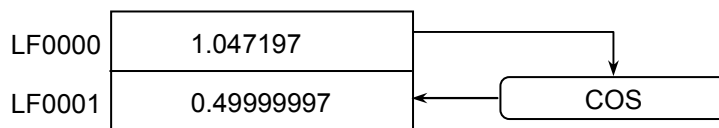
○ : 指定できます。

- : 指定できません。

(4) プログラム例



入力条件R000がONのとき、LF0000の内容でCOS演算し、演算結果をLF0001に格納します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	—

E : リザルト (R) が0ではなく、かつ下記範囲外るとき1、それ以外るとき0  
 $\pm 2^{-126} \sim \pm 2^{128}$

E以外のフラグはすべて保持します。

● EフラグON時は、無処理 (リザルト (R) は変化なし) です。



# TAN 正接：TAN

## (1) 入力形式

TAN S -> R

S：TAN（正接）演算する角度データの格納レジスタまたは定数

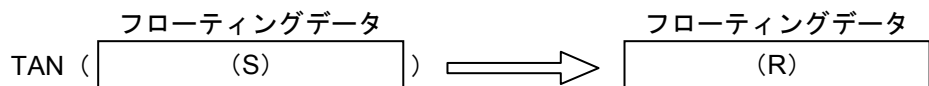
R：演算結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) で指定した角度のTAN（正接）値を演算し、演算結果をリザルト (R) に格納します。

ソース (S) に指定する角度はラジアン値（角度 × π / 180）を設定します。



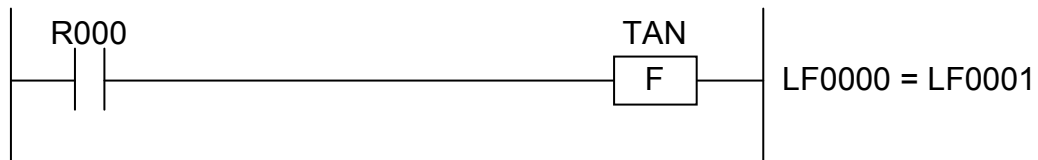
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	—	—	—	—	○	○	○
R	—	—	—	—	○	—	○

○：指定できます。

—：指定できません。

## (4) プログラム例



入力条件R000がONのとき、LF0000の内容でTAN演算し、演算結果をLF0001に格納します。



## (5) エラー処理

## ● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	—

E : リザルト (R) が0ではなく、かつ下記範囲外の時1、それ以外の時0  
 $\pm 2^{-126} \sim \pm 2^{128}$

E以外のフラグはすべて保持します。

## ● EフラグON時は、無処理 (リザルト (R) は変化なし) です。

# ASIN 逆正弦 : $\text{SIN}^{-1}$

## (1) 入力形式

ASIN S -> R

S :  $\text{SIN}^{-1}$  (逆正弦) 演算する角度データの格納レジスタまたは定数

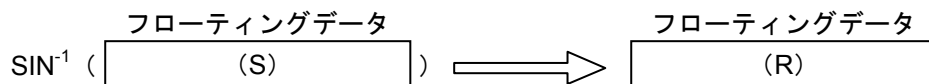
R : 演算結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) で指定したSIN値から角度を演算し、演算結果をリザルト (R) に格納します。

ソース (S) で指定するSIN値は、-1.0~1.0の範囲で設定できます。



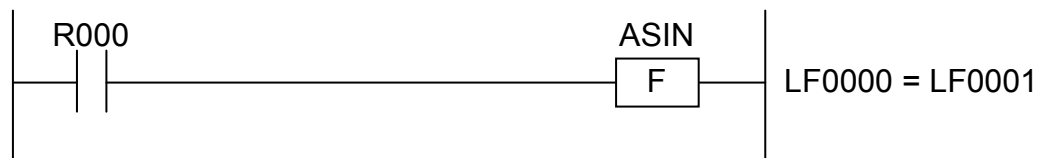
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	—	—	—	—	○	○	○
R	—	—	—	—	○	—	○

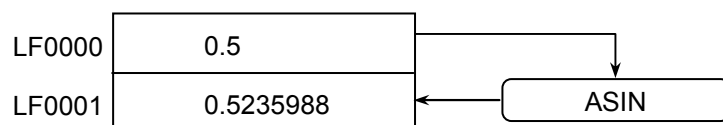
○ : 指定できます。

— : 指定できません。

## (4) プログラム例



入力条件R000がONのとき、LF0000の内容で $\text{SIN}^{-1}$ 演算し、演算結果をLF0001に格納します。



## (5) エラー処理

## ● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	—

E : ソース (S) で指定した値が、-1.0~1.0の範囲外の時1、範囲内の時0

リザルト (R) が0ではなく、かつ下記範囲外の時1、それ以外の時0

$$\pm 2^{-126} \sim \pm 2^{128}$$

E以外のフラグはすべて保持します。

## ● EフラグON時は、無処理 (リザルト (R) は変化なし) です。

# ACOS 逆余弦 : $\cos^{-1}$

## (1) 入力形式

ACOS S -> R

S :  $\cos^{-1}$  (逆余弦) 演算する角度データの格納レジスタまたは定数

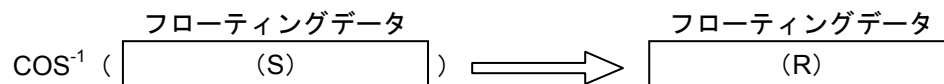
R : 演算結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) で指定したCOS値から角度を演算し、演算結果をリザルト (R) に格納します。

ソース (S) で指定するCOS値は、-1.0~1.0の範囲で設定できます。



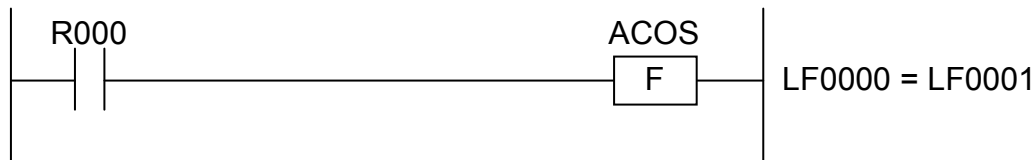
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	—	—	—	—	○	○	○
R	—	—	—	—	○	—	○

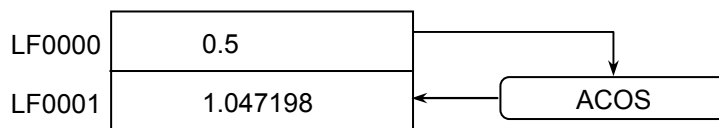
○ : 指定できます。

— : 指定できません。

## (4) プログラム例



入力条件R000がONのとき、LF0000の内容で $\text{COS}^{-1}$ 演算し、演算結果をLF0001に格納します。



## (5) エラー処理

## ● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	—

E : ソース (S) で指定した値が、 $-1.0 \sim 1.0$ の範囲外るとき1、範囲内るとき0

リザルト (R) が0ではなく、かつ下記範囲外るとき1、それ以外るとき0

$$\pm 2^{-126} \sim \pm 2^{128}$$

E以外のフラグはすべて保持します。

## ● EフラグON時は、無処理 (リザルト (R) は変化なし) です。

# ATAN 逆正接 : $TAN^{-1}$

## (1) 入力形式

ATAN S -> R

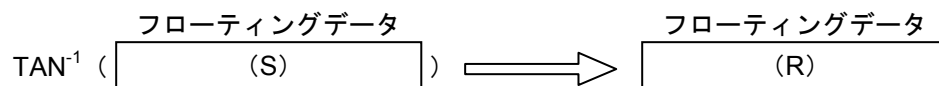
S :  $TAN^{-1}$  (逆正弦) 演算する角度データの格納レジスタまたは定数

R : 演算結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) で指定した角度のTAN値から角度を演算し、演算結果をリザルト (R) に格納します。



## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	—	—	—	—	○	○	○
R	—	—	—	—	○	—	○

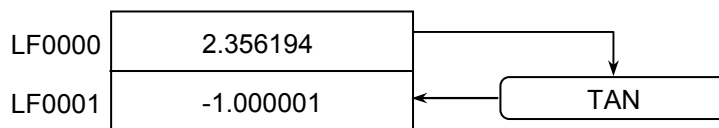
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がONのとき、LF0000の内容で $\text{TAN}^{-1}$ 演算し、演算結果をLF0001に格納します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	—

E : リザルト (R) が0ではなく、かつ下記範囲外るとき1、それ以外るとき0  
 $\pm 2^{-126} \sim \pm 2^{128}$

E以外のフラグはすべて保持します。

● EフラグON時は、無処理 (リザルト (R) は変化なし) です。



# EXP 指数演算 : EXPONENTIAL

## (1) 入力形式

EXP S -> R

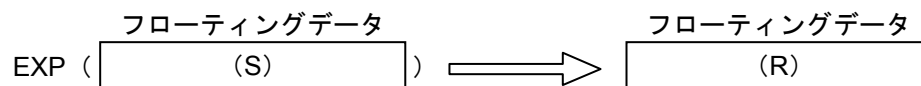
S : 指数演算するデータの格納レジスタまたは定数

R : 演算結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) で指定した値の指数を演算し、演算結果をリザルト (R) に格納します。



## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	—	—	—	—	○	○	○
R	—	—	—	—	○	—	○

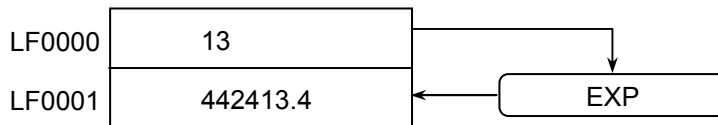
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がONのとき、LF0000の内容で指数演算し、演算結果をLF0001に格納します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	—

E : リザルト (R) が0ではなく、かつ下記範囲外るとき1、それ以外るとき0  
 $\pm 2^{-126} \sim \pm 2^{128}$

E以外のフラグはすべて保持します。

● EフラグON時は、無処理 (リザルト (R) は変化なし) です。

# LOG 自然対数 : LOGARITHM

## (1) 入力形式

LOG S -> R

S : 自然対数を演算するデータの格納レジスタまたは定数

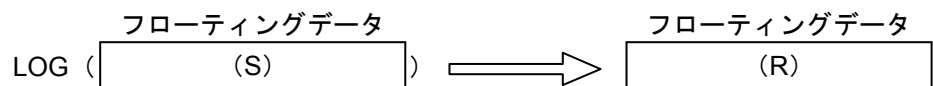
R : 演算結果を格納するレジスタ

(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「->」は省略できます。

## (2) 機能

ソース (S) で指定した値の、自然対数 (e) を底としたときの対数を演算し、結果をリザルト (R) に格納します。

ソース (S) で指定する値は、正の整数のみ設定できます。



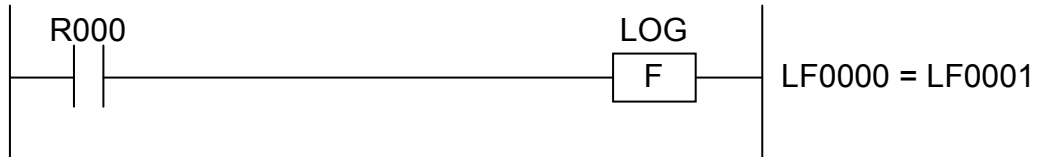
## (3) データタイプ

	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	—	—	—	—	○	○	○
R	—	—	—	—	○	—	○

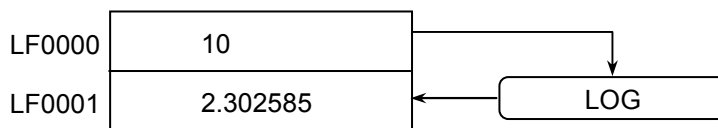
○ : 指定できます。

— : 指定できません。

(4) プログラム例



入力条件R000がONのとき、LF0000の内容で自然対数を演算し、演算結果をLF0001に格納します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	↕	—	—	—	—

E : ソース (S) で指定した値が負のとき1、正のとき0

リザルト (R) が0ではなく、かつ下記範囲外のとき1、それ以外のとき0  
 $\pm 2^{-126} \sim \pm 2^{128}$

E以外のフラグはすべて保持します。

● EフラグON時は、無処理 (リザルト (R) は変化なし) です。

# MAX 最大 : MAXIMUM

## (1) 入力形式

```
MAX S : D -> R
```

S : ソース格納レジスタまたは定数  
D : デスティネーション格納レジスタまたは定数  
R : 最大値を格納するレジスタ

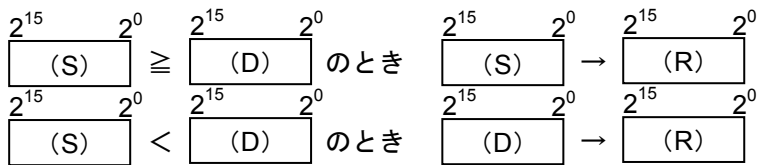
(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

指定されたソース (S) とデスティネーション (D) を大小比較し、大きい方の値をリザルト (R) に格納します。

### ● ワードデータの最大値取り込み

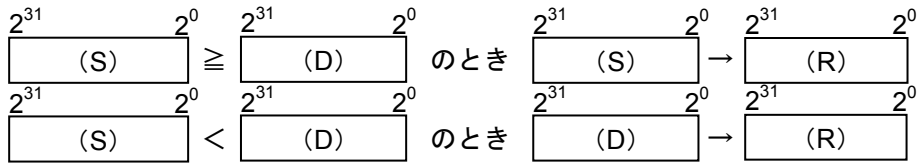
・下記の16ビットデータを大小比較し、大きい方の値をリザルト (R) に格納します。



・ソース (S) , デスティネーション (D) に指定できる値の範囲は、-32768~32767です。

### ● ロングワードデータの最大値取り込み

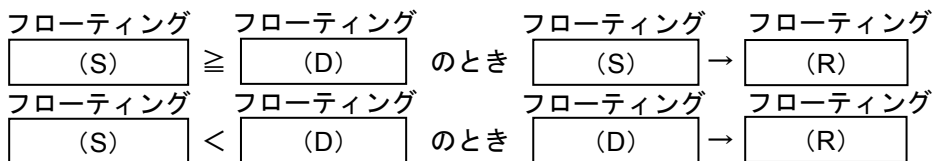
・下記の32ビットデータを大小比較し、大きい方の値をリザルト (R) に格納します。



・ソース (S) , デスティネーション (D) に指定できる値の範囲は、-2147483648~2147483647です。

### ● フローティングデータの最大値取り込み

・下記のフローティングデータを大小比較し、大きい方の値をリザルト (R) に格納します。



・ソース (S) , デスティネーション (D) に指定できる値の範囲は下記です。  
 $0, \pm 2^{-126} \sim \pm 2^{128}$

## (3) データタイプ

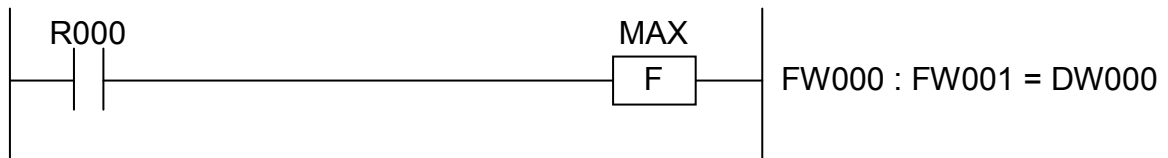
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○ : 指定できます。

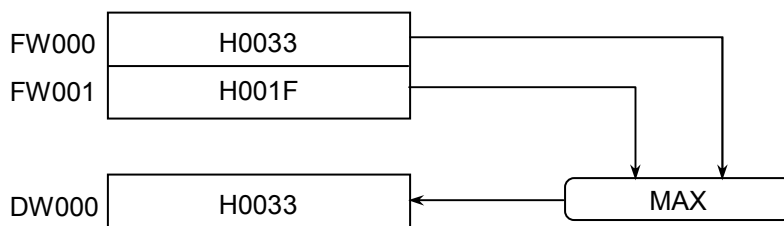
— : 指定できません。

S, D, およびRの型 (ワード/ロングワード/フローティング) はすべて合わせてください。異なった型は指定できません (入力エラーになります)。

## (4) プログラム例



入力条件R000がONのとき、FW000の内容とFW001の内容を対象比較して、大きい値をDW000に格納します。



## (5) エラー処理

## ● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# MIN 最小 : MINIMUM

## (1) 入力形式

MIN S : D -> R

S : ソース格納レジスタまたは定数  
 D : デスティネーション格納レジスタまたは定数  
 R : 最小値を格納するレジスタ

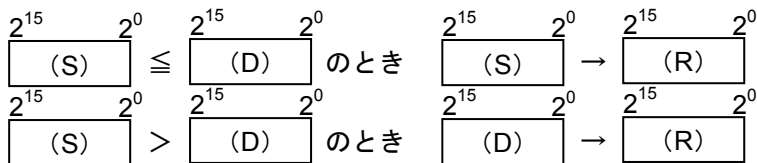
(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

## (2) 機能

指定されたソース (S) とデスティネーション (D) を大小比較し、小さい方の値をリザルト (R) に格納します。

### ● ワードデータの最小値取り込み

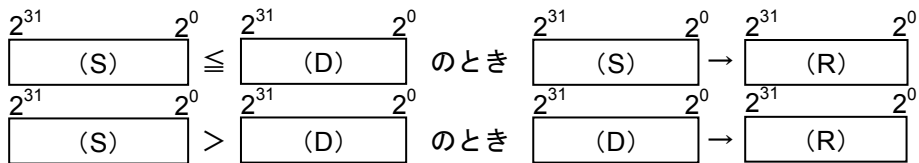
・下記の16ビットデータを大小比較し、小さい方の値をリザルト (R) に格納します。



・ソース (S) , デスティネーション (D) に指定できる値の範囲は、-32768~32767です。

### ● ロングワードデータの最小値取り込み

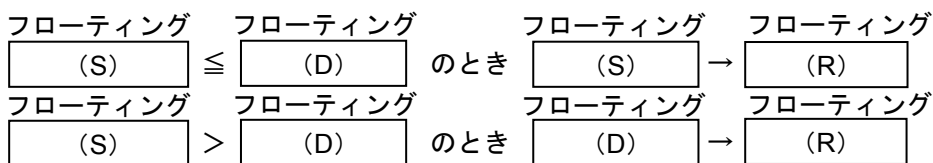
・下記の32ビットデータを大小比較し、小さい方の値をリザルト (R) に格納します。



・ソース (S) , デスティネーション (D) に指定できる値の範囲は、-2147483648~2147483647です。

### ● フローティングデータの最小値取り込み

・下記のフローティングデータを大小比較し、小さい方の値をリザルト (R) に格納します。



・ソース (S) , デスティネーション (D) に指定できる値の範囲は下記です。  
 $0, \pm 2^{-126} \sim \pm 2^{128}$

(3) データタイプ

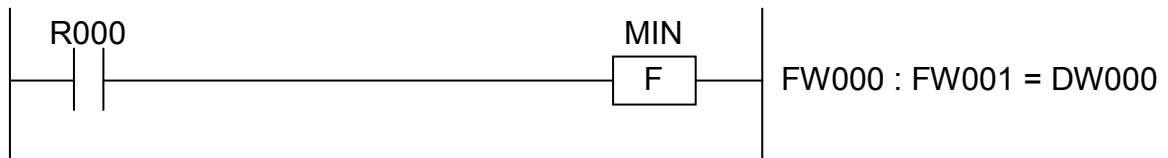
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○ : 指定できます。

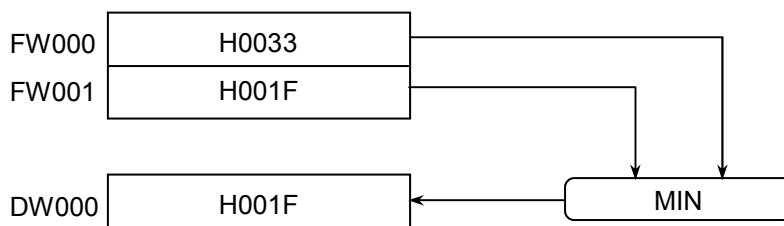
— : 指定できません。

S, D, およびRの型 (ワード/ロングワード/フローティング) はすべて合わせてください。異なった型は指定できません (入力エラーになります)。

(4) プログラム例



入力条件R000がONのとき、FW000の内容とFW001の内容を対象比較して、大きい値をDW000に格納します。



(5) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。



# CLR クリア : CLEAR

---

## (1) 入力形式

XCLR
YCLR
GCLR
RCLR
KCLR
TCLR
UCLR
CCLR
VCLR
ECLR
FCLR

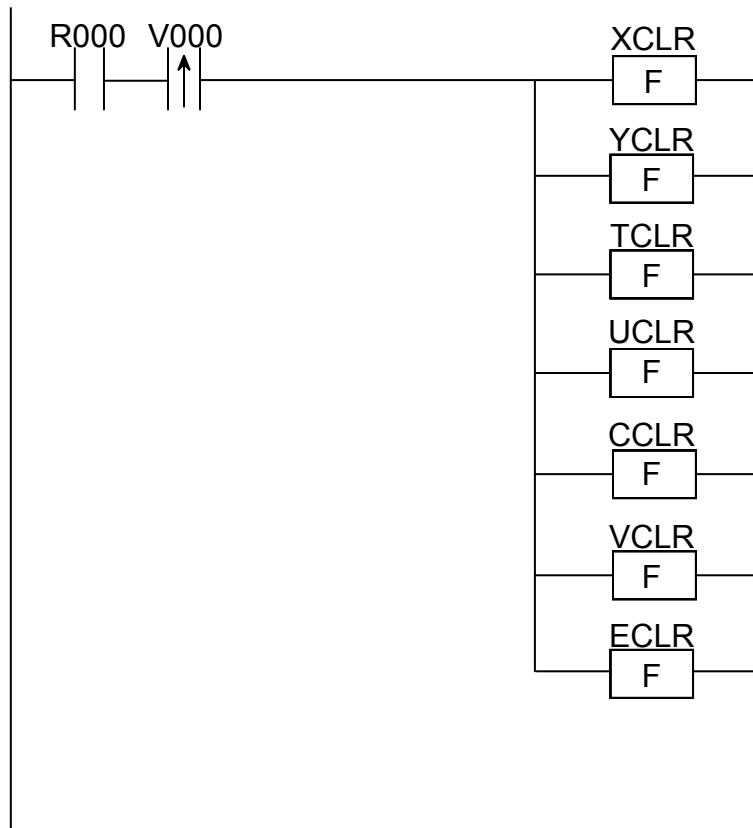
(\*) この命令にパラメーターはありません。

## (2) 機能

指定のI/Oエリアをクリアします。

- ① XCLR : Xエリア (外部入力) をクリアします。
- ② YCLR : Yエリア (外部出力) をクリアします。
- ③ GCLR : Gエリア (グローバルリンクレジスター) をクリアします。
- ④ RCLR : Rエリア (内部レジスター) をクリアします。
- ⑤ KCLR : Kエリア (キープリレー) をクリアします。
- ⑥ TCLR : Tエリア (オンディレイタイマーおよび計数值) をクリアします。
- ⑦ UCLR : Uエリア (ワンショットタイマーおよび計数值) をクリアします。
- ⑧ CCLR : Cエリア (U/Dカウンターおよび計数值) をクリアします。
- ⑨ VCLR : Vエリア (エッジ接点) をクリアします。
- ⑩ ECLR : Eエリア (イベントレジスター) をクリアします。
- ⑪ FCLR : 演算結果フラグ (X, E, P, N, Z, V) をクリアします。

## (3) プログラム例



入力条件R000がOFF→ONの変化時、1回のみX, Y, T, U, C, V, Eエリアをクリアします。

## (4) エラー処理

この命令は、常に正常終了します。

## JT 条件成立ジャンプ : JUMP if True

### (1) 入力形式

JT LAB

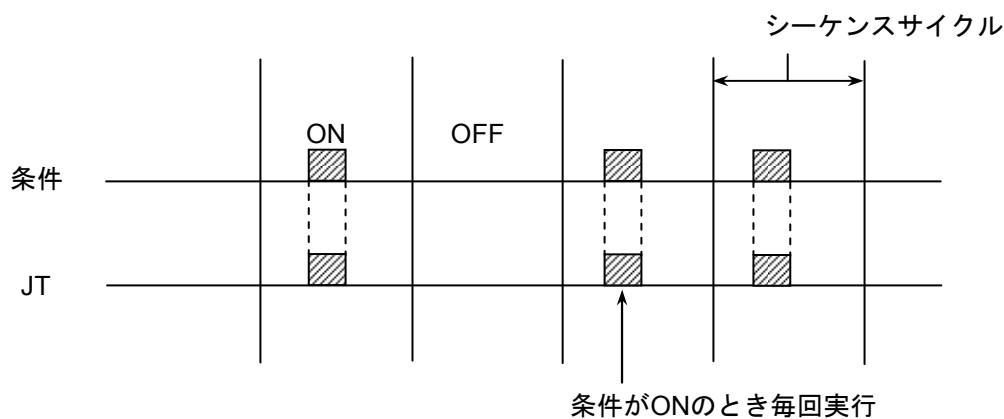
LAB : ジャンプ先ラベル名称

(\*) ファンクション名称とパラメーターの間には、必ずスペースを入力してください。

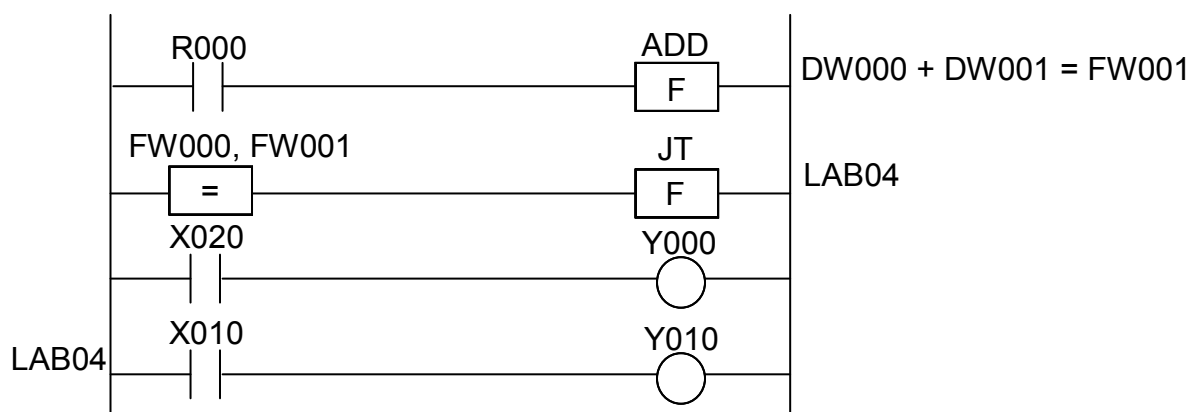
### (2) 機能

条件がONのとき、指定されたラベルにジャンプします。

条件がOFFのとき、次のステップを実行します。



### (3) プログラム例



入力条件R000がONのとき、DW000とDW001を加算し、結果をFW001に格納します。次にFW000とFW001が一致した場合、ラベルLAB04にジャンプし、X010がONのときY010をONします。FW000とFW001が不一致の場合、ラベルLAB04にジャンプせず、X020がONのときY000をONし、その後にラベルLAB04以降を実行します。

(5) エラー処理

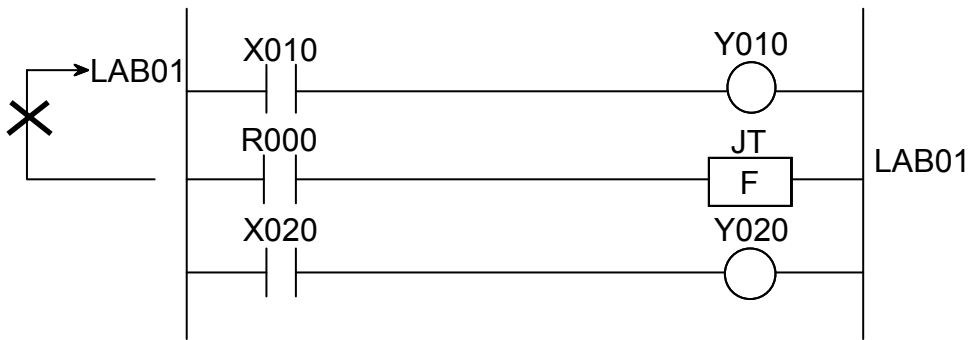
● 演算結果フラグ

X	E	P	N	Z	V
-	-	-	-	-	-

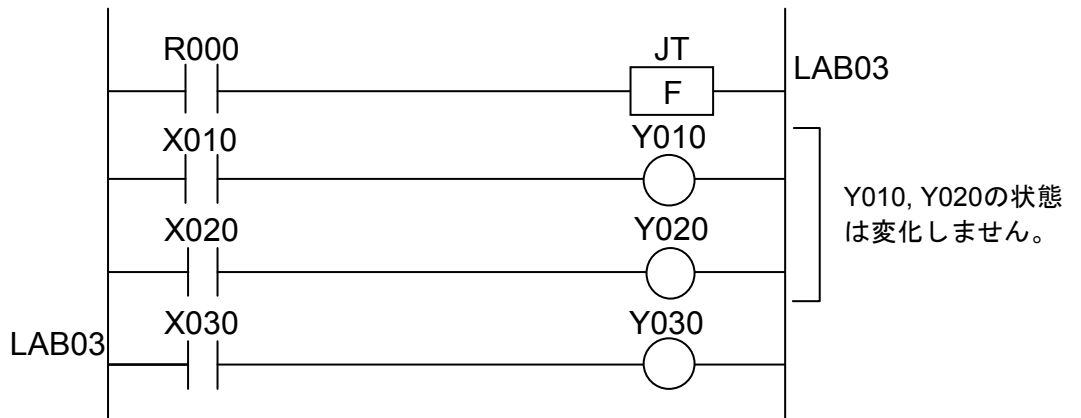
フラグはすべて保持します。

(注)

- ・ジャンプ命令では、実行中のステップより前のラベルへのジャンプは指定できません (ラダープログラム無限ループ防止のため)。



- ・ジャンプにより飛ばされた回路のコイルの状態は変化しません。



# JMP 無条件ジャンプ : JUMP

## (1) 入力形式

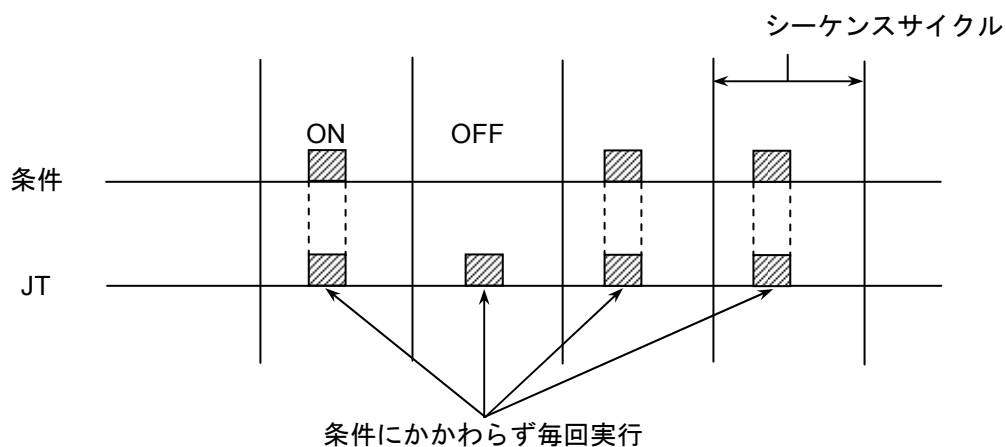
```
JMP LAB
```

LAB : ジャンプ先ラベル名称

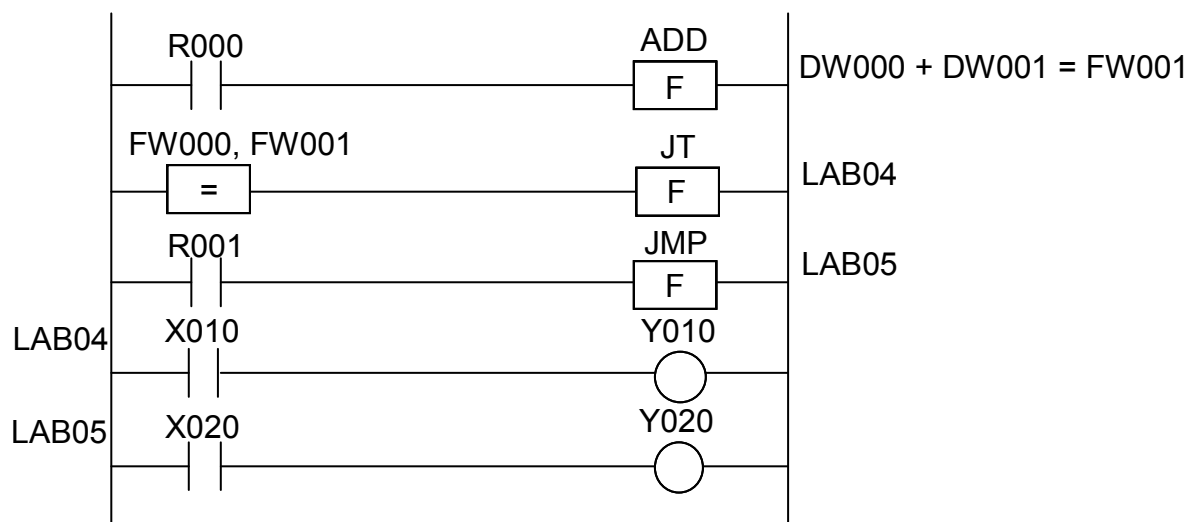
(\*) ファンクション名称とパラメーターの間には、必ずスペースを入力してください。

## (2) 機能

無条件に指定されたラベルにジャンプします。



## (3) プログラム例



入力条件R000がONのとき、DW000とDW001を加算し、結果をFW001に格納します。次にFW000とFW001が一致した場合、ラベルLAB04にジャンプし、X010がONのときY010をONし、さらにX020がONのときY020をONします。FW000とFW001が不一致の場合、R001の状態（ON/OFF）にかかわらず無条件にラベルLAB05にジャンプし、X020がONのときY020をONします。

(4) エラー処理

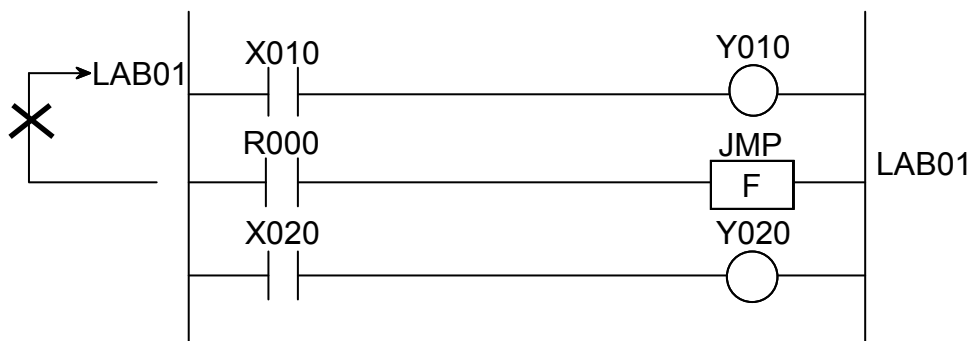
● 演算結果フラグ

X	E	P	N	Z	V
-	-	-	-	-	-

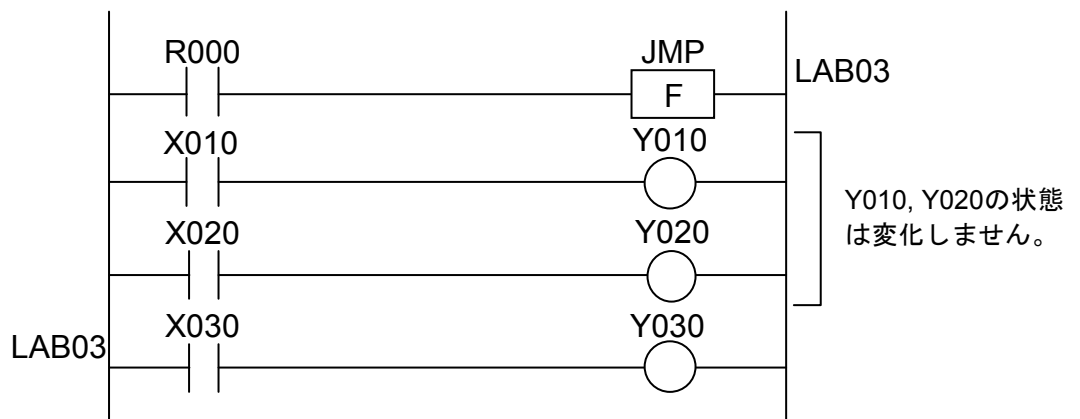
フラグはすべて保持します。

(注)

- ・ジャンプ命令では、実行中のステップより前のラベルへのジャンプは指定できません (ラダープログラム無限ループ防止のため)。



- ・ジャンプにより飛ばされた回路のコイルの状態は変化しません。



## JSE 条件成立SENDジャンプ : JUMP to SEND

### (1) 入力形式

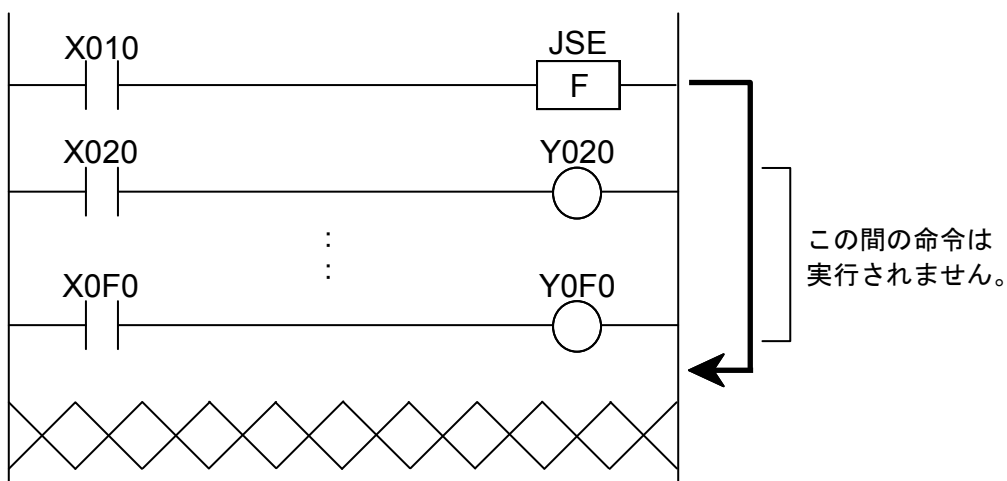


### (2) 機能

条件がONのとき、現在実行しているNコイルの最後 (SEND (\*)) にジャンプします。

(\* ) SEND : シーケンスエンド (SequenceEND) 各Nコイルの最後を意味します。

### (3) プログラム例



入力条件X010がONのとき、SENDにジャンプします。

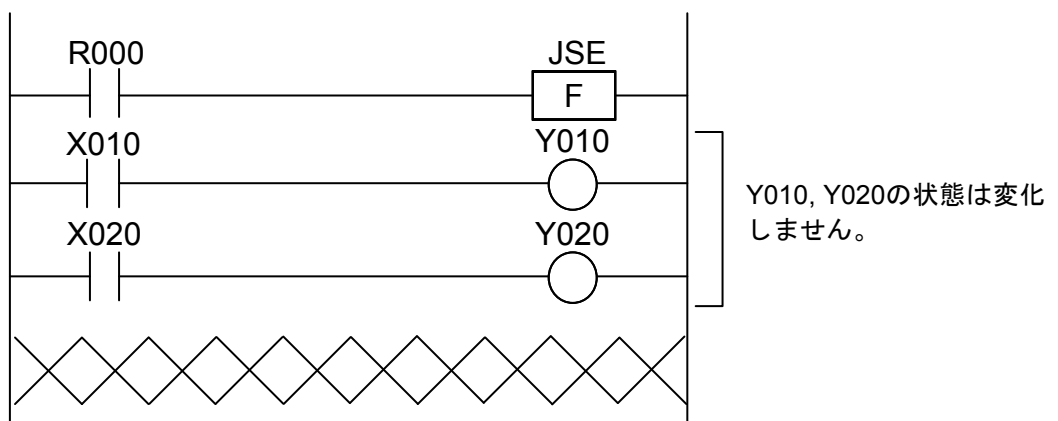
(4) エラー処理

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

(注) ジャンプにより飛ばされた回路のコイルの状態は変化しません。





## 2 演算ファンクション

### 2.7 イーサネット通信命令

#### 2.7.1 機能概要

ラダープログラムでTCP通信、UDP通信を行う場合、イーサネット通信システム拡張演算ファンクションを使用してください。

ラダー図システムのイーサネット通信システム拡張演算ファンクションとして、以下のインタフェースをサポートします。

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

システム拡張演算ファンクションに従った通信の仕様を以下に示します。

項目	仕様	備考
同時使用可能ソケット数	CMU : 16	TCPの送受信、UDPの送受信の合計
	ET.NET (メイン) : 16	
	ET.NET (サブ) : 16	
	OPTET (Module 0~3共用) : 32	
送受信データサイズ	TCP通信 : 0~4096バイト	
	UDP通信 : 0~1472バイト	
ポート番号	1~65535	10000~59999の使用を推奨、60000以降はシステム予約

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

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

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

## 2 演算ファンクション

イーサネット通信システム拡張演算ファンクションを実行すると、各管理番号ごとに実行結果がシステムレジスタS9C0～S9EF、S690～S6AFに設定されます。

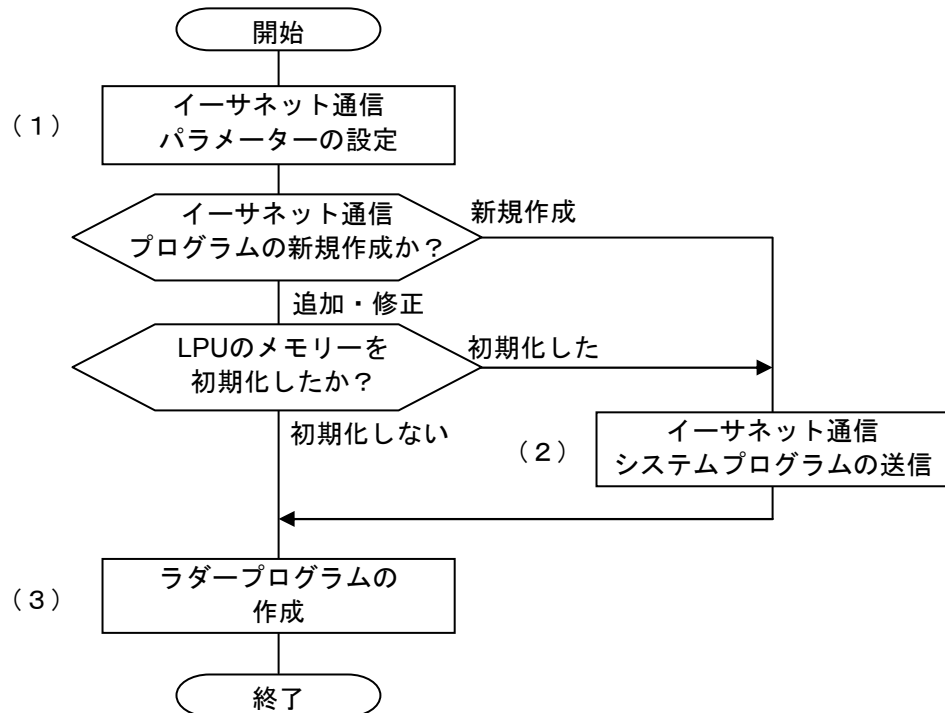
正常終了時は0、異常終了時は1が管理番号に該当するシステムレジスタに設定されます。

管理番号とは、使用できるソケットに対応付けている番号のことです。

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

## 2.7.2 使用方法

イーサネット通信システム拡張演算ファンクションは、ラダー図システムの [イーサネット通信設定] 画面で設定するパラメーター情報をもとに動作します。そのため、次のフローに示すように、ラダープログラム作成前に [イーサネット通信設定] 画面でパラメーター情報を設定しておく必要があります。

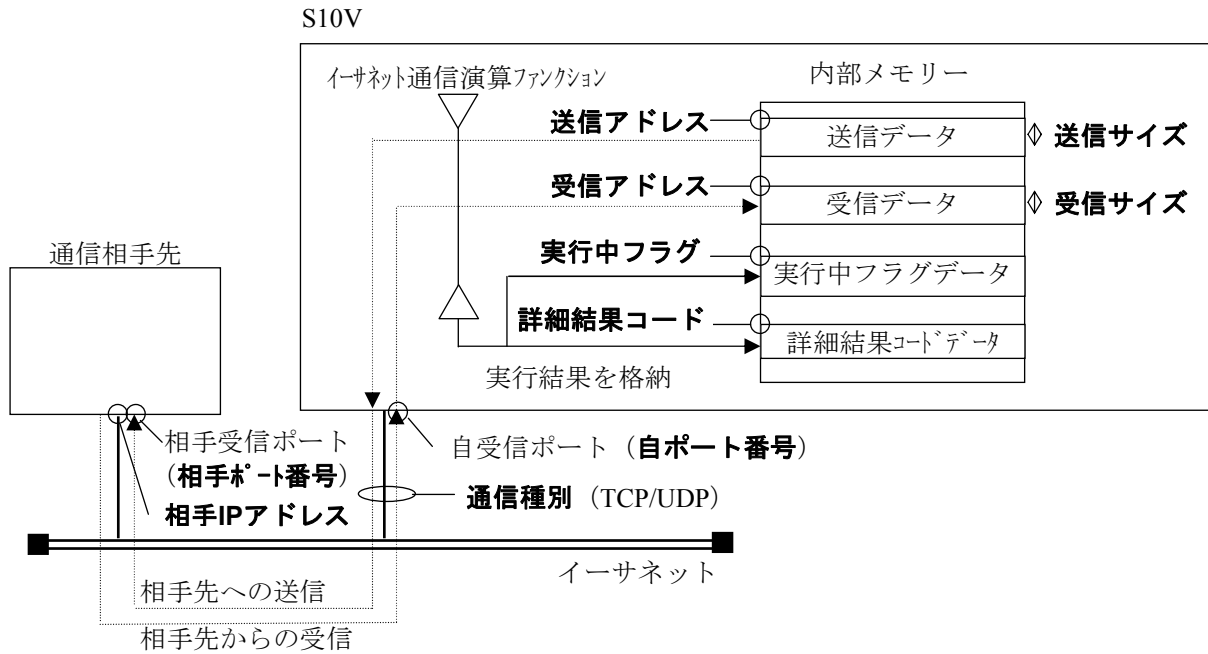


## 2 演算ファンクション

パラメーター情報は、以下の図を参考にして設定してください。

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

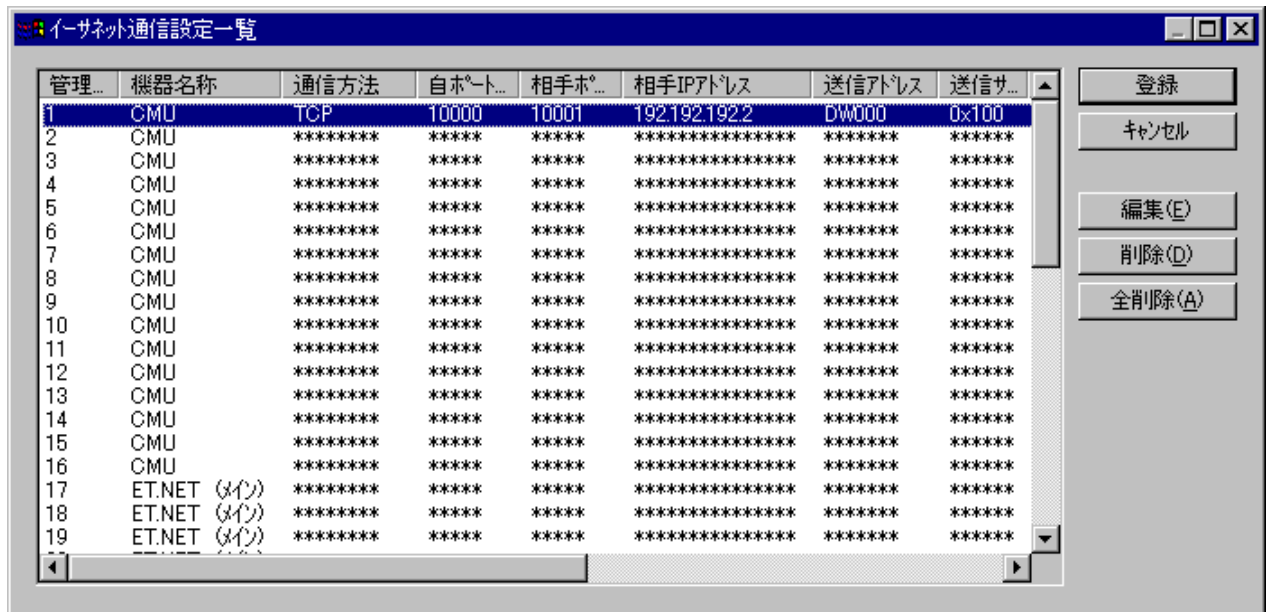
[イーサネット通信設定]画面の設定情報の詳細については、「(1) イーサネット通信パラメーターの設定」を参照してください。



## (1) イーサネット通信パラメーターの設定

[イーサネット通信設定] 画面でパラメーター情報を設定する場合、ラダー図システムのメニューから [ユーティリティ] - [イーサネット通信設定] - [パラメータ設定] を選択すると、以下の [イーサネット通信設定一覧] 画面が表示されます。

< [イーサネット通信設定一覧] 画面 >



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

なお、設定の詳細は「ソフトウェアマニュアル オペレーション S10V ラダー図 For Windows® (マニュアル番号 SVJ-3-131)」を参照してください。

## 2 演算ファンクション

< [イーサネット通信設定] 画面 >

イーサネット通信設定

管理番号 : 1

機器名称 : CMU

通信方法 : TCP

接続情報

自ポート番号 : 10000

相手ポート番号 : 10001

相手IPアドレス : 192 192 192 2

送受信エリア

送信アドレス : DW000 ~ DW07F

送信サイズ : / 100 Byte

受信アドレス : FW000 ~ FW0FF

受信サイズ : / 200 Byte

受信タイムアウト時間 : 10 (\*100ms)

結果格納エリア

実行中フラグ : R000

詳細結果コード : LWL0000

ソケット切断方法 : 未送信データ送信待ち

OK

キャンセル

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

管理番号 :

[イーサネット通信設定一覧] 画面で指定された管理番号を表示します。

機器名称 :

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

機器名称は管理番号に対して固定で、以下のモジュールが表示されます。

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

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

### 通信方法：

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

### 自ポート番号：

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

### 相手ポート番号：

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

### 相手IPアドレス：

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

### 送信アドレス：

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

### 送信サイズ：

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

TCP : 0x0～0x1000 (0～4096)

UDP : 0x0～0x5C0 (0～1472)

### 受信アドレス：

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

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



## 2 演算ファンクション

### 受信サイズ：

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

TCP：0x0～0x1000（0～4096）

UDP：0x0～0x5C0（0～1472）

### 受信タイムアウト時間：

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

### 実行中フラグ：

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

### 詳細結果コード：

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

### ソケット切断方法：

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

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

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

以下に [イーサネット通信設定] 画面で指定可能なレジスターを示します。

<設定レジスター一覧>

(1/2)

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

○：指定できます。

—：指定できません。

## 2 演算ファンクション

(2/2)

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

○：指定できます。

—：指定できません。

以下にイーサネット通信システム拡張演算ファンクションの詳細結果コードを示します。

< 詳細結果コード一覧 >

(1/2)

値	意味	対処方法
0	正常 (TOP, TPOP, TCLO, UOP, UCLO)	—
0~4096	正常 (送受信データサイズ) (TRCV, TSND, URCV, USND)	—
0x80000005 (EIO)	アダプター (デバイス) に重障害発生	エラーログ情報 (*1) に記載の対処方法を参照してください。
0x8000000D (EACCES)	宛先IPアドレスにブロードキャストアドレス指定	イーサネット通信設定が異常です。設定を見直してください。
0x80000016 (EINVAL)	切断されたソケットを指定または受信バッファ長が負の値	イーサネット通信設定が異常です。設定を見直してください。
0x800000DA (EMSGSIZE)	送信データ長範囲外	イーサネット通信設定が異常です。設定を見直してください。
0x800000E2 (EADDRINUSE)	ポート番号が別ソケットで使用中	使用ポート番号を見直してください。
0x800000E3 (EADDRNOTAVAIL)	ポート番号、IPアドレス指定誤り	イーサネット通信設定が異常です。設定を見直してください。
0x800000E4 (ENETDOWN)	デバイス未初期化または停止された。	エラーログ情報 (*1) に記載の対処方法を参照してください。
0x800000E5 (ENETUNREACH)	宛先IPアドレスの経路情報なし	CMUモジュール/ET.NETモジュールの経路情報設定を見直してください。 (*2)
0x800000E7 (ECONNABORTED)	コネクションの接続が打ち切られた。	・ケーブルの配線を見直してください。 ・接続先ホストのプログラムを見直してください。
0x800000E8 (ECONNRESET)	接続先ホストのTCPよりコネクションがリセット	接続先ホストのプログラムを見直してください。
0x800000E9 (ENOBUFS)	メモリー確保失敗	エラーログ情報 (*1) に記載の対処方法を参照してください。
0x800000EB (ENOTCONN)	コネクションが確立していないソケットに送信	TOP/TPOPが実行に失敗しています。プログラムを見直してください。
0x800000EC (ESHUTDOWN)	他タスクからソケットが解放された。	ラダー/HI-FLOWプログラムで同じ管理番号を使用していないか確認してください。
0x800000EE (ETIMEDOUT)	コネクション要求タイムアウト	・ケーブルの配線を見直してください。 ・接続先ホストのプログラムを見直してください。
0x800000EF (ECONNREFUSED)	接続先ソケットが存在しない。 (サーバータスクが未bind)	接続先ホストのプログラムを見直してください。
0x800000F6 (EWOULDBLOCK)	受信データなし。TCPの送信ウィンドウが満杯で送信できない。	プログラムを見直してください。
0x800000F9 (ENSOCK)	オープンできるソケット数オーバー	同時に使用するソケットが1モジュールあたり16以内になるよう、プログラムを見直してください。
0x80000516 (EBADF)	ソケット未オープン	TOP/TPOP/UOPが実行されていないか、実行に失敗しています。プログラムを見直してください。

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

(\*2) 経路情報の設定は、各モジュールの設定ツールから行ってください。

## 2 演算ファンクション

(2/2)

値	意味	対処方法
0xFFFFFFFFB	イーサネットモジュールがダウン中	LPUモジュールを一度リセットしてET.NETモジュールを再立ち上げしてください。再立ち上げしても同じエラーが返る場合は、ET.NETモジュールが故障している可能性があります。ET.NETモジュールを交換してください。
0xFFFFFFFFC	イーサネットモジュールが未実装	CMUモジュール/ET.NETモジュールの実装を確認してください。
0xFFFFFFFFD	タスク起動失敗	CMUモジュールがイーサネット通信応用命令に対応しているバージョンであるか確認してください。
0xFFFFFFFFE	通信識別子異常（管理番号が使用中）	同じ管理番号のイーサネット通信設定を使用しているラダー/HI-FLOWがないか確認してください。
0xFFFFFFFFF	使用種別不整合（パラメーター情報の送信方法、通信種別と異なる）	イーサネット通信設定の通信方法と、ラダー/HI-FLOWプログラムの通信種別が同一であるか確認してください。

### ・エラー種別

0x8XXXXXXXX : CPMSソケットマクロエラー（実際のCPMSソケットマクロエラーに0x80000000が加算された値）

0xFXXXXXXXX : システムプログラム、タスクでのエラー

以下にイーサネット通信時のエラートレースについて説明します。

各管理番号ごとに、8ケースのエラートレースを収集しており、基本システムの [イーサ通信エラーログ情報 (ラダー、HI-FLOW)] 画面から参照できます。 [イーサ通信エラーログ情報 (ラダー、HI-FLOW)] 画面に表示されているエラートレース情報からエラー内容を判断する方法については「ユーザーズマニュアル 基本モジュール (マニュアル番号 SVJ-1-100)」を参照してください。

< [イーサ通信エラーログ情報 (ラダー、HI-FLOW)] 画面 >

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

## 2 演算ファンクション

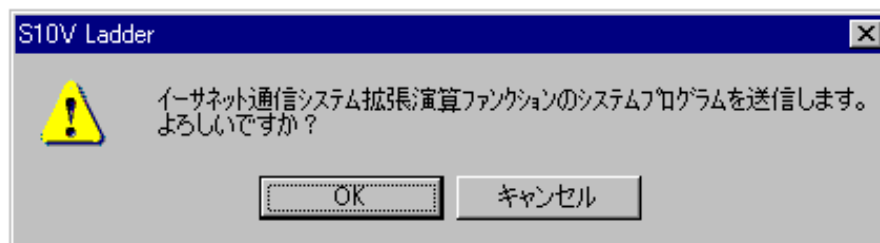
### (2) イーサネット通信システムプログラムの送信

LPUモジュールにイーサネット通信機能（イーサネット通信システム拡張演算ファンクション）のシステムプログラムを送信します。イーサネット通信機能のシステムプログラムは、ラダー図システムで提供しているため、イーサネット通信機能を使用するときは、最初にLPUモジュールにシステムプログラムを送信してください。

なお、この操作は、オンライン時のみ行えます。

システムプログラムの送信は、[ユーティリティ]メニューから[イーサネット通信設定] - [システムプログラム送信]を選択すると、以下の[システムプログラム送信確認]画面が表示されます。

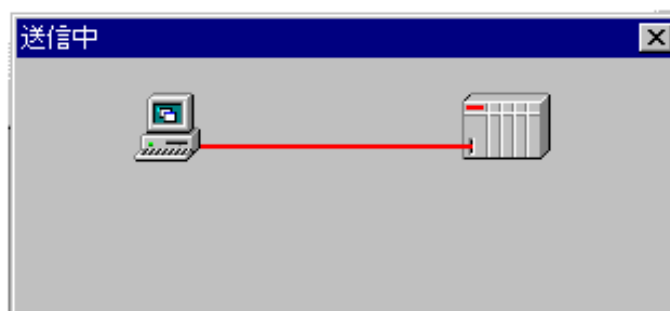
< [システムプログラム送信確認] 画面 >



送信する場合は、**OK** ボタンをクリックしてください。送信しない場合は **キャンセル** ボタンをクリックしてください。

**OK** ボタンをクリックすると、以下の[送信中]画面が表示されます。

< [送信中] 画面 >



送信が完了すると[送信中]画面が自動的に閉じます。

(注)

- システムプログラムを送信（＝イーサネット通信システム拡張演算ファンクションを実装）しなくてもラダープログラムは作成できます。ただし、ラダープログラムにイーサネット通信システム拡張演算ファンクションが使用されている場合、ラダープログラムの送信時にイーサネット通信システム拡張演算ファンクションの実装チェックを行います。実装されていない場合、エラーメッセージダイアログボックスを表示してラダープログラムの送信は行いません。

エラーメッセージ：“システム拡張演算ファンクションのXXXXが実装されていません。”

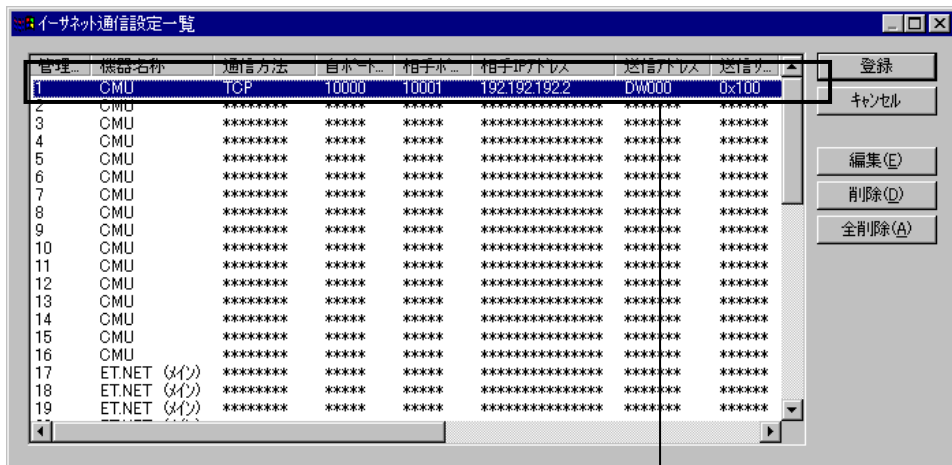
(XXXXはイーサネット通信システム拡張演算ファンクションの演算ファンクション名称)

- LPUモジュールのメモリを初期化した場合、送信したシステムプログラムがクリアされてしまいますので、再度システムプログラムの送信が必要になります。

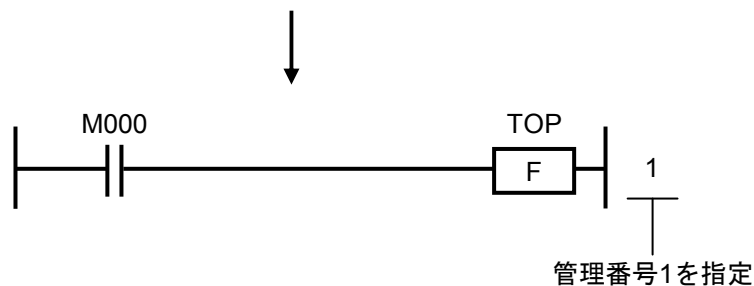
(3) ラダープログラムの作成

イーサネット通信システム拡張演算ファンクションのパラメーターには、[イーサネット通信設定]画面で設定した管理番号のみを指定して、ラダープログラムを作成します。イーサネット通信システム拡張演算ファンクションは、管理番号から[イーサネット通信設定]画面で設定した情報に従って動作します。

<作成例>



[イーサネット通信設定]画面で管理番号1の情報を設定





## 2 演算ファンクション

### 2.7.3 命令の詳細

イーサネット通信システム拡張演算ファンクション命令の説明は、以下に示す形式になっています。

#### (1) 入力形式

命令の入力形式を示します。

#### (2) 機能

命令が処理する機能について説明しています。説明文中のシステムレジスターとは、イーサネット通信システム拡張演算ファンクションの実行結果を格納するシステムレジスター（S9C0～S9FF）のことを指します。

#### (3) データタイプ

パラメーターに指定できるデータタイプに○を付けています。

(例)

	DW000などのレジスターやH0001などの定数が使用できるかを表しています。		LLL0000などのレジスターやH04231556などの定数が使用できるかを表しています。		LF0000などのレジスターや1.12E-002などの定数が使用できるかを表しています。		
	ワード		ロングワード		フローティング		インデックス指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	—

○：指定できます。  
—：指定できません。

レジスターが指定できる場合、インデックス指定ができるかを表しています。

この例では、S（ソース）にはワード、ロングワード、フローティングデータのアドレス（インデックス指定含む）、および定数を指定できます。

(注) R000, Y1FFなどビットのI/Oエリアは、演算ファンクションではワードデータとして扱います。

この場合、LSBのみ有効で、他のビットの読み出し時は常に0、書き込みは無効です。詳細は「2.3.2 ビットレジスターの扱い」を参照してください。

#### (4) プログラム例

簡単なラダープログラム例と処理の内容を示しています。

#### (5) エラー処理

エラー発生時の条件と演算結果フラグに反映されるフラグを示しています。

このページは白紙です。

## (1) 入力形式

TOP S
-------

S : ソース（通信識別子）。モジュールにより以下に示す管理番号を指定します。

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

TCP, UDPあわせて、1つのモジュールにつき同時に最大16までの通信が可能です。

OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。

## (2) 機能

### ● TCPコネクションのオープン（クライアント）

TOPは、ソケットをオープンし、[イーサネット通信設定]画面で設定した相手ポート番号、相手IPアドレスで相手側へコネクションの接続を行います。処理が完了しなくても命令は終了します。処理の結果は、システムレジスターと[イーサネット通信設定]画面で設定した「実行中フラグ」、「詳細結果コード」に格納されます。

実行中フラグ：処理実行中は1、処理完了時は0が設定されます。処理が完了したとき、システムレジスター、「詳細結果コード」に処理の結果が設定されます。そのため、命令の実行後、処理が完了するまで「実行中フラグ」を監視してください。

詳細結果コード：命令の処理が完了したとき、処理の結果が設定されます。「実行中フラグ」が0になった後、「詳細結果コード」を参照してください。

## (3) データタイプ

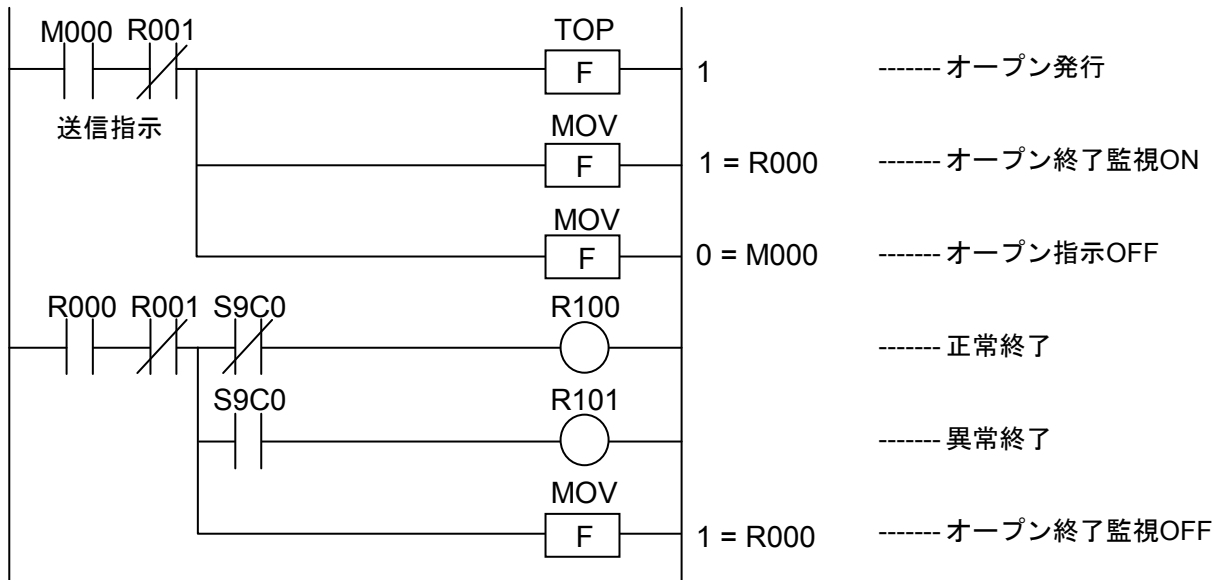
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	—

○：指定できます。

—：指定できません。

(4) プログラム例

管理番号 = 1、実行中フラグ = R001でオープン



(5) エラー処理

TCPコネクションのオープン（クライアント）が成功すると、システムレジスターに0、「詳細結果コード」に0が設定されます。失敗した場合は、システムレジスターに1、「詳細結果コード」にエラーNo. (0以外の値) が設定されます。

処理の成功、失敗はシステムレジスターで判定してください。

失敗時のエラー要因は、「詳細結果コード」を参照してください。

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# TPOP TCPコネクションのオープン（サーバー）：TPOP

## (1) 入力形式

TPOP S
--------

S：ソース（通信識別子）。モジュールにより以下に示す管理番号を指定します。

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

TCP, UDPあわせて、1つのモジュールにつき同時に最大16までの通信が可能です。

OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。

## (2) 機能

### ● TCPコネクションのオープン（サーバー）

TPOPは、ソケットをオープンし、[イーサネット通信設定]画面で設定したサーバーの自ポート番号でクライアント側からのコネクション要求を受け付けサーバーとクライアントを接続します。処理が完了しなくても命令は終了します。処理の結果は、システムレジスターと[イーサネット通信設定]画面で設定した「実行中フラグ」、「詳細結果コード」に格納されます。

実行中フラグ：処理実行中は1、処理完了時は0が設定されます。処理が完了したとき、システムレジスター、「詳細結果コード」に処理の結果が設定されます。そのため、命令の実行後、処理が完了するまで「実行中フラグ」を監視してください。

詳細結果コード：命令の処理が完了したとき、処理の結果が設定されます。「実行中フラグ」が0になった後、「詳細結果コード」を参照してください。

## (3) データタイプ

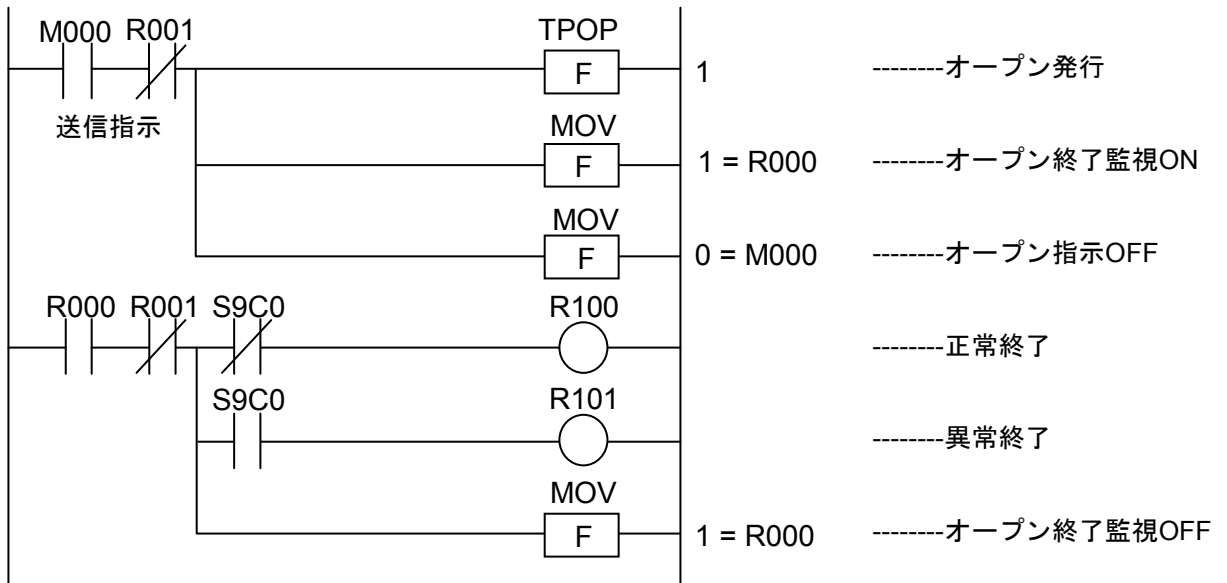
	ワード		ロングワード		フローティング		インデックス指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	—

○：指定できます。

—：指定できません。

(4) プログラム例

管理番号 = 1、実行中フラグ = R001でオープン



(5) エラー処理

TCPコネクションのオープン（サーバー）が成功すると、システムレジスターに0、「詳細結果コード」に0が設定されます。失敗した場合は、システムレジスターに1、「詳細結果コード」にエラーNo. (0以外の値) が設定されます。

処理の成功、失敗はシステムレジスターで判定してください。

失敗時のエラー要因は、「詳細結果コード」を参照してください。

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# TCLO TCP接続のクローズ : TCLO

## (1) 入力形式

TCLO S
--------

S : ソース (通信識別子)。モジュールにより以下に示す管理番号を指定します。

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

TCP, UDPあわせて、1つのモジュールにつき同時に最大16までの通信が可能です。

OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。

## (2) 機能

### ● TCP接続のクローズ

TCLOは、[イーサネット通信設定]画面で設定した「ソケット切断方法」で通信路を切断し、ソケットの解放を行います。処理が完了しなくても命令は終了します。処理の結果は、システムレジスターと[イーサネット通信設定]画面で設定した「実行中フラグ」、「詳細結果コード」に格納されます。

実行中フラグ : 処理実行中は1、処理完了時は0が設定されます。処理が完了したとき、システムレジスター、「詳細結果コード」に処理の結果が設定されます。そのため、命令の実行後、処理が完了するまで「実行中フラグ」を監視してください。

詳細結果コード : 命令の処理が完了したとき、処理の結果が設定されます。「実行中フラグ」が0になった後、「詳細結果コード」を参照してください。

## (3) データタイプ

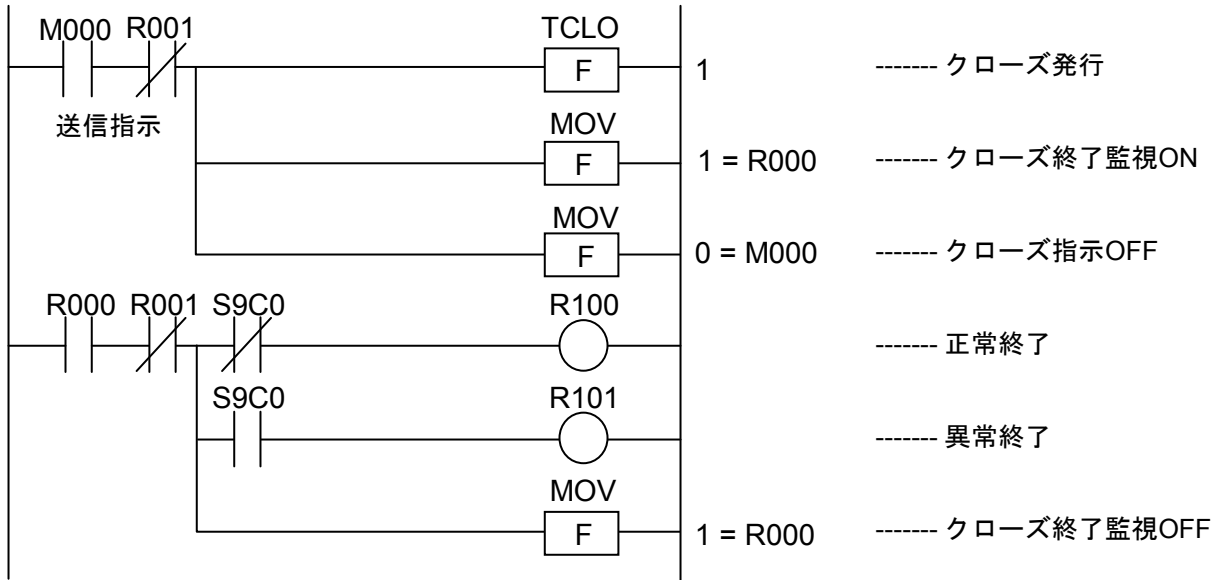
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	—

○ : 指定できます。

— : 指定できません。

(4) プログラム例

管理番号 = 1、実行中フラグ = R001でクローズ



(5) エラー処理

TCP接続のクローズが成功すると、システムレジスターに0、「詳細結果コード」に0が設定されます。失敗した場合は、システムレジスターに1、「詳細結果コード」にエラーNo. (0以外の値)が設定されます。

処理の成功、失敗はシステムレジスターで判定してください。

失敗時のエラー要因は、「詳細結果コード」を参照してください。

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。



## (1) 入力形式

TRCV S
--------

S : ソース (通信識別子)。モジュールにより以下に示す管理番号を指定します。

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

TCP, UDPあわせて、1つのモジュールにつき同時に最大16までの通信が可能です。

OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。

## (2) 機能

### ● TCP受信

TRCVは、[イーサネット通信設定]画面で設定した「受信サイズ」、「受信アドレス」をもとに、ソケットからのメッセージを「受信サイズ」分受信し、「受信アドレス」で指定したエリアに受信データを格納します。処理が完了しなくても命令は終了します。処理の結果は、システムレジスターと[イーサネット通信設定]画面で設定した「実行中フラグ」、「詳細結果コード」に格納されます。

命令を発行したときに受信データがない場合、[イーサネット通信設定]画面で設定した「受信タイムアウト時間」分受信の監視を行います。データが受信されず「受信タイムアウト時間」が経過した場合、「詳細結果コード」にEWOULDBLOCKを設定して受信処理を終了します。

「詳細結果コード」がEWOULDBLOCKで、受信を継続したい場合は、TRCVを再発行してください。

実行中フラグ : 処理実行中は1、処理完了時は0が設定されます。処理が完了したとき、システムレジスター、「詳細結果コード」に処理の結果が設定されます。そのため、命令の実行後、処理が完了するまで「実行中フラグ」を監視してください。

詳細結果コード : データを受信した場合、受信データサイズ (正の値) が設定されます。

受信データが「受信サイズ」と異なる場合は、以下の状態になります。

「受信サイズ」 > 受信データ … 受信データ分読み込み

「受信サイズ」 < 受信データ … 「受信サイズ」分読み込み。オーバーした

受信データはそのまま保持され、再度TRCVを発行することで残りの受信データが受信可能です。

データ受信に失敗した場合、エラーコード (負の値) が設定されます。

(3) データタイプ

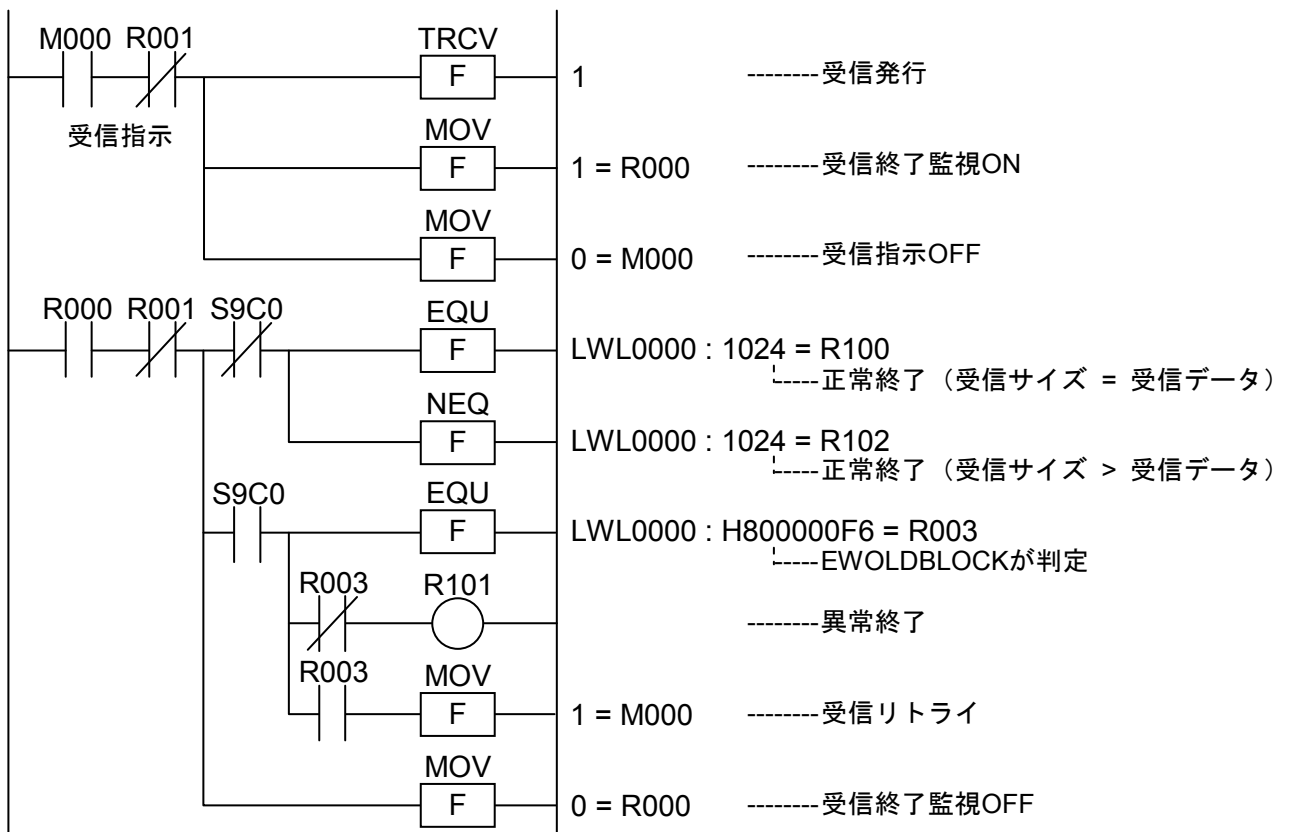
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	—

○ : 指定できます。

— : 指定できません。

(4) プログラム例

管理番号 = 1、実行中フラグ = R001、詳細結果コード = LWL0000、受信サイズ = 1024バイトで受信。  
受信データがない場合は、受信リトライ。



### (5) エラー処理

受信に成功すると、システムレジスターに0、「詳細結果コード」に受信データサイズが設定されます。受信に失敗した場合、システムレジスターに1、「詳細結果コード」にエラーNo. (負の値) が設定されます。処理の成功、失敗はシステムレジスターで判定してください。

失敗時のエラー要因は、「詳細結果コード」を参照してください。なお、「詳細結果コード」がEWOULDBLOCKの場合、TRCVの再発行が可能です。「詳細結果コード」がEWOULDBLOCKで、受信を継続したい場合は、TRCVを再発行してください。

#### ● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

このページは白紙です。

## (1) 入力形式

TSND S
--------

S : ソース (通信識別子)。モジュールにより以下に示す管理番号を指定します。

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

TCP, UDPあわせて、1つのモジュールにつき同時に最大16までの通信が可能です。

OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。

## (2) 機能

### ● TCP受信

TSNDは、[イーサネット通信設定]画面で設定した「送信サイズ」、「送信アドレス」をもとに、「送信アドレス」で指定したエリアの送信データを「送信サイズ」分ソケットへ送信します。処理が完了しなくても命令は終了します。処理の結果は、システムレジスターと[イーサネット通信設定]画面で設定した「実行中フラグ」、「詳細結果コード」に格納されます。

実行中フラグ : 処理実行中は1、処理完了時は0が設定されます。処理が完了したとき、システムレジスター、「詳細結果コード」に処理の結果が設定されます。そのため、命令の実行後、処理が完了するまで「実行中フラグ」を監視してください。

詳細結果コード : 命令の処理が完了したとき、処理の結果が設定されます。「実行中フラグ」が0になった後、「詳細結果コード」を参照してください。

## (3) データタイプ

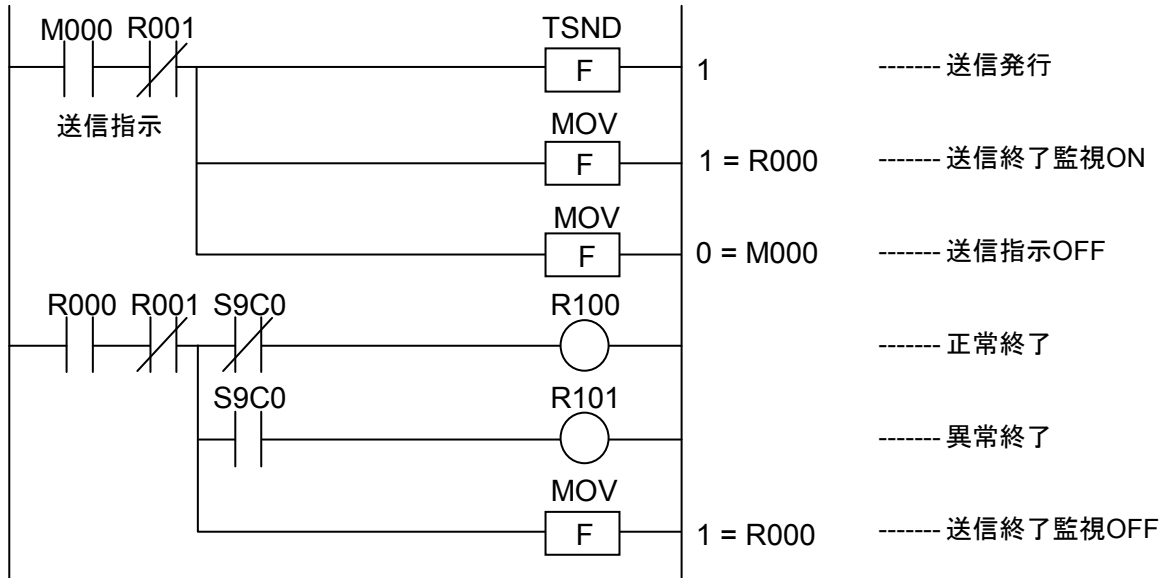
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	-	-	-	-	-

○ : 指定できます。

- : 指定できません。

(4) プログラム例

管理番号 = 1、実行中フラグ = R001で送信



(5) エラー処理

送信に成功すると、システムレジスターに0、「詳細結果コード」に0が設定されます。  
 失敗した場合、システムレジスターに1、「詳細結果コード」にエラーNo. (0以外の値) が設定されます。

処理の成功、失敗はシステムレジスターで判定してください。

失敗時のエラー要因は、「詳細結果コード」を参照してください。

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

# UOP UDPのオープン : UOP

## (1) 入力形式

UOP S
-------

S : ソース (通信識別子)。モジュールにより以下に示す管理番号を指定します。

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

TCP, UDPあわせて、1つのモジュールにつき同時に最大16までの通信が可能です。

OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。

## (2) 機能

### ● UDPのオープン

UOPは、ソケットをオープンし、[イーサネット通信設定]画面で設定した自ポート番号でソケットへのアドレス情報の割り付けを行います。処理が完了しなくても命令は終了します。処理の結果は、システムレジスターと[イーサネット通信設定]画面で設定した「実行中フラグ」、「詳細結果コード」に格納されます。

実行中フラグ : 処理実行中は1、処理完了時は0が設定されます。処理が完了したとき、システムレジスター、「詳細結果コード」に処理の結果が設定されます。そのため、命令の実行後、処理が完了するまで「実行中フラグ」を監視してください。

詳細結果コード : 命令の処理が完了したとき、処理の結果が設定されます。「実行中フラグ」が0になった後、「詳細結果コード」を参照してください。

## (3) データタイプ

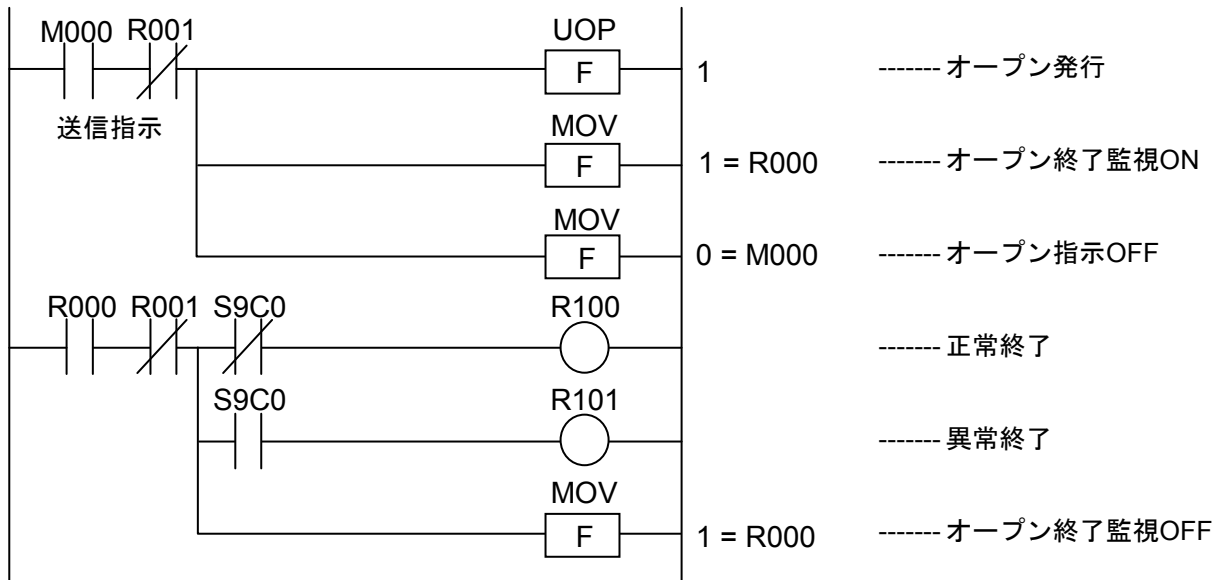
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	—

○ : 指定できます。

— : 指定できません。

(4) プログラム例

管理番号 = 1、実行中フラグ = R001でオープン



(5) エラー処理

UDPのオープンが成功すると、システムレジスターに0、「詳細結果コード」に0が設定されます。失敗した場合は、システムレジスターに1、「詳細結果コード」にエラーNo. (0以外の値) が設定されます。

処理の成功、失敗はシステムレジスターで判定してください。

失敗時のエラー要因は、「詳細結果コード」を参照してください。

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。



## (1) 入力形式

UCLO S
--------

S : ソース (通信識別子)。モジュールにより以下に示す管理番号を指定します。

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

TCP, UDPあわせて、1つのモジュールにつき同時に最大16までの通信が可能です。

OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。

## (2) 機能

### ● UDPのクローズ

UCLOは、通信路を切断しソケットの解放を行います。処理が完了しなくても命令は終了します。処理の結果は、システムレジスターと [イーサネット通信設定] 画面で設定した「実行中フラグ」、「詳細結果コード」に格納されます。

実行中フラグ : 処理実行中は1、処理完了時は0が設定されます。処理が完了したとき、システムレジスター、「詳細結果コード」に処理の結果が設定されます。そのため、命令の実行後、処理が完了するまで「実行中フラグ」を監視してください。

詳細結果コード : 命令の処理が完了したとき、処理の結果が設定されます。「実行中フラグ」が0になった後、「詳細結果コード」を参照してください。

## (3) データタイプ

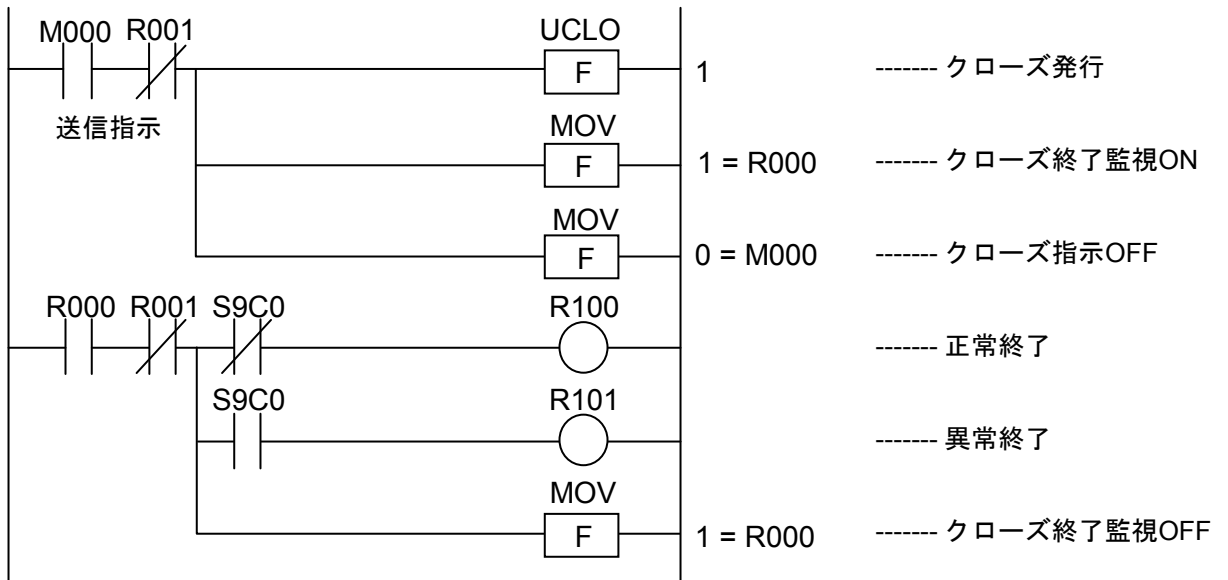
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	-	-	-	-	-

○ : 指定できます。

- : 指定できません。

(4) プログラム例

管理番号 = 1、実行中フラグ = R001でクローズ



(5) エラー処理

UDPのクローズが成功すると、システムレジスターに0、「詳細結果コード」に0が設定されます。失敗した場合は、システムレジスターに1、「詳細結果コード」にエラーNo. (0以外の値) が設定されます。

処理の成功、失敗はシステムレジスターで判定してください。

失敗時のエラー要因は、「詳細結果コード」を参照してください。

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。

## (1) 入力形式

URCV S
--------

S : ソース (通信識別子)。モジュールにより以下に示す管理番号を指定します。

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

TCP, UDPあわせて、1つのモジュールにつき同時に最大16までの通信が可能です。

OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。

## (2) 機能

### ● UDP受信

URCVは、[イーサネット通信設定]画面で設定した「受信サイズ」、「受信アドレス」をもとに、ソケットからのメッセージを「受信データ」長分受信し、受信バッファエリアに受信データを格納します。処理が完了しなくても命令は終了します。処理の結果は、システムレジスターと[イーサネット通信設定]画面で設定した「実行中フラグ」、「詳細結果コード」に格納されます。

命令を発行したときに受信データがない場合、[イーサネット通信設定]画面で設定した「受信タイムアウト時間」分受信の監視を行います。データが受信されず「受信タイムアウト時間」が経過した場合、「詳細結果コード」にEWOULDBLOCKを設定して受信処理を終了します。

「詳細結果コード」がEWOULDBLOCKで、受信を継続したい場合は、URCVを再発行してください。

実行中フラグ : 処理実行中は1、処理完了時は0が設定されます。処理が完了したとき、システムレジスター、「詳細結果コード」に処理の結果が設定されます。そのため、命令の実行後、処理が完了するまで「実行中フラグ」を監視してください。

詳細結果コード : データを受信した場合、受信データサイズ (正の値) が設定されます。

受信データが「受信サイズ」と異なる場合は、以下の状態になります。

「受信サイズ」 > 受信データ ... 受信データ分読み込み

「受信サイズ」 < 受信データ ... 「受信サイズ」分読み込み。オーバーした受信データはそのまま保持され、再度URCVを発行することで残りの受信データが受信できます。

データ受信に失敗した場合、エラーコード (負の値) が設定されます。

(3) データタイプ

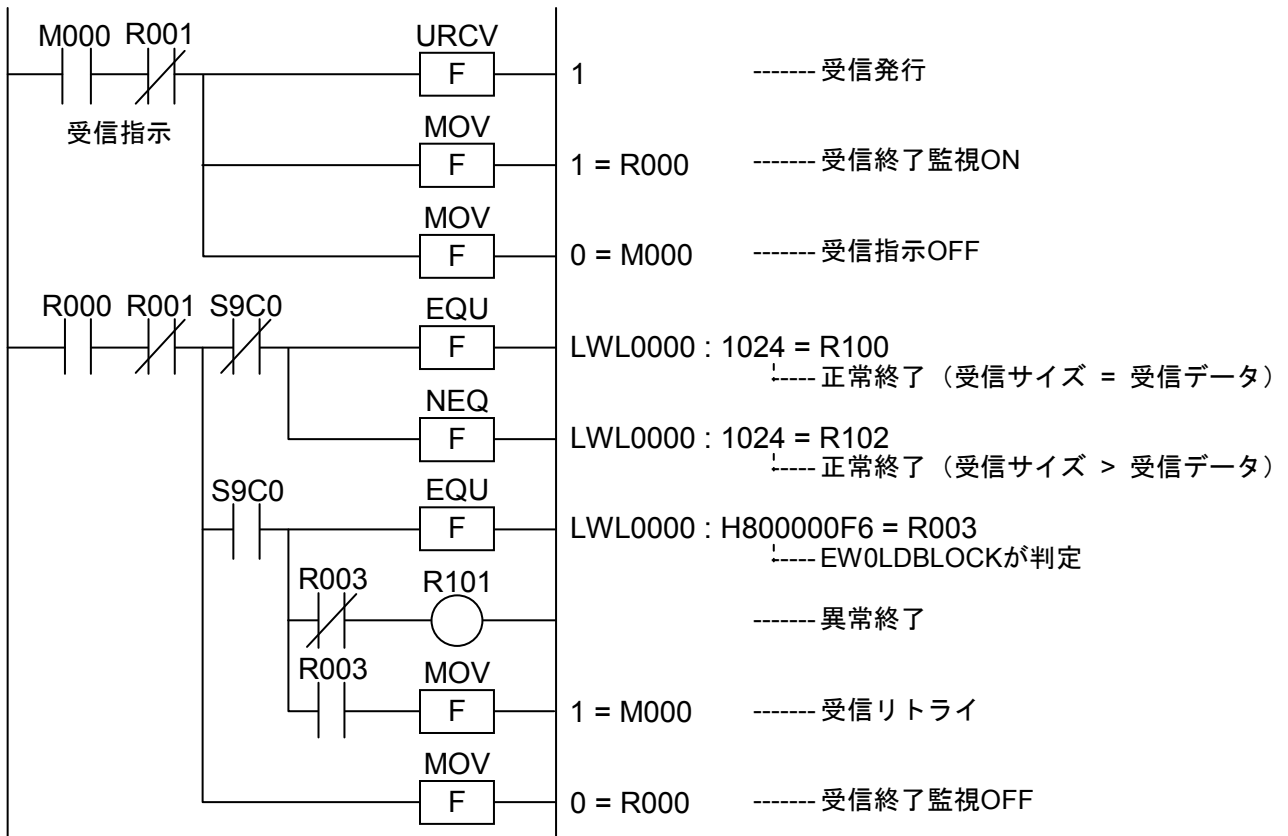
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	—

○ : 指定できます。

— : 指定できません。

(4) プログラム例

管理番号 = 1、実行中フラグ = R001、詳細結果コード = LWL0000で受信



### (5) エラー処理

受信に成功すると、システムレジスターに0、「詳細結果コード」に受信データサイズが設定されます。受信に失敗した場合、システムレジスターに1、「詳細結果コード」にエラーNo. (負の値) が設定されます。処理の成功、失敗はシステムレジスターで判定してください。

失敗時のエラー要因は、「詳細結果コード」を参照してください。なお、「詳細結果コード」がEWOULDBLOCKの場合、URCVの再発行が可能です。「詳細結果コード」がEWOULDBLOCKで、受信を継続したい場合は、URCVを再発行してください。

#### ● 演算結果フラグ

X	E	P	N	Z	V
-	-	-	-	-	-

フラグはすべて保持します。

このページは白紙です。

## (1) 入力形式

USND S
--------

S : ソース (通信識別子)。モジュールにより以下に示す管理番号を指定します。

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

TCP, UDPあわせて、1つのモジュールにつき同時に最大16までの通信が可能です。

OPTETモジュールは4モジュールの合計でTCP, UDPあわせて、最大32までの通信が可能です。

## (2) 機能

### ● UDP送信

USNDは、[イーサネット通信設定]画面で設定した「送信サイズ」、「送信アドレス」をもとに、送信バッファエリアの送信データを「送信データ」長分ソケットへ送信します。処理が完了しなくても命令は終了します。処理の結果は、システムレジスターと[イーサネット通信設定]画面で設定した「実行中フラグ」、「詳細結果コード」に格納されます。

実行中フラグ : 処理実行中は1、処理完了時に0が設定されます。処理が完了したとき、システムレジスター、「詳細結果コード」に処理の結果が設定されます。そのため、命令の実行後、処理が完了するまで「実行中フラグ」を監視してください。

詳細結果コード : 命令の処理が完了したとき、処理の結果が設定されます。「実行中フラグ」が0になった後、「詳細結果コード」を参照してください。

## (3) データタイプ

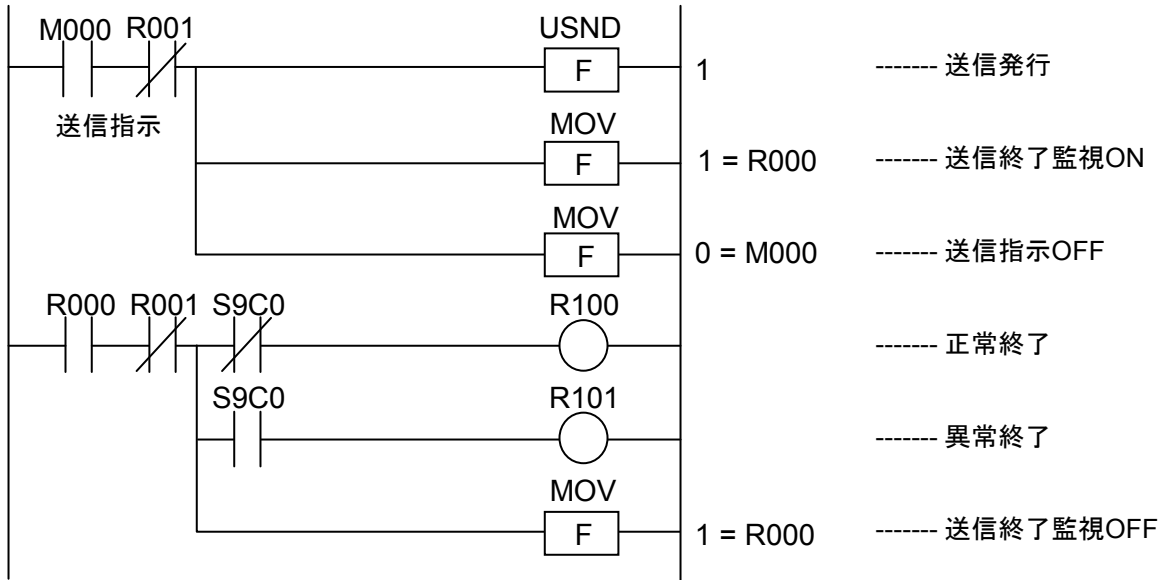
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	-	-	-	-	-

○ : 指定できます。

- : 指定できません。

(4) プログラム例

管理番号 = 1、実行中フラグ = R001で送信



(5) エラー処理

送信に成功すると、システムレジスタに0、「詳細結果コード」に0が設定されます。失敗した場合は、システムレジスタに1、「詳細結果コード」にエラーNo. (0以外の値) が設定されます。

処理の成功、失敗はシステムレジスタで判定してください。

失敗時のエラー要因は、「詳細結果コード」を参照してください。

● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

フラグはすべて保持します。



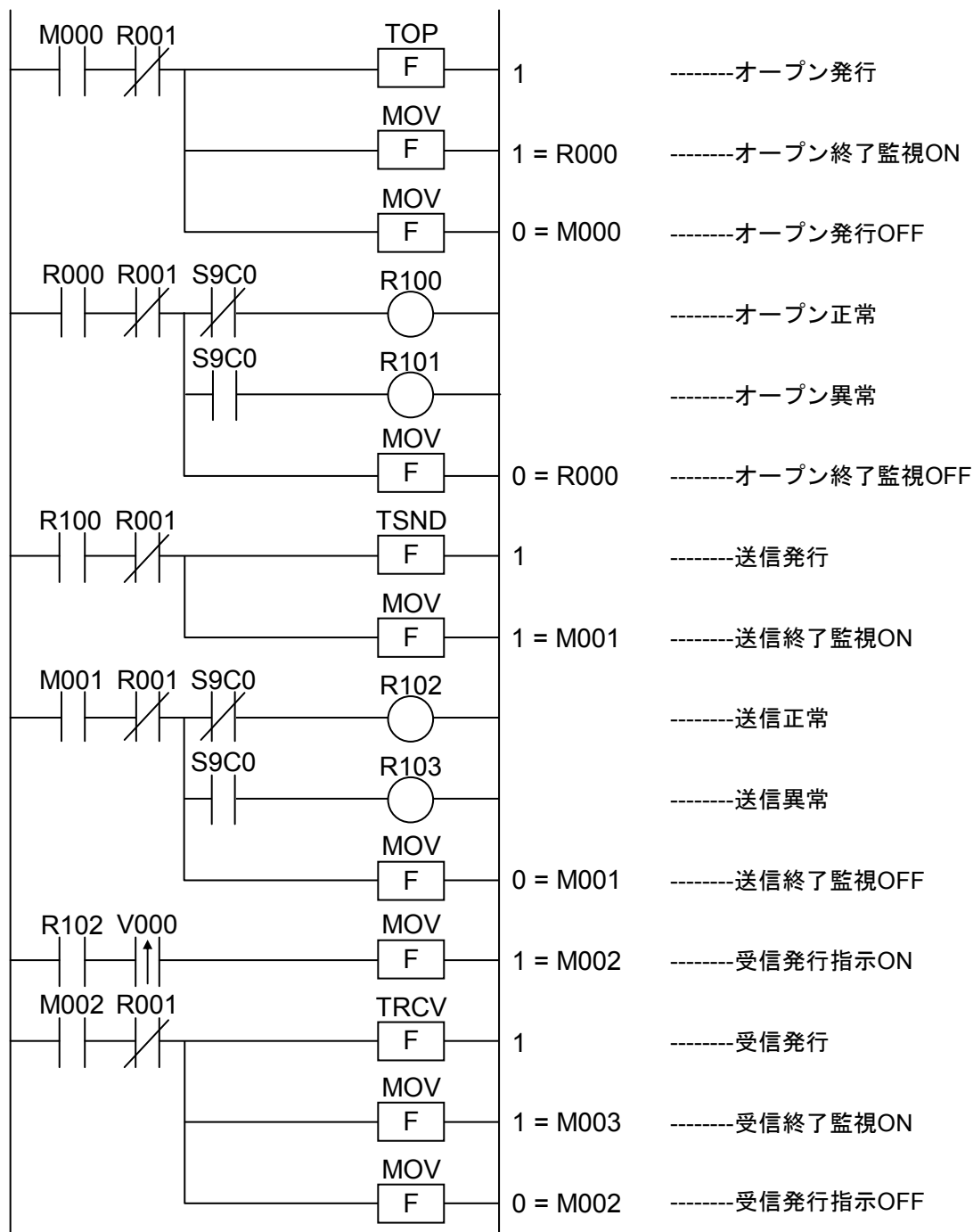
## 2 演算ファンクション

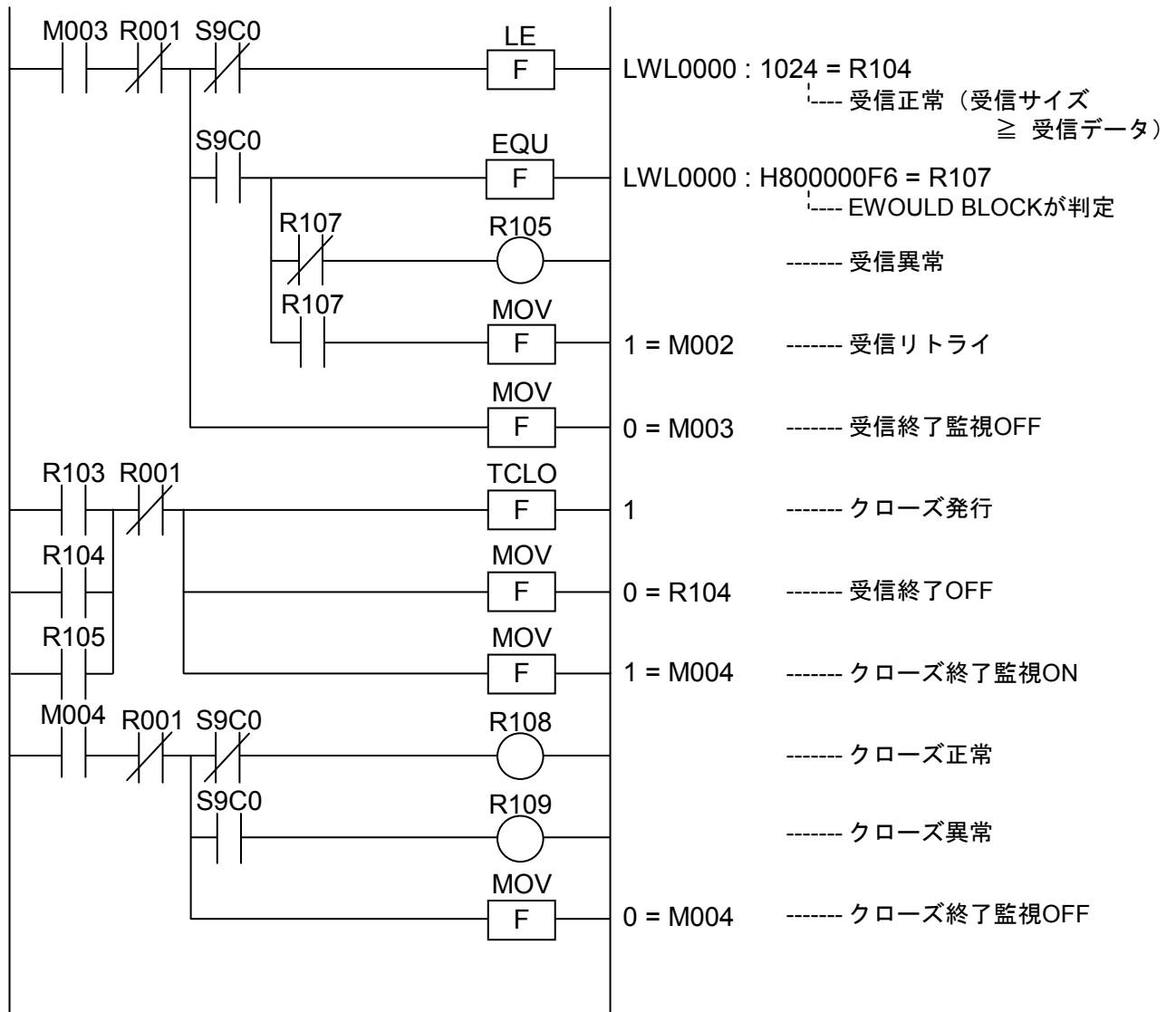
### 2.7.4 サンプルプログラム

イーサネット通信システム演算ファンクションを使用してソケットのオープン、送受信、ソケットのクローズするまでのサンプルプログラムを記載します。

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

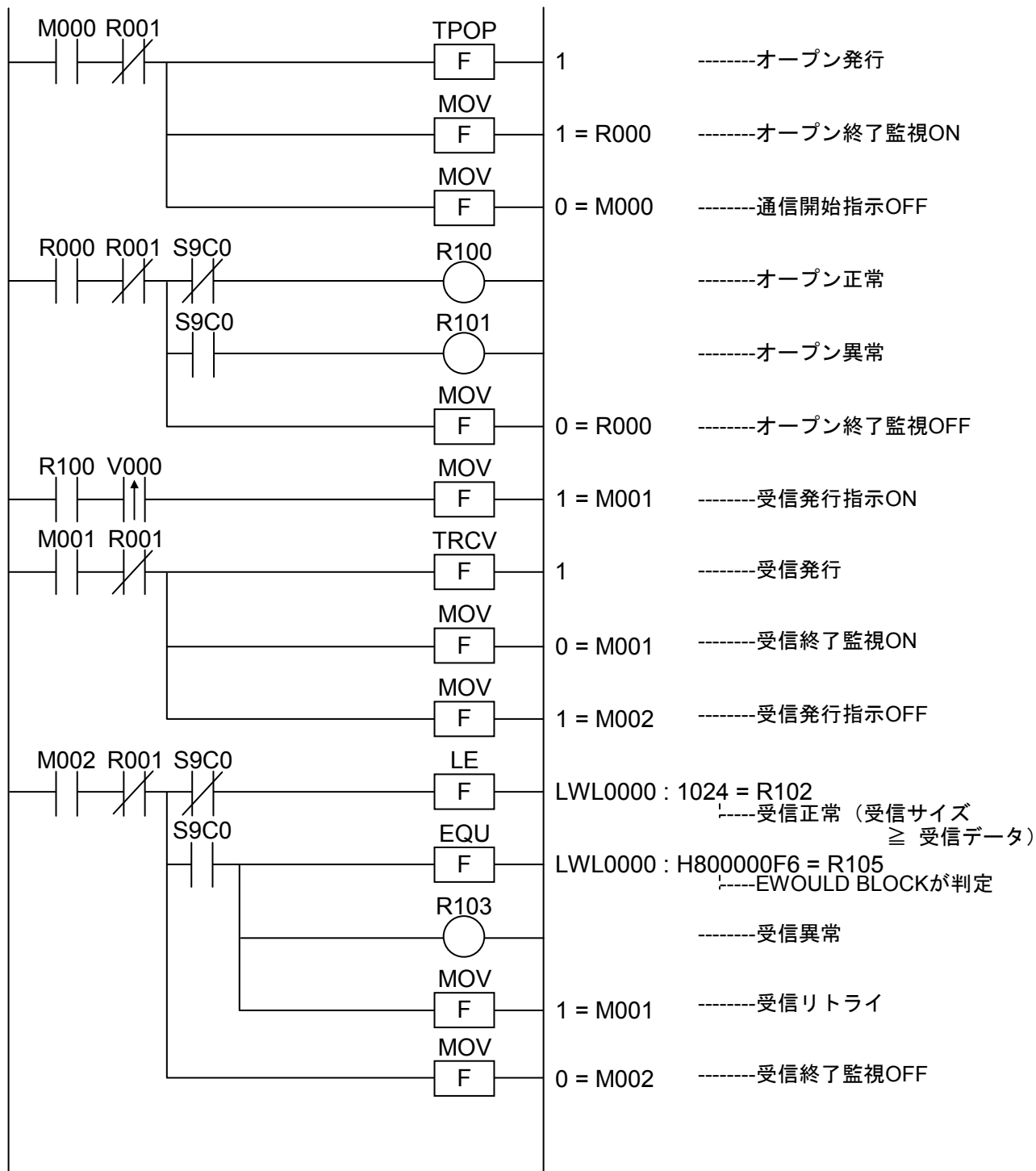
#### (1) TCPクライアント

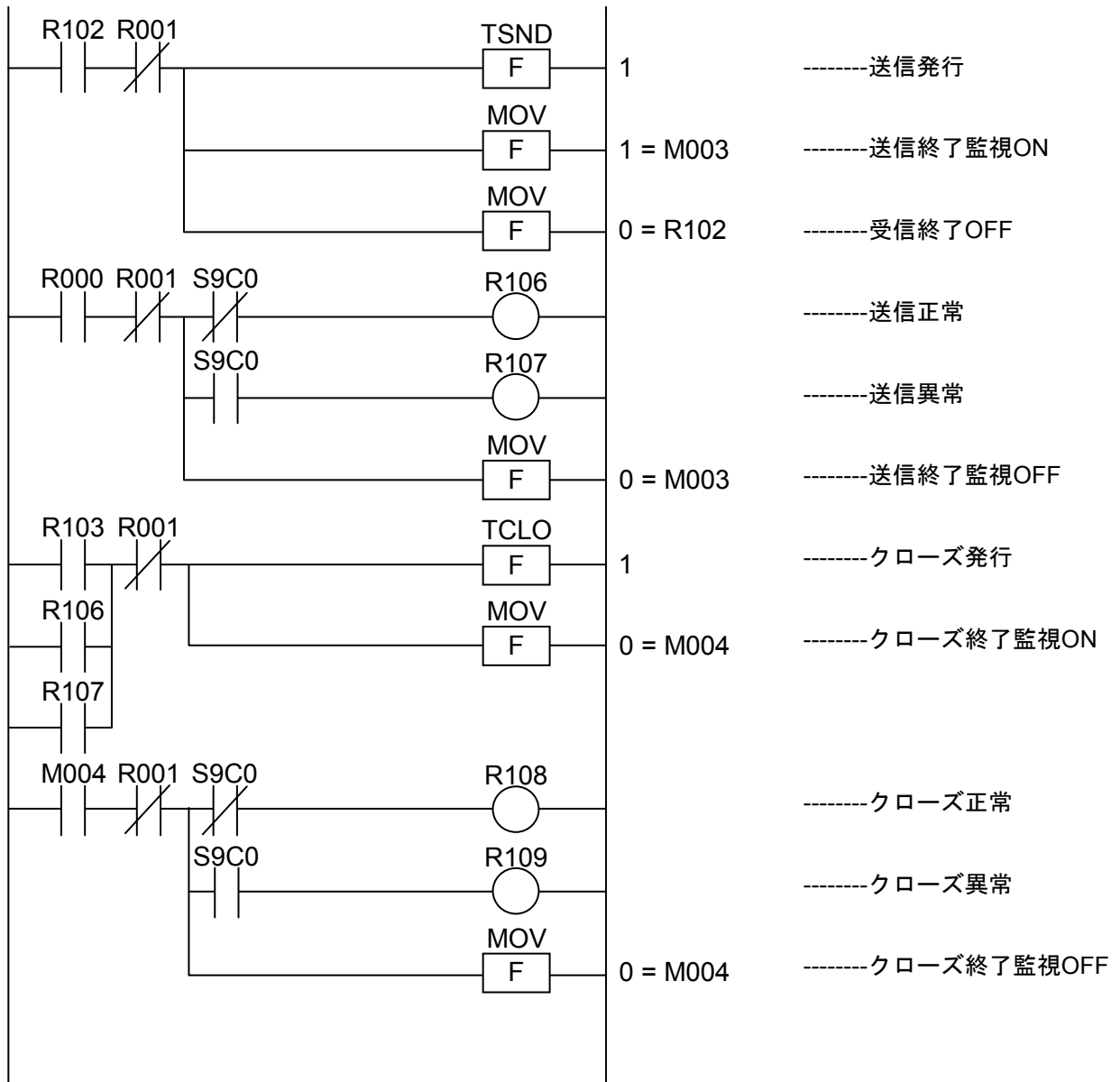




## 2 演算ファンクション

### (2) TCPサーバー





このページは白紙です。

補 足

## 補足 A 平均スキャンタイムの確認

ラダープログラムの平均スキャンタイムを確認する方法には、S10V ラダー図システム（型式：S-7895-02）のシーケンスサイクルモニター機能を使用する方法とラダープログラムに回路を追加する方法の2種類があります。

### A. 1 S10V ラダー図システムによる確認

S10V ラダー図システムのシーケンスサイクルモニター機能を使用すると、スキャンタイムの現在値、最大値、最小値、平均値を確認できます。

起動方法は、S10V ラダー図システムのメニューから [ユーティリティ] - [制御状態モニタ] - [シーケンスサイクルモニタ] を選択します。

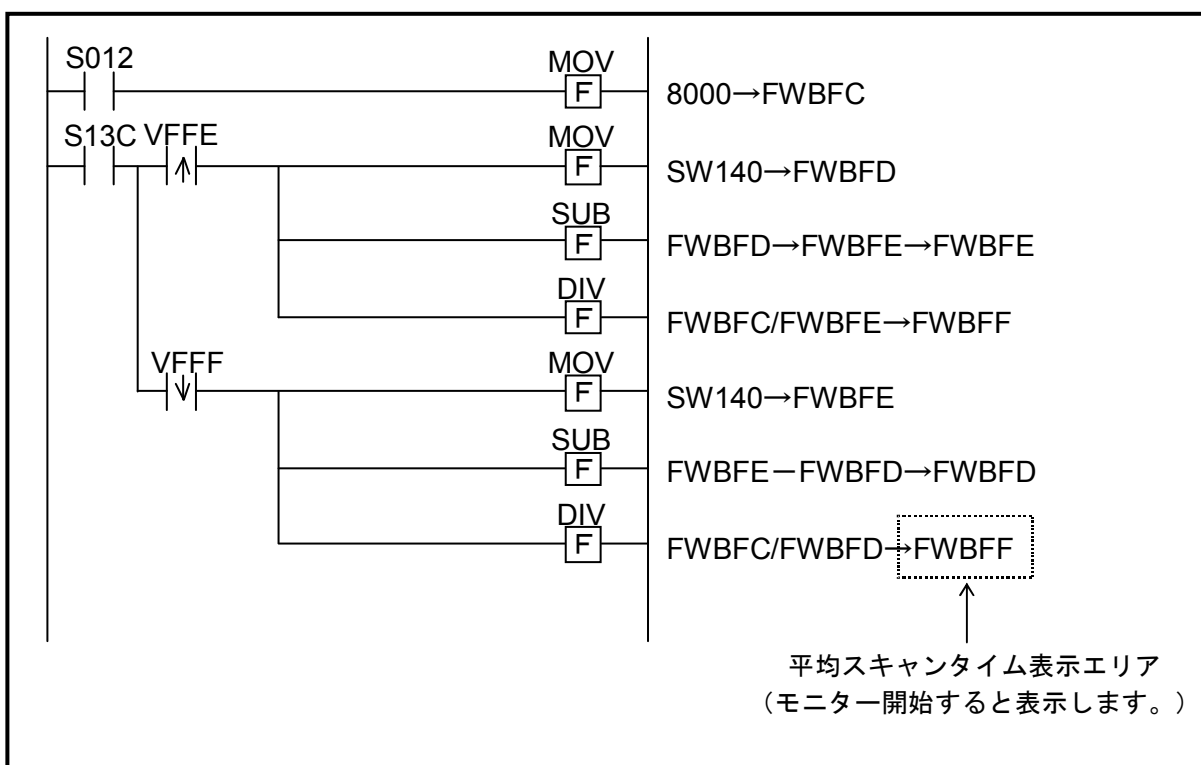
シーケンスサイクルモニターの操作方法は、「ソフトウェアマニュアル オペレーション S10V ラダー図 For Windows®（マニュアル番号 SVJ-3-131）」を参照してください。

< [シーケンスサイクルタイムモニタ] 画面 >



## A. 2 ラダープログラムによる確認

プログラム平均スキャンタイムは、以下に示す回路をプログラムに追加することにより確認できます。この値はユーザープログラムRUN時、8秒ごとの平均スキャンタイム（ms）を示し、FWBFFに格納します。



図A-1 スキャンタイム表示プログラム



## 補足 B 注意事項

## B. 1 ラダープログラムコンバート時の注意事項

S10V ラダー図システム（型式：S-7895-02）のコンバーター機能を使用すると、S10/2 $\alpha$ シリーズおよびS10miniシリーズの右下がりラダープログラムをS10Vの水平ラダープログラムに変換できます。以下にコンバート時の注意事項を示します。

コンバーター手順および機能については、「ソフトウェアマニュアル オペレーション S10V ラダー図 For Windows®（マニュアル番号 SVJ-3-131）」を参照してください。

- S10VにおいてS10/2 $\alpha$ シリーズ、S10miniシリーズで使用されていた下記演算ファンクションが廃止、変更になりました。コンバーター機能では、一部を除き自動でS10/2 $\alpha$ シリーズ、S10miniシリーズの演算ファンクションをS10Vの演算ファンクションへ変換しますのでユーザーは意識する必要はありません。

S10/2 $\alpha$ シリーズ、S10miniシリーズの演算ファンクション	S10Vの演算ファンクション	廃止・変更理由
ソースインダイレクト転送 (MSI) デスティネーションインダイレクト転送 (MDI)	一括転送 (MOM)	MOMにてインデックス指定（間接アドレッシング指定）が可能になったため
データセット (DST)	転送 (MOV)	MOVにて定数が指定可能になったため
平方根 (ROT)	平方根 (SQR)	名称変更
FIFO書き込み (PSH)	FIFO書き込み (PSHO)	S10/2 $\alpha$ , S10mini互換FIFOをサポートしたため （ただし、ラダー図システムのVer.-Rev. : 01-16以降）
FIFO読み込み (POP)	FIFO読み込み (POPO)	S10/2 $\alpha$ , S10mini互換FIFOをサポートしたため （ただし、ラダー図システムのVer.-Rev. : 01-16以降）

- S10Vは、S10/2 $\alpha$ シリーズ、S10miniシリーズに比べて処理性能が向上しています。そのため、ラダープログラムの処理時間が短くなり外部との入出力タイミングがS10/2 $\alpha$ シリーズ、S10miniシリーズとは異なる可能性があります。テストを十分に行ってから設備に使用してください。
- 演算ファンクションでアドレスをイミディエート値として指定している場合は、アドレスの変換をしません。水平ラダープログラムに変換後、アドレスを修正してください。特に拡張メモリーを使用していた場合は注意が必要です。

- LPUモジュールレビジョンL (Ver.-Rev. : 02-05) 以前のS10Vでは、ハードウェアの制限からロングワード型レジスタはロングワード境界でアクセスするレジスタ番号 (XL000, FL004など) を指定する必要があり、ワード境界指定 (XL010, FL003など) を行えません。コンバーター機能では、これらの変換をしていません。奇数番号のロングワード型レジスタがコンパイル時にエラーとして検出された場合、偶数番号のロングワード型レジスタとなるように修正してください (LPUモジュールレビジョンM (Ver.-Rev. : 02-06) 以降は下記の修正は不要です)。

[修正例]

(修正前)		(修正後)
XL010	→	XL000
FL003	→	FL004

ロングワード型レジスタに指定可能なナンバーを以下に示します。

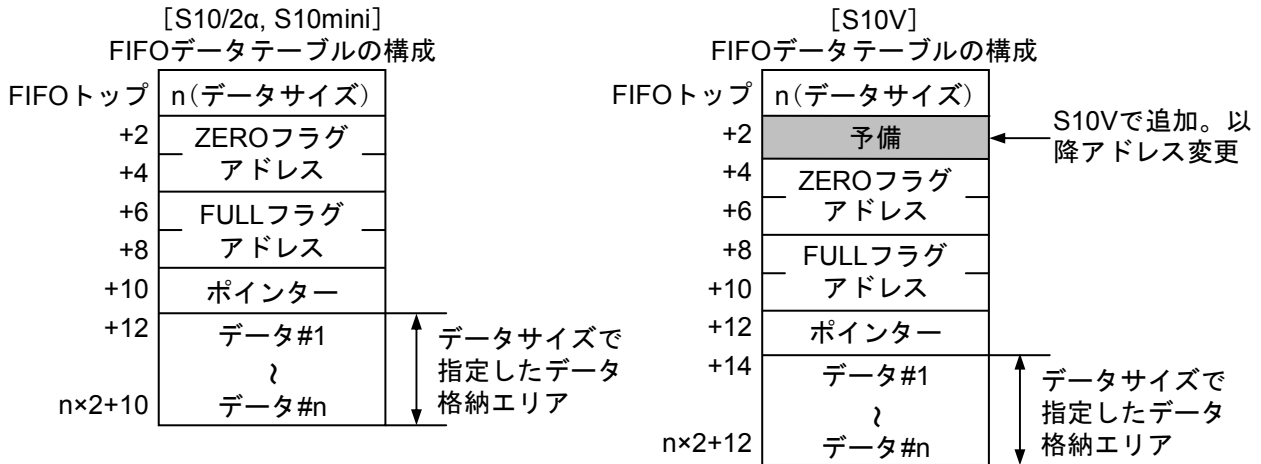
ロングワード型レジスタ指定可能ナンバー一覧 (モジュールレビジョンM (Ver.-Rev. : 02-06以降) のLPUは除く)

レジスタ種別	レジスタ名	指定可能ナンバー
I/Oレジスタ (ロングワード)	XL, JL, YL, QL, GL, AL, RL, ML, KL, TL, UL, CL, NL, PL, VL, EL, ZL, SL, LBL, LRL, LVL	XL0 <sup>□</sup> 0, LBL00 <sup>□</sup> 0のように、□の部分 部分が偶数の場合のみ入力可能
ワークレジスタ (ロングワード)	DL, FL, LWL, LXL	DL00 <sup>□</sup> , LWL000 <sup>□</sup> のように、□の部分 部分が偶数の場合のみ入力可能

## 通 知

Ver.-Rev. : 01-16以降のラダー図システムで作成した、奇数番号のロングワードレジスタまたはPSHO/POPOを含むラダープログラムを、Ver.-Rev. : 01-15以前のラダー図システムまたは一括ロードにてモジュールレビジョンL (Ver.-Rev. : 02-05) 以前のLPUに送信すると、LPUが“不当命令検出”にて停止します。

- FIFO書き込み（PSH）、FIFO読み出し（POP）のFIFOテーブル構成変更  
FIFOテーブル構成がS10/2αシリーズ、S10miniシリーズから変更になりました。



<LPUモジュールレビジョンM（Ver.-Rev.：02-06）以降の場合>

S10Vラダー図システム（Ver.-Rev.：01-16以降）のコンバーター機能では、演算ファンクションPSH/POPからPSHO/POPOに変換し、上記の [S10/2α, S10mini] のFIFOデータテーブルが使用できません。

<LPUモジュールレビジョンL（Ver.-Rev.：02-05）以前の場合>

S10Vラダー図システム（Ver.-Rev.：01-16以降）のコンバーター機能では、演算ファンクションPSH/POPからPSHO/POPOに変換しますが、この変換命令は使用できませんので、LPUモジュールをモジュールレビジョンがM（Ver.-Rev.：02-06）以降のものに交換するか、ラダープログラムをPSH/POPに修正し、上記 [S10V] のFIFOデータテーブルを使用してください。