

**HITACHI**

ソフトウェアマニュアル

プログラミング

ラダー図 For Windows<sup>®</sup>

---

***S10VE***

SEJ-3-121(A)

ソフトウェアマニュアル

プログラミング

ラダー図 For Windows®

---

***S10VE***

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

2018年 8月 (第1版) SEJ-3-121 (A)

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

## 安全上のご注意

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

### 警告表示の定義



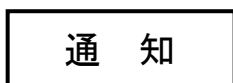
: この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こす危険の存在を示す。



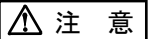
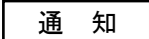
: この表示を無視して誤った取り扱いをすると、死亡または重大な傷害を引き起こすおそれのある危険の存在を示す。



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



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

なお、 **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。

## 通 知

- 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。

(1-35ページ)
  
- マスターコントロールでは、1つのラダーシート上に同一のNコイルを複数配置して、NコイルのON/OFFが混在する場合、以下の2つの問題があります。
  - ・ エッジ接点は、常に立ち上がりを検出、またはOFFのままで立ち下りが検出されない場合があります。そのため、Nコイルを複数配置した場合は、エッジ接点を使用しないでください。
  - ・ 1シーケンス内でNコイルの状態によりコイルがON/OFFします。途中でコイルがONしても最後に配置されたNコイルが未実行の場合、回路モニター、MCSなどでコイルの値を参照したとき、OFFに見えますので注意してください。

(1-39ページ)
  
- 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。

(1-45ページ)

## 来歴一覧表

改訂No.	来歴（改訂内容）	発行年月	備考
A	新規作成	2018.8	

## はじめに

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。

### <関連マニュアル>

S10VE ソフトウェアマニュアル オペレーション ラダー図 For Windows®

(マニュアル番号 SEJ-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

第1章 ラダー命令	1-1
1. 1 ラダープログラム	1-1
1. 2 ラダープログラムの動作順序	1-4
1. 3 ラダープログラム命令一覧	1-5
1. 3. 1 ラダープログラム命令一覧	1-5
1. 3. 2 a接点	1-8
1. 3. 3 b接点	1-9
1. 3. 4 立ち上がり／立ち下がりエッジ接点	1-10
1. 3. 5 演算結果push, read, pop	1-11
1. 3. 6 演算結果push+a接点, read+a接点, pop+a接点	1-11
1. 3. 7 演算結果push+b接点, read+b接点, pop+b接点	1-12
1. 3. 8 ブロック結合（並列接続）（ORB）	1-12
1. 3. 9 NOT	1-13
1. 3. 10 コイル	1-14
1. 3. 11 セットコイル／リセットコイル	1-14
1. 3. 12 比較命令	1-15
1. 3. 13 ラダー命令のインデックス指定方式	1-18
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	1-65
1. 6. 1 ラダーウォッチドッグタイマー動作概要	1-65
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	1-66
1. 6. 3 ラダーWDTエラー発生時のエラー情報	1-66
第2章 演算ファンクション	2-1
2. 1 機能概要	2-1
2. 2 機能仕様	2-3
2. 3 演算ファンクションで使用するレジスター	2-7
2. 3. 1 演算ファンクションで使用するレジスター	2-7
2. 3. 2 ビットレジスターの扱い	2-10
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	2-11
2. 4 演算ファンクション入力	2-12
2. 5 演算ファンクション一覧表	2-17

2. 6	命令の詳細 .....	2-22
2. 7	イーサネット通信命令 .....	2-171
2. 7. 1	機能概要 .....	2-171
2. 7. 2	使用方法 .....	2-173
2. 7. 3	命令の詳細 .....	2-182
2. 7. 4	サンプルプログラム .....	2-203

## 補足

補足A	平均スキャンタイムの確認 .....	Z-1
A. 1	LADDER DIAGRAM SYSTEM/S10VEによる確認 .....	Z-1
A. 2	ラダープログラムによる確認 .....	Z-2

## 図目次

図 Z-1	スキャンタイム表示プログラム .....	Z-2
-------	----------------------	-----

## 表目次

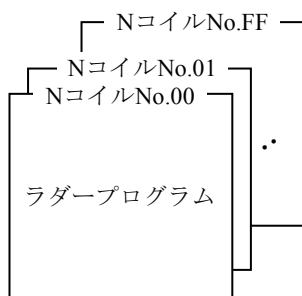
表 1-1	ラダー命令（基本命令）一覧 .....	1-5
表 1-2	ラダー命令（比較命令）一覧 .....	1-7
表 1-3	演算ファンクション命令一覧 .....	1-7
表 1-4	使用できるレジスター一覧 .....	1-22
表 1-5	レジスターナンバー一覧 .....	1-24
表 1-6	システムレジスター一覧 .....	1-46
表 2-1	演算ファンクションで使用できるレジスター一覧 .....	2-7

## 第1章 ラダー命令

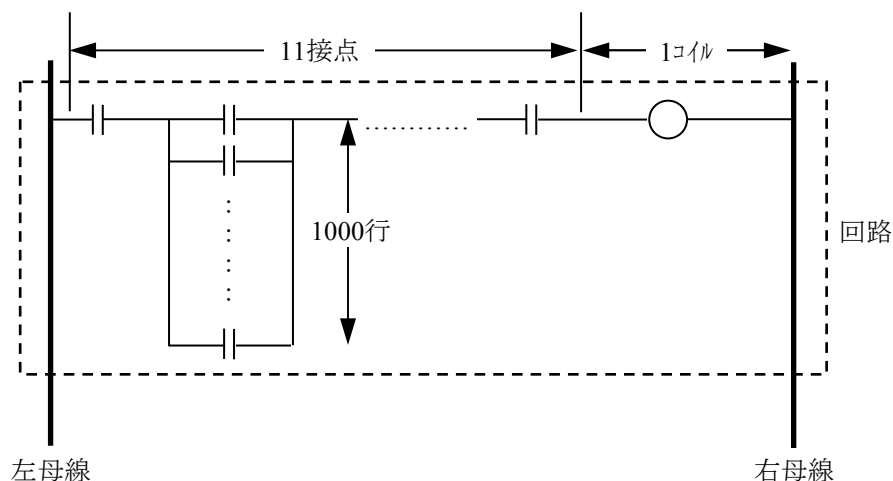
### 1. 1 ラダープログラム

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。

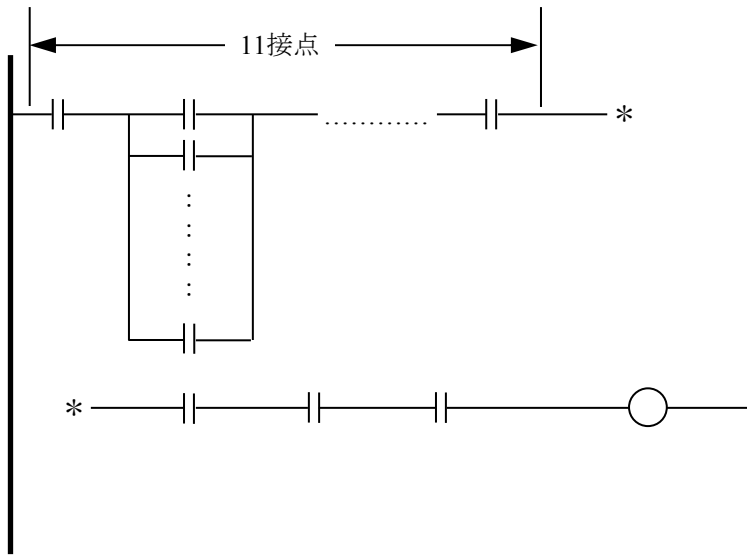
どのラダープログラムも、ネスティングコイル（以下、Nコイル）と呼ばれる1つ以上のプログラミング単位で構築します。1つのラダープログラム内では、Nコイルは00～FF（計256個）作成できます。NコイルNo.00はマスターコイルと呼ばれ、ラダープログラム実行時にシーケンスサイクルごとに必ず主ルーチンとして実行されます。NコイルNo.01～FFはサブNコイルと呼ばれ、マスターNコイルから、または他のサブNコイルからサブルーチンとして起動されます。



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。

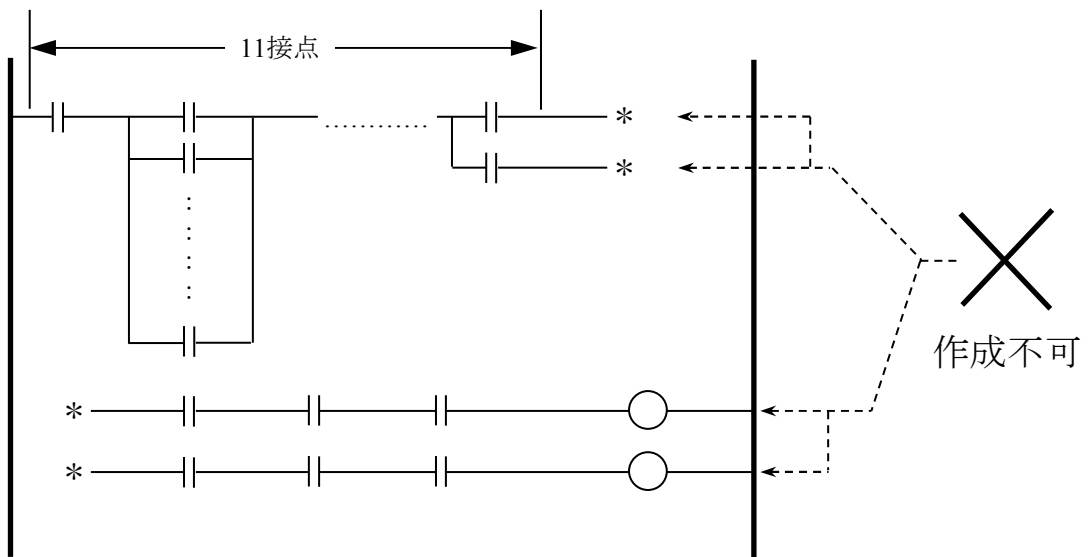


11接点以上をAND接続する場合、図に示すような折り返し回路が作成できます。ただし、下記の制限事項があります。

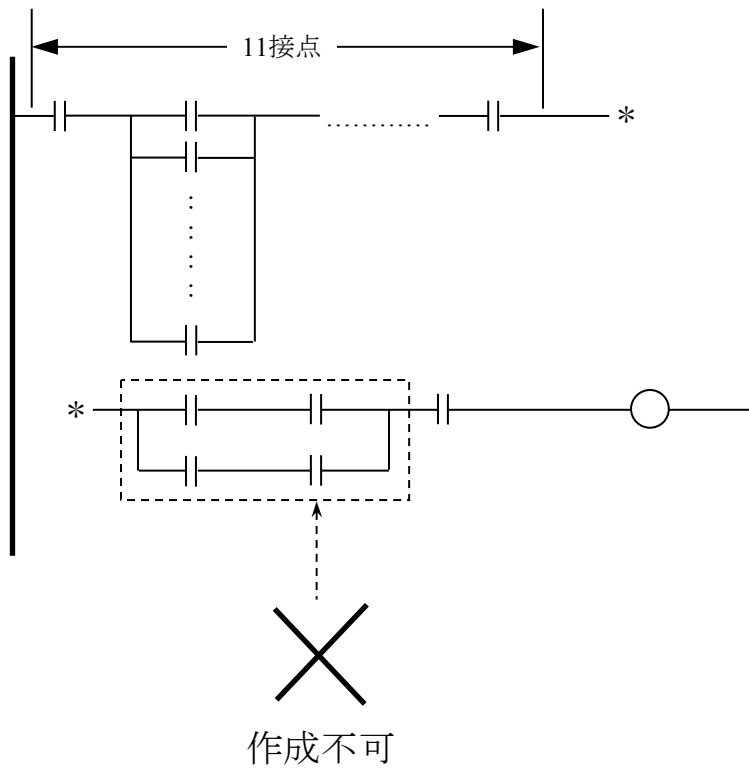


<制限1> 並列のAND接続不可

\*印の前に分岐回路は作成できません。



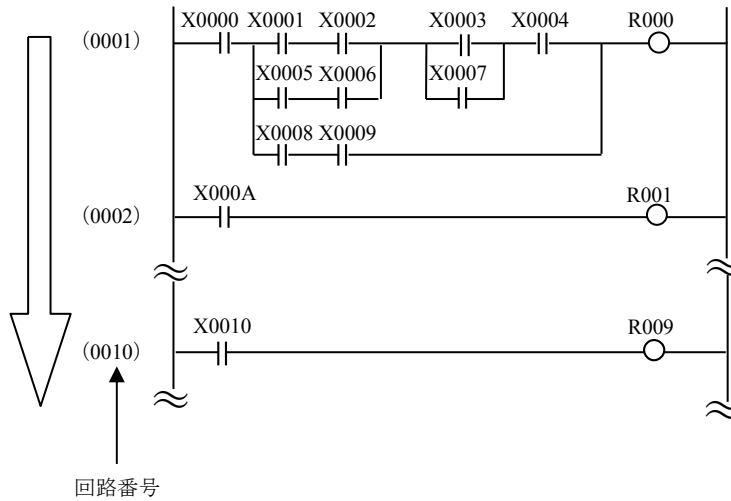
<制限2> \*印の後ろに分岐回路は作成できません。



1.2 ラダープログラムの動作順序

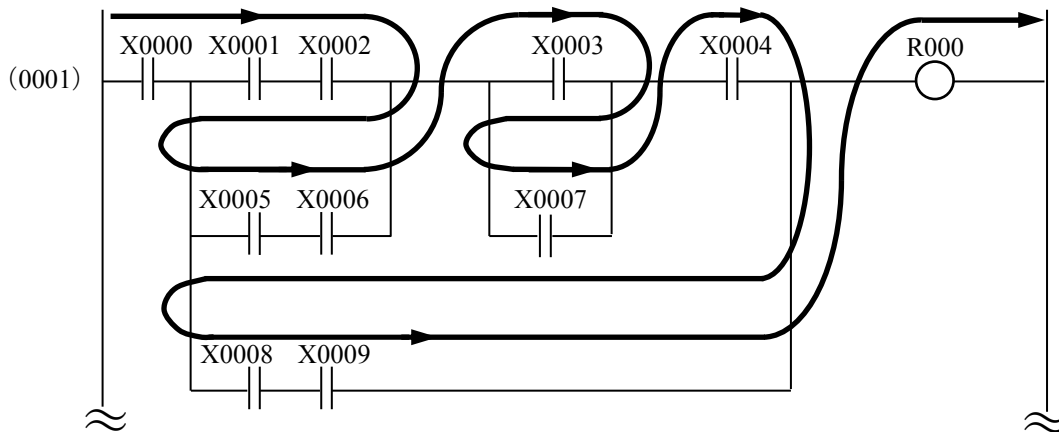
ラダープログラムは、回路番号の順番（昇順）に従って動作します。

【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. 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)












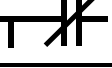


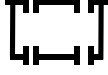
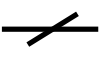

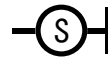
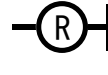
命令名称	記号	オペコード	機能
a接点開始		LD	a接点の開始を意味します。 (a接点：指定したレジスター値が1のときON)
a接点 直列接続		LAND	a接点と直前の命令との直列接続を意味します。
b接点開始		LDN	b接点の開始を意味します。 (b接点：指定したレジスター値が0のときON)
b接点 直列接続		LANDN	b接点と直前の命令との直列接続を意味します。
立ち上がり エッジ接点		EGP	入力の立ち上がりエッジを検出した1シーケンスサイクルだけONします。
立ち下がり エッジ接点		EGF	入力の立ち下がりエッジを検出した1シーケンスサイクルだけONします。
演算結果push		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。
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。



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		LNOT	入力を反転して出力します。
コイル		OUT	指定したレジスターに出力します。 指定したレジスター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：ネステイングコイル、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。

## (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/ロングワード/フローティング指定したレジスターを比較し、条件成立でON、非成立でOFFを出力します。 <ul style="list-style-type: none"> <li>比較データに定数を指定できます。</li> <li>定数、変数 (レジスター内容) とともに最上位ビットをサインビットとみなして比較します。</li> </ul> ただし、ロングワード/フローティング指定したレジスターの奇数ワード指定は行えません。 (例: XL0010、FL001)
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 = 1~54ステップになります。

演算ファンクション命令の詳細は、「第2章 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

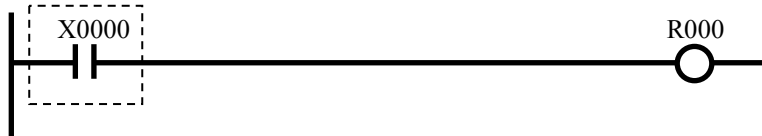
命令名称	記号	オペコード	機能
演算ファンクション		-	ワード/ロングワード/フローティング指定したレジスター/定数を用いて各種演算命令を実行します。

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。

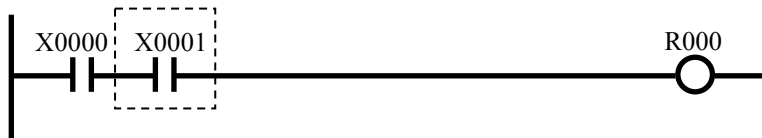
(1) a接点開始 (LD)

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、X0000がONのときにR000がONします。



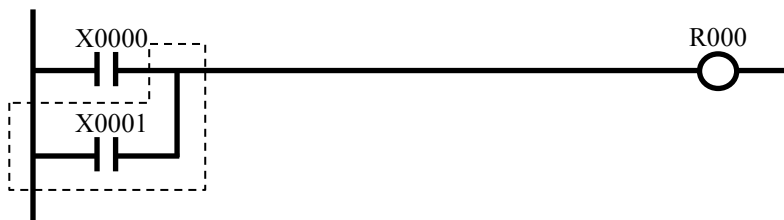
(2) a接点直列接続 (LAND)

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、X0000がON、かつX0001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、X0000またはX0001がONのときにR000がONします。

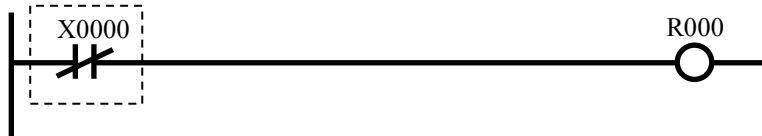


## 1. 3. 3 b接点

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。

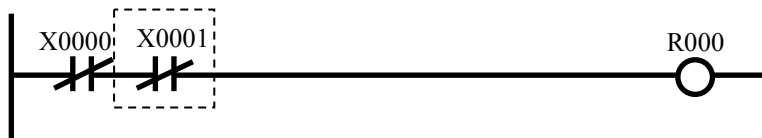
## (1) b接点開始 (LDN)

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、X0000がOFFのときにR000がONします。



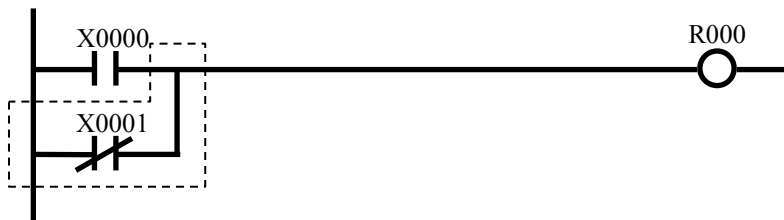
## (2) b接点直列接続 (LANDN)

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、X0000がOFF、かつX0001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、X0000がONまたはX0001がOFFのときにR000がONします。



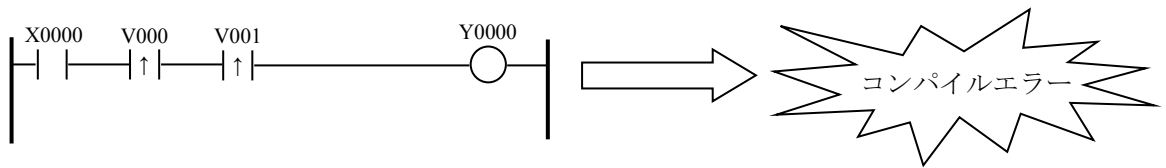
1. 3. 4 立ち上がり／立ち下がりエッジ接点

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。

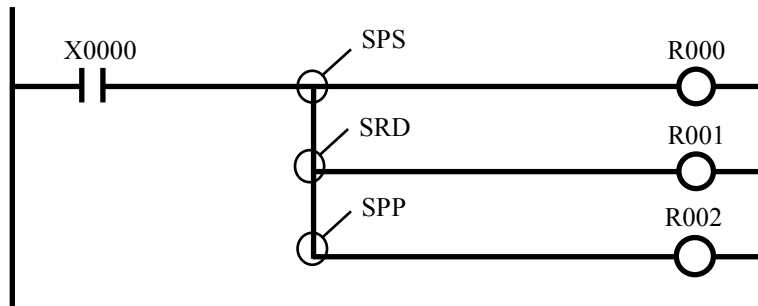
詳細は、「1. 5 レジスター一覧」の「V エッジ接点」を参照してください。

(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. 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。

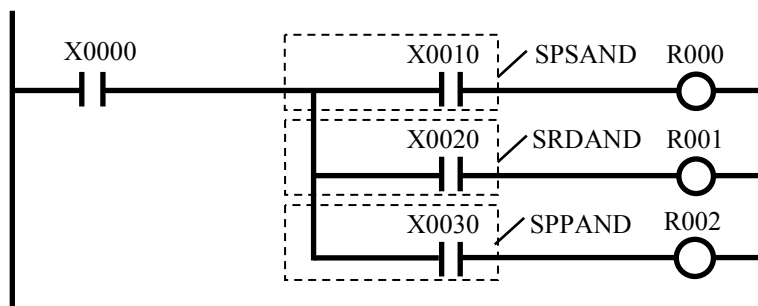
## (3) 演算結果pop (SPP)

・ 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、X0000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。

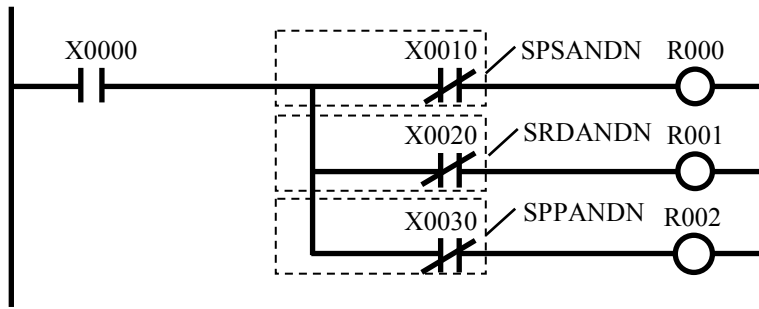
## (3) 演算結果pop+a接点 (SPPAND)

・ 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、X0000がON、かつX0010がONのときにR000がONします。X0000がON、かつX0020がONのときにR001がONします。X0000がON、かつX0030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)

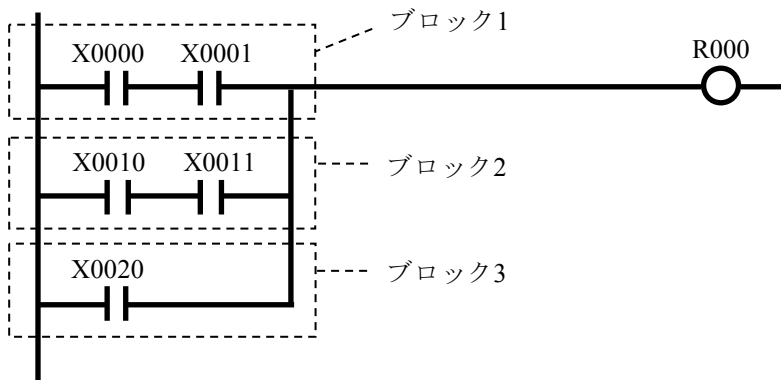
- ・SPS, SPSAND、またはSPSANDNで記憶した演算結果を読み出し、読み出した演算結果で次のb接点を演算します。

- ・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、X0000がON、かつX0010がOFFのときにR000がONします。X0000がON、かつX0020がOFFのときにR001がONします。X0000がON、かつX0030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)

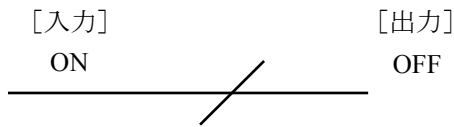
ブロック結合は、ブロックが複数ある回路のブロック間のOR演算を行います。下記回路例では、ブロック1~ブロック3のいずれかのブロックの演算結果がONの場合、R000がONします。



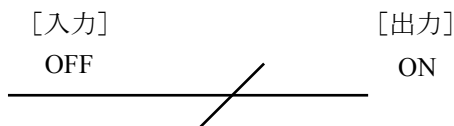
1. 3. 9 NOT

入力を反転して出力します。

<入力がONの場合>

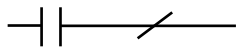


<入力がOFFの場合>

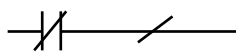


入りに指定できる命令は、a接点、b接点、エッジ接点、比較、並列接続のいずれか、または入力シンボルなしとなります。

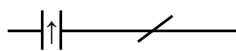
<a接点の結果を反転する場合>



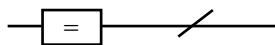
<b接点の結果を反転する場合>



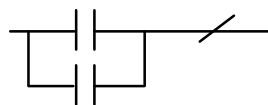
<エッジ接点の結果を反転する場合>



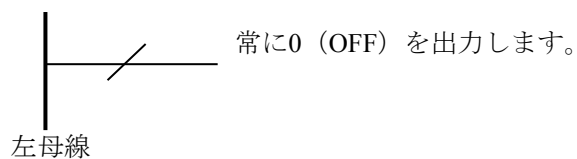
<比較の結果を反転する場合>



<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>

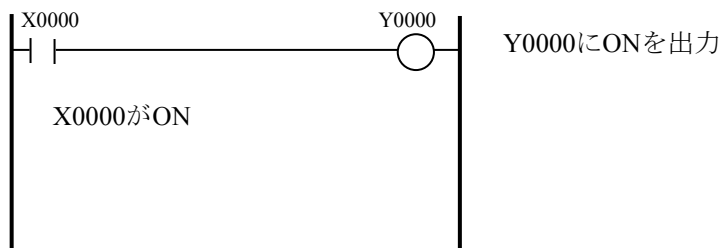




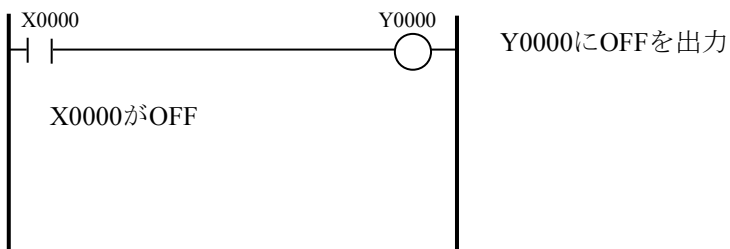
1. 3. 10 コイル

指定したレジスターに、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レジスター）、ワンショット（Uレジスター）、カウンター（Cレジスター）を指定した場合の動作は、「1. 5 レジスター一覧」の「T オンディレイタイマー」、「U ワンショット」、「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 レジスター一覧」の「K キープリレー」を参照してください。

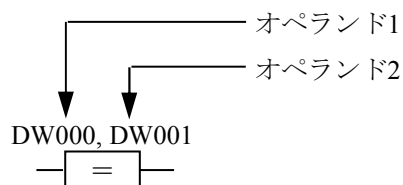
## 1. 3. 12 比較命令

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)

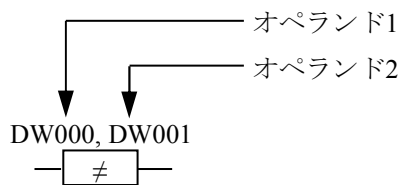
オペランド1とオペランド2の値が一致した場合、出力がONになります。不一致の場合は、OFFになります。



オペランドには、ワード/ロングワード/フローティング型のレジスターと定数が指定できます。 (\*)

## (2) 比較 (NEQ)

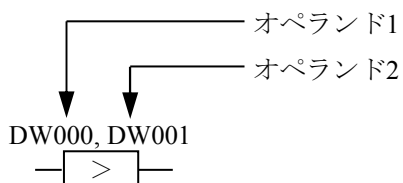
オペランド1とオペランド2の値が不一致の場合、出力がONになります。等しい場合は、OFFになります。



オペランドには、ワード/ロングワード/フローティング型のレジスターと定数が指定できます。 (\*)

## (3) 比較 (GT)

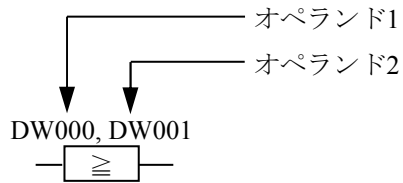
オペランド1がオペランド2より大きい場合、出力がONになります。それ以外の場合は、OFFになります。



オペランドには、ワード/ロングワード/フローティング型のレジスターと定数が指定できます。 (\*)

(4) 比較 (GE)

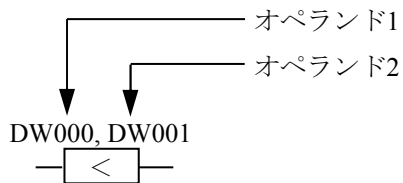
オペランド1がオペランド2より大きいかまたは等しい場合、出力がONになります。それ以外の場合は、OFFになります。



オペランドには、ワード/ロングワード/フローティング型のレジスターと定数が指定できます。 (\*)

(5) 比較 (LT)

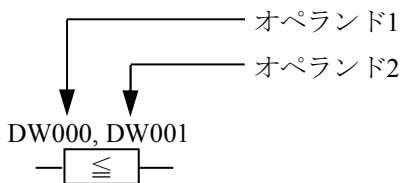
オペランド1がオペランド2より小さい場合、出力がONになります。それ以外の場合は、OFFになります。



オペランドには、ワード/ロングワード/フローティング型のレジスターと定数が指定できます。 (\*)

(6) 比較 (LE)

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



オペランドには、ワード/ロングワード/フローティング型のレジスターと定数が指定できます。 (\*)

(\*) 定数はオペランド2のみ指定できます。

定数（整数）の範囲はワード型の場合-32768～32767になります。

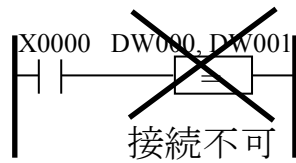
ロングワード型の場合-2147483648～2147483647になります。

浮動小数点の範囲は±2-126～±2128になります。

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。

(注1) 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>



(注2) 浮動小数点でオペランドの値が、非正規化数(0.0を除く-1.175494E-038(0x807FFFFFFF)～1.175494E-038(0x007FFFFFFF))の場合は、0.0として演算します。

(注3) ロングワードレジスター/フローティングレジスターの奇数ワード指定は行えません。

例：XL0010、FL001

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。

ベースレジスターに指定できるレジスターの型は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ーに指定できるレジスターの型はワード型のみです。

(例) X0020 (FW000)

X0020 (FW000) のとき、FW000の内容をH0020としたとき、X0020 + H0020 -> X0040を示します。

(注1) FW000の内容が、H0FF0やH1200などX0020に加算してXFFFF (Xの最大値) を超えるとき、他のレジスターをアクセスする場合があるため、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

比較命令の場合は、下記になります。

実行レジスターのアドレス = 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 × H0010  
(16進数)

(例) XW0000 (FW001)

FW001がH0040の場合、下記になります。

0000 (ベースレジスターの番号) + H0040 (インデックスレジスターの内容) × H0010 =  
XW0400

[コンパイル時の注意点]

コイルに使用するベースレジスターが同一である場合には、たとえインデックスの指定が異なる場合でも同一のコイルとみなしてコンパイルエラーとなります。

## [制限事項]

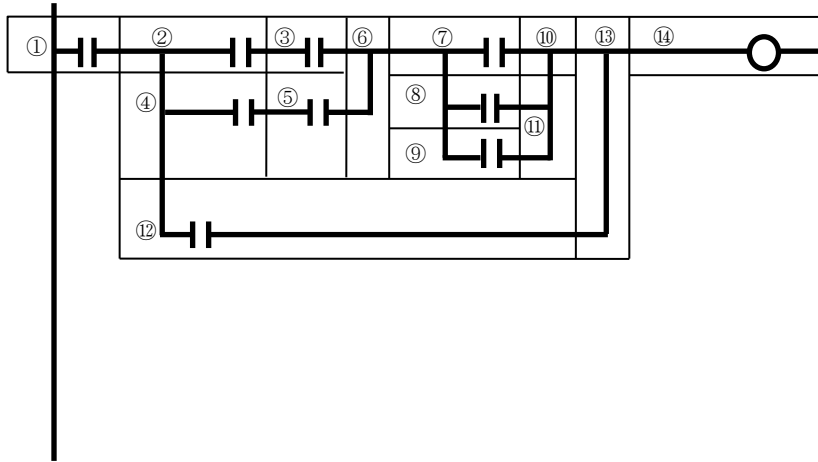
コイルに下記レジスター名を使用した場合は、入力時エラーになりインデックス指定はできません。

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

## 第1章 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	クリア	保持



第1章 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	—(*1)	—	—(*1)	—	—(*1)	—(*1)	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		
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		

● : 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
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
ファンクション データレジスタ	BW(*2)	—	—	—	—	—	●	エリア依存
	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

(\*1) Zレジスタはコンパイル時エラーとなりませんが使用しないでください。

(\*2) 間接アドレス方式でアクセスします。

Zレジスタはコンパイル時エラーとなりませんが使用しないでください。

## 第1章 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ナンバーの一覧を表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	X0000~XFFFF	XW0000~XWFFF0	XL0000~XLFFE0	—
2	Y	Y0000~YFFFF	YW0000~YWFFF0	YL0000~YLFFE0	—
3	R	R000~RFFF	RW000~RWFF0	RL000~RLFEO	—
4	M	M0000~MFFFF	MW0000~MWFFF0	ML0000~MLFFE0	—
5	A	A000~AFFF	AW000~AWFF0	AL000~ALFE0	—
6	K	K000~KFFF	KW000~KWFF0	KL000~KLFE0	—
7	T	T000~T7FF	TW000~TW7F0	TL000~TL7EO	—
8	TS	—	TS000~TS1FF	—	—
9	TC	—	TC000~TC1FF	—	—
10	U	U000~U0FF	UW000~UW0F0	UL000~UL0EO	—
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~CL0EO	—
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~NL0EO	—
23	P	P001~P080	PW000~PW080	PL000~PL060	—
24	E	E0000~EFFFF	EW0000~EWFFF0	EL0000~ELFFE0	—
25	V	V000~VFFF	VW000~VWFF0	VL000~VLFE0	—
26	Z (*1)	—	—	—	—
27	S	S0000~SBFFF	SW0000~SWBFF0	SL0000~SLBFE0	—
28	J	J000~JFFF	JW000~JWFF0	JL000~JLFE0	—
29	Q	Q0000~QFFFF	QW0000~QWFFF0	QL0000~QLFFE0	—

— : アクセスできません。

表1-5 レジスターナンバー一覧

(2/2)

No.	レジスター名	アクセス			
		ビット	ワード	ロングワード	単精度浮動小数点
30	LB	LB0000~LBFFFF	LBW0000~LBWFFF0	LBL0000~LBLFFE0	—
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 (*2)	—	BW000~BW1FE	BL000~BL1FE	—
37	DW	—	DW000~DWFFF	DL000~DLFFE	—
38	FW	—	FW000~FWBFF	FL000~FLBFE	—
39	LW	—	LWW0000~LWWFFFF	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。

(\*1) Zレジスタはコンパイル時エラーとなりませんが使用しないでください。

(\*2) 間接アドレス方式でアクセスします。

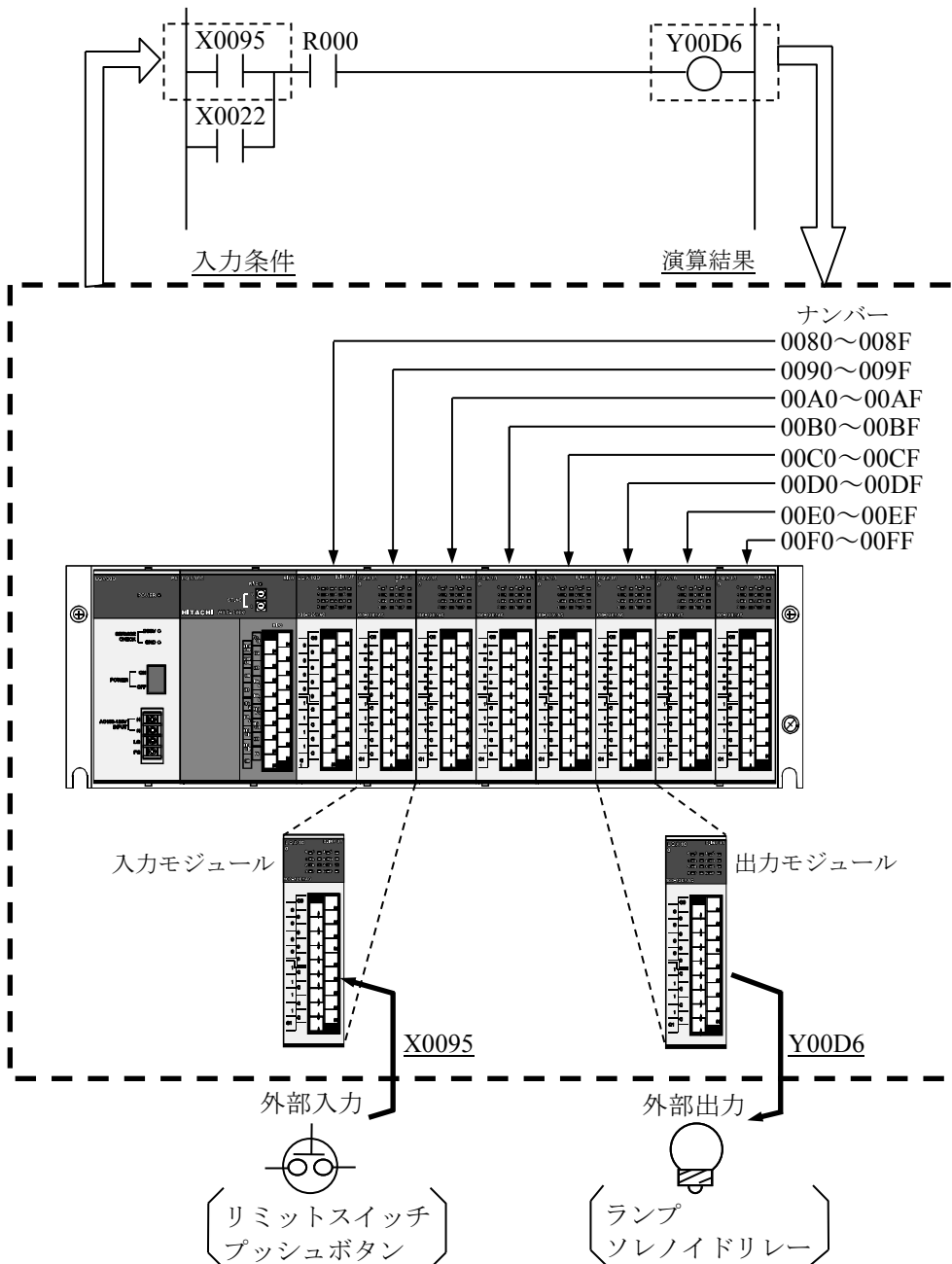
X, Y 外部入出力

ナンバーの 範囲	000~FFFF
リモートI/O の入出力範囲	0000~07FF

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X0095がONのとき、出力モジュールのY00D6に対して信号を出力します。

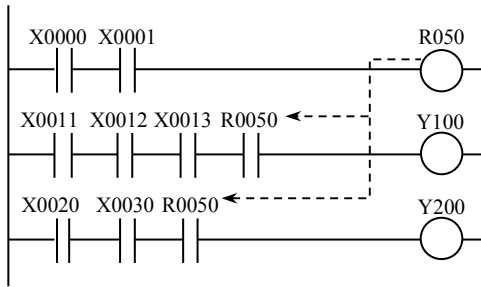


R, M, A, LB 内部レジスタ

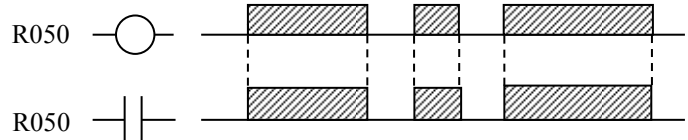
レジスタ名	R, A	M	LB
ナンバーの範囲	000~FFF	0000~FFFF	0000~FFFF

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。

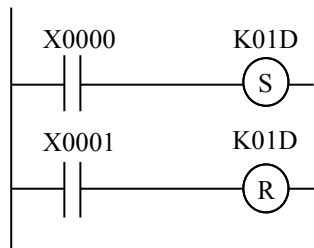
K キープリレー

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	プログラムの方がある方が 優先されます。

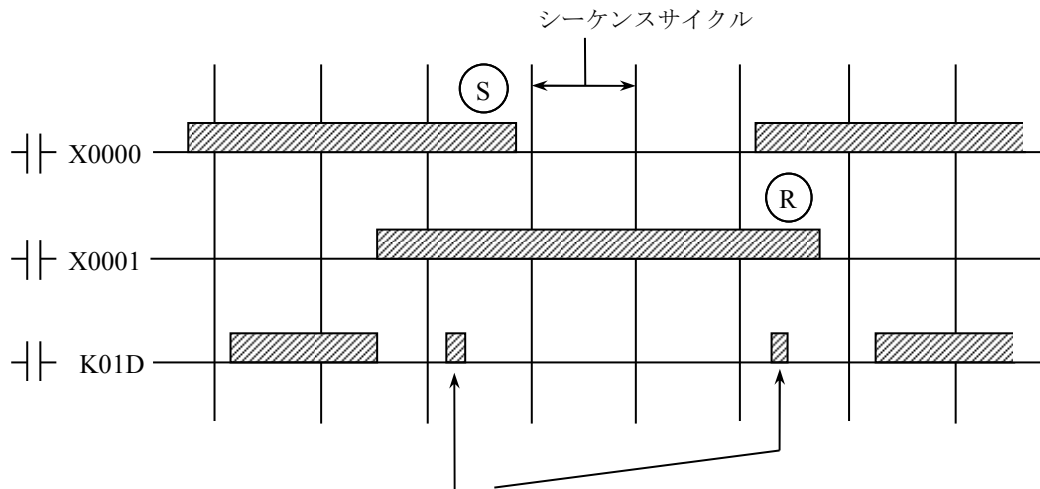
キープリレーは、セットコイルがONすると、リセットコイルがONするまで接点がONし、接点の状態は停電保持されます。また、セットコイルとリセットコイルが同時にONした場合、プログラムの下方にある方が優先されます。

(1) リセット優先回路

● 回路例



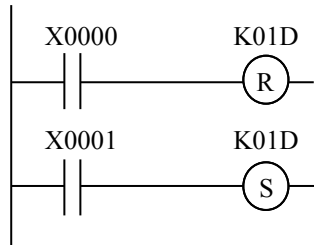
● タイムチャート



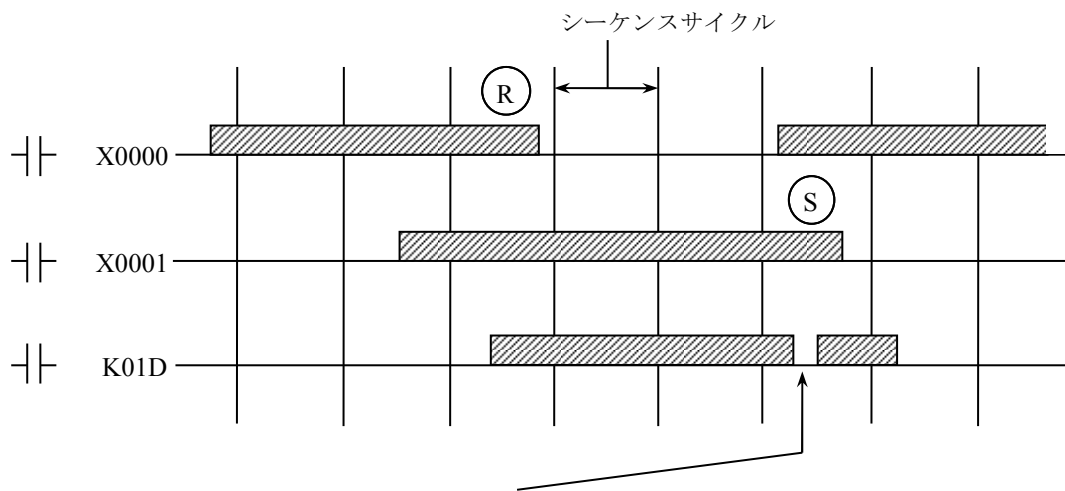
セット、リセットが同時ONの場合、シーケンスサイクル内で (S) が実行され、(R) が実行されるまでの間、キープリレー接点がONします。

(2) セット優先回路

● 回路例



● タイムチャート



セット、リセットが同時ONの場合、シーケンスサイクル内で (R) が実行され、(S) が実行されるまでの間、キーブリレー接点がOFFします。



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	000～1FFまたは000～7FF	000～00F
設定値	0～65535 (0.0～6553.5秒)	0～65535 (0.0～655.35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 (*1)	最小100ms	最小10ms

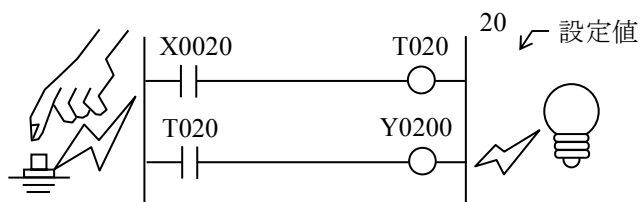
(\*1) セットリングパルス幅とは、オンディレイタイマーのコイルをONする接点がONしていなければならない最小時間のことを意味します。

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、0.0～6553.5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。

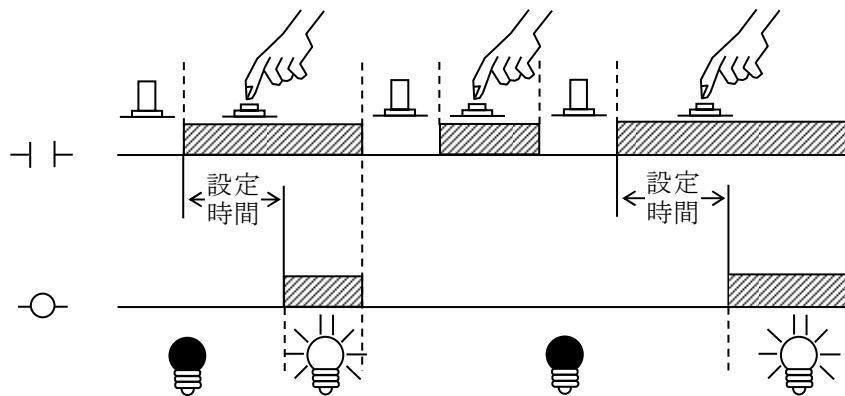
設定は、S10VE用プログラミングソフトウェア LADDER DIAGRAM SYSTEM/S10VE (型式 : S-7898-02) の [ユーティリティ] - [PCsエディション] - [PCsエディション]で行います。操作方法は、「S10VEソフトウェアマニュアル オペレーション ラダー図 For Windows® (マニュアル番号 SEJ-3-131)」を参照してください。

● 回路例



図に示した回路の場合、押しボタン (X0020) を押してから、設定時間後 (この場合2秒後) にランプ (Y0200) 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。
- ・ 計数值は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。

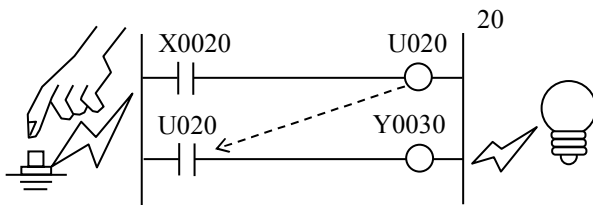
U ワンショット

ナンバーの範囲	000~0FF
設定値	0~65535 (0.0~6553.5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、0.0~6553.5秒で0.1秒単位に設定できます (LADDER DIAGRAM SYSTEM/S10VEからの設定は、0~65535となります)。

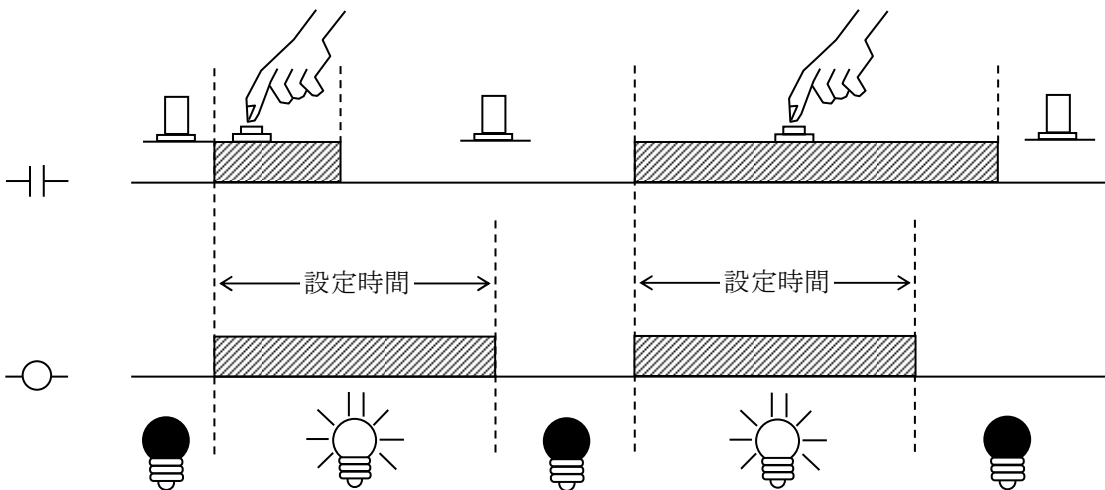
(\*) セットリングパルス幅とは、ワンショットのコイルをONする接点がONしていなければならない最小時間のことを意味します。

● 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 (Y0030) 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、ワンショットが動作しない場合があります。ワンショット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	
		C0	
設定値	0~65535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		

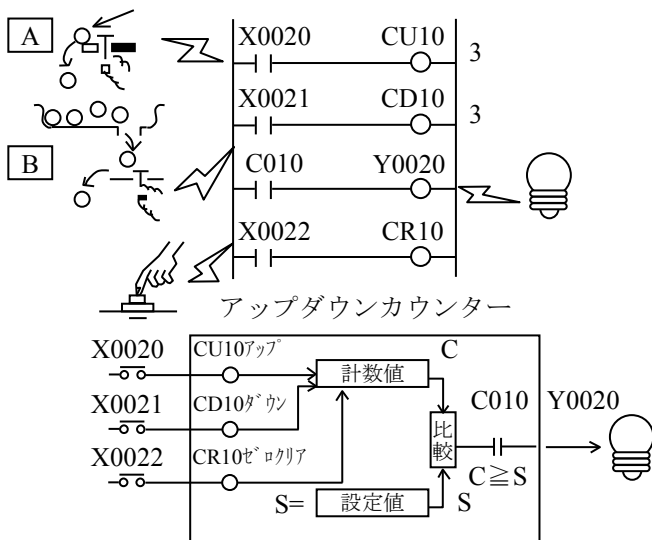
アップダウンカウンターは、アップカウンター（CU）とダウンカウンター（CD）から構成され、計数値はアップカウンターのコイル立ち上がり時にカウントアップ（+1）し、ダウンカウンターのコイル立ち上がり時にカウントダウン（-1）します。

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

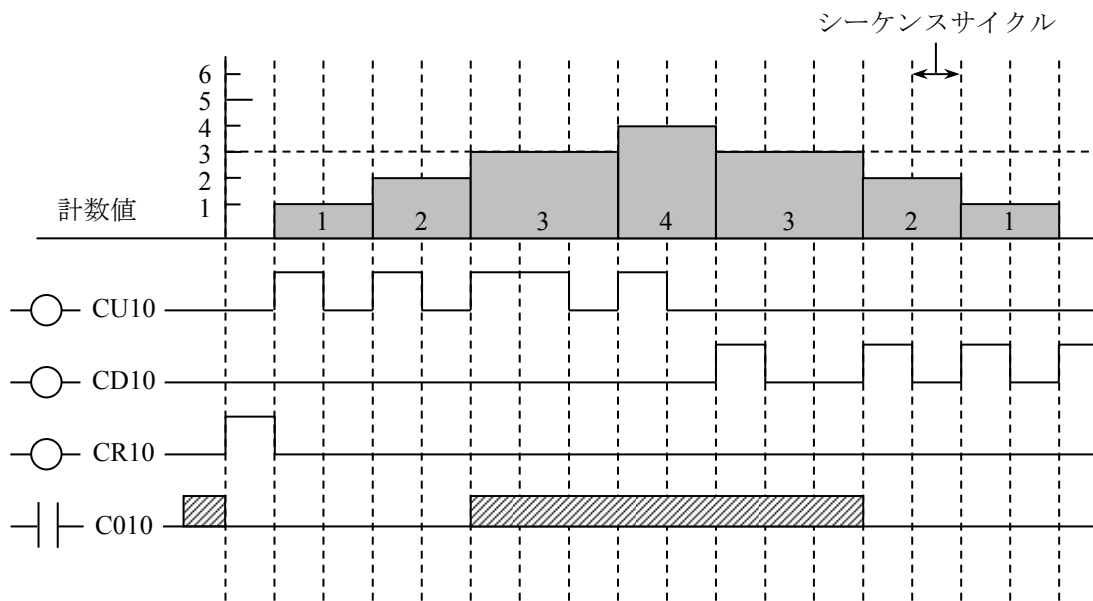
(\*) セットリングパルス幅とは、アップカウンター、ダウンカウンター、リセットのコイルをONする接点がONしていなければならない最小時間のことを意味します。

● 回路例



- 図に示した回路の場合、上方からバスケットに入ったボールの数をスイッチ **A** (X0020) でカウントし、下方へ落ちたボールの数をスイッチ **B** (X0021) 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 (Y0020) が点灯します。また、押しボタン (X0022) を押すと計数値が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状態のカウンタ接点はOFF状態になります。
- ・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。

## 通知

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。

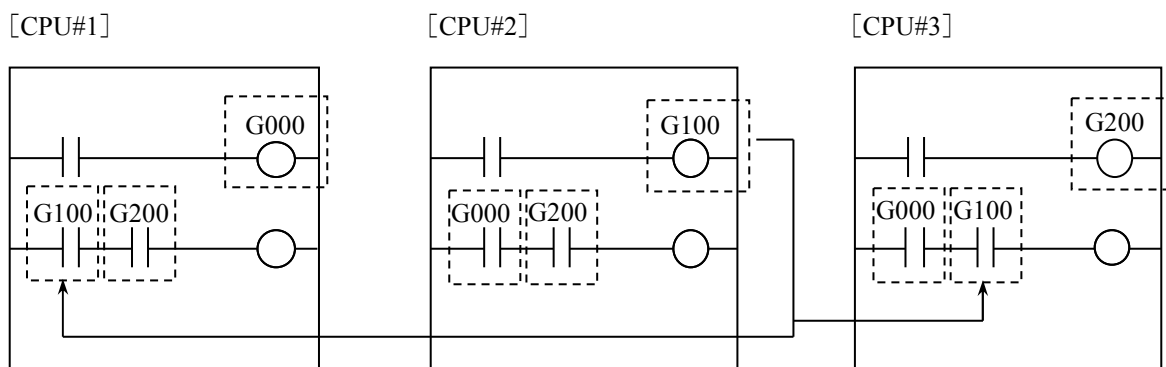
G グローバル リンクレジスター

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

グローバルリンクレジスター（Gレジスター）は、OD.RING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ーです。

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。

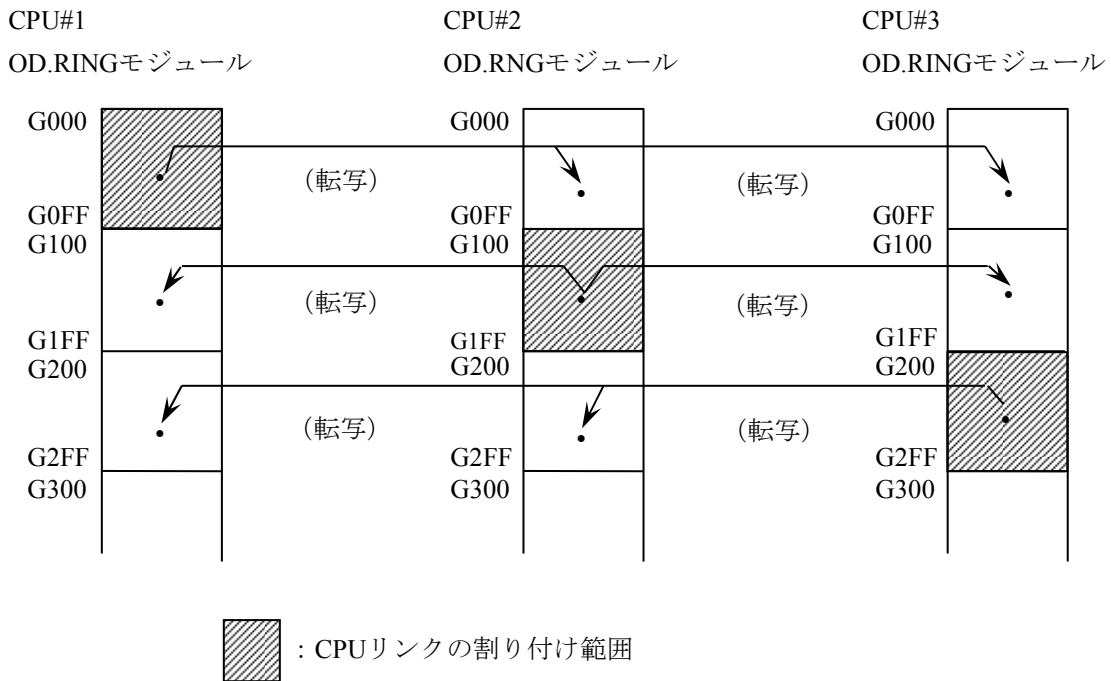
● ラダープログラム回路例（OD.RINGモジュール使用時）



[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。

● OD.RINGモジュールの動作



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



N ネスティングコイル

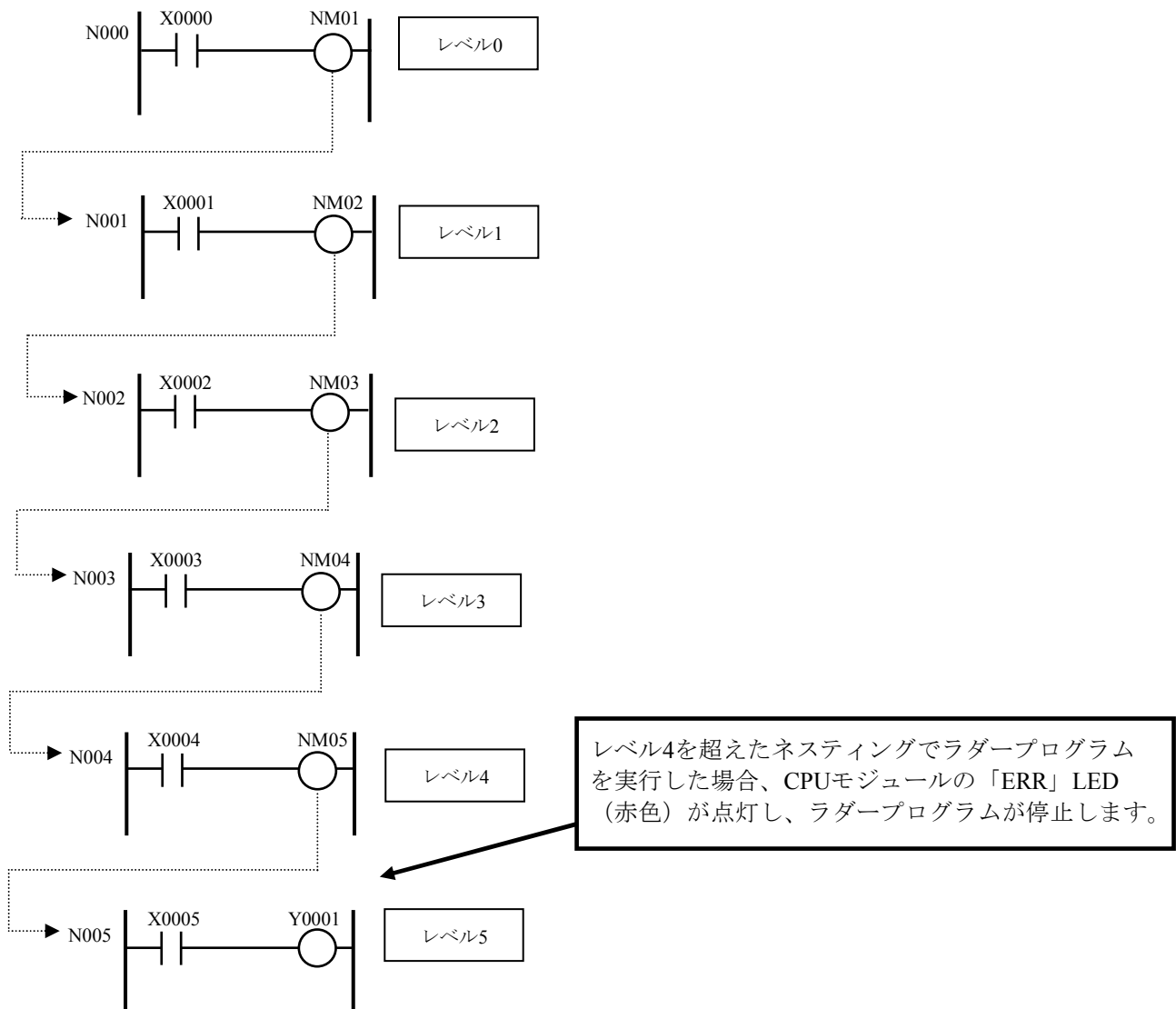
ナンバーの範囲	000~0FF
ネスティング最大レベル	4レベル

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（N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コイルはラダー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せん。

N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エッジ接点とコイルを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。

他のコイルと異なり、複数の同じコイルを同一ラダーシート上に配置することができます。

<最大ネスティングを超えたマスターコントロールの例>



(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、Nコイルの立ち下がり (ON→OFF) のタイミングでOFFになるコイルの違いにより、2つの動作モードがあります。

<マスターコントロールの動作モード>

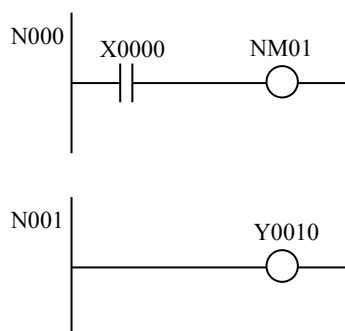
通常モード：立ち下がったNコイル以下のネスティングコイルで使用されている立ち上がり（立ち下がり）エッジ接点、ノーマルコイルがOFFになります。

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。

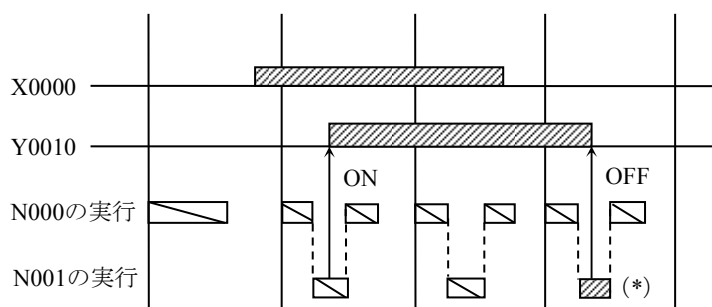
動作モードは、LADDER DIAGRAM SYSTEM/S10VE（型式：S-7898-02）の [ユーティリティ] - [PCsエディション] - [PCsエディション] で切り替えます。操作方法は、「S10VE ソフトウェアマニュアル オペレーション ラダー図 For Windows®（マニュアル番号 SEJ-3-131）」を参照してください。

## ● 回路例

N000からN001に対してマスターコントロールで呼び出します。



## ● 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。

## 通 知

マスターコントロールでは、1つのラダーシート上に同一のNコイルを複数配置して、NコイルのON/OFFが混在する場合、以下の2つの問題があります。

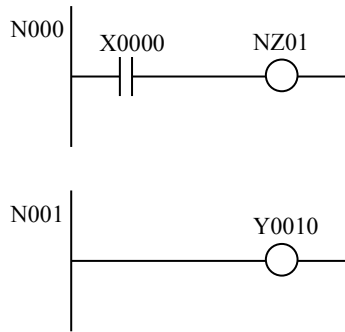
- ・エッジ接点は、常に立ち上がりを検出、またはOFFのままで立ち下りが検出されない場合があります。そのため、Nコイルを複数配置した場合は、エッジ接点を使用しないでください。
- ・1シーケンス内でNコイルの状態によりコイルがON/OFFします。途中でコイルがONしても最後に配置されたNコイルが未実行の場合、回路モニター、MCSなどでコイルの値を参照したとき、OFFに見えますので注意してください。

(2) ゾーンコントロール (NZ)

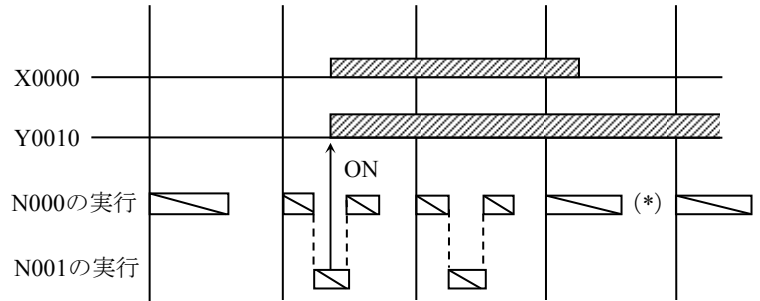
ゾーンコントロールは動作モードの指定はありません。

● 回路例

N000からN001に対してゾーンコントロールで呼び出します。




● 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。

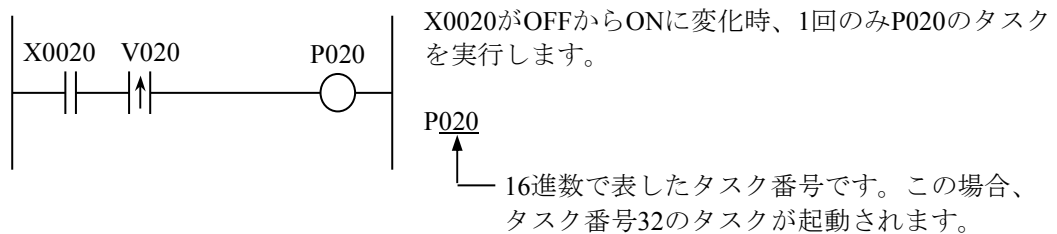
このコイルをONすることにより、ナンバー（タスク番号）に対応したタスクが起動されます。タスクの実行は、CPUモジュールのCP側で行われます。ユーザーはCP側でのみタスクの登録を行うことができます。

<プロセスレジスターの割り付け>

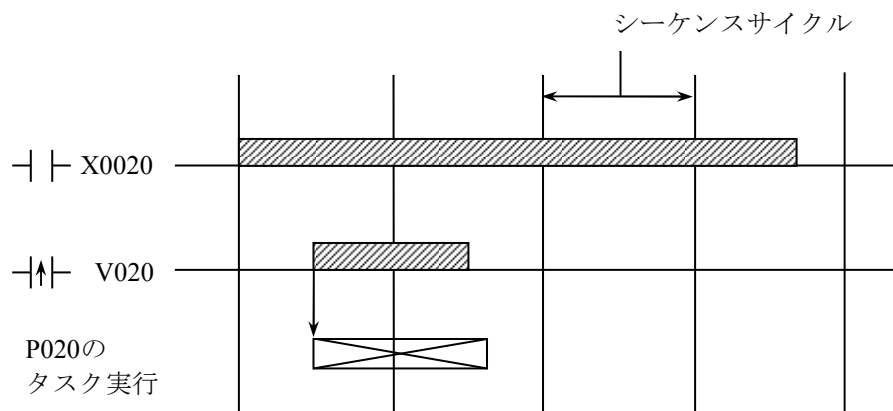
区分	ナンバー	名称	内容
ユーザー作成	P001	イニシャルタスク	CPUモジュールをリセットまたは停復電したときに必ず起動されるタスクです。システムをイニシャライズするプログラムを割り付けてください。
	P002 }	ユーザータスク	ユーザーが作成したプログラムを割り付けてください。
	P080		

(1) ON時1回実行

● 回路例

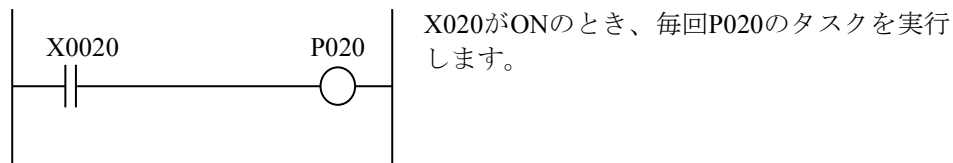


● タイムチャート

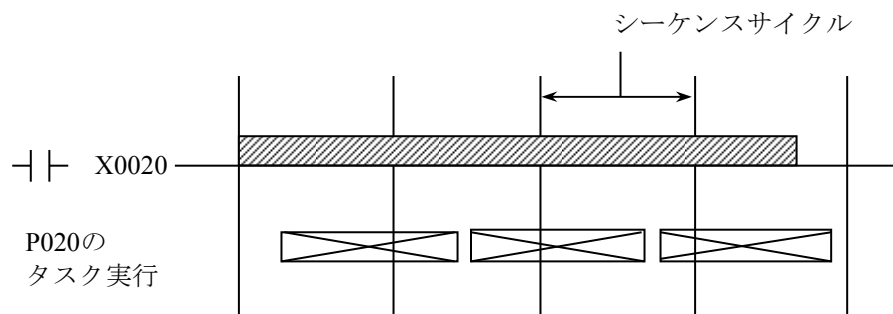


(2) ON時毎回実行

● 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（プロセスレジスタナンバーでは001～080）までです。タスク番号129～300は、このコイルから起動できません。
- ・タスクの実行はCPUモジュールのCP側で行われます。
- ・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。

- ・ラダープログラムはCPへタスク起動要求を発行後、タスクが起動されなかった場合でもそのまま動作を続けます。

E イベントレジスター

ナンバーの範囲	000~FFFF
インディケータ表示範囲	000~01FF
4チャンネルアナログ、 パルスカウンター 入出力範囲	0400~23FF

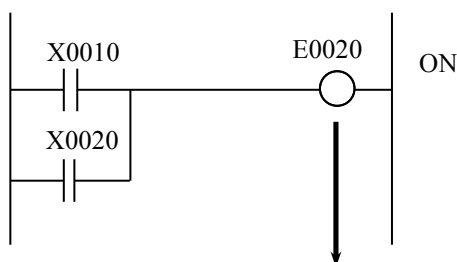
ユーザーエラーなどのイベント情報を出力します。出力したイベント情報は、BASE SYSTEM/S10VEのイベントレジスターモニター機能により監視できます。また、E0000~E01FFいずれかがONした場合、CPUモジュールのインディケータに表示します。E0000~E01FFがすべてOFFした場合、インディケータの表示が消えます。

BASE SYSTEM/S10VE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CPUモジュールのインディケータで表示される範囲は、E0000~E01FFまでとなります。

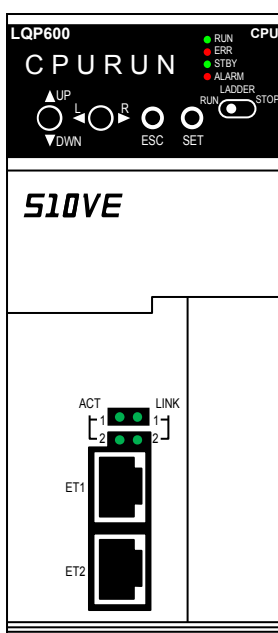
また、ナンバー0400~23FFは、リモートI/Oに接続されたアナログモジュールおよびパルスカウンターモジュールの入出力用として使用します。使用するためには、LADDER DIAGRAM SYSTEM/S10VE（型式：S-7898-02）の [ユーティリティ] - [PCsエディション] - [アナログ/カウンター] で設定してください。操作方法は、「S10VE ソフトウェアマニュアル オペレーション ラダー図 For Windows®（マニュアル番号 SEJ-3-131）」を参照してください。

（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

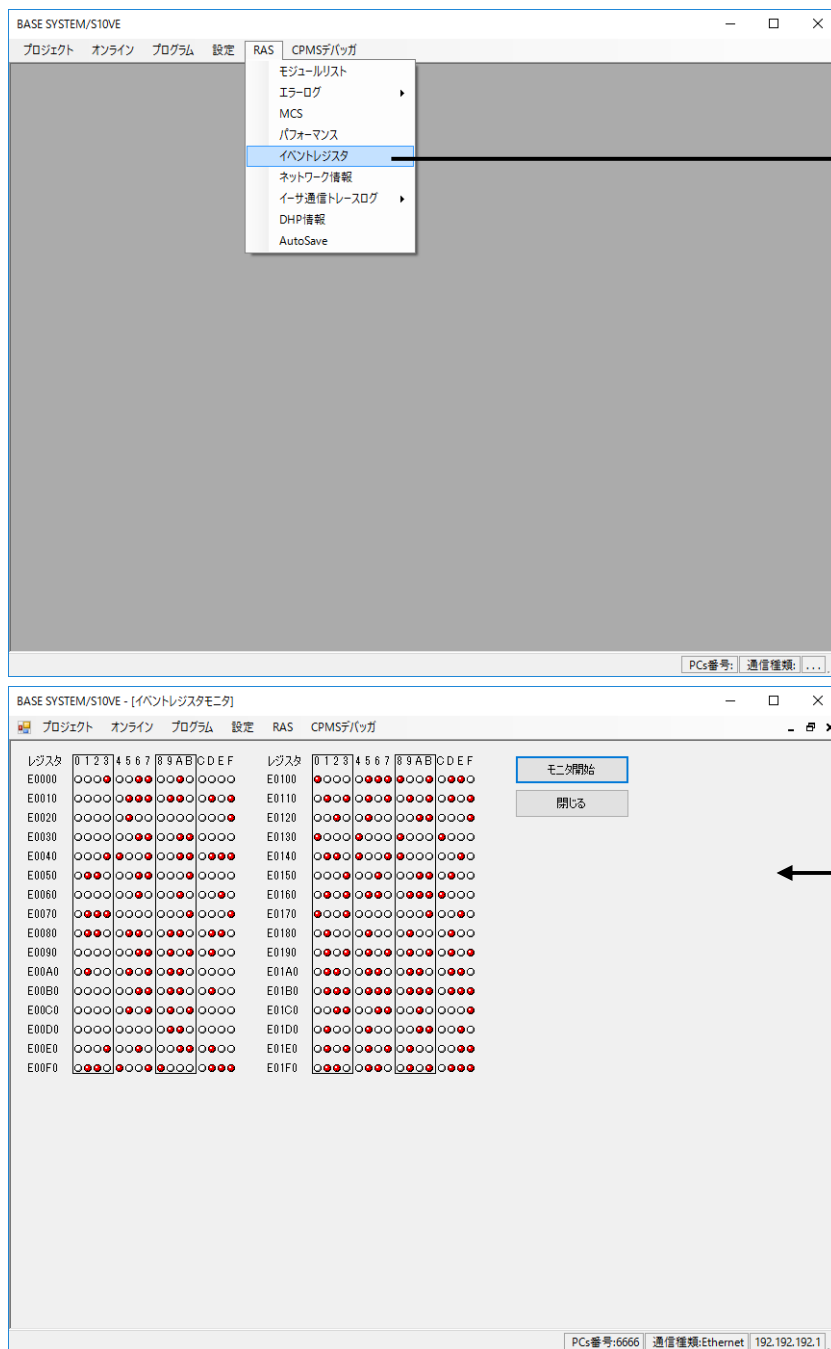


イベントレジスターがONするとCPUのインディケータに表示します。複数イベントレジスターがONしたときの表示については、「S10VE ユーザーズマニュアル 総合編（マニュアル番号 SEJ-1-001）」を参照してください。



イベント情報の確認は、BASE SYSTEM/S10VEのイベント情報表示機能で確認します（下図参照）。

BASE SYSTEM/S10VEによるイベントレジスタモニター



V エッジ接点

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

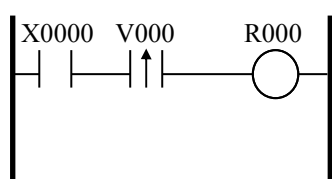
エッジ接点には立ち上がりエッジ検出接点 (↑) と立ち下がりエッジ検出接点 (↓) があり、それぞれのエッジを検出した1シーケンスサイクルだけONします。

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。

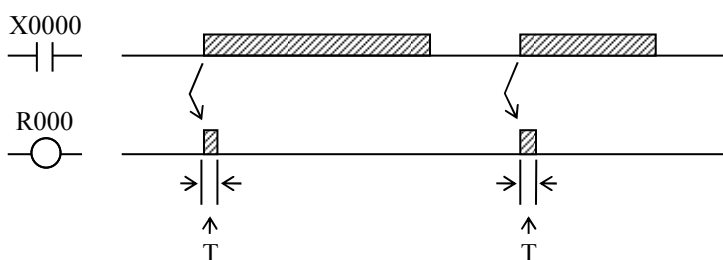
(1) 立ち上がりエッジ接点

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>

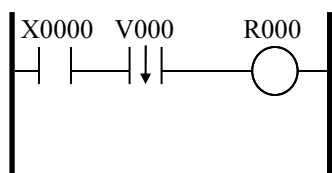


T : 1シーケンスサイクル

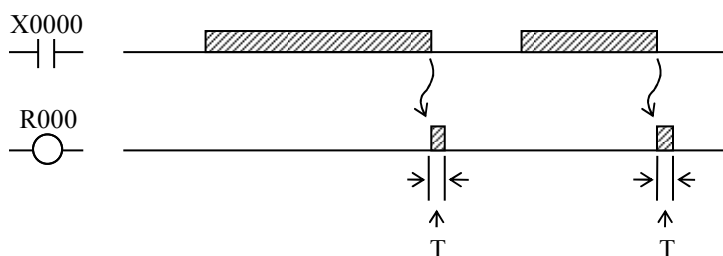
(2) 立ち下がりエッジ接点

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

通知

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。



S システムレジスター

ナンバーの 範囲	0000～BFFF
-------------	-----------

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。

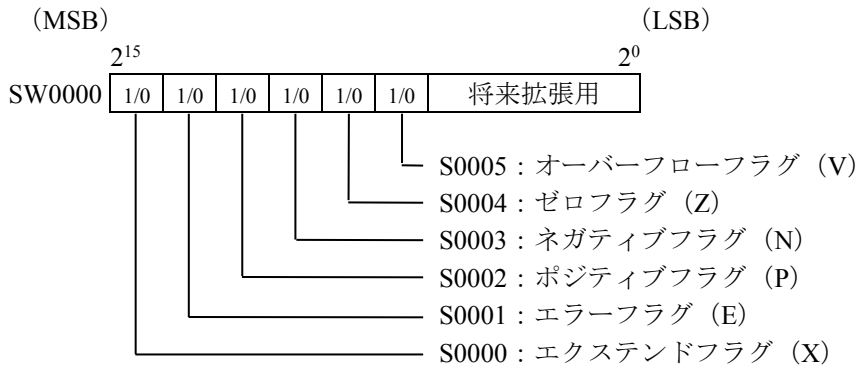
システムレジスター一覧を表1-6に示します。

表1-6 システムレジスター一覧

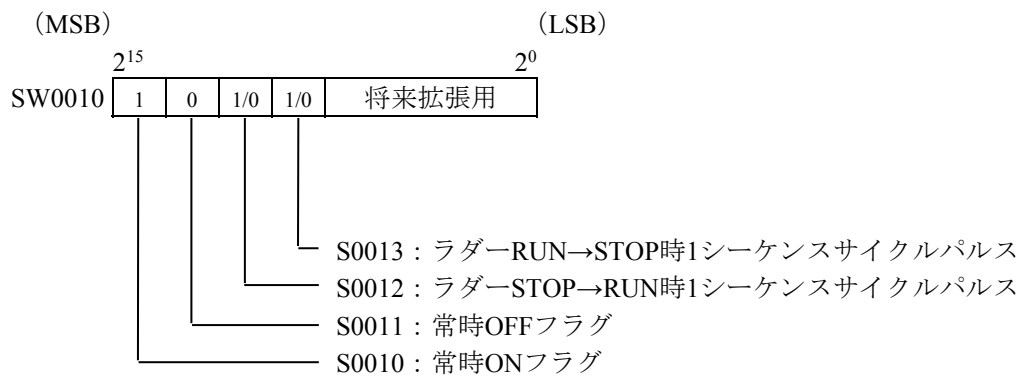
No.	レジスター ナンバー	概要
1	S0000～S000F	演算ファンクションフラグレジスター
2	S0010～S001F	ラダープログラム制御レジスター
3	S0020～S002F	HI-FLOW応用命令実行結果フラグレジスター
4	S0030～S00FF	システム予約
5	S0100～S015F	ラダープログラム制御カウンター
6	S0160～S01FF	システム予約
7	S0200～S020F	時刻制御レジスター
8	S0210～S027F	システム予約
9	S0280～S02EF	時刻格納レジスター
10	S02F0～S02FF	システム予約
11	S0300～S047F	リモートI/O状態レジスター
12	S0480～S053F	システム予約
13	S0540～S057F	オプションモジュール状態レジスター (D.NET)
14	S0580～S08BF	システム予約
15	S08C0～S08FF	オプションモジュール状態レジスター (ET.NET)
16	S0900～S093F	シーケンスサイクル時間
17	S0940～S097F	ラダー実行時間
18	S0980～S09BF	オプションモジュール状態レジスター (D.NET)
19	S09C0～S09FF	イーサネット通信実行結果フラグレジスター
20	S0A00～S0ADF	オプションモジュール状態レジスター (J.NET)
21	S0AE0～S0BEF	システム予約
22	S0BF0～S0BFF	CPUステータスレジスター
23	S0C00～S0CFF	構成制御レジスター
24	S0D00～S0EFF	オプションモジュール状態出力
25	S0F00～S0FFF	オプションパラメーター設定出力レジスター
26	S1000～S2FFF	システム予約
27	S3000～S31FF	オプションモジュール実装状態出力レジスター
28	S3200～SAFFF	システム予約
29	SB000～SBFFF	ユーザー開放

(1) 演算ファンクションフラグレジスタ

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



タイムチャート

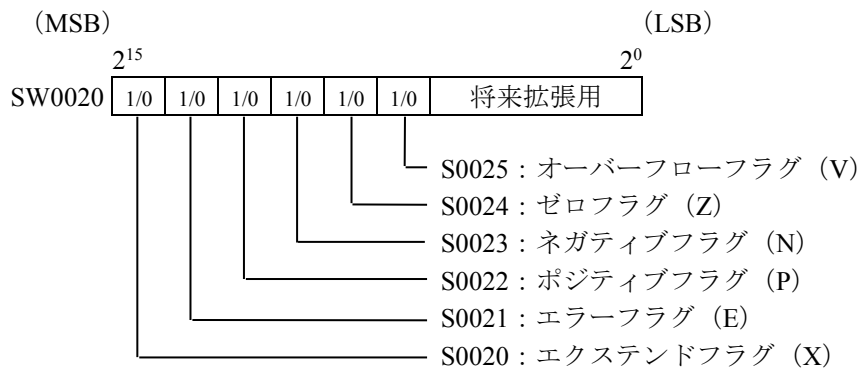
スイッチ操作



(注) 停電した場合は、S0010, S0011, S0012, S0013はONしません。

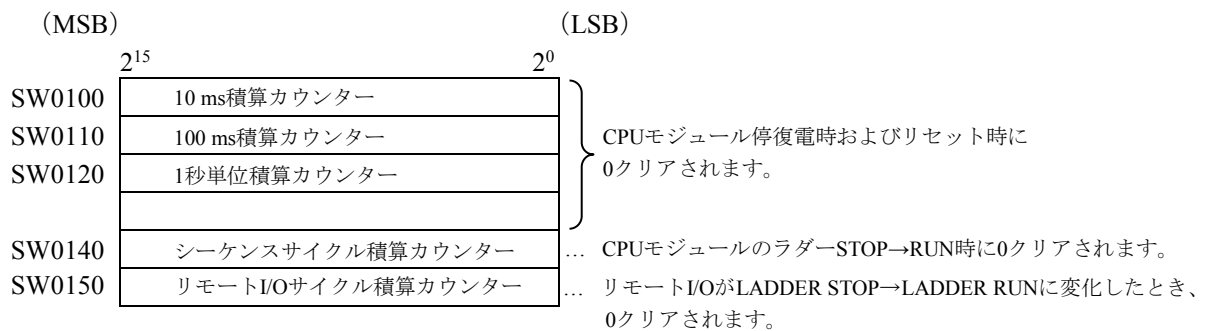
(3) HI-FLOW応用命令実行結果フラグレジスタ

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

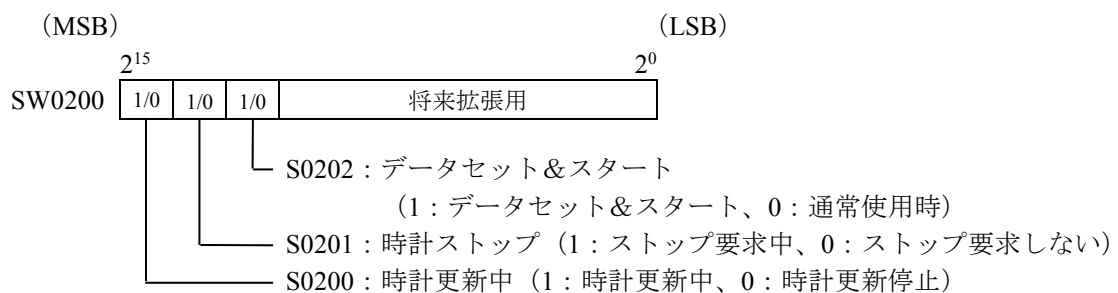
シーケンス制御で使用できる積算カウンターです。



- 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

CPUモジュールの現在時刻の設定を制御するレジスタです。CPU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

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

	(MSB)	(LSB)
	2 <sup>15</sup>	2 <sup>8</sup> 2 <sup>7</sup> 2 <sup>0</sup>
SW0280	未使用	秒
SW0290	未使用	分
SW02A0	未使用	時
SW02B0	未使用	日
SW02C0	未使用	月
SW02D0	年 (西暦)	
SW02E0	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)

(7) リモートI/O状態レジスター

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ーです。

<レジスターの割り付け>

S0300	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ーに「1」が設定されます。 (*)
S0380	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ーに「1」が設定されます。 (*)
S0400	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ーに「1」が設定されます。 (*)
S047F		(*) 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	0000~00F	S0300	S0380	S0400
1	0010~01F	S0301	S0381	S0401
2	0020~02F	S0302	S0382	S0402
3	0030~03F	S0303	S0383	S0403
4	0040~04F	S0304	S0384	S0404
⋮	⋮	⋮	⋮	⋮
124	07C0~7CF	S037C	S03FC	S047C
125	07D0~7DF	S037D	S03FD	S047D
126	07E0~7EF	S037E	S03FE	S047E
127	07F0~7FF	S037F	S03FF	S047F

(8) オプションモジュール状態レジスタ (ET.NET)

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です。詳細は、「S10VE ユーザーズマニュアル オプション ET.NET (LQE260-E) (マニュアル番号 SEJ-1-105)」を参照してください。

	(MSB)		(LSB)
	2 <sup>15</sup>		2 <sup>0</sup>
SW08C0	MAIN CH1 エラー情報		
SW08D0	MAIN CH2 エラー情報		
SW08E0	SUB CH1 エラー情報		
SW08F0	SUB CH2 エラー情報		

(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 <sup>15</sup>		2 <sup>0</sup>
SW0900	最新測定データ (ms)		
SW0910	最大測定データ (ms)		
SW0920	最小測定データ (ms)		
SW0930	最近16回の平均データ (ms) (*)		

ラダーを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。

(\*) 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 <sup>15</sup>		2 <sup>0</sup>
SW0940	最新測定データ (ms)		
SW0950	最大測定データ (ms)		
SW0960	最小測定データ (ms)		
SW0970	最近16回の平均データ (ms) (*)		

ラダーを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。

(\*) 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 (D.NET)

各モジュール (モジュール0~3) に発生したエラー情報を格納するレジスタです。詳細は、「S10VE ユーザーズマニュアル オプション D.NET (LQE770-E) (マニュアル番号 SEJ-1-103)」を参照してください。

	(MSB)		(LSB)
	2 <sup>15</sup>		2 <sup>0</sup>
SW0980	モジュール0 チャンネル0エラー情報		
SW0990	モジュール0 チャンネル1エラー情報		
SW09A0	モジュール1 チャンネル0エラー情報		
SW09B0	モジュール1 チャンネル1エラー情報		

	(MSB)		(LSB)
	2 <sup>15</sup>		2 <sup>0</sup>
SW0540	モジュール2 チャンネル0エラー情報		
SW0550	モジュール2 チャンネル1エラー情報		
SW0560	モジュール3 チャンネル0エラー情報		
SW0570	モジュール3 チャンネル1エラー情報		

## (12) イーサネット通信実行結果フラグレジスタ

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です。

管理番号ごとに実行結果がシステムレジスタS09C0～S09EF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。

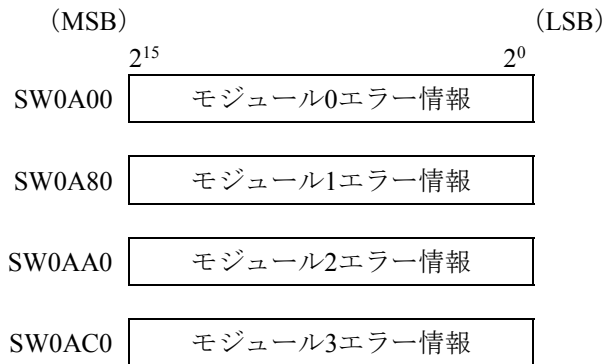
管理番号とは、使用できるソケットに対応付けている番号のことです。

レジスタ		管理番号	備考
ワード	ビット		
SW09C0	S09C0	1	CPU イーサネット通信用
	S09C1	2	
	}	}	
	S09CE	15	
	S09CF	16	
SW09D0	S09D0	17	ET.NET(メイン) イーサネット通信用
	S09D1	18	
	}	}	
	S09DE	31	
	S09DF	32	
SW09E0	S09E0	33	ET.NET(サブ) イーサネット通信用
	S09E1	34	
	}	}	
	S09EE	47	
	S09EF	48	



(13) オプションモジュール状態レジスタ (J.NET)

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です。詳細は、「S10VE ユーザーズマニュアル オプション J.NET (LQE540-E)（マニュアル番号 SEJ-1-102）」を参照してください。



(14) CPUステータスレジスター

現在のCPU状態を示すレジスターです。

<CPUステータスのビット構成>

	(MSB)	$2^{15}$	$2^{14}$	$2^{13}$	$2^{12}$	$2^{11}$	$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	(LSB)
SW0BF0		1/0	1/0	*	1/0	1/0	*	1/0	*	1/0	1/0	1/0	1/0	*	1/0	1/0	1/0
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	S0BF0	ラダーSTOP中	ラダーRUN中
1	S0BF1	シミュレーション中	通常RUN中
2	S0BF2	将来拡張用	
3	S0BF3	プロテクトSW ON状態	プロテクトSW OFF状態
4	S0BF4	リモートI/O動作中	リモートI/O停止中
5	S0BF5	将来拡張用	
6	S0BF6	ラダー書き換え中	ラダー書き換え完了
7	S0BF7	将来拡張用	
8	S0BF8	1次電池状態 電圧低下	1次電池状態 正常
9	S0BF9	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	S0BFA	FUSE断ステーションあり	FUSE断ステーションなし
B	S0BFB	オプションモジュールエラー (*1) あり	オプションモジュールエラー (*1) なし
C	S0BFC	将来拡張用	
D	S0BFD	GR (復電) またはリセット時に0クリア	
E	S0BFE	HPシステムタスク停止中	HPシステムタスク正常運転中
F	S0BFF	CPMS停止中 (CPダウン中)	CPMS動作中 (CP正常動作中)

(\*1) オプションモジュールエラー：CPU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) 構成制御レジスタ

構成制御を要求するレジスタです。

<構成制御のビット構成>

	(MSB)															(LSB)
	2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
SW0C00	1/0	*	*	1/0	*	*	*	*	*	*	*	*	*	*	*	*
SW0C10	1/0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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。

S0C00～S0C0F

ビット No.	ビット レジスタNo.	各ビットの内容	
		ON (=1)	OFF (=0)
0	S0C00	CP立ち上げ完了	CP立ち上げ中
1	S0C01	将来拡張用	
2	S0C02	将来拡張用	
3	S0C03	システムレジスタクリア完了	立ち上げ中
4	S0C04	将来拡張用	
5	S0C05	将来拡張用	
6	S0C06	将来拡張用	
7	S0C07	将来拡張用	
8	S0C08	将来拡張用	
9	S0C09	将来拡張用	
A	S0C0A	将来拡張用	
B	S0C0B	将来拡張用	
C	S0C0C	将来拡張用	
D	S0C0D	将来拡張用	
E	S0C0E	将来拡張用	
F	S0C0F	将来拡張用	

## S0C10～S0C1F（書き換え禁止・システム専用）

ビット No.	ビット レジスタNo.	各ビットの内容	
		ON (=1)	OFF (=0)
0	S0C10	HPタスクエラー停止	正常動作
1	S0C11	将来拡張用	
2	S0C12	将来拡張用	
3	S0C13	将来拡張用	
4	S0C14	将来拡張用	
5	S0C15	将来拡張用	
6	S0C16	将来拡張用	
7	S0C17	将来拡張用	
8	S0C18	将来拡張用	
9	S0C19	将来拡張用	
A	S0C1A	将来拡張用	
B	S0C1B	将来拡張用	
C	S0C1C	将来拡張用	
D	S0C1D	将来拡張用	
E	S0C1E	将来拡張用	
F	S0C1F	将来拡張用	

第 1 章 ラダー命令

	(MSB)		(LSB)
	$2^{15}$		$2^0$
SW0C20	シーケンスサイクル時間 (ms)		
SW0C30	WDT監視時間 (ms)		
SW0C40	将来拡張用		
SW0C50	将来拡張用		
SW0C60	シーケンスサイクル積算カウンター		
SW0C70	将来拡張用		
⋮	⋮		
SW0CF0	将来拡張用		

(16) オプションモジュール状態出力レジスタ

オプションモジュールの動作状態を出力するレジスタです。

	(MSB)	$2^{15}$	$2^0$	(LSB)
SW0D00	FL.NET(メイン)			
SW0D10	FL.NET(サブ)			
SW0D20	OD.RING(LQE510/515)(メイン)			
SW0D30	OD.RING(LQE510/515)(サブ)			
SW0D40	将来拡張用			
:	:			
SW0EF0	将来拡張用			

<オプションモジュール状態出力レジスタのビット構成>

	(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)
		1/0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	
ビットNo.		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

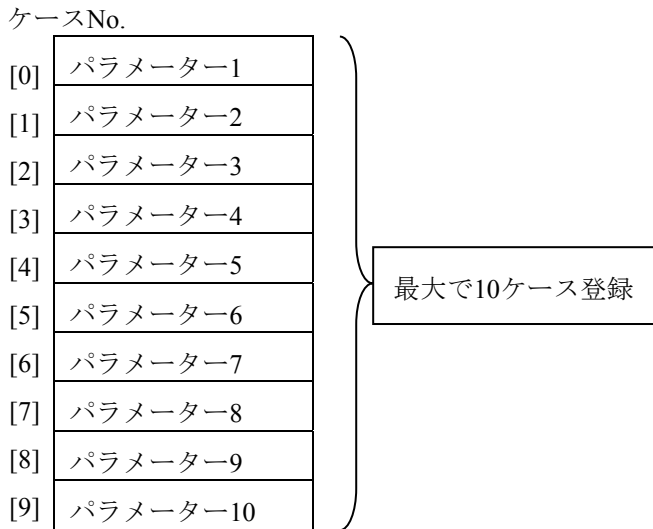
\*は将来拡張用です。

ビット No.	各ビットの内容	
	ON (=1)	OFF (=0)
0	モジュールエラー停止中	モジュールエラー停止中以外
1	将来拡張用	
:	:	
:	:	
F	将来拡張用	

(17) オプションパラメーター設定出力レジスタ

オプションモジュールパラメーター設定の有効／無効および設定書込み正常／異常を出力するレジスタです。オプションモジュールパラメーターは、各オプションモジュールのツールからCPUの「オプションモジュール設定パラメーター」エリアの任意のケースNo.に設定します。オプションモジュールパラメーターを最大で10ケース登録できます。

<オプションモジュール設定パラメーター>



	(MSB)										(LSB)					
	2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
SW0F00	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	*	*	*	*	*	*
SW0F10	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	1/0	*	*	*	*	*	*
SW0F20	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
⋮																
⋮																
SW0FF0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
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。

## S0F00～S0F0F

ビット No.	ビット レジスタ-No.	各ビットの内容	
		ON (=1)	OFF (=0)
0	S0F00	パラメーター1設定有効	パラメーター1設定無効
1	S0F01	パラメーター2設定有効	パラメーター2設定無効
2	S0F02	パラメーター3設定有効	パラメーター3設定無効
3	S0F03	パラメーター4設定有効	パラメーター4設定無効
4	S0F04	パラメーター5設定有効	パラメーター5設定無効
5	S0F05	パラメーター6設定有効	パラメーター6設定無効
6	S0F06	パラメーター7設定有効	パラメーター7設定無効
7	S0F07	パラメーター8設定有効	パラメーター8設定無効
8	S0F08	パラメーター9設定有効	パラメーター9設定無効
9	S0F09	パラメーター10設定有効	パラメーター10設定無効
A	S0F0A	将来拡張用	
B	S0F0B	将来拡張用	
C	S0F0C	将来拡張用	
D	S0F0D	将来拡張用	
E	S0F0E	将来拡張用	
F	S0F0F	将来拡張用	



S0F10～S0F1F

ビット No.	ビット レジスタNo.	各ビットの内容	
		ON (=1)	OFF (=0)
0	S0F10	パラメーター1書込みエラー	パラメーター1書込み正常
1	S0F11	パラメーター2書込みエラー	パラメーター2書込み正常
2	S0F12	パラメーター3書込みエラー	パラメーター3書込み正常
3	S0F13	パラメーター4書込みエラー	パラメーター4書込み正常
4	S0F14	パラメーター5書込みエラー	パラメーター5書込み正常
5	S0F15	パラメーター6書込みエラー	パラメーター6書込み正常
6	S0F16	パラメーター7書込みエラー	パラメーター7書込み正常
7	S0F17	パラメーター8書込みエラー	パラメーター8書込み正常
8	S0F18	パラメーター9書込みエラー	パラメーター9書込み正常
9	S0F19	パラメーター10書込みエラー	パラメーター10書込み正常
A	S0F1A	将来拡張用	
B	S0F1B	将来拡張用	
C	S0F1C	将来拡張用	
D	S0F1D	将来拡張用	
E	S0F1E	将来拡張用	
F	S0F1F	将来拡張用	

## (18) オプションモジュール実装状態出力レジスタ

オプションモジュールの実装状態を出力するレジスタで、実装されているオプションモジュールの該当ビットに“1”が出力されます。

オプションモジュール	モジュール番号	ビットレジスタ
OD.RING	MAIN	S3010
	SUB	S3011
FL.NET	MAIN	S3020
	SUB	S3021
J.NET	MODULEx	S303x
D.NET	MODULEx	S304x
ET.NET	MAIN	S30E1
	SUB	S30E2

x : モジュール番号

LR, LV ラダーコンバーター専用ワークレジスタ

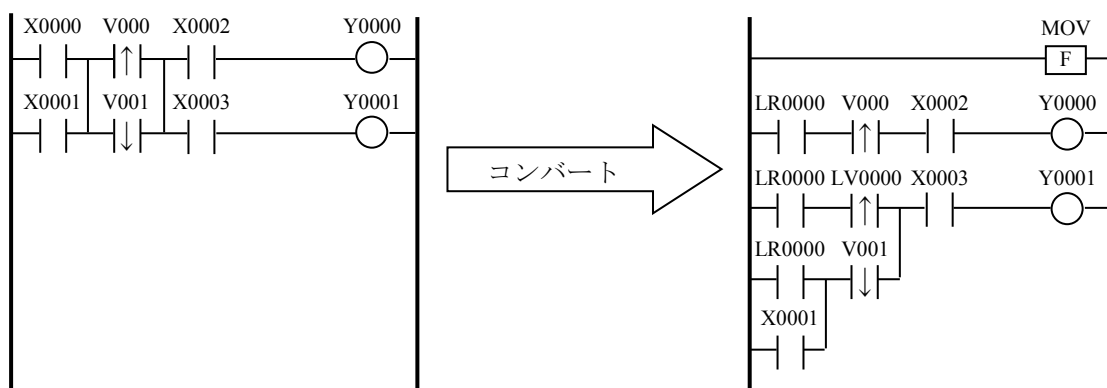
ナンバーの 範囲	0000~0FFF
-------------	-----------

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に、コンバーターが使用する専用レジスタです。S10VEのラダー図システムでは、S10/2αシリーズまたはS10miniシリーズのラダー図システムで作成したラダープログラムをS10VE向けに直接コンバートすることはできません。コンバートの機能はありませんが、内部レジスタとして使用することはできません。

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



## 1. 6 ラダーウォッチドッグタイマー

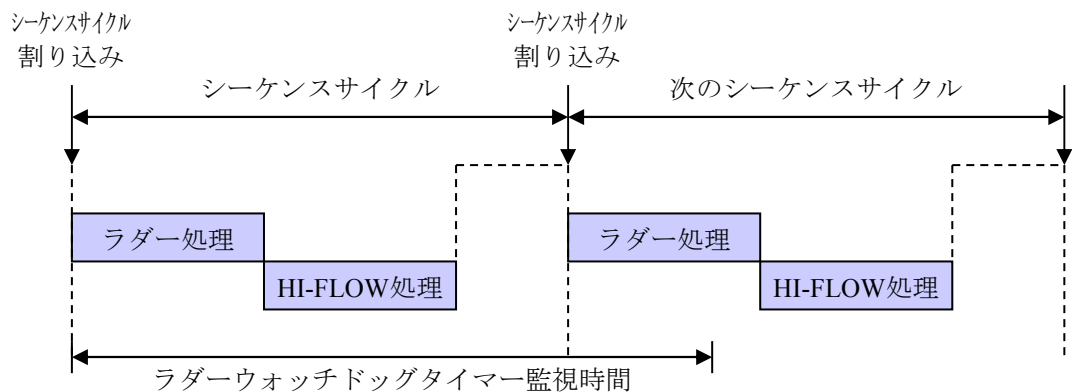
ラダーウォッチドッグタイマーは、ラダー処理とHI-FLOW処理が設定した監視時間内に終了することを監視するためのものです。設定した監視時間内に処理が終了しなかった場合、ラダープログラムウォッチドッグタイマータイムアウトエラー（以降、ラダーWDTエラーと略します。）になり、次の動作をします。

- CPUのERR LED点灯、エラー詳細情報を記録し、CPUを停止します。
- ラダー、HI-FLOW、リモートI/O通信など、HP側タスクが停止します。
- PCsOK信号およびCPU 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

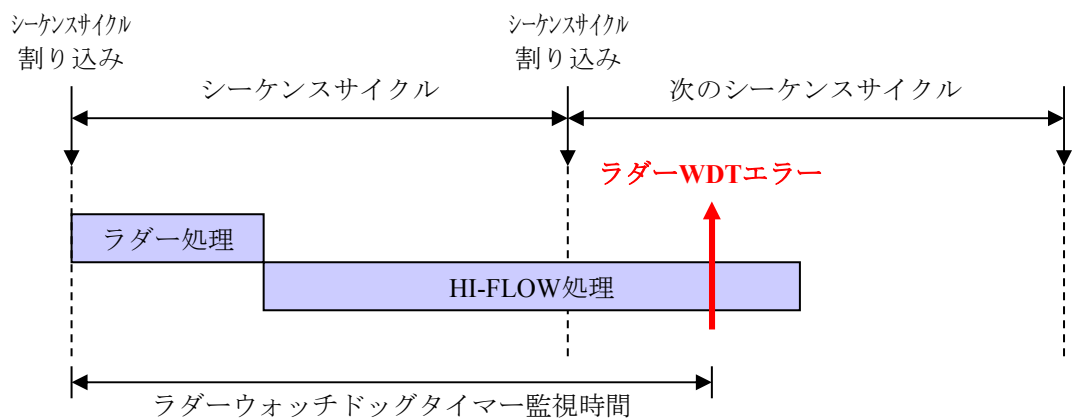
#### <通常動作>

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・HI-FLOWの処理時間およびユーザーの使用環境を考慮し、実際に要する時間よりも十分大きい値を設定してください。必要な余裕は構成するシステムに依存します。
- ・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。そのため、監視時間をシーケンスサイクル以下に設定することはできません。

### 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、CPU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、BASE SYSTEM/S10VE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：05C70000 内容：ウォッチドッグタイマタイムアウトエラー」がBASE SYSTEM/S10VEに表示されます。BASE SYSTEM/S10VEの操作方法および説明は、「S10VE ユーザーズマニュアル 総合編（マニュアル番号 SEJ-1-001）」を参照してください。

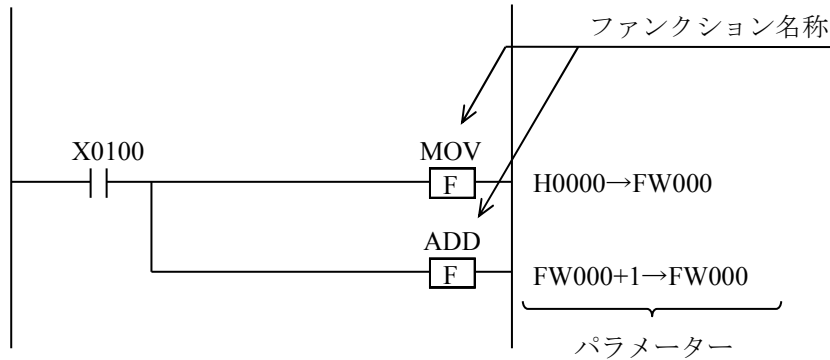
## 第2章 演算ファンクション

### 2.1 機能概要

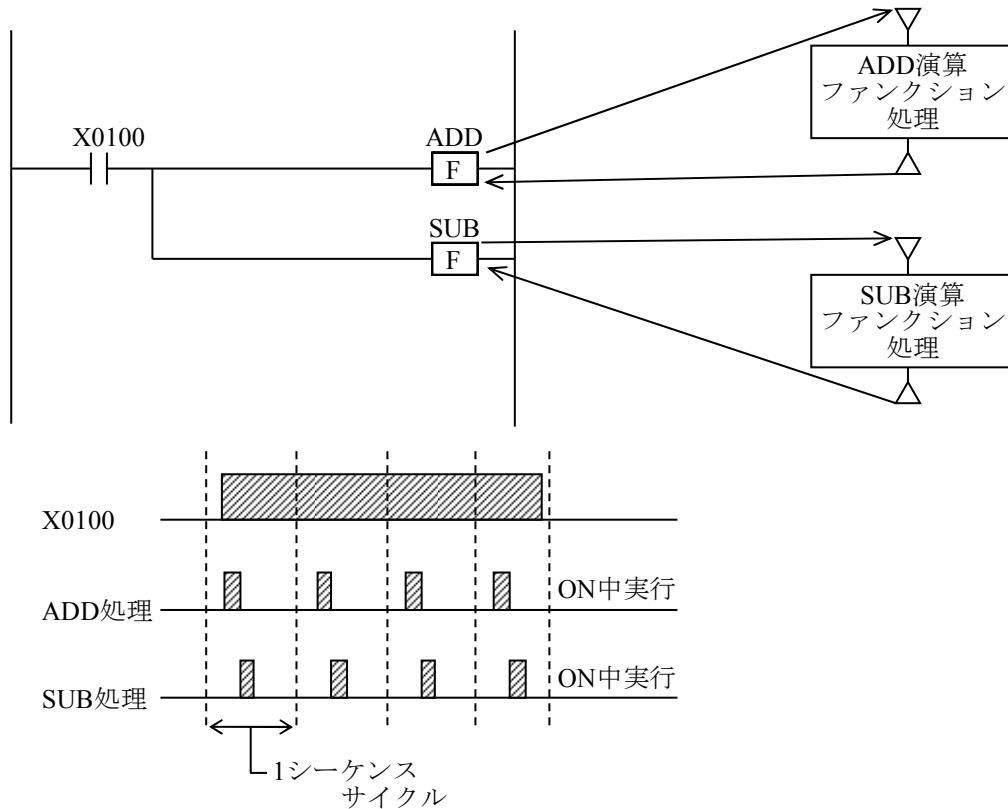
ラダープログラムで算術演算をしたい場合は、演算ファンクションを使用すると簡単にプログラムできます。

● 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。

## 第2章 演算ファンクション

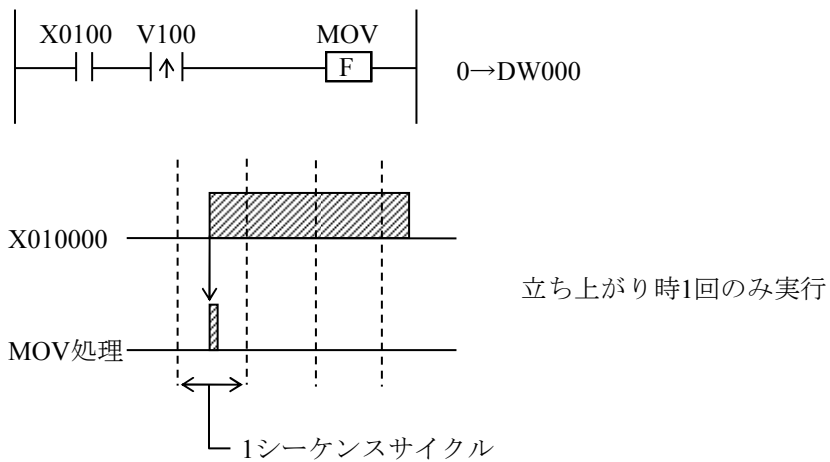
### (1) パラメーター

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。

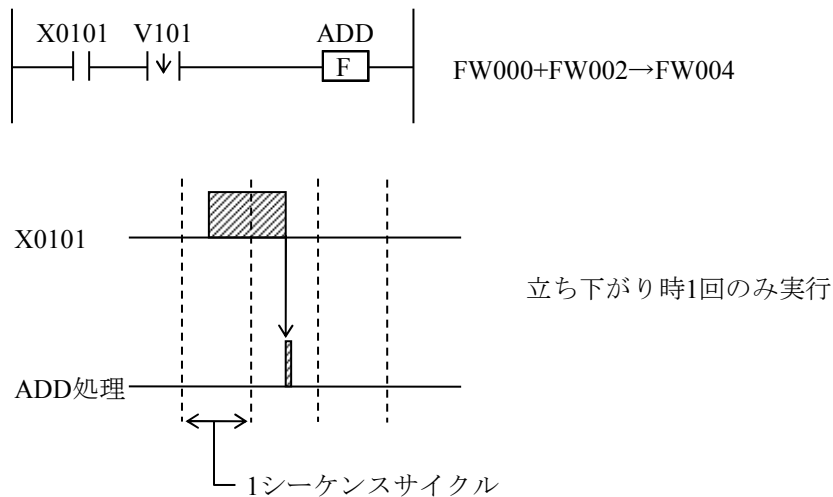
### (2) 動作

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

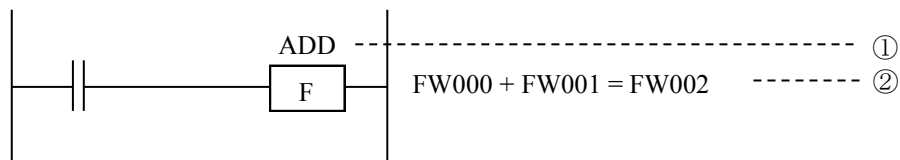


(例2) 立ち下がりエッジ接点との組み合わせ



## 2. 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。

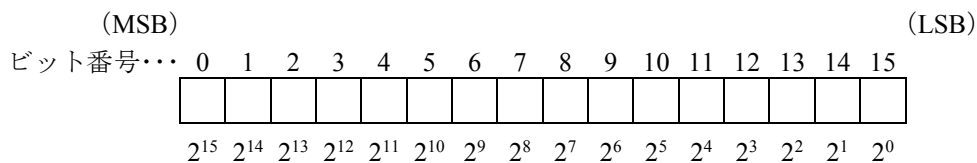
## (2) データフォーマット

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 (X0000, 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 (U)、およびアップダウンカウンター (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。

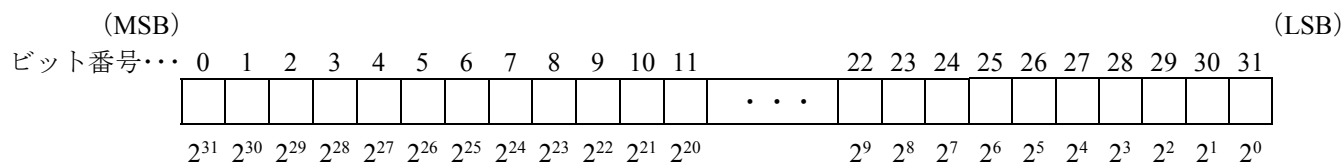


## 第2章 演算ファンクション

- 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。

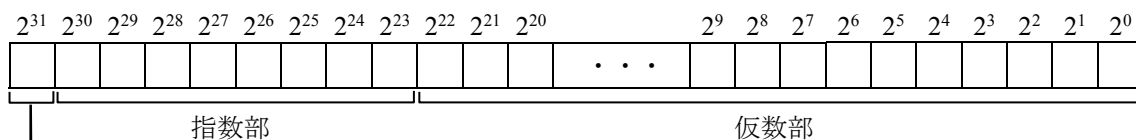
10進数 : -2147483648~+2147483647

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,  $\pm 2^{-126}$  ~  $\pm 2^{128}$

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。

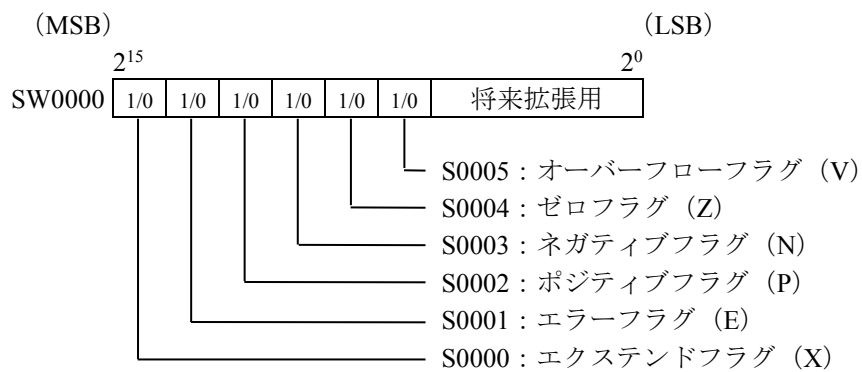
アンダーフロー : 正しい符号を持った0を返します。

(注) 入力において値が、非正規化数 (0.0を除く  $-1.175494E-038$  ( $0x807FFFFFFF$ ) ~

$1.175494E-038$  ( $0x007FFFFFFF$ ) ) の場合は、0.0として演算します。

## (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) 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。

第2章 演算ファンクション

<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	V	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	-	-	-	-	-	●			
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	-	●	-	-	-	-			E: 演算結果が異常 (*) のとき1、 以外のとき0
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 (インダイレクト指定時のみ)

## 2. 3 演算ファンクションで使用するレジスタ

「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にはレジスタを指定できます。ここでは、演算ファンクションで使用するレジスタについて説明します。

## 2. 3. 1 演算ファンクションで利用できるレジスタ

表2-1に演算ファンクションで利用できるレジスタ一覧を示します。各レジスタにはそれぞれ専用の用途とレジスタ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利用できるレジスタ一覧

(1/3)

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
外部入力	X (ビット)	0000~FFFF	リモートI/Oに接続された入力モジュールからのデータ入力用	クリア
	XW (ワード)	0000~FFF0		
	XL (ロングワード)	0000~FFE0		
外部出力	Y (ビット)	0000~FFFF	リモートI/Oに接続された出力モジュールへのデータ出力用	クリア
	YW (ワード)	0000~FFF0		
	YL (ロングワード)	0000~FFE0		
内部レジスタ	R, A (ビット)	000~FFF	演算結果中継用	クリア
	M (ビット)	0000~FFFF		
	RW, AW (ワード)	000~FF0		
	MW (ワード)	000~FFF0		
	RL, AL (ロングワード)	000~FE0		
	ML (ロングワード)	000~FFE0		
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		
ワンディレイタイマー (設定値)	TS (ワード)	000~1FF	ワンディレイタイマー	保持
ワンディレイタイマー (計数値)	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
	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	ワンショットタイマー	保持
ワンショットタイマー (計数値)	UC (ワード)	000~1FF		クリア

## 第2章 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ー一覧

(2/3)

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
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ー	G (ビット)	000~FFF	PLC間リンクエージ用	クリア
	GW (ワード)	000~FF0		
	GL (ロングワード)	000~FE0		
ネスティングコイル	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		
プロセスレジスター	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		
イベントレジスター	E (ビット)	0000~FFFF	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)	0000~FFF0		
	EL (ロングワード)	0000~FFE0		
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		
ゼットレジスター	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		
システムレジスター	S (ビット)	0000~BFFF	システム状態表示用	初期値
	SW (ワード)	0000~BFF0		
	SL (ロングワード)	0000~BFE0		
HI-FLOWとラダー間 共有データレジスター	J (ビット)	000~FFF	HI-FLOWとラダー間 データ共有用	クリア
	JW (ワード)	000~FF0		
	JL (ロングワード)	000~FE0		
	Q (ビット)	0000~FFFF		
	QW (ワード)	0000~FFF0		
	QL (ロングワード)	0000~FFE0		
内部拡張レジスター	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		

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

(3/3)

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
コンバーター 専用レジスター	LR, LV (ビット)	0000~0FFF	内部レジスターとして 使用	クリア
	LRW, LVW (ワード)	0000~0FF0		
	LRL, LVL (ロングワード)	0000~0FE0		
I/O入出力用レジスター (将来用)	IW, OW (ワード)	000~FFF	将来用	クリア
内部レジスター	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 データレジスター	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 ワークレジスター	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 ワークレジスター	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 ワークレジスター	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 ワークレジスター	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ー	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ー	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ー	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レジスターは他のレジスターと比較し、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。

## 第2章 演算ファンクション

### 2. 3. 2 ビットレジスタの扱い

演算ファンクションでは、X0000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使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。

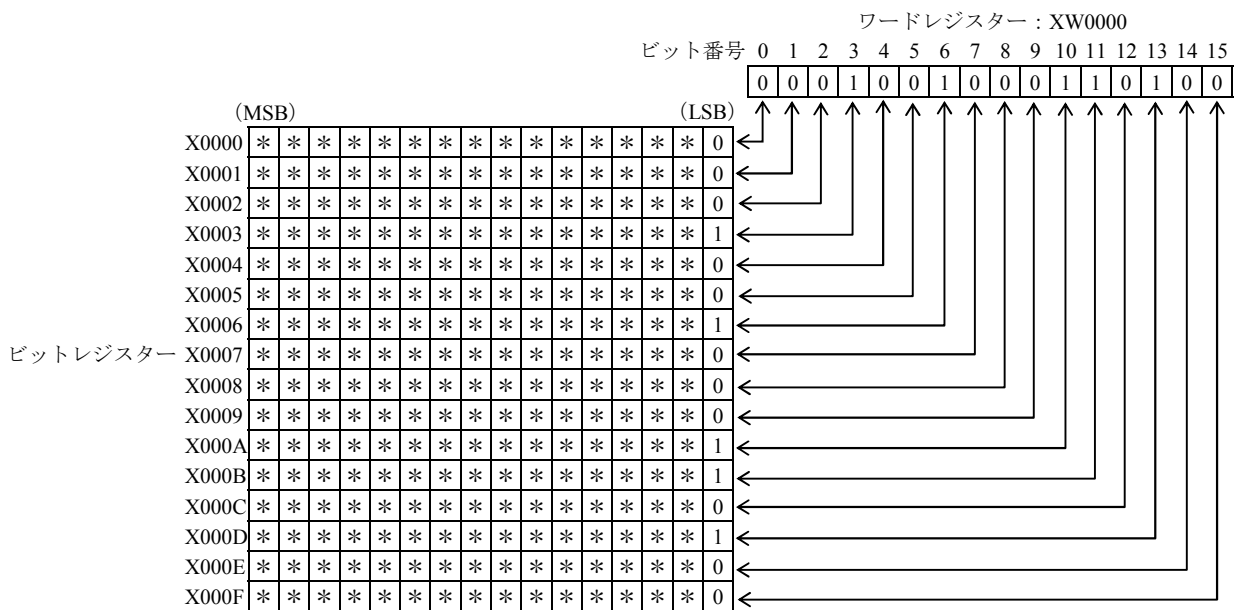
(例2) MOV FW010 -> LB0000実行後MOV LB0000 -> DW000

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. 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X0000～X000FがワードレジスターXW0000に対応していることを示しています。以降、X0010～X001FがXW0010、X0020～X002FがXW0020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X0000～X001FがロングワードレジスターXL0000に対応します。



## 第2章 演算ファンクション

### 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)	X0000	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)	YW0000	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) 定数 (イミディエート) 入力の場合

(a) 10進数入力

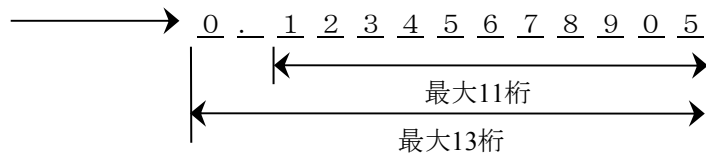
- ① 直接数値を入力 (正の10進数)  $\longrightarrow$  1 2 5 3
- ② + / - を入力後、数値入力  $\longrightarrow$  ± 3 2 1 0 5
- ③ 入力桁数は  $\longrightarrow$  - 1 2 5
  - ・ワードの場合 最大5桁  $\longrightarrow$  (± / -) 3 2 7 6 7
  - ・ロングワードの場合 最大10桁  $\longrightarrow$  (± / -) 2 1 4 7 4 8 3 6 4 7

(b) 16進数入力

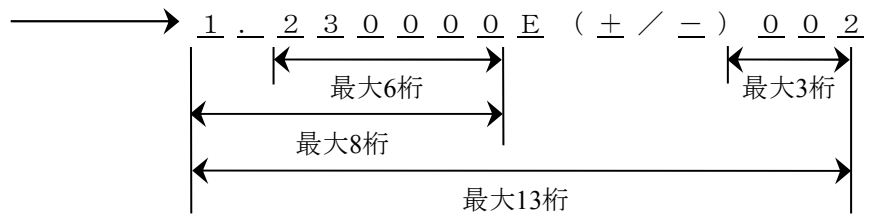
- ① 「H」を入力後、数値入力
- ② 入力桁数は
  - ・ワードの場合 最大4桁  $\longrightarrow$  H 0 5 F 3
  - ・ロングワードの場合 最大8桁  $\longrightarrow$  H 1 2 3 4 A B C D

(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し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H0FFF0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。

## (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	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)
5	ワークレジスタ (ロングワード)	DL, FL, LWL, LXL	
6	ロングワード専用レ ジスタ	BD, LLL, LML	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)
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。

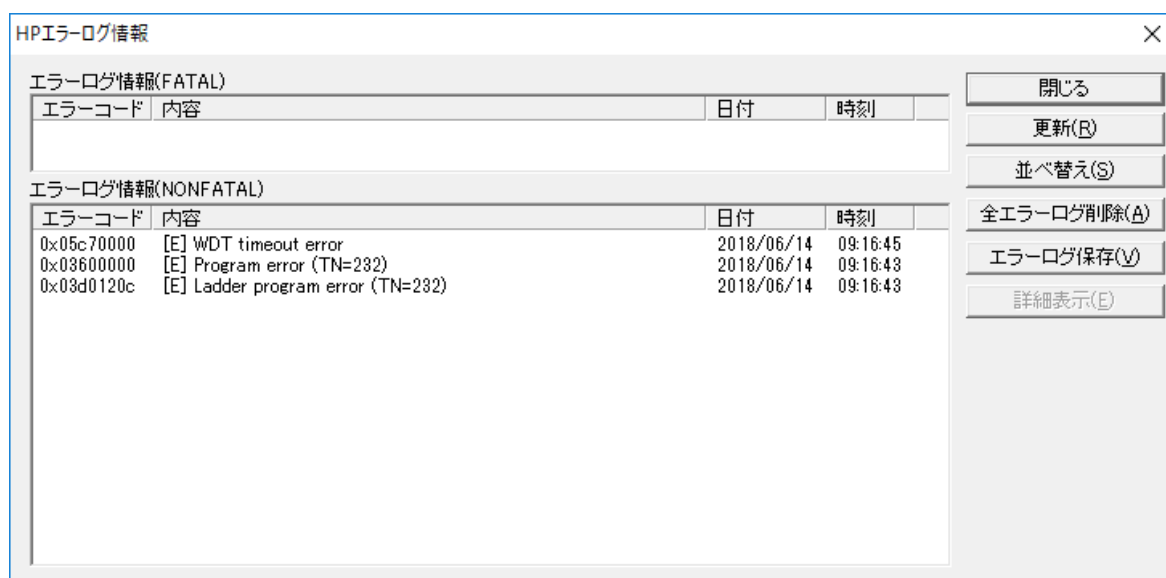
【補足】 BASE SYSTEM/S10VEでの演算ファンクションアドレスエラー発生確認方法

S10VE CPU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BASE SYSTEM/S10VEを使用して切り分けます。

BASE SYSTEM/S10VEでの切り分け手順を以下に示します。

- ① BASE SYSTEM/S10VEを起動します。
- ② [プロジェクト] ⇒ [開く] で該当のプロジェクトを開きます。
- ③ [RAS] ⇒ [エラーログ] ⇒ [HPエラーログ出力] をクリックします。
- ④ [エラーログ情報] 画面が表示されるので、エラーコード「0x03d0120c」、内容「[E] Ladder program error (TN=232)」のエラー情報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。



2.5 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) → (R)	-	-	-	-	-	●	2-23
			ロング		-	-	-	-	-	●	
			フローティング		-	●	-	-	-	-	
	減算	SUB	ワード	(S) - (D) → (R)	-	-	-	-	-	●	2-26
			ロング		-	-	-	-	-	●	
			フローティング		-	●	-	-	-	-	
	+ 1	INC	ワード	(S) + 1 → (S)	-	-	-	-	-	●	2-29
		DEC	ワード	(S) - 1 → (S)	-	-	-	-	-	●	2-31
			ロング		-	-	-	-	-	●	
	乗算	MUL	ワード	(S) × (D) → (R)	-	-	-	-	-	●	2-33
ロング			-		-	-	-	-	●		
フローティング			-		●	-	-	-	-		
除算	DIV	ワード	(S) / (D) → (R) (商)	-	●	-	-	-	●	2-36	
		ロング		-	●	-	-	-	●		
		フローティング		-	●	-	-	-	-		
剰余	MOD	ワード	(S) / (D) → (R) (剰余)	-	●	-	-	-	●	2-39	
スケール演算	SCL	ワード	(S) × (D1) / (D2) → (R)	-	●	-	-	-	●	2-41	
		ロング		-	●	-	-	-	●		
		フローティング		-	●	-	-	-	●		
論理演算命令	論理積	AND	ワード	(S) ∧ (D) → (R)	-	-	-	-	-	-	2-44
	論理和	OR	ワード	(S) ∨ (D) → (R)	-	-	-	-	-	-	2-46
			ロング		-	-	-	-	-	-	
	排他的論理和	EOR	ワード	(S) ⊕ (D) → (R)	-	-	-	-	-	-	2-48
否定	NOT	ワード	$\bar{S}$ → (R)	-	-	-	-	-	-	2-50	
		ロング		-	-	-	-	-	-		
比較命令	=	EQU	ワード	(S) = (D) のとき 1 → (R)	-	-	-	-	-	-	2-52
			ロング	(S) ≠ (D) のとき 0 → (R)	-	-	-	-	-	-	
			フローティング	(S) ≠ (D) のとき 0 → (R)	-	-	-	-	-	-	
	≠	NEQ	ワード	(S) ≠ (D) のとき 1 → (R)	-	-	-	-	-	-	2-54
			ロング	(S) = (D) のとき 0 → (R)	-	-	-	-	-	-	
			フローティング	(S) = (D) のとき 0 → (R)	-	-	-	-	-	-	
	>	GT	ワード	(S) > (D) のとき 1 → (R)	-	-	-	-	-	-	2-56
			ロング	(S) ≤ (D) のとき 0 → (R)	-	-	-	-	-	-	
フローティング			(S) ≤ (D) のとき 0 → (R)	-	-	-	-	-	-		
<	LT	ワード	(S) < (D) のとき 1 → (R)	-	-	-	-	-	-	2-58	
		ロング	(S) ≥ (D) のとき 0 → (R)	-	-	-	-	-	-		
		フローティング	(S) ≥ (D) のとき 0 → (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

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 <sup>°</sup>	(S) ≧ (D) のとき 1 → (R)	-	-	-	-	-	-	2-60
			ロング <sup>°</sup>	(S) < (D) のとき 0 → (R)							
			フローティング <sup>°</sup>								
	≦	LE	ワード <sup>°</sup>	(S) ≦ (D) のとき 1 → (R)	-	-	-	-	-	-	2-62
			ロング <sup>°</sup>	(S) > (D) のとき 0 → (R)							
			フローティング <sup>°</sup>								
テスト	TST	ワード <sup>°</sup> ロング <sup>°</sup> フローティング <sup>°</sup>	(S) をテストし、P, N, Zフラグをセットします。	-	-	●	●	●	-	2-64	
転送命令	転送	MOV	ワード <sup>°</sup> ロング <sup>°</sup> フローティング <sup>°</sup>	(S) → (D)	-	-	-	-	-	-	2-66
	一括転送	MOM	ワード <sup>°</sup> ロング <sup>°</sup>	(S, n) → (D)	-	-	-	-	-	-	2-68
	同一データ一括転送	INI	ワード <sup>°</sup> ロング <sup>°</sup>	(S) → (D, n)	-	-	-	-	-	-	2-70
	交換	EXC	ワード <sup>°</sup> ロング <sup>°</sup>	(S) ↔ (D)	-	-	-	-	-	-	2-72
	FIFO書き込み	PSH	ワード <sup>°</sup>	(S) → FIFOテーブル	-	-	-	-	-	-	2-74
	FIFO読み出し	POP	ワード <sup>°</sup>	FIFOテーブル → (D)	-	-	-	-	-	-	2-76
	FIFO書き込み	PSHO	ワード <sup>°</sup>	(S) → FIFOテーブル	-	-	-	-	-	-	2-78
	FIFO読み出し	POPO	ワード <sup>°</sup>	FIFOテーブル → (D)	-	-	-	-	-	-	2-80
	アドレスセット	AST	ロング <sup>°</sup>	Sのアドレス値 → (D)	-	-	-	-	-	-	2-82
	サーチ	SCH	ワード <sup>°</sup> ロング <sup>°</sup> フローティング <sup>°</sup>	(S) をDからm (サーチステップ数) サーチし、一致No. → (R)	-	-	-	-	-	-	2-84
	変換命令	BIN→FLOAT	BTF	ワード <sup>°</sup> ロング <sup>°</sup>	BIN → FLOAT (S) → (R)	-	-	-	-	-	-
FLOAT→BIN		FTB	ワード <sup>°</sup> ロング <sup>°</sup>	FLOAT → BIN (S) → (R)	-	-	-	-	-	-	2-89
BIN→BCD		BTD	ワード <sup>°</sup> ロング <sup>°</sup>	BIN → BCD (S) → (R)	-	●	-	-	-	●	2-91
BCD→BIN		DTB	ワード <sup>°</sup> ロング <sup>°</sup>	BCD → BIN (S) → (R)	-	●	-	-	-	-	2-93
BIN→7SEG		SEG	ワード <sup>°</sup> ロング <sup>°</sup>	BIN → 7SEG (S) → (R)	-	-	-	-	-	-	2-95

- : 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

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)	-	-	-	-	-	-	2-97	
		ASU	ワード	BIN → ASCII (アンパックモード) (S) → (R)	-	-	-	-	-	-	2-99	
	ASCII→BIN	APB	ワード	ASCII → BIN (パックモード) (S) → (R)	-	●	-	-	-	-	2-101	
		AUB	ワード	ASCII → BIN (アンパックモード) (S) (R)	-	●	-	-	-	-	2-103	
	SINGLE→DOUBLE	STD	ワード	(S) W → (R) L	-	-	-	-	-	-	2-105	
	DOUBLE→SINGLE	DTS	ロング	(S) L → (R) W	-	-	-	-	-	●	2-107	
	絶対値	ABS	ワード	(S)   → (R)	-	-	-	-	-	-	●	2-109
			ロング									
			フローティング									
	+/-	NEG	ワード	- (S) → (R)	-	-	-	-	-	-	●	2-111
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	-	-	-	-	-	-	-	2-113	
		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)	-	●	-	-	-	-	-	2-115	
		ロング										
シフト命令	論理右	LSR	ワード	(S) 右 (D) ビットシフト → 0, (R)	-	-	-	-	-	-	2-117	
			ロング									
	論理左	LSL	ワード	(S) 左 (D) ビットシフト → (R) , 0	-	-	-	-	-	-	-	2-119
			ロング									
算術右	ASR	ワード	(S) 右 (D) ビットシフト → MSB, (R)	-	-	-	-	-	-	-	2-121	
		ロング										
算術左	ASL	ワード	(S) 左 (D) ビットシフト → (R) , V	-	-	-	-	-	●	2-123		
ローテイト命令	右回転	ROR	ワード	(S) 右 (D) ビット回転 → (R)	-	-	-	-	-	-	2-125	
			ロング									
左回転	ROL	ワード	(S) 左 (D) ビット回転 → (R)	-	-	-	-	-	-	-	2-127	
		ロング										
関数処理命令	LIMITER	LIM	ワード	(D1) < (S) .. (D1) → (R)	-	●	-	-	-	-	2-129	
			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)	-	●	-	-	-	●	2-132	
			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



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>△</sup>	$(S) > 0 \dots (S) + (D1) \rightarrow (R)$	-	●	-	-	-	●	2-136
			ロンク <sup>△</sup>	$(S) = 0 \dots 0 \rightarrow (R)$	-	●	-	-	-	●	
			フローティング <sup>△</sup>	$(S) < 0 \dots (S) + (D2) \rightarrow (R)$	-	●	-	-	-	-	
	平方根	SQR	ワート <sup>△</sup>	$(S) \geq 0 \dots \text{SQR}(S) \rightarrow (R)$	-	-	-	-	-	-	2-140
			ロンク <sup>△</sup>	$(S) < 0 \dots 0 \rightarrow (R)$	-	-	-	-	-	-	
			フローティング <sup>△</sup>		-	-	-	-	-	-	
	サイン	SIN	フローティング <sup>△</sup>	$\text{SIN}(S) \rightarrow (R)$	-	-	-	-	-	-	2-143
	コサイン	COS	フローティング <sup>△</sup>	$\text{COS}(S) \rightarrow (R)$	-	-	-	-	-	-	2-145
	タンジェント	TAN	フローティング <sup>△</sup>	$\text{TAN}(S) \rightarrow (R)$	-	●	-	-	-	-	2-147
	アークサイン	ASIN	フローティング <sup>△</sup>	$\text{SIN}^{-1}(S) \rightarrow (R)$	-	●	-	-	-	-	2-149
	アークコサイン	ACOS	フローティング <sup>△</sup>	$\text{COS}^{-1}(S) \rightarrow (R)$	-	●	-	-	-	-	2-151
	アークタンジェント	ATAN	フローティング <sup>△</sup>	$\text{TAN}^{-1}(S) \rightarrow (R)$	-	-	-	-	-	-	2-153
	指数演算	EXP	フローティング <sup>△</sup>	$\text{EXP}(S) \rightarrow (R)$	-	●	-	-	-	-	2-155
自然対数	LOG	フローティング <sup>△</sup>	$\text{LOG}(S) \rightarrow (R)$	-	●	-	-	-	-	2-157	
最大値	MAX	ワート <sup>△</sup>	$(S) \geq (R) \dots (S) \rightarrow (R)$	-	-	-	-	-	-	2-159	
		ロンク <sup>△</sup>	$(S) < (R) \dots (D) \rightarrow (R)$	-	-	-	-	-	-		
		フローティング <sup>△</sup>		-	-	-	-	-	-		
最小値	MIN	ワート <sup>△</sup>	$(S) \leq (R) \dots (S) \rightarrow (R)$	-	-	-	-	-	-	2-161	
		ロンク <sup>△</sup>	$(S) > (R) \dots (D) \rightarrow (R)$	-	-	-	-	-	-		
		フローティング <sup>△</sup>		-	-	-	-	-	-		
特殊命令	クリア	XCLR	-	Xのエリアをクリアします。	-	-	-	-	-	-	2-163
		YCLR	-	Yのエリアをクリアします。	-	-	-	-	-	-	2-163
		GCLR	-	Gのエリアをクリアします。	-	-	-	-	-	-	2-163
		RCLR	-	Rのエリアをクリアします。	-	-	-	-	-	-	2-163
		KCLR	-	Kのエリアをクリアします。	-	-	-	-	-	-	2-163
		TCLR	-	Tのエリアおよび計数値をクリアします。	-	-	-	-	-	-	2-163
		UCLR	-	Uのエリアおよび計数値をクリアします。	-	-	-	-	-	-	2-163
		CCLR	-	Cのエリアおよび計数値をクリアします。	-	-	-	-	-	-	2-163
		VCLR	-	Vのエリアをクリアします。	-	-	-	-	-	-	2-163
		ECLR	-	Eのエリアをクリアします。	-	-	-	-	-	-	2-163
		FCLR	-	演算結果フラグをクリアします。	-	-	-	-	-	-	2-163

- ：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

(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	—	—	—	—	—	—	2-165
	無条件	JMP	—	無条件で指定したラベルへジャンプ	—	—	—	—	—	—	2-167
	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	—	—	—	—	—	—	2-169
イーサネット通信命令	TCP通信	TOP	—	TCP接続のオープン	—	—	—	—	—	—	2-183
		TPOP	—	TCP接続のオープン	—	—	—	—	—	—	2-185
		TCLO	—	TCP接続のクローズ	—	—	—	—	—	—	2-187
		TRCV	—	TCP受信	—	—	—	—	—	—	2-189
		TSND	—	TCP送信	—	—	—	—	—	—	2-192
	UDP通信	UOP	—	UDPのオープン	—	—	—	—	—	—	2-194
		UCLO	—	UDPのクローズ	—	—	—	—	—	—	2-196
		URCV	—	UDP受信	—	—	—	—	—	—	2-198
		USND	—	UDP送信	—	—	—	—	—	—	2-201

—：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, Y01FF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

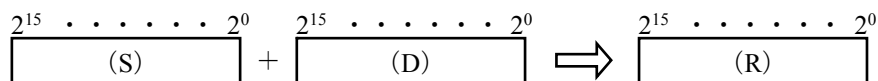
R : リザルト (演算結果) を格納するレジスタ

(\*) 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

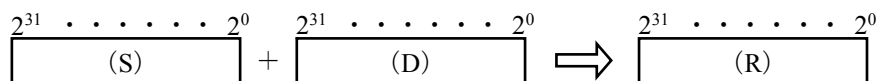
ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータを加算し、結果をリザルト (R) に格納します。



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。

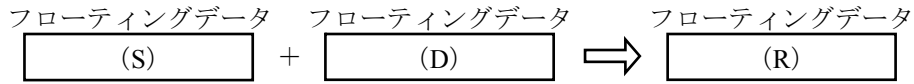


ソース (S) 、デスティネーション (D) 、リザルト (R) に指定できる値および格納される値の範囲は、-2147483648~2147483647です。

## 第2章 演算ファンクション

### ● 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

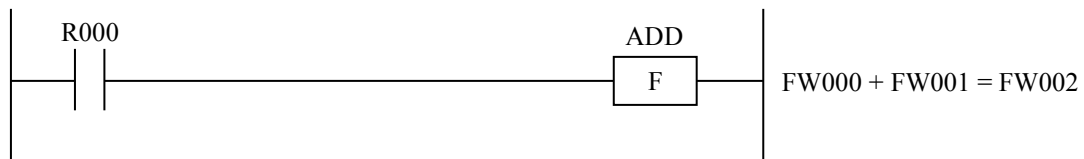
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○：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。

(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  
(インダイレクト指定のときのみ)

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

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。

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

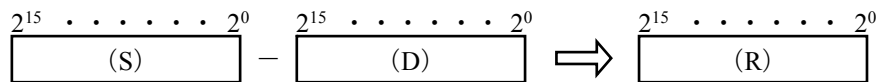
R : リザルト (演算結果) を格納するレジスタまたは定数

(\*) 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

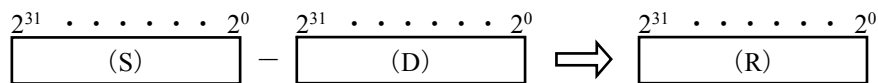
ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータを減算し、結果をリザルト (R) に格納します。



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

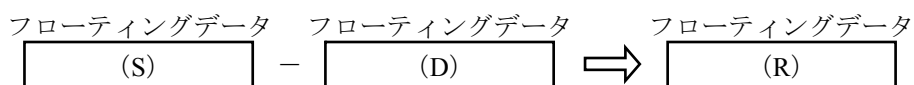
ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータを減算し、結果をリザルト (R) に格納します。



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

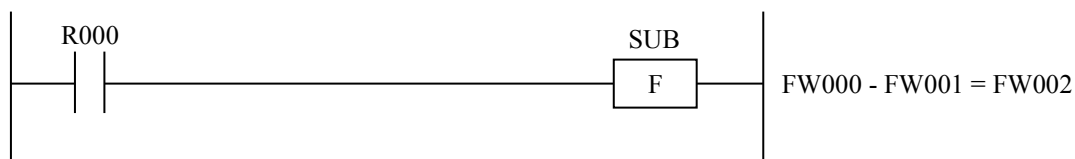
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	-	○	-	○	-	○

○：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。



## 第2章 演算ファンクション

### (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  
(インダイレクト指定のときのみ)

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

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。

INC +1 : INCREMENT

(1) 入力形式

INC S

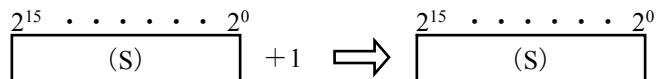
S : INC (+1) を行うデータの格納レジスタ

(\*) 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

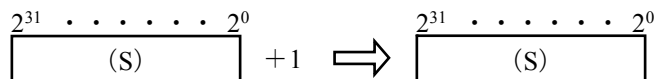
ソース (S) で指定された16ビットデータの内容を+1します。



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

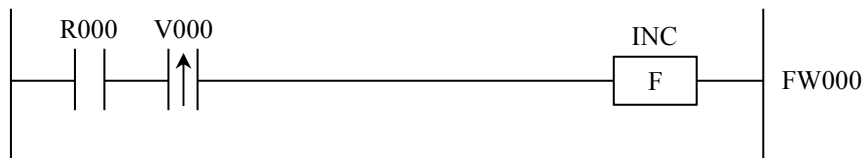
	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	—	○	—	—	—	○

○ : 指定できます。

— : 指定できません。

## 第2章 演算ファンクション

### (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

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

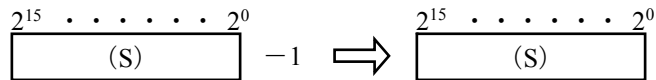
S : DEC (-1) を行うデータの格納レジスター

(\* ) 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

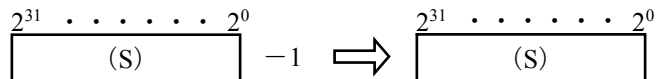
ソース (S) で指定された16ビットデータの内容を-1します。



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

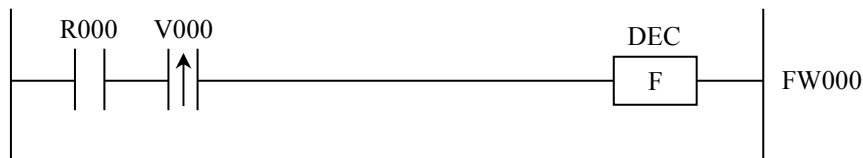
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	—	○	—	—	—	○

○ : 指定できます。

— : 指定できません。

## 第2章 演算ファンクション

### (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

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

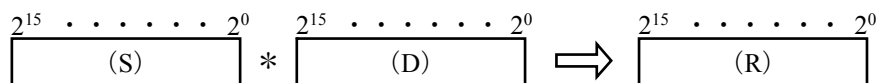
R : リザルト (演算結果) を格納するレジスタ

(\* 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

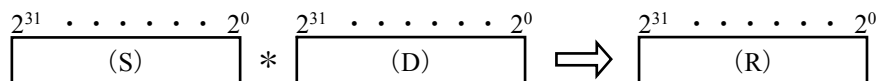
ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータを乗算し、結果をリザルト (R) に格納します。



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。

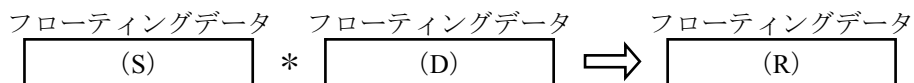


ソース (S) 、デスティネーション (D) 、リザルト (R) に指定できる値および格納される値の範囲は、-2147483648~2147483647です。

## 第2章 演算ファンクション

### ● 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

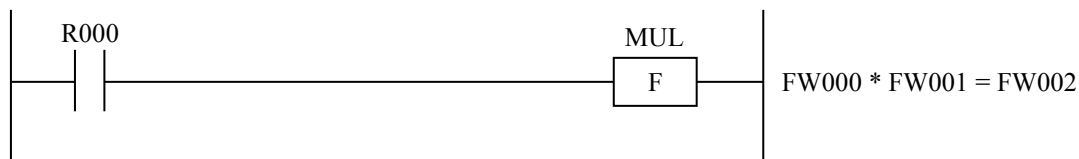
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○：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。

(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  
(インダイレクト指定のときのみ)

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

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。



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

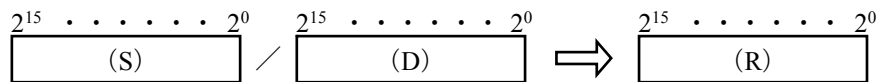
R : リザルト (演算結果) を格納するレジスタ

(\*) 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

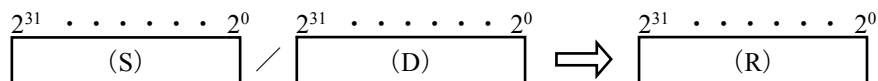
ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータを除算し、結果 (商) をリザルト (R) に格納します。



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

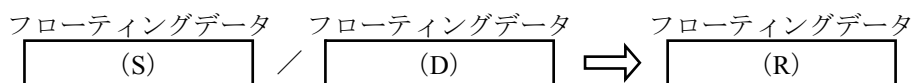
ソース (S) で指定された32ビットデータとデスティネーション (D) で指定された32ビットデータを除算し、結果 (商) をリザルト (R) に格納します。



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

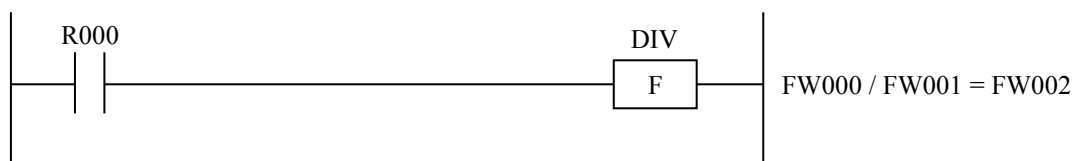
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○：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。

## 第2章 演算ファンクション

### (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、それ以外のとき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）。

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  
(インダイレクト指定のときのみ)

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

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。

## 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

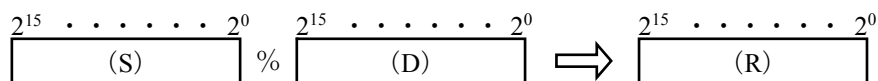
R : リザルト (演算結果) を格納するレジスタ

(\*) 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

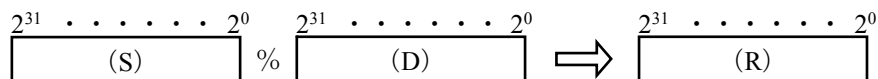
ソース (S) で指定された16ビットデータとデスティネーション (D) で指定された16ビットデータを除算し、除算結果の剰余をリザルト (R) に格納します。



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。

## 第2章 演算ファンクション

### (3) データタイプ

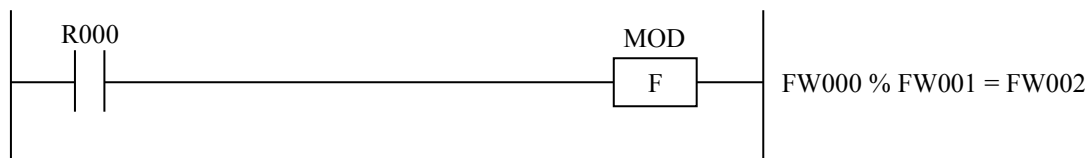
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
D	○	○	○	○	—	—	○
R	○	—	○	—	—	—	○

○：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、それ以外るとき0

ロングワードデータ時

- ・商 = 2147483648のとき1、それ以外るとき0

E：デスティネーション (D) = 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0が格納されます。

## SCL スケール演算 : SCALE

## (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

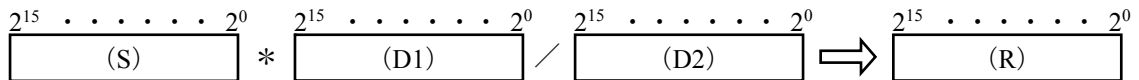
R : リザルト (演算結果) を格納するレジスタ

(\*) 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

## ● ワードデータのスケール演算

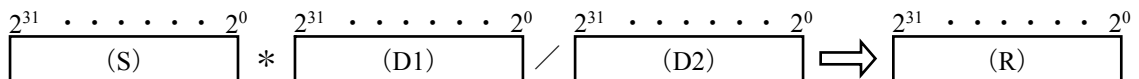
ソース (S) で指定された16ビットデータとデスティネーション (D1) / (D2) を乗算し、結果をリザルト (R) に格納します。



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。

## ● ロングワードデータのスケール演算

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 (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、-2147483648~2147483647です。

## 第2章 演算ファンクション

### ● フローティングデータのスケール演算

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 (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。

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

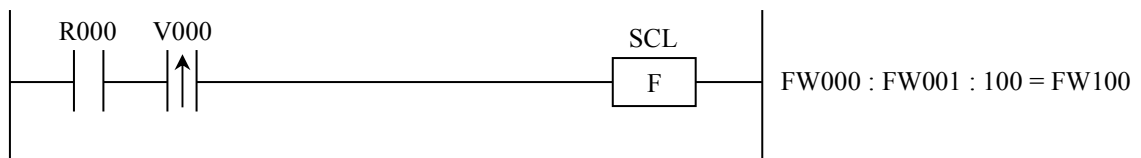
### (3) データタイプ

	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	○	—	○	—	○	—	○

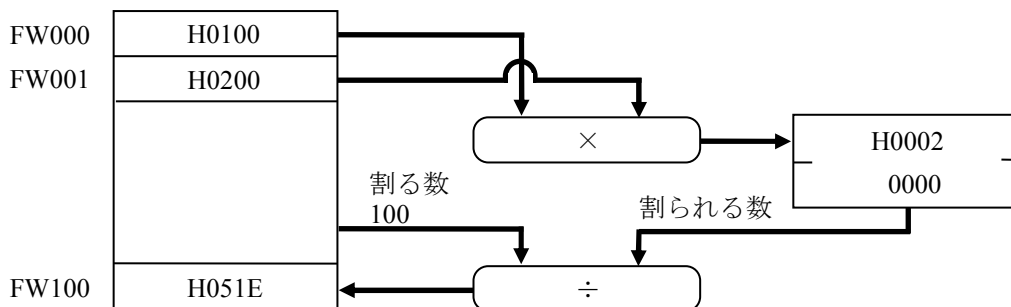
○ : 指定できます。

— : 指定できません。

### (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：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

- ・D=0のとき1、それ以外のとき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

(インダイレクト指定または、ロングワードサイズ指定のみ)

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

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



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

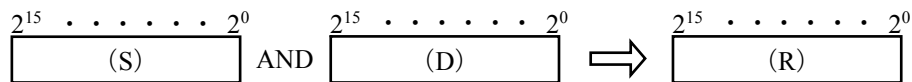
R : リザルト (演算結果) を格納するレジスタ

(\* ) 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

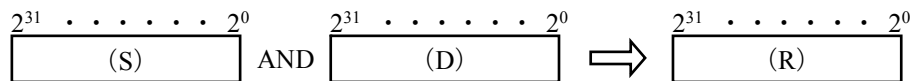
● ワードデータの論理積

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

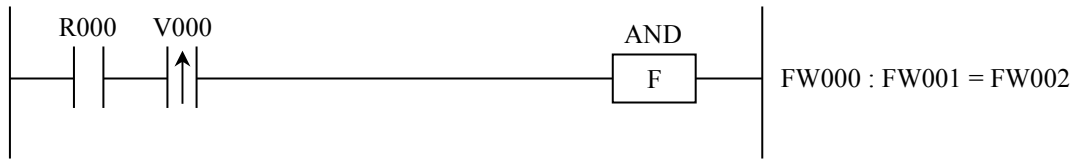
	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	—	—	○
D	○	○	○	○	—	—	○
R	○	—	○	—	—	—	○

○ : 指定できます。

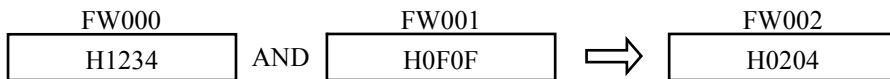
— : 指定できません。

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

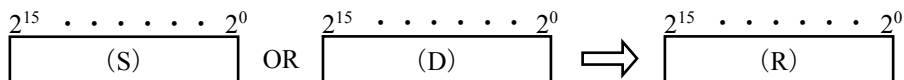
R : リザルト (演算結果) を格納するレジスタ

(\* ) 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

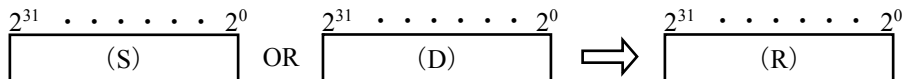
● ワードデータの論理和

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

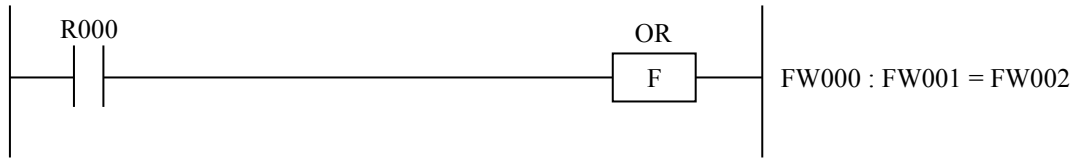
	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	—	—	○
D	○	○	○	○	—	—	○
R	○	—	○	—	—	—	○

○ : 指定できます。

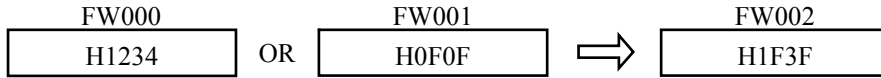
— : 指定できません。

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

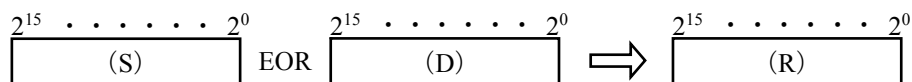
R : リザルト (演算結果) を格納するレジスタ

(\* ) 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

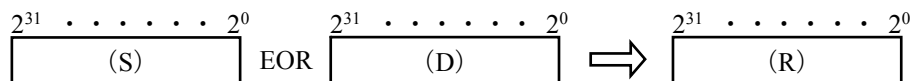
● ワードデータの排他的論理和

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

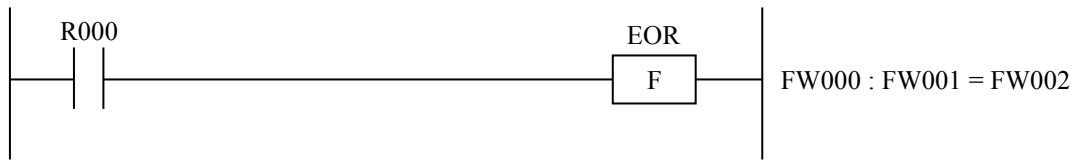
	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	—	—	○
D	○	○	○	○	—	—	○
R	○	—	○	—	—	—	○

○ : 指定できます。

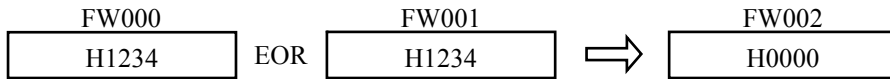
— : 指定できません。

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

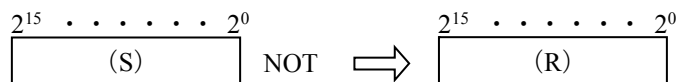
R : リザルト (演算結果) を格納するレジスタ

(\*) 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。「->」は省略できます。

(2) 機能

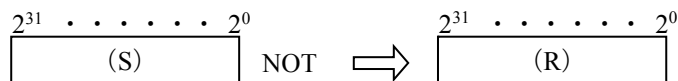
● ワードデータの否定

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

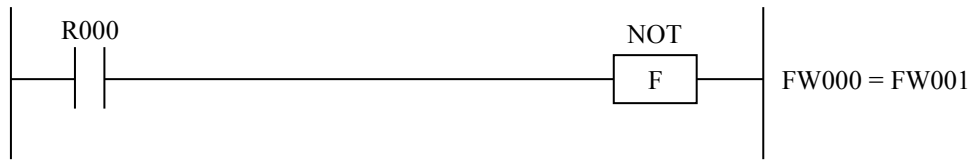
	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	—	—	○
R	○	—	○	—	—	—	○

○ : 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



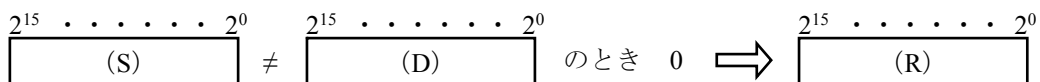
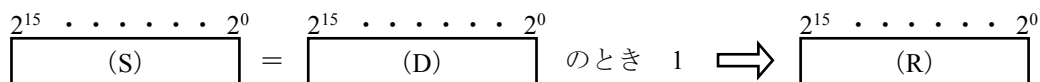
S : ソース格納レジスタまたは定数  
 D : デスティネーション格納レジスタまたは定数  
 R : リザルト (演算結果) を格納するレジスタ

(\*) 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

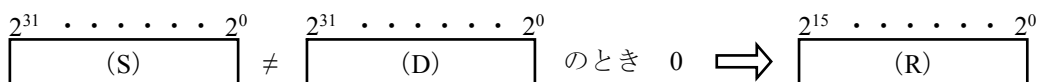
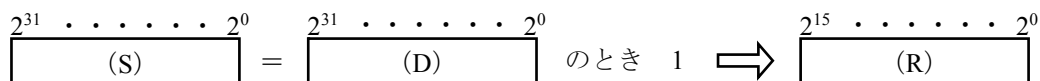
● ワードデータの比較

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。

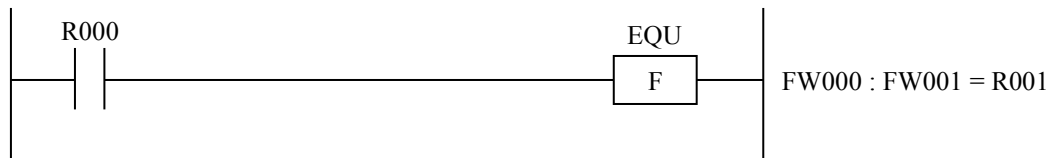
(3) データタイプ

	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	○	—	—	—	—	—	○

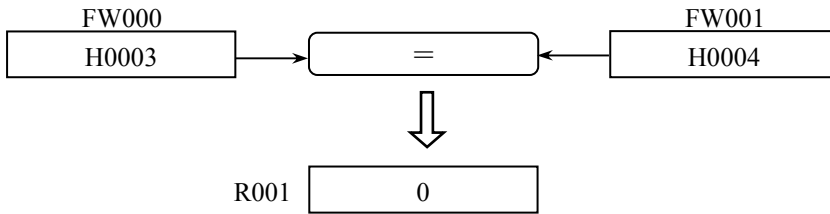
○：指定できます。  
—：指定できません。

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 ≠ : NOT EQUAL

(1) 入力形式



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

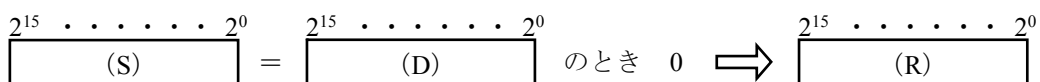
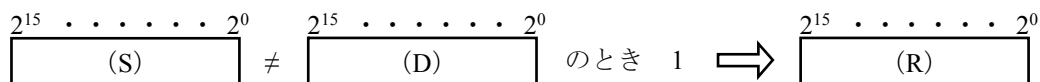
R : リザルト (演算結果) を格納するレジスタ

(\*) ファンクション名称とパラメータおよびパラメータとパラメータの間には、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

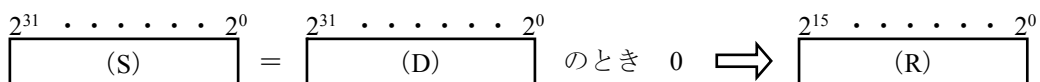
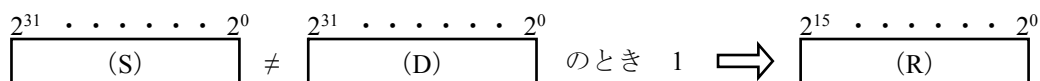
● ワードデータの比較

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。



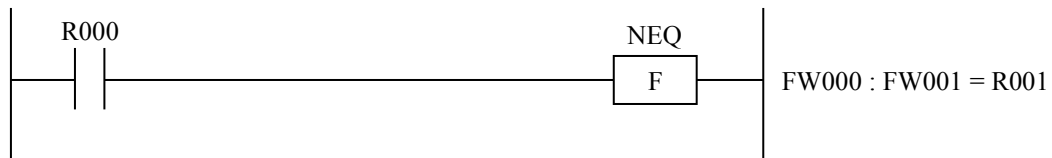
(3) データタイプ

	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	○	—	—	—	—	—	○

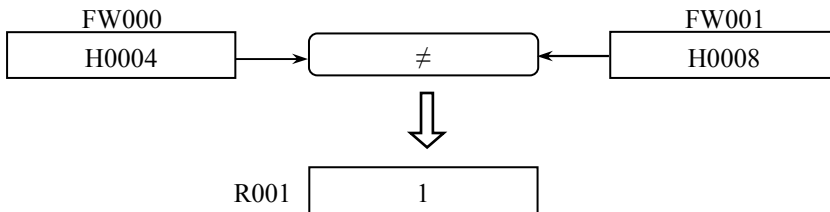
○：指定できます。  
—：指定できません。

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



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

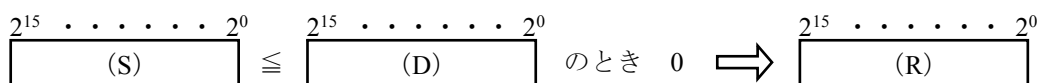
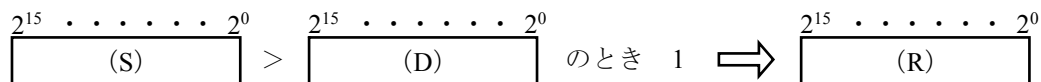
R : リザルト (演算結果) を格納するレジスタ

(\* ) 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

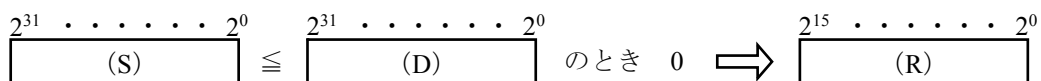
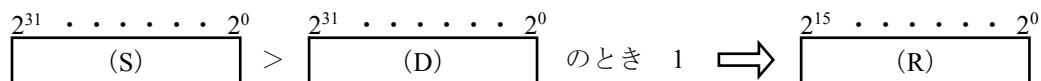
● ワードデータの比較

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。



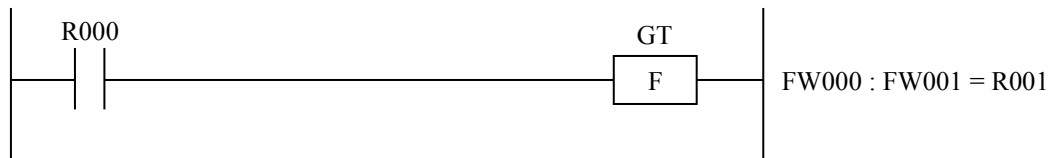
(3) データタイプ

	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	○	—	—	—	—	—	○

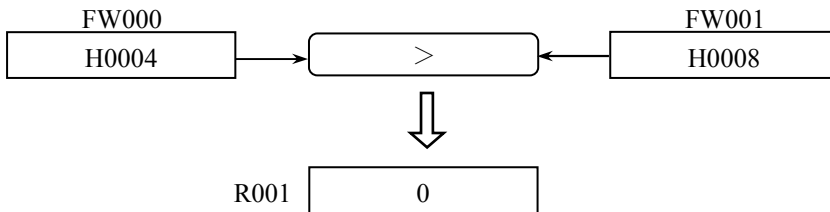
○：指定できます。  
—：指定できません。

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



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

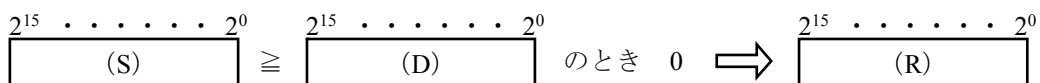
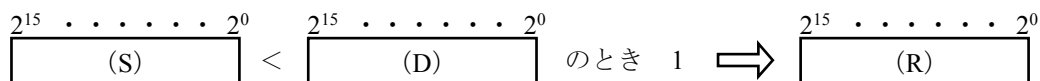
R : リザルト (演算結果) を格納するレジスタ

(\*) ファンクション名称とパラメータおよびパラメータとパラメータの間には、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

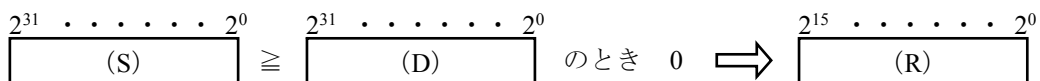
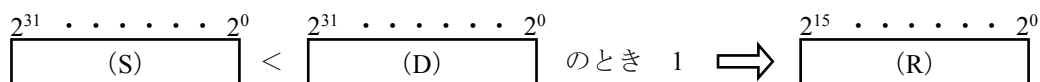
● ワードデータの比較

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。



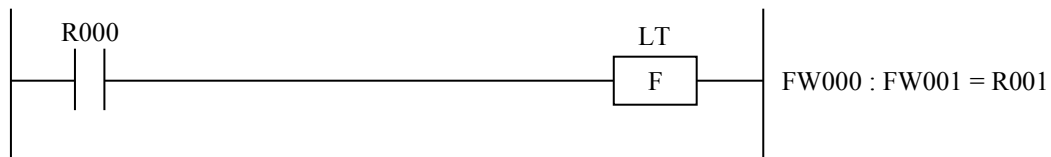
(3) データタイプ

	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	○	—	—	—	—	—	○

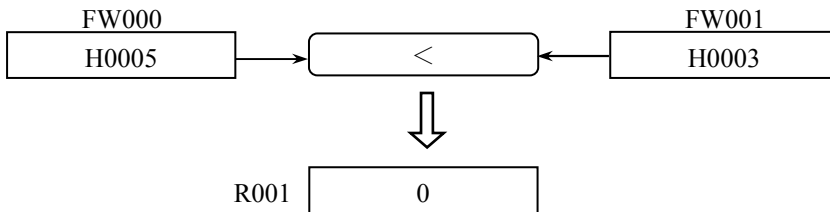
○：指定できます。  
—：指定できません。

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



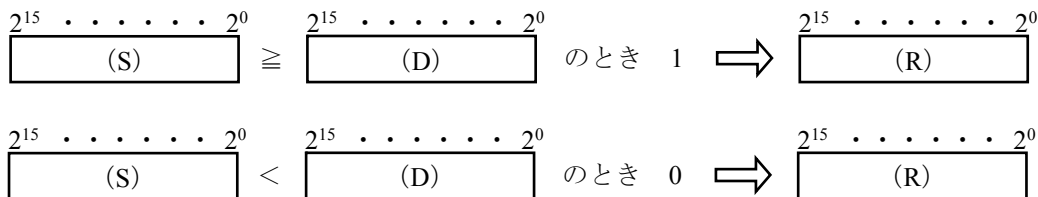
- S : ソース格納レジスタまたは定数
- D : デスティネーション格納レジスタまたは定数
- R : リザルト (演算結果) を格納するレジスタ

(\* 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

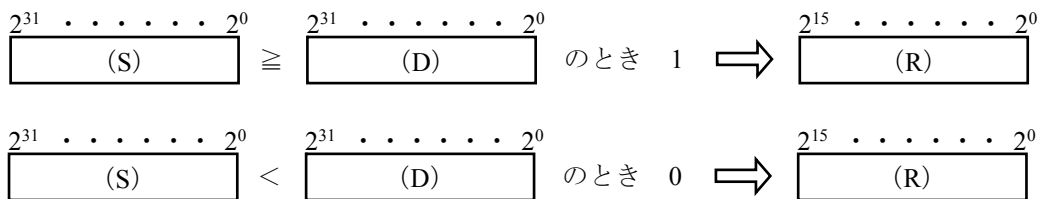
● ワードデータの比較

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。



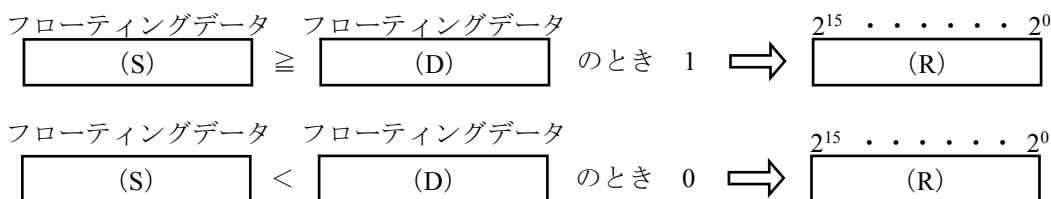
● ロングワードデータの比較

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。



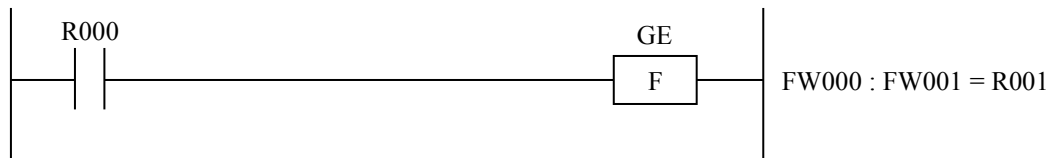
(3) データタイプ

	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	○	—	—	—	—	—	○

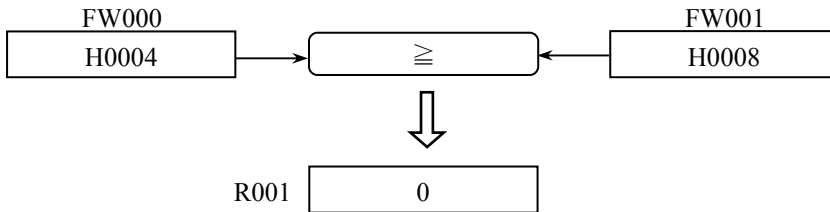
○：指定できます。  
—：指定できません。

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



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

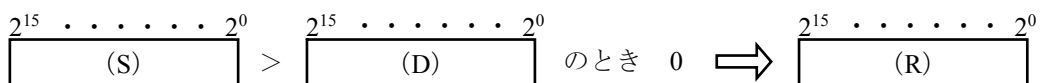
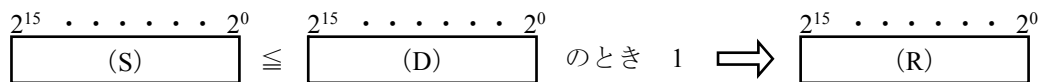
R : リザルト (演算結果) を格納するレジスタ

(\* ) ファンクション名称とパラメータおよびパラメータとパラメータの間には、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

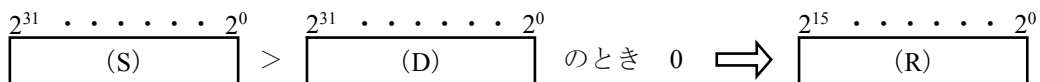
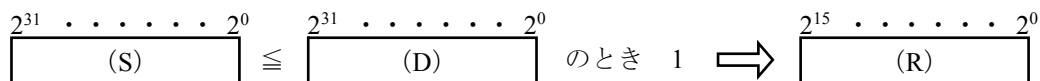
● ワードデータの比較

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	○	—	—	—	—	—	○

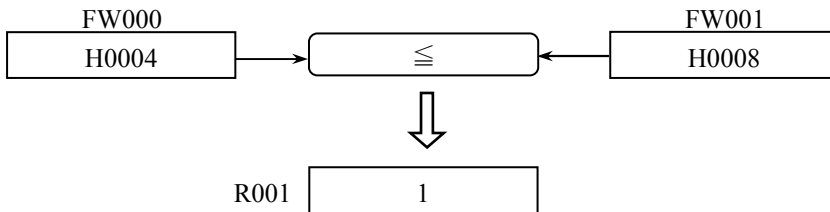
○：指定できます。  
—：指定できません。

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

(\*) 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

$$\begin{matrix} \text{フローティングデータ} \\ \boxed{(S)} \end{matrix} > 0 \quad : \text{P ON (N, Z OFF)}$$

$$\begin{matrix} \text{フローティングデータ} \\ \boxed{(S)} \end{matrix} = 0 \quad : \text{Z ON (P, N OFF)}$$

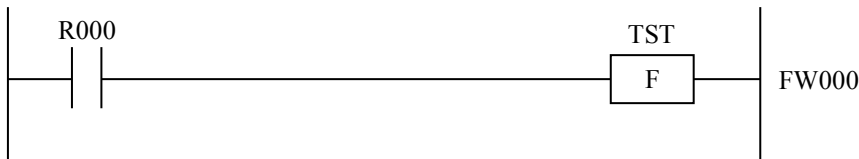
$$\begin{matrix} \text{フローティングデータ} \\ \boxed{(S)} \end{matrix} < 0 \quad : \text{N ON (P, Z OFF)}$$

(3) データタイプ

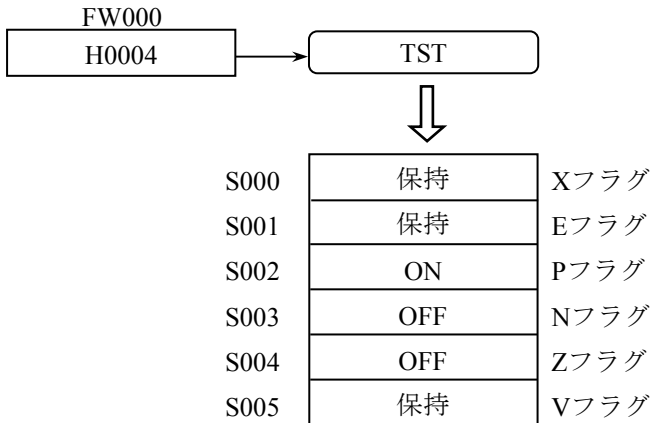
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	—	○	—	○	—	○

○：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。

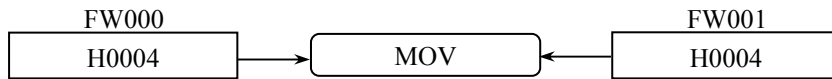
— : 指定できません。

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ー

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

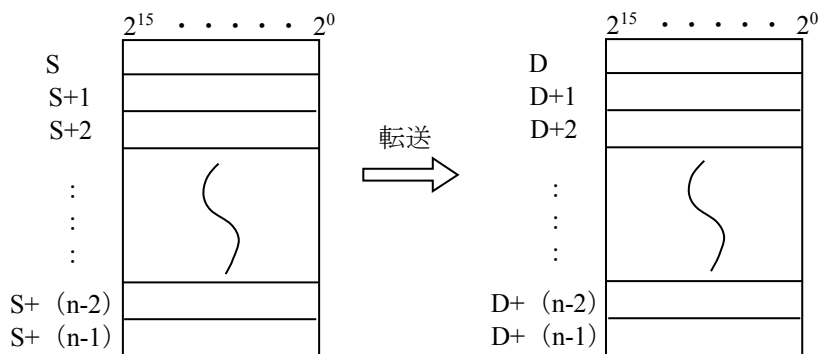
(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

(2) 機能

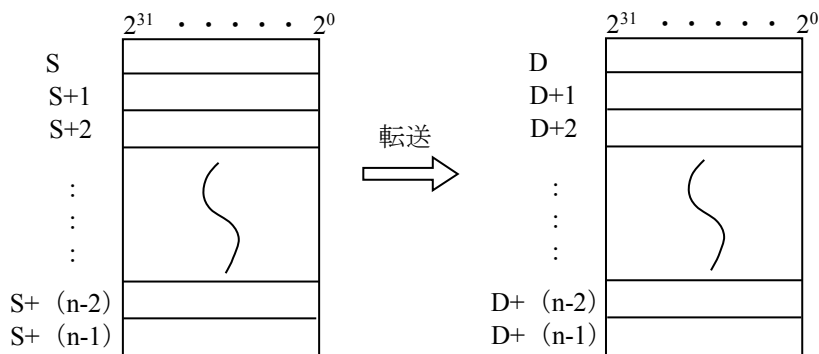
ソース (S) からnステップをデスティネーション (D) からnステップに転送します。

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



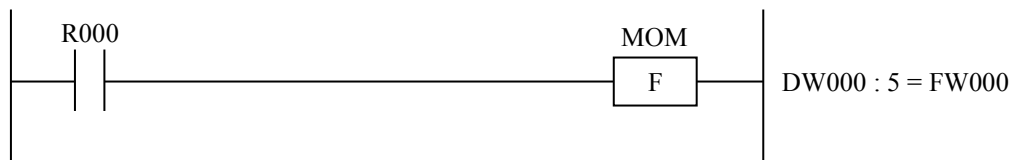
(3) データタイプ

	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	○	—	○	—	○	—	○

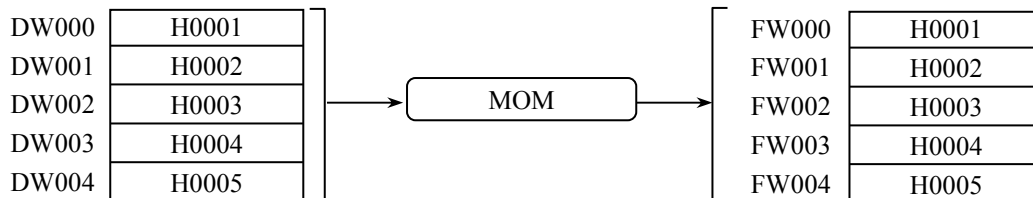
○：指定できます。  
—：指定できません。

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

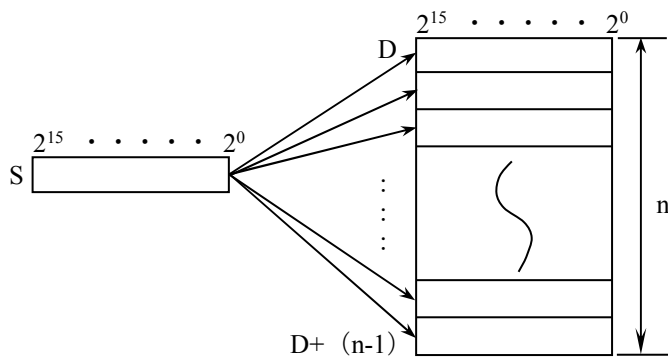
(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

(2) 機能

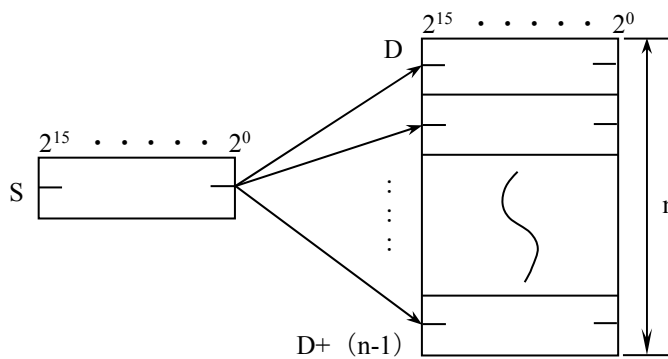
ソース (S) の内容をデスティネーション (D) からnステップに転送します。

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



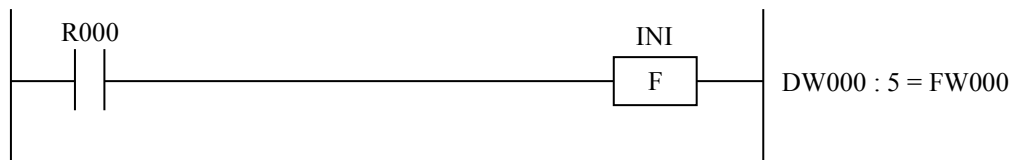
(3) データタイプ

	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	○	—	○	—	○	—	○

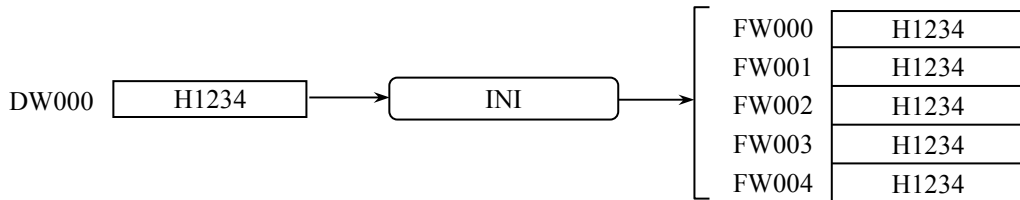
○：指定できます。  
—：指定できません。

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

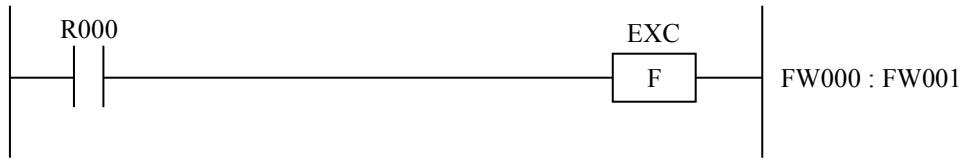
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	—	○	—	○	—	○
D	○	—	○	—	○	—	○

○ : 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
```

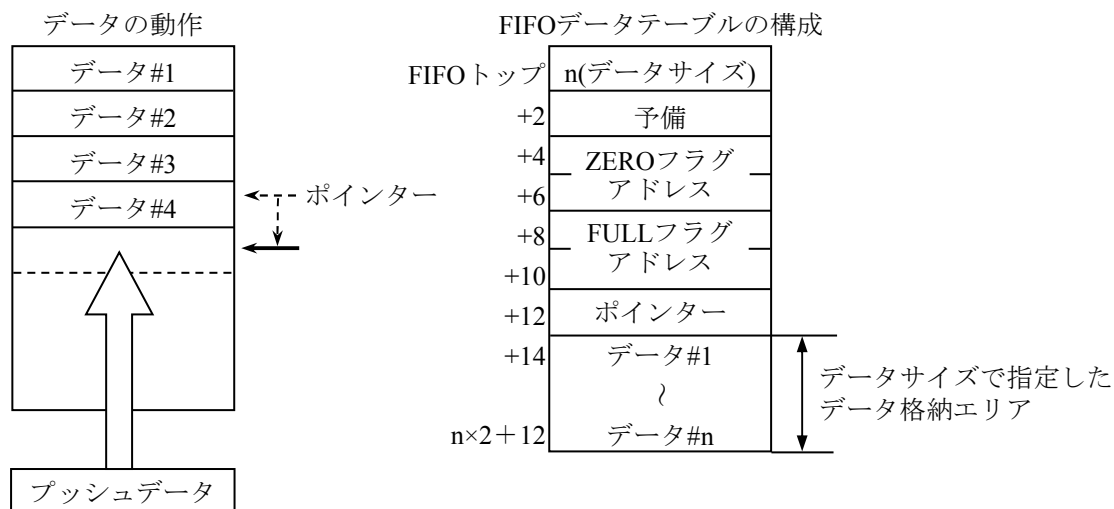
S : ソース格納レジスター

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)

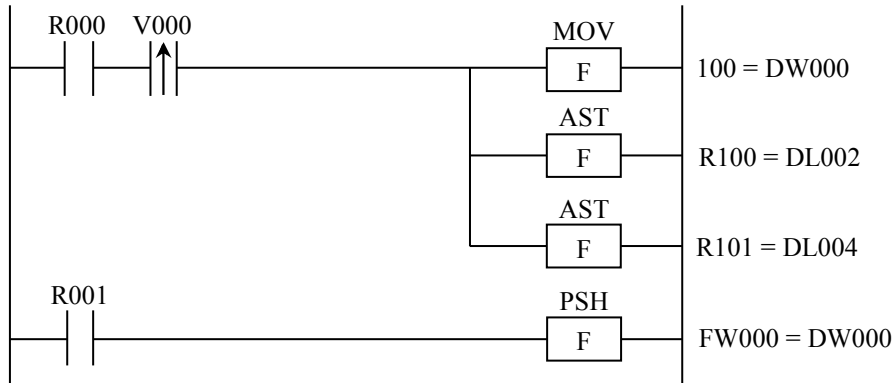
- プッシュ処理前にポインターがnであれば、FULLフラグをONし、プッシュ無処理とします (ただし、ZEROフラグはOFFします)。また、プッシュ処理後、ポインターをインクリメントしてnになった場合も、FULLフラグをONします。それ以外の場合は、FULLフラグをOFFします。
- 演算実行後、ZEROフラグはOFFします (下記条件により無処理の場合を除きます)。
- データサイズ  $n \leq 0$  または  $256 < n$  の場合は、無処理です。
- ポインター  $< 0$  または  $n < \text{ポインター}$  の場合は、無処理です。

(3) データタイプ

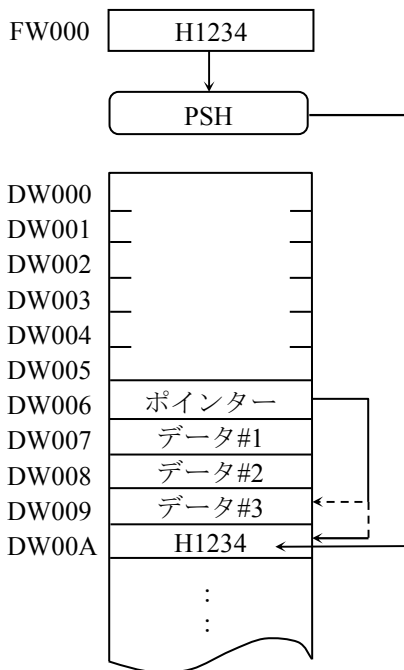
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	—	—	—	—	—	○
TB	○	—	—	—	—	—	○

○ : 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

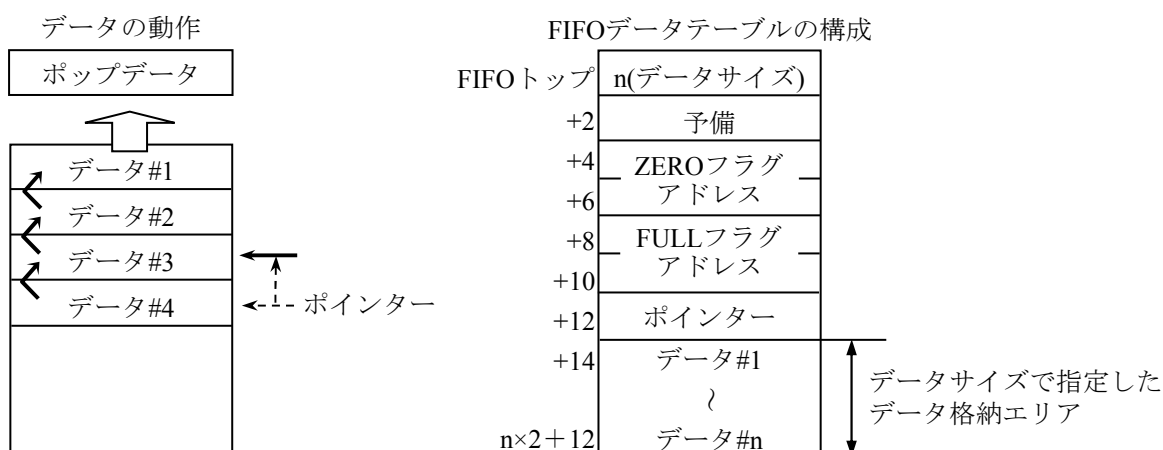
TB : FIFOテーブル先頭アドレス (レジスター)

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

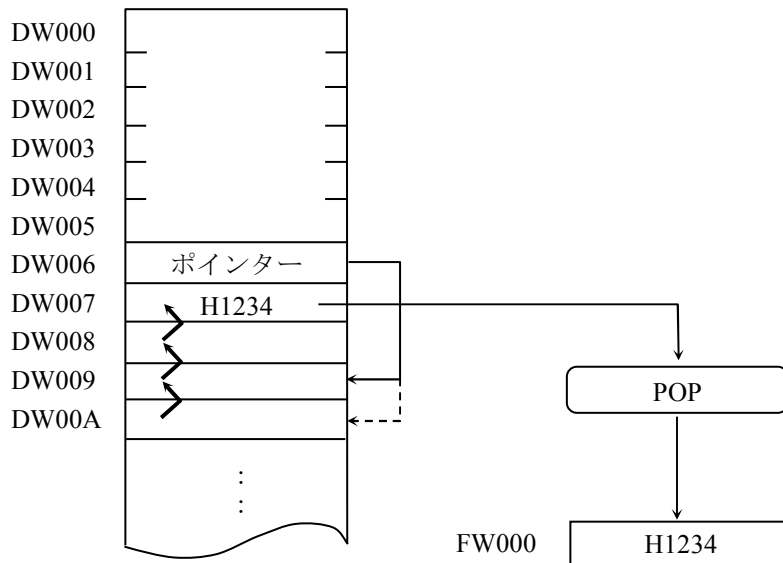
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
D	○	—	—	—	—	—	○
TB	○	—	—	—	—	—	○

○：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よりの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

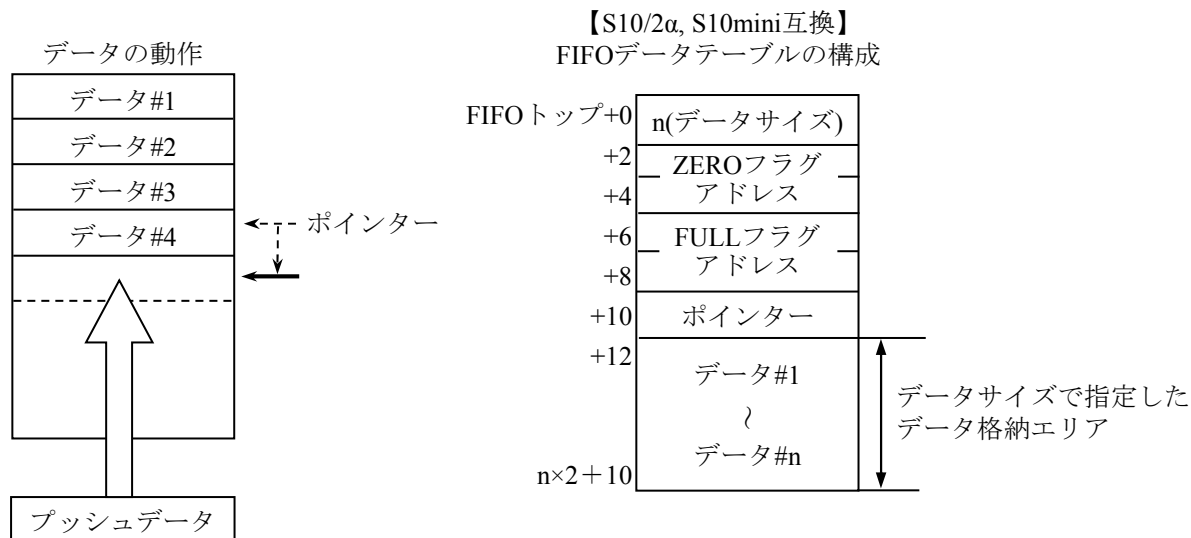
S : ソース格納レジスタ

TB : FIFOテーブル先頭レジスタ

(\*) 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)

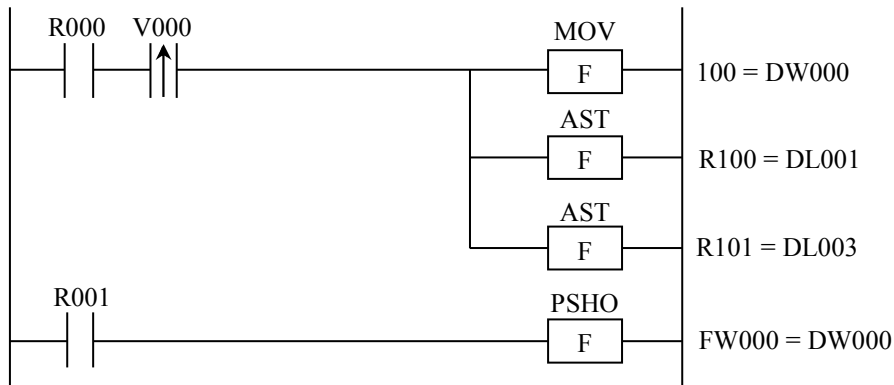
- プッシュ処理前にポインタがnであれば、FULLフラグをONし、プッシュ無処理とします (ただし、ZEROフラグはOFFします)。また、プッシュ処理後、ポインタをインクリメントしてnになった場合も、FULLフラグをONします。それ以外の場合は、FULLフラグをOFFします。
- 演算実行後、ZEROフラグはOFFします (下記条件により無処理の場合を除きます)。
- データサイズ $n \leq 0$ または $256 < n$ の場合は、無処理です。
- ポインタ $< 0$ の場合は、無処理です。
- $n < \text{ポインタ}$ の場合、FULLフラグがONします。

(3) データタイプ

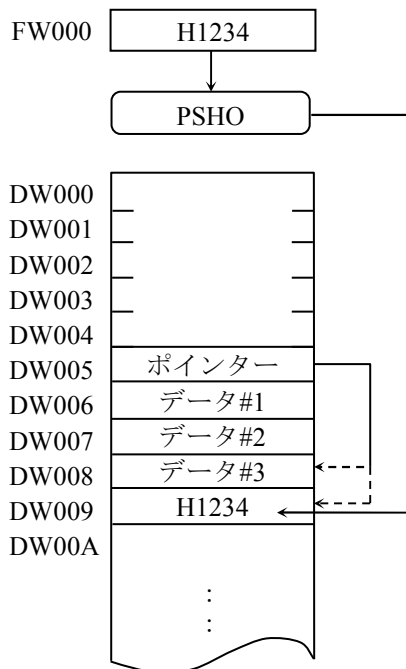
	ワード		ロングワード		フローティング		インデックス指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	—	—	—	—	—	○
TB	○	—	—	—	—	—	○

○ : 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

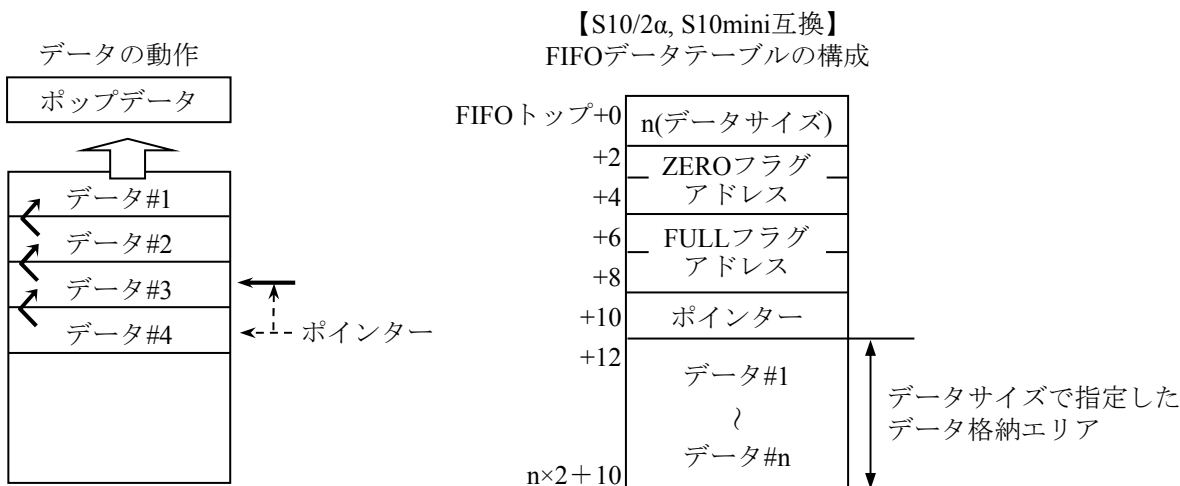
TB : FIFOテーブル先頭アドレス (レジスター)

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  $< 0$  の場合は、無処理です。
- $n < \text{ポインター}$  の場合、ZEROフラグがONします。

(3) データタイプ

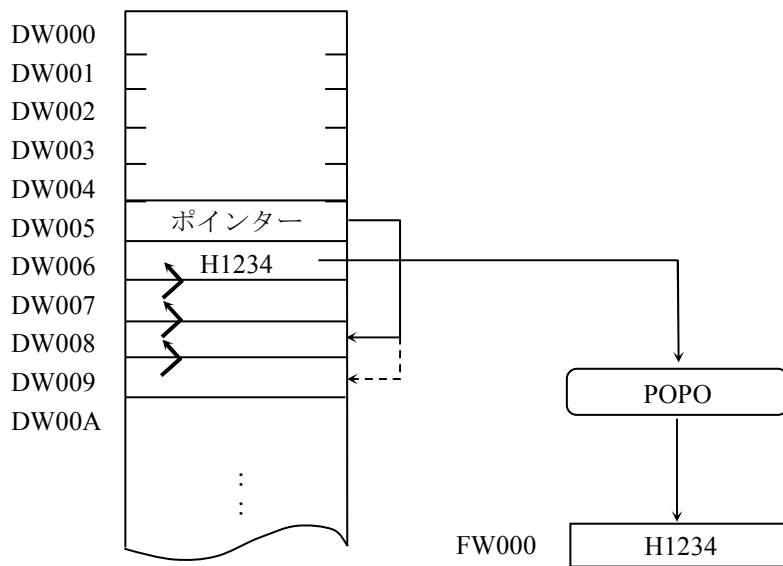
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
D	○	—	—	—	—	—	○
TB	○	—	—	—	—	—	○

○：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よりの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

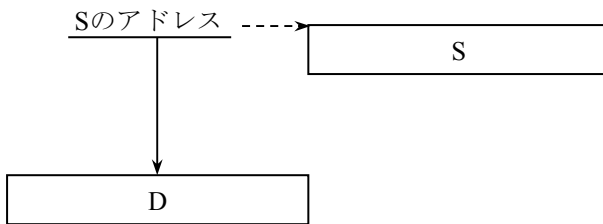
S : ソース格納レジスタ

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	—	—	○	—	—	—	○

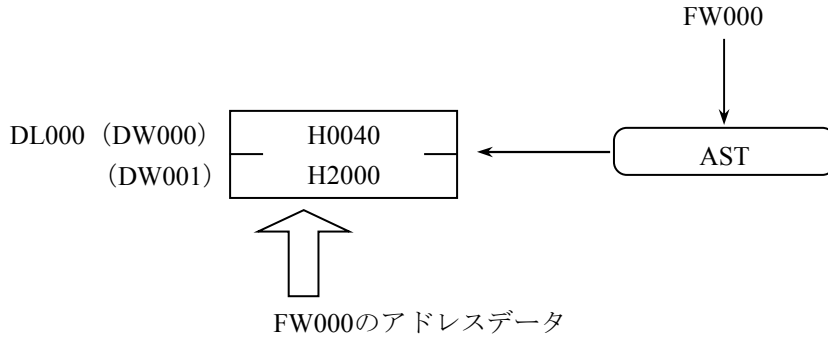
○ : 指定できます。

— : 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

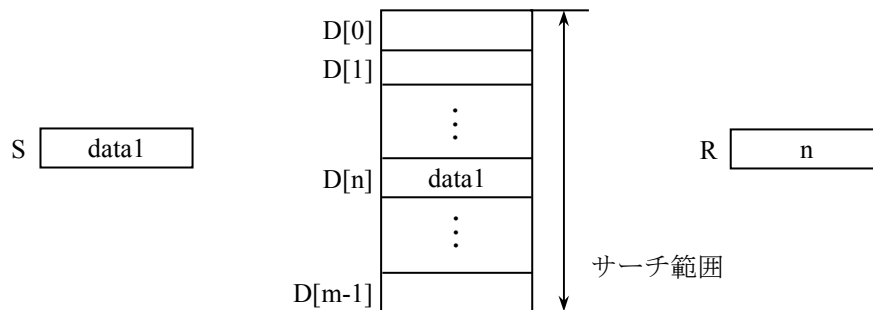
- S : ソース格納レジスタまたは定数
- D : デスティネーション格納レジスタ
- m : サーチステップ数 (定数およびワードレジスタ)
- R : リザルト (演算結果) を格納するレジスタ

(\* ) 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

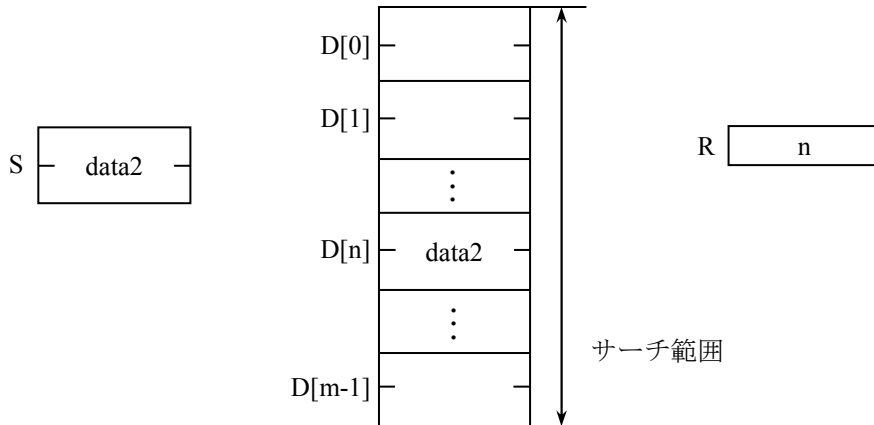
ソース (S) で指定されたワードデータの内容が、指定されたデスティネーション (D) から m ステップ数内に存在するかどうかサーチし、ソース (S) で指定されたワードデータと一致するデータを探し出すまでのステップ数を指定されたリザルト (R) に格納します。



- 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 (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。

● ログワードデータのサーチ

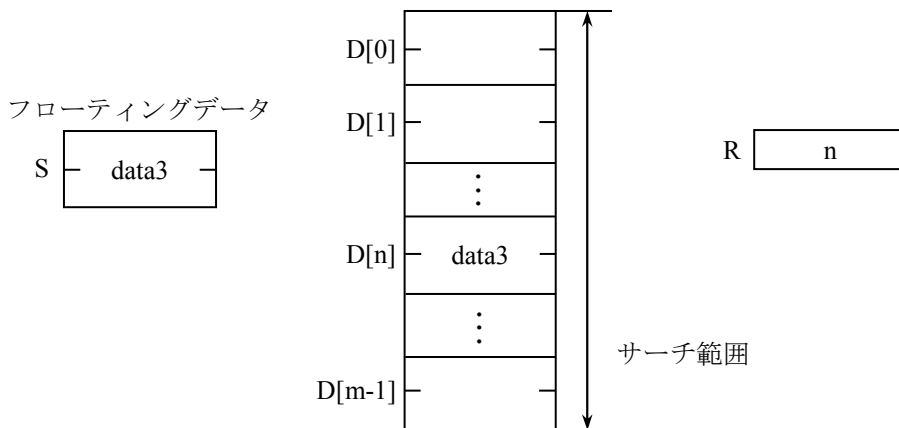
ソース (S) で指定されたログワードデータの内容が、指定されたデスティネーション (D) からmステップ数内に存在するかどうかをサーチし、ソース (S) で指定されたログワードデータと一致するデータを探し出すまでのステップ数を指定されたリザルト (R) に格納します。



- ・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 (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をサーチし、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。

## 第2章 演算ファンクション

### (3) データタイプ

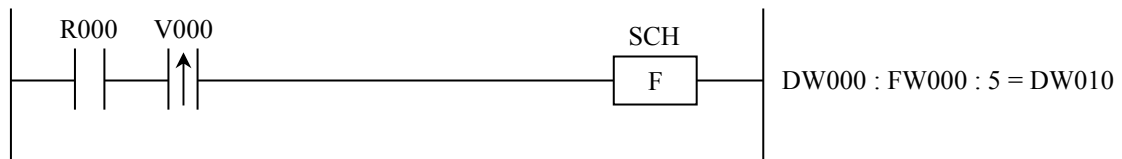
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	—	○	—	○	—	○
m	○	○	—	—	—	—	○
R	○	—	—	—	—	—	○

○：指定できます。

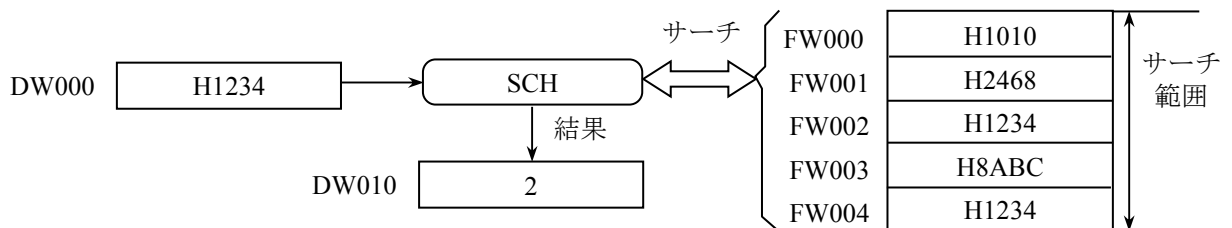
—：指定できません。

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。

## 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ーまたはバイナリー定数

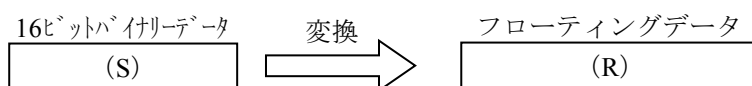
R：変換したフローティングデータを格納するレジスター

(\*）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

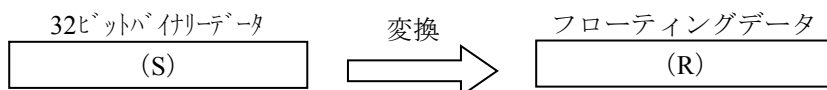
ソース (S) で指定された16ビットのバイナリーデータをフローティングデータに変換し、結果をリザルト (R) に格納します。



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	—	—	—	—	○	—	○

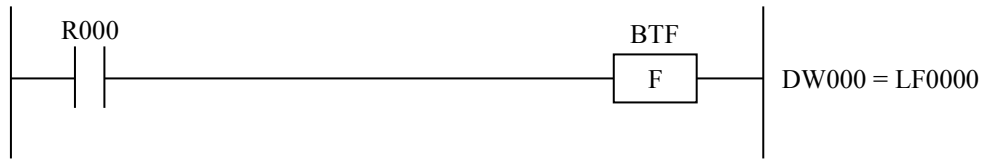
○：指定できます。

—：指定できません。

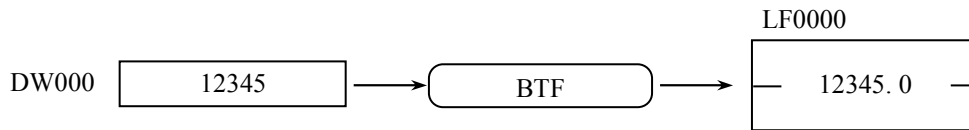
Rはフローティング型固定です。

## 第2章 演算ファンクション

### (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。

## FTB FLOAT→BINARY変換

## (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

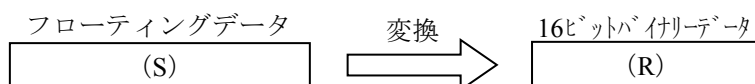
R：変換したバイナリデータを格納するレジスタ

(\*) 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

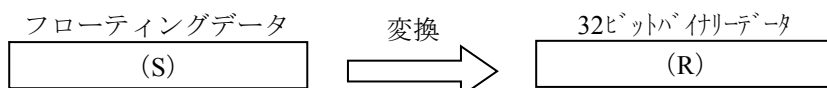
ソース (S) で指定されたフローティングデータを16ビットのバイナリデータに変換し、結果をリザルト (R) に格納します。



- ・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

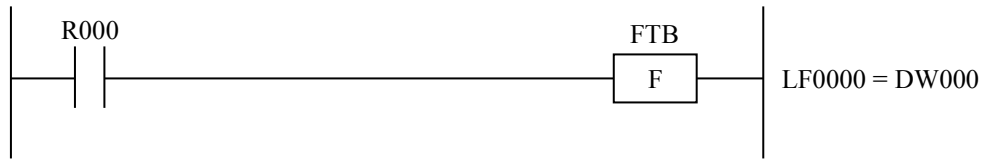
	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	—	—	—	—	○	○	○
R	○	—	○	—	—	—	○

○：指定できます。

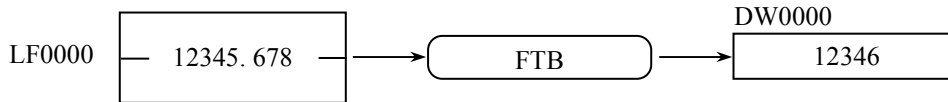
—：指定できません。

## 第2章 演算ファンクション

### (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、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
—	—	—	—	—	↕

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

- ・  $-2147483648 < \text{リザルト (R)} < 2147483647$  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ーまたはバイナリー定数

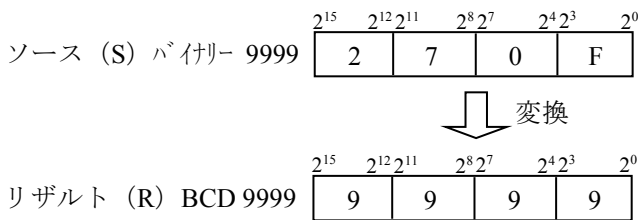
R : 変換したBCDデータを格納するレジスター

(\*) 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

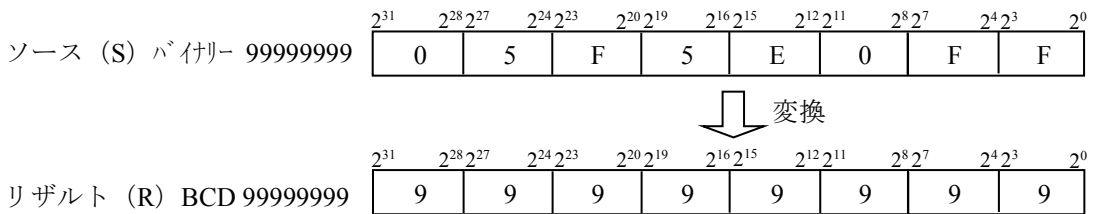
● ワードデータのBCD変換

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)。

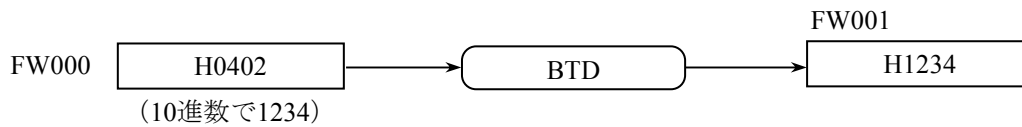


## 第2章 演算ファンクション

### (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

DTB BCD→BINARY変換

(1) 入力形式

DTB S -> R

S : BCDデータ格納レジスタまたはBCD定数

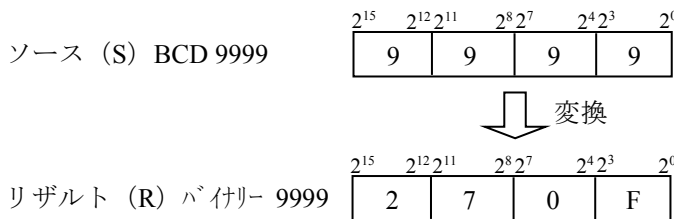
R : 変換したバイナリーデータを格納するレジスタ

(\*) 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

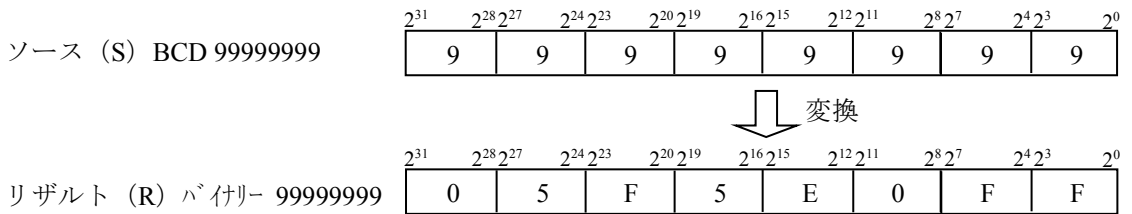
● ワードBCDデータのバイナリー変換

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

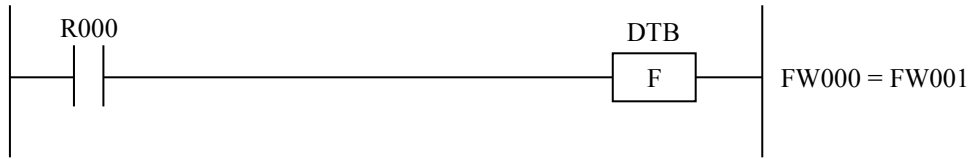
	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	—	—	○
R	○	—	○	—	—	—	○

○ : 指定できます。  
 — : 指定できません。

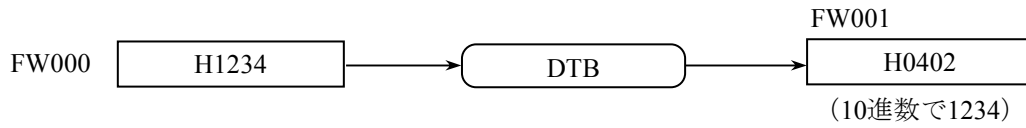
SとRの型 (ワード/ロングワード) は合わせてください。異なった型は指定できません (入力エラーになります)。

## 第2章 演算ファンクション

### (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

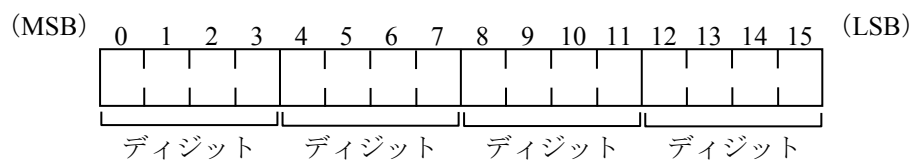
#### ● 演算結果フラグ

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

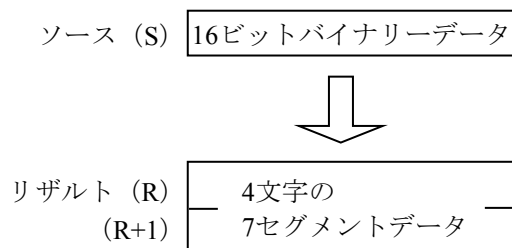
R: 変換した7セグメントデータを格納するレジスタ

(\*) 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

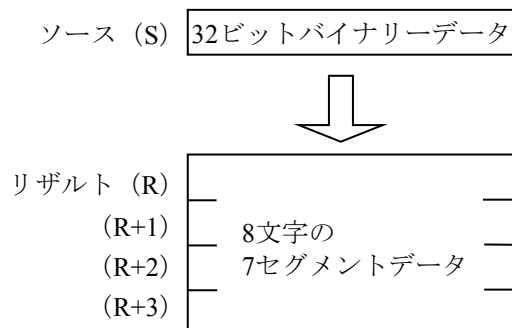
## ● ワードデータのセグメント変換

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。



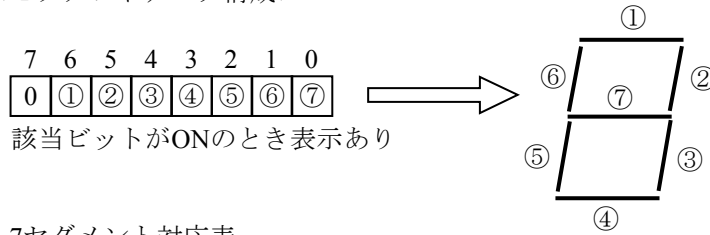
## ● ロングワードデータのセグメント変換

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。



## 第2章 演算ファンクション

<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																
データ	H7E	H30	H6D	H79	H33	H5B	H5F	H70	H7F	H7B	H77	H1F	H4E	H3D	H4F	H47

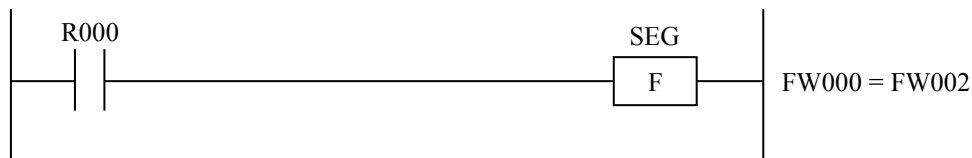
(3) データタイプ

	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	○	—	○	—	—	—	○

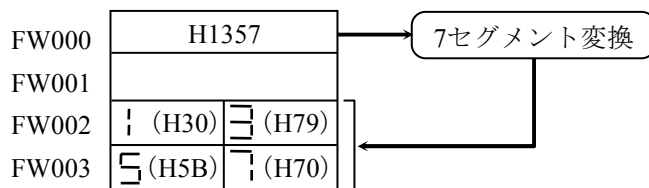
○：指定できます。  
—：指定できません。

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

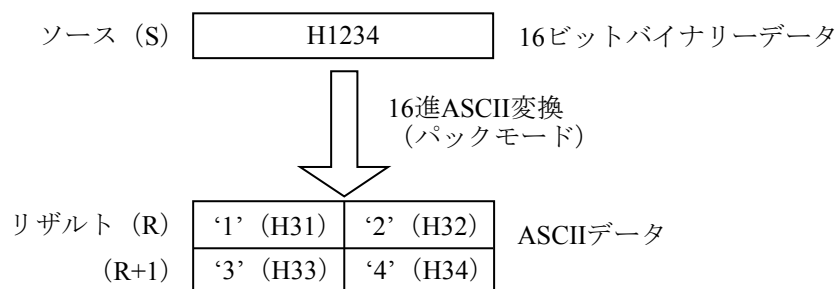
S : バイナリーデータ格納レジスタまたはバイナリー定数

R : 変換したASCIIデータを格納するレジスタ

(\* ) 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	定数	レジスタ	定数	レジスタ	定数	
S	○	○	—	—	—	—	○
R	○	—	—	—	—	—	○

○ : 指定できます。

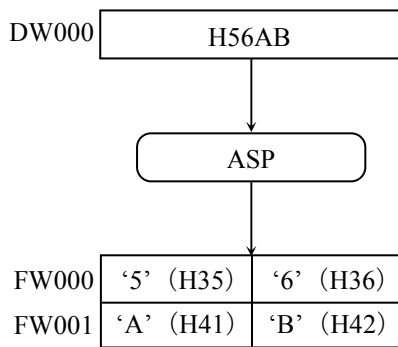
— : 指定できません。

## 第2章 演算ファンクション

### (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

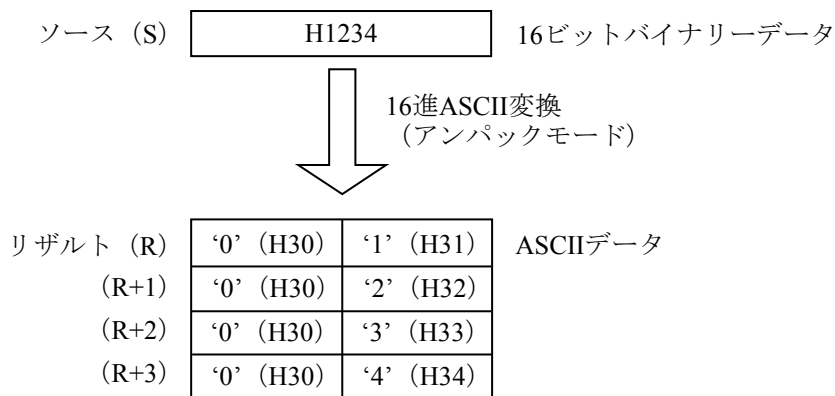
S : バイナリーデータ格納レジスタまたはバイナリー定数

R : 変換したASCIIデータを格納するレジスタ

(\*) 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

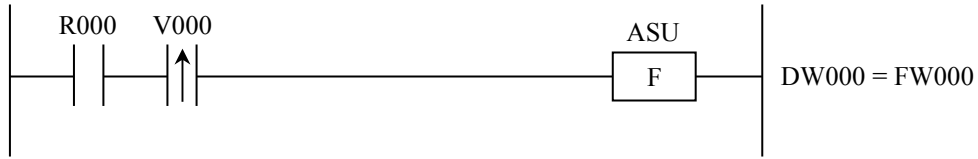
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	○
R	○	—	—	—	—	—	○

○ : 指定できます。  
— : 指定できません。

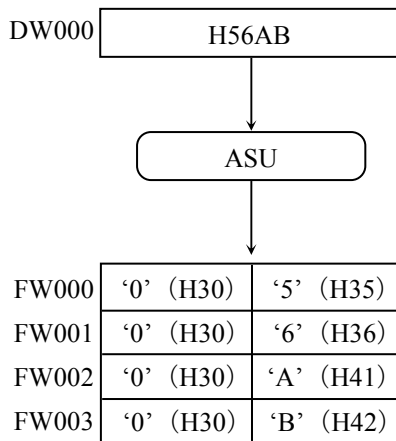


## 第2章 演算ファンクション

### (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

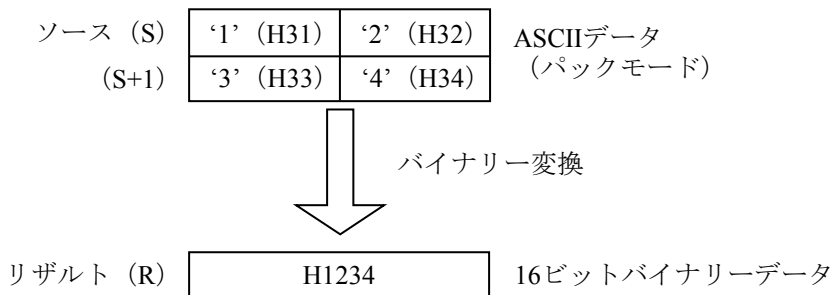
S : ASCIIデータ格納レジスター

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	○	—	—	—	—	—	○

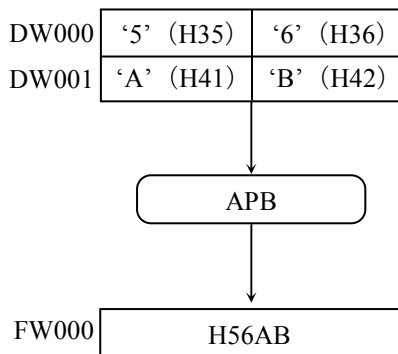
○ : 指定できます。  
— : 指定できません。

## 第2章 演算ファンクション

### (4) プログラム例



入力条件R000がOFF→ONに変化時、1回のみDW000の内容をバックモードの16進ASCIIデータからバイナリデータに変換して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

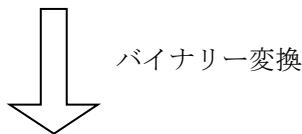
R : 変換したバイナリーデータを格納するレジスタ

(\*) 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

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)	‘0’ (H30)	‘1’ (H31)	ASCIIデータ (アンパックモード)
(S+1)	‘0’ (H30)	‘2’ (H32)	
(S+2)	‘0’ (H30)	‘3’ (H33)	
(S+3)	‘0’ (H30)	‘4’ (H34)	



リザルト (R) H1234 16ビットバイナリーデータ

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

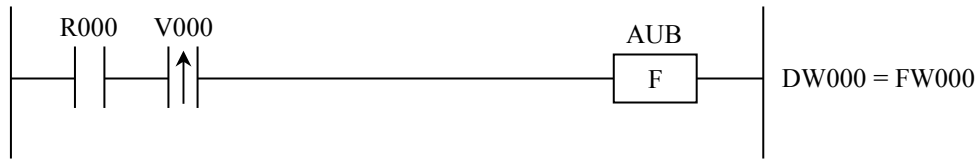
/	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	—	—	—	—	—	○
R	○	—	—	—	—	—	○

○ : 指定できます。

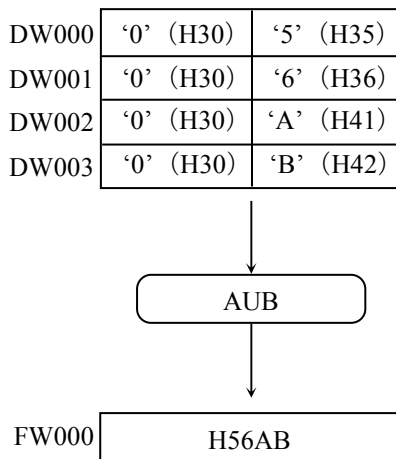
— : 指定できません。

## 第2章 演算ファンクション

### (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

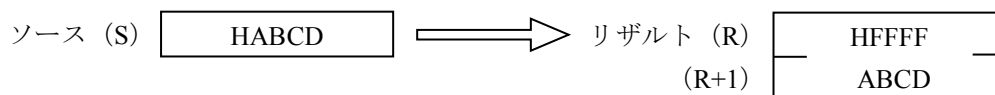
R: 変換した32ビットバイナリーデータを格納するレジスター

(\*) 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。

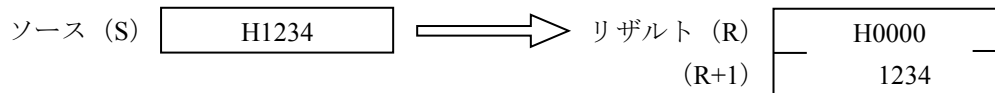
(2) 機能

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

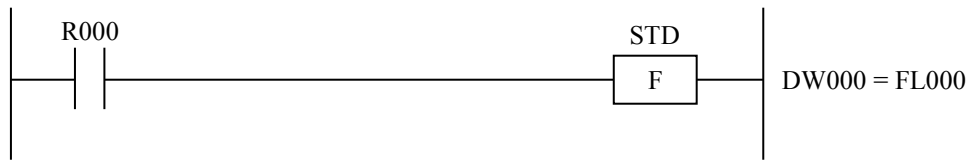
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	○
R	—	—	○	—	—	—	○

○: 指定できます。

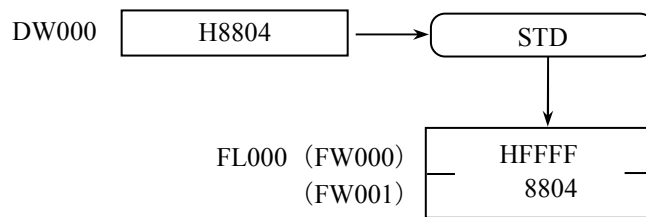
—: 指定できません。

## 第2章 演算ファンクション

### (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

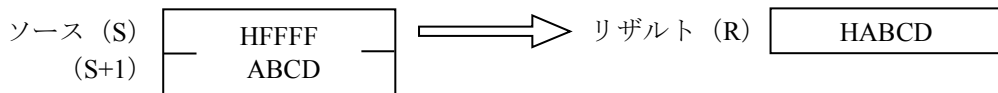
R : 変換した16ビットバイナリーデータを格納するレジスター

(\*) 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。

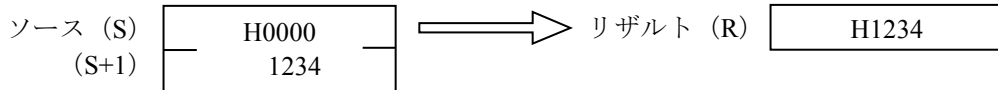
(2) 機能

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	○	—	—	—	—	—	○

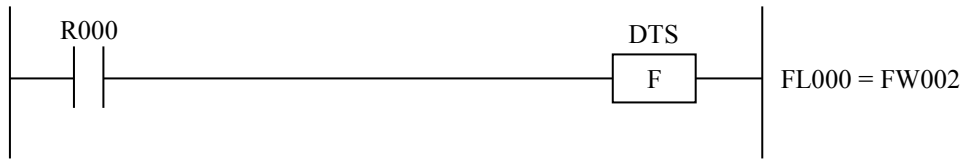
○ : 指定できます。

— : 指定できません。

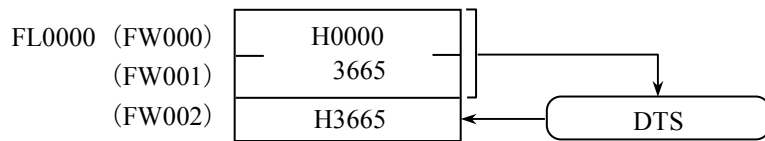


## 第2章 演算ファンクション

### (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 (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



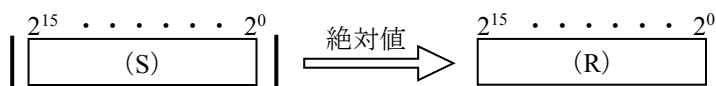
S : ソース格納レジスタまたは定数

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。

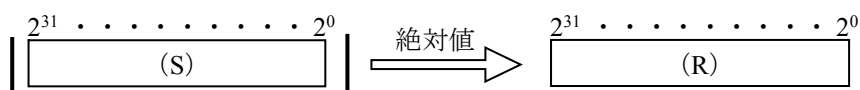
(2) 機能

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

(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。

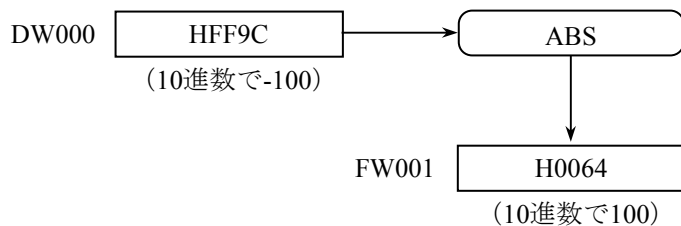
SとRの型 (ワード/ロングワード/フローティング) は合わせてください。異なった型は指定できません (入力エラーになります) 。

## 第2章 演算ファンクション

### (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

NEG 符号変換 : SIGN CHANGE

(1) 入力形式



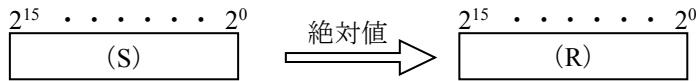
S : 符号変換するデータの格納レジスタまたは定数

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。

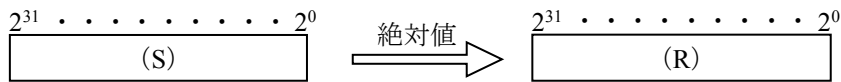
(2) 機能

- 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

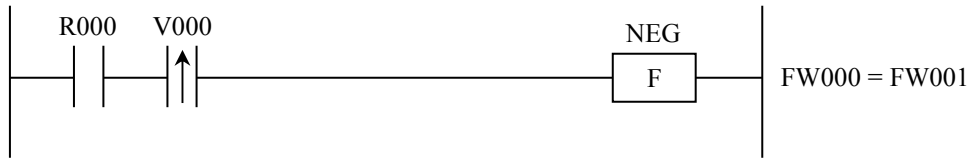
	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○ : 指定できます。  
 — : 指定できません。

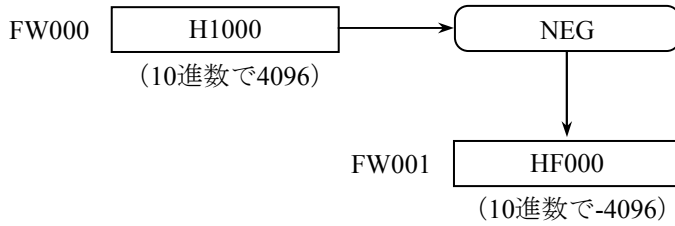
SとRの型 (ワード/ロングワード/フローティング) は合わせてください。異なった型は指定できません (入力エラーになります) 。

## 第2章 演算ファンクション

### (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

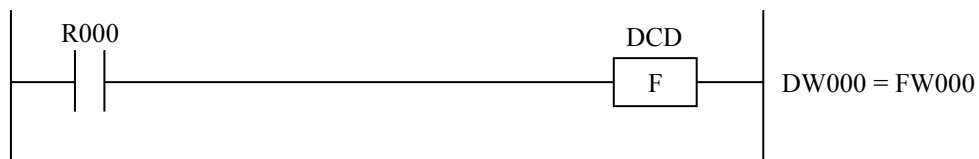
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	-	-	-	-	○
R	○	-	○	-	-	-	○

○ : 指定できます。

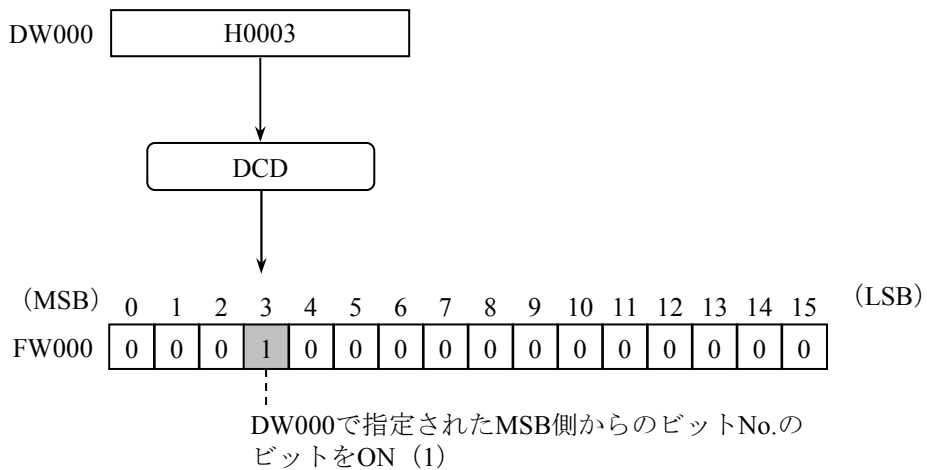
- : 指定できません。

## 第2章 演算ファンクション

### (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ーまたは定数

R : エン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

- ・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	○	-	-	-	-	-	○

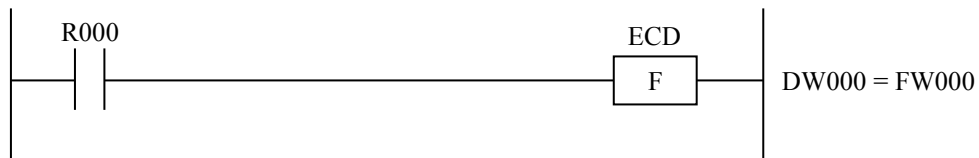
○ : 指定できます。

- : 指定できません。

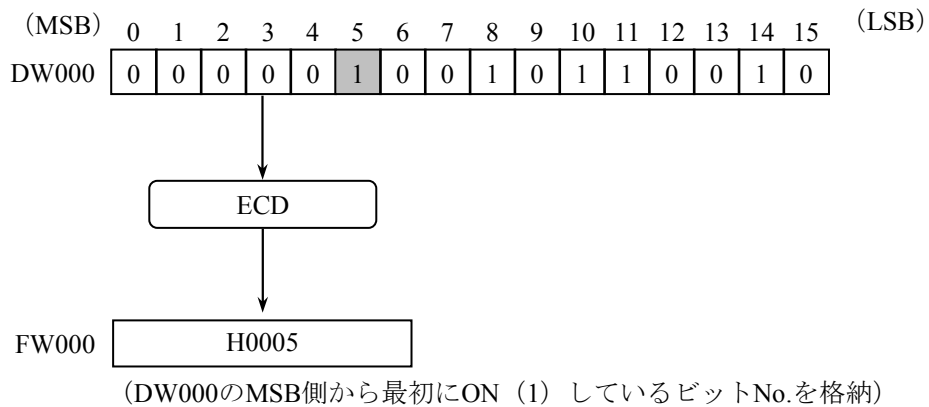


## 第2章 演算ファンクション

### (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、それ以外のとき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

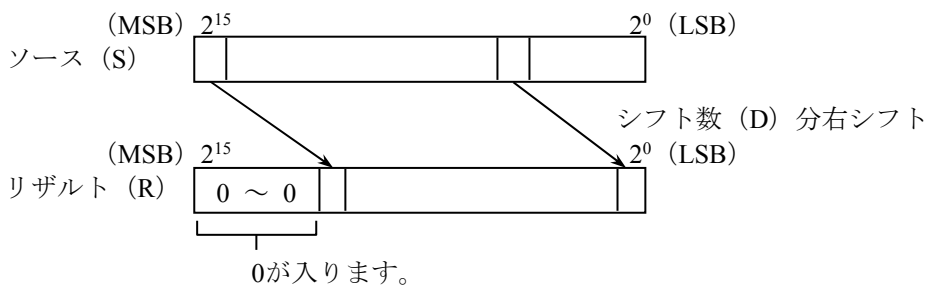
S : シフトするデータを格納したレジスタまたは定数  
 D : シフトするビット数を格納したレジスタまたは定数  
 R : シフトした結果を格納するレジスタ

(\*) 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

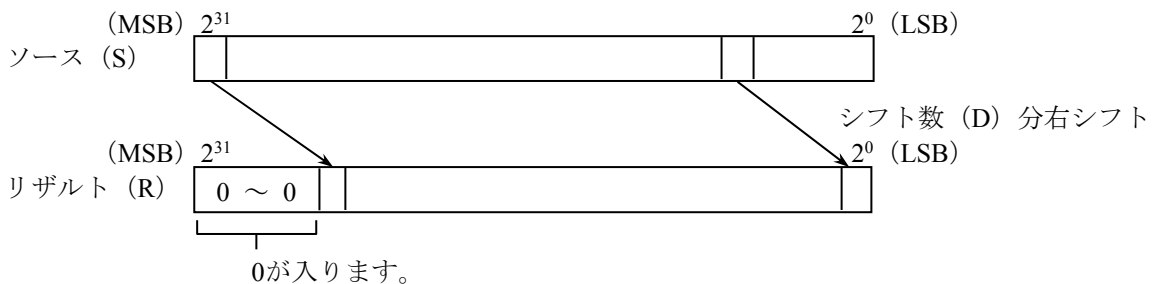
ソース (S) で指定された16ビットデータを、デスティネーション (D) で指定されたシフト数分右にシフトし、結果をリザルト (R) に格納します。



- ・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。

## 第2章 演算ファンクション

### (3) データタイプ

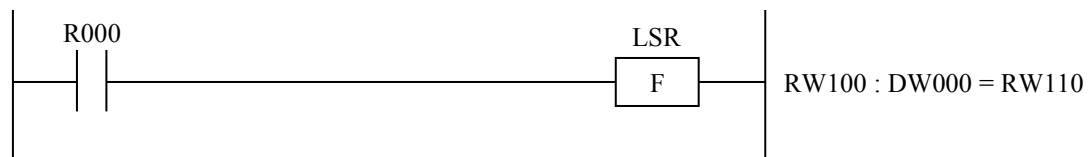
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
D	○	○	—	—	—	—	○
R	○	—	○	—	—	—	○

○：指定できます。

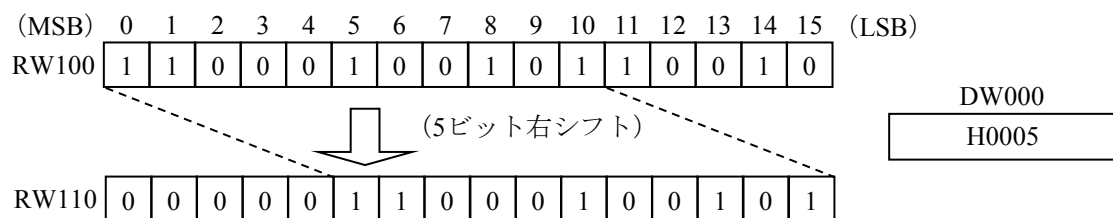
—：指定できません。

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

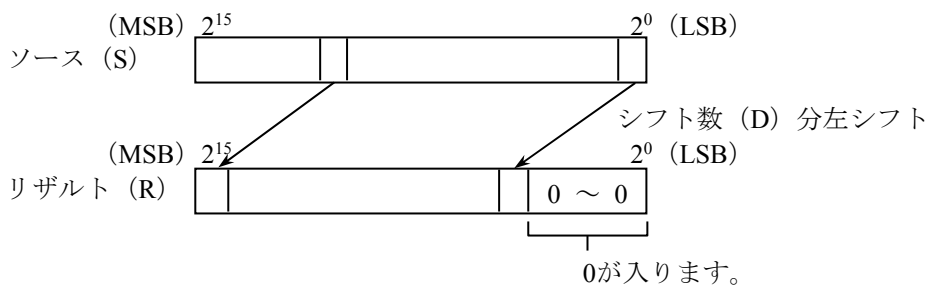
S : シフトするデータを格納したレジスタまたは定数  
 D : シフトするビット数を格納したレジスタまたは定数  
 R : シフトした結果を格納するレジスタ

(\*) 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

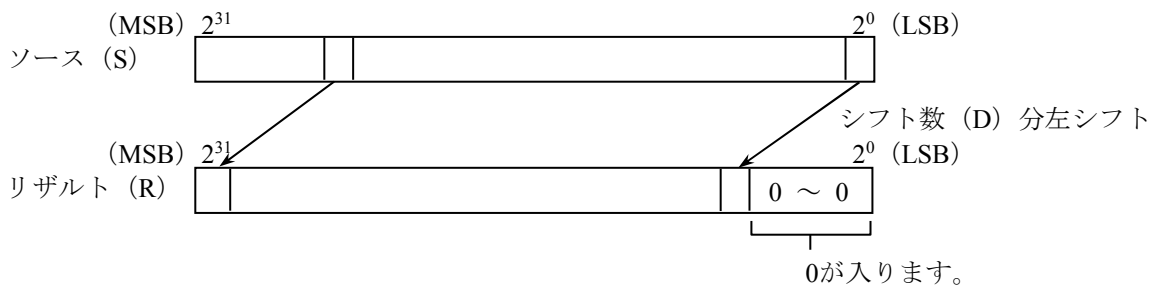
ソース (S) で指定された16ビットデータを、デスティネーション (D) で指定されたシフト数分左にシフトし、結果をリザルト (R) に格納します。



- ・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。

## 第2章 演算ファンクション

### (3) データタイプ

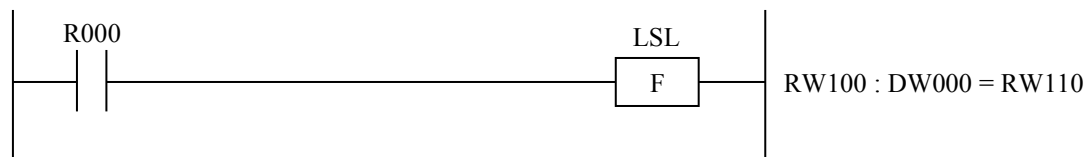
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
D	○	○	—	—	—	—	○
R	○	—	○	—	—	—	○

○：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

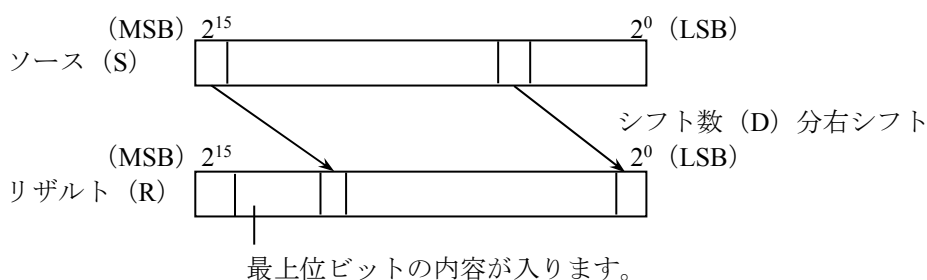
S : シフトするデータを格納したレジスタまたは定数  
 D : シフトするビット数を格納したレジスタまたは定数  
 R : シフトした結果を格納するレジスタ

(\*) 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

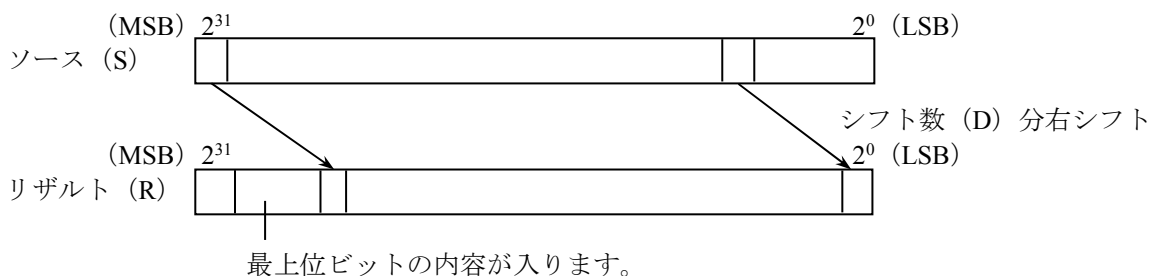
ソース (S) で指定された16ビットデータを、デスティネーション (D) で指定されたシフト数分右にシフト (符号ビット保持) し、結果をリザルト (R) に格納します。



- ・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。

## 第2章 演算ファンクション

### (3) データタイプ

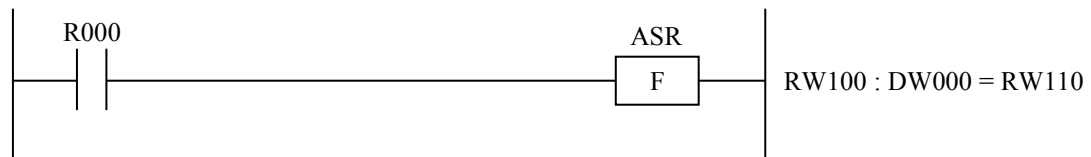
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
D	○	○	—	—	—	—	○
R	○	—	○	—	—	—	○

○：指定できます。

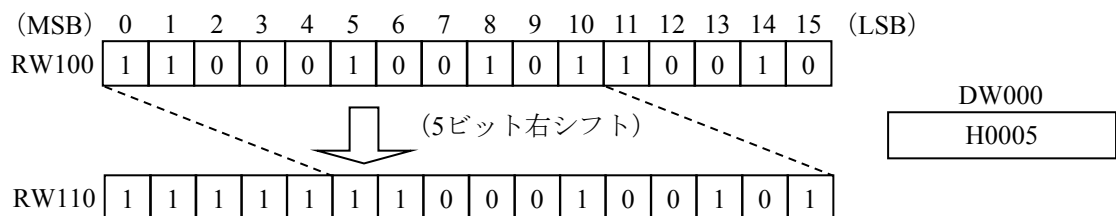
—：指定できません。

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

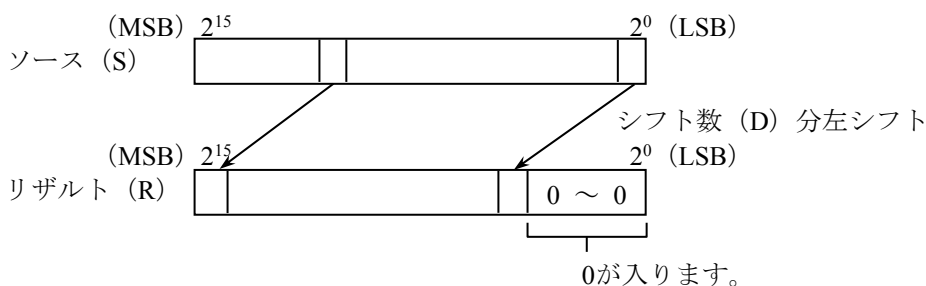
R : シフトした結果を格納するレジスタ

(\*) 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

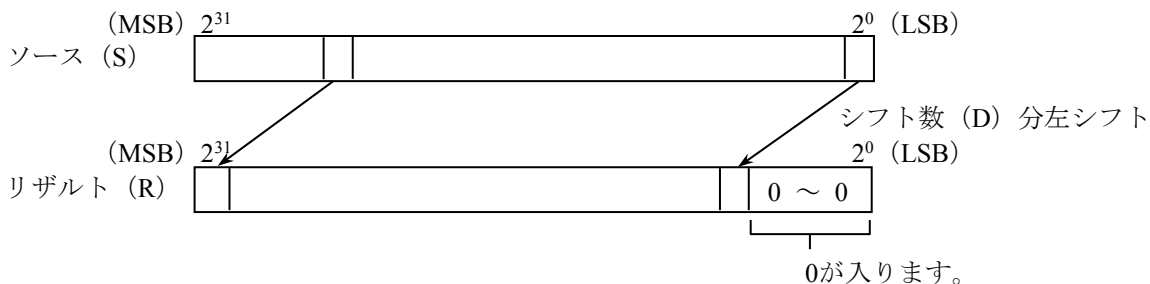
ソース (S) で指定された16ビットデータを、デスティネーション (D) で指定されたシフト数分左にシフトし、結果をリザルト (R) に格納します。



- ・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。



## 第2章 演算ファンクション

### (3) データタイプ

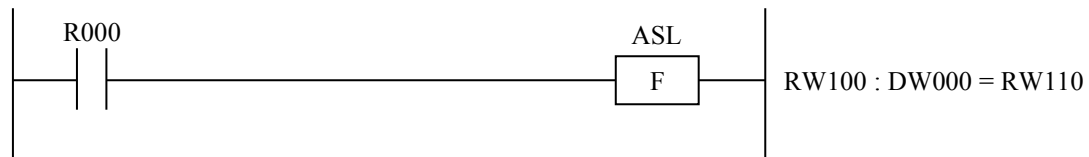
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
D	○	○	—	—	—	—	○
R	○	—	○	—	—	—	○

○：指定できます。

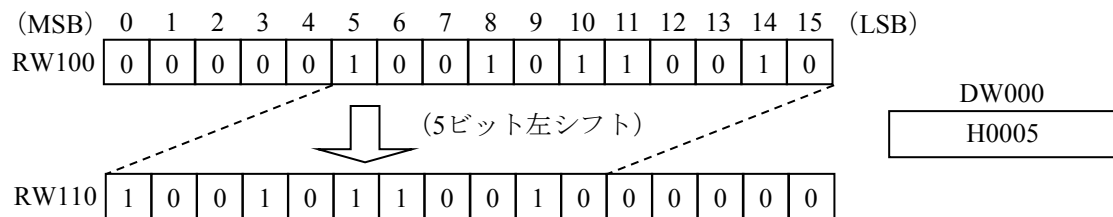
—：指定できません。

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

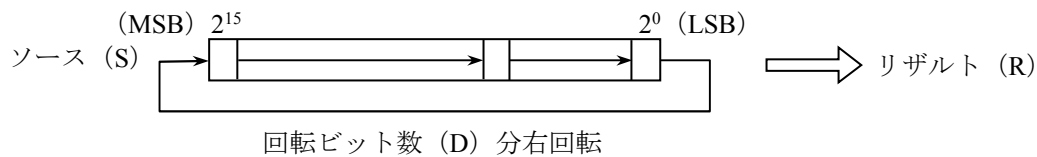
- S : 回転するデータを格納したレジスタまたは定数  
 D : 回転するビット数を格納したレジスタまたは定数  
 R : 回転した結果を格納するレジスタ

(\*) 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

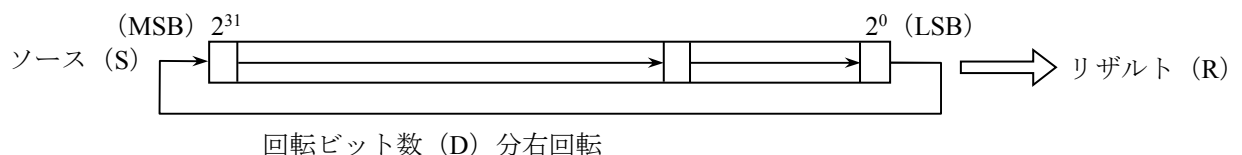
ソース (S) で指定された16ビットデータを、デスティネーション (D) で指定された回転ビット数分右に回転し、結果をリザルト (R) に格納します。



- ・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。

## 第2章 演算ファンクション

### (3) データタイプ

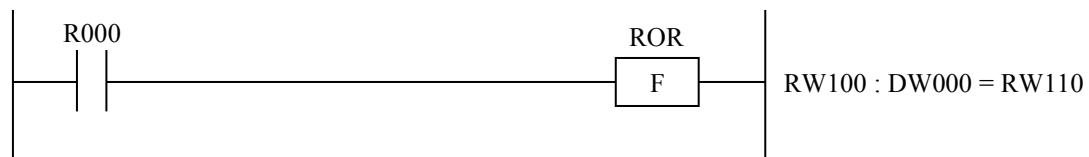
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
D	○	○	—	—	—	—	○
R	○	—	○	—	—	—	○

○：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

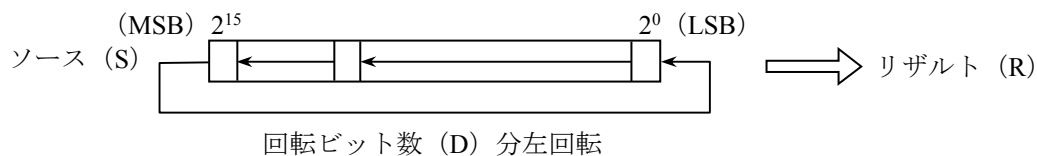
R : 回転した結果を格納するレジスタ

(\*) 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

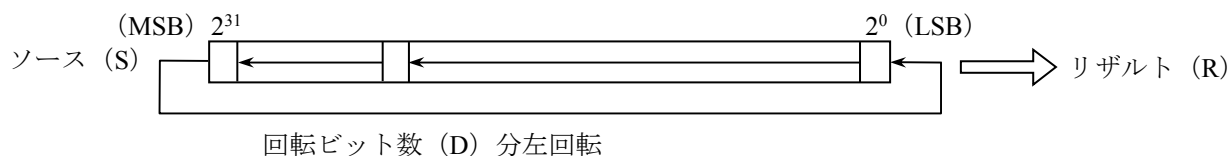
ソース (S) で指定された16ビットデータを、デスティネーション (D) で指定された回転ビット数分左に回転し、結果をリザルト (R) に格納します。



- ・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。

## 第2章 演算ファンクション

### (3) データタイプ

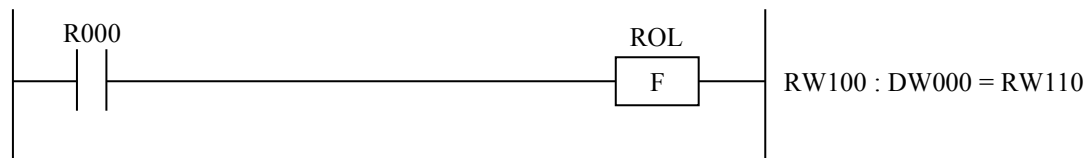
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	—	—	○
D	○	○	—	—	—	—	○
R	○	—	○	—	—	—	○

○：指定できます。

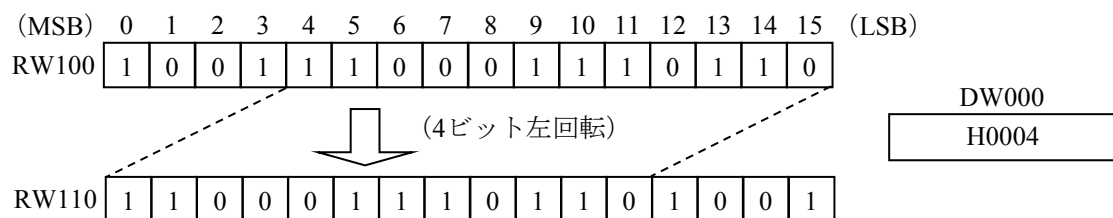
—：指定できません。

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

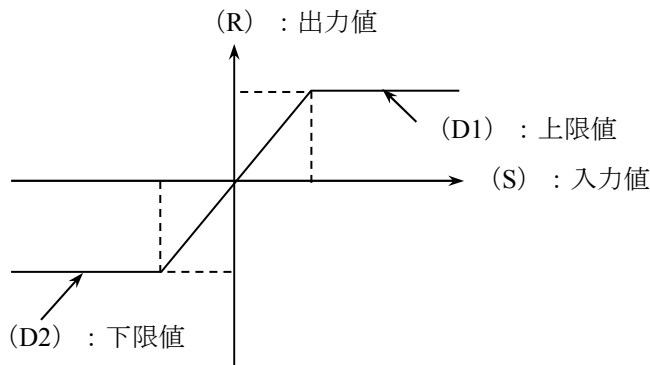
LIM S : D1 : D2 -> R

- S : 入力値を格納したレジスターまたは定数
- D1 : 上限値を格納したレジスターまたは定数
- D2 : 下限値を格納したレジスターまたは定数
- 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

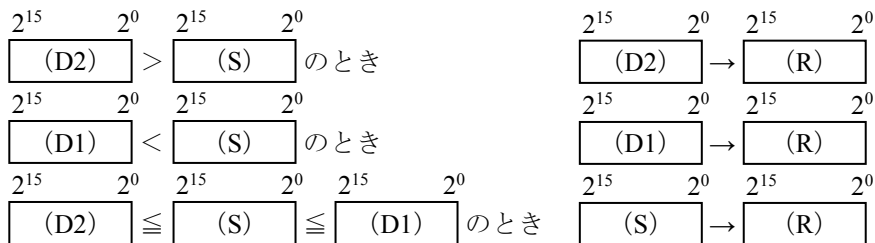
(\* ) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

(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。

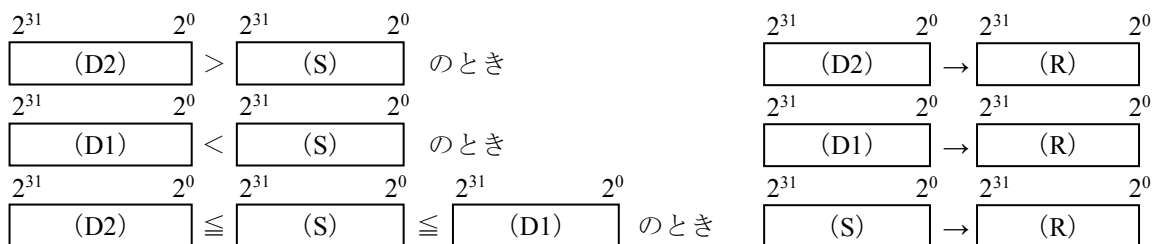


- ・ ソース (S) , デスティネーション (D1) , (D2) に指定できる値の範囲は、-32768~32767です。

## 第2章 演算ファンクション

### ● ロングワードデータのリミッター

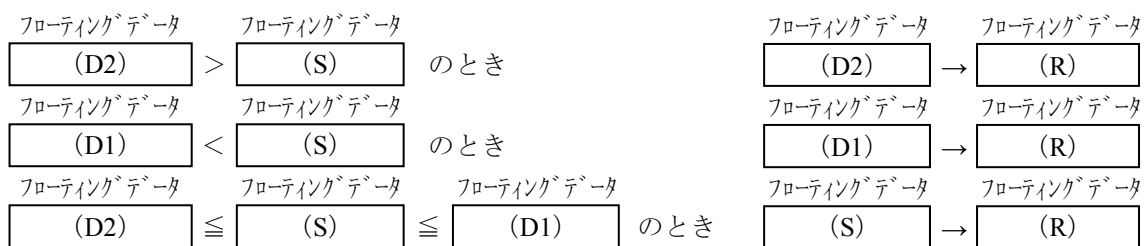
- ・下記の32ビットデータをリミッター制御します。



- ・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

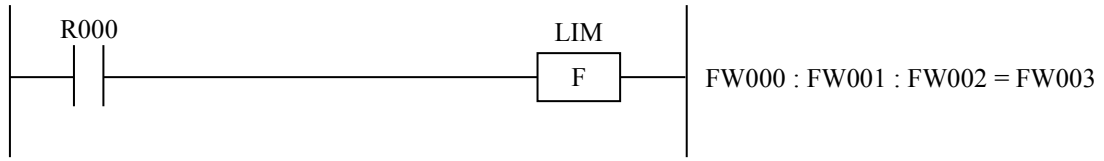
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D1	○	○	○	○	○	○	○
D2	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○ : 指定できます。

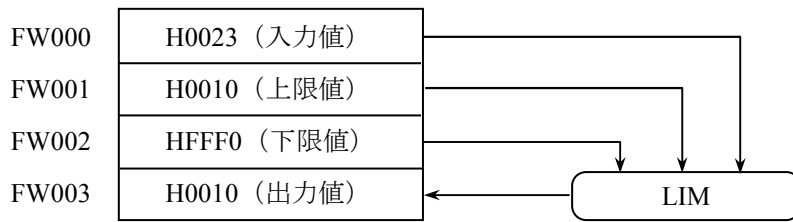
— : 指定できません。

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)。

(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、それ以外るとき0  
E以外のフラグはすべて保持します。

- EフラグON時は、(D2) と比較しません。(D1) とのみ比較されます。



BND デッドバンド : DEAD BAND

(1) 入力形式

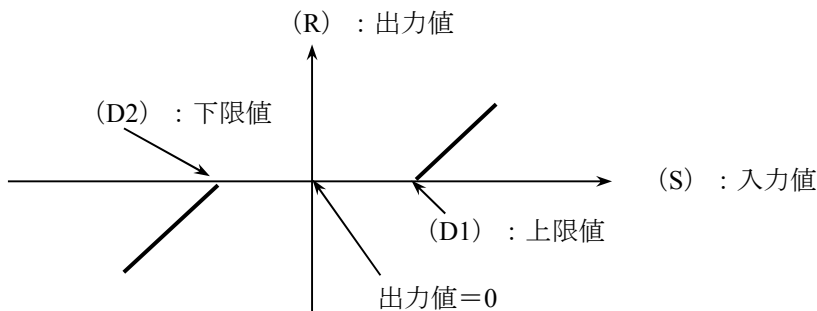
BND S : D1 : D2 -> R

- 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)。



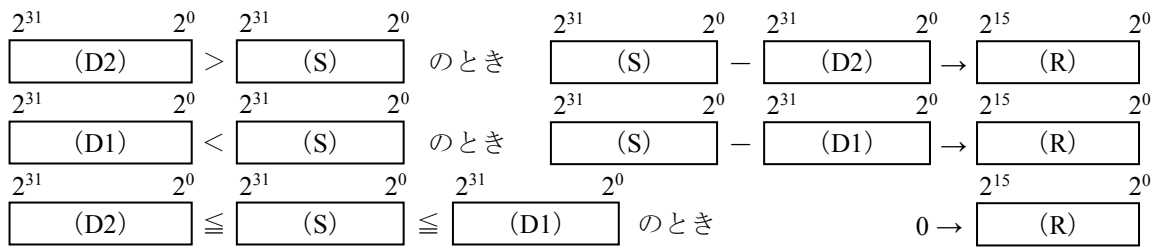
- ワードデータの不感帯制御
  - ・下記の16ビットデータを不感帯制御します。

$$\begin{array}{l}
 \begin{array}{ccc}
 \boxed{\begin{smallmatrix} 2^{15} & 2^0 \\ (D2) \end{smallmatrix}} > \boxed{\begin{smallmatrix} 2^{15} & 2^0 \\ (S) \end{smallmatrix}} \text{ のとき} & & \boxed{\begin{smallmatrix} 2^{15} & 2^0 \\ (S) \end{smallmatrix}} - \boxed{\begin{smallmatrix} 2^{15} & 2^0 \\ (D2) \end{smallmatrix}} \rightarrow \boxed{\begin{smallmatrix} 2^{15} & 2^0 \\ (R) \end{smallmatrix}} \\
 \boxed{\begin{smallmatrix} 2^{15} & 2^0 \\ (D1) \end{smallmatrix}} < \boxed{\begin{smallmatrix} 2^{15} & 2^0 \\ (S) \end{smallmatrix}} \text{ のとき} & & \boxed{\begin{smallmatrix} 2^{15} & 2^0 \\ (S) \end{smallmatrix}} - \boxed{\begin{smallmatrix} 2^{15} & 2^0 \\ (D1) \end{smallmatrix}} \rightarrow \boxed{\begin{smallmatrix} 2^{15} & 2^0 \\ (R) \end{smallmatrix}} \\
 \boxed{\begin{smallmatrix} 2^{15} & 2^0 \\ (D2) \end{smallmatrix}} \leq \boxed{\begin{smallmatrix} 2^{15} & 2^0 \\ (S) \end{smallmatrix}} \leq \boxed{\begin{smallmatrix} 2^{15} & 2^0 \\ (D1) \end{smallmatrix}} \text{ のとき} & & 0 \rightarrow \boxed{\begin{smallmatrix} 2^{15} & 2^0 \\ (R) \end{smallmatrix}}
 \end{array}$$

- ・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、-32768~32767です。

● ロングワードデータの不感帯制御

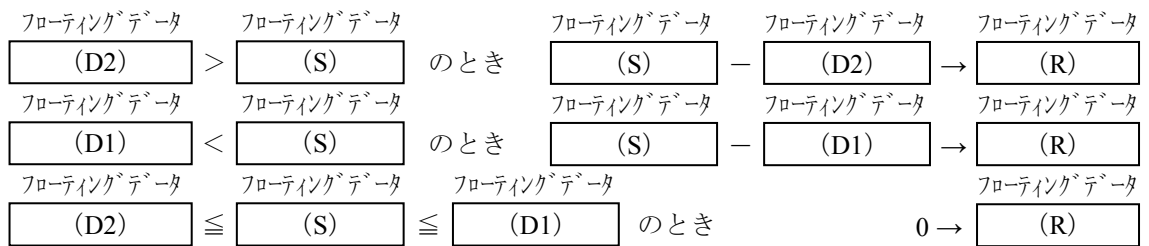
・下記の32ビットデータを不感帯制御します。



・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	○	—	○	—	○	—	○

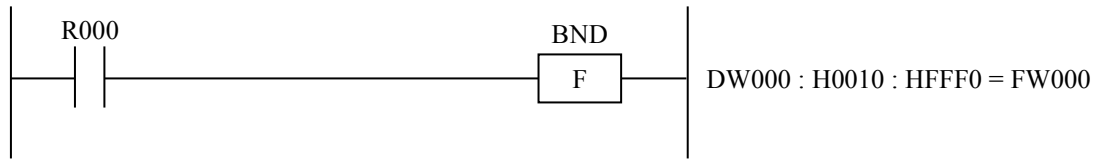
○ : 指定できます。

— : 指定できません。

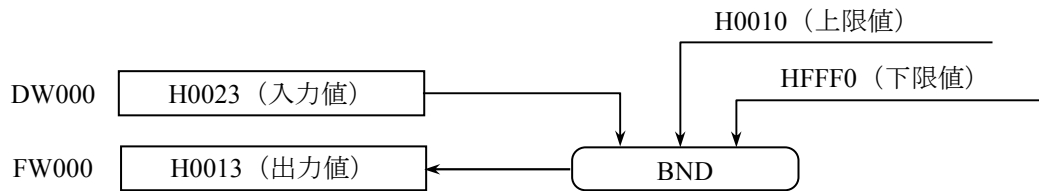
S, D1, D2, Rの型 (ワード/ロングワード/フローティング) はすべて合わせてください。異なった型は指定できません (入力エラーになります) 。

## 第2章 演算ファンクション

### (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  
(インダイレクト指定のときのみ)

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

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

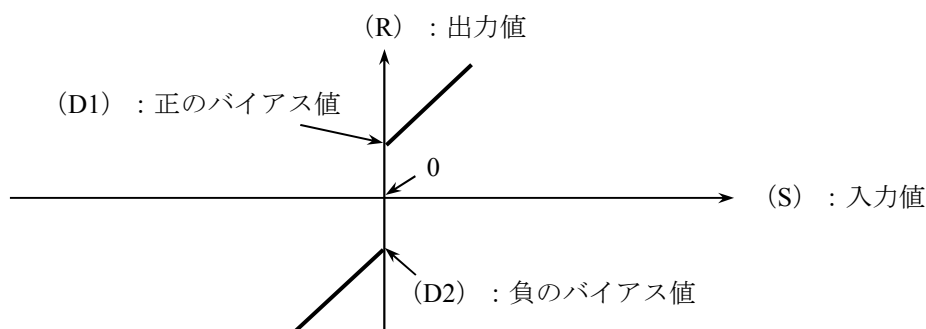
D2 : 入力値に加算する負のバイアス値を格納したレジスタまたは定数

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。

$$\begin{array}{l}
 \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \\
 \boxed{\text{(S)}} > 0 \text{ のとき } \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} + \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \text{(D1)} \rightarrow \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \text{(R)} \\
 \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \\
 \boxed{\text{(S)}} = 0 \text{ のとき } \quad \quad \quad 0 \rightarrow \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \text{(R)} \\
 \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \\
 \boxed{\text{(S)}} < 0 \text{ のとき } \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} + \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \text{(D2)} \rightarrow \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \text{(R)}
 \end{array}$$

- ・ 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、-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。

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

- ・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。

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

- ・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, ±2<sup>-126</sup>～±2<sup>128</sup>

(3) データタイプ

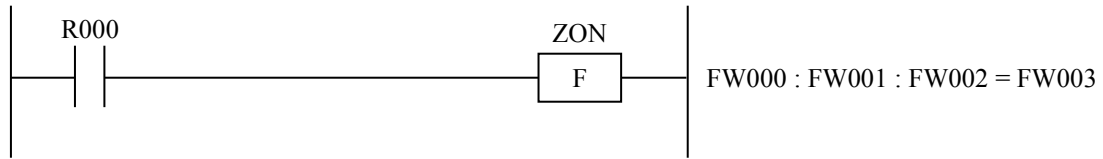
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D1	○	○	○	○	○	○	○
D2	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

- : 指定できます。
- : 指定できません。

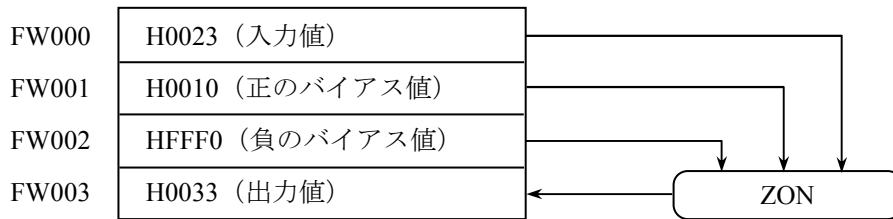
S, D1, D2, Rの型 (ワード/ロングワード/フローティング) はすべて合わせてください。異なった型は指定できません (入力エラーになります)。

## 第2章 演算ファンクション

### (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  
(インダイレクト指定のときのみ)

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

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



S : 平方根演算するデータの格納レジスタまたは定数

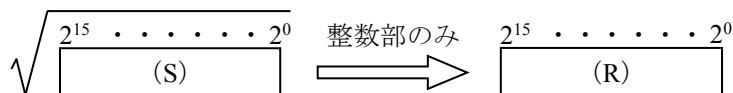
R : 演算結果を格納するレジスタ

(\*) 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

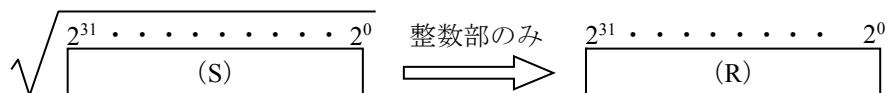
- ・ソース (S) で指定された16ビットデータの平方根を演算し、整数部のみリザルト (R) に格納します。



- ・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

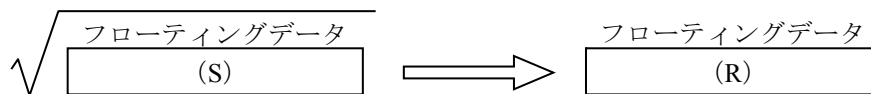
- ・ソース (S) で指定された32ビットデータの平方根を演算し、整数部のみリザルト (R) に格納します。



- ・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

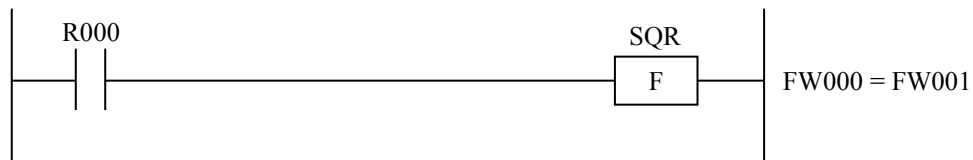
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○：指定できます。  
—：指定できません。

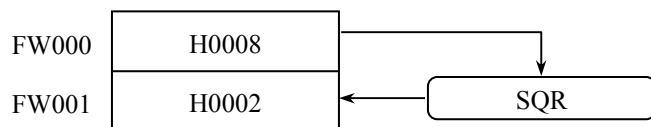
SとRの型（ワード/ロングワード/フローティング）は合わせてください。異なった型は指定できません（入力エラーになります）。

(4) プログラム例

● ワードデータの平方根



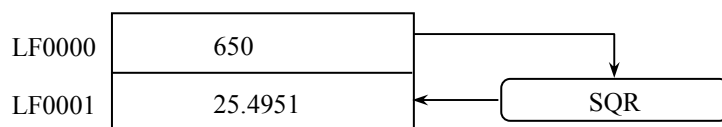
入力条件R000がONのとき、FW000の内容で平方根演算し、演算結果（整数部のみ）をFW001に格納します。



● 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。



## 第2章 演算ファンクション

### (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。

## SIN 正弦 : SIN

## (1) 入力形式

SIN S -> R
------------

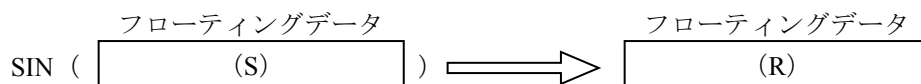
S : SIN (正弦) 演算する角度データの格納レジスターまたは定数

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 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

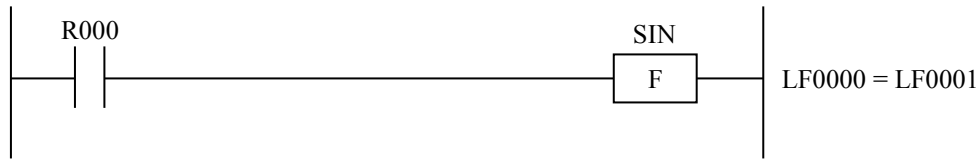
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	—	—	—	—	○	○	○
R	—	—	—	—	○	—	○

○ : 指定できます。

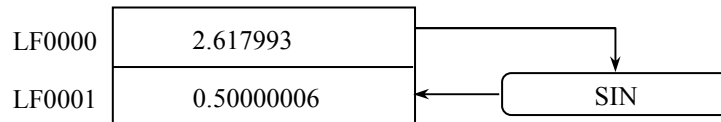
— : 指定できません。

## 第2章 演算ファンクション

### (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。

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

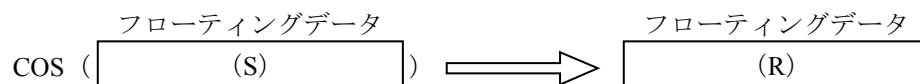
R : 演算結果を格納するレジスター

(\* ) 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

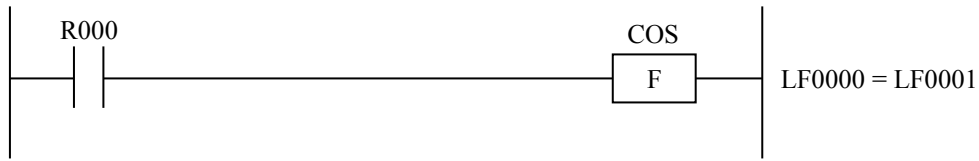
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	—	—	—	—	○	○	○
R	—	—	—	—	○	—	○

○ : 指定できます。

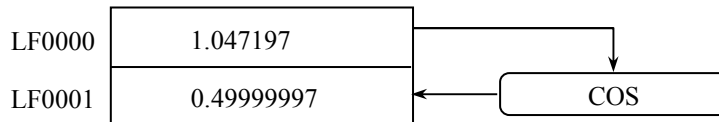
— : 指定できません。

## 第2章 演算ファンクション

### (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

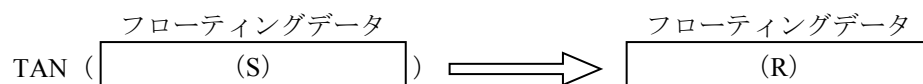
R : 演算結果を格納するレジスター

(\*) 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  $\times \pi / 180$ ) 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	—	—	—	—	○	—	○

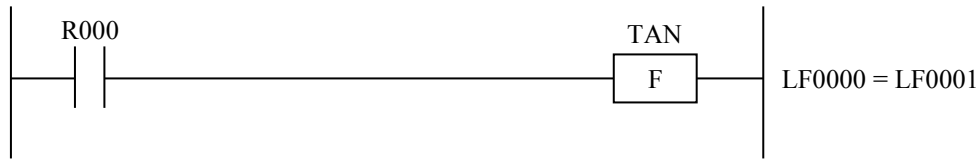
○ : 指定できます。

— : 指定できません。

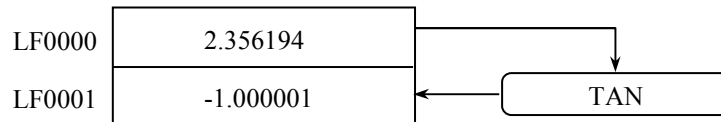


## 第2章 演算ファンクション

### (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、かつ下記範囲外するとき1、それ以外するとき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。

ASIN 逆正弦 : SIN<sup>-1</sup>

(1) 入力形式

ASIN S -> R

S : SIN<sup>-1</sup> (逆正弦) 演算する角度データの格納レジスターまたは定数

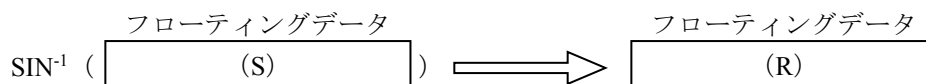
R : 演算結果を格納するレジスター

(\* ) 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

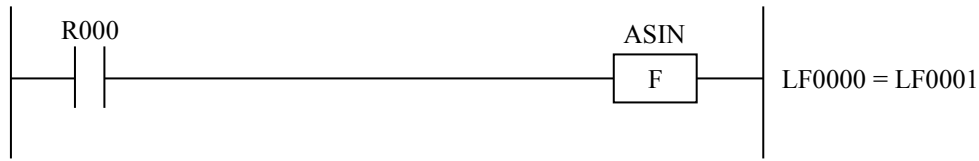
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	—	—	—	—	○	○	○
R	—	—	—	—	○	—	○

○ : 指定できます。

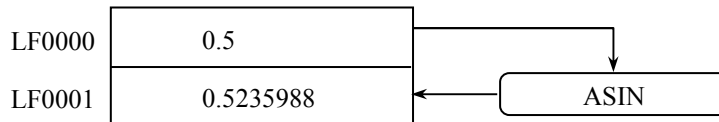
— : 指定できません。

## 第2章 演算ファンクション

### (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、 $-1.0 \sim 1.0$ 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。

ACOS 逆余弦 :  $\text{COS}^{-1}$ 

## (1) 入力形式

ACOS S -> R
-------------

S :  $\text{COS}^{-1}$  (逆余弦) 演算する角度データの格納レジスタまたは定数

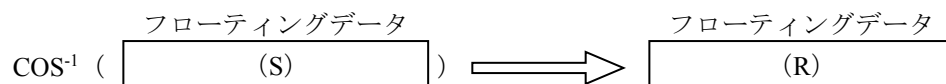
R : 演算結果を格納するレジスタ

(\*) 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

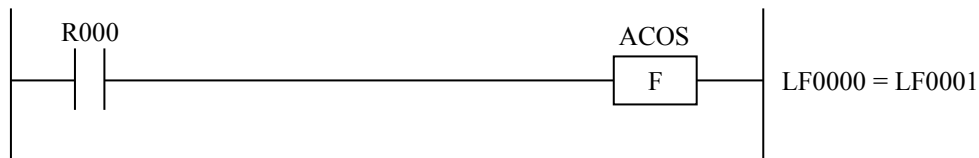
	ワード		ロングワード		フローティング		インデックス 指定
	レジスタ	定数	レジスタ	定数	レジスタ	定数	
S	—	—	—	—	○	○	○
R	—	—	—	—	○	—	○

○ : 指定できます。

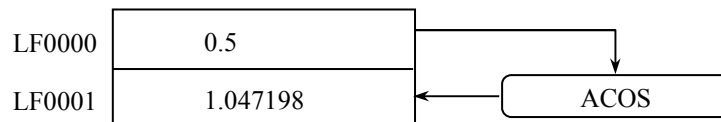
— : 指定できません。

## 第2章 演算ファンクション

### (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、-1.0~1.0の範囲外の時1、範囲内のとき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。

ATAN 逆正接 : TAN<sup>-1</sup>

(1) 入力形式

ATAN S -> R
-------------

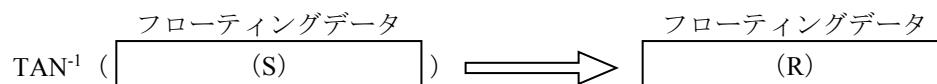
S : TAN<sup>-1</sup> (逆正弦) 演算する角度データの格納レジスタまたは定数

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。

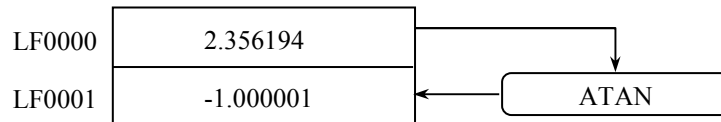
— : 指定できません。

## 第2章 演算ファンクション

### (4) プログラム例



入力条件R000がONのとき、LF0000の内容で $TAN^{-1}$ 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。

EXP 指数演算 : EXPONENTIAL

(1) 入力形式

EXP S -> R

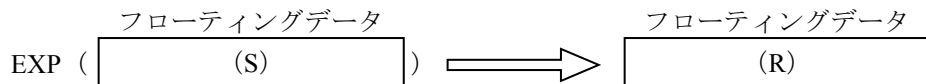
S : 指数演算するデータの格納レジスタまたは定数

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	—	—	—	—	○	—	○

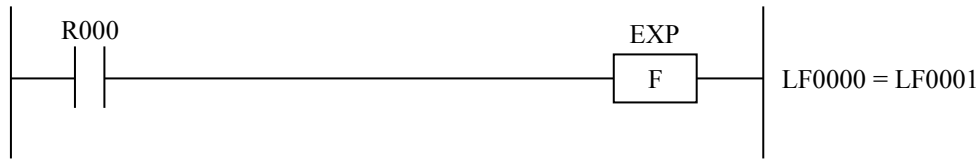
○ : 指定できます。

— : 指定できません。

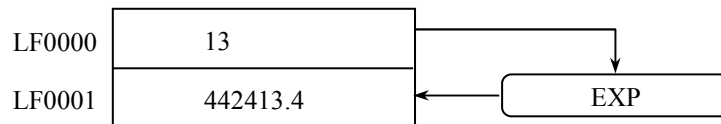


## 第2章 演算ファンクション

### (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

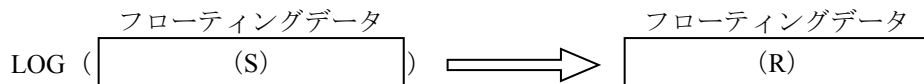
R : 演算結果を格納するレジスター

(\* ) 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。

## 第2章 演算ファンクション

### (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

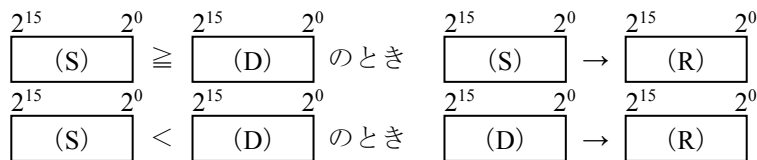
(\*) ファンクション名称とパラメーターおよびパラメーターとパラメーターの間には、必ずスペースを入力してください。「:」および「->」は省略できます。

(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

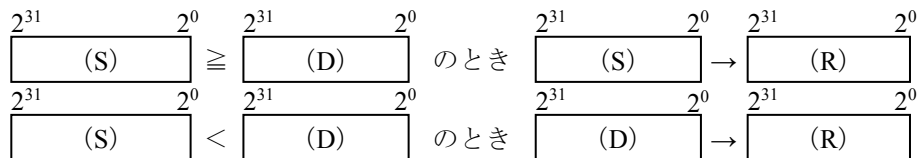
・下記の16ビットデータを大小比較し、大きい方の値をリザルト (R) に格納します。



・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

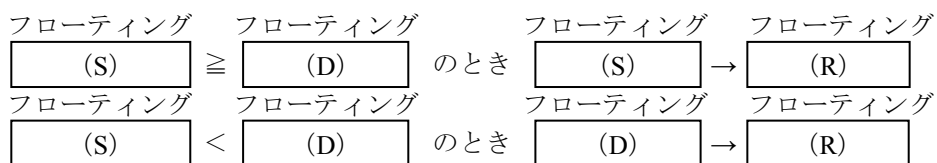
・下記の32ビットデータを大小比較し、大きい方の値をリザルト (R) に格納します。



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

## 第2章 演算ファンクション

### (3) データタイプ

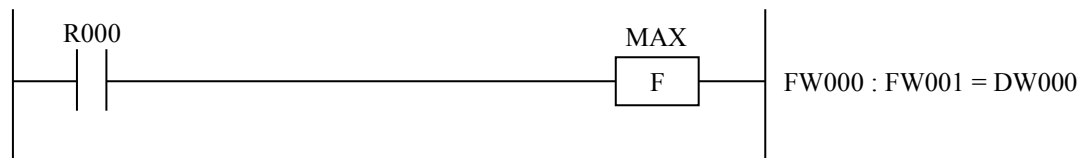
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○：指定できます。

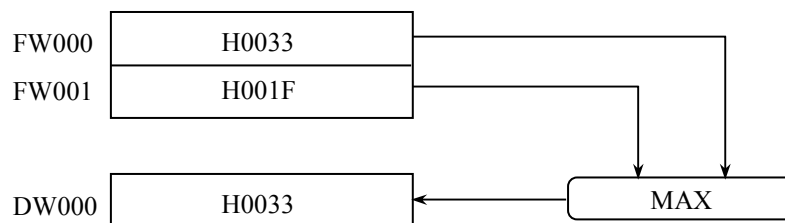
—：指定できません。

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。

$$\begin{array}{l}
 \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (S) \\ \hline \end{array} \leq \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (D) \\ \hline \end{array} \text{ のとき } \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (S) \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (R) \\ \hline \end{array} \\
 \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (S) \\ \hline \end{array} > \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (D) \\ \hline \end{array} \text{ のとき } \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (D) \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline 2^{15} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (R) \\ \hline \end{array}
 \end{array}$$

- 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。

$$\begin{array}{l}
 \begin{array}{|c|} \hline 2^{31} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (S) \\ \hline \end{array} \leq \begin{array}{|c|} \hline 2^{31} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (D) \\ \hline \end{array} \text{ のとき } \begin{array}{|c|} \hline 2^{31} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (S) \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline 2^{31} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (R) \\ \hline \end{array} \\
 \begin{array}{|c|} \hline 2^{31} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (S) \\ \hline \end{array} > \begin{array}{|c|} \hline 2^{31} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (D) \\ \hline \end{array} \text{ のとき } \begin{array}{|c|} \hline 2^{31} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (D) \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline 2^{31} \\ \hline \end{array} \begin{array}{|c|} \hline 2^0 \\ \hline \end{array} \begin{array}{|c|} \hline (R) \\ \hline \end{array}
 \end{array}$$

- 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。

$$\begin{array}{l}
 \begin{array}{|c|} \hline \text{フローティング} \\ \hline \end{array} \begin{array}{|c|} \hline (S) \\ \hline \end{array} \leq \begin{array}{|c|} \hline \text{フローティング} \\ \hline \end{array} \begin{array}{|c|} \hline (D) \\ \hline \end{array} \text{ のとき } \begin{array}{|c|} \hline \text{フローティング} \\ \hline \end{array} \begin{array}{|c|} \hline (S) \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline \text{フローティング} \\ \hline \end{array} \begin{array}{|c|} \hline (R) \\ \hline \end{array} \\
 \begin{array}{|c|} \hline \text{フローティング} \\ \hline \end{array} \begin{array}{|c|} \hline (S) \\ \hline \end{array} > \begin{array}{|c|} \hline \text{フローティング} \\ \hline \end{array} \begin{array}{|c|} \hline (D) \\ \hline \end{array} \text{ のとき } \begin{array}{|c|} \hline \text{フローティング} \\ \hline \end{array} \begin{array}{|c|} \hline (D) \\ \hline \end{array} \rightarrow \begin{array}{|c|} \hline \text{フローティング} \\ \hline \end{array} \begin{array}{|c|} \hline (R) \\ \hline \end{array}
 \end{array}$$

- 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。

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

## 第2章 演算ファンクション

### (3) データタイプ

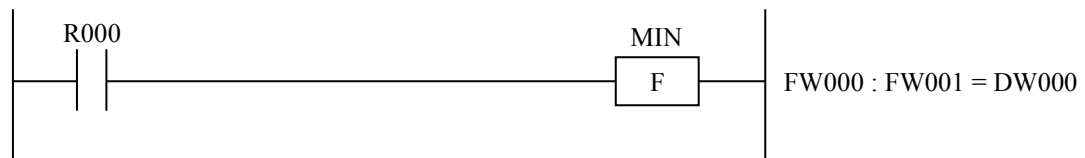
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○
R	○	—	○	—	○	—	○

○：指定できます。

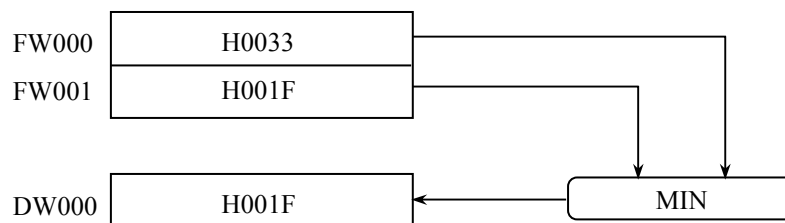
—：指定できません。

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。

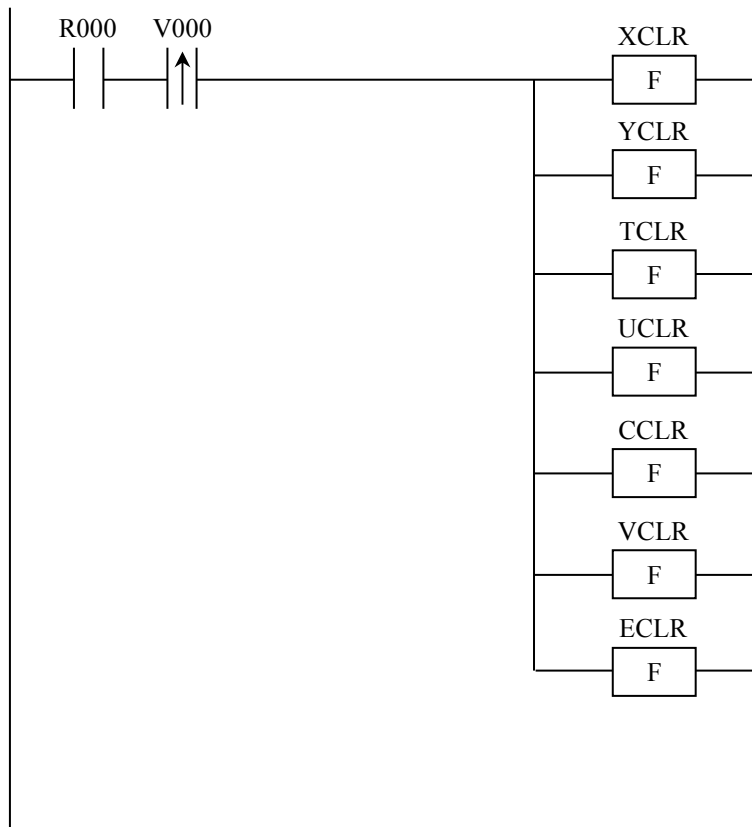
## (2) 機能

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



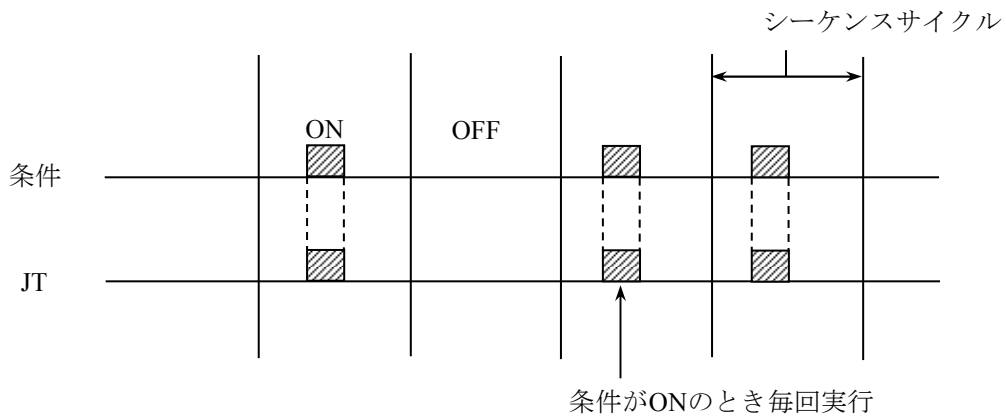
LAB : ジャンプ先ラベル名称

(\*) 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。

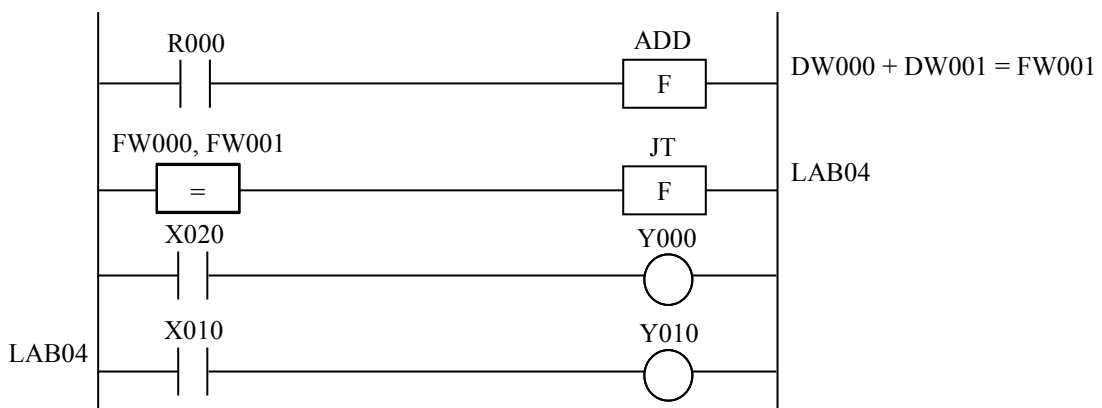
(2) 機能

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、その後ラベルLAB04以降を実行します。

(4) エラー処理

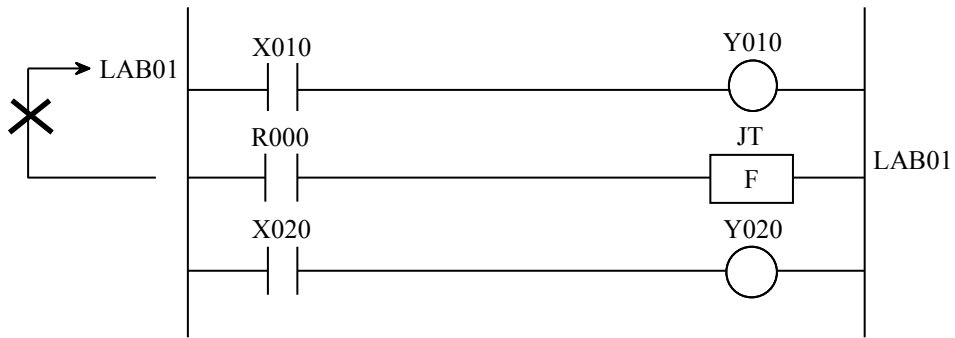
● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

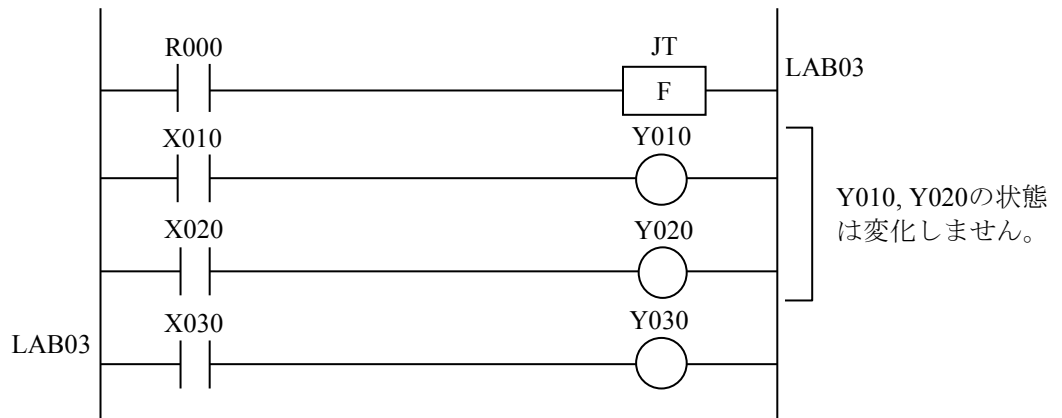
フラグはすべて保持します。

(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。



JMP 無条件ジャンプ : JUMP

(1) 入力形式

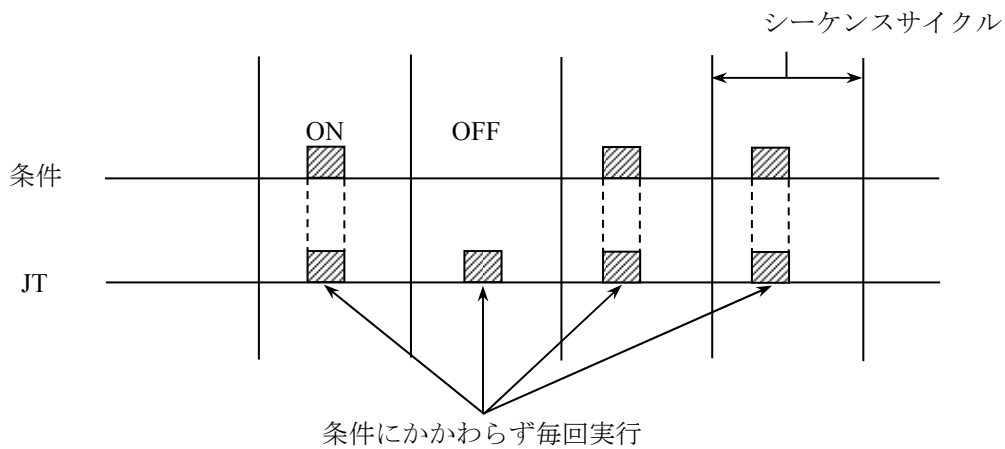
```
JMP LAB
```

LAB : ジャンプ先ラベル名称

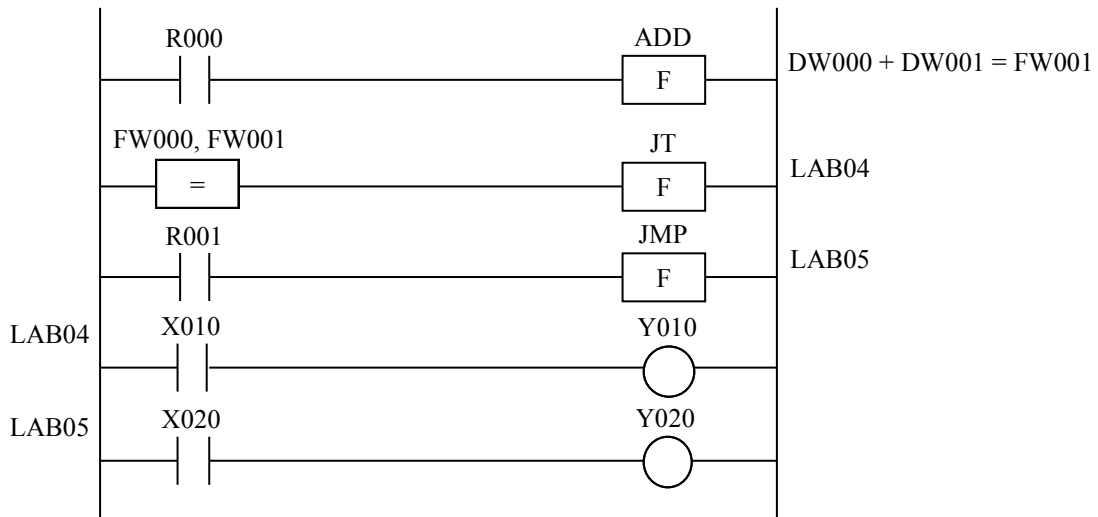
(\*) ファンクション名称とパラメーターの間には、必ずスペースを入力してください。

(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

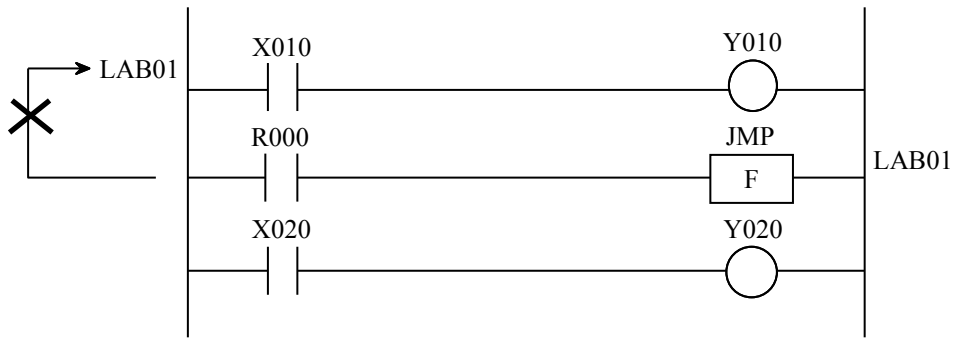
● 演算結果フラグ

X	E	P	N	Z	V
—	—	—	—	—	—

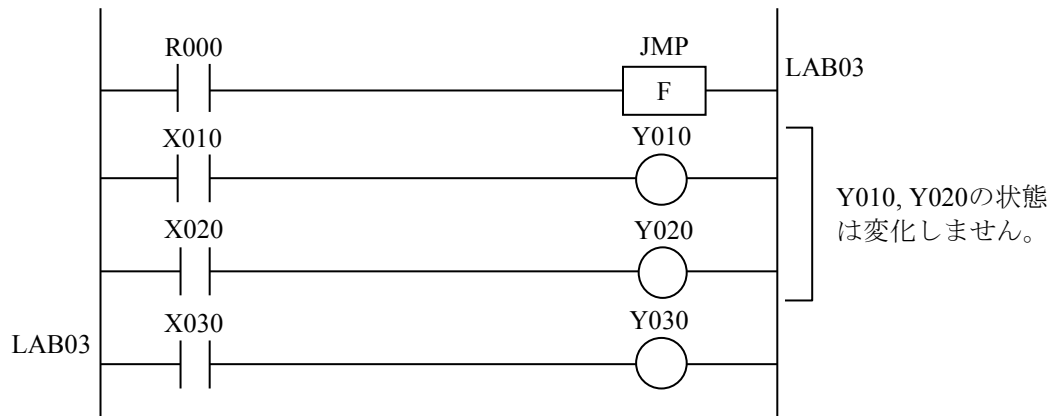
フラグはすべて保持します。

(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

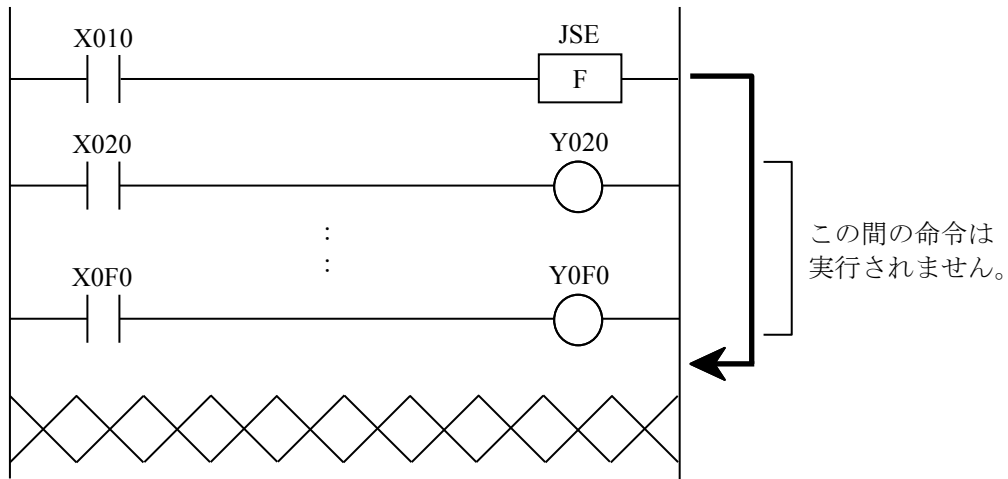


(2) 機能

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。

## 第2章 演算ファンクション

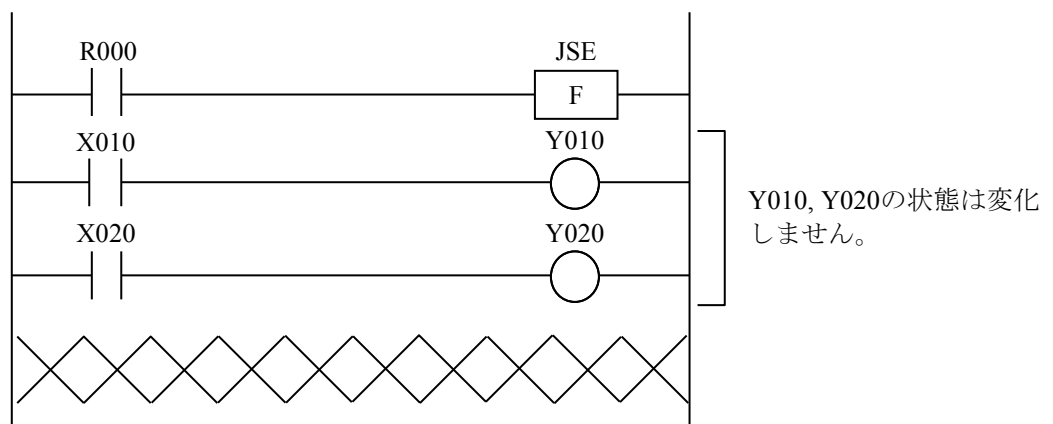
### (4) エラー処理

#### ● 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。



## 2.7 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演算ファンクションを使用してください。

LADDER DIAGRAM SYSTEM/S10VEのイーサネット通信システム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コネクションのオープン (クライアント)
TPOP	TCPコネクションのオープン (サーバー)
TCLO	TCPコネクションのクローズ
TRCV	TCP受信
TSND	TCP送信
UOP	UDPのオープン
UCLO	UDPのクローズ
URCV	UDP受信
USND	UDP送信

システム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	CPU : 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	
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



## 第2章 演算ファンクション

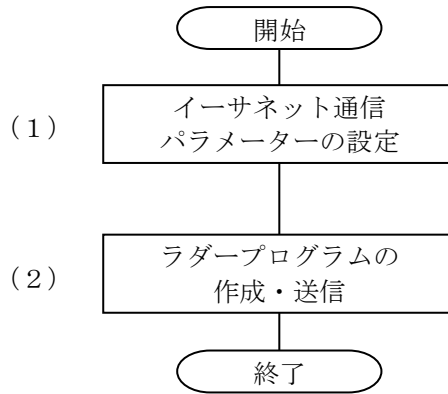
イーサネット通信システム演算ファンクションを実行すると、各管理番号ごとに実行結果がシステムレジスタS09C0～S09EFに設定されます。

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		
SW09C0	S09C0	1	CPU イーサネット1ch, 2ch (ユーザ指定) 通信 用
	S09C1	2	
	}	}	
	S09CE	15	
	S09CF	16	
SW09D0	S09D0	17	ET.NET(メイン) イーサネット1ch, 2ch (ユーザ指定) 通信 用
	S09D1	18	
	}	}	
	S09DE	31	
	S09DF	32	
SW09E0	S09E0	33	ET.NET(サブ) イーサネット1ch, 2ch (ユーザ指定) 通信 用
	S09E1	34	
	}	}	
	S09EE	47	
	S09EF	48	

2. 7. 2 使用方法

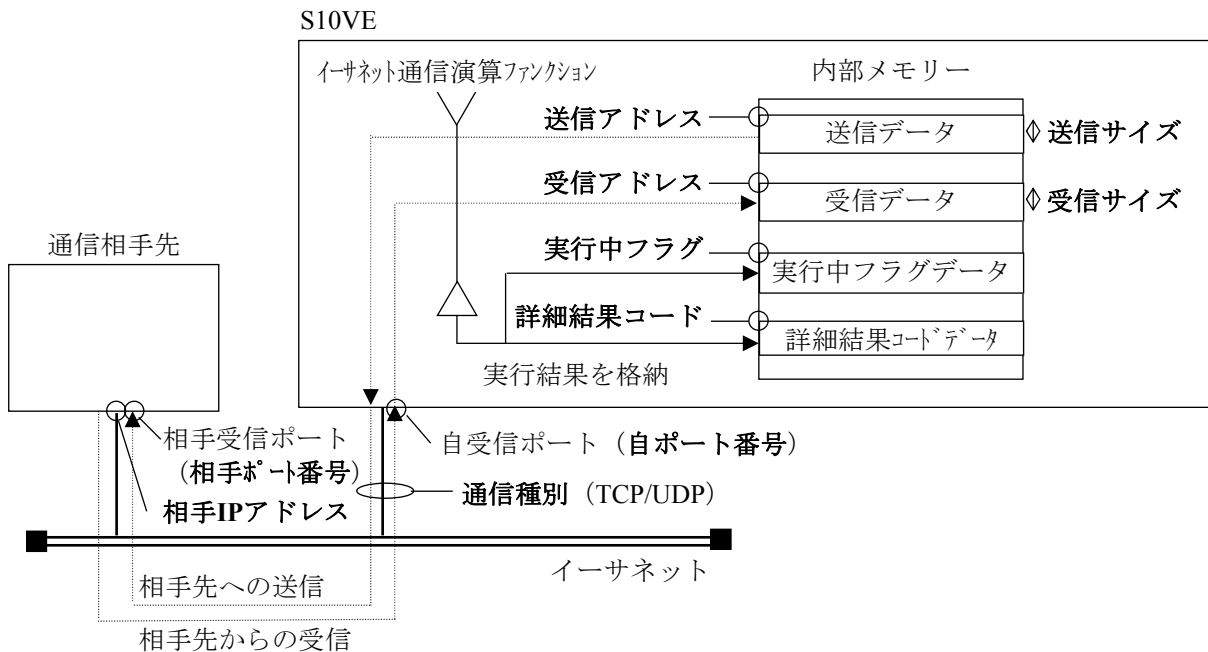
イーサネット通信システム演算ファンクションは、LADDER DIAGRAM SYSTEM/S10VEの [イーサネット通信設定] 画面で設定するパラメーター情報をもとに動作します。そのため、次のフローに示すように、ラダープログラム作成前に [イーサネット通信設定] 画面でパラメーター情報を設定しておく必要があります。



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、LADDER DIAGRAM SYSTEM/S10VE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「S10VE ソフトウェアマニュアル オペレーション ラダー図 For Windows® (マニュアル番号 SEJ-3-131)」を参照してください。

< [イーサネット通信設定] 画面 >

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

管理番号：

[イーサネット通信設定一覧] 画面で指定された管理番号を表示します。

機器名称：

[イーサネット通信設定一覧] 画面で指定された通信を行うモジュールを表示します。  
機器名称は管理番号に対して固定で、以下のモジュールが表示されます。

管理番号	機器名称	ポート	備考
1～16	CPU	ET1	(*1)
		ET2	(*2)
17～32	ET.NET(メイン)	CH1	(*3)
		CH2	(*4)
33～48	ET.NET(サブ)	CH1	(*3)
		CH2	(*4)

(\*1) ET1: CPUのイーサネットポートのET1を使用して通信する場合に選択します。

(\*2) ET2: CPUのイーサネットポートのET2を使用して通信する場合に選択します。

(\*3) CH1: S10VE ET.NET(メイン)またはET.NET(サブ)でイーサネットポートのCH1を使用して通信する場合に選択します。

(\*4) CH2: S10VE ET.NET(メイン)またはET.NET(サブ)でイーサネットポートのCH2を使用して通信する場合に選択します。

### 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。

## 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。

## 第2章 演算ファンクション

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ー	C	○	○	○	○
8	グローバルリンク レジスター	G	○	○	○	○
9	ネスティングコイル	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	システムレジスター	S	○	○	○	○
14	データレジスター	DW	○	○	—	○
15	ワークレジスター	FW	○	○	—	○
16	内部レジスター	M	○	○	○	○
17	内部レジスター (ロングワード)	BD	—	—	—	—
18	高速リモートI/O入力用	I	○	○	—	○
19	高速リモートI/O出力用	O	○	○	—	○
20	HI-FLOW—ラダー間	J	○	○	○	○
21	共有データレジスター	Q	○	○	○	○
22	ワークレジスター	LB	○	○	○	○
23	ワード専用ワーク レジスター	LW	○	○	—	○
24	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。

以下にイーサネット通信システム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	経路情報設定を見直してください。 (*2)
0x800000E7 (ECONNABORTED)	コネクションの接続が打ち切られた。	<ul style="list-style-type: none"> <li>ケーブルの配線を見直してください。</li> <li>接続先ホストのプログラムを見直してください。</li> </ul>
0x800000E8 (ECONNRESET)	接続先ホストのTCP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	<ul style="list-style-type: none"> <li>ケーブルの配線を見直してください。</li> <li>接続先ホストのプログラムを見直してください。</li> </ul>
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。	<ul style="list-style-type: none"> <li>ケーブルの配線を見直してください。</li> <li>プログラムを見直してください。</li> </ul>
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。(内蔵Ethernetを使用した場合のみ)
0x80000516 (EBADF)	イーサネット通信命令失敗	イーサネット通信命令の実行に失敗しています。CPUをリセットしてください。

(\*1) エラーログ情報を参照する方法については、「S10VE ユーザーズマニュアル 総合編 (マニュアル番号 SEJ-1-001)」の「8. 4. 6. 2 エラーログ情報」を参照してください。

(\*2) 経路情報の設定は、BASE SYSTEM/S10VEの [ネットワーク設定] 画面から行ってください。



値	意味	対処方法								
0xFFFFFFFFFA	管理番号未使用	通信ポートがオープンされていません。 ラダープログラムを見直してください。								
0xFFFFFFFFFB	イーサネットモジュールがダウン中	CPU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発生する場合は、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。								
0xFFFFFFFFFC	イーサネットモジュールが未実装	ET.NETモジュールの実装を確認してください。								
0xFFFFFFFFFD	タスク起動失敗	CPUモジュールを一度リセットしてください。再立ち上げしても同じエラーが発生する場合は、モジュールが故障している可能性があります。 モジュールを交換してください。 <table border="1" style="margin: 10px auto;"> <thead> <tr> <th>管理番号</th> <th>交換対象モジュール</th> </tr> </thead> <tbody> <tr> <td>1～16</td> <td>CPUモジュール</td> </tr> <tr> <td>17～32</td> <td>ET.NET(MAIN)</td> </tr> <tr> <td>33～48</td> <td>ET.NET(SUB)</td> </tr> </tbody> </table>	管理番号	交換対象モジュール	1～16	CPUモジュール	17～32	ET.NET(MAIN)	33～48	ET.NET(SUB)
管理番号	交換対象モジュール									
1～16	CPUモジュール									
17～32	ET.NET(MAIN)									
33～48	ET.NET(SUB)									
0xFFFFFFFFFE	管理番号使用中	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がないか確認してください。								
0xFFFFFFFFFF	使用種別不整合 (パラメーター情報の通信種別と異なる)	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

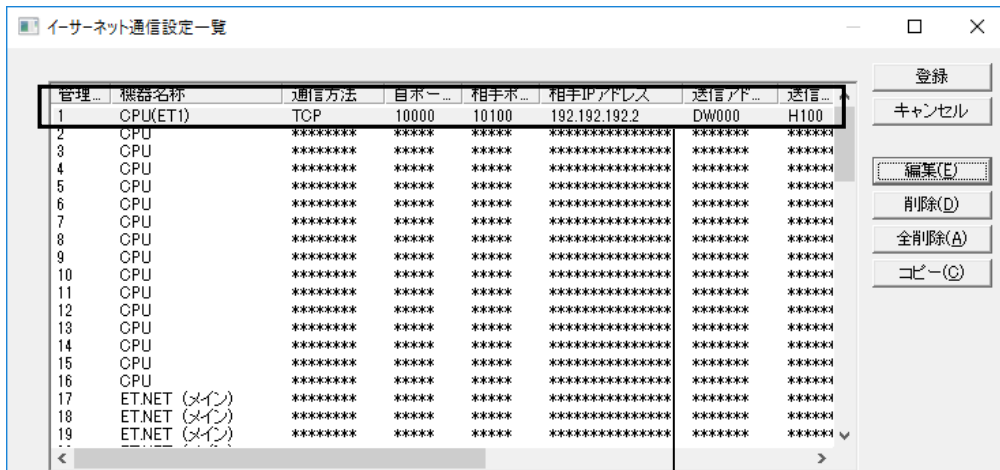
0x8XXXXXXXX : CPMSソケットマクロエラー (実際のCPMSソケットマクロエラーに0x80000000が加算された値)

0xFXXXXXXXX : システムプログラム、タスクでのエラー

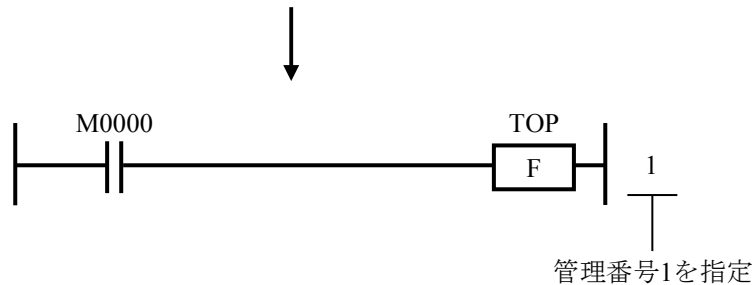
(2) ラダープログラムの作成

イーサネット通信システム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



<ラダープログラム作成時の注意>

イーサネット通信システム演算ファンクションの管理番号に、インデックス指定しないでください。インデックス指定するとラダープログラムのコンパイルはできますが、S10VEで実行するとエラーコード「0x3d00 0006」、内容「[E] Ladder Program error (Illegal SH Instruction)(TN=232)」のエラーログが発生します。

2. 7. 3 命令の詳細

イーサネット通信システム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。

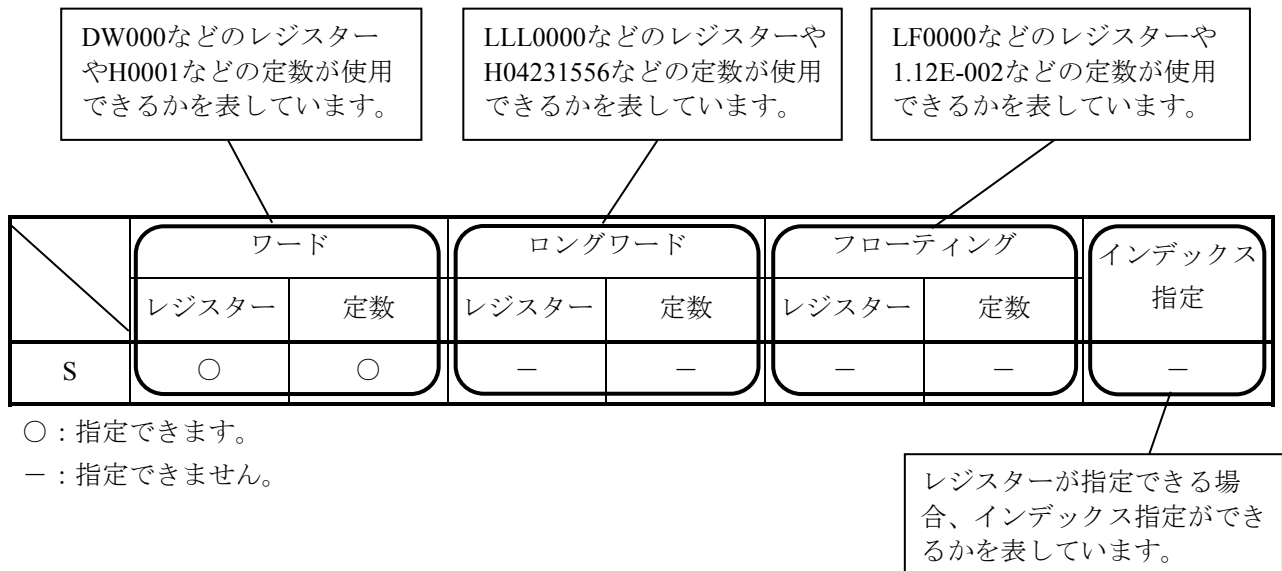
(2) 機能

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（S09C0～S09EF）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)



この例では、S（ソース）にはワードデータのアドレス、および定数を指定できます。

(注) R000, Y01FF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。

## TOP TCPコネクションのオープン（クライアント）：TOP

## (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	CPU
17～32	ET.NET(メイン)
33～48	ET.NET(サブ)

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。

## (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	○	○	—	—	—	—	—

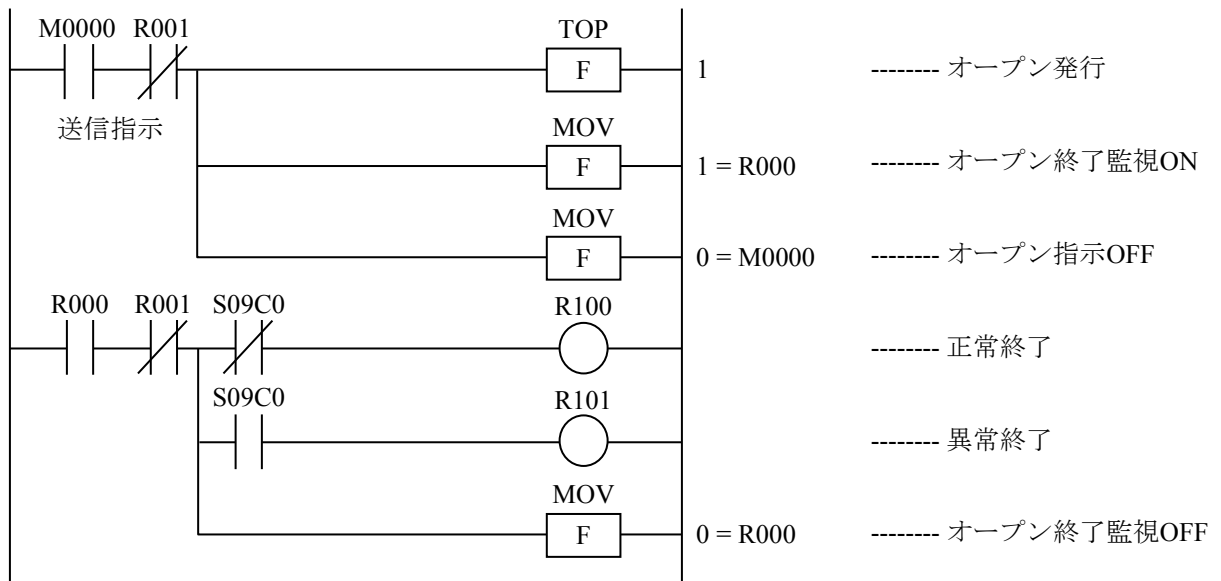
○：指定できます。

—：指定できません。

## 第2章 演算ファンクション

### (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接続のオープン（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	CPU
17～32	ET.NET(メイン)
33～48	ET.NET(サブ)

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。

## (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	○	○	—	—	—	—	—

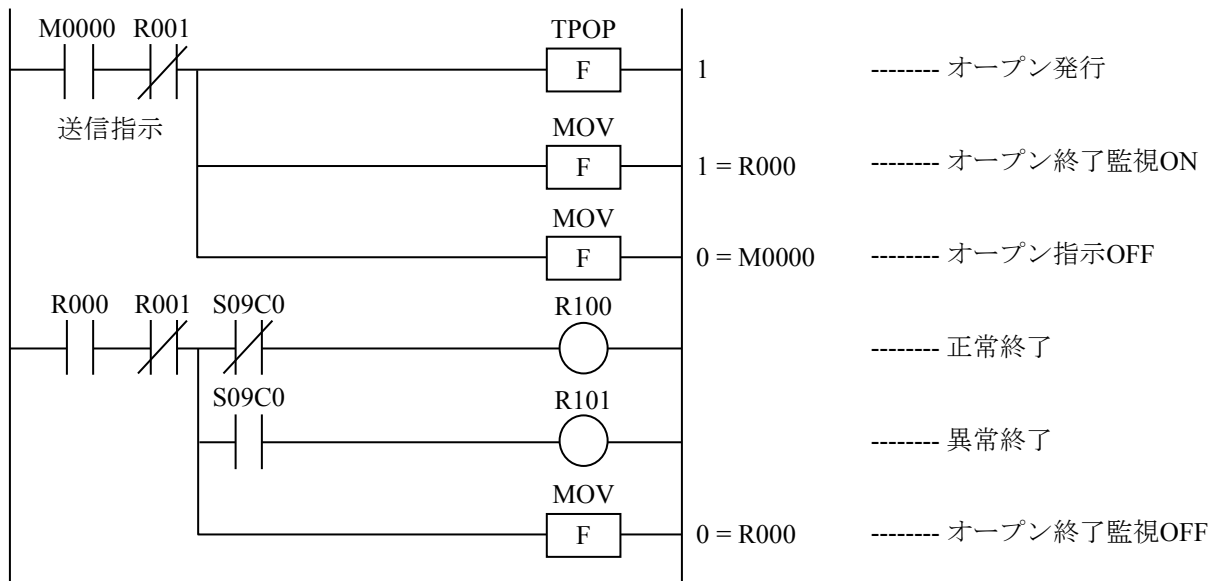
○：指定できます。

—：指定できません。

## 第2章 演算ファンクション

### (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接続のオープン（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コネクションのクローズ : 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	CPU
17~32	ET.NET(メイン)
33~48	ET.NET(サブ)

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。

## (2) 機能

## ● TCPコネクションのクローズ

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。

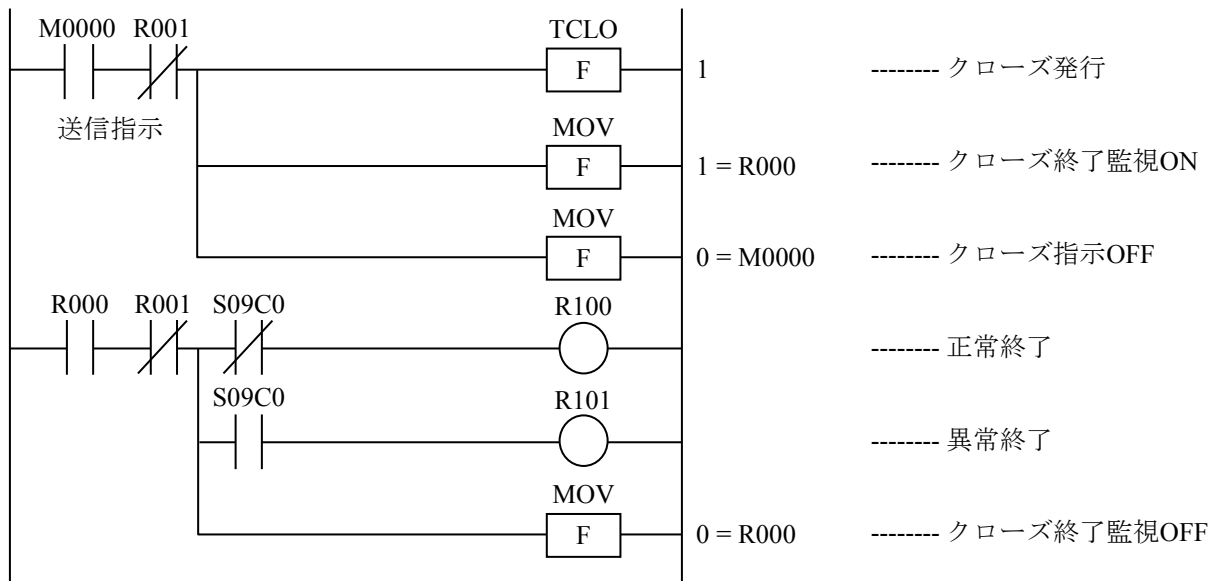
— : 指定できません。



## 第2章 演算ファンクション

### (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。

## TRCV TCP受信 : TRCV

## (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	CPU
17~32	ET.NET(メイン)
33~48	ET.NET(サブ)

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。

## (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。

## 第2章 演算ファンクション

### (3) データタイプ

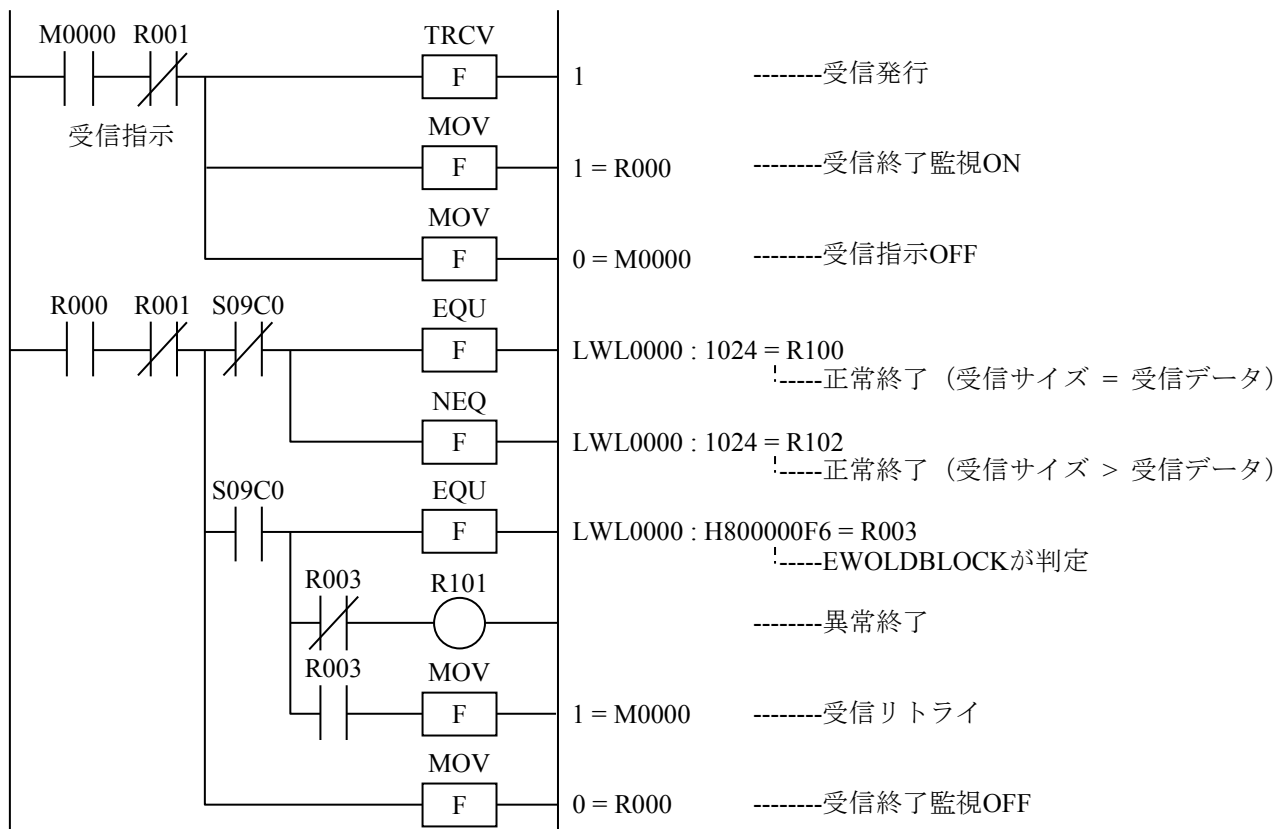
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	—

○：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。

TSND TCP送信 : TSND

(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	CPU
17~32	ET.NET(メイン)
33~48	ET.NET(サブ)

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。

(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

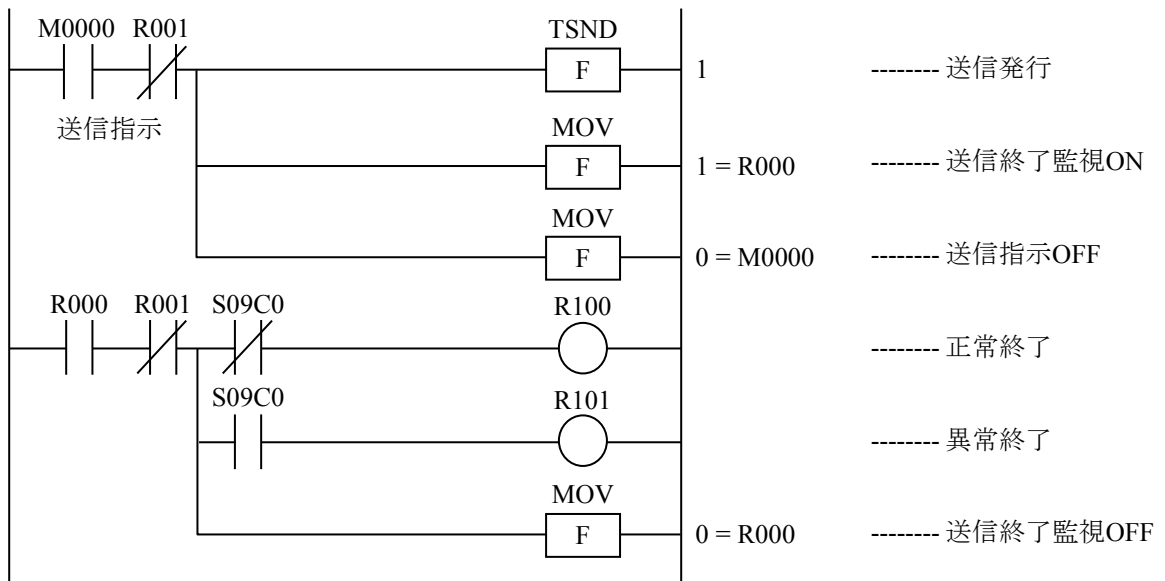
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	—

○ : 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	CPU
17～32	ET.NET(メイン)
33～48	ET.NET(サブ)

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。

(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

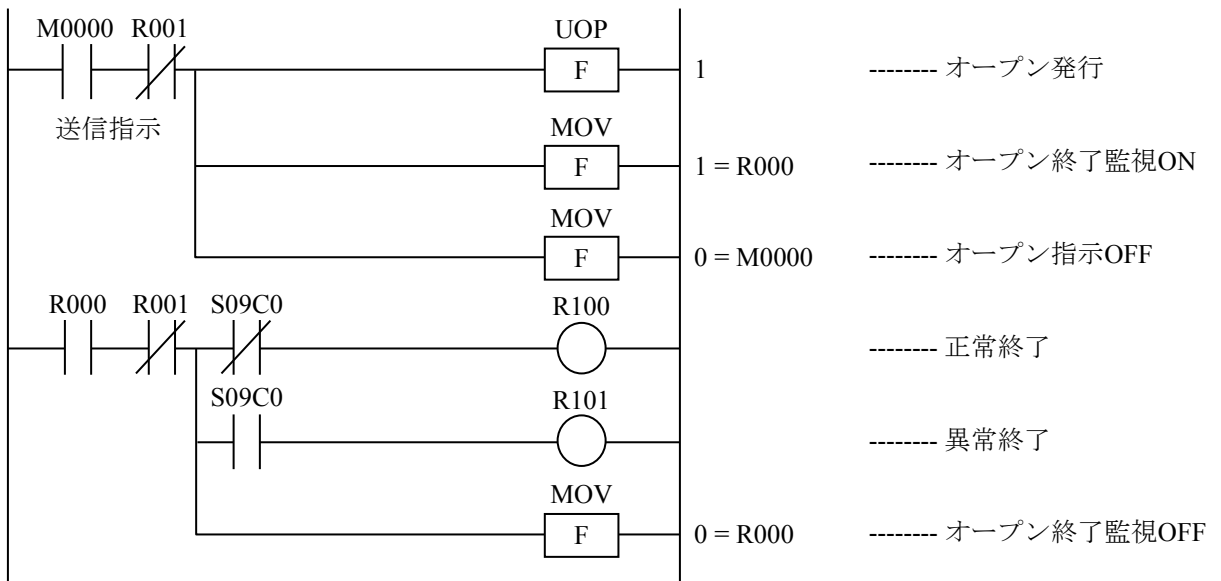
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	—

○ : 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。



UCLO UDPのクローズ : UCLO

(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	CPU
17~32	ET.NET(メイン)
33~48	ET.NET(サブ)

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。

(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

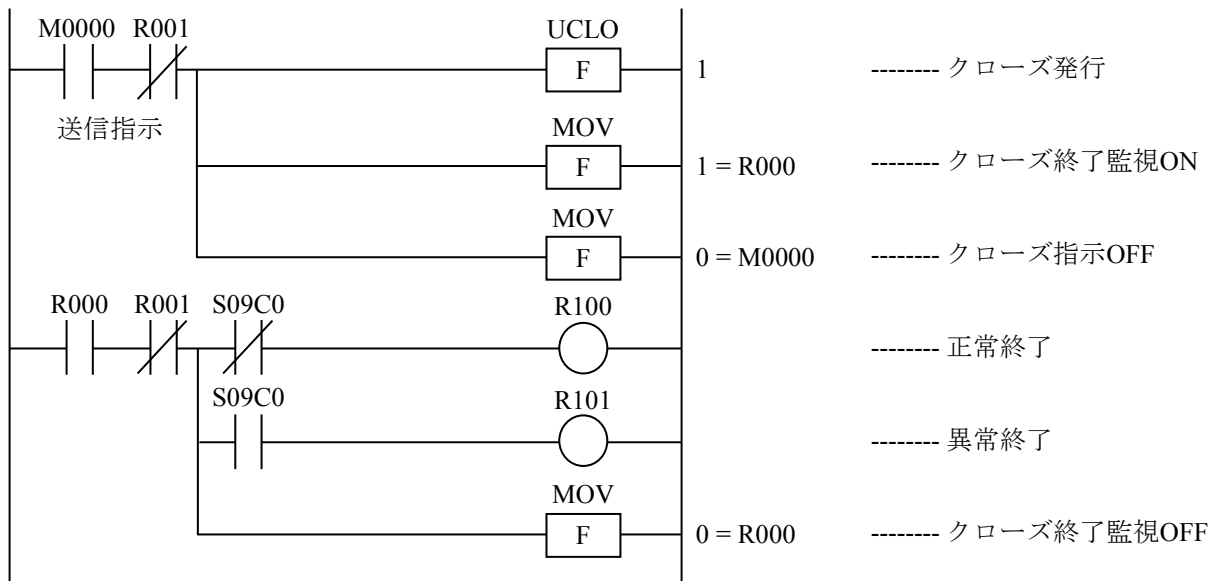
	ワード		ロングワード		フローティング		インデックス 指定
	レジスター	定数	レジスター	定数	レジスター	定数	
S	○	○	—	—	—	—	—

○ : 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。

URCV UDP受信 : URCV

(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	CPU
17～32	ET.NET(メイン)
33～48	ET.NET(サブ)

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。

(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。

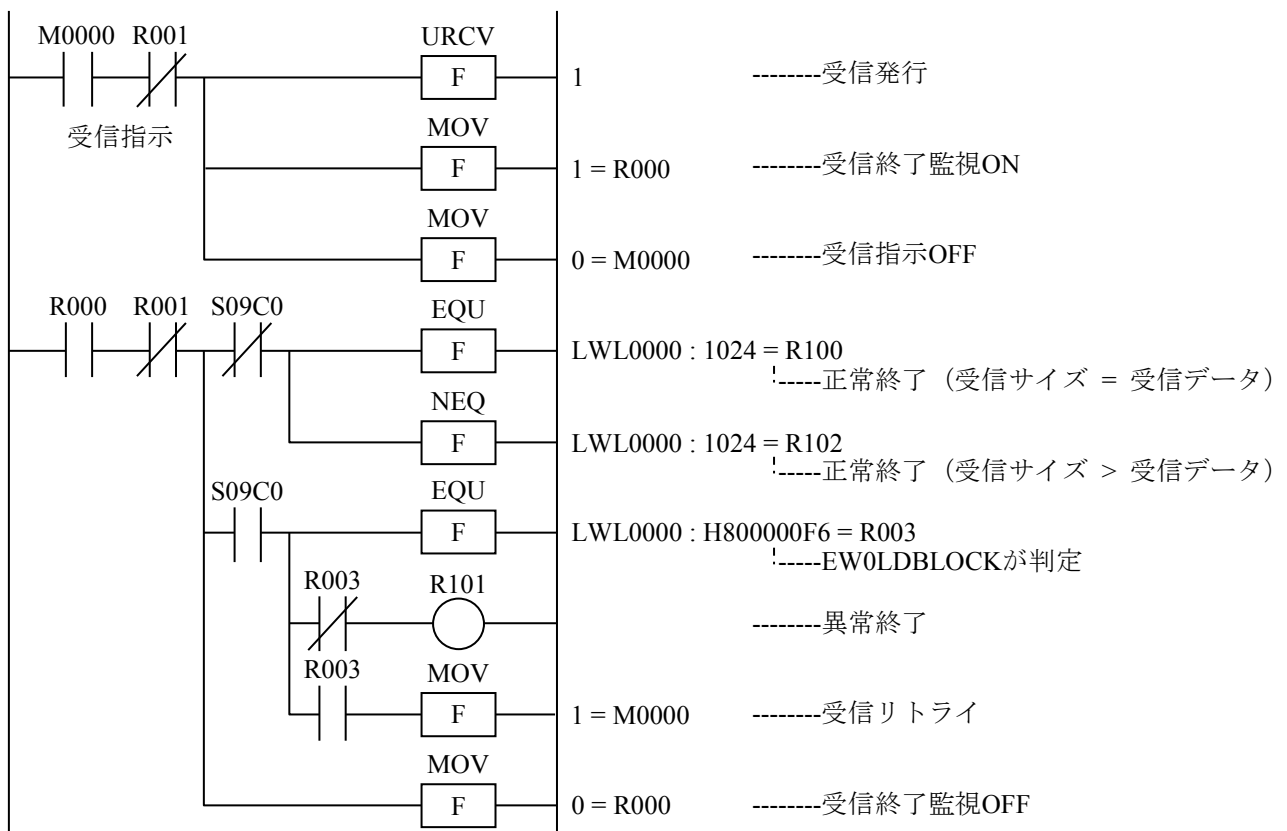
(3) データタイプ

	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



## 第2章 演算ファンクション

### (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。

## USND UDP送信 : USND

## (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	CPU
17~32	ET.NET(メイン)
33~48	ET.NET(サブ)

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。

## (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	○	○	—	—	—	—	—

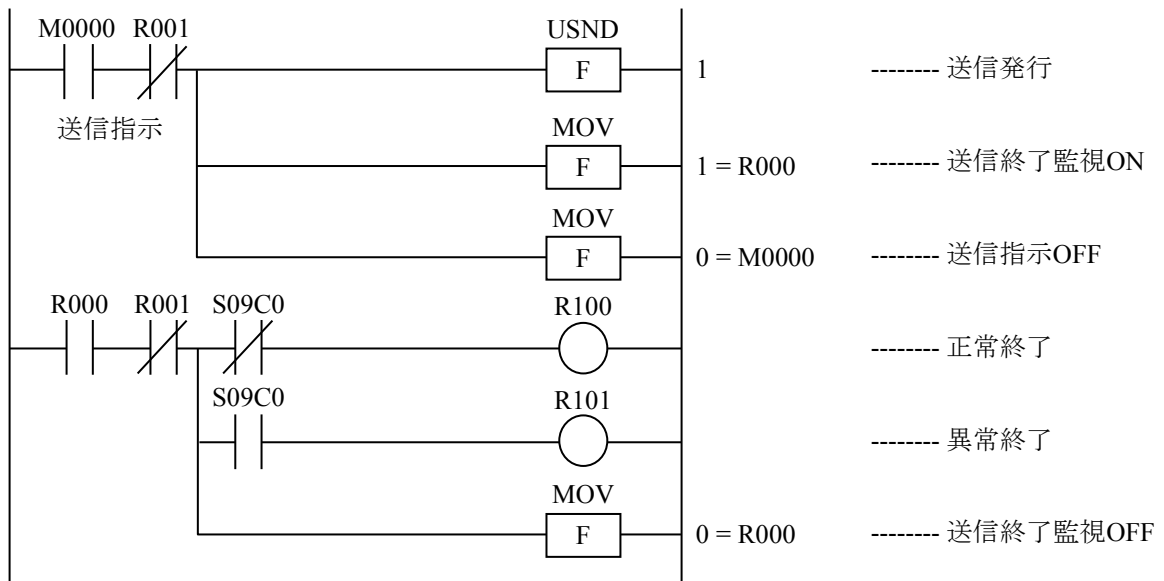
○ : 指定できます。

— : 指定できません。

## 第2章 演算ファンクション

### (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
—	—	—	—	—	—

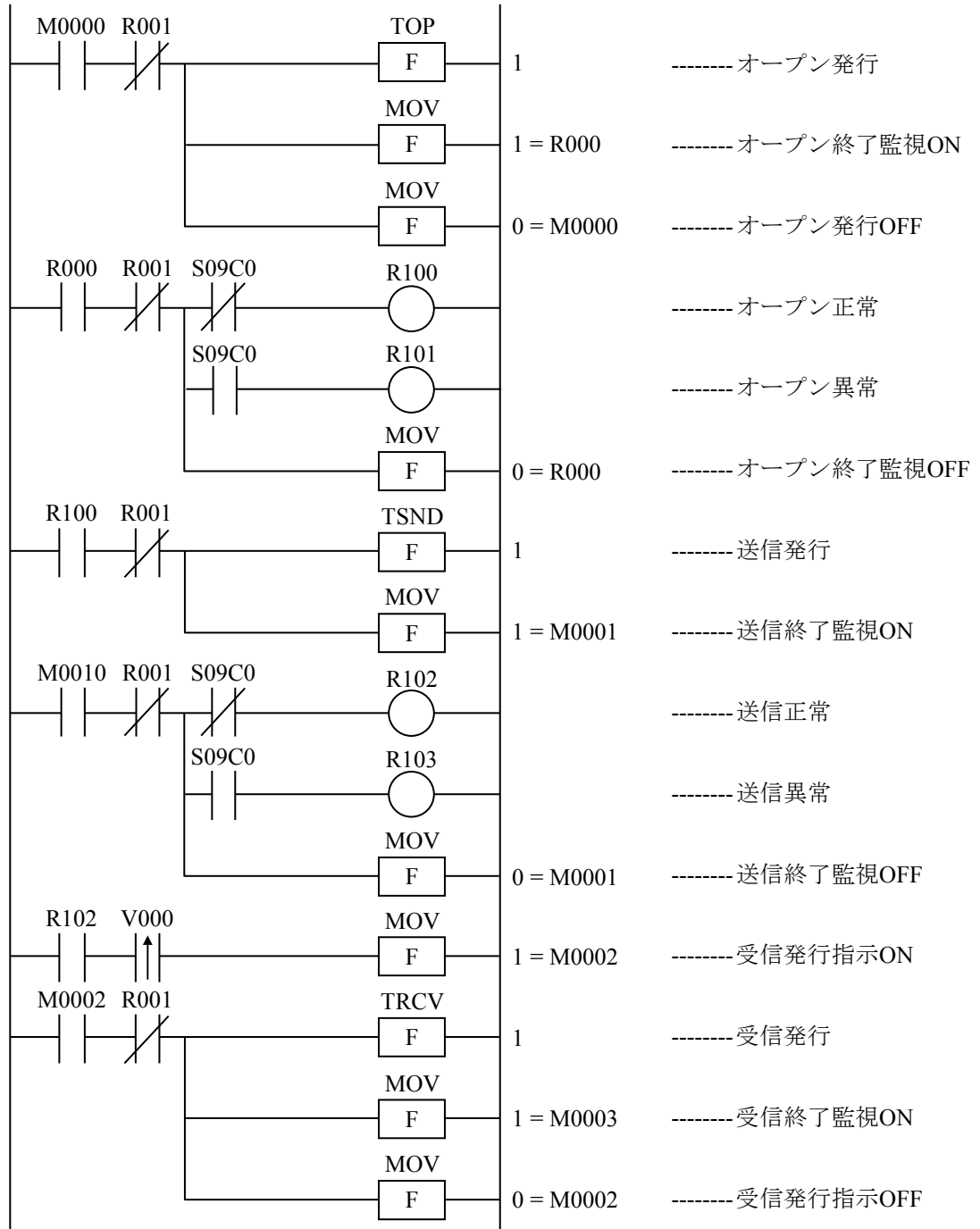
フラグはすべて保持します。

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。

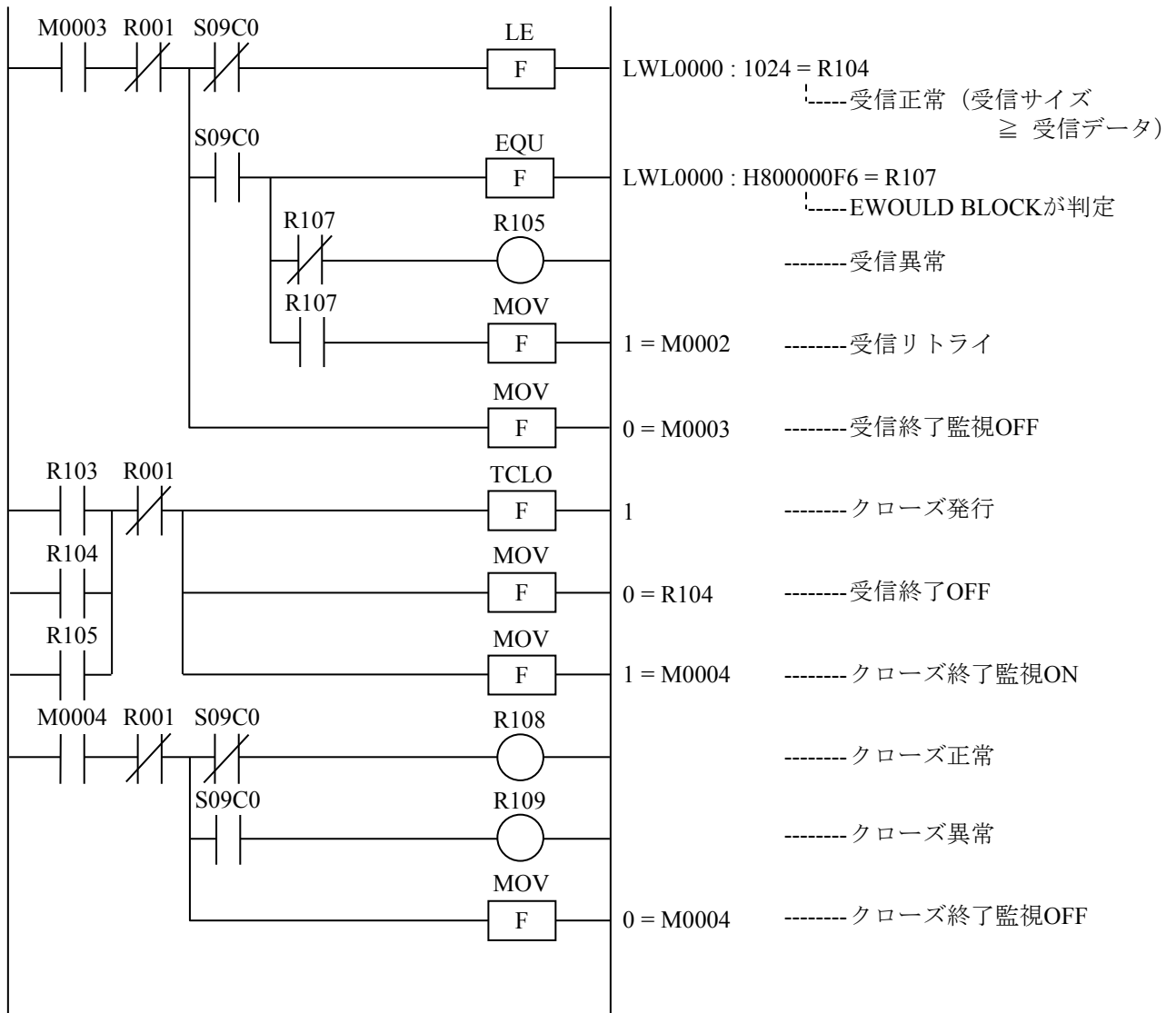
このサンプルプログラムは、パラメーター設定情報に管理番号 = 1、実行中フラグ = R001、詳細結果コード = LWL0000を設定したものです。

(1) TCPクライアント

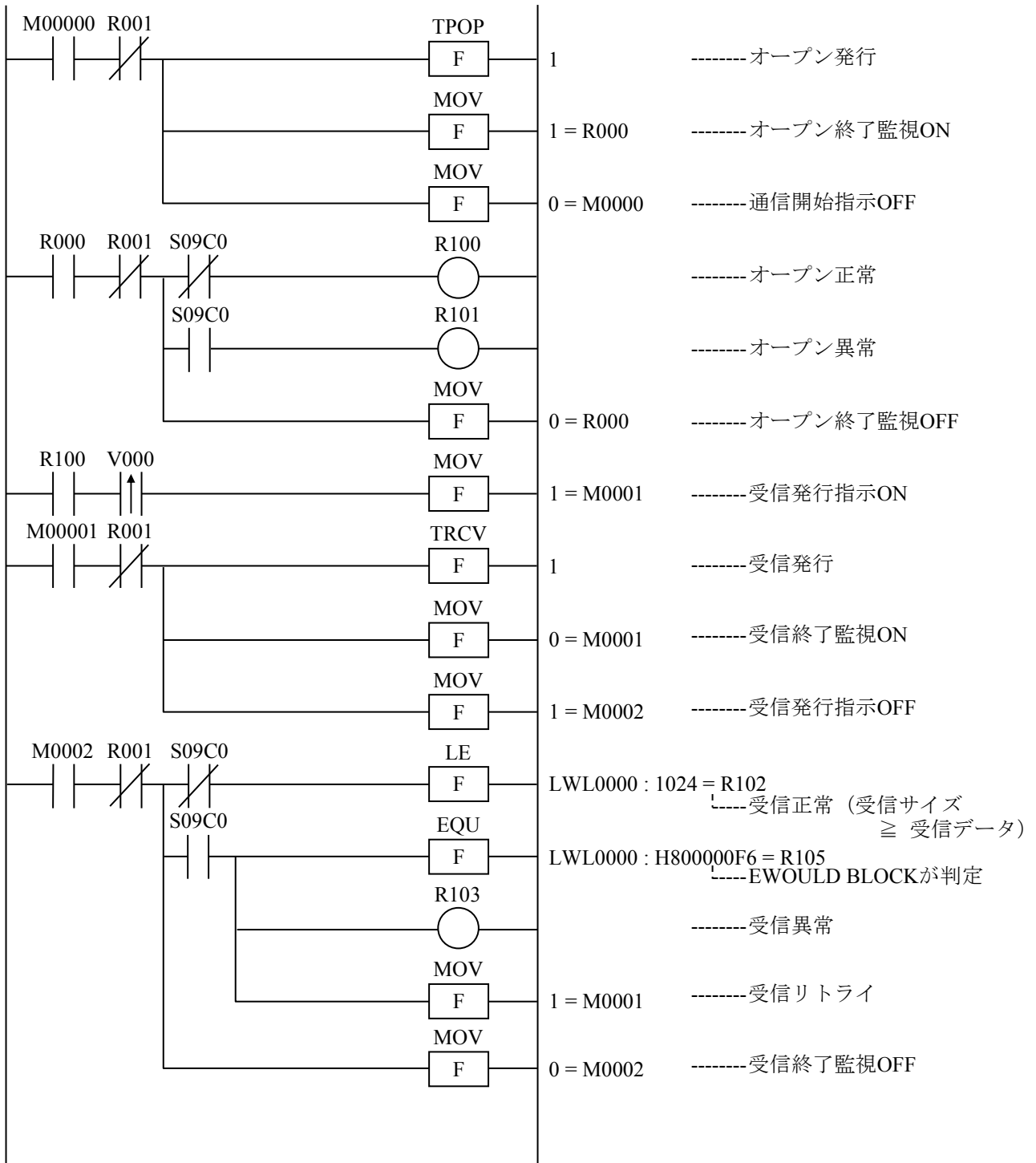


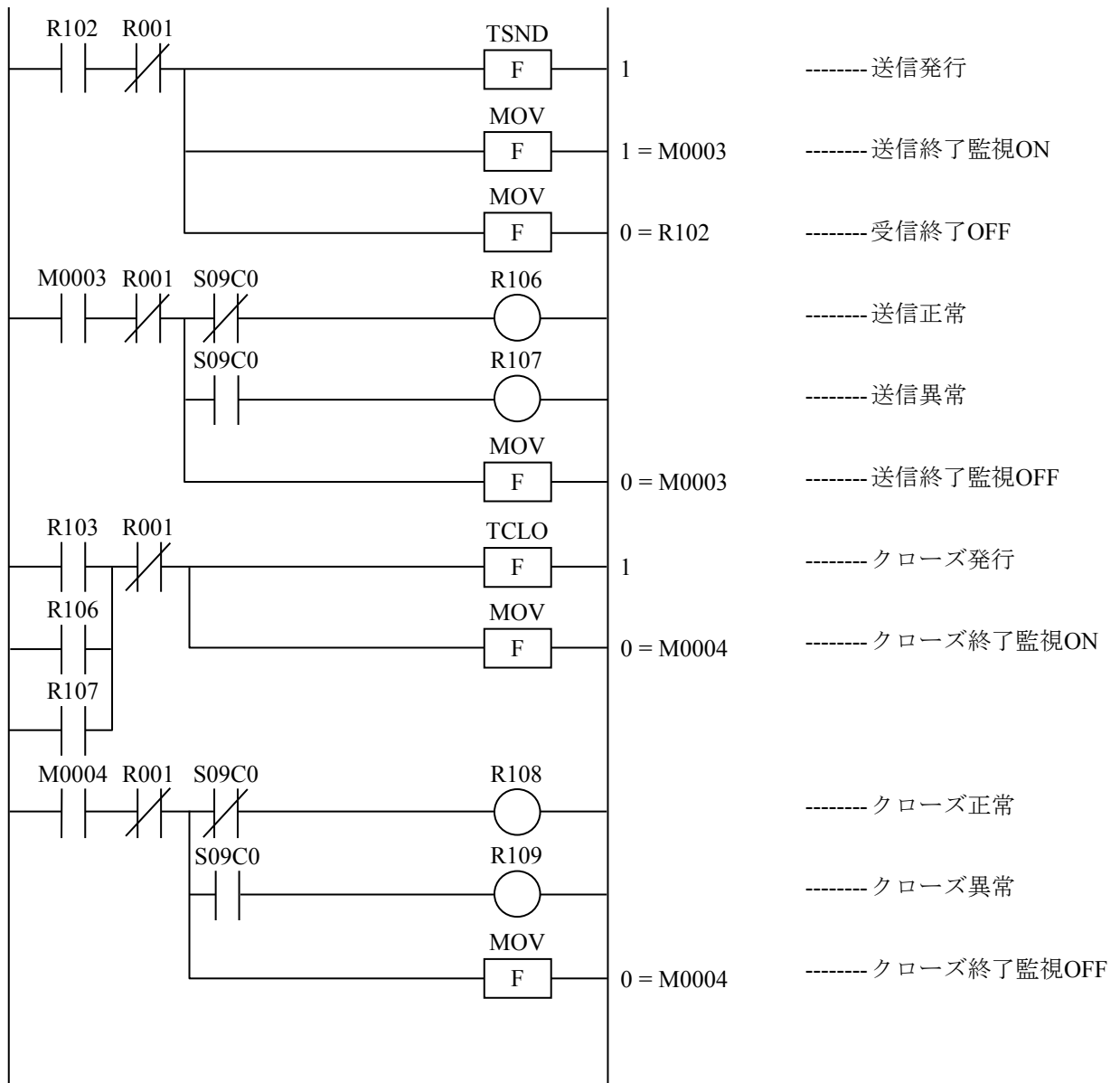


第2章 演算ファンクション



(2) TCPサーバー





## 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、LADDER DIAGRAM SYSTEM/S10VE（型式：S-7898-02）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。

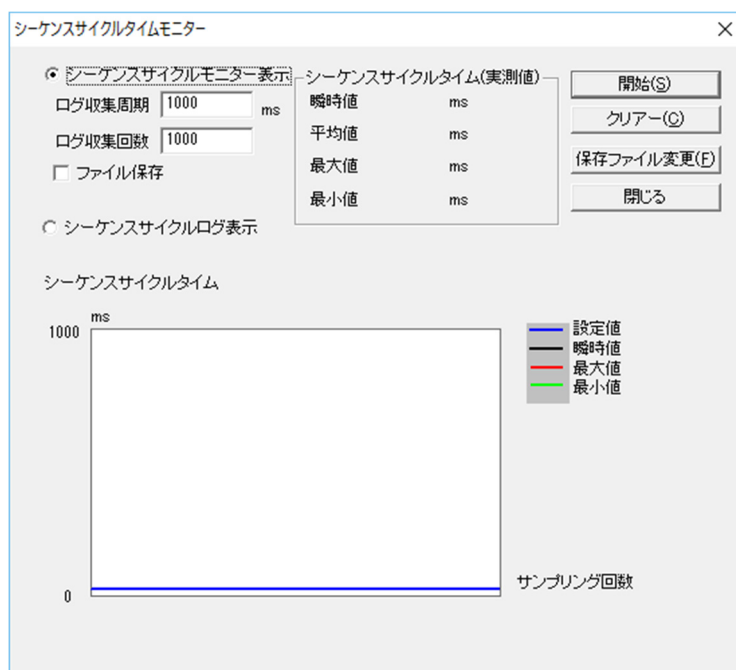
## A. 1 LADDER DIAGRAM SYSTEM/S10VEによる確認

LADDER DIAGRAM SYSTEM/S10VE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、LADDER DIAGRAM SYSTEM/S10VEのメニューから [ユーティリティ] - [制御状態モニター] - [シーケンスサイクルモニター] を選択します。

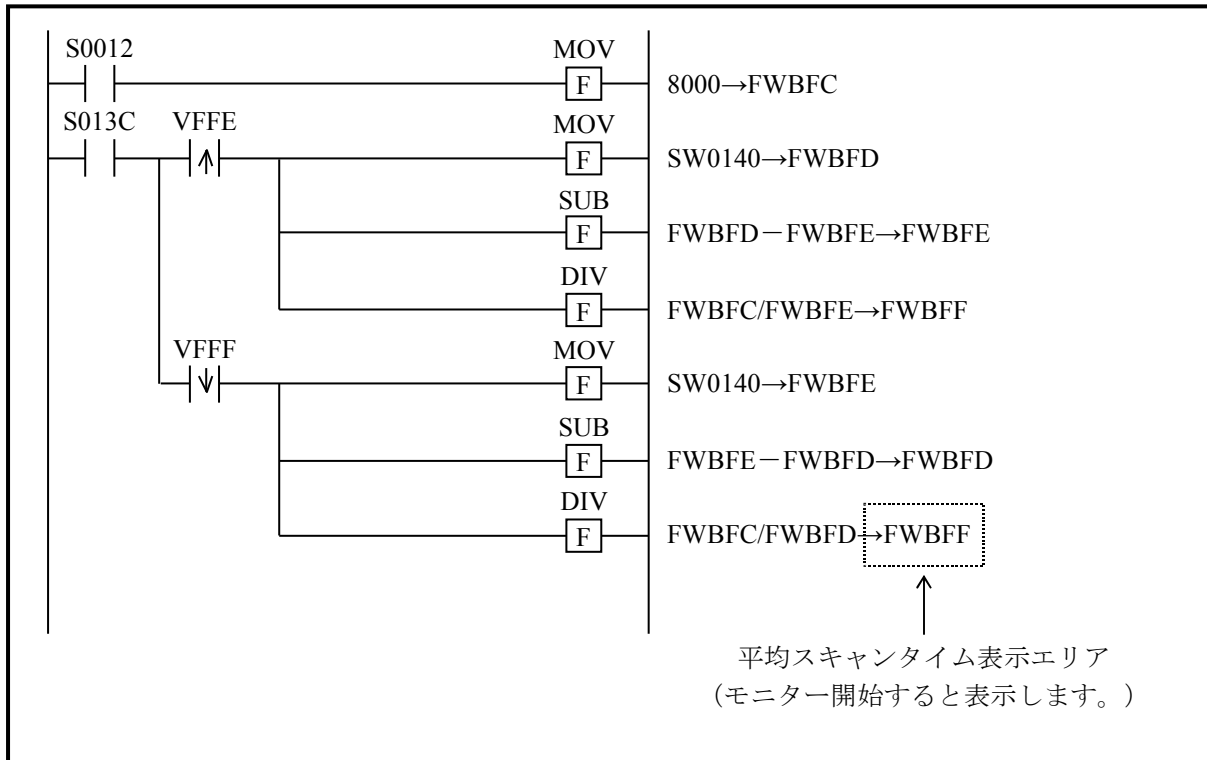
シーケンスサイクルモニターの操作方法は、「S10VE ソフトウェアマニュアル オペレーション ラダー図 For Windows®（マニュアル番号 SEJ-3-131）」を参照してください。

< [シーケンスサイクルタイムモニター] 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Z-1 スキャンタイム表示プログラム