

プログラマブルコントローラ

FP7 CPUユニット

**命令語マニュアル**

---



## はじめに

このたびは、パナソニック製品をお買い上げいただき、誠にありがとうございます。ご使用前に、施工説明書およびユーザーズマニュアルをよくお読みになり、十分に内容をご理解いただいたうえで、正しくご使用くださいますようお願いいたします。

## マニュアルの種類

- FP7 シリーズのユーザーズマニュアルには、以下の種類があります。使用するユニット、用途に合わせてご参照ください。
- マニュアルは、当社ホームページ [http://industrial.panasonic.com/ac/j/dl\\_center/manual/](http://industrial.panasonic.com/ac/j/dl_center/manual/) よりダウンロードすることができます。

ユニット名 または 用途	マニュアル名	マニュアル記号
FP7 電源ユニット	FP7 CPU ユニット ユーザーズマニュアル (ハードウェア編)	WUMJ-FP7CPUH
FP7 CPU ユニット	FP7 CPU ユニット 命令語マニュアル	WUMJ-FP7CPUPGR
	FP7 CPU ユニット ユーザーズマニュアル (ロギングトレース機能編)	WUMJ-FP7CPULOG
	FP7 CPU ユニット ユーザーズマニュアル (セキュリティ機能編)	WUMJ-FP7CPUSEC
	内蔵 COM ポート使用方法	FP7 シリーズユーザーズマニュアル(SCU 通信編)
FP7 拡張(通信)カセット (RS-232C,RS485 タイプ)		
FP7 拡張(通信)カセット (Ethernet タイプ)	WUMJ-FP7CCET	
内蔵 LAN ポート使用方法	FP7 CPU ユニット ユーザーズマニュアル (LAN ポート通信編)	WUMJ-FP7LAN
FP7 デジタル入出力ユニット	FP7 デジタル入出力ユニット ユーザーズマニュアル	WUMJ-FP7DIO
FP7 アナログ入力ユニット	FP7 アナログ入力ユニット ユーザーズマニュアル	WUMJ-FP7AIH
FP7 アナログ出力ユニット	FP7 アナログ出力ユニット ユーザーズマニュアル	WUMJ-FP7AOH
FP7 高速カウンタユニット	FP7 高速カウンタユニット ユーザーズマニュアル	WUMJ-FP7HSC
FP7 位置決めユニット	FP7 位置決めユニット ユーザーズマニュアル	WUMJ-FP7POSP
FP7 シリアルコミュニケーションユニット	FP7 シリーズユーザーズマニュアル(SCU 通信編)	WUMJ-FP7COM
PHLS システム	PHLS システム ユーザーズマニュアル	WUMJ-PHLS
プログラミングソフトウェア FPWIN GR7	FPWIN GR7 導入ガイド	WUMJ-FPWINGR7

# もくじ

<b>1 命令語概要</b> .....	<b>1-1</b>
1-1 命令の構成 .....	1-2
基本命令の構成パターン .....	1-2
応用命令の構成パターン .....	1-4
1-2 演算単位 .....	1-8
1-3 演算用デバイス一覧 .....	1-10
1-4 デバイス番号の指定 .....	1-12
1-5 リレーの説明 .....	1-14
X 外部入力 .....	1-14
Y 外部出力 .....	1-14
R 内部リレー .....	1-15
SR システムリレー .....	1-15
T タイマ .....	1-16
C カウンタ .....	1-16
L リンクリレー .....	1-17
P パルスリレー .....	1-18
E 異常報知リレー .....	1-19
IN 直接入力 .....	1-21
OT 直接出力 .....	1-21
1-6 メモリエリアの説明 .....	1-22
DT データレジスタ .....	1-22
LD リンクレジスタ .....	1-23
UM ユニットメモリ .....	1-24
SD システムデータレジスタ .....	1-24
WX, WY, WR, WL, WI, WO .....	1-25
TS, CSタイマ / カウンタ設定値レジスタ .....	1-26
TE, CEタイマ / カウンタ経過値レジスタ .....	1-27
I0~IE インデックスレジスタ .....	1-28
1-7 定数の説明 .....	1-32
K 符号付き 10 進定数 .....	1-32
U 符号なし 10 進定数 .....	1-33
H 16 進定数 .....	1-33

SF単精度浮動小数点形実数定数.....	1-34
DF倍精度浮動小数点形実数定数.....	1-35
“” 文字定数.....	1-36
1-8 グローバルデバイスとローカルデバイス.....	1-37
1-9 PLC内部で扱えるデータの範囲.....	1-38
1-10 オーバーフロー・アンダーフロー.....	1-40

## 2 基本命令 .....2-1

ST, ST/, OT (スタート、スタートノット、アウト).....	2-2
AN, AN/ (アンド、アンドノット).....	2-2
OR, OR/ (オア、オアノット).....	2-2
ST ↑, ST ↓ (立ち上がり、立ち下がり接点命令).....	2-5
AN ↑, AN ↓ (立ち上がり、立ち下がり接点命令).....	2-5
OR ↑, OR ↓ (立ち上がり、立ち下がり接点命令).....	2-5
/ (ノット).....	2-7
DF, DF/ (立ち上がり微分、立ち下がり微分).....	2-8
DFI (立ち上がり微分(イニシャル実行型)).....	2-8
ANS (アンドスタック).....	2-14
ORS (オアスタック).....	2-15
PUSHS (プッシュスタック).....	2-16
RDS (リードスタック).....	2-16
POPS (ポップスタック).....	2-16
NOP (ノップ).....	2-19
↑OT, ↓OT (立ち上がり、立ち下がり検出アウト).....	2-20
KP (キープ).....	2-21
SET, RST (セット、リセット).....	2-22
ALT (オルタネートアウト).....	2-24
TM (タイマ).....	2-25
SPTM (符号なし 32 ビット加算式補助タイマ).....	2-33
CT (ダウンカウンタ).....	2-37
UDC (アップダウンカウンタ).....	2-44
SR (シフトレジスタ).....	2-47
LRSR (左右シフトレジスタ).....	2-51
MC (マスタコントロールリレー).....	2-54
MCE (マスタコントロールリレーエンド).....	2-54
JP,LBL (ジャンプ、ラベル).....	2-59

LOOP, LBL (ループ、ラベル) .....	2-64
ED (エンド) .....	2-69
EDPB (エンドプログラムブロック) .....	2-70
CNDE (条件付きエンド) .....	2-71
EJECT (エジェクト) .....	2-72
SSTP (スタートステップ).....	2-73
NSTL (ネクストステップ).....	2-73
CSTP (クリアステップ) .....	2-73
STPE (ステップエンド) .....	2-73
ZRST (ブロッククリア) .....	2-82
CALL (サブルーチンコール).....	2-84
SBL (サブルーチンラベル).....	2-84
RET (サブルーチンリターン) .....	2-84
FCAL (出力OFF型ローカルサブルーチンコール) .....	2-88
ECAL (サブルーチンコール(PB番号指定)).....	2-90
EFCAL (出力強制OFF型サブルーチンコール(PB番号指定)) .....	2-92
COMOUT (コメントアウト).....	2-94
ENDCOM (コメントアウト終了).....	2-94
ST=,ST<>,ST>,ST>=,ST<,ST<= (データ比較(スタート)) .....	2-95
AN=,AN<>,AN>,AN>=,AN<,AN<= (データ比較(アンド)) .....	2-95
OR=,OR<>,OR>,OR>=,OR<,OR<= (データ比較(オア)).....	2-95

### 3 応用命令 .....3-1

CMP (データ比較).....	3-2
WIN (帯域比較).....	3-5
MV (データ転送).....	3-8
MV/ (データ反転転送).....	3-10
BKMV (ブロック転送).....	3-12
COPY (ブロック複写).....	3-14
RST (リセット).....	3-16
BTM (ビットブロック転送) .....	3-18
ZRST (ブロッククリア).....	3-20
DGT (デジタルデータ転送).....	3-22
XCH (データ交換).....	3-25
SWAP (上位バイト下位バイト交換).....	3-27
PUSHIX (インデックスレジスタ退避).....	3-28
POPIX (インデックスレジスタ復帰).....	3-29

ADD (加算).....	3-30
SUB (減算).....	3-32
MUL (乗算).....	3-34
DIV (除算).....	3-36
DIVMOD (除算).....	3-38
INC (インクリメント).....	3-40
DEC (デクリメント).....	3-41
BCDADD (BCDデータ加算).....	3-42
BCDSUB (BCDデータ減算).....	3-44
BCDMUL (BCDデータ乗算).....	3-46
BCDDIV (BCDデータ除算).....	3-48
BCDINC (BCDデータインクリメント).....	3-50
BCDDEC (BCDデータデクリメント).....	3-52
AND (論理積).....	3-54
OR (論理和).....	3-56
XOR (排他的論理和).....	3-58
XNR (排他的論理和否定).....	3-60
COMB (結合).....	3-62
BCC (ブロックチェックコード計算).....	3-64
CRC (CRCコード計算).....	3-67
HEXA (HEX→16進ASCII変換).....	3-70
AHEX (16進ASCII→HEX変換).....	3-72
BCDA (BCD→10進ASCII変換).....	3-75
ABCD (10進ASCII→BCD変換).....	3-78
BINA (BIN→10進ASCII変換).....	3-82
ABIN (10進ASCII→BIN変換).....	3-84
BTOA (BIN→ASCII変換).....	3-86
ATOB (ASCII→BIN変換).....	3-98
ACHK (ASCIIデータチェック).....	3-110
INV (データ反転).....	3-112
NEG (符号反転).....	3-113
ABS (絶対値).....	3-114
EXT (符号拡張).....	3-115
BCD (BCDデータ変換).....	3-116
BIN (BCD→BIN変換).....	3-117
DECO (デコード).....	3-118
SEGT (7セグメントデコード).....	3-120
ENCO (エンコード).....	3-122

UNIT (デジッ結合).....	3-124
DIST (デジッ分離).....	3-125
BUNI (バイトデータ結合).....	3-126
BDIS (バイトデータ分離).....	3-128
GRY (バイナリ→グレイコード変換).....	3-130
GBIN (グレイコード→BIN変換).....	3-132
COLM (ビット行→ビット列変換).....	3-134
LINE (ビット列→ビット行変換).....	3-136
SHR (nビット右シフト).....	3-138
SHL (nビット左シフト).....	3-140
BSR (n桁右シフト).....	3-142
BSL (n桁左シフト).....	3-144
BITR (nビット分の一括右シフト).....	3-146
BITL (nビット分の一括左シフト).....	3-148
WSHR (ブロック領域のnワードの右シフト).....	3-150
WSHL (ブロック領域のnワードの左シフト).....	3-152
ROR (データの右回転).....	3-154
ROL (データの左回転).....	3-156
RCR (データのキャリー込み右回転).....	3-158
RCL (データのキャリー込み左回転).....	3-160
CMPR (圧縮シフト読み出し).....	3-162
CMPW (圧縮シフト書き込み).....	3-164
DEFBUF (バッファ定義).....	3-166
FIFR (データ読み出し: 先入・先出).....	3-168
BUFW (データ書き込み).....	3-170
LIFR (データ読み出し(後入・先出)).....	3-172
BTI (ビット反転).....	3-174
BTT (ビットテスト).....	3-175
STC (キャリーのセット).....	3-177
CLC (キャリーのリセット).....	3-178
SSET (文字定数→ASCIIコード変換).....	3-179
SCMP (文字列比較).....	3-182
SADD (文字列加算).....	3-184
LEN (文字列長取得).....	3-185
SSRC (文字列検索).....	3-186
RIGHT (文字列右側取り出し).....	3-188
LEFT (文字列左側取り出し).....	3-190
MIDR (文字列任意位置からの読み出し).....	3-192



MIDW (文字列任意位置からの書き換え).....	3-194
SREP (文字列置き換え).....	3-197
SRC (データサーチ).....	3-200
BCU (ONビットカウント).....	3-203
MAX (最大値取得).....	3-205
MIN (最小値取得).....	3-209
MEAN (合計・平均値取得).....	3-214
SORT (並び替え).....	3-218
SCAL (線形化).....	3-221
EVENTC (事象発生回数計数命令).....	3-224
EVENTT (事象発生時間計数命令).....	3-227
PID (PID演算).....	3-230
EZPID (PID演算:PWM出力可能).....	3-235
DTR (データ変化検出).....	3-244
RAMP (傾斜出力).....	3-246
LIMIT (上下限リミット制御).....	3-248
BAND (不感帯制御).....	3-250
ZONE (ゾーン制御).....	3-252
FILTR (時定数処理).....	3-254
SIN (正弦演算).....	3-256
COS (余弦演算).....	3-257
TAN (正接演算).....	3-258
ASIN (逆正弦演算).....	3-259
ACOS (逆余弦演算).....	3-260
ATAN (逆正接演算).....	3-261
ATAN2 (座標データ→角度ラジアン変換).....	3-262
SINH (双曲線正弦演算).....	3-263
COSH (双曲線余弦演算).....	3-264
TANH (双曲線正接演算).....	3-265
EXP (指数演算).....	3-266
LN (自然対数演算).....	3-267
LOG (常用対数演算).....	3-268
PWR (べき乗演算).....	3-269
SQR (平方根演算).....	3-270
RAD (角度→ラジアン変換).....	3-271
DEG (ラジアン→角度変換).....	3-272
FINT (浮動小数点形実数データ 小数点以下切り捨て).....	3-273
FRINT (浮動小数点形実数データ 小数点四捨五入).....	3-274

FNEG (浮動小数点形実数データ 符号変換).....	3-275
FABS (浮動小数点形実数データ 絶対値).....	3-276
FLT (整数→浮動小数点形実数データ変換).....	3-277
INT (浮動小数点形実数データ→整数変換) (超えない最大).....	3-279
FIX (浮動小数点形実数データ→整数 (小数点以下切り捨て)変換).....	3-282
ROFF (浮動小数点形実数データ→整数変換) (小数点以下四捨五入).....	3-285
HMSS (時間(時分秒)→秒データ変換).....	3-288
SHMS (秒データ→時間(時分秒)変換).....	3-289
CADD (時刻加算).....	3-290
CSUB (時刻減算).....	3-291
TMSEC (時刻データ→基準時刻からの秒データ).....	3-292
SECTM (基準時刻からの秒データ→時刻データ).....	3-293
TIMEWT (カレンダー設定).....	3-294
ERR (自己診断エラーコードセット).....	3-296
WDTRES (ウォッチドッグタイマーリセット).....	3-297
UNITSEL (通信ユニットスロットポート指定).....	3-298
GPSEND (汎用通信送信命令).....	3-300
GPRECV (汎用通信受信命令).....	3-306
SEND (MEWTOCOLマスタ / MODBUSマスタ).....	3-312
SEND (MODBUSマスタ:ファンクションコード指定).....	3-318
RECV (MEWTOCOLマスタ / MODBUSマスタ).....	3-324
RECV (MODBUSマスタ:ファンクションコード指定).....	3-330
PMSET (SCUのパラメータ変更).....	3-336
PMGET (SCUのパラメータ取得).....	3-339
RDET (ET-LANステータス読み出し).....	3-342
POSSET (位置決め開始テーブル設定).....	3-344
PSTRD (軸ステータス取得).....	3-346
PERRD (位置決めユニットエラー/ワーニング取得).....	3-348
UCLR (エラー/ワーニングクリア).....	3-349

## 4 プログラム上のご注意.....4-1

4-1 共通の注意事項.....	4-2
4-2 時刻、時間データ.....	4-3
4-3 文字列命令のデータテーブル.....	4-4
4-4 浮動小数点形実数演算.....	4-5

4-5 RUN中のタイマ・カウンタ設定値の変更.....	4-6
4-6 2重出力の使用について.....	4-9
4-7 立ち上がり検出方式について.....	4-11
4-8 プログラム記述上のご注意.....	4-16
<b>5 命令語一覧.....</b>	<b>5-1</b>
5-1 基本命令一覧.....	5-2
5-2 応用命令一覧.....	5-6



# 1

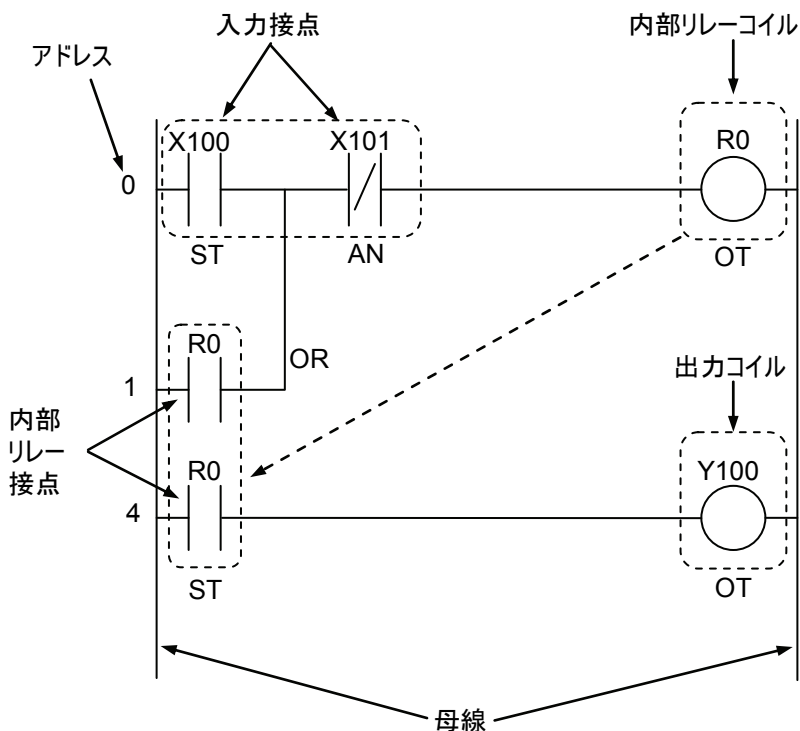
## 命令語概要

# 1-1 命令の構成

## 基本命令の構成パターン

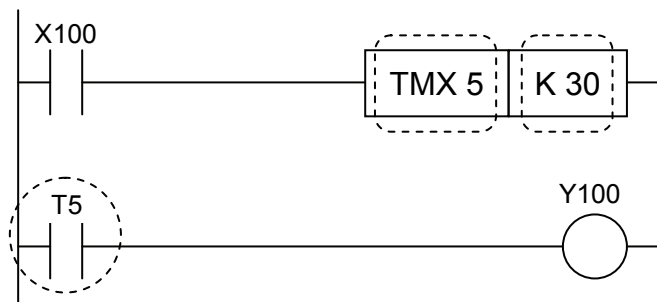
### ■ シーケンス基本命令

- リレーシーケンス回路をもとにした最も基本となる命令群で、ビット単位で論理演算をする命令です。下図のように、リレーのコイルと接点の組み合わせで表現されます。
- 命令によって指定できるリレーが異なります。各々の命令語の説明をご覧ください。



### ■ 基本機能命令

- タイマ、カウンタ、シフトレジスタなどの機能を実行する命令群です。



タイマ番号 No.5 の 0.1 秒タイマで、3.0 秒を設定している例です。X100 が ON のときに計時し、3.0 秒に達すると接点 T5 が ON になります。

## ■ 制御命令

- プログラムを実行する順序、流れを決める命令群です。条件によって、実行する部分を変更したり、必要な部分だけを実行することができます。
- 主な制御命令には、以下のようなものがあります。

ニモニック	名称	機能
MC-MCE	マスタコントロールリレー	実行条件が OFF のとき、MC～MCE 間の出力を OFF にします。
JP/LBL	ジャンプ・ラベル	実行条件が ON のとき、同じ番号を持つ LBL 命令へジャンプします。LBL 命令が JP 命令より後ろのアドレスにあるとき、JP～LBL 間のプログラムは実行されないため、実行時間が短縮します。
LOOP/LBL	ループ・ラベル	実行条件が ON のとき、指定した回数分 LBL～LOOP 間のプログラムを繰り返します。

## ■ ステップラダー命令

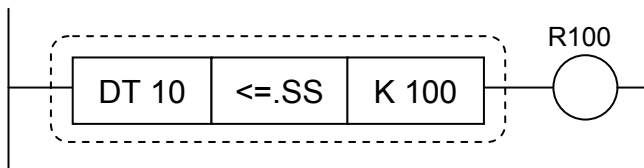
- SSTP～STPE で指定した区間を独立した「工程」として扱い、順序実行や分岐実行を行います。

## ■ サブルーチン命令

- 演算処理など繰り返し実行するプログラムをサブルーチン(SBL・RETで指定)として、必要な時に呼び出して実行します。

## ■ データ比較命令

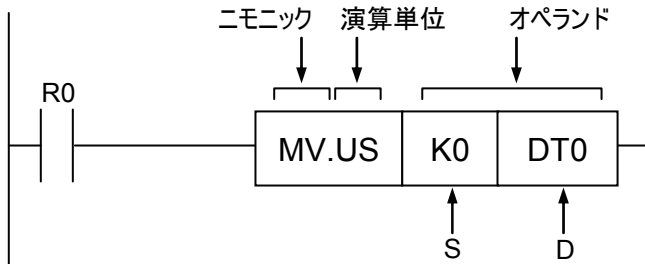
2つのデータを比較する命令群です。比較結果に応じて ON/OFF する接点として動作します。



# 応用命令の構成パターン

## ■ 応用命令の構成

- ニモニックは、演算を行なう内容(例えば、データの転送、算術演算など)を示します。
- 演算単位は、命令で演算する単位(US,SS、UL,SL、SF、DF など)を示します。
- オペランドは、演算の対象や演算処理のしかたを示します。[S]、[D]、[n]の3種類があります。命令語によって指定する必要があるオペランドの数、種類は異なります。



## ■ オペランドの種類

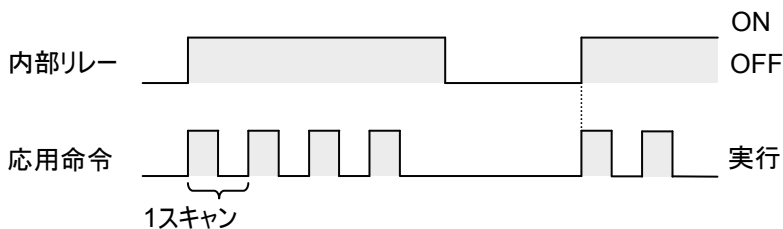
記号	略称		機能
S	Source	ソース	演算の対象となるデータや処理の方法を指定します。
D	Destination	デスティネーション	演算結果の格納先を指定します。
n	Number	ナンバー	演算の対象となるデータや処理の方法を指定する数値データです。

## ■ 毎スキャン実行型と微分実行型

- 応用命令には、毎スキャン実行型と微分実行型があります。

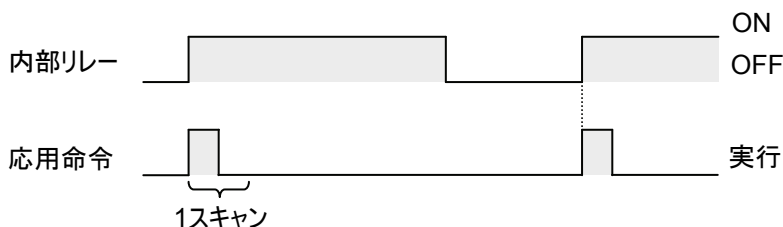
### (1) 毎スキャン実行型

内部リレーが成立している間、毎スキャン、繰り返して実行します。



### (2) 微分実行型

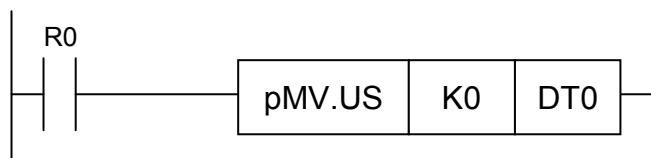
内部リレーの立ち上がりまたは立下りを検出して、1スキャンのみ実行します。





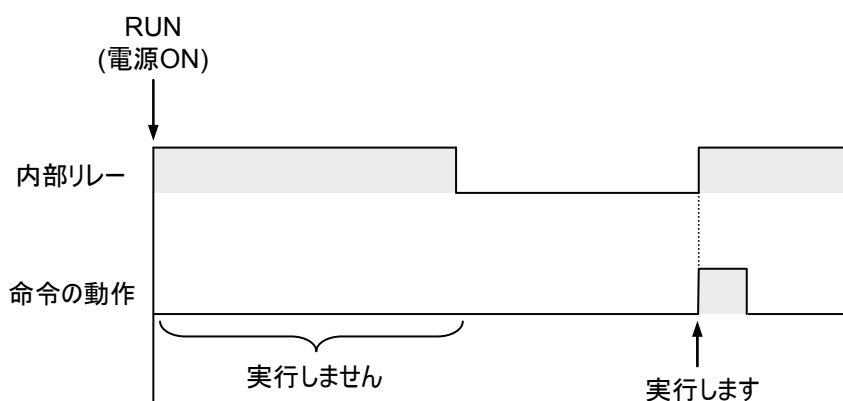
## ■ 微分実行型命令の入力

- ツールソフトウェア FPWINGR で微分実行型命令を入力するには、[Shift]キー+[F6]キーを押してから、命令リストダイアログボックスで任意の命令を選択します。
- 微分実行型命令の先頭には、(P)が付与されます。



## ■ 微分実行型命令の動作

- 微分実行型命令の実行条件となっている内部リレーが ON しつづけている間は、立ち上がり時のみ命令を実行して、以後は実行しません。
- RUN モードに切り替えた時や RUN モードで電源を投入した時に微分実行型命令の内部リレーが最初から成立している場合には最初のスキャンでは命令を実行しません。



- 微分実行型命令(P 命令)を MC~MCE 命令、JP~LBL 命令など命令を実行する順序を変える命令(下記①~⑥)と合わせて使用する場合、命令の実行とカウント入力のタイミングにより、命令の動作が変わりますのでご注意ください。

- ①MC~MCE 命令
- ②JP~LBL 命令
- ③LOOP~LBL 命令
- ④CNDE 命令
- ⑤ステップラダー命令
- ⑥サブルーチン命令



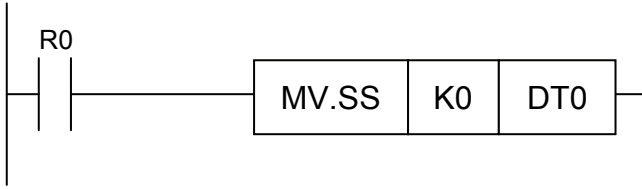
### ◆ 参照

- 詳細については、4-7立ち上がり検出方式についてご覧ください。
- 微分実行型応用命令を、アンドスタック命令やポップスタック命令と組み合わせるときは、正しくない記述にならないようご注意ください。詳しくは、4-8プログラム記述上のご注意をご覧ください。

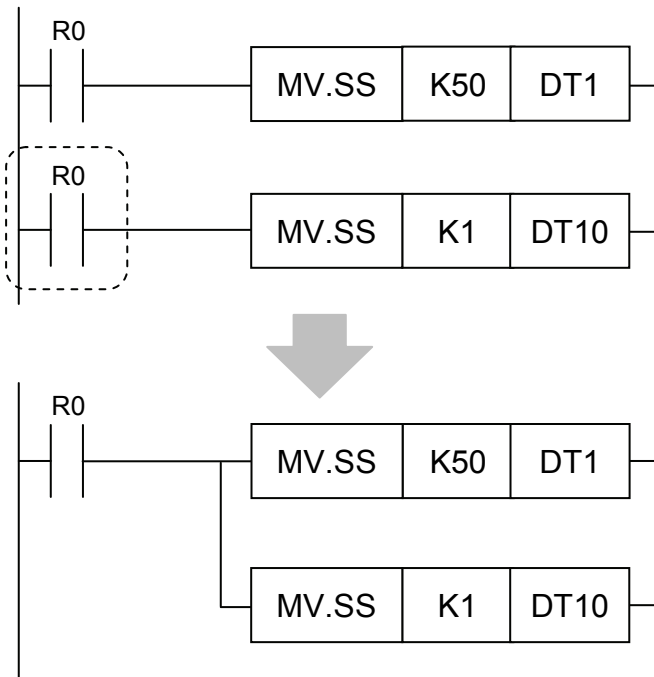
## ■ 応用命令と内部リレー

- 応用命令は、必ず内部リレーとペアにして使います。
- 内部リレーに指定したリレーシーケンス回路の演算結果が ON の時、応用命令は実行されます。

例) 内部リレーR0 が ON のとき、MV 命令が実行され、DT0 に K0 が転送されます。



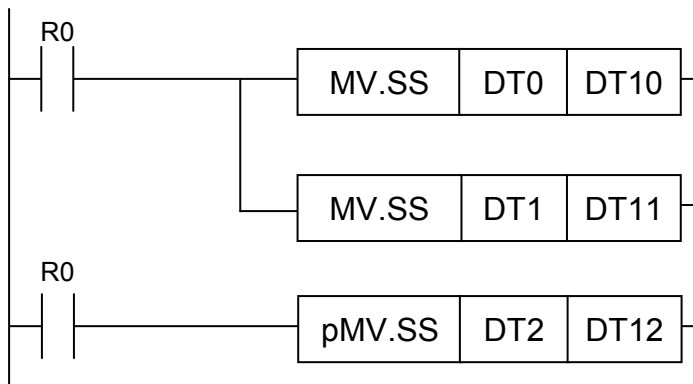
- 応用命令を連続して使う場合、内部リレーが同じ場合は、2 つ目から内部リレーを省略することができます。



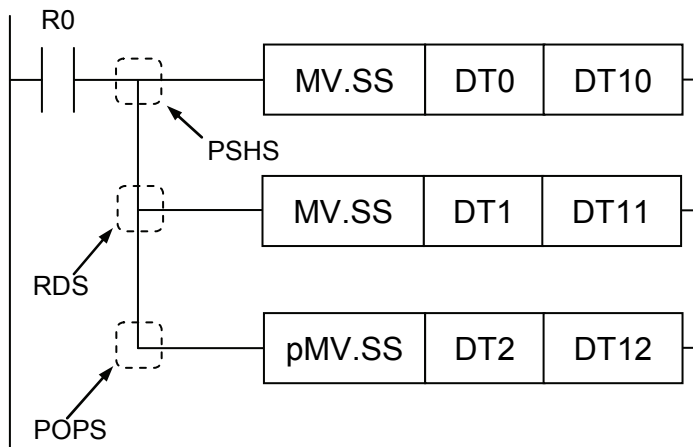
## ■ 内部リレーを省略する場合のご注意

- 一つの内部リレーに対して、毎スキャン実行型命令と微分型実行命令が混在しているときは、次のようにプログラミングしてください。

例 1) 毎スキャン実行型と微分実行型、それぞれに分けて記述する



例 2) PUSH・RDS・POPS 命令を使用する



# 1-2 演算単位

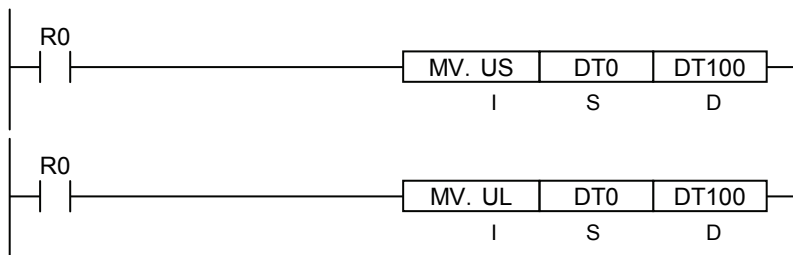
## ■ 演算単位とは

- 各命令で演算を行なう基準とする単位を指定します。
- 指定する演算単位により、同じ命令でも、演算結果が異なる結果が得られることがあります。また、演算の対象となるオペランドの範囲や演算結果を格納するワード数に変動します。
- 命令により、指定できる演算単位が異なります。また、演算単位を指定しない命令もあります。各命令語の説明をご参照ください。記号“i”で表示しています。

例 1) 転送命令で演算単位が異なる場合

演算単位 US (符号なし 16 ビットデータ) のとき、DT0 の値を DT100 に転送します。

演算単位 UL (符号なし 32 ビットデータ) のとき、DT0-DT1 の値を DT100-DT101 に転送します。

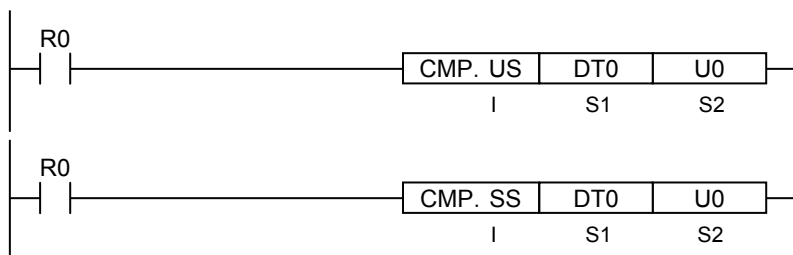


例 2) 比較命令で演算単位が異なる場合

DT0 (S1) にバイナリデータ“1111 1111 1111 1111”が格納されているとき

演算単位 US (符号なし 16 ビットデータ) の場合、(S1) の値は正の数“65535”として扱われ、(S1) > (S2) となり、比較フラグ SRA が ON になります。

演算単位 SS (符号付き 16 ビットデータ) の場合、(S1) の値は負の数“-1”として扱われ、(S1) < (S2) となり、比較フラグ SRC が ON になります。



## ■ 演算単位の種類

記号	名称	指定できる範囲
bit	1ビットデータ	0,1
US	符号なし16ビットデータ	0~65535
SS	符号付き16ビットデータ	-32768~+32767
UL	符号なし32ビットデータ	0~4294967295
SL	符号付き32ビットデータ	-2147483548~+2147483547
SF	単精度浮動小数点形実数データ	-1.175494E-38~-3.402823E+38 0 +1.175494E-38~+3.402823E+38
DF	倍精度浮動小数点形実数データ	-2.2250738585072014E-308~-1.7976931348623158E+308 0 +2.2250738585072014E-308~+1.7976931348623158E+308

■ 演算単位と指定できる定数(●:指定可能)

演算単位			K定数		U定数		H定数		実数		文字
名称	サイズ	符号	16bit	32bit	16bit	32bit	16bit	32bit	SF	DF	“”
US	16bit	なし	—	—	●	—	●	—	—	—	—
SS		あり	●	—	—	—	●	—	—	—	—
UL	32bit	なし	—	—	—	●	—	●	—	—	—
SL		あり	—	●	—	—	—	●	—	—	—
SF	32bit	あり	—	—	—	—	—	—	●	—	—
DF	64bit	あり	—	—	—	—	—	—	—	●	—

# 1-3 演算用デバイス一覧

## ■ 演算用デバイス一覧

ビット数	シンボル		名称	点数	範囲	説明
	グローバル	ローカル				
1	X	_X	外部入力	8192 点	0～511F	外部からの入力です ON/OFF します
	Y	_Y	外部出力	8192 点	0～511F	外部に ON/OFF 状態を出力します
	R	_R	内部リレー	32768 点	0～2047F	プログラム上でのみ ON/OFF するリレーです
	L	_L	リンクリレー	16384 点	0～1023F	PLC リンク時に共有して使用するリレーです
	T	_T	タイマ	4096 点	0～4095	タイマが設定時間に達すると ON になります。
	C	_C	カウンタ	1024 点	0～1023	カウンタが設定値に達すると ON になります。
	SR	-	システムリレー	約 1120 点		特定の条件で ON/OFF し、フラグ等として使用するリレーです。
	P	-	パルスリレー	4096 点	0～255F	実行条件の立ち上がり 1 スキャンのみ、ON するリレーです。
	E	-	異常報知リレー	4096 点	0～4095	ユーザーが任意に割り付けた異常条件をメモリに記憶させておくためのリレーです。
	IN	-	直接入力	最大 1008 点	0～62F	通常の I/O リフレッシュによらず、演算処理中に入出力処理を行なうためのリレーです。
OT	-	直接出力	最大 1008 点	0～62F		
16	WX	_WX	外部入力	512 ワード	0～511	外部入力 16 点分をまとめて、16 ビット 1 ワードのデータとして指定する時の記号です。
	WY	_WY	外部出力	512 ワード	0～511	外部出力 16 点分をまとめて、16 ビット 1 ワードのデータとして指定する時の記号です。
	WR	_WR	内部リレー	2048 ワード	0～2047	内部リレー 16 点分をまとめて、16 ビット 1 ワードのデータとして指定する時の記号です。
	WL	_WL	リンクリレー	1024 ワード	0～1023	リンクリレー 16 点分をまとめて、16 ビット 1 ワードのデータとして指定する時の記号です。
	DT	_DT	データレジスタ	最大 1M ワード	0～999423	プログラム上で使用するデータメモリです
	LD	_LD	リンクレジスタ	16384 ワード	0～16383	PLC リンク時に共有して使用するデータメモリです
	UM	-	ユニットメモリ	最大 512kw/1 ユニット	0～524287	高機能ユニットのユニットメモリにアクセスするためのデバイスです。ユニットにより大きさが変わり、あらかじめ割り付けられています。
	SD	-	システムデータ	約 80 ワード	—	特定の内容を格納するデータメモリです。各種の設定やエラーコードが格納されています。
	WI	-	直接入力	最大 62 ワード	0～62	ユニットの入力をアクセスします
WO	-	直接出力	最大 62 ワード	0～62	ユニットの出力をアクセスします	
32	TS	_TS	タイマ設定値	4096 ダブルワード	0～4095	タイマの目標値を格納するデータメモリです。タイマ番号に対応しています。
	TE	_TE	タイマ経過値	4096 ダブルワード	0～4095	タイマの経過値を格納するデータメモリです。タイマ番号に対応しています。
	CS	_CS	カウンタ設定値	1024 ダブルワード	0～1023	カウンタの目標値を格納するデータメモリです。カウンタの番号に対応します
	CE	_CE	カウンタ経過値	1024 ダブルワード	0～1023	カウンタの経過値を格納するデータメモリです。カウンタの番号に対応します
	I0～IE	-	インデックス修飾レジスタ	15 ダブルワード	I0～IE	メモリエリアのアドレス、定数の修飾用レジスタです。

- (注 1):表中の値は、プログラム上で使用できるデバイス点数を示します。実際に入出力として使用できる点数は、構成により異なります。  
 (注 2):演算用デバイスには、停電時または PROG.モードへの切り替え時に直前の状態を記憶する保持型とリセットされる非保持型があります。非保持型の領域は、電源 ON 時または PROG/RUN の切り替え時にゼロクリアします

演算用デバイスの種類	非保持型/保持型の設定
内部リレー (R)、データレジスタ (DT)、リンクリレー (L)、リンクレジスタ (LD)	ツールソフトウェアにより、非保持型または保持型デバイスとして設定可能
カウンタ (C)、カウンタ設定値 (CS)、カウンタ設定値 (CE)、異常報知リレー (E)	保持型
入力 (X)、出力 (Y)、タイマ (T)、タイマ設定値 (TS)、タイマ設定値 (TE)、パルスリレー (P)、直接入力 (IN)、直接出力 (OT)、インデックスレジスタ (I)、ユニットメモリ (UM)、システムデータレジスタ (SD)	非保持型

(注 3):直接入力 (IN)、直接出力 (OT)、ユニットメモリ (UM) の各デバイスは、制御対象となるユニットの-slot No.とメモリアドレスを命令で指定することにより使用します。

(注 4):使用できるデータレジスタ (DT) の数は、CPU ユニットの種類およびメモリコンフィグレーションの設定により変動します。

ユニットの種類	メモリの種類	メモリ選択パターン				
		1	2	3	4	5
CPS4E	データレジスタ容量(ワード)	65,536	131,072	262,144	524,288	999,424
CPS3E CPS3	データレジスタ容量(ワード)	131,072	262,144	425,985	589,824	

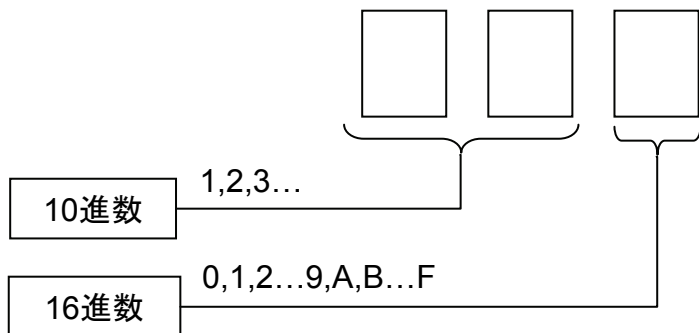
## ■ 定数一覧

シンボル	名称	演算ビット数	範囲
K	符号付き 10 進定数	16	-32768~+32767
		32	-2147483548~+2147483547
U	符号なし 10 進定数	16	0~65535
		32	0~4294967295
H	16 進定数	16	0~FFFF
		32	0~FFFF FFFF
SF	単精度浮動小数点形実数定数	32	-1.175494E-38~-3.402823E+38 0 +1.175494E-38~+3.402823E+38
DF	倍精度浮動小数点形実数定数	64	-2.2250738585072014E-308~-1.7976931348623158E+308 0 +2.2250738585072014E-308~+1.7976931348623158E+308
“”	文字定数	8	最大 256 文字

# 1-4 デバイス番号の指定

## ■ 外部入力(X)、外部出力(Y)、内部リレー(R)、リンクリレー(L)、パルスリレー(P)、システムリレー(SR)、直接入力(IN)、直接出力(OT)の場合

リレーは 16 点単位で扱うこともあるため、番号は 10 進数と 16 進数を組み合わせで指定します。



<例> 外部入力Xの場合

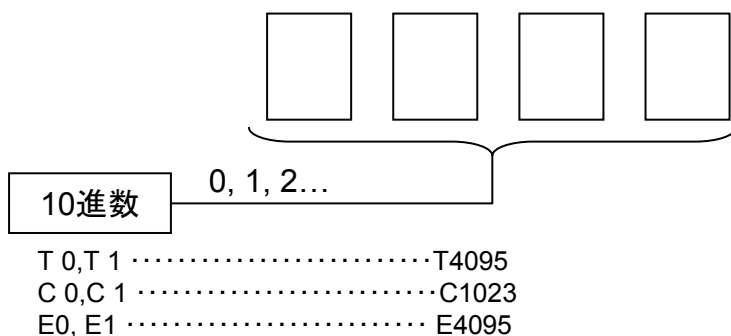
X	0, X	1	.....	X	F
X	10, X	11	.....	X	1F
X	20, X	21	.....	X	2F
	⋮				⋮

## ■ 外部入出力のリレー番号について

- 外部入力(X)で使用できるのは、実際に入力接点に割り付けられているもののみです。
- 外部出力(Y)で出力できるのは、実際に出力接点に割り付けられているもののみです。割り付けられていない Y は、内部リレーとして使用できます。
- 番号の割り付けは、ユニットの組み合わせに従って決まります。

## ■ タイマ接点(T)、カウンタ接点(C)、異常報知リレー(E)の場合

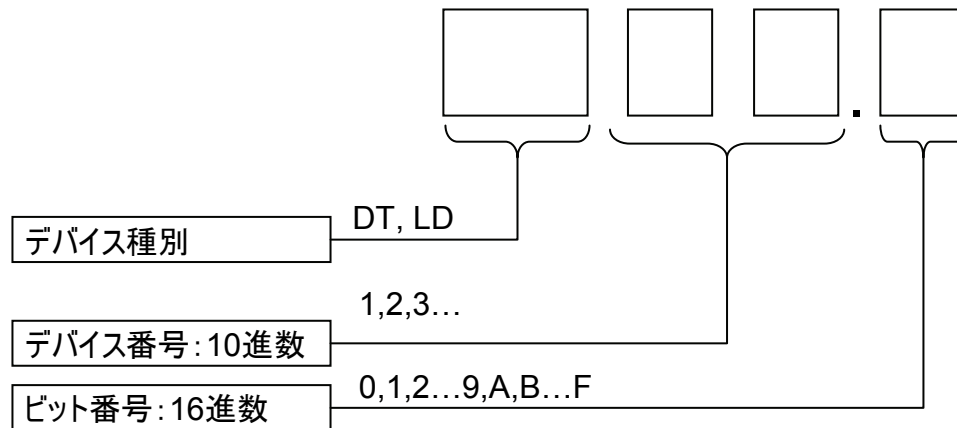
- タイマ/カウンタ接点の番号は、タイマ/カウンタの番号に対応しており、10 進数で指定します。
- 異常報知リレー番号は、10 進数で指定します。





## ■ ワードデバイスのビット指定 (DT\*.n, LD\*.n) の場合

- データレジスタ DT とリンクレジスタ LD は、デバイス種別、デバイス番号、ビット番号を指定することにより、16ビットデータの特定のビットを抽出し、ビットデータとして使用することができます。
- デバイス番号とビット番号の間には、ピリオド(.)で区切ります。
- 使用できる命令語については、各命令語の説明をご参照ください。



<例>ワードデバイス DT\*.n の場合

DT0.0, DT0.1, ----- DT0.F,  
DT10.0, DT10.1, ---- DT10.F  
DT20.0, DT20.1, ---- DT20.F  
⋮

## ■ メモリエリア番号の数え方

データレジスタ DT など各メモリエリアの番号は、10進数で指定します。(インデックスレジスタを除く。)

<例>データレジスタ DT の場合

DT0, DT1, -----DT9  
DT10, DT11, -----DT19  
⋮

# 1-5 リレーの説明

## X 外部入力

### ■ 外部入力 (X) のはたらき

- 外部入力は、入りに接続したリミットスイッチや光電スイッチなどの外部入力機器から送られる信号を取り込むためのリレーです。

### ■ 使用上の制限について

- 割り付けられていない外部入力は使用できません。
- 演算(プログラム)で外部入力の ON/OFF 状態を変化させることはできません。
- プログラム上での使用回数に制限はありません。

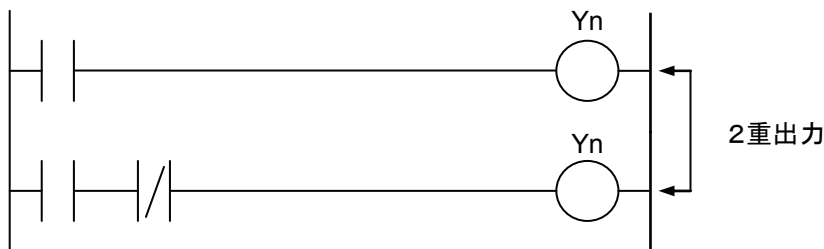
## Y 外部出力

### ■ 外部出力 (Y) のはたらき

- 外部出力は、出力に接続したソレノイドや表示器などの負荷に、プログラム演算の結果を信号として送るためのリレーです。外部出力の ON/OFF 状態が制御信号として出力されます。

### ■ 使用上の制限について

- 割り付けられていない外部出力は内部リレーとして使用できます。(ただし、保持型に指定することはできません)
- 接点として使用するときは、使用回数に制限はありません。
- 演算結果の出力先として指定する場合(OT 命令、KP 命令)は、原則として1プログラム中に1回のみ使用できます。(2重出力の禁止)
- ツールソフトウェア FPWIN GR7 の“CPU コンフィグレーション”ダイアログボックスの“動作選択”の“2重出力の許可”の設定で、2重出力を許可することもできます。
- SET 命令、RST 命令での使用は、2重出力になりません。



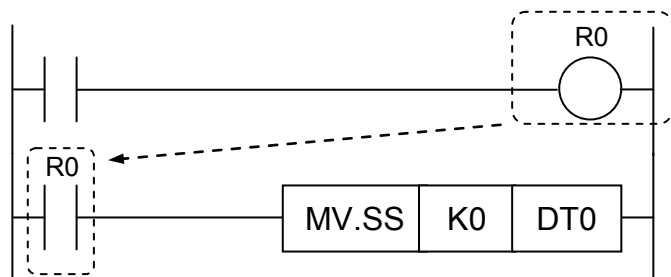
### ◆ ご注意！

- ツールソフトウェア FPWINGR7 上で、プロジェクトのトータルチェックを実行しても、先頭に使用されている命令は表示されません。2重使用となっている2番目以降の出力を表示します。

# R 内部リレー

## ■ 内部リレー (R) のはたらき

- 内部リレーは、プログラム上でのみ動作するリレーです。ON/OFF 状態は、外部に出力されません。
- 演算結果が出力されて ON になると(コイル:ON)、接点として使用している同じリレーが ON になります。



## ■ 非保持型リレーと保持型リレー

- 内部リレーには、電源を切ったり、RUN から PROG.モードに切り替えたときに、
  - 1) 停止する直前の ON/OFF 状態を記憶し、復帰後、その状態で運転を再開させることができる「保持型リレー」
  - 2) 停止するときリセットされる「非保持型リレー」の 2 種類があります。
- ツールソフトウェア FPWIN GR7 の“メモリコンフィグレーション”ダイアログボックスの“ローカルデバイスの使用数設定(全体)”の“グローバル保持開始番号”で、保持型にするものと非保持型にするものを指定できます。
- 保持型リレーの先頭をワード番号で指定すると、それより前が非保持型に、それ以降が保持型になります。
- 保持型/非保持型の設定は、グローバルデバイス/ローカルデバイス各々について設定できます。

“メモリコンフィグレーション”の  
“ローカルデバイスの使用数設定(全体)”の  
“グローバル保持開始番号”

非保持型  
-----  
保持型

## ■ 使用上の制限について

- 接点として使用するとき、使用回数に制限はありません。
- 演算結果の出力先として指定する場合(OT 命令、KP 命令)は、原則として 1 プログラム中に 1 回のみ使用できます。(2 重出力の禁止)
- “CPU コンフィグレーション”の“動作選択”の“2 重出力の許可”の設定で、2 重出力を許可することもできます。また、SET 命令、RST 命令での使用は、2 重出力になりません。

# SR システムリレー

## ■ システムリレー (SR) のはたらき

- システムリレーは、特定の条件で ON/OFF するリレーです。
- ON/OFF 状態は、外部に出力されません。プログラム上でのみ動作します。

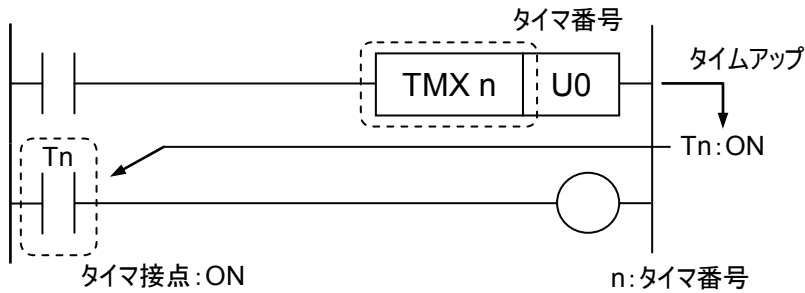
## ■ システムリレー (SR) の種類

分類	はたらき	例
動作状態フラグ	ON/OFFで動作状態を知らせます。	運転(RUN)中(SR20)、強制入出力中(SR29)、比較命令の結果(SRA~SRC)
エラーフラグ	エラー発生時にONになり、異常を知らせます。	演算エラー(SR7,SR8)
特殊な条件でON/OFFするリレー	プログラム上で必要な条件を選んで利用できます。	常時ONリレー(SR10)、1スキャンごとにON/OFF(SR12) イニシャルパルスリレー(SR13~SR14)、クロックパルス(SR18~SR1E)

# T タイマ

## ■ タイマ(T)のはたらき

- タイマが動作して、設定した時間に達すると、タイマと同じ番号のタイマ接点が ON します。
- タイマがタイムアップした状態のときに、タイマの実行条件が OFF になると、タイマ接点は OFF になります。タイマは、すべて非保持型です。



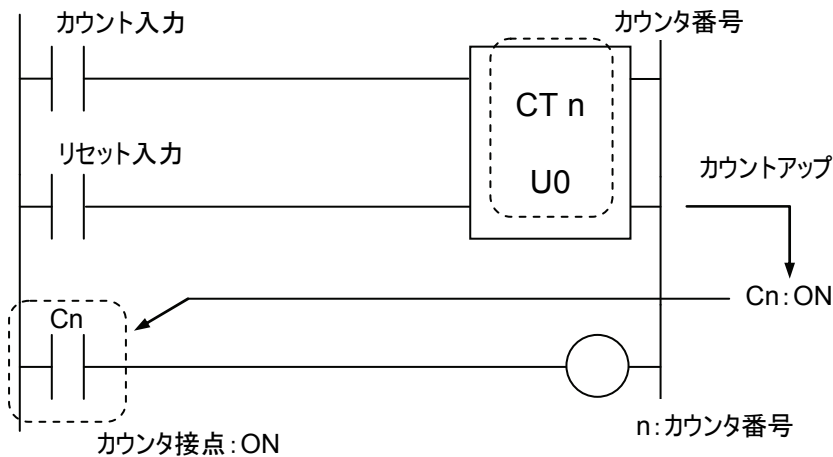
## ■ 使用上の制限について

- 接点として使用する時は、使用回数に制限はありません。

# C カウンタ

## ■ カウンタ(C)のはたらき

- 減算式プリセットカウンタが動作して、0 に達すると、カウンタと同じ番号のカウンタ接点が ON します。
- カウンタのリセット入力を ON にすると、カウンタ接点は OFF になります。
- カウンタは、すべて保持型です。



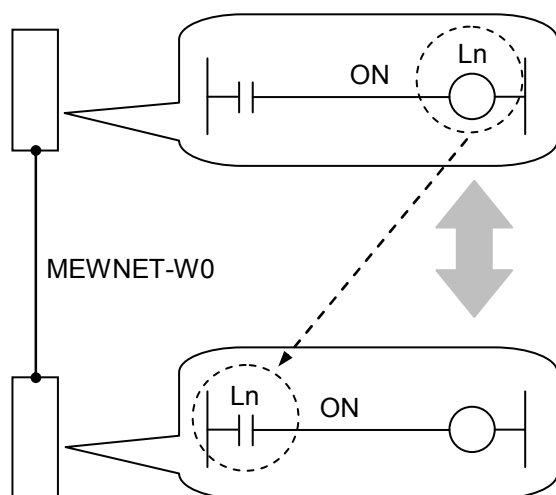
## ■ 使用上の制限について

- 接点として使用する時は、使用回数に制限はありません。

# L リンクリレー

## ■ リンクリレー(L)のはたらき

- リンクリレーは、複数のコントローラをネットワークで接続しているときに、その間で共有して使用する「PLC リンク」用のリレーです。
- ある PLC でリンクリレー(コイル)に演算結果を出力すると、ネットワークに接続されている別の PLC に送信され、同じ番号のリンクリレー(接点)にその演算結果が反映されます。
- リンクリレーを使用すると、このように、PLC 間でビット情報をやり取りすることができます。



## ■ リンクリレーの使用範囲

- リンクリレーは、ネットワークの種類やユニットの組み合わせによって、使用できる範囲が異なります。
- 使用する範囲や点数は、ネットワークごとに設定する必要があります。

## ■ 保持型リレーと非保持型リレーの設定

- リンクリレーには、電源を切ったり、RUN モードから PROG.モードに切り替えて停止したときに、
  - ① 停止から運転再開までの間、停止する直前の ON/OFF 状態を保持する「保持型リレー」
  - ② 停止時にリセットされる「非保持型リレー」の 2 種類があります。
- ツールソフトウェア FPWIN GR7 の“メモリコンフィグレーションダイアログボックスの“ローカルデバイスの使用数設定(全体)”の“グローバル保持開始番号”で(非保持型、保持型の範囲を指定できます。
- 保持型リレーの先頭をワード番号で指定すると、それ以前が非保持型に、それ以降が保持型になります。たとえば、“メモリコンフィグレーション”の“ローカルデバイスの使用数設定(全体)”の“グローバル保持開始番号”を 10 にすると、L0～L9F が非保持型、L100～L63F が保持型になります。
- 受信用リンクリレーとして使用する場合は、システムコンフィグレーションで保持型に指定していても、保持動作は行われませんので、ご注意ください。
- 保持型/非保持型の設定は、グローバルデバイス/ローカルデバイス各々について設定することができます。

## ■ 使用上の制限について

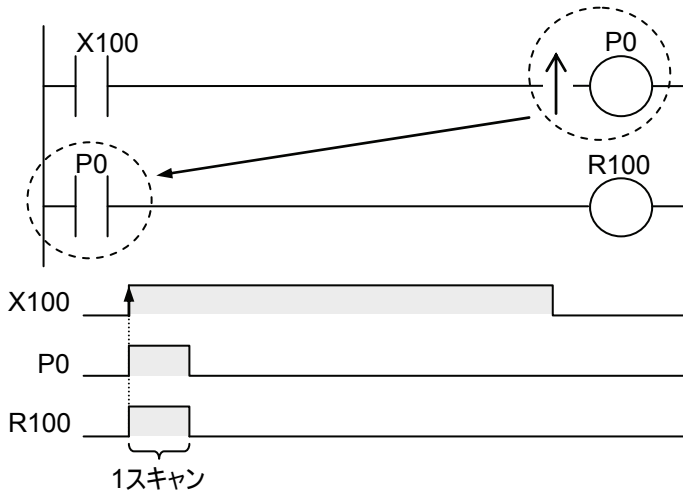
- 接点として使用するときは、使用回数に制限はありません。
- 演算結果の出力先として指定する場合(OT 命令、KP 命令)は、原則として 1 プログラム中に 1 回のみ使用できます。(2 重出力の禁止)
- “CPU コンフィグレーション”の“動作選択”の“2 重出力の許可”の設定で、2 重出力を許可することもできます。また、SET 命令、RST 命令での使用は、2 重出力になりません。

# P パルスリレー

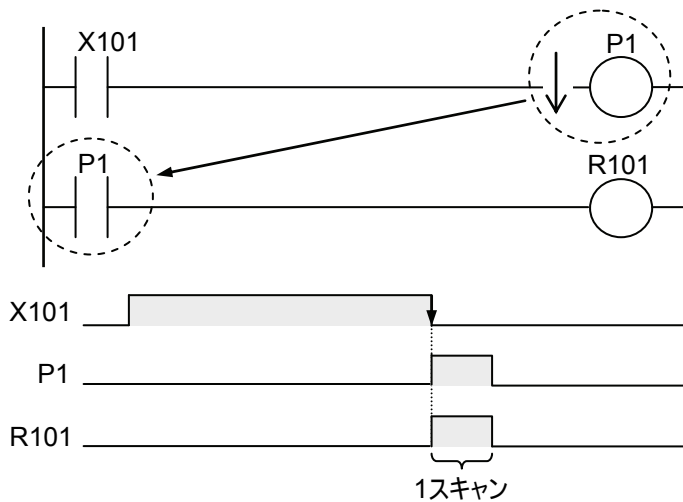
## ■ パルスリレー(P)のはたらき

- パルスリレーは、1 スキャンのみ ON するリレーです。ON/OFF 状態は、外部に出力されません。プログラム上でのみ動作します。
- 立ち上がり検出アウト命令(OT↑)または立ち下がり検出アウト命令(OT↓)を実行することによってのみ ON します。
- パルスリレーを実行条件に使用すると、立ち上がりまたは立ち下がりを検出した 1 スキャンのみ動作します。

<例 1> 入力 X100 の立ち上がりで微分実行する。



<例 2> 入力 X101 の立ち下がりで微分実行する。



## ■ 使用上の制限について

- パルスリレーは、電源を切るとクリアされます。
- OT↑ 命令または OT↓ 命令の出力先として指定する場合は、原則として、1 プログラム中に 1 回のみ使用できます。(2 重出力の禁止)
- 接点として使用するときには、使用回数に制限はありません。
- OT 命令、KP 命令、SET 命令、RST 命令、ALT 命令の出力先として、パルスリレーを指定することはできません。
- 応用命令の格納先として、ワード単位のパルスリレー(WP)を指定することはできません。

# E 異常報知リレー

## ■ 異常報知リレー (E) のはたらき

- 異常報知リレーは、ユーザが任意に割り付けた異常条件を内部リレーに反映させ、メモリに記憶しておくためのリレーです。
- 異常報知リレーは、ユーザプログラムの SET/RST 命令により ON/OFF させます。
- 異常報知リレーが ON になると、CPU ユニット内のメモリエリアに、ON になっている異常報知リレーの個数、リレー番号、および最初に ON になった時のカレンダータイマのデータが格納されます。

システムデータレジスタ SD 番号	内容	
SD50, SD51	年/月データ	最初にリレーが ON になったときのカレンダータイマのデータ
SD52, SD53	日/時データ	
SD54, SD55	分/秒データ	
SD60	ON になっているリレーの個数	
SD61~SD79	ON になっているリレー番号	

- 異常報知リレーは、最大 500 個分までメモリエリアに記憶されます。ただし、ユーザがモニタまたは操作できるのは、SD61~SD79 の範囲のみです。

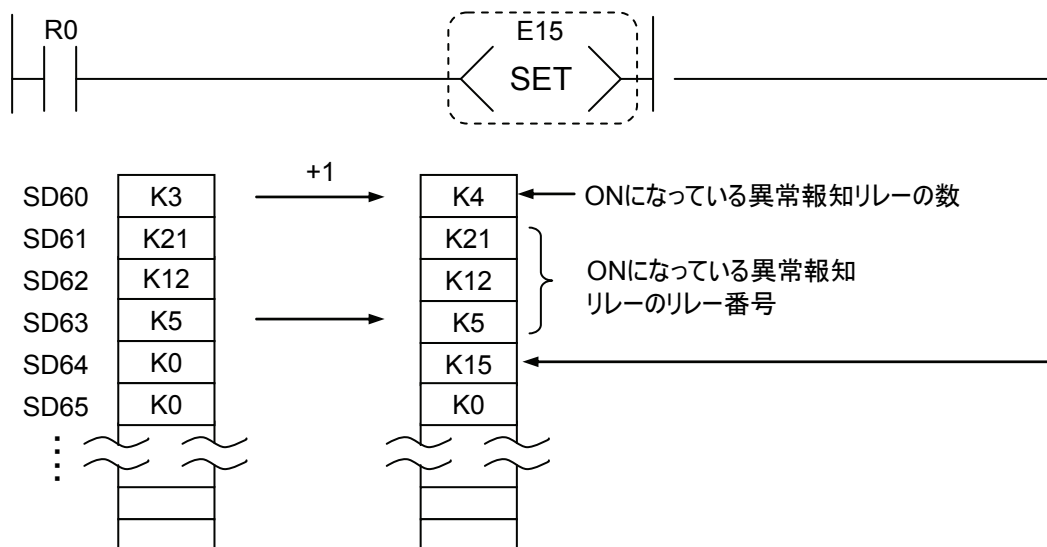
## ■ 使用上の制限

- 異常報知リレー E を OT 命令、KP 命令、ALT 命令の出力先に指定することはできません。
- 異常報知リレー E をプログラム上の複数個所で SET/RST 命令により ON/OFF させることができます。ただし、2 重使用のチェックは行われません。

## ■ 異常報知リレーのセットプログラム

- 異常報知条件で異常報知リレーを ON させるためには、下記のように SET 命令を使用してください。
- 異常報知リレーは、異常条件が OFF になっても保持されます。

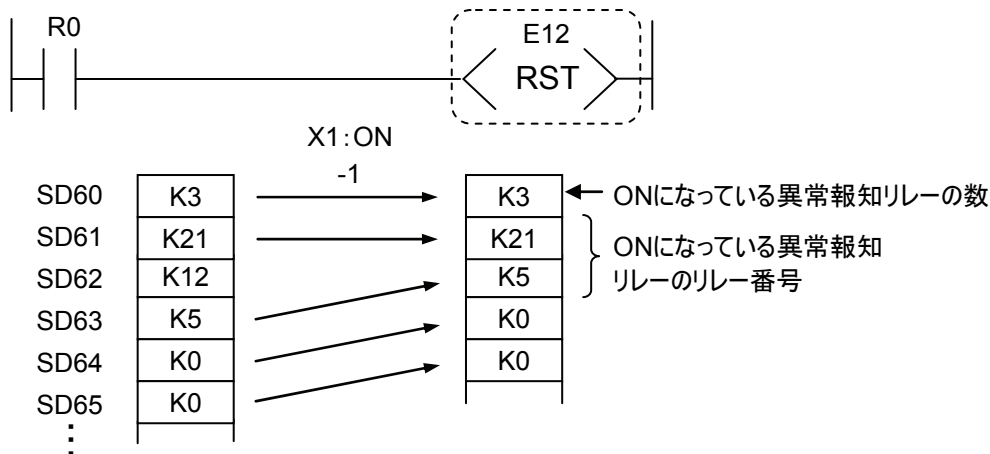
<例> 異常が発生すると R0 が ON になる場合



### ■ 任意の異常報知リレーのリセットプログラム

- 異常が修復された場合に、異常報知リレーを OFF するには、RST 命令を使用してください。

<例> 異常が修復すると R0 が ON になる場合



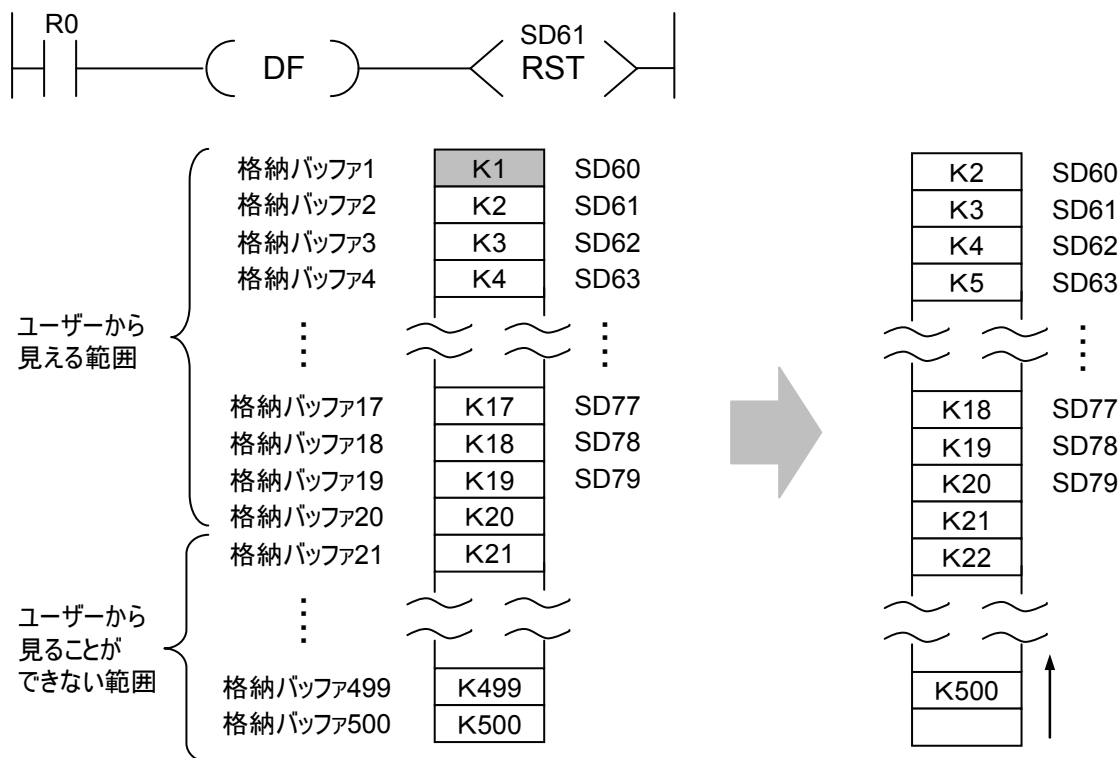
### ■ 全バッファエリアのクリア

- 異常報知リレーをすべてリセットするには、次ページの方法と同様に RST 命令を使って、システムデータレジスタ SD60 を指定してください。

### ■ バッファエリア、先頭のクリア

- リレー番号格納エリアのうち、SD60 と SD61 のみ、RST 命令にてシステムデータレジスタを直接指定してクリアすることができます。
- SD60 を指定した場合、バッファ内のエラー情報が全てクリアされ、SD61 を指定した時はバッファエリア先頭のリレー番号がクリアされ、下記の例のようにバッファが詰まります。

<例> RST 命令にて SD61 の内容を削除した時





## IN 直接入力

### ■ 直接入力 (IN) のはたらき

- 外部入力(X)は入出力リフレッシュ実行時に一括して読み出し更新しますが、直接入力(IN)を使用すると、演算実行時に外部入力を読み出し更新します。
- 高速応答を要求される制御に有効です。
- プログラム上では、スロット番号 S1、S2……と合わせて指定します。

## OT 直接出力

### ■ 直接出力 (OT) のはたらき

- 外部出力(Y)は入出力リフレッシュ実行時に一括して出力しますが、直接出力(OT)を使用すると、演算実行時に外部出力にそれまでの演算結果(ON/OFF)を出力します。
- 高速応答を要求される制御に有効です。
- プログラム上では、スロット番号 S1、S2……と合わせて指定します。



# 1-6 メモリエリアの説明

## DT データレジスタ

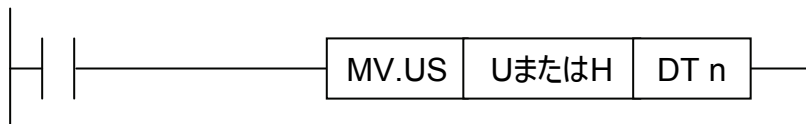
### ■ データレジスタ (DT) のはたらき

- データレジスタは、ワード(16ビット)単位で扱うメモリエリアで、16ビットで構成される数値データ等を格納します。

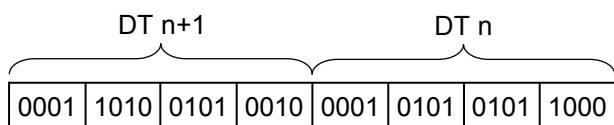
DTn 

0	0	0	1	1	0	1	0	0	1	0	1	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

[DTnに数値を書き込むプログラム例]



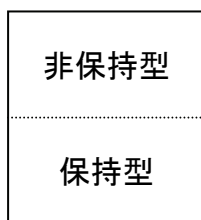
- 32ビットデータを扱う場合は、2個のデータレジスタをひとつのダブルワードのデータレジスタとして扱います。プログラム上では、下位16ビットのデータレジスタの番号を指定します。



### ■ 非保持型データと保持型データ

- データレジスタには、電源を切ったり、RUN から PROG.モードに切り替えたときに、
  - ①停止する直前の ON/OFF 状態を記憶し、復帰後、その状態で運転を再開させることができる「保持型データレジスタ」
  - ②停止するときにリセットされる「非保持型データレジスタ」の2種類があります。
- 保持型/非保持型の設定は、グローバルデバイス/ローカルデバイス各々について設定できます。
- ツールソフトウェア FPWIN GR7 の“メモリコンフィグレーション”ダイアログボックスの“ローカルデバイスの使用数設定(全体)”の“グローバル保持開始番号”で、保持型にするものと非保持型にするものを指定できます。保持型データレジスタの先頭をワード番号で指定すると、それより前が非保持型に、それ以降が保持型になります。

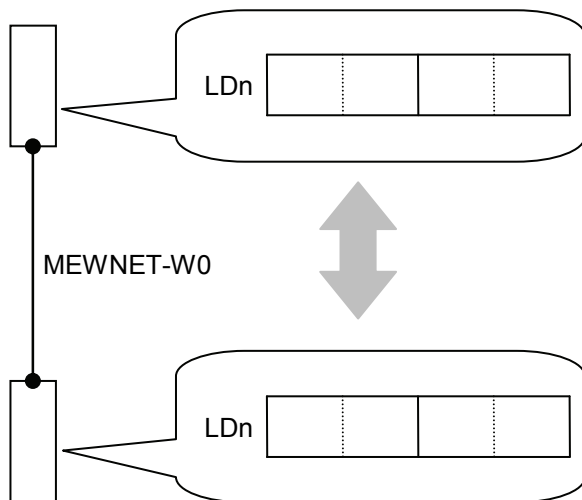
“メモリコンフィグレーション”の  
“ローカルデバイスの使用数設定(全体)”の  
“グローバル保持開始番号”



# LD リンクレジスタ

## ■ リンクレジスタ(LD)のはたらき

- リンクレジスタは、コントロールユニットをネットワークで接続しているときに、その間で共有して使用する「PLC リンク」用のデータメモリです。
- ある PLC でリンクレジスタにデータを書き込むと、ネットワークで接続されている別の PLC で、同じ番号のリンクレジスタにその内容が格納されます。
- リンクレジスタを使用すると、このように、データを書き込むだけで PLC 間でデータをやりとりすることができます。



## ■ リンクレジスタの使用範囲

- リンクレジスタは、ネットワークの種類やユニットの組み合わせによって使用できる範囲が異なります。
- 使用する範囲や点数は、ネットワークごとに設定する必要があります。

## ■ 保持型レジスタと非保持型レジスタの設定

- リンクレジスタには、電源を切ったり、RUN モードから PROG モードに切り替えて停止したときに、
  - ①停止から運転開始までの間、停止する直前の内容を保持する「保持型レジスタ」
  - ②停止時にリセットされる「非保持型レジスタ」の 2 種類があります。
- “メモリコンフィグレーション”の“ローカルデバイスの使用数設定(全体)”の“グローバル保持開始番号”で非保持型、保持型の範囲を指定できます。
- 受信リンクレジスタとして使用する場合は、コンフィグレーションで保持型に指定していても、保持動作は行われませんので、ご注意ください。
- 保持型／非保持型の設定は、グローバルデバイス／ローカルデバイス各々について設定できます。

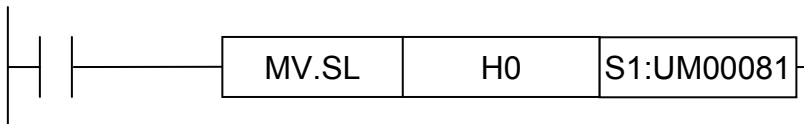
# UM ユニットメモリ

## ■ ユニットメモリ(UM)のはたらき

- CPU ユニット／高機能ユニット間でデータの受け渡しを行うときに使用します。
- ユニットメモリのデータ数や割り付けは、ユニットの種類により異なります。
- ユニットメモリのアドレスは、5桁の16進数で指定します。(H0～H7FFFF(最大))
- プログラム上では、スロット番号 S1、S2……と合わせて指定します。

## ■ ユニットメモリ(UM)の使用例

スロット No:1 のユニットメモリ(UM)のアドレス:H 00081 に定数 H0 を転送する場合。



# SD システムデータレジスタ

## ■ システムデータレジスタ (SD) のはたらき

- システムデータレジスタは、特定の内容が格納されるメモリエリアです。
- 呼び出し専用のエリアと書き込みもできるエリア、システムで使用しているエリアがあります。
- システムデータレジスタには、主に次のようなものがあります。

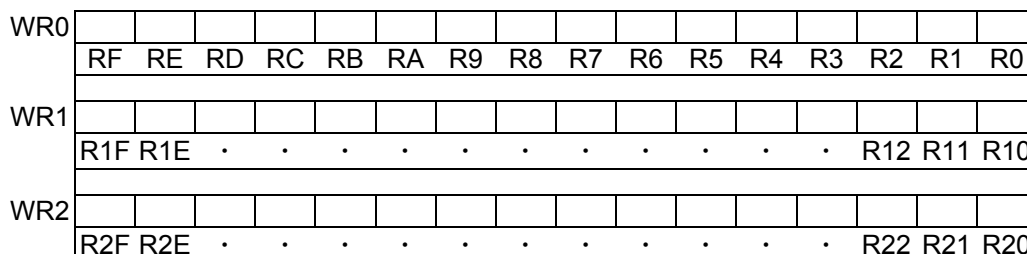
## ■ システムデータレジスタ (SD) の種類

分類	機能
環境設定、動作状態	コンフィグレーションデータや各種の命令などで設定されているPLCの動作状態が格納されています。 例) スキャンタイム
エラー内容	異常が発生したユニット等が格納されます。 例) 自己診断エラーコード、異常が発生したユニットのスロットNo.、演算エラーが発生したアドレス
カレンダータイム	カレンダータイムで計時している年、月、日、時、分、秒、曜日が格納されます。

# WX, WY, WR, WL, WI, WO

## ■ WX, WY, WR, WL, WI, WOのはたらき

- リレー(X, Y, R, L, IN, OT)を 16 点まとめて扱うことができます。
- パルスリレー(P)、異常報知リレー(E)をワード単位で扱うことはできません。
- 1ワード(16ビット)のメモリエリアですので、データメモリとして扱うこともできます。
- ワード単位扱いのメモリエリアの構成は次のとおりです。各々次のように番号が対応します。

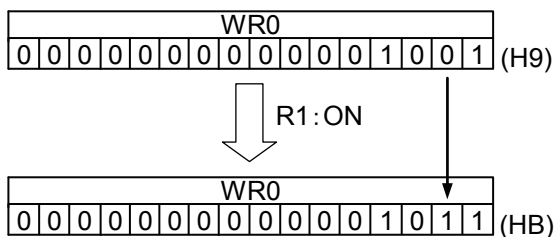


## ■ WX, WY, WR, WL, WI, WOの使用例

- WX や WY は、高機能ユニット入力の取り込みやワード単位の外部入出力に使用することができます。
- WR は、シフトレジスタにも使用します。
- いずれもワード単位の 16 点分のモニタに使用することができます。

## ■ 使用上のご注意

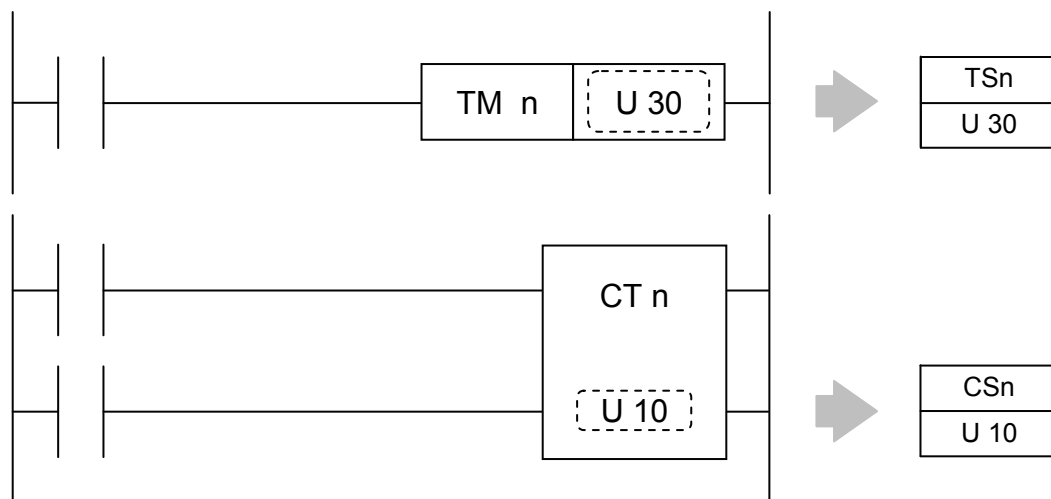
- メモリエリアを構成する各リレーの ON/OFF 状態が変化すれば、メモリエリアの値も変わります。



# TS, CSタイマ / カウンタ設定値レジスタ

## ■ 設定値エリア(TS, CS)のはたらき

- タイマの設定値(TS)、カウンタの設定値(CS)は、タイマ/カウンタと同じ番号の設定値エリアに格納されます。
- 設定値は、プログラムの TM 命令または CT 命令を記述するときに 10 進数または、設定値エリア(TS/CS)の No.で指定します。
- 設定値エリア(TS/CS)は、32 ビットのメモリエリアで 0~4,294,967,295 の 10 進数が格納されます。



## ■ 設定値エリア(TS, CS) の利用

- RUN モード中でも、設定値エリアを書き換えることによってタイマ/カウンタ設定値の変更を行なうことができます。
- 設定値エリアの値は、データ転送命令などのユーザプログラムで書き換えることができます。
- プログラミングツールで読み出し、書き込むこともできます。

## ■ 参考

- TS と TE はタイマ番号、CS と CE はカウンタ番号と 1 対 1 で対応します。

タイマ・カウンタ番号	設定値エリア	経過値エリア
T0	TS0	TE0
T1	TS1	TE1
⋮	⋮	⋮
T4095	TS4095	TE4095
C0	CS0	CE0
C1	CS1	CE1
⋮	⋮	⋮
C1023	CS1023	CE1023



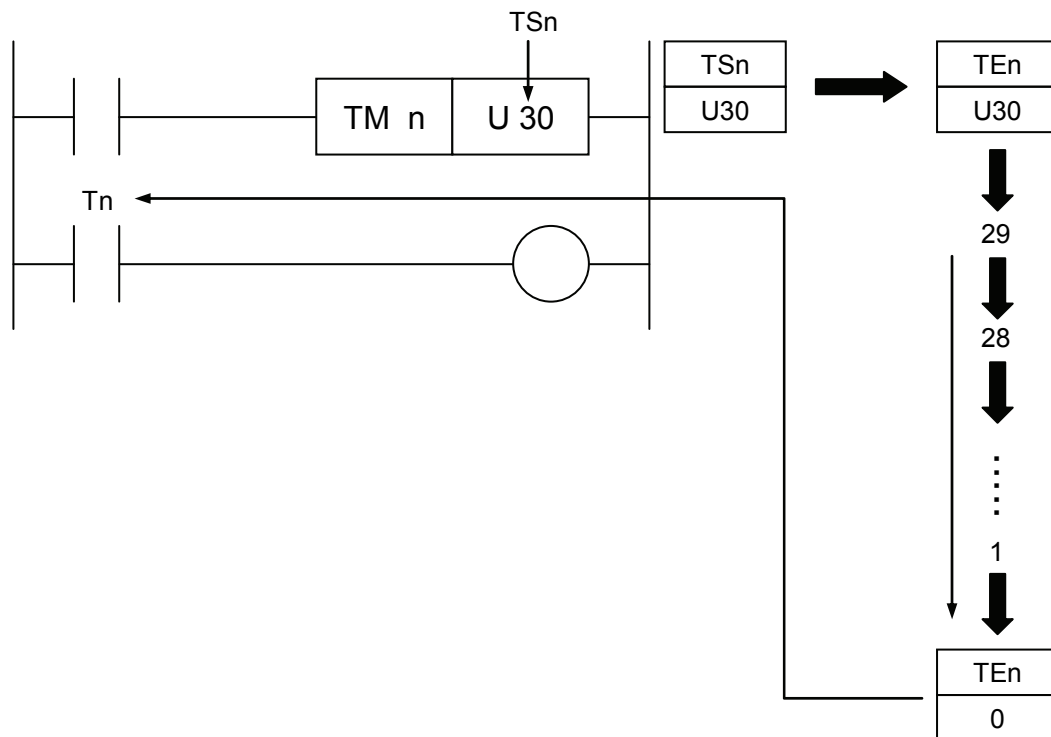
## ◆ ご注意！

- FP7 シリーズのタイマカウンタ設定は、符号なし定数(U 定数)で指定します。
- 設定値エリア(TS, CS) は、32 ビットのエリアを占有します。

# TE, CEタイマ／カウンタ経過値レジスタ

## ■ 経過値エリア(TE, CE)のはたらき

- タイマ、カウンタが実行されている間の経過値が、タイマ／カウンタと同じ番号の経過値エリア(TE/CE)に格納されます。
- 経過値エリア(TE/CE)の値が 0 になった時、同じ番号のタイマ接点またはカウンタ接点を ON にします。
- 経過値エリア(TE/CE)は、32 ビットのメモリエリアで 0～4,294,967,295 の 10 進数が格納されます。



## ■ 経過値エリア(TE, CE)の利用

- 動作中のタイマ／カウンタの経過値を変更して、動作を延長したり短縮することができます。
- 経過値エリアの値は、データ転送命令などのユーザプログラムで書き換えることができます。
- プログラミングツールで読み出し、書き込むこともできます。

## ■ プログラム上のご注意

- FP7 シリーズのタイマカウンタ設定は、符号なし定数(U 定数)で指定します。



### ◆ ご注意！

- 経過値エリア(TE, CE) は、32 ビットのエリアを占有します。

# I0～IE インデックスレジスタ

## ■ インデックスレジスタのはたらき

- インデックスレジスタは、メモリエリアのアドレスや定数を間接的に指定する場合に使用するレジスタです。
- インデックスレジスタの値によって、アドレスや定数を変えることを「インデックス修飾」といいます。
- I0～IE の 32 ビットレジスタ 15 点があります。

## ■ インデックスレジスタ使用時のご注意

- アドレス修飾の結果が各メモリエリアの領域を超える場合は、演算エラーになります。  
例) 修飾の結果、アドレスが負の値になったり大きな値になる場合

## ■ インデックスレジスタで修飾できるもの

- 基本命令で使用するメモリエリアの番号
- スロット番号および、メモリエリアの番号
- 応用命令で使用するメモリエリアの番号
- 応用命令で指定する定数の値
- K 定数(16 ビット、32 ビット)
- U 定数(16 ビット、32 ビット)
- H 定数(16 ビット、32 ビット)

## ■ インデックスレジスタで修飾できないもの

- 浮動小数点データ
- 命令語によりインデックス修飾で指定できないものもあります。各命令語の説明のページの「指定できるメモリエリアの種類」の項でご確認ください。



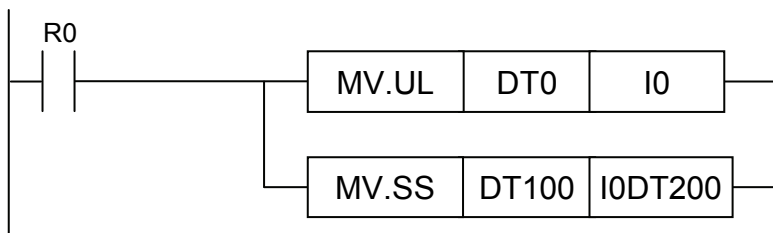
## ◆ ご注意！

- インデックスレジスタ(I0～IE) は、32 ビットのエリアを占有します。

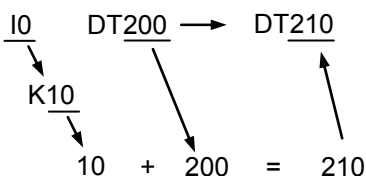
## ■ インデックス修飾のしかた

<例 1> 転送先をインデックス修飾する

DT0 の値によって、転送先のデータレジスタ DT のアドレスが変わります。

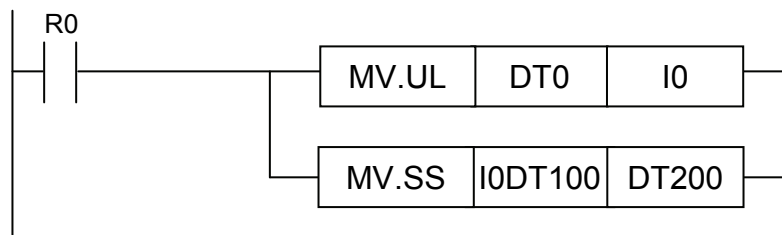


例) DT0 の値が K10 のとき、DT100 の値が、DT210 に書き込まれます。



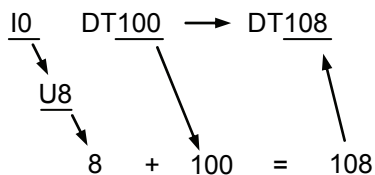


<例 2> 転送元をインデックス修飾する



DT0 の値によって、転送元のデータレジスタ DT のアドレスが変わります。

例) DT0 の値が U8 のとき、DT108 の値が DT200 に転送されます。



■ アドレスの修飾

アドレス = 基準アドレス + I0 ~ IE の値

**I0DT11**

基準アドレス		I0の値	=	対象アドレス
11	+	K0	=	DT11
11	+	K10	=	DT21
11	+	K-10	=	DT1

■ 定数の修飾

定数 = 基準値 + I0 ~ IE の値

**I0K100**

基準値		I0の値	=	定数
K100	+	K0	=	K100
K100	+	K10	=	K100
K100	+	K-10	=	K90

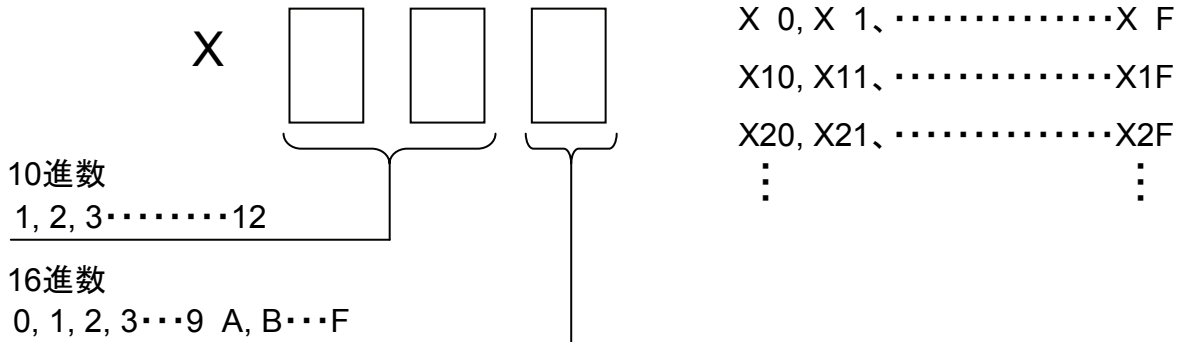
**I0H10**

基準値		I0の値	=	定数
H10	+	HA	=	H1A
H10	+	H10	=	H20

### ■ リレー番号のインデックス修飾時の注意

- 外部リレー(X)、外部出力リレー(Y)、内部リレー(R)について、リレー番号をインデックス修飾する場合は、リレー番号の下1桁目が16進数、上位桁目が10進数であることに注意してください。

<例> 外部入力 X の場合



### ■ インデックス修飾の例

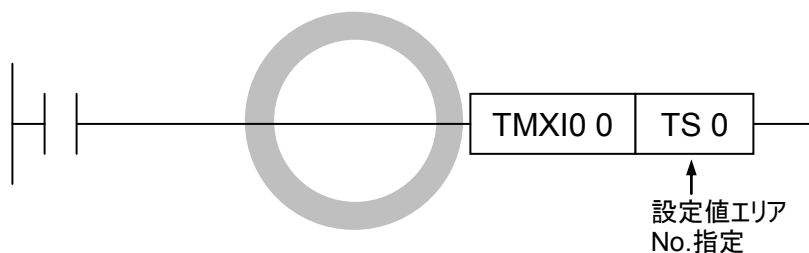
インデックスレジスタの値		インデックス後の リレー番号
K	H	
0	0	X0
1	1	X1
⋮	⋮	⋮
9	9	X9
10	A	XA
⋮	⋮	⋮
15	F	XF
16	10	X10
⋮	⋮	⋮
31	1F	X1F
⋮	⋮	⋮
159	9F	X9F
160	A0	X100
161	A1	X101
⋮	⋮	⋮
255	FF	X15F
256	100	X160
257	101	X161
⋮	⋮	⋮
265	10A	X16A
267	10B	X16B
⋮	⋮	⋮

## ■ 基本命令の命令番号の修飾

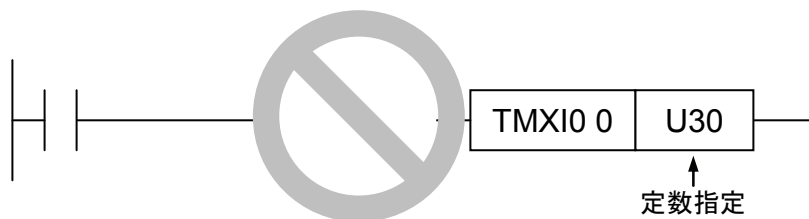
対象	修飾例	
タイマ番号	TML20を修飾する	TML I020
カウンタ番号	CT200を修飾する	CT I0200
シフトレジスタ番号	SRWR0を修飾する	SR I0WR0
ジャンプ命令でのラベル番号指定	JP1を修飾する	JP I01
ループ命令でのラベル番号指定	LOOP5を修飾する	LOOP I05
サブルーチンプログラム番号	CALL10を修飾する	CALL I010

## ■ タイマ/カウンタ番号の修飾の制限

- タイマ番号およびカウンタ番号は、設定値にメモリエリアを指定している時のみ修飾できます。



- 定数で設定値を指定しているときは、修飾できません。



# 1-7 定数の説明

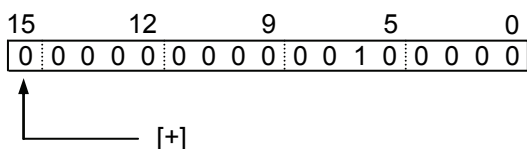
## K 符号付き 10 進定数

### ■ 符号付き10進定数(K)のはたらき

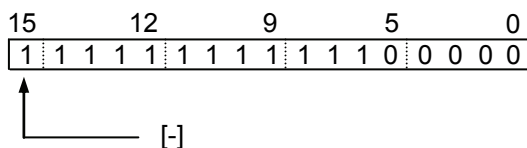
- バイナリデータを 10 進数に換算したものです。入力時には、数値の前に“K”を付けて指定してください。
- 主にデータの大きさや数量を指定するときに使います。
- 10 進定数 K は、PLC 内部では、下記のように 16 ビットワード単位の BIN データとして処理します。
- 最上位ビットを符号ビットとなり、正負が決まります。0 のときは正の数(+)、1 のときは負の数(-)。
- データは通常 1 ワード(16 ビット)単位で扱いますが、2 ワード(32 ビット)をまとめて扱うこともあります。この場合も最上位ビットが符号ビットになります。

### ■ 符号付き10進定数(K)の形式

<例> 10 進数 “+32” (K32)



<例> 10 進数 “-32” (K-32)



### ■ 10進定数(K)で指定できる範囲

演算	指定できる範囲
16ビット演算時	K-32,768~K32,767
32ビット演算時	-2,147,483,648~K2,147,483,647

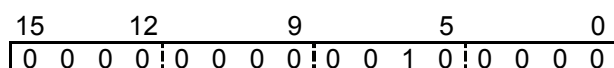
## U 符号なし 10 進定数

### ■ 符号なし10進定数(U)のはたらき

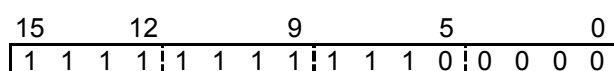
- バイナリデータを 10 進数に換算したものです。入力時および読み出し時には、数値の前に“U”を付けて指定してください。
- タイマの設定値など、主にデータの大きさや数量を指定するときに使います。
- 10 進定数 U は、PLC 内部では、下記のように 16 ビット単位の BIN データとして処理します。
- U 符号なしは、全 16 ビットで数値を表すため負の数扱うことはできません。
- データは通常 1 ワード(16 ビット)単位で扱いますが、2 ワード(32 ビット)をまとめて扱うこともあります。

### ■ 符号なし10進定数(U)の形式

<例> 10 進数 “+32” の場合 (U32)



<例> 10 進数 “+65504” の場合 (U65504)



### ■ 10進定数(U)で指定できる範囲

演算	指定できる範囲
16ビット演算時	U0~U65,535
32ビット演算時	U0~U4,294,967,295

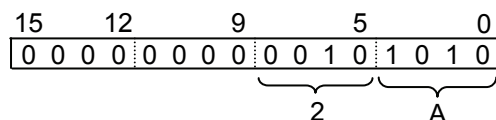
## H 16 進定数

### ■ 16進定数(H)のはたらき

- バイナリデータを 16 進数に換算したものです。入力時および読み出し時には、数値の前に “H” を付けて指定してください。
- システムデータレジスタの設定や応用命令のコントロールデータの指定など、主に 16 ビットデータの 1 と 0 の並びを指定するときに使います。また、BCD データを指定するときにも使用します。
- 16 進定数 H は、PLC 内部では、下記のように 16 ビット単位の BIN データとして処理します。
- データは通常 1 ワード(16 ビット)単位で扱いますが、2 ワード(32 ビット)をまとめて扱うこともあります。

### ■ 16進定数(H)の形式

<例> 16 進数 “2A” の場合 (H2A)



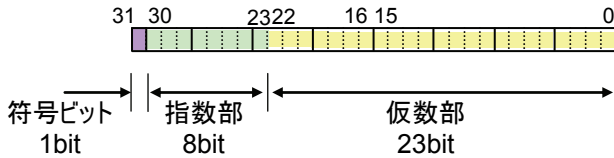
### ■ 16進定数(H)で指定できる範囲

演算	指定できる範囲
16ビット演算時	H0~HFFFF
32ビット演算時	H0~HFFFFFFF

# SF 単精度浮動小数点形実数定数

## ■ 単精度浮動小数点形の形式

- IEEE754 フォーマットに準拠し、符号ビット 1 ビット、指数部 8 ビット、仮数部 23 ビットから構成されています。
- 実数进行处理する演算命令、実数整数間の変換命令が用意されているため、データフォーマットを考慮してプログラムする必要はありません。

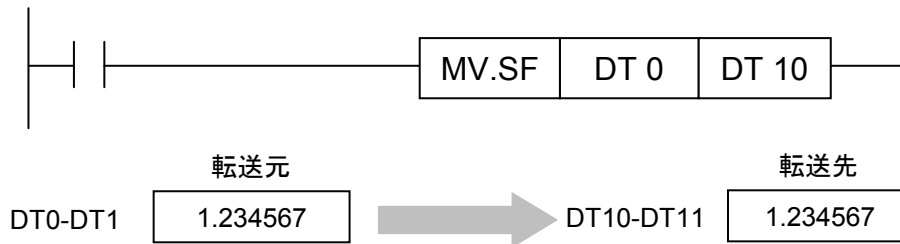


## ■ 単精度浮動小数点形実数定数の範囲

演算	指定できる範囲
32ビット演算	負の範囲: $-1.175494\text{E}-38 \sim -3.402823\text{E}+38$ 0 正の範囲: $3.402823\text{E}+38 \sim 1.175494\text{E}-38$

## ■ 単精度浮動小数点形実数定数を格納するエリア

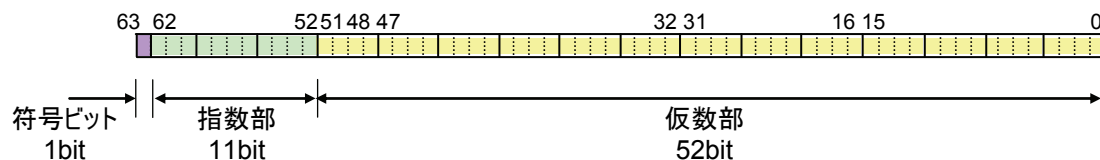
- 単精度浮動小数点形実数定数演算命令で、実数に変換されたデータを格納するエリアは 1 データあたり 2 ワード 32 ビット分のエリアを使用します。
  - 実数データを格納したエリアを転送命令などで操作する場合は、演算単位に SF を入力してください。
- <例> 命令コードの演算単位をSFに設定すると、DTには2ワード分のデータが入ります。



# DF 倍精度浮動小数点形実数定数

## ■ 倍精度浮動小数点形の形式

- IEEE754 フォーマットに準拠し、符号ビット 1 ビット、指数部 11 ビット、仮数部 52 ビットから構成されています。
- 実数処理する演算命令、実数整数間の変換命令が用意されているため、データフォーマットを考慮してプログラムする必要はありません。

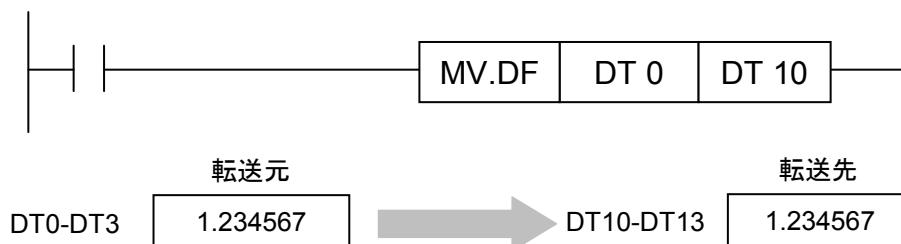


## ■ 倍精度浮動小数点形実数定数の範囲

演算	指定できる範囲
64ビット演算	負の範囲: $-2.2250738585072014E-308 \sim -1.7976931348623158E+308$ 0 正の範囲: $+2.2250738585072014E-308 \sim +1.7976931348623158E+308$

## ■ 倍精度浮動小数点形実数定数を格納するエリア

- 倍精度浮動小数点形実数定数演算命令で、実数に変換されたデータを格納するエリアは 1 データあたり 4 ワード 64 ビット分のエリアを使用します。
  - 実数データを格納したエリアを転送命令などで操作する場合は、演算単位に DF を入力してください。
- <例> 命令コードの演算単位を DF に設定すると、DT には 4 ワード分のデータが入ります。



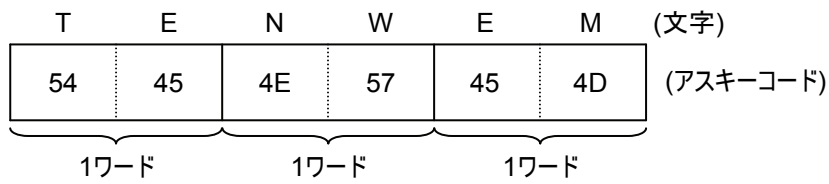
# “” 文字定数

## ■ 文字定数(“”)のはたらき

- バイナリデータを ASCII コードとして扱うものです。入力時には、“”で文字を括って指定してください。
- 文字定数を設定できる命令は、SSET<文字定数→ASCII コード変換>命令です。ツールソフトウェアでのみ入力できます。
- 文字定数(“”)は、PLC 内部では下記のように指定されたメモリエリアに BIN データとして格納されます。

## ■ 文字定数(“”)の形式

<例> 文字定数“MEWNET”を入力した場合

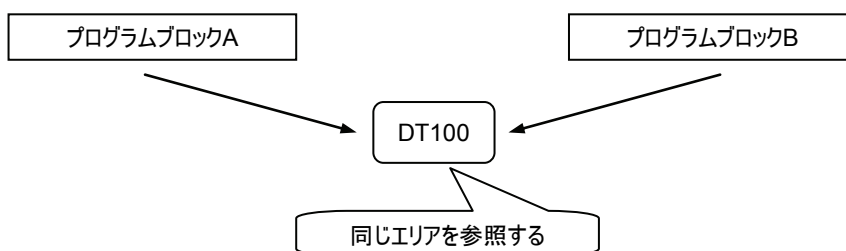




# 1-8 グローバルデバイスとローカルデバイス

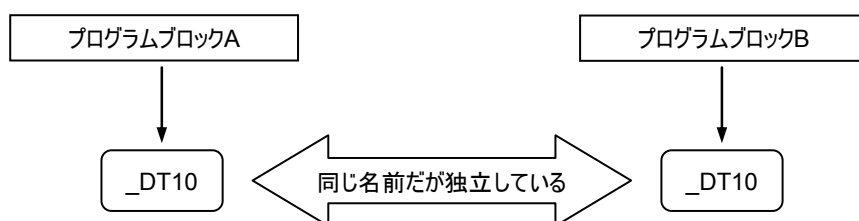
## ■ グローバルデバイス

- グローバルデバイスはプログラム全体で固有の番号を持ちますが、ローカルデバイスはプログラムブロックの内部で固有の番号を持ちます。
- 例えば、グローバルデバイス DT100 がプログラムブロック A と B で変更される場合、同じデータメモリを表すため他方のプログラムブロックによって内容が上書きすることになります。
- 外部への入出力など、複数のプログラムブロック (PB)、演算用メモリを共通で使用する場合に用います。



## ■ ローカルデバイス

一方、ローカルデバイス dt10 がプログラムブロック A と B で変更される場合は、実際には別のデータメモリに割り当てられているため、他方のプログラムブロックによって内容が上書きされることがありません。



## ■ ローカルデバイスが使用可能なメモリエリアの一覧 (●:使用可能)

ビットデバイス			16ビットデバイス		32ビットデバイス
X, Y, R, L, T, C	P, E, SR, IN, OT	DT.n, LD.n	WX, WY, WR, WL, DT, LD	SD, WI, WO, UM	TS, CS, TE, CE
●		●	●		●

# 1-9 PLC内部で扱えるデータの範囲

## ■ 16ビットの場合

PLC内部で扱われるデータ(2進16ビット)	10進数換算		16進数換算
	符号なし	符号付き	
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	U32767	K32767	H7FFF
⋮	⋮	⋮	⋮
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	U 1	K 1	H 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	U 0	K 0	H 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	U65535	K -1	HFFFF
⋮	⋮	⋮	⋮
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	U32768	K-32768	H8000

## ■ 32ビットの場合

PLC内部で扱われるデータ(2進32ビット)	10進数換算		16進数換算
	符号なし	符号付き	
01111111111111111111111111111111	U2147483547	K2147483547	H7FFFFFFF
⋮	⋮	⋮	⋮
00000000000000000000000000000001	U 1	K 1	H 1
00000000000000000000000000000000	U 0	K 0	H 0
11111111111111111111111111111111	U4294967295	K -1	HFFFFFFFF
⋮	⋮	⋮	⋮
10000000000000000000000000000000	U2147438648	K-2147438648	H80000000

## ■ PLC内部で扱えるデータの範囲

演算の種類		演算できる範囲	
バイナリ演算時	16ビット	符号なし	U 0 ~ U 65,535
		符号あり	K -32,768 ~ K 32,767
	32ビット	符号なし	U 0 ~ U 4,294,967,295
		符号あり	K -2,147,483,548 ~ K 2,147,483,54
BCD演算時	16ビット	符号なし	H 0 ~ H 9999
	32ビット	符号なし	H 0 ~ H 99999999

(注)いずれの場合も、上記の範囲を超えると、オーバーフロー、あるいは、アンダーフローとなります。

## ■ PLC内部での10進数の表現

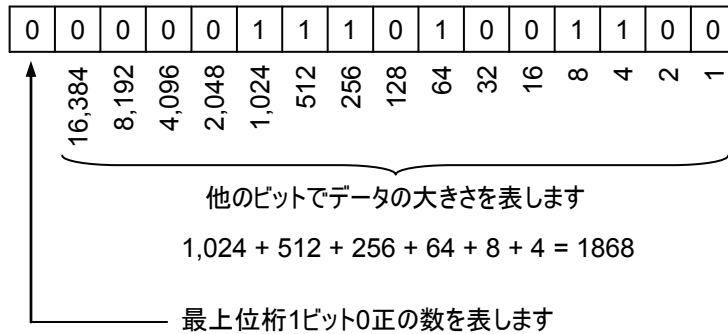
• 10進数は、16ビットあるいは32ビットのバイナリデータとして処理されます。

• 符号付きデータの場合

① 最上位1ビットは符号ビットで正の数の場合“0”、負の数の場合“1”となります。

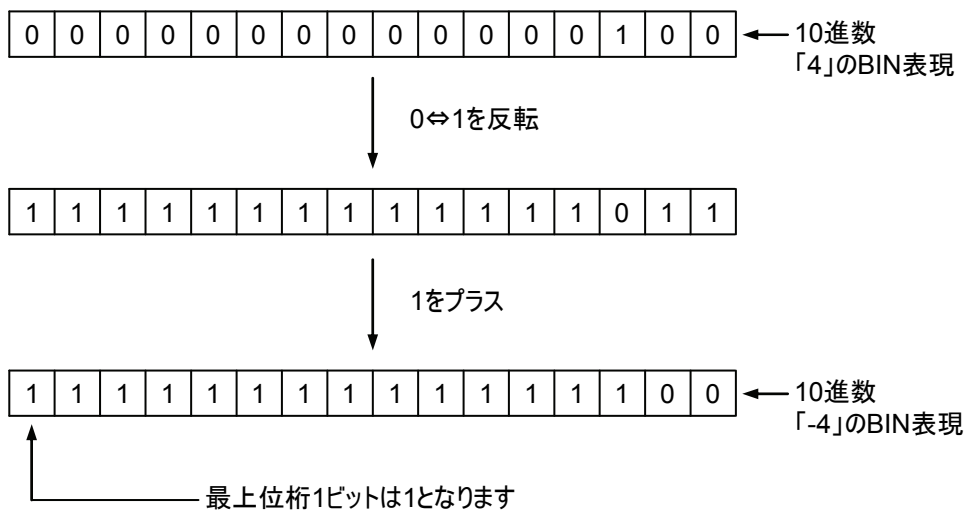
② 正の数の場合、最上位桁を除く他のビットでデータの大きさを表します。

<例> 10進数「1868」の表現



③ 負の数の場合、2の補数で表現されます。2の補数は、正の数のバイナリデータ  $0 \leftrightarrow 1$  を反転し、1を加えたバイナリデータです。

<例> 10進数「-4」の表現



# 1-10 オーバーフロー・アンダーフロー

## ■ オーバーフロー・アンダーフローとは？

- 演算命令では、演算時に取り扱える値を超えてしまう場合があります。
- このとき、最大値を超える場合を「オーバーフロー」、最小値を超える場合を「アンダーフロー」と呼びます。

## ■ バイナリ演算のオーバーフロー・アンダーフロー

- 以下の数値を超えた場合、オーバーフロー・アンダーフローとなります。

<16ビット演算>

(最大値を超えるとオーバーフロー)

	符号なしの場合		符号付きの場合	
最大値	U65535	HFFFF	K32767	H7FFF
	⋮	⋮	⋮	⋮
	U32768	H8000	K 1	H 1
	U32767	H7FFF	K 0	H 0
	⋮	⋮	K -1	HFFFF
	U 1	H 1	⋮	⋮
最小値	U 0	H 0	K-32768	H8000

(最小値を下回るとアンダーフロー)

<32ビット演算>

(最大値を超えるとオーバーフロー)

	符号なしの場合		符号付きの場合	
最大値	U4294967295	HFFFFFFFF	K2147483647	H7FFFFFFF
	⋮	⋮	⋮	⋮
	U2147483648	H80000000	K 1	H 1
	U2147483647	H7FFFFFFF	K 0	H 0
	⋮	⋮	K -1	HFFFFFFFF
	U 1	H 1	⋮	⋮
最小値	U 0	H 0	K-2147483648	H80000000

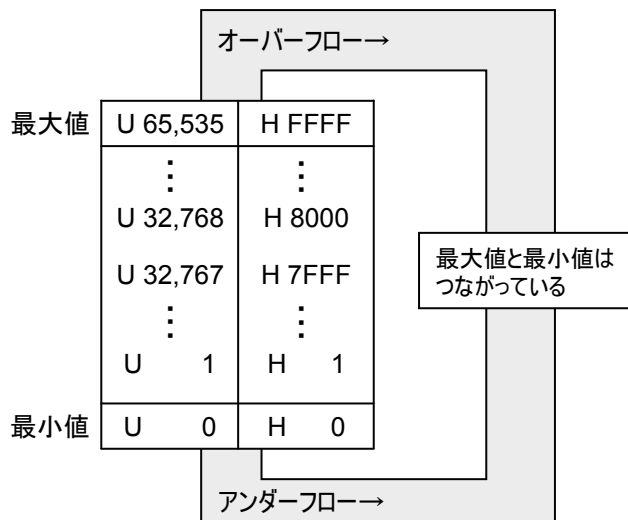
(最小値を下回るとアンダーフロー)

## ■ オーバーフロー・アンダーフロー時の値

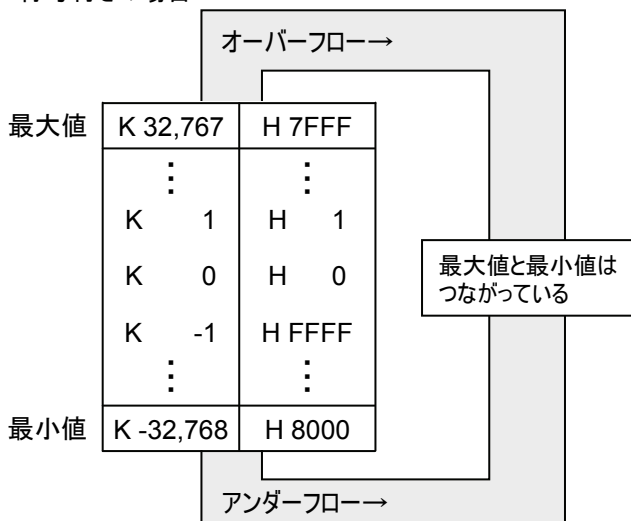
演算時取り扱える数値は全て、図のように最大値と最小値のつながった“ループ状”となっています。

例1) 16ビットバイナリ演算時

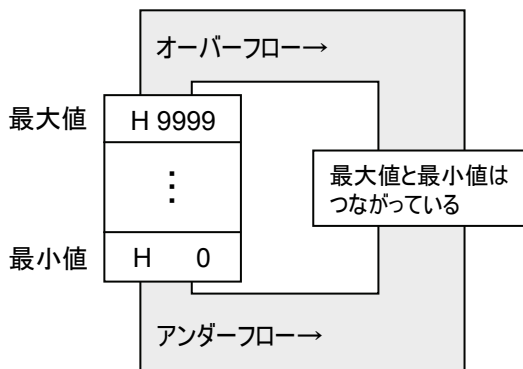
・符号なしの場合



・符号付きの場合



例2) 16ビットBCD演算時





# 2

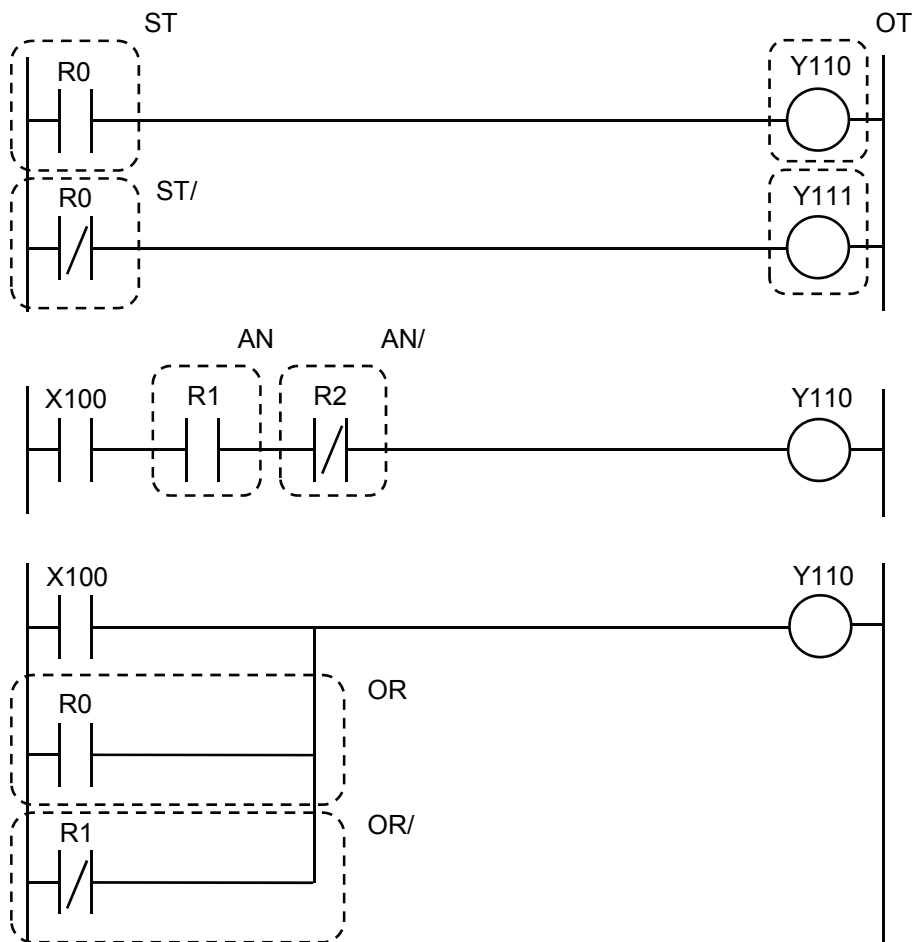
## 基本命令

## ST, ST/, OT (スタート、スタートノット、アウト)

## AN, AN/ (アンド、アンドノット)

## OR, OR/ (オア、オアノット)

### ■ ラダー表記



### ■ 指定可能なデバイス (●:指定可能)

オペランド		ビットデバイス										ワードデバイスのビット指定		インデックス修飾		
		X	Y	R	L	T	C	P	E	S	IN	OT	DT.b		LD.b	
ST ST/ AN, AN/ OR, OR/	bit	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
OT		●	●	●	●				●			●	●	●	●	



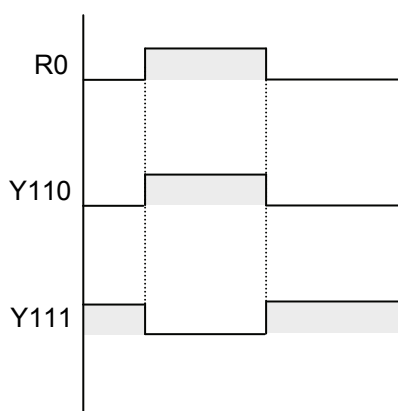
## ■ 動作説明

命令の種類	動作
ST	「ST」命令で指定した入力接点を常開接点(a接点)として扱い、論理演算を開始します。
ST/	「ST/」命令で指定した入力接点を常閉接点(b接点)として扱い、論理演算を開始します。
AN	「AN」命令で指定した入力接点を常開接点(a接点)として扱い、直列接続した直前の演算結果との論理積を演算します。
AN/	「AN/」命令で指定した入力接点を常閉接点(b接点)として扱い、直列接続した直前の演算結果との論理積を演算します。
OR	「OR」命令で指定した入力接点を常開接点(a接点)として扱い、並列接続した直前の演算結果との論理和を演算します。
OR/	「OR/」命令で指定した入力接点を常閉接点(b接点)として扱い、並列接続した直前の演算結果との論理和を演算します。
OT/	「OT」命令は、演算結果を指定したコイルに出力します。

## ■ 動作例

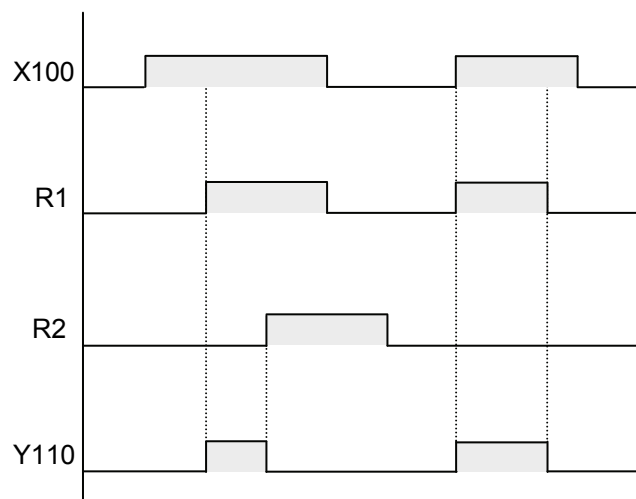
### (1) ラダー表記にある「ST」、「ST/」、「OT」のプログラム動作

R0がONのとき、Y110に出力し、R0がOFFのとき、Y111に出力します。



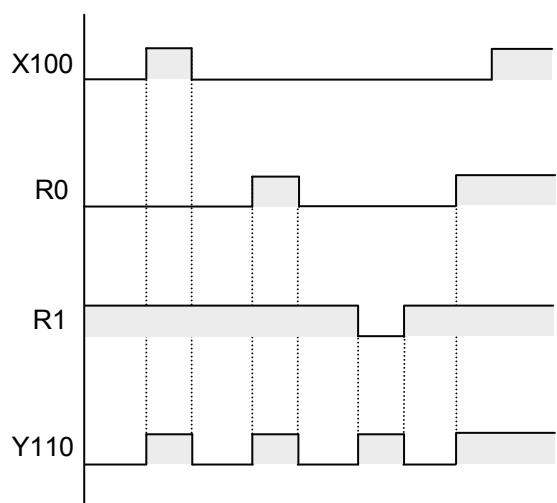
### (2) ラダー表記にある「AN」、「AN/」のプログラム動作

X100がON、R1がON、R2がOFFのとき、Y110に出力します。



### (3) ラダー表記にある「OR」、「OR/」のプログラム動作

X100がON、R0がON、R1がOFFのいずれかの条件が満たされたときY110に出力します。



#### ■ プログラム上のご注意

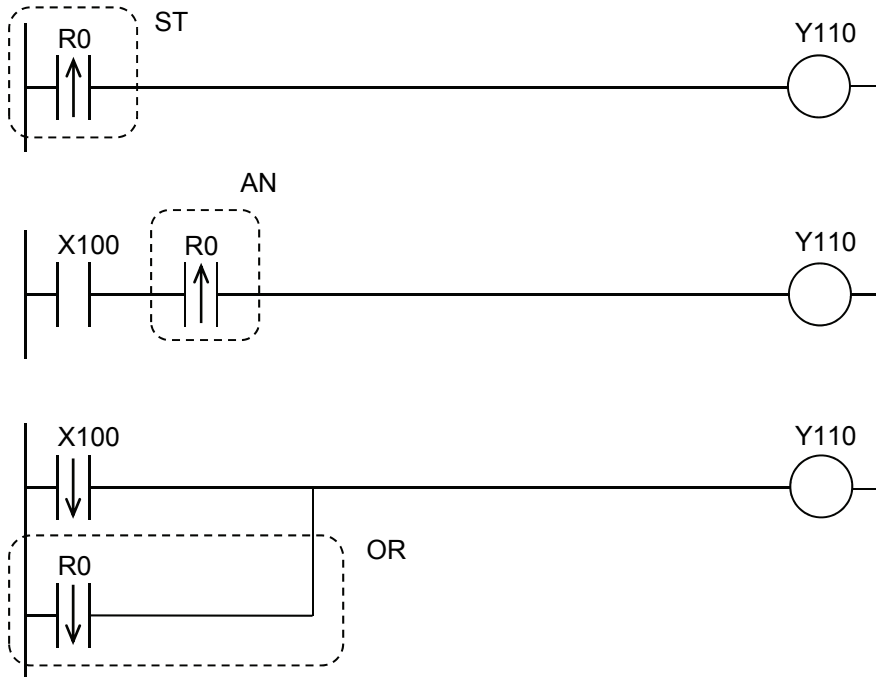
- 「ST」、「ST/」、「OR」、「OR/」命令は、母線から始まります。
- 「AN」、「AN/」、「OR」、「OR/」命令は、連続して使用できます。
- 非常停止スイッチなどのように、外部スイッチが **b** 接点の場合、プログラム上では「ST」命令を使うように注意してください。

**ST ↑, ST ↓ (立ち上がり、立ち下がり接点命令)**

**AN ↑, AN ↓ (立ち上がり、立ち下がり接点命令)**

**OR ↑, OR ↓ (立ち上がり、立ち下がり接点命令)**

■ ラダー表記



■ 指定可能なデバイス (●:指定可能)

オペランド	ビットデバイス											ワードデバイスのビット指定		インデックス修飾
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b	LD.b	
bit	●	●	●	●	●	●	●	●	●	●	●	●	●	

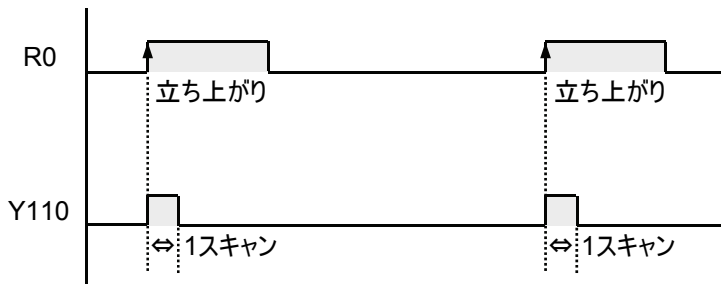
■ 動作説明

命令の種類	動作
ST ↑	信号が OFF 状態から ON 状態に変化(立ち上がり)したその 1 スキャンのみ導通します。 入力接点を常開接点(a 接点)もしくは、常閉接点(b 接点)として扱い、論理演算を開始します。
AN ↑	信号が OFF 状態から ON 状態に変化(立ち上がり)したその 1 スキャンのみ導通します。 入力接点を常開接点(a 接点)もしくは、常閉接点(b 接点)として扱い、直列接続した直前の演算結果との論理積を演算します。
OR ↑	信号が OFF 状態から ON 状態に変化(立ち上がり)したその 1 スキャンのみ導通します。 入力接点を常開接点(a 接点)もしくは、常閉接点(b 接点)として扱い、並列接続した直前の演算結果との論理和を演算します。
ST ↓	信号が ON 状態から OFF 状態に変化(立ち下がり)したその 1 スキャンのみ導通します。 入力接点を常開接点(a 接点)もしくは、常閉接点(b 接点)として扱い、論理演算を開始します。
AN ↓	信号が ON 状態から OFF 状態に変化(立ち下がり)したその 1 スキャンのみ導通します。 入力接点を常開接点(a 接点)もしくは、常閉接点(b 接点)として扱い、直列接続した直前の演算結果との論理積を演算します。
OR ↓	信号が ON 状態から OFF 状態に変化(立ち下がり)したその 1 スキャンのみ導通します。 入力接点を常開接点(a 接点)もしくは、常閉接点(b 接点)として扱い、並列接続した直前の演算結果との論理和を演算します。

## ■ 動作例

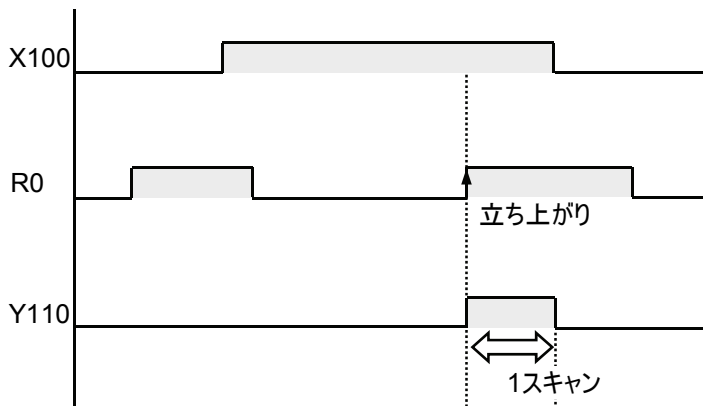
### (1) ラダー表記にある「ST↑」のプログラム動作

R0がOFF状態からON状態に変化した(立ち上がり)1スキャンのみY110に出力します。



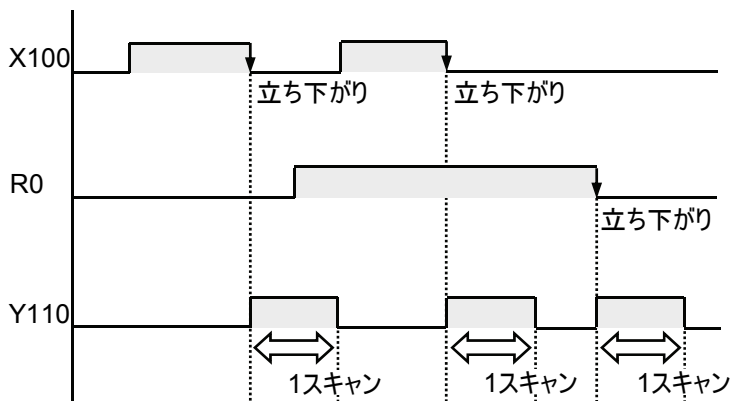
### (2) ラダー表記にある「AN↑」のプログラム動作

X100がON状態にあり、R0がOFF状態からON状態に変化した(立ち上がり)1スキャンのみY110に出力します。



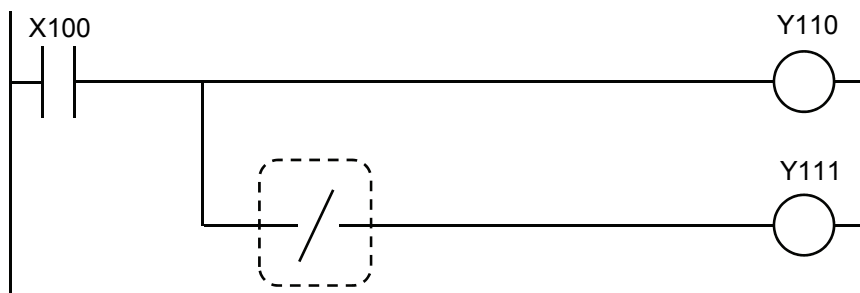
### (3) ラダー表記にある「OR↑」のプログラム動作

X100またはR0がON状態からOFF状態に変化した(立ち下がり)1スキャンのみY110に出力します。



# / (ノット)

## ■ ラダー表記



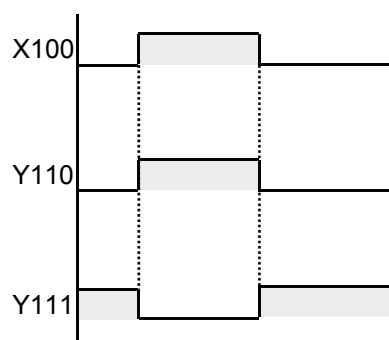
## ■ 動作説明

- 「/」命令は、この命令の直前までの演算結果を反転します。

## ■ 動作例

X100がONのとき、Y110をON、Y111をOFFとします。

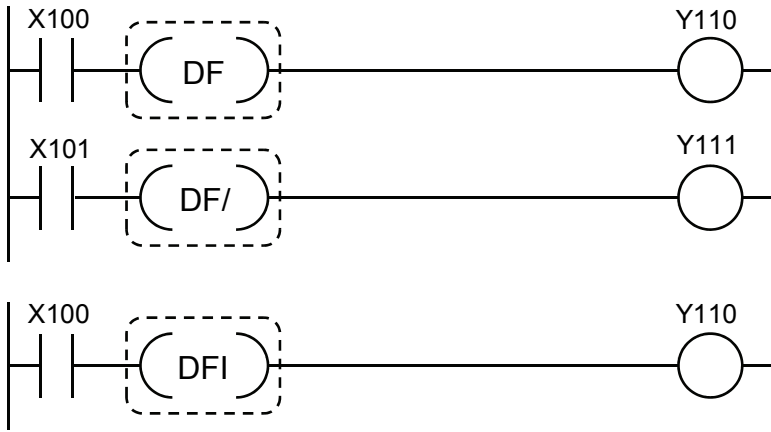
X100がONのとき、Y110をON、Y111をOFFとします。



# DF, DF/ (立ち上がり微分、立ち下がり微分)

# DFI (立ち上がり微分(イニシャル実行型))

## ■ ラダー表記



## ■ 動作説明

命令の種類	動作
DF	「DF」命令は、実行条件が OFF 状態から ON 状態に変化(立ち上がり)したその 1 スキャンのみ出力(微分出力)します。
DF/	「DF/」命令は、実行条件が ON 状態から OFF 状態に変化(立ち下がり)したその 1 スキャンのみ出力(微分出力)します。
DFI	「DFI」命令は、実行条件が OFF 状態から ON 状態に変化(立ち上がり)したその 1 スキャンのみ出力(微分出力)します。

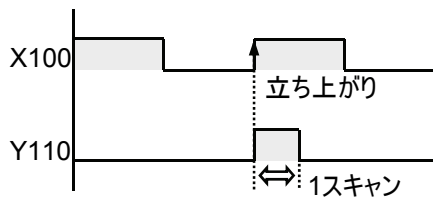
## ■ プログラム上のご注意

- 微分命令(「DF」、「DF/」、「DFI」)の使用回数に制限はありません。
- 微分命令では、接点の ON/OFF 状態の変化のみを検出します。
- 「DF」、「DF/」命令は、RUN モードに切り替えたときや、RUN モードで電源を立ち上げたときに実行条件が最初から成立している(ONしている)場合は、出力しません。
- 「DFI」命令は、RUN を開始する前から、実行条件が成立している場合も、最初の 1 スキャンで出力(微分出力)を行います。
- RUN モードに切り替えたときや RUN モードで電源を立ち上げたときに、実行条件が成立する可能性がある場合は、「DF」命令での最初の 1 スキャンでの出力は得られませんので、「DFI」命令を使用してください。
- MC~MCE 命令、JP~LBL 命令など命令を実行する順序を変える命令(下記1~6)と合わせて微分命令を使用するときは、注意が必要です。
  - 1) MC~MCE 命令
  - 2) JP~LBL 命令
  - 3) LOOP~LBL 命令
  - 4) CNDE 命令
  - 5) ステップラダー命令
  - 6) サブルーチン命令
- 微分命令を、アンドスタック命令やポップスタック命令と組み合わせるときは、正しくない記述にならないようご注意ください。

## ■ 動作例

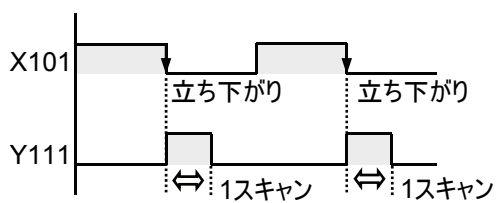
(1) ラダー表記にある「DF」のプログラム動作

X100がOFF状態からON状態に変化した(立ち上がり)1スキャンのみY110に出力します。



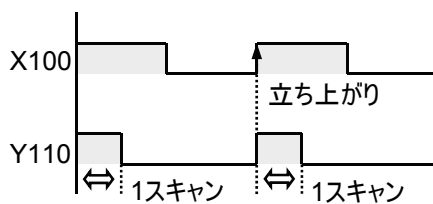
(2) ラダー表記にある「DF/」のプログラム動作

X101がON状態からOFF状態に変化した(立ち下がり)1スキャンのみY111に出力します。



(3) ラダー表記にある「DFI」のプログラム動作

X100がOFF状態からON状態に変化した(立ち上がり)1スキャンのみY110に出力します。

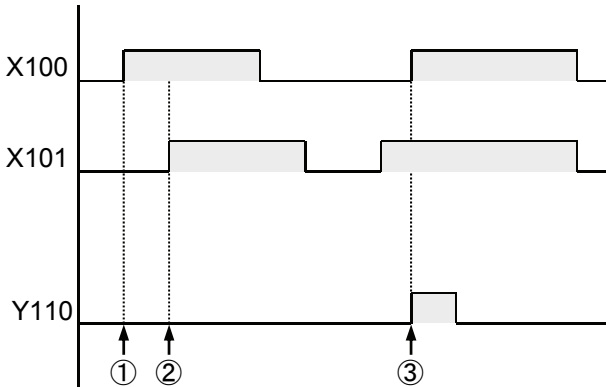
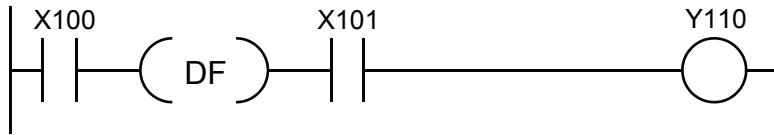


## ■ サンプルプログラム

• 下図のような回路では、動作は次のようになります。

### (1) 「DF」命令を使用した参考例1

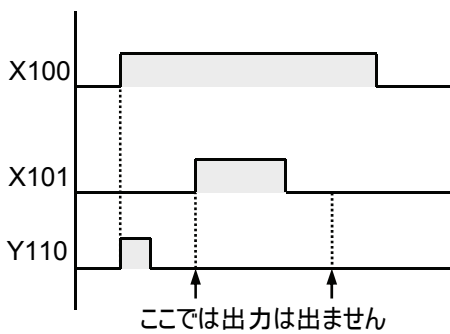
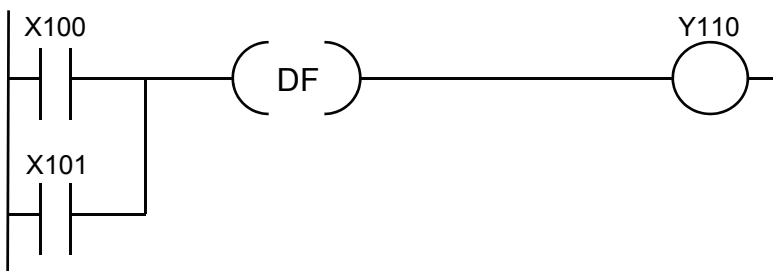
→ 入力情報(X100、X101)の間に立ち上がり微分を設定した場合



- 1) X101がOFFのときは、X100が立ち上がってもY110はOFFのままです。
- 2) X100がONのときにX101が立ち上がってもY110はOFFのままです。
- 3) X101がONのとき、X100が立ち上がるとY110は1スキャンの間ONとなります。

### (2) 「DF」命令を使用した参考例2

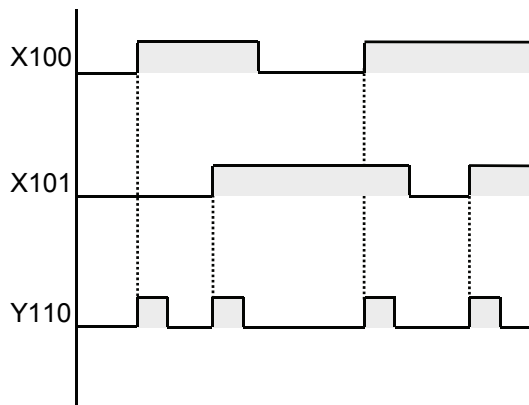
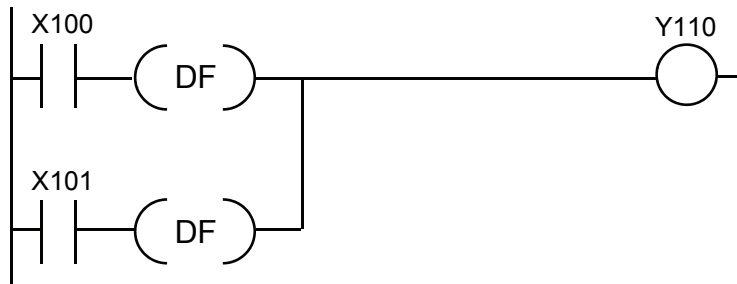
→ 入力情報(X100、X101)の並列接続後に立ち上がり微分を設定した場合





(3) 「DF」命令を使用した参考例3

→入力情報(X100、X101)のそれぞれに立ち上がり微分を設定した場合

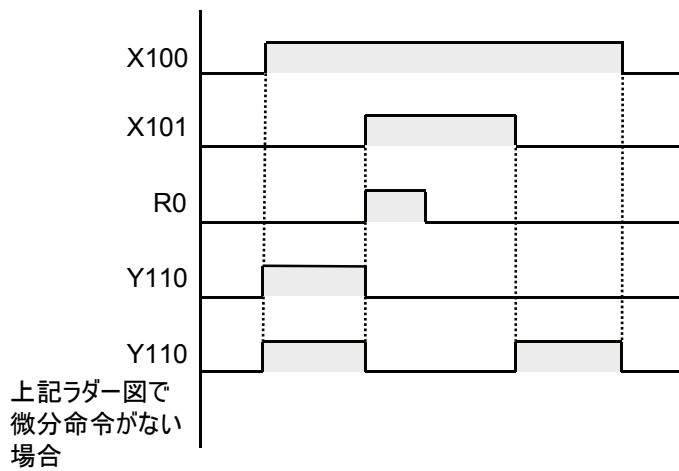
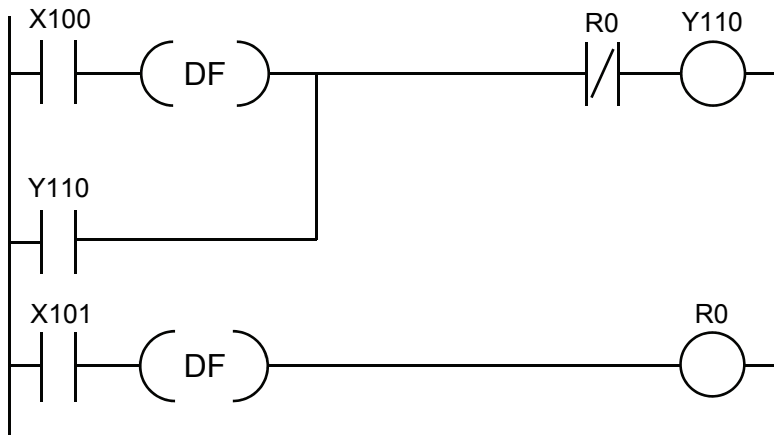


## ■ 微分命令の応用例

- 微分命令を使うとプログラムの作成調整が楽になります。

< 自己保持回路への応用例 >

- 微分命令を使うと、入力信号が長い場合などに対応できます。

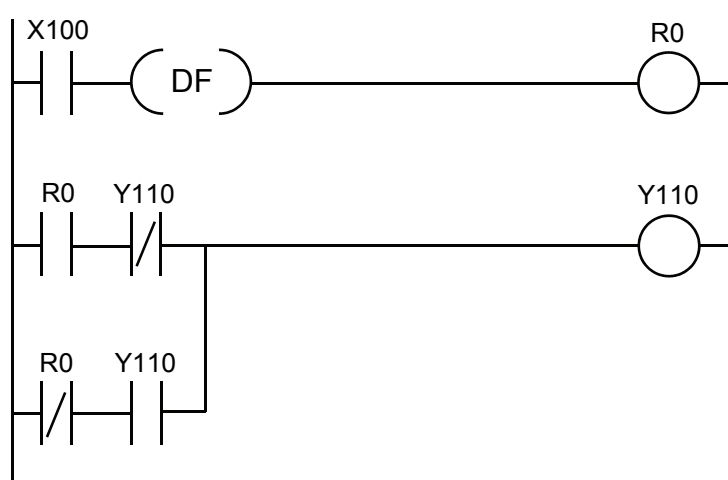


上記ラダー図で  
微分命令がない  
場合

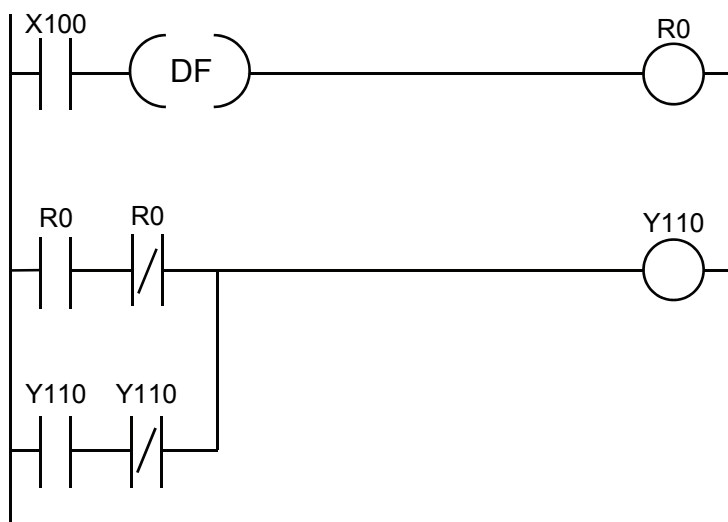
<オルタネイト回路への応用例>

- 1つの信号で、回路の保持・解除をさせる、オルタネイト回路にも応用できます。

<例1>

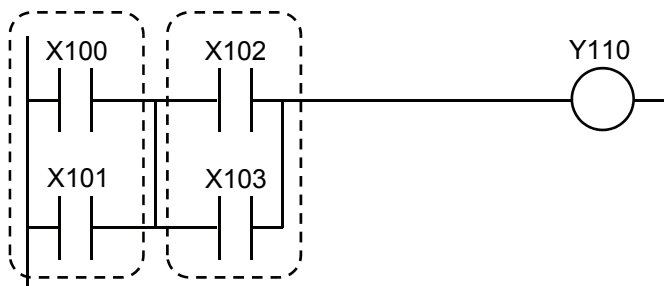


<例2>



# ANS (アンドスタック)

## ■ ラダー表記



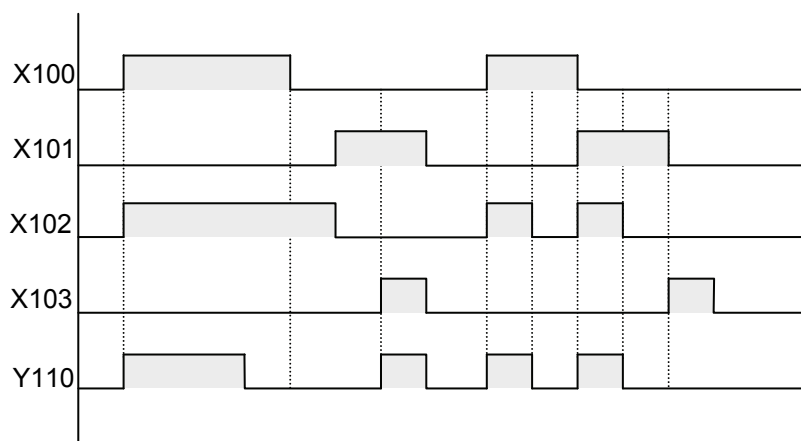
## ■ 動作説明

- 並列接続したブロックを直列に接続します。
- 各ブロックの最初は、「ST」命令で始めます。

## ■ 動作例

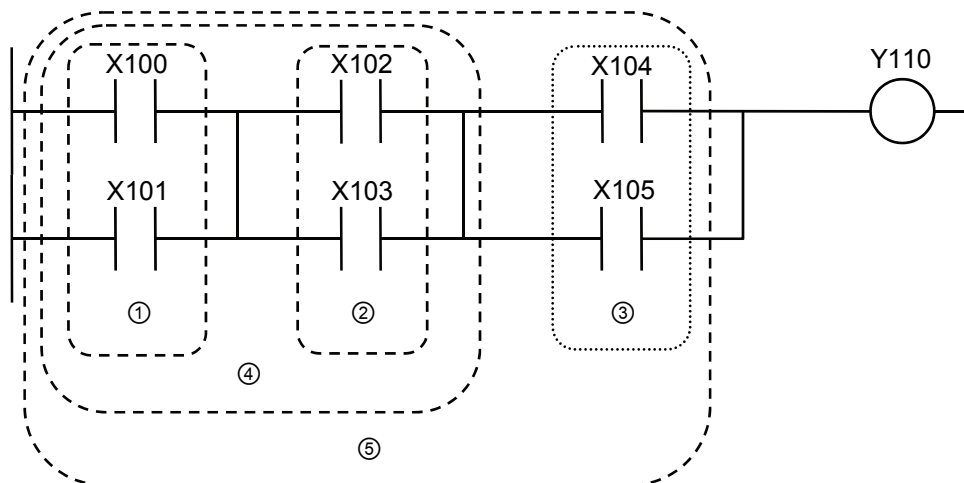
X100またはX101がONでかつ、X102またはX103がONのとき、Y110に出力します。

**(X100 AND X101) OR (X102 AND X103) → Y110**



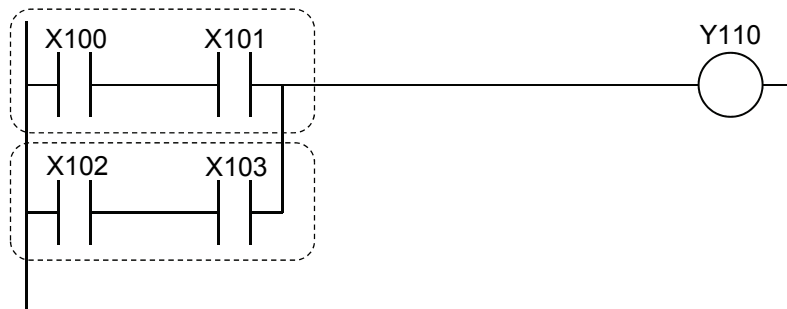
## ■ ブロックが連続する場合

- ブロックが連続する場合は、次のようにブロックを分けて考えてください。



# ORS (オアスタック)

## ■ ラダー表記



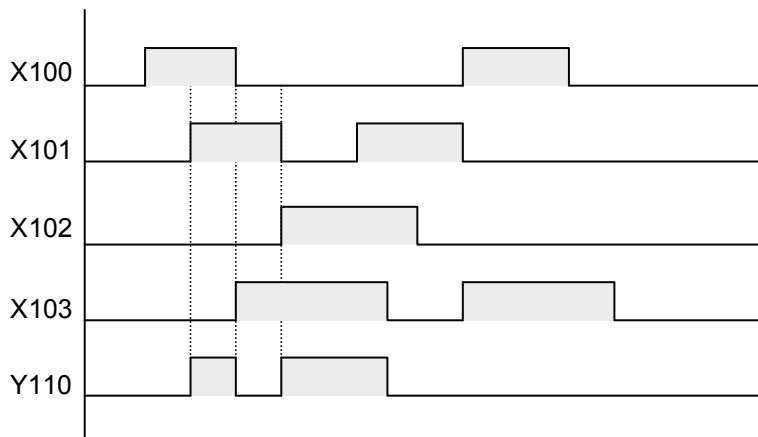
## ■ 動作説明

- 直列接続したブロックを並列に接続します。
- 各ブロックの最初は、「ST」命令で始めます。

## ■ 動作例

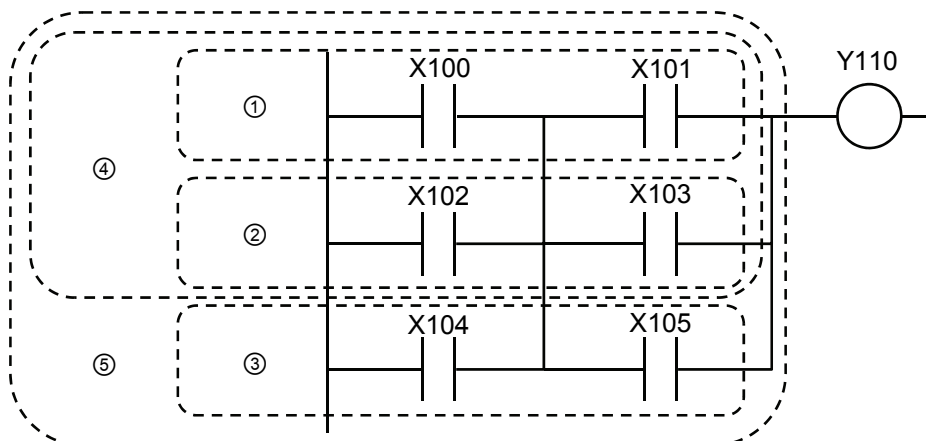
X100かつX101がONか、または、X102かつX103がONのとき、Y110に出力します。

**(X100 OR X101) AND (X102 OR X103) → Y110**



## ■ ブロックが連続する場合

- ブロックが連続する場合は、次のようにブロックを分けて考えてください。

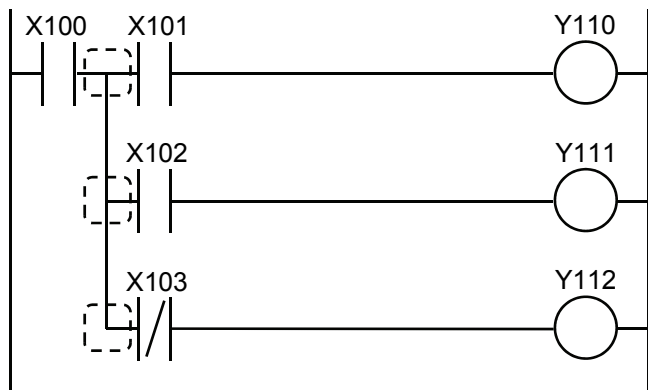


# PUSHS (プッシュスタック)

# RDS (リードスタック)

# POPS (ポップスタック)

## ■ ラダー表記



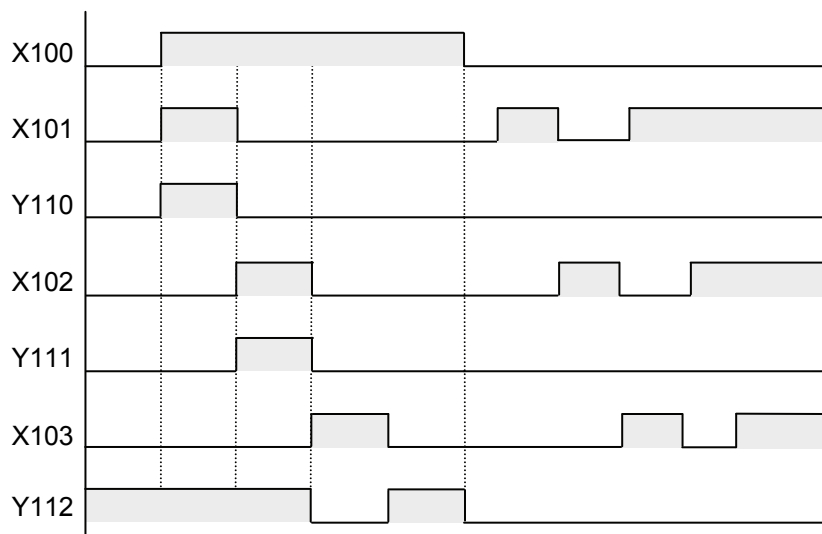
## ■ 動作説明

命令の種類	動作
PUSHS	本命令の直前の演算結果を記憶し、次のステップから演算を実行します。
RDS	「PUSHS」命令で記憶した演算結果を読み出し、その内容で次のステップから演算を続行します。
POPS	「PUSHS」命令で記憶した演算結果を読み出し、その内容で次のステップから演算を続行、「PUSHS」命令で記憶した演算結果をリセットします。

- 1つの演算結果を記憶し、読み出して複数の処理を行います。
- この命令は、ひとつの接点から分岐して、その後にさらに接点を接続する場合に使用します。

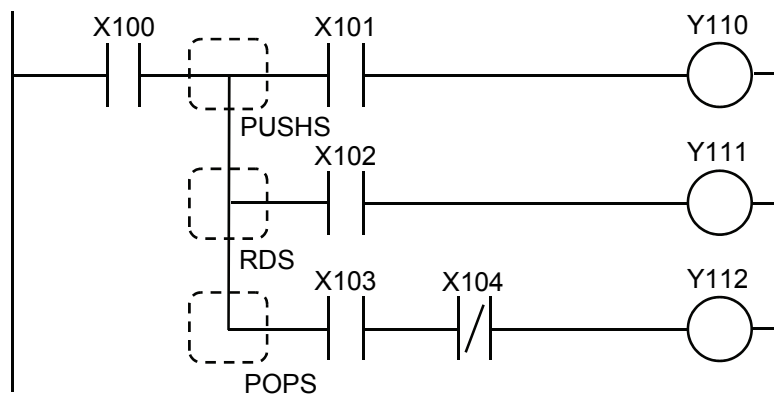
## ■ 動作例

- 1) X100 が ON のとき、「PUSH」命令で演算結果を記憶し、X101 が ON なら Y110 に出力します。
- 2) 「RDS」命令で演算結果を読み出し、X102 が ON なら Y111 に出力します。
- 3) 「POP」命令で演算結果を読み出し、X103 が OFF なら Y112 に出力し、「PUSH」命令で記憶した演算結果をリセットします。

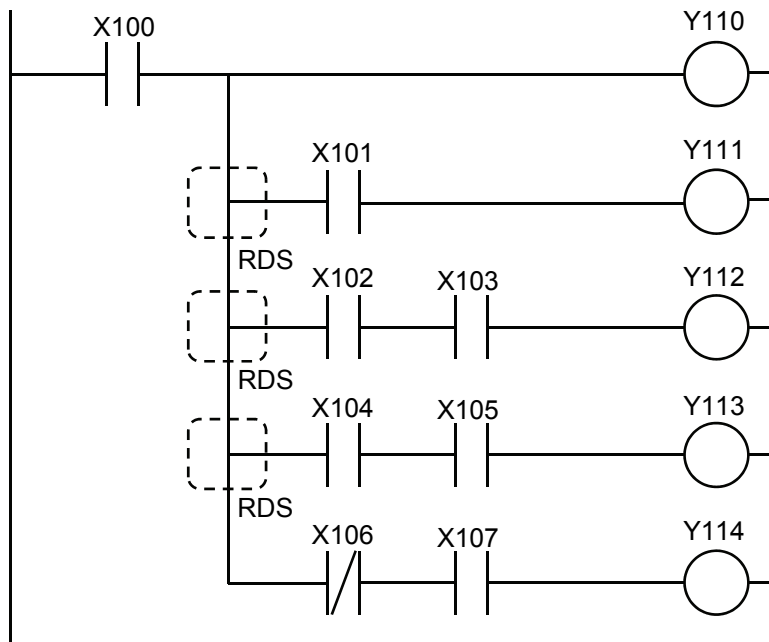


## ■ プログラム上のご注意

- 続けて演算結果を使用する場合は「RDS」命令、終わらせる場合は「POPS」命令を使います。（「POPS」命令は必ず入れてください。）

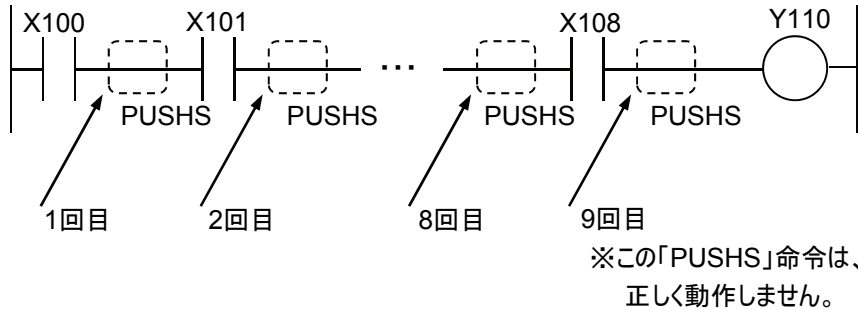


- 「RDS」命令は続けて何回でも使用できます。

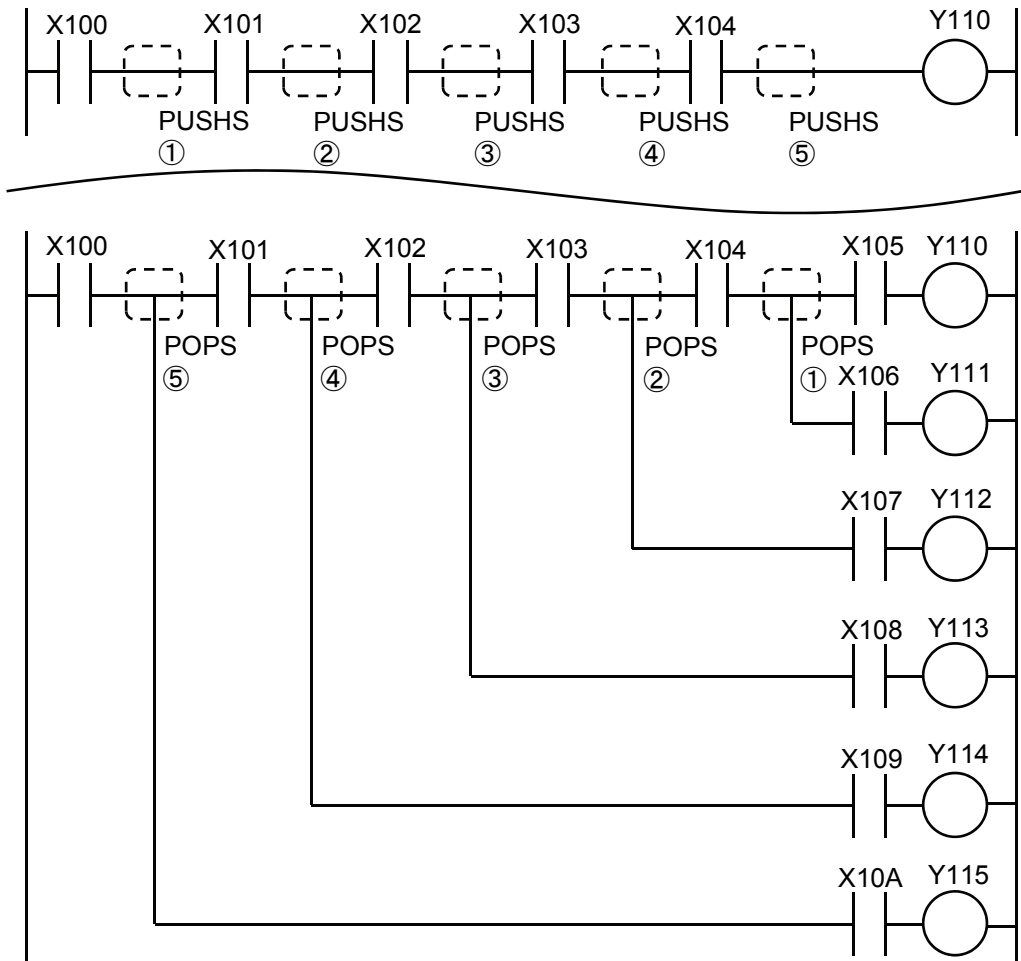


## ■ 「PUSHS」命令を連続して使用する時のご注意

- 「PUSHS」命令は、連続して記述する回数に制限はあります。次に「POP」命令を記述するまで連続して使用できる回数は最大 8 回までとなります。
- 制限回数よりも多く連続して使用すると、正しく動作しませんので、ご注意ください。



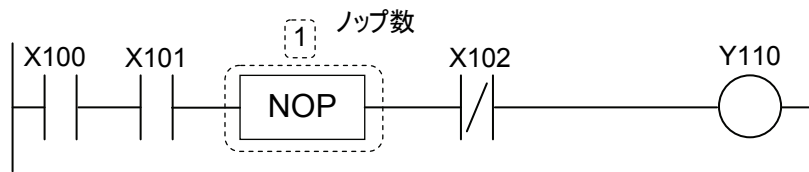
- 「PUSHS」命令を連続して使用している状態で「POPS」命令を使用すると、最後に「PUSHS」命令で記憶した内容から順に読み出して行きます。図内の番号が、それぞれの命令の内容に対応した番号になります。





# NOP (ノップ)

## ■ ラダー表記



## ■ 動作説明

- それまでの演算結果には何も影響を与えません。「NOP」命令がなくても同じ動作を行います。
- 「NOP」命令は、プログラムの見直しや修正の際にプログラムを見やすくするために利用できます。
- アドレスを変えずに命令を消去したいとき、「NOP」命令を書き込んで(上書きして)ください。
- プログラムを変えずに、ある部分のアドレスを送りたいとき、「NOP」命令を挿入してください。
- 例えば、長いプログラムをいくつかのブロックごとに分けておくときなどに便利です。

## ■ サンプルプログラム

アドレス		アドレス	
0	ST X100	0	ST X100
1	AN X101	1	AN X101
2	AN/ X102	2	NOP
-----			
3	OT Y100	3	AN/ X102
		4	OT Y100

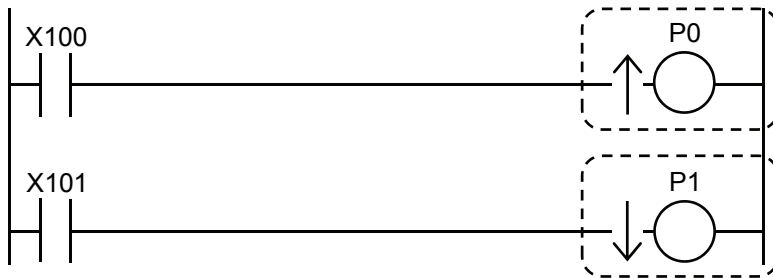
←ここにNOP命令が入り、アドレスが変化する。

## ■ 「NOP」命令の削除

- プログラム作成後、プログラミングツールを使ってプログラム中の「NOP」命令をすべて削除できます。

# ↑ OT, ↓ OT (立ち上がり、立ち下がり検出アウト)

## ■ ラダー表記



## ■ 指定可能なデバイス (●: 指定可能)

オペランド	ビットデバイス											ワードデバイスのビット指定		インデックス修飾
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b	LD.b	
bit							●							●

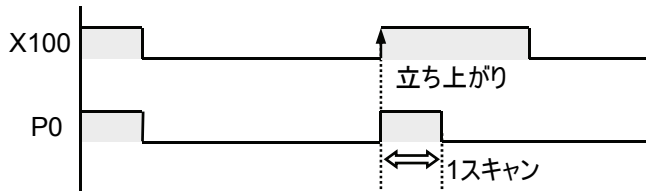
## ■ 動作説明

命令の種類	動作
↑ OT	直前までの演算結果が OFF 状態から ON 状態に変化(立ち上がり)したその 1 スキャンのみ出力します。
↓ OT	直前までの演算結果が ON 状態から OFF 状態に変化(立ち下がり)したその 1 スキャンのみ出力します。

## ■ 動作例

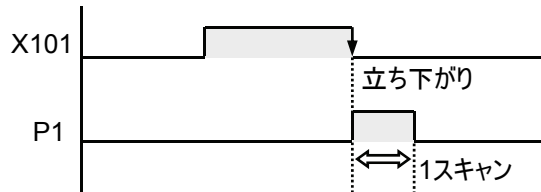
### (1) ラダー表記にある「↑OT」のプログラム動作

X100がOFF状態からON状態に変化した(立ち上がり)1スキャンのみパルスリレーP0に出力します。  
第1スキャンよりONしているときもP0に出力します。



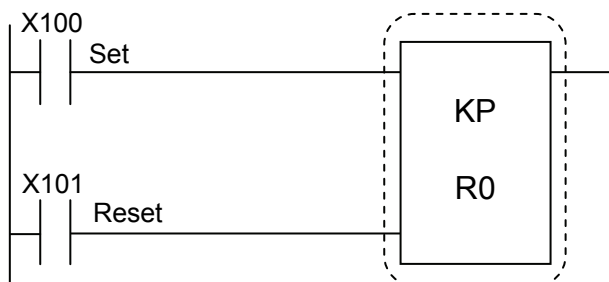
### (2) ラダー表記にある「↓OT」のプログラム動作

X101がON状態からOFF状態に変化した(立ち下がり)1スキャンのみパルスリレーP1に出力します。



# KP (キープ)

## ■ ラダー表記



## ■ 指定可能なデバイス (●:指定可能)

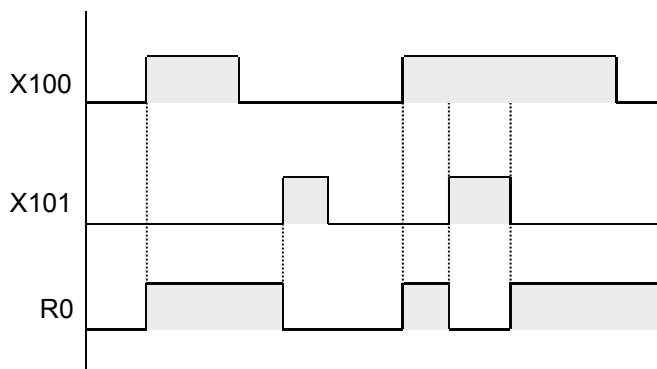
オペランド	ビットデバイス										ワードデバイスのビット指定		インデックス修飾	
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b		LD.b
bit	●	●	●	●				●			●	●	●	●

## ■ 動作説明

- セット入力(X100)が ON すると、指定したコイルの出力を ON にし、その状態を保持します。リセット入力(X101)が ON すると、保持を解除します。
- 保持中は、セット入力(X100)の ON/OFF にかかわらず、リセット入力(X101)があるまで、出力を保持します。
- セット入力(X100)と、リセット入力(X101)が同時に ON した場合、リセット入力(X101)が優先されます。

## ■ 動作例

- 1) X100がONすると、指定したコイル:R0の出力をONにし、その状態を保持します。
- 2) X101がONすると、保持を解除します。

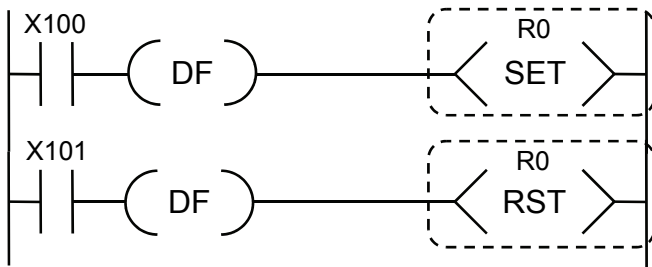


## ■ プログラム上のご注意

- 「MC」命令の動作中であっても、出力先はその状態を保持します。
- 「RUNモード」から「PROG.モード」への切り替え時や、電源OFF時には、リセットされます。ただし、保持型に設定した内部リレーを出力先に指定した場合を除きます。

# SET, RST (セット、リセット)

## ■ ラダー表記



## ■ 指定可能なデバイス (●:指定可能)

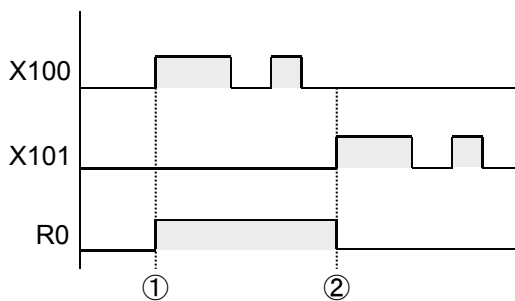
オペランド	ビットデバイス											ワードデバイスのビット指定		インデックス修飾
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b	LD.b	
bit	●	●	●	●				●			●	●	●	●

## ■ 動作説明

命令の種類	動作
SET	実行条件が ON のときに、出力を ON し、実行条件の状態変化にかかわらず、状態を保持します。
RST	実行条件が ON のときに、出力コイルを OFF し、実行条件の状態変化にかかわらず、OFF の状態を保持します。

## ■ 動作例

- 1) X100がONになると、R0をONし、その状態を保持します。
- 2) X101がONになると、R1をOFFし、その状態を保持します。

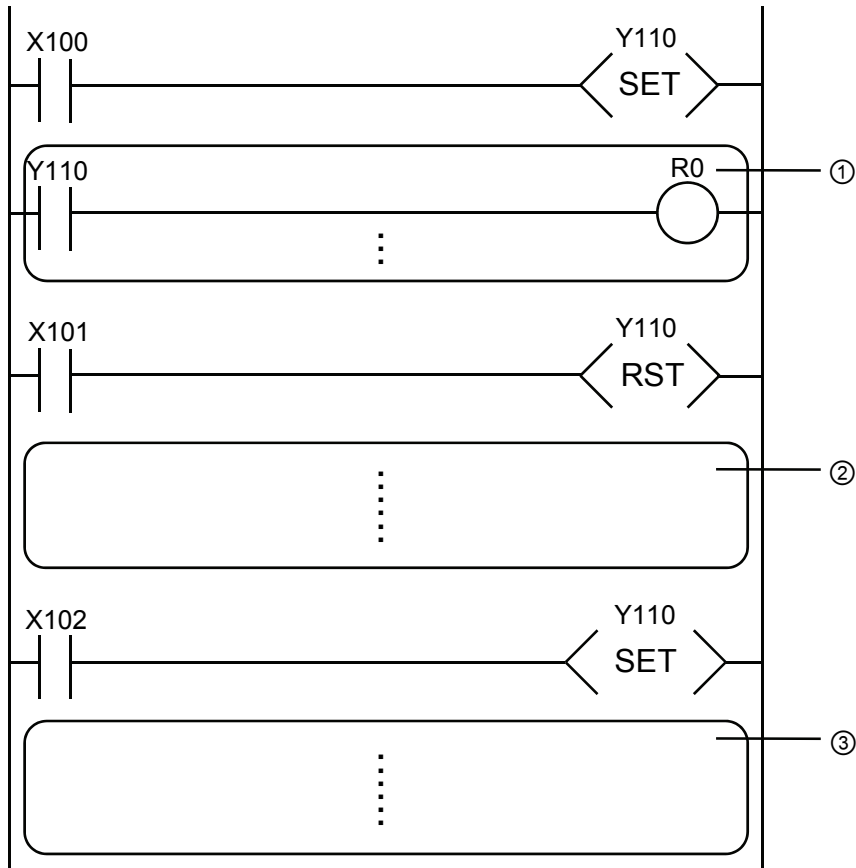


## ■ プログラム上のご注意

- 「SET」、「RST」命令の出力先には、同一の出力コイルを何度でも指定できます。
- 「SET」、「RST」命令での各種リレーの使用は 2 重出力にはなりません。トータルチェックをかけても文法エラー扱いにはなりません。
- 「RST」命令を使用することにより、リレーをオフすることができます。
- 「MC」命令の動作中であっても、「SET」命令の出力先はその状態を保持します。
- 「RUN モード」から「PROG.モード」への切り替え時や電源 OFF 時には、「SET」命令の出力先はリセットされます。ただし、保持型に設定した内部リレーを出力先に指定した場合を除きます。
- 「SET」、「RST」命令の出力先として、パルスリレー P を指定することはできません。
- RST SD60 にて、異常報知バッファの全クリアができます。
- RST SD61 にて、異常報知バッファの先頭がクリアできます。

## ■ 「SET」、「RST」命令使用時の処理のしくみ

- 演算処理中は、ステップ毎に出力の内容が書き換わります。
- I/O リフレッシュは、「ED」命令実行時に行われますので、実際に出力されるデータは、最終的な演算結果で決まります。
- 途中の演算結果を出力したい場合は、直接出力(OT)を使用してください。



上記プログラムで X100～X102 がすべて ON のとき、

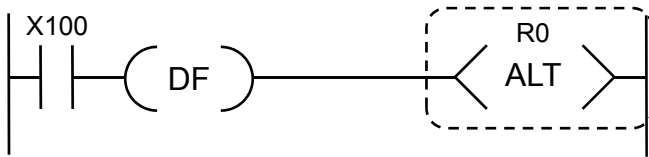
- 1) Y110 は ON であるものとして、処理します。
- 2) Y110 は OFF であるものとして、処理します。
- 3) Y110 は ON であるものとして、処理します。

## ■ 「SET」、「RST」命令は微分命令とセットで

- 「SET」、「RST」命令の前に「微分 DF」命令を入れるとプログラム作成・調整が楽になります。
- 特に、同じ出力先をプログラム中で何ヶ所も使う場合に有効です。

# ALT (オルタネートアウト)

## ■ ラダー表記



## ■ 指定可能なデバイス (●: 指定可能)

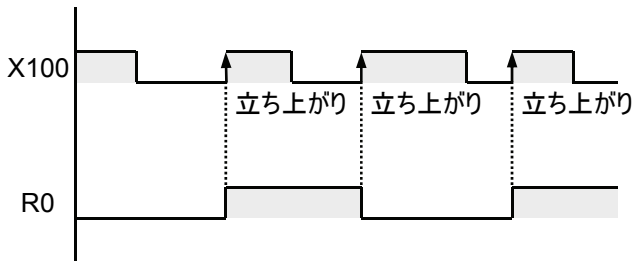
オペランド	ビットデバイス											ワードデバイスのビット指定		インデックス修飾
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b	LD.b	
bit	●	●	●	●				●				●	●	

## ■ 動作説明

- 直前までの演算結果が OFF 状態から ON 状態に変化(立ち上がり)すると、指定したコイルの ON/OFF を反転します。
- 指定したコイルの ON/OFF 状態は、そのコイルを指定している「ALT」命令が次に立ち上がるまで保持されます。(フリップフロップ制御)

## ■ 動作例

X100がOFF状態からON状態に変化(立ち上がり)するたびに、出力R0のON/OFF状態が反転します。



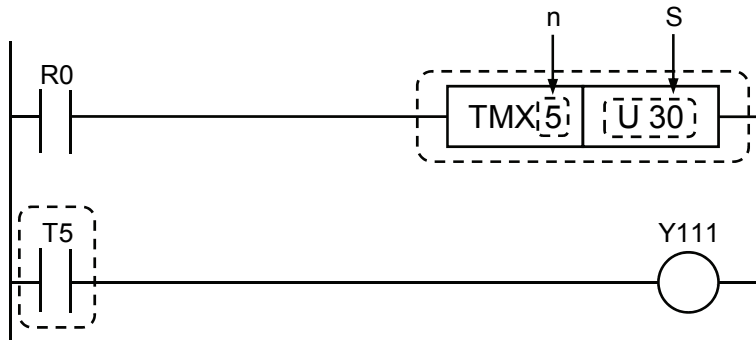
## ■ プログラム上のご注意

- 「ALT」命令では、入力の OFF→ON の立ち上がりを検出して、出力を反転します。
- 入力 ON しつづけている間は、立ち上がり時のみ反転して、以後は反転しません。
- RUN に切り替えた時や RUN モードで電源を投入したときに入力が最初から ON している場合には最初のスキャンでは反転が行われません。
- MC～MCE 命令、JP～LBL 命令など命令を実行する順序を変える命令(下記 1～6)と合わせて使用する場合、命令の実行と入力のタイミングにより、命令の動作が変わりますので、ご注意ください。

- 1) MC～MCE 命令
- 2) JP～LBL 命令
- 3) LOOP～LBL 命令
- 4) CNDE 命令
- 5) ステップラダー命令
- 6) サブルーチン命令

# TM (タイマ)

## ■ ラダー表記



## ■ オペランド一覧

オペランド	説明
n	タイマ番号
S	タイマ設定値

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS *1	TE CE	IX	K	U	H	SF	DF	" "	
n						●										●					●
S	●	●	●			●	●	●				●				●					●

\*1: TM 命令の場合、TS のみ指定可能

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

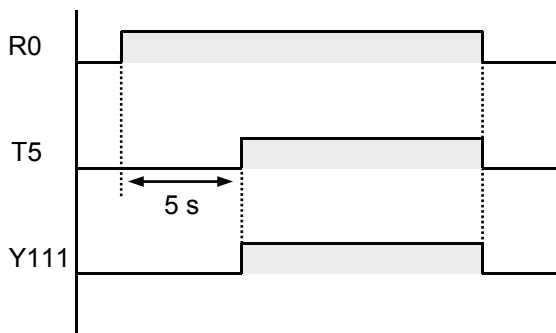
- タイマは、電源を切ったり、RUN モード→PROG.モードへ切り替えたりするとリセットされる非保持型です。
- 実行条件が ON のとき、設定時間[S]を減算動作し、経過値が 0 になるとタイマ接点[Tn](nはタイマ接点番号)が ON になります。
- 減算動作中に実行条件が OFF すると、動作を中断し経過値をリセット(0 クリア)します。
- タイマコイルのすぐ後に OT 命令を記述することもできます。

## ■ タイマ時間の設定について

- タイマの設定時間は、(タイマ単位) × (タイマ設定値)となります。
- タイマ設定値[S]は、U1～U4294967295 の範囲で、10 進数定数を使用して設定します。

「TMS」は、0.00001 秒単位で 0.00001～42949.67295 秒。
「TML」は、0.001 秒単位で 0.001～4294967.295 秒。
「TMR」は、0.01 秒単位で 0.01～42949672.95 秒。
「TMX」は、0.1 秒単位で 0.1～429496729.5 秒。
「TMY」は、1 秒単位で 1～4294967295 秒。

## ■ 動作例



## ■ プログラム上のご注意

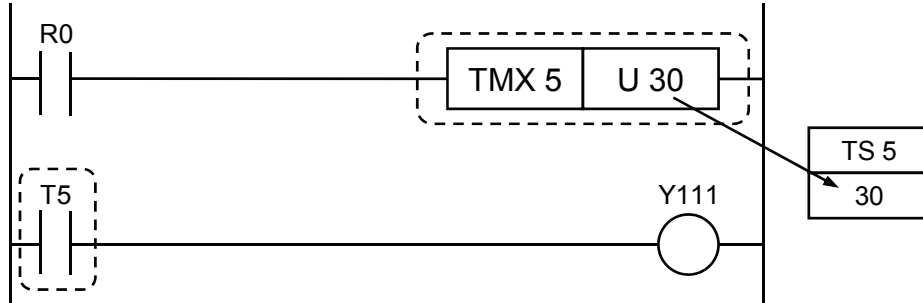
- タイマ設定値エリア TS、タイマ経過値エリア TE は、いずれも 32 ビットのエリアを占有します。オペランド[S]に DT などのデバイスを使用する場合も同様です。他のプログラムで上書きすることがないようにしてください。
- 減算動作は演算時に行いますので、1 スキャンタイム中に 1 回減算するようにプログラムを作成してください。割り込み処理プログラム中やジャンプ/ループ命令などにより、1 スキャン中に複数回演算したとき、あるいは、一度も減算できなかつたときは、正しい結果が得られません。
- タイマ命令を、アンドスタック命令やポップスタック命令と組み合わせるときは、正しくない記述にならないようご注意ください。



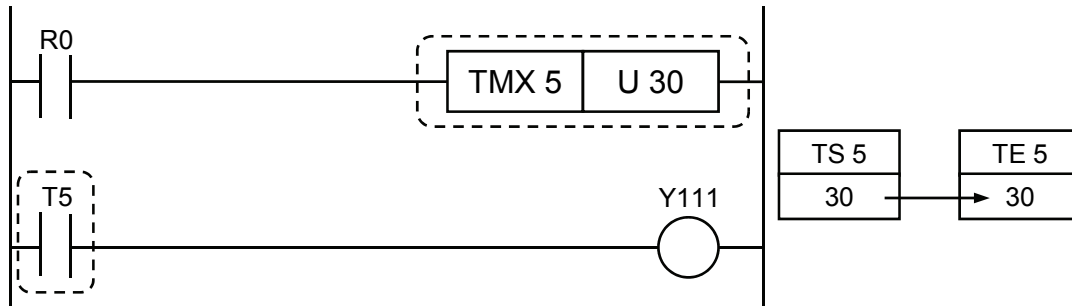
## ■ タイマ動作のしくみ

• 設定値を U 定数で指定したときの例です。設定値エリア No.を指定するときの動作は、次ページをご参照ください。

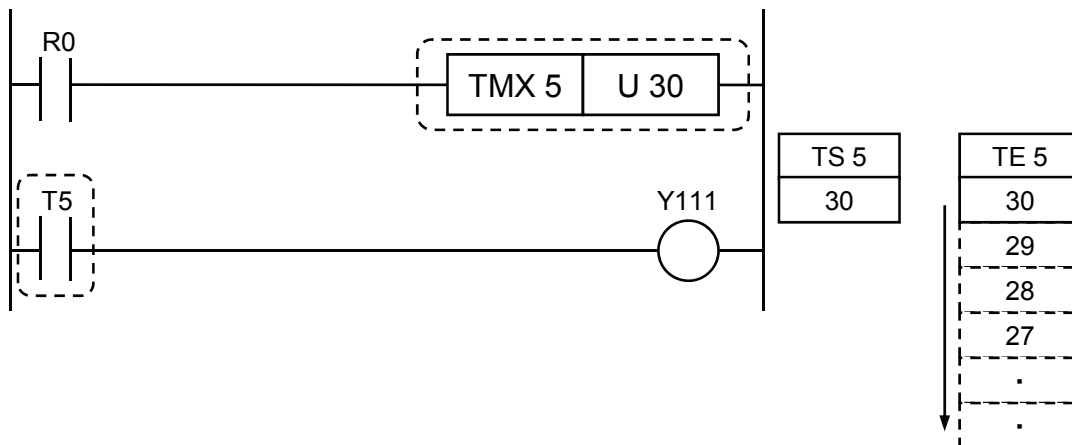
1) RUN モードに切り替えたとき、あるいは、RUN モードで電源を ON にしたとき、タイマ設定値が同じ番号の設定値エリア「TS」に転送されます。



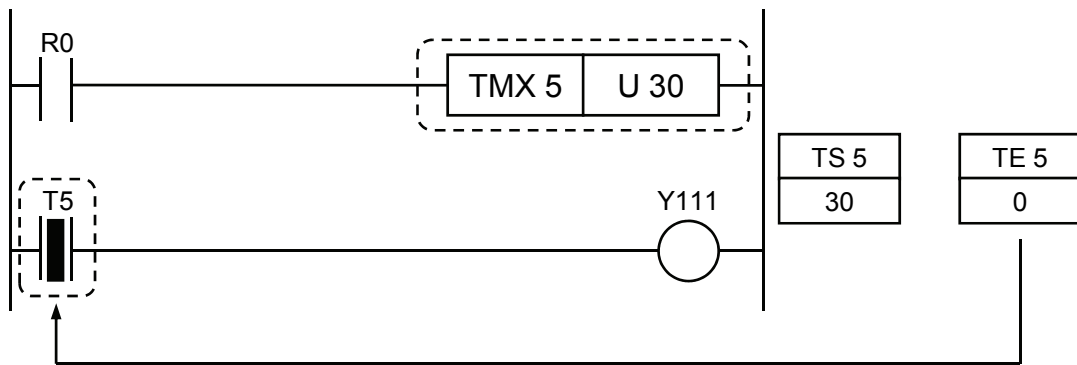
2) タイマ実行条件の OFF→ON の立ち上がりで、設定値エリア「TS」から同じ番号の経過値エリア「TE」へ転送します。  
(実行条件が ON 状態で、RUN モードに切り替えたときにも同様の動作となります。)



3) スキャンごとに、実行条件が ON ならば経過値エリア「TE」の値を減算します。

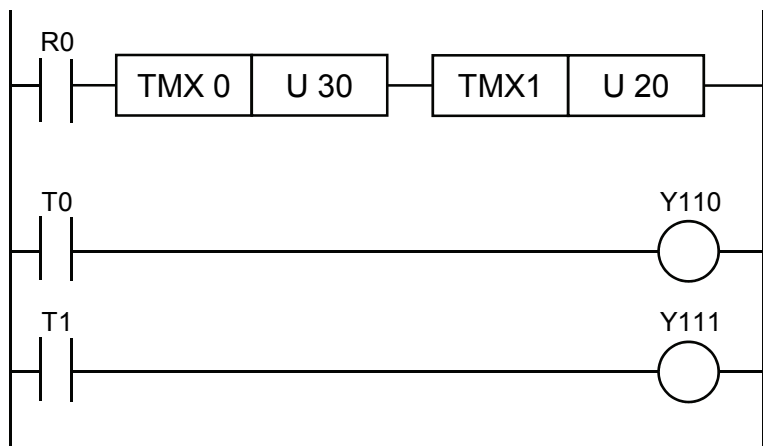


4) 経過値エリア「TE」の値が 0 になれば、同じ番号のタイマ接点「T」が ON になります。

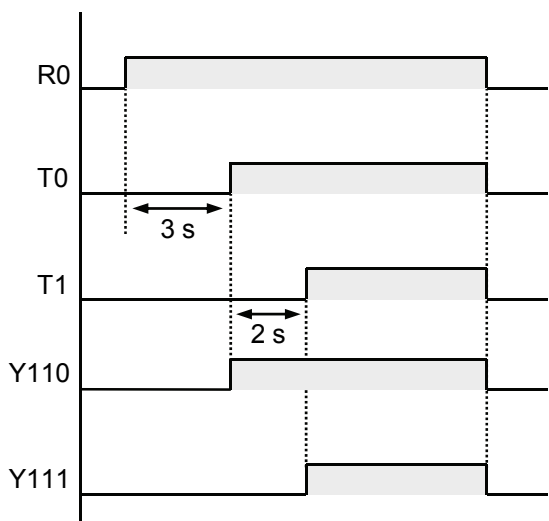


■ タイマ命令の応用例<タイマの直列接続>

●ラダー表記

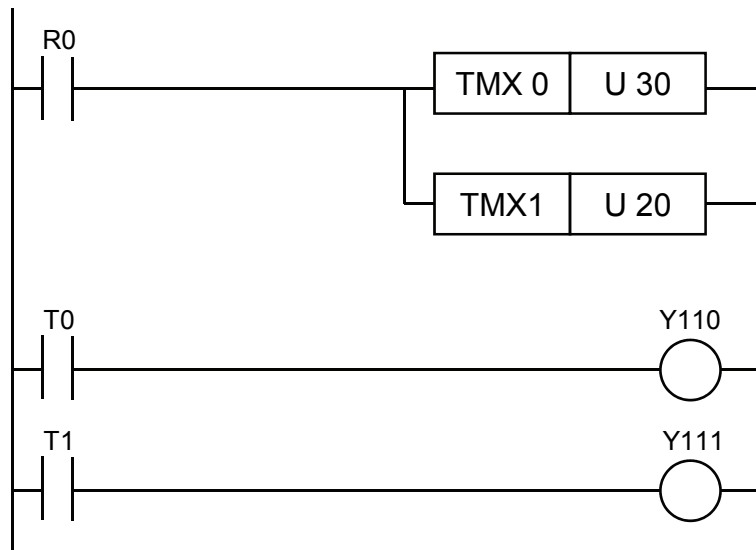


●タイムチャート

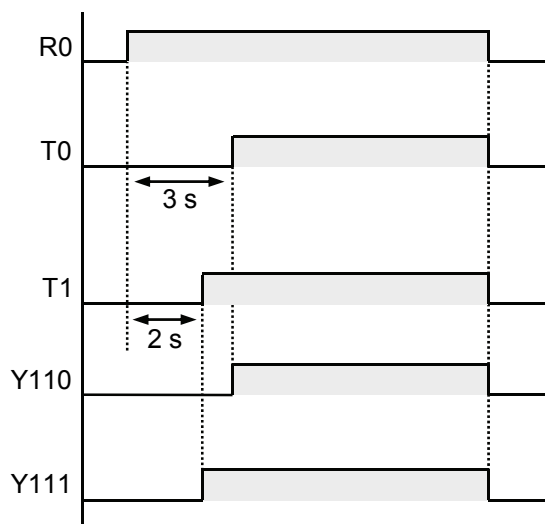


## ■ タイマ命令の応用例<タイマの並列接続>

### ●ラダー表記

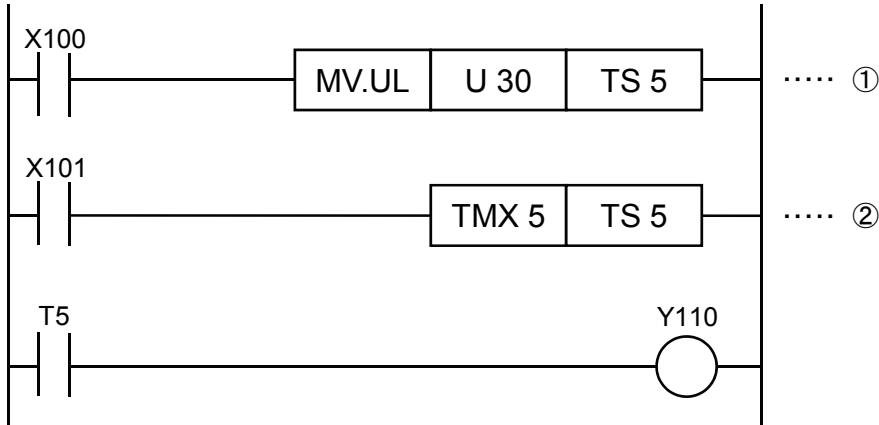


### ●タイムチャート



## ■ タイマ設定値に設定値エリアNo.を直接指定する方法について

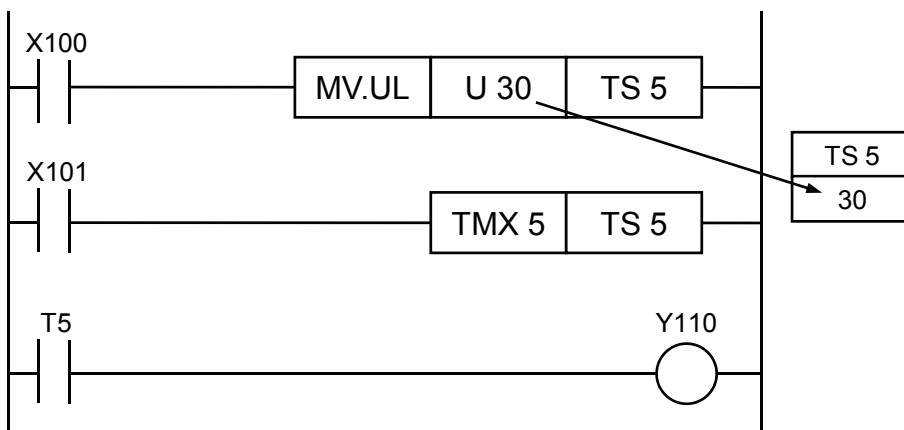
- 以下のプログラムの場合、タイマ設定値を設定値エリアに直接設定しているパターンです。
  - [n]に指定する設定エリア「TS」の番号はタイマの番号と同じにしてください。
  - 設定値に TS5 を指定している上記プログラムは次のように動作します。
- 1) 実行条件 X100 が ON のとき、データ転送命令「MV」を実行して、TS5 に U30 として、減算動作を開始します。
  - 2) 実行条件 X101 が ON になると、設定値を 30 として、減算動作を開始します。



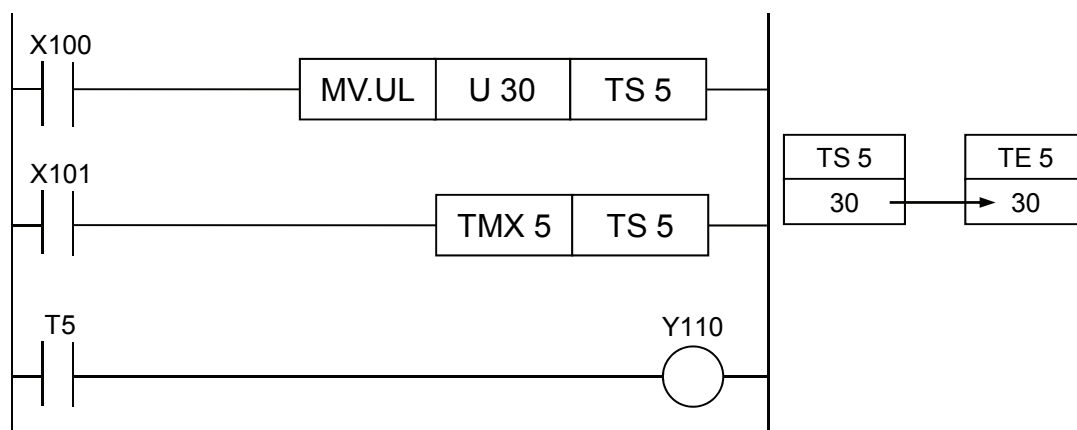
- 減算動作中に、設定値エリア「TS」の値を変更しても、変更前の設定で減算動作を続行します。
- 変更した値でタイマ動作が開始されるのは、減算動作が完了または中断したあとで、次に実行条件が OFF から ON になったときです。
- 設定エリア「TS」は、通常、電源を切ったり、RUN モード→PROG.モードに切り替えたときにリセットされる非保持型になっています。

## ■ 設定値エリアNo.を直接指定したときのタイマ動作のしくみ

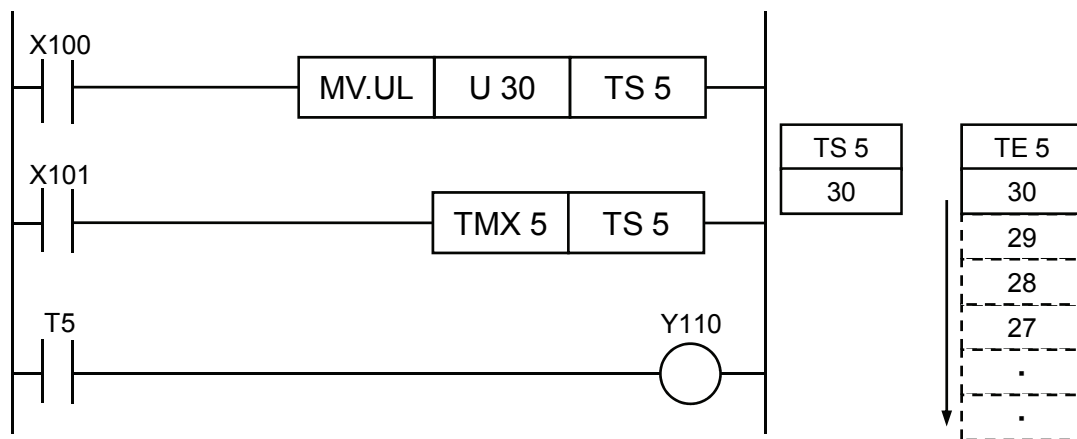
- 1) 応用命令の実行条件が ON のとき、設定値エリア「TS」に設定をセットします。下図は、「MV」命令を使用する例です。



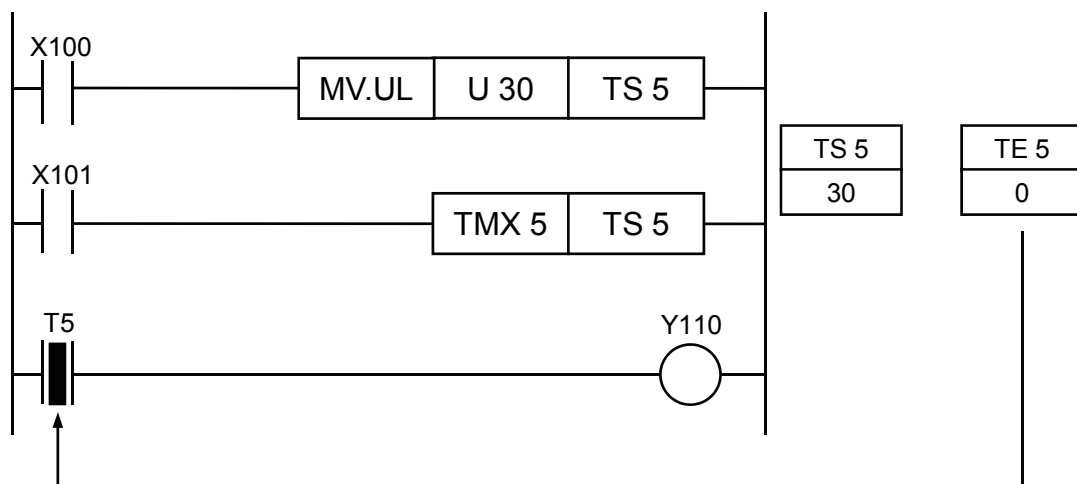
2) タイマ実行条件の OFF→ON の立ち上がりで、設定値エリア「TS」から同じ番号の経過値エリア「TE」へ転送されます。(実行条件が ON の状態で、RUN モードに切り替えたときにも同様の動作 となります。)



3) スキャンごとに、実行条件が ON ならば、経過値エリア「TE」の値を減算します。



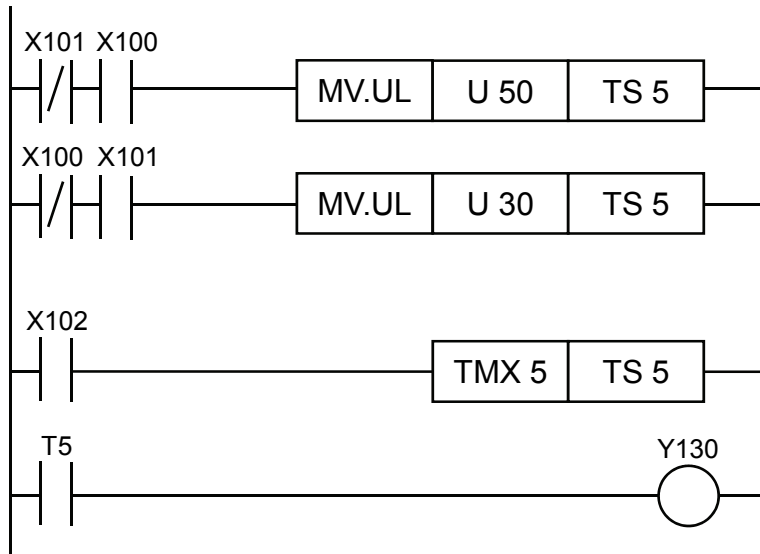
4) 経過値エリア「TE」の値が 0 になれば、同じ番号のタイマ接点「T」が ON になります。



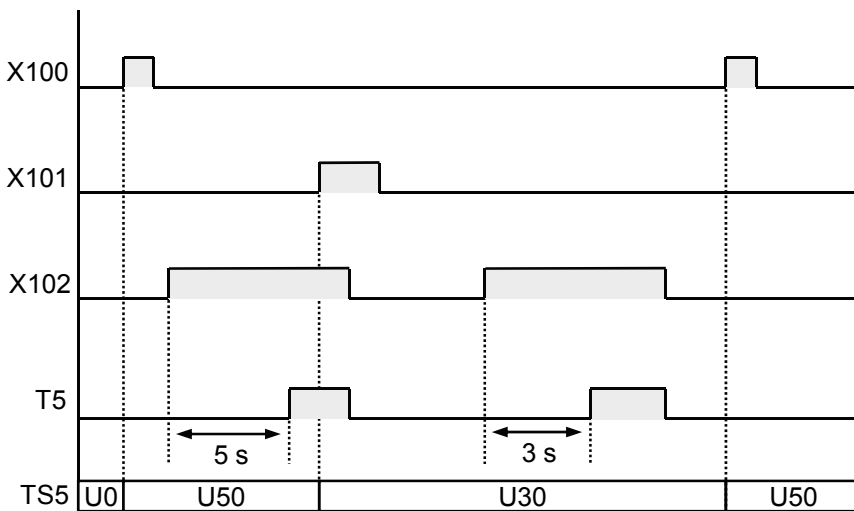
## ■ 設定値エリアNo.を直接指定する場合の応用例

<例>条件に応じて、設定値を切り替える例

### ●ラダー表記

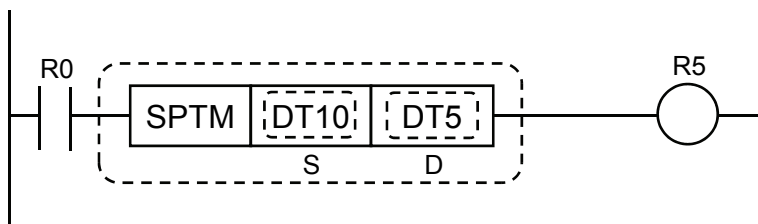


### ●タイムチャート



# SPTM (符号なし 32 ビット加算式補助タイマ)

## ■ ラダー表記



## ■ オペランド一覧

オペランド	説明
S	タイマ設定値
D	タイマ経過値

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *3
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS *1	TE CE *2	IX	K	U	H	SF	DF	""	
S	●	●	●	●			●	●	●			●				●	●				●
D	●	●	●	●			●	●	●				●								●

\*1: 第1オペランド[S]でCS指定不可

\*2: 第2オペランド[D]でCE指定不可

\*3:16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- 0.01秒単位のオンディレータイマとして動作します。内部リレーがONのとき、設定時間を減算動作し、経過値[D]が0になると、システムリレーSRDがONになります。(内部リレーOFF時および減算中はOFFしています。)
- 内部リレーOFF時、経過値エリアは0クリアされます。OT命令で使用しているリレーはOFFになります。
- システムリレーSRDもタイムアップするとONになります。システムリレーSRDをタイマ接点として使用することもできます。(内部リレーOFF時および減算中はOFFしています。)

## ■ タイマ時間の設定について

- タイマ時間は、 $0.01 \times (\text{タイマ設定値})$ となります。
- タイマ設定値は、U1(H1)~U4294967295(H7FFFFFFF)の範囲でU定数で設定します。

「SPTM」は、0.01秒単位で0.01~42949672.95秒。

例) 設定値がU500のとき、設定時間は、 $0.01 \times 500 = 5$ 秒となります。

## ■ 動作例

設定値[S]: DT10の値がU500として実行した場合

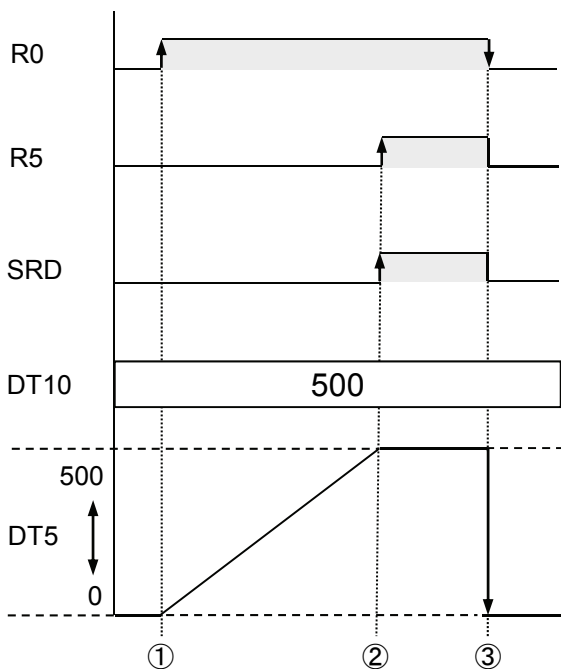
1) R0がONでタイマ動作を開始します。

経過値エリア: DT5に“0”を転送します

2) 経過値エリア: DT5が設定値エリア: DT10の値(U500)まで到達すると、

システムリレーSRDと出カコイルR0がONします。

3) R0がOFFになると、タイマ動作を中断し、経過値エリア: DT5に“0”を転送します。



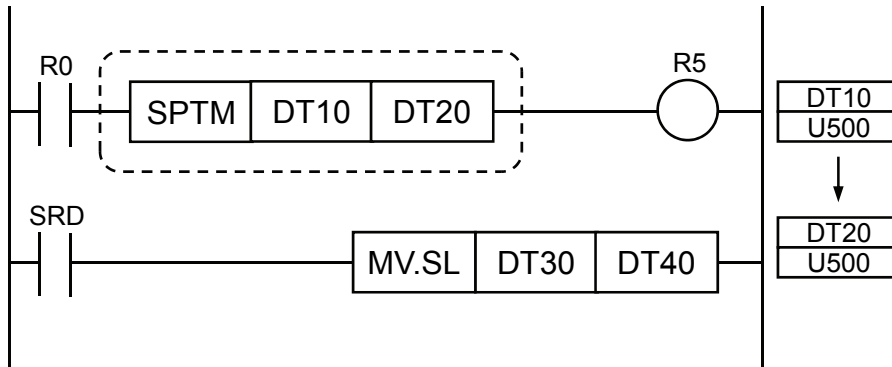
## ■ プログラム作成上のご注意

- 設定値を格納しているエリア、経過値エリアの指定は他のタイマ／カウンタ命令や応用命令の演算用メモリエアと重ならないようにしてください。
- 減算動作は演算時に行いますので、1 スキャンタイム中に 1 回減算するようにプログラムを作成してください。割り込み処理プログラム中やジャンプ/ループ命令などにより、1 スキャン中に複数回演算したとき、あるいは、一度も減算できなかったときは、正しい結果が得られません。

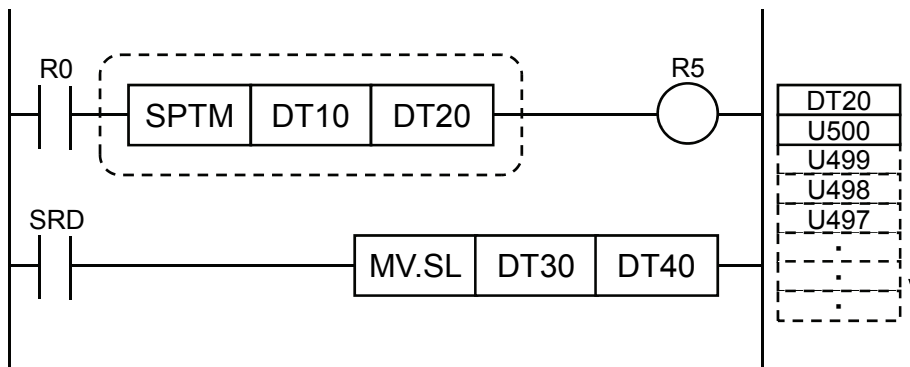


## ■ 補助タイマ動作のしくみ

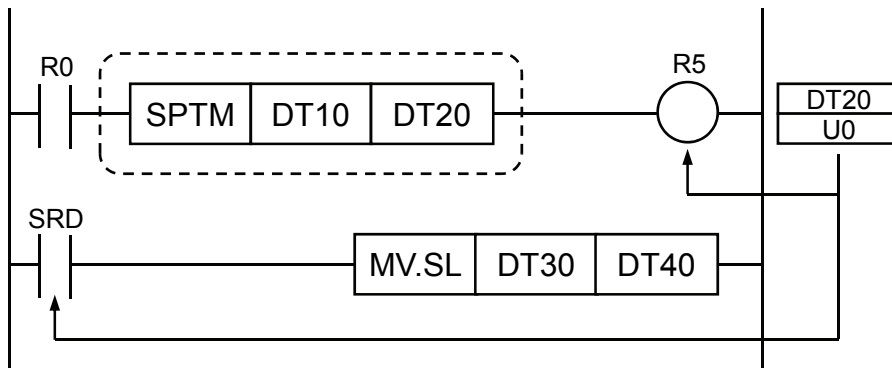
1) 内部リレーが OFF→ON になると、[S]で指定している設定値が経過値エリア[D]に転送されます。



2) スキャンごとに、内部リレーが ON ならば、経過値エリア[D]の値を減算します。



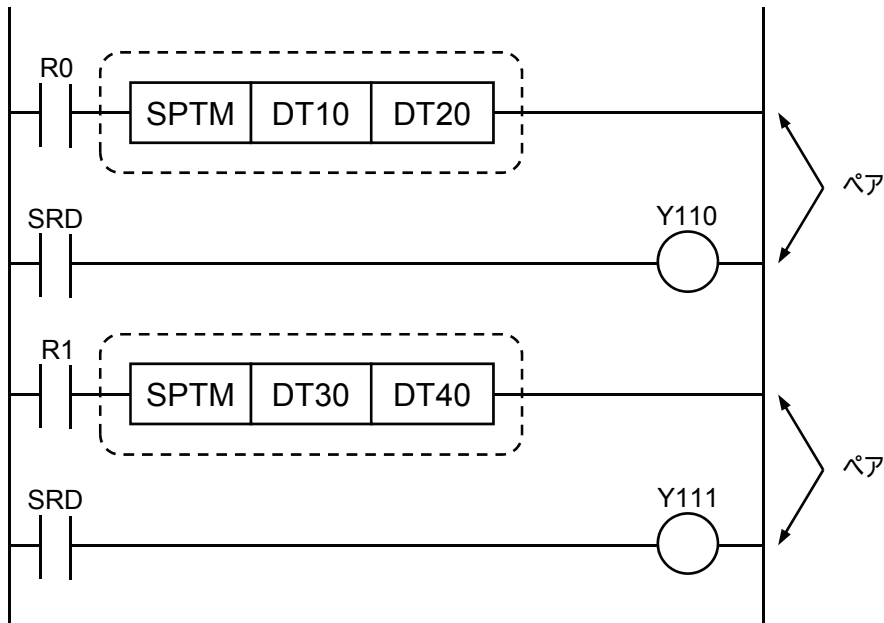
3) 経過値エリア[D]の値が 0 になれば、つづけて記述している OT 命令で使用しているリレーが ON になります。システムリレー SRD も ON になります。



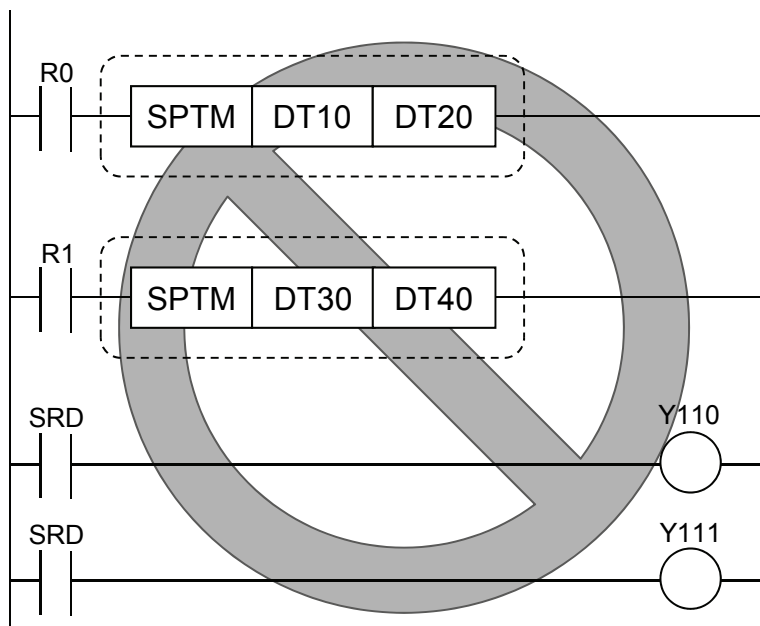
■ システムリレーSRD使用時のご注意

- SRD を使用して複数の補助タイマを使用するときは、SRD を必ず補助タイマ命令の次の行で使用してください。

<例>

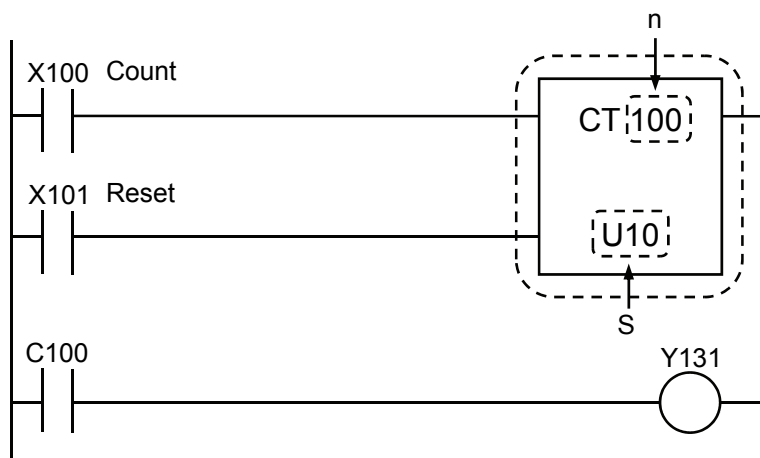


- 下記のように記述すると正しい動作が得られません。



# CT (ダウンカウンタ)

## ■ ラダー表記



## ■ オペランド一覧

オペランド	説明
n	カウンタ番号
S	カウンタ設定値

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS *1	TE CE	IX	K	U	H	SF	DF	" "		
n																●						●
S	●	●	●	●			●	●				●				●						●

\*1: CT 命令の場合、CS のみ指定可能

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- カウンタはすべて減算式プリセットカウンタです。
- リセット入力の ON→OFF への立ち上がり時に、設定値エリア「CS」の値を経過値エリア「CE」へプリセットします。
- リセット入力 ON のとき、経過値をリセット状態(0)にします。
- カウント入力 OFF→ON に変化したとき、設定値を減算動作し、経過値が 0 になると、カウンタ接点に出力します。
- カウント入力とリセット入力が同時に ON した場合、リセット入力が優先されます。
- カウント入力の立ち上がりとリセット入力の立ち下がりが同時の場合は、カウント入力は無視されて、プリセットのみ実行されます。
- カウンタ命令のすぐ後に OT 命令を記述することもできます。

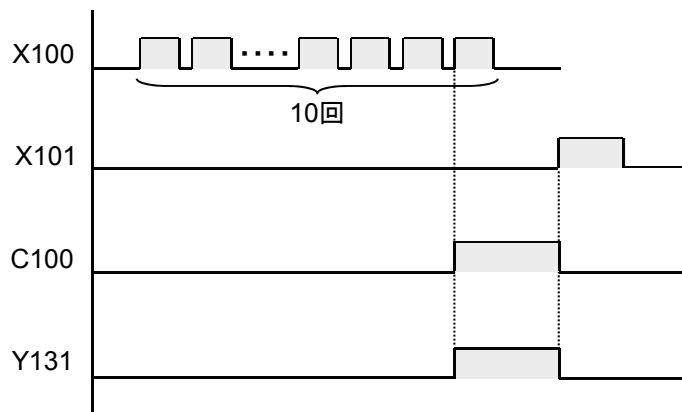
## ■ カウント値の設定について

- カウント値の設定範囲は U1~4294967295 で、10 進数(U 定数)で設定します。

## ■ 動作例

プログラムに対し、カウンタ番号[n]=100、設定値[S]=10とした場合

- 1) X100が10回ONすると、C100がONし、Y131がONします。
- 2) X101がONすると、経過値をリセットします。

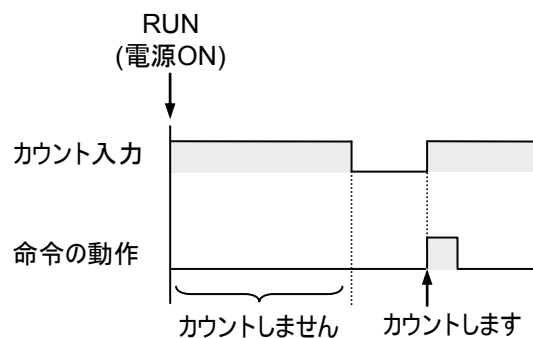


## ■ プログラム上のご注意

- カウンタ設定値エリア CS、カウンタ経過値エリア CE は、いずれも 32 ビットのエリアを占有します。オペランド[S]に DT などのデバイスを使用する場合も同様です。他のプログラムで上書きすることがないようにしてください。
- カウンタ命令をアンドスタック命令やポップスタック命令と組み合わせるときは、正しくない記述にならないようにご注意ください。

## ■ カウント入力検出についてのご注意

- CT 命令では、カウント入力の OFF→ON の立ち上がりを検出して、減算します。
- カウント入力 が ON しつづけている間は、立ち上がり時のみカウントして、以後はカウントしません。
- RUN に切り替えたときや RUN モードで電源を投入したときにカウント入力 が最初から ON している場合は、最初のスキャンでは減算動作が行われません。



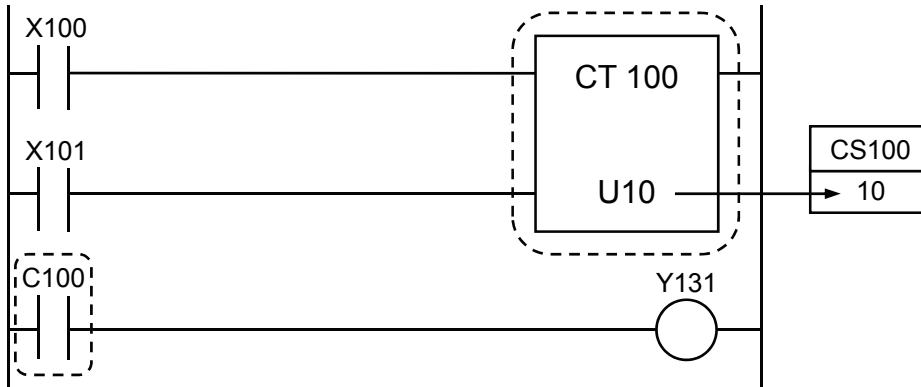
- MC~MCE 命令、JP~LBL 命令など命令を実行する順序を変える命令(下記 1~6)と合わせて使用する場合、命令の実行とカウント入力のタイミングにより、命令の動作が変わりますのでご注意ください。

- 1) MC~MCE 命令
- 2) JP~LBL 命令
- 3) LOOP~LBL 命令
- 4) CNDE 命令
- 5) ステップラダー命令
- 6) サブルーチン命令

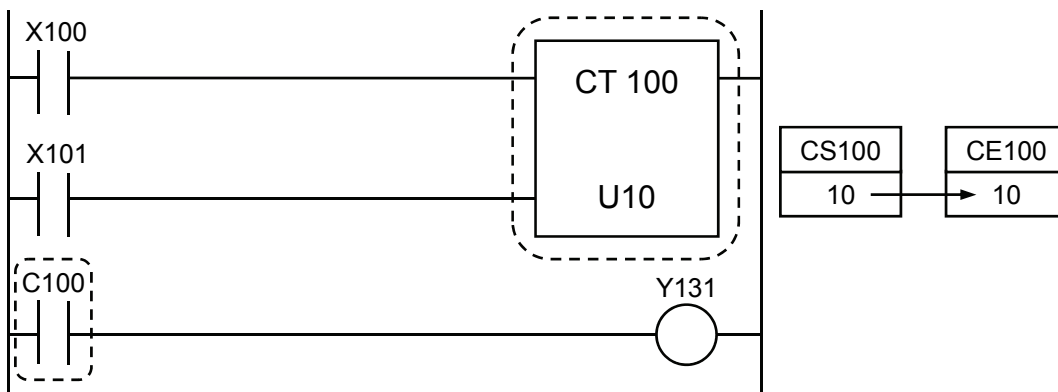
## ■ ダウンカウンタ動作のしくみ

● 設定値を U 定数で指定したときの例です。設定値エリア No.を指定するときの動作は、次ページをご参照ください。  
(100 をカウンタに指定しているときの例です。)

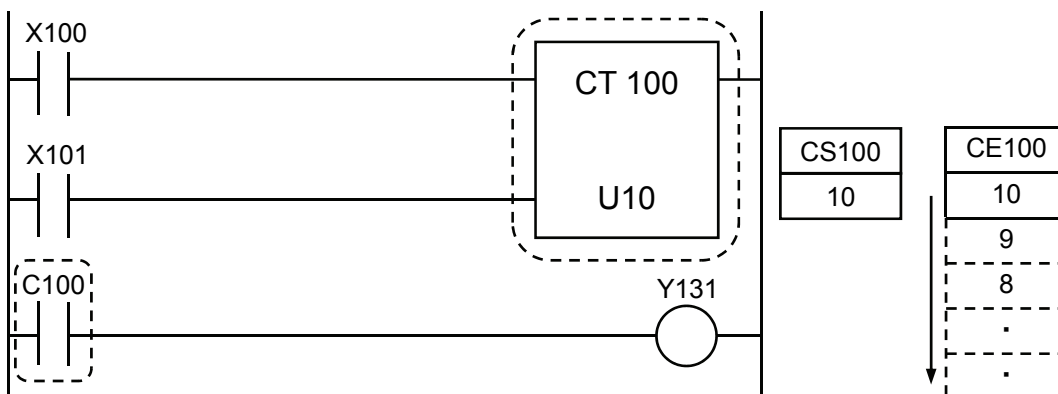
1) RUN モードに切り替えたとき、または、RUN モードで電源を ON にしたとき、カウンタ設定値が同じ番号の設定値エリア「CS」に転送されます。



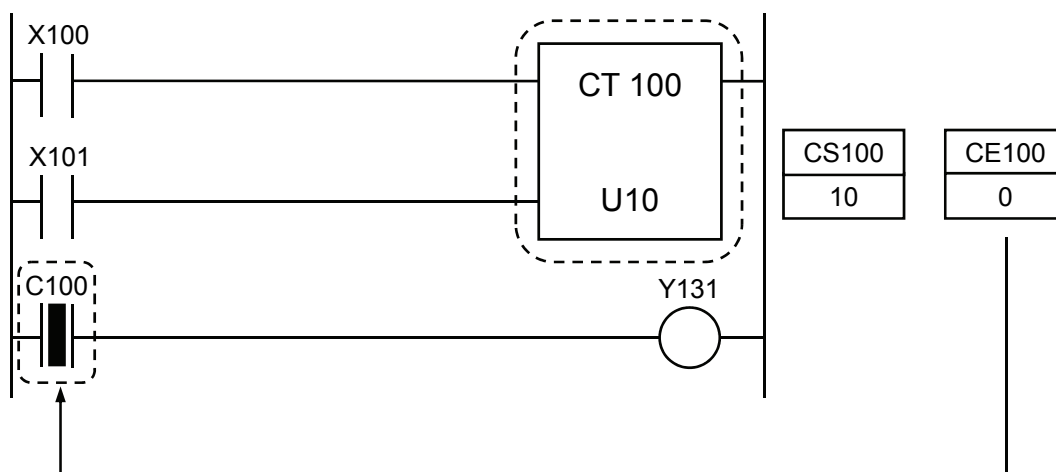
2) リセット入力の立ち下がりにて、設定値エリア「CS」の値が経過値エリア「CE」にプリセットされます。



3) カウント入力 X100 が ON するごとに経過値エリア「CE」の値を減算します。



4) 経過値エリア「CE」の値が 0 になれば、同じ番号のカウンタ接点「C」が ON になります。

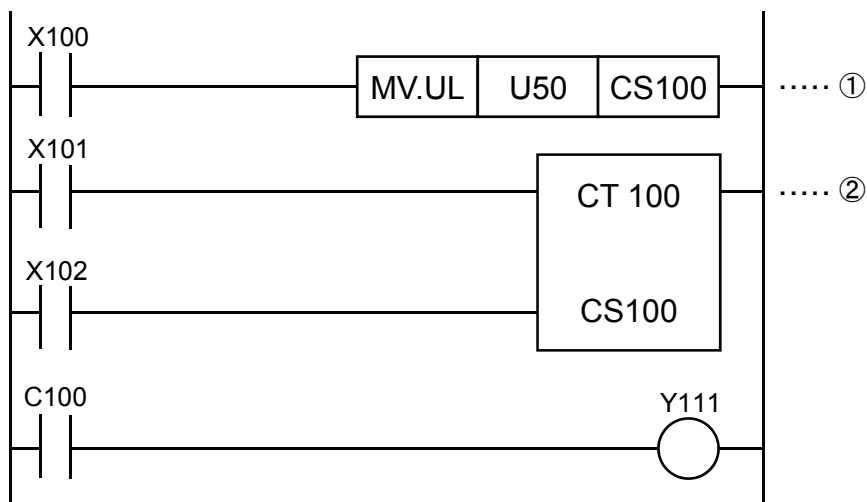


## ■ 設定値エリアNo.を直接指定する方法

●設定値に CS100 を指定している上記プログラムは次のように動作します。

- 1) 実行条件 X100 が ON のとき、データ転送命令「MV」を実行して、CS100 に U30 をセットします。
- 2) カウント入力 X101 が ON のとき、設定値を 30 として、減算動作を行います。

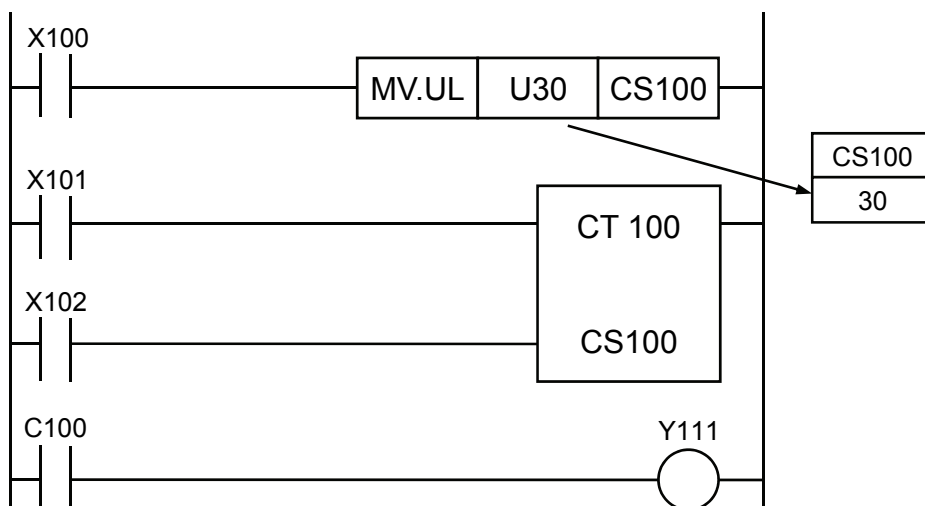
●[S]に指定する設定エリア「CS」のアドレスはカウンタの番号と同じにしてください。



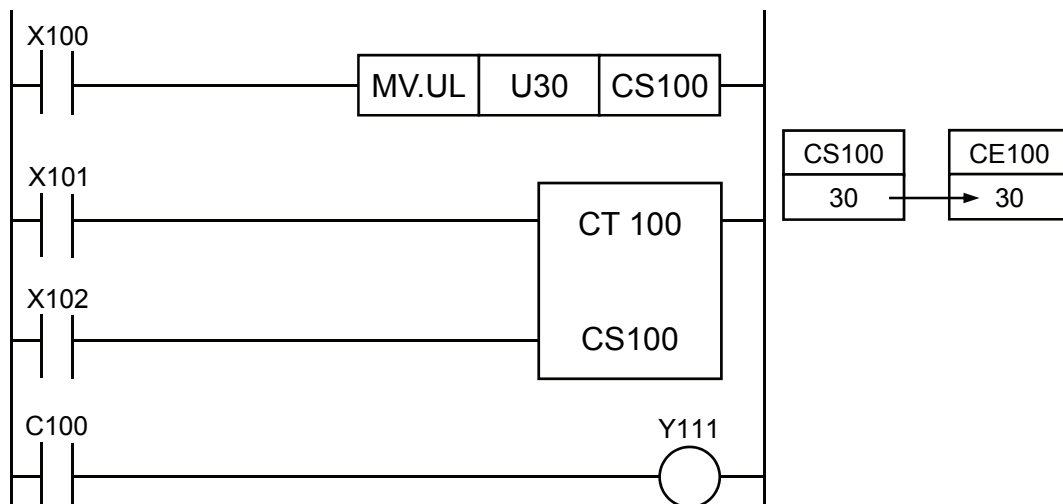
●減算動作中に、設定値エリア「CS」の値を変更しても、変更前の設定で減算動作を続行します。変更した値でカウンタ動作が開始されるのは、カウンタがリセットされたあとで、次にカウンタ入力 OFF から ON になったときです。

## ■ 設定値エリアNo.を直接指定したときのダウンカウンタ動作のしくみ

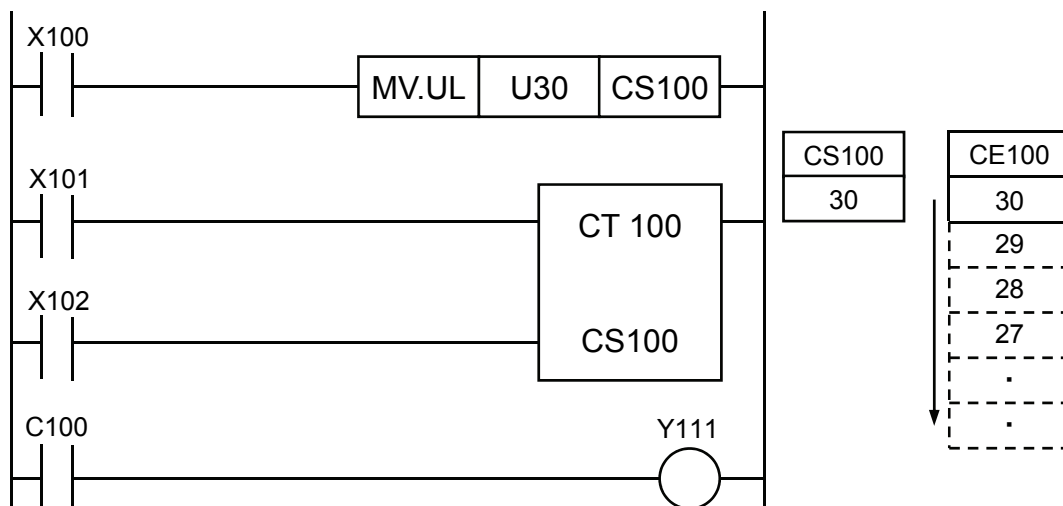
1) 応用命令の実行条件が ON のとき、設定値エリア「CS」に設定をセットします。下図は、「MV」命令を使用する例です。



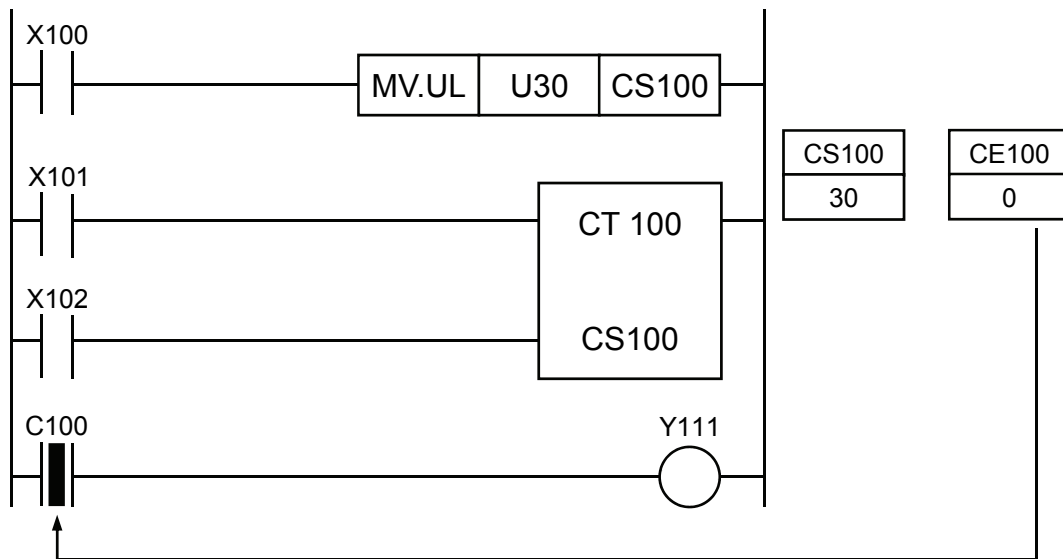
2) リセット入力の立ち下がりにて、設定値エリア「CS」の値が経過値エリア「CE」にプリセットします。



3) カウント入力 X101 が ON するごとに経過値エリア「CE」の値を減算します。



4) 経過値エリア「CE」の値が 0 になれば、同じ番号のカウンタ接点「C」が ON になります。

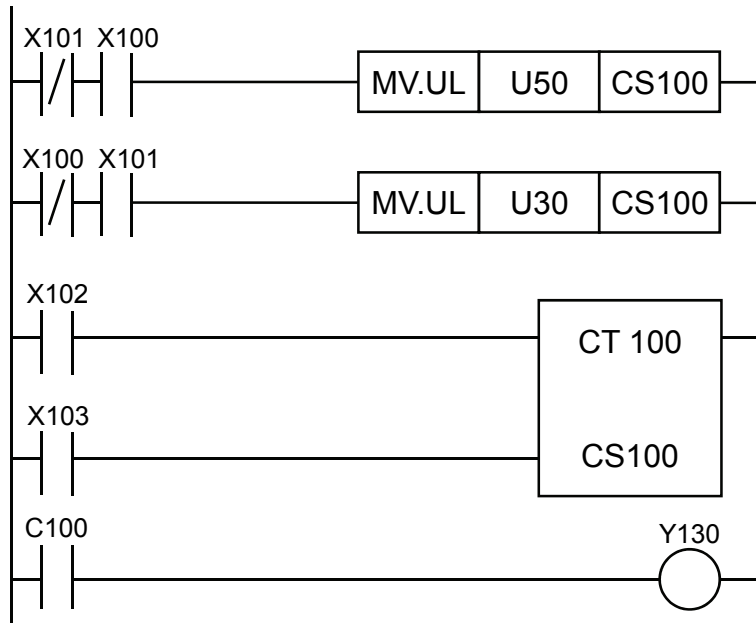




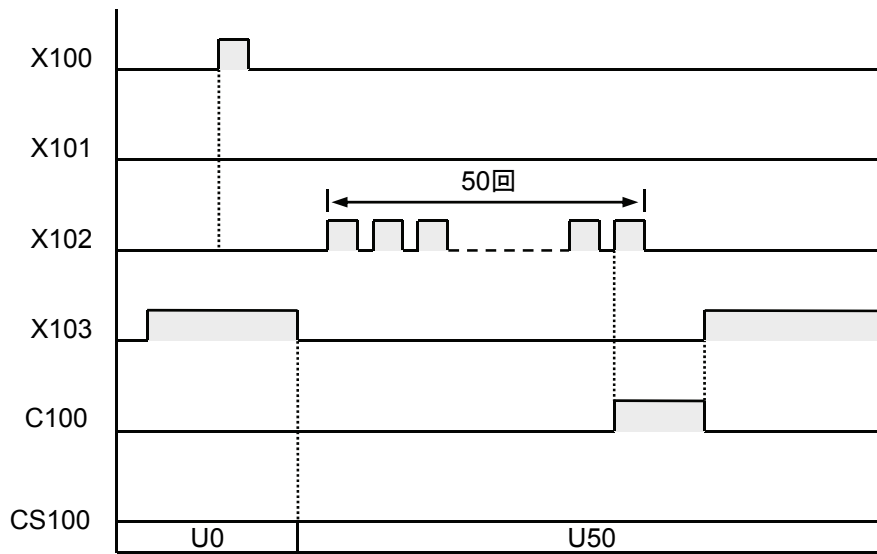
■ 設定値エリアNo.を直接指定する場合の応用例

<例> 条件に応じて、設定値を切り替える例

●ラダー表記

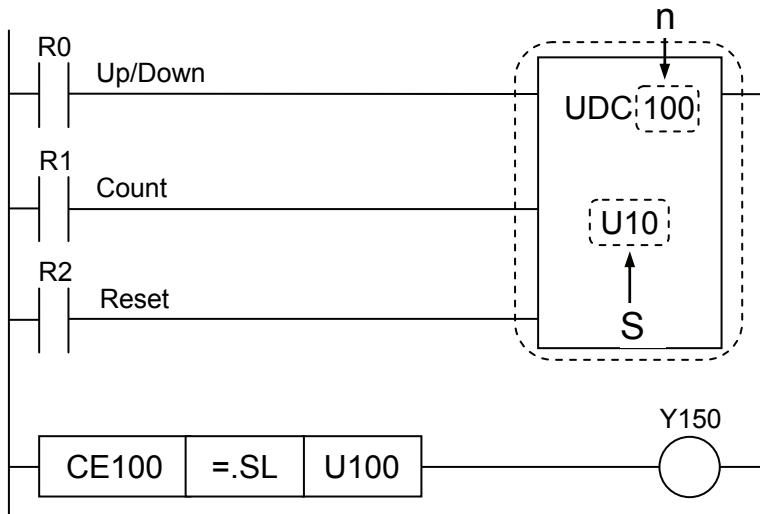


●タイムチャート



# UDC (アップダウンカウンタ)

## ■ ラダー表記



## ■ オペランド一覧

オペランド	説明
n	カウンタ番号
S	カウンタ設定値

## ■ 指定可能なデバイス (●: 指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾*2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS *1	TE CE	IX	K	U	H	SF	DF	""		
n																●						●
S	●	●	●	●			●	●				●				●						●

\*1: CT 命令の場合、CS のみ指定可能

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

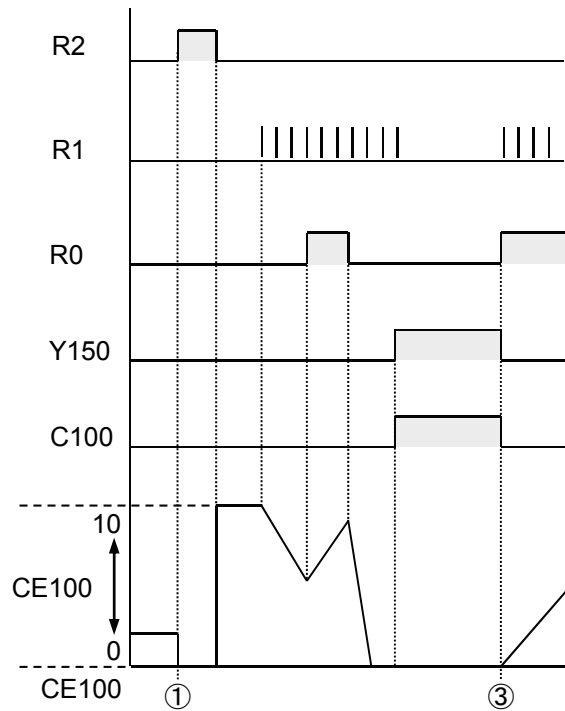
- アップ/ダウン入力に指定されたリレーの ON/OFF によって、アップカウント(加算)、ダウンカウント(減算)を切り替えられるカウンタです。
- カウント動作は、アップ/ダウン入力が入力 ON のときはアップカウント(+1)、OFF のときはダウンカウント(-1)です。経過値は、[CEn]エリアに格納されます。
- リセット入力が入力 ON→OFF になるときに、[S]のプリセット値が[CEn]に転送されます。
- カウント入力が入力 OFF→ON に変化したときの、[CEn]に設定された値を初期値としてカウント動作します。
- リセット入力が入力 ON のとき、[CEn]の経過値エリアをクリアします。
- カウント結果は、[CEn]の経過値をデータ比較命令で任意の設定値と比較することによって判定できます。
- データ比較命令は「UDC」命令の直後に実行してください。

## ■ カウント値の設定について

- カウント値の計数範囲は U1~4294967295 で、10 進数(U 定数)で設定します。

## ■ 動作例

- 1) リセット入力R2がONからOFFになったとき、設定値:U100が経過値:CE100に転送されます。  
この値を目標値とします。
- 2) R0がOFFのときにカウンタ入力R1がONすると、経過値:CE100の値を-1します(減算カウント)。  
R0がONのときにカウンタ入力R1がONすると、経過値:CE100の値を+1します(加算カウント)。
- 3) カウンタ経過値:CE100の値とU0とを比較し、CE100=U0のとき、外部出力Y150をONします。

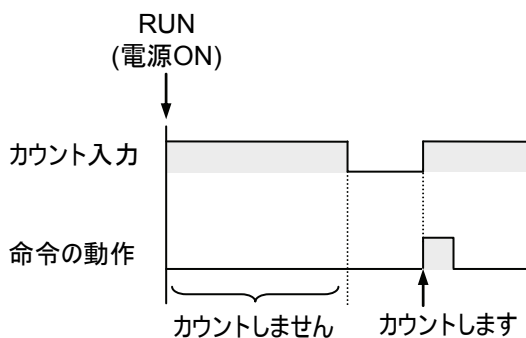


## ■ プログラム上のご注意

- 経過値エリアに保持型のメモリエリアを指定した場合、経過値は保持されている内容に従って動作します。
- 演算開始時に設定値は経過値エリアへ自動的にプリセットされませんのでご注意ください。プリセットする時は、リセット入力を ON→OFF にしてください。
- 「UDC」命令を、アンドスタック命令やポップスタック命令と組み合わせるときは、正しくない記述にならないようご注意ください。

## ■ カウント入力検出についてのご注意

- 「UDC」命令では、カウント入力の OFF→ON の立ち上がりを検出して、加減算します。
- カウント入力が ON しつづけている間は、立ち上がり時のみカウントして、以後はカウントしません。
- RUN に切り替えたときや RUN モードで電源を投入したときにカウント入力が最初から ON している場合には最初のスキャンでは加減算動作が行われません。

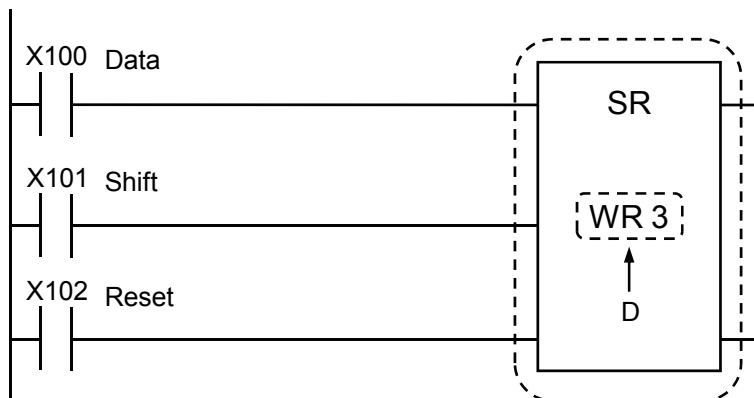


- MC～MCE 命令、JP～LBL 命令など命令を実行する順序を変える命令(下記 1～6)と合わせて使用する場合、命令の実行とカウント入力のタイミングにより、命令の動作が変わりますのでご注意ください。

- 1) MC～MCE 命令
- 2) JP～LBL 命令
- 3) LOOP～LBL 命令
- 4) CNDE 命令
- 5) ステップラダー命令
- 6) サブルーチン命令

# SR (シフトレジスタ)

## ■ ラダー表記



## ■ オペランド一覧

オペランド	説明
D	シフトするデバイス

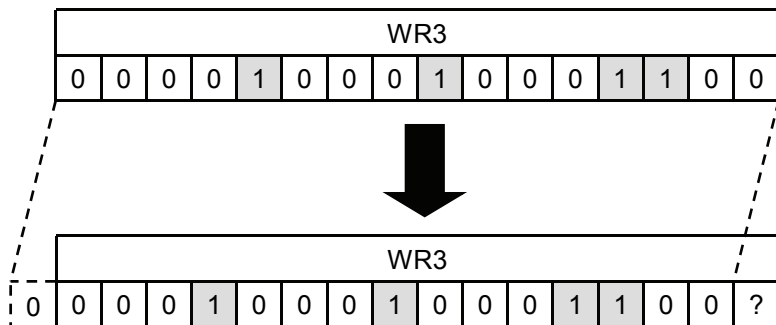
## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
D			●																		

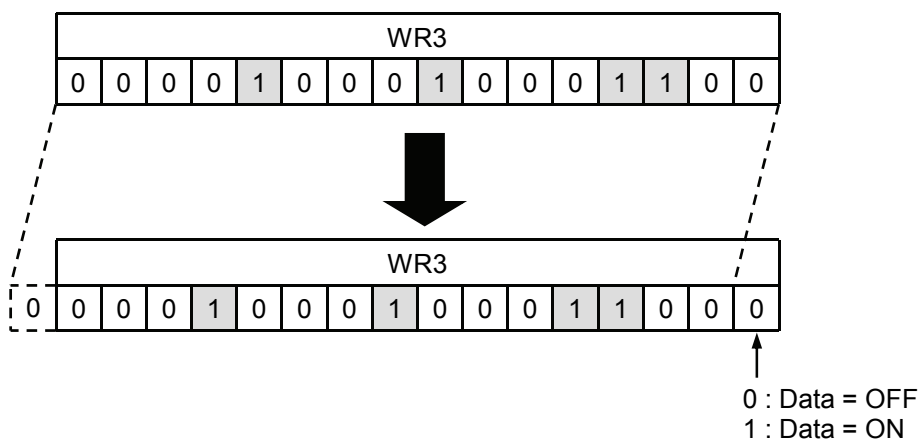
## ■ 動作説明

• 指定したレジスタ WR (指定した演算単位) の内容を、左に 1 ビット移動 (シフト) させる命令です。

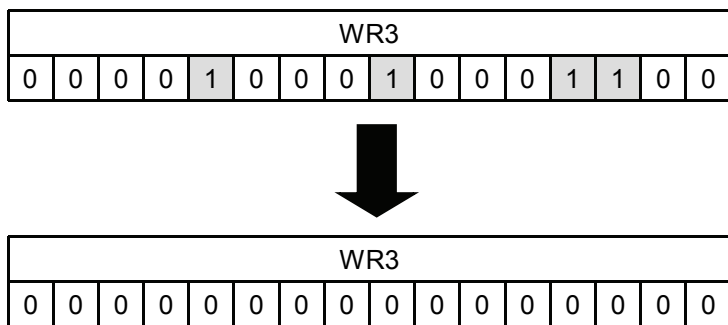
1) シフト入力が ON (立ち上がり) すると、WR の内容を 1 ビット左シフト。



2) シフト時、データ入力が ON なら 1、OFF なら 0 を、空きビット (最下位ビット) にセットします。

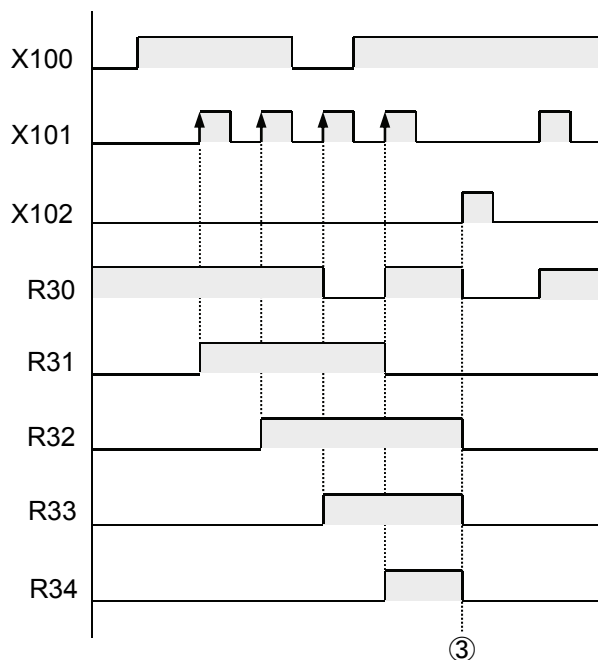


3) リセット入力が ON のとき、指定したレジスタの内容はクリアされます。



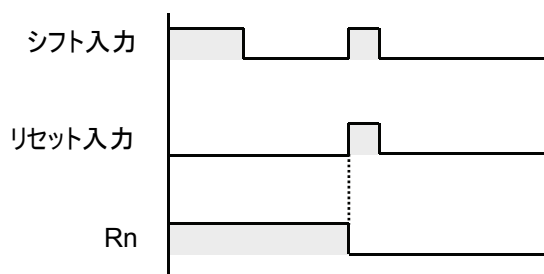
## ■ 動作例

- 1) X102がOFFの状態ではX101をONするとWR3の内容(内部リレーR30~R3F)を1ビット左へシフトします。
- 2) 左シフトで空いたビット(R30)には、X100がONのとき1、OFFのとき0をセットします。
- 3) X102がONになると、WR3の内容がリセットされ0となります。



## ■ プログラム上のご注意

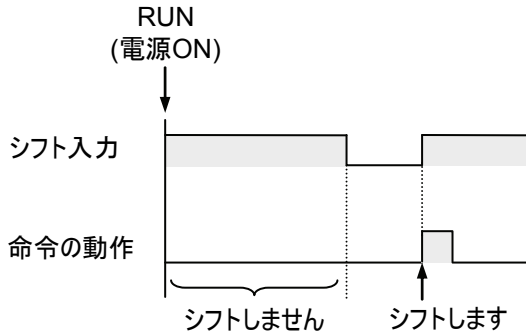
- 「SR」命令には、データ入力、シフト入力、リセット入力が必要です。
- リセット入力とシフト入力と同時に立ち上がった場合は、リセット入力優先されます。



- シフトレジスタに保持型のメモリエリアを指定するときは、電源 ON 時に自動リセットを行いませんのでご注意ください。
- シフトレジスタ命令をアンドスタック命令やポップスタック命令と組み合わせるときは、正しくない記述にならないようにご注意ください。

## ■ シフト入力検出についてのご注意

- SR 命令では、シフト入力の OFF→ON の立ち上がりを検出して、シフトします。
- シフト入力 ON している間は、立ち上がり時のみシフトして、以後はシフトしません。
- RUN に切り替えたときや RUN モードで電源を投入したときにシフト入力最初から ON している場合には最初のスキャンではシフト動作が行われません。



- MC～MCE 命令、JP～LBL 命令など命令を実行する順序を変える命令(下記 1～6)と合わせて使用する場合、命令の実行とシフト入力のタイミングにより、命令の動作が変わりますのでご注意ください。

- 1) MC～MCE 命令
- 2) JP～LBL 命令
- 3) LOOP～LBL 命令
- 4) CNDE 命令
- 5) ステップラダー命令
- 6) サブルーチン命令

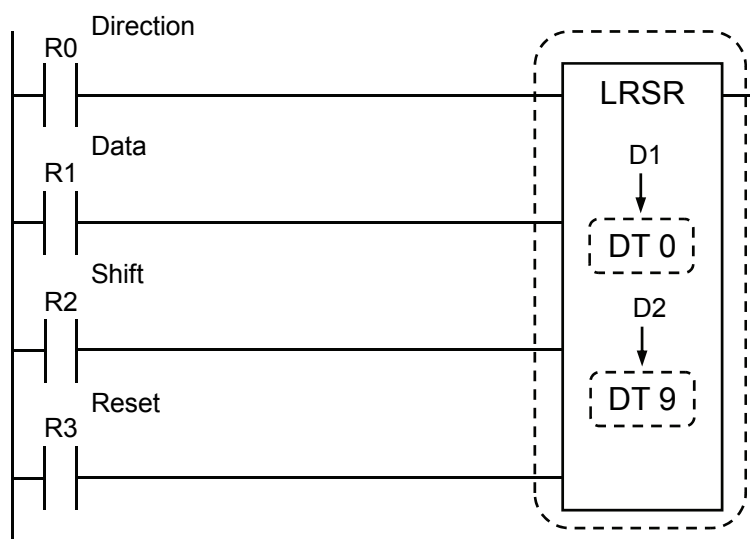
## ■ 関連命令

- シフトレジスタは、本命令の他に、左右シフトレジスタ(「LRSR」)があります。また、「データシフト」命令や「データ回転」命令を使用しても同様の動作を実現することができます。



# LRSR（左右シフトレジスタ）

## ■ ラダー表記



## ■ オペランド一覧

オペランド	説明
D1	シフト開始位置
D2	シフト終了位置

## ■ 指定可能なデバイス（●:指定可能）

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""	
D1	●	●	●	●			●	●	●		●										
D2	●	●	●	●			●	●	●		●										

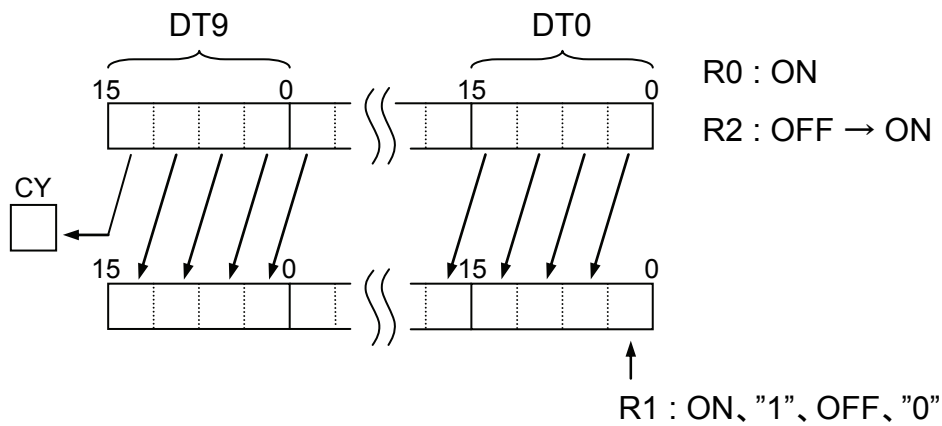
## ■ 動作説明

- 左右シフト入力に指定されたリレーの ON/OFF によって、左(上位ビット方向)か右(下位ビット方向)か、1ビット移動(シフト)する方向を切り替えられるシフトレジスタです。
- シフト動作は、左右シフト入力 ON のときは左シフト、OFF のときは右シフトです。
- [D1]と[D2]は同じ種類のエリアを指定してください。また、[D1] ≤ [D2]になるようにしてください。

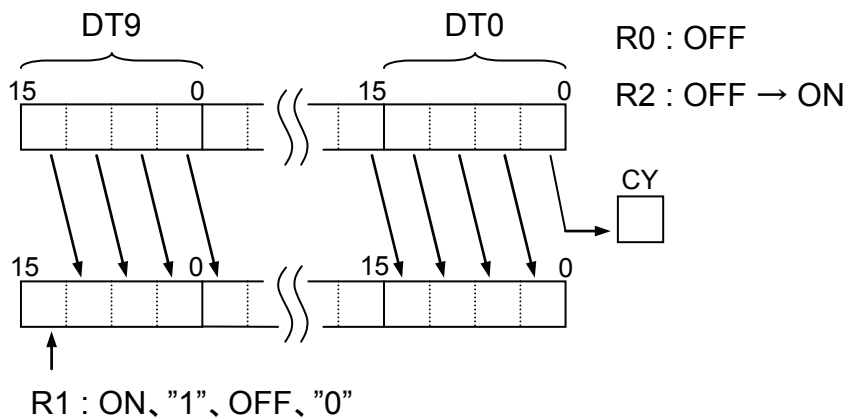
## ■ 左右シフトレジスタの動作

- 1) シフト入力 OFF → ON に変化すると(リセット入力は OFF 状態)、[D1]と[D2]で指定したエリアの内容を左または右に1ビットシフトします。
- 2) シフト動作したとき、データ入力が ON なら 1、OFF なら 0 をシフトによってできた空きビット(最上位ビットまたは最下位ビット)にセットします。また、シフトによって押し出されるビット(左シフト時は最上位ビット、右シフト時は最下位ビット)がシステムリレー SR9(キャリーフラグ)にセットされます。
- 3) リセット入力が ON のとき、指定されたエリアの内容は 0 クリアされます。

■ 動作例: 左シフトの場合



■ 動作例: 右シフトの場合

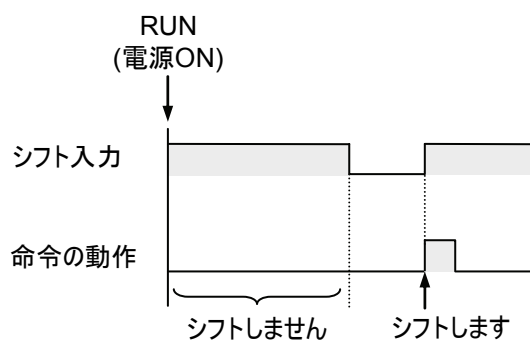


■ フラグ動作

名称	説明
SR7 SR8 (ER)	[D1]のアドレス > [D2]のアドレスのとき、ON になります
SR9(CY)	シフトによって押し出される値が"1"の場合は、ON になります

### ■ シフト入出力についてのご注意

- 「LRSR」命令では、シフト入力の OFF→ON の立ち上がりを検出して、シフトします。
- シフト入力 が ON しつづけている間は、立ち上がり時のみシフトして、以後はシフトしません。
- RUN に切り替えたときや RUN モードで電源を投入したときにシフト入力 が最初から ON している場合には最初のスキャンではシフト動作が行われません。



- MC～MCE 命令、JP～LBL 命令など命令を実行する順序を変える命令(下記 1～6)と合わせて使用する場合、命令の実行とシフト入力のタイミングにより、命令の動作が変わりますのでご注意ください。

- 1) MC～MCE 命令
- 2) JP～LBL 命令
- 3) LOOP～LBL 命令
- 4) CNDE 命令
- 5) ステップラダー命令
- 6) サブルーチン命令

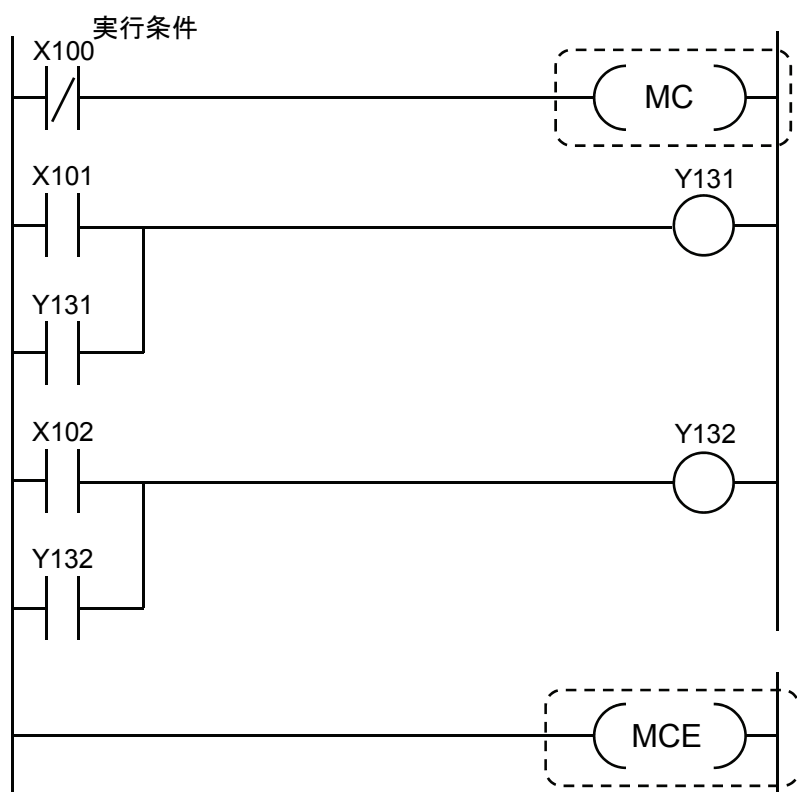
### ■ プログラム上のご注意

- 「LRSR」命令を、アンドスタック命令やポップスタック命令と組み合わせるときは、正しくない記述にならないようご注意ください。

## MC (マスタコントロールリレー)

## MCE (マスタコントロールリレーエンド)

### ■ ラダー表記



## ■ 動作説明

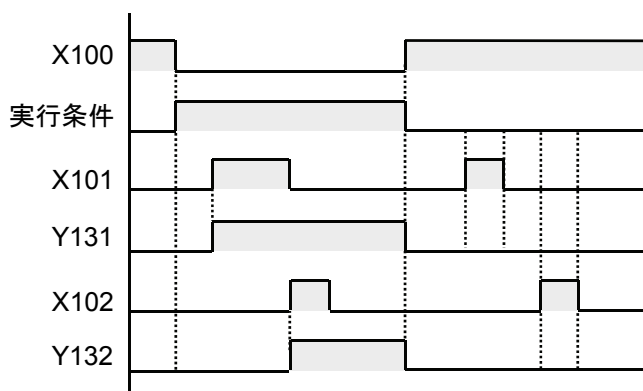
- 実行条件が ON のとき、「MC」命令と「MCE」命令の間にはさまれたプログラムを動作させます。
- 実行条件が OFF のとき、各入出力リレーの状態は以下のようになります。

命令の種類	動作
OT	すべて OFF にします。
KP	状態を保持します
SET	
RST	
TM	リセットします。
CT	途中経過を保持します。
SR	
微分命令	MC～MCE命令間に微分命令を使用した場合と同様です。 <a href="#">MC～MCE間の微分命令の動作</a> の項をご参照ください。
その他の命令	実行しません。

- 微分命令など実行条件の立ち上がりを検出して実行する命令(下記 1～6)を使用する場合は注意が必要です。

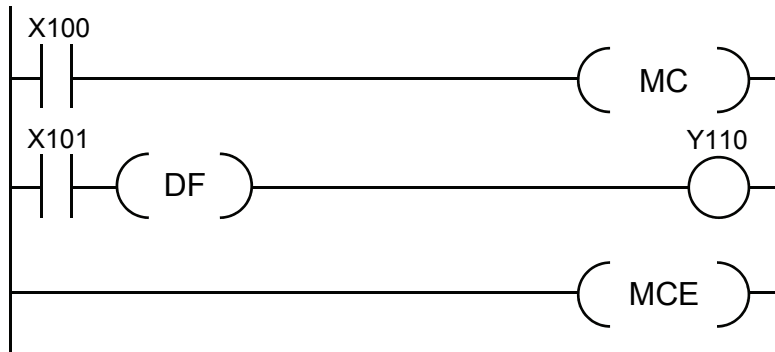
- 1) DF(立ち上がり微分)
- 2) CT(カウンタ)のカウンタ入力
- 3) UDC(アップダウンカウンタ)のカウンタ入力
- 4) SR(シフトレジスタ)のシフト入力
- 5) LRSR(左右シフトレジスタ)のシフト入力
- 6) 微分実行型応用命令(pと命令名で指定する命令)

## ■ 動作例

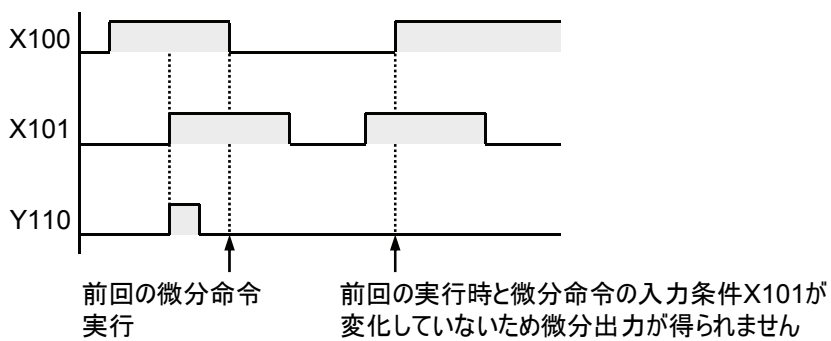


## ■ MC～MCE間の微分命令の動作

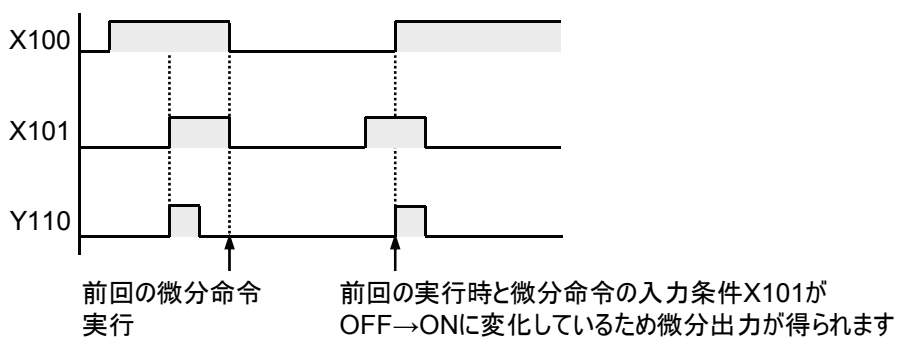
- MC～MCE 間に微分命令を使用した場合、MC の実行条件と微分命令の入力タイミングにより、得られる出力が下記のように異なりますのでご注意ください。



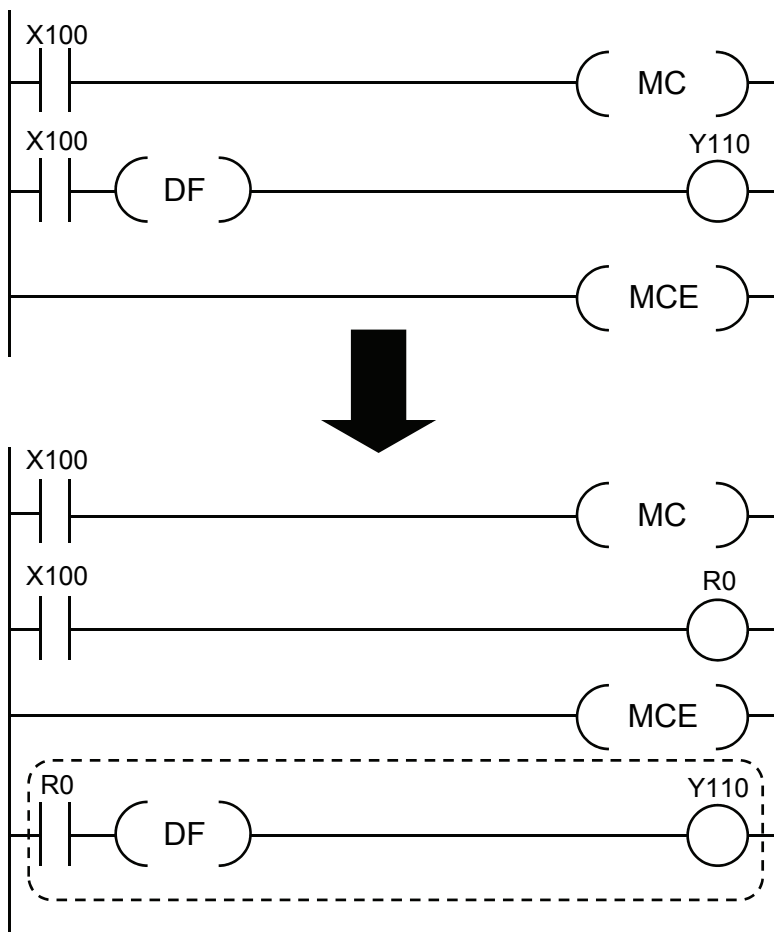
<タイミングチャート1>



<タイミングチャート2>

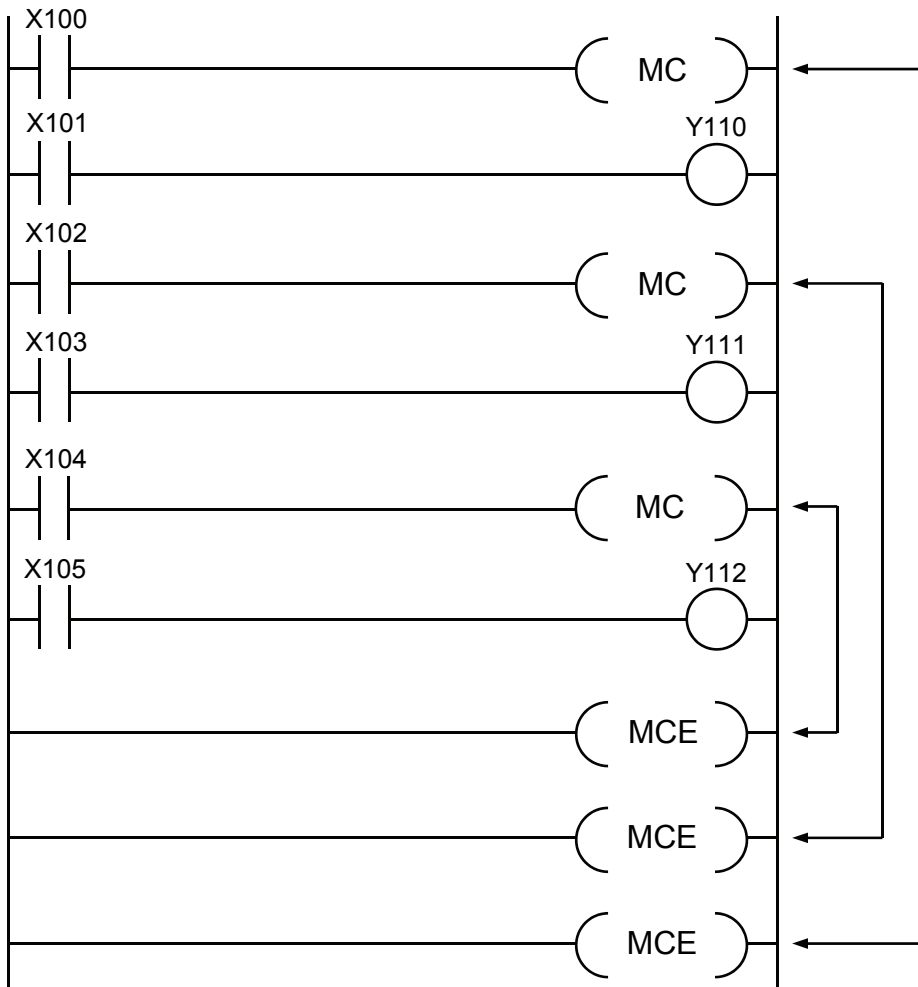


- MC の実行条件と微分命令の実行条件に同じ条件を指定した場合は出力されません。出力が必要な場合は、MC ~MCE 命令の外に、微分命令を記述してください。



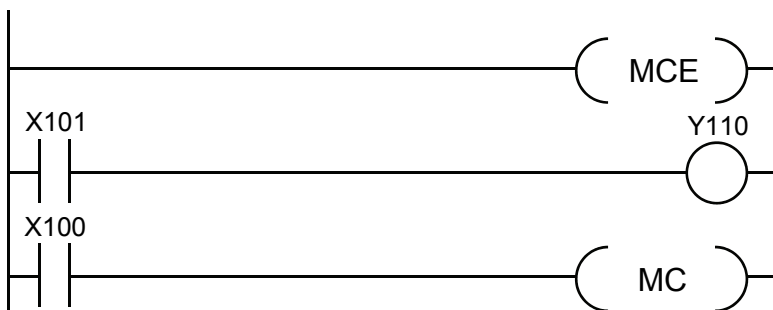
## ■ プログラム記述上のご注意

- 「MC」「MCE」命令の中に、別の「MC」「MCE」命令のペアを入れ、記述すること(ネスティング)もできます。(MC 命令のネスティングは 30 回までの制限があります。)



- 次の場合、プログラムを実行することができません。

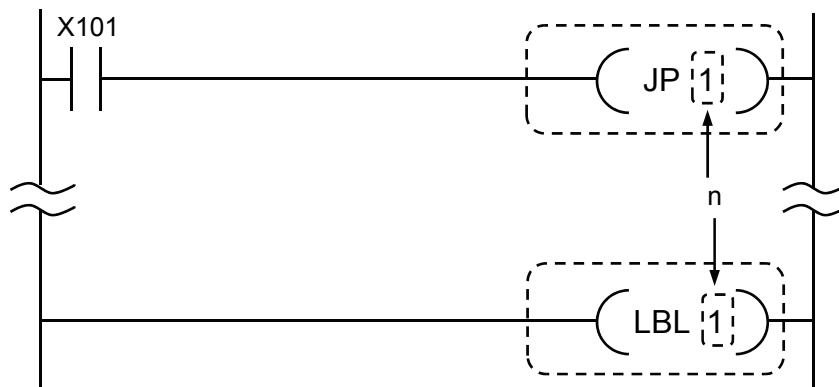
- 1) 「MC」、「MCE」のどちらかが欠けている場合。
- 2) 「MC」、「MCE」の順序が逆の場合。





# JP,LBL (ジャンプ、ラベル)

## ■ ラダー表記



## ■ オペランド一覧

オペランド	説明
n	ラベル番号(LBL への相対ジャンプ・ポイント)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U *1	H	SF	DF	" "		
n																●						

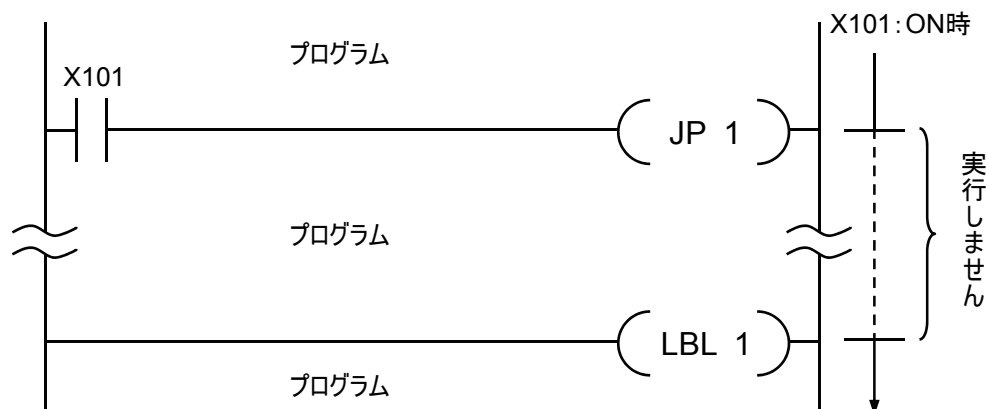
\*1: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

## ■ 動作説明

- 実行条件が ON のとき、指定している番号と同一番号のラベル(「LBL」命令)までジャンプします。
- プログラムの実行は、飛び先ラベル以降の命令から引き続き行われます。

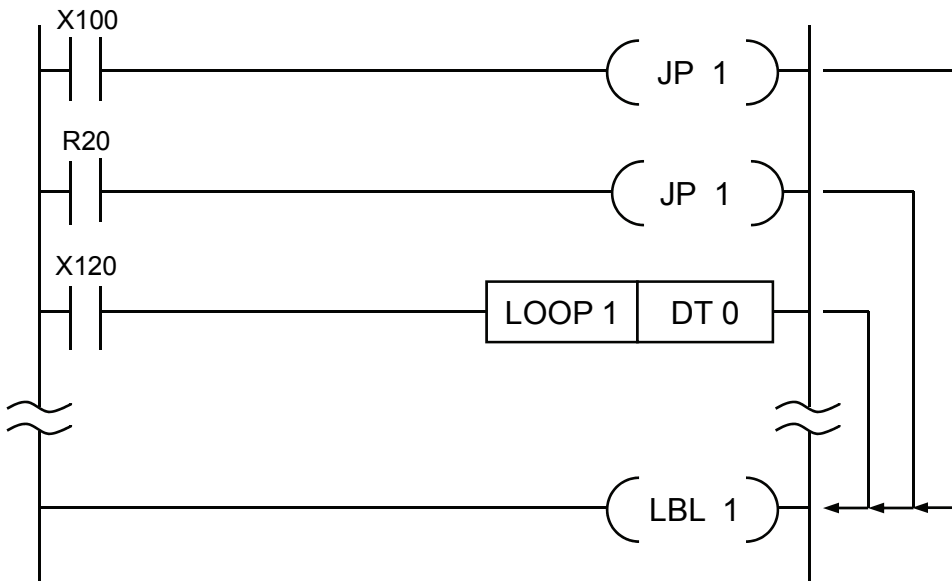
## ■ 動作例

- 実行条件 X101 が ON のとき、ラベル1までジャンプします。



## ■ プログラム上のご注意

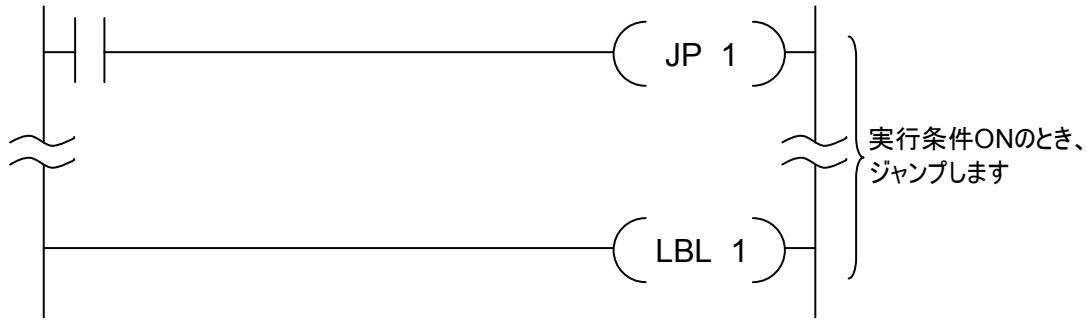
- 同じラベル番号の「JP」命令を複数回使用することができます。



- 同じ番号を持った「LBL」命令を 2 個以上、1 プログラム中に書き込むことはできません。
- 飛び先のラベルがプログラムされていない場合は文法エラーになります。
- ラベルが、「JP」命令よりも前のアドレスに書き込まれている場合、スキャンを終了することができずに演算渋滞エラーが発生する可能性がありますので、ご注意ください。
- ステップラダー領域（「SSTP」～「STPE」の範囲）では、「JP」命令・「LBL」命令は使用できません。
- 主プログラムから副プログラム（「ED」命令以降のサブルーチンや割り込みプログラム）へのジャンプ、副プログラムから主プログラムへのジャンプ、副プログラムから別の副プログラムへのジャンプはできません。
- 微分命令など実行条件の立ち上がりを検出して実行する命令（下記 1～6）を使用する場合は注意が必要です。
  - 1) DF（立ち上がり微分）
  - 2) CT（カウンタ）のカウンタ入力
  - 3) UDC（アップダウンカウンタ）のカウンタ入力
  - 4) SR（シフトレジスタ）のシフト入力
  - 5) LRSR（左右シフトレジスタ）のシフト入力
  - 6) 微分実行型応用命令（p と命令名で指定する命令）

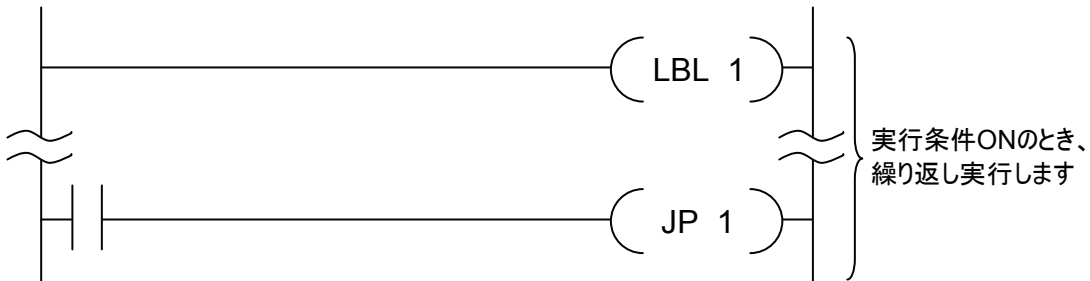
## ■ JP～命令間の「TM」「CT」「SR」命令の動作について

- 「LBL」命令が「JP」命令より後のアドレスにある場合、「JP」命令実行時の各命令の処理は次のようになります。



- 1) TM 命令: 計時されません。1 スキャン中に 1 回計時されなければ時間が保証できませんので、ご注意ください。
- 2) CT 命令: カウント入力 が ON になっても、カウントされません。経過値は保持されます。
- 3) SR 命令: シフト入力 が ON になっても、シフトしません。指定のレジスタの内容は保持されます。

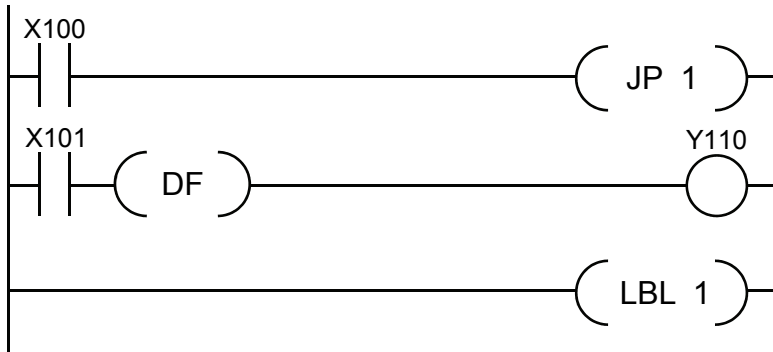
- 「LBL」命令が「JP」命令より前のアドレスにある場合、「JP」命令実行時の各命令の処理は次のようになります。



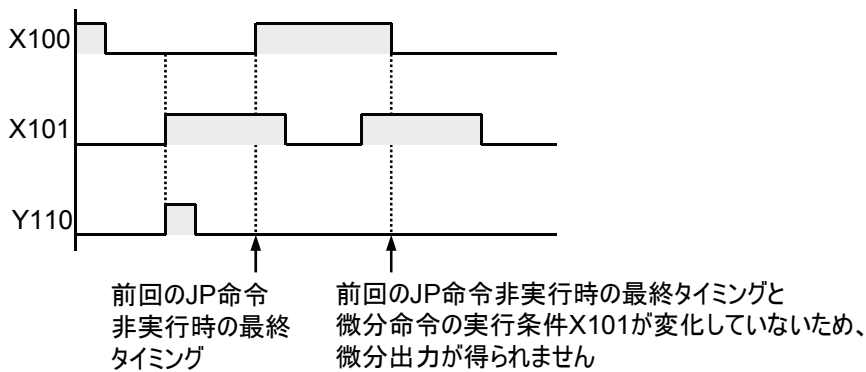
- 1) TM 命令: 1 スキャン中に何回も計時するため、時間が保証できません(注)。
- 2) CT 命令: カウント入力の状態が、そのスキャン中に変化しなければ、通常と同様に動作します。
- 3) SR 命令: シフト入力の状態が、そのスキャン中に変化しなければ、通常と同様に動作します。

## ■ JP～LBL命令間の微分命令の動作について

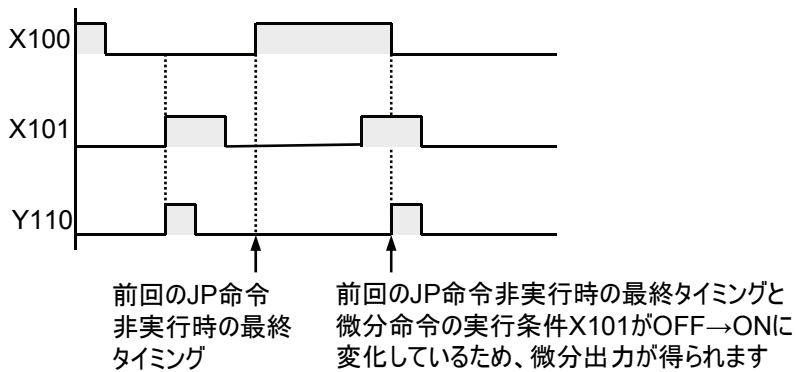
- 「JP」～「LBL」間に微分命令を使用した場合、「JP」の実行条件と微分命令の入力のタイミングにより、得られる出力が下記のように異なりますのでご注意ください。



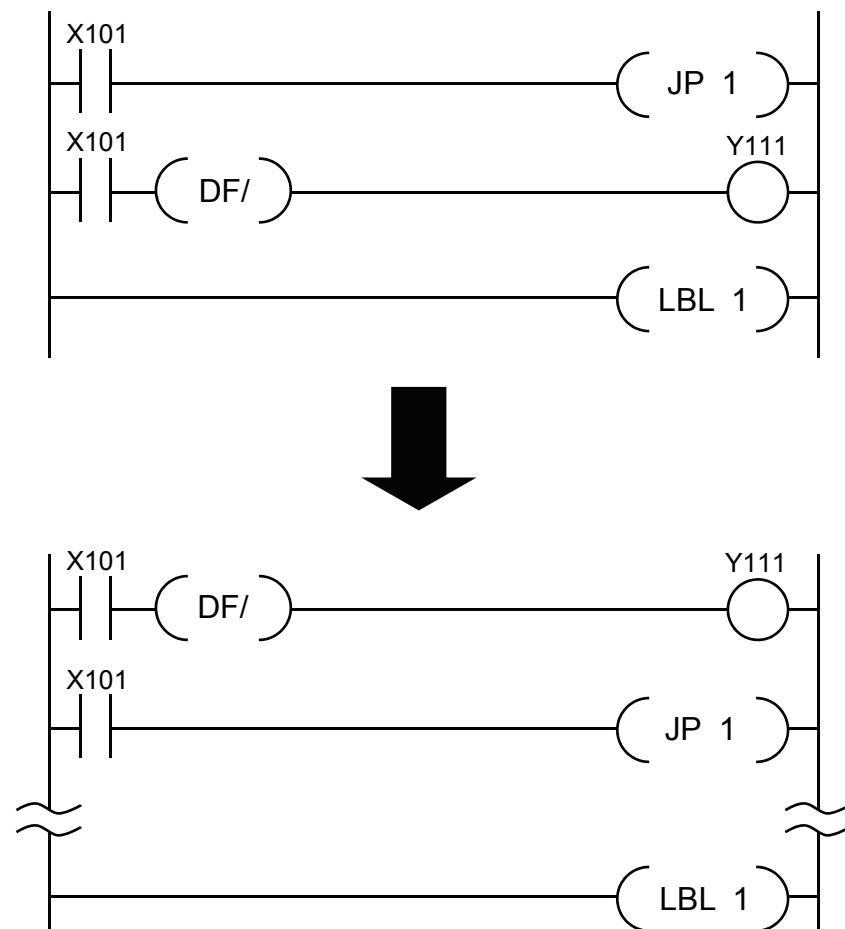
<タイミングチャート1>



<タイミングチャート2>

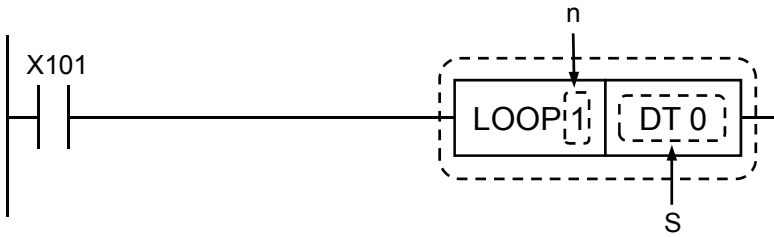


- 「JP」命令の実行条件と微分命令の実行条件が同じである場合、微分命令の実行条件の立ち上がり(または立ち下がり)は検出されません。微分出力が必要な場合は、微分命令を「JP」～「LBL」の外に記述してください。



# LOOP, LBL (ループ、ラベル)

## ■ ラダー表記



## ■ オペランド一覧

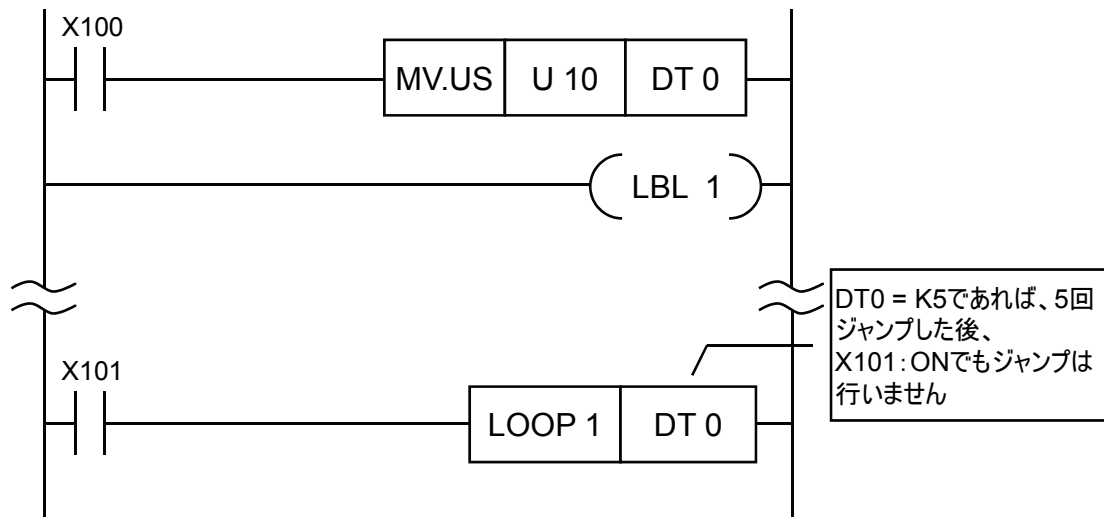
オペランド	説明
n	ラベル番号
S	ループ回数

## ■ 指定可能なデバイス (●: 指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデックス 修飾		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		""	
n																●						
S	●	●	●	●			●	●														●

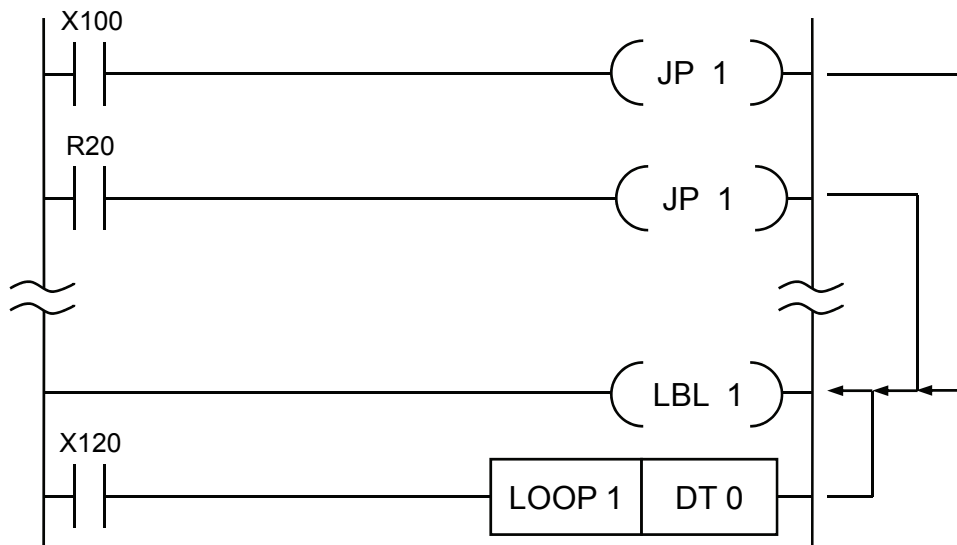
## ■ 動作説明

- 条件が ON のとき、[S]の内容から-1してその結果が 0 でないとき、指定している番号と同一番号のラベル(「LBL」命令)までジャンプします。
- プログラムの実行は、飛び先ラベル以降の命令から引き続き行われます。
- 「LOOP」命令では、実行する回数を設定します。[S]で設定した回数が終了すると、実行条件が成立していても、ジャンプしません。



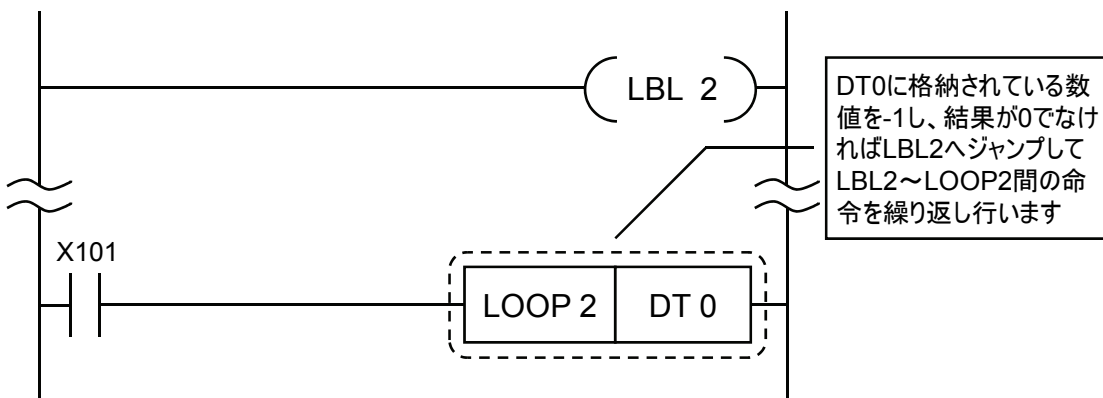
- [S]で指定したメモリエリアの内容が最初から 0 の場合にはラベル番号には飛ばず、次の処理を行います。

- ラベルは、「JP」命令、「LOOP」命令で共用です。どの命令からでも何回でも飛び先として利用することができます。



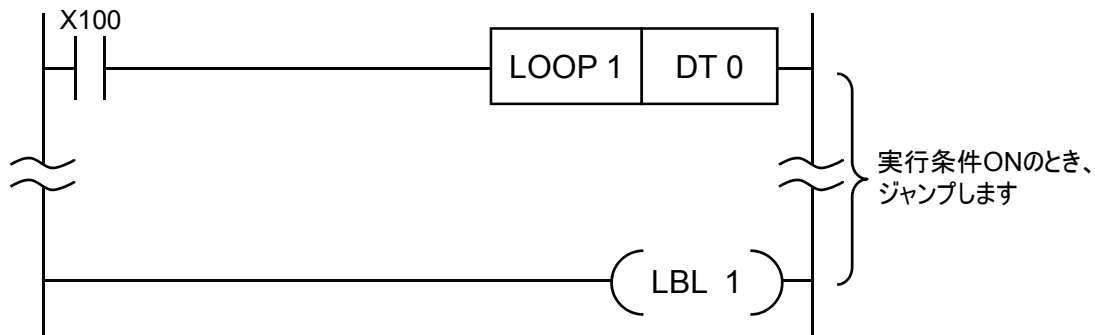
- 同じ番号を持った「LBL」命令を2個以上、1プログラム中に書き込むことはできません。
- 飛び先のラベルがプログラムされていない場合は文法エラーになります。

#### ■ 動作例

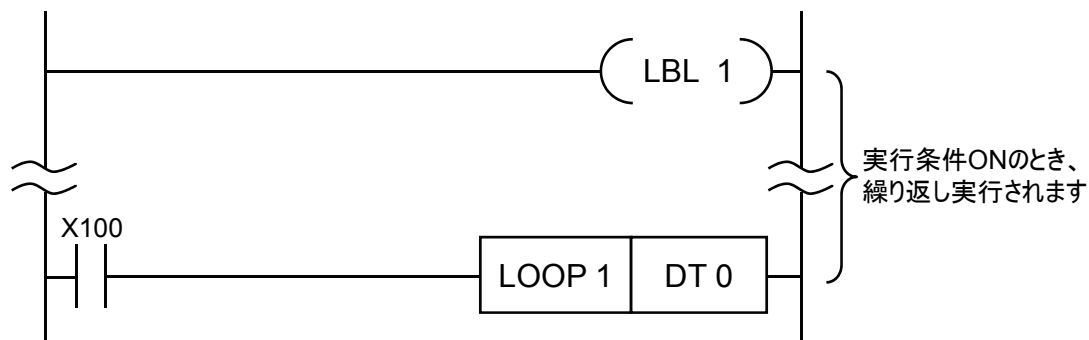


## ■ LOOP～LBL命令間の「TM」「CT」「SR」命令の動作について

- 「LBL」命令が「LOOP」命令より後のアドレスにある場合、「LOOP」命令実行時の各命令の処理は次のようになります。



- 1) TM 命令: 計時されません。1 スキャン中に 1 回計時されなければ時間が保証できませんので、ご注意ください。
  - 2) CT 命令: カウント入力 が ON になっていても、カウントされません。経過値は保持されます。
  - 3) SR 命令: シフト入力 が ON になっていても、シフトしません。指定のレジスタの内容は保持されます。
- 「LBL」命令が「LOOP」命令より前のアドレスにある場合、「LOOP」命令実行時の各命令の処理は次のようになります。



- 1) TM 命令: 1 スキャン中に何回も計時するため、時間が保証できません。
- 2) CT 命令: カウント入力の状態が、そのスキャン中に変化しなければ、通常と同様に動作します。
- 3) SR 命令: シフト入力の状態が、そのスキャン中に変化しなければ、通常と同様に動作します。

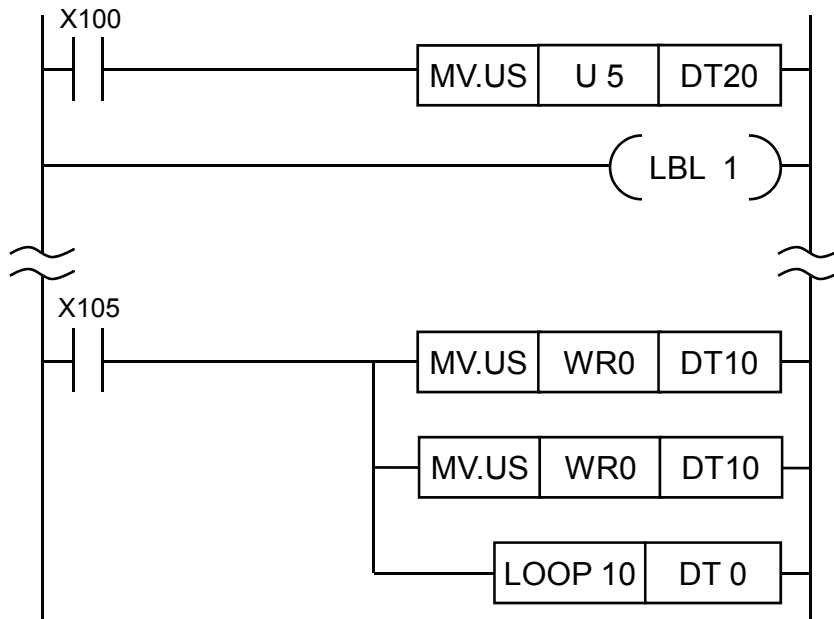


## ■ プログラム上のご注意

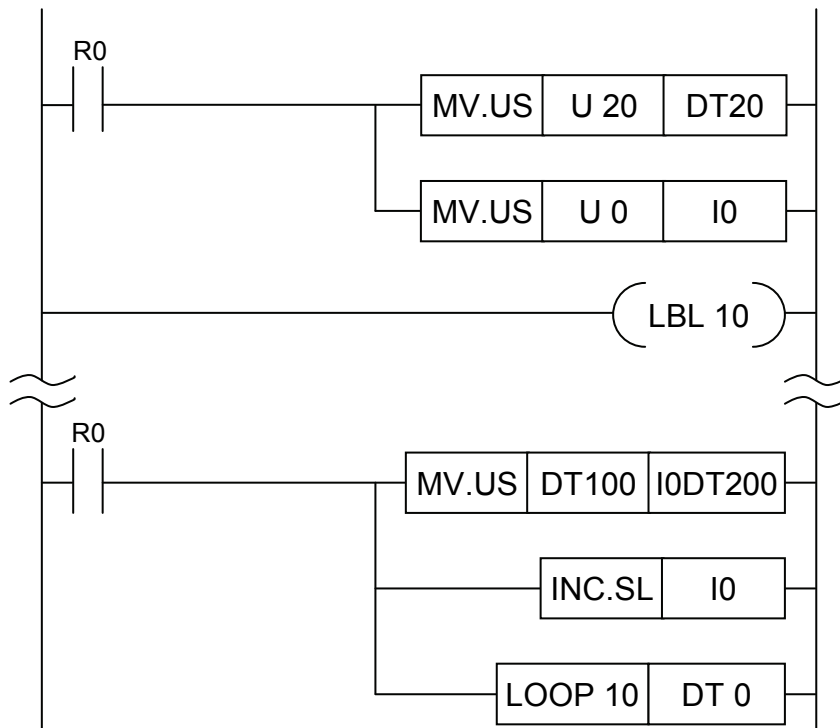
•ラベルを、「LOOP」命令よりも前のアドレスに書き込む場合、以下の点にご注意ください。

- 1) ループ回数を設定する命令は、必ず「LBL」～「LOOP」よりも前に書き込んでください。
- 2) 「LBL」～「LOOP」間で繰り返し実行する各命令は、「LOOP」命令と同じ条件で実行するように記述してください。
- 3) 繰り返している間に一回のスキャンが演算渋滞監視時間を超えてしまい、演算渋滞エラーが発生する可能性があります。

<例1> X105がONのときに2つのMV命令を5回繰り返す



<例2> DT100の値をDT200～DT219に転送する



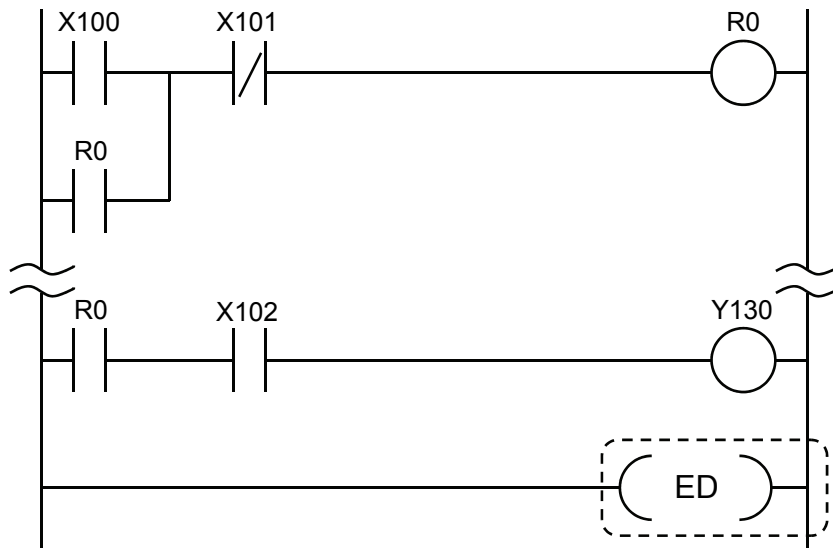
- ステップラダー領域(「SSTP」～「STPE」の範囲)では「LOOP」命令・「LBL」命令は使用できません。
- 主プログラムから副プログラム(「ED」命令以降のサブルーチンや割り込みプログラム)へのジャンプ、副プログラムから主プログラムへのジャンプ、副プログラムから別の副プログラムへのジャンプはできません。
- 微分命令など実行条件の立ち上がりを検出して実行する命令(下記 1～6)を使用する場合は注意が必要です。
  - 1) DF(立ち上がり微分)
  - 2) CT(カウンタ)のカウント入力
  - 3) UDC(アップダウンカウンタ)のカウント入力
  - 4) SR(シフトレジスタ)のシフト入力
  - 5) LRSR(左右シフトレジスタ)のシフト入力
  - 6) 微分実行型応用命令(pと命令名で指定する命令)

#### ■ フラグ動作

名称	説明
SR7 SR8 (ER)	[S]の内容が負の数るとき(最上位ビットが 1)、ON

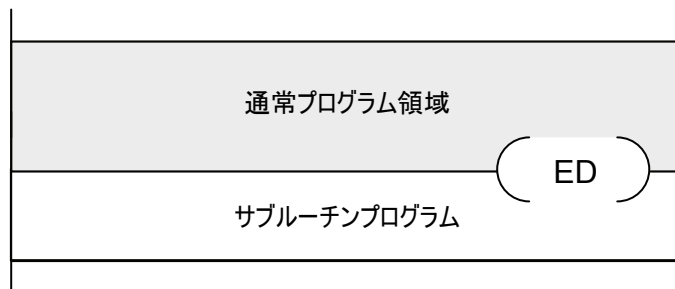
# ED (エンド)

## ■ ラダー表記



## ■ 動作説明

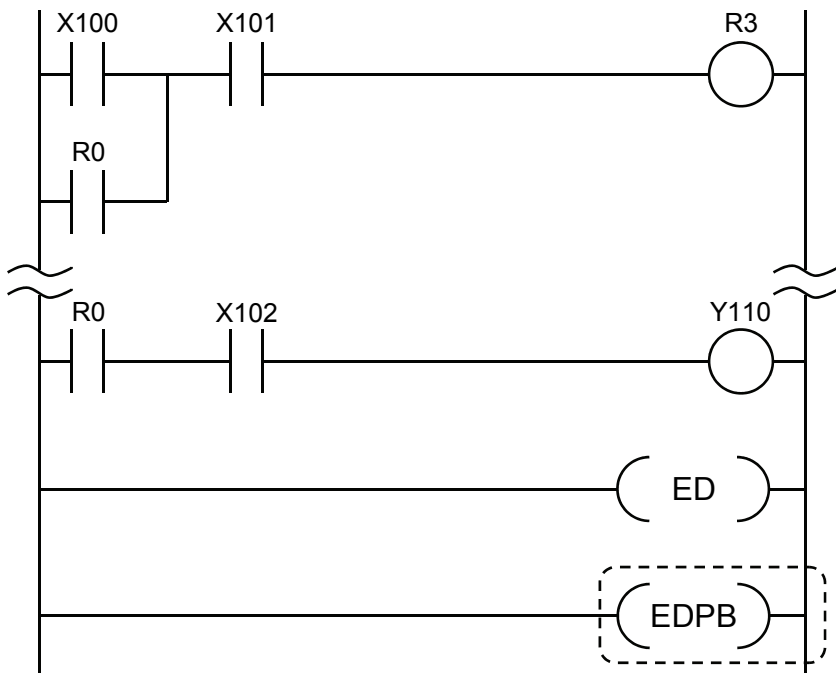
- 通常プログラム領域の最後には、「ED」命令を記述してください。



- プログラム領域はこの命令によって、通常プログラム領域(主プログラム)と、“サブルーチン”や“割り込みプログラム”の領域(副プログラム)に分けられます。
- サブルーチンプログラム、および割り込みプログラムは、この「ED」命令以降に書き込むようにしてください。

# EDPB (エンドプログラムブロック)

## ■ ラダー表記

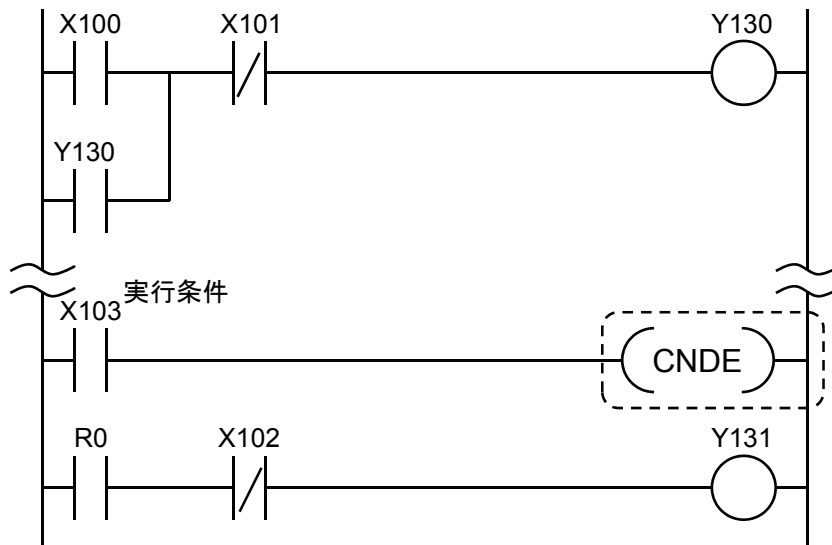


## ■ 動作説明

- PB (プログラム・ブロック)の終了を示します。

# CNDE（条件付きエンド）

## ■ ラダー表記

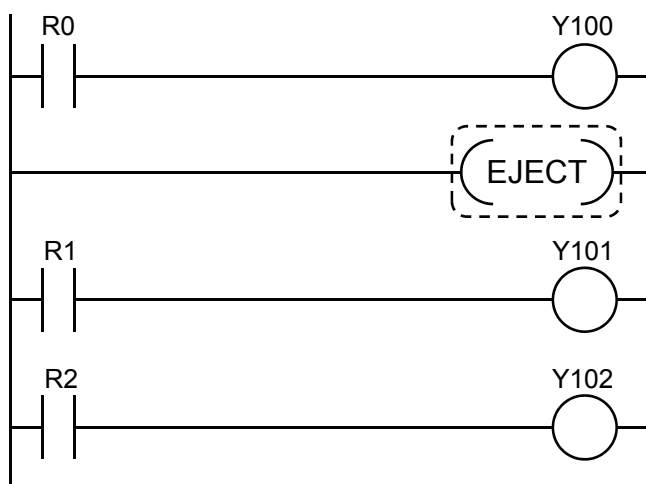


## ■ 動作説明

- 任意のアドレスでプログラムの演算処理を終了することができます。
- 実行条件が ON になると、プログラムの演算処理を終了し、入出力などの処理を行います。処理が終わると、先頭アドレスに戻ります。
- 必要なプログラムのスキャンが終わり次第、処理を行うことによって、処理タイミングを調整することができます。
- 「CNDE」命令は、サブルーチンなどの副プログラムには使用できません。主プログラム領域で使用してください。
- 「CNDE」は、主プログラム内で何度でも記述できます。
- 微分命令など実行条件の立ち上がりを検出して実行する命令(下記 1～6)を使用する場合は注意が必要です。
  - 1) DF(立ち上がり微分)
  - 2) CT(カウンタ)のカウント入力
  - 3) UDC(アップダウンカウンタ)のカウント入力
  - 4) SR(シフトレジスタ)のシフト入力
  - 5) LRSR(左右シフトレジスタ)のシフト入力
  - 6) 微分実行型応用命令(p と命令名で指定する命令)

# EJECT (エジェクト)

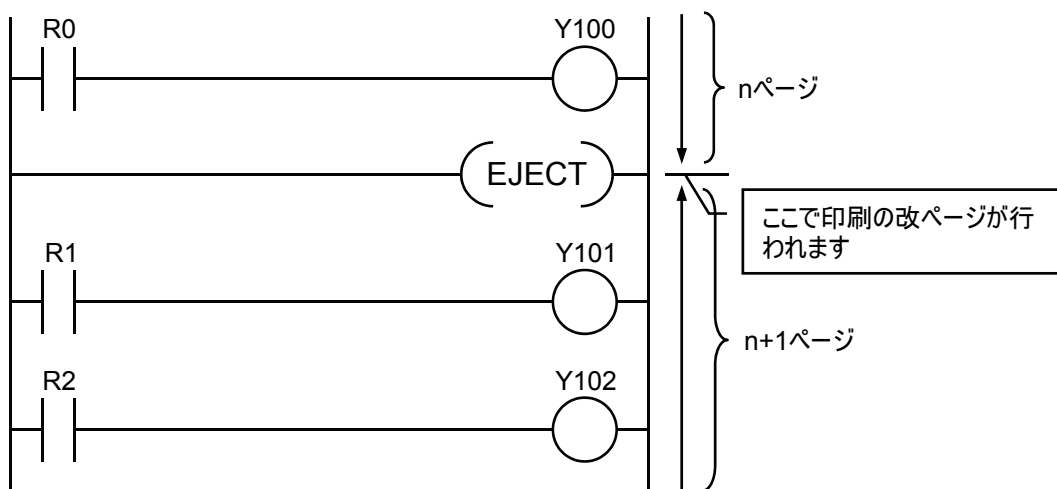
## ■ ラダー表記



## ■ 動作説明

- ツールソフトウェアで作成したプログラムをプリントアウトする際、この命令が挿入された位置で改ページが行われます。
- NOP 命令同様、プログラム上の処理は行われません。

## ■ 動作例



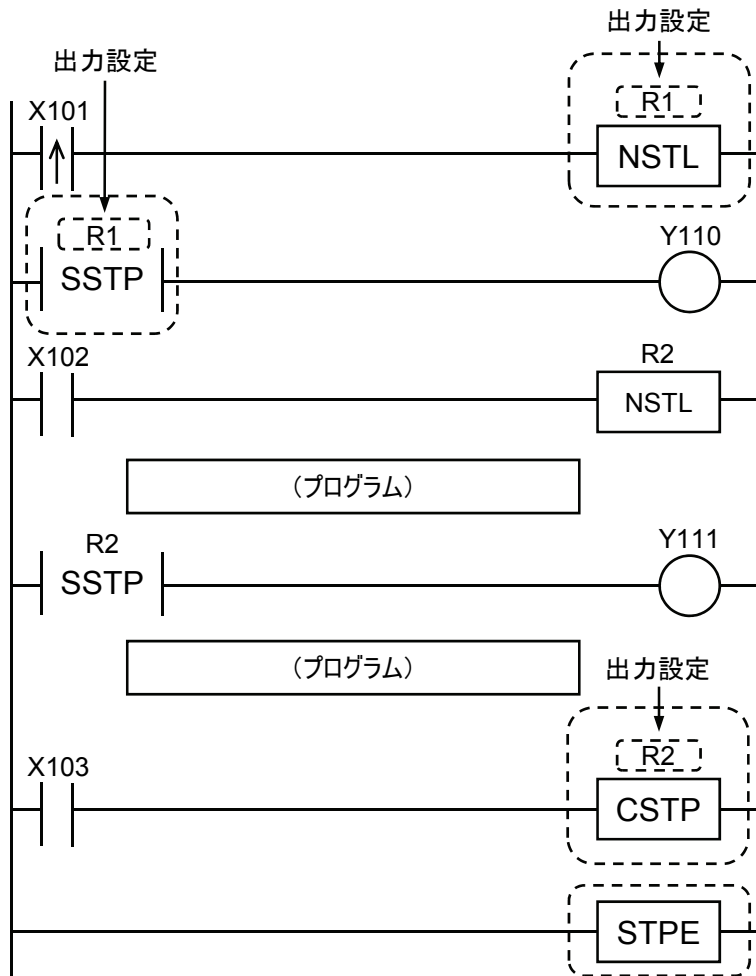
# SSTP (スタートステップ)

# NSTL (ネクストステップ)

# CSTP (クリアステップ)

# STPE (ステップエンド)

## ■ ラダー表記



## ■ 指定可能なデバイス (●:指定可能)

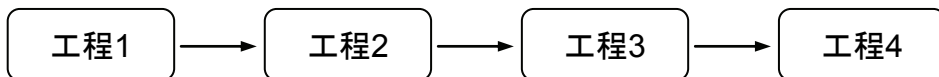
オペランド	ビットデバイス											ワードデバイスのビット指定		インデックス修飾
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b	LD.b	
bit	●	●	●	●								●	●	

## ■ 動作説明

- 「NSTL」命令を実行すると、「SSTP」命令から始まる指定の番号の工程を起動し、実行します。
- 「SSTP」命令から次の「SSTP」命令または「STPE」命令までのプログラムが、ひとつの工程です。
- 順序制御、選択分岐制御、並列分岐合流制御などが容易に実現できます。

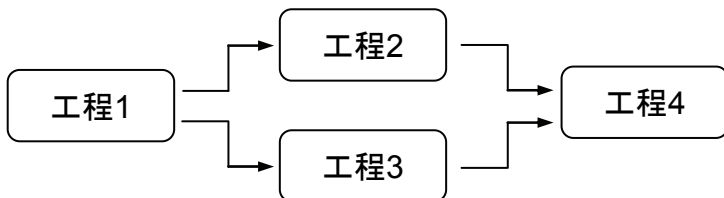
### 1) 順序制御

必要な工程のみを順次、切り替えて実行します。



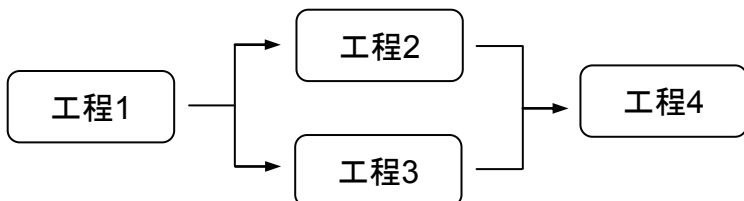
### 2) 選択分岐制御

条件に応じて、工程を選択して実行します。



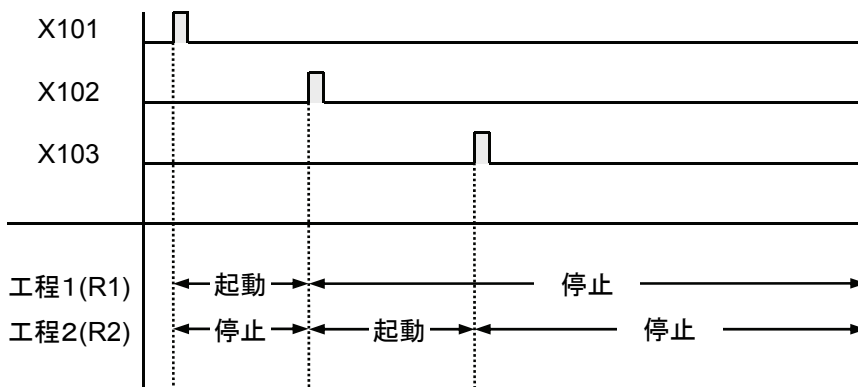
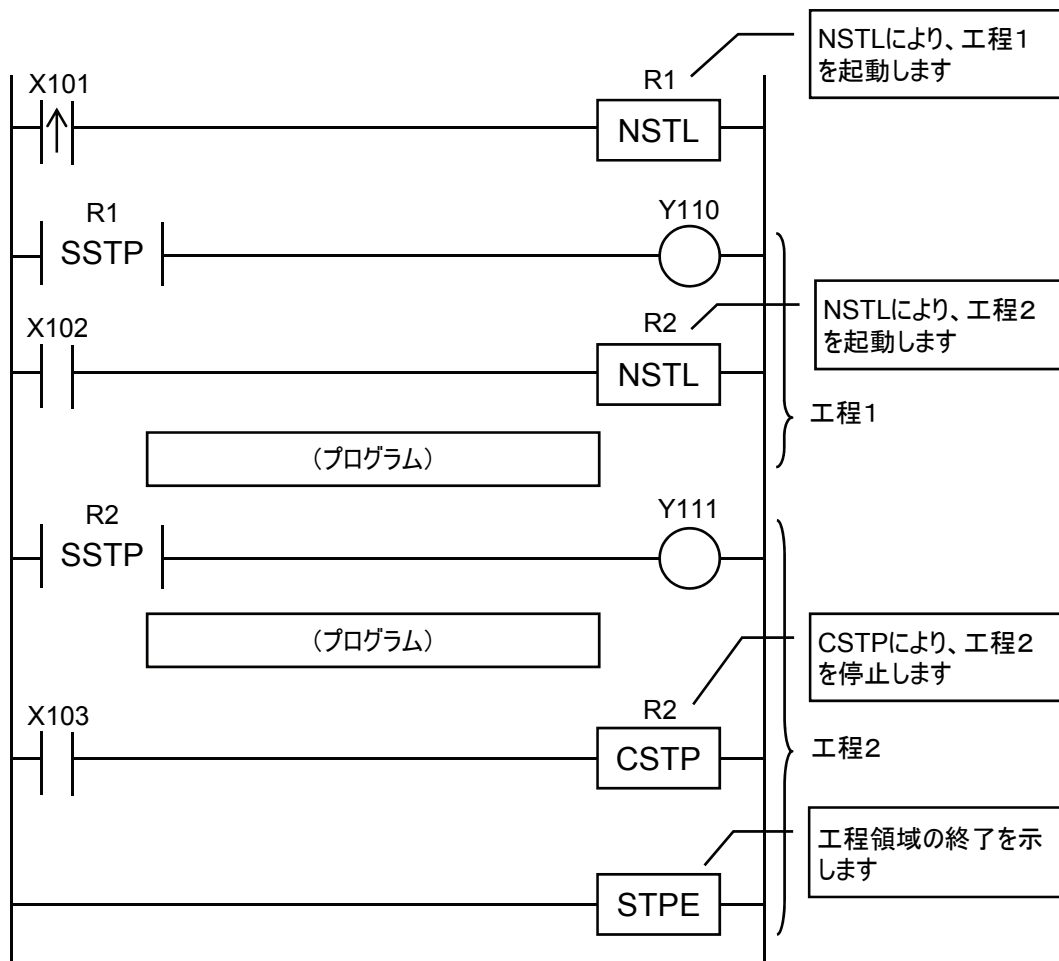
### 3) 並列分岐合流制御

複数の工程を同時に実行することができます。複数の工程の各々が完了してから次の工程を実行することができます。





■ 動作例



## ■ SSTP スタートステップ命令

- 「出力番号[n]の開始」を示します。出力番号[n]のプログラムの先頭に必ず「SSTP n」を記述してください。
- 「SSTP n」から次の「SSTP」または「STPE」までが出力番号[n]の領域になります。
- 同じ番号の工程を二重に定義することはできません。
- 「SSTP」命令の直後では、「OUT」命令を直接母線から接続することができます。
- 「SSTP」命令は、副プログラム中(サブルーチン)には記述できません。
- 最初に記述されている「SSTP」命令から「STPE」命令までを「ステップラダー領域」といい、この間のプログラムはすべて工程として制御されます。これに対して、その他の領域を「通常ラダー領域」といいます。
- ステップラダーのある工程を起動した時に 1 スキャンだけ ON するシステムリレーがあります。(SR15:ステップラダーイニシャルパルスリレー)。このリレーは、カウンタのリセットなど工程起動 1 スキャンだけ処理をさせる場合に利用できます。

## ■ NSTL ネクストステップ命令(毎スキャン実行型)

- 「NSTL R」命令が実行されると、リレー番号[R]で指定された工程が起動します。
- ネクストステップ命令の実行条件は、工程の起動条件になります。
- 最初に起動する工程については、ネクストステップ命令を通常ラダー領域に書き込んでください。
- 工程は通常ラダー領域からでも起動している工程からでも起動できます。
- ただし、ある工程のなかで別の工程を起動するネクストステップ命令を実行した場合は、その命令を含む起動中の工程を自動的にクリアし、指定の工程を起動します。
- 「NSTL」命令を MC～MCE 命令、JP～LBL 命令など命令を実行する順序を変える命令(下記 1～6)と合わせて使用する場合、命令の実行と実行条件のタイミングにより、命令の動作が変わりますのでご注意ください。

1) MC～MCE 命令

2) JP～LBL 命令

3) LOOP～LBL 命令

4) CNDE 命令

5) ステップラダー命令

6) サブルーチン命令

- NSTL 命令を、アンドスタック命令やポップスタック命令と組み合わせるときは、正しくない記述にならないようご注意ください。

## ■ CSTP クリアステップ命令

- 「CSTPn」命令が実行されると、n で指定された工程 n がクリアされます。最終工程のクリアや、並列分岐合流制御時に並列で起動している各工程をクリアする場合に使用します。
- 工程のクリアは、通常ラダー領域からでも起動している工程からでも実行できます。

## ■ STPE ステップエンド命令

- 「ステップラダー領域の終了」を示します。最後に記述する工程の末尾に必ず書き込んでください。最終工程は、「SSTP」から「STPE」までの間になります。
- 「STPE」命令は、1 プログラム中に 1 回のみ、主プログラムに記述することができます。(サブルーチンプログラム、割り込みプログラムなどの副プログラム中に記述することができません。)

## ■ プログラム上のご注意

- 工程は、番号の順序で記述する必要はありません。
- ステップラダー領域中では、次の命令語が使用できませんので、ご注意ください。

- 1) ジャンプ命令 (「JP」・「LBL」)
- 2) ループ命令 (「LOOP」・「LBL」)
- 3) マスタコントロール命令 (「MC」・「MCE」)
- 4) サブルーチン命令 (「SBL」・「RET」)
- 5) 「ED」命令
- 6) 「CNDE」命令

注) 「CALL」命令は、ステップラダー領域中で使用できます。

- 全ての工程をまとめてクリアする場合、マスタコントロールリレーを利用してプログラムしてください。
- 工程の起動は、番号の順序で行う必要はありません。複数の工程を同時に起動することもできます。
- 起動していない工程のなかにある出力を強制 ON または強制 OFF した場合、強制を解除しても起動するまで状態を保持します。

## ■ ステップラダーの動作

- プログラムの実行は、通常ラダー領域のプログラムおよびネクストステップ命令 (「NSTL」) で起動された工程中のプログラムについて行われます。起動していない工程中のプログラムは無視されます。
- ある工程が起動し、第 1 スキャンが行われている間、ステップイニシャルパルスリレー (SR15) が ON します。第 2 スキャン以後は OFF になります。カウンタやシフトレジスタのリセットに利用できます。
- 工程の起動／停止状態は、下記の特特殊データレジスタに格納されます。

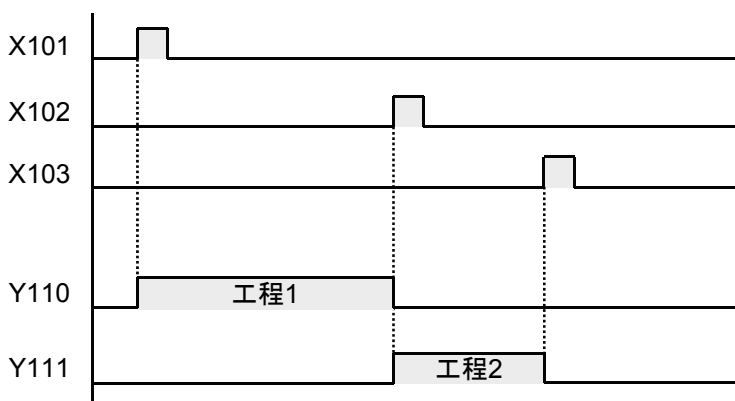
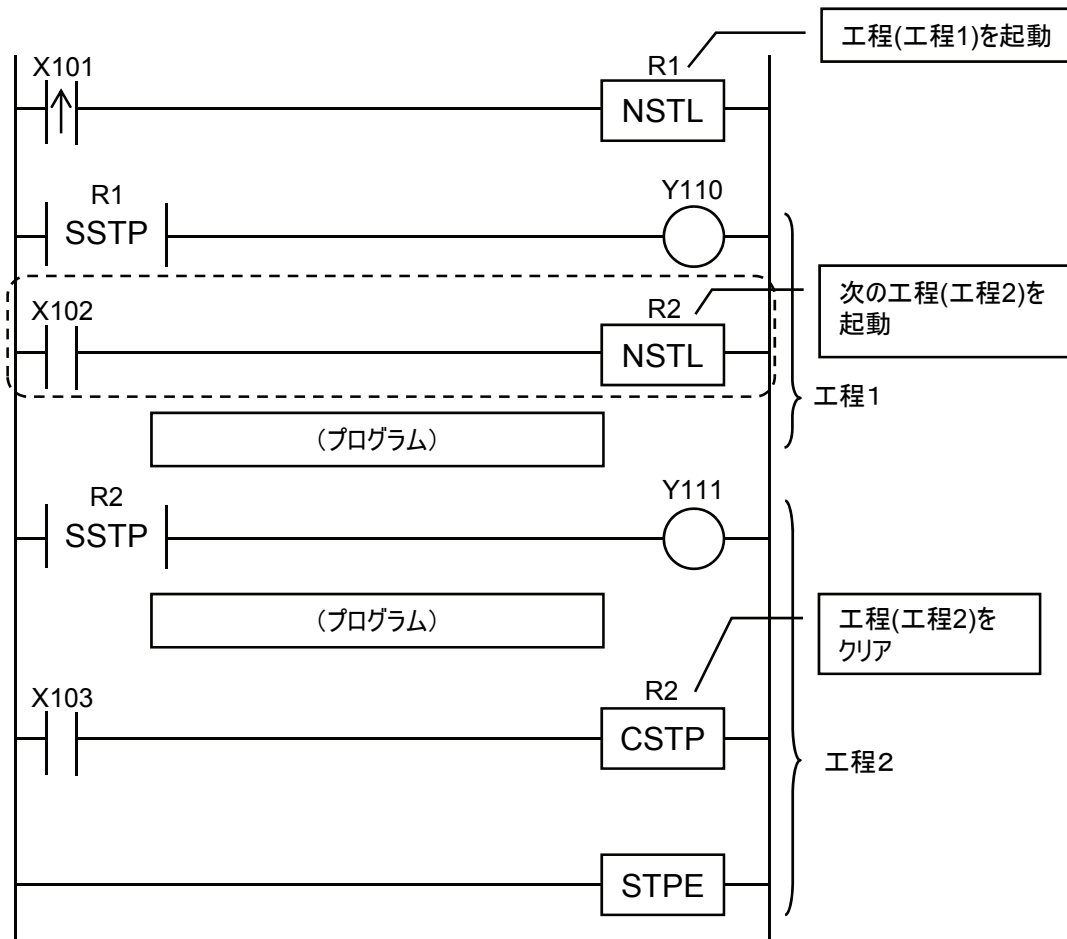
## ■ 工程のクリアについてのご注意

- 起動中の工程のプログラムでネクストステップ命令が実行されると、その起動中の工程を自動的にクリアしますが、実際にクリア動作が行われるのは、次のスキャン時になります。このため、工程が移行するときに 1 スキャンだけ 2 つの工程が同時に起動していることがありますので、同時に ON してはならない一組の出力の間では、同時 ON をさけるインタロックを取るようプログラムしてください。プログラム上でインタロックを取っていても、ハードウェア上の応答遅れなどにより同時 ON する可能性がある場合は、応答遅れを考慮したハードウェア上の処置を施してください。
- 工程がクリアされると、その工程で使用している各命令の動作は、次のようになります。
- 微分命令など実行条件の立ち上がりを検出して実行する命令 (下記 1～6) を使用する場合は注意が必要です。

- 1) DF (立ち上がり微分)
- 2) CT (カウンタ) のカウント入力
- 3) UDC (アップダウンカウンタ) のカウント入力
- 4) SR (シフトレジスタ) のシフト入力
- 5) LRSR (左右シフトレジスタ) のシフト入力
- 6) 微分実行型応用命令 (p と命令名で指定する命令)

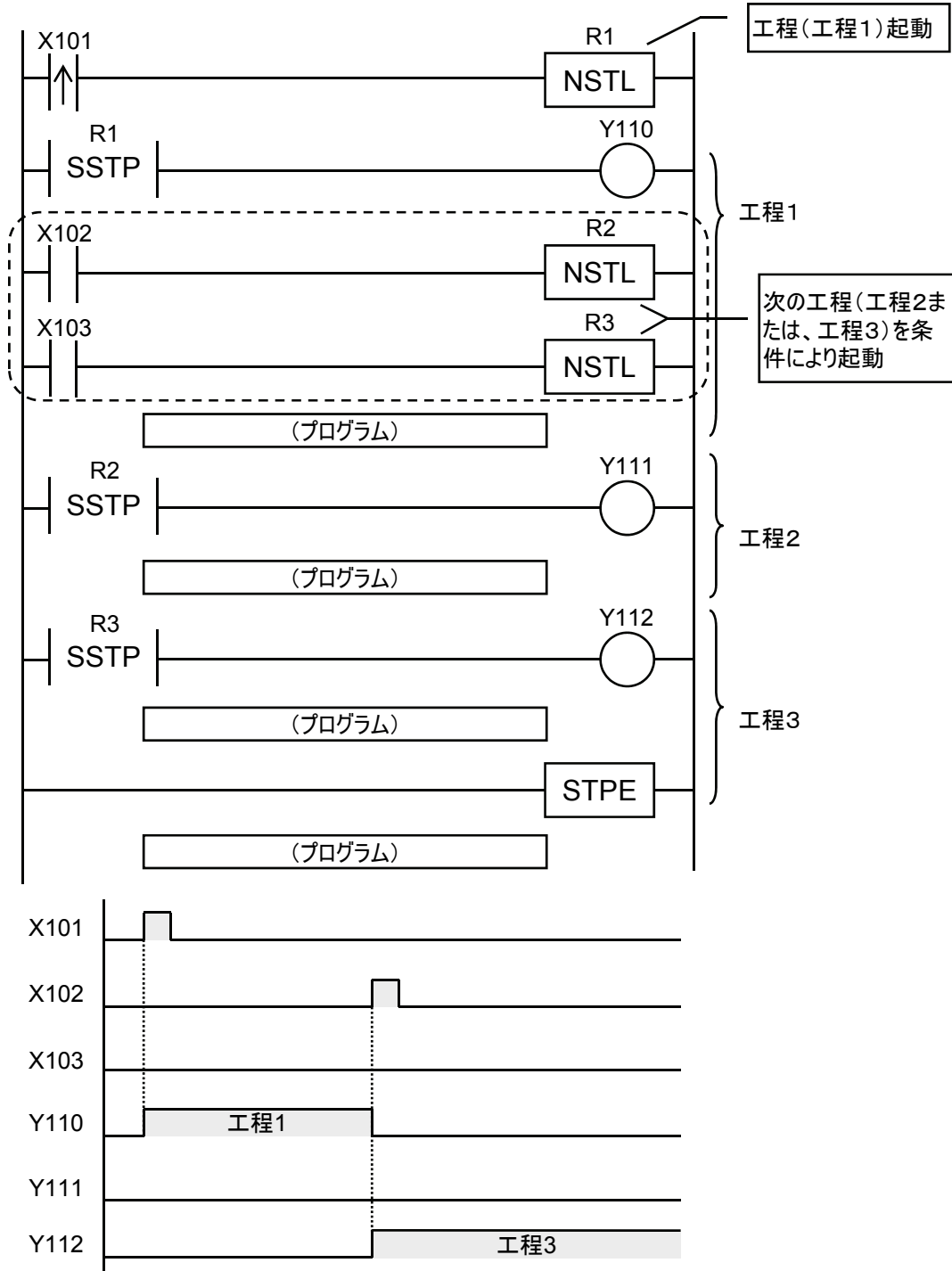
## ■ 使用例 1) 順序制御

- ある工程での作業を終了するまで繰り返し、終了した時点で次の工程に移るプログラムです。
- ある工程のプログラムの中に、次に実行する工程を起動する命令を記述してください。起動命令が実行されると、次の工程を起動し、それまで起動していた工程をクリアします。
- 工程番号の順序で実行する必要はありません。また、条件に応じて、前工程を起動するプログラムも可能です。



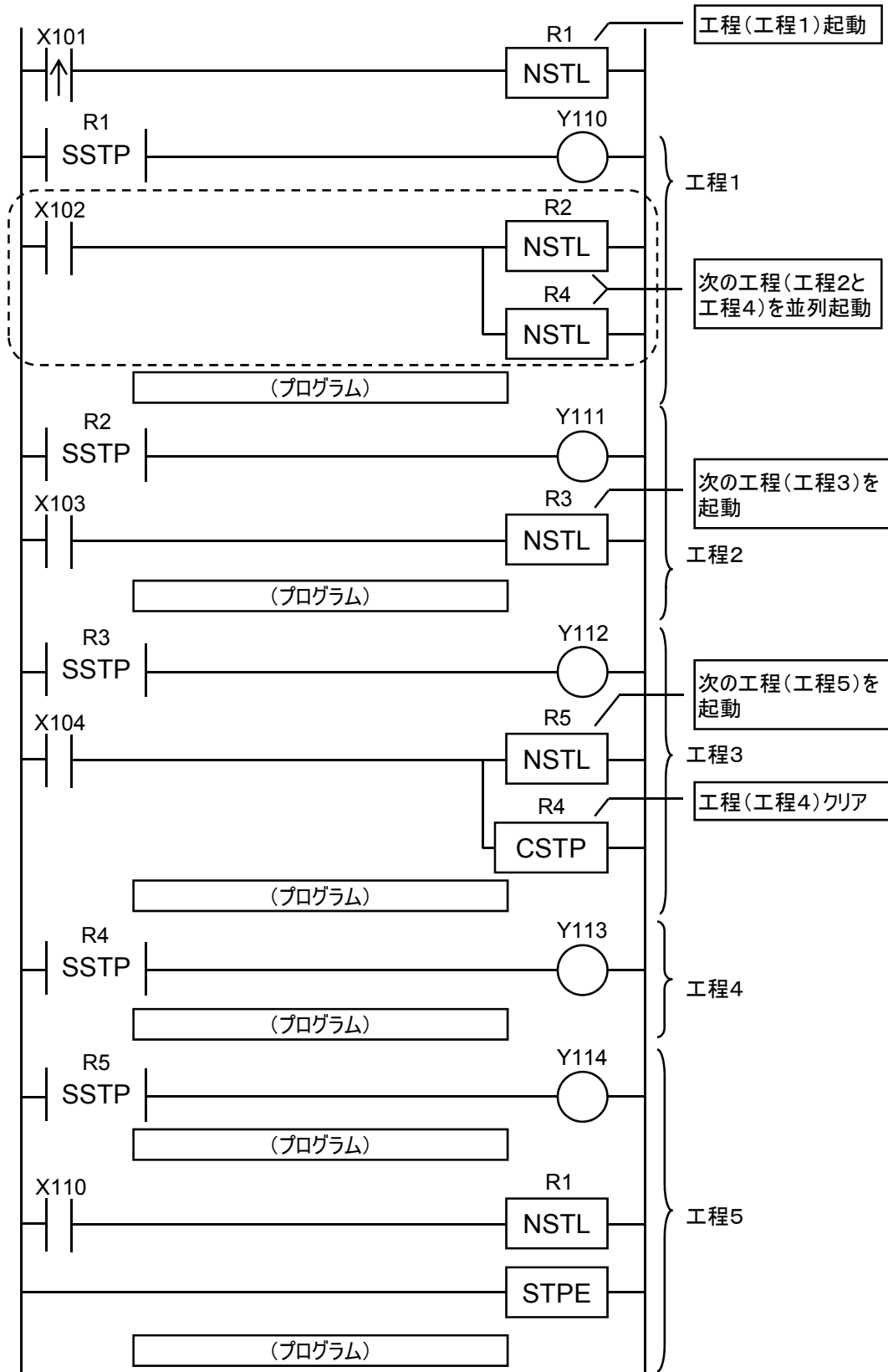
## ■ 使用例 (2) 工程の選択分岐制御

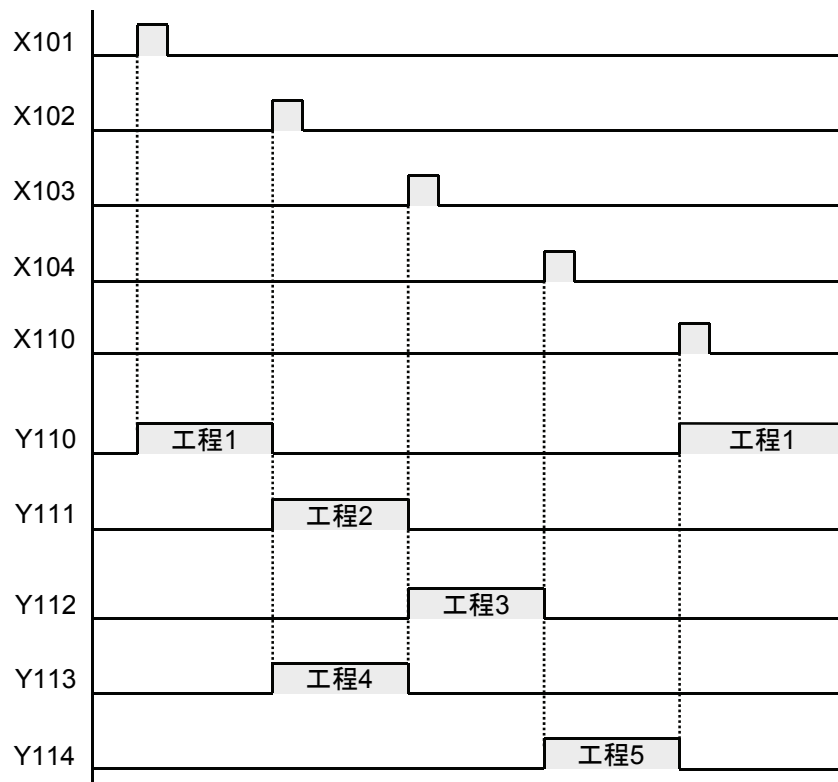
- ある工程での作業内容や結果に応じて、次の工程を選択し、切り替えるプログラムです。各工程では、作業が終了するまで、その工程を繰り返します。
- ある工程のプログラムの中に、次に実行する工程を起動する命令を記述してください。実行条件に応じて、次の工程を選んで起動し、移行します。



### ■ 使用例 (3) 工程の並列分岐合流制御

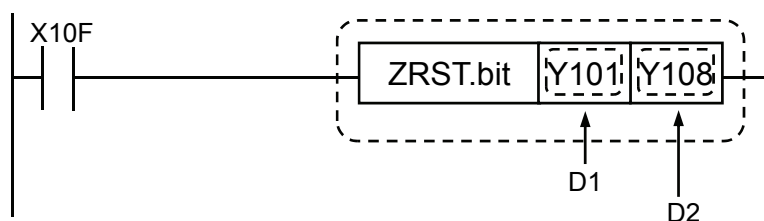
- 複数の工程を同時に起動するプログラムです。分岐したそれぞれの工程で作業が終了した時点で合流し、次の工程に移行します。
- ある工程のプログラムの中に、一つの実行条件に対して、連続して、複数の工程移行命令を記述してください。
- 合流する場合は、次の工程への移行条件に、他の工程の状態を示すフラグを含めます。合流して次工程を起動するとき、同時に、クリアされていない工程をクリアしてください。





# ZRST (ブロッククリア)

## ■ ラダー表記



## ■ 指定可能な演算単位 (○:指定可能 –:指定不可)

演算単位	bit	US	SS	UL	SL	SF	DF
i	●						

## ■ オペランド一覧

オペランド	説明
D1	工程クリア開始番号
D2	工程クリア終了番号

## ■ 指定可能なデバイス (●:指定可能)

オペランド	ビットデバイス											ワードデバイスのビット指定		インデックス修飾
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b	LD.b	
D1	●	●	●	●							●	●	●	●
D2	●	●	●	●							●	●	●	●

## ■ 動作説明

- 「ZRST」命令が実行されると、工程[D1]から工程[D2]までの範囲の起動中の工程がまとめてクリアされます。
- [D1]で指定したエリア(ビットアドレス)から[D2]で指定したエリア(ビットアドレス)までを 0 クリア(リセット)することにも使用できます。

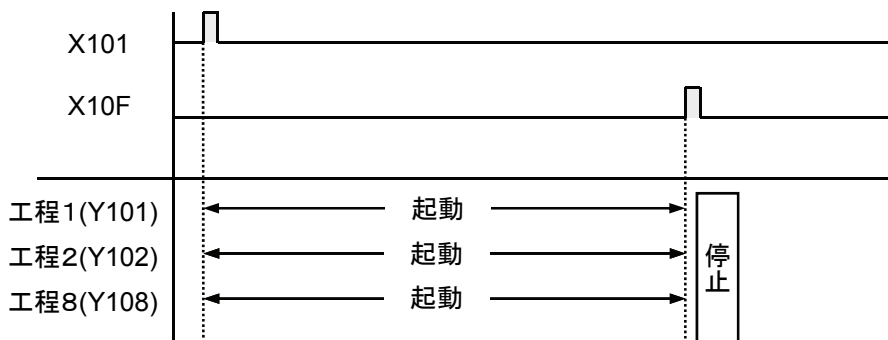
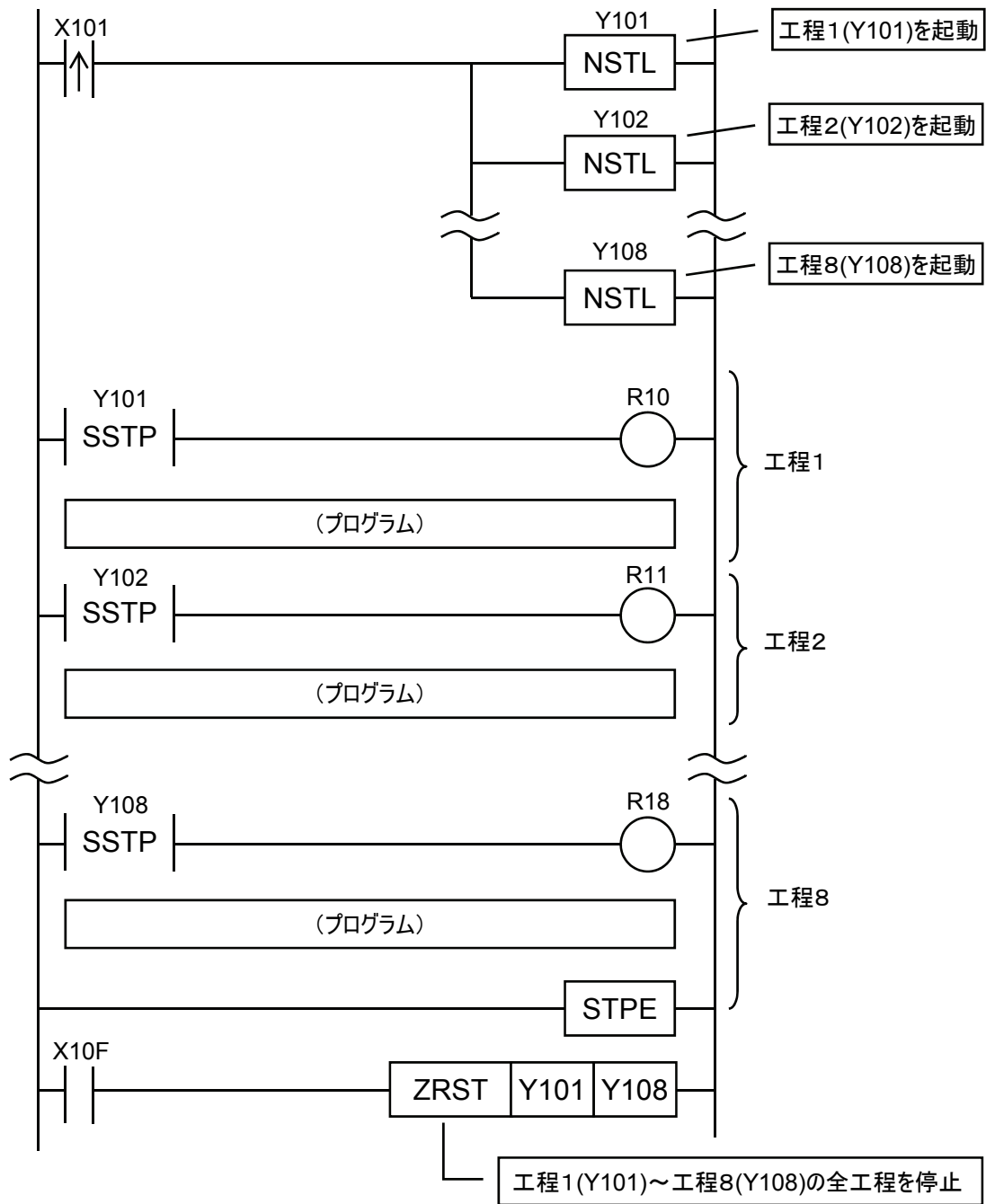
(詳細は、“第3章 応用命令 ZRST”を参照願います。)

## ■ プログラム上のご注意

- [D1]<[D2]となるように設定してください。
- 通常ラダー領域からでも起動している工程からでも実行できます。



■ 動作例

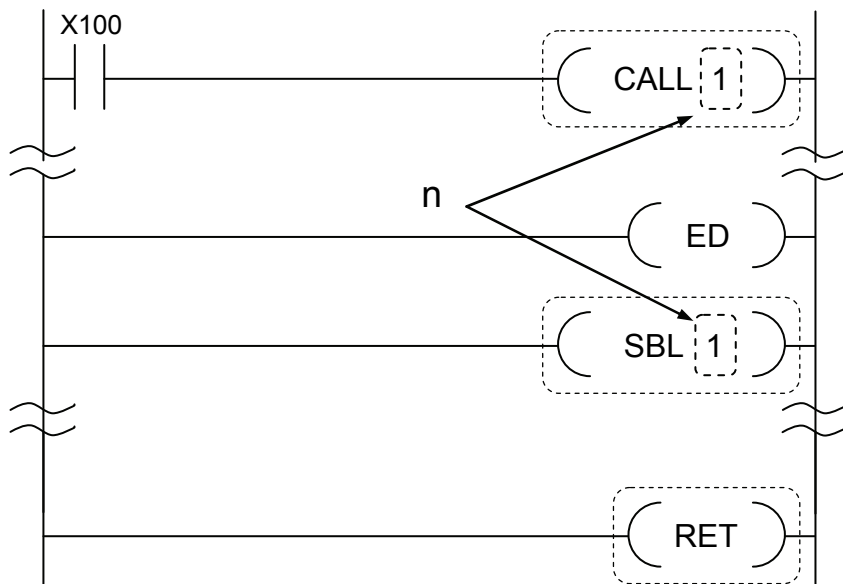


## CALL (サブルーチンコール)

## SBL (サブルーチンラベル)

## RET (サブルーチンリターン)

### ■ ラダー表記



### ■ オペランド一覧

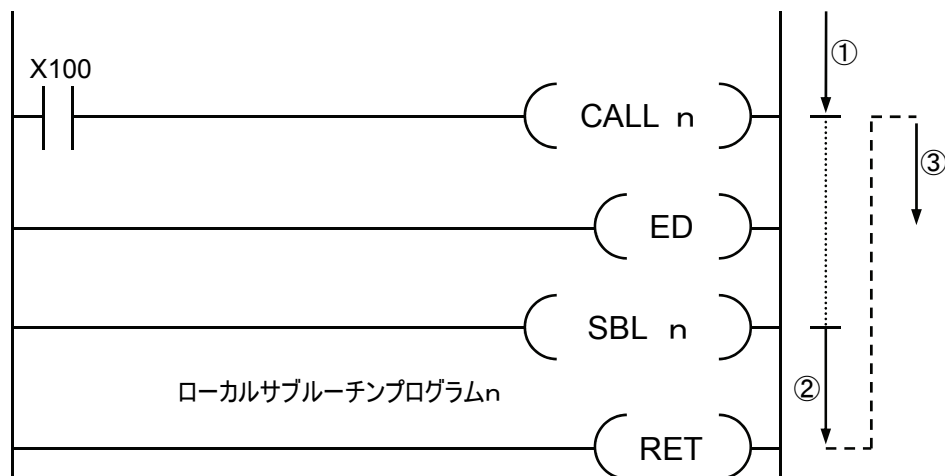
オペランド	説明
n	同一 PB 内のローカルサブルーチンプログラム番号 データ指定可能範囲: 0~65535 (0 から順に詰めて指定することを推奨します)

### ■ 指定可能なデバイス (●: 指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデック ス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		" "
n																●					

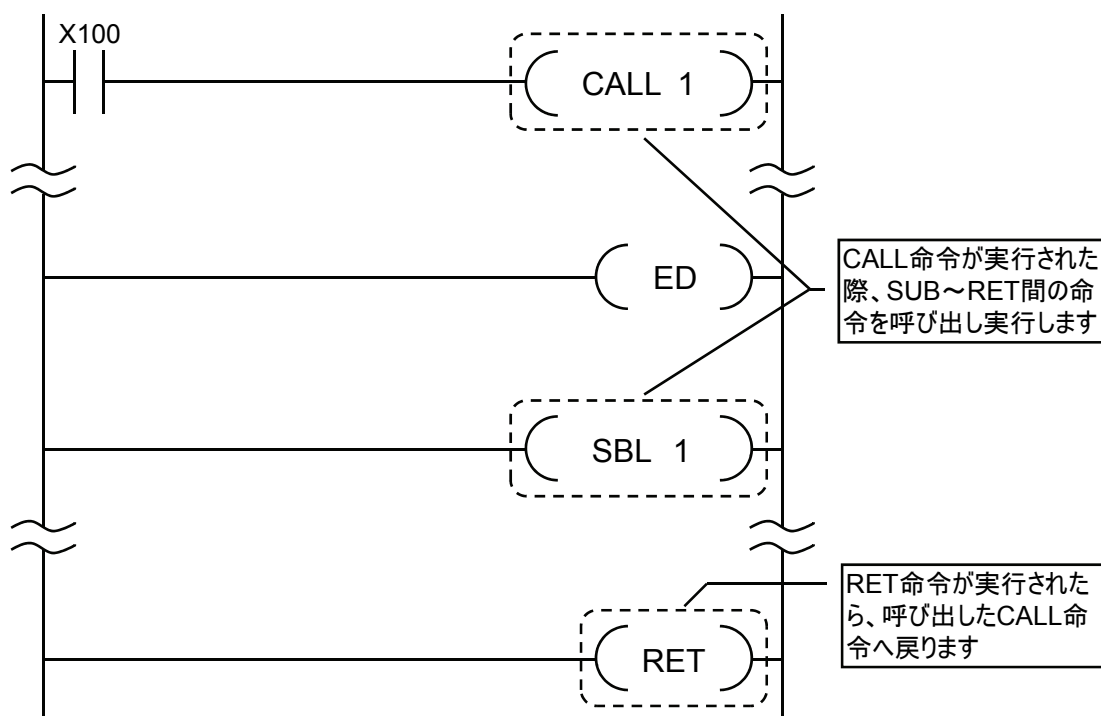
## ■ 動作説明

- 実行条件が ON のとき「CALL」命令を実行し、「SBL」命令から始まる指定の番号のローカルサブルーチンプログラムを実行します。実行条件 OFF 時は、無処理となります。
- 「RET」命令まで実行すると、主プログラムの「CALL」命令の次のアドレスに戻り、主プログラムの実行を続けます。



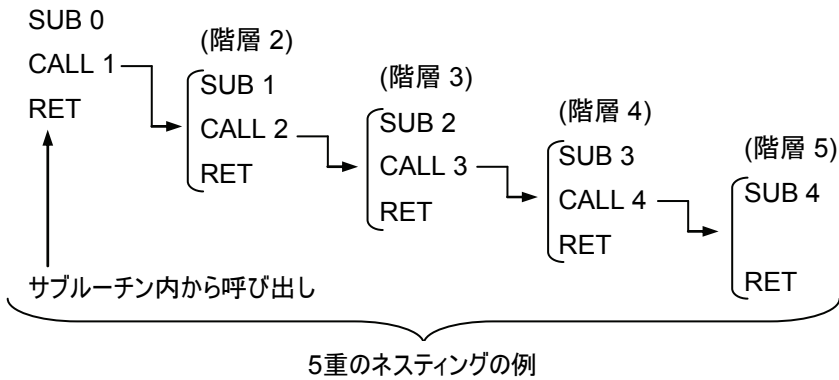
「CALL n」実行時、①～③の順で実行されます

## ■ 動作例



## ■ サブルーチンプログラムの文法

- 「サブルーチンプログラム n」は、「SBL n」命令と「RET」命令までの間のプログラムです。必ず、「ED」命令より後のアドレスに書き込んでください。
- 「CALL n」命令は、主プログラムの他、別のサブルーチンプログラム、ステップラダー中のいずれにも記述可能です。また、同じ番号の「CALL」命令を繰り返し記述することができます。
- サブルーチンのネスティングは、5重まで可能です。



## ■ プログラム上のご注意

• 微分命令など実行条件の立ち上がりを検出して実行する命令(下記 1~6)をサブルーチン内で使用する場合は注意が必要です。

- 1) DF(立ち上がり微分)
- 2) CT(カウンタ)のカウント入力
- 3) UDC(アップダウンカウンタ)のカウント入力
- 4) SR(シフトレジスタ)のシフト入力
- 5) LRSR(左右シフトレジスタ)のシフト入力
- 6) 微分実行型応用命令(p と命令名で指定する命令)

• SBL 命令で n は 0 から順に詰めて使用してください。

## ■ CALL命令の実行条件がOFFの時の動作

• 「CALL」命令の実行条件が OFF になると、そのサブルーチンの演算をしなくなります(マスタコントロールやステップラダーの中のコールでも、同様です。)このときサブルーチンで使用している各命令の動作は次のようになります。

命令の種類	動作
OT	状態を保持します
KP	
SET	
RST	
TM	計時されません。 1スキャン中に1回計時されなければ時間が保証できなくなりますので、ご注意ください。
CT	途中経過を保持します。
SR	
微分命令	MC~MCE命令間に微分命令を使用した場合と同様です。 <a href="#">MC~MCE間の微分命令の動作</a> の項をご参照ください。
その他の命令	実行しません。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	16 重のネスタイングを行っている場合に、16 重目のサブルーチンで「CALL」命令を実行すると、ON

# FCAL（出力OFF型ローカルサブルーチンコール）

## ■ ラダー表記



## ■ オペランド一覧

オペランド	説明
n	ローカルサブルーチンプログラム番号(実行条件 OFF 時 1 スキャンのみ MC 処理)

## ■ 指定可能なデバイス（●: 指定可能）

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデック ス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U *1	H	SF	DF		" "
n																●					

\*1: 演算単位が符号なし整数 (US, UL) の場合のみ指定可能

## ■ 動作説明

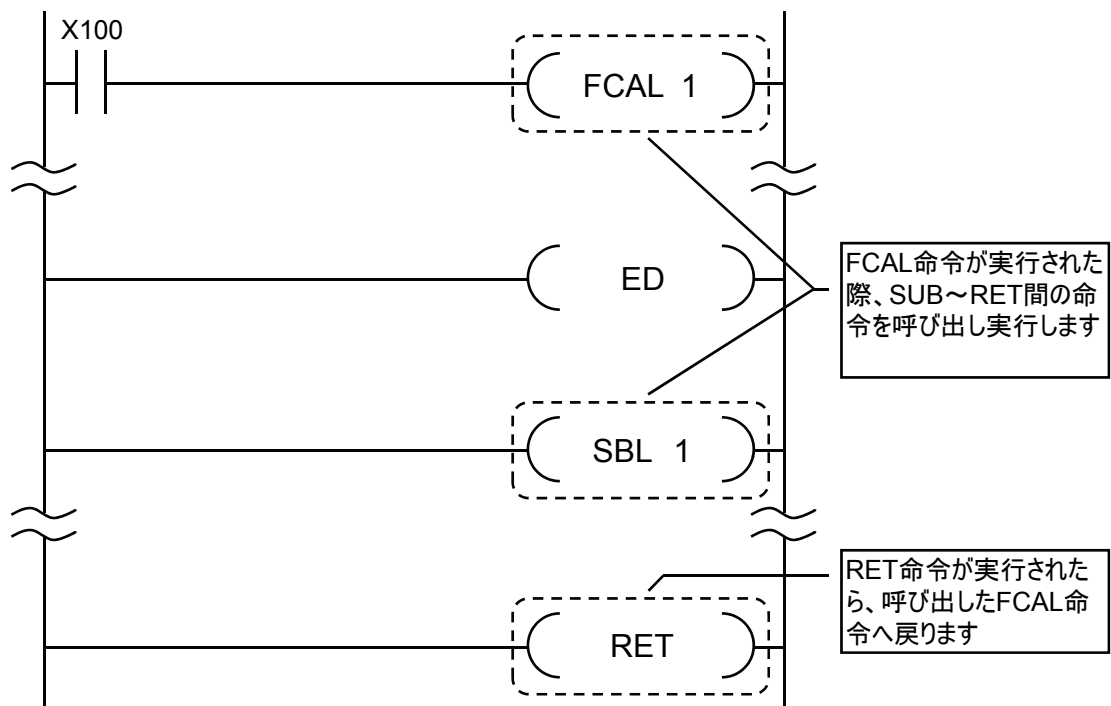
- 動作、文法は通常のローカルサブルーチンコール命令と同じです。ただし、以下の点が異なります。

### ■ 「FCAL」命令の条件がOFFの時の動作

- サブルーチン実行条件が OFF になると、そのサブルーチンの演算は行われませんが、そのサブルーチンで使用している命令の動作は以下のようになります。

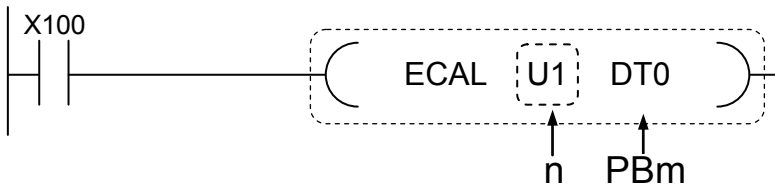
命令の種類	動作
OT	すべて OFF にします。CALL 命令の場合と動作と異なります。
KP	状態を保持します
SET	
RST	
TM	リセットします。CALL 命令の場合と動作と異なります。
CT	途中経過を保持します。
SR	
微分命令	MC～MCE命令間に微分命令を使用した場合と同様です。 <a href="#">MC～MCE間の微分命令の動作</a> の項をご参照ください。
その他の命令	実行しません。

■ 動作例



# ECAL (サブルーチンコール(PB番号指定))

## ■ ラダー表記



## ■ オペランド一覧

オペランド	説明
n	サブルーチン番号:0~65535 / 1PB
PBm	対象 PB 番号: n で指定したサブルーチンが格納されているPB番号

## ■ 指定可能なデバイス (●:指定可能)

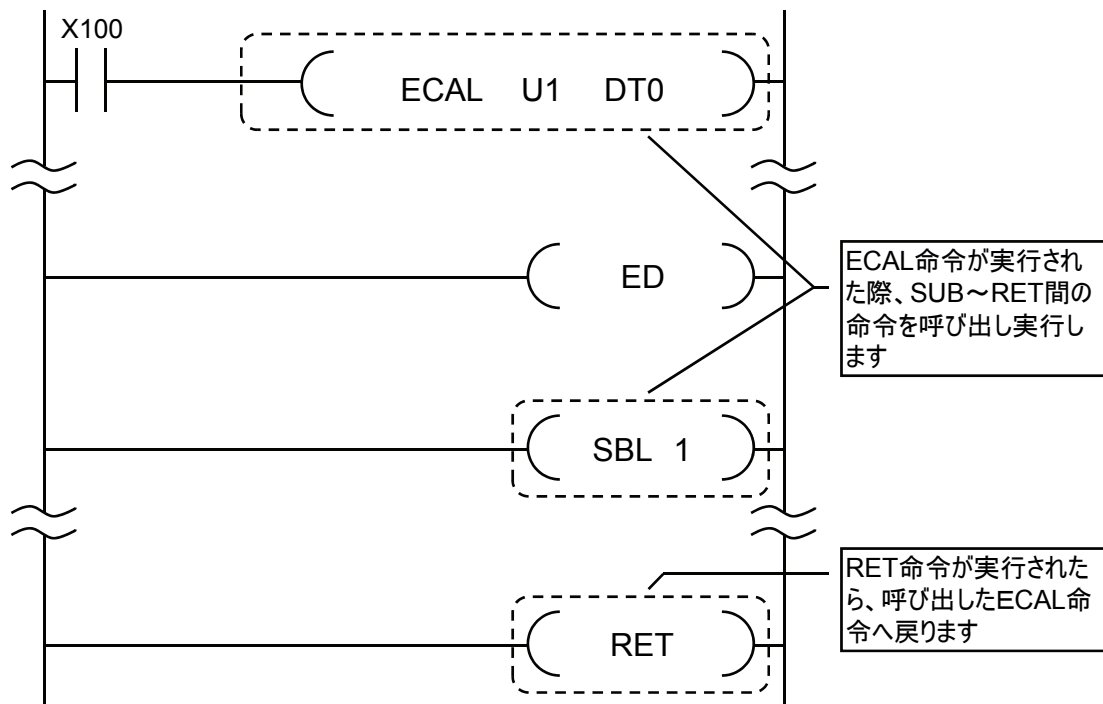
オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデックス 修飾
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""	
n																●					
PBm	●	●	●	●			●	●								●					

## ■ 動作説明

- 動作、文法は通常のローカルサブルーチンコール命令と同じです。
- 実行条件がONの時、PBm の SBLn サブルーチン呼び出します。( PBm の SBLn へ分岐します)
- RET 命令実行により呼び出した ECAL 命令の次のアドレスに戻ります。
- 呼び出し先のサブルーチンで使用されているローカルデバイスは、呼び出し先の PBm のローカルデバイスが使用されます。
- 何回もサブルーチンコールを実行すると実行するプログラムの量が増えるので注意が必要です。
- 実行条件が OFF の時、サブルーチンを実行しなくなるだけです。サブルーチン中の出力や実行状態はすべて保持されます
- 但し、EFCAL 強制コール命令で呼び出した場合、実行条件 OFF 時には、サブルーチン内の出力を OFF します。この動作は、マスターコントロールの動作と同様の動作となります。



## ■ 動作例

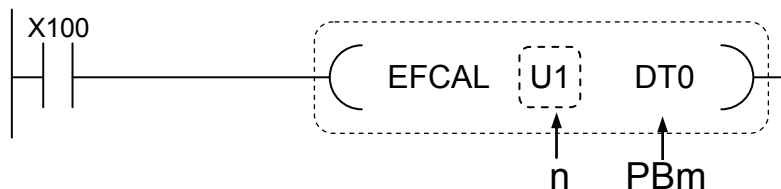


## ■ 使用上のご注意

- サブルーチン内から別のサブルーチンを呼び出すことができます。
- 次々別のサブルーチンを呼び出す場合、最大で16重まで実行可能ですが17以上になると演算エラーとなります。

# EFCAL (出力強制OFF型サブルーチンコール(PB番号指定))

## ■ ラダー表記



## ■ オペランド一覧

オペランド	説明
n	番号 : 0~65535/1PB
PBm	対象 PB 番号 : n で指定したサブルーチンが格納されているPB番号

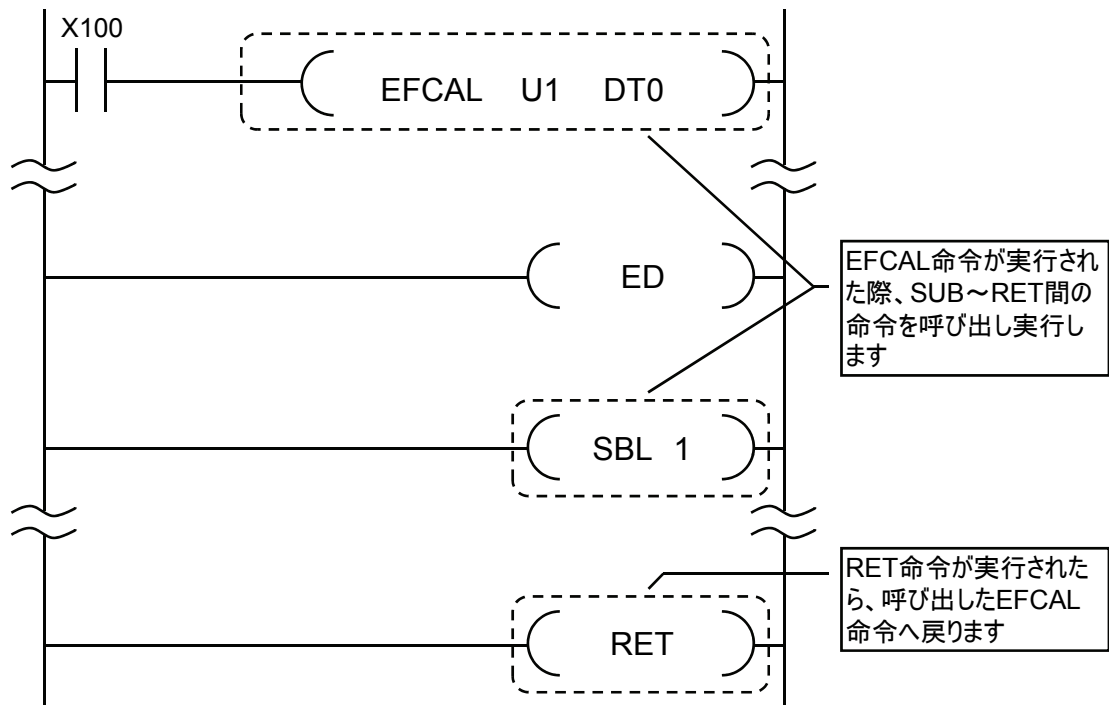
## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデックス 修飾
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""	
n																●					
PBm	●	●	●	●			●	●								●					

## ■ 動作説明

- 動作、文法は通常のローカルサブルーチンコール命令と同じです。
- 実行条件が ON の時、PBm の SBLn サブルーチン呼び出します。( PBm の SBLn へ分岐します)
- RET 命令実行により呼び出した EFCAL 命令の次のアドレスに戻ります。
- 呼び出し先のサブルーチンで使用されているローカルデバイスは、呼び出し先の PBm のローカルデバイスが使用されます。
- 何回もサブルーチンコールを実行すると実行するプログラムの量が増えるので注意が必要です。
- EFCAL 強制コール命令で呼び出した場合、実行条件 OFF 時には、サブルーチン内の出力を OFF します。この動作は、マスターコントロールの動作と同様の動作となります。

## ■ 動作例



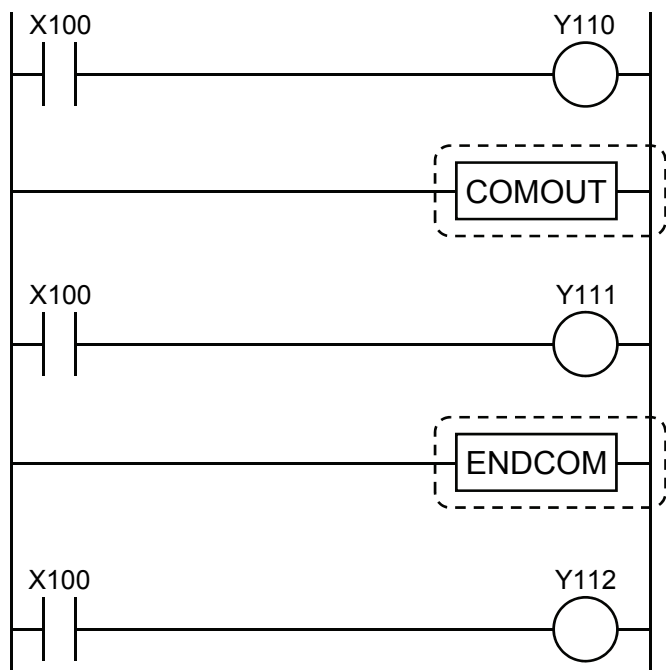
## ■ 使用上のご注意

- サブルーチン内から別のサブルーチンを呼び出すことができます。
- 次々別のサブルーチンを呼び出す場合、最大で16重まで実行可能ですが17以上になると演算エラーとなります。

## COMOUT (コメントアウト)

## ENDCOM (コメントアウト終了)

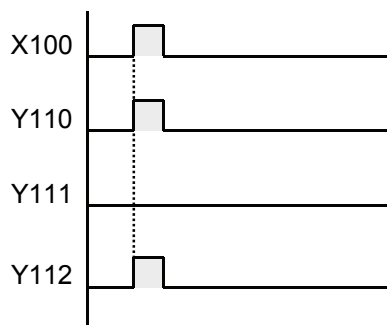
### ■ ラダー表記



### ■ 動作説明

- 「COMOUT」命令～「ENDCOM」命令の間の命令をコメントアウトします。

### ■ 動作例

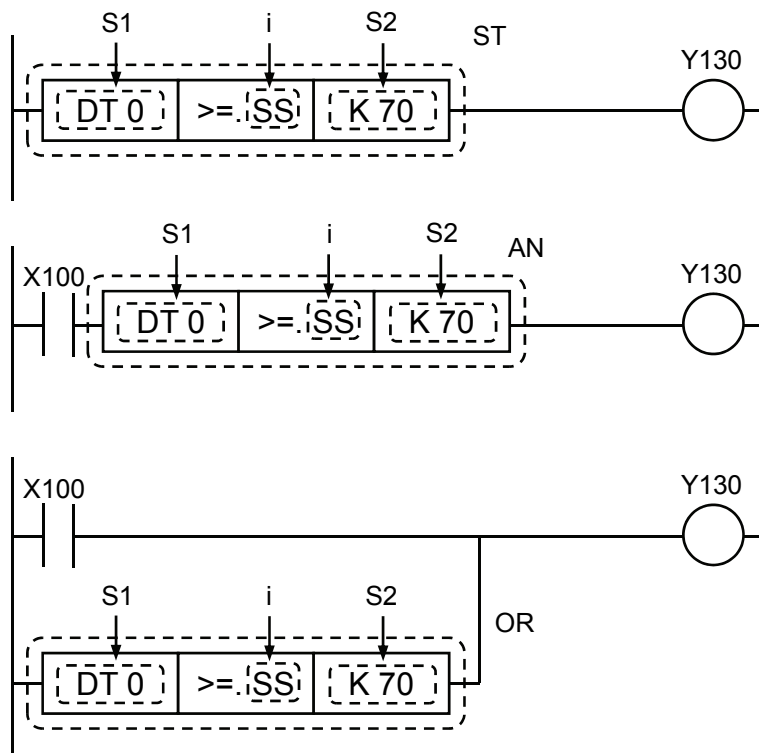


**ST=,ST<>,ST>,ST>=,ST<,ST<=**  
**(データ比較(スタート))**

**AN=,AN<>,AN>,AN>=,AN<,AN<=**  
**(データ比較(アンド))**

**OR=,OR<>,OR>,OR>=,OR<,OR<=**  
**(データ比較(オア))**

■ ラダー表記



■ 指定可能な演算単位 (○:指定可能 -:指定不可)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

■ オペランド一覧

オペランド	説明
S1	比較データ1
S2	比較データ2

## ■ 指定可能なデバイス (●: 指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8	..	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

\*1: 演算単位が16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

\*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*8: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

命令の種類	動作
ST= ST<> ST> ST>= ST< ST<=	[S1]で指定した符号付きデータと[S2]で指定した符号付きデータを比較します。 比較結果が指定の状態(=,<,>,…)のときに導通する接点として論理演算を開始します。
AN= AN<> AN> AN>= AN< AN<=	[S1]で指定した符号付きデータと[S2]で指定した符号付きデータを比較します。 比較結果が指定の状態(=,<,>,…)のときに導通する接点として直列接続します。
OR= OR<> OR> OR>= OR< OR<=	[S1]で指定した符号付きデータと[S2]で指定した符号付きデータを比較します。 比較結果が指定の状態(=,<,>,…)のときに導通する接点として並列接続します。

## ■ 比較結果と動作

比較命令	[S1]と[S2]の関係		
	[S1] < [S2]	[S1] = [S2]	[S1] > [S2]
ST=,AN,OR=	OFF	ON	OFF
ST<>,AN<>,OR<>	ON	OFF	ON
ST>,AN>,OR>	OFF	OFF	ON
ST>=,AN>=,OR>=	OFF	ON	ON
ST<>,AN<>,OR<>	ON	OFF	OFF
ST<=,AN<=,OR<=	ON	ON	OFF

注)「<>」は、「≠」を表します

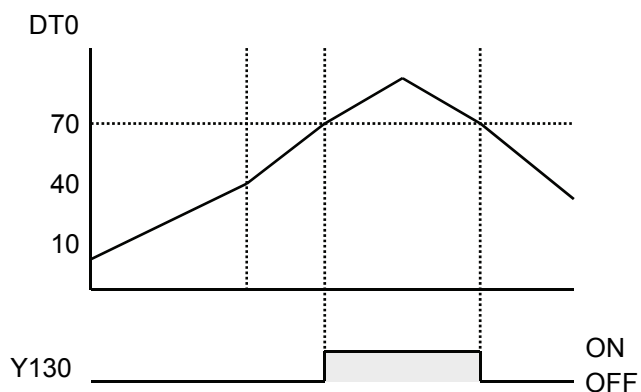
「>=」は、「≧」を表します

「<=」は、「≦」を表します

## ■ 動作例

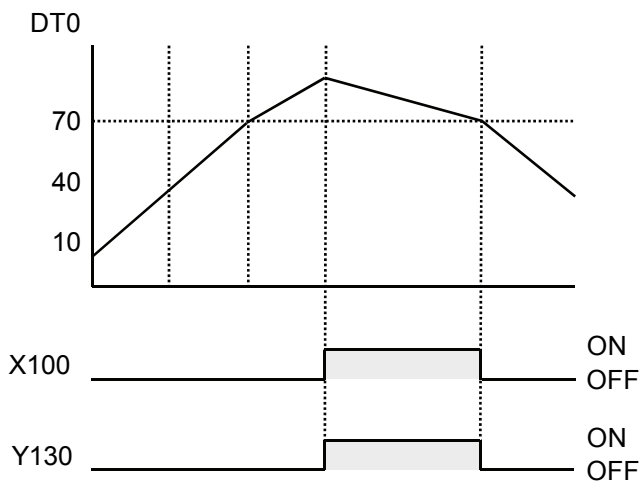
### (1) ラダー表記にある「ST >=」のプログラム動作

データレジスタDT0の値とK70を比較し、 $DT0 > K70$ のとき、外部出力Y130がONになります。



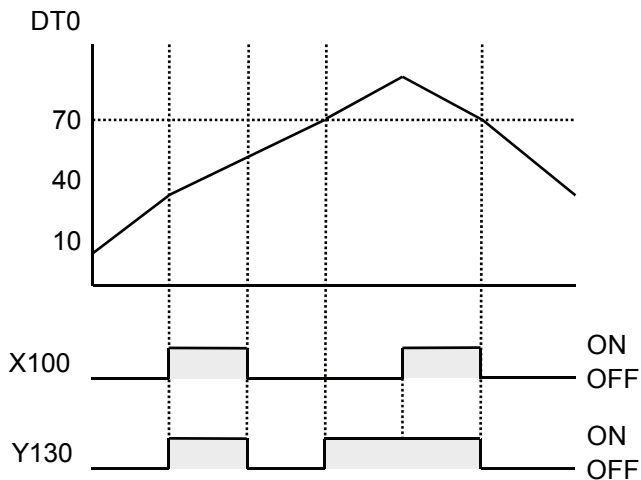
### (2) ラダー表記にある「AN >=」のプログラム動作

外部出力X100がONで、DT0の値とK70を比較し、 $DT0 > K70$ のとき、外部出力Y130がONになります。



### (3) ラダー表記にある「OR >=」のプログラム動作

外部出力X100がONのとき、またはDT0の値とK70の比較結果が $DT0 > K70$ のとき、外部出力Y130がONになります。



## ■ 使用上のご注意

- 「ST=」、「ST<>」、「ST>」、「ST>=」、「ST<」、「ST<=」、「OR=」、「OR<>」、「OR>」、「OR>=」、「OR<」、「OR<=」命令は、母線から始まります。
- 「AN=」、「AN<>」、「AN>」、「AN>=」、「AN<」、「AN<=」、「OR=」、「OR<>」、「OR>」、「OR>=」、「OR<」、「OR<=」命令は、連続して使用できます。
- BCD データの場合は、最上位ビットが 1 のときは負の値とみなして比較しますので、正しい比較結果が得られないおそれがあります。このような場合は、「BIN」命令などでバイナリデータに変換してから比較してください。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	インデックス修飾時にエリアを超えたとき ON

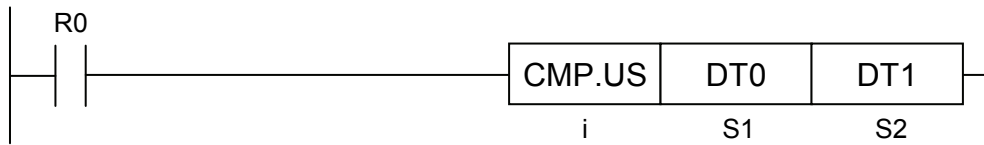


# 3

## 応用命令

# CMP (データ比較)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	比較データ 1(デバイスアドレスまたは定数)
S2	比較データ 2(デバイスアドレスまたは定数)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8	""	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0～IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

\*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*8: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S1]と[S2]をそれぞれ先頭とするエリアに格納されているデータの比較を行います。
- 比較結果は、システムリレーSRA～SRC(比較命令の判定フラグ)に出力します。

## ■ 処理内容

- [S1]と[S2]の関係により比較フラグ(SR(システムリレー))の結果は下記ようになります

[S1]と[S2]の 関係	比較フラグ(SR(システムリレー))		
	SRA	SRB	SRC
	>	=	<
[S1] < [S2]	OFF	OFF	ON
[S1] = [S2]	OFF	ON	OFF
[S1] > [S2]	ON	OFF	OFF

- 例1) 演算単位が符号なし16ビット(US)の場合(SRC(<) ON)

[i]...US

[S1]...DT0 [S2]...DT1

	16進	符号なし10進	符号付き10進
DT0	H 6000	<b>K 24576</b>	K 24576
DT1	H 8500	<b>K 34048</b>	K -31488

実行時のフラグ動作

	SRA(>)	SRB(=)	SRC(<)
DT0 < DT1	OFF	OFF	ON

- 例2) 演算単位が符号付き16ビット(SS)の場合(SRA(>) ON)

[i]...SS

[S1]...DT0 [S2]...DT1

	16進	符号なし10進	符号付き10進
DT0	H 6000	K 24576	<b>K 24576</b>
DT1	H 8500	K 34048	<b>K -31488</b>

実行時のフラグ動作

	SRA(>)	SRB(=)	SRC(<)
DT0 > DT1	ON	OFF	OFF

- 例3) 演算単位が符号なし16ビット(US)の場合(SRB(=) ON)

[i]...US

[S1]...DT0 [S2]...DT1

	16進	符号なし10進	符号付き10進
DT0	H 1234	<b>K 4660</b>	K 4660
DT1	H 1234	<b>K 4660</b>	K 4660

実行時のフラグ動作

	SRA(>)	SRB(=)	SRC(<)
DT0 = DT1	OFF	ON	OFF

例4) 演算単位が符号付き32ビット(SL)の場合(SRC(<) ON)

[i]...SL  
[S1]...I0 [S2]...TS0

	16進	符号なし10進	符号付き10進
I0	H 85000000	K 2231369728	<b>K -2063597568</b>
TS0	H 60000000	K 1610612736	<b>K 1610612736</b>

実行時のフラグ動作

	SRA(>)	SRB(=)	SRC(<)
I0 < TS0	OFF	OFF	ON

例5) 演算単位が単精度浮動小数点形実数(SF)の場合(SRA(>) ON)

[i]...SF  
[S1]...DT0 [S2]...LD0

	値(実数10進)
DT0・DT1	<b>SF 1.234E+00</b>
LD0・LD1	<b>SF -1.234E+00</b>

実行時のフラグ動作

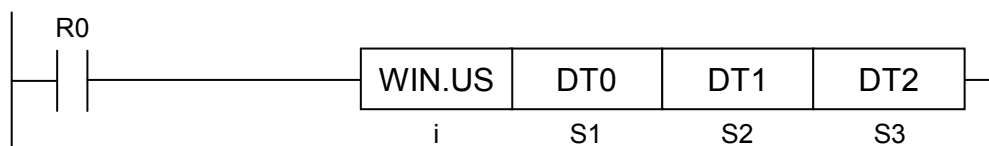
	SRA(>)	SRB(=)	SRC(<)
DT0・DT1 > LD0・LD1	ON	OFF	OFF

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします 演算単位が実数(SF)の場合、[S1] または [S2]に非実数を指定した時にセットします
SRA(>)	比較結果により変動します
SRB(=)	
SRC(<)	

# WIN (帯域比較)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	比較データ(デバイスアドレスまたは定数)
S2	下限値(デバイスアドレスまたは定数)
S3	上限値(デバイスアドレスまたは定数)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8	..	
S1	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●		●
S3	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●		●

\*1: 演算単位が 16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

\*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*8: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S1](比較データ)が[S2](下限値)と[S3](上限値)の範囲内にあるか比較を行います。
- 比較結果は、システムリレーSRA~SRC(比較命令の判定フラグ)に出力します。

## ■ 処理内容

- [S1]と[S2]と[S3]の関係によりフラグ(SR(システムリレー))の結果は下記ようになります

[S1]と[S2]と[S3] の関係	比較フラグ(SR(システムリレー))		
	SRA	SRB	SRC
	>	=	<
[S1] < [S2]	OFF	OFF	ON
[S2] ≤ [S1] ≤ [S3]	OFF	ON	OFF
[S1] > [S3]	ON	OFF	OFF

- 例1) 演算単位が符号なし16ビット(US)の場合(SRB(=) ON)

[i]...US

[S1]...DT0 [S2]...DT1 [S3]...DT2

	16進	符号なし10進	符号付き10進
DT0	H 8000	<b>K 32768</b>	K -32767
DT1	H 7000	<b>K 28672</b>	K 28672
DT2	H 9000	<b>K 36864</b>	K -28671

実行時のフラグ動作

	SRA( > )	SRB( = )	SRC( < )
DT1 < DT0 < DT2	OFF	ON	OFF

- 例2) 演算単位が符号なし16ビットの場合(SRC(<) ON)

[i]...US

[S1]...DT0 [S2]...DT1 [S3]...DT2

	16進	符号なし10進	符号付き10進
DT0	H 6000	<b>K 24576</b>	K 24576
DT1	H 7000	<b>K 28672</b>	K 28672
DT2	H 9000	<b>K 36864</b>	K -28671

実行時のフラグ動作

	SRA( > )	SRB( = )	SRC( < )
DT0 < DT1	OFF	OFF	ON

例3) 演算単位が符号付き16ビット(SS)の場合(演算エラー)

[i]...SS  
[S1]...DT0 [S2]...DT1 [S3]...DT2

	16進	符号なし10進	符号付き10進
DT0	H 6000	K 32768	<b>K -32767</b>
DT1	H 7000	K 28672	<b>K 28672</b>
DT2	H 9000	K 36864	<b>K -28671</b>

実行時のフラグ動作

	SRA( > )	SRB( = )	SRC( < )
DT1 > LD2	OFF	OFF	OFF

※[S2] > [S3]のため演算エラー(SR7(最新エラー)、SR8(保持エラー)をセット)

例4) 演算単位が単精度浮動小数点形実数(SF)の場合(SRA(>) ON)

[i]...SF  
[S1]...DT0 [S2]...LD0 [S3]...LD2

	値(実数10進)
DT0・DT1	SF 8.000E+02
LD0・LD1	SF 5.000E+02
LD2・LD3	SF 7.000E+02

実行時のフラグ動作

	SRA( > )	SRB( = )	SRC( < )
DT0・DT1 > LD2・LD3	ON	OFF	OFF

■ プログラム上のご注意

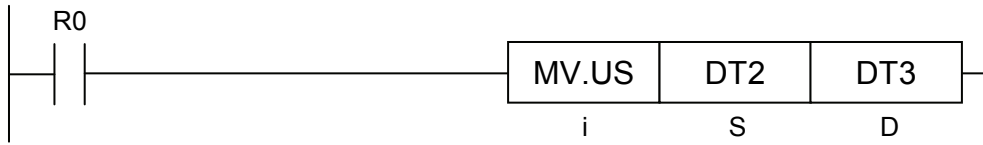
- 直接アドレスとインデックス修飾アドレスの場合、[S3] ≥ [S2]となるように指定してください。

■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S2] > [S3]の場合にセットします
(ER)	演算単位が実数(SF)の場合、[S1] または [S2] または [S3]に非実数を指定した時にセットします
SRA(>)	比較結果により変動します
SRB(=)	
SRC(<)	

# MV (データ転送)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S	転送元のデバイスアドレスまたは定数
D	転送先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8		..
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●	●		●	●	●								●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

\*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*8: 演算単位が単精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [S]で指定したデバイスアドレスまたは定数から[D]で指定したデバイスアドレスへ[i]で指定した演算単位のデータを転送します。

[S] → [D]



## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[S] ...DT2 [D] ...DT3

DT0	H 0011	DT0	H 0011
DT1	H 2233	DT1	H 2233
DT2	<b>H 4455</b>	DT2	H 4455
DT3	H 6677	DT3	<b>H 4455</b>
DT4	H 8899	DT4	H 8899

例2) 演算単位が32ビット(UL,SL,SF)の場合

[i]...UL,SL,SF

[S] ...TS1 [D] ...TS4

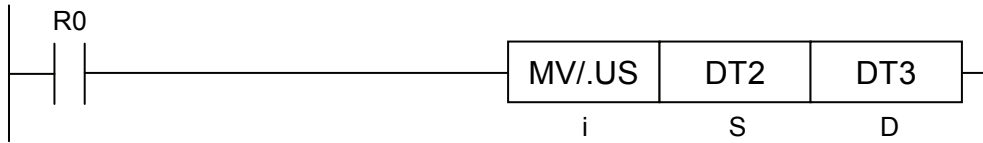
TS0	H 11223344	TS0	H 11223344
TS1	<b>H 55667788</b>	TS1	H 55667788
TS2	H 9900AABB	TS2	H 9900AABB
TS3	H CCDDEEFF	TS3	H CCDDEEFF
TS4	H 12345678	TS4	<b>H 55667788</b>

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# MV/ (データ反転転送)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S	転送元のデバイスアドレスまたは定数
D	転送先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8		..
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

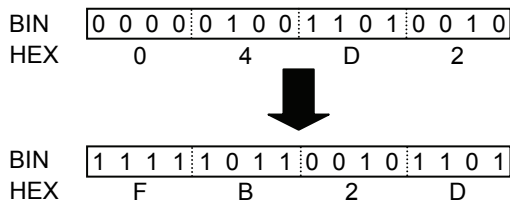
\*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*8: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [S]で指定したデバイスアドレスまたは定数から[D]で指定したデバイスアドレスへ[i]で指定した演算単位のデータを論理反転して転送します。

[S] → [D]



## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[I]...US,SS

[S] ...DT2 [D] ...DT3

DT0	H 0011	DT0	H 0011
DT1	H 2233	DT1	H 2233
DT2	<b>H 4455</b>	DT2	H 4455
DT3	H 6677	DT3	<b>H BBAA</b>
DT4	H 8899	DT4	H 8899

例2) 演算単位が32ビット(UL,SL,SF)の場合

[I]...UL,SL,SF

[S] ...TS1 [D] ...TS4

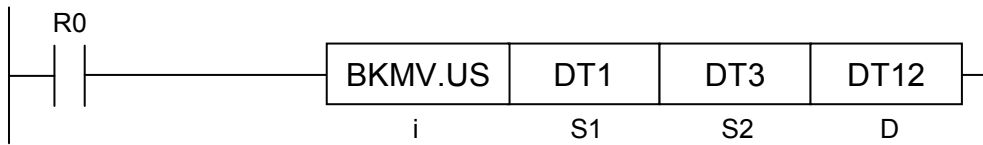
TS0	H 11223344	TS0	H 11223344
TS1	<b>H 55667788</b>	TS1	H 55667788
TS2	H 9900AABB	TS2	H 9900AABB
TS3	H CCDDEEFF	TS3	H CCDDEEFF
TS4	H 12345678	TS4	<b>H AA998877</b>

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# BKMV (ブロック転送)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	転送元データの先頭デバイスアドレス
S2	転送元データの終端デバイスアドレス
D	データの転送先の先頭デバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	""		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●								●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●								●
D	●	●	●	●			●	●	●		●	●	●	●								●

\*1: 演算単位が16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイスのみ修飾可能(整数定数、実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

• [S1]~[S2]で指定されたエリアのデータを[D]で指定されたエリア以降に一括で転送します。

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[S1]...DT1 [S2]...DT3 [D]...DT12

DT0	H 0011		DT10	H 0011
DT1	H 2233	↘	DT11	H 2233
DT2	H 4455	↘	DT12	H 2233
DT3	H 6677	↘	DT13	H 4455
DT4	H 8899	↘	DT14	H 6677

例2) 演算単位が32ビット(UL,SL,SF)の場合

[i]...UL,SL,SF

[S1]...TS1 [S2]...TS4 [D]...DT10

TS0	H 11223344		DT10・DT11	H 55667788
TS1	H 55667788	↘	DT12・DT13	H 9900AABB
TS2	H 9900AABB	↘	DT14・DT15	H CCDDEEFF
TS3	H CCDDEEFF	↘	DT16・DT17	H 12345678
TS4	H 12345678	↘	DT18・DT19	H 24680ACE

## ■ プログラム上のご注意

- 直接アドレスとインデックス修飾アドレスの場合、[S1]と[S2]は同じ種類のデバイスを指定してください。かつ、[S2] ≥ [S1]と指定してください。
- 演算単位毎に転送し[S2]が含まれるデバイスで終了します。

例1) [S2]のデバイスアドレスが上位ワードにきた場合(演算単位=32ビット)

[S1]...DT2 [S2]...DT6 [D]...DT12

DT0・DT1	H 11223344		DT10・DT11	H 11111111
DT2・DT3	<b>H 55667788</b>	→	DT12・DT13	<b>H 55667788</b>
DT4・DT5	<b>H 9900AABB</b>	→	DT14・DT15	<b>H 9900AABB</b>
DT6・DT7	<b>H CCDDEEFF</b>	→	DT16・DT17	<b>H CCDDEEFF</b>
DT8・DT9	H 12345678		DT18・DT19	H 55555555

例2) [S2]のデバイスアドレスが下位ワードに来た場合(演算単位=32ビット)

[S1]...DT2 [S2]...DT7 [D]...DT12

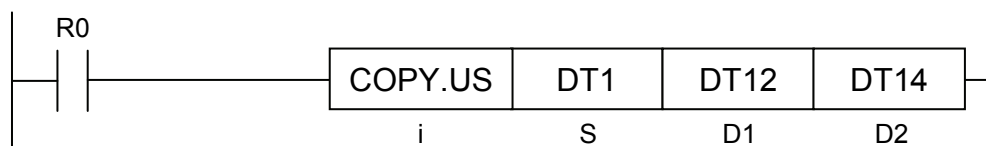
DT0・DT1	H 11223344		DT10・DT11	H 11111111
DT2・DT3	<b>H 55667788</b>	→	DT12・DT13	<b>H 55667788</b>
DT4・DT5	<b>H 9900AABB</b>	→	DT14・DT15	<b>H 9900AABB</b>
DT6・DT7	<b>H CCDDEEFF</b>	→	DT16・DT17	<b>H CCDDEEFF</b>
DT8・DT9	H 12345678		DT18・DT19	H 55555555

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S1] > [S2]の場合にセットします
(ER)	転送先範囲がアクセス可能範囲外の場合にセットします

# COPY (ブロック複写)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S	転送元データのデバイスアドレスまたは定数
D1	転送先エリアの先頭デバイスアドレス
D2	転送先エリアの終端デバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8		..
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D1	●	●	●	●			●	●	●		●	●	●	●							●
D2	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)。

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

\*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*8: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

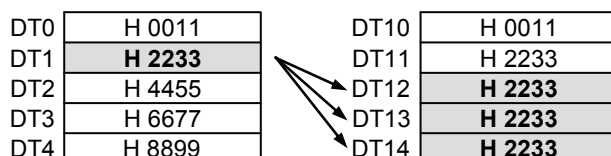
## ■ 動作説明

- [S]で指定されたデータを[D1]~[D2]のエリアに複写します。

## ■ 処理内容

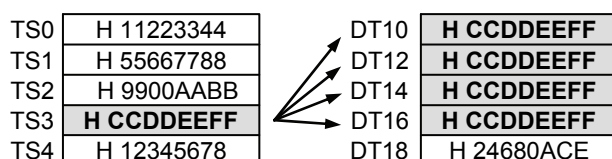
例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS  
[S]...DT1 [D1]...DT12 [D2]...DT14



例2) 演算単位が32ビット(UL,SL,SF)の場合

[i]...UL,SL,SF  
[S]...TS3 [D1]...DT10 [D2]...DT16

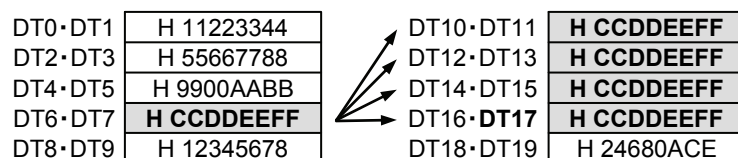


## ■ プログラム上のご注意

- 直接アドレスとインデックス修飾アドレスの場合、[D1]と[D2]は同じ種類のデバイスを指定してください。かつ、[D2]≧[D1]と指定してください。
- 演算単位毎に転送し[D2]が含まれるデバイスで終了します。

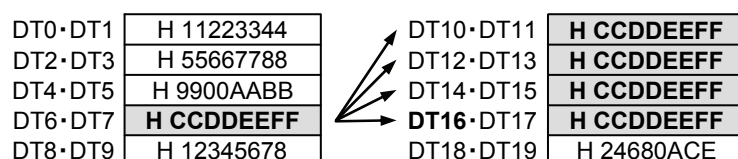
例1) [D2]のデバイスアドレスが上位ワードにきた場合(演算単位=32ビット)

[S]...DT6 [D1]...DT10 [D2]...DT17



例2) [D2]のデバイスアドレスが下位ワードに来た場合(演算単位=32ビット)

[S]...DT6 [D1]...DT10 [D2]...DT16

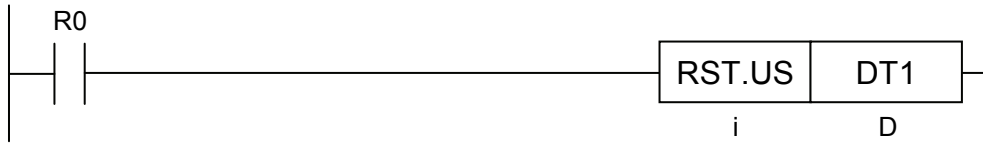


## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	[D1] > [D2]の場合にセットします

# RST (リセット)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
D	リセットする対象

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2			
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS	TE	IX	K	U	H	SF	DF	""				
						*4						CS	CE	*3										
D	●	●	●	●		●	●	●			●	●	●	●										●

\*1: 演算単位が 16 ビット整数(SS, US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイスのみ修飾可能(整数定数、実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: SD60/SD61 のみ許可

## ■ 動作説明

- 演算単位が 16 ビット(US,SS) または 32 ビット(UL,SL,SF)の場合、[D]で指定されたエリアをリセット(0 クリア)します。

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[D]...DT1

DT0	H 0011	→	DT0	H 0011
DT1	<b>H 2233</b>		DT1	<b>H 0000</b>
DT2	H 4455		DT2	H 4455
DT3	H 6677		DT3	H 6677
DT4	H 8899		DT4	H 8899

例2) 演算単位が32ビット(UL,SL,SF)の場合

[i]...UL,SL,SF

[D]...TS3

TS0	H 11223344	→	TS0	H 11223344
TS1	H 55667788		TS1	H 55667788
TS2	H 9900AABB		TS2	H 9900AABB
TS3	<b>H CCDDEEFF</b>		TS3	<b>H 00000000</b>
TS4	H 12345678		TS4	H 12345678

※SFの場合、32ビット全て0(符号と指数部と仮数部を0)で0.00e+00になります

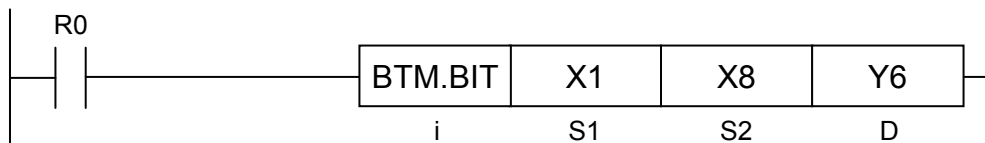


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# BTM (ビットブロック転送)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i	●						

## ■ オペランド一覧

オペランド	説明
S1	転送元データの先頭ビットアドレス
S2	転送元データの終端ビットアドレス
D	データの転送先の先頭ビットアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	ビットデバイス											ワードデバイスのビット指定		インデックス修飾	
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b	LD.b		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
D	●	●	●	●							●	●	●	●	●

## ■ 動作説明

- [S1]で指定したエリア(ビットアドレス)から[S2]で指定したエリア(ビットアドレス)までを[D]で指定されたエリアにビット転送します。

## ■ 処理内容

例1) X1~X8をY6~YDに転送

[S1]...X1 [S2]...X8 [D]...Y6

		WX0																
bit		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
BIN		0	1	1	0	1	0	0	1	1	1	0	0	0	0	0	1	1



		WY0																
bit		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
BIN		0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0

例2) X1~X8をYD~Y14に転送

[S1]...X1 [S2]...X8 [D]...YD

		WX0															
bit		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN		0	1	1	0	1	0	0	1	1	1	0	0	0	0	1	1



		WY1																WY0															
bit		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN		0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

## ■ プログラム上のご注意

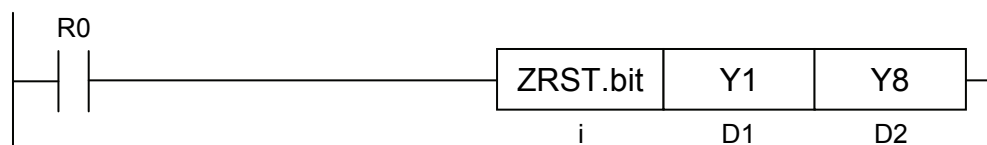
- 直接アドレスとインデックス修飾アドレスの場合、[S1]と[S2]は同じ種類のデバイスを指定してください。かつ、[S2] ≥ [S1]と指定してください。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S1] > [S2]の場合にセットします
(ER)	転送先範囲がアクセス可能範囲外の場合にセットします

# ZRST (ブロッククリア)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i	●						

## ■ オペランド一覧

オペランド	説明
D1	リセットデータの先頭ビットアドレス
D2	リセットデータの終端ビットアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	ビットデバイス											ワードデバイスのビット指定			インデックス修飾
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b	LD.b	SD.b	
D1	●	●	●	●							●	●	●		●
D2	●	●	●	●							●	●	●		●

## ■ 動作説明

- [D1]で指定したエリア(ビットアドレス)から[D2]で指定したエリア(ビットアドレス)までを 0 クリア(リセット)します。
- ステップラダーにおいて、工程[D1]から工程[D2]までの範囲の起動中の工程をまとめてクリアすることにも使用できます。

## ■ 処理内容

例1) Y1～Y8をリセット

[D1]...Y1 [D2]...Y8

		WY0															
bit		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1



		WY0															
bit		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN		1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1

例2) YD～Y14をリセット

[D1]...YD [D2]...Y14

		WY1																WY0															
bit		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	



		WY1																WY0															
bit		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN		1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1		

## ■ プログラム上のご注意

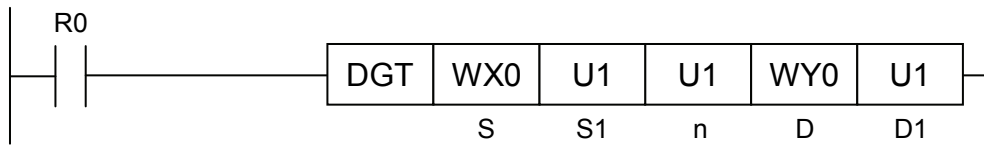
- 直接アドレスとインデックス修飾アドレスの場合、[D1]と[D2]は同じ種類のデバイスを指定してください。かつ、[D2] ≧ [D1]と指定してください。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	[D1] > [D2]の場合にセットします

# DGT (デジタルデータ転送)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S	転送元データのデバイスアドレスまたは定数
S1	転送元の転送開始桁(データ指定可能範囲:0~3)
n	転送する桁数(データ指定可能範囲:1~4)
D	転送先データのデバイスアドレス
D1	転送先の転送開始桁(データ指定可能範囲:0~3)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""	
S	●	●	●	●	●	●	●	●	●	●	●				●	●	●				●
S1(*1)	●	●	●	●			●	●	●	●	●					●	●				●
n(*1)	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●
D1(*1)	●	●	●	●			●	●	●	●	●					●	●				●

\*1: 演算単位にかかわらず 16ビット符号なし整数(US)として扱います

\*2: 16ビットデバイス、整数定数のみ修飾可能(32ビットデバイス、実数定数、文字定数には指定不可)

## ■ 動作説明

- [S]で指定したエリアの[S1]桁目から[n]桁分、[D]で指定した 16 ビットデータの[D1]桁に転送します。
- 転送開始桁は下位から 4bit ずつ第 0 桁目、第 1 桁目、第 2 桁目、第 3 桁目となります。

bit	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
	第3桁目				第2桁目				第1桁目				第0桁目			

## ■ 処理内容

例1) 第1デジットから第1デジットへ転送する場合

[S]...WX0 [S1]...U1(H1)  
 [n]...U1(H1)  
 [D]...WY0 [D1]...U1(H1)

		X																
bit		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
BIN		0	1	1	0	1	0	0	1	1	1	0	0	0	0	0	1	1

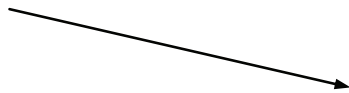


		Y																
bit		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
BIN		0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

例2) 1デジットずらして転送する場合

[S]...WX0 [S1]...U3(H3)  
 [n]...U1(H1)  
 [D]...WY0 [D1]...U0(H0)

		X															
bit		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN		0	1	1	0	1	0	0	1	1	1	0	0	0	0	1	1



		Y															
bit		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN		0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

例3) 複数デジットを平行に転送する場合

[S]...WX0 [S1]...U2(H2)  
 [n]...U2(H2)  
 [D]...WY0 [D1]...U2(H2)

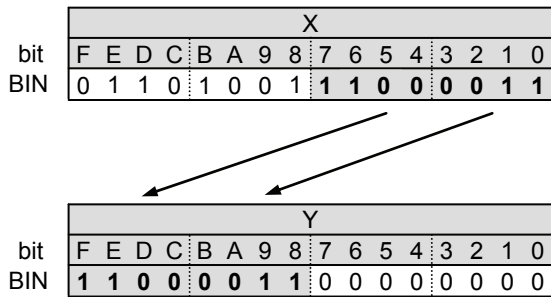
		X															
bit		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN		0	1	1	0	1	0	0	1	1	1	0	0	0	0	1	1



		Y															
bit		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
BIN		0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0

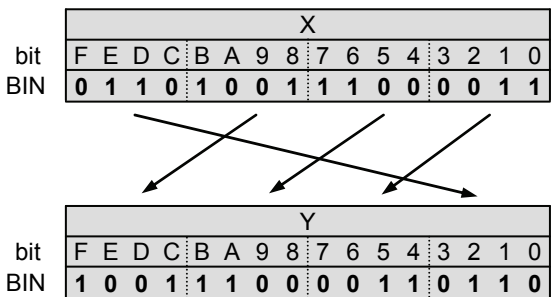
例4) 複数デジットをずらして転送する場合

[S]...WX0 [S1]...U0(H0)  
 [n]...U2(H2)  
 [D]...WY0 [D1]...U2(H2)



例5) 4デジット分を転送する場合

[S]...WX0 [S1]...U0(H0)  
 [n]...U4(H4)  
 [D]...WY0 [D1]...U1(H1)



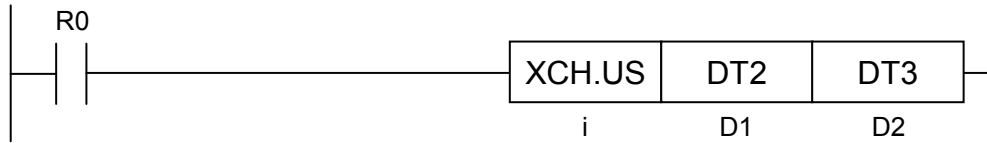
■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	[S1]、[n]、[D1]のオペランドが指定範囲外の場合にセットします



# XCH (データ交換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
D1	交換データ1のデバイスアドレス
D2	交換データ2のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	" "		
S1	●	●	●	●			●	●	●		●	●	●	●								●
S2	●	●	●	●			●	●	●		●	●	●	●								●

\*1: 演算単位が16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイスのみ修飾可能(整数定数のみ、実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [D1]で指定したデバイスアドレスと[D2]で指定したデバイスアドレスを[i]で指定した演算単位に従ってデータ交換します。

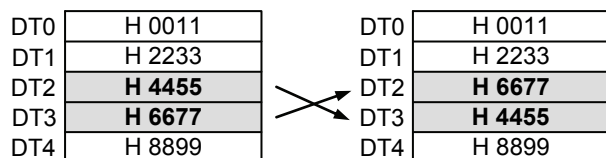
[D1] ⇔ [D2]

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

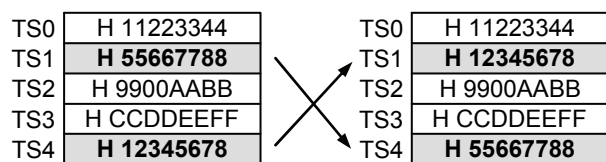
[D1]...DT2 [D2]...DT3



例2) 演算単位が32ビット(UL,SL,SF)の場合

[i]...UL,SL,SF

[D1]...TS1 [D2]...TS4



## ■ プログラム上のご注意

- 交換データの範囲が重複しないように指定してください。

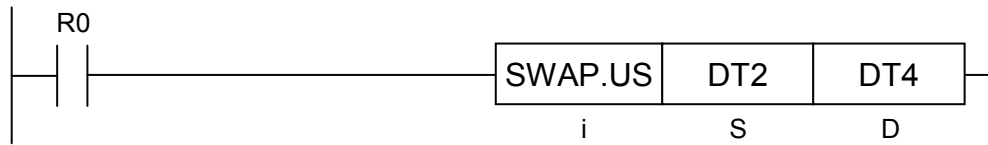
## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

•

# SWAP (上位バイト下位バイト交換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
S	上位バイトと下位バイトを交換する転送元のデバイスアドレス
D	交換データの転送先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデックス 修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "		
S	●	●	●	●	●	●	●	●	●	●	●											●
D	●	●	●	●			●	●	●		●											●

\*1: 16ビットデバイスのみ修飾可能(32ビットデバイス、整数定数、実数定数、文字定数には指定不可)

## ■ 動作説明

- [S]で指定したデバイスアドレスの上位バイトと下位バイトを交換して[D]で指定したデバイスアドレスに転送します。

## ■ 処理内容

例) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[S]...DT2 [D]...DT4

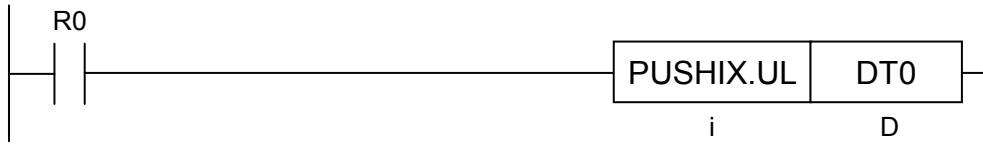
DT0	H 0011	DT0	H 0011
DT1	H 2233	DT1	H 2233
DT2	<b>H 4455</b>	DT2	H 4455
DT3	H 6677	DT3	H 6677
DT4	H 8899	DT4	<b>H 5544</b>

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# PUSHIX (インデックスレジスタ退避)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i				●	●		

## ■ オペランド一覧

オペランド	説明
D	退避先のデバイスアドレス先頭

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF	DF	""		
D	●	●	●	●			●	●														●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(整数定数、実数定数、文字定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

## ■ 動作説明

- インデックスレジスタの値を、[D]を先頭に 15 データ(30 ワード)分退避します。  
(I0 ~ IE) → ([D] ~ [D] + 29)
- 退避元のインデックスレジスタの値は変化しません。
- 「PUSHIX」命令は、サブルーチンなどの副プログラムでインデックスレジスタを使用する場合に、主プログラムから副プログラムに移行する前に、インデックスレジスタの内容を退避するために使用します。
- 「POPIX」(インデックスレジスタ復帰)命令と組み合わせて使用してください。

## ■ 処理内容

例) 第1オペランド[D]にDT0を指定

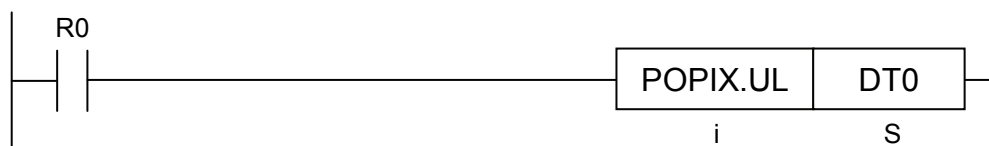
		退避前	退避後
I0	H 00112233	DT0 H 00000000	H 00112233
I1	H 44556677	DT2 H 00000000	H 44556677
I2	H 8899AABB	DT4 H 00000000	H 8899AABB
⋮	⋮	⋮	⋮
IC	H CCDDEEFF	DT24 H 00000000	H CCDDEEFF
ID	H 12345678	DT26 H 00000000	H 12345678
IE	H 90ABCDEF	DT28 H 00000000	H 90ABCDEF

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	退避先範囲がアクセス可能範囲外の場合にセットします

# POPIX (インデックスレジスタ復帰)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i				●	●		

## ■ オペランド一覧

オペランド	説明
S	復帰元のデバイスアドレス先頭

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF	DF	" "		
S	●	●	●	●			●	●														●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(整数定数、実数定数、文字定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [S]を先頭に 15 データ(30 ワード)分インデックスレジスタの値に復帰します。
- ([S] ~ [S] + 29) → (I0 ~ IE)
- 「POPIX」命令は、サブルーチンなどの副プログラムでインデックスレジスタを使用する場合に、副プログラムから主プログラムに移行する前に、「PUSHIX」(インデックスレジスタ退避)命令で、退避した内容を復帰させるために使用します。
- 「PUSHIX」(インデックスレジスタ退避)命令と組み合わせて使用してください。

## ■ 処理内容

例) 第1オペランド[S]にDT0を指定

		復帰前	復帰後
DT0	H 00112233	I0 H 00000000	H 00112233
DT2	H 44556677	I1 H 00000000	H 44556677
DT4	H 8899AABB	I2 H 00000000	H 8899AABB
⋮	⋮	⋮	⋮
DT24	H CCDDEEFF	IC H 00000000	H CCDDEEFF
DT26	H 12345678	ID H 00000000	H 12345678
DT28	H 90ABCDEF	IE H 00000000	H 90ABCDEF

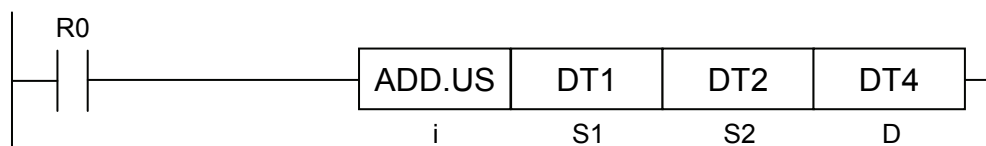
→ 復帰

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	復帰元範囲がアクセス可能範囲外の場合にセットします

# ADD (加算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	演算対象データ 1(デバイスアドレスまたは定数)
S2	演算対象データ 2(デバイスアドレスまたは定数)
D	演算結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデックス 修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8	..	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

\*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*8: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って[S1]と[S2]の値を加算します。
- 演算結果は、[D]を先頭とするアドレスに格納します。

[S1] + [S2] → [D]

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[I]...US,SS

[S1]...DT1 [S2]...DT2 [D]...DT4

DT0	K100		DT0	K100
DT1	<b>K110</b>	}	DT1	K110
DT2	<b>K120</b>		DT2	K120
DT3	K130		DT3	K130
DT4	K140	→	DT4	<b>K230</b>

例2) 演算単位が32ビット(UL,SL,SF)の場合

[I]...UL,SL,SF

[S1]...TS1 [S2]...TS3 [D]...TS0

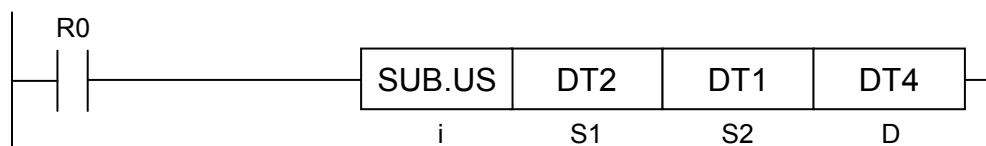
TS0	K500	}	TS0	<b>K3000</b>
TS1	<b>K1000</b>		TS1	K1000
TS2	K1500		TS2	K1500
TS3	<b>K2000</b>		TS3	K2000
TS4	K2500		TS4	K2500

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	
	演算単位が実数(SF)の場合、[S1] または [S2]に非実数を指定した時にセットします

# SUB (減算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	演算対象データ 1(デバイスアドレスまたは定数)
S2	演算対象データ 2(デバイスアドレスまたは定数)
D	演算結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8	..	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

\*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*8: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可

## ■ 動作説明

- [i]の演算単位に従って[S1]から[S2]の値を減算します。
- 演算結果は、[D]を先頭とするアドレスに格納します。

[S1] - [S2] → [D]



## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[I]...US,SS

[S1]...DT2 [S2]...DT1 [D]...DT4

DT0	K 100		DT0	K 100
DT1	<b>K 110</b>	}	DT1	K 110
DT2	<b>K 120</b>		DT2	K 120
DT3	K 130		DT3	K 130
DT4	K 140	→	DT4	<b>K 10</b>

例2) 演算単位が32ビット(UL,SL,SF)の場合

[I]...UL,SL,SF

[S1]...TS3 [S2]...TS1 [D]...TS0

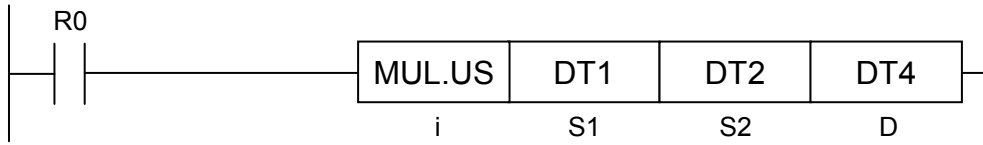
TS0	K 500	}	TS0	<b>K 1000</b>
TS1	<b>K 1000</b>		TS1	K 1000
TS2	K 1500		TS2	K 1500
TS3	<b>K 2000</b>		TS3	K 2000
TS4	K 2500		TS4	K 2500

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	
	演算単位が実数(SF)の場合、[S1] または [S2]に非実数を指定した時にセットします

# MUL (乗算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	演算対象データ 1(デバイスアドレスまたは定数)
S2	演算対象データ 2(デバイスアドレスまたは定数)
D	演算結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8	..	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

\*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*8: 演算単位が単精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

● [i]の演算単位に従って[S1]と[S2]の値を乗算します。

● 演算結果は、[D]を先頭とするアドレスに格納します。

$$[S1] \times [S2] \rightarrow ([D], [D]+1)$$

● [D]のサイズは整数の場合は演算単位の 2 倍とします。

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[S1]...DT1 [S2]...DT2 [D]...DT3

DT0	K 100		DT0	K 100
DT1	<b>K 110</b>	}	DT1	K 110
DT2	<b>K 120</b>		DT2	K 120
DT3	K 130	}	DT3	<b>K 13200</b>
DT4	K 140		DT4	

例2) 演算単位が32ビット(UL,SL,SF)の場合

[i]...UL,SL,SF

[S1]...TS2 [S2]...TS3 [D]...TS0

TS0	K 500		TS0	<b>K 3000000</b>
TS1	K 1000	}	TS1	
TS2	<b>K 1500</b>		TS2	K 1500
TS3	<b>K 2000</b>	}	TS3	K 2000
TS4	K 2500		TS4	K 2500

## ■ プログラム上のご注意

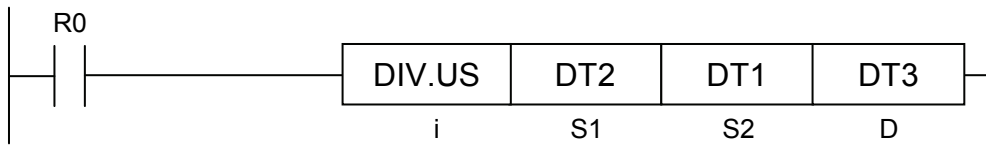
- 演算結果が格納されるエリアは、演算単位の2倍となります。他のエリアを上書きしないように、メモリエリアを割り付けてください。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	演算単位が実数(SF)の場合、[S1] または [S2]に非実数を指定した時にセットします

# DIV (除算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	演算対象データ 1(デバイスアドレスまたは定数)
S2	演算対象データ 2(デバイスアドレスまたは定数)
D	演算結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8		..
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

\*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*8: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って[S1]を[S2]の値で除算します。
- 演算結果は、[D]を先頭とするアドレスに格納します。

$$[S1] \div [S2] \rightarrow \text{商}([D])$$

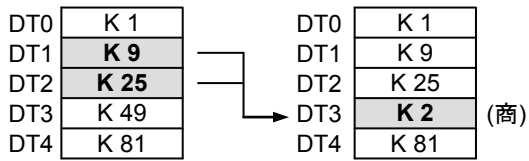
- 余りは出力しません。余りが必要な場合は、DIVMOD 命令を使用してください。

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

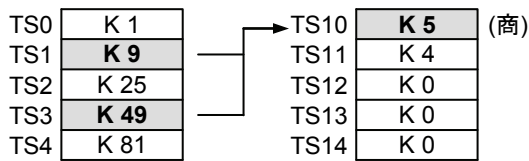
[S1]...DT2 [S2]...DT1 [D]...DT3



例2) 演算単位が32ビット(UL,SL,SF)の場合

[i]...UL,SL,SF

[S1]...TS3 [S2]...TS1 [D]...TS10



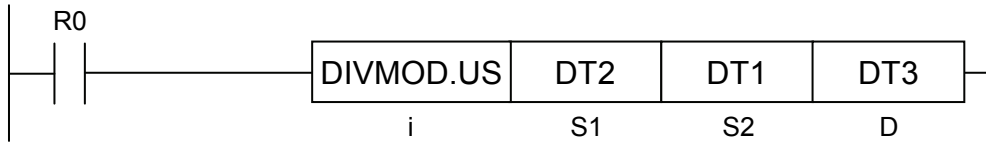
※実数の場合、余りはでないため設定しません

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	演算単位が実数(SF)の場合、[S1] または [S2]に非実数を指定した時にセットします
(ER)	[S2]に 0 を指定した場合にセットします

# DIVMOD (除算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S1	演算対象データ 1(デバイスアドレスまたは定数)
S2	演算対象データ 2(デバイスアドレスまたは定数)
D	演算結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF	DF	..	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

## ■ 動作説明

● [i]の演算単位に従って[S1]を[S2]の値で除算します。

● 演算結果は、[D]を先頭とするアドレスに格納します。

[S1] ÷ [S2] → 商([D]), 余り([D]+1)

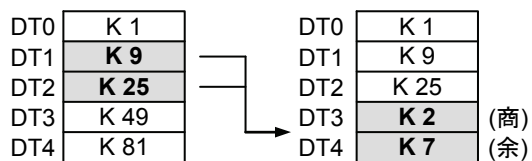
● 整数の場合、余りは[D+1]に設定します。

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[I]...US,SS

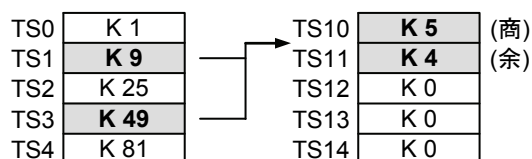
[S1]...DT2 [S2]...DT1 [D]...DT3



例2) 演算単位が32ビット(UL,SL,SF)の場合

[I]...UL,SL,SF

[S1]...TS3 [S2]...TS1 [D]...TS10



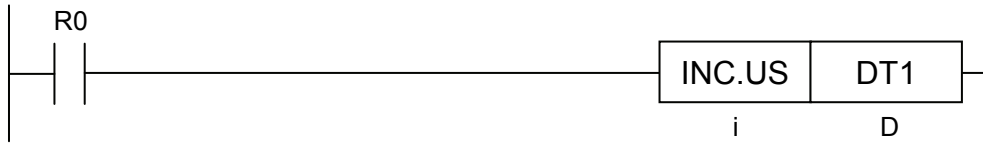
※実数の場合、余りはでないため設定しません

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	演算単位が実数(SF)の場合、[S1] または [S2]に非実数を指定した時にセットします
(ER)	[S2]に 0 を指定した場合にセットします

# INC (インクリメント)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
D	演算結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス*1			整数			実数		文字	インデックス修飾*2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF		""
D	●	●	●	●			●	●	●		●	●	●								●

\*1: 演算単位が16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイスのみ修飾可能(整数定数、実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [i]の演算単位に従って、[D]の値に1を加算します。
- 演算結果は、[D]を先頭とするアドレスに格納します。

$$[D] + 1 \rightarrow [D]$$

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[D]...DT1

DT0	K 100	→	DT0	K 100
DT1	<b>K 110</b>		DT1	<b>K 111</b>
DT2	K 120		DT2	K 120
DT3	K 130		DT3	K 130

例2) 演算単位が32ビット(UL,SL,SF)の場合

[i]...UL,SL,SF

[D]...TS1

TS0	K 500	→	TS0	K 500
TS1	<b>K 1000</b>		TS1	<b>K 1001</b>
TS2	K 1500		TS2	K 1500
TS3	K 2000		TS3	K 2000

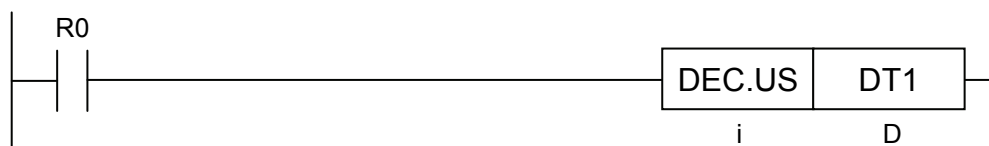
## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	演算単位が実数(SF)の場合、[D]に非実数を指定した時にセットします



# DEC (デクリメント)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
D	演算対象データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデック ス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		" "
D	●	●	●	●			●	●	●		●	●	●								●

\*1: 演算単位が16ビット整数(SS, US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイスのみ修飾可能(整数定数、実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [i]の演算単位に従って、[D]の値から1を減算します。
- 演算結果は、[D]を先頭とするアドレスに格納します。

[D] - 1 → [D]

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[D]...DT1

DT0	K 100	→	DT0	K 100
DT1	<b>K 110</b>		DT1	<b>K 109</b>
DT2	K 120		DT2	K 120
DT3	K 130		DT3	K 130

例2) 演算単位が32ビット(UL,SL,SF)の場合

[i]...UL,SL,SF

[D]...TS1

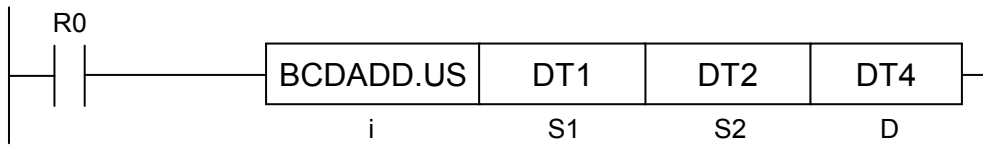
TS0	K 500	→	TS0	K 500
TS1	<b>K 1000</b>		TS1	<b>K 999</b>
TS2	K 1500		TS2	K 1500
TS3	K 2000		TS3	K 2000

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	演算単位が実数(SF)の場合、[D]に非実数を指定した時にセットします

# BCDADD (BCDデータ加算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
S1	演算対象データ 1(デバイスアドレスまたは定数)(指定可能データは各桁 H0~H9)
S2	演算対象データ 2(デバイスアドレスまたは定数)(指定可能データは各桁 H0~H9)
D	演算結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	""	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイスのみ修飾可能(整数定数、実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [i]の演算単位に従って[S1]と[S2]の BCD データを加算します。
- 演算結果は、[D]を先頭とするアドレスに格納します。

$$[S1] + [S2] \rightarrow [D]$$

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[i]...US

[S1]...DT1 [S2]...DT2 [D]...DT4

DT0	H 0500	}	DT0	H 0500
DT1	<b>H 0510</b>		DT1	H 0510
DT2	<b>H 0520</b>		DT2	H 0520
DT3	H 0530		DT3	H 0530
DT4	H 0540		DT4	<b>H 1030</b>

例2) 演算単位が32ビット(UL)の場合

[i]...UL

[D1]...TS2 [D2]...TS3 [D]...TS0

TS0	H 00005000	}	TS0	<b>H 00012500</b>
TS1	H 00005500		TS1	H 00005500
TS2	<b>H 00006000</b>		TS2	H 00006000
TS3	<b>H 00006500</b>		TS3	H 00006500
TS4	H 00007000		TS4	H 00007000

例3) 演算単位が16ビット(US)の場合

[i]...US

[S1]...DT1 [S2]...DT4 [D]...DT3

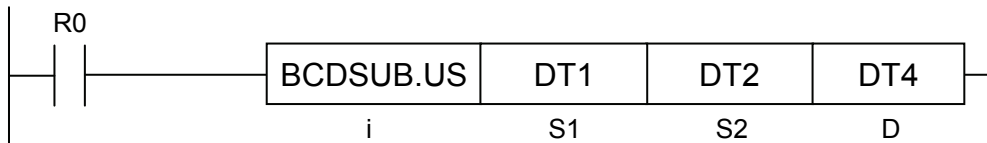
DT0	H 5000	}	DT0	H 5000
DT1	<b>H 6000</b>		DT1	H 6000
DT2	H 7000		DT2	H 7000
DT3	H 8000		DT3	<b>H 5000</b>
DT4	<b>H 9000</b>		DT4	H 9000

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	

# BCDSUB (BCDデータ減算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
S1	演算対象データ 1(デバイスアドレスまたは定数)(指定可能データは各桁 H0~H9)
S2	演算対象データ 2(デバイスアドレスまたは定数)(指定可能データは各桁 H0~H9)
D	演算結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	""	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイスのみ修飾可能(整数定数、実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [i]の演算単位に従って[S1]から[S2]の BCD データを減算します。
- 演算結果は、[D]を先頭とするアドレスに格納します。

[S1] - [S2] → [D]

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[i]...US

[S1]...DT1 [S2]...DT2 [D]...DT4

DT0	H 0540	}	DT0	H 0540
DT1	<b>H 0530</b>		DT1	H 0530
DT2	<b>H 0520</b>		DT2	H 0520
DT3	H 0510		DT3	H 0510
DT4	H 0500		DT4	<b>H 0010</b>

例2) 演算単位が32ビット(UL)の場合

[i]...UL

[D1]...TS2 [D2]...TS3 [D]...TS0

TS0	H 00020000	}	TS0	<b>H 00005000</b>
TS1	H 00019500		TS1	H 00019500
TS2	<b>H 00019000</b>		TS2	H 00019000
TS3	<b>H 00018500</b>		TS3	H 00018500
TS4	H 00018000		TS4	H 00018000

例3) 演算単位が16ビット(US)の場合

[i]...US

[S1]...DT1 [S2]...DT4 [D]...DT3

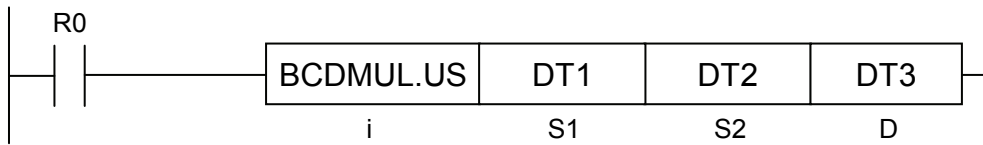
DT0	H 1000	}	DT0	H 1000
DT1	<b>H 2000</b>		DT1	H 2000
DT2	H 3000		DT2	H 3000
DT3	H 4000		DT3	<b>H 7000</b>
DT4	<b>H 5000</b>		DT4	H 5000

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	演算対象データ[S1]、[S2]に BCD 以外のデータを指定した場合、セットします

# BCDMUL (BCDデータ乗算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
S1	演算対象データ 1(デバイスアドレスまたは定数) (指定可能データは各桁 H0~H9)
S2	演算対象データ 2(デバイスアドレスまたは定数) (指定可能データは各桁 H0~H9)
D	演算結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	""	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイスのみ修飾可能(整数定数、実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [i]の演算単位に従って[S1]と[S2]の BCD データを乗算します。
- 演算結果は、[D]を先頭とするアドレスに格納します。  
[S1] × [S2] → ([D]、[D]+1)
- [D]の演算結果の出力は演算単位の 2 倍になります。

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[i]...US

[S1]...DT1 [S2]...DT2 [D]...DT3

DT0	H 0500	}	DT0	H 0500
DT1	<b>H 0510</b>		DT1	H 0510
DT2	<b>H 0520</b>		DT2	H 0520
DT3	H 0530		DT3	<b>H 00265200</b>
DT4	H 0540		DT4	

例2) 演算単位が32ビット(UL)の場合

[i]...UL

[S1]...TS2 [S2]...TS3 [D]...TS0

TS0	H 00005000	}	TS0	<b>H 0000000039000000</b>
TS1	H 00005500		TS1	
TS2	<b>H 00006000</b>		TS2	H 00006000
TS3	<b>H 00006500</b>		TS3	H 00006500
TS4	H 00007000		TS4	H 00007000

## ■ プログラム上のご注意

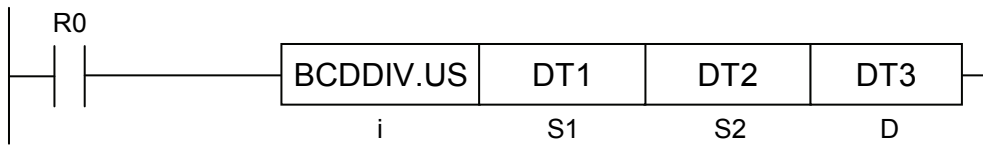
- [D]に演算デバイスの終端付近を指定すると次のデバイスのメモリを演算単位の2倍破壊することがあります。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	
	演算対象データ[S1]、[S2]にBCD以外のデータを指定した場合、セットします

# BCDDIV (BCDデータ除算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
S1	演算対象データ 1(デバイスアドレスまたは定数)(指定可能データは各桁 H0~H9)
S2	演算対象データ 2(デバイスアドレスまたは定数)(指定可能データは各桁 H0~H9)
D	演算結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	""		
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●					●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●			●					●
D	●	●	●	●			●	●			●	●	●	●								●

\*1: 演算単位が 16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイスのみ修飾可能(整数定数、実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [i]の演算単位に従って[S1]を[S2]の BCD データで除算します。
- 演算結果は、[D]を先頭とするアドレスに格納します。  
[S1] ÷ [S2] → 商([D])、余り([D]+1)
- 余りは([D]+1)に格納します。



## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[I]...US

[S1]...DT1 [S2]...DT2 [D]...DT3

DT0	H 0064	}	DT0	H 0064
DT1	<b>H 0049</b>		DT1	H 0049
DT2	<b>H 0025</b>		DT2	H 0025
DT3	H 0009		DT3	<b>H 0001</b>
DT4	H 0001		DT4	<b>H 0024</b>

例2) 演算単位が32ビット(UL)の場合

[I]...UL

[S1]...TS2 [S2]...TS3 [D]...TS0

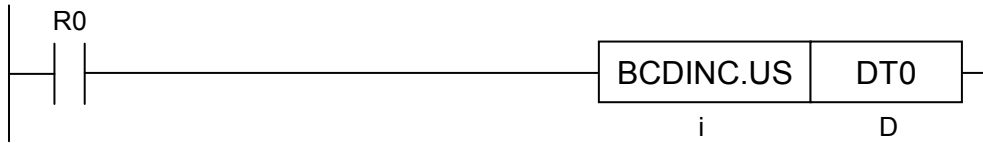
TS0	H 11111100	}	TS0	<b>H 00033333</b>
TS1	H 22222200		TS1	<b>H 00000300</b>
TS2	<b>H 33333300</b>		TS2	H 33333300
TS3	<b>H 00001000</b>		TS3	H 00001000
TS4	H 44444400		TS4	H 44444400

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S2]に 0 を指定した場合にセットします
(ER)	演算対象データ[S1]、[S2]に BCD 以外のデータを指定した場合、セットします

# BCDINC (BCDデータインクリメント)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
D	演算対象データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス*1			整数			実数		文字	インデックス修飾*2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	""	
D	●	●	●	●			●	●	●		●	●	●								●

\*1: 演算単位が16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイスのみ修飾可能(整数定数、実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [i]の演算単位に従って、[D]のBCDデータに1を加算します。
- 演算結果は、[D]を先頭とするアドレスに格納します。

$$[D] + 1 \rightarrow [D]$$

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[i]...US  
[D]...DT0

DT0	H 0100	→	DT0	H 0101
DT1	H 0200		DT1	H 0200
DT2	H 0300		DT2	H 0300
DT3	H 0400		DT3	H 0400

例2) 演算単位が32ビット(UL)の場合

[i]...UL  
[D]...TS1

TS0	H 01000000	→	TS0	H 01000000
TS1	<b>H 01999999</b>		TS1	<b>H 02000000</b>
TS2	H 03000000		TS2	H 03000000
TS3	H 04000000		TS3	H 04000000

例3) 演算単位が16ビット(US)の場合

[i]...US  
[D]...DT0

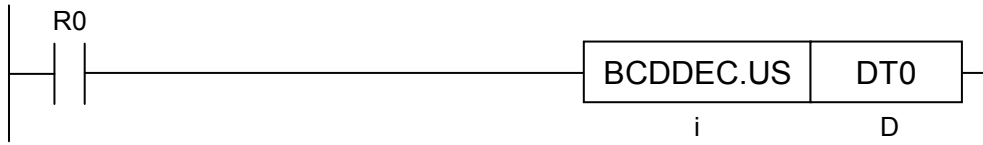
DT0	<b>H 9999</b>	→	DT0	<b>H 0000</b>
DT1	H 0200		DT1	H 0200
DT2	H 0300		DT2	H 0300
DT3	H 0400		DT3	H 0400

### ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	演算対象データ[D]に BCD 以外のデータを指定した場合、セットします

# BCDDEC (BCDデータデクリメント)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
D	演算対象データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス*1			整数			実数		文字	インデックス修飾*2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	""	
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイスのみ修飾可能(整数定数、実数定数、文字定数には指定不可)

\*3:インデックスレジスタ(I0~IE)

## ■ 動作説明

- [i]の演算単位に従って、[D]のBCDデータから1を減算します。
- 演算結果は、[D]を先頭とするアドレスに格納します。

[D] - 1 → [D]

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[i]...US

[D]...DT0

DT0	H 0100	→	DT0	H 0099
DT1	H 0200		DT1	H 0200
DT2	H 0300		DT2	H 0300
DT3	H 0400		DT3	H 0400

例2) 演算単位が32ビット(UL)の場合

[i]...UL  
[D]...TS1

TS0	H 01000000	→	TS0	H 01000000
TS1	<b>H 02000000</b>		TS1	<b>H 01999999</b>
TS2	H 03000000		TS2	H 03000000
TS3	H 04000000		TS3	H 04000000

例3) 演算単位が16ビット(US)の場合

[i]...US  
[D]...DT0

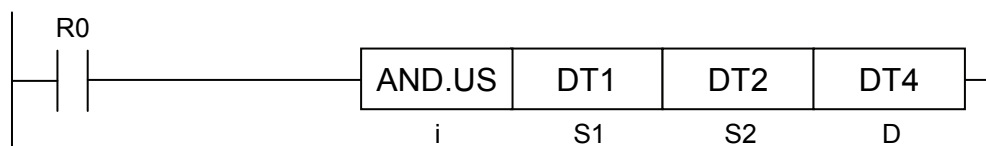
DT0	<b>H 0000</b>	→	DT0	<b>H 9999</b>
DT1	H 0200		DT1	H 0200
DT2	H 0300		DT2	H 0300
DT3	H 0400		DT3	H 0400

### ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	演算対象データ[D]に BCD 以外のデータを指定した場合、セットします

# AND (論理積)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S1	演算対象データ 1(デバイスアドレスまたは定数)
S2	演算対象データ 2(デバイスアドレスまたは定数)
D	演算結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデックス 修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0～IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って[S1]と[S2]の論理積を求めます。
- 演算結果は、[D]を先頭とするアドレスに格納します。

[S1] ∧ [S2] → [D]

## ■ 処理内容

### ■ 論理積(AND)

[S1]のビット	[S2]のビット	論理積
0	0	0
0	1	0
1	0	0
1	1	1

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[S1]...DT1 [S2]...DT2 [D]...DT4

DT0	H 1234	DT0	H 1234
DT1	<b>H FF00</b>	DT1	H FF00
DT2	<b>H 5678</b>	DT2	H 5678
DT3	H 00FF	DT3	H 00FF
DT4	H 90AB	DT4	<b>H 5600</b>

例2) 演算単位が32ビット(UL,SL)の場合

[i]...UL,SL

[S1]...TS1 [S2]...TS3 [D]...TS0

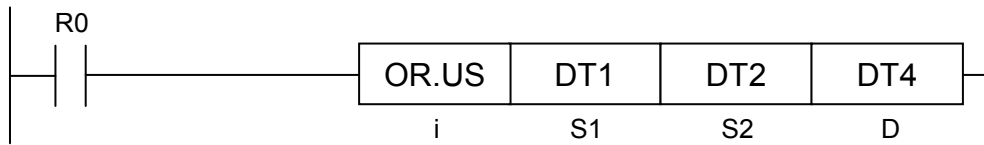
TS0	H 12345678	TS0	<b>H A0A0A0A0</b>
TS1	<b>H F0F0F0F0</b>	TS1	H F0F0F0F0
TS2	H 0F0F0F0F	TS2	H 0F0F0F0F
TS3	<b>H AAAAAAAAAA</b>	TS3	H AAAAAAAAAA
TS4	H 5AA5A55A	TS4	H 5AA5A55A

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# OR (論理和)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S1	演算対象データ 1(デバイスアドレスまたは定数)
S2	演算対象データ 2(デバイスアドレスまたは定数)
D	演算結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って[S1]と[S2]の論理和を求めます。
- 演算結果は、[D]を先頭とするアドレスに格納します。

[S1] ∨ [S2] → [D]



## ■ 処理内容

### ■ 論理和(OR)

[S1]のビット	[S2]のビット	論理和
0	0	0
0	1	1
1	0	1
1	1	1

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[S1]...DT1 [S2]...DT2 [D]...DT4

DT0	H 1234		DT0	H 1234
DT1	<b>H FF00</b>	}	DT1	H FF00
DT2	<b>H 5678</b>		DT2	H 5678
DT3	H 00FF		DT3	H 00FF
DT4	H 90AB		DT4	<b>H FF78</b>

例2) 演算単位が32ビット(UL,SL)の場合

[i]...UL,SL

[S1]...TS1 [S2]...TS3 [D]...TS0

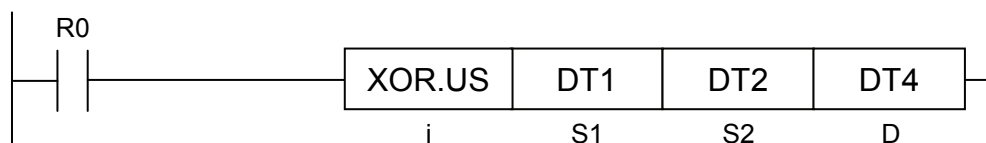
TS0	H 12345678		TS0	<b>H FAFAFafa</b>
TS1	<b>H F0F0F0F0</b>	}	TS1	H F0F0F0F0
TS2	H 0F0F0F0F		TS2	H 0F0F0F0F
TS3	<b>H AAAAAAAAAA</b>		TS3	H AAAAAAAAAA
TS4	H 5AA5A55A		TS4	H 5AA5A55A

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# XOR (排他的論理和)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S1	演算対象データ 1(デバイスアドレスまたは定数)
S2	演算対象データ 2(デバイスアドレスまたは定数)
D	演算結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って[S1]と[S2]の排他的論理和を求めます。
- 演算結果は、[D]を先頭とするアドレスに格納します。

$$\{ [S1] \wedge \neg [S2] \} \vee \{ \neg [S1] \wedge [S2] \} \rightarrow [D]$$

## ■ 処理内容

### ■ 排他的論理和(XOR)

[S1]のビット	[S2]のビット	排他的論理和
0	0	0
0	1	1
1	0	1
1	1	0

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[S1]...DT1 [S2]...DT2 [D]...DT4

DT0	H 1234	}	DT0	H 1234
DT1	<b>H FF00</b>		DT1	H FF00
DT2	<b>H 5678</b>		DT2	H 5678
DT3	H 00FF		DT3	H 00FF
DT4	H 90AB		DT4	<b>H A978</b>

例2) 演算単位が32ビット(UL,SL)の場合

[i]...UL,SL

[S1]...TS1 [S2]...TS3 [D]...TS0

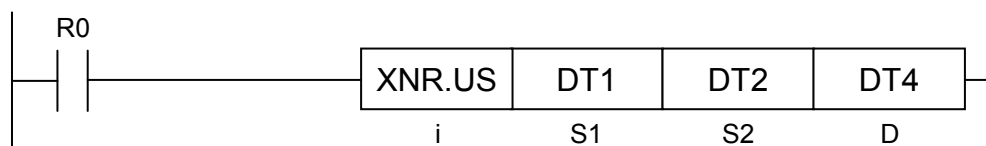
TS0	H 12345678	}	TS0	<b>H 5A5A5A5A</b>
TS1	<b>H F0F0F0F0</b>		TS1	H F0F0F0F0
TS2	H 0F0F0F0F		TS2	H 0F0F0F0F
TS3	<b>H AAAAAAAAAA</b>		TS3	H AAAAAAAAAA
TS4	H 5AA5A55A		TS4	H 5AA5A55A

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# XNR (排他的論理和否定)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S1	演算対象データ 1(デバイスアドレスまたは定数)
S2	演算対象データ 2(デバイスアドレスまたは定数)
D	演算結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF	DF	" "	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
S2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可

## ■ 動作説明

- [i]の演算単位に従って[S1]と[S2]の排他的論理和否定を求めます。
- 演算結果は、[D]を先頭とするアドレスに格納します。
- $\{ [S1] \wedge [S2] \} \vee \{ \neg [S1] \wedge \neg [S2] \} \rightarrow [D]$

## ■ 処理内容

### ■ 排他的論理和否定(XNR)

[S1]のビット	[S2]のビット	排他的論理和否定
0	0	1
0	1	0
1	0	0
1	1	1

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[S1]...DT1 [S2]...DT2 [D]...DT4

DT0	H 1234	DT0	H 1234
DT1	<b>H FF00</b>	DT1	H FF00
DT2	<b>H 5678</b>	DT2	H 5678
DT3	H 00FF	DT3	H 00FF
DT4	H 90AB	DT4	<b>H 5687</b>

例2) 演算単位が32ビット(UL,SL)の場合

[i]...UL,SL

[S1]...TS1 [S2]...TS3 [D]...TS0

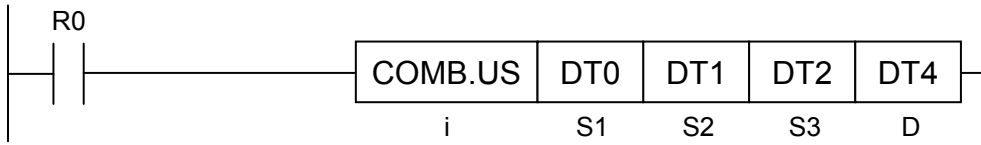
TS0	H 12345678	TS0	<b>H A5A5A5A5</b>
TS1	<b>H F0F0F0F0</b>	TS1	H F0F0F0F0
TS2	H 0F0F0F0F	TS2	H 0F0F0F0F
TS3	<b>H AAAAAAAAAA</b>	TS3	H AAAAAAAAAA
TS4	H 5AA5A55A	TS4	H 5AA5A55A

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# COMB (結合)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S1	結合データ 1(デバイスアドレスまたは定数)
S2	結合データ 2(デバイスアドレスまたは定数)
S3	結合用マスクデータ(デバイスアドレスまたは定数)
D	結合結果データ(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデックス 修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF	DF	" "	
S1	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●				●
S2	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●				●
S3	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ (I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可

## ■ 動作説明

- [i]の演算単位と[S3]に格納されたマスクデータに従って、[S1] と [S2] のデータを結合します。
- 演算結果は、[D]を先頭とするアドレスに格納します。
- [S3]で指定したマスクデータが ON のビットは[S1]から、OFF のビットは[S2]のデータを結合対象にします。  
 $\{ [S1] \wedge [S3] \} \vee \{ [S2] \wedge \neg [S3] \} \rightarrow [D]$

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[I]...US,SS

[S1]...DT0 [S2]...DT1 [S3]...DT2 [D]...DT4

DT0	<b>H 1234</b>	}	DT0	H 1234
DT1	<b>H 5678</b>		DT1	H 5678
DT2	<b>H FF00</b>		DT2	H FF00
DT3	H 00FF		DT3	H 00FF
DT4	H 90AB		DT4	<b>H 1278</b>

例2) 演算単位が32ビット(UL,SL)の場合

[I]...UL,SL

[S1]...TS0 [S2]...TS1 [S3]...TS3 [D]...TS4

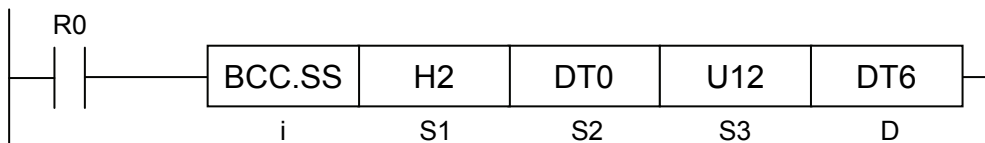
TS0	<b>H 12345678</b>	}	TS0	H 12345678
TS1	<b>H 90ABCDEF</b>		TS1	H 90ABCDEF
TS2	H 0F0F0F0F		TS2	H 0F0F0F0F
TS3	<b>H F0F0F0F0</b>		TS3	H F0F0F0F0
TS4	H 5AA5A55A		TS4	<b>H 103B5D7F</b>

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# BCC (ブロックチェックコード計算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
S1	計算方法を指定するデータを格納しているエリアまたは定数データ
S2	対象データを格納しているエリアの先頭アドレス
S3	対象データの長さ(バイト数)を格納しているエリア、または定数データ
D	演算結果を格納するエリア

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF	DF		" "
S1	●	●	●	●			●	●	●	●	●				●	●	●				●
S2	●	●	●	●			●	●	●	●	●										●
S3	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*2: 演算単位が符号付き整数(SS)の場合のみ指定可能

\*3: 演算単位が符号なし整数(US)の場合のみ指定可能

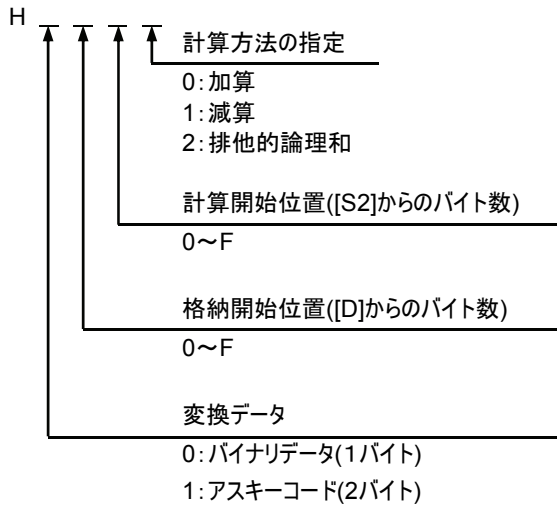
\*4: 演算単位が整数(US、SS)の場合のみ指定可能

## ■ 動作説明

- ブロックチェックコード(BCC)を計算します。
- [S2]で指定された計算開始位置より[S3]で指定されたバイト数分の対象データを、[S1]で指定する計算方法でブロックチェックコード(BCC)を計算します。
- 演算結果は、[D]と[S1]で指定された格納位置から、[S1]で指定された変換方法に従って格納します。

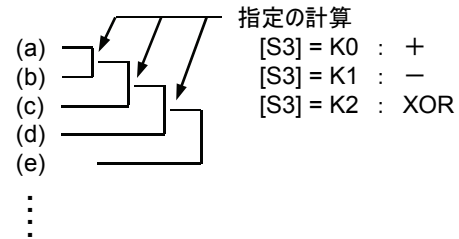


## ■ コントロールデータ[S1]の指定



計算は、次の様に8ビット単位ずつ、  
 指定の計算を行います

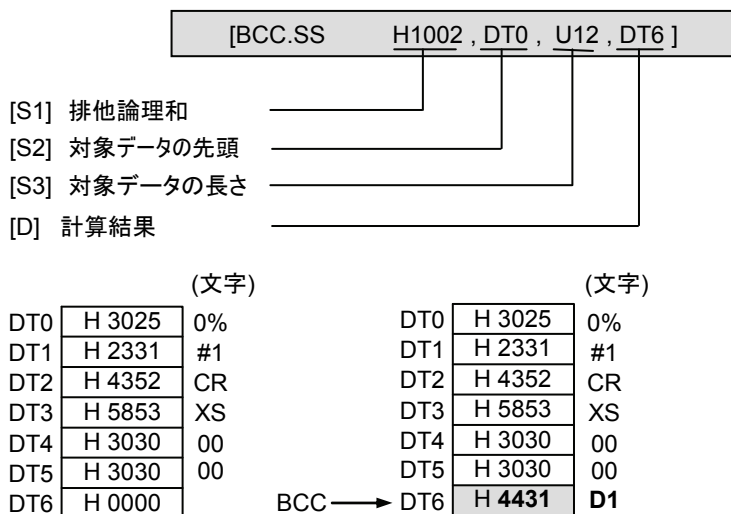
%	→ H 25	→ 00100101	(a)
0	→ H 30	→ 00110000	(b)
1	→ H 31	→ 00110001	(c)
#	→ H 23	→ 00100011	(d)
R	→ H 52	→ 01010010	(e)
			⋮



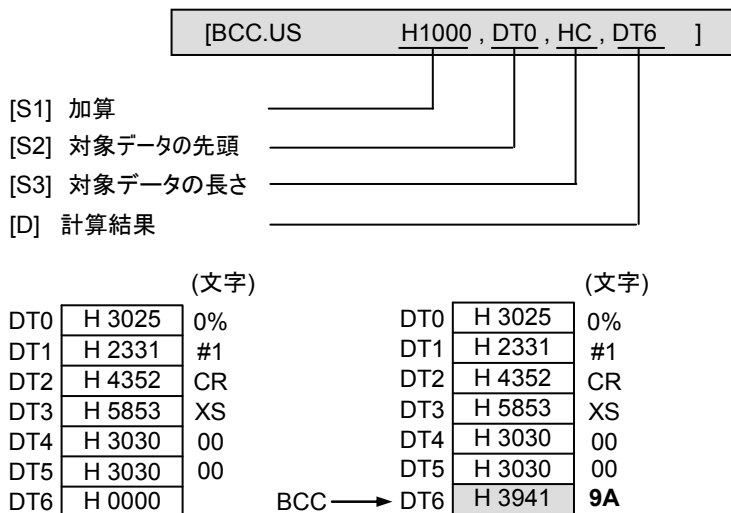
## ■ 変換例

- 送信するメッセージ"%01#RCSX0000"のブロックチェックコード(BCC)を計算して、メッセージの後に付ける例を示します。

例1) 演算単位が16ビット(SS)の場合 / 計算方法: 排他論理和、変換データ: アスキーコード



例2) 演算単位が16ビット(US)の場合 / 計算方法: 加算、変換データ: アスキーコード

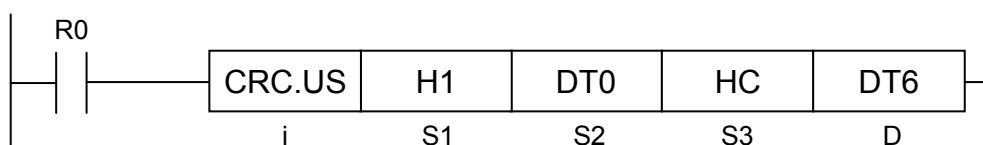


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# CRC (CRCコード計算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
S1	計算方法を指定するデータを格納しているエリアまたは定数データ
S2	対象データを格納しているエリアの先頭アドレス
S3	対象データの長さ(バイト数)を格納しているエリア、または定数データ
D	演算結果を格納するエリア

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K *2	U *3	H *4	SF	DF	" "	
S1	●	●	●	●			●	●	●	●	●				●	●	●				●
S2	●	●	●	●			●	●	●	●	●										●
S3(*5)	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*2: 演算単位が符号付き整数(SS)の場合のみ指定可能

\*3: 演算単位が符号無し整数(US)の場合のみ指定可能

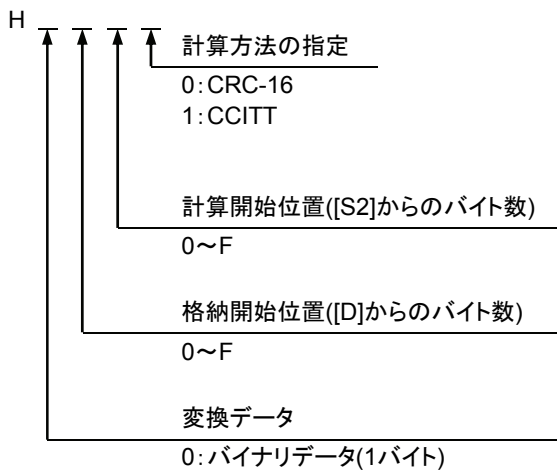
\*4: 演算単位が整数(US、SS)の場合のみ指定可能

\*5: 演算単位にかかわらず 16ビット符号無し整数(US)として扱

## ■ 動作説明

- ブロックチェックコード(BCC)をCRC(16ビットCRCコード)計算方式で計算します。
- [S2]で指定された計算開始位置より[S3]で指定されたバイト数分の対象データを、[S1]で指定する計算方法でブロックチェックコード(BCC)をCRC計算方式で計算します。
- 演算結果は、[D]と[S1]で指定された格納位置から、[S1]で指定された変換方法に従って格納します。

## ■ コントロールデータ[S1]の指定

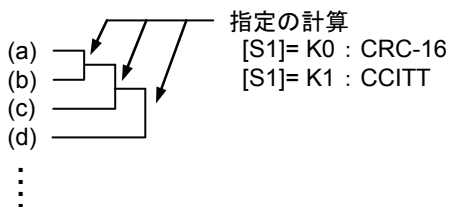


- 計算方法は、つぎのように 16 ビット単位ずつ、指定の計算を行います。
- 計算方式の指定を CRC にした場合、下記の生成多項式を用いて計算を行います。  
(MODBUS-RTUと同じ計算方法です)

・生成多項式:(CRC-16)  $X^{16} + X^{15} + X^2 + 1$     ・生成多項式:(CRCITT)  $X^{16} + X^{12} + X^5 + 1$

%	→	H 25	→	00100101	(a)
0	→	H 30	→	00110000	(b)
1	→	H 31	→	00110001	(c)
#	→	H 23	→	00100011	(d)
R	→	H 52	→	01010010	(e)

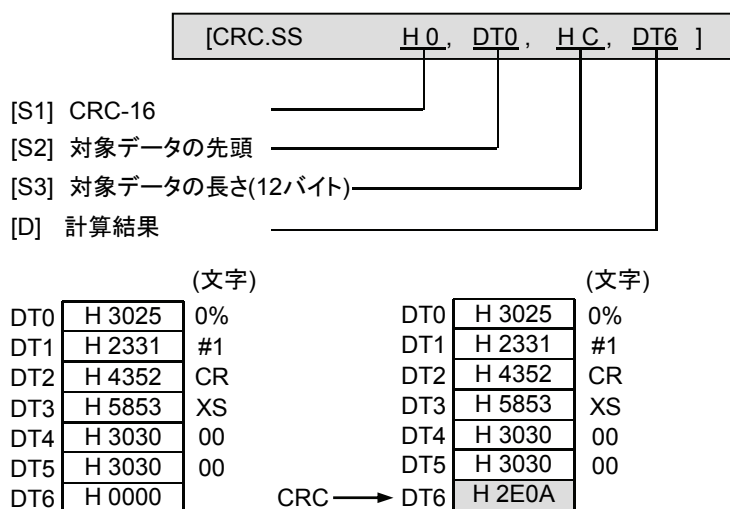
⋮



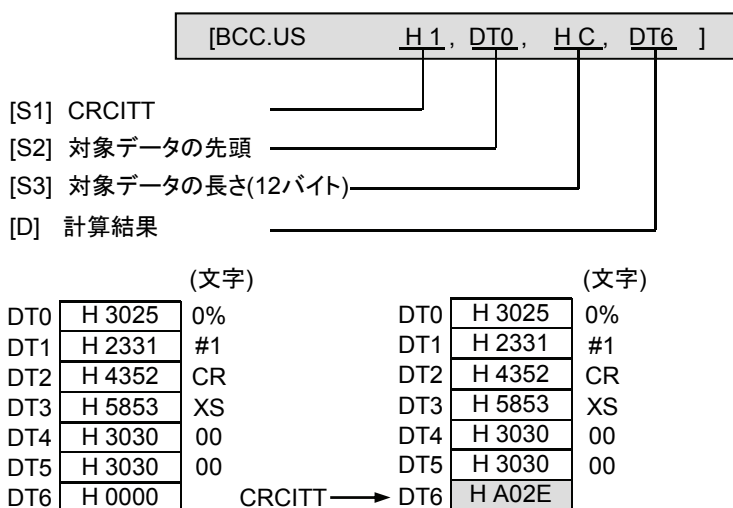
## ■ 変換例

- 送信するメッセージ"%01#RCSX0000"のブロックチェックコードを計算して、メッセージの後に付ける例を示します。

例1.演算単位が16ビット(SS)の場合 /計算方法:CRC-16、変換データ:バイナリデータ



例2.演算単位が16ビット(US)の場合 /計算方法:CRCITT、変換データ:バイナリデータ

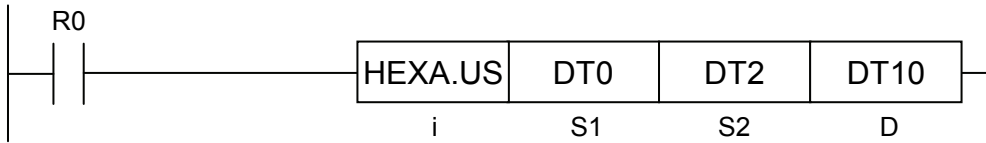


## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S1]で指定した計算方法が指定範囲外の場合
(ER)	[S1]で指定した変換データが指定範囲外の場合

# HEXA (HEX→16進ASCII変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S1	16進数の数値を格納しているエリアの先頭番号
S2	変換する数値の長さ(バイト数)を格納しているエリア、または定数
D	変換結果のASCIIコードを格納するエリアの先頭番号

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾 *1		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K *2	U *3	H *4	SF	DF		""	
S1	●	●	●	●	●	●	●	●													●	
S2	●	●	●	●			●	●							●	●	●					●
D	●	●	●	●			●	●														●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*2: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*3: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*4: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

## ■ 動作説明

- 16進数の数値を、ASCIIコード化します。
- [S1]で指定するエリアに格納されている16進数の数値データをASCIIコードに変換し、[D]で指定するエリアに格納します。
- データは、[S2]で指定するバイト数分、変換します。
- 結果(ASCIIコード)の量は、変換したデータの2倍です。

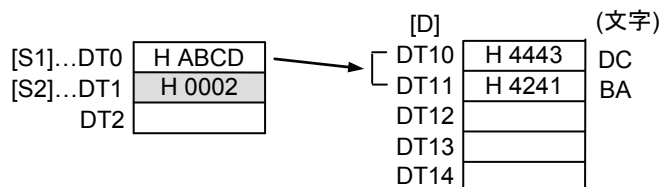
## ■ 変換例

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[S1]...DT0 [S2]...DT1

[D]...DT10

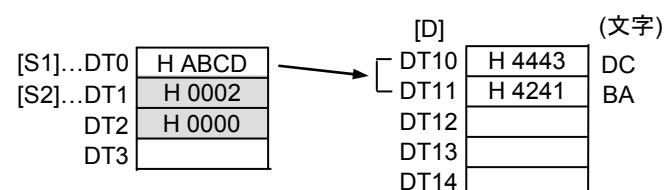


例2) 演算単位が32ビット(UL,SL)の場合

[i]...UL,SL

[S1]...DT0 [S2]...DT1

[D]...DT10

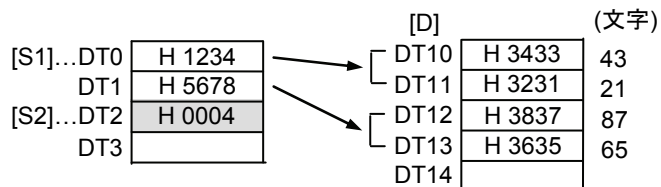


例3) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

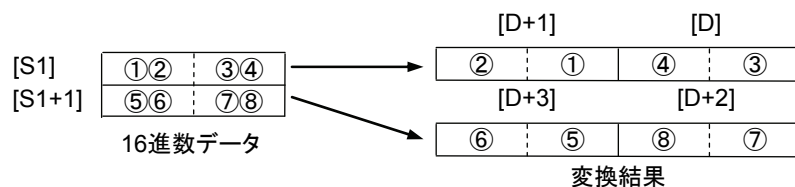
[S1]...DT0 [S2]...DT2

[D]...DT10



## ■ プログラム上のご注意

- 1 バイト分を構成する 2 文字は、変換すると入れ替わって格納されます。
- 2 バイト分を一区切りにして変換します。

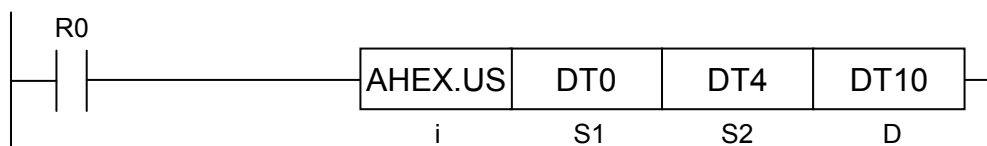


## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S2]のバイト数指定で、変換範囲がエリアを超えるとセットします
(ER)	変換結果がエリアを超えるとセットします
	[S2]の指定が"0"のとき、または、(-)負値のときセットします

# AHEX (16進ASCII→HEX変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S1	ASCII コードを格納しているエリアの先頭番号
S2	変換する ASCII コードの数(文字数)を格納しているエリア、または定数
D	変換結果の 16 進数の数値を格納するエリアの先頭番号

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデックス 修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K *2	U *3	H *4	SF	DF		" "
S1	●	●	●	●	●	●	●	●													●
S2	●	●	●	●			●	●							●	●	●				●
D	●	●	●	●			●	●													●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*2: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*3: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*4: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

## ■ 動作説明

- ASCII コードの文字列を、16 進の数値に変換します。
- [S1]で指定するエリアに格納されている ASCII コードを、16 進数の数値に変換して、[D]で指定するエリアに格納します。
- 変換する ASCII コードの数(文字数)は、[S2]で指定します。
- 変換結果は、バイト単位で格納されます。
- 変換する文字数が奇数の場合、変換結果の最終データ(バイト)のビット(0~3)には、「0」が入ります。



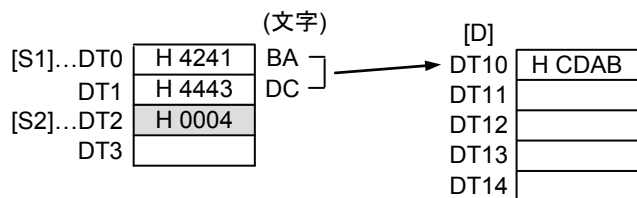
## ■ 変換例

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[S1]...DT0 [S2]...DT2

[D]...DT10

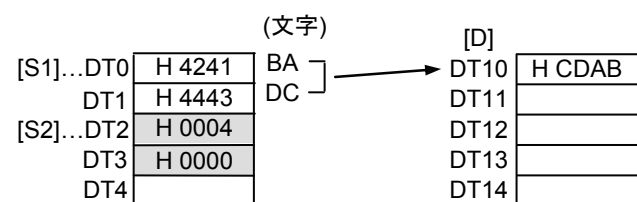


例2) 演算単位が32ビット(UL,SL)の場合

[i]...UL,SL

[S1]...DT0 [S2]...DT2

[D]...DT10

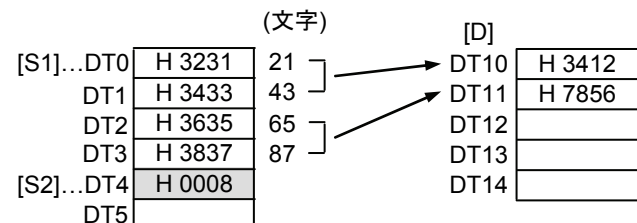


例3) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[S1]...DT0 [S2]...DT4

[D]...DT10

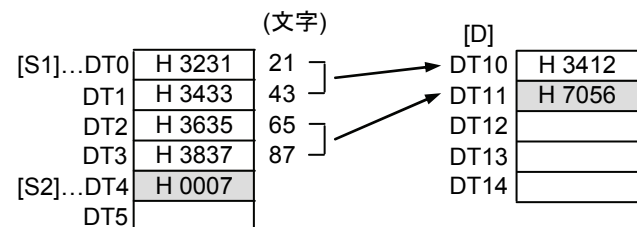


例4) 演算単位が16ビット(US,SS)で、変換する文字数が奇数の場合

[i]...US,SS

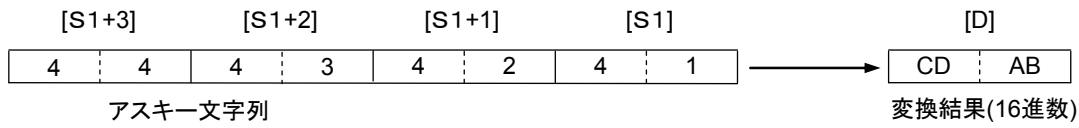
[S1]...DT0 [S2]...DT4

[D]...DT10



### ■ プログラム上のご注意

- ASCIIコード2文字分が1バイトの数値2桁に変換されます。このとき、上位の文字と下位の文字が入れ替わります。
- 4文字分を一区切りにして変換します。

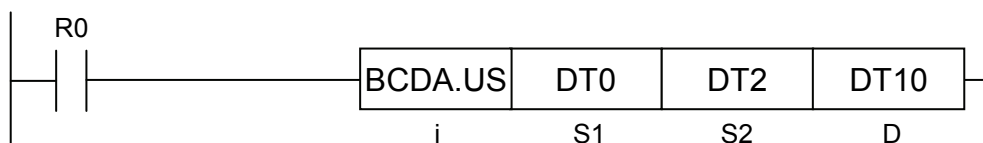


### ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	[S2]の文字数指定で、変換範囲がエリアを超えるときセットします
	変換結果がエリアを超えるときセットします
	[S2]の指定が"0"のとき、または、(-)負値のときセットします
	[S1]で指定した ASCII コード中に「0~F」以外の文字コードがあるときセットします

# BCDA (BCD→10進ASCII変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
S1	BCDの数値を格納しているエリアの先頭番号
S2	変換するデータ量および変換方向を表すデータを格納しているエリア、または定数
D	変換結果のASCIIコードを格納するエリアの先頭番号

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾*1		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		""	
S1	●	●	●	●	●	●	●	●													●	
S2	●	●	●	●			●	●								●	●					●
D	●	●	●	●			●	●														●

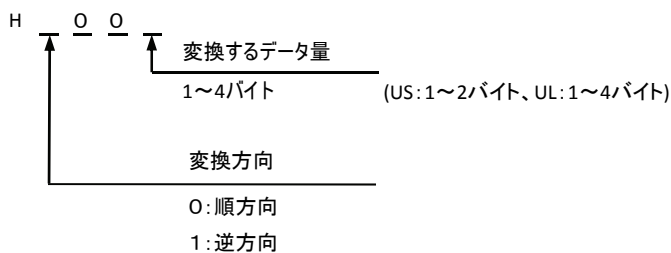
\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- 最大 8 桁の BCD データを ASCII コードの文字列に変換します。
- [S1]で指定するエリアに格納されている BCD の数値を、ASCII コードに変換します。
- 変換結果は、[D]を先頭とするエリアに格納します。
- 変換する BCD データ量(バイト数)および変換方向は、[S2]で指定します。
- 変換結果(ASCII コード)の量は、変換したデータの 2 倍です。
- 変換するデータ量の最大値は、演算単位により変わります。(US:2 バイト、UL:4 バイト)
- 変換するデータ量は、バイト単位で指定しますので、1ワードデータの下位バイトのみを変換することも可能です。

## ■ 変換データ量と変換方向の設定[S2]

4桁H定数で指定してください。



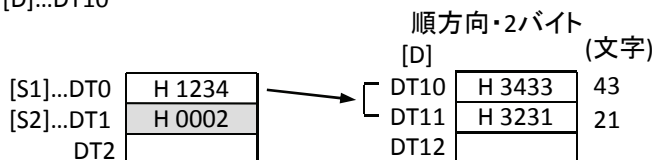
## ■ 変換例

例1)演算単位が16ビット(US)の場合

[i]...US

[S1]...DT0 [S2]...DT1

[D]...DT10

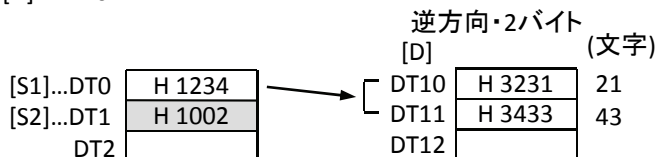


例2)演算単位が16ビット(US)の場合

[i]...US

[S1]...DT0 [S2]...DT1

[D]...DT10

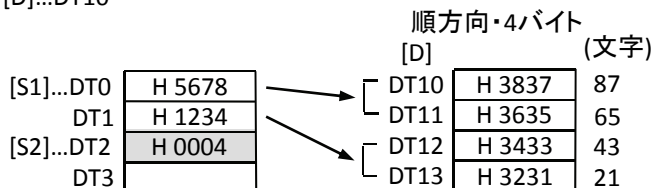


例3)演算単位が32ビット(UL)の場合

[i]...UL

[S1]...DT0 [S2]...DT2

[D]...DT10

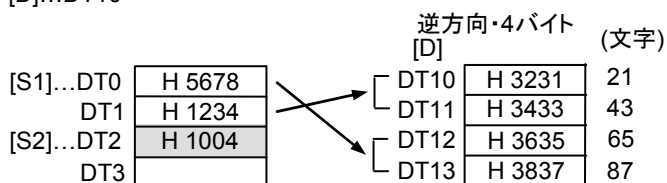


例4)演算単位が32ビット(UL)の場合

[i]...UL

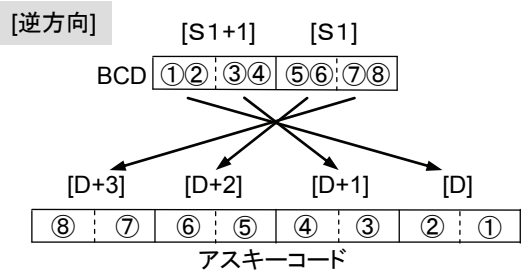
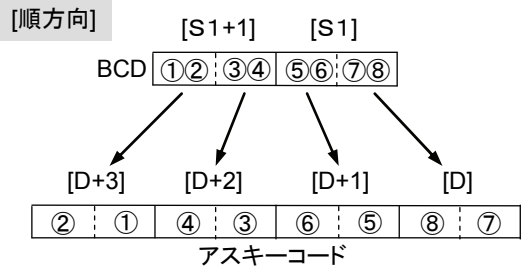
[S1]...DT0 [S2]...DT2

[D]...DT10



## ■ プログラム上のご注意

- 1 バイト分の構成する2文字は、変換すると入れ替わって格納されます。
- 2 バイト分を一区切りにして変換します。

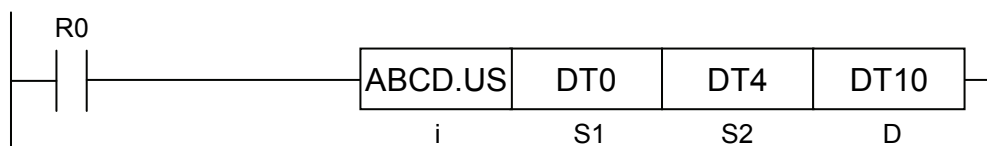


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	[S1]で指定したデータ中にBCD以外のデータがあるときセットします
	[S2]で指定したバイト数が[S1]のエリアを超えるときセットします
	変換結果がエリアを超えるときセットします
	[S2]の指定が"0"のときセットします
	[S2]の変換方向が範囲外の場合セットします
	[S2]の変換するデータ量が範囲外の場合セットします

# ABCD (10進ASCII→BCD変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
S1	ASCII コードを格納しているエリアの先頭番号
S2	変換する ASCII コード数および変換方向を表すデータを格納しているエリア、または定数
D	変換結果の BCD の数値を格納するエリアの先頭番号

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾*1		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		""	
S1	●	●	●	●	●	●	●	●													●	
S2	●	●	●	●			●	●								●	●					●
D	●	●	●	●			●	●														●

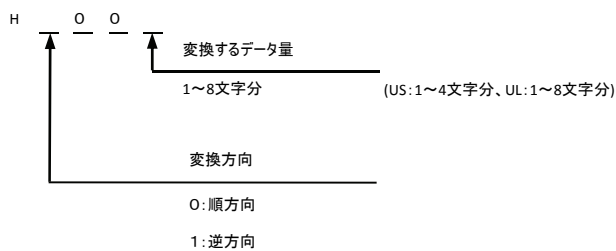
\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定)

## ■ 動作説明

- 最大 8 文字の ASCII コードの文字列を BCD データに変換します。
- [S1]で指定するエリアに格納されている ASCII コードを、BCD のデータに変換して、[D]で指定するエリアに格納します。
- 変換する ASCII コードの数(文字数)および変換方向は、[S2]で指定します。
- 変換結果(BCD データ)の量は、変換した ASCII コード列の半分です。
- 変換するデータ量の最大値は、演算単位により変わります。(US:4 文字、UL:8 文字)
- 変換するデータ量は、バイト単位で指定しますので、1ワードデータの下位バイトのみを変換することも可能です。
- 変換するデータ量が奇数の場合、変換結果の最終データのビットに 0 が入ります。
  - i) ビット 0~3 に「0」が入ります。(順方向の場合)
  - ii) ビット 4~7 に「0」が入ります。(逆方向の場合)

## ■ 変換データ量と変換方向の設定[S2]

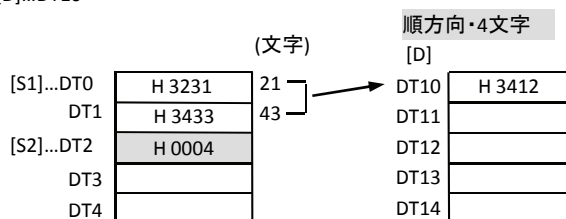
次の形式に従って、4桁のBCD(H定数)で指定してください。



## ■ 変換例

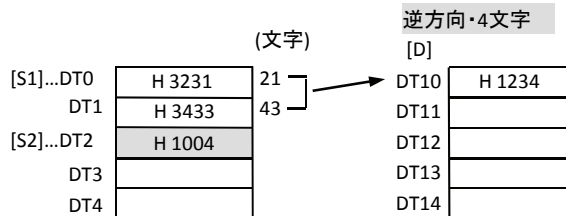
例1) 演算単位が16ビット(US)の場合

[i]...US  
[S1]...DT0 [S2]...DT2  
[D]...DT10



例2) 演算単位が16ビット(US)の場合

[i]...US  
[S1]...DT0 [S2]...DT2  
[D]...DT10



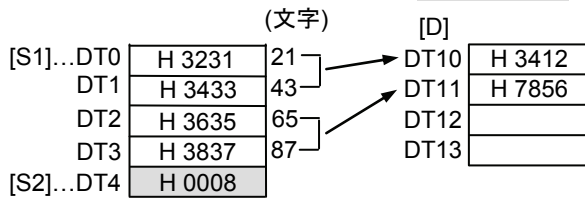
例3) 演算単位が32ビット(UL)の場合

[i]...UL

[S1]...DT0 [S2]...DT4

[D]...DT10

順方向・8文字



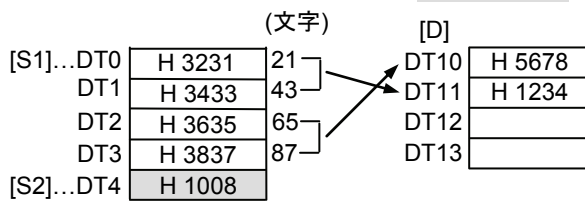
例4) 演算単位が32ビット(UL)の場合

[i]...UL

[S1]...DT0 [S2]...DT4

[D]...DT10

逆方向・8文字



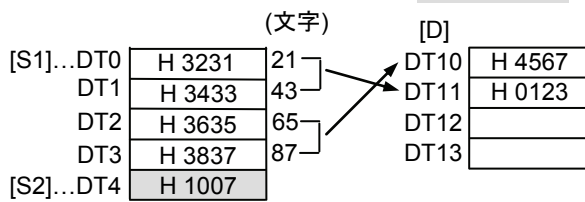
例5) 演算単位が32ビット(UL)の場合

[i]...UL

[S1]...DT0 [S2]...DT4

[D]...DT10

逆方向・7文字

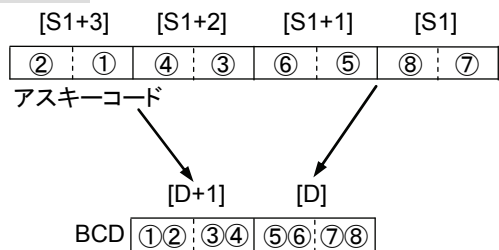




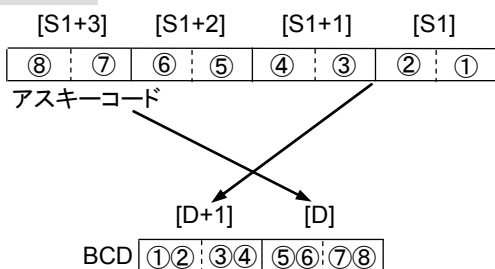
## ■ プログラム上のご注意

- 1 バイト分を構成する 2 文字は、変換すると入れ替わって格納されます。
- 2 バイト分を一区切りにして変換します。

### [順方向]



### [逆方向]

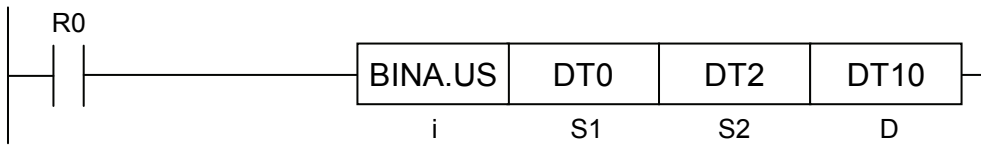


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	[S1]で指定した ASCII コード中に 0~9 以外のデータがあるときセットします
	[S2]で指定したバイト数が[S1]のエリアを超えるときセットします
	変換結果がエリアを超えるときセットします
	[S2]の指定が"0"のときセットします
	[S2]の変換方向が範囲外の場合セットします
[S2]の変換するデータ量が範囲外の場合セットします	

# BINA (BIN→10 進ASCII変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S1	10 進数を表す BIN データの数値を格納しているエリア、または、定数データ
S2	変換結果を格納するエリアのバイト数を格納しているエリアまたは、定数データ
D	変換結果の ASCII コードを格納するエリアの先頭番号

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデック ス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K *2	U *3	H *4	SF	DF		""
S1	●	●	●	●	●	●	●	●							●	●	●				●
S2	●	●	●	●			●	●							●	●	●				●
D	●	●	●	●			●	●													●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*2: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*3: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

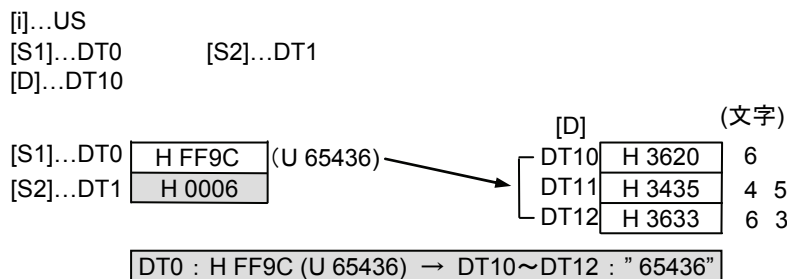
\*4: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可

## ■ 動作説明

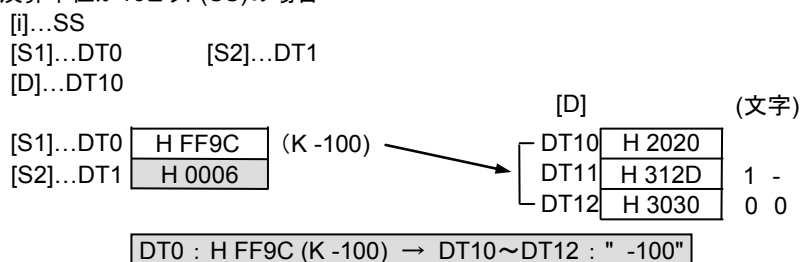
- 10 進数を表す BIN データを ASCII コードの文字列に変換します。
- [S1]で指定する 10 進数で表す BIN データを ASCII コードに変換して、[D]で指定するエリアに格納します。
- 格納エリアは、[D]で先頭を指定し、[S2]で大きさを指定します。
- 変換後の ASCII コードのバイト数(負の符号を含みます)が、[S2]で指定したバイト数よりも大きくなる時は、演算エラーになります。
- 符号を含めた変換対象の桁数を[S2]で指定してください。

## ■ 変換例

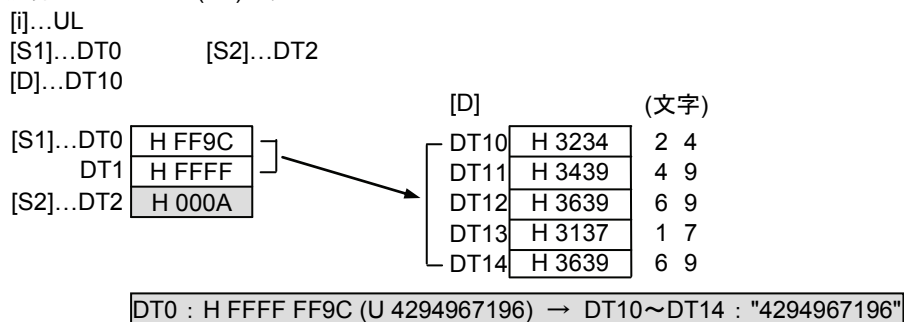
例1) 演算単位が16ビット(US)の場合



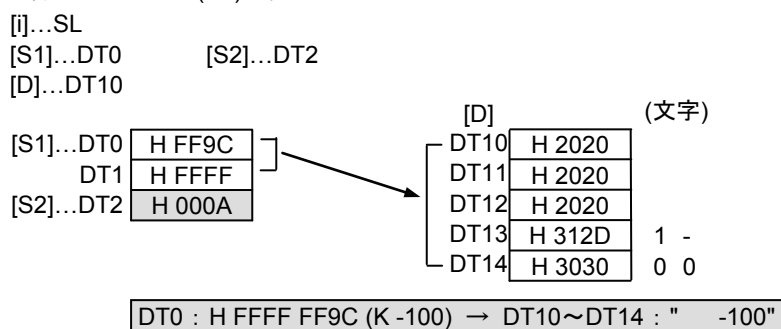
例2) 演算単位が16ビット(SS)の場合



例3) 演算単位が32ビット(UL)の場合



例4) 演算単位が32ビット(SL)の場合

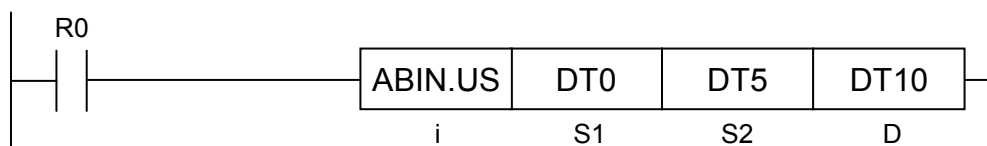


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	[S2]で指定したバイト数が[D]のエリアを超えるときセットします
	変換結果がエリアを超えるときセットします
	変換結果のバイト数が[S2]で指定しているバイト数を超えるときセットします

# ABIN (10 進ASCII→BIN変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S1	変換する ASCII コードを格納しているエリアの先頭番号
S2	変換するバイト数を格納しているエリアまたは、定数データ
D	変換結果を格納するエリア

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデックス 修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K *2	U *3	H *4	SF	DF		""
S1	●	●	●	●	●	●	●	●													●
S2(*5)	●	●	●	●			●	●							●	●	●				●
D	●	●	●	●			●	●													●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*2: 演算単位が符号付き整数(SS, SL)の場合のみ指定可能

\*3: 演算単位が符号なし整数(US, UL)の場合のみ指定可能

\*4: 演算単位が整数(US, SS, UL, SL)の場合のみ指定可

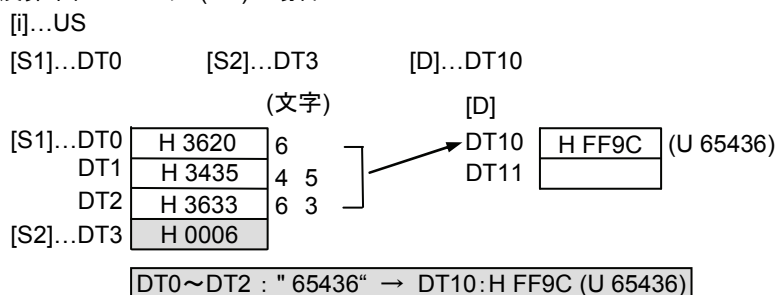
\*5: 演算単位にかかわらず 16ビット整数(US, SS)として扱

## ■ 動作説明

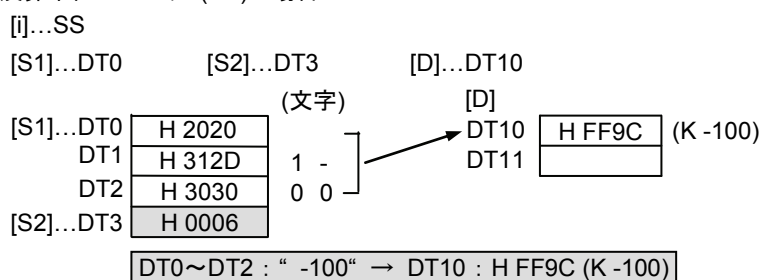
- ASCII コードの文字列を、10 進数を表す BIN データに変換します。
- [S1]で指定したエリアから[S2]で指定したバイト数(=文字数)分の 10 進数の数値を表す ASCII コードを、10 進数の数値に変換し[D]で指定したエリアに格納します。
- [S1]で指定する ASCII コードに、負の符号(-)が含まれる場合は、演算単位 SS, SL を指定してください。演算単位 US, UL の場合は、演算エラーとなります。

## ■ 変換例

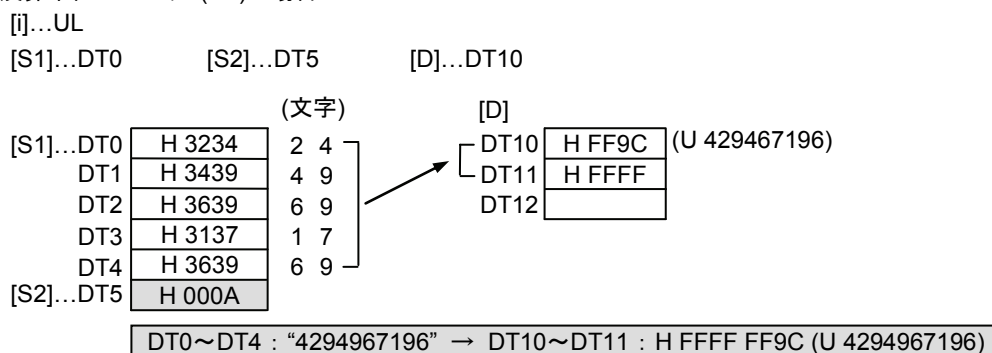
例1) 演算単位が16ビット(US)の場合



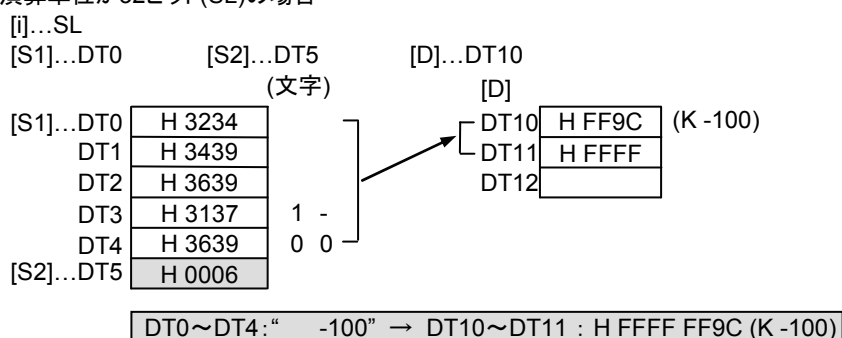
例2) 演算単位が16ビット(SS)の場合



例3) 演算単位が32ビット(UL)の場合



例4) 演算単位が32ビット(SL)の場合

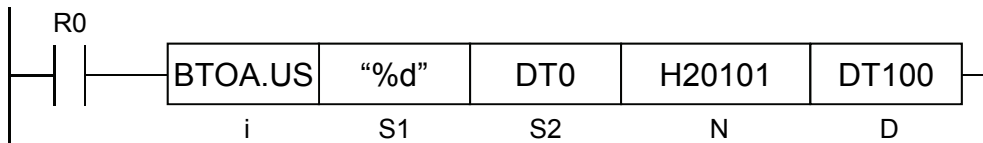


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	[S2]で指定したバイト数が[D]のエリアを超えるときセットします
	変換結果がエリアを超えるときセットします
	[S1]に0~9、符号コード、スペース以外の文字のASCIIコードが含まれているときセットします

# BTOA (BIN→ASCII変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	制御文字列(2~16文字)
S2	バイナリデータを格納しているエリアの先頭番号
N	変換方法
D	変換結果の ASCII コードを格納するエリアの先頭番号

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *1		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "			
S1	●	●	●	●			●	●													●	●	
S2	●	●	●	●	●	●	●	●															●
N(*2)	●	●	●	●			●	●								●	●						●
D	●	●	●	●			●	●															●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*2: 演算単位にかかわらず 32ビット整数(UL)として扱う

## ■ 動作説明

- [S2]を先頭とするエリアに格納されているバイナリデータを、[S1]と[N]で指定された変換方法により、ASCII コードに変換します。
- 演算結果は、[D]を先頭とするエリアに格納します。

## ■ 制御文字列の指定 [S1]

制御文字列	変換データの形式	指定可能な演算単位	使用例
"%d" or "%i"	16 ビットデータを 10 進 ASCII データへ変換 (符号あり整数)	SS	"%d", "%5d", "%+5d", "%-5d" "%05d", "%10.5d", "%d," "% d"
	32 ビットデータを 10 進 ASCII データへ変換 (符号あり整数)	SL	
"%u"	16 ビットデータを 10 進 ASCII データへ変換 (符号なし整数)	US	"%u", "%5u", "%+5u", "%-5u" "%05u", "%10.5u", "%u,"
	32 ビットデータを 10 進 ASCII データへ変換 (符号なし整数)	UL	
"%x"	16 ビットデータを 16 進 ASCII データへ変換 (順/逆方向)	US	"%x", "%5x", "%-5x", "%05x" "%10.5x", "%x,", "%#x", "%X"
	32 ビットデータを 16 進 ASCII データへ変換 (順/逆方向)	UL	
"%b"	16 ビット BCD データを 10 進 ASCII データへ変換 (順/逆方向)	US	"%b", "%5b", "%-5b", "%05b" "%10.5b", "%b,"
	32 ビット BCD データを 10 進 ASCII データへ変換 (順/逆方向)	UL	
"%f"	32 ビット単精度実数データを 浮動小数点 ASCII データへ変換	SF	"%f", "%5.2f", "%+5.2f" "%-5.2f", "%05.2f", "%f," "%#f", "% f"
"%e"	32 ビット単精度実数データを 指数表示 ASCII データへ変換	SF	"%e", "%5.2e", "%+5.2e" "%-5.2e", "%05.2e", "%e," "%#5.2e", "% e", "%E"
"%s"	32 ビット単精度実数データを 指数表示 ASCII データ または、浮動小数点 ASCII データへ変換 (表現できる短い方の書式)	SF	"%g", "%5.2g", "%+5.2g" "%-5.2g", "%05.2g", "%g," "%#5.2g", "%G"

## ■ 制御文字列の形式[S1]

1) アルファベット大文字/小文字の指定

16 進数・指数表示 ASCII データで使用するアルファベットの大文字、小文字を指定します。

制御文字列	バイナリデータ	ASCIIデータ
%x	H ABCD	"abcd"
%X	H ABCD	"ABCD"
%e	1234.5678	"1.2345678e+3"
%E	1234.5678	"1.2345678E+3"

## 2)表示桁数の指定

表示桁数の指定は、「全体の文字数」と「精度の文字数」で指定します。

指定がなければデフォルトが使用されます。

「n.m」「n」「.m」などで指定します。

n:全体の文字数、m:精度の文字数

<精度の文字数について>

[d, i, u, x, X, b]は、数値列の文字数を表します。

[f, e, E]は、小数部以下の文字数を表します。

[g, G]は、有効桁数を表します。

制御文字列	バイナリデータ	ASCIIデータ
%d	K 100	"100"
%5d	K 100	"@@100"
%10.5d	K 100	"@@@@@00100"
%x	H 12A	"12a"
%5x	H 12A	"@@12a"
%10.5x	H 12A	"@@@@@0012a"
%b	H 123	"123"
%5b	H 123	"@@123"
%f	123.45678	"123.45678"
%8.3f	123.45678	"@123.456"
%e	1234.5678	"1.2345678e+3"
%9.3e	1234.5678	"@1.234e+3"
%g	1234.5678	"1234.5678"
%8.6g	1234.5678	"@1234.56"

(注)"@"は、スペースを表します

## 3)ゼロパディングの指定

表示数の指定がある場合、ゼロ詰めを指定することができます。表示桁数の前にゼロ(0)を付けてください。

制御文字列	バイナリデータ	ASCIIデータ
%05d	K 100	"00100"
%05x	H 12A	"0012a"
%05b	H 123	"00123"
%08.3f	123.45678	"0123.456"
%09.3e	1234.5678	"01.234e+3"

## 4)後詰・前詰の指定

デフォルトは、後詰です。前詰にしたいときは、桁数指定の前にマイナス(-)を付けてください。

制御文字列	バイナリデータ	ASCIIデータ
%-5d	K 100	"100@@"
%-5x	H 12A	"12a@@"
%-5b	H 123	"123@@"
%-8.3f	123.45678	"123.456@"
%-9.3e	1234.5678	"1.234e+3@"

(注)"@"は、スペースを表します



### 5) 符号の指定

デフォルトでは、プラス記号(+)を付加しません。プラス記号(+)を付加したいときは、(+)を付加してください。

制御文字列	バイナリデータ	ASCIIデータ
%+d	K 100	"+100"
%+d	K -100	"-100"
%+5d	K 100	"@+100"
%+8.3f	123.45678	"+123.456"
%+9.3e	1234.5678	"+1.234e+3"

(注)"@"は、スペースを表します

### 6) 数値位置の指定

正数の場合にスペースを挿入し、負数の場合と位置をあわせませす。位置をあわせる場合は、(空白)を付加してください。

制御文字列	バイナリデータ	ASCIIデータ
%_d	K 100	"@100"
%_d	K -100	"-100"
%_8.3f	123.45678	"@123.456"
%_8.3f	-123.45678	"-123.456"
%_9.3e	1234.5678	"@1.234e+3"
%_9.3e	-1234.5678	"-1.234e+3"

(注)"@"は、スペースを表します

### 7) 数値データ型に対する別の出力形式の指定

"#"を付加することにより、別の出力形式を自動的に付加します。

制御文字列	バイナリデータ	ASCIIデータ	備考
%#x	H 12A	"0x12a"	"0x"を付加します
%#X	H 12A	"0X12A"	"0X"を付加します
%#8.0f	123.45678	"@@@@"123."	"."を必ず付加します
%#9.0e	1234.5678	"@@@@@1.e+3"	
%#9.3E	1234.5678	"@@@@@1.E+3"	
%#9.0g	1234	"@@@@"1234.0"	"."を必ず付加し、さらに小数点に続く"0"を省略しません
%#.9G	1234	"1234.56780"	

(注)"@"は、スペースを表します

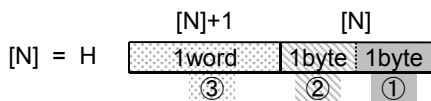
### 8) 数値の後に文字付加を指定

変換指定文字(d, x, b, f, e, g など)の後ろの文字を数値 ASCII データの後ろに付加します。

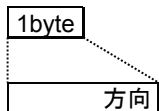
制御文字列	バイナリデータ	ASCIIデータ
%d,	K 100	"100,"
%x,	H 100	"100,"
%xH	H 100	"100H"
%bBCD	H 100	"100BCD"

## ■ 変換方法 [N] の指定

• 変換方法[N]は、32ビットエリアに Hex 形式で指定します。



### ① : アスキーデータの方向



方向:「0」=「順方向」、 「1」=「逆方向」

下記の方向を制約とします

制御文字列	方向
"%d"	逆方向のみ
"%u"	逆方向のみ
"%x"	順・逆方向
"%b"	順・逆方向
"%f"	逆方向のみ
"%e"	逆方向のみ
"%g"	逆方向のみ

### ② : 格納開始位置

[D]の下位バイトを「0」として指定

指定範囲 : 「0」~アスキーデータ桁数の最大値以内

例)変換開始位置(1)の場合

[D]	0byte	**	←[D]のHighアドレスから格納される
	1byte	31	
[D]+1	2byte	32	
	3byte	33	
[D]+2	4byte		
	.		
	.		

### ③ : 変換データ数

変換データ数を指定

指定範囲 : 「1~65535」

## ■ 変換例

例1) 演算単位が16ビット(US)の場合

[i]...US

[S1]..."%-6u," : 16ビットデータを10進アスキーデータ(6桁)へ変換

[S2]...DT0

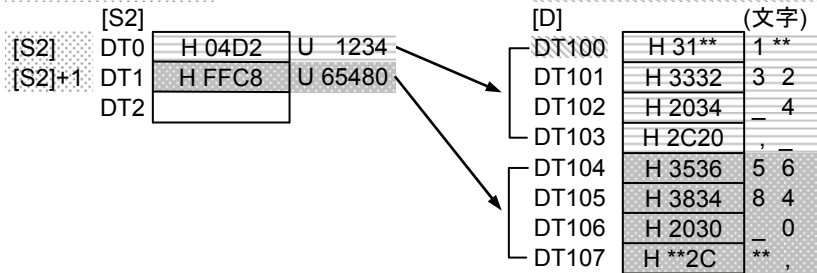
[N]...H 00020101 : ①アスキーデータ(逆方向)

[D]...DT100 : ②格納開始位置(1) → [D]+1バイト

③変換データ数(2) → [S2], [S2]+1 を変換

③変換データ数 = 2

②格納開始位置 = +1バイト



DT0 : U 1234 (H 04D2) → DT100~DT103 : "1234 ,"

DT1 : U 65480 (H FFC8) → DT104~DT107 : "65480 ,"

例2) 演算単位が16ビット(SS)の場合

[i]...SS

[S1]..."%05d" : 16ビットデータを10進アスキーデータ(5桁)へ変換(0パディング)

[S2]...DT0

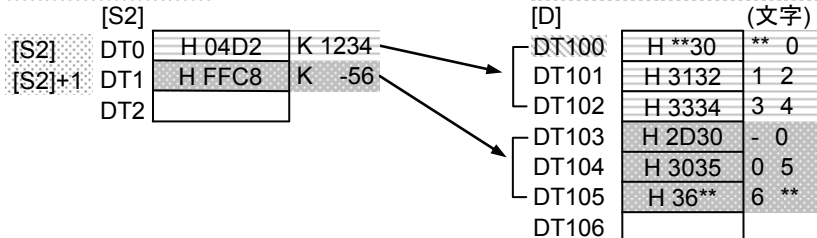
[N]...H 00020101 : ①アスキーデータ(逆方向)

[D]...DT100 : ②格納開始位置(1) → [D]+1バイト

③変換データ数(2) → [S2], [S2]+1 を変換

③変換データ数 = 2

②格納開始位置 = +1バイト



DT0 : K 1234 (H 04D2) → DT100~DT102 : "01234"

DT1 : K -56 (H FFC8) → DT103~DT105 : "-0056"



例5) 演算単位が16ビット(US)の場合

[i]...US

[S1]..."%+10.5u" : 16ビットデータを10進アスキーデータ(10桁)へ変換(符号の指定)

[S2]...DT0

[N]...H 00020001 : ①アスキーデータ(逆方向)  
 ②格納開始位置(0) → [D]+0バイト  
 ③変換データ数(2) → [S2], [S2]+1 を変換

③変換データ数 = 2

[S2]	DT0	H 04D2	U 1234
[S2]+1	DT1	H FFC8	U 65480
	DT2		

②格納開始位置 = +0バイト  
 [D] (文字)

[D]	DT100	H 2020	
	DT101	H 2020	
	DT102	H 2B30	+ 0
	DT103	H 3132	1 2
	DT104	H 3334	3 4
	DT105	H 2020	
	DT106	H 2020	
	DT107	H 2B36	+ 6
	DT108	H 3534	5 4
	DT109	H 3830	8 0

DT0 : U 1234 (H 04D2) → DT100~DT104 : " +01234"  
 DT1 : U 65480 (H FC8) → DT105~DT109 : " +65480"

例6) 演算単位が32ビット(UL)の場合

[i]...UL

[S1]..."%-11u" : 32ビットデータを10進アスキーデータ(11桁)へ変換(符号の指定)

[S2]...DT0

[N]...H 00020001 : ①アスキーデータ(逆方向)  
 ②格納開始位置(0) → [D]+0バイト  
 ③変換データ数(2) → [S2], [S2]+2 を変換

③変換データ数 = 2

[S2]	DT0	H 04D2	U 1234
	DT1 <td>H 0000 <td></td> </td>	H 0000 <td></td>	
[S2]+2	DT2 <td>H FFC8 <td>U 4294967240</td> </td>	H FFC8 <td>U 4294967240</td>	U 4294967240
	DT3 <td>H FFFF <td></td> </td>	H FFFF <td></td>	
	DT4 <td></td> <td></td>		

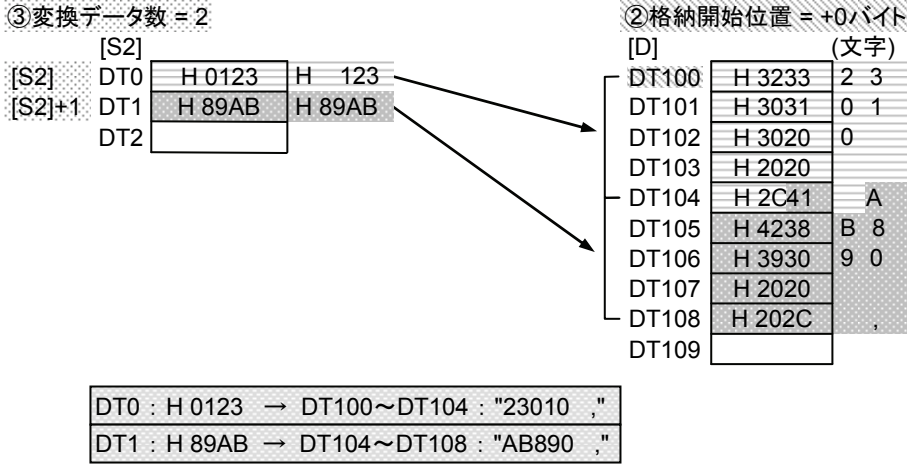
②格納開始位置 = +0バイト  
 [D] (文字)

[D]	DT100	H 3132	1 2
	DT101 <td>H 3334 <td>3 4</td> </td>	H 3334 <td>3 4</td>	3 4
	DT102 <td>H 2020 <td></td> </td>	H 2020 <td></td>	
	DT103 <td>H 2020 <td></td> </td>	H 2020 <td></td>	
	DT104 <td>H 2020 <td></td> </td>	H 2020 <td></td>	
	DT105 <td>H 2034 <td>4</td> </td>	H 2034 <td>4</td>	4
	DT106 <td>H 3239 <td>2 9</td> </td>	H 3239 <td>2 9</td>	2 9
	DT107 <td>H 3439 <td>4 9</td> </td>	H 3439 <td>4 9</td>	4 9
	DT108 <td>H 3637 <td>6 7</td> </td>	H 3637 <td>6 7</td>	6 7
	DT109 <td>H 3234 <td>2 4</td> </td>	H 3234 <td>2 4</td>	2 4
	DT110 <td>H 3020 <td>0</td> </td>	H 3020 <td>0</td>	0
	DT111 <td></td> <td></td>		

DT0~DT1 : U 1234 (H 0000 04D2) → DT100~DT105 : "1234 "  
 DT2~DT3 : U 4294967240 (H FFFF FFC8) → DT105~DT110 : "4294967240 "

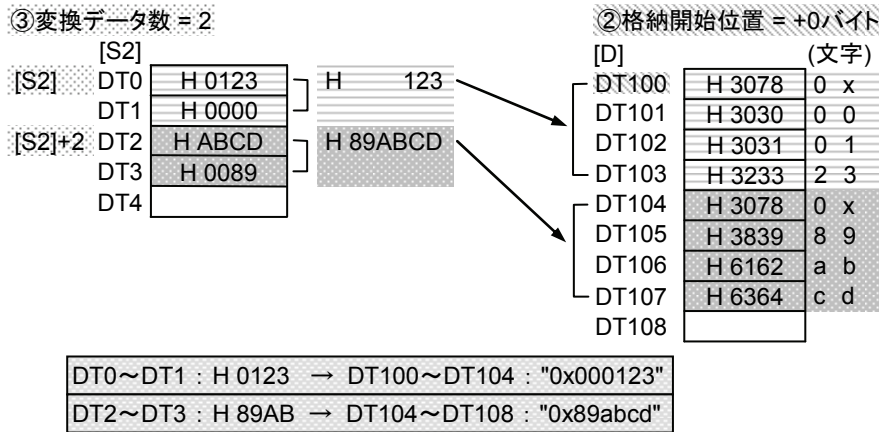
例7) 演算単位が16ビット(US)の場合

[i]...US  
 [S1]..."%8.5X," : 16ビットデータを16進アスキーデータ(8桁)へ変換  
 [S2]...DT0  
 [N]...H 00020000 : ①アスキーデータ(順方向)  
 [D]...DT100 : ②格納開始位置(0) → [D]+0バイト  
 : ③変換データ数(2) → [S2], [S2]+1 を変換



例8) 演算単位が32ビット(UL)の場合

[i]...UL  
 [S1]..."%#8.6x" : 32ビットデータを16進アスキーデータ(8桁)へ変換  
 [S2]...DT0  
 [N]...H 00020001 : ①アスキーデータ(逆方向)  
 [D]...DT100 : ②格納開始位置(0) → [D]+0バイト  
 : ③変換データ数(2) → [S2], [S2]+2 を変換



例9) 演算単位が32ビット(UL)の場合

[I]...UL  
 [S1]..."%05b" : 16ビットデータを10進アスキーデータ(5桁)へ変換  
 [S2]...DT0  
 [N]...H 00020000 : ①アスキーデータ(順方向)  
 [D]...DT100 : ②格納開始位置(0) → [D]+0バイト  
 : ③変換データ数(2) → [S2], [S2]+1 を変換

③変換データ数 = 2

[S2]	DT0	H 0123	H 123
[S2]+1	DT1	H 3456	H 3456
	DT2		

②格納開始位置 = +0バイト

[D]	DT100	H 3032	0 2	(文字)
	DT101	H 3330	3 0	
	DT102	H 3130	1 0	
	DT103	H 3536	5 6	
	DT104	H 3334	3 4	
	DT105			

DT0 : H 0123 → DT100~DT104 : "02301"  
 DT1 : H 89AB → DT104~DT108 : "05634"

例10) 演算単位が32ビット(UL)の場合

[I]...UL  
 [S1]..."%+10.7" : 32ビットBCDデータを10進アスキーデータ(10桁)へ変換  
 [S2]...DT0  
 [N]...H 00020001 : ①アスキーデータ(逆方向)  
 [D]...DT100 : ②格納開始位置(0) → [D]+0バイト  
 : ③変換データ数(2) → [S2], [S2]+2 を変換

③変換データ数 = 2

[S2]	DT0	H 0123	H 123
[S2]	DT1	H 0000	
[S2]+2	DT2	H 3456 <th>H 123456</th>	H 123456
	DT3	H 0012	
	DT4		

②格納開始位置 = +0バイト

[D]	DT100	H 2020		(文字)
	DT101	H 2B30	+ 0	
	DT102	H 3030	0 0	
	DT103	H 3031	0 1	
	DT104	H 3233	2 3	
	DT105	H 3239		
	DT106	H 3439	+ 0	
	DT107	H 3637	1 2	
	DT108	H 3234	3 4	
	DT109	H 3536	5 6	
	DT110			

DT0~DT1 : U 1234 (H 0000 04D2) → DT100~DT104 : " +0000123"  
 DT2~DT3 : U 4294967240 (H FFFF FFC8) → DT105~DT109 : " +0123456"

例11) 演算単位が32ビット(SF)の場合

[i]...SF

[S1]..."%#8.0f" : 32ビット単精度実数データを浮動小数点アスキーデータ(8桁)へ変換

[S2]...DT0

[N]...H 00020101 : ①アスキーデータ(逆方向)

[D]...DT100 : ②格納開始位置(1) → [D]+1バイト

③変換データ数(2) → [S2], [S2]+4 を変換

③変換データ数 = 2

[S2]	DT0	DT4	DT8
	123.45678	-12.3456789	

②格納開始位置 = +1バイト

[D]	(文字)
DT100	H **20 **
DT101	H 2020
DT102	H 2031 1
DT103	H 3233 2 3
DT104	H 2E20 .
DT105	H 2020
DT106	H 202D -
DT107	H 3132 1 2
DT108	H 2E20 .
DT109	

DT0~DT3 : 123.45678	→	DT100~DT104 : " 123."
DT4~DT7 : -12.3456789	→	DT104~DT108 : " -12."

例12) 演算単位が32ビット(SF)の場合

[i]...SF

[S1]..."%-10.2e" : 32ビット単精度実数データを指数表示アスキーデータ(10桁)へ変換

[S2]...DT0

[N]...H 00020001 : ①アスキーデータ(逆方向)

[D]...DT100 : ②格納開始位置(0) → [D]+0バイト

③変換データ数(2) → [S2], [S2]+4 を変換

③変換データ数 = 2

[S2]	DT0	DT4	DT8
	123.45678	-12.3456789	

②格納開始位置 = +0バイト

[D]	(文字)
DT100	H 312E 1 .
DT101	H 3233 2 3
DT102	H 652B E +
DT103	H 3220 2
DT104	H 2020
DT105	H 2D31 - 1
DT106	H 2E32 . 2
DT107	H 3365 3 e
DT108	H 2B31 + 1
DT109	H 2020
DT110	

DT0~DT3 : 123.45678	→	DT100~DT104 : "1.23e+2 "
DT4~DT7 : -12.3456789	→	DT105~DT109 : "-1.23e+1 "



例13) 演算単位が32ビット(SF)の場合

[I]...SF

[S1]..."%#9.7g" : 32ビット単精度実数データを指数表示アスキーデータ(9桁)または、浮動小数点アスキーデータへ変換(表現できる短い方の書式)

[S2]...DT0

[N]...H 00020001 : ①アスキーデータ(逆方向)  
 [D]...DT100 : ②格納開始位置(0) → [D]+0バイト  
 ③変換データ数(2) → [S2], [S2]+4 を変換

③変換データ数 = 2

[S2]	[S2]
DT0	123.45678
[S2]+4	-12.3456789
DT8	

②格納開始位置 = +0バイト

[D]	(文字)
DT100	H **20 **
DT101	H 3233 1 2
DT102	H 332E 3 .
DT103	H 3435 4 5
DT104	H 3030 0 0
DT105	H 2D31 - 1
DT106	H 3233 2 3
DT107	H 342E 4 .
DT108	H 3536 5 6
DT109	H 37** 7 **
DT110	

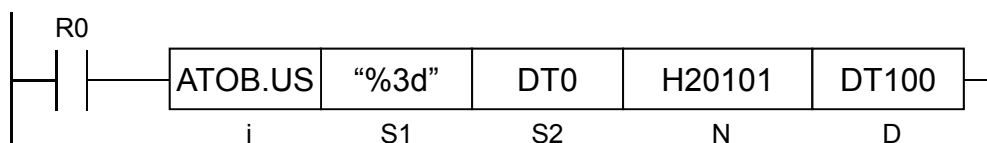
DT0~DT3 : 123.45678 → DT100~DT104 : " 123.4500"  
 DT4~DT7 : -12.3456789 → DT105~DT109 : "-1234.567"

■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	[S1]で指定した変換フォーマットが指定可能な演算単位でない場合にセットする
	[S1]で指定した変換フォーマットが制御文字で無い場合にセットする
	[N]で指定した ASCII コード桁数が、[S1]で指定した制御文字の最大桁数を超過しているときセットします
	[N]で指定した格納開始位置が範囲外るときセットします
	[N]で指定した変換データ数が[S2]のエリアを超えるときセットします
	[N]で指定した変換データ数が、範囲外るときセットします
	変換結果が[N]で指定した ASCII コード格納エリアの大きさを超過するときセットします
変換結果が[D]で指定したエリアを超える場合セットします	

# ATOB (ASCII→BIN変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	制御文字列(2から16文字)
S2	ASCIIコードを格納しているエリアの先頭番号
N	変換方法
D	変換結果のバイナリデータを格納するエリアの先頭番号

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス 修飾 *1			
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		""		
S1	●	●	●	●			●	●													●		
S2	●	●	●	●	●	●	●	●															●
N(*2)	●	●	●	●			●	●								●	●						●
D	●	●	●	●			●	●															●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*2: 演算単位にかかわらず32ビット整数(UL)として扱

## ■ 動作説明

- 指定の ASCII コード文字列を指定のデータコードに変換します。
- [S1] で指定した制御文字に従い、[S2]で指定したエリアに格納されている ASCII コードを[N]の変換方法により、バイナリデータに変換します。
- 演算結果は、[D]を先頭とするエリアに格納します。

## ■ 制御文字列の指定 [S1]

制御文字列	変換データの形式	指定可能な演算単位	データ範囲
"%nd"	10 進 ASCII データを 16 ビットデータへ変換	US	U0 ~ U65,535
	10 進 ASCII データを 32 ビットデータへ変換	SS	K-32,768 ~ K32,767
	16 進 ASCII データを 16 ビットデータへ変換	UL	U0 ~ U4,294,967,000
	16 進 ASCII データを 32 ビットデータへ変換	SL	K-2,147,483,000 ~ K2,147,483,000
"%nx"	16 進 ASCII データを 16 ビットデータへ変換 (順/逆方向)	US	H0 ~ HFFFF
	16 進 ASCII データを 32 ビットデータへ変換 (順/逆方向)	UL	H0 ~ HFFFFFFFF
"%nb"	10 進 ASCII データを BCD データへ変換 (順/逆方向)	US	H0 ~ H9999
	10 進 ASCII データを BCD データへ変換 (順/逆方向)	UL	H0 ~ H99999999
"%nf"	浮動小数点 ASCII データを 32 ビット単精度実数データへ変換	SF	3.402823E+38 ~ 2.802597E-45 -2.802597E-45 ~ -3.402823E+38
		DF	1.797693134862231E+308 ~ 4.940656458412465E-324 1.797693134862231E+308 ~ 4.940656458412465E-324
	浮動小数点 ASCII データを 64 ビット倍精度実数データへ変換	SF	3.402823E+38 ~ 2.802597E-45 -2.802597E-45 ~ -3.402823E+38
		DF	1.797693134862231E+308 ~ 4.940656458412465E-324 1.797693134862231E+308 ~ 4.940656458412465E-324

## ■ ASCIIデータの形式 [S2]

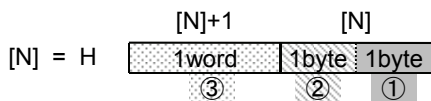
形式	データ例
整数 (10 進)	"1234567890",
	" 1234567", "1234567 ",
整数 (16 進)	"123456789ABCDEF", "123456789abcdef",
	" 123456789", "123456789 ",
	"0000000123456789",
浮動小数点	"1234.56789",
	" 1234.567", "1234.567 ",
指数表示	"1.234E+12"
	" 1.23E+12", "1.23E+12 ",
	"1.234e+12",

(注 1): ASCII データがカンマ(",")で区切られている場合は、カンマの所でデータを区切る。

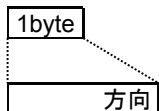
(注 2): ASCII データがカンマ(",")で終了している場合は、カンマの所で読み取り終了する。

## ■ 変換方法の指定 [N]

- 変換方法[N]は、32ビットエリアに Hex 形式で指定します。



### ① : アスキーデータの方向



方向:「0」=「順方向」、 「1」=「逆方向」

下記の方向を制約とします

制御文字列	方向
"%nd"	逆方向のみ
"%nx"	順・逆方向
"%nb"	順・逆方向
"%nf"	逆方向のみ
"%ne"	逆方向のみ

### ② : 変換開始位置

[S2]の下位バイトを「0」として指定

指定範囲:「0」~アスキーデータ桁数の最大値以内

例)変換開始位置(1)の場合

[S2]	0byte	**	←[S2]のHighアドレスから格納される
	1byte	H 31	
[S2] +1	2byte	H 32	
	3byte	H 33	
[S2] +2	4byte	.	

### ③ : 変換データ数

変換データ数を指定

指定範囲:「1~65535」

## ■ ASCIIデータの形式

形式	データ例
整数(10進)	"1234567890",
	" 1234567", "1234567 ",
整数(16進)	"123456789ABCDEF", "123456789abcdef",
	" 123456789", "123456789 ",
	"0000000123456789",
浮動小数点	"1234.56789",
	" 1234.567", "1234.567 ",
指数表示	"1.234E+12"
	" 1.23E+12", "1.23E+12 ",
	"1.234e+12",

(注 1):ASCII データがカンマ(",")で区切られている場合は、カンマの所でデータを区切る。

(注 2):ASCII データがカンマ(",")で終了している場合は、カンマの所で読み取り終了する。

## ■ 変換例

例1) 10進アスキーデータ→バイナリ変換

[I]...US

[S1]..."%d," : 10進アスキーデータを16ビットデータへ変換(データの終端が','の場合)

[S2]...DT0

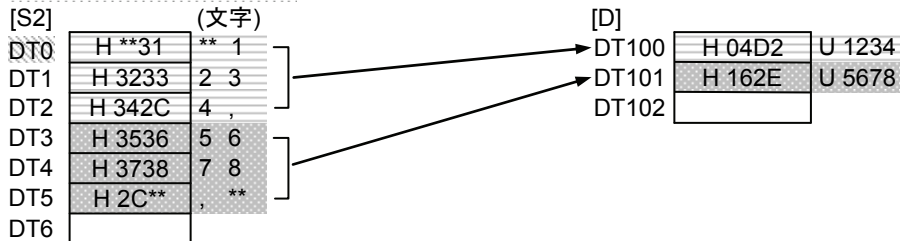
[N]...H 00020101 : ①アスキーデータ(逆方向)

[D]...DT100 : ②格納開始位置(1) → [S2]+1バイト

③変換データ数(2) → ; 2つまで読み込み

②格納開始位置 = +1バイト

③変換データ数 = 2



例2) 16進アスキーデータ→バイナリ変換

[I]...US

[S1]..."%x" : 16進アスキーデータを16ビットデータへ変換(データの終端が','の場合)

[S2]...DT0

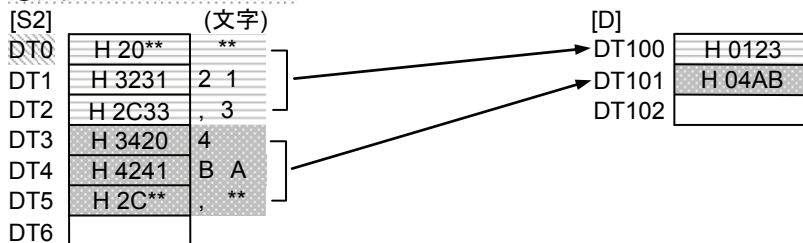
[N]...H 00020101 : ①アスキーデータ(逆方向)

[D]...DT100 : ②格納開始位置(1) → [S2]+1バイト

③変換データ数(2) → ; 2個まで読み込む

②格納開始位置 = +1バイト

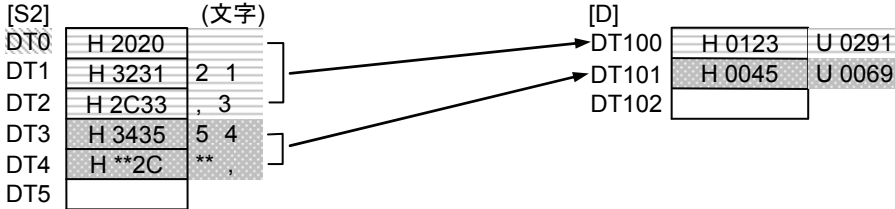
③変換データ数 = 2



例3) 10進アスキーデータ→BCDデータ変換

[i]...US  
 [S1]..."%b" : 10進アスキーデータをBCDデータへ変換 (データの終端が','の場合)  
 [S2]...DT0  
 [N]...H 00020001 : ①アスキーデータ(逆方向)  
 [D]...DT100 : ②格納開始位置(0) → [S2]+0バイト  
 : ③変換データ数(2) → ',' 2個まで読み込む

②格納開始位置 = +0バイト  
 ③変換データ数 = 2

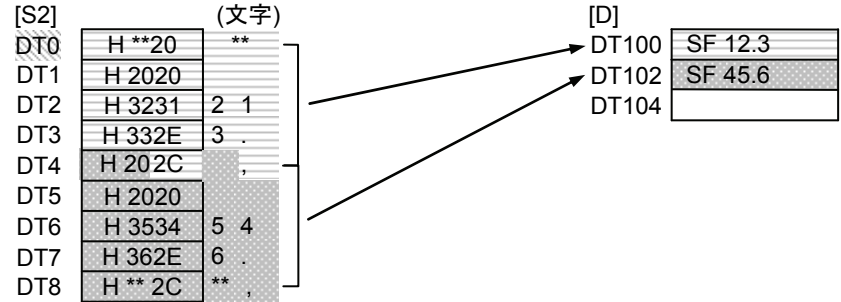


DT0~DT2 : " 123," → DT100 : U 0291 (H 0123)  
 DT3~DT5 : "45," → DT101 : U 0069 (H 0045)

例4) 浮動小数点アスキーデータ→実数データ変換

[i]...SF  
 [S1]..."%f" : 浮動小数点アスキーデータを32ビット実数データへ変換 (データの終端が','の場合)  
 [S2]...DT0  
 [N]...H 00020101 : ①アスキーデータ(逆方向)  
 [D]...DT100 : ②格納開始位置(1) → [S2]+1バイト  
 : ③変換データ数(2) → ',' 2個まで読み込む

②格納開始位置 = +1バイト  
 ③変換データ数 = 2



DT0~DT4 : " 12.3," → DT100~DT101 : SF 12.3  
 DT4~DT8 : " 45.6," → DT102~DT103 : SF 45.6

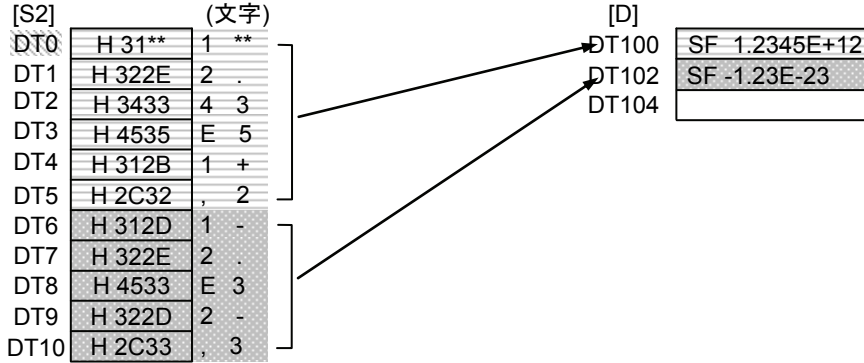
例5) 演算単位が32ビット(SF)の場合

[I]...SF  
 [S1]..."%e" : 指数表示アスキーデータを32ビット単精度実数データへ変換  
 (データの終端が','の場合)

[S2]...DT0

[N]...H 00020101 : ①アスキーデータ(逆方向)  
 [D]...DT100 : ②格納開始位置(1) → [S2]+1バイト  
 ③変換データ数(2) → ',' 2個まで読み込む

②格納開始位置 = +1バイト  
 ③変換データ数 = 2



DT0~DT5 : "1.2345E+12,"	→	DT100~DT101 : SF 1.2345E+12
DT6~DT10 : "-1.23E-23,"	→	DT102~DT103 : SF -1.23E-23

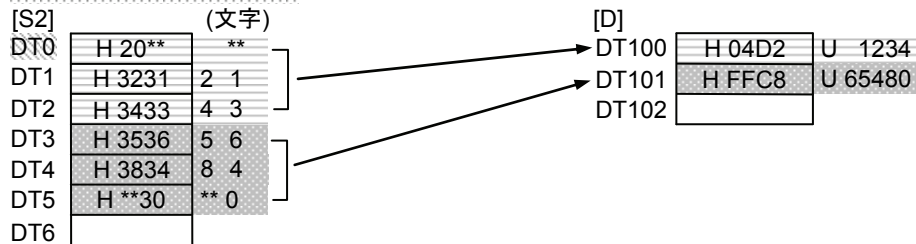
例6) 演算単位が16ビット(US)の場合

[I]...US  
 [S1]..."%5d" : 10進アスキーデータ(5桁)を16ビットデータへ変換

[S2]...DT0

[N]...H 00020101 : ①アスキーデータ(逆方向)  
 [D]...DT100 : ②格納開始位置(1) → [S2]+1バイト  
 ③変換データ数(2) → DT0.H~DT2.HとDT3.L~DT5.L を変換

②格納開始位置 = +1バイト  
 ③変換データ数 = 2



DT0~DT2 : " 1234"	→	DT100 : U 1234 (H 04D2)
DT3~DT5 : "65480"	→	DT101 : U 65480 (H FFC8)

例7) 演算単位が16ビット(SS)の場合

[i]...SS

[S1]..."%5d" : 10進アスキーデータ(5桁)を16ビットデータへ変換

[S2]...DT0

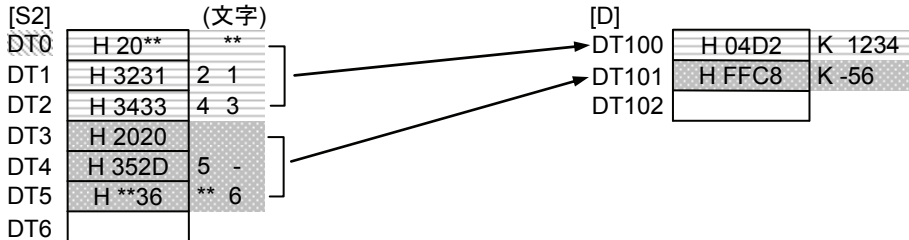
[N]...H 00020101 : ①アスキーデータ(逆方向)

[D]...DT100 : ②格納開始位置(1) → [S2]+1バイト

③変換データ数(2) → DT0.H~DT2.HとDT3.L~DT5.L を変換

②格納開始位置 = +1バイト

③変換データ数 = 2



DT0~DT2 : " 1234" → DT100 : K 1234 (H 04D2)  
 DT3~DT5 : "-56" → DT101 : K -56 (H FFC8)

例8) 演算単位が32ビット(UL)の場合

[i]...UL

[S1]..."%10d" : 10進アスキーデータ(10桁)を32ビットデータへ変換

[S2]...DT0

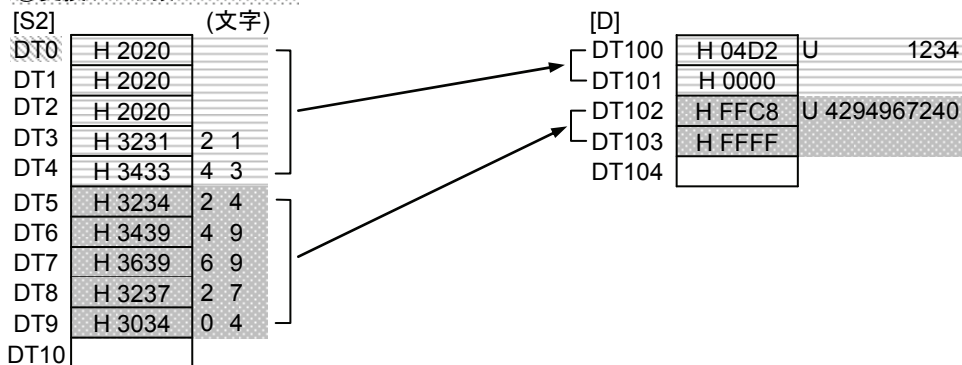
[N]...H 00020001 : ①アスキーデータ(逆方向)

[D]...DT100 : ②格納開始位置(0) → [S2]+0バイト

③変換データ数(2) → DT0.L~DT4.HとDT5.L~DT9.H を変換

②格納開始位置 = +0バイト

③変換データ数 = 2



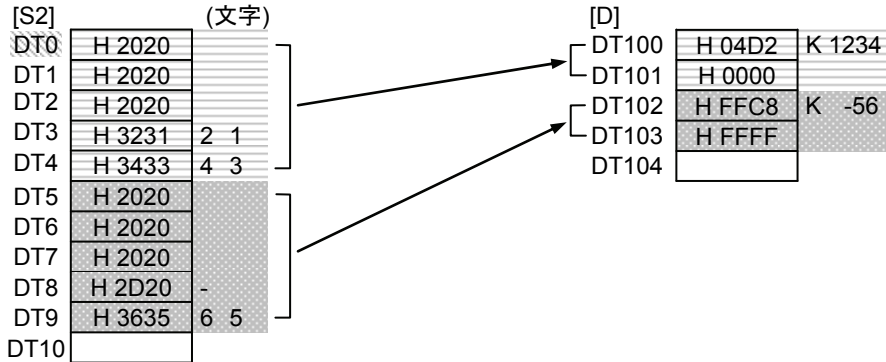
DT0~DT4 : " 1234" → DT100~DT101 : U 1234 (H 0000 04D2)  
 DT5~DT9 : "4294967240" → DT102~DT103 : U 4294967240 (H FFFF FFC8)



例9) 演算単位が32ビット(SL)の場合

- [I]...SL
- [S1]..."%10d" : 10進アスキーデータ(10桁)を32ビットデータへ変換
- [S2]...DT0
- [N]...H 00020001 : ①アスキーデータ(逆方向)
- [D]...DT100 : ②格納開始位置(0) → [S2]+0バイト
- ③変換データ数(2) → DT0.L~DT4.HとDT5.L~DT9.H を変換

- ②格納開始位置 = +0バイト
- ③変換データ数 = 2

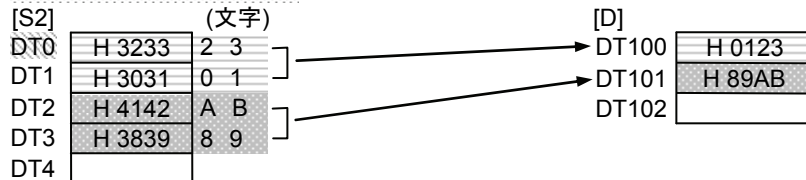


DT0~DT4 : " 1234" → DT100~DT101 : K 1234
DT5~DT9 : " -56" → DT102~DT103 : K -56

例10) 演算単位が16ビット(US)の場合

- [I]...US
- [S1]..."%4x" : 16進アスキーデータ(4桁)を16ビットデータへ変換
- [S2]...DT0
- [N]...H 00020000 : ①アスキーデータ(順方向)
- [D]...DT100 : ②格納開始位置(0) → [S2]+0バイト
- ③変換データ数(2) → DT0.L~DT1.HとDT2.L~DT3.H を変換

- ②格納開始位置 = +0バイト
- ③変換データ数 = 2

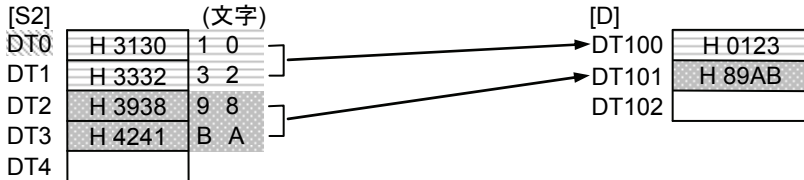


DT0~DT1 : "0123" → DT100 : H 0123
DT2~DT3 : "89AB" → DT101 : H 89AB

例11) 演算単位が16ビット(US)の場合

- [i]...US
- [S1]..."%4x" : 16進アスキーデータ(4桁)を16ビットデータへ変換
- [S2]...DT0
- [N]...H 00020001 : ①アスキーデータ(逆方向)
- [D]...DT100 : ②格納開始位置(0) → [S2]+0バイト  
③変換データ数(2) → DT0.L~DT1.HとDT2.L~DT3.H を変換

- ②格納開始位置 = +0バイト
- ③変換データ数 = 2

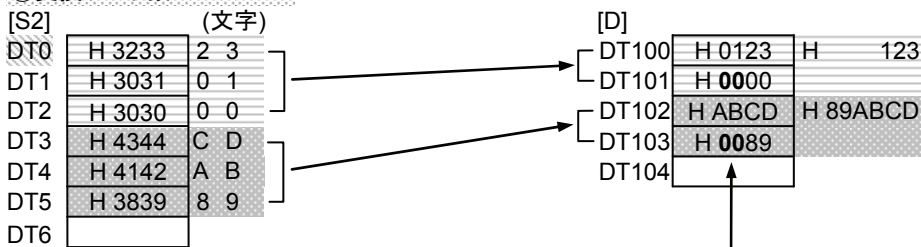


DT0~DT1 : "0123" → DT100 : H 0123  
 DT2~DT3 : "89AB" → DT101 : H 89AB

例12) 演算単位が32ビット(UL)の場合

- [i]...UL
- [S1]..."%6x" : 16進アスキーデータ(6桁)を32ビットデータへ変換
- [S2]...DT0
- [N]...H 00020000 : ①アスキーデータ(順方向)
- [D]...DT100 : ②格納開始位置(0) → [S2]+0バイト  
③変換データ数(2) → ; 2個まで読み込む

- ②格納開始位置 = +0バイト
- ③変換データ数 = 2



※文字数が変換ビットより小さい場合は、「0」で埋める

DT0~DT2 : "000123" → DT100 : H 0000 0123  
 DT3~DT5 : "89ABCD" → DT101 : H 0089 ABCD

例13) 演算単位が32ビット(UL)の場合

[i]...UL

[S1]... "%6x" : 16進アスキーデータ(6桁)を32ビットデータへ変換

[S2]...DT0

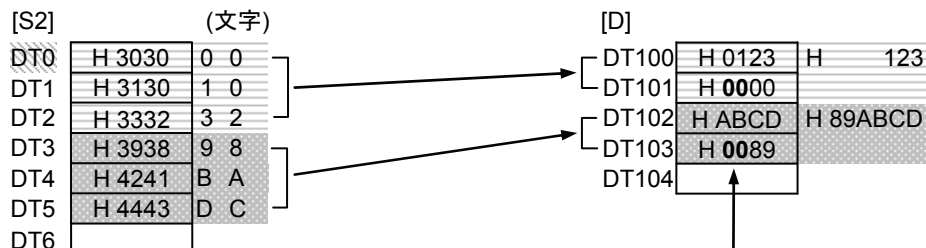
[N]...H 00020001 : ①アスキーデータ(逆方向)

[D]...DT100 : ②格納開始位置(0) → [S2]+0バイト

③変換データ数(2) → ; 2個まで読み込む

②格納開始位置 = +0バイト

③変換データ数 = 2



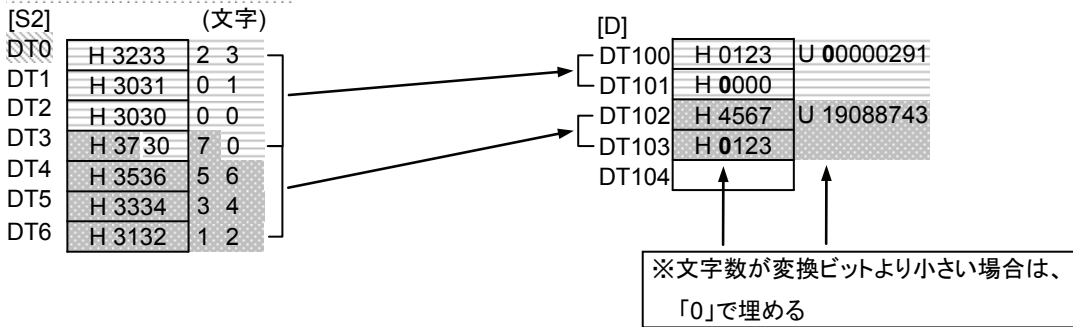
※文字数が変換ビットより小さい場合は、「0」で埋める

DT0~DT2 : "000123" → DT100 : H 0000 0123
DT3~DT5 : "89ABCD" → DT101 : H 0089 ABCD

例14) 演算単位が32ビット(UL)の場合

- [i]...UL
- [S1]..."%7b" : 10進アスキーデータ(7桁)を32ビットBCDデータへ変換
- [S2]...DT0
- [N]...H 00020000 : ①アスキーデータ(順方向)
- [D]...DT100 : ②格納開始位置(0) → [S2]+0バイト
- ③変換データ数(2) → DT0.L~DT3.L と DT3.H~DT6.H を変換

- ②格納開始位置 = +0バイト
- ③変換データ数 = 2

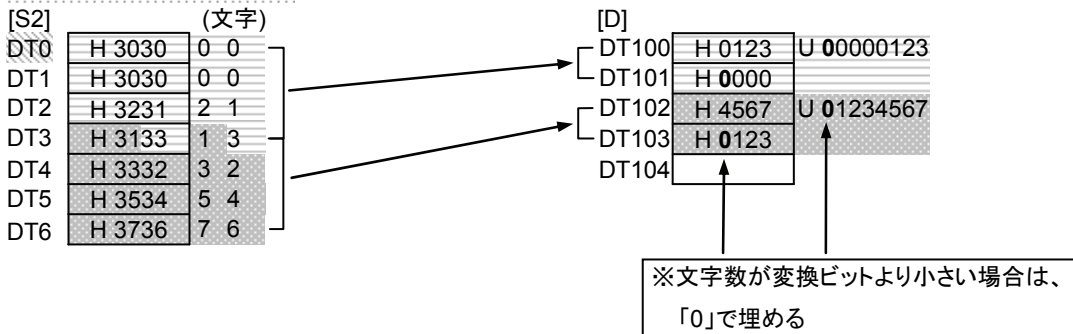


DT0~DT3 : "0000123" → DT100~DT101 : H 0000 0123  
 DT3~DT6 : "1234567" → DT102~DT103 : H 0123 4567

例15) 演算単位が32ビット(UL)の場合

- [i]...UL
- [S1]..."%7b" : 10進アスキーデータ(7桁)を32ビットBCDデータへ変換
- [S2]...DT0
- [N]...H 00020001 : ①アスキーデータ(逆方向)
- [D]...DT100 : ②格納開始位置(0) → [S2]+0バイト
- ③変換データ数(2) → DT0.L~DT3.L と DT3.H~DT6.H を変換

- ②格納開始位置 = +0バイト
- ③変換データ数 = 2



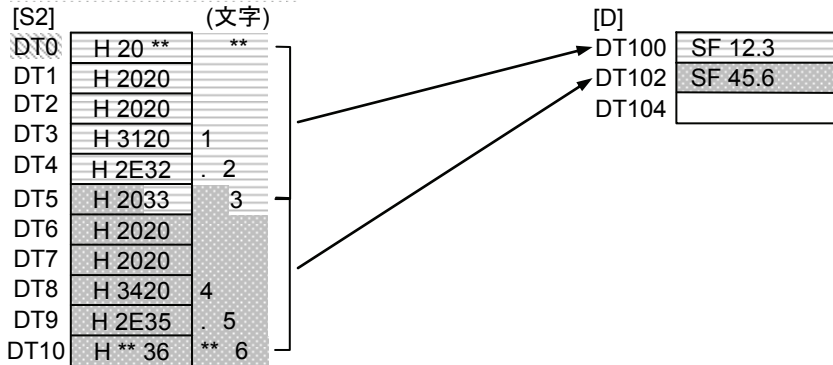
DT0~DT3 : "0000123" → DT100~DT101 : H 0000 0123  
 DT3~DT6 : "1234567" → DT102~DT103 : H 0123 4567

例16) 演算単位が32ビット(SF)の場合

[i]...SF  
 [S1]..."%10f" : 浮動小数点アスキーデータ(10文字)を32ビット実数データへ変換  
 [S2]...DT0  
 [N]...H 00020101 : ①アスキーデータ(逆方向)  
 [D]...DT100 : ②格納開始位置(1) → [S2]+1バイト  
 : ③変換データ数(2) → DT0.H~DT5.L と DT5.H ~ DT10.L を変換

②格納開始位置 = +1バイト

③変換データ数 = 2



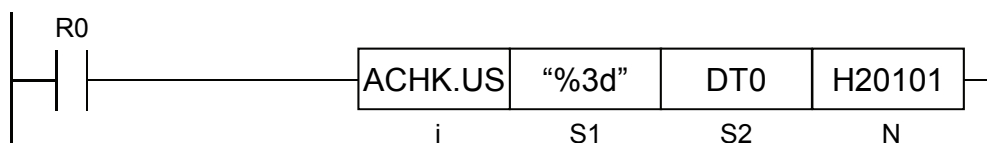
DT0~DT5 : " 12.3" → DT100~DT101 : SF 12.3
DT5~DT10 : " 45.6" → DT102~DT103 : SF 45.6

■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	[S1]で指定した変換フォーマットが指定可能な演算単位でない場合にセットする
	[S1]で指定した変換フォーマットが制御文字で無い場合にセットする
	[N]で指定した ASCII コード桁数が、[S1]で指定した制御文字の最大桁数を超過しているときセットします
	[N]で指定した格納開始位置が範囲外るときセットします
	[N]で指定した変換データ数が[S2]のエリアを超えるとときセットします
	[N]で指定した変換データ数が、範囲外るときセットします
	変換結果が[D]で指定したエリアを超える場合セットします

# ACHK (ASCIIデータチェック)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	制御文字列
S2	ASCII コードを格納しているエリアの先頭番号
N	変換方法

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデックス 修飾 *1		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		""	
S1	●	●	●	●			●	●													●	
S2	●	●	●	●	●	●	●	●														●
N(*2)	●	●	●	●			●	●								●	●					●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*2: 演算単位にかかわらず 32ビット整数(UL)として扱う

## ■ 動作説明

- 指定された ASCII コード文字列が ATOB 命令で変換できるかどうかをチェックします。
- [S1]で指定した制御文字に従い、[S2]で指定したエリアに格納されている ASCII コードを[N]で指定した変換方法により、正常に変換できるかをチェックします。
- チェックした結果が正常であれば、システムリレー SRB が ON(1),異常であれば OFF(0)になります。
- [S1]の制御文字列、[S2]の変換元の先頭、[N]の変換方法は、ATOB 命令と同じ値をセットしてください。
- ASC 最大文字列長、変換データ値の有効範囲等の動作は、ATOBと同様になっています。

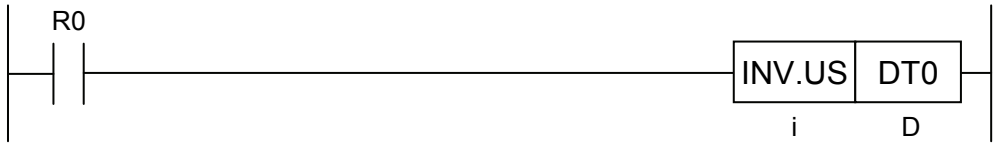
## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	[S1]で指定した変換フォーマットが指定可能な演算単位でない場合にセットする
	[S1]で指定した変換フォーマットが制御文字で無い場合にセットする
	[N]で指定した ASCII コード桁数が、[S1]で指定した制御文字の最大桁数を超過しているときセットします
	[N]で指定した格納開始位置が範囲外のときセットします
	[N]で指定した変換データ数が[S2]のエリアを超えるとときセットします
[N]で指定した変換データ数が、範囲外のときセットします	

■ MEMO

# INV (データ反転)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
D	反転するデータが格納されているデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス*1			整数			実数		文字	インデックス修飾*2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	""	
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [D]で指定したデバイスアドレスの値を論理反転します。

/[D] → [D]

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[D]...DT2

DT0	H 0011	DT0	H 0011
DT1	H 2233	DT1	H 2233
DT2	H 4455	DT2	H BBAA
DT3	H 6677	DT3	H 6677
DT4	H 8899	DT4	H 8899

例2) 演算単位が32ビット(UL,SL)の場合

[i]...UL,SL

[D]...TS1

TS0	H 11223344	TS0	H 11223344
TS1	H 55667788	TS1	H AA998877
TS2	H 9900AABB	TS2	H 9900AABB
TS3	H CCDDEEFF	TS3	H CCDDEEFF
TS4	H 12345678	TS4	H 12345678

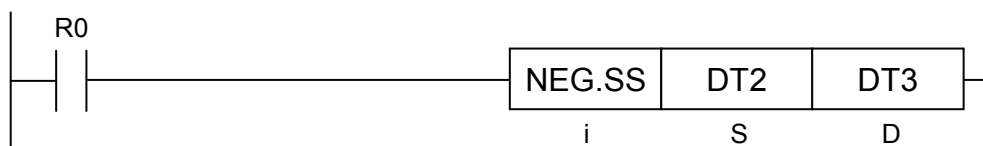
## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします



# NEG (符号反転)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i			●		●		

## ■ オペランド一覧

オペランド	説明
S	符号反転するデータの格納されているデバイスアドレスまたは定数
D	格納先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が16ビット整数(SS)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [S]で指定したデバイスアドレスの値または定数について、2の補数を求め、データの符号を反転します。
- 演算結果は、[D]で指定したデバイスアドレスに格納します。

## ■ 処理内容

例1) 演算単位が16ビット(SS)の場合

[i]...SS

[S]...DT2 [D]...DT3

DT0	K 100	DT0	K 100
DT1	K 110	DT1	K 110
DT2	<b>K 120</b>	DT2	K 120
DT3	K 130	DT3	<b>K -120</b>
DT4	K 140	DT4	K 140

例2) 演算単位が32ビット(SL)の場合

[i]...SL

[S]...TS1 [D]...TS4

TS0	K 500	TS0	K 500
TS1	<b>K 1000</b>	TS1	K 1000
TS2	K 1500	TS2	K 1500
TS3	K 2000	TS3	K 2000
TS4	K 2500	TS4	<b>K -1000</b>

## ■ プログラム上のご注意

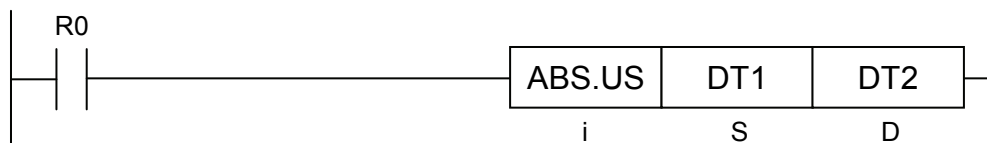
- 負の最小値を指定した場合、結果は負の最小値となります。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# ABS (絶対値)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S	絶対値を取るデータの格納されているデバイスアドレスまたは定数
D	格納先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス *1			整数			実数		文字	インデックス 修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H	SF	DF	""	
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

## ■ 動作説明

- [S]で指定したデバイスアドレスの値または定数の絶対値を取り、[D]で指定したデバイスアドレスに格納します。

## ■ 処理内容

例1) 演算単位が符号付き16ビット(SS)の場合

[i]...SS  
[S]...DT1 [D]...DT2

DT0	K-100	DT0	K-100
DT1	<b>K-110</b>	DT1	K-110
DT2	K-120	DT2	<b>K110</b>
DT3	K-130	DT3	K-130
DT4	K-140	DT4	K-140

例2) 演算単位が符号付き32ビット(SL)の場合

[i]...SL  
[S]...TS1 [D]...TS2

TS0	K-500	TS0	K-500
TS1	<b>K-1000</b>	TS1	K-1000
TS2	K-1500	TS2	<b>K1000</b>
TS3	K-2000	TS3	K-2000
TS4	K-2500	TS4	K-2500

## ■ プログラム上のご注意

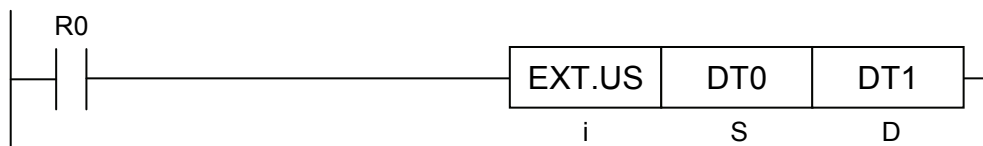
- 符号なし整数(US、UL)を指定した場合は同じ値が格納先に格納されます。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	
	[S]に負の最小値を指定した場合にセットします

# EXT (符号拡張)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
S	符号拡張を行うデータの格納されているデバイスアドレスまたは定数
D	格納先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス*1			整数			実数		文字	インデックス修飾*2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H	SF	DF		" "
S	●	●	●	●	●	●	●	●	●	●	●				●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が符号付き整数(SS)の場合のみ指定可能

\*4: 演算単位が符号なし整数(US)の場合のみ指定可能

## ■ 動作説明

- [S]で指定したデバイスアドレスの値または定数の符号拡張を行い、[D]で指定したデバイスアドレスに格納します。

## ■ 処理内容

例1) 演算単位が符号付き16ビット(SS)の場合

[i]...SS

[S]...DT0 [D]...DT0

DT0  $\boxed{\text{K -2(H FFFE)}}$  → DT0·DT1  $\boxed{\text{K-2(H FFFFFFFE)}}$   
 DT1  $\boxed{\text{K 0(H 0000)}}$

例2) 演算単位が符号なし16ビット(US)の場合

[i]...US

[S]...DT0 [D]...DT0

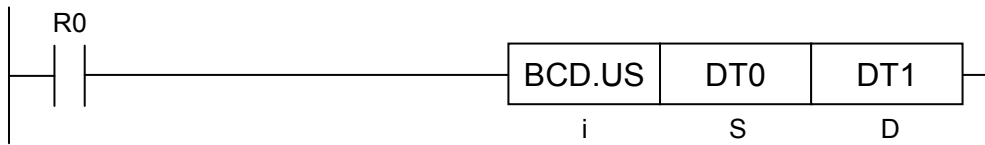
DT0  $\boxed{\text{H FFFE}}$  → DT0·DT1  $\boxed{\text{H 0000FFFFE}}$   
 DT1  $\boxed{\text{H 1234}}$

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# BCD (BCDデータ変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
S	変換するバイナリデータを格納しているデバイスアドレスまたは定数
D	変換結果を格納するデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデックス 修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		" "
S	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

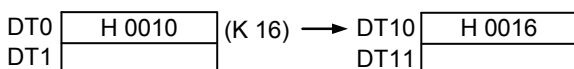
- [S]で指定したデバイスアドレスの値または定数をバイナリ→BCD変換し、[D]で指定したデバイスに格納します。

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[i]...US

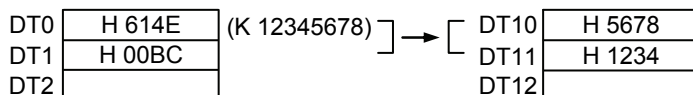
[S]...DT0 [D]...DT10



例2) 演算単位が32ビット(UL)の場合

[i]...UL

[S]...DT0 [D]...DT10



## ■ プログラム上のご注意

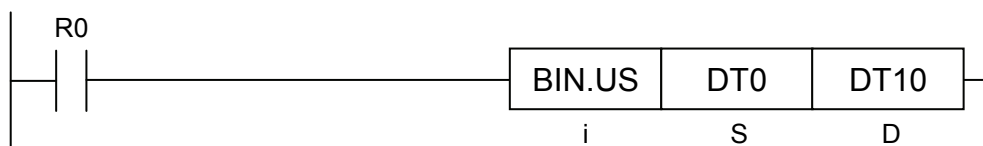
- 符号なし整数(US、UL)を指定した場合は同じ値が格納先に格納されます。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	バイナリデータがBCD変換できる範囲を超えているときセットします (USの場合:K 9999、ULの場合:K 99999999を超える場合など)

# BIN (BCD→BIN変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
S	変換するBCDデータを格納しているデバイスアドレスまたは定数
D	変換結果を格納するデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		""
S	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

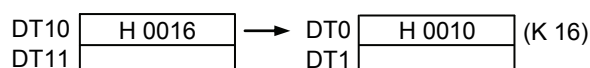
## ■ 動作説明

- [S]で指定したデバイスアドレスの値または定数をBCD→バイナリ変換し、[D]で指定したデバイスに格納します。

## ■ 処理内容

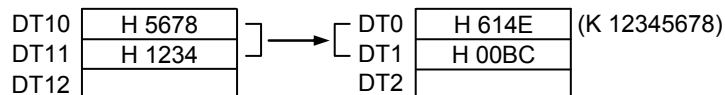
例1) 演算単位が16ビット(US)の場合

[i]...US  
[S]...DT0 [D]...DT10



例2) 演算単位が32ビット(UL)の場合

[i]...UL  
[S]...DT0 [D]...DT10



## ■ プログラム上のご注意

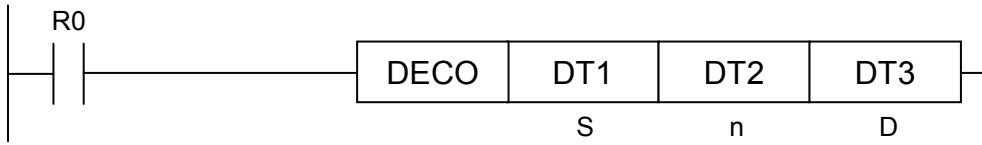
- 符号なし整数(US、UL)を指定した場合は同じ値が格納先に格納されます。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	[S]がBCDデータでないときセットします

# DECO (デコード)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S	デコードを行うデータの格納されているデバイスアドレスまたは定数
N	制御データ(変換開始ビットの指定、変換有効ビット長の指定)の格納されているデバイスアドレスまたは定数
D	格納先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

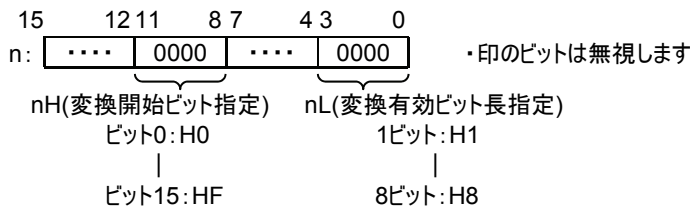
オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデックス 修飾 *1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""	
S	●	●	●	●	●	●	●	●	●	●	●				●	●	●				●
n	●	●	●	●	●	●	●	●	●	●	●						●				●
D	●	●	●	●			●	●	●		●										●

\*1: 16ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- [S]で指定したデバイスアドレスの値または定数についてデコードし、デコード結果を[D]で指定したデバイスアドレスに格納します。
- デコードの対象部分は、[n]の制御データで指定します。
- デコード結果を格納するのに必要なデバイスアドレスの長さは、デコード対象の長さによって変わります。

## ■ 制御データ[N]の指定



## ■ 変換例

変換データ	デコード結果(16ビット)
0000	0000 0000 0000 0001
0001	0000 0000 0000 0010
0010	0000 0000 0000 0100
0011	0000 0000 0000 1000
0100	0000 0000 0001 0000
0101	0000 0000 0010 0000
0110	0000 0000 0100 0000
0111	0000 0000 1000 0000
1000	0000 0001 0000 0000
1001	0000 0010 0000 0000
1010	0000 0100 0000 0000
1011	0000 1000 0000 0000
1100	0001 0000 0000 0000
1101	0010 0000 0000 0000
1110	0100 0000 0000 0000
1111	1000 0000 0000 0000

## ■ nLの指定と演算結果の長さ

nLの値 変換有効ビット長	デコード結果の 占有長	デコード結果の 有効ビット長	[D]内の 有効ビット長以外の値
1	1ワード	2	0
2	1ワード	4	0
3	1ワード	8(1バイト)	0
4	1ワード	16(1バイト)	-
5	2ワード	32(2バイト)	-
6	4ワード	64(4バイト)	-
7	8ワード	128(8バイト)	-
8	16ワード	256(16バイト)	-

## ■ 処理内容

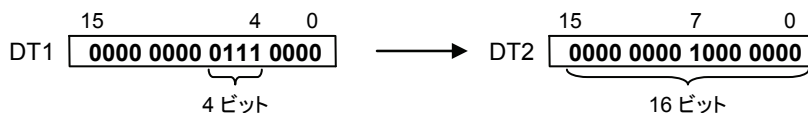
例) ビット4から4ビットデコードする場合

[S]...DT1 [n]...H 0404 [D]...DT2

※ 指定部分("0111"=7)をデコードした結果をDT2を先頭とする

16ビット(2<sup>4</sup>ビット)のデバイスアドレスに格納します

※ DT2を先頭とする16ビットのエリアのビット7がONになり、それ以外のビットは0になります

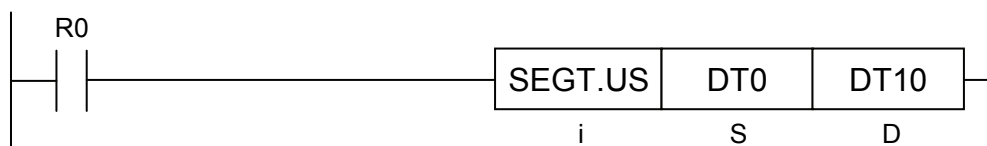


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	変換有効ビット長(nL)が、 $1 \leq nL \leq 8$ でない場合セットします
	変換開始ビット No.(nH)と変換有効ビット長(nL)を足した値が、 $1 \leq nH + nL \leq 16$ でない場合セットします
	デコード結果を[D]で指定されたデバイスアドレスへ格納するとエリアを超える場合セットします

# SEGT (7 セグメントデコード)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●					

## ■ オペランド一覧

オペランド	説明
S	デコードを行うデータの格納されているデバイスアドレスまたは定数
D	格納先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデック ス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		" "
S	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

\*1: 16ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- [S]で指定したデバイスアドレスの値または定数を4桁分の7セグメント表示用データに変換し、[D]を先頭とする2ワードのデバイスアドレスに格納します。

## ■ 処理内容

例) [S]にH ABCDが格納されている場合

[S]...DT0 [D]...DT10

DT0	H ABCD	→	DT10	0011	1001	0101	1110
			DT11	0111	0111	0111	1100

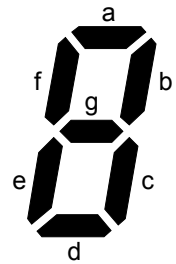
## ■ プログラム上のご注意

- [S]に符号なし定数 U を指定した場合、HEX データとして変換します。



### ■ 表示内容とデータ

数値	変換データ1桁分 [S]				7セグメント表示用データ1桁分 [D]							7セグメント表示	
					g	f	e	d	c	b	a		
0	0	0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	0	0	0	0	1	1	0	1
2	0	0	1	0	0	1	0	1	1	0	1	1	2
3	0	0	1	1	0	1	0	0	1	1	1	1	3
4	0	1	0	0	0	1	1	0	0	1	1	0	4
5	0	1	0	1	0	1	1	0	1	1	0	1	5
6	0	1	1	0	0	1	1	1	1	1	0	1	6
7	0	1	1	1	0	0	1	0	0	1	1	1	7
8	1	0	0	0	0	1	1	1	1	1	1	1	8
9	1	0	0	1	0	1	1	0	1	1	1	1	9
A	1	0	1	0	0	1	1	1	0	1	1	1	A
B	1	0	1	1	0	1	1	1	1	1	0	0	b
C	1	1	0	0	0	0	1	1	1	0	0	1	c
D	1	1	0	1	0	1	0	1	1	1	1	0	d
E	1	1	1	0	0	1	1	1	1	0	0	1	e
F	1	1	1	1	0	1	1	1	0	0	0	1	f

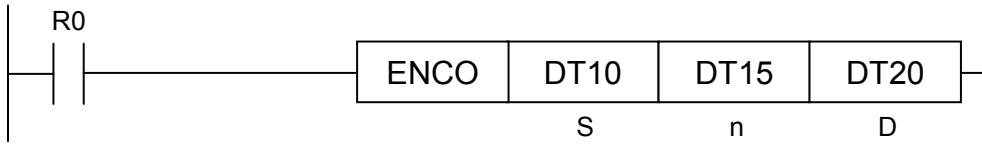


### ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	変換結果を[D]で指定されたデバイスアドレスへ格納するとエリアを超える場合にセットします

# ENCO (エンコード)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S	エンコードを行うデータの格納されているデバイスアドレスまたは定数
n	制御データ(結果出力開始ビットの指定、変換有効ビット長の指定)の格納されているデバイスアドレスまたは定数
D	格納先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

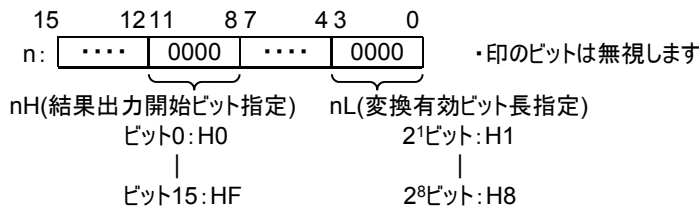
オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""		
S	●	●	●	●	●	●	●	●	●	●	●											●
n	●	●	●	●	●	●	●	●	●	●	●						●					●
D	●	●	●	●			●	●	●	●	●											●

\*1: 16ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- [S]で指定したデバイスアドレスの値または定数のある部分についてエンコードし、エンコード結果を[D]で指定したデバイスアドレスに格納します。
- エンコードの対象部分は、[n]の制御データで指定します。
- エンコードの対象部分で複数のビットが ON している場合は、上位のビットが有効です。
- [S]で指定したデバイスアドレスを先頭に 2nL ビット分の内容をエンコードします。エンコード結果は、nH で指定したビットから 8 ビット以内に 10 進数で格納します。
- [D]で指定したデバイスアドレスのうち、変換結果を格納する部分以外は 0 になります。

## ■ 制御データ[N]の指定



## ■ 変換例

変換データ(16ビット)	エンコード結果
0000 0000 0000 0001	0000
0000 0000 0000 0010	0001
0000 0000 0000 0100	0010
0000 0000 0000 1000	0011
0000 0000 0001 0000	0100
0000 0000 0010 0000	0101
0000 0000 0100 0000	0110
0000 0000 1000 0000	0111
0000 0001 0000 0000	1000
0000 0010 0000 0000	1001
0000 0100 0000 0000	1010
0000 1000 0000 0000	1011
0001 0000 0000 0000	1100
0010 0000 0000 0000	1101
0100 0000 0000 0000	1110
1000 0000 0000 0000	1111

## ■ nLの指定と結果の長さ

nL の値	変換有効ビット長	nL の値	変換有効ビット長
1	2	5	32(2 バイト)
2	4	6	64(4 バイト)
3	8(1 バイト)	7	128(8 バイト)
4	16(1 バイト)	8	256(16 バイト)

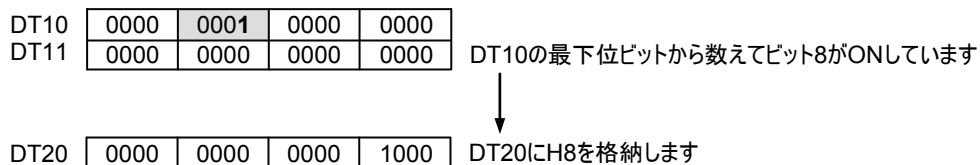
## ■ 処理内容

例) [S]...DT10 [n]...H 0005 [D]...DT20

DGT S, S1, n, D, D1

[S]で指定したエリアの[S1]桁目から[n]桁分、[D]で指定した16ビットデータの[D1]桁に転送します  
 転送開始は下位から4bitづつ第0桁目、第1桁目、第2桁目、第3桁目となります

変換有効ビットは、DT10から32ビット分のDT10～DT11です  
 この2ワードのエリアでONしているビットの番号を10進数でDT20のビット0から格納します



## ■ プログラム上のご注意

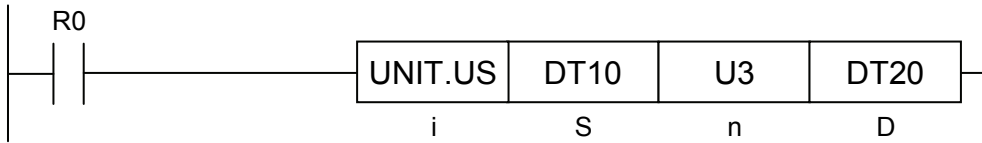
- [S]に符号なし定数 U を指定した場合、Hex データとして変換します。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	変換有効ビット長:nL が、 $1 \leq nL \leq 8$ でない場合にセットします
	結果出力開始ビット No.:nH と変換有効ビット長:nL を足した値が $1 \leq nH+nL \leq 16$ でない場合にセット(整合性)します
	エンコードするデータが、全て"0"の場合にセットします

# UNIT (デジット結合)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●					

## ■ オペランド一覧

オペランド	説明
S	結合するデータの格納されているデバイスアドレスまたは定数(データ形式:符号なし 16ビット整数)
n	結合データ数が格納されているデバイスアドレスまたは定数(データ形式:符号なし 16ビット整数)
D	格納先のデバイスアドレス(データ形式:演算単位に従う)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""		
S	●	●	●	●	●	●	●	●	●	●	●											●
n	●	●	●	●	●	●	●	●	●	●	●					●	●					●
D	●	●	●	●			●	●	●		●											●

\*1: 16ビットデバイス、整数定数のみ修飾可能(32ビットデバイス、実数定数、文字定数には指定不可)

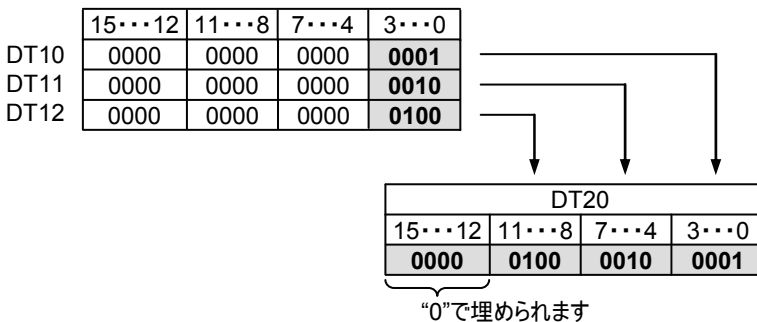
## ■ 動作説明

- [S]を先頭とする[n]ワードの 16 ビットデータの 下位 4 ビットを 16 ビットデータに結合します。
- 結合データ数[n]の指定可能範囲は 0~4 です。[n]=0 の場合、無処理とします。
- [D]の残りの部分には、"0"を格納します。

## ■ 処理内容

[i]...US

[S]...DT10 [n]...U3 [D]...DT20

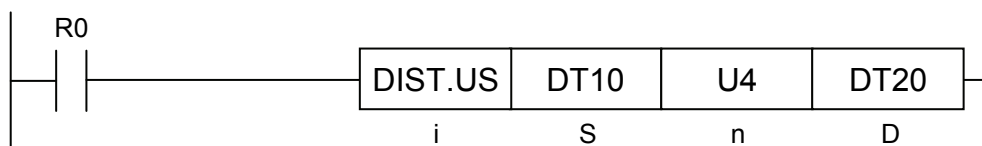


## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	結合するデータ数[n]が指定範囲外の場合にセットします

# DIST (デジット分離)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●					

## ■ オペランド一覧

オペランド	説明
S	分離するデータの格納されているデバイスアドレスまたは定数(データ形式:演算単位に従う)
n	分離する点数を格納しているデバイスアドレスまたは定数(データ形式:符号なし 16 ビット整数)
D	格納先のデバイスアドレス(データ形式:符号なし 16 ビット整数)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""	
S	●	●	●	●	●	●	●	●	●	●	●					●	●				
n	●	●	●	●	●	●	●	●	●	●	●					●	●				
D	●	●	●	●			●	●	●	●	●										

\*1: 16ビットデバイス、整数定数のみ修飾可能(32ビットデバイス、実数定数、文字定数には指定不可)

## ■ 動作説明

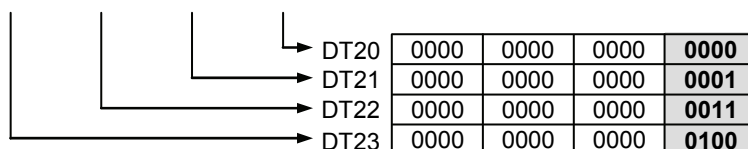
- [S]で指定された 16 ビットデバイスを 4 ビット毎に 16 ビットデータに分離する。(分離データ数:[n]の指定可能範囲は 0~4)
- [n]=0 の場合、無処理とします。

## ■ 処理内容

例) 演算単位が16ビット(US)の場合

[i]...US  
[S]...DT10 [n]...U4 [D]...DT20

DT10			
15...12	11...8	7...4	3...0
0100	0011	0001	0000

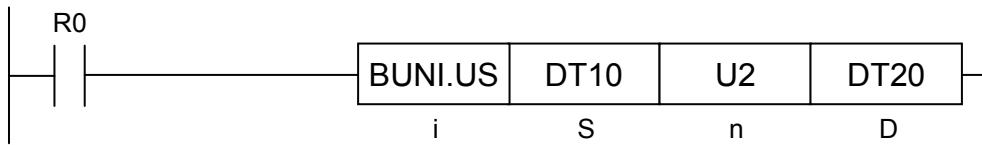


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	分離するデータ点数:[n]が指定範囲外の場合にセットします
	[D]で指定したアドレスから[n]個分データを転送すると、デバイスアドレスを超える場合にセットします

# BUNI (バイトデータ結合)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
S	結合するデータの格納されているデバイスアドレスまたは定数(データ形式:符号なし 16ビット整数)
n	結合データ数が格納されているデバイスアドレスまたは定数(データ形式:符号なし 16ビット整数)
D	格納先のデバイスアドレス(データ形式:演算単位に従う)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF	DF	""		
S	●	●	●	●	●	●	●	●	●	●	●											●
n	●	●	●	●	●	●	●	●	●	●	●					●	●					●
D	●	●	●	●			●	●	●		●											●

\*1: 16ビットデバイス、整数定数のみ修飾可能(32ビットデバイス、実数定数、文字定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

## ■ 動作説明

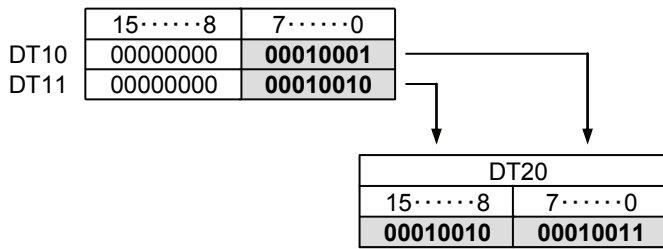
- 演算単位が 16ビット(US)の場合、[S]を先頭とする[n]ワードの 16ビットデータの 下位 1 バイトを 16ビットデータに結合します。(結合データ数:[n]の指定可能範囲は 0~2)
- 演算単位が 32ビット(UL)の場合、[S]を先頭とする[n]ワードの 16ビットデータの 下位 1 バイトを 32ビットデータに結合します。(結合データ数:[n]の指定可能範囲は 0~4)
- [n]=0 の場合、無処理とします。
- [D]の残りの部分は"0"で埋めます。

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[i]...US

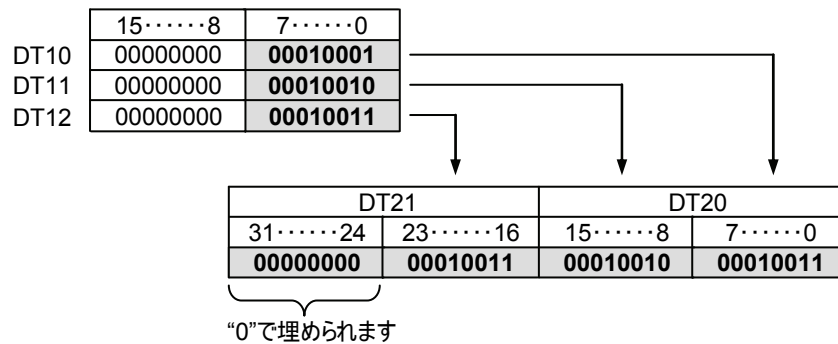
[S]...DT10 [n]...U2 [D]...DT20



例2) 演算単位が32ビット(UL)の場合

[i]...UL

[S]...DT10 [n]...U3 [D]...DT20

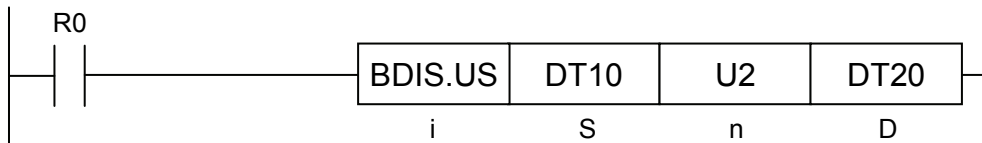


## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	結合するデータ数[n]が指定範囲外の場合にセットします

# BDIS (バイトデータ分離)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
S	分離するデータの格納されているデバイスアドレスまたは定数
n	分離する点数を格納しているデバイスアドレスまたは定数
D	格納先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾*1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●					●	●				●
n	●	●	●	●	●	●	●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

\*1: 16ビットデバイス、整数定数のみ修飾可能(32ビットデバイス、実数定数、文字定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

## ■ 動作説明

- 演算単位が 16 ビット(US)の場合、[S]で指定された 16 ビットデバイスを 1 バイト毎に 16 ビットデータに分離します。(分離データ数:[n]の指定可能範囲は 0~2)
- 演算単位が 32 ビット(UL)の場合、[S]で指定された 32 ビットデバイスを 1 バイト毎に 16 ビットデータに分離します。(分離データ数:[n]の指定可能範囲は 0~4)
- [n]=0 の場合、無処理とします。



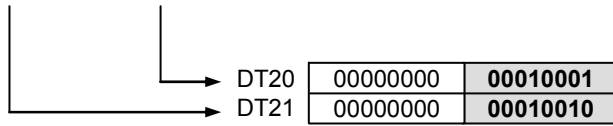
## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[i]...US

[S]...DT10 [n]...U2 [D]...DT20

DT10	
15.....8	7.....0
<b>00010010</b>	<b>00010001</b>

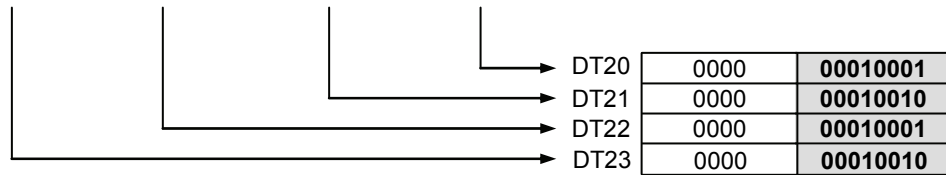


例2) 演算単位が32ビット(UL)の場合

[i]...UL

[S]...DT10 [n]...U4 [D]...DT20

DT11		DT10	
31.....24	23.....16	15.....8	7.....0
<b>00010010</b>	<b>00010011</b>	<b>00010010</b>	<b>00010011</b>

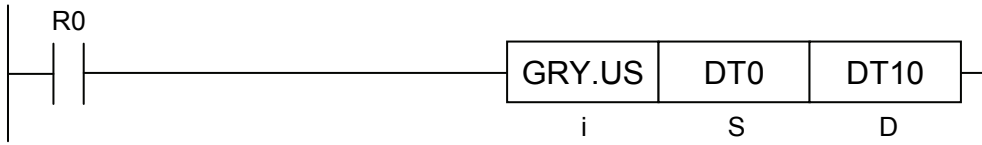


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	分離するデータ点数:[n]が指定範囲外の場合にセットします
	[D]で指定したアドレスから[n]個分データを転送すると、デバイスアドレスを超える場合にセットします

# GRY (バイナリ→グレイコード変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
S	変換を行うデータの格納されているデバイスアドレスまたは定数
D	格納先のデバイスアドレス

(注)グレイコードについては"BIN/グレイコード対応表"を参照願います

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス*1			整数			実数		文字	インデックス修飾*2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	" "		
	S	●	●	●	●	●	●	●	●	●	●	●					●	●				
D	●	●	●	●			●	●			●											●

\*1: 演算単位が16ビット整数(US)の場合は指定不可

\*2: 16ビットデバイス、整数定数のみ修飾可能(32ビットデバイス、実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [S]で指定したデバイスアドレスの値または定数をグレイコードに変換して、[D]で指定したデバイスアドレスに格納します。

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[i]...US

[S]...DT0 [D]...DT10

DT0 [0000000000011000] → DT10 [000000000010100]

例2) 演算単位が32ビット(UL)の場合

[i]...UL

[S]...DT0 [D]...DT10

DT0, DT1 [00000000000000000000000000001111]  
↓  
DT10, DT11 [00000000000000000000000000001000]

## ■ フラグ動作

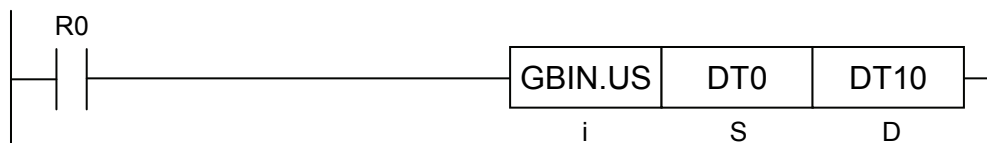
名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

■ DEC / BIN / Gray code 対応表

10進数 Decimal	2進数 Binary	グレイコード Gray code
0	0000 0000 0000 0000	0000 0000 0000 0000
1	0000 0000 0000 0001	0000 0000 0000 0001
2	0000 0000 0000 0010	0000 0000 0000 0011
3	0000 0000 0000 0011	0000 0000 0000 0010
4	0000 0000 0000 0100	0000 0000 0000 0110
5	0000 0000 0000 0101	0000 0000 0000 0111
6	0000 0000 0000 0110	0000 0000 0000 0101
7	0000 0000 0000 0111	0000 0000 0000 0100
8	0000 0000 0000 1000	0000 0000 0000 1100
9	0000 0000 0000 1001	0000 0000 0000 1101
10	0000 0000 0000 1010	0000 0000 0000 1111
11	0000 0000 0000 1011	0000 0000 0000 1110
12	0000 0000 0000 1100	0000 0000 0000 1010
13	0000 0000 0000 1101	0000 0000 0000 1011
14	0000 0000 0000 1110	0000 0000 0000 1001
15	0000 0000 0000 1111	0000 0000 0000 1000
16	0000 0000 0001 0000	0000 0000 0001 1000
17	0000 0000 0001 0001	0000 0000 0001 1001
18	0000 0000 0001 0010	0000 0000 0001 1011
19	0000 0000 0001 0011	0000 0000 0001 1010
20	0000 0000 0001 0100	0000 0000 0001 1110
21	0000 0000 0001 0101	0000 0000 0001 1111
22	0000 0000 0001 0110	0000 0000 0001 1101
23	0000 0000 0001 0111	0000 0000 0001 1100
24	0000 0000 0001 1000	0000 0000 0001 0100
25	0000 0000 0001 1001	0000 0000 0001 0101
26	0000 0000 0001 1010	0000 0000 0001 0111
27	0000 0000 0001 1011	0000 0000 0001 0110
28	0000 0000 0001 1100	0000 0000 0001 0010
29	0000 0000 0001 1101	0000 0000 0001 0011
30	0000 0000 0001 1110	0000 0000 0001 0001
31	0000 0000 0001 1111	0000 0000 0001 0000
32	0000 0000 0010 0000	0000 0000 0011 0000
-	-	-
63	0000 0000 0010 1111	0000 0000 0010 0000
64	0000 0000 0100 1111	0000 0000 0110 0000
-	-	-
255	0000 00001111 1111	0000 0000 1000 0000

# GBIN (グレイコード→BIN変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
S	変換を行うデータの格納されているデバイスアドレスまたは定数
D	格納先のデバイスアドレス

(注)グレイコードについては"BIN/グレイコード対応表"を参照願います

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2			
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	" "				
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●							●
D	●	●	●	●			●	●	●		●	●	●	●										●

\*1: 演算単位が 16 ビット整数(US)の場合は指定不可

\*2: 16 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ (I0~IE)

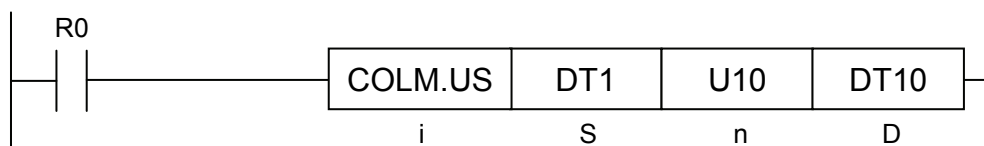
## ■ 動作説明

- [S]で指定したデバイスアドレスの値または定数のグレイコードをバイナリデータに変換して、[D]で指定したデバイスアドレスに格納します。



# COLM (ビット行→ビット列変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●					

## ■ オペランド一覧

オペランド	説明
S	変換を行うデータの格納されているデバイスアドレスまたは定数
n	ビット位置指定を格納しているデバイスアドレスまたは定数(データ指定可能範囲:0~15)
D	ビット列を書換えるデバイスの先頭アドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●				●
n	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●	●				●
D	●	●	●	●	●			●	●		●	●	●	●							●

\*1: 16ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

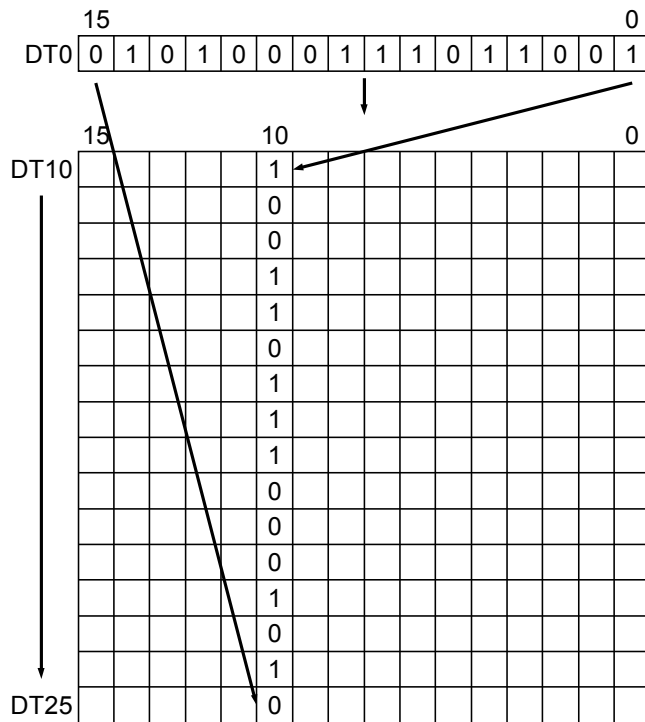
- [S]で指定された 16 ビット行データを、[D]で指定された 16 ワードのデバイス領域の[n]ビット列に転送します。
- 指定されたビット列以外は変更されません。

### ■ 処理内容

例) 演算単位が 16 ビット(US)の場合

[i]...US

[S]...DT1 [n]...U10 [D]...DT10

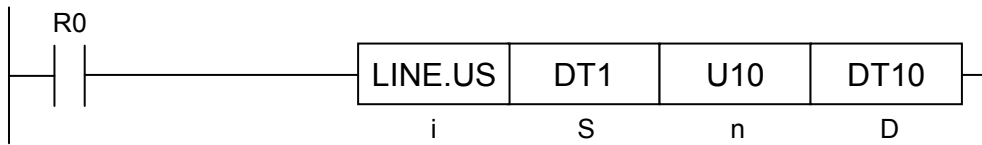


### ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	ビット位置指定:[n]が、 $0 \leq [n] \leq 15$ でない場合にセットします
	変換結果を[D]で指定されたデバイスアドレスに格納すると、エリアを超える場合にセットします

# LINE (ビット列→ビット行変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●					

## ■ オペランド一覧

オペランド	説明
S	ビット列を読み込むデバイスアドレスの先頭アドレス
n	ビット位置指定を格納しているデバイスアドレスまたは定数(データ指定可能範囲:0~15)
D	格納先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""		
S	●	●	●	●	●	●	●	●	●	●	●											●
n	●	●	●	●	●	●	●	●	●	●	●					●	●					●
D	●	●	●	●			●	●	●		●											●

\*1: 16ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- [S]で指定された 16 ワードのデバイス領域の[n]ビット列のデータを[D]で指定された 16 ビットデータに転送します。

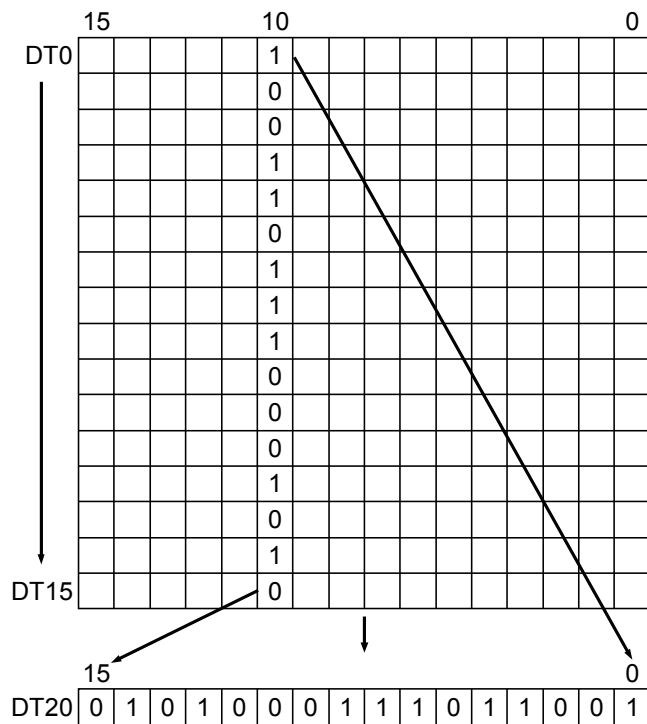


■ 処理内容

例) 演算単位が 16 ビット(US)の場合

[i]...US

[S]...DT1 [n]...U10 [D]...DT20

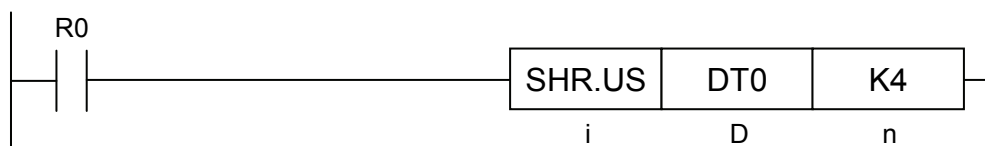


■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	ビット位置指定:[n]が、 $0 \leq [n] \leq 15$ でない場合にセットします
(ER)	[S]で指定された変換範囲がデバイスアドレスをを超える場合にセットします

# SHR (nビット右シフト)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
D	シフト対象のデータを格納しているデバイスアドレス
n	シフトビット数を格納しているデバイスアドレスまたは定数

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H	SF	DF	" "	
D	●	●	●	●			●	●	●		●	●	●								●
n	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●				●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

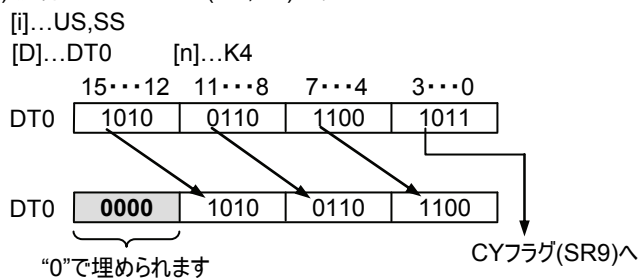
\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

## ■ 動作説明

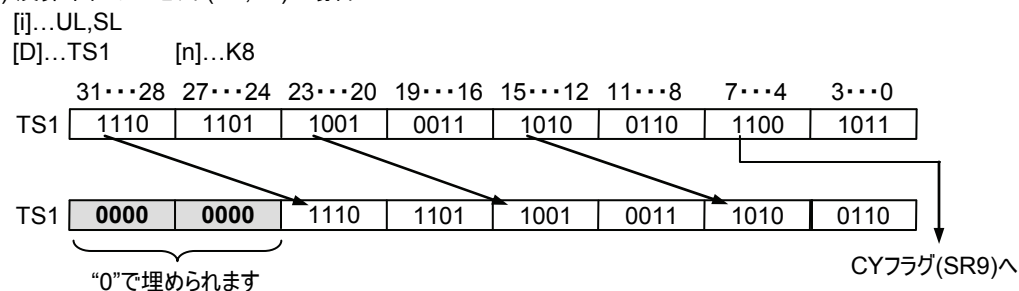
- [D]で指定されたデータを、[n](10進数で指定)で指定したビット分右(下位ビット方向)へシフトします。
- シフトを行うと、最上位ビットから[n]ビット分0で埋めます。また、最下位ビットから[n]ビット目の内容を、SR9(CY)に格納します。
- [n]はデータ中の下位 8 ビットのみ有効です。シフト量は、0 ビット~255 ビットの範囲で指定します。

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合



例2) 演算単位が32ビット(UL,SL)の場合

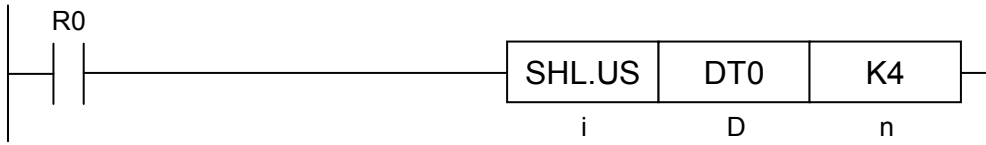


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR9(CY)	[n](シフトビット数)が演算単位より大きい場合はリセットし、 上記以外の場合は、最下位ビットから[n]ビット目の内容をセットします

# SHL (nビット左シフト)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
D	シフト対象のデータを格納しているデバイスアドレス
n	シフトビット数を格納しているデバイスアドレスまたは定数

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H	SF	DF		" "
D	●	●	●	●			●	●	●		●	●	●								●
n	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●				●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可

## ■ 動作説明

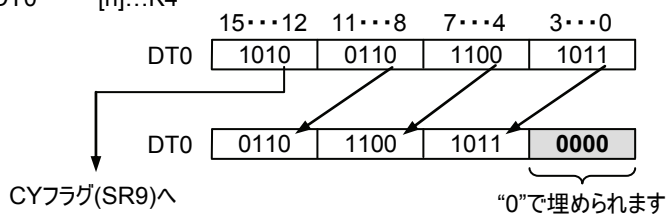
- [D]で指定されたデータを、[n](10進数で指定)で指定したビット分左(上位ビット方向)へシフトします。
- シフトを行うと、最下位ビットから[n]ビット分0で埋めます。また、最上位ビットから[n]ビット目の内容を、SR9(CY)に格納します。
- [n]はデータ中の下位 8 ビットのみ有効です。シフト量は、0 ビット~255 ビットの範囲で指定します。

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

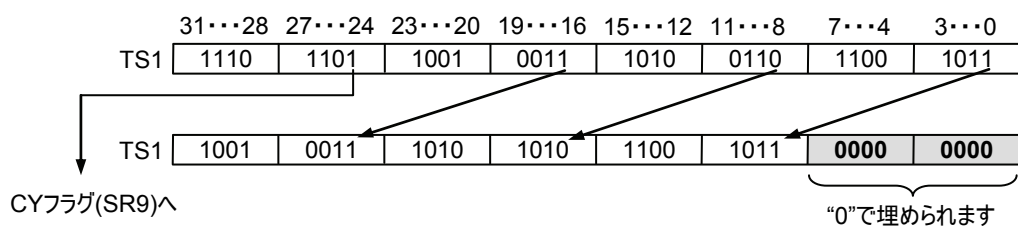
[D]...DT0 [n]...K4



例2) 演算単位が32ビット(UL,SL)の場合

[i]...UL,SL

[D]...SV1 [n]...K8

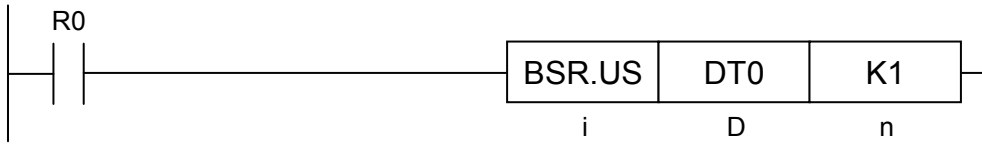


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR9(CY)	[n](シフトビット数)が演算単位より大きい場合はリセットし、 上記以外の場合は、最上位ビットから[n]ビット目の内容をセットします

# BSR (n桁右シフト)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
D	シフト対象のデータを格納しているデバイスアドレス
n	シフトデジット数を格納しているデバイスアドレスまたは定数

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H	SF	DF		" "
D	●	●	●	●			●	●	●		●	●	●								●
n	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●				●

\*1: 演算単位が 16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(10~1E)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

## ■ 動作説明

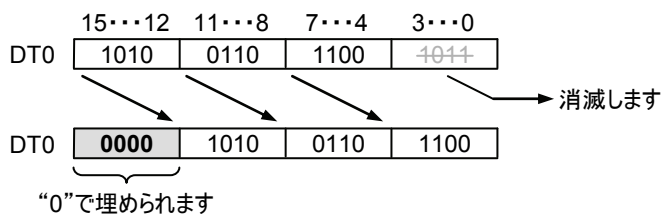
- [D]で指定されたデータを、[n](10進数で指定)デジット(4ビット)分右(下位方向)へシフトします。
- シフトを行うと、シフト実行前の最上位デジットから[n]デジット分 0 で埋めます。
- [n]はデータ中の下位 8 ビットのみ有効です。
- 演算単位が 16ビット(US,SS)の場合、シフト量は、1 デジット~4 デジットの範囲で指定します。
- 演算単位が 32ビット(UL,SL)の場合、シフト量は、1 デジット~8 デジットの範囲で指定します。

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

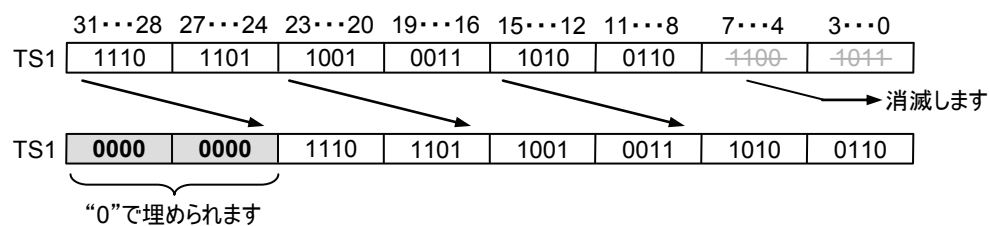
[D]...DT0 [n]...K1



例2) 演算単位が32ビット(UL,SL,SF)の場合

[i]...UL,SL,SF

[D]...TS1 [n]...K2



## ■ プログラム上のご注意

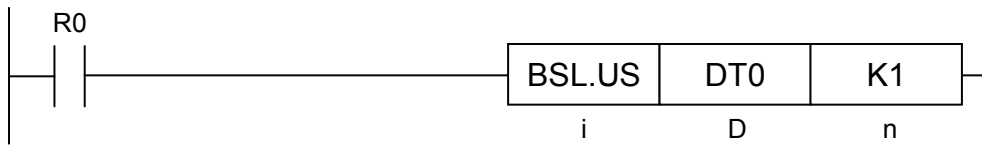
- シフトアウトした桁データは破棄されます。シフト演算補助レジスタがありません。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# BSL (n桁左シフト)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
D	シフト対象のデータを格納しているデバイスアドレス
n	シフトデジット数を格納しているデバイスアドレスまたは定数

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H	SF	DF	""		
D	●	●	●	●			●	●	●		●	●	●									●
n	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●					●

\*1: 演算単位が 16 ビット整数(SS, US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS, SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US, UL)の場合のみ指定可能

## ■ 動作説明

- [D]で指定されたデータを、[n](10進数で指定)デジット(4ビット)分右(下位方向)へシフトします。
- シフトを行うと、シフト実行前の最上位デジットから[n]デジット分 0 で埋めます。
- n はデータ中の下位 8 ビットのみ有効です。
- 演算単位が 16 ビット(US, SS)の場合、シフト量は、1 デジット~4 デジットの範囲で指定します。
- 演算単位が 32 ビット(UL, SL)の場合、シフト量は、1 デジット~8 デジットの範囲で指定します。

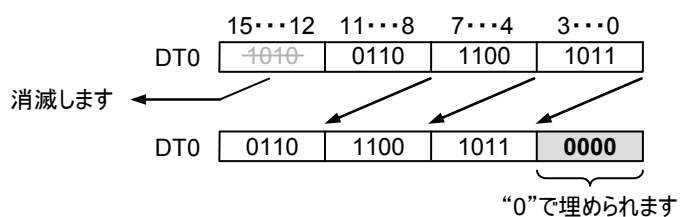


## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

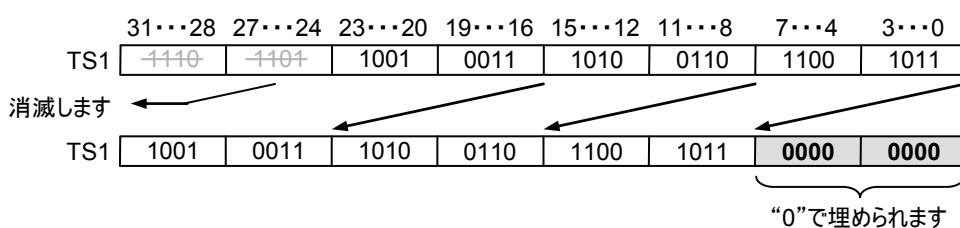
[D]...DT0 [n]...K1



例2) 演算単位が32ビット(UL,SL)の場合

[i]...US,SL

[D]...TS1 [n]...K2



## ■ プログラム上のご注意

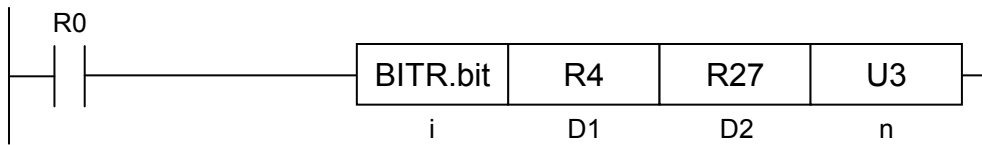
- シフトアウトした桁データは破棄されます。シフト演算補助レジスタがありません。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# BITR (nビット分の一括右シフト)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i	●						

## ■ オペランド一覧

オペランド	説明
D1	シフトするデバイスの先頭アドレス(データ形式:演算単位に従う)
D2	シフトするデバイスの終端アドレス(データ形式:演算単位に従う)
n	シフトビット数を格納しているデバイスアドレスまたは定数

## ■ 指定可能なビットデバイス (●:指定可能)

オペランド	ビットデバイス											ワードデバイスのビット指定			インデックス修飾
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b	LD.b	SD.b	
D1	○	○	○	○	—	—	—	—	—	—	○	○	○	—	○
D2	○	○	○	○	—	—	—	—	—	—	○	○	○	—	○

## ■ 指定可能なワードデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE *3	IX *3	K	U	H	SF	DF	""		
n	●	●	●	●		●	●	●	●	●	●					●	●					●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

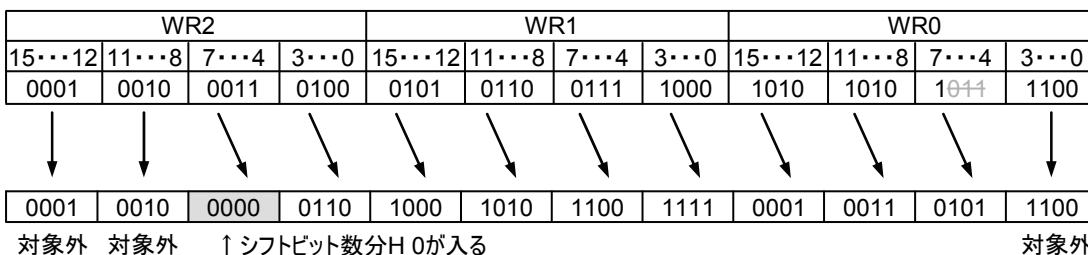
## ■ 動作説明

- [D1]～[D2]の範囲を右に[n]ビットシフトします。
- [D1]でビットの先頭アドレスを、[D2]でビットの終端アドレスを指定します。
- シフトを行うと、実行前の[D1]の下位[n]ビットは消滅、実行後の[D2]の上位[n]ビットを0で埋めます。
- [n]の設定範囲は0～15です。[n]=0のときは、シフトしません。

## ■ 処理内容

例) R4～R27を3ビットシフトする場合

[D1]...R4 [D2]...R27 [n]...U3



## ■ プログラム上のご注意

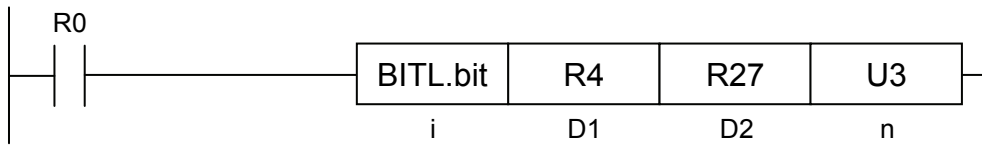
- 直接アドレスとインデックス修飾アドレスの場合、[D1]と[D2]は同じ種類のデバイスを指定してください。かつ、[D2]  $\geq$  [D1]と指定してください。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[D1] > [D2] の場合にセットします
(ER)	[n] $\geq$ 16 の場合にセットします

# BITL (nビット分の一括左シフト)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i	●						

## ■ オペランド一覧

オペランド	説明
D1	シフトするデバイスの先頭アドレス(データ形式:演算単位に従う)
D2	シフトするデバイスの終端アドレス(データ形式:演算単位に従う)
n	シフトビット数を格納しているデバイスアドレスまたは定数

## ■ 指定可能なビットデバイス (●:指定可能)

オペランド	ビットデバイス											ワードデバイスのビット指定			インデックス修飾	
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b	LD.b	SD.b		
D1	●	●	●	●								●	●	●		●
D2	●	●	●	●								●	●	●		●

## ■ 指定可能なワードデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾*1		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		""	
n	●	●	●	●			●	●	●	●	●					●	●					●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

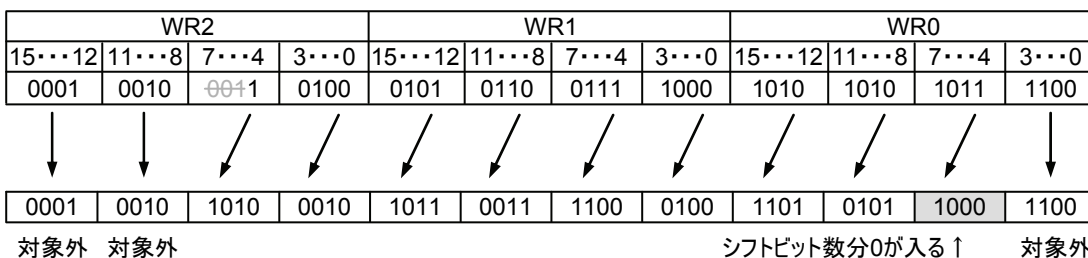
## ■ 動作説明

- [D1]~[D2]の範囲を左に[n]ビットシフトします。
- [D1]でビットの先頭アドレスを、[D2]でビットの終端アドレスを指定します。
- シフトを行うと、実行前の[D1]の上位[n]ビットは消滅、実行後の[D2]の下位[n]ビットを0で埋めます。
- [n]の設定範囲は0~15です。[n]=0のときは、シフトしません。

## ■ 処理内容

例) R4~R27を3ビットシフトする場合

[D1]...R4 [D2]...R27 [n]...U3



### ■ プログラム上のご注意

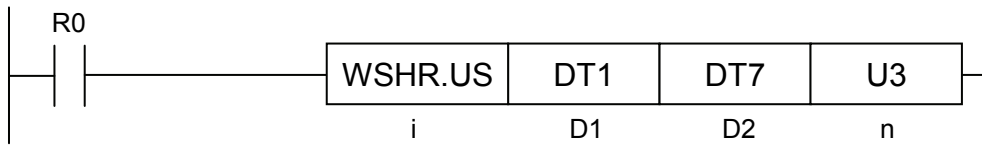
- 直接アドレスとインデックス修飾アドレスの場合、[D1]と[D2]は同じ種類のデバイスを指定してください。かつ、[D2]  $\geq$  [D1]と指定してください。

### ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[D1] > [D2] の場合にセットします
(ER)	[n] $\geq$ 16 の場合にセットします

# WSHR (ブロック領域のnワードの右シフト)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
D1	シフト対象の先頭アドレス
D2	シフト対象の終端アドレス
n	右シフトするワード数(データ指定可能範囲:0~255ワード)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾*1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""	
D1	●	●	●	●			●	●	●		●										●
D2	●	●	●	●			●	●	●		●										●
n	●	●	●	●			●	●	●	●	●				●	●	●				●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*2: 演算単位が符号付き整数(SS)の場合のみ指定可能

\*3: 演算単位が符号なし整数(US)の場合のみ指定可能

## ■ 動作説明

- [D1]で指定されたエリアから[D2]で指定されたエリアを[n]ワード右シフトします。
- シフト対象の先頭アドレスから終端アドレスまでをシフトワード数分、右にシフトします。
- 先頭アドレスからシフトワード数分消滅し、終端アドレスからシフトワード数分 H0 が入ります。
- シフト対象範囲よりシフトワード数が大きい場合、シフト対象範囲が全て H0000 になります。

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[i]...US

[D1]...DT1 [D2]...DT7 [n]...U3

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	<b>EEFF</b>	<b>CCDD</b>	<b>AABB</b>	<b>8899</b>	<del>6677</del>	<del>4455</del>	<del>2233</del>	0011	(HEX)

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	<b>0000</b>	<b>0000</b>	<b>0000</b>	<b>EEFF</b>	<b>CCDD</b>	<b>AABB</b>	<b>8899</b>	0011	(HEX)

↑シフトビット数分H 0000が入る

例2) 演算単位が16ビット(SS)の場合

[i]...SS

[D1]...DT1 [D2]...DT7 [n]...K2

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	<b>EEFF</b>	<b>CCDD</b>	<b>AABB</b>	<b>8899</b>	<b>6677</b>	<del>4455</del>	<del>2233</del>	0011	(HEX)

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0	
5678	1234	<b>0000</b>	<b>0000</b>	<b>EEFF</b>	<b>CCDD</b>	<b>AABB</b>	<b>8899</b>	<b>6677</b>	0011	(HEX)

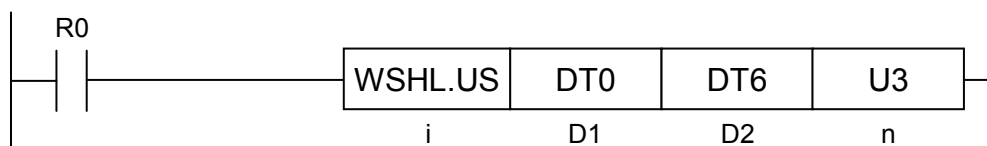
↑シフトビット数分H 0000が入る

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[D1] > [D2]の場合にセットします
(ER)	[n](シフトワード数)が指定可能範囲外の場合にセットします

# WSHL (ブロック領域のnワードの左シフト)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
D1	シフト対象の先頭アドレス
D2	シフト対象の終端アドレス
n	右シフトするワード数(データ指定可能範囲:0~255ワード)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾*1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K *2	U *3	H	SF	DF	""	
D1	●	●	●	●			●	●	●		●										●
D2	●	●	●	●			●	●	●		●										●
n	●	●	●	●			●	●	●	●	●				●	●	●				●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*2: 演算単位が符号付き整数(SS)の場合のみ指定可能

\*3: 演算単位が符号なし整数(US)の場合のみ指定可能

## ■ 動作説明

- [D1]で指定されたエリアから[D2]で指定されたエリアを[n]ワード左シフトします。
- シフト対象の先頭アドレスから終端アドレスまでをシフトワード数分、左にシフトします。
- 終端アドレスからシフトワード数分消滅し、先頭アドレスからシフトワード数分 H0 が入ります。
- シフト対象範囲よりシフトワード数が大きい場合、シフト対象範囲が全て H0000 になります。



## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[i]...US

[D1]...DT0 [D2]...DT6 [n]...U3

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
5678	1234	EEFF	<del>GGDD</del>	<del>AABB</del>	<del>8899</del>	<b>6677</b>	<b>4455</b>	<b>2233</b>	<b>0011</b>

(HEX)

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
5678	1234	EEFF	<b>6677</b>	<b>4455</b>	<b>2233</b>	<b>0011</b>	<b>0000</b>	<b>0000</b>	<b>0000</b>

(HEX)

↑シフトビット数分H 0000が入る

例2) 演算単位が16ビット(SS)の場合

[i]...SS

[D1]...DT1 [D2]...DT6 [n]...K2

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
5678	1234	EEFF	<del>GGDD</del>	<del>AABB</del>	<b>8899</b>	<b>6677</b>	<b>4455</b>	<b>2233</b>	0011

(HEX)

DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
5678	1234	<b>EEFF</b>	<b>8899</b>	<b>6677</b>	<b>4455</b>	<b>2233</b>	<b>0000</b>	<b>0000</b>	0011

(HEX)

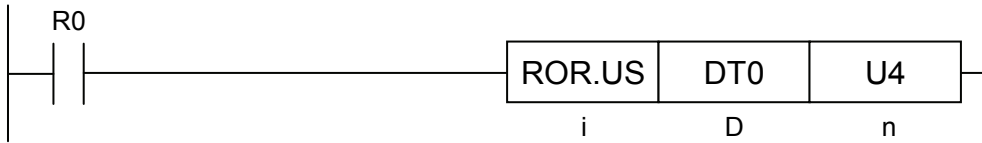
↑シフトビット数分H 0000が入る

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[D1] > [D2]の場合にセットします
(ER)	[n](シフトワード数)が指定可能範囲外の場合にセットします

# ROR (データの右回転)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
D	回転対象のデータを格納しているデバイスアドレス
n	回転ビット数を格納しているデバイスアドレスまたは定数(データ指定可能範囲:0~255)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "		
D	●	●	●	●			●	●	●		●	●	●	●								●
n	●	●	●	●			●	●	●	●	●	●	●	●		●	●					●

\*1 演算単位が 16 ビット整数(US)の場合は指定不可

\*2 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

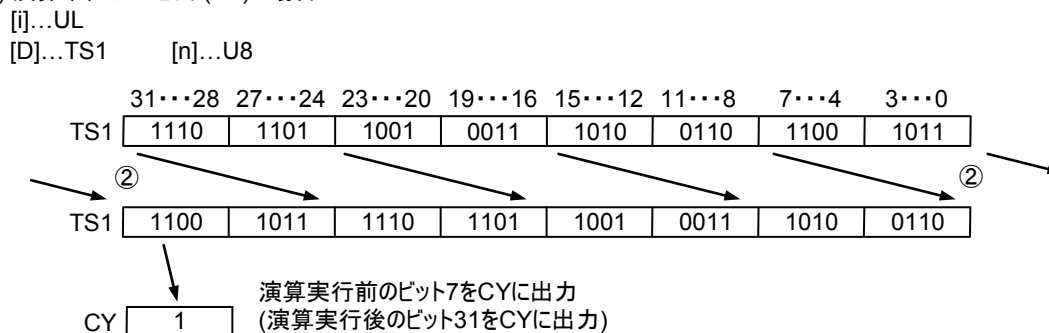
- [D]で指定されたデータを、[n](10 進数で指定)で指定したビット分右(下位ビット方向)へ回転します。
- [n]はデータ中の下位 8 ビットのみ有効です。回転量は、0 ビット~255 ビットの範囲で指定します。
- SR9(CY)には(回転数 - 1)ビットが出力されます。
- 演算単位が 16 ビット(US)の場合、[n]が 0 または 16 の倍数の場合は回転数を 0 とみなして無処理となります。
- 演算単位が 32 ビット(UL)の場合、[n]が 0 または 32 の倍数の場合は回転数を 0 とみなして無処理となります。

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合



例2) 演算単位が32ビット(UL)の場合

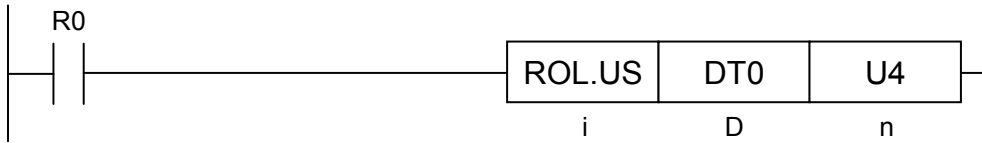


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR9(CY)	演算実行前のデータの(回転数 - 1)ビットが出力されます 演算単位が 16 ビット(US)の場合、[n]が 0 または 16 の倍数の場合は回転数を 0 とみなして変化しません 演算単位が 32 ビット(UL)の場合、[n]が 0 または 32 の倍数の場合は回転数を 0 とみなして変化しません

# ROL (データの左回転)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
D	回転対象のデータを格納しているデバイスアドレス
n	回転ビット数を格納しているデバイスアドレスまたは定数(データ指定可能範囲:0~255)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "		
D	●	●	●	●			●	●	●		●	●	●	●								●
n	●	●	●	●			●	●	●	●	●	●	●	●		●	●					●

\*1: 演算単位が 16 ビット整数(US)の場合は指定不可

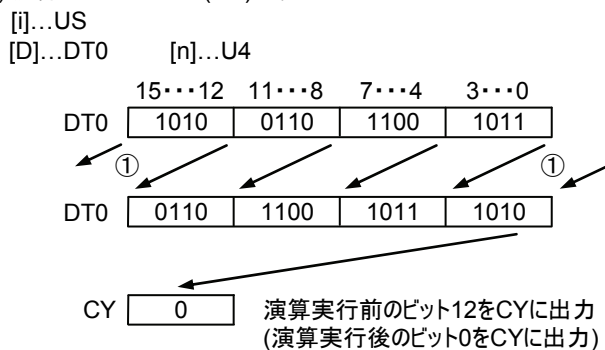
\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

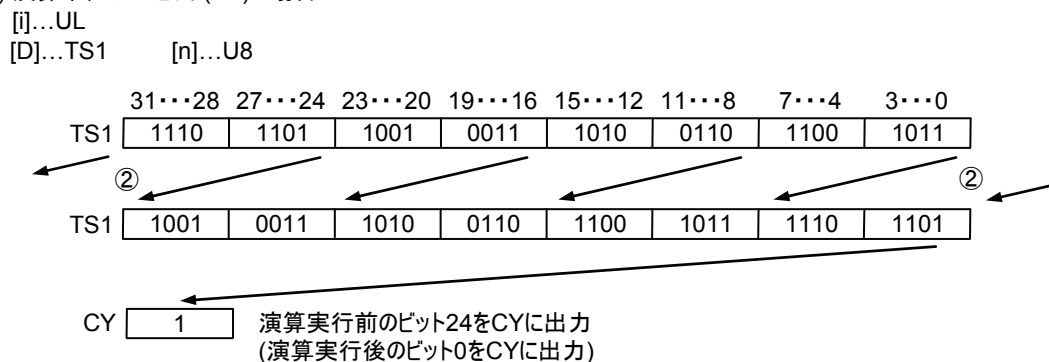
- [D]で指定されたデータを、[n](10 進数で指定)で指定したビット分左(上位ビット方向)へ回転します。
- [n]はデータ中の下位 8 ビットのみ有効です。回転量は、0 ビット~255 ビットの範囲で指定します。
- SR9(CY)には(演算単位のビット長 - 回転数)ビットが出力されます。
- 演算単位が 16 ビット(US)の場合、[n]が 0 または 16 の倍数の場合は回転数を 0 とみなして無処理となります。
- 演算単位が 32 ビット(UL)の場合、[n]が 0 または 32 の倍数の場合は回転数を 0 とみなして無処理となります。

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合



例2) 演算単位が32ビット(UL)の場合

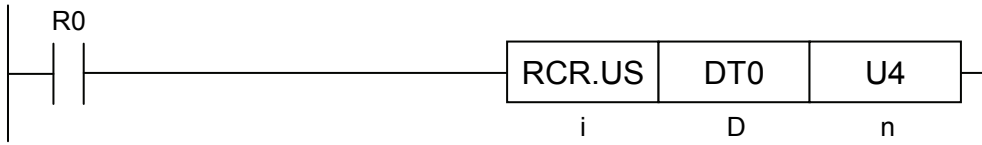


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR9(CY)	演算実行前のデータの(演算単位のビット長 - 回転数)ビットが出力されます 演算単位が 16 ビット(US)の場合、[n]が 0 または 16 の倍数の場合は回転数を 0 とみなして変化しません 演算単位が 32 ビット(UL)の場合、[n]が 0 または 32 の倍数の場合は回転数を 0 とみなして変化しません

# RCR (データのキャリー込み右回転)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
D	回転対象のデータを格納しているデバイスアドレス
n	回転ビット数を格納しているデバイスアドレスまたは定数(データ指定可能範囲:0~255)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""		
D	●	●	●	●			●	●	●		●	●	●									●
n	●	●	●	●			●	●	●	●	●	●	●	●		●	●					●

\*1: 演算単位が 16 ビット整数(US)の場合は指定不可

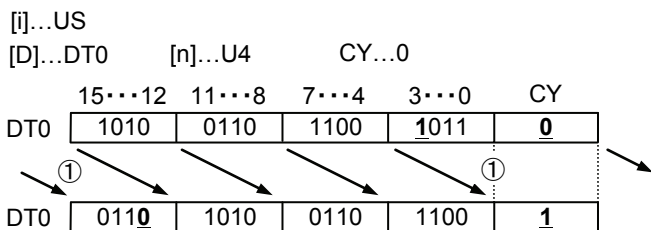
\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- [D]で指定されたデータを、[n](10進数で指定)で指定したビット分右(下位ビット方向)へ SR9(CY)込みで回転します。
- [n]はデータ中の下位 8 ビットのみ有効です。回転量は、0 ビット~255 ビットの範囲で指定します。
- SR9(CY)には(回転数 - 1)ビットが出力されます。
- 演算単位が 16 ビット(US)の場合、[n]が 0 または 17 の倍数の場合は回転数を 0 とみなして無処理となります。
- 演算単位が 32 ビット(UL)の場合、[n]が 0 または 33 の倍数の場合は回転数を 0 とみなして無処理となります。

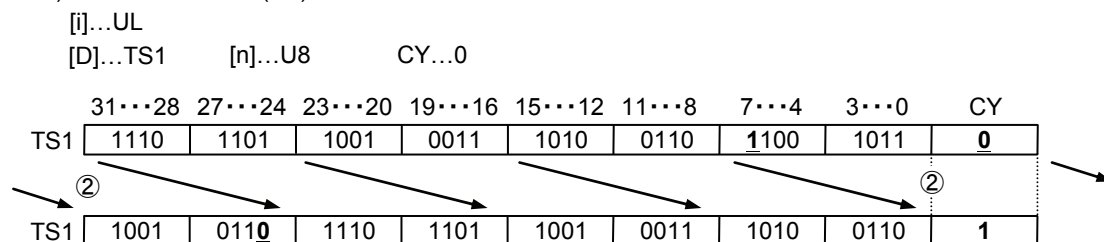
## ■ 処理内容

例1) 演算単位が16ビット(US)の場合



演算実行前のビット3をCYに出力  
演算実行前のCYをビット12に出力

例2) 演算単位が32ビット(UL)の場合



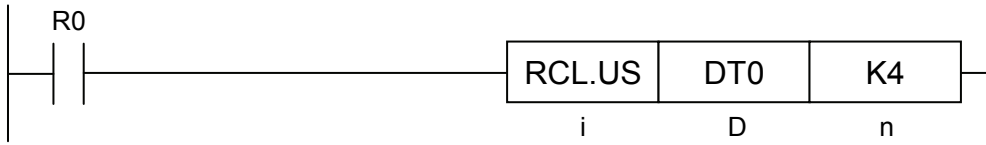
演算実行前のビット7をCYに出力  
演算実行前のCYをビット24に出力

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR9(CY)	演算実行前のデータの(回転数 - 1)ビットが出力されます 演算単位が 16 ビット(US)の場合、[n]が 0 または 17 の倍数の場合は回転数を 0 とみなして変化しません 演算単位が 32 ビット(UL)の場合、[n]が 0 または 33 の倍数の場合は回転数を 0 とみなして変化しません

# RCL (データのキャリー込み左回転)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
D	回転対象のデータを格納しているデバイスアドレス
n	回転ビット数を格納しているデバイスアドレスまたは定数(データ指定可能範囲:0~255)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""	
D	●	●	●	●			●	●	●		●	●	●								●
n	●	●	●	●			●	●	●	●	●	●	●	●		●	●				●

\*1:演算単位が 16 ビット整数(US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- [D]で指定されたデータを、[n](10 進数で指定)で指定したビット分左(上位ビット方向)へ SR9(CY)込みで回転します。
- [n]はデータ中の下位 8 ビットのみ有効です。回転量は、0 ビット~255 ビットの範囲で指定します。
- SR9(CY)には(演算単位のビット長 - 回転数)ビットが出力されます。
- 演算単位が 16 ビット(US)の場合、[n]が 0 または 17 の倍数の場合は回転数を 0 とみなして無処理となります。
- 演算単位が 32 ビット(UL)の場合、[n]が 0 または 33 の倍数の場合は回転数を 0 とみなして無処理となります。

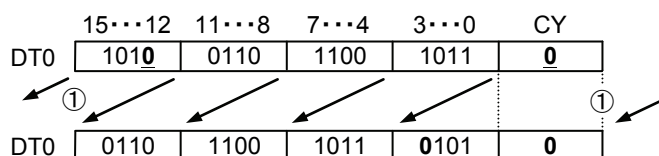


## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[i]...US

[D]...DT0 [n]...K4 CY...0



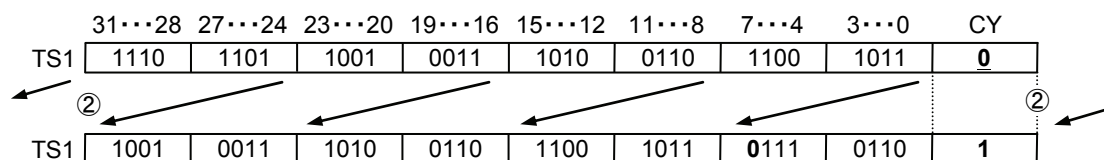
演算実行前のビット12をCYに出力

演算実行前のCYをビット3に出力

例2) 演算単位が32ビット(UL)の場合

[i]...UL

[D]...TS1 [n]...K8 CY...0



演算実行前のビット24をCYに出力

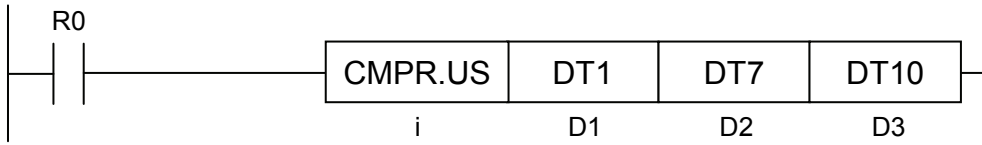
演算実行前のCYをビット7に出力

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR9(CY)	演算実行前のデータの(演算単位のビット長 - 回転数)ビットが出力されます 演算単位が 16 ビット(US)の場合、[n]が 0 または 17 の倍数の場合は回転数を 0 とみなして変化しません 演算単位が 32 ビット(UL)の場合、[n]が 0 または 33 の倍数の場合は回転数を 0 とみなして変化しません

# CMPR (圧縮シフト読み出し)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
D1	バッファの先頭アドレス
D2	バッファの終端アドレス
D3	読み出しデータを格納するデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

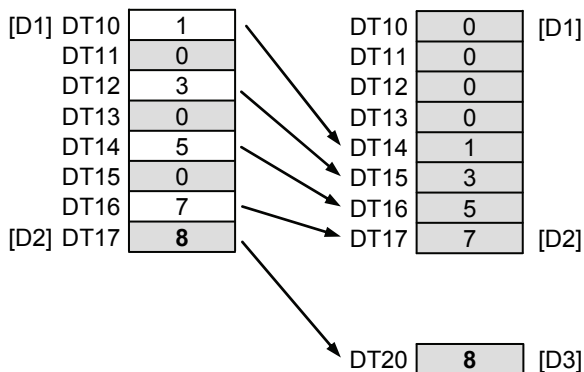
オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *1	K	U	H	SF	DF		""	
D1	●	●	●	●			●	●	●		●											●
D2	●	●	●	●			●	●	●		●											●
D3	●	●	●	●			●	●	●		●											●

\*1: インデックスレジスタ (I0~IE)

## ■ 動作説明

- [i]の演算単位に従って[D2]を[D3]へ転送し[D1]~[D2]で指定されたエリアを圧縮します。(圧縮する時に[D3]へ転送したデータは除く)
- 指定されたエリア内の0を除くデータを指定されたエリアのアドレス上位から順に配置、残りのエリアを0クリアします。

[D1]にDT10、[D2]にDT17、[D3]にDT20を指定した場合の圧縮シフト読み出しの例



## ■ 処理内容

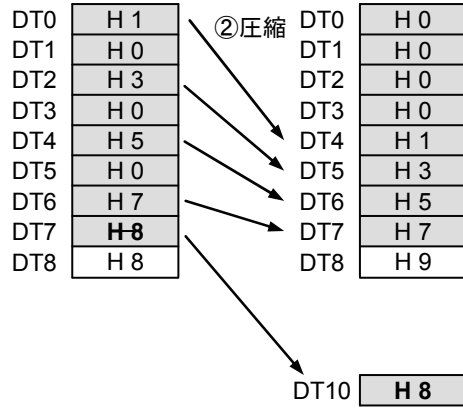
- 1) バッファの終端を読み出しデータに転送します。
- 2) バッファから終端を含むデータを除いて圧縮します。

例) 演算単位が16ビット(US,SS)の場合(2回実行した場合)

[i]...US,SS

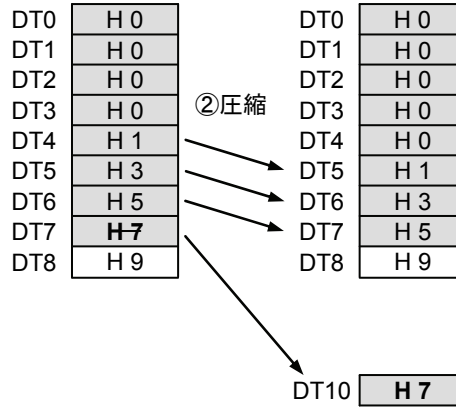
[D1]...DT1 [D2]...DT7 [D3]...DT10

1回目実行



①読み出しデータ転送

2回目実行



①読み出しデータ転送

## ■ プログラム上の注意

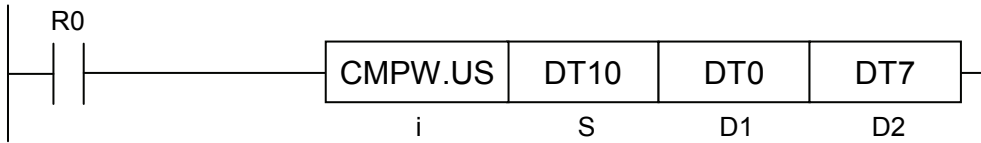
- 直接アドレスとインデックス修飾アドレスの場合、[D1]と[D2]は同じ種類のデバイスを指定してください。かつ、[D2] ≥ [D1]と指定してください。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	[D1] > [D2]の場合にセットします

# CMPW (圧縮シフト書き込み)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
S	書き込みデータ
D1	バッファの先頭アドレス
D2	バッファの終端アドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K *3	U *4	H	SF	DF	" "		
S	●	●	●	●	●	●	●	●	●	●	●				●	●	●					●
D1	●	●	●	●			●	●														●
D2	●	●	●	●			●	●														●

\*1: 16ビットデバイス、整数定数のみ修飾可能

\*2: インデックスレジスタ(I0~IE)

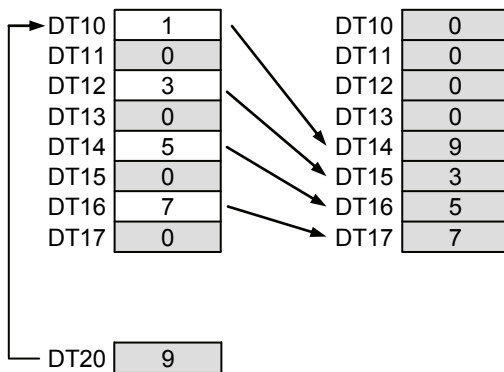
\*3: 演算単位が符号付き整数(SS)の場合のみ指定可能

\*4: 演算単位が符号なし整数(US)の場合のみ指定可

## ■ 動作説明

- [n]の演算単位に従って[S]を[D1]へ転送し[D1]~[D2]で指定されたエリアを圧縮します。
- 指定されたエリア内の 0 を除くデータを指定されたエリアのアドレス上位から順に配置、残りのエリアを 0 クリアします。

[D1]にDT10、[D2]にDT17、[D3]にDT20を指定した場合の圧縮シフト読み出しの例



## ■ 処理内容

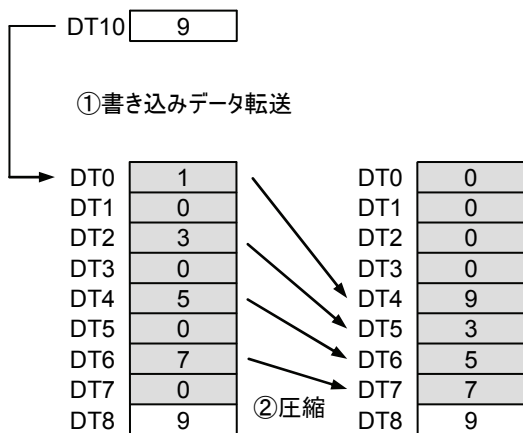
- 1) 書き込みデータをバッファ先頭に転送します。(先頭データを上書きします)
- 2) バッファの先頭～終端の範囲で圧縮します。

例) 演算単位が16ビット(US,SS)の場合(2回実行した場合)

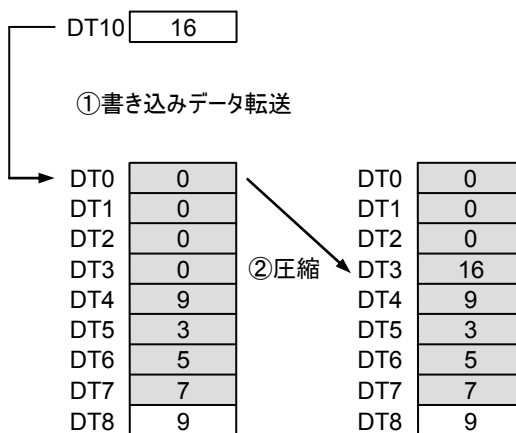
[i]...US,SS

[S]...DT10 [D1]...DT0 [D2]...DT7

1回目実行



2回目実行



## ■ プログラム上の注意

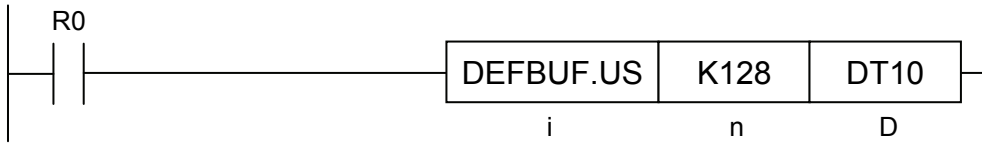
- 直接アドレスとインデックス修飾アドレスの場合、[D1]と[D2]は同じ種類のデバイスを指定してください。かつ、[D2] ≧ [D1]と指定してください。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	[D1] > [D2]の場合にセットします

# DEFBUF (バッファ定義)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
n	バッファサイズを指定するデバイスアドレスまたは定数(データ指定可能範囲:1 ~ 4096)
D	データバッファの先頭のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K *3	U *4	H	SF	DF		" "
n	●	●	●	●			●	●	●		●				●	●	●				●
D							●	●													●

\*1: 16ビットデバイス、整数定数のみ修飾可能

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*4: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って[D]のエリアを先頭に[n]データ分のデータバッファを定義します。
- ([D]+1)(使用サイズ)~([D]+3)(書き込みポインタ)までを初期化(0クリア)します。

## ■ データバッファの形式(FIFOバッファの場合)

[D]	バッファサイズ	…データバッファエリアのサイズ	初期値:[n](バッファサイズ)
[D]+1	格納データ数	…格納データ数(演算単位別)	初期値:H 0000
[D]+2	読み出しポインタ	…[D]+4からの相対番号	初期値:H 0000
[D]+3	書き込みポインタ	…[D]+4からの相対番号	初期値:H 0000
⋮			
⋮			
⋮			

} データバッファエリア  
※データバッファエリアはクリアしません

## ■ データバッファの形式(LIFOバッファの場合)

[S]	バッファサイズ	…データバッファエリアのサイズ	初期値:[n](バッファサイズ)
[S]+1	格納データ数	…格納データ数(演算単位別)	初期値:H 0000
[S]+2	0固定	…0固定	初期値:H 0000
[S]+3	LIFOポインタ	…[D]+4からの相対番号	初期値:H 0000
⋮			
⋮			
⋮			

} データバッファエリア  
※データバッファエリアはクリアしません

## ■ 処理内容

- 1) [D](バッファ先頭)に[n](バッファサイズ)を設定します。
- 2) [D]+1(格納データ数)~[D]+3(書き込みポインタ)までを 0 クリアします。

例) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[n]...K 128(U 128) [D]...DT0

DT0	K 0	→	DT0	K 128	(バッファサイズ)
DT1	K 1		DT1	K 0	(格納データ数)
DT2	K 2		DT2	K 0	(読み出しポインタ)
DT3	K 3		DT3	K 0	(書き込みポインタ)
DT4	K 4		DT4	K 4	
DT5	K 5		DT5	K 5	
DT6	K 6		DT6	K 6	
DT7	K 7		DT7	K 7	
DT8	K 8		DT8	K 8	

## ■ 関連命令

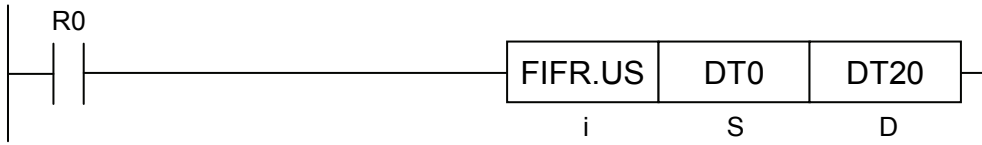
- FIFR(16・32 ビットデータバッファからデータ読み出し(先入・先出))
- BUFW(16・32 ビットデータバッファデータ書き込み)
- LIFR(16・32 ビットデータバッファからデータ読み出し(後入・先出))

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[n](バッファサイズ)が指定可能範囲外の場合にセットします
(ER)	[D](バッファ先頭)+[n](バッファサイズ)の範囲がアクセス可能範囲外の場合にセットします

# FIFR (データ読み出し: 先入・先出)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
S	データバッファの先頭のデバイスアドレス
D	読み出しデータのデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF	DF	" "		
S							●	●														●
D	●	●	●	●			●	●	●		●											●

\*1: 16ビットデバイス、整数定数のみ修飾可能

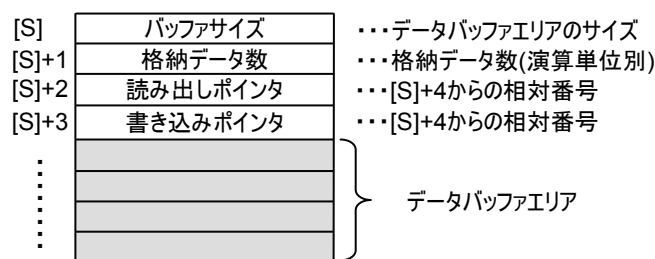
\*2: インデックスレジスタ (I0~IE)

## ■ 動作説明

- [S]で指定した FIFO バッファからデータを読み出し[D]に設定します。  
([S]のバッファエリアは DEFBUF 命令で先にバッファを定義する必要があります)
- 実行前のバッファ整合性チェック(下記の場合、演算エラーとなります)
  - 1) [S](バッファサイズ) > 4096 または [S](バッファサイズ) = 0
  - 2) [S]+1(格納データ数) = 0
  - 3) [S]+1(格納データ数) > [S](バッファサイズ)
  - 4) [S]+2(読み出しポインタ) > [S](バッファサイズ)
  - 5) バッファエリアが指定されたデバイスの上限値オーバー
- [i]の演算単位に従って[S]+2(読み出しポインタ)が指定するエリアのデータを[D]に設定します。
- [S]+2(読み出しポインタ)をインクリメント(+1)します。
- インクリメント(+1)後、[S]+2(読み出しポインタ) = [S](バッファサイズ)の場合、[S]+2(読み出しポインタ)に 0 を設定します。
- [S]+1(格納データ数)をデクリメント(-1)します。



## ■ データバッファの形式(FIFOバッファの場合)



## ■ 処理内容

- 1) ([S]+2)(読み出しポインタ)が指定するエリアを[D](読み出しデータ)に設定します。
- 2) ([S]+2)(読み出しポインタ)をインクリメント(+1)します。
- 3) ([S]+1)(格納データ数)をデクリメント(-1)します。

例) 演算単位が16ビット(US,SS)の場合

[i]…US,SS

[S]…DT0 [D]…DT20

[S]	DT0	K 5	(バッファサイズ)	[S]	DT0	K 5	(バッファサイズ)	
[S]+1	DT1	K 4	(格納データ数)	[S]+1	DT1	K 3	(格納データ数)	③デクリメント(-1)
[S]+2	DT2	K 0	(読み出しポインタ)	[S]+2	DT2	K 1	(読み出しポインタ)	②インクリメント(+1)
[S]+3	DT3	K 4	(書き込みポインタ)	[S]+3	DT3	K 4	(書き込みポインタ)	
[S]+4	DT4	H 101	(0番目)	[S]+4	DT4	H 101	(0番目)	
[S]+5	DT5	H 102	(1番目)	[S]+5	DT5	H 102	(1番目)	
[S]+6	DT6	H 103	(2番目)	[S]+6	DT6	H 103	(2番目)	
[S]+7	DT7	H 104	(3番目)	[S]+7	DT7	H 104	(3番目)	
[S]+8	DT8	H 105	(4番目)	[S]+8	DT8	H 105	(4番目)	

DT20 H 20

DT20 H 101

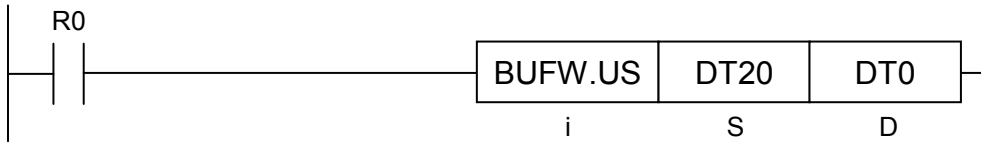
①[S]+2(読み出しポインタ)が0を指しているので  
バッファの0番目のデータをDに転送

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	[S](バッファサイズ) > 4096 または [S](バッファサイズ) = 0 の場合にセットします
	[S]+1(格納データ数)が 0 の場合にセットします
	[S]+1(格納データ数) > [S](バッファサイズ)の場合にセットします
	[S]+2(読み出しポインタ) ≥ [S](バッファサイズ)の場合にセットします
	バッファエリアが指定されたデバイスの上限值オーバーの場合にセットします

# BUFV (データ書き込み)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
S	書き込みデータのデバイスアドレスまたは定数
D	データバッファの先頭のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K *3	U *4	H	SF	DF		" "
S	●	●	●	●	●	●	●	●	●	●	●				●	●	●				●
D							●	●													●

\*1: 16ビットデバイス、整数定数のみ修飾可能

\*2: インデックスレジスタ (I0~IE)

\*3: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*4: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

## ■ 動作説明

- [D]で指定したバッファに[S]で指定したデータを設定します。  
([D]のバッファエリアは DEFBUF 命令で先にバッファを定義する必要があります)
- 実行前のバッファ整合性チェック(下記の場合、演算エラーとなります)
  - 1) [D](バッファサイズ) > 4096 または [D](バッファサイズ) = 0
  - 2) [D]+1(格納データ数) ≥ [D](バッファサイズ)
  - 3) [D]+3(書き込みポインタ) ≥ [D](バッファサイズ)
  - 4) バッファエリアが指定されたデバイスの上限值オーバー
- [i]の演算単位に従って[D]+3(書き込みポインタ)が指定するエリアに[S]を設定します。
- [D]+3(書き込みポインタ)をインクリメント(+1)します。
- インクリメント(+1)後、「[D]+3(書き込みポインタ) = [S](バッファサイズ)」の場合、
- [D]+3(書き込みポインタ)に 0 を設定します。
- [S]+1(格納データ数)をインクリメント(+1)します。

## ■ データバッファの形式(FIFOの場合)

[D]	バッファサイズ	…データバッファエリアのサイズ
[D]+1	格納データ数	…格納データ数(演算単位別)
[D]+2	読み出しポインタ	…[D]+4からの相対番号
[D]+3	書き込みポインタ	…[D]+4からの相対番号
⋮		
⋮		
⋮		
⋮		

} データバッファエリア

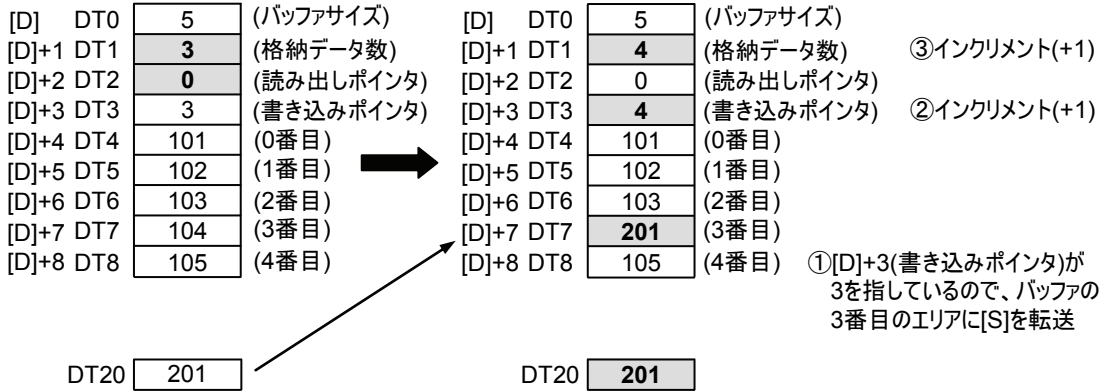
## ■ 処理内容

- 1) [D]+3(書き込みポインタ)が指定するエリアに[S](書き込みデータ)を設定します。
- 2) [D]+3(書き込みポインタ)をインクリメント(+1)します。
- 3) [D]+1(格納データ数)をインクリメント(+1)します。

例) 16ビット(US,SS)の場合

[i]...US,SS

[S]...DT20 [D]...DT0

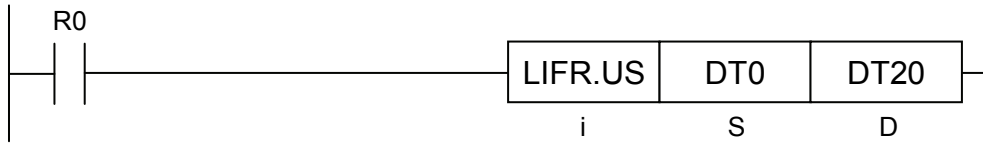


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	[D](バッファサイズ) > 4096 または [D](バッファサイズ) = 0 の場合にセットします
	[D]+1(格納データ数) ≥ [D](バッファサイズ)の場合にセットします
	[D]+3(書き込みポインタ) ≥ [D](バッファサイズ)の場合にセットします
	バッファエリアが指定されたデバイスの上限值オーバーの場合にセットします

# LIFR (データ読み出し(後入・先出))

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
S	データバッファの先頭のデバイスアドレス
D	読み出しデータのデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾 *1		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF	DF		""	
S							●	●														●
D	●	●	●	●			●	●	●		●											●

\*1: 16ビットデバイス、整数定数のみ修飾可能

\*2: インデックスレジスタ (I0~IE)

## ■ 動作説明

- [S]で指定した LIFO バッファからデータを読み出し[D]に設定します。  
([S]のバッファエリアは DEFBUF 命令で先にバッファを定義する必要があります)
- 実行前のバッファ整合性チェック(下記の場合、演算エラーとなります)
  - 1) [S](バッファサイズ) > 4096 または [S](バッファサイズ) = 0
  - 2) [S]+1(格納データ数) = 0
  - 3) [S]+2 ≠ 0
  - 4) [S]+1(格納データ数) > [S](バッファサイズ)
  - 5) [S]+3(LIFOポインタ) ≥ [S](バッファサイズ)
  - 6) バッファエリアが指定されたデバイスの上限値オーバー
- [S]+3(LIFO ポインタ) が 0 の場合、[S]+3(LIFO ポインタ)に[S](バッファサイズ)を設定します。
- [S]+3(LIFO ポインタ)をデクリメント(-1)します。
- [i]の演算単位に従って[S]+3(LIFO ポインタ)が指定するエリアのデータを[D]に設定します。
- [S]+1(格納データ数)をデクリメント(-1)します。

## ■ データバッファの形式(LIFOの場合)

[S]	バッファサイズ	…データバッファエリアのサイズ
[S]+1	格納データ数	…格納データ数(演算単位別)
[S]+2	0固定	…0固定
[S]+3	LIFOポインタ	…[S]+4からの相対番号
⋮		
⋮		
⋮		
⋮		

} データバッファエリア

## ■ 処理内容

- 1) [S]+3(LIFO ポインタ)をデクリメント(-1)します。
- 2) [S]+3(LIFO ポインタ)が指定するエリアのデータを[D](読み出しデータ)に設定します。
- 3) [S]+1(格納データ数)をデクリメント(-1)します。

例) 16ビット(US,SS)の場合

[i]...US,SS

[S]...DT0 [D]...DT20

[S]	DT	Value	Label	[S]	DT	Value	Label	Operation
[S]	DT0	K 5	(バッファサイズ)	[S]	DT0	K 5	(バッファサイズ)	
[S]+1	DT1	<b>K 4</b>	(格納データ数)	[S]+1	DT1	<b>K 3</b>	(格納データ数)	③デクリメント(-1)
[S]+2	DT2	K 0	-	[S]+2	DT2	K 0	-	
[S]+3	DT3	<b>K 4</b>	(LIFOポインタ)	[S]+3	DT3	<b>K 3</b>	(LIFOポインタ)	①デクリメント(-1)
[S]+4	DT4	H 101	(0番目)	[S]+4	DT4	H 101	(0番目)	
[S]+5	DT5	H 102	(1番目)	[S]+5	DT5	H 102	(1番目)	
[S]+6	DT6	H 103	(2番目)	[S]+6	DT6	H 103	(2番目)	
[S]+7	DT7	<b>H 104</b>	(3番目)	[S]+7	DT7	H 104	(3番目)	
[S]+8	DT8	H 105	(4番目)	[S]+8	DT8	H 105	(4番目)	

DT20 [ H 10 ]

DT20 [ **H 104** ]

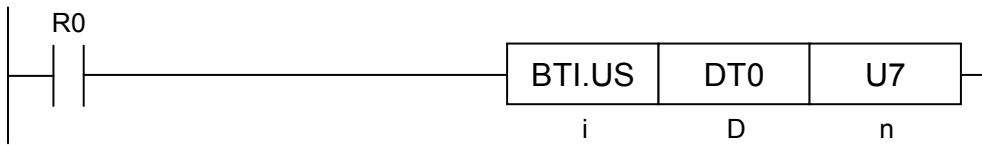
②[S]+3(LIFOポインタ)が3を指しているので  
バッファの3番目のデータを[D]に転送

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	[S](バッファサイズ) > 4096 または [S](バッファサイズ) = 0 の場合にセットします
	[S]+1(格納データ数)が 0 の場合にセットします
	[S]+2 が 0 以外の場合にセットします
	[S]+1(格納データ数) > [S](バッファサイズ)の場合にセットします
	[S]+3(LIFO ポインタ) ≥ [S](バッファサイズ)の場合にセットします
	バッファエリアが指定されたデバイスの上限值オーバーの場合にセットします

# BTI (ビット反転)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●					

## ■ オペランド一覧

オペランド	説明
D	反転対象データ(デバイスアドレス)
n	ビット番号(デバイスアドレスまたは定数)(データ指定可能範囲:0~15)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデックス 修飾 *1		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF	DF		""	
S							●	●														●
D	●	●	●	●			●	●	●		●											●

\*1: 16ビットデバイス、整数定数のみ修飾可能

\*2: インデックスレジスタ (I0~IE)

## ■ 動作説明

- [i]の演算単位に従って[D]で指定されたエリアの[n]番目のビットを反転します。

## ■ 処理内容

例) 演算単位が16ビット(US)の場合

[i]...US

[D]...DT0 [n]...U7 <ビットの7を反転>

		DT0																
ビット		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
BIN		0	1	0	1	1	0	0	0	0	0	1	1	1	0	0	0	0



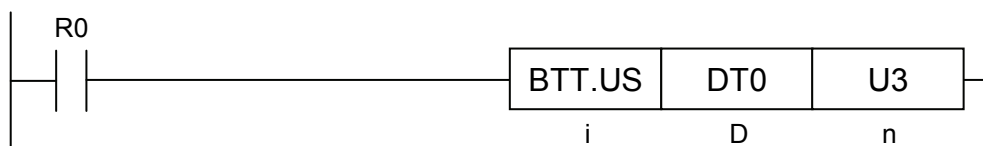
		DT0																
ビット		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
BIN		0	1	0	1	1	0	0	0	1	1	1	1	0	0	0	0	0

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	

# BTT (ビットテスト)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●					

## ■ オペランド一覧

オペランド	説明
D	テスト対象データ(デバイスアドレス)
n	ビット番号(デバイスアドレスまたは定数)(データ指定可能範囲:0~15)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""		
D	●	●	●	●			●	●	●		●											●
n	●	●	●	●			●	●	●	●	●					●	●					●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(整数定数、実数定数、文字定数には指定不可)

## ■ 動作説明

- [i]の演算単位に従って[D]で指定されたエリアの[n]番目のビットをテスト(ON/OFF判定)し、結果をSRB(=)に出力します。

## ■ 処理内容

指定されたビットの状態によりSRB(=)の結果が下記ようになります

指定ビットの状態	SRB(=)
ON(1)	OFF(0)
OFF(0)	ON(1)

例1) 演算単位が16ビット(US)の場合(SRBがOFF)

[i]...US

[D]...DT0 [n]...U3

		DT0																
ビット		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
BIN		0	1	0	1	1	0	0	0	0	1	1	1	1	1	0	0	0

実行時のフラグ動作

指定ビットの状態	SRB(=)
ON(1)	OFF(0)

例2) 演算単位が16ビット(US)の場合(SRBがON)

[i]...US

[D]...DT0 [n]...U3

		DT0																
ビット		F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	
BIN		0	1	0	1	1	0	0	0	0	1	1	1	1	0	0	0	0

実行時のフラグ動作

指定ビットの状態	SRB(=)
OFF(0)	ON(1)

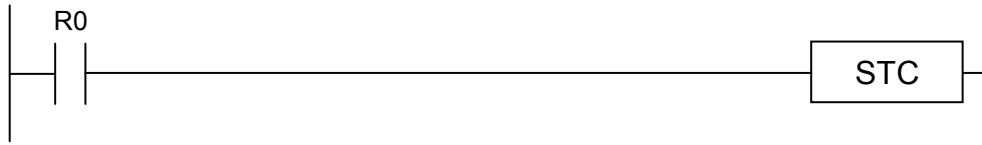
## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
SRB(=)	テストビット(ビット[n])が'0'のときセットします
	テストビット(ビット[n])が'1'のときリセットします



# STC (キャリーのセット)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

- なし。

## ■ 指定可能なデバイス (●:指定可能)

- なし。

## ■ 動作説明

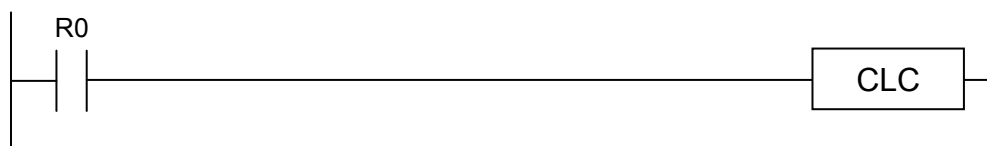
- SR9 (CY)を ON にします。

## ■ フラグ動作

名称	説明
SR9 (CY)	本命令を実行した後にセットします

# CLC (キャリーのリセット)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

- なし。

## ■ 指定可能なデバイス (●:指定可能)

- なし。

## ■ 動作説明

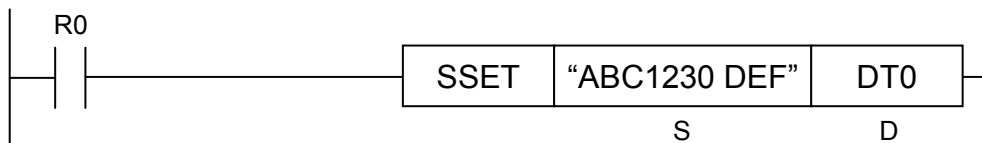
- SR9 (CY)を OFF にします。

## ■ フラグ動作

名称	説明
SR9 (CY)	本命令を実行した後にリセットします

# SSET (文字定数→ASCIIコード変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S	転送元の文字列
D	転送先の先頭デバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

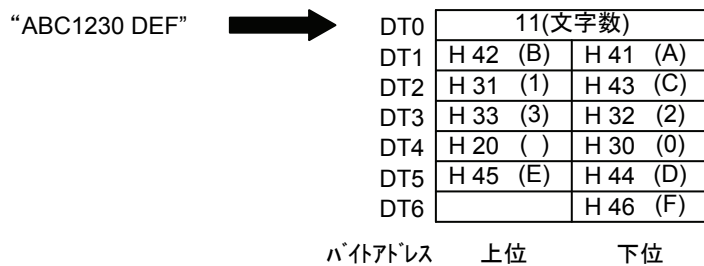
オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *1		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""			
S																						●	
D	●	●	●	●			●	●	●														●

## ■ 動作説明

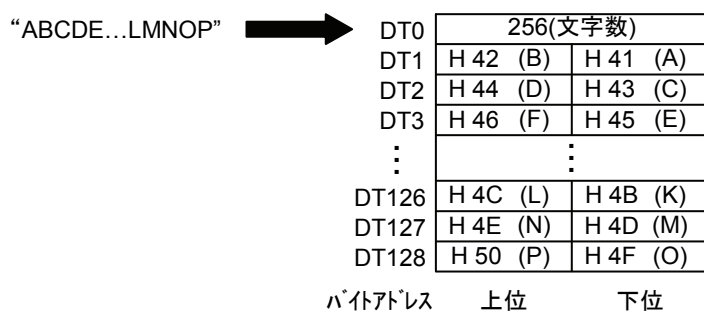
- [S]で指定した文字定数を ASCII コードに変換し、[D]を先頭とするエリアに格納します。
- エリアの先頭に文字数を 1 ワードで設定した上でその後のエリアに文字データを格納します。
- 文字定数は"(ダブルクォーテーション)で括って指定します。
- 文字定数は 1~256 文字まで設定可能です。
- 設定時に文字の終端に NULL(00)などは付加しません。

## ■ 処理内容

例1) 文字列"ABC1230 DEF"の11文字をDT0に設定する場合



例2) A~Pの16文字を1セットとして繰り返し256文字分をDT0に設定する場合



## ■ プログラム上のご注意

- 演算される前の[D]のエリアの文字データは、上書きされます。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	転送先範囲がアクセス可能範囲外の場合にセットします

■ 参考 ASCIIコード表

										b7								
										b6	0	0	0	0	1	1	1	1
										b5	0	0	1	1	0	0	1	1
										b4	0	1	0	1	0	1	0	1
b7	b6	b5	b4	b3	b2	b1	b0	R	C	0	1	2	3	4	5	6	7	
0	0	0	0	0	0	0	0	NUL	DEL	SPACE	0	@	P	`	p			
0	0	0	1	1	1	1	1	SOH	DC1	!	1	A	Q	a	q			
0	0	1	0	0	0	0	0	STX	DC2	"	2	B	R	b	r			
0	0	1	1	1	1	1	1	ETX	DC3	#	3	C	S	c	s			
0	1	0	0	0	0	0	0	EOT	DC4	\$	4	D	T	d	t			
0	1	0	1	1	1	1	1	ENQ	NAK	%	5	E	U	e	u			
0	1	1	0	0	0	0	0	ACK	SYN	&	6	F	V	f	v			
0	1	1	1	1	1	1	1	BEL	ETB	'	7	G	W	g	w			
1	0	0	0	0	0	0	0	BS	CAN	(	8	H	X	h	x			
1	0	0	1	1	1	1	1	HT	EM	)	9	I	Y	i	y			
1	0	1	0	0	0	0	0	A	LF	SUB	*	:	J	Z	j	z		
1	0	1	1	1	1	1	1	B	VT	ESC	+	;	K	[	k	{		
1	1	0	0	0	0	0	0	C	FF	FS	,	<	L	¥	l			
1	1	0	1	1	1	1	1	D	CR	GS	-	=	M	]	m	}		
1	1	1	0	0	0	0	0	E	SO	RS	.	>	N	^	n	~		
1	1	1	1	1	1	1	1	F	SI	US	/	?	O	_	o	DEL		

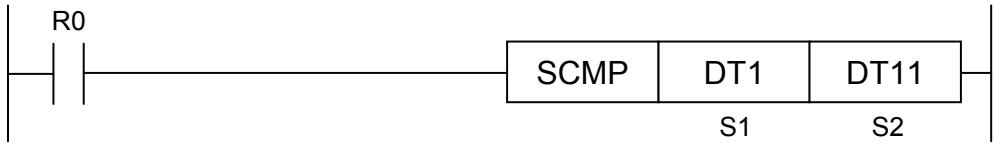
■ 参考 JIS8コード表

										0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
										0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
										0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
										0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
b7	b6	b5	b4	b3	b2	b1	b0	列 行	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	0	0	0	0	0	0	0	NUL	TC7 (DEL)	(SP)	0	@	P	`	p	↑	↑	未定義	一	タ	ミ	↑	↑		
0	0	0	1	1	1	1	1	TC1 (SOH)	DC1	!	1	A	Q	a	q	↑	↑	。	ア	チ	ム	↑	↑		
0	0	1	0	0	0	0	0	TC2 (STX)	DC2	"	2	B	R	b	r	↑	↑	「	イ	ツ	メ	↑	↑		
0	0	1	1	1	1	1	1	TC3 (ETX)	DC3	#	3	C	S	c	s	↑	↑	」	ウ	テ	モ	↑	↑		
0	1	0	0	0	0	0	0	TC4 (EOT)	DC4	\$	4	D	T	d	t	↑	↑	、	エ	ト	ヤ	↑	↑		
0	1	0	1	1	1	1	1	TC5 (ENQ)	TC8 (NAK)	%	5	E	U	e	u	↑	↑	・	オ	ナ	ユ	↑	↑		
0	1	1	0	0	0	0	0	TC6 (ACK)	TC9 (SYN)	&	6	F	V	f	v	↑	↑	ヲ	カ	ニ	ヨ	↑	↑		
0	1	1	1	1	1	1	1	BEL	ETB	'	7	G	W	g	w	↑	↑	ア	キ	ヌ	ラ	↑	↑		
1	0	0	0	0	0	0	0	EE0 (BS)	CAN	(	8	H	X	h	x	↑	↑	イ	ク	ネ	リ	↑	↑		
1	0	0	1	1	1	1	1	EE1 (HT)	EM	)	9	I	Y	i	y	↑	↑	ウ	ケ	ノ	ル	↑	↑		
1	0	1	0	0	0	0	0	EE2 (LF)	SUB	*	:	J	Z	j	z	↑	↑	エ	コ	ハ	レ	↑	↑		
1	0	1	1	1	1	1	1	EE3 (VT)	ESC	+	;	K	[	k		↑	↑	オ	サ	ヒ	ロ	↑	↑		
1	1	0	0	0	0	0	0	EE4 (FF)	IS4 (FS)	,	<	L	¥	l		↑	↑	ヤ	シ	フ	ワ	↑	↑		
1	1	0	1	1	1	1	1	EE5 (CR)	IS3 (GS)	-	=	M	]	m		↑	↑	ユ	ス	ヘ	ン	↑	↑		
1	1	1	0	0	0	0	0	E	SO	IS2 (RS)	.	>	N	^	n	~	↑	↑	ヨ	セ	ホ	”	↑	↑	
1	1	1	1	1	1	1	1	F	SI	IS1 (US)	/	?	O	_	o	DEL	↑	↑	ッ	ソ	マ	'	↑	↑	

JIS8コード表の未定義の部分は使用しないでください。

# SCMP (文字列比較)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S1	比較する文字列 1
S2	比較する文字列 2

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデック ス修飾			
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		""		
S1	●	●	●	●			●	●	●													●	
S2	●	●	●	●			●	●	●														●

## ■ 動作説明

- [S1]で指定した文字列と[S2]で指定した文字列を比較し、その判定結果をシステムリレーSRA～SRC(比較命令の判定フラグ)に出力します。

- システムリレーSRA～SRCは、大小関係により以下の通りとなります

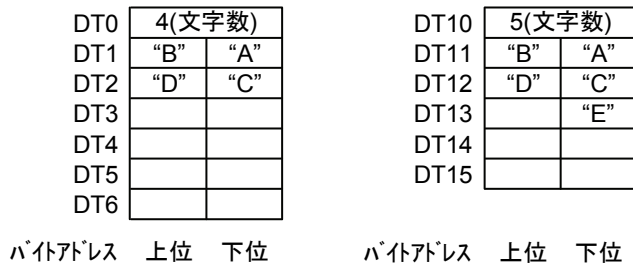
[S1]と[S2]の 関係	比較命令の判定フラグ		
	SRA	SRB	SRC
	>	=	<
[S1] < [S2]	OFF	OFF	ON
[S1] = [S2]	OFF	ON	OFF
[S1] > [S2]	ON	OFF	OFF

- 比較文字数が異なる場合の大小関係は、次の通りとなります

[S1]	大小	[S2]
"ABCDE"	=	"ABCDE"
"ABCD"	<	"ABCDE"
"B"	>	"ABCDE"

## ■ 処理内容

例) DT1とDT11を比較する場合



[S1]と[S2]の 関係	比較命令の判定フラグ		
	SRA	SRB	SRC
	>	=	<
[S1] < [S2]	OFF	OFF	ON

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします [S1]、[S2]で指定された文字列範囲がアクセス可能範囲外の場合にセットします
SRA(>)	比較結果により変動します
SRB(=)	
SRC(<)	

# SADD (文字列加算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S1	連結する文字列 1 の先頭デバイスアドレス
S2	連結する文字列 2 の先頭デバイスアドレス
D	連結後の文字列を格納する先頭デバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデックス 修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		""
S1	●	●	●	●			●	●	●												●
S2	●	●	●	●			●	●	●												●
D	●	●	●	●			●	●	●												●

## ■ 動作説明

- [S1]で指定した文字列と[S2]で指定した文字列を連結し、連結した文字列を[D]で指定したデバイスアドレスに設定します。
- 連結結果の最大文字数は 4096 文字です。

## ■ 処理内容

例) DT0とDT10の文字列を連結し、結果をDT20に設定する場合

DT0	5(文字数)	DT10	3(文字数)	DT20	8(文字数)
DT1	“B” “A”	DT11	“2” “1”	DT21	“B” “A”
DT2	“D” “C”	DT12	“ ” “3”	DT22	“D” “C”
DT3	“ ” “E”	DT13	“ ” “ ”	DT23	“1” “E”
DT4	“ ” “ ”	DT14	“ ” “ ”	DT24	“3” “2”
DT5	“ ” “ ”	DT15	“ ” “ ”	DT25	“ ” “ ”
DT6	“ ” “ ”	DT16	“ ” “ ”	DT26	“ ” “ ”
バイトアドレス	上位 下位	+	バイトアドレス 上位 下位	=	バイトアドレス 上位 下位

## ■ プログラム上のご注意

- 演算される前の[D]のエリアの文字データは、上書きされます。

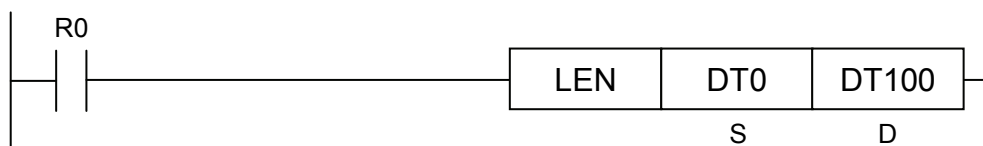
## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	[S1]、[S2]で指定された文字列範囲がアクセス可能範囲外の場合にセットします
	転送先範囲がアクセス可能範囲外の場合にセットします
	連結最大文字数を越えた場合にセットします



# LEN (文字列長取得)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S	文字列の先頭デバイスアドレス
D	文字列長を格納する先頭デバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

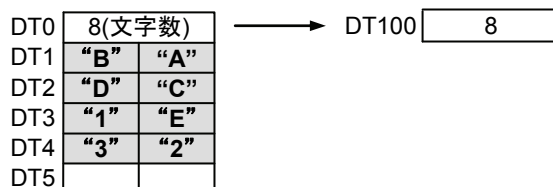
オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *1	K	U	H	SF	DF	""		
S	●	●	●	●			●	●	●													●
D	●	●	●	●			●	●	●		●											●

## ■ 動作説明

- [S]で指定した文字列の先頭に格納されている文字数を[D]で指定したデバイスアドレスに設定します。

## ■ 処理内容

例) DT0の文字数をDT100に設定する場合



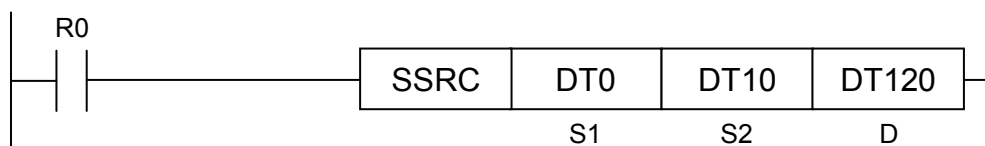
バイトアドレス 上位 下位

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	取得した文字数が 4096(文字)をえている場合にセットします
(ER)	[S]で指定された文字列範囲がアクセス可能範囲外の場合にセットします

# SSRC (文字列検索)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S1	検索する文字列データの先頭デバイスアドレス
S2	検索される文字列の先頭デバイスアドレス
D	検索結果を格納する先頭デバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

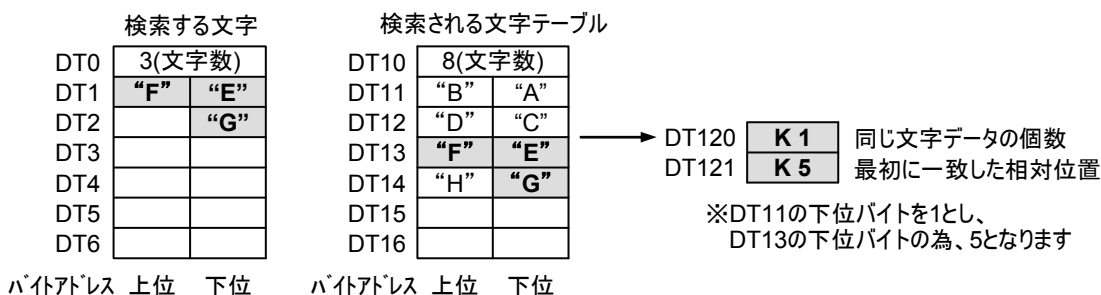
オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""		
S1	●	●	●	●			●	●	●													●
S2	●	●	●	●			●	●	●													●
D	●	●	●	●			●	●	●		●											●

## ■ 動作説明

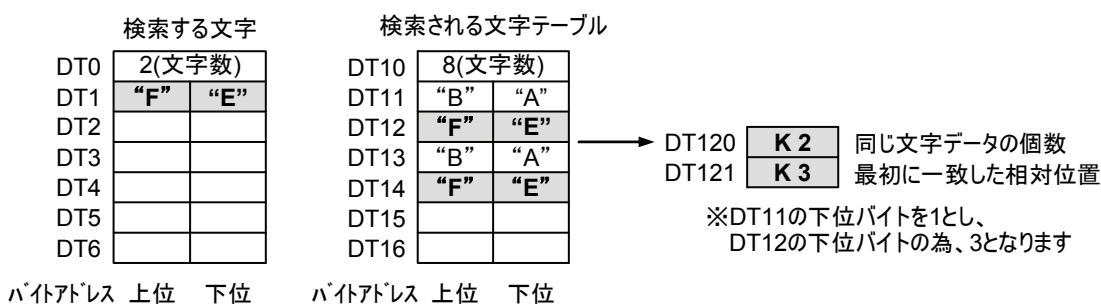
- [S1]で指定した文字データを[S2]で指定した文字列より検索します。
- 検索結果は、同じ文字データの個数を[D]で指定したデバイスアドレスに格納し、最初に一致した相対位置(バイト単位)を[D]+1 に格納します。

## ■ 処理内容

例1) DT0とDT10の文字列を連結し、結果をDT20に設定する場合



例2) 検索文字データが2つ検索される場合



## ■ プログラム上のご注意

- 検索する側の文字列の文字数[S1]には、検索を行う文字数分を指定してください。

例)

DT0	2(文字数)
DT1	"B" "A"
DT2	"D" "C"

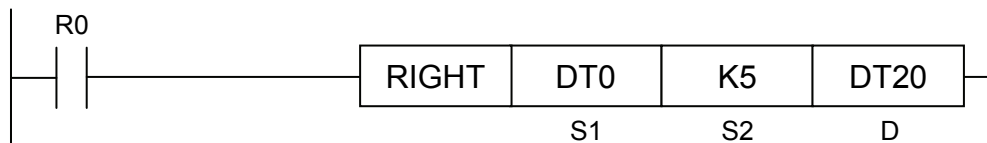
文字数を1と指定すると文字"A"として検索し、  
文字数を2と指定すると文字"AB"をひとつの固まりとして検索します

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S1]文字数 > [S2]の文字数の場合にセットします
(ER)	[S1]、[S2]で指定された文字列範囲がアクセス可能範囲外の場合にセットします

# RIGHT (文字列右側取り出し)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S1	取り出し元の先頭デバイスアドレス
S2	取り出したい文字数(データ指定可能範囲:1~4096)
D	取り出し結果を格納する先頭デバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""		
S1	●	●	●	●			●	●	●													●
S2	●	●	●	●	●	●	●	●	●	●	●				●	●	●					●
D	●	●	●	●			●	●	●													●

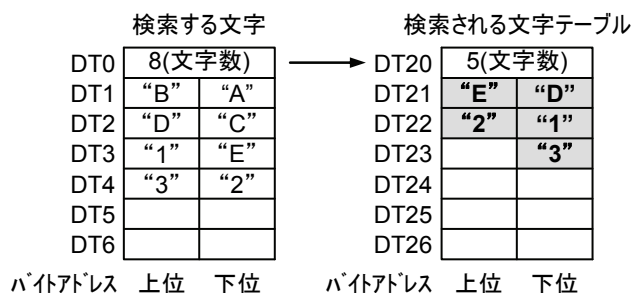
## ■ 動作説明

- [S1]で指定された文字列の右側(文字データの最後)から[S2]で指定した文字数分を取り出し、[D]で指定したデバイスアドレスに格納します。

## ■ 処理内容

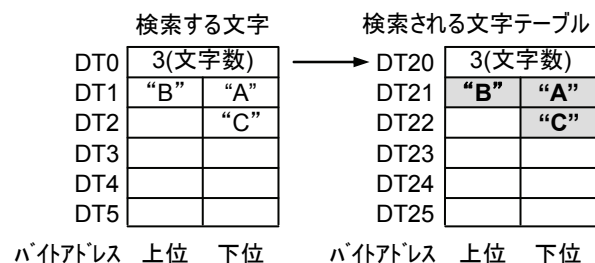
例1) DT0の文字列の最後の5文字を取り出し、DT20へ転送する場合

[S1]...DT0 [S2]...K5 [D]...DT20



例2) [S2]の文字数>[S1]の文字列の文字数の場合

[S1]...DT0 [S2]...K7 [D]...DT20



## ■ プログラム上のご注意

- 演算される前の[D]のエリアの文字データは、上書きされます。
- [S2]の文字数>[S1]の文字列の文字数の場合、[S1]の文字数分転送します。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	転送先範囲がアクセス可能範囲外の場合にセットします
(ER)	[S2](文字数)が範囲外の場合にセットします

# LEFT (文字列左側取り出し)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

	説明
S1	取り出し元の先頭デバイスアドレス
S2	取り出したい文字数(データ指定可能範囲:1~4096)
D	取り出し結果を格納する先頭デバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *1	K	U	H	SF	DF	""		
S1	●	●	●	●			●	●	●													●
S1	●	●	●	●	●	●	●	●	●	●	●				●	●	●					●
D	●	●	●	●			●	●	●													●

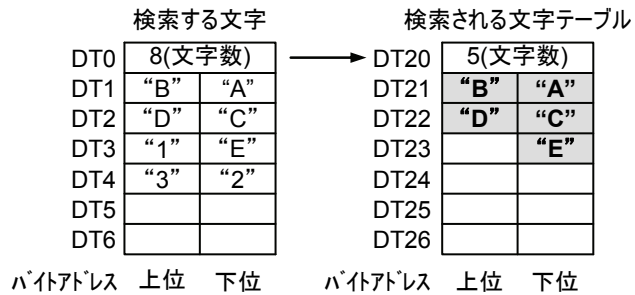
## ■ 動作説明

- [S1]で指定された文字列の左側(文字データの先頭)から[S2]で指定した文字数分を取り出し、[D]で指定したデバイスアドレスに格納します。

## ■ 処理内容

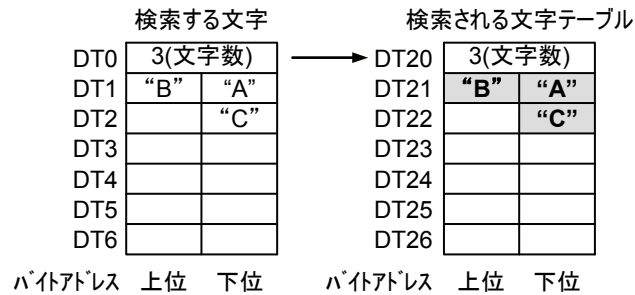
例1) DT0の文字列の最初の5文字を取り出し、DT20へ転送する場合

[S1]...DT0 [S2]...K5 [D]...DT20



例2) [S2]の文字数 > [S1]の文字列の文字数の場合

[S1]...DT0 [S2]...K7 [D]...DT20



## ■ プログラム上のご注意

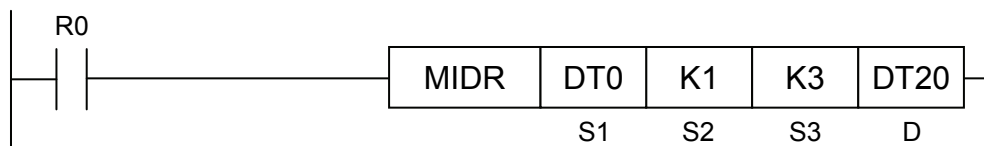
- 演算される前の[D]のエリアの文字データは、上書きされます。
- [S2]の文字数 > [S1]の文字列の文字数の場合、[S1]の文字数分転送します。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	転送先範囲がアクセス可能範囲外の場合にセットします
(ER)	[S2](文字数)が範囲外の場合にセットします

# MIDR (文字列任意位置からの読み出し)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●: 指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S1	取り出し元の先頭デバイスアドレス
S2	開始位置(データ指定可能範囲:0~4095)
S3	取り出したい文字数(データ指定可能範囲:1~4096)
D	取り出し結果を格納する先頭デバイスアドレス

## ■ 指定可能なデバイス (●: 指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデックス 修飾
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *1	K	U	H	SF	DF	""	
S1	●	●	●	●			●	●	●												●
S2	●	●	●	●	●	●	●	●	●	●	●				●	●	●				●
S3	●	●	●	●	●	●	●	●	●	●	●				●	●	●				●
D	●	●	●	●			●	●	●												●

## ■ 動作説明

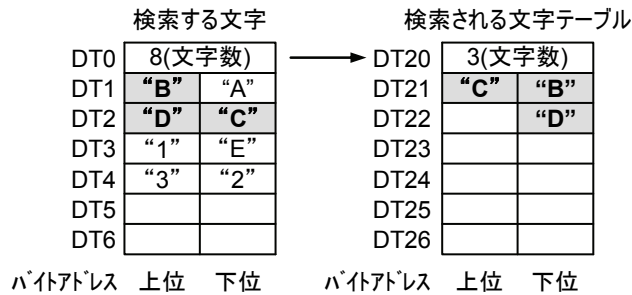
- [S1]で指定した文字列の[S2]で指定した位置から[S3]で指定した文字数分を取り出し、[D]で指定したデバイスアドレスに格納します。



## ■ 処理内容

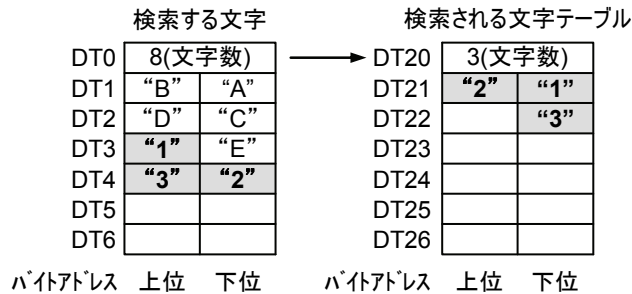
例1) DT0の文字列の1バイト目(2文字目)から3文字を取り出し、DT20へ転送する場合

[S1]...DT0 [S2]...K1 [S3]...K3 [D]...DT20



例2) [S3]の文字数 > [S2]の位置から[S1]の文字列の文字数の場合

[S1]...DT0 [S2]...K5 [S3]...K5 [D]...DT20



## ■ プログラム上のご注意

- 演算される前の[D]のエリアの文字データは、上書きされます。
- [S3]の文字数 > [S2]の位置から[S1]の文字列の文字数の場合、[S1]の文字数分転送します。
- [S2]の位置は、最下位バイトを K0(バイト 0)とし、下位から順番に 0,1,2...と数えて指定します。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合に、セットします
	転送先範囲がアクセス可能範囲外の場合にセットします
	[S3](文字数)が範囲外の場合にセットします
	[S1]の文字数 < [S2]の場合にセットします

# MIDW (文字列任意位置からの書き換え)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S1	取り出し元の先頭デバイスアドレス
S2	文字数(データ指定可能範囲:1~4096)
D	転送先の先頭デバイスアドレス
n	転送先文字列の開始位置(データ指定可能範囲:0~4095)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデックス 修飾
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *1	K	U	H	SF	DF	""	
S1	●	●	●	●			●	●	●												●
S2	●	●	●	●	●	●	●	●	●	●	●				●	●	●				●
D	●	●	●	●			●	●	●		●										●
n	●	●	●	●	●	●	●	●	●	●	●				●	●	●				●

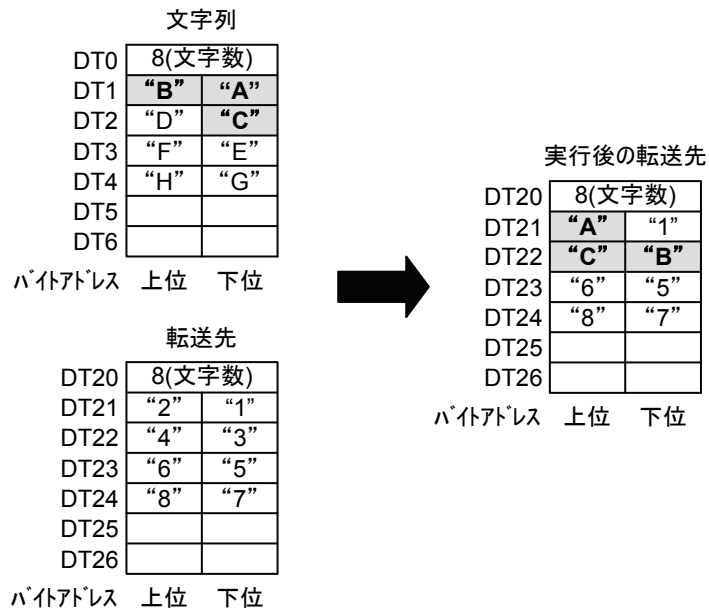
## ■ 動作説明

- [S1]で指定した文字列から[S2]で指定した文字数分を取り出し、[D]で指定した文字列の[n]の位置へ転送します。

## ■ 処理内容

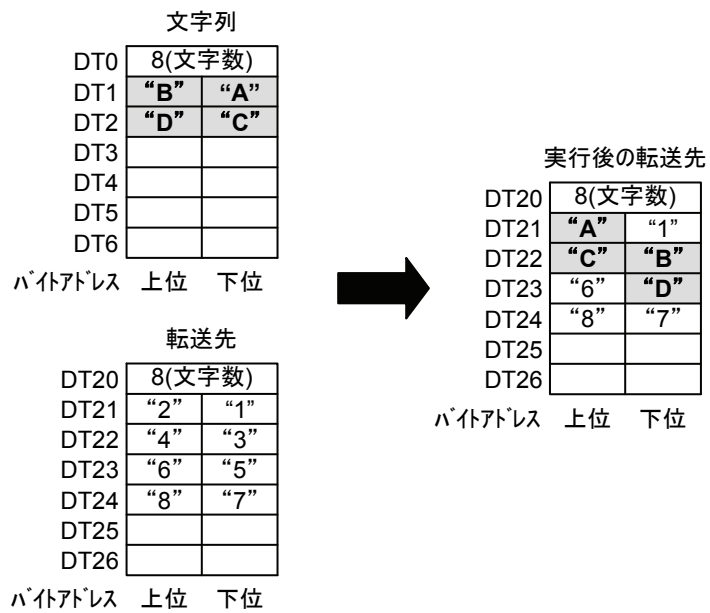
例1) DT0の文字列から3文字取り出し、DT20の文字列の1バイト目(2文字目)へ転送する場合

[S1]...DT0 [S2]...K3 [D]...DT20 [n]...K1



例2) [S2]の文字数>[S1]の文字列の文字数の場合

[S1]...DT0 [S2]...K5 [D]...DT20 [n]...K1



## ■ プログラム上のご注意

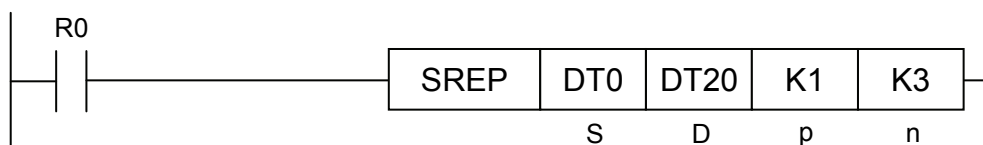
- 演算される前の[D]のエリアの文字データは、上書きされます。
- [S2]の文字数 > [S1]の文字列の文字数の場合、[S1]の文字列の文字数分を転送します。
- [n]の位置は、最下位バイトを K0(バイト 0)とし、下位から順番に 0,1,2...と数えて指定します。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S2](文字数)が範囲外の場合にセットします
(ER)	[D]の文字数 < [n]の場合にセットします

# SREP (文字列置き換え)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S	置換元文字列の先頭デバイスアドレス
D	置換先文字列の先頭デバイスアドレス
p	置換先文字列の置き換え開始位置(データ指定可能範囲:0~4095)
n	置き換え文字数(データ指定可能範囲:1~4096)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""			
S	●	●	●	●			●	●	●													●	
D	●	●	●	●			●	●	●														●
p	●	●	●	●	●	●	●	●	●	●	●				●	●	●						●
n	●	●	●	●	●	●	●	●	●	●	●				●	●	●						●

## ■ 動作説明

- [S]で指定した文字列を、[D]で指定した文字列の[p]の位置から[n]で指定した文字数分に置き換えます。

## ■ 処理内容

例1) DT0の文字列をDT20の1バイト目(2文字目)から3文字分と置き換える場合

[S1]...DT0 [D]...DT20 [p]...K1 [n]...K3

文字列

DT0	5(文字数)	
DT1	"B"	"A"
DT2	"D"	"C"
DT3		"E"
DT4		
DT5		
DT6		

バイトアドレス 上位 下位

転送先

DT20	8(文字数)	
DT21	"2"	"1"
DT22	"4"	"3"
DT23	"6"	"5"
DT24	"8"	"7"
DT25		
DT26		

バイトアドレス 上位 下位

実行後の転送先

DT20	10(文字数)	
DT21	"A"	"1"
DT22	"C"	"B"
DT23	"E"	"D"
DT24	"6"	"5"
DT25	"8"	"7"
DT26		

バイトアドレス 上位 下位

※この場合、置換先となった元々DT20の  
1バイト目～4バイト目までのデータは削除されます  
結果、文字数は8文字から10文字に増加します

例2) [n]の文字数が[p]で指定された以降の[S1]の文字列の文字数より大きい場合

[S1]...DT0 [D]...DT20 [p]...K4 [n]...K8

文字列

DT0	5(文字数)	
DT1	"B"	"A"
DT2	"D"	"C"
DT3		"E"
DT4		
DT5		
DT6		

バイトアドレス 上位 下位

転送先

DT20	10(文字数)	
DT21	"2"	"1"
DT22	"4"	"3"
DT23	"6"	"5"
DT24	"8"	"7"
DT25	"0"	"9"
DT26		

バイトアドレス 上位 下位

実行後の転送先

DT20	9(文字数)	
DT21	"2"	"1"
DT22	"4"	"3"
DT23	"B"	"A"
DT24	"D"	"C"
DT25		"E"
DT26		

バイトアドレス 上位 下位

## ■ プログラム上のご注意

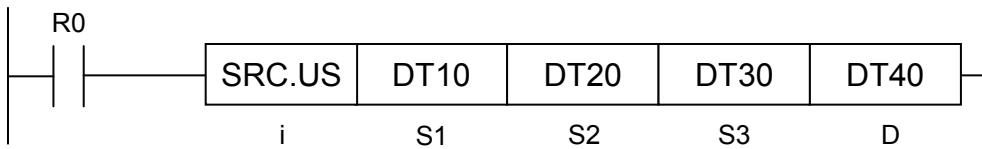
- 演算される前の[D]のエリアの文字データは、クリアされません。(上書きされます)
- [n]の文字数が[p]で指定された以降の[S1]の文字列の文字数より大きい場合、[p]で指定された以降の[S1]の文字列の文字数分が置き換えされます。
- [p]の位置は、最下位バイトを K0(バイト 0)とし、下位から順番に 0,1,2・・・と数えて指定します。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	文字数 > 文字列サイズの場合にセットします
	[D]の文字数 < [n]の場合にセットします
	[S2](文字数)が範囲外の場合にセットします

# SRC (データサーチ)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	サーチデータが格納されているデバイスアドレスまたは定数(データ形式:演算単位に従う)
S2	サーチ範囲の開始位置(データ形式:演算単位に従う)
S3	サーチ範囲の終了位置(データ形式:演算単位に従う)
D	サーチ結果を格納するデバイスアドレス(データ形式:符号なし 32 ビット整数)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8	..	
S1	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●			●	●	●	●	●	●	●	●							●
S3	●	●	●	●			●	●	●	●	●	●	●	●							●
D	●	●	●	●			●	●	●		●	●	●	●							●

- \*1: 演算単位が 16 ビット整数(SS, US)の場合は指定不可
- \*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)
- \*3: インデックスレジスタ(I0~IE)
- \*4: 演算単位が符号付き整数(SS, SL)の場合のみ指定可能
- \*5: 演算単位が符号なし整数(US, UL)の場合のみ指定可能
- \*6: 演算単位が整数(US, SS, UL, SL)の場合のみ指定可能
- \*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能
- \*8: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [S1]で指定したサーチデータを[S2]と[S3]で指定した範囲のエリアを検索します。
- サーチ結果は、データ形式を符合なし 32 ビット整数とし下記のように格納します。

16ビットデバイス	32ビットデバイス	出力内容
[D]、[D]+1	[D]	同じ値を持つデータの個数を10進数で格納
[D]+2、[D]+3	[D]+1	最初に一致したデータの位置 (最初のデータを0とした相対位置)を格納

- 指定できるデータ数の上限は 30000 データとなります。
- サーチは、[S2]⇒[S3]の方向で行われます。

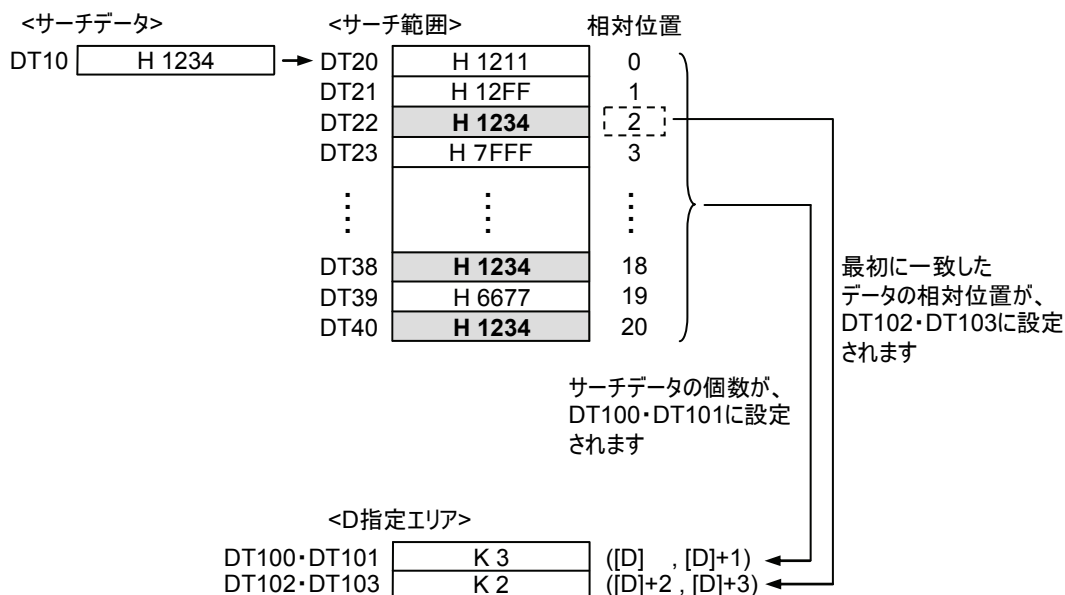


## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

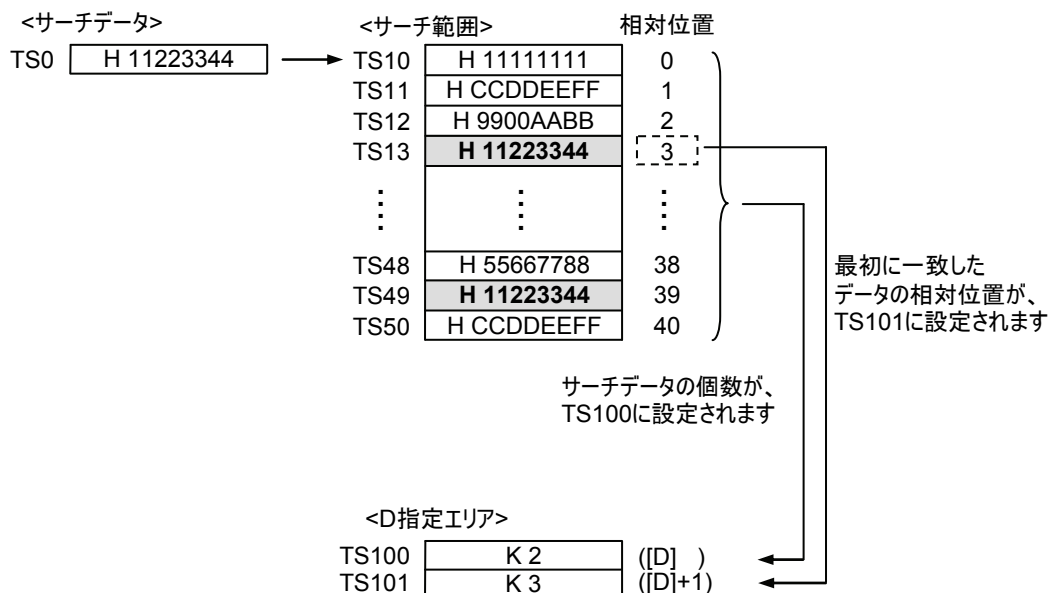
[S1]...DT10 [S2]...DT20 [S3]...DT40 [D]...DT100



例2) 演算単位が32ビット(UL,SL,SF)の場合

[i]...UL,SL,SF

[S1]...TS0 [S2]...TS10 [S3]...TS50 [D]...TS100



## ■ プログラム上のご注意

- サーチ範囲の終了位置は[S3]が含まれるデバイスまでをサーチ対象とします。

例) 演算単位に32ビットを指定した場合、[S3]のデバイスアドレスに上位アドレス、  
下位アドレスのいずれを指定しても検索範囲は同じになります

[S2]...DT2    [S3]...DT6                      [S2]...DT2    [S3]...DT6

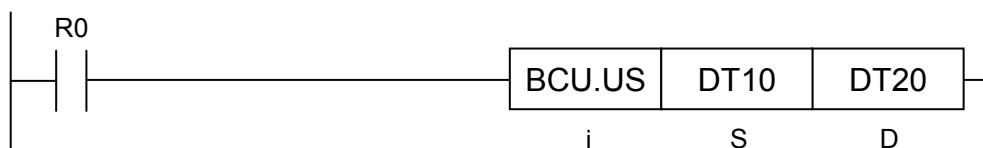
DT0・DT1	H 11223344	} サーチ対象	DT0・DT1	H 11223344	} サーチ対象
DT2・DT3	<b>H 55667788</b>		DT2・DT3	<b>H 55667788</b>	
DT4・DT5	<b>H 9900AABB</b>		DT4・DT5	<b>H 9900AABB</b>	
<b>DT6・DT7</b>	<b>H CCDDEEFF</b>		<b>DT6・DT7</b>	<b>H CCDDEEFF</b>	
DT8・DT9	H 12345678		DT8・DT9	H 12345678	

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾・ポインタアクセス)に範囲外となった場合にセットします
SR8	[S2] > [S3]の場合、セットします
(ER)	[S2]と[S3]が異なるエリアの場合、セットします

# BCU (ONビットカウント)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●		●			

## ■ オペランド一覧

オペランド	説明
S	対象のデバイスアドレスまたは定数(データ形式:演算単位に従う)
D	結果を格納するデバイスアドレス(データ形式:符号なし 16ビット整数)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス*1			整数			実数		文字	インデック ス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""		
S	●	●	●	●	●	●	●	●	●	●	●	●	●									●
D	●	●	●	●			●	●	●		●											●

\*1: 演算単位が 16ビット整数(SS、US)の場合は指定不可

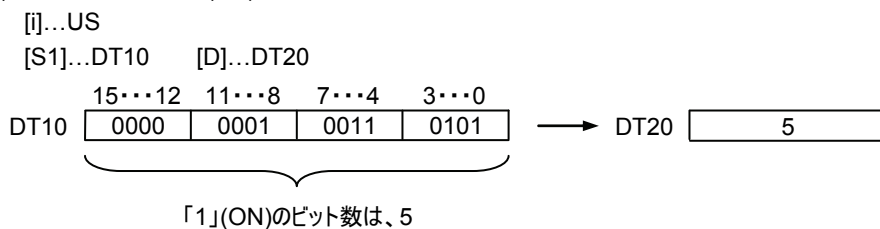
\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

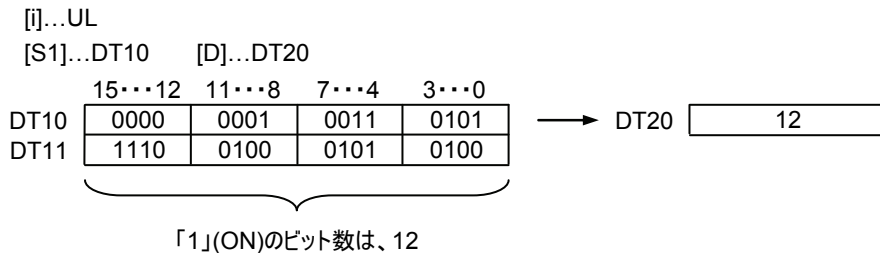
- [S]で指定されたデータ中の ON ビット(値が 1 のビット)の数をカウントし、カウント結果を[D]で指定したデバイスアドレスに格納します。
- 結果は符号なし 16 ビット整数で格納します。

## ■ 処理内容

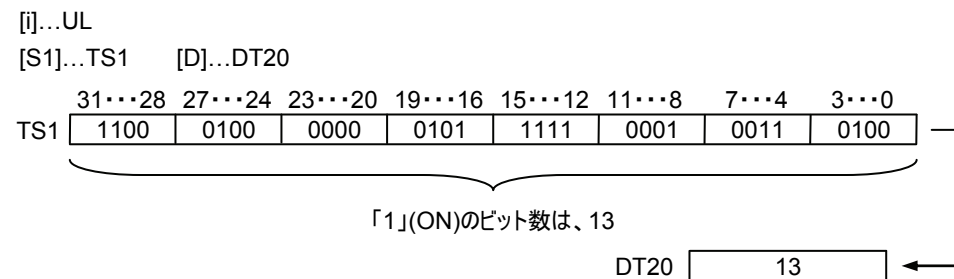
例1) 演算単位が16ビット(US)の場合



例2) 演算単位が32ビット(UL)の場合(デバイスに16ビットデバイスを指定)



例3) 演算単位が32ビット(UL)の場合(デバイスに32ビットデバイスを指定)

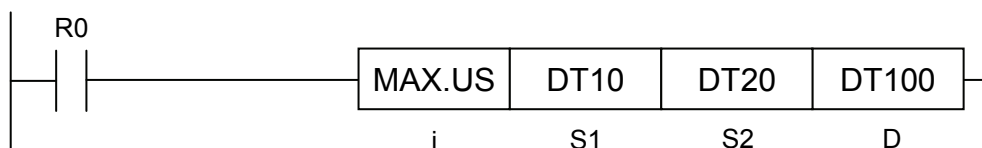


## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾・ポインタアクセス)に範囲外となった場合にセットします

# MAX (最大値取得)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	最大値検索範囲の開始位置(データ形式:演算単位に従う)
S2	最大値検索範囲の終了位置(データ形式:演算単位に従う)
D	最大値検索結果を格納するデバイスアドレス(データ形式:演算単位別)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●	●	●	●	●	●							●
S2	●	●	●	●			●	●	●	●	●	●	●	●							●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [S1]と[S2]で指定されたデバイスエリアの範囲内で、最大値を検索します。
- 検索された値は、[D]で指定したデバイスエリアに格納し、[n]に[S1]からの相対アドレス値を格納します。
- 相対アドレス格納位置([D]+[n])は、演算単位により異なります。
- 指定できるデータ数の上限は 30000 データとなります。
- [D]は、演算単位別に下記形式となります。

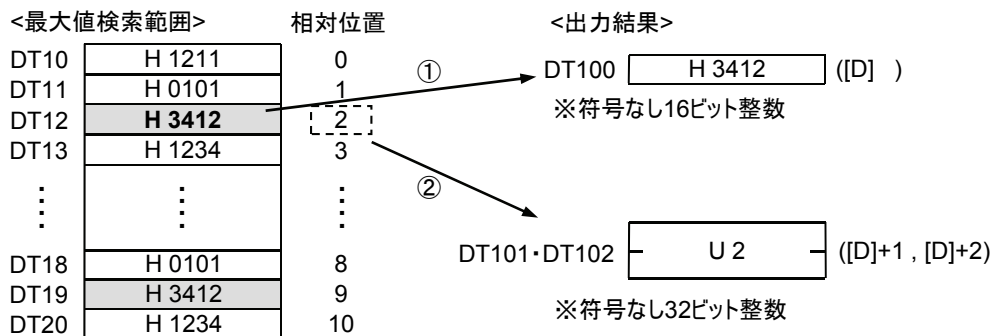
演算単位	16ビット (US, SS)		32ビット (UL, SL, SF)		64ビット (DF)		出力内容
	16ビット	32ビット	16ビット	32ビット	16ビット	32ビット	
結果格納 エリア	[D]	指定不可	[D] ~[D+1]	[D]	[D] ~[D+3]	[D] ~[D+1]	最大値を格納
	[D+1] ~[D+2]	指定不可	[D+2] ~[D+3]	[D+1]	[D+4] ~[D+5]	[D+2]	最初に検出した最大値の位置。[S1]先頭を 0 としたt相対位置。

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[I]...US

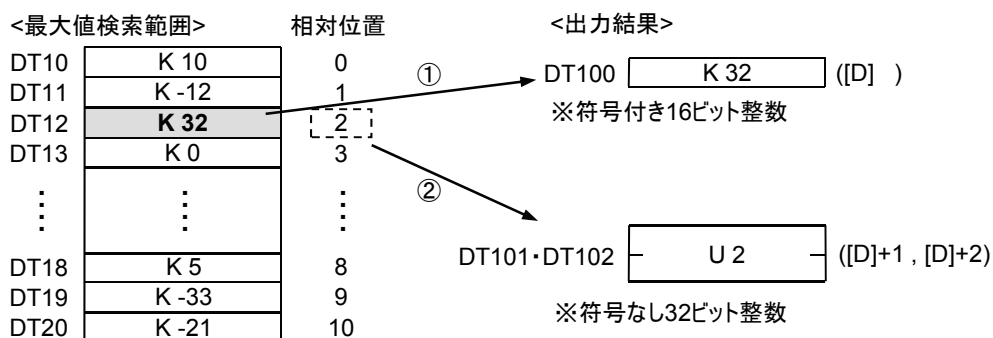
[S1]...DT10 [S2]...DT20 [D]...DT100



例2) 演算単位が16ビット(SS)の場合

[I]...SS

[S1]...DT10 [S2]...DT20 [D]...DT100



例3) 演算単位が32ビット(UL)の場合(32ビットデバイスを指定)

[i]...UL

[S1]...TS10 [S2]...TS50 [D]...TS100

<最大値検索範囲>		相対位置	<出力結果>
TS10	H 11111111	0	TS100 <span style="border: 1px solid black; padding: 2px;">H FFFFFFFF</span> ([D] ) ※符号なし32ビット整数
TS11	H CCDDEEFF	1	
TS12	H 9900AABB	2	
TS13	H 1234	3	
⋮	⋮	⋮	
TS48	H 55667788	38	TS101 <span style="border: 1px solid black; padding: 2px;">U 39</span> ([D]+1) ※符号なし32ビット整数
TS49	H FFFFFFFF	39	
TS50	H 9900AABB	40	

例4) 演算単位が32ビット(SL)の場合(32ビットデバイスを指定)

[i]...SL

[S1]...TS10 [S2]...TS50 [D]...TS100

<最大値検索範囲>		相対位置	<出力結果>
TS10	K 200	0	TS100 <span style="border: 1px solid black; padding: 2px;">K 300</span> ([D] ) ※符号付き16ビット整数
TS11	K -100	1	
TS12	K 100	2	
TS13	K 0	3	
⋮	⋮	⋮	
TS48	K 300	38	TS101 <span style="border: 1px solid black; padding: 2px;">U 38</span> ([D]+1) ※符号なし32ビット整数
TS49	K -300	39	
TS50	K -400	40	

例5) 演算単位が32ビット(UL)の場合(16ビットデバイスを指定)

[i]...UL

[S1]...DT10 [S2]...DT30 [D]...DT100

<最大値検索範囲>		相対位置	<出力結果>
DT10・DT11	H 1234FFFF	0	DT100・DT101 <span style="border: 1px solid black; padding: 2px;">H AAAA0000</span> ([D] , [D]+1 ) ※符号なし32ビット整数
DT12・DT13	H 789A3456	1	
DT14・DT15	H 9900AABB	2	
DT16・DT17	H 12345678	3	
⋮	⋮	⋮	
DT26・DT27	H 55667788	8	DT102・DT103 <span style="border: 1px solid black; padding: 2px;">U 10</span> ([D]+2 , [D]+3) ※符号なし32ビット整数
DT28・DT29	H 9900AABB	9	
DT30・DT31	H AAAA0000	10	

例6) 演算単位が32ビット(SL)の場合(16ビットデバイスを指定)

[I]...SL

[S1]...DT10 [S2]...DT30 [D]...DT100

<最大値検索範囲>		相対位置	<出力結果>	
DT10・DT11	K -10	0	DT100・DT101	K 200 (([D] , [D]+1)
DT12・DT13	K -20	1		※符号付き32ビット整数
DT14・DT15	K 100	2		
DT16・DT17	K 0	3		
⋮	⋮	⋮		
DT26・DT27	<b>K 200</b>	8	DT102・DT103	U 8 (([D]+2 , [D]+3)
DT28・DT29	K -300	9		※符号なし32ビット整数
DT30・DT31	K -400	10		

例7) 演算単位が単精度浮動小数点形実数(SF)の場合

[I]...SF

[S1]...DT10 [S2]...DT90 [D]...DT100

<最大値検索範囲>		相対位置	<出力結果>	
DT10・DT11	SF 1.11	0	DT100・DT101	SF 5.55 (([D] , [D]+1)
DT12・DT13	SF 3.33	1		※単精度浮動小数点形実数(32ビット)
DT14・DT15	SF 2.22	2		
DT16・DT17	SF 4.44	3		
⋮	⋮	⋮		
DT86・DT87	<b>SF 5.55</b>	38	DT102・DT103	U 38 (([D]+2 , [D]+3)
DT88・DT89	SF 1.11	39		※符号なし32ビット整数
DT90・DT91	SF 3.33	40		

■ プログラム上のご注意

- 最大値検索範囲の終了位置は[S2]が含まれるデバイスまでを検索対象とします。

例) 演算単位に32ビットを指定した場合、[S2]のデバイスアドレスに上位アドレス、下位アドレスのいずれを指定しても最大値検索範囲は同じになります

[S1]...DT2	[S2]...DT6	[S1]...DT2	[S2]...DT7
DT0・DT1	H 11223344	DT0・DT1	H 11223344
DT2・DT3	<b>H 55667788</b>	DT2・DT3	<b>H 55667788</b>
DT4・DT5	<b>H 9900AABB</b>	DT4・DT5	<b>H 9900AABB</b>
DT6・DT7	<b>H CCDDEEFF</b>	DT6・DT7	<b>H CCDDEEFF</b>
DT8・DT9	H 12345678	DT8・DT9	H 12345678

- [D](最大値検索結果)を最大値検索範囲内に指定するとデータを上書きします。

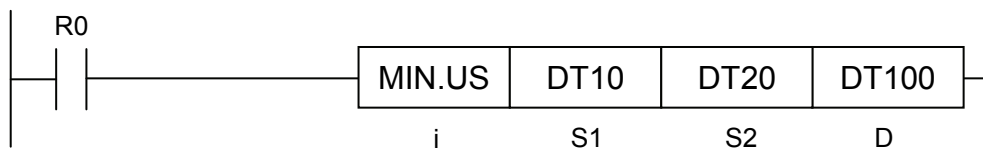
■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾・ポインタアクセス)に範囲外となった場合にセットします
SR8	[S1] > [S2] の場合にセットします
(ER)	[S1] と [S2] のデバイスが異なる場合にセットします



# MIN (最小値取得)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	最小値検索範囲の開始位置(データ形式:演算単位に従う)
S2	最小値検索範囲の終了位置(データ形式:演算単位に従う)
D	最小値検索結果を格納するデバイスアドレス(データ形式:演算単位別)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●	●	●	●	●	●							●
S2	●	●	●	●			●	●	●	●	●	●	●	●							●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16ビット整数(SS, US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [S1]と[S2]で指定されたデバイスエリアの範囲内で最小値を検索し、[D]で指定したデバイスエリアに格納し、[D]+[n]に[S1]からの相対アドレス値を格納します。
- 相対アドレス格納位置([D]+[n])は、演算単位により異なります。
- 指定できるデータ数の上限は 30,000 データとなります。
- [D]は、演算単位別に下記形式となります。

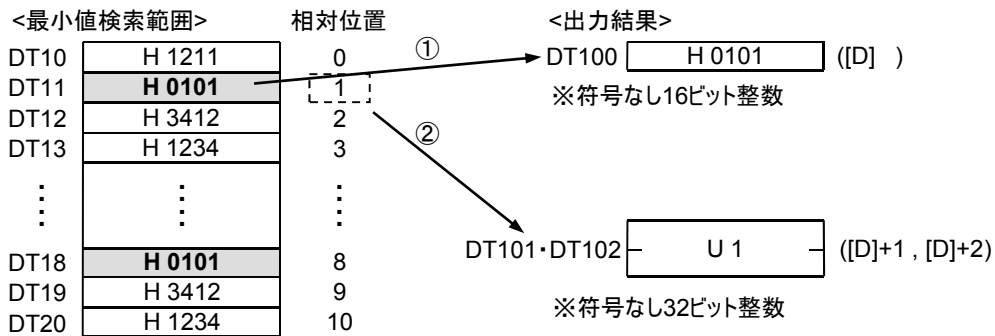
演算単位	16ビット (US, SS)		32ビット (UL, SL, SF)		64ビット (DF)		出力内容
	16ビット	32ビット	16ビット	32ビット	16ビット	32ビット	
結果格納 エリア	[D]	指定不可	[D] ~[D+1]	[D]	[D] ~[D+3]	[D] ~[D+1]	最小値を格納
	[D+1] ~[D+2]	指定不可	[D+2] ~[D+3]	[D+1]	[D+4] ~[D+5]	[D+2]	最初に検出した最大値の位置。[S1]先頭を 0 としたt相対位置。

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[i]...US

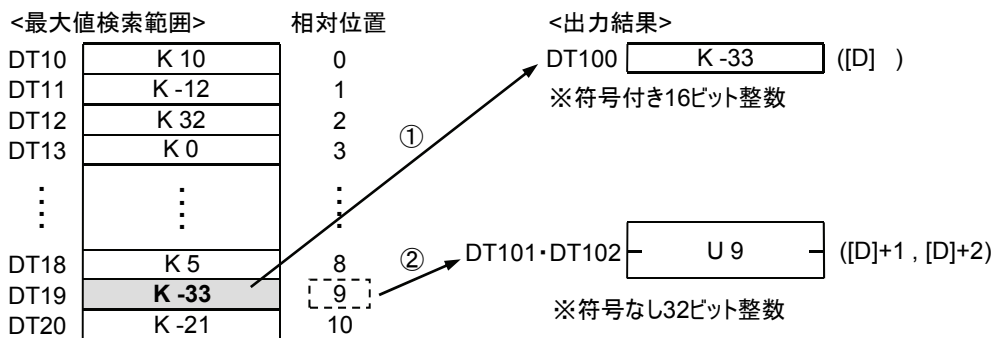
[S1]...DT10 [S2]...DT20 [D]...DT100



例2) 演算単位が16ビット(SS)の場合

[i]...SS

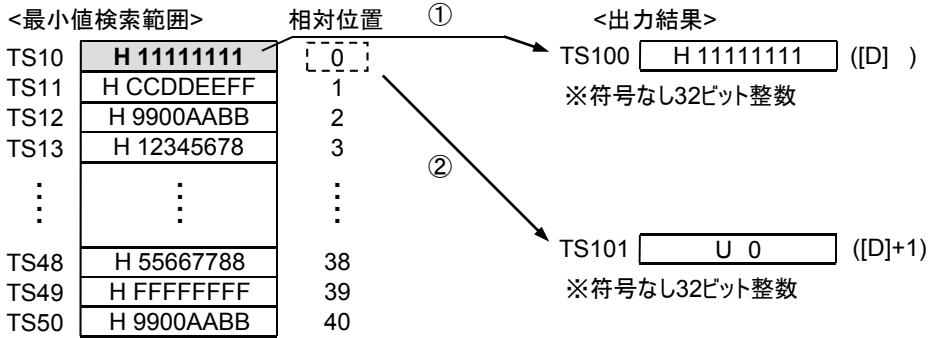
[S1]...DT10 [S2]...DT20 [D]...DT100



例3) 演算単位が32ビット(UL)の場合(32ビットデバイスを指定)

[i]...UL

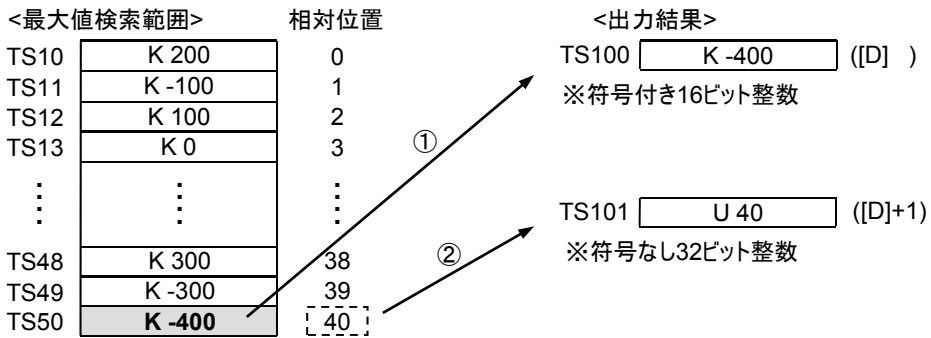
[S1]...TS10 [S2]...TS50 [D]...TS100



例4) 演算単位が32ビット(SL)の場合(32ビットデバイスを指定)

[i]...SL

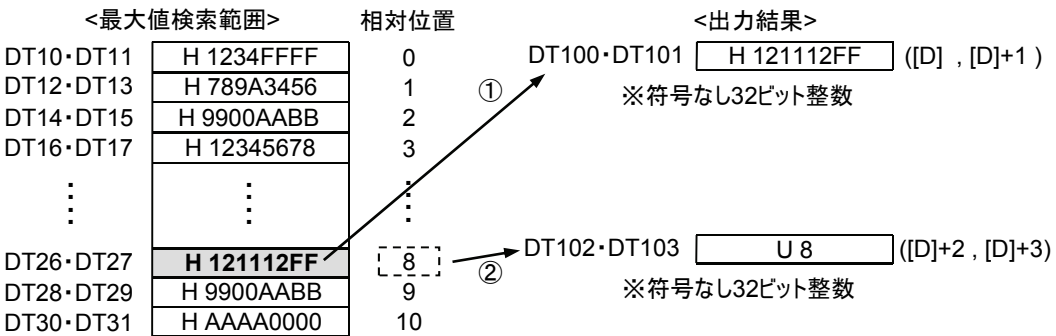
[S1]...TS10 [S2]...TS50 [D]...TS100



例5) 演算単位が32ビット(UL)の場合(16ビットデバイスを指定)

[i]...UL

[S1]...DT10 [S2]...DT30 [D]...DT100



例6) 演算単位が32ビット(SL)の場合(16ビットデバイスを指定)

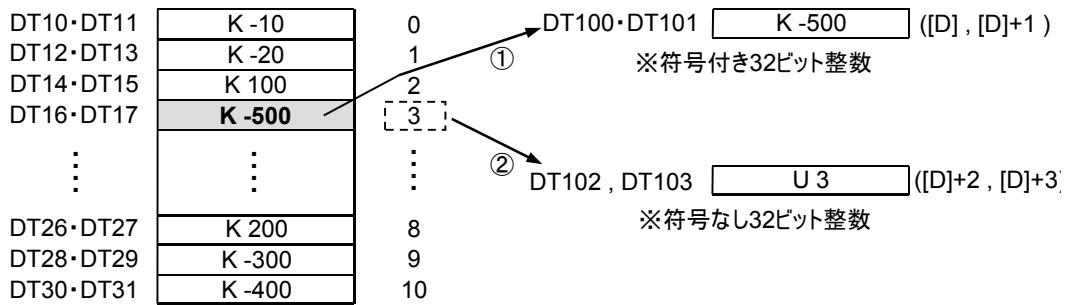
[I]...SL

[S1]...DT10 [S2]...DT30 [D]...DT100

<最大値検索範囲>

相対位置

<出力結果>



例7) 演算単位が単精度浮動小数点形実数(SF)の場合

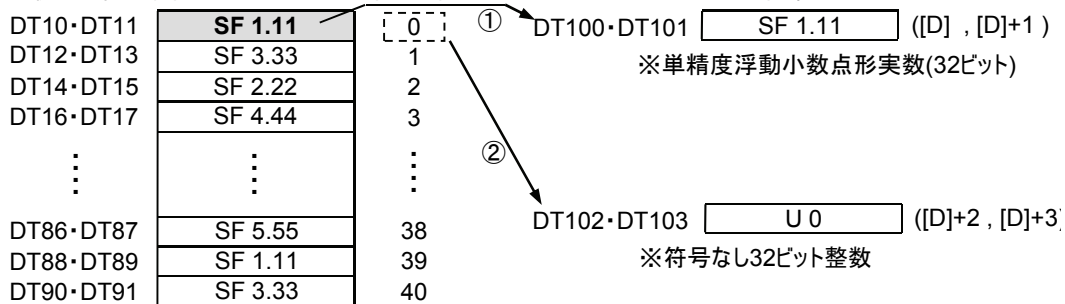
[I]...SF

[S1]...DT10 [S2]...DT90 [D]...DT100

<最小値検索範囲>

相対位置

<出力結果>



## ■ プログラム上のご注意

- 最小値検索範囲の終了位置は[S2]が含まれるデバイスまでを検索対象とします。

例) 演算単位に32ビットを指定した場合、[S2]のデバイスアドレスに上位アドレス、  
下位アドレスのいずれを指定しても最小値検索範囲は同じになります

[S1]...DT2	[S2]...DT6	[S1]...DT2	[S2]...DT7																				
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>DT0・DT1</td><td>H 11223344</td></tr> <tr><td>DT2・DT3</td><td><b>H 55667788</b></td></tr> <tr><td>DT4・DT5</td><td><b>H 9900AABB</b></td></tr> <tr><td>DT6・DT7</td><td><b>H CCDDEEFF</b></td></tr> <tr><td>DT8・DT9</td><td>H 12345678</td></tr> </table>	DT0・DT1	H 11223344	DT2・DT3	<b>H 55667788</b>	DT4・DT5	<b>H 9900AABB</b>	DT6・DT7	<b>H CCDDEEFF</b>	DT8・DT9	H 12345678	} 最小値検索範囲	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>DT0・DT1</td><td>H 11223344</td></tr> <tr><td>DT2・DT3</td><td><b>H 55667788</b></td></tr> <tr><td>DT4・DT5</td><td><b>H 9900AABB</b></td></tr> <tr><td>DT6・DT7</td><td><b>H CCDDEEFF</b></td></tr> <tr><td>DT8・DT9</td><td>H 12345678</td></tr> </table>	DT0・DT1	H 11223344	DT2・DT3	<b>H 55667788</b>	DT4・DT5	<b>H 9900AABB</b>	DT6・DT7	<b>H CCDDEEFF</b>	DT8・DT9	H 12345678	} 最小値検索範囲
DT0・DT1	H 11223344																						
DT2・DT3	<b>H 55667788</b>																						
DT4・DT5	<b>H 9900AABB</b>																						
DT6・DT7	<b>H CCDDEEFF</b>																						
DT8・DT9	H 12345678																						
DT0・DT1	H 11223344																						
DT2・DT3	<b>H 55667788</b>																						
DT4・DT5	<b>H 9900AABB</b>																						
DT6・DT7	<b>H CCDDEEFF</b>																						
DT8・DT9	H 12345678																						

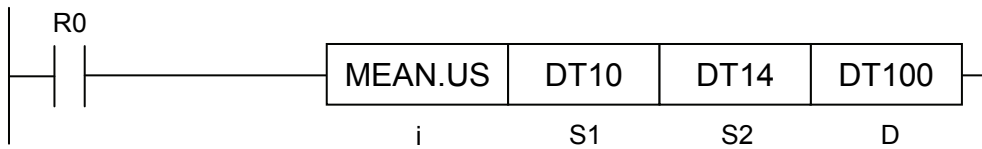
- [D](最小値検索結果)を最大値検索範囲内に指定するとデータを上書きします。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾・ポインタアクセス)に範囲外となった場合にセットします
SR8	[S1] > [S2] の場合にセットします
(ER)	[S1] と [S2] のデバイスが異なる場合にセットします

# MEAN (合計・平均値取得)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	対象エリアの開始位置(データ形式:演算単位に従う)
S2	対象エリアの終了位置(データ形式:演算単位に従う)
D	結果を格納するデバイスアドレス(データ形式:演算単位別)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	""		
S1	●	●	●	●			●	●	●	●	●	●	●	●								●
S2	●	●	●	●			●	●	●	●	●	●	●	●								●
D	●	●	●	●			●	●	●		●	●	●	●								●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- [S1]と[S2]で指定されたデバイスエリアの範囲内の合計値及び平均値を、[D]で指定したデバイスエリアに格納します。
- [D]を先頭アドレスとする合計値・平均値の格納位置は演算単位により異なります。
- 合計値は演算単位の 2 倍のエリアサイズ、平均値は演算単位と同じエリアサイズを使用します。但し、浮動小数点実数(SF)の場合は合計値も演算単位と同じエリアサイズとなります。
- 指定できるデータ数の上限は、30,000 データとなります。
- 平均値は演算単位が US SS UL SL の場合、小数点以下が切り捨てられた整数となります。
- [D]は、演算単位別に下記形式となります。

演算単位	16ビット(US, SS)		32ビット(UL, SL)		32ビット(SF)		64ビット(DF)		出力内容
	16ビット	32ビット	16ビット	32ビット	16ビット	32ビット	16ビット	32ビット	
結果格納 エリア	[D] ~[D+1]	指定不可	[D] ~[D+3]	[D] ~[D+1]	[D] ~[D+1]	[D]	[D] ~[D+3]	[D] ~[D+1]	合計値
	[D+2]	指定不可	[D+4] ~[D+5]	[D+2]	[D+2] ~[D+3]	[D+1]	[D+4] ~[D+7]	[D+2] ~[D+3]	平均値

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合  
合計値は32ビットデータ、平均値は16ビットデータとなります

[i]...US

[S1]...DT10 [S2]...DT14 [D]...DT100

<合計・平均計算範囲>

	値
DT10	H 1111
DT11	H 5555
DT12	H 7777
DT13	H AAAA
DT14	H FFFF



<出力結果>

DT100・DT101 

値
H 00028886

 合計値([D]、[D]+1)  
※符号なし32ビット整数

DT102 

値
H 81B4

 平均値([D]+2)  
※符号なし16ビット整数

例2) 演算単位が16ビット(SS)の場合  
合計値は32ビットデータ、平均値は16ビットデータとなります

[i]...SS

[S1]...DT10 [S2]...DT14 [D]...DT100

<合計・平均計算範囲>

	値
DT10	K 11
DT11	K -33
DT12	K 44
DT13	K 55
DT14	K -22



<出力結果>

DT100・DT101 

値
K -55

 合計値([D]、[D]+1)  
※符号付き32ビット整数

DT102 

値
K 11

 平均値([D]+2)  
※符号付き16ビット整数

例3) 演算単位が32ビット(UL)の場合(32ビットデバイスを指定)  
合計値は64ビットデータ、平均値は32ビットデータとなります

[i]...UL

[S1]...TS10 [S2]...TS14 [D]...TS100

<合計・平均計算範囲>

	値
TS10	H 00000000
TS11	H 22222222
TS12	H 33333333
TS13	H 44444444
TS14	H 55555555



<出力結果>

TS100・TS101 

値
H EEEEEEEE

 合計値([D]、[D]+1)  
※符号なし64ビット整数

TS102 

値
H 2FC962FC

 平均値([D]+2)  
※符号なし32ビット整数

例4) 演算単位が32ビット(SL)の場合(32ビットデバイスを指定)  
合計値は64ビットデータ、平均値は32ビットデータとなります

[i]...SL

[S1]...TS10 [S2]...TS14 [D]...TS100

<合計・平均計算範囲>

	値
TS10	K 1000
TS11	K 2000
TS12	K -3000
TS13	K -4000
TS14	K -5000



<出力結果>

	値	
TS100・TS101	<b>K -9000</b>	合計値([D]、[D]+1) ※符号付き64ビット整数
TS102	<b>K -1800</b>	平均値([D]+2) ※符号付き32ビット整数

例5) 演算単位が32ビット(UL)の場合(16ビットデバイスを指定)  
合計値は64ビットデータ、平均値は32ビットデータとなります

[i]...UL

[S1]...DT10 [S2]...DT14 [D]...DT100

<合計・平均計算範囲>

	値
DT10・DT11	H 11110000
DT12・DT13	H 33332222
DT14・DT15	H 55554444
DT16・DT17	H 77776666



<出力結果>

	値	
DT100・DT103	<b>H 11110CCCC</b>	合計値([D]～[D]+3) ※符号なし64ビット整数
DT104・DT105	<b>H 44443333</b>	平均値([D]+4、[D]+5) ※符号なし32ビット整数

例6) 演算単位が32ビット(SL)の場合(16ビットデバイスを指定)  
合計値は64ビットデータ、平均値は32ビットデータとなります

[i]...SL

[S1]...DT10 [S2]...DT16 [D]...DT100

<合計・平均計算範囲>

	値
DT10・DT11	K -100
DT12・DT13	K 600
DT14・DT15	K 500
DT16・DT17	K -200



<出力結果>

	値	
DT100・DT103	<b>K 800</b>	合計値([D]～[D]+3) ※符号付き64ビット整数
DT104・DT105	<b>K 200</b>	平均値([D]+4、[D]+5) ※符号付き32ビット整数



例7) 演算単位が単精度浮動小数点形実数(SF)の場合(16ビットデバイスを指定)  
**合計値は32ビットデータ、平均値は32ビットデータとなります**

[I]...SF

[S1]...DT10 [S2]...DT24 [D]...DT100

<合計・平均計算範囲>

	値
DT10・DT11	SF 3.33E+00
DT12・DT13	SF 1.11E+00
DT14・DT15	SF 4.44E+00
DT16・DT17	SF 5.55E+00
DT18・DT19	SF 2.22E+00
DT20・DT21	SF 1.11E+00
DT22・DT23	SF 3.33E+00
DT24・DT25	SF 5.55E+00



<出力結果>

	値	
DT100・DT101	<b>SF 2.66E+01</b>	合計値([D]、[D]+1) ※単精度浮動小数点形実数(32ビット)
DT102・DT103	<b>SF 3.33E+00</b>	平均値([D]+2、[D]+3) ※単精度浮動小数点形実数(32ビット)

## ■ プログラム上のご注意

- 合計・平均計算範囲の終了位置は[S2]が含まれるデバイスまでを計算対象とします。

例) 演算単位が 32 ビットるとき、上位、下位いずれのデバイス番号を指定した場合も計算範囲は同様になります。

[S1]...DT2 [S2]...DT6

DT0・DT1	H 11223344	} 合計・平均 計算範囲
DT2・DT3	<b>H 55667788</b>	
DT4・DT5	<b>H 9900AABB</b>	
DT6・DT7	<b>H CCDDEEFF</b>	
DT8・DT9	H 12345678	

[S1]...DT2 [S2]...DT7

DT0・DT1	H 11223344	} 合計・平均 計算範囲
DT2・DT3	<b>H 55667788</b>	
DT4・DT5	<b>H 9900AABB</b>	
DT6・DT7	<b>H CCDDEEFF</b>	
DT8・DT9	H 12345678	

- [D](合計・平均計算結果)を合計・平均計算範囲内に指定するとデータを上書きします。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾・ポインタアクセス)に範囲外となった場合にセットします
SR8	[S1] > [S2] の場合にセットします
(ER)	[S1] と [S2] のデバイスが異なる場合にセットします

# SORT (並び替え)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	対象エリアの開始位置(データ形式:演算単位に従う)
S2	対象エリアの終了位置(データ形式:演算単位に従う)
S3	ソート条件(データ形式:符号なし 16ビット整数)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "		
S1							●	●														●
S2							●	●														●
S3	●	●	●	●			●	●								●	●					●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- [S1]で指定されたエリアから[S2]で指定されたエリアの範囲内のデータを、[S3]で指定されたソート条件に従い昇順または、降順で並び替えます。
- [S3]に指定可能なソート条件には、以下を指定します。  
 U0:昇順  
 U1:降順

## ■ 処理内容

例1) 演算単位が16ビット(US)の場合

[i]...US

[S1]...DT10 [S2]...DT19 [S3]...U0 (昇順)

<ソート前>

DT10	H 0123
DT11	H 1111
DT12	H 3210
DT13	H 2222
DT14	H 3333
DT15	H 0000
DT16	H 3210
DT17	H 4321
DT18	H 3333
DT19	H 5432



<ソート後>

DT10	H 0000
DT11	H 0123
DT12	H 1111
DT13	H 2222
DT14	H 3210
DT15	H 3210
DT16	H 3333
DT17	H 3333
DT18	H 4321
DT19	H 5432



例2) 演算単位が16ビット(SS)の場合

[i]...SS

[S1]...DT10 [S2]...DT19 [S3]...U1 (降順)

<ソート前>

DT10	K 300
DT11	K10
DT12	K 3
DT13	K -1
DT14	K 1000
DT15	K -30
DT16	K 100
DT17	K 30
DT18	K 1
DT19	K -3



<ソート後>

DT10	K 1000
DT11	K 300
DT12	K 100
DT13	K 30
DT14	K 10
DT15	K 3
DT16	K 1
DT17	K -1
DT18	K -3
DT19	K 30



例3) 演算単位が32ビット(UL)の場合

[i]...UL

[S1]...DT10 [S2]...DT19 [S3]...U0 (昇順)

<ソート前>

DT10・DT11	H 22220000
DT12・DT13	H 11113333
DT14・DT15	H 55550000
DT16・DT17	H 22222222
DT18・DT19	H 11114444



<ソート後>

DT10・DT11	H 11113333
DT12・DT13	H 11114444
DT14・DT15	H 22220000
DT16・DT17	H 22222222
DT18・DT19	H 55550000



例4) 演算単位が32ビット(SL)の場合

[i]...SL

[S1]...DT10 [S2]...DT19 [S3]...U1 (降順)

<ソート前>

DT10・DT11	K 11
DT12・DT13	K 33
DT14・DT15	K 55
DT16・DT17	K 22
DT18・DT19	K 44



<ソート後>

DT10・DT11	K 55
DT12・DT13	K 44
DT14・DT15	K 33
DT16・DT17	K 22
DT18・DT19	K 11



例5) 演算単位が単精度浮動小数点形実数(SF)の場合

[I]...SF

[S1]...DT10 [S2]...DT19 [S3]...U1 (降順)

<ソート前>

DT10・DT11	SF 3.33
DT12・DT13	SF 11.11
DT14・DT15	SF 2.22
DT16・DT17	SF 1111.1
DT18・DT19	SF 4.44



<ソート後>

DT10・DT11	SF 1111.1
DT12・DT13	SF 11.11
DT14・DT15	SF 4.44
DT16・DT17	SF 3.33
DT18・DT19	SF 2.22



■ プログラム上のご注意

- データの比較時間は、データ数の 2 乗に比例して増加する為、ソートデータ数が多い場合は、実行時間が長くなりますので注意が必要です。
- ソート実行中の[S1]~[S2]のデータは、ソート条件に従い順次並び替えられます。
- ソート範囲の終了位置は[S2]が含まれるデバイスまでをソート対象とします。  
例) 演算単位が 32 ビットのととき、上位、下位いずれのデバイス番号を指定した場合も、ソートする範囲は同様になります。

[S1]...DT2 [S2]...DT6

DT0・DT1	H 11223344
DT2・DT3	H 55667788
DT4・DT5	H 9900AABB
DT6・DT7	H CCDDEEFF
DT8・DT9	H 12345678

} ソート対象

[S1]...DT2 [S2]...DT7

DT0・DT1	H 11223344
DT2・DT3	H 55667788
DT4・DT5	H 9900AABB
DT6・DT7	H CCDDEEFF
DT8・DT9	H 12345678

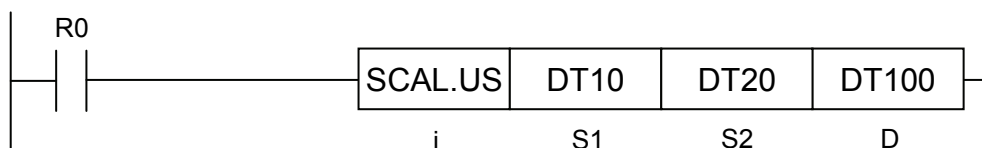
} ソート対象

■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾・ポインタアクセス)に範囲外となった場合にセットします
SR8	[S1] > [S2] の場合にセットします
(ER)	[S1] と [S2] のデバイスが異なる場合にセットします

# SCAL (線形化)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	入力値 X に相当するデータまたは格納しているエリア(データ形式:演算単位に従う)
S2	スケーリング(線形化)に使用するデータテーブルの先頭アドレス(データ形式:演算単位別)
D	出力結果 Y が格納されるエリア(データ形式:演算単位に従う)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス*1			整数			実数		文字	インデックス修飾*2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8	..	
S1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●		●
S2	●	●	●	●	●	●	●	●	●	●	●										●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16ビット整数(SS, US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS, SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US, UL)の場合のみ指定可能

\*6: 演算単位が整数(US, SS, UL, SL)の場合のみ指定可能

\*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*8: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

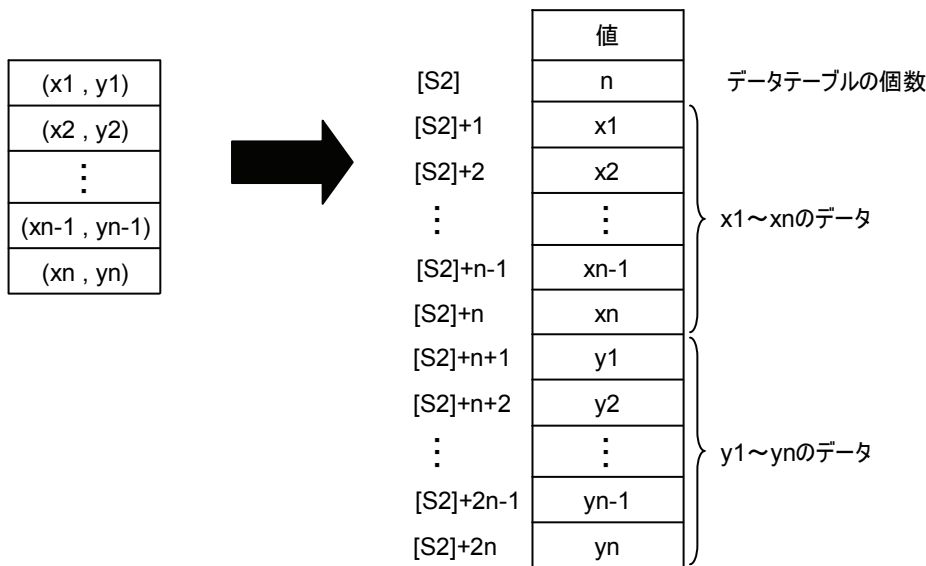
●[S1]で指定されたデータを[S2]で指定されたデータテーブルに従ってスケーリングを行い、[D]で指定したデバイスエリアに格納します。

●[S2]は、演算単位別に下記形式となります。

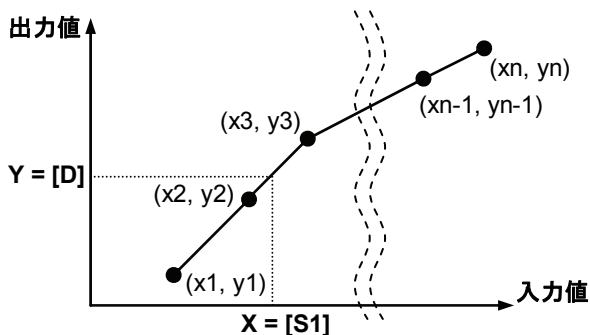
演算単位	16ビット(US, SS)		32ビット(UL, SL, SF)		64ビット(DF)		出力内容
	16ビット	32ビット	16ビット	32ビット	16ビット	32ビット	
結果格納 エリア	[S2]	指定不可	[S2]	指定不可	[S2]	指定不可	データテーブルの個数 指定可能範囲:2~256
	[S2+1]	指定不可	[S2+1] ~[S2+2]	指定不可	[S2+1] ~[S2+4]	指定不可	X1~Xn のデータ
	[S2+n+1]	指定不可	[S2+n+1] ~ [S2+n+2]	指定不可	[S2+n+1] ~ [S2+n+4]	指定不可	y1~yn のデータ

■ 処理内容

• 演算単位に関わらず、以下のように、[S1]の入力値 X に対する、[D]の出力値 Y を求めます。



[S1]で指定された入力値Xの情報に対し、[S2]のデータテーブルに従い、出力値Yの情報を算出し、[D]のエリアに格納します



<<演算単位(n)別のデバイス参照について>>

例) 演算単位が32ビット(UL,SL,SF)で16ビットデバイスを指定した場合

	値
DT10・DT11	H 22220000
DT12・DT13	H 11113333
DT14・DT15	H55550000
DT16・DT17	H 22222222
DT18・DT19	H 11114444

- 下位ワードと上位ワードをひとつの32ビットデータとして参照します
- 32ビットデータとして扱う為、デバイス指定がワード境界の場合、[S2]+1までが範囲となります

### ■ プログラム上のご注意

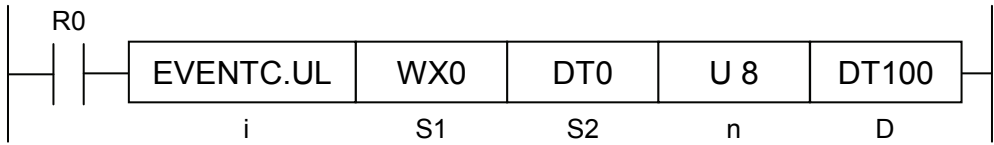
- [S2]: データテーブル内のデータについて、[  $X_{n-1} < X_n$  ]としてください。
- [  $X([S1]) < x_1$  ]の場合、[  $Y([D]) = y_1$  ]となります。
- [  $X([S1]) > x_n$  ]の場合、[  $Y([D]) = y_n$  ]となります。
- [S2]: データテーブル内のデータ個数である[n]の最大は **256** となります。

### ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾・ポインタアクセス)に範囲外となった場合にセットします
	[S2]: データテーブル内のデータ個数である[n]が、[ $n < 2$ ]または、[ $n > 256$ ]の場合にセットします
	[S2]: データテーブル内が、エリアを超えた場合にセットします
	$X_n$ が昇順になっていない場合にセットします

# EVENTC (事象発生回数計数命令)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i				●			

## ■ オペランド一覧

オペランド	説明
S1	計数開始位置先頭アドレス
S2	計数用作業領域の先頭アドレス
n	計数するビット数
D	計数結果を格納する先頭デバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデックス 修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""		
S1	●	●	●	●	●	●	●	●	●	●	●											●
S2	●	●	●	●			●	●														●
n	●	●	●	●			●	●	●	●	●					●	●					●
D	●	●	●	●			●	●	●	●	●											●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- [S1]で指定されたデバイスから[n]ビット分の ON 回数を計数します。
- 演算結果は、[D]を先頭とする[n]\*2ワードの領域に格納します。



## ■ 処理内容

例1) WX0から8点分の計数を行う場合

[i]...UL            [S1]...WX0            [S2]...DT0            [n]...U 8            [D]...DT10

[S1]  
WX0 

X0-X7 の 8ビットが対象ビット
--------------------

[S2]  
DT0 

計数用作業領域
---------

[D]		
DT10	X0 計数值	1
DT11		
DT12	X1 計数值	2
DT13		
DT14	X2 計数值	3
DT15		
DT16	X3 計数值	4
DT17		
DT18	X4 計数值	5
DT19		
DT20	X5 計数值	6
DT21		
DT22	X6 計数值	7
DT23		
DT24	X7 計数值	8
DT25		

例2) WX0から17点分の計数を行う場合

[i]...UL                    [S1]...WX0                    [S2]...DT0                    [n]...U 17                    [D]...DT10

[S1] WX0 WX1	X0 - X10 の17ビットが対象ビット
--------------------	-----------------------

[S2] DT0 DT1	計数作業領域
--------------------	--------

[D] DT10	X0 計数值	1
DT11		
DT12	X1 計数值	2
DT13		
DT14	X2 計数值	3
DT15		
DT16	X3 計数值	4
DT17		
DT18	X4 計数值	5
DT19		
DT20	X5 計数值	6
DT21		
DT22	X6 計数值	7
DT23		
DT24	X7 計数值	8
DT25		

●  
●  
●  
●  
●

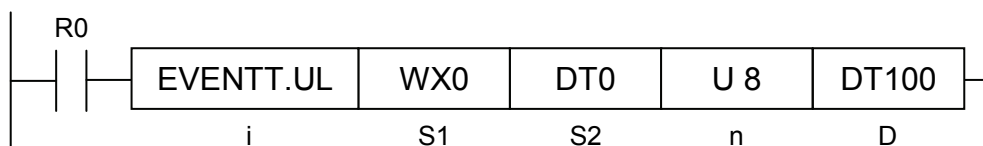
DT38	XE 計数值	15
DT39		
DT40	XF 計数值	16
DT41		
DT42	X10 計数值	17
DT43		

■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾・ポインタアクセス)に範囲外となった場合にセットします
	計数対象領域、計数作業領域、計数結果格納領域のいずれかがデバイス領域を超える場合

# EVENTT (事象発生時間計数命令)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i				●			

## ■ オペランド一覧

オペランド	説明
S1	計数開始位置先頭アドレス
S2	計数作業領域の先頭アドレス
n	計数するビット数
D	計数結果を格納する先頭デバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "		
S1	●	●	●	●	●	●	●	●	●	●	●											●
S2	●	●	●	●			●	●														●
n	●	●	●	●			●	●	●	●	●					●	●					●
D	●	●	●	●			●	●	●	●	●											●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- [S1]で指定されたデバイスから[n]ビット分の ON 時間を計数します。
- 演算結果は、[D]を先頭とする[n]\*2ワードの領域に格納します。

## ■ 処理内容

例1)WX0から8点分の計数を行う場合

[i]...UL            [S1]...WX0            [S2]...DT0            [n]...U 8            [D]...DT10

[S1]  
WX0 

X0-X7 の 8ビットが対象ビット
--------------------

[S2]  
DT0 

計数用作業領域
---------

[D]		
DT10	X0 計数值	1
DT11		
DT12	X1 計数值	2
DT13		
DT14	X2 計数值	3
DT15		
DT16	X3 計数值	4
DT17		
DT18	X4 計数值	5
DT19		
DT20	X5 計数值	6
DT21		
DT22	X6 計数值	7
DT23		
DT24	X7 計数值	8
DT25		

例2) WX0から17点分の計数を行う場合

[i]...UL            [S1]...WX0            [S2]...DT0            [n]...U 17            [D]...DT10

[S1] WX0 WX1	X0 - X10 の17ビットが対象ビット
--------------------	-----------------------

[S2] DT0 DT1	計数用作業領域
--------------------	---------

[D] DT10	X0 計数值	1
DT11	X1 計数值	2
DT12	X2 計数值	3
DT13	X3 計数值	4
DT14	X4 計数值	5
DT15	X5 計数值	6
DT16	X6 計数值	7
DT17	X7 計数值	8
DT18	X8 計数值	9
DT19	X9 計数值	10
DT20	X10 計数值	11
DT21	X11 計数值	12
DT22	X12 計数值	13
DT23	X13 計数值	14
DT24	X14 計数值	15
DT25	X15 計数值	16

●  
●  
●  
●

DT38	XE 計数值	15
DT39	XF 計数值	16
DT40	XF 計数值	16
DT41	XF 計数值	16
DT42	XF 計数值	16
DT43	X10 計数值	17

### ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾・ポインタアクセス)に範囲外となった場合にセットします
SR8 (ER)	計数対象領域、計数作業領域、計数結果格納領域のいずれかがデバイス領域を超える場合

# PID (PID演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S	PID 演算用パラメータエリア(30 ワード)の先頭番号

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""	
S							●														

## ■ 動作説明

- [S+2] に格納された測定値 PV を、[S+1]で指定する設定値 SP に一致させて保つために PID 演算を行います。
- 演算結果は、出力値 MV として[S+3]で指定されたエリアに格納します。
- PID 演算の方法(微分先行型/比例微分先行型、逆動作/正動作)および PID 演算に用いる係数(比例ゲイン、積分時間、微分時間)や演算の種類/周期は、パラメータテーブル[S]~[S+29]に設定します。

## ■ PID演算の種類

項目	説明
逆動作/正動作	プロセスに変化があったときの出力の上下方向が選択します。 測定値が下がると、出力を上げる場合(加熱など)は、「逆動作」を指定します。 測定値が上がると、出力を上げる場合(冷却)は、「正動作」を指定します。
微分先行型 PID /比例微分先行型 PID	微分先行型 PID: 一般に、設定値を変更したときに、出力の変動が大きくなりますが、収束は早くなります。
	比例微分先行型 PID: 一般に、設定値を変更したときに、出力の変動は小さくなりますが、収束が遅くなります。
オートチューニング	プロセスの応答を計測する事により、PID パラメータの Kp, Ti, Td の最適な値を計測します。オートチューニングを実行すると、推測された結果はオートチューニング終了後にパラメータのエリアに反映されます。

## ■ パラメータテーブルの設定

[S]			制御モード
[S+1]			設定値(SP)
[S+2]			測定値(PV)
[S+3]			出力値(MV)
[S+4]			出力下限値
[S+5]			出力上限値
[S+6]			比例ゲイン(Kp)
[S+7]			積分時間(Ti)
[S+8]			微分時間(Td)
[S+9]			制御周期(Ts)
[S+10]			オートチューニング進捗状況
[S+11]			} PID演算用ワークエリア
≈		≈	
[S+29]			

## ■ パラメータの設定:[S]~[S+23]

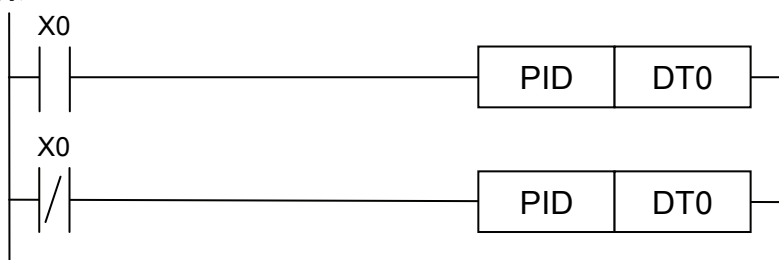
オペランド	パラメータ名	設定範囲	設定方法
[S]	制御モード	H0~H3	オートチューニング非実行時 H0:微分先行型、逆動作 H1:微分先行型、正動作 H2:比例微分先行型、逆動作 H3:比例微分先行型、正動作
		H8000~H8003	オートチューニング実行時 H8000:微分先行型、逆動作 H8001:微分先行型、正動作 H8002:比例微分先行型、逆動作 H8003:比例微分先行型、正動作
[S+1]	設定値(SP)	K0~K10000	プロセスの制御量の目標値を、下記の範囲で設定してください。
[S+2]	測定値(PV)	K0~K10000	プロセスの制御量の現在値を、A/D変換ユニットなどを使用して、入力してください。
[S+3]	出力値(MV)	K0~K10000	PID演算結果が格納されます。D/A変換ユニットなどを使用して、出力してください。
[S+4]	出力下限値	K0~K9999	出力値MVの範囲を指定してください。指定した範囲の値が出力されます。
[S+5]	出力上限値	K1~K10000	
[S+6]	比例ゲイン(Kp)	K1~K9999 (0.1~999.9)	PID演算に用いる係数を指定してください。設定値×0.1が、実際の比例ゲインになります。(注1)
[S+7]	積分時間(Ti)	K1~K30000 (0.1~3000 s)	PID演算に用いる係数を指定してください。設定値×0.1が実際の積分時間になります。0を指定すると、積分を行いません。(注1)
[S+8]	微分時間(Td)	K0~K10000 (0~1000 s)	PID演算に用いる係数を指定してください。設定値×0.1が実際の微分時間になります。0を指定すると、積分を行いません。(注1)
[S+9]	制御時間(Ts)	K1~6000 (0.01~60.0 s)	PID演算を実行する周期を指定してください。設定値×0.01が、実際の制御周期になります。
[S+10]	オートチューニング 進行状況	K0~K5	制御モードにてオートチューニングを指定した場合、オートチューニングの進行度を示します。初期値[0]から進行状況に応じてK1~K5の値が格納され、オートチューニング終了後は初期値に戻ります。
[S+11] ~[S+29]	PID演算用ワーク エリア	-	システムが演算に使用する必要なワークエリアです。

(注1):制御モード指定で、オートチューニングを指定すると、自動的に調整されて、設定値が書き換えられます。

## ■ プログラム上のご注意

- パラメータテーブルとして、演算用のワークエリアを含めて、30ワードのエリアを必要とします。他の命令で、このエリア値を書き換えることがないようにご注意ください。
- パラメータテーブルがエリアを超えても、エラー検知を行いません。[S]を指定する場合に、最終番号から最低30ワード以前の番号を指定してください。
- インデックス修飾でエリアを超えることがないようにご注意ください。エリアを超えても、エラー検知を行いません。
- 下図のように、同じテーブルを指定した2つ以上のPID命令をプログラム中に記述した場合、正しく動作しません。PID命令は、実行条件が成立していない場合も、指定されたテーブルを使用して、内部的に動作しています。このような場合には、テーブルを別々のアドレスに設定してください。

<例>



## ■ オートチューニング実行時のご注意

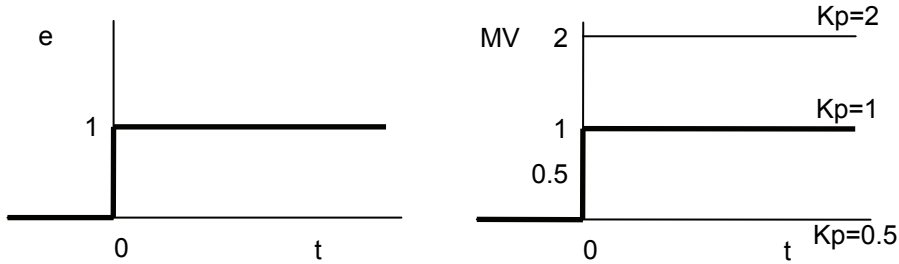
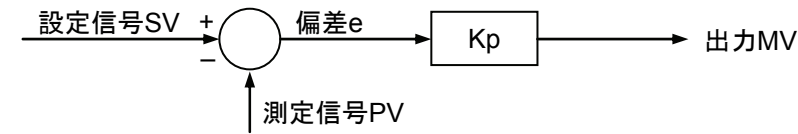
- プロセスによってはオートチューニングが実行できない場合があります。その場合、元のパラメーター演算に戻ります。
- オートチューニング終了後、制御モード[S]のエリアは、H 8000～H 8003 から H 0～H 3 へ自動的に書き換えられます。プログラム等で再度書き替わらないようにしてください。
- オートチューニング終了後、比例ゲイン[Kp]、積分時間[Ti]、微分時間[Td]には最適な値が格納されますが、実行前には、設定範囲内で適当な値(例えば下限値)を指定する必要があります。
- オートチューニング終了後、比例ゲイン[Kp]、積分時間[Ti]、微分時間[Td]:に最適な値を格納します。格納された値が書き換わらないよう、ご注意ください。
- オートチューニングは設定値(SP)に対して、測定値(PV)を上下させるように、出力値(MV)を上限値の値にしたときの測定値(PV)の変化と出力値(MV)を下限値にしたときの測定値(PV)の変化を測定することにより、最適な、Kp, Ti, Td の値を算出します。
- オートチューニングの出力値(MV)の変化は最短で、上限値出力ー下限値出力ー上限値出力の3回の変化で完了します。複数回以上の変化でもオートチューニング進行状況が0のまま変化しない場合は、制御同期 Ts を短くして再度オートチューニングを実行してください。



## ■ PID制御の概要

### ①比例動作

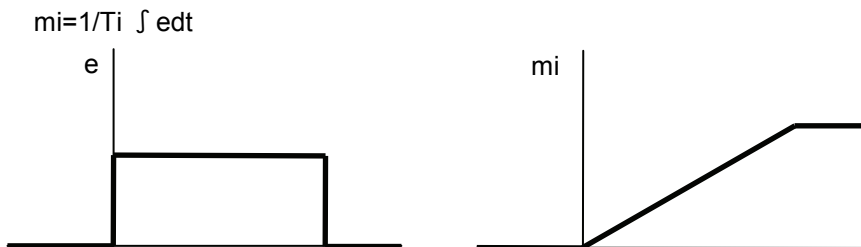
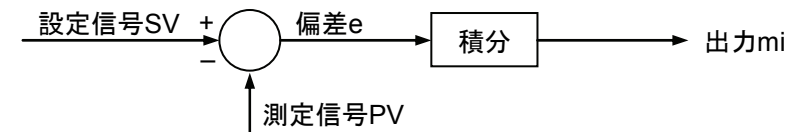
入力に比例する大きさの出力を出す制御動作



- 制御量を一定に保つ。
- オフセット(定常偏差)が残る。
- $K_p$  の値が大きいほど、比例動作は強く働く。

### ②積分動作

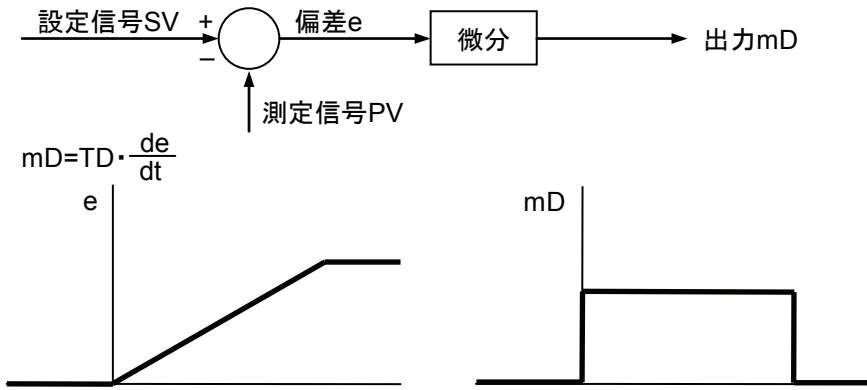
入力の積分時間に比例する大きさの出力を出す制御動作。



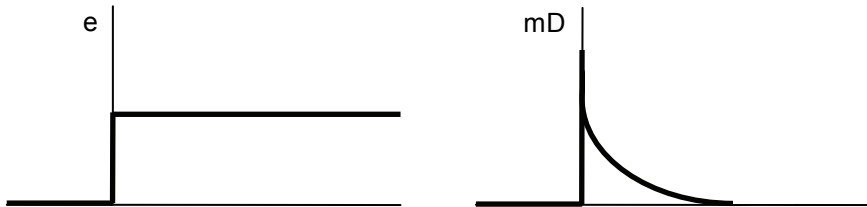
- 比例動作、または比例微分動作と組み合わせられ、生じるオフセットを取り除く。
- $T_i$  が小さいほど積分動作は強く働く。

### ③微分動作

入力的时间微分値に比例する大きさの出力を出す制御動作。

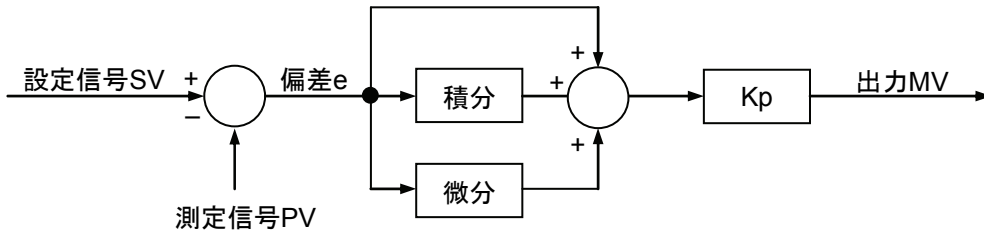


- 微分動作の進み特性により、プロセスの遅れ特性が制御に与える悪影響を減少させる。
- $T_d$  が大きいほど微分動作は強く働く。
- 純粋な微分動作の場合、ノイズ等が入力されると一時的に無効になり、操作端に悪影響を与えるため不完全微分を実行している。



### ④PID動作

比例、積分、微分の各動作を組み合わせた物をPID動作という。



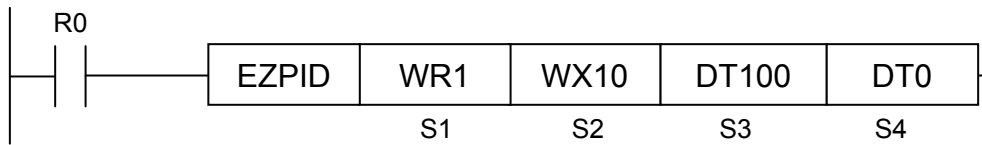
- PID 制御でパラメータが最適であれば制御量を目標値に早く一致させて保つことができる。

### ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	パラメータ設定範囲外の場合にセットします

# EZPID (PID演算: PWM出力可能)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●: 指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S1	オートチューニング、出力値(MV)の方法を決定するコントロールデータを設定する 1 ワードのエリア
S2	測定値(PV) を入力する 1 ワードのエリア
S3	設定値(SV)、比例ゲイン(Kp)、積分時間(Ti)、微分時間(Td) を設定する 4 ワードのエリア
S4	下記から構成される 30 ワードのエリア 出力値(MV)の格納エリア、制御周期(Ts)、制御モード、オートチューニング関連パラメータの設定エリア、演算用ワークエリア

## ■ 指定可能なデバイス(●: 指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデック ス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		" "
S1	●	●	●	●			●	●													
S2	●	●	●	●	●	●	●	●													
S3	●	●	●	●			●	●													
S4	●	●	●	●			●	●													

## ■ 動作説明

- [S+2] に格納された測定値(PV)を、[S3] で指定する設定値(SP)に一致させて保つための PID 演算を行います。
- 演算結果は、[S4] で指定されたエリアに出力値(MV)として格納します。命令の直後に OUT 命令を記述すると、出力値に応じた PWM 出力(ON-OFF 出力) を得ることができます。
- PID 演算に用いるパラメータ(比例ゲイン、積分時間、微分時間)は、[S3+1]~[S3+3] に設定します。これらの値を自動的に計算するオートチューニング機能もあります。
- PID 演算の方法(微分先行型/比例微分先行型、逆動作/正動作)や制御周期 Ts などを変更する場合は、[S4]~[S4+9] のエリアに設定します。

## ■ PID演算の種類

項目	説明
逆動作/正動作	プロセスに変化があったときの出力の上下方向が選択します。 測定値が下がると、出力を上げる場合(加熱など)は、「逆動作」を指定します。 測定値が上がると、出力を上げる場合(冷却)は、「正動作」を指定します。
微分先行型 PID /比例微分先行型 PID	微分先行型 PID: 一般に、設定値を変更したときに、出力の変動が大きくなりますが、収束は早くなります。 比例微分先行型 PID: 一般に、設定値を変更したときに、出力の変動は小さくなりますが、収束が遅くなります。
オートチューニング	プロセスの応答を計測する事により、PID パラメータの Kp, Ti, Td の最適な値を計測します。オートチューニングを実行すると、推測された結果はオートチューニング終了後にパラメータのエリアに反映されます。

(注 1): デフォルト値では、逆動作、微分先行型で制御します。演算方法を変更する場合は、2 回目の EZPID 命令実行前に、[S4+5]のエリアの値を変更します。

(注 2): オートチューニングを行なう場合は、コントロールデータ [S1] のエリアで設定します。

## ■ パラメータ [S1]～[S3] の設定

オペランド	パラメータ名	設定範囲	設定方法
[S1]	コントロール データ	bit 0	bit0=1 (ON) : オートチューニング要求を行いません。オートチューニングが完了すると、EZPID 命令実行時にリセットします。オートチューニングを中止する場合も、このビットをリセットしてください。 bit0=0 (OFF) : PID 制御を行いません。
		bit 1	オートチューニングが正常に完了したとき、bit 1=1 (ON) がセットされます。命令実行開始時に bit 1=0 (OFF) にリセットします。
		bit 2	命令の実行条件が OFF > ON になったときに [S4] 出力値 (MV) をクリアするかどうかを指定します。 bit2=0 (OFF) : 前回命令を実行時ときの出力値 (MV) をクリアします。 bit2=1 (ON) : 前回命令を実行したときの出力値 (MV) を保持します。
		bit 3	PID 演算で演算結果を出力する方式を選択します。 bit3=0 (OFF) : PWM 出力 bit3=1 (ON) : アナログ出力
		bit 4	出力 (MV) の内部計算値の最大値、最小値の範囲を指定します。 bit4=0 (OFF) : 出力上限値－出力下限値の+20%、-20%とする。 bit4=1 (ON) : 出力上限値、出力下限値とする。
		bit F～bit5	ビット 5～F は、予約ビットです。0 で使用してください。
[S2]	測定値(PV)	K -30000 ～ K 30000	プロセス制御量の現在値を、アナログ入力ユニットなどを使用して、入力します。アナログ入力ユニットの入力データエリア WXn を直接指定することもできます。
[S3]	設定値(SP)	K -30000 ～ K 30000	プロセスの制御量の目標値を設定します。命令または表示器など外部機器から設定します。
[S3+1]	比例ゲイン (Kp)	U1～U9999 (0.1～999.9)	PID 演算に用いるパラメータを指定します。 設定値×0.1 が、それぞれのパラメータ値になります。 オートチューニングを実行すると、オートチューニング完了時に、データが格納されま す。
[S3+2]	積分時間 (Ti)	U0～U0000 (0～3000 s)	
[S3+3]	微分時間 (Td)	U0～U10000 (0～1000 s)	

(注 1) : [S1] のエリアは、ビット操作ができる非保持型の演算用メモリエリア (WR) に割り付けて使用することをおすすめします。

(注 2) : [S2] のエリアは、非保持型の演算用メモリエリアに割り付けて使用することをおすすめします。

(注 3) : [S3] ～ [S3+3] のエリアは、保持型の演算用メモリエリアに割り付けて使用することをおすすめします。

(注 4) : EZPID 命令起動時、[S3+1] ～ [S3+3] のエリアがすべて"0"の場合、比例ゲイン (Kp)=1、積分時間 (Ti)=0、微分時間 (Td)=0 として演算を継続します。

## ■ パラメータ [S4]～[S4+29] の設定

オペランド	パラメータ名	初期値	設定範囲	設定方法
[S4]	出力値 (MV)	K0	K -10000 ～K 10000	<ul style="list-style-type: none"> <li>●PID 演算の結果を格納します。</li> <li>●[S1]の bit3 でPWM 出力を選択した場合は、0～10000 の範囲で計算し、PWM 出力のデューティ比 (0～100%) となります。</li> <li>●[S1]の bit3 でアナログ出力を選択した場合は、下記計算式で得られた値を格納します。格納された値をアナログ出力ユニットのレンジに変換して、出力します。 (出力上限値－出力下限値)×内部計算値/10000+下限値</li> </ul>
[S4+1]	出力下限値	K0	K-10000 ≦下限値 <上限値 ≦K 10000	出力値 MV の範囲を指定してください。指定した範囲の値が出力されます。
[S4+2]	出力上限値	K10000		
[S4+3]	100%出力帯	U0 (0%)	U0～U80 (0%～80%)	<ul style="list-style-type: none"> <li>●測定値(PV)が設定値の何%以上に達すると、PID 制御に移行するかを指定します。指定した測定値(PV)に達するまでの間は、出力値 (MV) は 100% となります。測定値(PV)が設定値(SP)より小さいときは、設定値(SP)への到達時間を短縮する効果があります。</li> <li>●設定値が K80 のとき、測定値 (PV) が設定値(SP)の 80% に達するまで 100% 出力を行ない、80% を超えると PID 制御切り替えます。設定値が K0 (初期値) のときは、最初から PID 制御を実行します。</li> </ul>
[S4+4]	制御周期 (Ts)	U100 (1s)	U1～6000 (0.01～60.0 s)	PID 演算を実行する周期を指定します。 設定値×0.01 が、実際の制御周期になります。
[S4+5]	制御モード	U0 (微分 先行型、 逆動作)	U0～U3	U0:微分先行型、逆動作 U1:微分先行型、正動作 U2:比例微分先行型、逆動作 U3:比例微分先行型、正動作
[S4+6]	オートチューニング バイアス値	U0	U0	オートチューニング中の温度の過上昇を抑えたい場合に、オートチューニング中の設定値と実際の設定値との差を設定します。 (注 3)
[S4+7]	オートチューニング 比例ゲイン (Kp) 補正係数	U125 (125%)	U50～U500 (50～500%)	左記の係数セットし、オートチューニングを実行すると、オートチューニングで得られたパラメータ値をそれぞれの係数倍し、各パラメータの設定エリア [S3+1] ～ [S3+3] に格納します。 例)[S4+7] に U200 を設定すると、オートチューニングで得られた比例ゲイン Kp 値の 200% の値を [S3+1] にセットします。
[S4+8]	オートチューニング 積分時間 (Ti) 補正係数	U200 (200%)	U50～U500 (50～500%)	
[S4+9]	オートチューニング 微分時間 (Td) 補正係数	U100 (100%)	U50～U500 (50～500%)	
[S4+10]	オートチューニング 進行状況	U0	U0～U5	オートチューニングの進行度を示します。初期値 [0] から進行状況に応じて U1～U5 の値が格納され、オートチューニング終了後は初期値 U0 に戻ります。
[S4+11] ～[S4+29]	PID 演算用 ワークエリア	K0	-	システムが演算に使用する必要なワークエリアです。プログラムで上書きしないようにしてください。

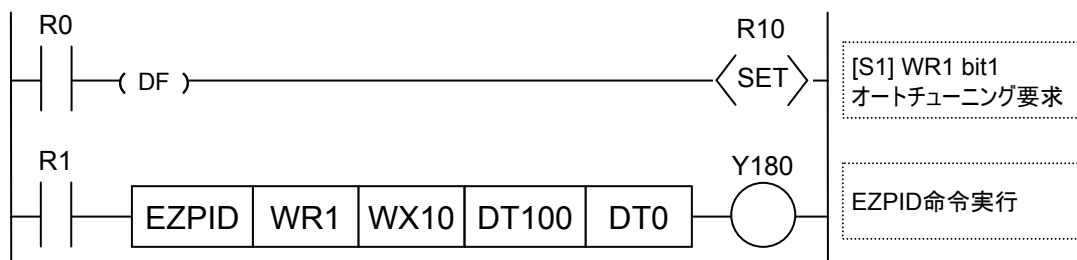
(注 1) : デフォルトでは、EZPID 命令起動時に、[S4]出力値 (MV) の値をクリアします。前回実行時の値を保持する必要があるときは、EZPID 命令の実行前に、オペランド[S1]の bit2 に 1 をセットしてください。

(注 2) : [S4+1]～[S4+29]のエリアは、EZPID 命令の起動時に、初期値にプリセットします。

(注 3) : オートチューニングバイアス値の詳細については、オートチューニング バイアス値 [S4+6] の設定の項をご参照ください。

## ■ サンプルプログラム(PWM出力)

- オートチューニングで比例ゲイン (Kp)、積分時間 (Ti)、微分時間 (Td) を設定した後、PID 演算による PWM 出力を行なう場合を示します。



- 設定値(SP)をオペランド[S3]のエリアに、命令または表示器により設定します。
- 命令または表示器により、[S1]の bit0 (オートチューニング要求フラグ) をセットしたあと、EZPID 命令の実行条件を ON にすると、オートチューニング動作を開始します。
- オートチューニングが正常に完了すると、オペランド[S3+1]～[S3+3]のエリアにパラメータ Kp, Ti, Td がセットされます。このとき、オペランド[S1]の bit0 (オートチューニング要求フラグ) は OFF にリセットされ、オペランド[S1]の bit1 (オートチューニング完了フラグ) が ON になります。
- EZPID 命令の実行条件が ON であれば、次のスキャンより PID 制御を開始し、PWM 出力を開始します。
- オペランド[S4]出力値 (MV) の値 (K0～K10000) を 0～100%のデューティ比に換算し、PWM 出力を行ないます。
- オペランド[S4]～[S4+29]のエリアは、EZPID 命令の実行条件の立ち上がり時に、初期値にプリセットされます。[S4] 出力値 (MV) をクリアしたくない場合は、[S1]の bit2 に 1 をセットしてから EZPID 命令を実行します。
- PID 制御中に EZPID 命令の実行条件が OFF になると、PWM 出力 Y180 も OFF になります。このとき、オペランド [S4]の出力値 (MV) の値は保持されます。

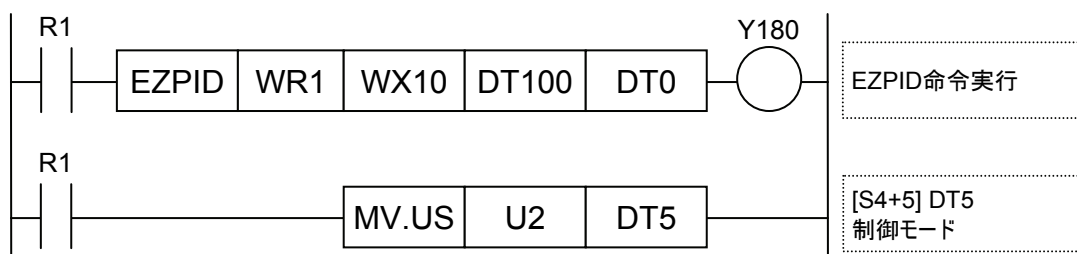
## ■ PWM 出力の周期とデューティ比

- デューティ比は、[S4]に格納される出力値 (MV) が K0～K10000 のどれだけの割合を占めているかによって、PWM のデューティが決まります。
- 出力値 (MV) が K0 のとき PWM 出力は常時 OFF、K10000 のとき PWM 出力は常時 ON となります。
- PWM 出力の周期は、オペランド[S4+4]に指定する制御周期の値で決まります。初期値では周期 1 秒となります。

## ■ サンプルプログラム(制御条件の変更)

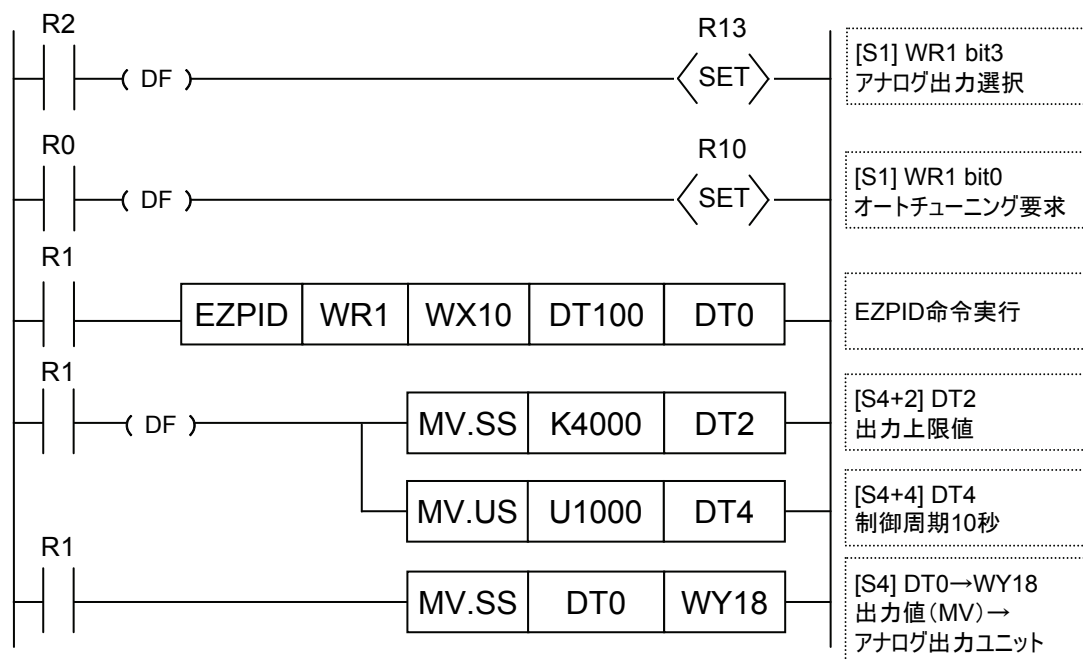
- デフォルトでは制御条件は、演算周期 Ts: 1 秒、制御方法: 微分先行型逆動作(加熱)で演算します。
- 制御条件を変更するには、オペランド[S4+1]～[S4+9]の値を MV 命令などにより変更してください。
- EZPID 命令実行後、2 回目の実行前に変更してください。

[例] 制御モードを比例微分先行型に切り替え、PWM出力を行なう場合



## ■ サンプルプログラム(アナログ出力)

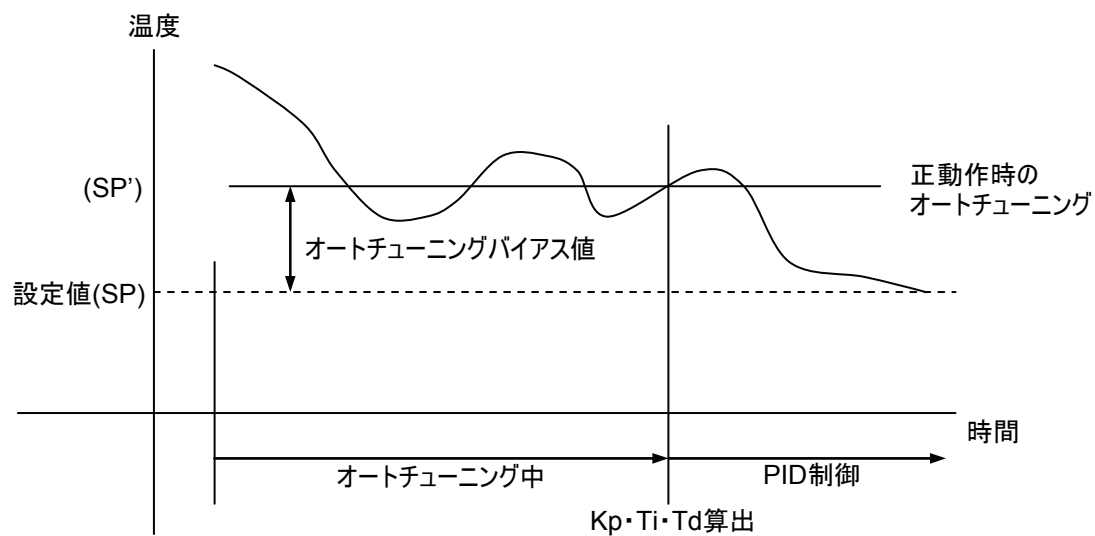
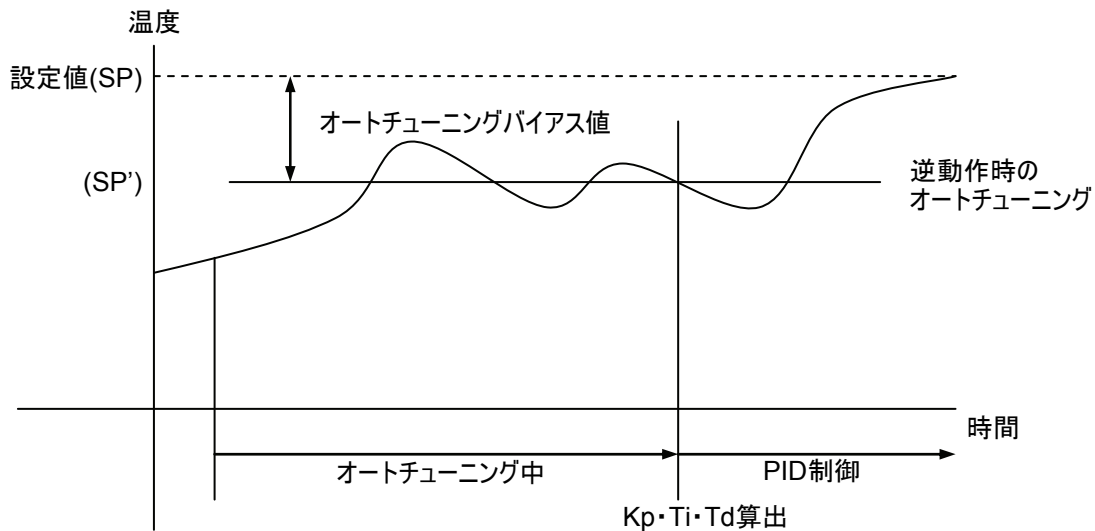
[例] アナログ出力、出力上限値[S4+2]を K4000、制御周期[S4+4]を K1000 (10 秒)に変更する場合



- 出力にアナログ出力を使用する場合は、オペランド[S1]の bit3 に 1 をセットします。
- 出力下限値[S4+1]、出力上限値[S4+2]は、アナログ出力ユニットの出力範囲に合わせて設定します。
- 制御周期(Ts):[S4+4]の値は、アナログ入力ユニットの入力更新周期(通常 0.1 秒以上)に合わせて変更します。  
(例) [S4+4]の値が K10 のとき、Ts=100ms となります。
- 制御モードなど他のパラメータは、必要に応じて変更します。
- オペランド[S4]に格納された出力値(MV)は、アナログ出力ユニットの出力値に対応するデジタル値の出力エリア WY へ転送します。
- [S4]出力値(MV)には、出力の内部計算値(K0~K10000)が下記の変換式で換算され、格納されます。  
変換式: (出力上限値-出力下限値) × (内部計算値) / 10000 + (出力下限値)
- [S4]が保持型エリアに割り付けられている場合、EZPID 命令の実行条件が OFF になっても、出力値(MV)の値は保持されます。
- 出力にアナログ出力を使用する場合は、EZPID 命令の後に OUT 命令を記述する必要ありません。また、アナログ出力を使用する場合は、PWM 出力は OFF になります。

## ■ オートチューニング バイアス値 [S4+6] の設定

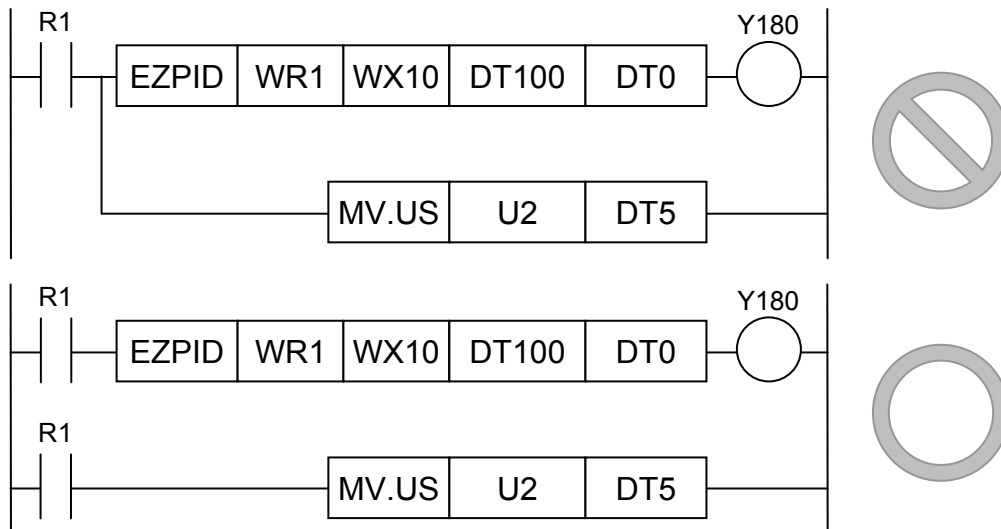
- 温度の過上昇を防ぐなど、オートチューニング時の出力を抑えたい場合に、実際の PID 制御運転時の設定値(SP) とオートチューニング時の設定値(SP') との差をバイアス値として設定します。
- 逆動作(加熱)の場合、 $(SP) \geq (SP')$  となり、その差がバイアス値となります。
- 正動作(冷却)の場合、 $(SP) \leq (SP')$  となり、その差がバイアス値となります。
- 測定値 (PV) が設定値(SP)の近傍にある場合にオートチューニングを起動する場合でも、バイアス値が加減された設定値(SP') でオートチューニングを実行します。





## ■ プログラム上のご注意

- 実行条件の立ち上がり時に、オペランド[S4]～[S4+29]のエリアを初期化します。初期値以外の値に設定する場合は、MV 命令などで、書き込んでください。
- PID 演算命令は、演算の周期や PWM 出力のタイミングを内部で常時演算しています。必ず1スキャン中に1回のみ演算するようにしてください。従って、サブルーチンや割り込みプログラム中での実行はしないようにしてください。また、同じオペランドを指定した EZPID 命令を複数記述することはできません。
- PID 演算中は実行条件を OFF しないでください。PID 演算できなくなります。
- 複数の対象を制御する場合、PWM の出力周期を同期させたくない場合は、起動条件の立ち上がり時間などを調整して、起動タイミングをずらして使用してください。
- 同じテーブルを指定した2つ以上の PID 命令をプログラム中に記述した場合、正しく動作しません。PID 命令は、実行条件が成立していない場合も、指定されたテーブルを使用して、内部的に動作しています。このような場合には、テーブルを別々のアドレスに設定してください。
- PID 演算用のワークエリアは、他の命令で、書き換えることがないようにご注意ください。
- パラメータテーブルがエリアを超えても、エラー検知を行いません。[S4]のエリアを指定する場合は、最終番号から最低 30ワード以前の番号を指定してください。また、インデックス修飾でエリアを超えることがないようにご注意ください。エリアを超えても、エラー検知を行いません。
- EZPID 命令の直後では PWM 出力と同様に実行条件が変化しますので、後続の命令を記述する場合は、正しく動作しません。



## ■ オートチューニング実行時のご注意

- プロセスによってはオートチューニングが実行できない場合があります。その場合、元のパラメーター演算に戻ります。
- オートチューニング終了後、比例ゲイン[Kp]、積分時間[Ti]、微分時間[Td]には、最適な値が格納されますが、実行前には、設定範囲内で適当な値(例えば下限値)を指定する必要があります。
- オートチューニング終了後、比例ゲイン[Kp]、積分時間[Ti]、微分時間[Td]に最適な値を格納します。格納された値が書き換わらないよう、ご注意ください。
- オートチューニング実行中は、設定値(SP)に対して、測定値(PV)を上下させるように、出力値(MV)を上限値の値にしたときの測定値(PV)の変化と出力値(MV)を下限値にしたときの測定値(PV)の変化を測定することにより、最適な、Kp, Ti, Td の値を算出します。
- オートチューニングの出力値(MV)の変化は最短で、上限値出力ー下限値出力ー上限値出力の 3 回の変化で完了します。複数回以上の変化でもオートチューニング進行状況が 0 のまま変化しない場合は、制御同期 Ts を短くして再度オートチューニングを実行してください。

## ■ 演算動作

- 実行条件立ち上がりで、[S4]～[S4+29]の初期化を行います。
- PID 動作開始時、Kp, Ti, Td の各パラメータがすべて 0 の場合は、それぞれ 1, 0, 0 で初期化して演算継続します。
- オートチューニング要求信号の立ち上がりで、[S1]の bit2 (オートチューニング完了フラグ)、[S4+10] (オートチューニング完了コード)をクリアします。
- オートチューニングの設定値は、設定値(SP)ーバイアス値を目標値として、動作します。
- オートチューニング正常終了時は、計算結果の Kp, Ti, Td に、[S4+7]～[S4+9]に指定した補正係数を乗じた結果を格納します。
- オートチューニング正常終了時は、[S1]の bit2 (オートチューニング完了フラグ)、オートチューニング完了コードを [S4+10]に格納します
- [S1]の bit3=0: PWM 出力のとき、0～10000 の範囲で出力し、  
変換式: (上限値ー下限値) × 内部計算値/10000 とした時のデューティで出力する。
- [S1]の bit3=1: アナログ出力のとき、内部計算値は 0～10000 の範囲で出力し、下式で換算し、オペランド[S4]に出力値(MV)としてセットします。  
変換式: (上限値ー下限値) × 内部計算機 / 10000 + 下限値

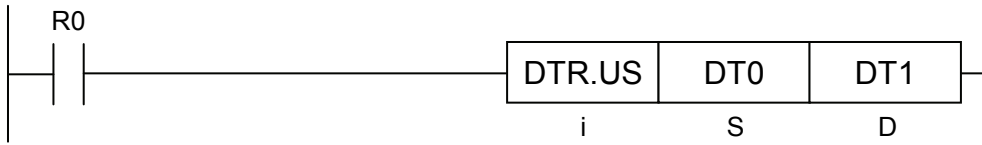
## ■ フラグ動作

名称	説明
SR7	[S2]: 測定値(PV)、[S3]: 設定値(SP)、[S3]+1: KP、[S3]+2: TI、[S3]+3: TD、[S4]+4～[S4]+9の各パラメータが設定範囲外になった場合にセットします
SR8 (ER)	[S3]または[S4]で指定されたエリアが、指定演算デバイスの上限を超えた場合にセットします
	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

■ MEMO

# DTR (データ変化検出)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S	データ値の変化を検出するデバイスアドレス
D	前回実行時のデータ値を格納するデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "		
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●								●
D	●	●	●	●			●	●	●		●	●	●	●								●

\*1: 演算単位が16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

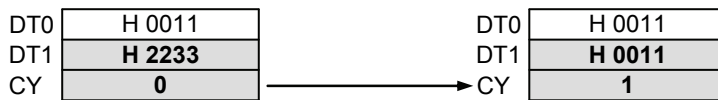
- [S]で指定されたデバイスアドレスのデータ値が前回実行時のデータ値から変化していれば、SR9(CY)をONにします。

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

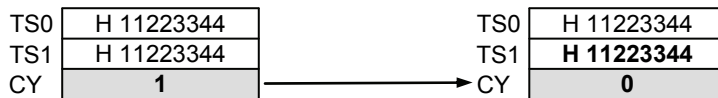
[S]...DT0 [D]...DT1



例2) 演算単位が32ビット(UL,SL,SF)の場合

[i]...UL,SL,SF

[S]...TS0 [D]...TS1



## ■ プログラム上のご注意

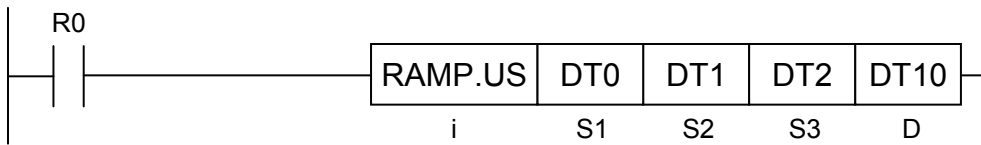
- 演算単位が実数の場合でも、データ変化のみチェックを行い非実数等の型チェックは行いません。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾・ポインタアクセス)に範囲外となった場合にセットします
SR9(CY)	[S]の値が[D]と違う値の場合、セットします
	[S]の値が[D]と同じ値の場合、リセットします

# RAMP (傾斜出力)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	初期値を格納しているデバイスアドレスや定数
S2	目標値を格納しているデバイスアドレスや定数
S3	時間幅を格納しているデバイスアドレスや定数(データ指定可能範囲:1~30000)
D	出力格納先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8		..
S1	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
S2	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
S3	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●				●	●	●							●

\*1: 演算単位が16ビット整数(SS、US)の場合は指定不可

\*2: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

\*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*8: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [S1]、[S2]、[S3]で指定した出力初期値、出力目標値、出力時間(ms 単位)からスケーリングを行い、実行開始からの経過時間に応じた線形出力を行います。

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

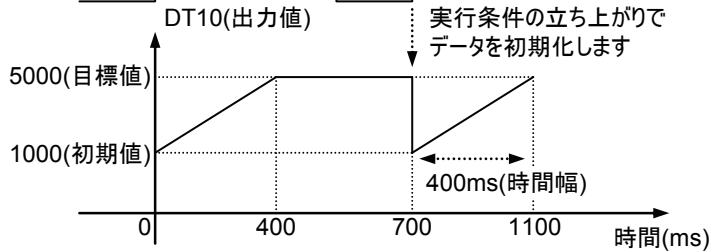
[i]...US,SS

[S1]...DT0:K1000 [S2]...DT1:K5000 [S3]...DT2:K400 [D]...DT10

実行条件

ON

OFF



例2) 演算単位が32ビット(UL,SL)の場合

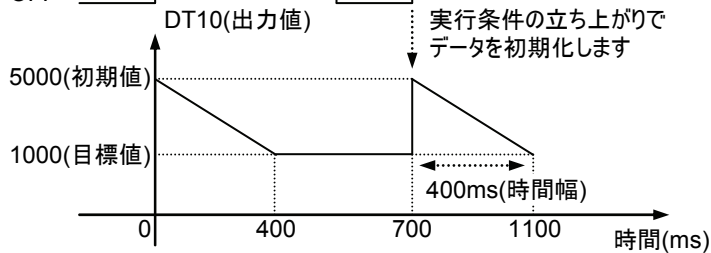
[i]...UL,SL

[S1]...DT0:K5000 [S2]...DT1:K1000 [S3]...DT2:K400 [D]...DT10

実行条件

ON

OFF



## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾・ポインタアクセス)に範囲外となった場合にセットします
SR8	
(ER)	

# LIMIT (上下限リミット制御)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	下限値を格納しているデバイスアドレスや下限値データ
S2	上限値を格納しているデバイスアドレスや上限値データ
S3	入力値を格納しているデバイスアドレスや入力値データ
D	出力格納先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8	" "	
S1	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
S2	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
S3	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●				●	●	●							●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

\*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*8: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能



## ■ 動作説明

- [S3]で指定した入力値が[S1]、[S2]で指定した上下限リミット値の範囲内か否かにより、[D]で指定したデバイスアドレスに格納する出力値を制御します。
- 出力値は、次のように定義します。

下限値:[S1] > 入力値:[S3]のとき	下限値:[S1] → 出力値:[D]
上限値:[S2] < 入力値:[S3]のとき	上限値:[S2] → 出力値:[D]
下限値:[S1] ≤ 入力値:[S3] ≤ 上限値:[S2]のとき	入力値:[S3] → 出力値:[D]

- 上限リミット値のみによる制御を行う場合は、下限リミット値:[S1]に演算単位別の最小値を指定します。
- 下限リミット値のみによる制御を行う場合は、上限リミット値:[S1]に演算単位別の最大値を指定します。

	US	SS	UL	SL	SF
最小値	0	<b>-32768</b>	0	<b>-2147483648</b>	<b>負の無限大</b>
最大値	65535	32767	4294967295	2147483647	<b>正の無限大</b>

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

[i]...US,SS

[S1]...DT0 [S2]...DT1 [S3]...DT2 [D]...DT10

<table border="1" style="margin-bottom: 5px;"> <tr><td>DT0</td><td style="text-align: center;">K 100</td></tr> <tr><td>DT1</td><td style="text-align: center;">K 1000</td></tr> <tr><td>DT2</td><td style="text-align: center;">K 500</td></tr> </table>	DT0	K 100	DT1	K 1000	DT2	K 500	→	<table border="1" style="margin-bottom: 5px;"> <tr><td>DT0</td><td style="text-align: center;">K 100</td></tr> <tr><td>DT1</td><td style="text-align: center;">K 1000</td></tr> <tr><td>DT2</td><td style="text-align: center;">K 500</td></tr> </table>	DT0	K 100	DT1	K 1000	DT2	K 500
DT0	K 100													
DT1	K 1000													
DT2	K 500													
DT0	K 100													
DT1	K 1000													
DT2	K 500													
DT10 <table border="1" style="display: inline-table;"><tr><td style="text-align: center;"><b>K 0</b></td></tr></table>	<b>K 0</b>		DT10 <table border="1" style="display: inline-table;"><tr><td style="text-align: center;"><b>K 500</b></td></tr></table>	<b>K 500</b>										
<b>K 0</b>														
<b>K 500</b>														

例2) 演算単位が32ビット(UL,SL)の場合

[i]...UL,SL

[S1]...TS0 [S2]...TS1 [S3]...TS2 [D]...DT10

<table border="1" style="margin-bottom: 5px;"> <tr><td>TS0</td><td style="text-align: center;">K 100</td></tr> <tr><td>TS1</td><td style="text-align: center;">K 1000</td></tr> <tr><td>TS2</td><td style="text-align: center;">K 90</td></tr> </table>	TS0	K 100	TS1	K 1000	TS2	K 90	→	<table border="1" style="margin-bottom: 5px;"> <tr><td>TS0</td><td style="text-align: center;">K 100</td></tr> <tr><td>TS1</td><td style="text-align: center;">K 1000</td></tr> <tr><td>TS2</td><td style="text-align: center;">K 90</td></tr> </table>	TS0	K 100	TS1	K 1000	TS2	K 90
TS0	K 100													
TS1	K 1000													
TS2	K 90													
TS0	K 100													
TS1	K 1000													
TS2	K 90													
DT10 <table border="1" style="display: inline-table;"><tr><td style="text-align: center;"><b>K 0</b></td></tr></table>	<b>K 0</b>		DT10 <table border="1" style="display: inline-table;"><tr><td style="text-align: center;"><b>K 100</b></td></tr></table>	<b>K 100</b>										
<b>K 0</b>														
<b>K 100</b>														

例3) 演算単位が単精度浮動小数点形実数(SF)の場合

[i]...SF

[S1]...DT0 [S2]...DT4 [S3]...DT8 [D]...DT50

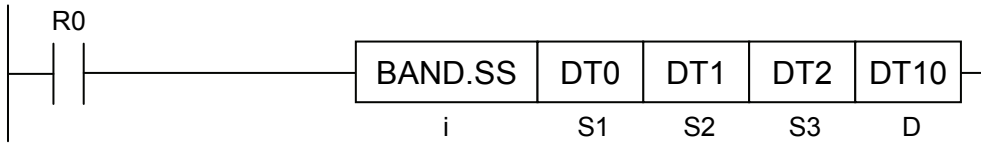
<table border="1" style="margin-bottom: 5px;"> <tr><td>DT0・DT1</td><td style="text-align: center;">SF 1.00E+02</td></tr> <tr><td>DT4・DT5</td><td style="text-align: center;">SF 1.00E+03</td></tr> <tr><td>DT8・DT9</td><td style="text-align: center;">SF 2.00E+03</td></tr> </table>	DT0・DT1	SF 1.00E+02	DT4・DT5	SF 1.00E+03	DT8・DT9	SF 2.00E+03	→	<table border="1" style="margin-bottom: 5px;"> <tr><td>DT0・DT1</td><td style="text-align: center;">SF 1.00E+02</td></tr> <tr><td>DT4・DT5</td><td style="text-align: center;">SF 1.00E+03</td></tr> <tr><td>DT8・DT9</td><td style="text-align: center;">SF 2.00E+03</td></tr> </table>	DT0・DT1	SF 1.00E+02	DT4・DT5	SF 1.00E+03	DT8・DT9	SF 2.00E+03
DT0・DT1	SF 1.00E+02													
DT4・DT5	SF 1.00E+03													
DT8・DT9	SF 2.00E+03													
DT0・DT1	SF 1.00E+02													
DT4・DT5	SF 1.00E+03													
DT8・DT9	SF 2.00E+03													
DT50・DT51 <table border="1" style="display: inline-table;"><tr><td style="text-align: center;"><b>SF 0.00E+00</b></td></tr></table>	<b>SF 0.00E+00</b>		DT50・DT51 <table border="1" style="display: inline-table;"><tr><td style="text-align: center;"><b>SF 1.00E+03</b></td></tr></table>	<b>SF 1.00E+03</b>										
<b>SF 0.00E+00</b>														
<b>SF 1.00E+03</b>														

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾・ポインタアクセス)に範囲外となった場合にセットします
SR8	
(ER)	[S1] > [S2] の場合にセットします

# BAND (不感帯制御)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i			●		●	●	●

## ■ オペランド一覧

オペランド	説明
S1	下限値を格納しているデバイスアドレスや下限値データ
S2	上限値を格納しているデバイスアドレスや上限値データ
S3	入力値を格納しているデバイスアドレスや入力値データ
D	出力格納先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス *1			整数			実数		文字	インデックス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U	H *5	SF *6	DF *7	""	
S1	●	●	●	●			●	●				●	●	●	●		●	●	●		●
S2	●	●	●	●			●	●				●	●	●	●		●	●	●		●
S3	●	●	●	●			●	●				●	●	●	●		●	●	●		●
D	●	●	●	●			●	●				●	●	●							●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

\*6: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*7: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [S3]で指定した入力値が[S1]、[S2]で指定した上下限値の範囲内か否かにより、[D]で指定したデバイスアドレスに格納する出力値を制御します。
- 出力値[D]には、以下の内容を格納します。

下限値:[S1] > 入力値:[S3]のとき	入力値:[S3] – 下限値:[S1] → 出力値:[D]
上限値:[S2] < 入力値:[S3]のとき	上限値:[S3] – 下限値:[S2] → 出力値:[D]
下限値:[S1] ≤ 入力値:[S3] ≤ 上限値:[S2]のとき	0 → 出力値:[D]

## ■ 処理内容

例1) 演算単位が16ビット(SS)の場合

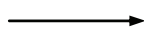
[i]...SS

[S1]...DT0 [S2]...DT1 [S3]...DT2 [D]...DT10

DT0	K 100
DT1	K 1000
DT2	K 500

DT0	K 100
DT1	K 1000
DT2	K 500

DT10 **K 100**



DT10 **K 0**

例2) 演算単位が32ビット(SL)の場合

[i]...SL

[S1]...TS0 [S2]...TS1 [S3]...TS2 [D]...DT10

TS0	K 100
TS1	K 1000
TS2	K 90

TS0	K 100
TS1	K 1000
TS2	K 90

DT10 **K 100**



DT10 **K -10**

例3) 演算単位が単精度浮動小数点形実数(SF)の場合

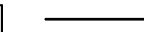
[i]...SF

[S1]...DT0 [S2]...DT4 [S3]...DT8 [D]...DT50

DT0・DT1	SF 1.00E+02
DT4・DT5	SF 1.00E+03
DT8・DT9	SF 3.00E+03

DT0・DT1	SF 1.00E+02
DT4・DT5	SF 1.00E+03
DT8・DT9	SF 3.00E+03

DT50・DT51 **SF 1.00E+02**



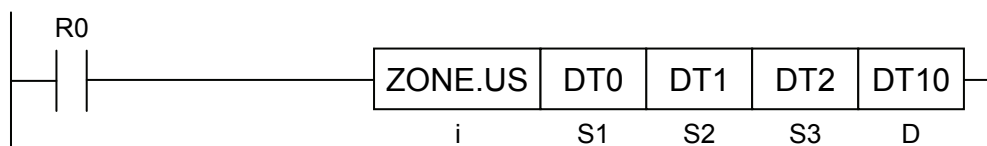
DT50・DT51 **SF 2.00E+03**

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾・ポインタアクセス)に範囲外となった場合にセットします
SR8	
(ER)	

# ZONE (ゾーン制御)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●	●	●

## ■ オペランド一覧

オペランド	説明
S1	負の場合のバイアス値を格納しているデバイスアドレスやバイアス値データ
S2	正の場合のバイアス値を格納しているデバイスアドレスやバイアス値データ
S3	入力値を格納しているデバイスアドレスや入力値データ
D	出力格納先のデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K *4	U *5	H *6	SF *7	DF *8	""	
S1	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
S2	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
S3	●	●	●	●			●	●				●	●	●	●	●	●	●	●		●
D	●	●	●	●			●	●				●	●	●							●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

\*4: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*5: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

\*6: 演算単位が整数(US、SS、UL、SL)の場合のみ指定可能

\*7: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*8: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [S3]で指定した入力値が[S1]、[S2]で指定したバイアス値を加算して、[D]で指定したデバイスアドレスに格納します。
- 出力値は、次のように定義します。

入力値:[S3] < 0 のとき	入力値:[S3] + 負のバイアス値:[S1] → 出力値:[D]
入力値:[S3] = 0 のとき	0 → 出力値:[D]
入力値:[S3] > 0 のとき	入力値:[S3] + 正のバイアス値:[S2] → 出力値:[D]

## ■ 処理内容

例1) 演算単位が16ビット(US,SS)の場合

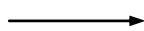
[i]...US,SS

[S1]...DT0 [S2]...DT1 [S3]...DT2 [D]...DT10

DT0	K -100
DT1	K 1000
DT2	K 500

DT0	K -100
DT1	K 1000
DT2	K 500

DT10 **K 100**



DT10 **K 600**

例2) 演算単位が32ビット(UL,SL)の場合

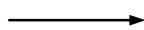
[i]...UL,SL

[S1]...TS0 [S2]...TS1 [S3]...TS2 [D]...DT10

TS0	K -100
TS1	K 1000
TS2	K -300

TS0	K -100
TS1	K 1000
TS2	K -300

DT10 **K 100**



DT10 **K -400**

例3) 演算単位が単精度浮動小数点形実数(SF)の場合

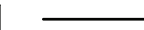
[i]...SF

[S1]...DT0 [S2]...DT4 [S3]...DT8 [D]...DT50

DT0・DT1	SF -1.00E+02
DT4・DT5	SF 1.00E+02
DT8・DT9	SF 0.00E+00

DT0・DT1	SF -1.00E+02
DT4・DT5	SF 1.00E+02
DT8・DT9	SF 0.00E+00

DT50・DT51 **SF 1.00E+02**



DT50・DT51 **SF 0.00E+00**

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾・ポインタアクセス)に範囲外となった場合にセットします

# FILTR (時定数処理)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S1	フィルタ処理対象データ(デバイスアドレス)
S2	フィルタ処理対象ビット(デバイスアドレスまたは定数)(データ指定可能範囲:H0000~HFFFF)
S3	フィルタ処理時間(デバイスアドレスまたは定数)(データ指定可能範囲:0~30000、ms 単位)
D	フィルタ処理結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデックス 修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "		
S1	●	●	●	●			●	●														●
S2	●	●	●	●			●	●								●	●					●
S3	●	●	●	●			●	●								●	●					●
D	●	●	●	●			●	●														●

\*1: 16ビットデバイス、32ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

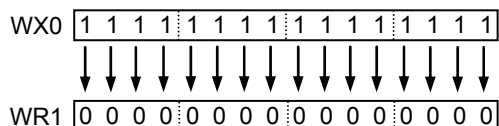
- [S1]で指定したデータのうち、[S2]で指定したビットが0のビットは直接出力、1のビットはフィルタ出力します。
- [S3]で指定した時間(0~30000ms 単位)で対象ビットについてフィルタ処理を行い、結果を[D]で指定したエリアに出力します。
- 実行条件立ち上がり時、[S1]で指定した入力の全ビットを無条件に直接出力します。
- フィルタ処理時間は、最大1スキヤンの遅れが発生する可能性があります。

## ■ 処理内容

対象データのうち、指定したビットが0のビットは直接出力、ビットが1のビットはフィルタ出力とし、指定した時間で対象ビットについてフィルタ処理を行い、結果を出力します。

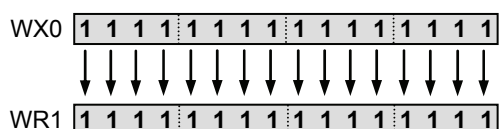
### ①初期条件

オペランド	説明	デバイス	設定値
[S1]	対象データ	WX0	H FFFF
[S2]	対象ビット	DT0	H FF00
[S3]	フィルタ時間	DT1	K500
[D]	処理結果	WR1	H 0000



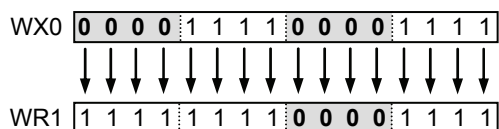
### ②実行条件立ち上がり時(入力の全ビットを無条件で直接出力する)

オペランド	説明	デバイス	設定値
[S1]	対象データ	WX0	H FFFF
[S2]	対象ビット	DT0	H FF00
[S3]	フィルタ時間	DT1	K500
[D]	処理結果	WR1	H FFFF



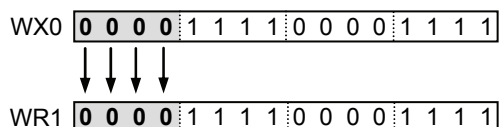
### ③フィルタ対象データ変更(対象外ビットのみ出力する)

オペランド	説明	デバイス	設定値
[S1]	対象データ	WX0	H 0F0F
[S2]	対象ビット	DT0	H FF00
[S3]	フィルタ時間	DT1	K500
[D]	処理結果	WR1	H FF0F



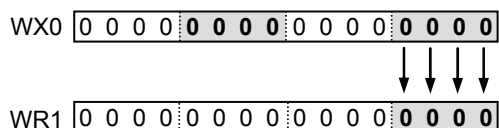
### ④フィルタ処理時間経過(対象ビットを出力する)

オペランド	説明	デバイス	設定値
[S1]	対象データ	WX0	H 0F0F
[S2]	対象ビット	DT0	H FF00
[S3]	フィルタ時間	DT1	K500
[D]	処理結果	WR1	H 0F0F



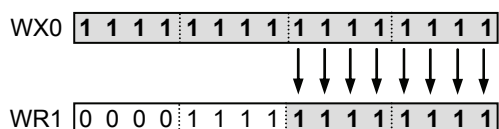
### ⑤フィルタ対象データ変更(対象外ビットのみ出力する)

オペランド	説明	デバイス	設定値
[S1]	対象データ	WX0	H 0000
[S2]	対象ビット	DT0	H FF00
[S3]	フィルタ時間	DT1	K500
[D]	処理結果	WR1	H 0F00



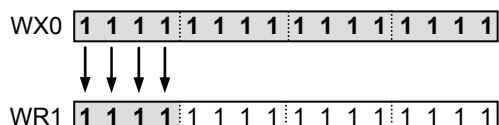
### ⑥フィルタ処理時間経過前にフィルタ対象データ変更(対象外ビットのみ出力する)

オペランド	説明	デバイス	設定値
[S1]	対象データ	WX0	H FFFF
[S2]	対象ビット	DT0	H FF00
[S3]	フィルタ時間	DT1	K500
[D]	処理結果	WR1	H 0FFF



### ⑦フィルタ処理時間経過(対象ビットを出力する)

オペランド	説明	デバイス	設定値
[S1]	対象データ	WX0	H FFFF
[S2]	対象ビット	DT0	H FF00
[S3]	フィルタ時間	DT1	K500
[D]	処理結果	WR1	H FFFF

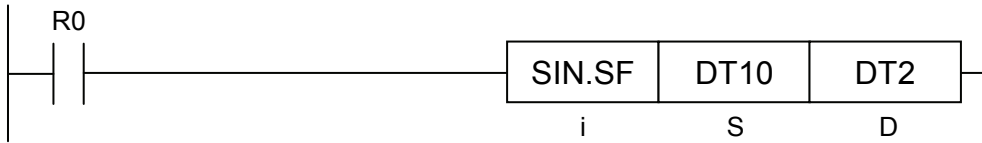


## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	フィルタ処理時間[S3]が範囲外となった場合にセットします

# SIN (正弦演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	角度データ(デバイスアドレスまたは定数)(単位:ラジアン)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4		""
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]に格納された角度データ(単位:ラジアン)の SIN を求めます。
- 演算結果は[D]を先頭とするエリアに格納します。  
SIN([S]) → [D]

## ■ 処理内容

例) 演算単位が単精度浮動小数点形実数(SF)の場合([S]に30° のラジアンを指定)

[i]...SF  
[S]...DT10 [D]...DT2

角度	値(ラジアン)	DT0-DT1	値
30°	SF 5.235988E-01	DT2-DT3	SF 5.000000E-01
60°	SF 1.047198E+00	DT4-DT5	SF 0.000000E+00
90°	SF 1.570796E+00		SF 0.000000E+00

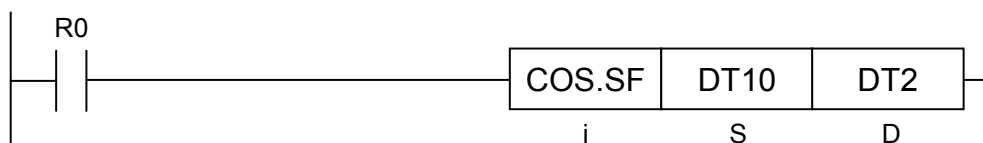
## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	



# COS (余弦演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	角度データ(デバイスアドレスまたは定数)(単位:ラジアン)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾 *1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4	..	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]に格納された角度データ(単位:ラジアン)のCOSを求めます。
- 演算結果は[D]を先頭とするエリアに格納します。  
COS([S]) → (D)

## ■ 処理内容

例) 演算単位が単精度浮動小数点形実数(SF)の場合([S]に30°のラジアンを指定)

[i]...SF

[S]...DT10 [D]...DT2

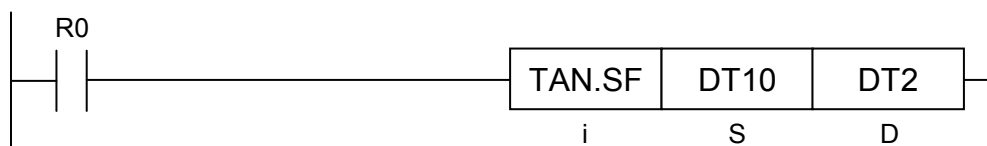
	角度	値(ラジアン)		値	
DT10・DT11	30°	SF 5.235988E-01	→	DT0・DT1	SF 0.000000E+00
DT12・DT13	60°	SF 1.047198E+00		DT2・DT3	SF 8.660254E-01
DT14・DT15	90°	SF 1.570796E+00		DT4・DT5	SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	[S](角度データ)に非実数を指定した場合にセットします

# TAN (正接演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	角度データ(デバイスアドレスまたは定数)(単位:ラジアン)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4		""
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]に格納された角度データ(単位:ラジアン)のTANを求めます。
- 演算結果は[D]を先頭とするエリアに格納します。  
TAN([S]) → (D)

## ■ 処理内容

例) 演算単位が単精度浮動小数点形実数(SF)の場合([S]に30°のラジアンを指定)

[i]...SF  
[S]...DT10 [D]...DT2

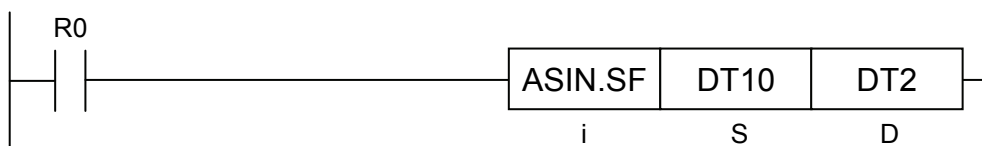
角度	値(ラジアン)	値
DT10・DT11	30° SF 5.235988E-01	DT0・DT1 SF 0.000000E+00
DT12・DT13	60° SF 1.047198E+00	DT2・DT3 SF 5.773503E-01
DT14・DT15	90° SF 1.570796E+00	DT4・DT5 SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	

# ASIN (逆正弦演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	角度データ(デバイスアドレスまたは定数)(SIN 値)(データ指定可能範囲:-1.0 ~ +1.0)
D	演算結果(デバイスアドレス)(単位:ラジアン)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾 *1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4	..	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]に格納された SIN 値の ASIN(逆正弦)を求めます。
- 演算結果は[D]を先頭とするエリアに格納します。  
ASIN([S]) → (D)

## ■ 処理内容

例) 演算単位が単精度浮動小数点実数(SF)の場合([S]に15° のSIN値を指定)

[i]...SF

[S]...DT10 [D]...DT2

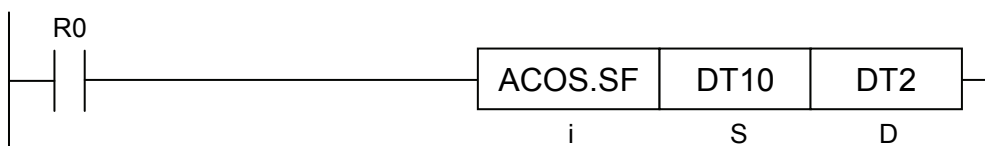
	角度	値(ラジアン)		値
DT10・DT11	15°	SF2.588190E-01	DT0・DT1	SF0.000000E+00
DT12・DT13	30°	SF1.047198E+00	DT2・DT3	SF2.617994E-01
DT14・DT15	45°	SF1.570796E+00	DT4・DT5	SF0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S](角度データ)に非実数を指定した場合にセットします
(ER)	[S](角度データ)が指定可能範囲外の場合にセットします

# ACOS (逆余弦演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	角度データ(デバイスアドレスまたは定数)(COS 値)(データ指定可能範囲:-1.0 ~ +1.0)
D	演算結果(デバイスアドレス)(単位:ラジアン)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4		" "
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]に格納された COS 値の ACOS(逆余弦)を求めます。  
ACOS([S]) → (D)

## ■ 処理内容

例) 演算単位が単精度浮動小数点形実数(SF)の場合([S]に15° のCOS値を指定)

[i]...SF  
[S]...DT10 [D]...DT2

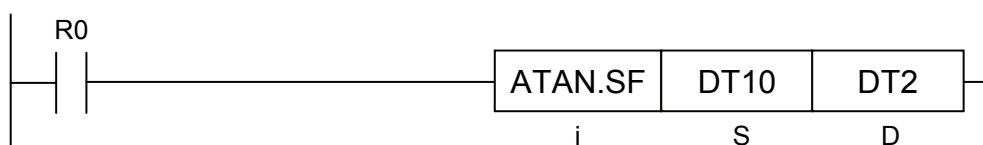
	角度	値(ラジアン)		DT0・DT1	値
DT10・DT11	15°	SF9.659258E-01	→	DT0・DT1	SF0.000000E+00
DT12・DT13	30°	SF1.047198E+00		DT2・DT3	SF2.617994E-01
DT14・DT15	45°	SF1.570796E+00		DT4・DT5	SF0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S](角度データ)に非実数を指定した場合にセットします
(ER)	[S](角度データ)が指定可能範囲外の場合にセットします

# ATAN (逆正接演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	角度データ(デバイスアドレスまたは定数)(TAN 値)
D	演算結果(デバイスアドレス)(単位:ラジアン)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾 *1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4	..	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]に格納された TAN 値の ATAN(逆正接)を求めます。
- 演算結果は[D]を先頭とするエリアに格納します。  
ATAN([S]) → (D)

## ■ 処理内容

例) 演算単位が単精度浮動小数点形実数(SF)の場合([S]に15° のTAN値を指定)

[i]...SF

[S]...DT10 [D]...DT2

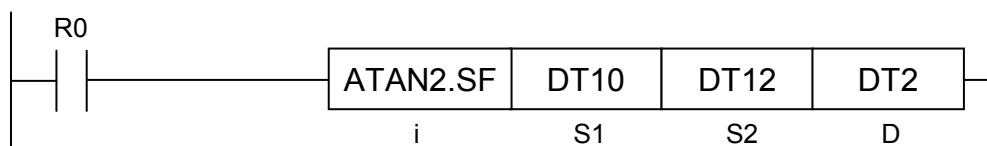
	角度	値(ラジアン)		値
DT10・DT11	15°	SF 2.679392E-01	DT0・DT1	SF 0.000000E+00
DT12・DT13	30°	SF 1.047198E+00	DT2・DT3	SF 2.617994E-01
DT14・DT15	45°	SF 1.570796E+00	DT4・DT5	SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	[S](角度データ)に非実数を指定した場合にセットします

# ATAN2 (座標データ→角度ラジアン変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S1	角度データの被除数(デバイスアドレスまたは定数)(y 座標)
S2	角度データ除数(デバイスアドレスまたは定数)(x 座標)
D	演算結果(デバイスアドレス)(単位:ラジアン)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4	""	*1
S1	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
S2	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●								●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S1]で指定された y 座標、[S2]で指定された x 座標から、ATAN (単位:ラジアン)を求めます。
- 演算結果は[D]を先頭とするエリアに格納します。  
ATAN2([S1]、[S2]) → [D]

## ■ 処理内容

例) 演算単位が単精度浮動小数点形実数(SF)の場合([S1](y座標)と[S2](x座標)に1.0を指定)

[i]...SF  
[S1]...DT10 [S2]...DT12 [D]...DT2

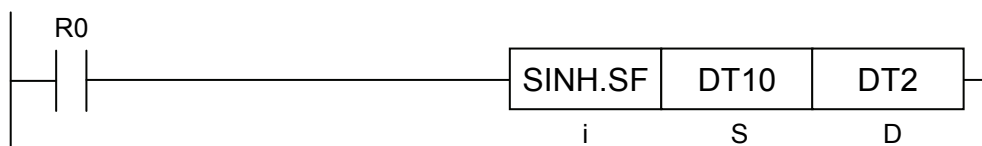
	値		値
DT10・DT11	SF 1.000000E+00	DT0・DT1	SF 0.000000E+00
DT12・DT13	SF 1.000000E+00	DT2・DT3	SF 7.853982E-01
DT14・DT15	SF 0.000000E+00	DT4・DT5	SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S1](y 座標) または [S2](x 座標)に非実数を指定した場合にセットします
(ER)	[S1](y 座標)に 0.0 かつ [S2](x 座標)に 0.0 を指定した場合にセットします

# SINH (双曲線正弦演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	角度データ(デバイスアドレスまたは定数)(単位:ラジアン)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾 *1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4	..	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可

## ■ 動作説明

- [i]の演算単位に従って、[S]に格納された角度データ(単位:ラジアン)の SINH(双曲線正弦)を求めます。
- 演算結果は[D]を先頭とするエリアに格納します。  
SINH([S]) → [D]

## ■ 処理内容

例) 演算単位が単精度浮動小数点形実数(SF)の場合([S]に30° のラジアンを指定)

[i]...SF

[S]...DT10 [D]...DT2

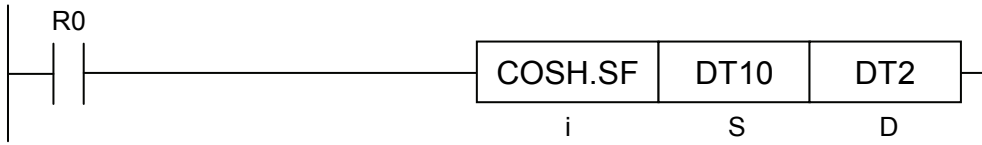
	角度	値(ラジアン)		値
DT10・DT11	30°	SF 5.235988E-01	DT0・DT1	SF 0.000000E+00
DT12・DT13	60°	SF 1.047198E+00	DT2・DT3	SF 5.478535E-01
DT14・DT15	90°	SF 1.570796E+00	DT4・DT5	SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	[S](角度データ)に非実数を指定した場合にセットします

# COSH (双曲線余弦演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	角度データ(デバイスアドレスまたは定数)(単位:ラジアン)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4		""
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可

## ■ 動作説明

- [i]の演算単位に従って、[S]に格納された角度データ(単位:ラジアン)の COSH(双曲線余弦) を求めます。
- 演算結果は[D]を先頭とするエリアに格納します。  
COSH([S]) → [D]

## ■ 処理内容

例) 演算単位が単精度浮動小数点形実数(SF)の場合([S]に30° のラジアンを指定)

[i]...SF  
[S]...DT10 [D]...DT2

	角度	値(ラジアン)		値
DT10・DT11	30°	SF 5.235988E-01	DT0・DT1	SF 0.000000E+00
DT12・DT13	60°	SF 1.047198E+00	DT2・DT3	SF 1.140238E+00
DT14・DT15	90°	SF 1.570796E+00	DT4・DT5	SF 0.000000E+00

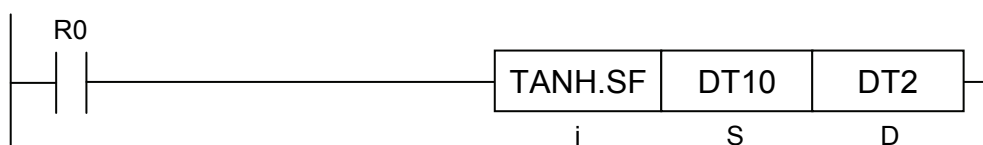
## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	



# TANH (双曲線正接演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	角度データ(デバイスアドレスまたは定数)(単位:ラジアン)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4	..	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可

## ■ 動作説明

- [i]の演算単位に従って、[S]に格納された角度データ(単位:ラジアン)のTANH(双曲線正接)を求めます。
- 演算結果は[D]を先頭とするエリアに格納します。  
TANH([S]) → [D]

## ■ 処理内容

例) 演算単位が単精度浮動小数点形実数(SF)の場合([S]に30°のラジアンを指定)

[i]...SF

[S]...DT10 [D]...DT2

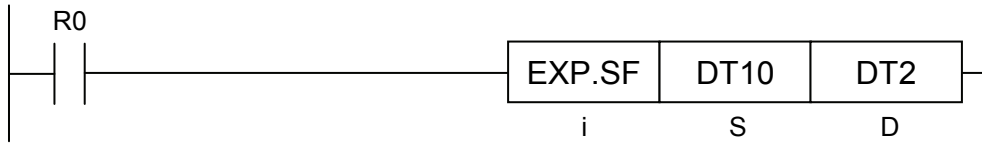
	角度	値(ラジアン)		値	
DT10・DT11	30°	SF 5.235988E-01	→	DT0・DT1	SF 0.000000E+00
DT12・DT13	60°	SF 1.047198E+00		DT2・DT3	SF 4.804728E-01
DT14・DT15	90°	SF 1.570796E+00		DT4・DT5	SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	

# EXP (指数演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	演算対象データ(デバイスアドレスまたは定数)(実数値)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4		" "
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]先頭とするエリアに格納された実数値の EXP(正接)を求めます。
- 演算結果は[D]を先頭とするエリアに格納します。  
EXP([S]) → [D]

## ■ 処理内容

例) 演算単位が単精度浮動小数点形実数(SF)の場合

[i]...SF

[S]...DT10 [D]...DT2

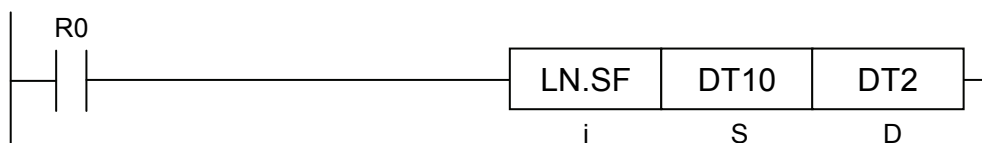
値(ラジアン)		値	
DT10・DT11	SF 3.000000E+00	DT0・DT1	SF 0.000000E+00
DT12・DT13	SF 4.000000E+00	DT2・DT3	SF 2.008554E+01
DT14・DT15	SF 5.000000E+00	DT4・DT5	SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	

# LN (自然対数演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	演算対象データ(デバイスアドレスまたは定数)(実数値)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾 *1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4	..	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(10~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]先頭とするエリアに格納された実数値のLN(自然対数)を求めます。
- 演算結果は[D]を先頭とするエリアに格納します。  
LN([S]) → [D]

## ■ 処理内容

- LN(演算対象データ)を演算し演算結果に設定します。

例) 演算単位が単精度浮動小数点形実数(SF)の場合

[i]...SF

[S]...DT10 [D]...DT2

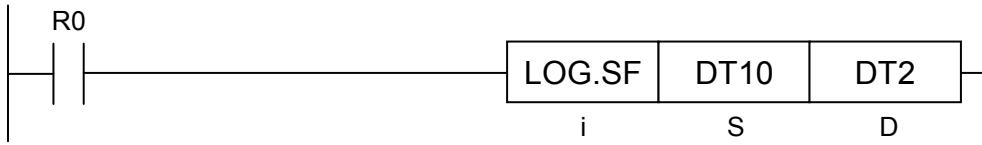
	値(ラジアン)		値
DT10・DT11	SF 3.000000E+00	→	DT0・DT1 SF 0.000000E+00
DT12・DT13	SF 4.000000E+00		DT2・DT3 SF 1.098612E+00
DT14・DT15	SF 5.000000E+00		DT4・DT5 SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S](演算対象データ)に非実数を指定した場合にセットします
(ER)	[S](演算対象データ)に0.0以下を指定の場合にセットします

# LOG (常用対数演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	演算対象データ(デバイスアドレスまたは定数)(実数値)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4		" "
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]先頭とするエリアに格納された実数値の LOG(常用対数)を求めます。
- 演算結果は[D]を先頭とするエリアに格納します。  
LOG([S]) → [D]

## ■ 処理内容

例) 演算単位が単精度浮動小数点形実数(SF)の場合

[i]...SF

[S]...DT10 [D]...DT2

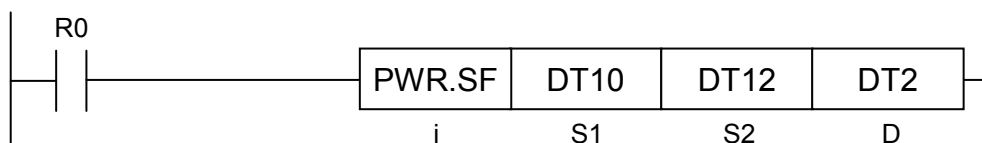
値(ラジアン)		値	
DT10・DT11	SF 3.000000E+00	DT0・DT1	SF 0.000000E+00
DT12・DT13	SF 4.000000E+00	DT2・DT3	SF 4.771213E-01
DT14・DT15	SF 5.000000E+00	DT4・DT5	SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S](演算対象データ)に非実数を指定した場合にセットします
(ER)	[S](演算対象データ)に0.0以下を指定の場合にセットします

# PWR (べき乗演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S1	被べき乗データ(デバイスアドレスまたは定数)(実数値)
S2	べき乗データ(デバイスアドレスまたは定数)(実数値)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾*1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4	""	*
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●	●	●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(10~IE)

\*3: 演算単位が単精度浮動小数点実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S1]で先頭とするエリアに格納された実数値と、[S2]で先頭とするエリアに格納された実数値のべき乗を求めます。
- 演算結果は[D]を先頭とするエリアに格納します。  
 $[S1] \wedge [S2] \rightarrow [D]$

## ■ 処理内容

例) 演算単位が単精度浮動小数点実数(SF)の場合

[i]...SF

[S1]...DT10 [S2]...DT12 [D]...DT2

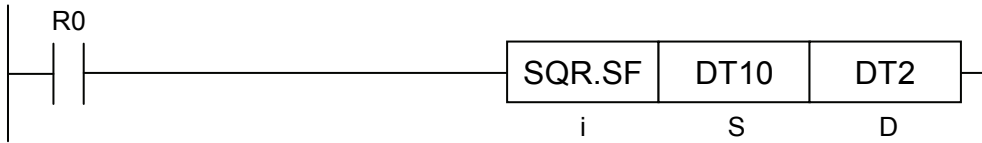
	値(ラジアン)		値	
DT10・DT11	SF 3.000000E+00	→	DT0・DT1	SF 0.000000E+00
DT12・DT13	SF 4.000000E+00		DT2・DT3	SF 8.100000E+01
DT14・DT15	SF 5.000000E+00		DT4・DT5	SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S1](被べき乗データ) または [S2](べき乗データ) に非実数を指定した場合にセットします
(ER)	[S1](被べき乗データ)に 0.0 かつ [S2](べき乗データ) に 0.0 以下を指定した場合にセットします
	[S1](被べき乗データ)に負の値 かつ [S2](べき乗データ) に整数値以外を指定した場合にセットします

# SQR (平方根演算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	演算対象データ(デバイスアドレスまたは定数)(実数値)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4		" "
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]を先頭とするエリアに格納された実数値の平方根を求めます。
- 演算結果は[D]を先頭とするエリアに格納します。  
SQR([S]) → [D]

## ■ 処理内容

例) 演算単位が単精度浮動小数点形実数(SF)の場合

[i]...SF

[S]...DT10 [D]...DT2

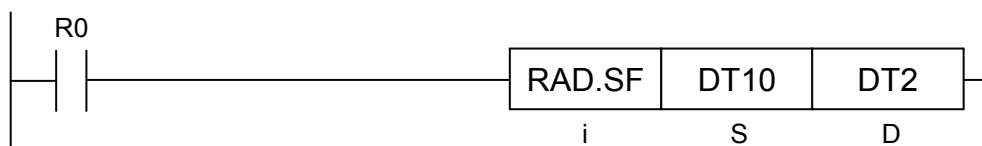
値(ラジアン)		値	
DT10・DT11	SF 3.000000E+00	DT0・DT1	SF 0.000000E+00
DT12・DT13	SF 4.000000E+00	DT2・DT3	SF 1.732051E+00
DT14・DT15	SF 5.000000E+00	DT4・DT5	SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S](演算対象データ)に非実数を指定した場合にセットします
(ER)	[S](演算対象データ)に負の値を指定の場合にセットします

# RAD (角度→ラジアン変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	角度データのデバイスアドレスまたは定数(単位:度)
D	角度データのデバイスアドレス(単位:ラジアン)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾 *1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4	..	
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]を先頭とするエリアに格納された角度データ(単位:度)を角度データ(単位:ラジアン)に変換します。
- 演算結果は[D]を先頭とするエリアに格納します。  
[S] × ( π ÷ 180 ) → [D]

## ■ 処理内容

例) 演算単位が単精度浮動小数点形実数(SF)の場合([S]に30(度)を指定)

[i]...SF

[S]...DT10

[D]...DT2

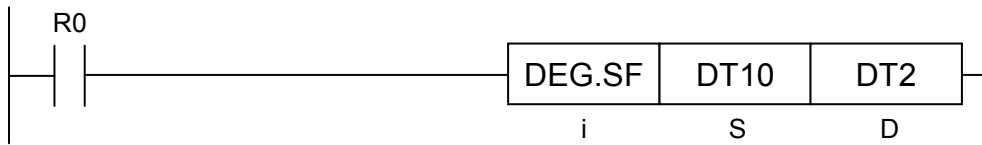
	値(度)		値(ラジアン)
DT10・DT11	SF 3.000000E+01	DT0・DT1	SF 0.000000E+00
DT12・DT13	SF 6.000000E+01	DT2・DT3	SF 5.235988E-01
DT14・DT15	SF 9.000000E+01	DT4・DT5	SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	[S](角度データ)に非実数を指定した場合にセットします

# DEG (ラジアン→角度変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	角度データのデバイスアドレスまたは定数(単位:ラジアン)
D	角度データのデバイスアドレス(単位:度)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4		""
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]を先頭とするエリアに格納された角度データ(単位:ラジアン)を角度データ(単位:度)に変換します。
- 演算結果は[D]を先頭とするエリアに格納します。  
 $[S] \times (180 \div \pi) \rightarrow [D]$

## ■ 処理内容

例) 演算単位が単精度浮動小数点形実数(SF)の場合([S]に30° のラジアンを指定)

[i]...SF  
[S]...DT10 [D]...DT2

角度	値(ラジアン)	値(度)
DT10・DT11	30° SF 5.235988E-01	DT0・DT1 SF 0.000000E+00
DT12・DT13	60° SF 1.047198E+00	DT2・DT3 SF 3.000000E+01
DT14・DT15	90° SF 1.570796E+00	DT4・DT5 SF 0.000000E+00

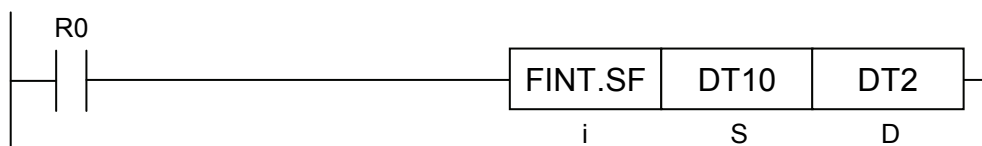
## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	[S](角度データ)に非実数を指定した場合にセットします



# FINT (浮動小数点形実数データ 小数点以下切り捨て)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	演算対象データ(デバイスアドレスまたは定数)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデック ス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4		..
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]を先頭とするエリアに格納された実数値の小数点以下を切り捨てます。
- 演算結果は[D]を先頭とするエリアに格納します。

## ■ 処理内容

例1) 演算単位が単精度浮動小数点形実数(SF)の場合(正の実数)

[i]...SF		
[S]...DT10	[D]...DT2	
DT10・DT11	SF 1.234560E+02	DT0・DT1
DT12・DT13	SF 3.456780E+02	DT2・DT3
DT14・DT15	SF 5.678900E+02	DT4・DT5
		SF 0.000000E+00
		SF 1.230000E+02
		SF 0.000000E+00

例2) 演算単位が単精度浮動小数点形実数(SF)の場合(負の値)

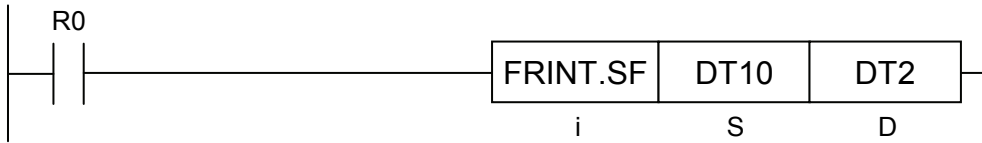
[i]...SF		
[S]...DT10	[D]...DT2	
DT10・DT11	SF -1.234560E+02	DT0・DT1
DT12・DT13	SF -3.456780E+02	DT2・DT3
DT14・DT15	SF -5.678900E+02	DT4・DT5
		SF 0.000000E+00
		SF -1.230000E+02
		SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	[S]に非実数を指定した場合にセットします

# FRINT (浮動小数点形実数データ 小数点四捨五入)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	演算対象データ(デバイスアドレスまたは定数)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4		""
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]を先頭とするエリアに格納された実数値の小数点以下第1位を四捨五入します。
- 演算結果は[D]を先頭とするエリアに格納します。

## ■ 処理内容

例1) 演算単位が単精度浮動小数点形実数(SF)の場合(正の実数)

[i]...SF

[S]...DT10 [D]...DT2

DT10・DT11	SF 1.234560E+02	DT0・DT1	SF 0.000000E+00
DT12・DT13	SF 3.456780E+02	DT2・DT3	SF 1.230000E+02
DT14・DT15	SF 5.678900E+02	DT4・DT5	SF 0.000000E+00

例2) 演算単位が単精度浮動小数点形実数(SF)の場合(負の値)

[i]...SF

[S]...DT10 [D]...DT2

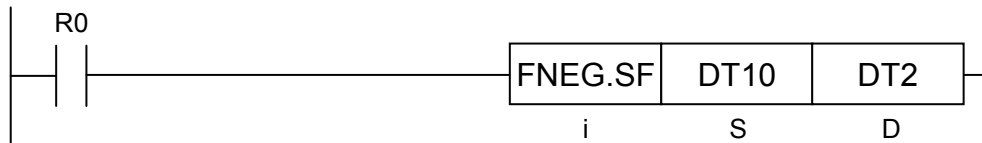
DT10・DT11	SF -1.234560E+02	DT0・DT1	SF 0.000000E+00
DT12・DT13	SF -3.456780E+02	DT2・DT3	SF -1.230000E+02
DT14・DT15	SF -5.678900E+02	DT4・DT5	SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	[S]に非実数を指定した場合にセットします

# FNEG (浮動小数点形実数データ 符号変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	演算対象データ(デバイスアドレスまたは定数)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデック ス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4		..
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]を先頭とするエリアに格納された実数値の符号を反転します。
- 演算結果は[D]を先頭とするエリアに格納します。

## ■ 処理内容

例1) 演算単位が単精度浮動小数点形実数(SF)の場合(正の実数)

[i]...SF		
[S]...DT10	[D]...DT2	
DT10・DT11	SF 1.234560E+02	DT0・DT1 SF 0.000000E+00
DT12・DT13	SF 3.456780E+02	DT2・DT3 SF -1.234560E+02
DT14・DT15	SF 5.678900E+02	DT4・DT5 SF 0.000000E+00

例2) 演算単位が単精度浮動小数点形実数(SF)の場合(負の値)

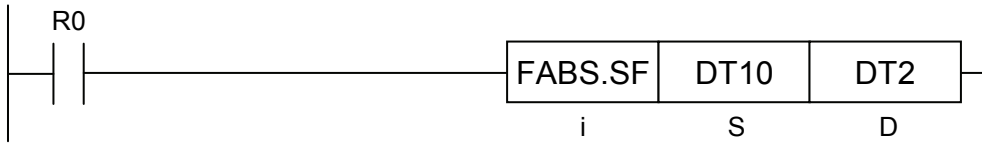
[i]...SF		
[S]...DT10	[D]...DT2	
DT10・DT11	SF -1.234560E+02	DT0・DT1 SF 0.000000E+00
DT12・DT13	SF -3.456780E+02	DT2・DT3 SF 1.234560E+02
DT14・DT15	SF -5.678900E+02	DT4・DT5 SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	[S]に非実数を指定した場合にセットします

# FABS (浮動小数点形実数データ 絶対値)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i						●	●

## ■ オペランド一覧

オペランド	説明
S	演算対象データ(デバイスアドレスまたは定数)
D	演算結果(デバイスアドレス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾*1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K	U	H	SF *3	DF *4		" "
S	●	●	●	●			●	●	●	●	●	●	●	●				●	●		●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 16ビットデバイス、32ビットデバイスのみ修飾可能(実数定数には指定不可)

\*2: インデックスレジスタ(I0~IE)

\*3: 演算単位が単精度浮動小数点形実数(SF)の場合のみ指定可能

\*4: 演算単位が倍精度浮動小数点形実数(DF)の場合のみ指定可能

## ■ 動作説明

- [i]の演算単位に従って、[S]を先頭とするエリアに格納された実数値の絶対値を求めます。
- 演算結果は[D]を先頭とするエリアに格納します。

## ■ 処理内容

例1) 演算単位が単精度浮動小数点形実数(SF)の場合(正の実数)

[i]...SF		
[S]...DT10	[D]...DT2	
DT10・DT11	SF 1.234560E+02	DT0・DT1
DT12・DT13	SF 3.456780E+02	DT2・DT3
DT14・DT15	SF 5.678900E+02	DT4・DT5
		SF 0.000000E+00
		SF 1.234560E+02
		SF 0.000000E+00

例2) 演算単位が単精度浮動小数点形実数(SF)の場合(負の値)

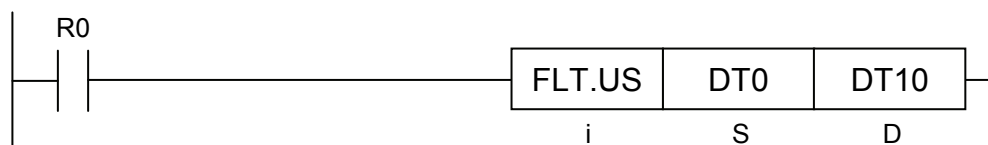
[i]...SF		
[S]...DT10	[D]...DT2	
DT10・DT11	SF -1.234560E+02	DT0・DT1
DT12・DT13	SF -3.456780E+02	DT2・DT3
DT14・DT15	SF -5.678900E+02	DT4・DT5
		SF 0.000000E+00
		SF 1.234560E+02
		SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	[S]に非実数を指定した場合にセットします

# FLT (整数→浮動小数点形実数データ変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S	変換対象データ(デバイスアドレスまたは定数(データ形式 : 演算単位に従う))
D	変換結果(デバイスアドレス(データ形式 : 単精度浮動小数点形実数データ))

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *2	K *3	U *4	H	SF	DF	""	
S	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				●
D	●	●	●	●			●	●	●		●	●	●	●							●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: インデックスレジスタ(I0～IE)

\*3: 演算単位が符号付き整数(SS、SL)の場合のみ指定可能

\*4: 演算単位が符号なし整数(US、UL)の場合のみ指定可能

## ■ 動作説明

- 演算単位[i]の内容に従って、[S]を先頭とするエリアに格納された整数値を単精度浮動小数点形実数値に変換します。
- 演算結果は[D]を先頭とするエリアに格納します。

## ■ 処理内容

例1) 符号なし16ビット(US)の場合

[i]...US

[S]...DT0 [D]...DT10

DT0	<b>U 123</b>
DT1	U 456
DT2	U 789



DT10・DT11	<b>SF 1.230000E+02</b>
DT12・DT13	SF 0.000000E+00
DT14・DT15	SF 0.000000E+00

例2) 符号付き16ビット(SS)の場合(正の値)

[i]...SS

[S]...DT20 [D]...DT10

DT20	<b>K 123</b>
DT21	K 456
DT22	K 789



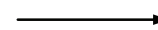
DT10・DT11	<b>SF 1.230000E+02</b>
DT12・DT13	SF 0.000000E+00
DT14・DT15	SF 0.000000E+00

例3) 符号付き16ビット(SS)の場合(負の値)

[i]...SS

[S]...DT20 [D]...DT10

DT20	<b>K -123</b>
DT21	K -456
DT22	K -789



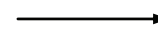
DT10・DT11	<b>SF -1.230000E+02</b>
DT12・DT13	SF 3.456780E+02
DT14・DT15	SF 5.678900E+02

例4) 符号なし32ビット(UL)の場合

[i]...UL

[S]...DT0 [D]...DT10

DT0・DT1	<b>U 12345</b>
DT2・DT3	U 67890
DT4・DT5	U 13579



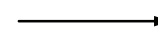
DT10・DT11	<b>SF 1.234500E+04</b>
DT12・DT13	SF 0.000000E+00
DT14・DT15	SF 0.000000E+00

例5) 符号付き32ビット(SL)の場合(正の値)

[i]...SL

[S]...DT20 [D]...DT10

DT20・DT21	<b>K 12345</b>
DT22・DT23	K 67890
DT24・DT25	K 13579



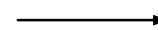
DT10・DT11	<b>SF 1.234500E+04</b>
DT12・DT13	SF 0.000000E+00
DT14・DT15	SF 0.000000E+00

例6) 符号付き32ビット(SL)の場合(負の値)

[i]...SL

[S]...DT20 [D]...DT10

DT20・DT21	<b>K -12345</b>
DT22・DT23	K -67890
DT24・DT25	K -13579



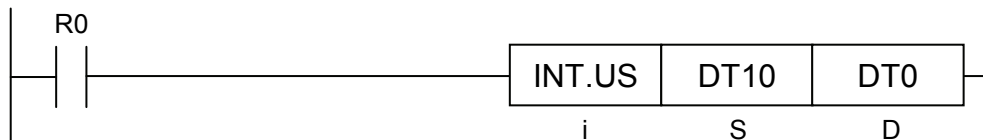
DT10・DT11	<b>SF -1.234500E+04</b>
DT12・DT13	SF 0.000000E+00
DT14・DT15	SF 0.000000E+00

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします

# INT (浮動小数点形実数データ→整数変換) (超えない最大)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S	変換対象データ(デバイスアドレスまたは定数(データ形式 : 演算単位に従う))
D	変換対象データ(デバイスアドレスまたは定数(データ形式 : 演算単位に従う))

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	""	
S	●	●	●	●			●	●	●	●	●	●	●	●	●	●	●	●			●
D	●	●	●	●			●	●	●	●	●	●	●	●							●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイスのみ修飾可能(実数定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- 演算単位[i]の内容に従って、[S]を先頭とするエリアに格納された単精度浮動小数点形実数値を整数値(超えない最大値)に変換します。
- 演算結果は[D]を先頭とするエリアに格納します。
- 演算単位別の変換対象データ[S]のデータ指定範囲を以下に示します。

	最小値	最大値
US	0	65,530
SS	-32,760	32,760
UL	0	4,294,967,000
SL	-2,147,483,000	2,147,483,000

## ■ 処理内容

例1) 符号なし16ビット(US)の場合(正の値)

[I]...US

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF 2.345670E+02</b>	→	DT0	<b>U 234</b>
DT12・DT13	SF 3.456780E+02		DT1	U 0
DT14・DT15	SF 4.567890E+02		DT2	U 0

例2) 符号なし16ビット(US)の場合(負の値)

[I]...US

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF -2.345670E+02</b>	→	DT0	U 0
DT12・DT13	SF -3.456780E+02		DT1	U 0
DT14・DT15	SF -4.567890E+02		DT2	U 0

※演算単位に符号なし整数を指定して負の値を変換すると演算エラーになります

例3) 符号付き16ビット(SS)の場合(正の値)

[I]...SS

[S]...DT10 [D]...DT20

DT10・DT11	<b>SF 2.345670E+02</b>	→	DT20	<b>K 234</b>
DT12・DT13	SF 3.456780E+02		DT21	K 0
DT14・DT15	SF 4.567890E+02		DT22	K 0

例4) 符号付き16ビット(SS)の場合(負の値)

[I]...SS

[S]...DT10 [D]...DT20

DT10・DT11	<b>SF -2.345670E+02</b>	→	DT20	<b>K -235</b>
DT12・DT13	SF -3.456780E+02		DT21	K 0
DT14・DT15	SF -4.567890E+02		DT22	K 0

例5) 符号なし32ビット(UL)の場合(正の値)

[I]...UL

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF 1.234567E+05</b>	→	DT0・DT1	<b>U 123456</b>
DT12・DT13	SF 2.468000E+02		DT2・DT3	K 0
DT14・DT15	SF 1.357000E+02		DT4・DT5	K 0



例6) 符号なし32ビット(UL)の場合(負の値)

[i]...UL

[S]...DT10 [D]...DT0

DT10・DT11	SF -1.234567E+05	→	DT0・DT1	U 0
DT12・DT13	SF -2.468000E+02		DT2・DT3	U 0
DT14・DT15	SF -1.357000E+02		DT4・DT5	U 0

※演算単位に符号なし整数を指定して負の値を変換すると演算エラーになります

例7) 符号付き32ビット(SL)の場合(正の値)

[i]...SL

[S]...DT10 [D]...DT20

DT10・DT11	SF 1.234567E+05	→	DT20・DT21	K 123456
DT12・DT13	SF 2.468000E+02		DT22・DT23	K 0
DT14・DT15	SF 1.357000E+02		DT24・DT25	K 0

例8) 符号付き32ビット(SL)の場合(負の値)

[i]...SL

[S]...DT10 [D]...DT20

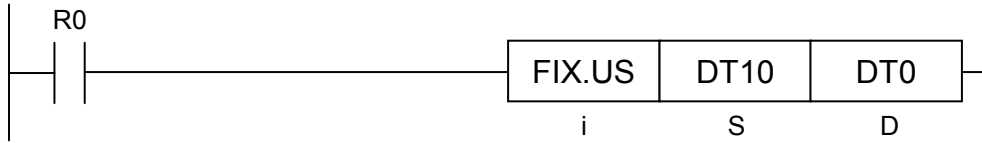
DT10・DT11	SF -1.234567E+05	→	DT20・DT21	K -12346
DT12・DT13	SF -2.468000E+02		DT22・DT23	K 0
DT14・DT15	SF -1.357000E+02		DT24・DT25	K 0

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S]に非実数を指定した場合にセットします
(ER)	[S](変換対象データ)に範囲外の値を指定された場合にセットします

# **FIX (浮動小数点形実数データ→整数 (小数点以下切り捨て)変換)**

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S	変換対象データ(デバイスアドレスまたは定数(データ形式 : 演算単位に従う))
D	変換対象データ(デバイスアドレスまたは定数(データ形式 : 演算単位に従う))

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス *1			整数			実数		文字	インデック ス修飾 *2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●	●	●	●	●	●	●				●			●
D	●	●	●	●			●	●	●	●	●	●	●	●							●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイスのみ修飾可能(実数定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- 演算単位[i]の内容に従って、[S]を先頭とするエリアに格納された単精度浮動小数点形実数値を整数値(小数点以下切り捨て)に変換します。
- 演算結果は[D]を先頭とするエリアに格納します。
- 演算単位別の変換対象データ[S]のデータ指定範囲を以下に示します。

	最小値	最大値
US	0	65,530
SS	-32,760	32,760
UL	0	4,294,967,000
SL	-2,147,483,000	2,147,483,000

## ■ 処理内容

例1) 符号なし16ビット(US)の場合(正の値)

[i]...US

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF 2.345670E+02</b>	→	DT0	<b>U 234</b>
DT12・DT13	SF 3.456780E+02		DT1	U 0
DT14・DT15	SF 4.567890E+02		DT2	U 0

例2) 符号なし16ビット(US)の場合(負の値)

[i]...US

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF -2.345670E+02</b>	→	DT0	U 0
DT12・DT13	SF -3.456780E+02		DT1	U 0
DT14・DT15	SF -4.567890E+02		DT2	U 0

※演算単位に符号なし整数を指定して負の値を変換すると演算エラーになります

例3) 符号付き16ビット(SS)の場合(正の値)

[i]...SS

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF 2.345670E+02</b>	→	DT0	<b>K 234</b>
DT12・DT13	SF 3.456780E+02		DT1	K 0
DT14・DT15	SF 4.567890E+02		DT2	K 0

例4) 符号付き16ビット(SS)の場合(負の値)

[i]...SS

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF -2.345670E+02</b>	→	DT0	<b>K -234</b>
DT12・DT13	SF -3.456780E+02		DT1	K 0
DT14・DT15	SF -4.567890E+02		DT2	K 0

例5) 符号なし32ビット(UL)の場合(正の値)

[i]...UL

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF 1.234567E+05</b>	→	DT0・DT1	<b>U 123456</b>
DT12・DT13	SF 2.468000E+02		DT2・DT3	U 0
DT14・DT15	SF 1.357000E+02		DT4・DT5	U 0

例6) 符号なし32ビット(UL)の場合(負の値)

[i]...UL

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF -1.234567E+05</b>	→	DT0・DT1	U 0
DT12・DT13	SF -2.468000E+02		DT2・DT3	U 0
DT14・DT15	SF -1.357000E+02		DT4・DT5	U 0

※演算単位に符号なし整数を指定して負の値を変換すると演算エラーになります

例7) 符号付き32ビット(SL)の場合(正の値)

[i]...SL

[S]...DT10 [D]...DT20

DT10・DT11	<b>SF 1.234567E+05</b>	→	DT20・DT21	<b>K 123456</b>
DT12・DT13	SF 2.468000E+02		DT22・DT23	K 0
DT14・DT15	SF 1.357000E+02		DT24・DT25	K 0

例8) 符号付き32ビット(SL)の場合(負の値)

[i]...SL

[S]...DT10 [D]...DT20

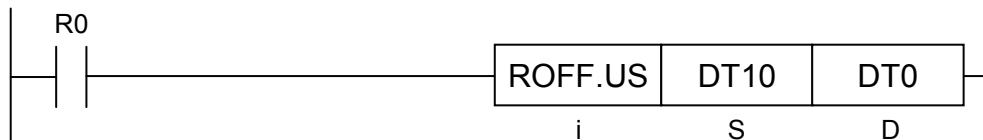
DT10・DT11	<b>SF -1.234567E+05</b>	→	DT20・DT21	<b>K -12345</b>
DT12・DT13	SF -2.468000E+02		DT22・DT23	K 0
DT14・DT15	SF -1.357000E+02		DT24・DT25	K 0

■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S]に非実数を指定した場合にセットします
(ER)	[S](変換対象データ)に範囲外の値を指定された場合にセットします

# ROFF (浮動小数点形実数データ→整数変換) (小数点以下四捨五入)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●	●	●		

## ■ オペランド一覧

オペランド	説明
S	変換対象データ(デバイスアドレスまたは定数(データ形式 : 演算単位に従う))
D	変換対象データ(デバイスアドレスまたは定数(データ形式 : 演算単位に従う))

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス*1			整数			実数		文字	インデックス修飾*2
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *3	K	U	H	SF	DF	""	
S	●	●	●	●			●	●	●	●	●	●	●	●				●			●
D	●	●	●	●			●	●	●	●	●	●	●	●							●

\*1: 演算単位が 16 ビット整数(SS、US)の場合は指定不可

\*2: 16 ビットデバイス、32 ビットデバイスのみ修飾可能(実数定数、文字定数には指定不可)

\*3: インデックスレジスタ(I0~IE)

## ■ 動作説明

- 演算単位[i]の内容に従って、[S]を先頭とするエリアに格納された単精度浮動小数点形実数値を整数値(小数点以下四捨五入)に変換します。
- 演算結果は[D]を先頭とするエリアに格納します。
- 演算単位別の変換対象データ[S]のデータ指定範囲を以下に示します。

	最小値	最大値
US	0	65,530
SS	-32,760	32,760
UL	0	4,294,967,000
SL	-2,147,483,000	2,147,483,000

## ■ 処理内容

例1) 符号なし16ビット(US)の場合(正の値)

[i]...US

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF 2.345670E+02</b>	→	DT0	<b>U 235</b>
DT12・DT13	SF 3.456780E+02		DT1	U 0
DT14・DT15	SF 4.567890E+02		DT2	U 0

例2) 符号なし16ビット(US)の場合(負の値)

[i]...US

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF -2.345670E+02</b>	→	DT0	U 0
DT12・DT13	SF -3.456780E+02		DT1	U 0
DT14・DT15	SF -4.567890E+02		DT2	U 0

※演算単位に符号なし整数を指定して負の値を変換すると演算エラーになります

例3) 符号付き16ビット(SS)の場合(正の値)

[i]...SS

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF 2.345670E+02</b>	→	DT0	<b>K 235</b>
DT12・DT13	SF 3.456780E+02		DT1	K 0
DT14・DT15	SF 4.567890E+02		DT2	K 0

例4) 符号付き16ビット(SS)の場合(負の値)

[i]...SS

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF -2.345670E+02</b>	→	DT0	<b>K -235</b>
DT12・DT13	SF -3.456780E+02		DT1	K 0
DT14・DT15	SF -4.567890E+02		DT2	K 0

例5) 符号なし32ビット(UL)の場合(正の値)

[i]...UL

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF 1.234567E+05</b>	→	DT0・DT1	<b>U 123457</b>
DT12・DT13	SF 2.468000E+02		DT2・DT3	U 0
DT14・DT15	SF 1.357000E+02		DT4・DT5	U 0

例6) 符号なし32ビット(UL)の場合(負の値)

[i]...UL

[S]...DT10 [D]...DT0

DT10・DT11	<b>SF -1.234567E+05</b>	→	DT0・DT1	U 0
DT12・DT13	SF -2.468000E+02		DT2・DT3	U 0
DT14・DT15	SF -1.357000E+02		DT4・DT5	U 0

※演算単位に符号なし整数を指定して負の値を変換すると演算エラーになります

例7) 符号付き32ビット(SL)の場合(正の値)

[i]...SL

[S]...DT10 [D]...DT20

DT10・DT11	<b>SF 1.234567E+05</b>	→	DT20・DT21	<b>K 123457</b>
DT12・DT13	SF 2.468000E+02		DT22・DT23	K 0
DT14・DT15	SF 1.357000E+02		DT24・DT25	K 0

例8) 符号付き32ビット(SL)の場合(負の値)

[i]...SL

[S]...DT10 [D]...DT20

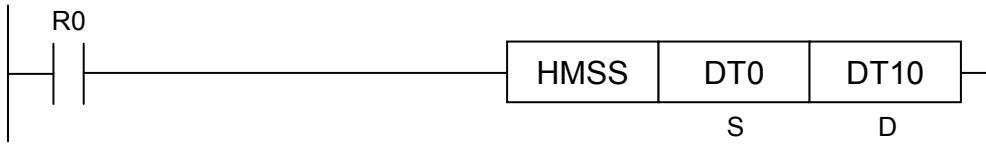
DT10・DT11	<b>SF -1.234567E+05</b>	→	DT20・DT21	<b>K -123457</b>
DT12・DT13	SF -2.468000E+02		DT22・DT23	K 0
DT14・DT15	SF -1.357000E+02		DT24・DT25	K 0

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S]に非実数を指定した場合にセットします
(ER)	[S](変換対象データ)に範囲外の値を指定された場合にセットします

# HMSS (時間(時分秒)→秒データ変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S	時間データの先頭デバイスアドレス
D	秒データのデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""		
S	●	●	●	●	●	●	●	●	●	●	●											●
D	●	●	●	●			●	●	●		●											●

## ■ 動作説明

- [S]を先頭とするエリアに格納された 時・分・秒を表す整数各 1 ワードのデータを、秒データを表す整数 2 ワードのデータに変換します。
- 演算結果は[D]を先頭とするエリアに格納します。

## ■ 処理内容

例) 3時間54分19秒を変換する場合

[S]...DT0 [D]...DT10

※1ワード

DT0	K 3	(時)
DT1	K 54	(分)
DT2	K 19	(秒)

変換

※2ワード

DT10・DT11	K 14059	(秒)

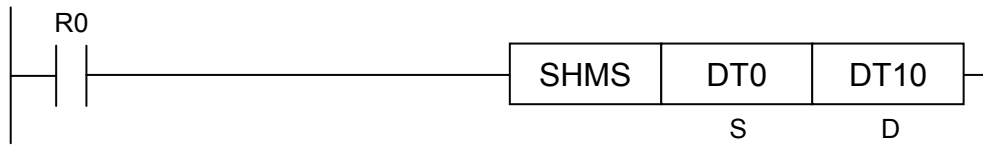
## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	
(ER)	



# SHMS (秒データ→時間(時分秒)変換)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S	秒データのデバイスアドレス(指定可能データ範囲 : 0~35,999,999)
D	時間データの先頭デバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *1	K	U	H	SF	DF	" "		
S	●	●	●	●	●	●	●	●	●	●	●											●
D	●	●	●	●			●	●	●		●											●

## ■ 動作説明

- [S]を先頭とするエリアに格納された 秒データを表す整数 2 ワードのデータを、時・分・秒を表す整数各 1 ワードのデータに変換します。
- 演算結果は[D]を先頭とするエリアに格納します。

## ■ 処理内容

例) 12,345秒を変換する場合

[S]...DT0 [D]...DT10

※2ワード

DT0	K 12345	(秒)
DT1		

変換

※1ワード

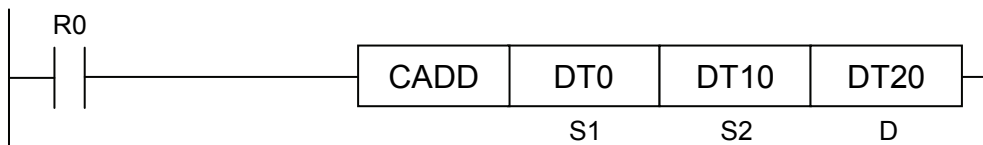
DT10	K 3	(時)
DT11	K 25	(分)
DT12	K 45	(秒)

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	秒データの範囲を超えている場合にセットします

# CADD (時刻加算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S1	時刻データの先頭デバイスアドレス
S2	時間データの先頭デバイスアドレス
D	加算結果の先頭デバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデッ クス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "		
S1	●	●	●	●			●	●														●
S2	●	●	●	●			●	●														●
D	●	●	●	●			●	●														●

## ■ 動作説明

- [S1]を先頭とするエリアに格納された時刻データ(年、月、日、時、分、秒)に[S2]を先頭とするエリアに格納された時間データ(時、分、秒)を加算します。
- 演算結果は[D]を先頭とするエリアに格納します。
- うるう年にも対応し、演算します。

## ■ 処理内容

例) 2012年01月01日8時54分19秒に20時間23分45秒を加算する場合  
[S1]...DT0 [S2]...DT10 [D]...DT20

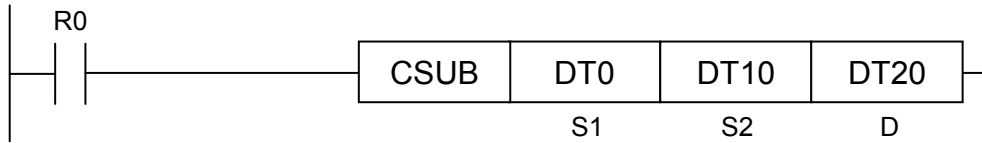
DT0	K 12	(年)																										DT20	K 12	(年)
DT1	K 1	(月)																									DT21	K 1	(月)	
DT2	K 1	(日)																									DT22	K 2	(日)	
DT3	K 8	(時)						K 20	(時)	+				K 5	(時)											DT23	K 5	(時)		
DT4	K 54	(分)						K 23	(分)					K 18	(分)											DT24	K 18	(分)		
DT5	K 19	(秒)						K 45	(秒)	→				K 4	(秒)											DT25	K 4	(秒)		

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	時刻データと時間データの範囲を超えている場合にセットします
(ER)	加算結果がアクセス可能範囲外の場合にセットします

# CSUB (時刻減算)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S1	時刻データの先頭デバイスアドレス
S2	時間データの先頭デバイスアドレス
D	減算結果の先頭デバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""		
S1	●	●	●	●			●	●														●
S2	●	●	●	●			●	●														●
D	●	●	●	●			●	●														●

## ■ 動作説明

- [S1]を先頭とするエリアに格納された時刻データ(年、月、日、時、分、秒)に[S2]を先頭とするエリアに格納された時間データ(時、分、秒)を減算します。
- 演算結果は[D]を先頭とするエリアに格納します。
- うるう年にも対応し、演算します。

## ■ 処理内容

例) 2012年01月01日8時54分19秒に20時間23分45秒を減算する場合  
[S1]...DT0 [S2]...DT10 [D]...DT20

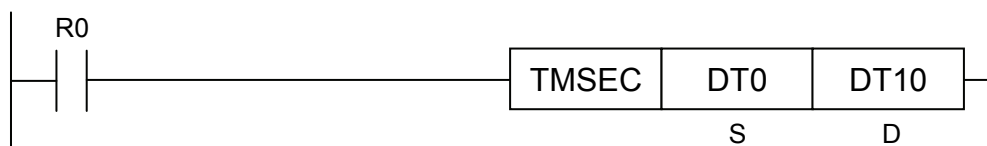
DT0	K 12	(年)	-	DT10	K 20	(時)	→	DT20	K 11	(年)
DT1	K 1	(月)		DT11	K 23	(分)		DT21	K 12	(月)
DT2	K 1	(日)		DT12	K 45	(秒)		DT22	K 31	(日)
DT3	K 8	(時)						DT23	K 12	(時)
DT4	K 54	(分)						DT24	K 30	(分)
DT5	K 19	(秒)						DT25	K 34	(秒)

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	時刻データと時間データの範囲を超えている場合にセットします
(ER)	減算結果がアクセス可能範囲外の場合にセットします

# TMSEC (時刻データ→基準時刻からの秒データ)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S	時刻データの先頭デバイスアドレス(指定可能データ範囲 : 1970/1/1 00:00:00~2106/2/6 23:59:59)
D	基準時刻からの秒データのデバイスアドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●	●	●	●	●	●	●	●										
D	●	●	●	●			●	●	●		●										

## ■ 動作説明

- [S]を先頭とするエリアに格納された時刻データ(年、月、日、時、分、秒)から、基準時刻からの時間を算出します。
- 演算結果は[D]を先頭とするエリアに格納します。
- 基準時刻は、2001年1月1日0時0分0秒です。

## ■ 処理内容

例) 2012年01月01日8時54分19秒から基準時刻からの秒データを計算する場合

[S]...DT0 [D]...DT10

※1ワード

DT0	K 12	(年)
DT1	K 1	(月)
DT2	K 1	(日)
DT3	K 8	(時)
DT4	K 54	(分)
DT5	K 19	(秒)

※2ワード

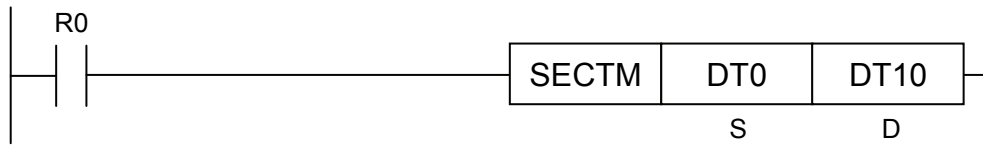
DT10・DT11	K 347100859	(基準秒データ)

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	時刻データの範囲を超えている場合にセットします

# SECTM (基準時刻からの秒データ→時刻データ)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S	基準時刻からの秒データのデバイスアドレス
D	時刻データの先頭デバイスアドレス(指定可能データ範囲 : 1970/1/1 00:00:00~2106/2/6 23:59:59)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *1	K	U	H	SF	DF	""		
S	●	●	●	●	●	●	●	●	●	●	●											●
D	●	●	●	●			●	●	●		●											●

## ■ 動作説明

- [S]を先頭とするエリアに格納された基準時刻からの時間データから、時刻データ(年、月、日、時、分、秒)を算出します。
- 演算結果は[D]を先頭とするエリアに格納します。
- 基準時刻=2001年1月1日0時0分0秒です。

## ■ 処理内容

例) 1,325,408,059秒からデータを計算する場合

[S]...DT0 [D]...DT10

※2ワード

DT0・DT1	K 1325408059	(基準秒データ)

※1ワード

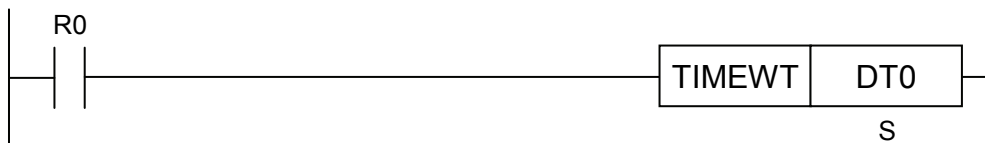
DT10	K 12	(年)
DT11	K 1	(月)
DT12	K 1	(日)
DT13	K 8	(時)
DT14	K 54	(分)
DT15	K 19	(秒)

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	時刻データの範囲を超えている場合にセットします

# TIMEWT (カレンダー設定)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S	時刻データの先頭デバイスアドレス

(注)この命令のみ曜日込みで合計7ワードとなります。

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデック ス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		" "
S	●	●	●	●	●	●	●	●	●												●

## ■ 動作説明

- [S]を先頭とするエリアに格納された時刻データ(年、月、日、時、分、秒)をCPUユニットのRTCデータとしてセットします。
- FP7 CPU ユニットのカレンダータイマに設定できる時刻データは、以下の範囲です。  
2000/1/1 00:00:00 ~ 2099/12/31 23:59:59

## ■ 処理内容

例) 2012年01月01日8時54分19秒を指定した場合  
[S]...DT0

※1ワード

DT0	K 12	(年)	更新 →	K 12	(年)
DT1	K 1	(月)		K 1	(月)
DT2	K 1	(日)		K 1	(日)
DT3	K 8	(時)		K 8	(時)
DT4	K 54	(分)		K 54	(分)
DT5	K 19	(秒)		K 19	(秒)
DT6	K 0	(曜日)		K 0	(曜日)

● 曜日データ

0	日
1	月
2	火
3	水
4	木
5	金
6	土

## ■ プログラム上のご注意

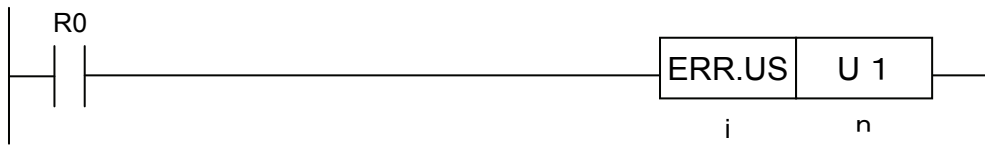
- 曜日データの日付との整合性はチェックしません。

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	時刻データの範囲を超えている場合にセットします
(ER)	曜日の範囲を超えている場合にセットします

# ERR (自己診断エラーコードセット)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●					

## ■ オペランド一覧

オペランド	説明
n	自己診断エラーコードを指定する

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデック ス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *1	K	U	H	SF	DF		" "
n																●	●				

## ■ 動作説明

- ユーザプログラムにより、任意のエラーコードをセットします。
- [n] で指定されている自己診断エラーコードを自己診断異常コードレジスタ(SD0)に格納し、自己診断異常発生フラグ(S0)をセットします。
- 同じエラーコードをセットする ERR 命令を記述することもできます。

## ■ 自己診断エラーコード[n]の設定

- n(自己診断エラーコード)は、U1000~U2999の範囲で設定できます。

n(自己診断エラーコード)の設定	エラー発生時の動作
U 1000 ~ 1999	運転停止
U 2000 ~ 2999	運転継続

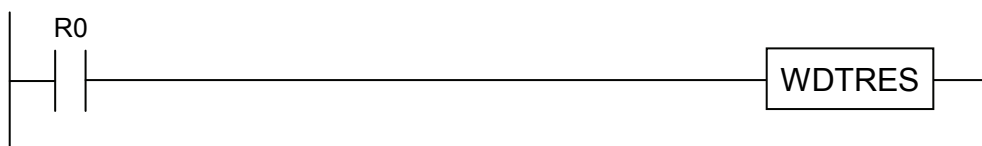
## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	nのエラーコードが範囲外となった場合にセットします



# WDTRES (ウォッチドッグタイマーリセット)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

- オペランドなし。

## ■ 指定可能なデバイス (●:指定可能)

- 指定可能なデバイスなし。

## ■ 動作説明

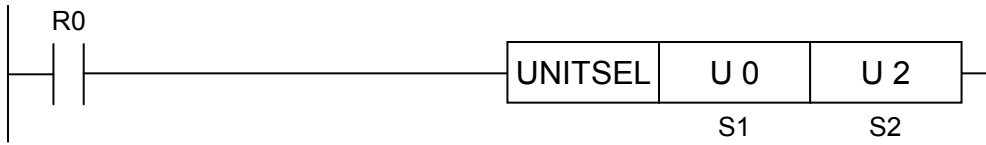
- ウォッチドッグタイマーをリセットします。

## ■ フラグ動作

- 変化ありません。

# UNITSEL (通信ユニットスロットポート指定)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S1	ユニットのスロット No.
S2	COM ポート No.またはコネクション No.

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX *1	K	U	H	SF	DF	""		
S1	●	●	●	●			●	●								●						●
S2	●	●	●	●			●	●								●						●

## ■ 動作説明

- 以下の通信命令の直前に記述し、各命令を実行する対象を指定します。  
GPSEND、GPRECV、SEND、RECV、RDET、PMSET、PMGET
- CPU ユニット内蔵 SCU の場合、スロット番号(U0)とCOM ポート番号を指定します。
- CPU ユニット内蔵 ET-LAN の場合、スロット番号(U100)とコネクション番号を指定します。
- シリアルコミュニケーションユニットの場合、スロット番号(U1~16)とCOM ポート番号を指定します。
- [S1]に指定したスロットの機種を取得し、[S2]に指定した通信ポート番号が設定可能範囲にあるかどうかを確認します。範囲外の場合はエラーとなります。
- SCU の場合、指定した通信ポート(COM ポート番号)に通信カセットが装備されているかどうかを確認します。指定 COM ポートに通信カセットがない場合は、エラーとなります。
- エラーが発生しなければ、[S1]、[S2]の値を、CPU ユニットのシステムデータレジスタ (SD40, SD41) にセットします。

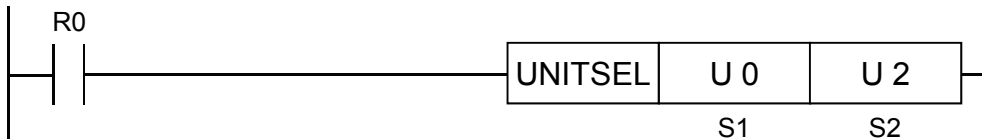
## ■ [S1]および[S2]の指定

- [S1]で通信ユニットのユニット番号を指定します。
- [S1]の設定値はシステムデータレジスタ SD40 に格納されます。
- [S2]で通信ポートを指定します。(SCU の場合:COM ポート番号、CPU ユニット内蔵 ET-LAN の場合:コネクション番号)
- [S2]の設定値はシステムデータレジスタ SD41 に格納されます。

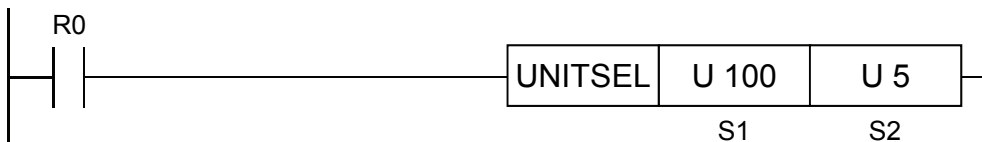
ユニット種類	[S1] ユニットNo.	[S2] COMポートNo. コネクションNo.
CPU ユニット内蔵 SCU	U0	U0~U2
CPU ユニット内蔵 ETLAN	U100	U1~U16
シリアルコミュニケーションユニット	U1~U16	U1~U4

## ■ プログラム例

例1) スロット0のCPU内蔵SCUのCOM2を指定する場合



例2) スロット100のCPU内蔵ET-LANのユーザーコネクション5を指定する場合

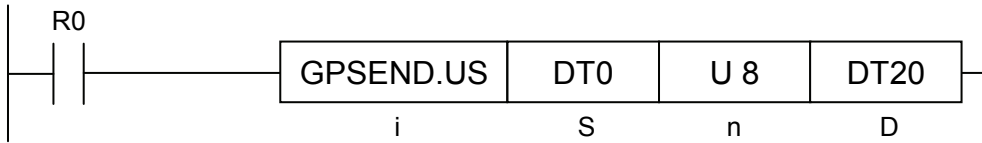


## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	[S2]で指定した COM ポートが存在しない場合(カセット未装着、通信カセットでない)
(ER)	[S2]で指定したコネクションが存在しないとき(コネクション番号範囲外)

# GPSEND (汎用通信送信命令)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
S	送信元データ領域の先頭番号
n	送信データバイト数(演算単位 i の指定にかかわらず、常に 16bit データ)
D	処理結果(1ワード)を格納する自局内のデバイス領域の先頭番号 (演算単位 i の指定にかかわらず、常に 1ワード(16ビット)データ)

(注) n に負の値を整数指定する場合は、演算単位に SS(符号付き)の指定が必要です。

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデック ス修飾 *1		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		""	
S1	●	●	●	●			●	●														●
n(*2)	●	●	●	●			●	●							●	●	●					●
D(*2)	●	●	●	●			●	●														●

\*1: UM,WI,WO 領域はインデックス修飾不可

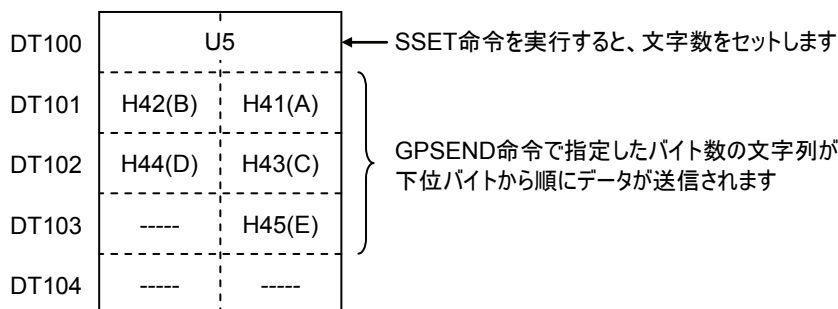
\*2: 演算単位 i の指定にかかわらず、常に 16 ビットデータ、16 ビットデバイス

## ■ 動作説明

- ユニットの通信ポートから、外部機器へのデータ送信します。
- UNITSEL 命令設定された通信ユニット・通信ポートをから、[S]で指定された送信データ領域の先頭アドレス(ワードアドレス)から、[n]バイト分のデータを送信します。
- 送信するデータは、[S]を先頭とする領域にユーザプログラムによりセットします。
- 処理結果は、[D]で指定されたエリアに格納します。

## ■ 送信データテーブル[S]の作成

- 送信するデータは、ユーザプログラムにより、任意のデータレジスタに格納します。



## ■ 送信データバイト数[n]の指定

ユニット種類	送信データ数設定	説明
SCU の場合 *1	1 ~ 4094	正の値のとき、コンフィグレーションメニューで設定した始端コード、終端コードを自動付加します。送信データには始端コード、終端コードを含めないでください。
	-1 ~ -4096	負の値のとき、終端コードを付加しません。
CPU ユニット内蔵 ET-LAN の場合	1 ~ 16372	「専用ヘッダを付ける」設定時(コネクション設定初期値) 送信データ、終端コードを区別しません。自動付加しません。
	1 ~ 16384 *2	「専用ヘッダを付けない」設定時 送信データ、終端コードを区別しません。自動付加しません。

\*1: SCU の設定が始端コード自動付加有効になっている場合は、最大送信データ数は1減少します。始端終端コードを含めて最大 4096 バイトとなります。

\*2: FP2 ETLAN へ送信する場合は、ユーザーコネクション設定で「専用ヘッダを付けない」を設定し、最大送信バイト数はユーザー指定で 8192 バイト以内としてください。

## ■ 処理結果 [D] の内容

状態	設定される値
送信完了時	送信済みバイト数
エラー発生時	FFFFH

## ■ プログラム上のご注意

- 通信を行なうためには、ツールソフトウェア FPWIN GR7 のコンフィグレーションメニューによる設定が必要です。通信モードの項で“汎用通信”を選択してください。
- GPSEND 命令の実行条件は、汎用送信中フラグが OFF となる送信完了まで、ON の条件を保持してください。
- 対象の COM ポートの汎用通信送信可フラグが ON していることを確認して、GPSEND 命令を実行してください。
- 送信中の通信ポートに対して、GPSEND 命令を実行した場合は、送信中フラグと実行結果の更新処理を行いません。
- GPSEND 命令は、割り込みプログラム内では使用できません。

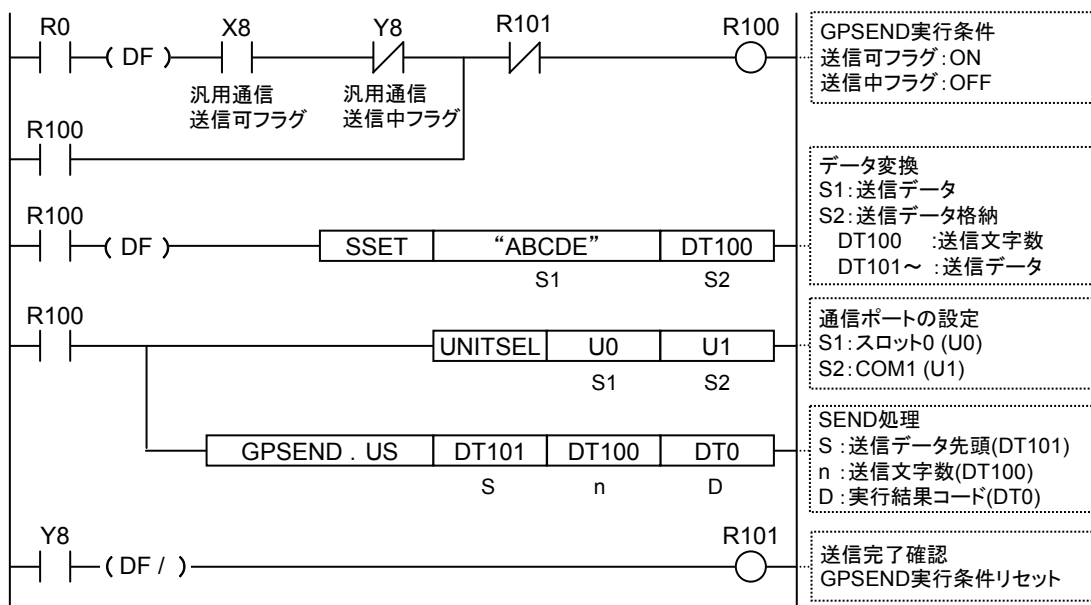


### ◆ ここがポイント

- SCU の場合とは、以下の組み合わせで使用した場合を示します。
  - CPU ユニット標準搭載の COM.0 ポート
  - CPU ユニットに通信カセットを装着 (COM.1~COM.2 ポート)
  - シリアルコミュニケーションユニットに通信カセットを装着 (COM.1~COM.4 ポート)
- 通信カセット (Ethernet タイプ) は、Ethernet-シリアル変換機能を持っているため、内部のインターフェイスは、SCU の場合と同様のプログラムで動作します。CPU ユニット内蔵 ET-LAN とは設定方法、プログラム方法が異なります。

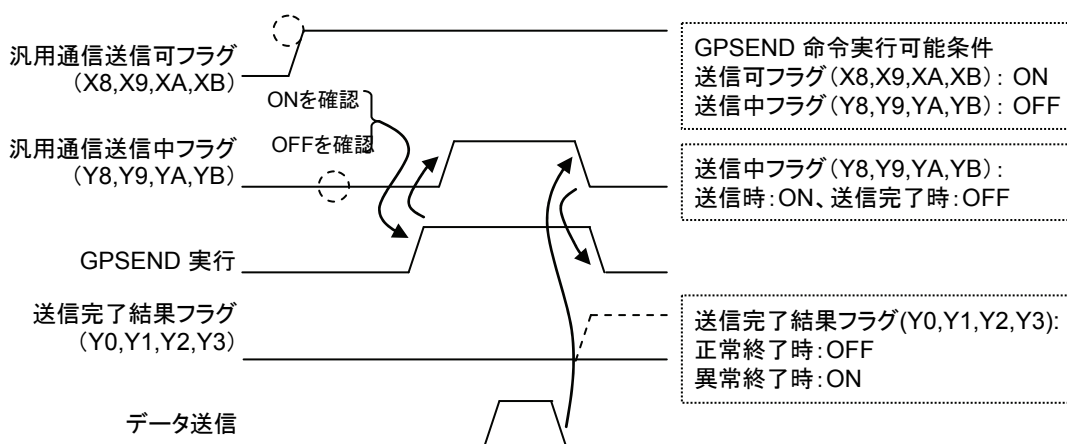
## ■ サンプルプログラム(SCUの場合)

- 汎用通信モードになっていること(X8)、同じポートに対し汎用送信が実行中でないこと(Y8)を確認し、送信プログラムを起動します。
- SSET 命令で、任意のメッセージを文字列に ASCII 変換して、送信文字列数をデータレジスタ(DT100)に、送信メッセージをデータレジスタ(DT101)からセットします。
- UNITSEL 命令で、スロット番号(U0)と COM.ポート No.(U1)を指定します。
- GPSEND 命令では、送信するメッセージを格納したテーブル先頭(DT101)とデータ文字数(DT100)を指定して実行します。



## ■ タイムチャート(SCUの場合)

- GPSEND 命令で指定したテーブルの[S]の下位バイトから順にデータを送信します。
- 送信中は、汎用通信送信中フラグ(Y8,Y9,YA,YB)が ON になります。送信が完了すると、OFF になります。(命令実行直後には OFF にならず、2 スキャン目の先頭で OFF します)
- 汎用通信送信完了結果フラグ(Y0,Y1,Y2,Y3)に送信結果(0: 正常終了,1: 異常終了)が格納されます。



## ■ I/Oの割り付け(CPUユニット内蔵SCUの場合)

COMポート番号			名称	名称
1	2	0		
X8	X9	XA	汎用送信可フラグ	汎用通信モードに設定している場合に ON になります。
Y8	Y9	YA	汎用通信送信中フラグ	汎用通信 GPSEND で、送信中に ON します。 送信完了時に OFF します。
Y0	Y1	Y2	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。(正常終了時:0 異常終了時:1)

(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。

(注 2): スキャン時間よりも送信時間が短い場合は、汎用通信送信中フラグ(Y8,Y9,YA,YB)は、データ送信を完了した次のスキャンの GPSEND 命令実行時に OFF になります。必ず、1 スキャンタイム以上 ON になります。

## ■ I/Oの割り付け(シリアルコミュニケーションユニットの場合)

COMポート番号				名称	説明
1	2	3	4		
X8	X9	XA	XB	汎用通信送信可フラグ	汎用通信モードに設定している場合に ON になります。
Y8	Y9	YA	YB	汎用通信送信中フラグ	汎用通信 GPSEND で、送信中に ON します。 送信完了時に OFF します。
Y0	Y1	Y2	Y3	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。(正常終了時:0 異常終了時:1)

(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。

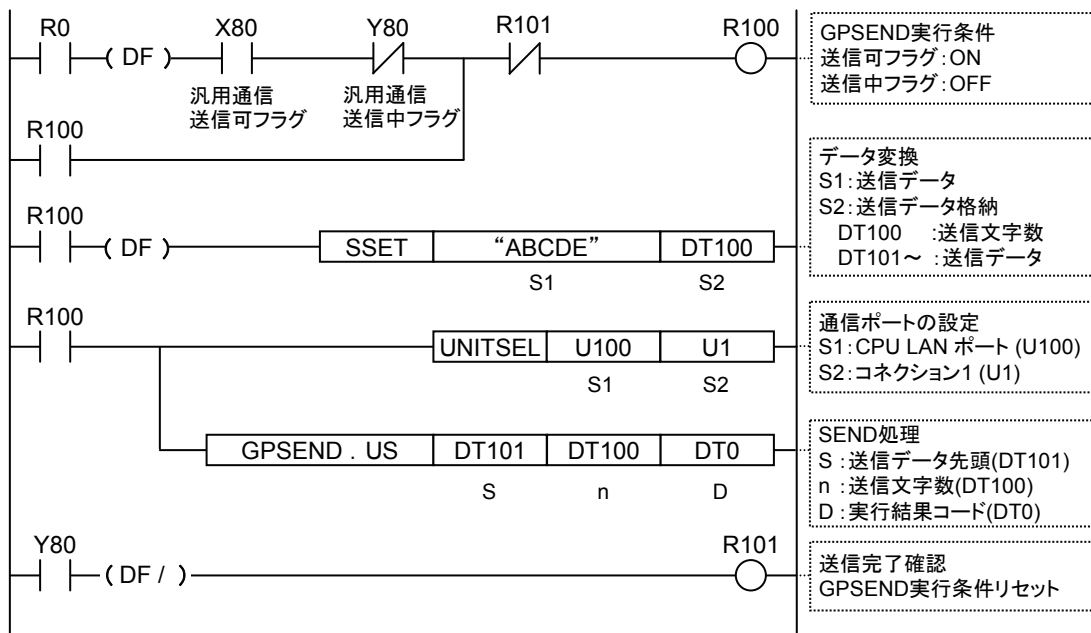
(注 2): スキャン時間よりも送信時間が短い場合は、汎用通信送信中フラグ(Y8,Y9,YA,YB)は、データ送信を完了した次のスキャンの GPSEND 命令実行時に OFF になります。必ず、1 スキャンタイム以上 ON になります。

## ■ プログラム上のご注意(SCUの場合)

- GPSEND 命令の直前に UNITSEL 命令を使用し、通信の対象となるポートを指定してください。
- 終端コードを付加しない場合は、GPSEND 命令送信データ数は、負の値を指定します。また、演算単位に "SS" を選択してください。
- 送信するデータには、コンフィグレーションメニューで指定した始端・終端コードが自動的に付加されます。送信データには、始端コード、終端コードを含めないでください。
- 送信できるデータは、最大 4,096 バイトです。始端コードを有効に設定している場合は、始端コード、終端コードを含めて、最大 4,096 バイトです。
- バイナリデータを送信することもできます。

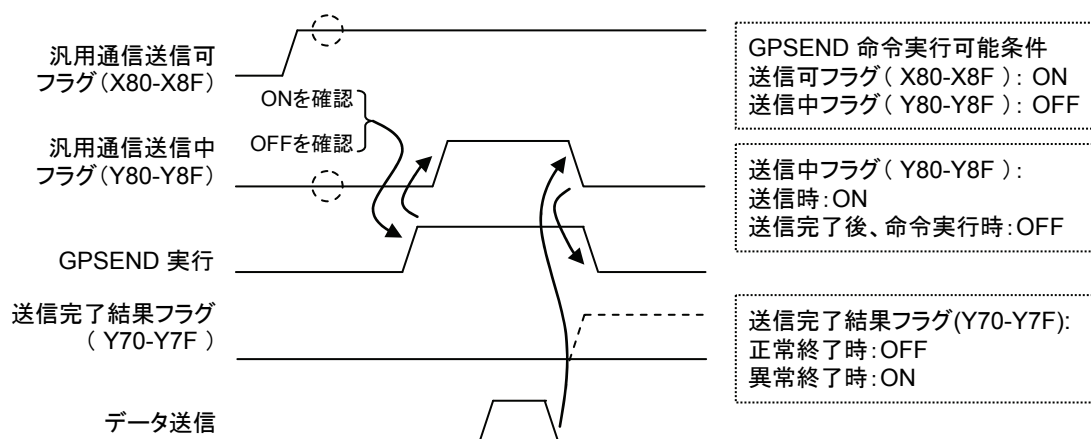
## ■ サンプルプログラム(CPUユニット内蔵 ET-LANの場合)

- 汎用通信モードでコネクション 1 が確立していること(X80)、同じポートに対し、汎用送信が実行中でないこと(Y80)を確認し、送信プログラムを起動します。
- SSET 命令で、任意のメッセージを文字列に ASCII 変換して、送信文字列数をデータレジスタ(DT100)に、送信メッセージをデータレジスタ(DT101)からセットします。
- UNITSEL 命令で、スロット番号(LAN ポート:U100)とコネクション No.(U1)を指定します。
- GPSEND 命令では、送信するメッセージを格納したテーブル先頭(DT101)とデータ文字数(DT100)を指定して実行します。



## ■ タイムチャート(CPUユニット内蔵 ET-LANの場合)

- GPSEND 命令で指定したテーブルの下位バイトから順にデータを送信します。
- 送信中は、コネクションに対応する汎用通信送信中フラグ(Y80~Y8F)が ON になります。送信が完了すると、OFF になります。
- 汎用通信送信完了結果フラグ(Y70~Y7F)に送信結果(0:正常終了,1:異常終了)が格納されます。





## ■ I/Oの割り付け(CPUユニット内蔵 ET-LANの場合)

I/O 番号	名称	説明
X80～X8F	汎用通信送信可フラグ	汎用通信でコネクションが接続状態にあるときに ON になります。
Y80～Y8F	汎用通信送信中フラグ	GPSEND 命令による送信中に ON になります。 送信完了後、次に GPSEND 命令が実行されたときに OFF になります。
Y70～Y7F	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。 (正常終了時:0、異常終了時:1)

(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。

## ■ プログラム上のご注意(CPUユニット内蔵 ET-LANの場合)

- GPSEND 命令の直前に UNITSEL 命令を使用し、通信の対象となるコネクションを指定してください。
- FP7 CPU ユニットから送信するデータには、始端コード、終端コードは付加されません。外部機器のプロトコルに合わせ、始端コード、終端コードを送信する必要がある場合は、送信用データの一部として格納してください
- FP7CPU ユニットの LAN ポートから GPSEND 命令によって、一度に送信できるデータは、最大 16,384 バイトです。



### ◆ ここがポイント

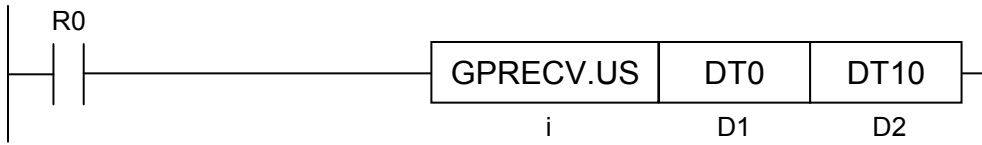
- 通信カセット(Ethernet タイプ)は、Ethernet-シリアル変換機能を持っているため、内部のインターフェイスは、SCU の場合と同様のプログラムで動作します。CPU ユニット内蔵 ET-LAN とは設定方法、プログラム方法が異なります。SCU の場合の項をご参照ください。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	UNITSEL 指定のコネクションが通信可能ではない場合
	UNITSEL 指定の通信ポートの通信モードが「汎用通信」ではない場合
	[S]で指定したデータデバイスが領域を超えるとき
	[n]で指定した送信データ数が 0 の場合。始端・終端コード含めて、最大値をこえる場合
	[n]で指定した送信データ数がデータ領域のエリアを超える場合
	ET-LAN への送信設定時に、[N]に 0 または負の値を設定した場合

# GPRECV (汎用通信受信命令)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
D1	受信データ格納先データ領域の先頭アドレス (演算単位 i の指定にかかわらず、常にワードデータの 16bit デバイス)
D2	受信データ格納先データ領域の最終アドレス (演算単位 i の指定にかかわらず、常にワードデータの 16bit デバイス)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデックス 修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""		
D1(*1)	●	●	●	●			●	●														●
D2(*1)	●	●	●	●			●	●														●

\*1: 演算単位 i の指定にかかわらず、常に 16 ビットデータ、16 ビットデバイス

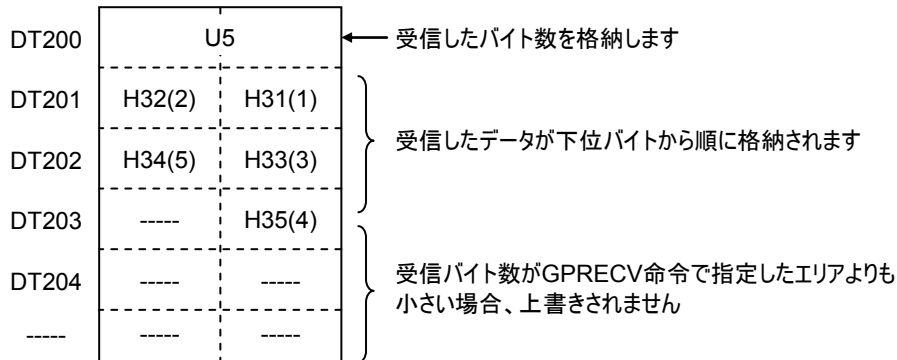
## ■ 動作説明

- ユニットの通信ポートに、外部機器から受信したデータを読み出します。
- UNITSEL 命令設定された通信ユニット・通信ポートから、受信データを読み出し、[D1]で指定されたエリアに受信バイト数、[D1+1]～[D2]の領域に受信データを格納します。
- SCU の場合、相手側の機器から受信したデータは、COM ポートごとに 8 つの受信バッファに格納されます。GPRECV 命令を実行すると受信バッファのデータを任意の演算用メモリにコピーすることができます。
- CPU ユニット内蔵 ET-LAN の場合、相手側の機器から受信したデータは、コネクションごとに 1 つの受信バッファに格納されます。GPRECV 命令を実行すると受信バッファのデータを任意の演算用メモリにコピーすることができます。

## ■ 受信データ数と終端コード

項目	SCU の場合	CPU ユニット内蔵 ET-LAN の場合
受信データ数	0~4096 (終端コード無し時)	0~16384
	0~4094, 4095 (終端コード有り時、終端設定による) ただし、受信バッファ、受信データ数から終端コードは 除かれます。	
終端コード識別	行う(SCU 通信設定(終端設定)に従う)	区別しない

## ■ 受信データの格納方法

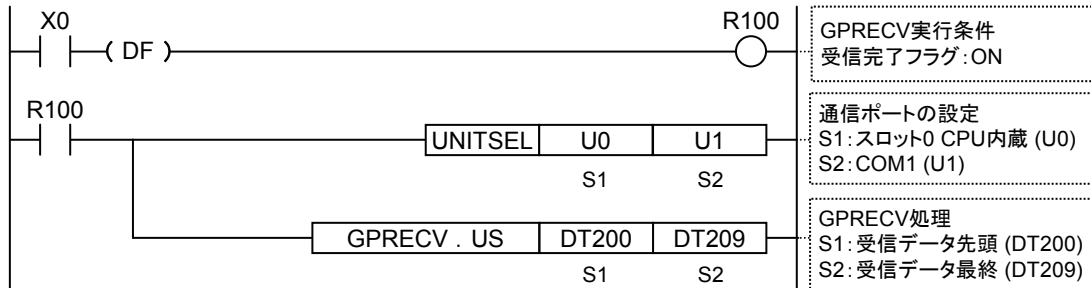


### ◆ ここがポイント

- SCU の場合とは、以下の組み合わせで使用した場合を示します。
  - CPU ユニット標準搭載の COM.0 ポート
  - CPU ユニットに通信カセットを装着 (COM.1~COM.2 ポート)
  - シリアルコミュニケーションユニットに通信カセットを装着 (COM.1~COM.4 ポート)
- 通信カセット (Ethernet タイプ) は、Ethernet-シリアル変換機能を持っているため、内部のインターフェイスは、SCU の場合と同様のプログラムで動作します。CPU ユニット内蔵 ET-LAN とは設定方法、プログラム方法が異なります。

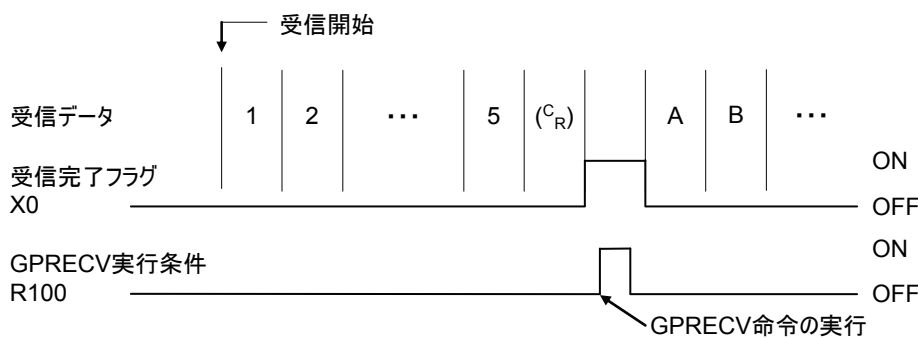
## ■ サンプルプログラム(SCUの場合)

- 受信完了フラグ(X0)が ON になると、GPREC V 命令による受信プログラムをを起動します。
- UNITSEL 命令で、スロット番号(U0)と COM.ポート No.(U1)を指定します。
- GPREC V 命令では、受信したメッセージを格納するデータテーブルの先頭(DT200)と最終アドレス(DT209)を指定して実行します。

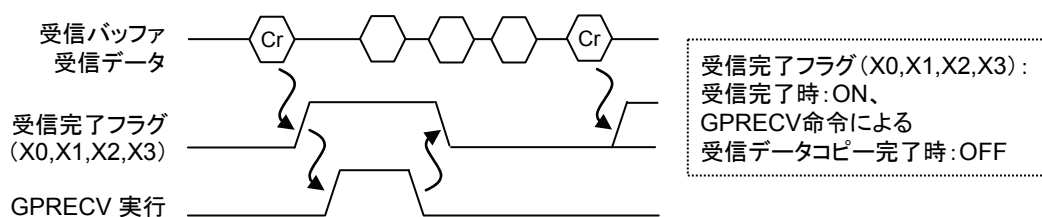


## ■ タイムチャート(SCUの場合)

- 外部機器から受信したデータは、受信用バッファに格納されます。
- 端末コードを受信すると、受信完了フラグ(X0,X1,X2,X3)が ON になります。以後、次のデータを受信するとバッファに格納します。連続して 8 個分のデータが受信可能です。



- GPREC V 命令を実行すると、指定したエリアにデータをコピーし、受信完了フラグ(X0,X1,X2,X3)が OFF になります。受信完了フラグ(X0,X1,X2,X3)は、次のスキャン以降の先頭で、I/O リフレッシュが実行されたときに OFF になります。



### ■ I/Oの割付(CPUユニットの場合)

COMポート番号			名称	名称
1	2	0		
X0	X1	X2	汎用通信受信完了フラグ	汎用通信モードで受信完了時に ON になります。
X4	X5	X6	汎用通信受信コピー完了フラグ	GPRECV 命令実行時に、受信データが指定した演算用メモリにコピーできたとき ON になります。データがない場合は OFF となります。

### ■ I/Oの割り付け(シリアルコミュニケーションユニットの場合)

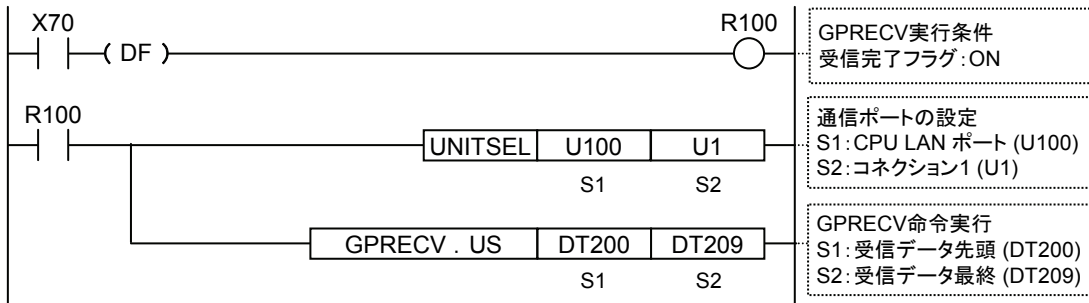
COMポート番号				名称	説明
1	2	3	4		
X0	X1	X2	X3	汎用通信受信完了フラグ	汎用通信モードで受信完了時に ON になります。
X4	X5	X6	X7	汎用通信受信コピー完了フラグ	GPRECV 命令実行時に、受信データが指定した演算用メモリにコピーできたとき ON になります。データがない場合は OFF となります。

### ■ プログラム上のご注意(SCUの場合)

- GPRECV 命令の直前に UNITSEL 命令を使用し、通信の対象となるポートを指定してください。
- 対象の COM ポートの汎用通信受信完了フラグが ON している場合に、GPRECV を実行してください。
- 多重受信している場合は、GPRECV 命令により受信データをコピーした後も、引き続き受信完了フラグが ON の状態ですので、受信完了信号の立ち上がりで受信データをコピーすることができません。
- GPRECV 命令でコピーする受信データには、始端コードおよび終端コードは含まれません。
- GPRECV 命令で、バイナリデータを受信することもできます。この場合、終端設定は、“時間”で指定します。
- 終端コードは受信データ、受信データ数には含まれません。(取り除かれます)
- SCU の場合は、内部に 4,096 バイトのバッファを 8 個を持っているため、連続して 8 回分のデータを受信することができます。
- SCU の受信バッファからデータを取り出す本命令 GPRECV が実行される前に、9 回目のデータが SCU に受信されたときは SCU のバッファ FULL エラーが発生し、9 回目の受信データは破棄されます。
- 受信バッファ FULL エラー状態に、GPRECV 命令が実行された場合、一番古い受信データが取り出され、受信バッファ FULL エラーは解除されます。
- 受信データがなかった場合は、汎用通信制御フラグ(受信完了コピーフラグ)は OFF となります。
- 受信データを取得し、CPU ユニットの演算用メモリにコピーが完了すると、汎用通信制御フラグ(受信完了コピーフラグ)が ON になります。
- 直接アドレスとインデックス修飾アドレスの場合、D1 と D2 は同じ種類のデバイスを指定してください。かつ、 $D2 \geq D1$  と指定してください。

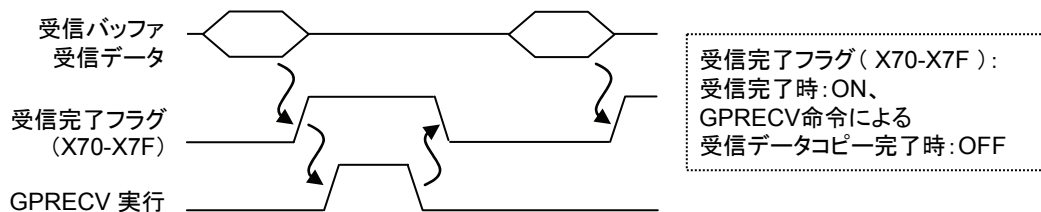
## ■ サンプルプログラム(CPUユニット内蔵 ET-LANの場合)

- コネクション 1 の受信完了フラグ (X70) が ON になると、GPREC V 命令による受信プログラムを起動します。
- UNITSEL 命令で、スロット番号 (LAN ポート: U100) とコネクション No. (U1) を指定します。
- GPREC V 命令では、受信したメッセージを格納するデータテーブルの先頭 (DT200) と最終アドレス (DT209) を指定して実行します。



## ■ タイムチャート(CPUユニット内蔵 ET-LANの場合)

- 外部機器から受信したデータは、コネクション毎に、受信用バッファに格納されます。
- データを受信すると、受信完了フラグ (X70~X7F) が ON になります。
- GPREC V 命令を実行すると、指定したエリアにデータをコピーし、受信完了フラグ (X70~X7F) が OFF になります。受信完了フラグ (X70~X7F) は、次のスキャン以降の先頭で、I/O リフレッシュが実行されたときに OFF になります。



## ■ I/O の割り付け(CPUユニット内蔵 ET-LANの場合)

I/Oの割り付け	名称	名称
X70~X7F	汎用通信受信完了フラグ	汎用通信モードで受信完了時に ON します。

## ■ プログラム上のご注意(CPUユニット内蔵 ET-LANの場合)

- GPREC V 命令の直前に UNITSEL 命令を使用し、通信の対象となるコネクションを指定してください。
- 対象のコネクションの汎用通信受信完了フラグが ON している場合に、GPREC V を実行してください。
- FP7CPU ユニットの LAN ポートから GPREC V 命令によって、一度に受信できるデータは、最大 16,384 バイトです。
- 外部機器の通信フォーマットによって、始端コード、終端コードが含まれている場合は、受信データの一部として、演算用メモリに格納されます。必要に応じて、データを抽出するプログラムを挿入してください。
- 直接アドレスとインデックス修飾アドレスの場合、D1 と D2 は同じ種類のデバイスを指定してください。かつ、D2 ≥ D1 と指定してください。



### ◆ ここがポイント

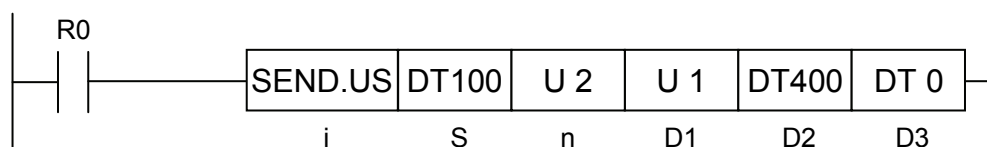
- 通信カセット (Ethernet タイプ) は、Ethernet-シリアル変換機能を持っているため、内部のインターフェイスは、SCU の場合と同様のプログラムで動作します。CPU ユニット内蔵 ET-LAN とは設定方法、プログラム方法が異なります。SCU の場合の項をご参照ください。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	転送先範囲がアクセス可能範囲外の場合にセットします
	UNITSEL 指定のスロットに SCU・ET-LAN ユニットが存在しない場合
	UNITSEL 指定の通信ポートの通信モードが「汎用通信」ではない場合
	UNITSEL 指定の COM ポートが存在しない場合
	UNITSEL 指定のコネクションが、「受信完了 OFF」で、「接続」状態ではない場合
	[D1],[D2]で指定したデータデバイスが領域を超えるとき
	[D1] ≥ [D2]を指定した場合
[D1]と[D2]に異なるデバイスが指定されていた場合	

# SEND (MEWTOCOLマスタ / MODBUSマスタ)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
S	送信元データ領域の先頭アドレス
n	送信データ数
D1	相手先局番
D2	相手局内の送信先データ領域の先頭アドレス
D3	実行結果コード(1ワード)を格納する自局内のデバイス領域の先頭アドレス

## ■ 指定可能なワードデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""		
S	●	●	●	●			●	●														●
n	●	●	●	●			●	●								●	●					●
D1	●	●	●	●			●	●								●	●					●
D2(*1)	*2	●	●	*2			●	*2														●
D3	●	●	●	●			●	●														●

\*1: 送信先が FP7 の場合は、グローバルデバイスのみを指定可能です(ローカルデバイスは指定できません)

\*2: MODBUS モードの場合、送信先に指定できません(ワードデバイス・ビットデバイス指定はできません)

## ■ 指定可能なビットデバイス (●:指定可能)

オペランド	ビットデバイス											ワードデバイスのビット指定		インデックス修飾
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b	LD.b	
S	●	●	●	●								●	●	●
n														
D1														
D2*1	*2	●	●	*2										●
D3														

\*1 送信先が FP7 の場合は、グローバルデバイスのみを指定可能です(ローカルデバイスは指定できません)

\*2 MODBUS モードの場合、ビットデバイス指定はできません



## ■ 動作説明

- ユニットの通信ポートからコマンドを送信し、外部機器とのデータ送受信を行います。プロトコルに合わせたメッセージは PLC が自動生成しますので、ユーザプログラムでは、局番やメモリアドレスを指定し、SEND/RECV 命令を実行するだけで、読み出し、書き込みを行うことができます。
- 通信モードは、ツールソフトウェア FPWIN GR7 のコンフィグレーションメニューで選択します。
- SEND 命令を実行すると、[S] を先頭とする自局内のデバイスからデータを読み出し、相手局の [D2] を先頭とするアドレスから格納します。
- [S] および [D2] で指定するデバイスの種類によって、転送方法(レジスタ転送/ビット転送)が変わります。
- 送信データ数 [n] は、レジスタ転送時はワード単位、ビット転送ワード数またはビット数に変わります。
- [D3] で指定した自局内の領域 1 ワードに実行結果コードを格納します。

## ■ 送信データ数 [n] の設定

送信種類	通信モード	送信データ数 n	補足
レジスタ送信	MEWTOCOL-COM	1 ~ 507 ワード	
	MEWTOCOL-DAT	1 ~ 1020 ワード	コネクション設定:MEWTOCOL 通信タイプの設定:FP2 ET-LAN と接続する
		1 ~ 2038 ワード	コネクション設定:MEWTOCOL 通信タイプの設定:FP2 ET-LAN と接続しない
	MODBUS	1 ~ 127 ワード	MODBUS のコマンド 15(WY, WR 書き込み), コマンド 16(DT 複数ワード書き込み)を使用します
ビット送信	MEWTOCOL-COM	1 ビット固定	MEWTOCOL-COM 時は、WCS コマンドを使用します
	MEWTOCOL-DAT	1 ビット固定	MEWTOCOL-DAT 時は、接点情報ライト 52H を使用します
	MODBUS	1 ~ 2040	複数点コイル強制コマンド 15 を使用します

(注 1):オペランド[S] および[D2]に指定するデバイスの種類によって、転送方法が変わります。16 ビットデバイスのときレジスタ転送、1 ビットデバイスのときビット転送となります。

(注 2):送信データ数は、レジスタ転送時にはワード単位、ビット転送時にはビット単位となります。

## ■ 相手先局番[D1]の指定

通信モード	SCU 使用時	ET-LAN 使用時
MEWTOCOL-COM	0 ~ 99	1 ~ 64
MEWTOCOL-DAT	SCU 非対応	
MODBUS	0 ~ 247	1 ~ 247

(注 1):相手先局番に "0" を指定した場合、グローバル転送となります。このとき、相手先からのレスポンスメッセージはありません。

(注 2):FP7 同士の接続時は"1"を指定します。接続先は IP アドレスで決まります。

## ■ 送信先アドレス[D2]の指定

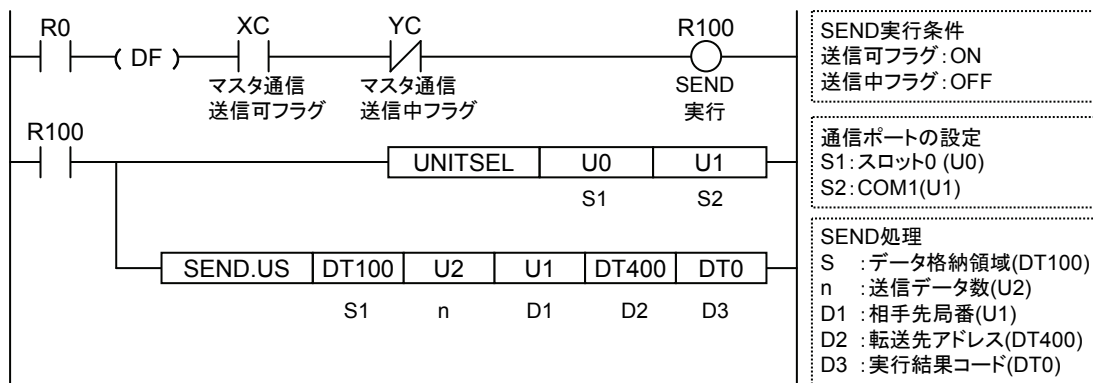
通信モード	アドレス範囲
MEWTOCOL-COM	0 ~ 9999
MEWTOCOL-DAT	0 ~ 65535 (H FFFF)
MODBUS	0 ~ 65535 (H FFFF)

## ■ 実行結果コード[D3]

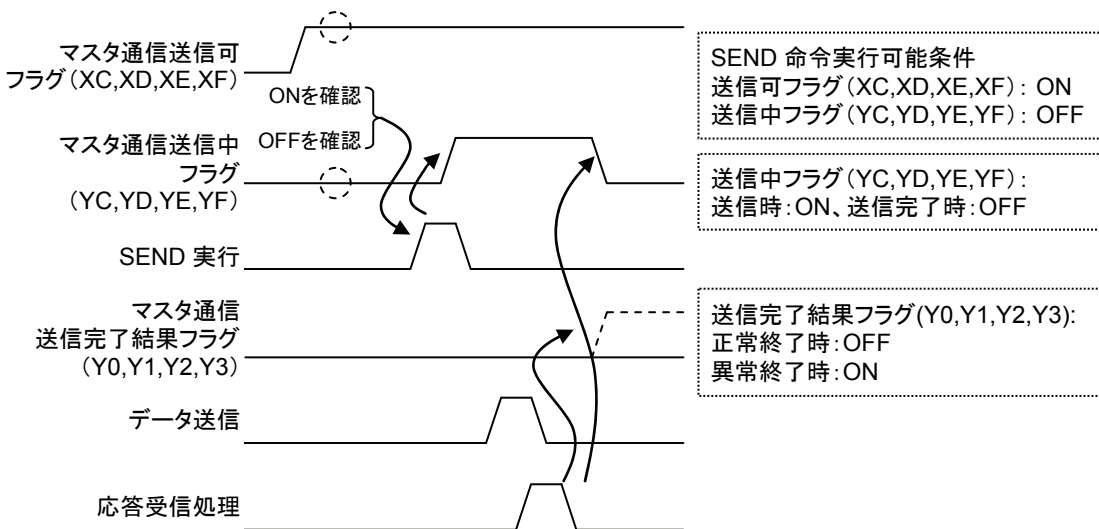
実行結果	実行結果コード
正常終了	0
通信ポートをマスター通信で使用中	1
通信ポートをスレーブ通信で使用中	2
マスター通信命令同時使用数を超過	3
送信タイムアウト	4
応答受信タイムアウト	5
受信データエラー	6

## ■ サンプルプログラム(SCUの場合)

- CPU ユニットの COM1 ポートからコマンドを送信し、PLC のデータレジスタ DT100~DT101 の内容を外部機器 (局番 1) のデータエリア DT400~DT401 へ書き込みます。
- マスタモードになっていること(XC)、同じポートに対し送信が実行中でないこと(YC)を確認し、SEND 命令を起動します。
- UNITSEL 命令で、スロット番号(U0)と COM.ポート No.(U1)を指定します。
- SEND 命令では、転送元の先頭アドレス(DT100)とデータ数(U2)、転送先の局番(U1)、先頭アドレス(DT400)を指定して実行します。



## ■ タイムチャート(SCUの場合)



### ◆ ここがポイント

- SCU の場合とは、以下の組み合わせで使用した場合を示します。
  - CPU ユニット標準搭載の COM.0 ポート
  - CPU ユニットに通信カセットを装着 (COM.1~COM.2 ポート)
  - シリアルコミュニケーションユニットに通信カセットを装着 (COM.1~COM.4 ポート)
- 通信カセット (Ethernet タイプ) は、Ethernet-シリアル変換機能を持っているため、内部のインターフェイスは、CPU ユニット内蔵 SCU の場合と同様のプログラムで動作します。CPU ユニット内蔵 ET-LAN とは設定方法、プログラム方法が異なります。また、通信カセット (Ethernet タイプ) は、MODBUS には対応していません。

### ■ I/Oの割り付け(CPUユニット内蔵SCUの場合)

COMポート番号			名称	説明
1	2	0		
XC	XD	XE	マスタ通信 送信可フラグ	通信モードで MEWTOCOL-COM、MEWTOCOL7、MODBUS-RTU が設定され、RUN モードになっているときに ON になります。
YC	YD	YE	マスタ通信 送信中フラグ	SEND,RECV 命令で送信中に ON します。 送信完了時に OFF します。
Y0	Y1	Y2	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。 (正常終了時:0 異常終了時:1)

(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。

### ■ I/Oの割り付け(シリアルコミュニケーションユニットの場合)

COMポート番号				名称	説明
1	2	3	4		
XC	XD	XE	XF	マスタ通信 送信可フラグ	通信モードで MEWTOCOL-COM、MEWTOCOL7、MODBUS-RTU が設定され、RUN モードになっているときに ON になります。
YC	YD	YE	YF	マスタ通信 送信中フラグ	SEND,RECV 命令で送信中に ON します。 送信完了時に OFF します。
Y0	Y1	Y2	Y3	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。 (正常終了時:0 異常終了時:1)

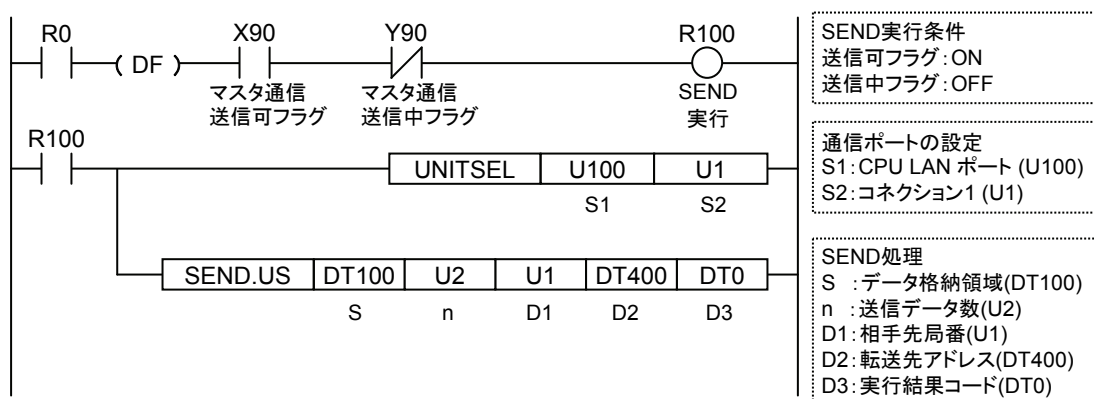
(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。

### ■ プログラム上のご注意(SCUの場合)

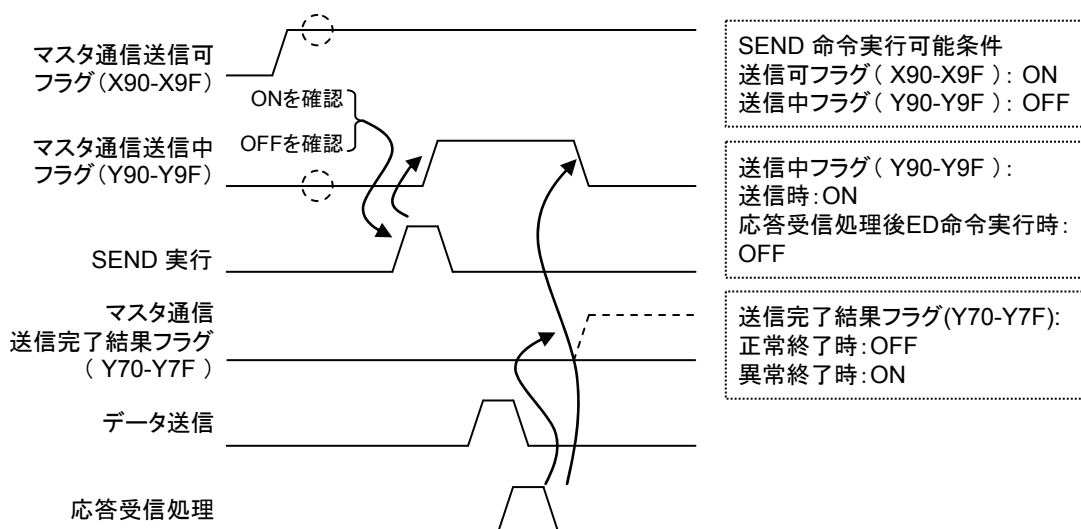
- SEND/RECV 命令の直前に UNITSEL 命令を使用し、通信の対象となるポートを指定してください。
- マスタ通信は、MEWTOCOL または MODBUS が選択されているときのみ有効です。対応するチャンネルの「マスタ通信送信可フラグ」(XC~XF)が ON になっていることを確認し、SEND/RECV 命令を実行してください。
- マスタ通信中の通信ポートに対して、他の SEND/RECV 命令を実行することはできません。「マスタ通信送信中フラグ」(YC~YF)が OFF になっていることを確認し、命令を実行してください。
- スレーブ通信中のポートに対して、SEND/RECV 命令を実行することはできません。
- 応答がない場合、「マスタ通信送信中フラグ」(YC~YF)は、CPU コンフィグレーションで設定されたタイムアウト設定時間まで ON の状態を継続します。
- SEND/RECV 命令は、異なる COM.ポートに対して、最大 16 命令まで同時実行することができます。

## ■ サンプルプログラム(CPUユニット内蔵 ET-LANの場合)

- CPU ユニットの LAN ポートからコマンドを送信し、PLC のデータレジスタ DT100～DT101 の内容を外部機器のデータエリア DT400～DT401 へ書き込みます。
- マスタモードで接続1が確立していること(X90)、同じポートに対して送信が実行中でないこと(Y90)を確認し、SEND 命令を起動します。
- UNITSEL 命令で、スロット番号(LAN ポート:U100)と接続 No.(U1)を指定します。
- SEND 命令では、転送元先頭アドレス(DT100)とデータ数(U2)、転送先の局番(U1)、先頭アドレス(DT400)を指定して実行します。



## ■ タイムチャート(CPUユニット内蔵 ET-LANの場合)



## ■ I/Oの割り付け(CPUユニット内蔵 ET-LANの場合)

I/O 番号	名称	説明
X90～X9F	マスタ通信送信可フラグ	マスタ通信で接続が接続状態にあるときに ON になります。
Y90～Y9F	マスタ通信送信中フラグ	SEND,RECV 命令で送信中に ON になります。 応答受信処理完了後、ED 命令が実行されたときに OFF になります。
Y70～Y7F	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。(正常終了時:0、異常終了時:1)

(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。

## ■ プログラム上のご注意(CPUユニット内蔵 ET-LANの場合)

- SEND/RECV 命令の直前に UNITSEL 命令を使用し、通信の対象となるコネクション No.を指定してください。
- マスタ通信は、MEWTOCOL または MODBUS が選択されているときのみ有効です。対応するコネクションの「マスタ通信送信可フラグ」(X90～X9F)が ON になっていることを確認し、SEND/RECV 命令を実行してください。
- マスタ通信中のコネクションに対して、他の SEND/RECV 命令を実行することはできません。「マスタ通信送信中フラグ」(Y90～Y9F)が OFF になっていることを確認し、命令を実行してください。
- スレーブ通信中のコネクションに対し、SEND/RECV 命令を実行することはできません。
- SEND/RECV 命令は、異なるコネクションに対して、最大 16 命令まで同時実行することができます。
- FP7 の LAN ポート同士で通信する場合、相手先局番には「U1」を指定します。送信先は IP アドレスで決まります。



### ◆ ここがポイント

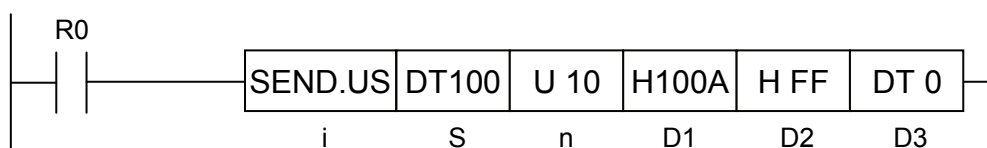
- 通信カセット(Ethernet タイプ)は、Ethernet-シリアル変換機能を持っているため、内部のインターフェイスは、CPU ユニット内蔵 SCU の場合と同様のプログラムで動作します。CPU ユニット内蔵 ET-LAN とは設定方法、プログラム方法が異なります。SCU の場合の項をご参照ください。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	送信元範囲がアクセス可能範囲外の場合にセットします
	UNITSEL 指定の COM ポート、コネクションが存在しないとき、コネクションが通信可能ではない
	[S]で指定したデータデバイスが不正、領域を超えるとき
	[n]で指定した送信データ数が不正
	[D1]で指定した局番が範囲外
	[D2]で指定したデータデバイスが不正、領域を超えている
	[D3]で指定した結果格納先デバイスが不正
	[D2]の整数指定は MODBUS 直接アドレス指定タイプのみ可。それ以外では不正。
[S]と[D2]のビットデバイス、16 ビットデバイス指定が異なっていた場合	

# SEND ( MODBUSマスタ:ファンクションコード指定)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

項目	設定内容	設定範囲					
i	演算単位を指定します。	US / SS					
S	送信元データ領域の先頭を指定します。	-					
n	送信データ数を指定します。	1 ~ 127ワード 1~2040ビット					
D1	使用する MODBUS コマンドと相手先局番を指定します。						
	<table border="1"> <tr> <td>上位バイト</td> <td>MODBUS ファンクションコードを示す 16 進数 2 桁</td> <td>H5、H6、HF、H10</td> </tr> <tr> <td>下位バイト</td> <td>局番を示す 16 進数 2 桁</td> <td>H0~HF7 (0 ~ 247)</td> </tr> </table>	上位バイト	MODBUS ファンクションコードを示す 16 進数 2 桁	H5、H6、HF、H10	下位バイト	局番を示す 16 進数 2 桁	H0~HF7 (0 ~ 247)
上位バイト	MODBUS ファンクションコードを示す 16 進数 2 桁	H5、H6、HF、H10					
下位バイト	局番を示す 16 進数 2 桁	H0~HF7 (0 ~ 247)					
D2	相手局内の送信先データ領域の MODBUS アドレスの先頭を指定します。	H0~HFFFF (0 ~ 65535)					
D3	実行結果コード(1ワード)を格納する自局内のデバイス領域を指定します。	-					

## ■ 指定可能なワードデバイス (●:指定可能)

オペランド	16ビットデバイス										32ビットデバイス			整数			実数		文字	インデックス修飾		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		" "	
S	●	●	●	●			●	●														●
n	●	●	●	●			●	●								●	●					●
D1	●	●	●	●			●	●								●	●					●
D2	●	●	●	●			●	●								●	*1					●
D3	●	●	●	●			●	●														●

\*1: 送信先が FP7 の場合は、グローバルデバイスのみを指定可能です(ローカルデバイスは指定できません)

## ■ 指定可能なビットデバイス (●:指定可能)

オペランド	ビットデバイス											ワードデバイスのビット指定		インデックス修飾
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b	LD.b	
S	●	●	●	●								●	●	●
n														
D1														
D2※1														
D3														

## ■ 動作説明

- ユニットの通信ポートから、MODBUS コマンドを送信し、外部機器とのデータ送受信を行います。
- プロトコルに合わせたメッセージは PLC が自動生成しますので、ユーザプログラムでは、局番やメモリアドレスを指定し、SEND/RCV 命令を実行するだけで、読み出し、書き込みを行うことができます。
- 通信モードは、ツールソフトウェア FPWIN GR7 のコンフィグレーションメニューで選択します。
- [D1]には、使用する MODBUS コマンドと相手先の MODBUS 局番を Hex 形式で指定します。
- SEND 命令を実行すると、[S] を先頭とする自局内のデバイスからデータを読み出し、相手局 の [D2] を先頭とするアドレスから格納します。
- [S]で指定するデバイスの種類、[n] で指定するデータ数によって、転送方法(レジスタ転送/ビット転送)と使用できる MODBUS コマンドの種類が変わります。
- 送信データ数 [n] は、レジスタ転送時はワード単位、ビット転送ワード数またはビット数に変わります。
- [D3] で指定した自局内の領域 1 ワードに実行結果コードを格納します。

## ■ [S] [n] [D1]の指定

- オペランド[S]で指定するデバイスの種類、[n]で指定する送信データ数によって、転送方法と使用できる MODBUS ファンクションコードが変わります。

[S] に指定するデバイスの種類	転送方法	送信データ数[n]	[D1] の上位バイトに指定できる値
16ビットデバイス WX,WY,WR,WL,DT,LD	レジスタ 送信	1	H6: レジスタ単点プリセット(06) HF: 複数点コイル強制(15) H10: 複数点レジスタプリセット(16)
		2~127	HF: 複数点コイル強制(15) H10: 複数点レジスタプリセット(16)
1ビットデバイス X,Y,R,L,DT,n,LD,n	ビット送信	1	H5: コイル単点強制(05) HF: 複数点コイル強制(15)
		2~2040	HF: 複数点コイル強制(15)

- 送信データ数[n]は、レジスタ転送時はワード単位、ビット転送時はビット単位となります。
- オペランド[D1] は、MODBUS ファンクションコード 16 進 2 桁と相手先局番を 16 進 2 桁の組合せで指定します。  
例: MODBUS ファンクションコード 16 (複数点レジスタプリセット)、局番 10 のとき、H100A と指定します。
- SCU の場合、相手先局番に "0" を指定した場合、グローバル転送となります。このとき、相手先からのレスポンスメッセージはありません。

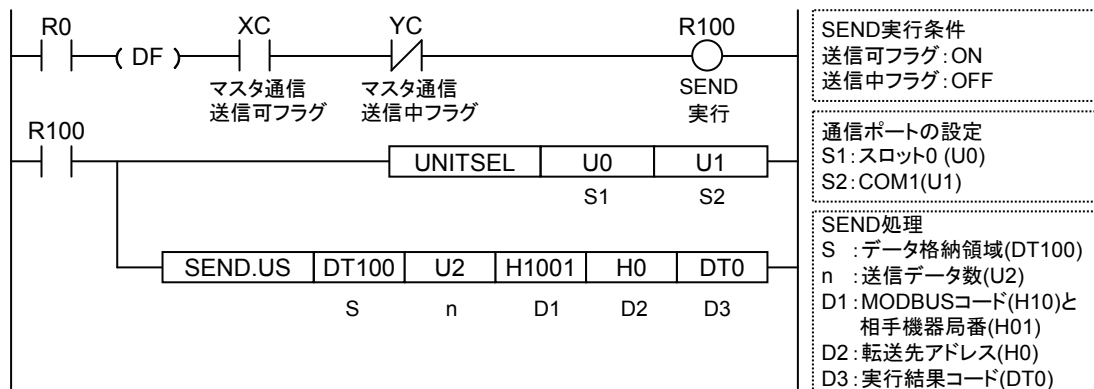
## ■ 実行結果コード[D3]

実行結果	実行結果コード
正常終了	0
通信ポートをマスター通信で使用中	1
通信ポートをスレーブ通信で使用中	2
マスター通信命令同時使用数を超過	3
送信タイムアウト	4
応答受信タイムアウト	5
受信データエラー	6

(注) 格納データ数は1ワードです

## ■ サンプルプログラム(SCUの場合)

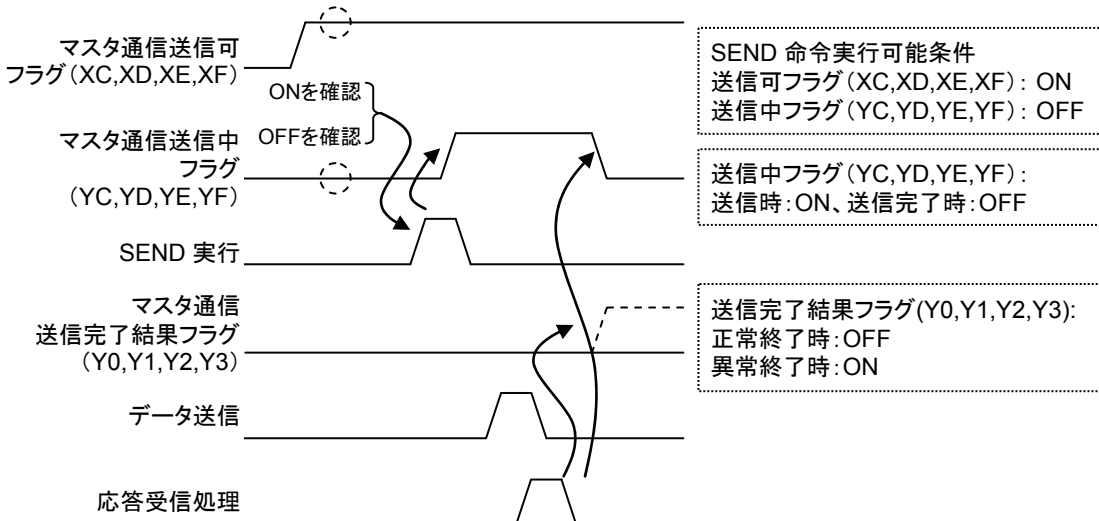
- CPU ユニットの COM1 ポートからコマンドを送信し、PLC のデータレジスタ DT100~DT101 の内容を、外部機器 (局番 1) のデータエリア 400001~400002 へ書き込みます。
- マスタモードになっていること(XC)、同じポートに対し送信が実行中でないこと(YC)を確認し、SEND 命令を起動します。
- UNITSEL 命令で、スロット番号(U0)と COM.ポート No.(U1)を指定します。
- SEND 命令では、PLC 側の先頭アドレス(DT100)とデータ数(U2)、使用する MODBUS ファンクションコード(16: H10)、相手機器の局番(H01)、先頭アドレス(H0)を指定して実行します。相手機器のアドレスについては、各機器の取扱説明書等でご確認ください。



(注1) : SEND 命令のオペランド[D1]は、MODBUS ファンクションコード 16 進 2 桁と相手機器局番 16 進 2 桁の組合せで指定します。MODBUS ファンクションコードが 16 のとき、[D1]の H10 を指定します。

(注2) : 相手機器が FP シリーズ PLC の場合、SEND 命令のオペランド[D2]は、デバイス番号で指定することができます。

## ■ タイムチャート(SCUの場合)



### ◆ ここがポイント

- SCU の場合とは、以下の組み合わせで使用した場合を示します。
  - CPU ユニット標準搭載の COM.0 ポート
  - CPU ユニットに通信カセットを装着 (COM.1~COM.2 ポート)
  - シリアルコミュニケーションユニットに通信カセットを装着 (COM.1~COM.4 ポート)
- 通信カセット(Ethernet タイプ)は、MODBUS には対応していません。



## ■ I/Oの割り付け(CPUユニット内蔵SCUの場合)

COMポート番号			名称	説明
1	2	0		
XC	XD	XE	マスタ通信 送信可フラグ	通信モードで MEWTOCOL-COM、MEWTOCOL7、MODBUS-RTU が設定され、RUN モードになっているときに ON になります。
YC	YD	YE	マスタ通信 送信中フラグ	SEND,RECV 命令で送信中に ON します。 送信完了時に OFF します。
Y0	Y1	Y2	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。 (正常終了時:0 異常終了時:1)

(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。

## ■ I/Oの割り付け(シリアルコミュニケーションユニットの場合)

COMポート番号				名称	説明
1	2	3	4		
XC	XD	XE	XF	マスタ通信 送信可フラグ	通信モードで MEWTOCOL-COM、MEWTOCOL7、MODBUS-RTU が設定され、RUN モードになっているときに ON になります。
YC	YD	YE	YF	マスタ通信 送信中フラグ	SEND,RECV 命令で送信中に ON します。 送信完了時に OFF します。
Y0	Y1	Y2	Y3	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。 (正常終了時:0 異常終了時:1)

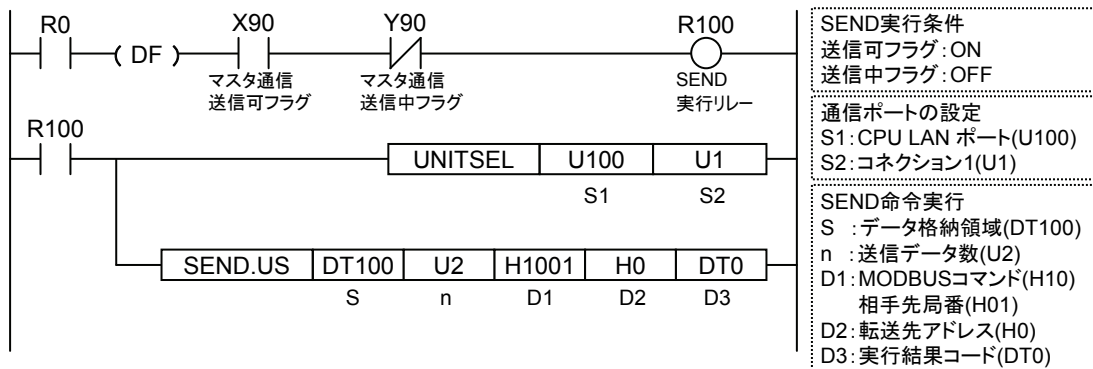
(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。

## ■ プログラム上のご注意(SCUの場合)

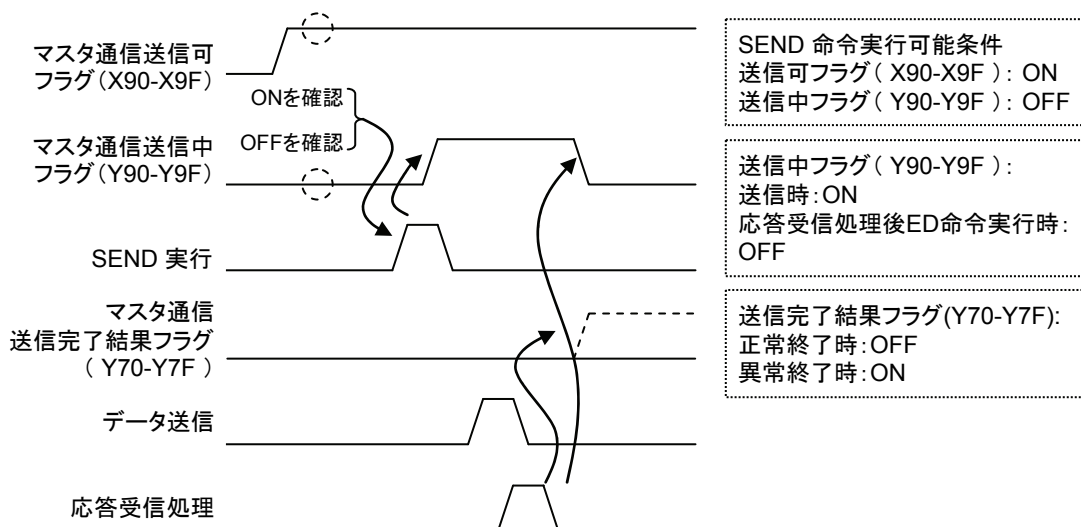
- SEND/RECV 命令の直前に UNITSEL 命令を使用し、通信の対象となるポートを指定してください。
- マスタ通信は、MEWTOCOL または MODBUS が選択されているときのみ有効です。対応するチャンネルの「マスタ通信送信可フラグ」(XC~XF)が ON になっていることを確認し、SEND/RECV 命令を実行してください。
- マスタ通信中の通信ポートに対して、他の SEND/RECV 命令を実行することはできません。「マスタ通信送信中フラグ」(YC~YF)が OFF になっていることを確認し、命令を実行してください。
- スレーブ通信中のポートに対して、SEND/RECV 命令を実行することはできません。
- 応答がない場合、「マスタ通信送信中フラグ」(YC~YF)は、CPU コンフィグレーションで設定されたタイムアウト設定時間まで ON の状態を継続します。
- SEND/RECV 命令は、異なる COM.ポートに対して、最大 16 命令まで同時実行することができます。

## ■ サンプルプログラム(CPUユニット内蔵 ET-LANの場合)

- CPU ユニットの LAN ポートから MODBUS コマンド (16) を送信し、PLC のデータレジスタ DT100~DT101 の内容を外部機器のデータエリア 400001~400002 (MODBUS アドレス 0000H~0001H) へ書き込みます。
- マスタモードで接続1が確立していること(X90)、同じポートに対して送信が実行中でないこと(Y90)を確認し、SEND 命令を起動します。
- UNITSEL 命令で、スロット番号(LAN ポート:U100)と接続 No.(U1)を指定します。
- SEND 命令では、PLC 側の先頭アドレス(DT100)とデータ数(U2)、MODBUS コマンド(16=H10)と相手機器の局番(H01)、先頭アドレス(H0)を指定して実行します。相手機器のアドレスについては、各機器の取扱説明書等でご確認ください。



## ■ タイムチャート



## ■ I/Oの割り付け

I/O 番号	名称	説明
X90~X9F	マスタ通信送信可フラグ	マスタ通信で接続が接続状態にあるときに ON になります。
Y90~Y9F	マスタ通信送信中フラグ	SEND, RECV 命令で送信中に ON になります。 応答受信処理完了後、ED 命令が実行されたときに OFF になります。
Y70~Y7F	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。 (正常終了時:0、異常終了時:1)

(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。

## ■ プログラム上のご注意(CPUユニット内蔵 ET-LANの場合)

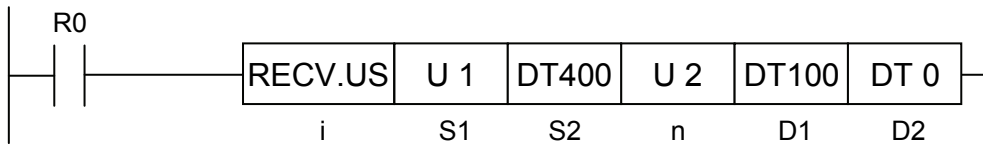
- SEND/RECV 命令の直前に UNITSEL 命令を使用し、通信の対象となるコネクション No.を指定してください。
- マスタ通信は、MEWTOCOL または MODBUS が選択されているときのみ有効です。対応するコネクションの「マスタ通信送信可フラグ」(X90～X9F)が ON になっていることを確認し、SEND/RECV 命令を実行してください。
- マスタ通信中のコネクションに対して、他の SEND/RECV 命令を実行することはできません。「マスタ通信送信中フラグ」(Y90～Y9F)が OFF になっていることを確認し、命令を実行してください。
- スレーブ通信中のコネクションに対し、SEND/RECV 命令を実行することはできません。
- SEND/RECV 命令は、異なるコネクションに対して、最大 16 命令まで同時実行することができます。
- MODBUS-TCP モードでは、SEND/RECV 命令のオペランドに相手先局番を指定します。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	送信元範囲がアクセス可能範囲外の場合にセットします
	UNITSEL 指定の COM ポート、コネクションが存在しないとき、コネクションが通信可能ではない
	[S]で指定したデータデバイスが不正、領域を超えるとき
	[n]で指定した送信データ数が不正
	[D1]で指定した MODBUS コマンド、局番が不正
	[D2]で指定したデータデバイスが不正、領域を超えている
	[D3]で指定した結果格納先デバイスが不正
	[D2]の整数指定は MODBUS アドレス直接指定タイプのみ。それ以外では不正。
	[D3]で指定した結果格納先デバイスが不正

# RECV (MEWTOCOL マスタ / MODBUS マスタ)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

オペランド	説明
S1	相手先局番
S2	相手局の送信元データ領域のデバイス先頭アドレス
n	受信データ数
D1	相手局内の送信先データ領域の先頭アドレス
D2	実行結果コード(1ワード)を格納する自局内のデバイス領域の先頭アドレス

## ■ 指定可能なワードデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾		
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "			
S1	●	●	●	●			●	●								●	●					●	
S2(*1)	●	●	●	●			●	●									●	●					●
n	●	●	●	●			●	●									●	●					●
D1	●	●	●	●			●	●															●
D2	●	●	●	●			●	●															●

\*1: 送信元が FP7 の場合は、グローバルデバイスのみを指定可能です(ローカルデバイスは指定できません)

## ■ 指定可能なビットデバイス (●:指定可能)

オペランド	ビットデバイス											ワードデバイスのビット指定		インデックス修飾	
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b	LD.b		
S1															
S2 *1	●	●	●	*2									*3	*3	●
n															
D1	●	●	●	●									●	●	●
D2															

\*1 送信元が FP7 の場合は、グローバルデバイスのみを指定可能です(ローカルデバイスは指定できません)

\*2 MODBUS モードの場合、ビットデバイス指定はできません

\*3 MEWTOCOL-COM モード、MODBUS モードの場合、ビットデバイス指定はできません

## ■ 動作説明

- ユニットの通信ポートからコマンドを送信し、外部機器とのデータ送受信を行います。
- プロトコルに合わせたメッセージは PLC が自動生成しますので、ユーザプログラムでは局番やメモリアドレスを指定し、SEND/RECV 命令を実行するだけで読み出し、書き込みを行うことができます。
- 通信モードは、ツールソフトウェア FPWIN GR7 のコンフィグレーションメニューで選択します。
- RECV 命令を実行すると、相手先局番[S1] の [S2] を先頭とするアドレスからデータを読み出し、[D1] を先頭とする自局内のエリアに格納します。
- [S2] および [D1] に指定するデバイスの種類によって、転送方法(レジスタ転送/ビット転送)が変わります。
- 送信データ数 [n] は、レジスタ転送時はワード単位、ビット転送ワード数またはビット数に変わります。
- [D2] で指定した自局内の領域 1 ワードに実行結果コードを格納します。

## ■ 相手先局番[S1]の指定

通信モード	SCU 使用時	ET-LAN 使用時
MEWTOCOL-COM	1 ~ 99	1 ~ 64
MEWTOCOL-DAT (注)	SCU 非対応	(注)
MODBUS	1 ~ 247	1 ~ 247

(注) : FP7 同士の接続時は"1"を指定します。接続先は IP アドレスで決まります。

## ■ 送信元データ領域の先頭アドレス[S2]の指定

通信モード	アドレス範囲
MEWTOCOL-COM	0 ~ 9999
MEWTOCOL-DAT	0 ~ 65535 (H FFFF)
MODBUS	0 ~ 65535 (H FFFF)

## ■ 受信データ数[n]の指定

送信種類	通信モード	通信ポート種類	設定範囲
レジスタ送信 *1	MEWTOCOL-COM	1 ~ 509 ワード	RCC コマンド、RD コマンドを使用します
	MEWTOCOL-DAT	1 ~ 1020 ワード	コネクション設定:MEWTOCOL 通信タイプの設定: FP2 ET-LAN と接続する
		1 ~ 2038 ワード	コネクション設定:MEWTOCOL 通信タイプの設定: FP2 ET-LAN と接続しない
	MODBUS	1 ~ 127 ワード	WY, WR 読み出しにはコマンド1、WX にはコマンド2を使用します DT 読み出しには、コマンド3を使用します、 WL, LD の読み出しには、コマンド4を使用します
ビット送信 *2	MEWTOCOL-COM	1ビット固定	MEWTOCOL-COM 時は、RCS コマンドを使用します
	MEWTOCOL-DAT	1ビット固定	MEWTOCOL-DAT 時は、接点情報リード 53H を使用します
	MODBUS	1 ~ 2040 ビット	Y, R 読み出しにはコマンド1、X にはコマンド2を使用します

\*1 送信元[S]、送信先[D2]に 16 ビットデバイスを指定したとき

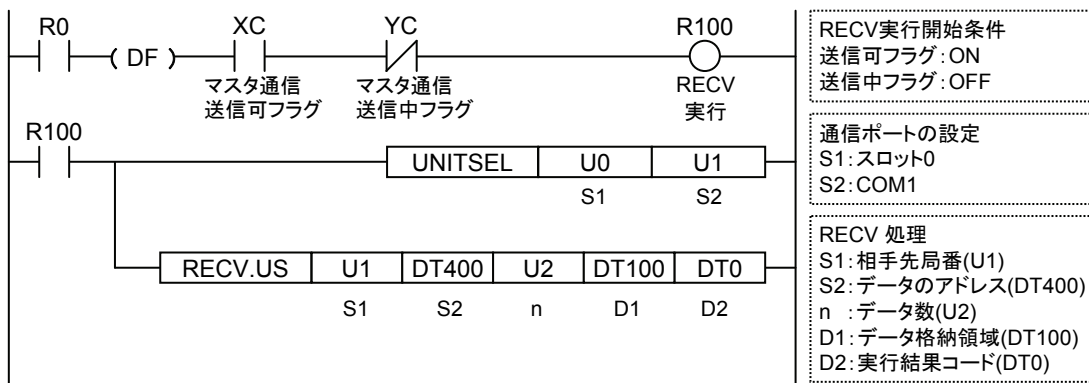
\*2 送信元[S]、送信先[D2]にビットデバイスを指定したとき

## ■ 実行結果コード[D2]

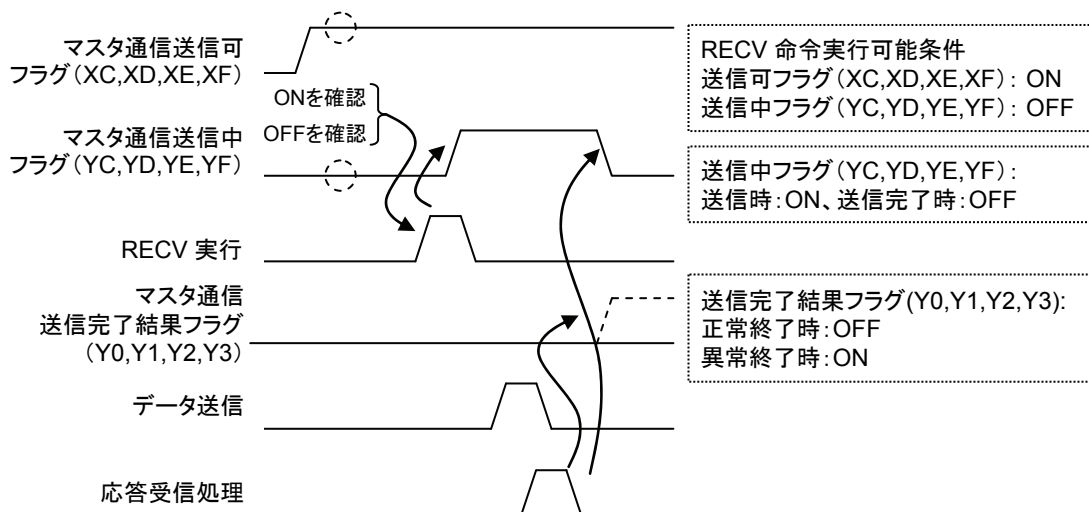
実行結果	実行結果コード
正常終了	0
通信ポートをマスター通信で使用中	1
通信ポートをスレーブ通信で使用中	2
マスター通信命令同時使用数を超過	3
送信タイムアウト	4
応答受信タイムアウト	5
受信データエラー	6

### ■ サンプルプログラム(SCUの場合)

- CPU ユニットの COM1 ポートからコマンドを送信し、外部機器(局番 1)のデータエリア DT400~DT401 からデータを読み出し、PLC のデータレジスタ DT100~DT101 へ書き込みます。
- マスタモードになっていること(XC)、同じポートに対し送信が実行中でないこと(YC)を確認し、SEND 命令を起動します。
- UNITSEL 命令で、スロット番号(U0)と COM.ポート No.(U1)を指定します。
- RECV 命令では、相手局の局番(U1)、先頭アドレス(DT400)、データ数(U2)、データを格納する PLC 側の先頭アドレス(DT100) を指定して実行します。



### ■ タイムチャート(SCUの場合)



#### ◆ ここがポイント

- SCU の場合とは、以下の組み合わせで使用した場合を示します。
  - CPU ユニット標準搭載の COM.0 ポート
  - CPU ユニットに通信カセットを装着 (COM.1~COM.2 ポート)
  - シリアルコミュニケーションユニットに通信カセットを装着 (COM.1~COM.4 ポート)
- 通信カセット (Ethernet タイプ) は、Ethernet-シリアル変換機能を持っているため、内部のインターフェイスは、CPU ユニット内蔵 SCU の場合と同様のプログラムで動作します。CPU ユニット内蔵 ET-LAN とは設定方法、プログラム方法が異なります。また、通信カセット (Ethernet タイプ) は、MODBUS には対応していません。

## ■ I/Oの割り付け(CPUユニット内蔵SCUの場合)

COMポート番号			名称	説明
1	2	0		
XC	XD	XE	マスタ通信 送信可フラグ	通信モードで MEWTOCOL-COM、MEWTOCOL7、MODBUS-RTU が設定され、RUN モードになっているときに ON になります。
YC	YD	YE	マスタ通信 送信中フラグ	SEND,RECV 命令で送信中に ON します。 送信完了時に OFF します。
Y0	Y1	Y2	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。 (正常終了時:0 異常終了時:1)

(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。

## ■ I/Oの割り付け(シリアルコミュニケーションユニットの場合)

COMポート番号				名称	説明
1	2	3	4		
XC	XD	XE	XF	マスタ通信 送信可フラグ	通信モードで MEWTOCOL-COM、MEWTOCOL7、MODBUS-RTU が設定され、RUN モードになっているときに ON になります。
YC	YD	YE	YF	マスタ通信 送信中フラグ	SEND,RECV 命令で送信中に ON します。 送信完了時に OFF します。
Y0	Y1	Y2	Y3	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。 (正常終了時:0 異常終了時:1)

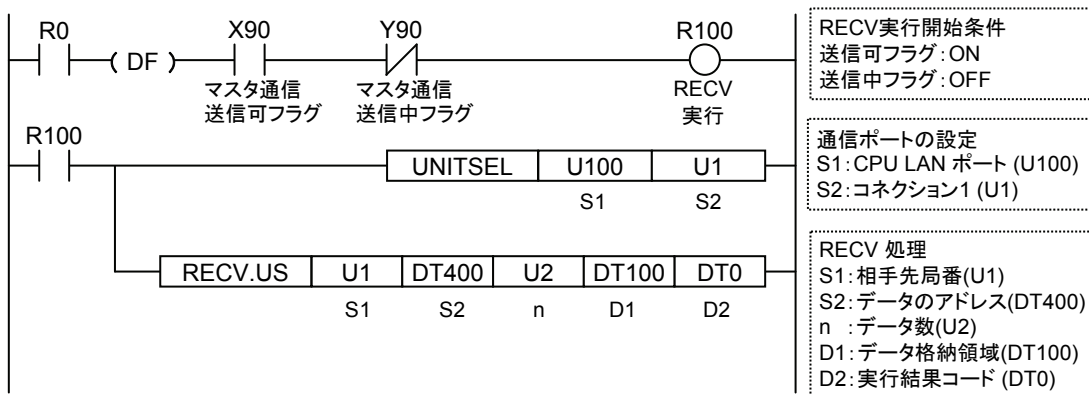
(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。

## ■ プログラム上のご注意(SCUの場合)

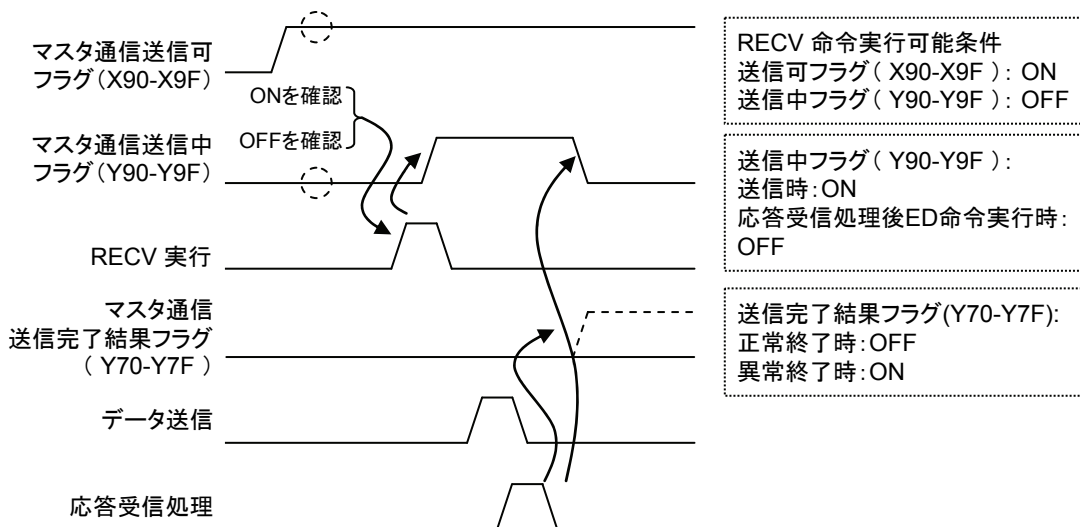
- SEND/RECV 命令の直前に UNITSEL 命令を使用し、通信の対象となるポートを指定してください。
- マスタ通信は、MEWTOCOL または MODBUS が選択されているときのみ有効です。対応するチャンネルの「マスタ通信送信可フラグ」(XC~XF)が ON になっていることを確認し、SEND/RECV 命令を実行してください。
- マスタ通信中の通信ポートに対して、他の SEND/RECV 命令を実行することはできません。「マスタ通信送信中フラグ」(YC~YF)が OFF になっていることを確認し、命令を実行してください。
- スレーブ通信中のポートに対して、SEND/RECV 命令を実行することはできません。
- 応答がない場合、「マスタ通信送信中フラグ」(YC~YF)は、CPU コンフィグレーションで設定されたタイムアウト設定時間まで ON の状態を継続します。
- SEND/RECV 命令は、異なる COM.ポートに対して、最大 16 命令まで同時実行することができます。

## ■ サンプルプログラム(CPUユニット内蔵 ET-LANの場合)

- CPU ユニットの LAN ポートからコマンドを送信し、外部機器のデータエリア DT400～DT401 からデータを読み出し、PLC のデータレジスタ DT100～DT101 の内容をへ書き込みます。
- マスタモードで接続1が確立していること(X90)、同じポートに対して送信が実行中でないこと(Y90)を確認し、SEND 命令を起動します。
- UNITSEL 命令で、スロット番号(LAN ポート:U100)と接続 No.(U1)を指定します。
- RECV 命令では、相手局の局番(U1)、先頭アドレス(DT400)、データ数(U2)、データを格納するPLC側の先頭アドレス(DT100)を指定して実行します。



## ■ タイムチャート(CPUユニット内蔵 ET-LANの場合)



## ■ I/Oの割り付け(CPUユニット内蔵 ET-LANの場合)

I/O 番号	名称	説明
X90～X9F	マスタ通信送信可フラグ	マスタ通信で接続1が接続状態にあるときに ON になります。
Y90～Y9F	マスタ通信送信中フラグ	SEND,RECV 命令で送信中に ON になります。 応答受信処理完了後、ED 命令が実行されたときに OFF になります。
Y70～Y7F	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。(正常終了時:0、異常終了時:1)

(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。



## ■ プログラム上のご注意(CPUユニット内蔵 ET-LANの場合)

- SEND/RECV 命令の直前に UNITSEL 命令を使用し、通信の対象となるコネクション No.を指定してください。
- マスタ通信は、MEWTOCOL または MODBUS が選択されているときのみ有効です。対応するコネクションの「マスタ通信送信可フラグ」(X90～X9F)が ON になっていることを確認し、SEND/RECV 命令を実行してください。
- マスタ通信中のコネクションに対して、他の SEND/RECV 命令を実行することはできません。「マスタ通信送信中フラグ」(Y90～Y9F)が OFF になっていることを確認し、命令を実行してください。
- スレーブ通信中のコネクションに対し、SEND/RECV 命令を実行することはできません。
- SEND/RECV 命令は、異なるコネクションに対して、最大 16 命令まで同時実行することができます。
- FP7 の LAN ポート同士で通信する場合、相手先局番には「U1」を指定します。送信先は IP アドレスで決まります。



### ◆ ここがポイント

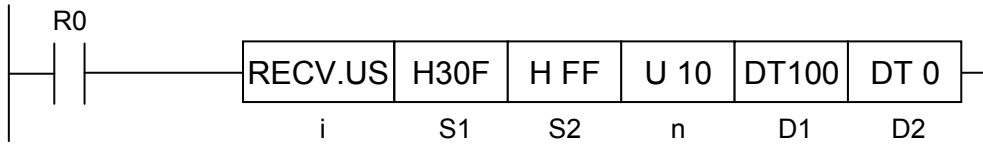
- 通信カセット(Ethernet タイプ)は、Ethernet-シリアル変換機能を持っているため、内部のインターフェイスは、CPU ユニット内蔵 SCU の場合と同様のプログラムで動作します。CPU ユニット内蔵 ET-LAN とは設定方法、プログラム方法が異なります。SCU の場合の項をご参照ください。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	転送先範囲がアクセス可能範囲外の場合にセットします
	UNITSEL 指定の COM ポート、コネクションが存在しないとき、コネクションが通信可能ではない
	[S1]で指定した相手局番が範囲外の時
	[S2]で指定した相手局送信元データデバイスが不正の時
	[n]で指定した送信データ数が不正
	[D1]で指定した自局内の送信先データ領域のデータデバイスが不正、領域を超えるとき
	[D2]で指定した結果格納先デバイスが不正
	[S2]と[D1]のビットデバイス、16ビットデバイス指定が異なっていた場合
[S2]の整数指定は MODBUS アドレス直接指定タイプのみ。それ以外では不正。	

# RECV (MODBUSマスタ:ファンクションコード指定)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

演算単位	bit	US	SS	UL	SL	SF	DF
i		●	●				

## ■ オペランド一覧

項目	設定内容	設定範囲					
i	演算単位を指定します。	US / SS					
S1	使用する MODBUS ファンクションコードと相手先局番を指定します。						
	<table border="1"> <tr> <td>上位バイト</td> <td>MODBUS ファンクションコードを示す 16 進数 2 桁</td> <td>H1 ~ H4 (1 ~ 4)</td> </tr> <tr> <td>下位バイト</td> <td>局番を示す 16 進数 2 桁</td> <td>H1 ~ HF7 (1 ~ 247)</td> </tr> </table>	上位バイト	MODBUS ファンクションコードを示す 16 進数 2 桁	H1 ~ H4 (1 ~ 4)	下位バイト	局番を示す 16 進数 2 桁	H1 ~ HF7 (1 ~ 247)
上位バイト	MODBUS ファンクションコードを示す 16 進数 2 桁	H1 ~ H4 (1 ~ 4)					
下位バイト	局番を示す 16 進数 2 桁	H1 ~ HF7 (1 ~ 247)					
S2	相手局の送信元 MODBUS アドレスを指定します。	H0 ~ HFFFF (0 ~ 65535)					
n	送信データ数を指定します。	1 ~ 127 ワード 1~2040 ビット					
D1	自局内の送信先データ領域のデバイス先頭アドレスを指定します。	—					
D2	実行結果コード(1 ワード)を格納する自局内のデバイス領域を指定します。						

## ■ 指定可能なワードデバイス (●:指定可能)

オペランド	16ビットデバイス											32ビットデバイス			整数			実数		文字	インデックス修飾
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""	
S1	●	●	●	●			●	●								●	●				●
S2	●	●	●	●			●	●								●	*1				●
n	●	●	●	●			●	●								●	●				●
D1	●	●	●	●			●	●													●
D2	●	●	●	●			●	●													●

\*1: MODBUS モードの「直接アドレス指定」(本命令)の場合にのみ、送信元アドレスに整数を設定できます

## ■ 指定可能なビットデバイス (●:指定可能)

オペランド	ビットデバイス											ワードデバイスのビット指定		インデックス修飾
	X	Y	R	L	T	C	P	E	S	IN	OT	DT.b	LD.b	
S1														
S2	●	●	●	●								●	●	●
n														
D1	●	●	●	●								●	●	●
D2														

## ■ 動作説明

- ユニットの通信ポートから MODBUS コマンドを送信し、外部機器とのデータ送受信を行います。
- プロトコルに合わせたメッセージは PLC が自動生成しますので、ユーザプログラムでは局番やメモリアドレスを指定し、SEND/RCV 命令を実行するだけで読み出し、書き込みを行うことができます。
- [S1] には、MODBUS コマンドと相手先の MODBUS アドレスを Hex 形式で指定します。
- RCV 命令を実行すると、相手局 の [S2] を先頭とするアドレスからデータを読み出し、[D1] を先頭とする自局内のエリアに格納します。
- [D1] に指定するデバイスの種類によって、転送方法(レジスタ転送/ビット転送)と使用できる MODBUS ファンクションコードが変わります。
- 送信データ数 [n] は、レジスタ転送時はワード単位、ビット転送ワード数またはビット数に変わります。
- [D2] で指定した自局内の領域 1 ワードに実行結果コードを格納します。

## ■ [S1] および [n] の指定

- オペランド[S1] は、MODBUS ファンクションコード 16 進 2 桁と相手先局番を 16 進 2 桁の組合せで指定します。  
例: MODBUS ファンクションコード 03 (保持レジスタの読み出し)、局番 15 のとき、H030F と指定します。
- オペランド[D1]で指定するデバイスの種類によって、転送方法と使用できる MODBUS ファンクションコードが変わります。

[D1]に指定するデバイス	転送方法	[S1]の上位バイトに指定できる値
16ビットデバイス WX,WY,WR,WL,DT,LD	レジスタ送信	H1:コイル状態読み出し(01) H2:入力状態読み出し(02) H3:保持レジスタの読み出し(03) H4:入力レジスタの読み出し(04)
1ビットデバイス X,Y,R,L,DT,n,LD,n	ビット送信	H1:コイル状態読み出し(01) H2:入力状態読み出し(02)

- 送信データ数は、レジスタ転送時はワード単位、ビット転送時はビット単位となります。

## ■ 実行結果コード[D2]

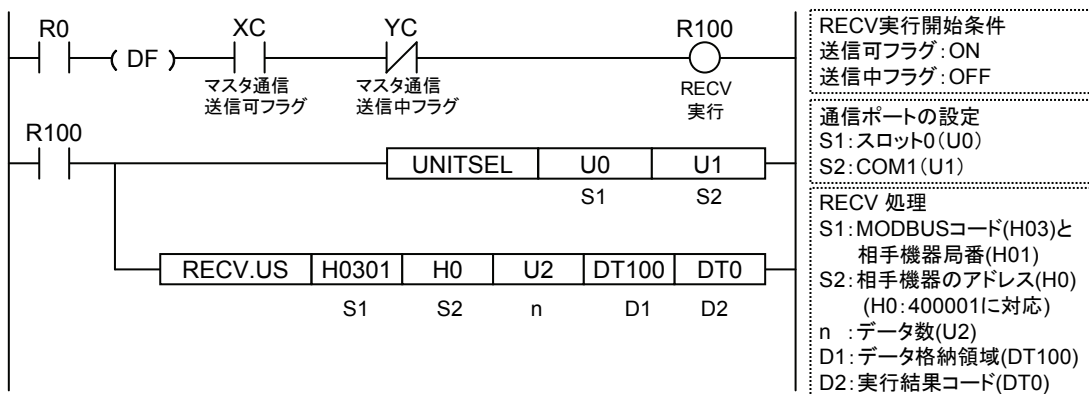
実行結果	実行結果コード
正常終了	0
通信ポートをマスター通信で使用中	1
通信ポートをスレーブ通信で使用中	2
マスター通信命令同時使用数を超過	3
送信タイムアウト	4
応答受信タイムアウト	5
受信データエラー	6

(注1):RCV 命令のオペランド[S1]は、MODBUS ファンクションコード 16 進 2 桁と相手機器局番は 16 進 2 桁の組合せで指定します。

(注2):相手機器が FP シリーズ PLC の場合、RCV 命令のオペランド[S2]は、デバイス番号で指定することができます。

## ■ サンプルプログラム(SCUの場合)

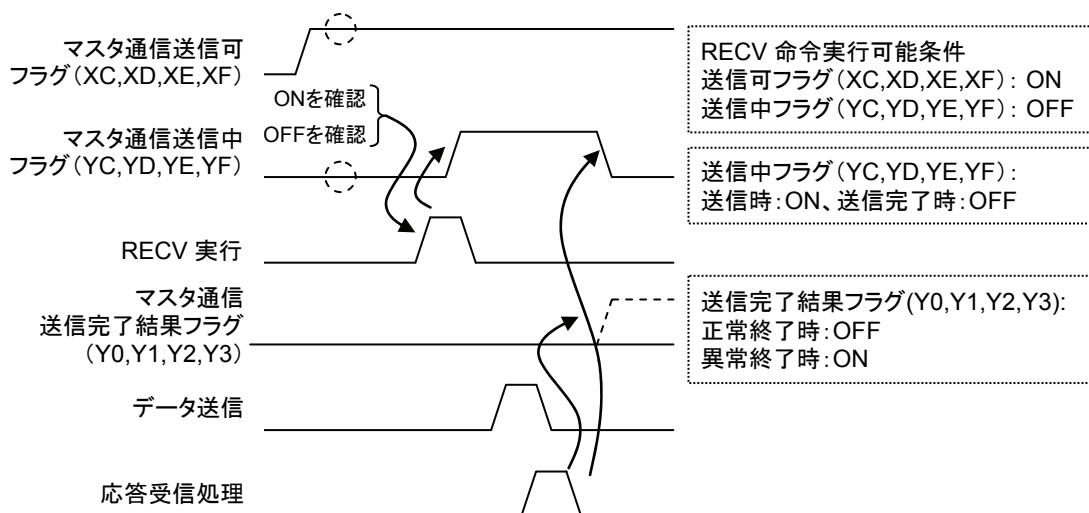
- CPU ユニットの COM1 ポートからコマンドを送信し、外部機器(局番 1)のデータエリア 400001~400002 からデータを読み出し、PLC のデータレジスタ DT100~DT101 へ書き込みます。
- マスタモードになっていること(XC)、同じポートに対し送信が実行中でないこと(YC)を確認し、SEND 命令を起動します。
- UNITSEL 命令で、スロット番号(U0)と COM.ポート No.(U1)を指定します。
- RECV 命令では、相手機器の局番(U1)、使用する MODBUS コマンドと相手機器局番(H0301)、先頭アドレス(400001)、データ数(U2)、データを格納する PLC 側の先頭アドレス(DT100)を指定して実行します。相手機器のアドレスについては、各機器の取扱説明書等でご確認ください。



(注1): RECV 命令のオペランド[S1]は、MODBUS ファンクションコード 16 進 2 桁と相手機器局番は 16 進 2 桁の組合せで指定します。

(注2): 相手機器が FP シリーズ PLC の場合、RECV 命令のオペランド[S2]は、デバイス番号で指定することができます。

## ■ タイムチャート(SCUの場合)



### ◆ ここがポイント

- SCU の場合とは、以下の組み合わせで使用した場合を示します。
  - CPU ユニット標準搭載の COM.0 ポート
  - CPU ユニットに通信カセットを装着 (COM.1~COM.2 ポート)
  - シリアルコミュニケーションユニットに通信カセットを装着 (COM.1~COM.4 ポート)
- 通信カセット(Ethernet タイプ)は、MODBUS には対応していません。

### ■ I/Oの割り付け(CPUユニット内蔵SCUの場合)

COMポート番号			名称	説明
1	2	0		
XC	XD	XE	マスタ通信 送信可フラグ	通信モードで MEWTOCOL-COM、MEWTOCOL7、MODBUS-RTU が設定され、RUN モードになっているときに ON になります。
YC	YD	YE	マスタ通信 送信中フラグ	SEND,RECV 命令で送信中に ON します。 送信完了時に OFF します。
Y0	Y1	Y2	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。 (正常終了時:0 異常終了時:1)

(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。

### ■ I/Oの割り付け(シリアルコミュニケーションユニットの場合)

COMポート番号				名称	説明
1	2	3	4		
XC	XD	XE	XF	マスタ通信 送信可フラグ	通信モードで MEWTOCOL-COM、MEWTOCOL7、MODBUS-RTU が設定され、RUN モードになっているときに ON になります。
YC	YD	YE	YF	マスタ通信 送信中フラグ	SEND,RECV 命令で送信中に ON します。 送信完了時に OFF します。
Y0	Y1	Y2	Y3	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。 (正常終了時:0 異常終了時:1)

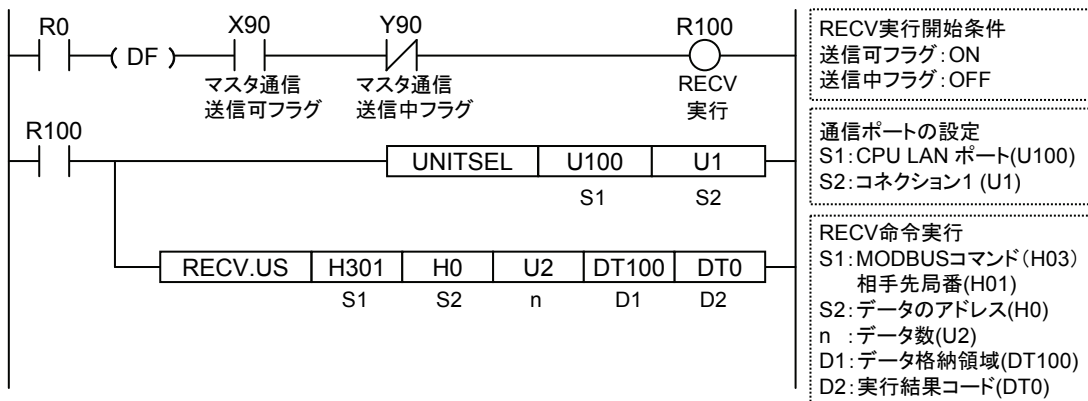
(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。

### ■ プログラム上のご注意(SCUの場合)

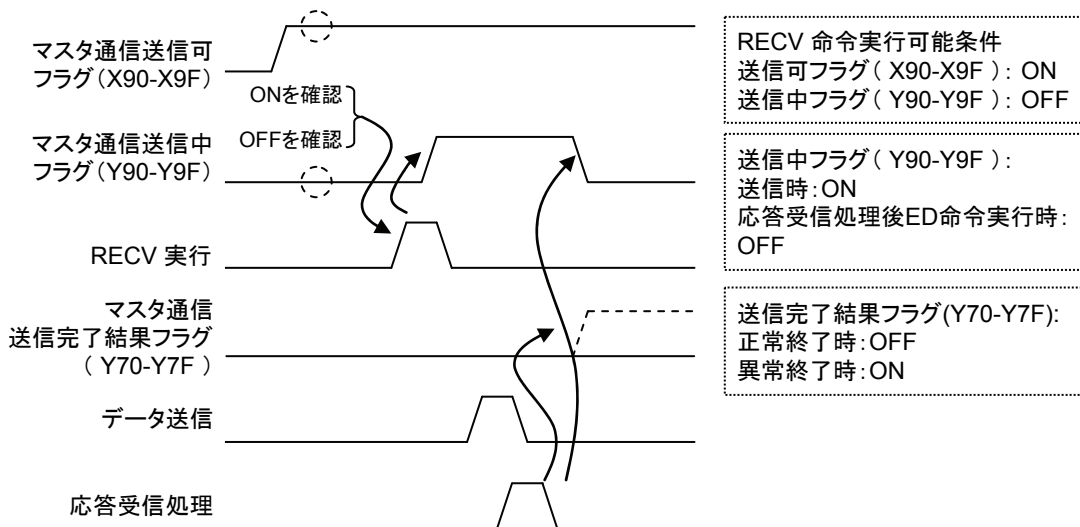
- SEND/RECV 命令の直前に UNITSEL 命令を使用し、通信の対象となるポートを指定してください。
- マスタ通信は、MEWTOCOL または MODBUS が選択されているときのみ有効です。対応するチャンネルの「マスタ通信送信可フラグ」(XC~XF)が ON になっていることを確認し、SEND/RECV 命令を実行してください。
- マスタ通信中の通信ポートに対して、他の SEND/RECV 命令を実行することはできません。「マスタ通信送信中フラグ」(YC~YF)が OFF になっていることを確認し、命令を実行してください。
- スレーブ通信中のポートに対して、SEND/RECV 命令を実行することはできません。応答がない場合、「マスタ通信送信中フラグ」(YC~YF)は、CPU コンフィグレーションで設定されたタイムアウト設定時間まで ON の状態を継続します。
- SEND/RECV 命令は、異なる COM.ポートに対して、最大 16 命令まで同時実行することができます。

## ■ サンプルプログラム(CPUユニット内蔵 ET-LANの場合)

- CPU ユニットの LAN ポートから MODBUS コマンド (03) を送信し、外部機器のデータエリア 400001~400002 (MODBUS アドレス 0000H~0001H) から、データを読み出し、PLC のデータレジスタ DT100~DT101 の内容をへ書き込みます。
- マスタモードで接続1が確立していること (X90)、同じポートに対して送信が実行中でないこと (Y90) を確認し、SEND 命令を起動します。
- UNITSEL 命令で、スロット番号 (LAN ポート: U100) と接続 No. (U1) を指定します。
- RECV 命令では、MODBUS コマンドの種類と相手機器の局番 (H0301)、先頭アドレス (H0)、データ数 (U2)、データを格納する PLC 側の先頭アドレス (DT100) を指定して実行します。相手機器のアドレスについては、各機器の取扱説明書等でご確認ください。



## ■ タイムチャート(CPUユニット内蔵 ET-LANの場合)



## ■ I/Oの割り付け(CPUユニット内蔵 ET-LANの場合)

I/O 番号	名称	説明
X90~X9F	マスタ通信送信可フラグ	マスタ通信で接続が接続状態にあるときに ON になります。
Y90~Y9F	マスタ通信送信中フラグ	SEND, RECV 命令で送信中に ON になります。 応答受信処理完了後、ED 命令が実行されたときに OFF になります。
Y70~Y7F	送信完了結果フラグ	汎用通信・マスタ通信で送信した場合の完了結果を報知します。 (正常終了時: 0、異常終了時: 1)

(注 1): 各接点は、動作状態を読み出すために使用します。ユーザプログラムによる書き込みは行わないでください。

## ■ プログラム上のご注意(CPUユニット内蔵 ET-LANの場合)

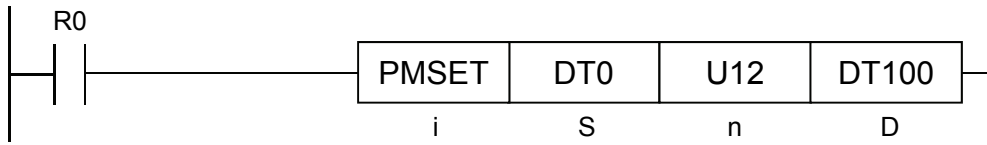
- SEND/RECV 命令の直前に UNITSEL 命令を使用し、通信の対象となるコネクション No.を指定してください。
- マスタ通信は、MEWTOCOL または MODBUS が選択されているときのみ有効です。対応するコネクションの「マスタ通信送信可フラグ」(X90～X9F)が ON になっていることを確認し、SEND/RECV 命令を実行してください。
- マスタ通信中のコネクションに対して、他の SEND/RECV 命令を実行することはできません。「マスタ通信送信中フラグ」(Y90～Y9F)が OFF になっていることを確認し、命令を実行してください。
- スレーブ通信中のコネクションに対し、SEND/RECV 命令を実行することはできません。
- SEND/RECV 命令は、異なるコネクションに対して、最大 16 命令まで同時実行することができます。
- MODBUS-TCP モードでは、SEND/RECV 命令のオペランドに相手先局番を指定します。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	転送先範囲がアクセス可能範囲外の場合にセットします
	UNITSEL 指定の COM ポート、コネクションが存在しないとき、コネクションが通信可能ではない
	[S1]で指定した相手局番が範囲外るとき
	[S2]で指定した相手局送信元データデバイスが不正のとき
	[n]で指定した送信データ数が不正
	[D1]で指定した自局内の送信先データ領域のデータデバイスが不正、領域を超えるとき
	[D2]で指定した結果格納先デバイスが不正
[S2]の整数指定は MODBUS アドレス直接指定タイプのみ。それ以外では不正。	

# PMSET (SCUのパラメータ変更)

## ■ ラダー表記



## ■ オペランド一覧

オペランド	説明
S	通信パラメータとして設定するデータを格納しているエリアの先頭
n	設定ワード数(設定範囲: 1~12)
D	処理結果(1ワード)を格納する自局内のデバイス領域の先頭アドレス

## ■ 指定可能なデバイス (●: 指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデックス 修飾	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""		
S	●	●	●	●			●	●													●	
n	●	●	●	●			●	●								●						●
D	●	●	●	●			●	●														●

## ■ 動作説明

- ユニットの COM ポートの通信パラメータをユーザプログラムにより変更します。
- PMSET 命令の直前に UNITSEL 命令を記述し、パラメータを変更するユニットのスロット No.と COM ポート No.を指定します。
- [S] を先頭とするエリアから、[n] ワード分に変更する通信パラメータをセットし、PMSET 命令を実行すると、ユニットに設定変更要求を行います。
- 変更処理中は、処理結果格納エリア[D] の bit15 が ON になり、処理が終了すると OFF になります。
- 処理結果は、[D]で指定したエリアに格納します。異常がある場合は、[D] の bit14 が ON になり、[D]の下位バイトにエラーコードを格納します。
- PMGET 命令により、設定パラメータを読み出し、変更が必要なパラメータを PMSET 命令でセットすることにより、設定を簡素化することができます。



## ■ パラメータ設定内容

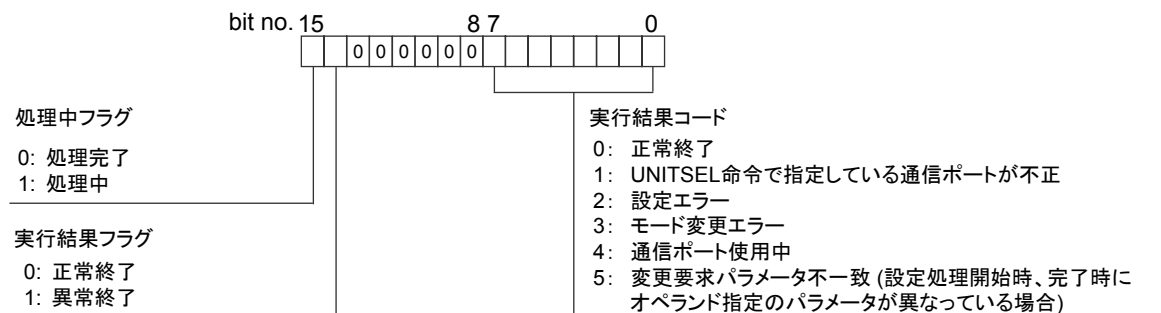
オペランド	パラメータ項目	範囲	設定内容
[S]	通信モード	U0 U1 U2 U8 U9	U0: MEWTOCOL-COM U1: MEWTOCOL7-COM U2: MODBUS-RTU U8: 汎用通信 U9: PLCリンク(ただし、局番のみが変更可能)
[S+1]	局番設定	U1~U999	局番 U1~U999 MEWTOCOL-COM :U1~U99 MEWTOCOL7-COM :U1~U999 MODBUS-RTU :U1~U247 PLCリンク :U1~U16(初期値 0)
[S+2]	通信速度設定	U0~U10	U0:300, U1:600, U2:1200, U3:2400, U4:4800, U5:9600, U6:19200, U7:38400, U8:57600., U9:115200, U10:230400 bps
[S+3]	データ長設定	U0, U1	U0:7ビット長, U1:8ビット長
[S+4]	パリティ設定	U0~U2	U0:パリティ無し, U1:奇数パリティ, U2:偶数パリティ
[S+5]	ストップビット長設定	U0, U1	U0:1ビット, U1:2ビット
[S+6]	RS/CS有効・無効 (注 1)	U0, U1	U0:無効, U1:有効
[S+7]	送信待ち時間	U0~U10000	U0:即時 有効時間=Un×0.01ms(0~100ms)
[S+8]	始端コードSTX	U0, U1	U0:無効, U1:有効
[S+9]	終端設定	U0~U3	U0:cR, U1:cR+Lf, U2:時間, U3:ETX
[S+10]	終端判定時間	U0~U10000	U0:32ビット分 有効時間=Un×0.01ms(但し、終端設定が時間の場合のみ有効)
[S+11]	モデム初期化	U0~U2	U0:モデム初期化しない。 U1:初回の初期化のみ実行。(注 2) U2:設定時にモデム初期化を再実行する。

(注 1): RS-232C の 1ch 5 線式切替カセット使用時のみ RS/CS を選択できます。

(注 2): 設定時(電源 ON、PMSET 命令実行時、RUN モード切替時)にモデムを初期化します。ただし、初回の初期化のみ実行します。(電源再投入を除く)

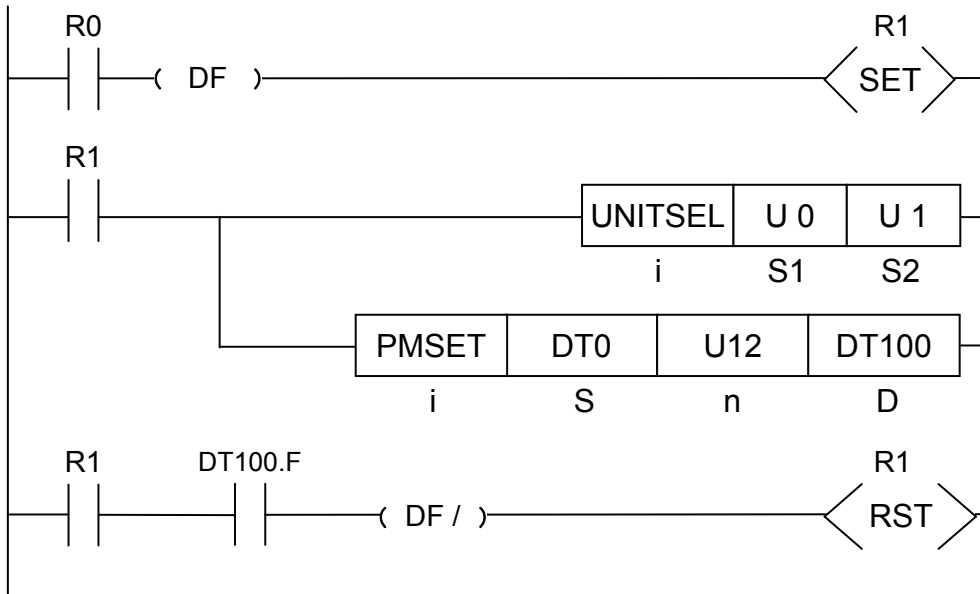
## ■ 処理結果[D]の内容

- 実行結果は 1 ワードのエリアに格納します。
- 下位バイトの実行結果コードは、bit15 処理中フラグが 0 のとき有効です。



## ■ プログラム例

- データレジスタ DT0 を先頭とするエリアに格納されている CPU 内蔵 SCU の COM1 ポートに通信パラメータ 12 ワードを設定する場合を示します。



DT0	U 0	通信モード
DT1	U 1	局番設定
DT2	U 5	通信速度設定
DT3	U 1	データ長設定
DT4	U 1	パリティ設定
DT5	U 0	ストップビット長設定
DT6	U 0	RS / CS 有効・無効
DT7	U 0	送信待ち時間
DT8	U 0	始端コードSTX
DT9	U 0	終端設定
DT10	U 0	終端判定時間
DT11	U 0	モデム初期化

	上位バイト	下位バイト	
DT100	H 0	H 0	処理結果 (正常終了であれば0)

## ■ プログラム上のご注意

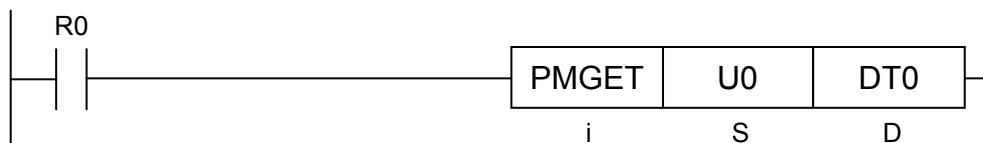
- 処理結果の確認は、[D]で指定したエリアの bit15 (処理中フラグ) が、1 から 0 に変化したときに行なってください。
- 送受信中の COM ポートに対してパラメータ変更を行うと、送受信を中止しパラメータ変更を行います。その際、受信データは失われます。また、送信は途中で中止します。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	UNITSEL 指定の COM ポートが存在しない場合にセットします
	[S]で指定した設定パラメータのデバイスが不正な場合にセットします
	[n]で指定したワード数が指定範囲外の場合にセットします
	[D]で指定した処理結果のデバイス指定が不正な場合にセットします

# PMGET (SCUのパラメータ取得)

## ■ ラダー表記



## ■ オペランド一覧

オペランド	説明
S	取得するデータの種類 0:通信パラメータ、1:COMポート動作状態モニタ
D	取得した通信パラメータ(モニター情報)を格納する領域の先頭アドレス

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●								●					●
D	●	●	●	●			●	●													●

## ■ 動作説明

- ユニットの COM ポートのパラメータを読み出します。
- PMGET 命令の直前に UNITSEL 命令を記述し、パラメータを取得するユニットの-slot No.と COM ポート No.を指定します。
- [S] の内容が "0" のとき、通信パラメータを取得し、[D]を先頭とする 26 ワードのエリアに格納します。
- [S] の内容が "1" のとき、通信モニタ領域状態を取得し、[D]を先頭とする 7 ワードのエリアに格納します。

## ■ SCU通信パラメータの取得内容

オペランド	パラメータ項目	範囲	設定内容
[D]	通信モード	U0 U1 U2 U8 U9	U0: MEWTOCOL-COM U1: MEWTOCOL7-COM U2: MODBUS-RTU U8: 汎用通信 U9: PLC リンク
[D+1]	局番設定	U1~999	局番 U1~U999 MEWTOCOL-COM :U1~U99 MEWTOCOL7-COM :U1~U999 MODBUS-RTU :U1~U247 PLC リンク :U1~U16(初期値 0)
[D+2]	通信速度設定	U0~10	U0:300, U1:600, U2:1200, U3:2400, U4:4800, U5:9600, U6:19200, U7:38400, U8:57600., U9:115200, U10:230400 bps
[D+3]	データ長設定	U0, U1	U0:7 ビット長, U1:8 ビット長
[D+4]	パリティ設定	U0~U2	U0:パリティ無し, U1:奇数パリティ, U2:偶数パリティ
[D+5]	ストップビット長設定	U0, U1	U0:1 ビット, U1:2 ビット
[D+6]	RS / CS 有効・無効 (注 1)	U0, U1	U0:無効, U1:有効
[D+7]	送信待ち時間	U0~10000	U0:即時 有効時間=Un×0.01ms(0~100ms)
[D+8]	始端コードSTX	U0, U1	U0:無効, U1:有効
[D+9]	終端設定	U0~U3	U0:CR, U1:CR+LF, U2:時間, U3:ETX
[D+10]	終端判定時間	U0~ 10000	U0:32 ビット分 有効時間=Un×0.01ms(但し、終端設定が時間の場合のみ有効)
[D+11]	モデム初期化	U0~U2	U0:モデム初期化しない。 U1:初回の初期化のみ実行。(注 2) U2:設定時にモデム初期化を再実行する。
[D+12]	予約領域	U0	予約領域
[D+13]	予約領域	U0	予約領域
[D+14]	リンク領域ブロック番号	U0,U1	リンクリレー・リンクレジスタ領域のブロック番号
[D+15]	PLC リンク W0 最大局番	U2~16	範囲外の数値の場合は、16 として扱う。
[D+16]	リンクリレー範囲	U0~64	通信に使用するリンクリレーの範囲指定 (指定ブロック内の相対値)
[D+17]	リンクレジスタ範囲	U0~128	通信に使用するリンクレジスタの範囲指定 (指定ブロック内の相対値)
[D+18]	リンクリレー送信開始番号	U0~63	リンクリレー送信開始番号 (ワード数指定、指定ブロック内の相対値)
[D+19]	リンクリレー送信サイズ	U0~64	リンクリレー送信サイズ (ワード数指定)
[D+20]	リンクレジスタ送信開始番号	U0~127	リンクレジスタ送信開始番号 (ワード数指定、指定ブロック内の相対値)
[D+21]	リンクレジスタ送信サイズ	U0~127	リンクレジスタ送信サイズ (ワード数指定)
[D+22]	予約領域	U0	予約領域
[D+23]	予約領域	U0	予約領域
[D+24]	予約領域	U0	予約領域
[D+25]	予約領域	U0	予約領域

(注 1): RS-232C の 1ch 5 線式切替カセット使用時のみ RS/CS を選択できます。

(注 2): 設定時(電源 ON、PMGET 命令実行時、RUN モード切替時)にモデムを初期化します。ただし、初回の初期化のみ実行します。(電源再投入を除く)

(注 3): [D+14]~[D+21]の設定は、COM1 ポートで通信モードが PLC リンクの場合のみ有効です。

## ■ SCU COMポート動作状態モニター情報

オペランド	モニター情報の種類	範囲	設定内容
[D]	動作モード	U 0 U 1 U 2 U 8 U 9	U0: MEWTOCOL-COM U1: MEWTOCOL7-COM U2: MODBUS-RTU U8: 汎用通信 U9: PLC リンク
[D+1]	通信カセット判別	U0 U232 U422 U485	U 0: 通信カセット無し U 232: RS-232C U 422: RS-422 U 485: RS-485
[D+2]	受信エラーコード	bit9: 受信バッファFULL bit8: 受信バッファオーバーフロー bit2: パリティ不一致 bit1: ストップビット未検出(フレームエラー) bit0: 受信バッファオーバーラン	
[D+3]	受信エラー発生回数	受信エラーコードの下位バイトに格納される受信エラーの検知回数(符号無し 16 ビット循環)	
[D+4]	設定エラーコード	bit9: 送信データ数異常 bit8: 通信パラメータの設定異常 bit0: モード設定・変更異常(設定・変更できないモード番号が指定された)	
[D+5]	エラーパラメータ No	U 1~12	範囲外データを指定したパラメータ番号 (通信パラメータの設定異常が発生している場合のみ有効)
[D+6]	モデム初期化状態	U 0000 U 0100 U 0200 U 02FF	無処理 初期化中 初期化完了成功 初期化完了失敗

## ■ プログラム上のご注意

- PMGET 命令の直前に、UNITSEL 命令を使用し、設定の対象となるポートを指定してください。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合
	転送先範囲がアクセス可能範囲外の場合
	UNITSEL 指定のスロットに SCU ユニットが存在しない場合
	UNITSEL 指定の COM ポートが存在しないとき
	[D]で指定したパラメータ格納先のデバイスが不正

# RDET (ET-LANステータス読み出し)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
D	ステータスを格納する領域の先頭[D]~[D+6] の 7 ワードのエリアに格納。

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	""	
D	●	●	●	●			●	●													●

## ■ 動作説明

- ET-LAN の全コネクションの状態を示すステータスサマリーを取得します。
- RDET 命令の直前に UNITSEL 命令を記述し、対象となる ET-LAN ポートを指定します。コネクション No.は 1~16 の範囲で任意の値を設定します。
- 取得した情報は、割り付けに従って Hex 形式の整数値に変換し、[D]を先頭とする 7 ワードのエリアに格納します。

## ■ ET-LANステータス情報

- 全コネクションのコネクション状況
- OPEN 状況
- OPEN 異常状況
- FTP サーバー接続中コネクション数

## ■ ET-LAN ステータス情報

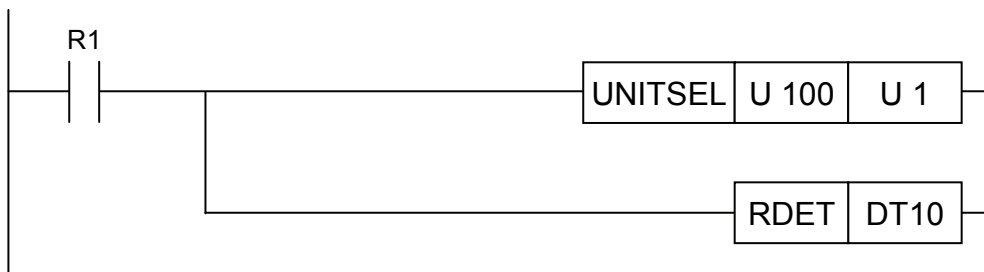
オペランド	データ名称	格納されるデータ
[D]	コネクション状況サマリー	下位ワード
[D+1]		上位ワード
[D+2]	OPEN 状況サマリー	下位ワード
[D+3]		上位ワード
[D+4]	OPEN 異常状況サマリー	下位ワード
[D+5]		上位ワード
[D+6]	FTP サーバ接続中コネクション数	

(注)コネクション状況サマリー、OPEN 状況サマリー、OPEN 異常状況サマリーのコネクションとビット対応

上位ワード							下位ワード															
b15-b9	b8	b7-b4	b3	b2	b1	b0	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-	FTP-S	-	U16	U15	U14	U13	U12	U11	U10	U9	U8	U7	U6	U5	U4	U3	U2	U1	S4	S3	S2	S1

## ■ プログラム例

- CPU 内蔵 ET-LAN の全コネクションサマリーを取得し、DT10 を先頭とする 7 ワードのエリアに格納します。



## ■ プログラム上のご注意

- UNITSEL 命令により、通信対象 ET-LAN のスロット番号、コネクション番号の設定が必要です。

## ■ フラグ動作

名称	説明
SR7 SR8 (ER)	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
	転送先範囲がアクセス可能範囲外の場合にセットします
	UNITSEL で指定したコネクションが存在しない場合や、数値が範囲外の場合
	[D]で指定したパラメータ格納先のデバイスが不正

# POSSET (位置決め開始テーブル設定)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S1	位置決めユニットを装着している スロット No.(符号無し 16 ビット整数)
S2	位置決めテーブルを起動する軸 No.(符号無し 16 ビット整数) 1~4:1~4 軸、8:仮想軸
S3	位置制御を起動するテーブル No.(符号無し 16 ビット整数) 1~600、10001~10025

(注): 開始テーブル No.に指定可能範囲外の値が設定された場合は位置決めユニットによりエラーを報知します

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●	●	●					●	●				●
S2	●	●	●	●			●	●	●	●	●					●	●				●
S3	●	●	●	●			●	●	●	●	●					●	●				●

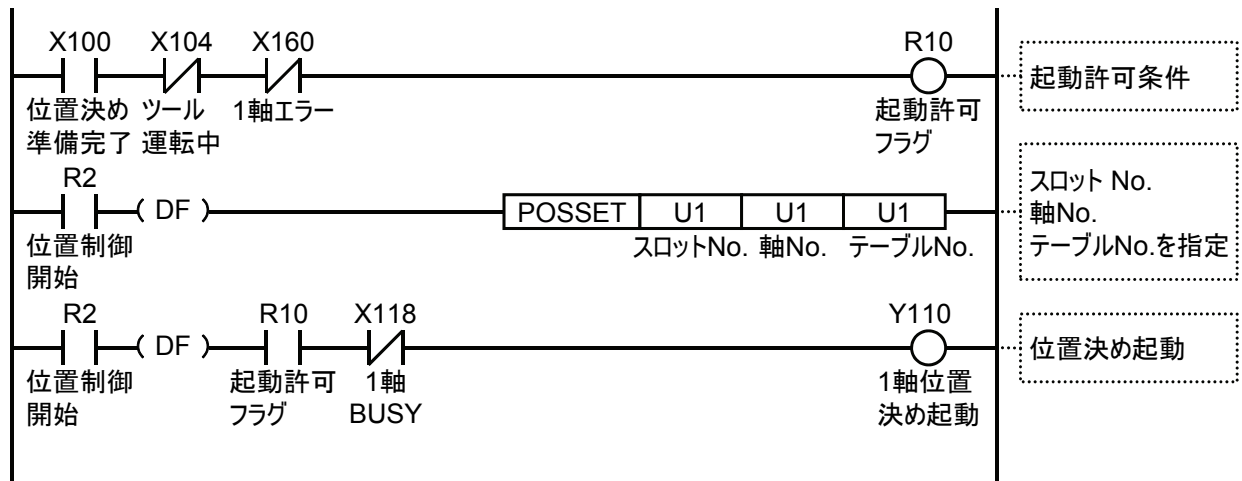
\*1: 16ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- 位置決めを起動するプログラムの直前に記述し、起動する位置決めデータテーブルをセットします。
- [S1] と [S2] で指定された位置決めユニットの軸 No.について、[S3]で指定したデータテーブルをセットします。
- 位置決めデータテーブルのデータは、ツールソフトウェア FPWIN GR7 内の Configurator PM7 またはユーザプログラムにより設定します。
- 位置決めパラメータは、ツールソフトウェア FPWIN GR7 のコンフィグレーションメニューで設定します。



## ■ プログラム例



## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	スロット No.、軸 No.が指定可能範囲外となった場合にセットします

# PSTRD (軸ステータス取得)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S1	位置決めユニットを装着している スロット No.(符号無し 16 ビット整数)
S2	軸ステータス情報を読み出す軸 No.(符号無し 16 ビット整数) 1~4:1~4 軸、8: 仮想軸
D	軸ステータス情報を格納するデバイスアドレス(符号無し 16 ビット整数)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス										32ビット デバイス			整数			実数		文字	インデック ス修飾 *1	
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF		" "
S1	●	●	●	●			●	●	●	●	●					●	●				●
S2	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

\*1: 16 ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- [S1] と [S2] で指定された位置決めユニットの軸 No.について、主なフラグの状態を軸ステータスとして取得します。
- 取得した情報は、割り付けに従って Hex 形式の整数値に変換し、[D]で指定したエリアに格納します。

## ■ 軸ステータス情報の種類

ステータス情報	内容
ツール運転中	指定した軸によらず、Configurator PM でツール運転している場合に ON する
エラー報知	指定した軸にエラーが発生した場合に ON する
ワーニング報知	指定した軸にワーニングが発生した場合に ON する
BUSY	指定した軸が動作中である場合に ON する
動作完了	指定した軸が動作完了した場合に ON する
原点復帰完了	指定した軸が原点復帰完了した場合に ON する

### ■ [D]に格納される軸ステータス情報の割り付け

bit	ステータス情報	1軸	2軸	3軸	4軸	仮想軸
0	ツール運転中	X4	X4	X4	X4	X4
1	エラー報知	X60	X61	X62	X63	X67
2	ワーニング報知	X68	X69	X6A	X6B	X6F
3	BUSY	X18	X19	X1A	X1B	X1F
4	動作完了	X20	X21	X22	X23	X27
5	原点復帰完了	X28	X29	X2A	X2B	X2F

### ■ 処理例

- スロット No.3 に装着されている位置決めユニットの 1 軸目の軸ステータス情報を読み出します。

bit	ステータス情報	1軸	値
0	ツール運転中	IN4	0
1	エラー報知	IN60	0
2	ワーニング報知	IN68	0
3	BUSY	IN18	1
4	動作完了	IN20	0
5	原点復帰完了	IN28	0

	値
DT10	H 0008

### ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	スロット No.、軸 No.が指定可能範囲外となった場合にセットします

# PERRD (位置決めユニットエラー/ワーニング取得)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S1	位置決めユニットを装着している スロット No.(符号無し 16ビット整数)
S2	エラー/ワーニング情報を読み出す軸 No.(符号無し 16ビット整数)1~4:1~4 軸、8:仮想軸
D	エラー/ワーニングコードを格納するデバイスアドレス(符号無し 16ビット整数)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデッ クス修飾 *1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S1	●	●	●	●			●	●	●	●	●					●	●				●
S2	●	●	●	●			●	●	●	●	●					●	●				●
D	●	●	●	●			●	●	●		●										●

\*1: 16ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- [S1] と [S2] で指定された位置決めユニットの軸 No.について、それぞれの報知バッファ 1 に格納されているエラーコード、ワーニングコードを取得します。
- エラーコードを[D]、ワーニングコードを [D+1] で指定したエリアに格納します。

## ■ 処理例

- スロット No.1 に装着されている位置決めユニットの 3 軸目のエラーコード、ワーニングコードを読み出します。

分類	名称	UM(Hex)	値
エラー情報	3軸エラーコード 報知/バッファ1	UM 0015A	H 4022
ワーニング情報	3軸ワーニングコード 報知/バッファ1	UM 001E2	H B010

・格納先

[D].....DT0 H 4022

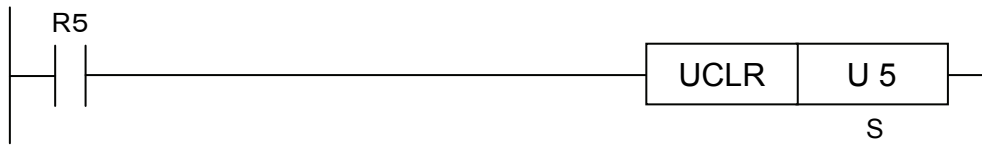
[D+1]...DT1 H B010

## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8	スロット No.、軸 No.が指定可能範囲外となった場合にセットします
(ER)	転送先範囲がアクセス可能範囲外の場合にセットします

# UCLR (エラー/ワーニングクリア)

## ■ ラダー表記



## ■ 指定可能な演算単位 (●:指定可能)

- 演算単位なし。

## ■ オペランド一覧

オペランド	説明
S	スロット No.を指定します(符号無し 16 ビット整数)

## ■ 指定可能なデバイス (●:指定可能)

オペランド	16ビット デバイス											32ビット デバイス			整数			実数		文字	インデック ス修飾 *1
	WX	WY	WR	WL	WS	SD	DT	LD	UM	WI	WO	TS CS	TE CE	IX	K	U	H	SF	DF	" "	
S	●	●	●	●			●	●		●						●	●				●

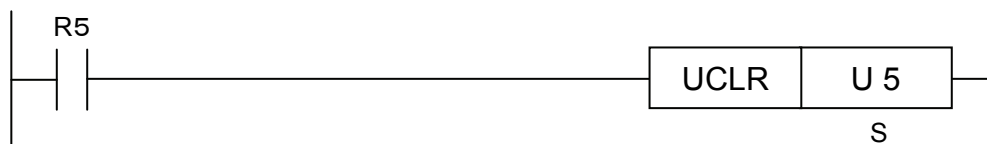
\*1: 16ビットデバイス、整数定数のみ修飾可能(実数定数、文字定数には指定不可)

## ■ 動作説明

- [S]で指定したスロット No.に装着されているユニットのエラー/ワーニングをクリアします。

## ■ プログラム例

例)スロットNo.5に装着されているユニットのエラー/ワーニングをクリアする場合  
[S]...U5



## ■ フラグ動作

名称	説明
SR7	間接アクセス時(インデックス修飾)に範囲外となった場合にセットします
SR8 (ER)	
	スロット No.、軸 No.が指定可能範囲外となった場合にセットします



# 4

## プログラム上のご注意

## 4-1 共通の注意事項

FP7 シリーズは、従来機種 FP シリーズと以下のような違いがあります。

### ■ キャリーフラグ、比較フラグの動作(●:変化する、-:変化しない)

命令の種類	ニモニック	命令名称	システムリレー番号			
			キャリーフラグ	比較フラグ		
			SR9(CY)	SRA(>)	SRB(=)	SRC(<)
データ比較命令	CMP	データ比較	-	●	●	●
	WIN	帯域比較	-	●	●	●
	BCMP	ブロック一致検出	-	-	●	-
ビット比較命令	BTT	16ビットデータの指定ビットテスト	-	-	●	-
	STC	キャリーのセット	●	-	-	-
	CLC	キャリーにリセット	●	-	-	-
文字列操作命令	SCMP	文字列比較	-	●	●	●

- 演算実行時のキャリーフラグ SR9(CY) 、比較フラグ SRA(>)、SRB(=)、SRC(<) は、上記命令で動作します。
- 上記以外の命令では、演算結果がオーバーフロー、アンダーフローとなった場合、演算結果が0であっても、変化しません。

### ■ オーバーフロー、アンダーフロー時の動作

- オーバーフロー、アンダーフロー時も、キャリーフラグ SR9(CY) は変化しません。
- 演算単位の設定を誤ると、オーバーフロー、アンダーフローが発生するおそれがありますので、ご注意ください。

### ■ エラーフラグ

- 正常演算時でも、演算エラーフラグをクリアしません。
- エラーフラグをクリアする場合は、ERR 命令を使用してください。

### ■ 演算用デバイス領域

- 演算単位が 32 ビット(UL,SL,SF)または 64 ビット(DF)の場合、各演算用デバイスの終端付近を指定すると、PLC 内部でアドレスが連続しているデバイスのメモリを上書きします。誤って、読み出し、書き込みが行なわれないようオペランドを指定してください。

例) 転送命令で演算単位が 32 ビットの場合、転送先 [D] に WY511 を指定すると、WL0 の先頭を上書きします。

DT0	H 0011	WY509	H 1111
DT1	H 2233	WY510	H 2222
DT2	H 4455	WY511	H 2233
DT3	H 6677	WL0	H 4455
DT4	H 8899	WL1	H 5555



## 4-2 時刻、時間データ

### ■ CPUユニット内蔵カレンダータイマ

- カレンダータイマは、ツールソフトウェア FPWINGR7 の“PLC 日付/時刻設定”メニュー、または、TIMEWT 命令で時刻合わせを行ないます。
- セットされた値は、システムデータレジスタレジスタ SD50～56 に以下のように格納されます。

デバイス番号	命令名称	データ範囲	備考
SD50	カレンダータイマ(年)	00～99	年号は西暦下 2 桁、2099 年まで対応。うるう年にも対応しています。設定時は、うるう年を考慮する必要があります。年が 4 で割り切れる場合は閏年で 29 日に設定可能。
SD51	カレンダータイマ(月)	01～12	
SD52	カレンダータイマ(日)	01～31	
SD53	カレンダータイマ(時)	0～23	
SD54	カレンダータイマ(分)	0～59	
SD55	カレンダータイマ(秒)	0～59	
SD56	カレンダータイマ(曜日)	0～6	0: 日、1: 月、2: 火、、3: 水、、4: 木、、5: 金、、6: 土

### ■ 時刻データ、時間データを扱う命令

ニモニック	オペランド	機能
HMSS	S,D	(時間データ) → (秒データ)
SHMS	S,D	(秒データ) → (時間データ)
CADD	S1,S2,D	(時刻データ) + (時間データ) → (時刻データ)
CSUB	S1,S2,D	(時刻データ) - (時間データ) → (時刻データ)
TMSEC	S,D	(時刻データ) → (基準時刻からの秒データ)
SECTM	S,D	(基準時刻からの秒データ) → (時刻データ)
TIMEWT	S	(時刻データ) → (SD50～SD56)

### ■ 時刻データのフォーマット

- 年、月、日、時、分、秒の各単位に 1 ワード 16 ビットバイナリデータが割り付けられ、6 ワードの固まりとして扱います。

例) オペランドに DT0 を指定した場合

	ワード	範囲
DT0	年	00～99
DT1	月	01～12
DT2	日	01～31
DT3	時	0～23
DT4	分	0～59
DT5	秒	0～59

### ■ 時間データのフォーマット

- 時、分、秒の各単位に 1 ワード 16 ビットバイナリデータが割り付けられ、3 ワードの固まりとして扱います。
- 時データは、0～9999 の範囲で指定することができます。

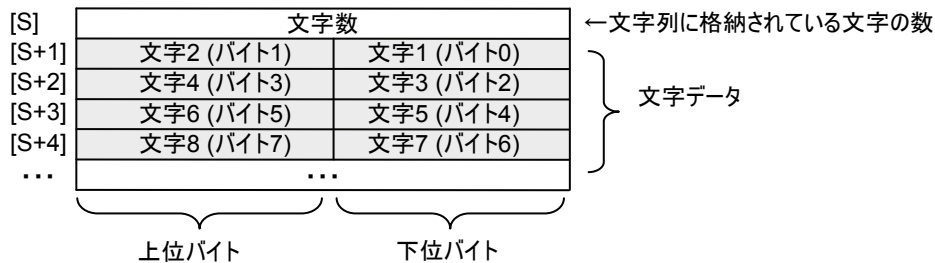
例) オペランドに DT0 を指定した場合

	ワード	範囲
DT0	時	0～9999
DT1	分	0～59
DT2	秒	0～59

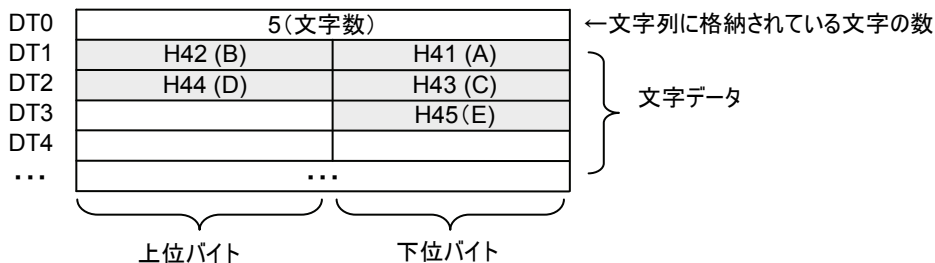
## 4-3 文字列命令のデータテーブル

### ■ データテーブルの構造

- 文字列命令で扱うデータは、先頭ワードに文字数、以降に文字データを格納します。
- 文字列データの最大指定サイズは 4,096 バイトです。



例) 文字数:5、文字データ:"ABCDE"の文字列データテーブルを指定した場合

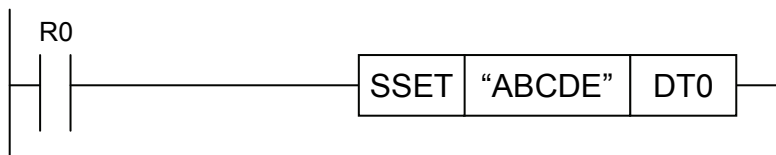


### ■ SSET命令による文字列データの変換

- SSET 命令を用いると、任意の文字列を文字列データテーブルに容易に変換することができます。

例) 文字列データ "ABCDE" 変換する場合

文字数を DT0、ASCII 変換した文字データを DT1 以降のデータレジスタに格納します。



## 4-4 浮動小数点形実数演算

浮動小数点形実数演算時に、演算結果または入力値が無限小、無限大になる場合の動作は以下のようになります。

### ■ 演算結果が無限小になる場合

命令	SIN	COS	TAN	ASIN	ACOS	ATAN	ATAN2	SINH	COSH	TANH	EXP	LN	LOG	PWR	SQR
入力値	—	—	$-\pi/2$	●	●	—	●	—	—	—	—	●	●	●	●

(注 1)：—印の命令は、無限小となりません。

(注 2)：●印の命令は、演算エラーとなります。

(注 3)：入力値が指定可能範囲外の場合は演算エラーを出力します

### ■ 演算結果が無限大になる場合

命令	SIN	COS	TAN	ASIN	ACOS	ATAN	ATAN2	SINH	COSH	TANH	EXP	LN	LOG	PWR	SQR
入力値	—	—	$\pi/2$	●	●	—	●	—	—	—	—	●	●	●	●

(注 1)：—印の命令は、無限大となりません。

(注 2)：●印の命令は、演算エラーとなります。

(注 3)：入力値が指定可能範囲外の場合は演算エラーを出力します

### ■ 入力値に無限小 ( $-\infty$ ) を指定した場合

命令	SIN	COS	TAN	ASIN	ACOS	ATAN	ATAN2	SINH	COSH	TANH	EXP	LN	LOG	PWR	SQR
入力値	nan	nan	nan	●	●	$-\pi/2$	●	$-\infty$	$+\infty$	-1.0	+0.0	●	●	●	●

(注 1)：nan 印の命令は、不定値となります。

(注 2)：●印の命令は、演算エラーとなります。

(注 3)：入力値が指定可能範囲外の場合は演算エラーを出力します

### ■ 入力値に無限大 ( $+\infty$ ) を指定した場合

命令	SIN	COS	TAN	ASIN	ACOS	ATAN	ATAN2	SINH	COSH	TANH	EXP	LN	LOG	PWR	SQR
入力値	nan	nan	nan	●	●	$\pi/2$	●	$+\infty$	$+\infty$	+1.0	$+\infty$	●	●	●	●

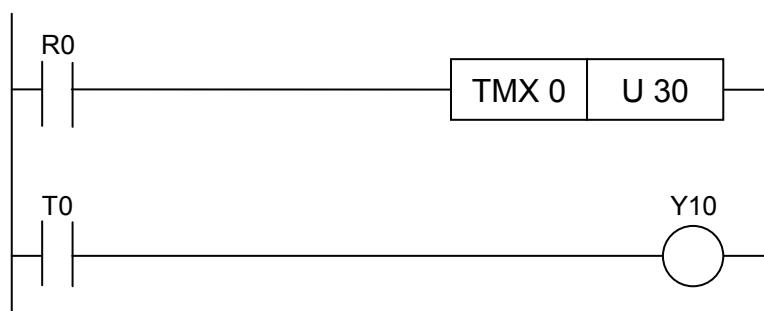
(注 1)：nan 印の命令は、不定値となります。

(注 2)：●印の命令は、演算エラーとなります。

(注 3)：入力値が指定可能範囲外の場合は演算エラーを出力します

## 4-5 RUN中のタイマ・カウンタ設定値の変更

(1) プログラム上の定数を書き換える方法



### ■ プログラム上の設定値(定数)

プログラム上の定数の書き換えは、次の条件で可能です。

- ・運転方法:RAM/ROM 運転時のみ
- ・書き換え方法:①ツールソフトウェアを使用する方法

### ■ ツールソフトウェア FPWIN GR7 を使用する方法

<手順>

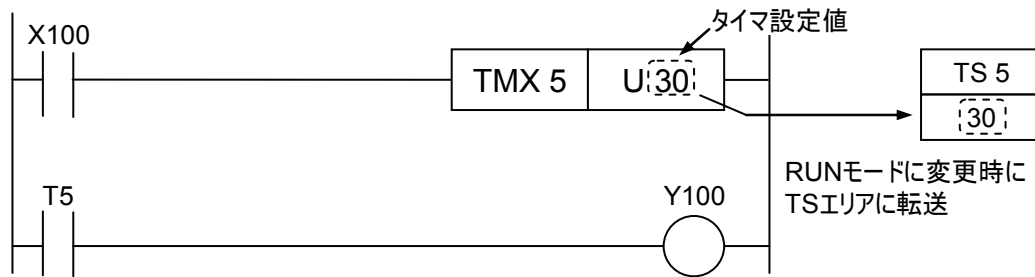
タイマ 0 の設定値を U30→U50 に変更する例です。

- 1) カーソルをタイマ 0 の設定値 "U30" に合わせる。
- 2) 新しい定数 "U50" を入力し、リターンキーを押す。
- 3) [PB 変換] を実行するため、<Ctrl>+<F1>キーを押す。
- 4) 確認のメッセージが出たら、[はい(Y)] を押す。

### ■ 変更後の動作と注意点

- 動作中のタイマ・カウンタは、そのまま動作を継続します。変更した設定値で動作を開始するのは、次に実行条件が OFF から ON になったときです。
- プログラム上の定数を書き換える場合プログラム自体が書き換わりますので、モードを切り替えて再度 RUN にした時や電源を入れ直した時には、変更した設定値でプリセットされます。

## (2)設定値エリアの値を書き換える方法



### ■ 設定値エリア TS/CS の値の変更

設定値エリア TS/CS の書き換えは、次の条件で可能です。

- ・運転方法:RAM/ROM 運転時
- ・書き換え方法:①ツールソフトウェアを使用する方法、②プログラム(応用命令)による方法

### ■ 変更後の動作と注意点

- 変更した時、動作中のタイマ・カウンタは、そのまま動作を継続します。変更した設定値で動作を開始するのは、次に実行条件が OFF から ON になったときです。
- この方法の場合、設定値エリア TS/CS の値は変わっても、プログラム自体は書き替わりませんので、モードを切り替えて再度 RUN にした時や電源を入れ直した場合には、以下のような動作になります。

#### 1) 設定値を U 定数で指定している場合

設定値エリア TS/CS には定数がプリセットされます。変更した値は無効になります。

#### 2) 設定値を設定値エリア No.で指定している場合

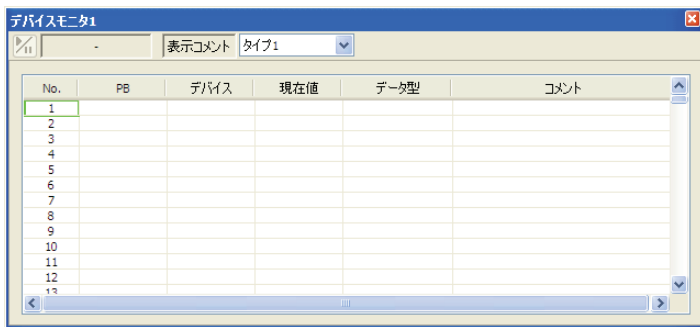
タイマの場合、設定値エリア TS には 0 をプリセットします。

カウンタの場合、設定値エリア CS には、次ページの方法で変更した値にプリセットします。

## ■ 方法1 ツールソフトウェア FPWIN GR7を使用する方法

<手順>

- 1) メニューバーから、"オンライン" → "デバイスモニタ"を選択する。  
"デバイスモニタ"ダイアログボックスが表示されます。



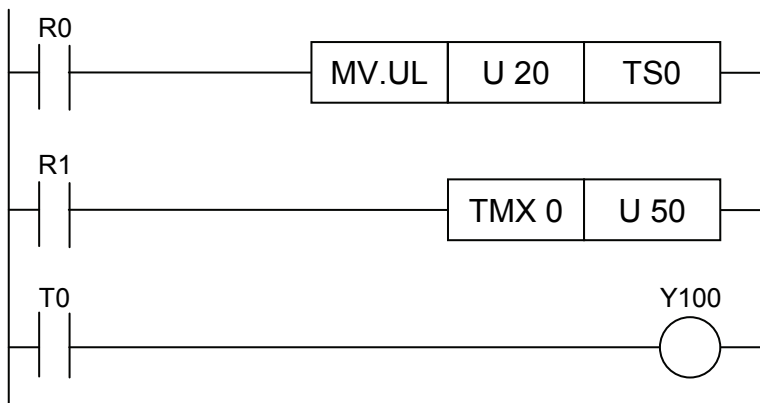
- 2) デバイスの項でダブルクリックする。  
"モニタデバイス登録"ダイアログボックスが表示されます。
- 3) デバイス種別から、TS(タイマ設定値)またはCS(カウンタ設定値)を選択し、任意のデバイス No.を入力する。  
デバイスモニタダイアログボックスに表示されます。
- 4) 現在値に、任意の値を入力し、リターンキーを押す。  
設定値エリアが更新されます。

## ■ 方法2 :プログラム(応用命令)による方法

- 入力の条件などにより、タイマ・カウンタの設定値を変更したい場合は、下記のように応用命令を使って、変更したいタイマまたはカウンタの設定値エリア TS の値を書き換えてください。

<例> 入力 R0 が ON の時に設定値を U20 とする場合

R0 が ON のときタイマの設定値が 5 秒→2 秒に変わります



- 設定値エリアにデータレジスタ DT を指定し、MV 命令などで転送する値を変えることによって、設定値を変更することができます。設定値エリアのオペランドに、設定値エリア (TS/CS) の番号を指定することもできます。

## 4-6 2重出力の使用について

### ■ 2重出力について

- 2重出力は、1つのシーケンスプログラム内に同じ出力を重複して指定している場合をいいます。
- 2重出力とみなされるのは「OT」命令、「KP」命令に同じ出力を指定した場合です。
- 「SET」命令、「RST」命令、応用命令(データ転送など)で同じ出力を使っても2重出力とみなされません。
- 2重出力の状態のまま、RUNモードにすると通常の場合エラーとなります。ERROR LEDが点灯し、自己診断フラグSR0がONとなります。

### ■ 2重出力のチェック方法

- プログラムが2重出力になっているかどうかは、ツールソフトウェア FPWIN GR7 を使って、以下の方法でチェックすることができます。

1) メニューバーから、“デバッグ” → “プロジェクトのトータルチェック”を選択する。

“プロジェクトのトータルチェック”ダイアログボックスが表示されます。

2) [実行(E)]ボタンを押す。

2重出力がある場合は、PB名、アドレス、エラー内容(2重使用定義エラー)が表示されます。

3) 任意の行を選択し、[ジャンプ]ボタンを押す。

2重出力が使用されている命令にカーソルが移動します。

### ■ 2重出力の許可

- プログラムの内容によって、出力を重複して使いたい場合は、2重出力を許可することができます。
- このような場合は、“CPUコンフィグレーション”の“動作選択”の“2重出力の許可”の内容を許可に変更してください。
- この場合は、プログラムを実行してもエラーとなりません。



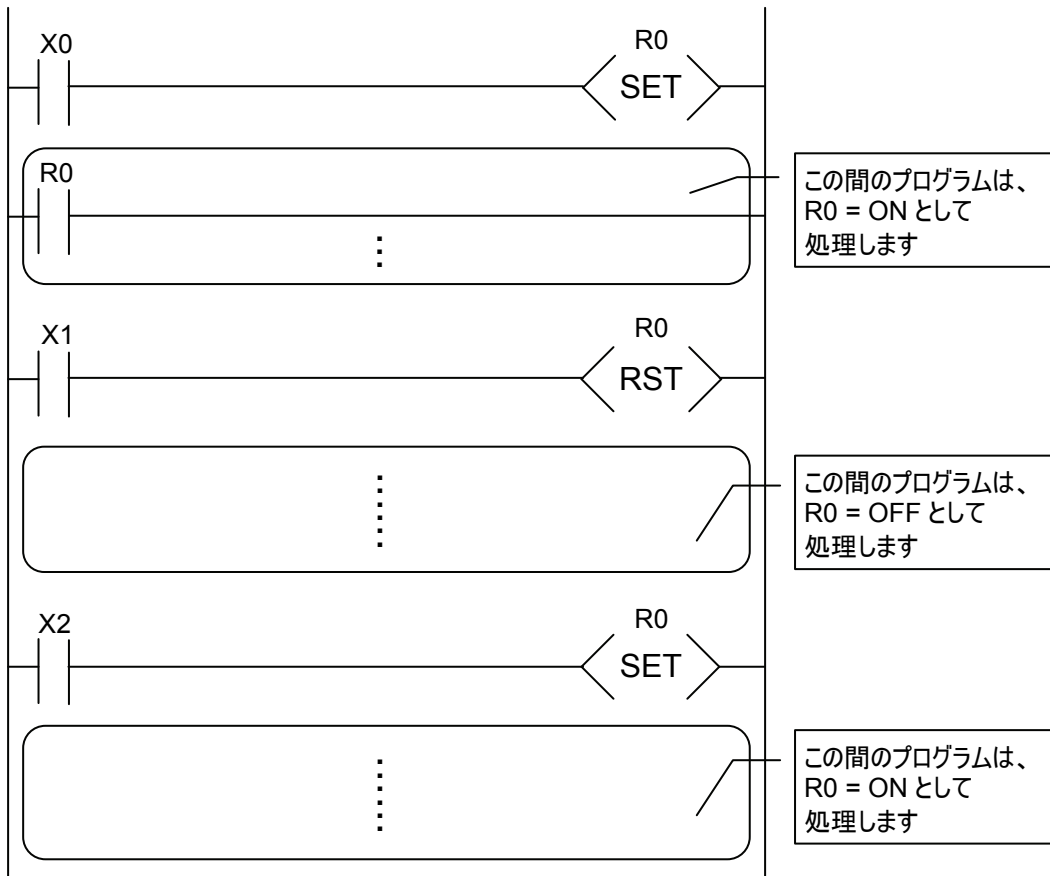
### ◆ ご注意！

- ツールソフトウェア FPWINGR7 上で、プロジェクトのトータルチェックを実行しても、先頭に使用されている命令は表示されません。2重使用となっている2番目以降の出力を表示します。

### ■ 同じ出力を重複して使用したときの処理

- OT 命令、KP 命令、SET 命令、RST 命令、転送命令などで内部リレーや出力リレーに出力する命令を同じコイルを指定した場合、演算中はステップ毎にその内容が書き替わります。

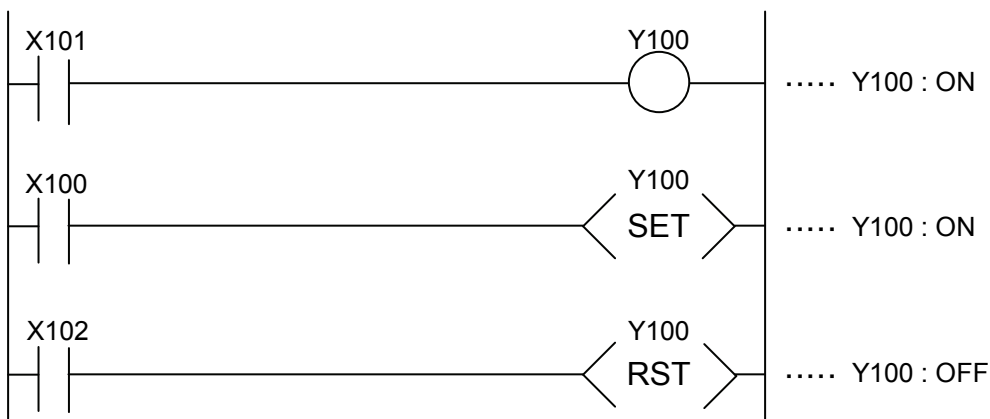
<例> SET、RST 命令、OT 命令を使った場合の処理:( X100~X102 がすべて ON の場合)



### ■ 演算結果の決定

- OT 命令、KP 命令、SET、RST 命令、転送命令などで同じ出力を重複して使用した場合、I/O リフレッシュ時に得られる出力は、最終的な演算結果で決まります。

<例> OT 命令、SET 命令、RST 命令で同じ出力リレー Y100 に出力する場合





## 4-7 立ち上がり検出方式について

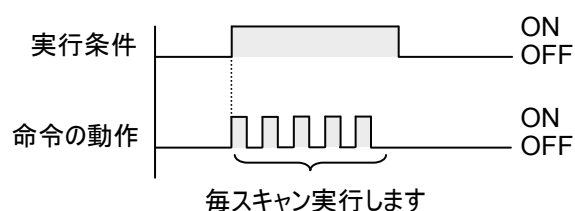
### ■ 立ち上がり検出を行う命令

- ①DF (立ち上がり微分)
- ②CT (カウンタ)のカウンタ入力
- ③UDC (アップダウンカウンタ)のカウンタ入力
- ④SR (シフトレジスタ)のシフト入力
- ⑤LRSR (左右シフトレジスタ)のシフト入力
- ⑥微分実行型応用命令 (P と番号で指定する命令)

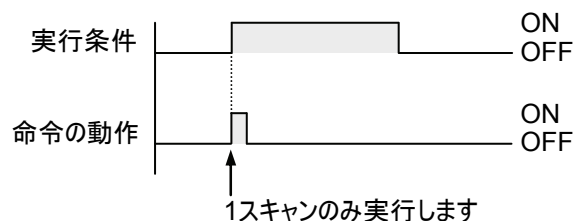
### ■ 立ち上がり検出方式とは

- 立ち上がり検出方式の命令は、実行条件が OFF から ON に変化した 1 スキャンのみ実行される命令です。

#### ①通常の入力検出



#### ②立ち上がり検出



### ■ 立ち上がり検出の方法

- 前回実行されたときの実行条件と今回の実行条件を比較して、前回は OFF で今回は ON の時のみ命令を実行します。
- それ以外の場合は、命令を実行しません。

### ■ 立ち上がり検出を行う命令を使用する時のご注意

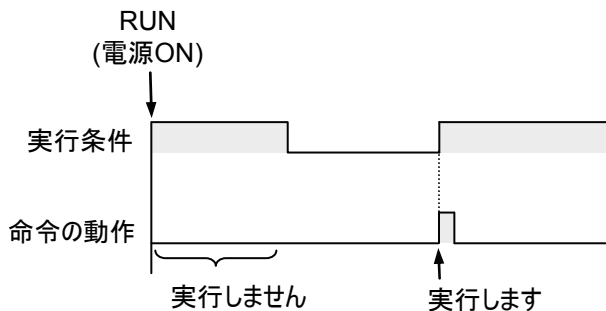
- 電源立ち上げ時など RUN 開始時には、実行条件の OFF→ON の変化が検出されないため、命令が実行されません。
- 下記の①～⑥のような、命令を実行する順序を変える命令と合わせて使用する場合、入力のタイミングにより、命令の動作が変わりますのでご注意ください。

<立ち上がり検出を行う命令を使用する場合に注意が必要な命令>

- ①MC～MCE 命令
- ②JP～LBL 命令
- ③LOOP～LBL 命令
- ④CNDE 命令
- ⑤ステップラダー命令
- ⑥サブルーチン命令

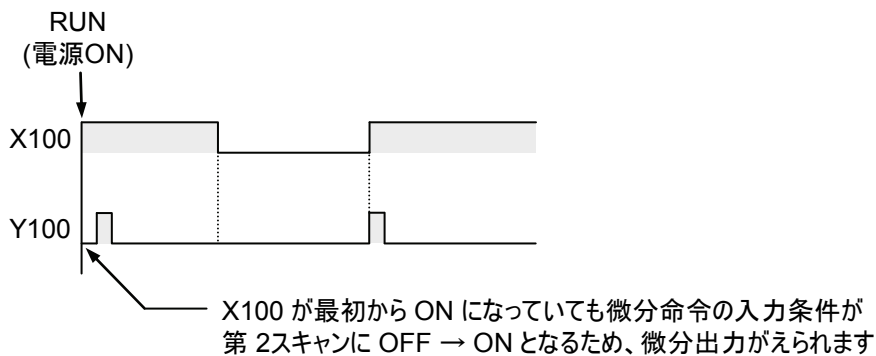
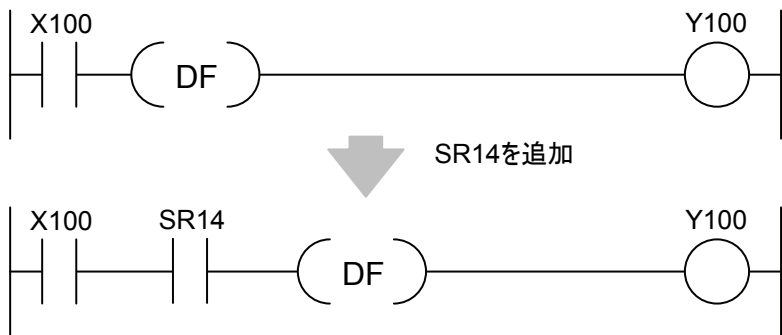
## ■ 運転開始時の動作と注意点

- 立ち上がり検出を行う命令は、RUN モードに切り替えた時や RUN モード状態で電源を立ち上げた時に、すでに実行条件が ON になっていると、実行されません。

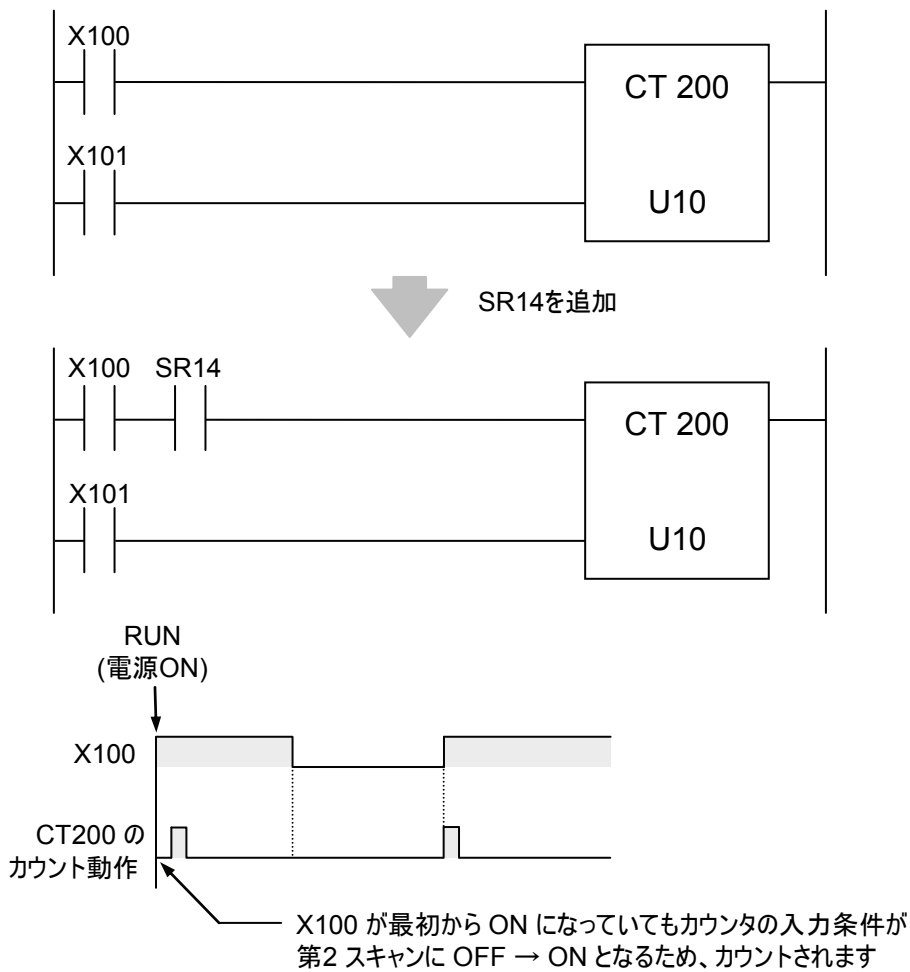


- RUN モードに切り替える前に既に ON になっている実行条件で命令を実行したい場合は、特殊内部リレー SR14 を用いて、下記のようなプログラムを組んでください。(SR14 は、第 1 スキャン時 OFF、第 2 スキャン以降に ON になる特殊内部リレーです)

<例 1> 立ち上がり微分命令 DF の場合



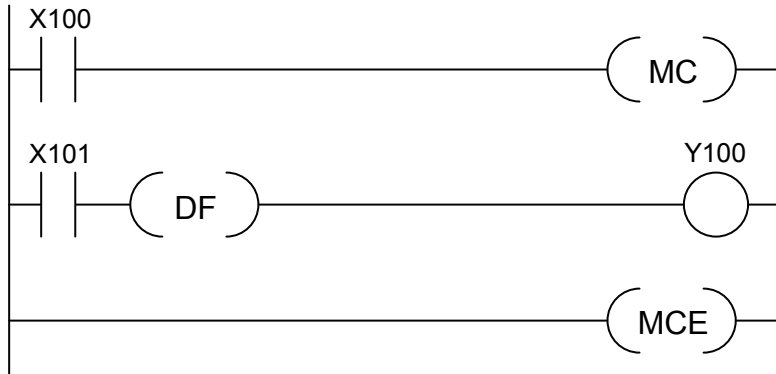
<例 2>カウンタ命令 CT の場合



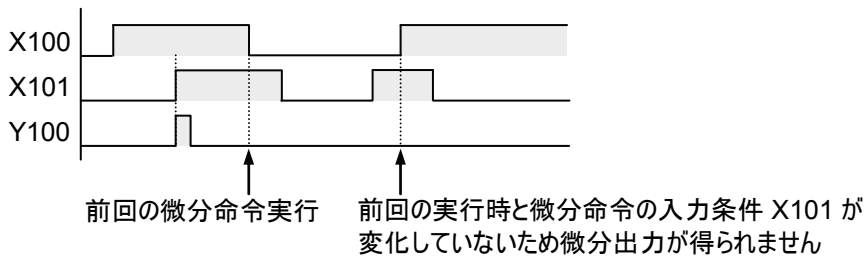
## ■ 制御命令を使用している場合の注意点

- 立ち上がり検出を行う命令は、前回その命令が実行された時の実行条件と今回の実行条件を比較して、OFF から ON になった場合にのみ実行されます。それ以外の場合は実行されません。
- 従って、MC・MCE、JP・LBL など命令を実行する順序を変える命令と合わせて、立ち上がり検出を使用する場合は、入力のタイミングにより、命令の動作が下記のように変わりますのでご注意ください。

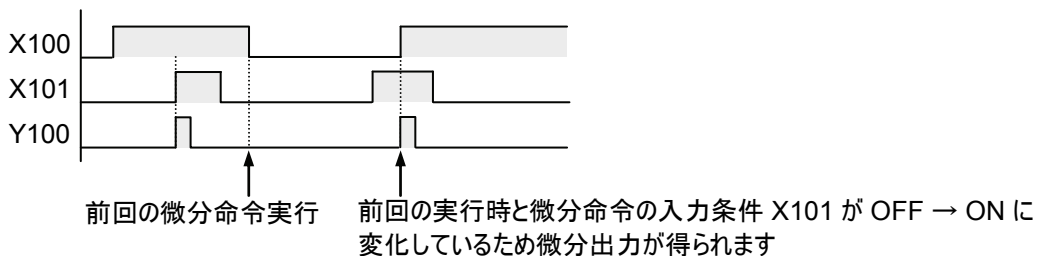
<例 1> MC～MCE 間に微分命令 DF を使用した場合



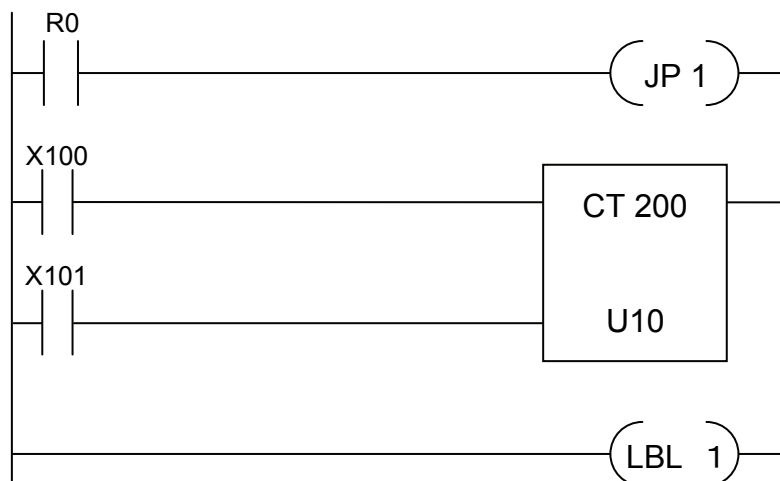
タイミングチャート 1



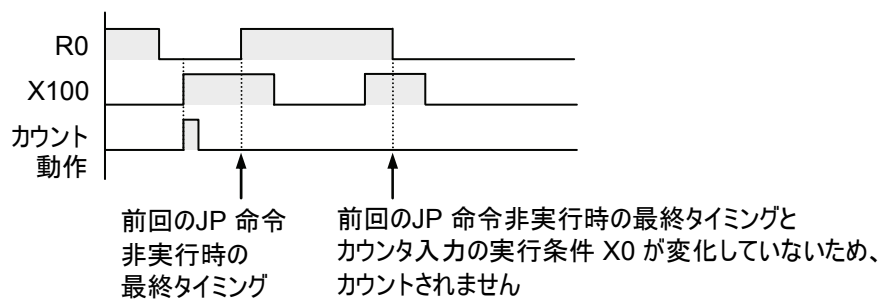
タイミングチャート 2



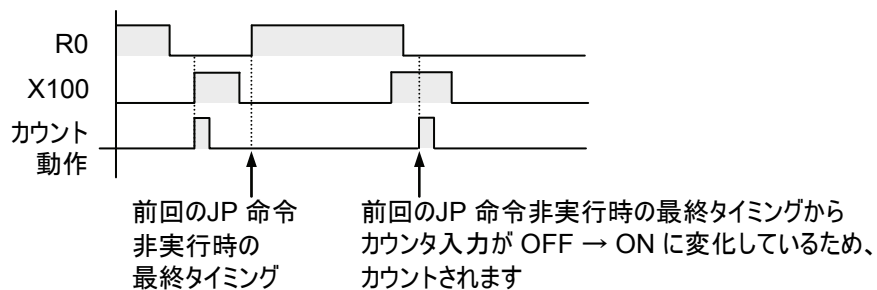
<例 2> JP～LBL 間にカウンタ命令を使用した場合



タイミングチャート 1



タイミングチャート 2

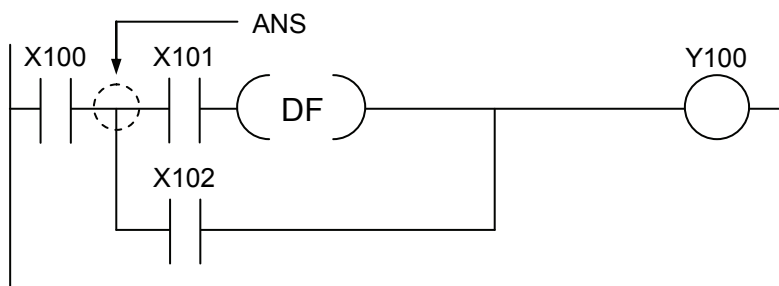


## 4-8 プログラム記述上のご注意

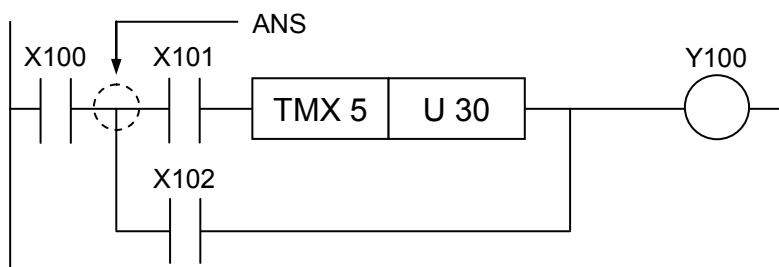
### ■ 正しく実行されないプログラム

• 下記のようなプログラムは正しく実行されませんので、記述しないでください。

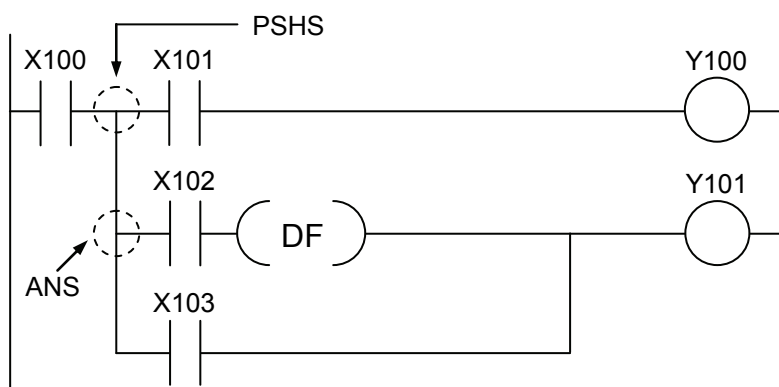
<例 1> X101 が先に ON している、X100 を ON しても Y100 は ON しません。



<例 2> X100 の ON/OFF に関係なく、X101 を ON すると TMX5 が働いてしまう。



<例 3> X102 が先に ON している、X100 を ON しても Y101 は ON しません。

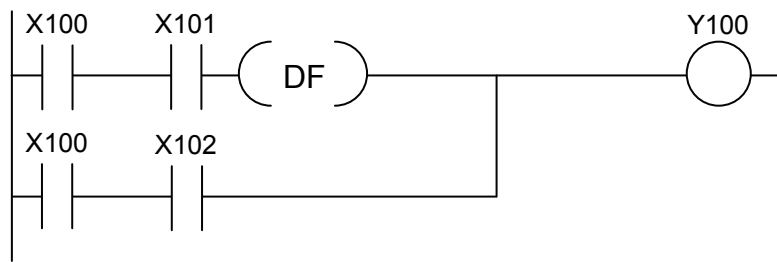


• 微分命令やタイマ命令の実行条件を、複数の接点を組み合わせて設定する場合は、ANS 命令や RDS 命令、POPS 命令を使わないでください。

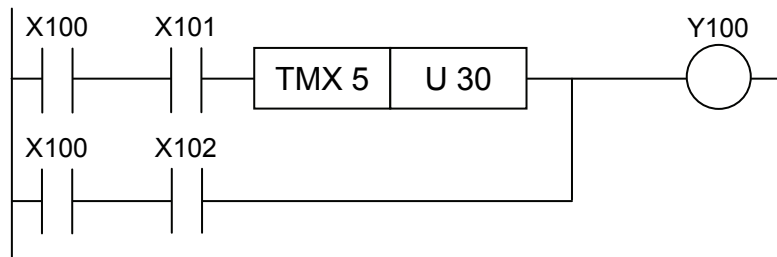
## ■ プログラム書き換え例

• 前述のプログラムを正しく書き換えた例です。

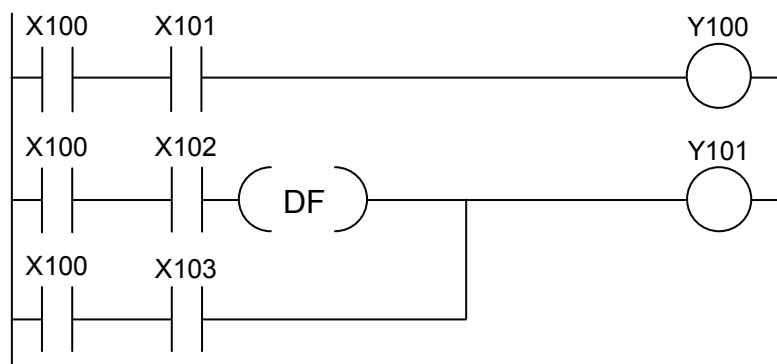
< 例 1 を書き換えたプログラム >



< 例 2 を書き換えたプログラム >



< 例 3 を書き換えたプログラム >







# 5

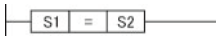
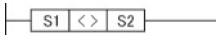
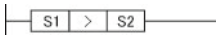
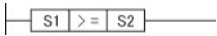
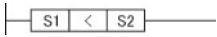
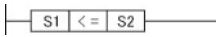
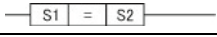
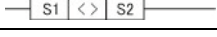
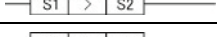
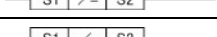
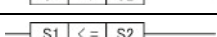
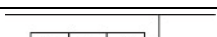
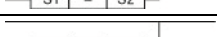
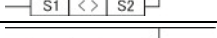
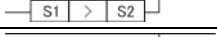
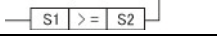
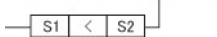
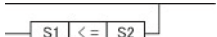
## 命令語一覽

# 5-1 基本命令一覧

名称	ニモニック	シンボル	機能概要	掲載ページ
<b>シーケンス基本命令</b>				
スタート	ST		常開接点で論理演算を開始。	2-2
スタートノット	ST/		常閉接点で論理演算を開始。	2-2
アウト	OT		演算結果を出力。	2-2
アンド	AN		常開接点を直列接続。	2-2
アンドノット	AN/		常閉接点を直列接続。	2-2
オア	OR		常開接点を並列接続。	2-2
オアノット	OR/		常閉接点を並列接続。	2-2
立ち上がり接点命令	ST↑		信号の立ち上がりを検出した 1 スキャンのみ ON する接点で論理演算を開始。	2-5
立ち下がり接点命令	ST↓		信号の立ち下がり検出した 1 スキャンのみ ON する接点で論理演算を開始。	2-5
立ち上がり接点命令	AN↑		信号の立ち上がりを検出した 1 スキャンのみ ON する接点を直列接続。	2-5
立ち下がり接点命令	AN↓		信号の立ち下がり検出した 1 スキャンのみ ON する接点を直列接続。	2-5
立ち上がり接点命令	OR↑		信号の立ち上がりを検出した 1 スキャンのみ ON する接点を並列接続。	2-5
立ち下がり接点命令	OR↓		信号の立ち下がり検出した 1 スキャンのみ ON する接点を並列接続。	2-5
ノット	/		直前までの演算結果を反転。	2-7
立ち上がり微分	DF		信号の立ち上がりを検出した 1 スキャンのみ接点を ON。	2-8
立ち下がり微分	DF/		信号の立ち下がり検出した 1 スキャンのみ接点を ON。	2-8
立ち上がり微分 (イニシャル実行型)	DFI		信号の立ち上がりを検出した 1 スキャンのみ接点を ON。第 1 スキャンでの立ち上がり検出が可能。	2-8
アンドスタック	ANS		複数のブロックを直列接続。	2-14
オアスタック	ORS		複数のブロックを並列接続。	2-15
プッシュスタック	PUSHS		直前までの演算結果を記憶。	2-16
リードスタック	RDS		PSHS で記憶した演算結果を読み出し。	2-16
ポップスタック	POPS		PSHS で記憶した演算結果を読み出し、読み出して、記憶をクリア。	2-16
NOP	NOP		無処理。	2-19
立ち上がり検出アウト	↑OT		信号の立ち上がりを検出した 1 スキャンのみ出力(パルススレーP 用)。	2-20
立ち下がり検出アウト	↓OT		信号の立ち下がり検出した 1 スキャンのみ出力(パルススレーP 用)。	2-20

名称	ニモニック	シンボル	機能概要	掲載ページ
キープ	KP		セット入力で ON し、リセットするまで保持。	2-21
セット	SET		出力を ON にして、その状態を保持。	2-22
リセット	RST		出力を OFF にして、その状態を保持。	2-22
オルタネートアウト	ALT		信号の立ち上がりを検出するたびに ON/OFF が反転する出力。	2-24
<b>基本機能命令</b>				
タイマ	TM		オンディレータイマ。設定時間 S で減算動作し、経過値 0 でタイマ接点を ON。S は 32 ビットデータ(U1～U4294967295)で指定。 TMS:0.1ms 単位 TML:1ms 単位 TMR:0.01s 単位 TMX:0.1s 単位 TMY:1s 単位	2-25
符号なし 32 ビット加算式補助タイマ	SPTM		0.01s 単位のオンディレータイマとして動作	2-33
ダウンカウンタ	CT		プリセットした設定値 S から減算カウント。経過値 0 でカウンタ接点を ON。S は 32 ビットデータ(U1～U4294967295)で指定。	2-37
アップダウンカウンタ	UDC		アップダウン入力に応じて、プリセットした設定値 S から加算または減算カウント。比較命令を直後に記述して使用。	2-44
シフトレジスタ	SR		D で指定したワードデバイスの内容を左に 1 ビットシフト。	2-47
左右シフトレジスタ	LRSR		D1～D2 で指定したエリアを左または右に 1 ビットシフト。	2-51
マスタコントロールリレー	MC		実行条件が OFF のとき、MC～MCE 間の出力を OFF。実行条件 ON のとき、MC～MCE 間のプログラムを実行	2-54
マスタコントロールリレーエンド	MCE		2-54	
ジャンプ	JP		実行条件が ON のとき、同じ番号の LBL 命令にジャンプし、プログラムを続行。	2-59
ラベル	LBL			2-59
ループ	LOOP		実行条件が ON のとき、同じ番号 LBL 命令にジャンプして、プログラムを続行。演算を繰り返す回数を S に設定。	2-64
ラベル	LBL		2-64	
エンド	ED		プログラムの演算を終了。主プログラムの終わりを示す。	2-69
エンドプログラムブロック	EDPB		PB(プログラム・ブロック)の終了。	2-70
条件付きエンド	CNDE		実行条件が ON のとき、プログラムの演算を終了。	2-71
エジェクト	EJECT		プリントアウト時に改ページ。	2-72

名称	ニモニック	シンボル	機能概要	掲載ページ
<b>ステップラダー命令</b>				
スタートステップ	SSTP		工程として制御するプログラム n の先頭。	2-76
ネクストステップ	NSTL		指定の工程 n を起動し、起動中の工程をクリア。(毎スキャン実行型)	2-76
クリアステップ	CSTP		起動中の工程 n をクリア。	2-76
ステップエンド	STPE		ステップラダー領域の終端。	2-76
ブロッククリア	ZRST		起動中の工程 n1～n2 をクリア。	2-82
<b>サブルーチン命令</b>				
サブルーチンコール	CALL		実行条件 ON 時:サブルーチンの実行 実行条件 OFF 時:サブルーチン非実行 サブルーチン内での出力は保持。	2-73
サブルーチンラベル	SBL		CALL 命令実行時に、同じ番号のラベルを持つサブルーチンラベルにジャンプし、サブルーチンを実行。RET 命令で主プログラムのアドレスに戻り、プログラムを実行。	2-84
サブルーチンリターン	RET			2-84
出力 OFF 型 ローカル サブルーチンコール	FCAL		実行条件 ON 時:サブルーチンの実行 実行条件 OFF 時:サブルーチン非実行 但し、サブルーチン内での出力はクリアします。	2-88
サブルーチンコール (PB 番号指定)	ECAL		プログラムブロック PBn 中のサブルーチンに対して、 実行条件 ON 時:サブルーチンへジャンプ 実行条件 OFF 時:サブルーチン非実行	2-90
出力 OFF 型 サブルーチンコール (PB 番号指定)	EFCAL		プログラムブロック PBn 中のサブルーチンに対して、 実行条件 ON 時:サブルーチンへジャンプ 実行条件 OFF 時:サブルーチン非実行 但し、サブルーチン内での出力はクリアします。	2-92
<b>コメント化命令</b>				
コメントアウト	COMOUT		「COMOUT」命令～「ENDCOM」命令の間の命令をコメントアウト。	2-94
コメントアウト 終了	ENDCOM			2-94

名称	演算単位	ニモニック	シンボル	機能概要	掲載ページ
<b>データ比較命令</b>					
データ比較 (スタート)	US, SS, UL, SL, SF, DF	ST=		S1=S2 のとき導通する接点で論理演算を開始。	2-95
		ST<>		S1≠S2 のとき導通する接点で論理演算を開始。	2-95
		ST>		S1>S2 のとき導通する接点で論理演算を開始。	2-95
		ST>=		S1≥S2 のとき導通する接点で論理演算を開始。	2-95
		ST<		S1<S2 のとき導通する接点で論理演算を開始。	2-95
		ST<=		S1≤S2 のとき導通する接点で論理演算を開始。	2-95
データ比較 (アンド)	US, SS, UL, SL, SF, DF	AN=		S1=S2 のとき導通する接点を直列接続。	2-95
		AN<>		S1≠S2 のとき導通する接点を直列接続。	2-95
		AN>		S1>S2 のとき導通する接点を直列接続。	2-95
		AN>=		S1≥S2 のとき導通する接点を直列接続。	2-95
		AN<		S1<S2 のとき導通する接点を直列接続。	2-95
		AN<=		S1≤S2 のとき導通する接点を直列接続。	2-95
データ比較 (オア)	US, SS, UL, SL, SF, DF	OR=		S1=S2 のとき導通する接点を並列接続。	2-95
		OR<>		S1≠S2 のとき導通する接点を並列接続。	2-95
		OR>		S1>S2 のとき導通する接点を並列接続。	2-95
		OR>=		S1≥S2 のとき導通する接点を並列接続。	2-95
		OR<		S1<S2 のとき導通する接点を並列接続。	2-95
		OR<=		S1≤S2 のとき導通する接点を並列接続。	2-95

## 5-2 応用命令一覧

名称	演算 単位	ニモニック	オペランド	機能概要	実行条件		掲載 ページ	
					レベル	↑		
<b>データ比較命令</b>								
データ比較	US, SS, UL, SL, SF, DF	CMP	(P)	S1, S2	S1とS2を比較し、判定結果をシステムリレー(SRA~SRC)に出力 (S1) > (S2) → SRA:ON (S1) = (S2) → SRB:ON (S1) < (S2) → SRC:ON	●	●	3-2
帯域比較	US, SS, UL, SL, SF, DF	WIN	(P)	S1, S2,S3	S1の値を下限値S2、上限値S3と比較し、判定結果をシステムリレー(SRA~SRC)に出力 (S1) > (S3) → SRA:ON (S2) ≤ (S1) ≤ (S3) → SRB:ON (S1) < (S2) → SRC:ON	●	●	3-2
<b>データ転送命令</b>								
データ転送	US, SS, UL, SL, SF, DF	MV	(P)	S, D	(S) → (D)	●	●	3-8
データ反転転送	US, SS, UL, SL, SF, DF	MV/	(P)	S, D	$\overline{(S)}$ → (D)	●	●	3-10
ブロック転送	US, SS, UL, SL, SF, DF	BKMV	(P)	S1, S2, D	S1~S2間のデータを、Dを先頭とするエリアに転送。	●	●	3-12
ブロック複写	US, SS, UL, SL, SF, DF	COPY	(P)	S, D1, D2	Sのデータを、D1~D2間のすべてのエリアに転送。	●	●	3-14
リセット	US, SS, UL, SL, SF, DF	RST	(P)	D	Dのエリアを0にリセット。	●	●	3-16
ビットブロック転送	US, SS, UL, SL, SF, DF	BTM	(P)	S1, S2, D	ビットアドレスS1~S2間のデータを、Dで指定したビットを下位とするエリアに転送。	●	●	3-18
ブロッククリア	bit	ZRST	(P)	D1, D2	ビットアドレスD1~D2で指定したエリアを0にリセット。	●	●	3-20
デジットデータ転送	-	DGT	(P)	S, S1, n, D, D1	S, S1で指定したデジットから、nデジットのデータを、D, D1で指定したデジットから格納。	●	●	3-22
データ交換	US, SS, UL, SL, SF, DF	XCH	(P)	D1, D2	(D1)→(D2), (D2)→(D1)	●	●	3-25
上位バイト 下位バイト交換	US, SS	SWAP	(P)	S, D	Sのデータの上位バイトと下位バイトを交換し、Dに格納。	●	●	3-27
<b>インデックスレジスタ操作命令</b>								
インデックス レジスタ退避	UL, SL	PUSHIX	(P)	D	(I0)~(IE) → (D)~(D+29)	●	●	3-28
インデックス レジスタ復帰	UL, SL	POPIX	(P)	S	(S)~(S+29) → (I0)~(IE)	●	●	3-29

名称	演算単位	ニモニック	オペランド	機能概要	実行条件		掲載ページ	
					レベル	↑		
<b>四則演算命令</b>								
加算	US, SS, UL, SL, SF, DF	ADD	(P)	S1, S2, D	$(S1) + (S2) \rightarrow (D)$	●	●	3-30
減算	US, SS, UL, SL, SF, DF	SUB	(P)	S1, S2, D	$(S1) - (S2) \rightarrow (D)$	●	●	3-32
乗算	US, SS, UL, SL, SF, DF	MUL	(P)	S1, S2, D	$(S1) \times (S2) \rightarrow (D, D+n)$ n は演算単位によって変動します。	●	●	3-34
除算	US, SS, UL, SL, SF, DF	DIV	(P)	S1, S2, D	$(S1) \div (S2) \rightarrow \text{商}(D)$	●	●	3-36
除算	US, SS, UL, SL	DIVMOD	(P)	S1, S2, D	$(S1) \div (S2) \rightarrow \text{商}(D), \text{余り}(D+n)$ n は演算単位によって変動します。	●	●	3-38
インクリメント	US, SS, UL, SL, SF, DF	INC	(P)	D	$(D) + 1 \rightarrow (D)$	●	●	3-40
デクリメント	US, SS, UL, SL, SF, DF	DEC	(P)	D	$(D) - 1 \rightarrow (D)$	●	●	3-41
<b>BCD データ算術演算命令</b>								
BCD データ加算	US, UL	BCDAD D	(P)	S1, S2, D	$(S1) + (S2) \rightarrow (D)$	●	●	3-42
BCD データ減算	US, UL	BCDSUB	(P)	S1, S2, D	$(S1) - (S2) \rightarrow (D)$	●	●	3-44
BCD データ乗算	US, UL	BCDMU L	(P)	S1, S2, D	$(S1) \times (S2) \rightarrow (D, D+n)$ n は演算単位によって変動します。	●	●	3-46
BCD データ除算	US, UL	BCDDIV	(P)	S1, S2, D	$(S1) \div (S2) \rightarrow \text{商}(D), \text{余り}(D+n)$ n は演算単位によって変動します。	●	●	3-48
BCD データインクリメント	US, UL	BCDINC	(P)	D	$(D) + 1 \rightarrow (D)$	●	●	3-50
BCD データデクリメント	US, UL	BCDDEC	(P)	D	$(D) - 1 \rightarrow (D)$	●	●	3-52
<b>論理演算命令</b>								
論理積	US, SS, UL, SL	AND	(P)	S1, S2, D	$(S1) \wedge (S2) \rightarrow (D)$	●	●	3-54
論理和	US, SS, UL, SL	OR	(P)	S1, S2, D	$(S1) \vee (S2) \rightarrow (D)$	●	●	3-56
排他的論理和	US, SS, UL, SL	XOR	(P)	S1, S2, D	$\{(S1) \wedge \overline{(S2)} \vee \overline{(S1)} \wedge (S2)\} \rightarrow (D)$	●	●	3-58
排他的論理和否定	US, SS, UL, SL	XNR	(P)	S1, S2, D	$\{(S1) \wedge (S2) \vee \overline{(S1)} \wedge \overline{(S2)}\} \rightarrow (D)$	●	●	3-60
結合	US, SS, UL, SL	COMB	(P)	S1, S2, S3, D	$\{(S1) \wedge (S3)\} \vee \{(S2) \wedge (S3)\} \rightarrow (D)$	●	●	3-62

名称	演算 単位	ニモニック		オペランド	機能概要	実行条件		掲載 ページ
						レベル	↑	
<b>文字列変換命令</b>								
ブロックチェック コード計算	US, SS	BCC	(P)	S1, S2,, S3, D	S2とS3で指定されたデータのブロック チェックコードを計算し、Dを先頭と するエリアに格納。計算方法はS1で 指定。	●	●	3-64
CRC コード 計算	US, SS	CRC	(P)	S1, S2, S3, D	S2とS3で指定されたデータのCRC コードを計算し、Dを先頭とするエリア に格納。計算方法はS1で指定。	●	●	3-67
HEX→16進 ASCII変換	US, SS, UL, SL	HEXA	(P)	S1, S2, D	S1とS2で指定された16進のデータ をASCIIコードに変換して、Dを先頭 とするエリアに格納。 例)HABCD→H <u>42 41 44 43</u> B A D C	●	●	3-70
16進ASCII →HEX変換	US, SS, UL, SL	AHEX	(P)	S1, S2, D	S1とS2で指定されたASCIIコード を16進のデータに変換して、Dを先 頭とするエリアに格納。 例)H <u>44 43 42 41</u> →HCDAB D C B A	●	●	3-72
BCD→10進 ASCII変換	US, UL	BCDA	(P)	S1, S2, D	S1とS2で指定されたBCDデータを ASCIIコードに変換して、Dを先頭と するエリアに格納。変換方法はS2で 指定。 例1)H1234→H <u>32 31 34 33</u> 2 1 4 3 例2)H1234→H <u>34 33 32 31</u> 4 3 2 1	●	●	3-75
10進ASCII →BCD変換	US, UL	ABCD	(P)	S1, S2, D	S1とS2で指定されたASCIIコード をBCDデータに変換して、Dを先頭 とするエリアに格納。変換方法はS2 で指定。 例1)H <u>34 33 32 31</u> →H 3412 4 3 2 1 例2)H <u>34 33 32 31</u> →H 1234 4 3 2 1	●	●	3-78
BIN→10進 ASCII変換	US, SS, UL, SL	BINA	(P)	S1, S2, D	S1で指定された10進数を表すBIN データをASCIIコードに変換して、D を先頭とするエリアに格納。変換結果 のバイト数はS2で指定。 例)K-100→H <u>30 30 31 2D 20 20</u> 0 0 1 -	●	●	3-82
10進ASCII →BIN変換	US, SS, UL, SL	ABIN	(P)	S1, S2, D	S1を先頭とするASCIIコードを、10 進数を表すBINデータに変換して、 Dを先頭とするエリアに格納。変換す るのバイト数はS2で指定。 例)H <u>30 30 31 2D 20 20</u> → K-100 0 0 1 -	●	●	3-84
BIN →ASCII変換	US, SS, UL, SL, SF, DF	BTOA	(P)	S1, S2, N, D	S2を先頭とするエリアに格納されて いるBINデータをASCIIデータに変 換し、Dを先頭とするエリアに格納。 変換方法はS1とNで指定。	●	●	3-86
ASCII →BIN変換	US, SS, UL, SL, SF, DF	ATOB	(P)	S1, S2, N, D	S2を先頭とするエリアに格納されて いるASCIIデータをBINデータに変 換し、Dを先頭とするエリアに格納。 変換方法はS1とNで指定。	●	●	3-98
ASCIIデータ チェック	US, SS, UL, SL, SF, DF	ACHK	(P)	S1, S2, N	ATOB命令で変換するASCIIデータ が正常に変換できるかチェック。	●	●	3-110



名称	演算 単位	ニモニック	オペランド	機能概要	実行条件		掲載 ページ
					レベル	↑	
<b>データ変換命令</b>							
データ反転	US, SS, UL, SL	INV	(P) D	Dのデータを各ビットを論理反転し、Dに格納。	●	●	3-112
符号反転	SS, SL	NEG	(P) S, D	Sのデータを符号反転し、Dに格納。	●	●	3-113
絶対値	US, SS, UL, SL	ABS	(P) S, D	Sのデータの絶対値を求め、Dに格納。	●	●	3-114
符号拡張	US, SS	EXT	(P) S, D	Sの16ビットBINデータの符号を維持したまま、32ビットBINデータに変換し、(D+1,D)に格納。	●	●	3-115
BCD データ変換	US, UL	BCD	(P) S, D	Sで指定されたBINデータをBCDデータに変換してDを先頭とするエリアに格納。 例) K100→H100	●	●	3-116
BCD→BIN 変換	US, UL	BIN	(P) S, D	Sで指定されたBCDデータをBINデータに変換してDを先頭とするエリアに格納。 例) H100→K100	●	●	3-117
デコード	-	DECO	(P) S, n, D	Sのデータの一部をデコードし、Dを先頭とするエリアに格納。変換する対象はnで指定。	●	●	3-118
7 セグメント デコード	US	SEGT	(P) S, D	Sのデータを7セグメント表示用に変換して、(D+1,D)に格納。	●	●	3-120
エンコード	-	ENCO	(P) S, n, D	Sのデータの一部をエンコードし、Dを先頭とするエリアに格納。変換する対象はnで指定。	●	●	3-122
デジット結合	US	UNIT	(P) S, n, D	Sを先頭とするnワード分の16ビットデータの最下位デジットを結合し、Dに格納。	●	●	3-124
デジット分離	US	DIST	(P) S, n, D	Sのデータの各デジットを分離し、Dを先頭とする16ビットデータの各最下位デジットに格納。	●	●	3-125
バイトデータ 結合	US, UL	BUNI	(P) S, n, D	Sを先頭とするnワード分の16ビットデータの下位バイトを結合し、Dを先頭とするエリアに格納。	●	●	3-126
バイトデータ 分離	US, UL	BDIS	(P) S, n, D	Sを先頭とするエリアのデータをバイト単位で分離し、Dを先頭とする16ビットデータの下位バイトに格納。	●	●	3-128
バイナリ→ グレイコード変換	US, UL	GRY	(P) S, D	Sに格納されているBINデータをグレイコードデータに変換し、Dを先頭とするエリアに格納。	●	●	3-130
グレイコード →BIN 変換	US, UL	GBIN	(P) S, D	Sに格納されているグレイコードデータをBINデータに変換し、Dを先頭とするエリアに格納。	●	●	3-132
ビット行 →ビット列変換	US	COLM	(P) S, n, D	Sのビット0~15の値をD~D+15のビットnに格納。	●	●	3-134
ビット列→ビット行 変換	US	LINE	(P) S, n, D	S~S+15のビットnの値をDのビット0~15に格納。	●	●	3-136

名称	演算 単位	ニモニック	オペランド	機能概要	実行条件		掲載 ページ	
					レベル	↑		
<b>データシフト命令</b>								
nビット右シフト	US, SS, UL, SL	SHR	(P)	D, n	Dのデータをnビット分、右へシフト。	●	●	3-138
nビット左シフト	US, SS, UL, SL	SHL	(P)	D, n	Dのデータをnビット分、左へシフト。	●	●	3-140
n桁右シフト	US, SS, UL, SL	BSR	(P)	D, n	Dのデータをnデジット分、右へシフト。	●	●	3-142
n桁左シフト	US, SS, UL, SL	BSL	(P)	D, n	Dのデータをnデジット分、左へシフト。	●	●	3-144
nビット分の一括右シフト	bit	BITR	(P)	D1, D2, n	D1~D2のエリアをnビット分、右へシフト。	●	●	3-146
nビット分の一括左シフト	bit	BITL	(P)	D1, D2, n	D1~D2のエリアをnビット分、左へシフト。	●	●	3-148
ブロック領域のnワードの右シフト	US, SS	WSHR	(P)	D1, D2, n	D1~D2のエリアをnワード分、右へシフト。	●	●	3-150
ブロック領域のnワードの左シフト	US, SS	WSHL	(P)	D1, D2, n	D1~D2のエリアをnワード分、左へシフト。	●	●	3-152
<b>データ回転命令</b>								
データの右回転	US, UL	ROR	(P)	D, n	Dのデータをnビット分、右へ回転。	●	●	3-154
データの左回転	US, UL	ROL	(P)	D, n	Dのデータをnビット分、左へ回転。	●	●	3-156
データのキャリー込み右回転	US, UL	RCR	(P)	D, n	DにCYフラグ(SR9)を加えたエリアをnビット分、右へ回転。	●	●	3-158
データのキャリー込み左回転	US, UL	RCL	(P)	D, n	DにCYフラグ(SR9)を加えたエリアをnビット分、左へ回転。	●	●	3-160
<b>データバッファ命令</b>								
圧縮シフト読み出し	US, SS	CMPR	(P)	D1, D2, D3	D2をD3に転送。D1~D2間でデータが0の部分の圧縮し、D2方向に順にシフト。	●	●	3-162
圧縮シフト書き込み	US, SS	CMPW	(P)	S, D1, D2	SをD1に転送。D1~D2間でデータが0の部分の圧縮し、D2方向に順にシフト。	●	●	3-164
バッファ定義	US, SS	DEFBUF	(P)	n, D	Dを先頭とするnワード分のエリアをFIFR/BUFV/LIFR命令で使用するデータバッファエリアに定義。	●	●	3-166
データ読み出し(先入・先出)	US, SS	FIFR	(P)	S, D	Sを先頭とするFIFOバッファの読み出しポイントが示すエリアから、データを読み出して、Dに格納。	●	●	3-168
データ書き込み	US, SS	BUFV	(P)	S, D	Sの値を、Dを先頭とするバッファの書き込みポイントが示すエリアに格納。	●	●	3-170
データ読み出し(後入・先出)	US, SS	LIFR	(P)	S, D	Sを先頭とするLIFOバッファのLIFOポイントが示すエリアから、データを読み出して、Dに格納。	●	●	3-172

名称	演算単位	ニモニック	オペラント	機能概要	実行条件		掲載ページ	
					レベル	↑		
<b>ビット操作命令</b>								
ビット反転	US	BTI	(P)	D, n	D のデータのビット n の値を反転。	●	●	3-174
ビットテスト	US	BTT	(P)	D, n	D のデータのビット n の値が 0 のとき、システムリレー (SRB) を ON、1 のとき、(SRB) を OFF。	●	●	3-175
キャリーのセット	-	STC	(P)		CY フラグ (SR9) を ON	●	●	3-177
キャリーのリセット	-	CLC	(P)		CY フラグ (SR9) を OFF	●	●	3-178
<b>文字列操作命令</b>								
文字定数→ASCIIコード変換	-	SSET	(P)	S, D	S で指定した文字定数を ASCII コードに変換し、D を先頭とするエリアに格納。	●	●	3-179
文字列比較	-	SCMP	(P)	S1, S2	S1 と S2 の文字列を比較し、判定結果をシステムリレー (SRA～SRC) に出力。 (S1) > (S2) → SRA:ON (S1) = (S2) → SRB:ON (S1) < (S2) → SRC:ON	●	●	3-182
文字列加算	-	SADD	(P)	S1, S2, D	S1 の文字列に S2 の文字列を連結し、D に格納。	●	●	3-184
文字列長取得	-	LEN	(P)	S, D	S を先頭とする文字列テーブルのデータに格納されている文字数を D に格納。	●	●	3-185
文字列検索	-	SSRC	(P)	S1, S2, D	S2 を先頭とする文字列テーブルのデータのなかから、データ S1 を検索し、最初に一致した相対位置を D に格納。	●	●	3-186
文字列右側取り出し	-	RIGHT	(P)	S1, S2, D	S1 を先頭とする文字列テーブルの右側 (デバイスアドレスが大きい方) から、S2 文字数分の文字列を取り出し、D に格納。	●	●	3-188
文字列左側取り出し	-	LEFT	(P)	S1, S2, D	S1 を先頭とする文字列テーブルの左側 (デバイスアドレスが小さい方) から、S2 文字数分の文字列を取り出し、D に格納。	●	●	3-190
文字列任意位置からの読み出し	-	MIDR	(P)	S1, S2, S3, D	S1 を先頭とする文字列テーブルの S2 バイト目から S3 文字列分の文字列を読み出し D に格納	●	●	3-192
文字列任意位置からの書き換え	-	MIDW	(P)	S1, S2, D, n	S1 を先頭とする文字列テーブルの S2 文字数分の文字列データを読み出し、D の n バイト目の位置から格納。	●	●	3-194
文字列置き換え	-	SREP	(P)	S, D, p, n	D を先頭とする文字列テーブルを S1 の文字列で置き換え。置き換える位置と文字数は p と n で指定。	●	●	3-197

名称	演算単位	ニモニック		オペランド	機能概要	実行条件		掲載ページ
						レベル	↑	
<b>データ処理命令</b>								
データサーチ	US, SS, UL, SL, SF, DF	SRC	(P)	S1, S2, S3, D	S2～S3 に格納されているデータから、S1 のデータを検索し、一致個数と最初にデータが一致した相対位置を D を先頭とするエリアに格納。	●	●	3-200
ON ビットカウント	US, UL	BCU	(P)	S, D	S のデータについて、ON のビット数をカウントし、個数を D に格納。	●	●	3-203
最大値取得	US, SS, UL, SL, SF, DF	MAX	(P)	S1, S2, D	S1～S2 に格納されているデータから、最大値を検索し、最大値とそれを最初に検出した相対位置を D を先頭とするエリアに格納。	●	●	3-205
最小値取得	US, SS, UL, SL, SF, DF	MIN	(P)	S1, S2, D	S1～S2 に格納されているデータから、最小値を検索し、最小値とそれを最初に検出した相対位置を D を先頭とするエリアに格納。	●	●	3-209
合計・平均値取得	US, SS, UL, SL, SF, DF	MEAN	(P)	S1, S2, D	S1～S2 に格納されているデータの合計値および平均値を求め、D を先頭とするエリアに格納。	●	●	3-214
並び替え	US, SS, UL, SL, SF, DF	SORT	(P)	S1, S2, S3	S1～S2 に格納されているデータを並び替え。S3 にはソート条件(昇順または降順)を指定。	●	●	3-218
線形化	US, SS, UL, SL, SF, DF	SCAL	(P)	S1, S2, D	S2 を先頭とするエリアにスケーリング(線形化)に使用するデータテーブルを作成。S1 (入力値 X) のデータをスケーリングし、D (出力値 Y) に格納。	●	●	3-221
<b>事象計数命令</b>								
事象発生回数計数命令	UL	EVENTC		S1, S2, n, D	S1 で指定された n ビット分のデータの ON 回数を計数し、D を先頭とするに格納。	●	-	3-224
事象発生時間計数命令	UL	EVENTT		S1, S2, n, D	S1 で指定された n ビット分のデータの ON 時間を秒単位で計時し、D を先頭とするに格納。	●	-	3-227
<b>PID 命令</b>								
PID 加算	-	PID		S	S～S+29 のパラメータに従って PID 演算を実行。演算結果は出力値 MV として、(S+3) に格納。	(注 1)	(注 1)	3-230
PID 演算: PWM 出力可能	-	EZPID		S1, S2, S3, S4	S1, S2, S3～S3+3, S4～S4+29 のパラメータに従って PID 演算を実行。演算結果は出力値 MV として、S4 に格納。直後に OUT 命令を記述し、PWM 出力が可能。	(注 1)	(注 1)	3-235

(注 1):PID 命令および EZPID 命令は、演算内容を保持するため、常時実行条件を ON にする必要があります。

名称	演算単位	ニモニック	オペランド	機能概要	実行条件		掲載ページ	
					レベル	↑		
<b>データ制御命令</b>								
データ変化検出	US, SS, UL, SL, SF, DF	DTR	(P)	S, D	S のデータ変化を検出し、CY フラグ (SR9) に反映。D は前回値データ保持用エリアとして使用。	●	●	3-244
傾斜出力	US, SS, UL, SL, SF, DF	RAMP		S1, S2, S3, D	指定された初期値 S1 から目標値 S2 まで、指定時間 S3 で線形出力。	(注 1)	(注 1)	3-246
上下限リミット制御	US, SS, UL, SL, SF, DF	LIMIT	(P)	S1, S2, S3, D	入力値 S3 が上限値 S1 と下限値 S2 の範囲内か否かにより、出力を求め、D に格納。 (S1) > (S3) のとき、(S1)→(D) (S2) < (S3) のとき、(S2)→(D) (S1) ≤ (S3) ≤ (S2) のとき、(S3)→(D)	●	●	3-248
不感帯制御	SS, SL, SF, DF	BAND	(P)	S1, S2, S3, D	入力値 S3 が S1 と S2 で指定した不感帯の範囲内か否かにより、出力を求め、D に格納。 (S1) > (S3) のとき、(S3)-(S1)→(D) (S2) < (S3) のとき、(S3)-(S2)→(D) (S1) ≤ (S3) ≤ (S2) のとき、0→(D)	●	●	3-250
ゾーン制御	US, SS, UL, SL, SF, DF	ZONE	(P)	S1, S2, S3, D	入力値 S3 の範囲により、S1 または S2 で指定したバイアス値を加え、D に格納。 (S3) < 0 のとき、(S3)+(S1)→(D) (S3) = 0 のとき、0→(D) (S3) > 0 のとき、(S3)+(S2)→(D)	●	●	3-252
<b>時定数処理命令</b>								
時定数処理	-	FILTR		S1, S2, S3, D	S1 と S2 で指定したデータのフィルタ処理を行ない、D に格納。	(注 1)	(注 1)	3-254
<b>浮動小数点形実数関数命令</b>								
正弦演算	SF, DF	SIN	(P)	S, D	SIN(S)→(D)	●	●	3-256
余弦演算	SF, DF	COS	(P)	S, D	COS(S)→(D)	●	●	3-257
正接演算	SF, DF	TAN	(P)	S, D	TAN(S)→(D)	●	●	3-258
逆正弦演算	SF, DF	ASIN	(P)	S, D	SIN <sup>-1</sup> (S)→(D)	●	●	3-259
逆余弦演算	SF, DF	ACOS	(P)	S, D	COS <sup>-1</sup> (S)→(D)	●	●	3-260
逆正接演算	SF, DF	ATAN	(P)	S, D	TAN <sup>-1</sup> (S)→(D)	●	●	3-261
座標データ→角度ラジアン変換	SF, DF	ATAN2	(P)	S1, S2, D	S1(X座標)、S2(Y座標)から逆正接を求め、D に格納。	●	●	3-262
双曲線正弦演算	SF, DF	SINH	(P)	S, D	SINH(S)→(D)	●	●	3-263
双曲線余弦演算	SF, DF	COSH	(P)	S, D	COSH(S)→(D)	●	●	3-264
双曲線正接演算	SF, DF	TANH	(P)	S, D	TANH(S)→(D)	●	●	3-265
指数演算	SF, DF	EXP	(P)	S, D	EXP(S)→(D)	●	●	3-266
自然対数演算	SF, DF	LN	(P)	S, D	LN(S)→(D)	●	●	3-267
常用対数演算	SF, DF	LOG	(P)	S, D	LOG(S)→(D)	●	●	3-268
べき乗演算	SF, DF	PWR	(P)	S1, S2, D	(S1) <sup>(S2)</sup> → (D)	●	●	3-269
平方根演算	SF, DF	SQR	(P)	S, D	SQR(S) → (D)	●	●	3-270

(注 1) : RAMP 命令および FILTER 命令は、演算内容を保持するため、常時実行条件を ON にする必要があります。

名称	演算 単位	ニモニック	オペランド	機能概要	実行条件		掲載 ページ	
					レベル	↑		
<b>浮動小数点形実数変換命令</b>								
角度→ ラジアン変換	SF, DF	RAD	(P)	S, D	Sの角度データ[度]を角度データ[ラジアン]に変換し、Dに格納。	●	●	3-271
ラジアン→ 角度変換	SF, DF	DEG	(P)	S, D	Sの角度データ[ラジアン]を角度データ[度]に変換し、Dに格納。	●	●	3-272
浮動小数点形 実数データ 小数 点以下切り捨て	SF, DF	FINT	(P)	S, D	Sに格納されている実数の小数点以下を切り捨て、Dに格納。	●	●	3-273
浮動小数点形 実数データ 小数点四捨五入	SF, DF	FRINT	(P)	S, D	Sに格納されている実数の小数点第1位を四捨五入し、Dに格納。	●	●	3-274
浮動小数点形 実数データ 符号変換	SF, DF	FNEG	(P)	S, D	Sに格納されている実数を符号変換し、Dに格納。	●	●	3-275
浮動小数点形 実数データ 絶対値	SF, DF	FABS	(P)	S, D	Sに格納されている実数の絶対値を求め、Dに格納。	●	●	3-276
整数→単精度浮 動小数点形実数 データ変換	US, SS, UL, SL	FLT	(P)	S, D	Sに格納されている整数データを、単精度浮動小数点形実数データに変換し、Dに格納。	●	●	3-277
単精度浮動小 数点形実数データ →整数変換 (超えない最大)	US, SS, UL, SL	INT	(P)	S, D	Sに格納されている単精度浮動小数点形実数を、整数(超えない最大)に変換し、Dに格納。	●	●	3-279
単精度浮動小 数点形実数データ →整数変換 (小数点以下切り 捨て)	US, SS, UL, SL	FIX	(P)	S, D	Sに格納されている単精度浮動小数点形実数を、整数(小数点以下切り捨て)に変換し、Dに格納。	●	●	3-282
単精度浮動小 数点形実数データ →整数変換 (小数点以下四捨 五入)	US, SS, UL, SL	ROFF	(P)	S, D	Sに格納されている単精度浮動小数点形実数を、整数(小数点以下四捨五入)に変換し、Dに格納。	●	●	3-285

名称	演算単位	ニモニック	オペランド	機能概要	実行条件		掲載ページ	
					レベル	↑		
<b>カレンダー、時刻命令</b>								
時間(時分秒) →秒データ変換	-	HMSS	(P)	S, D	S~S+2に格納されている時、分、秒を示すBINデータを、秒単位に換算し、(D+1,D)に格納。	●	●	3-288
秒データ→時間 (時分秒)変換	-	SHMS	(P)	S, D	(S+1,S)に格納されている秒データを、時、分、秒に換算してD~D+2にBINデータとして格納。	●	●	3-289
時刻加算	-	CADD	(P)	S1, S2, D	S1~S1+5に格納されている時刻データ(年、月、日、時、分、秒)に、S2~S2+2に格納されている時間データ(時、分、秒)を加算し、D~D+5に格納。	●	●	3-290
時刻減算	-	CSUB	(P)	S1, S2, D	S1~S1+5に格納されている時刻データ(年、月、日、時、分、秒)から、S2~S2+2に格納されている時間データ(時、分、秒)を減算し、D~D+5に格納。	●	●	3-291
時刻データ →基準時刻からの秒データ	-	TMSEC	(P)	S, D	Sで指定された時刻データを基準時刻(2001年1月1日)からの秒データに変換し、(D+1,D)に格納。	●	●	3-292
基準時刻からの秒データ →時刻データ	-	SECTM	(P)	S, D	(S+1,S)に格納されている基準時刻(2001年1月1日)からの秒データを時刻データに変換し、D~D+5に格納。	●	●	3-293
カレンダー設定	-	TIMEWT	(P)	S	S~S+6に格納されている時刻データ(年、月、日、時、分、秒)をCPUユニットのRTCデータとしてセット。	●	●	3-294
<b>特殊命令</b>								
自己診断エラーコードセット	US	ERR	(P)	n	ユーザプログラムにより、任意のエラーコード[n]をセット。自己診断エラーコードをはシステムデータレジスタ(SD0)に格納。自己診断異常発生フラグ(SR0)をセット。	●	●	3-296
ウォッチドッグタイマーリセット	-	WDTRS	(P)		ウォッチドッグタイマーをリセット。	●	●	3-297

名称	演算 単位	ニモニック		オペランド	機能概要	実行条件		掲載 ページ
						レベル	↑	
<b>通信命令</b>								
通信ユニット スロットポート 指定	-	UNITSE L	(P)	S1, S2	以下の通信命令の直前に記述し、各命令を実行する対象を指定。 GPSEND、GPRECV、SEND、 RECV、RDET、PMSET、PMGET	●	●	3-298
汎用通信送信 命令	US, SS	GPSEN D		S, n, D	S を先頭とするエリアに格納された n バイトのデータを、ユニットの COM ポート(SCU)または LAN ポート (ET-LAN) 経由で外部機器へ送信。	(注 1)	(注 1)	3-300
汎用通信受信 命令	US, SS	GPREC V	(P)	D1, D2	ユニットの COM ポート(SCU)または LAN ポート(ET-LAN) 経由で、外部 機器から受信したデータを D1~D2 の範囲にコピー。	●	●	3-306
MEWTOCOL /MODBUS マスタ 送信命令	US, SS	SEND	(P)	S, n, D1, D2, D3	S を先頭とする自局内のデバイスか らデータを読み出し、相手局 D1 の D2 を先頭とするアドレスから格納。 送信データ数は n で指定。	●	●	3-312
MODBUS マスタ 送信命令 :ファンクション コード指定	US, SS	SEND	(P)	S, n, D1 D2, D3	S を先頭とする自局内のデバイスか らデータを読み出し、相手局 D1 の D2 を先頭とするアドレスから格納。 MODBUS ファンクションコードと局番 は D1、送信データ数は n で指定。	●	●	3-318
MEWTOCOL /MODBUS マスタ 受信命令	US, SS	RECV	(P)	S1, S2, n D1, D2	相手先局番 S1 の S2 を先頭とする アドレスからデータを読み出し、D1 を先頭とする自局内のエリアに格納。 送信データ数は n で指定。	●	●	3-324
MODBUS マスタ 受信命令 :ファンクション コード指定	US, SS	RECV	(P)	S1, S2, n D1, D2	相手先局番 S1 の S2 を先頭とする アドレスからデータを読み出し、D1 を先頭とする自局内のエリアに格納。 MODBUS ファンクションコードと局番 は S1、送信データ数は n で指定。	●	●	3-330
SCU のパラメータ 変更	-	PMSET		S, n, D	S から n ワードのエリアに格納されて いるデータを、ユニットの COM ポート (SCU) の通信パラメータにセット。	(注 2)	(注 2)	3-336
SCU のパラメータ 取得	-	PMGET	(P)	S, D	ユニットの COM ポート(SCU) に設定 されている通信パラメータを読み出 し、D を先頭とするエリアに格納。	●	●	3-339
ET-LAN ステータ ス読み出し	-	RDET	(P)	D	ユニットの LAN ポート(ET-LAN) の全 コネクションの状態を読み出し D に格 納。	●	●	3-342

(注 1) : GPSEND 命令は、命令起動後、送信中フラグが OFF になるまでの間、実行条件を ON にする必要があります。

(注 2) : PMSET 命令は、命令起動後、処理中フラグが OFF になるまでの間、実行条件を ON にする必要があります。



名称	演算 単位	ニモニック	オペランド	機能概要	実行条件		掲載 ページ
<b>位置決めユニット制御命令</b>							
位置決め開始 テーブル設定	-	POSSET	(P) S1, S2, S3	位置決めを起動するプログラムの直前に記述し、起動する位置決めデータテーブルをセット。S1、S2、S3にスロット No.、軸 No.、テーブル No.を指定。	●	●	3-344
軸ステータス 取得	-	PSTRD	(P) S1, S2, D	スロット No. (S1) に装着されている位置決めユニットの軸 No. (S2) について、主なフラグの状態を軸ステータスとして読み出し、D に格納。	●	●	3-346
位置決めユニット エラー/ワーニング 取得	-	PERRD	(P) S1, S2, D	スロット No. (S1) に装着されている位置決めユニットの軸 No. (S2) について、報知パッファ 1 に格納されているエラーコード、ワーニングコードを読み出し、それぞれ D と D1 に格納。	●	●	3-348
<b>ユニット制御命令</b>							
エラー/ワーニング クリア	-	UCLR	S	S で指定されたスロット No. に装着されているユニットのエラー/ワーニングをクリア。	●	●	3-349



# 索引

---

/  
/ ..... 2-7

---

↑  
↑ OT ..... 2-20

---

↓  
↓ OT ..... 2-20

---

## A

ABCD ..... 3-78  
ABIN ..... 3-84  
ABS ..... 3-114  
ACHK ..... 3-110  
ACOS ..... 3-260  
ADD ..... 3-30  
AHEX ..... 3-72  
ALT ..... 2-24  
AN ..... 2-2  
AN/ ..... 2-2  
AN< ..... 2-95  
AN<= ..... 2-95  
AN<> ..... 2-95  
AN= ..... 2-95  
AN> ..... 2-95  
AN>= ..... 2-95  
AN↑ ..... 2-5  
AN↓ ..... 2-5  
AND ..... 3-54  
ANS ..... 2-14  
ASIN ..... 3-259  
ATAN ..... 3-261  
ATAN2 ..... 3-262  
ATOB ..... 3-98

---

## B

BAND ..... 3-250  
BCC ..... 3-64  
BCD ..... 3-116  
BCDA ..... 3-75  
BCDADD ..... 3-42  
BCDDEC ..... 3-52  
BCDDIV ..... 3-48  
BCDINC ..... 3-50  
BCDMUL ..... 3-46  
BCDSUB ..... 3-44  
BCU ..... 3-203  
BDIS ..... 3-128

BIN ..... 3-117  
BINA ..... 3-82  
BITL ..... 3-148  
BITR ..... 3-146  
BKMV ..... 3-12  
BSL ..... 3-144  
BSR ..... 3-142  
BTI ..... 3-174  
BTM ..... 3-18  
BTOA ..... 3-86  
BTT ..... 3-175  
BUFW ..... 3-170  
BUNI ..... 3-126

---

## C

C ..... 1-16  
CADD ..... 3-290  
CALL ..... 2-84  
CE ..... 1-27  
CLC ..... 3-178  
CMP ..... 3-2  
CMPR ..... 3-162  
CMPW ..... 3-164  
CNDE ..... 2-71  
COLM ..... 3-134  
COMB ..... 3-62  
COMOUT ..... 2-94  
COPY ..... 3-14  
COS ..... 3-257  
COSH ..... 3-264  
CRC ..... 3-67  
CS ..... 1-26  
CSTP ..... 2-73  
CSUB ..... 3-291  
CT ..... 2-37

---

## D

DEC ..... 3-41  
DECO ..... 3-118  
DEFBUF ..... 3-166  
DEG ..... 3-272  
DF ..... 1-35, 2-8  
DF/ ..... 2-8  
DFI ..... 2-8  
DGT ..... 3-22  
DIST ..... 3-125  
DIV ..... 3-36  
DIVMOD ..... 3-38  
DT ..... 1-22  
DTR ..... 3-244

---

## E

E ..... 1-19  
ECAL ..... 2-90  
ED ..... 2-69  
EDPB ..... 2-70

---

EFCAL .....	2-92
EJECT .....	2-72
ENCO .....	3-122
ENDCOM.....	2-94
ERR .....	3-296
EVENTC .....	3-224
EVENTT.....	3-227
EXP .....	3-266
EXT .....	3-115
EZPID .....	3-235

---

## **F**

FABS .....	3-276
FCAL .....	2-88
FIFR.....	3-168
FILTR.....	3-254
FINT.....	3-273
FIX .....	3-282
FLT .....	3-277
FNEG.....	3-275
FRINT .....	3-274

---

## **G**

GBIN.....	3-132
GPRECV.....	3-306
GPSEND.....	3-300
GRY .....	3-130

---

## **H**

H .....	1-33
HEXA .....	3-70
HMSS .....	3-288

---

## **I**

IO .....	1-28
IN .....	1-21
INC .....	3-40
INT .....	3-279
INV.....	3-112

---

## **J**

JP .....	2-59
----------	------

---

## **K**

K .....	1-32
KP.....	2-21

---

## **L**

L .....	1-17
LBL .....	
JP .....	2-59

LOOP.....	2-64
LD .....	1-23
LEFT .....	3-190
LEN .....	3-185
LIFR.....	3-172
LIMIT.....	3-248
LINE .....	3-136
LN .....	3-267
LOG .....	3-268
LOOP.....	2-64
LRSR.....	2-51

---

## **M**

MAX .....	3-205
MC .....	2-54
MCE .....	2-54
MEAN.....	3-214
MIDR.....	3-192
MIDW.....	3-194
MIN .....	3-209
MUL .....	3-34
MV.....	3-8
MV/.....	3-10

---

## **N**

NEG .....	3-113
NOP .....	2-19
NSTL.....	2-73

---

## **O**

OR.....	2-2, 3-56
OR/.....	2-2
OR< .....	2-95
OR< = .....	2-95
OR< > .....	2-95
OR= .....	2-95
OR> .....	2-95
OR> = .....	2-95
OR↑ .....	2-5
OR↓ .....	2-5
ORS .....	2-15
OT .....	1-21, 2-2

---

## **P**

P .....	1-18
PERRD .....	3-348
PID .....	3-230
PMGET .....	3-339
PMSET .....	3-336
POPIX.....	3-29
POPS.....	2-16
POSSET .....	3-344
PSTRD.....	3-346
PUSHIX.....	3-28
PUSHS.....	2-16
PWR.....	3-269

---

---

**R**

R	1-15
RAD	3-271
RAMP	3-246
RCL	3-160
RCR	3-158
RDET	3-342
RDS	2-16
RECV	
MEWTOCOL	3-324
MODBUS	3-330
RET	2-84
RIGHT	3-188
ROFF	3-285
ROL	3-156
ROR	3-154
RST	2-22, 3-16

---

**S**

SADD	3-184
SBL	2-84
SCAL	3-221
SCMP	3-182
SD	1-24
SECTM	3-293
SEGT	3-120
SEND	3-312
MEWTOCOL	3-312
MODBUS	3-318
SET	2-22
SF	1-34
SHL	3-140
SHMS	3-289
SHR	3-138
SIN	3-256
SINH	3-263
SORT	3-218
SPTM	2-33
SQR	3-270
SR	1-15, 2-47
SRC	3-200
SREP	3-197
SSET	3-179
SSRC	3-186
SSTP	2-73
ST	2-2
ST/	2-2
ST<	2-95
ST<=	2-95
ST<>	2-95
ST=	2-95
ST>	2-95
ST>=	2-95
ST↑	2-5
ST↓	2-5
STC	3-177

STPE	2-73
SUB	3-32
SWAP	3-27

---

**T**

T	1-16
TAN	3-258
TANH	3-265
TE	1-27
TIMEWT	3-294
TM	2-25
TMSEC	3-292
TS	1-26

---

**U**

U	1-33
UCLR	3-349
UDC	2-44
UM	1-24
UNIT	3-124
UNITSEL	3-298

---

**W**

WDTRES	3-297
WI	1-25
WIN	3-5
WL	1-25
WO	1-25
WR	1-25
WSHL	3-152
WSHR	3-150
WX	1-25
WY	1-25

---

**X**

X	1-14
XCH	3-25
XNR	3-60
XOR	3-58

---

**Y**

Y	1-14
---	------

---

**Z**

ZONE	3-252
ZRST	2-82, 3-20

---





---

●在庫・納期・価格など、販売に関するお問い合わせは

●技術に関するお問い合わせは

コールセンター・フリーダイヤル

TEL 0120-394-205 FAX 0120-336-394

※サービス時間／9：00～17：00(12：00～13：00、当社休業日を除く)

Webでのお問い合わせ[panasonic.net/id/pidsx](http://panasonic.net/id/pidsx)

**パナソニック デバイス SUNX 株式会社**

〒486-0901 愛知県春日井市牛山町2431-1

© Panasonic Industrial Devices SUNX Co., Ltd. 2013

本書からの無断の複製はかたくお断りします。

このマニュアルの記載内容は2013年9月現在のものです。

PRINTED IN JAPAN

WUMJ-FP7CPUPGR-03