SIEMENS

SIMATIC

S7-300 および S7-400 のプログラミング用ファンクションブロックダイアグラム (FBD)

リファレンスマニュアル

05/2017
A5E41733447-AA

法律上の注意

警告事項

本書には、ユーザーの安全を守るため、および製品や接続された機器の損傷を防ぐために遵守すべき注意事項が記載されています。ユーザーの安全性に関する注意事項は、安全警告サインで強調表示されています。このサインは、物的損傷に関する注意事項には表示されません。

⚠危険

適切な予防措置を講じなければ、きわめて高い可能性で、死亡、重傷、または機器の重大な損傷を引き起こす恐れがあります。

⚠警告

適切な予防措置を講じない場合、死亡、重傷、または機器の重大な損傷を引き起こす恐れがあります。

⚠注意

適切な予防措置を講じなければ、人体に軽度の傷害を引き起こす恐れがあります。

注意

適切な予防措置を講じなければ、機器の損傷を引き起こす恐れがあります。

複数の危険度が存在する場合、一番高い危険度を表す警告が使用されます。安全警告シンボルを伴う人的傷害に関する警告には、物的損害に関する警告も含まれる場合があります。

有資格者

装置/システムのセットアップおよび使用にあたっては必ず本マニュアルを参照してください。機器のインストールおよび操作は**有資格者**のみが行うものとします。有資格者とは、法的な安全規制/規格に準拠してアースの取り付け、電気回路、設備およびシステムの設定に携わることを承認されている技術者のことをいいます。

シーメンス製品を正しくお使いいただくために

以下の点に注意してください。

⚠警告

シーメンス製品は、カタログおよび付属の技術説明書の指示に従ってお使いください。他社の製品または部品との併用は、弊社の推奨もしくは許可がある場合に限ります。シーメンス製品を正しく安全にご使用いただくには、適切な運搬、保管、組み立て、据え付け、配線、始動、操作、保守を行ってください。ご使用になる場所は、許容された範囲を必ず守ってください。付属の技術説明書に記述されている指示を遵守してください。

商標

本書において®で識別されるすべての名称は、Siemens AGの登録商標です。その他、この文書に記載されている会社名や製品名は各社の商標であるため、第三者が自己の目的のためにこれらの名前を使用すると、商標所有者の権利を侵害する恐れがあります。

免責事項

本書の内容は、記載されているハードウェアやソフトウェアとの齟齬がないよう見直されています。しかしながら、相違点をすべて取り除くことはできないため、完全な一致を保証するものではありません。本マニュアルの内容は定期的に見直され、必要な修正は次回の版で行われます。

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E41733447-AA
Ⓟ 03/2017 予告なしに変更することがあります

はじめに

目的

このマニュアルはファンクションダイアグラム(FBD)プログラミング言語のユーザプログラムを作成する手引きです。

ファンクションブロックダイアグラムの構文やファンクションを説明するリファレンスが収録されています。

必要な基本知識

このマニュアルの対象者は、S7 プログラマ、オペレータ、保守/サービス要員です。

本マニュアルを理解するには、自動化技術の全般的な知識が必要です。

さらに、コンピュータについての知識、オペレーティングシステム MS Windows XP、MS Windows Server 2003、MS Windows 7 の下での PC(プログラミングデバイス等)同様やその他の動作機器の知識が必須です。

本マニュアルの対応バージョン

本マニュアルは STEP 7 プログラミングソフトウェアパッケージの 5.6 版用です。

標準との適合

FBD は IEC(国際電気標準会議)1131-3 で定義された「Function Block Diagram」言語に対応します。詳細は、STEP 7 ファイル NORM_TBL.RTF の標準表を参照してください。

オンラインヘルプ

本マニュアルはソフトウェアに組み込まれているオンラインヘルプと併せてご使用ください。このオンラインヘルプは、STEP 7 を使用する際に詳細なサポートを提供することを目的としています。

ソフトウェアに組み込まれたヘルプシステムでは、いくつかのインターフェースが利用できます。

- 状況に応じたヘルプは、現在のコンテキスト、たとえば開いたダイアログ、アクティブなウィンドウについての情報を提供します。状況に応じたヘルプは、メニューコマンドの[ヘルプ|状況に応じたヘルプ]を選択するか、[F1]キーを押すか、ツールバーの疑問符マークの記号を使います。
- [STEP 7 のヘルプ]を呼び出すには、メニューコマンドから[ヘルプ|目次]または状況に応じたヘルプウィンドウの[STEP 7 のヘルプ]ボタンを使用します。
- [用語集]ボタンで、STEP 7 アプリケーションの用語集を呼び出すことができます。

このマニュアルは、「ファンクションブロックマニュアルのヘルプ」から抽出されたものです。マニュアルとオンラインヘルプがほぼ同一構造なので、マニュアルとオンラインヘルプを交互に容易に利用できます。

その他のサポート

技術的な質問がある場合、シーメンスの担当者または代理店の担当者に連絡をとってください。

連絡先は下記のアドレスで検索できます。

http://www.siemens.com/automation/partner

各 SIMATIC 製品およびシステムの技術文書のガイドは、以下でご覧になれます。

http://www.siemens.com/simatic-tech-doku-portal

オンラインカタログおよび注文システムは以下にあります。

http://mall.automation.siemens.com/

トレーニングセンター

Siemens 社は、SIMATIC S7 オートメーションシステムに精通していただくためのたくさんのトレーニングコースを用意しております。

システム。詳しくは、該当地区のトレーニングセンターか、下記のドイツ D 90026 ニュルンベルクの中央トレーニングセンターにご連絡ください。

Internet: http://sitrain.automation.siemens.com/sitrainworld/

技術サポート

すべての産業用オートメーション&ドライブテクノロジ製品のテクニカルサポートは次のとおりです。

- サポートリクエスト Web フォーム経由
  http://www.siemens.com/automation/support-request

テクニカルサポートについての追加情報は、インターネットの

http://www.siemens.com/automation/serviceページにあります。

インターネットによるサービスとサポート

マニュアルの他に、ノウハウを次のインターネットで提供します。

http://www.siemens.com/automation/service&support

内容:

- ニュースレター、製品に関する最新情報を提供
- 「Service & Support」の検索機能を利用して必要な文書を検索
- フォーラム、世界中のユーザーや専門家がその経験を交換
- 産業用オートメーション&ドライブテクノロジを担当するお客様の最寄りのお問い合わせ先
- フィールドサービス、修理、スペアパーツ、コンサルティングについての情報

セキュリティ情報：

シーメンスの製品およびソリューションでは、プラント、システム、マシン、ネットワークの安全な操作をサポートする産業用セキュリティファンクションが提供されています。

プラント、システム、マシン、ネットワークをサイバー攻撃から保護するには、総合的な最新の産業用セキュリティコンセプトを実装し、継続的に維持する必要があります。シーメンスの製品およびソリューションのみが、このようなコンセプトの１つのエレメントを形成します。

お客様のプラント、システム、マシン、ネットワークへの未許可のアクセスを防止するのは、お客様の責任です。システム、マシン、コンポーネントは、必要な場合に必要な部分のみ、インターネットの企業ネットワークに接続します。この場合、適当な位置で適切なセキュリティ手段(たとえば、ファイアウォールおよびネットワークセグメンテーションの使用など)を使用します。

さらに、適切なセキュリティ手段に関するシーメンスのアドバイスを検討する必要があります。産業用セキュリティに関する詳細は、次を参照してください

http://www.siemens.com/industrialsecurity。

シーメンスの製品およびソリューションは、セキュリティ度を高めるための継続的な更新を続けます。製品更新が使用可能になったらすぐにそれを適用し、常に最新の製品バージョンを使用することをシーメンスは強く推奨します。サポートされなくなった製品バージョンを使用したり、最新の更新を適用しないままでいると、お客様のサーバー攻撃に曝される度合いが高まります。

製品更新に関する継続通知を希望する場合は、次の Siemens Industrial Security RSS Feed でその申し込みを行います

http://www.siemens.com/industrialsecurity。

目次

1 ビットロジック命令

1.1 ビット論理命令の概要

説明

ビット論理命令は、1 と 0 の 2 つの桁で動作します。これらの 2 つの桁は、2 進法という記数法の基本となります。1 と 0 の 2 つの数字は、2 進数またはビットと呼ばれます。AND、OR、XOR、および出力と組み合わせて、1 は論理 YES を表し、0 は論理 NO を表します。

ビット論理命令は 1 と 0 の信号状態を解読し、ブールロジックに従ってそれらを結合します。これらの結合により、"論理演算結果"(RLO)と呼ばれる 1 または 0 の結果が生成されます。

次のファンクションを実行するビット論理命令があります。

- AND、OR、排他的 OR: これらの命令は信号状態をチェックして、RLO ビットにコピーされるか、または RLO ビットと結合される結果を生成します。
- AND-before-OR 論理演算と OR-before-AND 論理演算
- 割り付け、中間出力: これらの命令は RLO を割り付けるか、または RLO を一時的に格納します。

次の命令は、1 の RLO に対して反応します。

- S : 出力の設定
- R : 出力のリセット
- SR : Set_Reset フリップフロップ
- RS : Reset_Set フリップフロップ

その他の命令は信号立ち上がりまたは信号立ち下がりへの移行に反応して、次のファンクションを実行します。

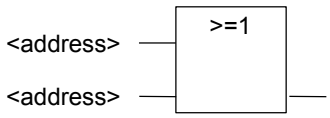
- N : 立ち下がり RLO エッジの検出
- P : 立ち上がり RLO エッジの検出
- NEG : アドレスにおける立ち下がりエッジの検出
- POS : アドレスにおける立ち上がりエッジの検出

その他の命令は、次の方法で直接 RLO に影響を及ぼします。

- バイナリ入力の挿入
- バイナリ入力の否定
- SAVE : RLO の BR メモリへの保存

1.2 >=1 : OR 論理演算

シンボル



パラメータ	データタイプ	メモリ領域	説明
<address>	BOOL	I, Q, M, T, C, D, L	アドレスは、信号状態をチェックするビットを表す。

説明

OR 命令により、OR ボックスの入力に指定されている複数のアドレスの信号状態をチェックすることができます。

1 つのアドレスの信号状態が 1 の場合、条件は満たされており、命令は結果を 1 とします。すべてのアドレスの信号状態が 0 の場合、条件は満たされておらず、命令は結果を 0 とします。

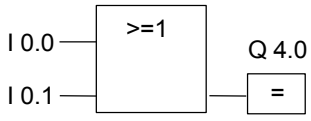
OR 命令が論理演算文字列内の最初の命令である場合、信号状態のチェック結果が RLO ビットに保存されます。

OR 命令が論理演算文字列内の最初の命令でない場合は、信号状態のチェック結果と RLO ビットに格納されている値が結合されます。値の結合は、OR 真理値表に従って行われます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

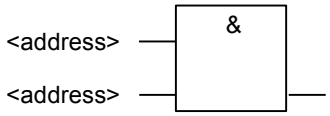
例



入力 I0.0 または入力 I0.1 の信号状態が 1 の場合、出力 Q4.0 が設定されます。

1.3 & : AND 論理演算

シンボル



パラメータ	データタイプ	メモリ領域	説明
<address>	BOOL	I, Q, M, T, C, D, L	アドレスは、信号状態をチェックするビットを表す。

説明

AND 命令により、AND ボックスの入力に指定されている複数のアドレスの信号状態をチェックすることができます。

すべてのオペランドの信号状態が 1 の場合、条件は満たされており、命令は結果を 1 とします。アドレスの信号状態が 0 の場合、条件は満たされておらず、命令は結果を 0 とします。

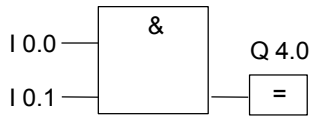
AND 命令が論理演算文字列内の最初の命令である場合、信号状態のチェック結果が RLO ビットに保存されます。

AND 命令が論理演算文字列内の最初の命令でない場合は、信号状態のチェック結果と RLO ビットに格納されている値が結合されます。値の結合は、AND 真理値表に従って行われます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例



入力 I0.0 および入力 I0.1 の信号状態が 1 の場合、出力 Q4.0 が設定されます。

1.4 AND-before-OR 論理演算と OR-before-AND 論理演算

説明

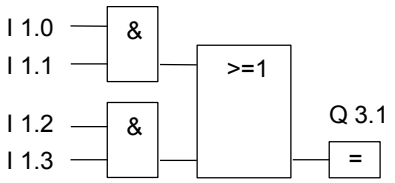
AND-before-OR 命令では、OR 真理値表に従って信号状態の結果をチェックできます。

AND-before-OR 論理演算では、少なくとも 1 つの AND 論理演算が満たされたときに信号状態が 1 になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例



AND 論理演算の少なくとも 1 つが満たされている場合、出力 Q3.1 の信号状態は 1 になります。

AND 論理演算がどれも満たされていない場合、出力 Q3.1 の信号状態は 0 になります。

説明

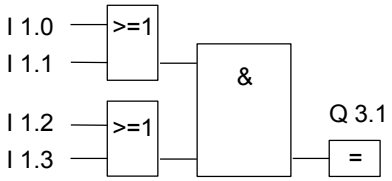
OR-before-AND 命令では、AND 真理値表に従って信号状態の結果をチェックできます。

OR-before-AND 論理演算では､すべての OR 論理演算が満たされたときに信号状態が 1 になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例

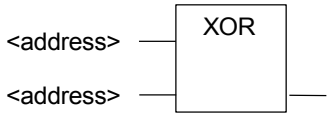


すべての OR 論理演算が満たされている場合、出力 Q3.1 の信号状態は 1 になります。

OR 論理演算の少なくとも 1 つが満たされない場合、出力 Q3.1 の信号状態は 0 になります。

1.5 XOR : 排他的 OR 論理演算

シンボル



パラメータ	データタイプ	メモリ領域	説明
<address>	BOOL	I, Q, M, T, C, D, L	アドレスは、信号状態をチェックするビットを表す。

説明

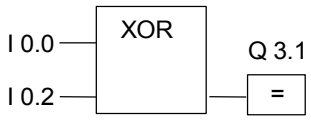
排他的 OR 命令により、排他的 OR 真理値表に従い、信号状態の結果をチェックすることができます。

排他的 OR 論理演算では、指定されている 2 つのアドレスのうちの 1 つの信号状態が 1 の場合に、信号状態が 1 になります。排他的 OR ファンクションは複数回使用できます。チェックされたアドレスの不良数が"1"の場合、相互の論理演算の結果が"1"になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例



入力 I0.0 または入力 I0.2 のどちらかの信号状態が 1 の場合（排他的に 1 である、すなわち両方が同時に 1 でない場合）、出力 Q3.1 の信号状態は 1 です。

1.6 バイナリ入力の挿入

シンボル

<address>

パラメータ	データタイプ	メモリ領域	説明
<address>	BOOL	I, Q, M, T, C, D, L	アドレスは、信号状態をチェックするビットを表す。

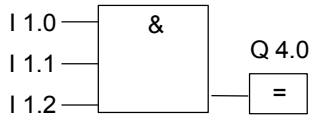
説明

バイナリ入力の挿入命令は、AND、OR、XOR のいずれかのボックスに、さらにバイナリ入力を挿入します。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	-	1	X	-

例



入力 I1.0 と入力 I1.1 と出力 I1.2 の信号状態が 1 の場合、出力 Q4.0 は 1 です。

1.7 バイナリ入力の否定

シンボル

—○|

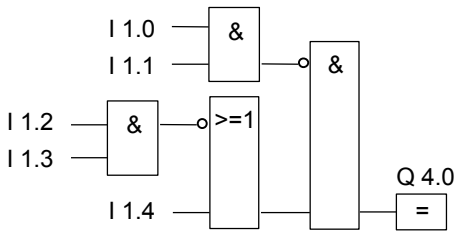
説明

バイナリ入力の否定命令は、RLO を否定します。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	-	1	X	-

例

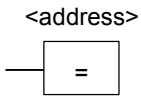


次の場合に、出力 Q4.0 は 1 になります。

- I1.0 および I1.1 の信号状態が 1 でない。
- I1.2 および I1.3 の信号状態が 1 でない。
- I1.4 の信号状態が 1 でない。

1.8 = : 割り付け

シンボル



パラメータ	データタイプ	メモリ領域	説明
<address>	BOOL	I、Q、M、D、L	アドレスは、論理演算の信号状態が割り付けられるビットを示します。

説明

割り付け命令は、論理演算の結果を生成します。論理演算の端にあるボックスには、次の基準に従い、1 または 0 の信号が割り付けられます。

- 出力ボックスの前の論理演算の条件が満たされた場合、出力の信号状態は 1 になります。
- 出力ボックス前の論理演算が条件を満たさない場合、出力の信号は 0 です。

FBD 論理演算は、命令でアドレス指定された出力に信号状態を割り付けます(RLO ビットの信号状態をアドレスに割り付けても、同じ結果が得られます)。FBD 論理演算の条件が満たされていれば、出力ボックスの信号状態は 1 になります。満たされていなければ信号状態は 0 になります。**割り付け**命令は、マスタコントロールリレー(MCR)に影響されます。

MCR のファンクションの詳細については、MCR のオン/オフを参照してください。

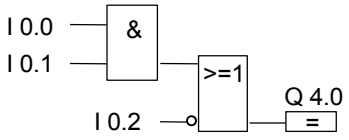
[**割り付け**]ボックスは、論理演算文字列の右端にしか配置することができません。ただし、複数の**割り付け**ボックスを使用することができます。

入力の否定 命令を使用すると、否定された割り付けを作成できます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	X	-	0

例

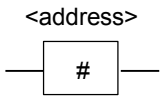


次のいずれかの場合に、出力 Q4.0 の信号状態は 1 になります。

- 入力 I0.0 および I0.1 の信号状態が 1
- 入力 I0.2 が 0

1.9 # : 中間出力

シンボル



パラメータ	データタイプ	メモリ領域	説明
<address>	BOOL	I, Q, M, D, *L	アドレスは、RLO が割り付けられるビットを示します。

* ローカルデータスタック内のアドレスは、コードブロック(FC、FB、OB)の TEMP 領域にある変数宣言テーブルで宣言されている場合にだけ使用できます。

説明

中間出力命令は、RLO をバッファに記憶する中間エレメントです。より正確に言うと、このエレメントは中間出力命令の直前に開かれた分岐のビット論理演算をバッファに記憶します。

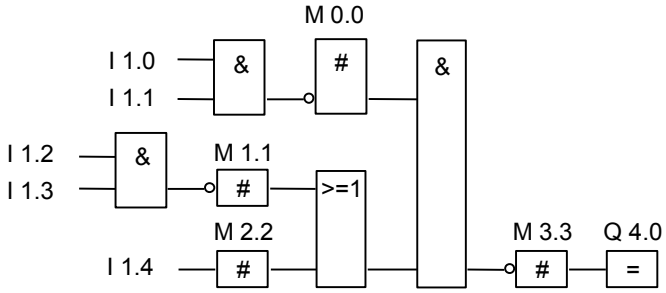
中間出力命令は、マスタコントロールリレー(MCR)の影響を受けます。MCR ファンクションの詳細については、MCR のオン/オフを参照してください。

中間出力の入力を否定することによって、中間出力を否定することができます。

ステータスワード

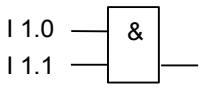
	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	X	-	1

例

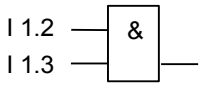


中間出力命令は、次の論理演算結果をバッファに入れます。

M0.0 は、次の否定 RLO をバッファに記憶します。



M1.1 は、次の否定 RLO を保存します。

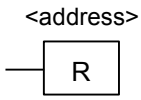


M2.2 は、I1.4 の RLO を保存します。

M3.3 はビット論理演算全体の否定 RLO を保存します。

1.10 R : 出力のリセット

シンボル



パラメータ	データタイプ	メモリ領域	説明
<address>	BOOL TIMER COUNTER	I, Q, M, T, C, D, L	アドレスは、リセットされるビットを表す。

説明

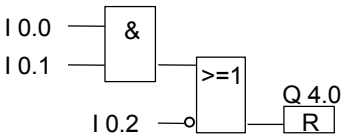
出力の設定命令は、RLO が 1 の場合にだけ実行されます。RLO が 1 の場合、この命令は指定されたアドレスを 0 に設定します。RLO が 0 の場合、この命令は指定されたアドレスを変更しません。

出力のリセット命令は、マスタコントロールリレー(MCR)の影響を受けます。MCR ファンクションの詳細については、MCR のオン/オフを参照してください。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	X	-	0

例



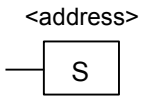
以下の場合のみ、出力 Q4.0 の信号状態は 0 にリセットされます。

- 入力 I0.0 および I0.1 の信号状態が 1
- OR 入力 I0.2 の信号状態が

ブランチの RLO が 0 の場合、出力 Q4.0 の信号状態は変更されません。

1.11 S : 出力の設定

シンボル



パラメータ	データタイプ	メモリ領域	説明
<address>	BOOL	I、Q、M、D、L	アドレスは、設定されるビットを表す。

説明

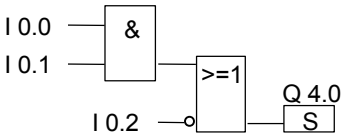
出力の設定命令は、RLO が 1 の場合にだけ実行されます。RLO が 1 の場合、この命令は指定されたアドレスを 1 に設定します。RLO が 0 の場合、この命令は指定されたアドレスを変更しません。

出力の設定命令は、マスタコントロールリレー(MCR)の影響を受けます。MCR ファンクションの詳細については、MCR のオン/オフを参照してください。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	X	-	0

例



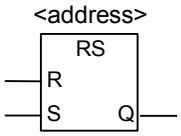
以下の場合のみ、出力 Q4.0 の信号状態は 1 に設定されます。

- 入力 I0.0 および I0.1 の信号状態が 1
- OR 入力 I0.2 の信号状態が

ブランチの RLO が 0 の場合、出力 Q4.0 の信号状態は変更されません。

1.12 RS : Reset_Set フリップフロップ

シンボル



パラメータ	データタイプ	メモリ領域	説明
<address>	BOOL	I、Q、M、D、L	アドレスは、設定またはリセットするビットを表す。
S	BOOL	I、Q、M、D、L、T、C	リセット命令の有効化
R	BOOL	I、Q、M、D、L、T、C	設定命令の有効化
Q	BOOL	I、Q、M、D、L	<address>の信号状態

説明

Reset_Set フリップフロップ命令は、RLO が 1 の場合にだけ、設定(S)またはリセット(R)命令を実行します。RLO が 0 の場合は、これらの命令に影響を与えず、命令で指定されたアドレスは変更されません。

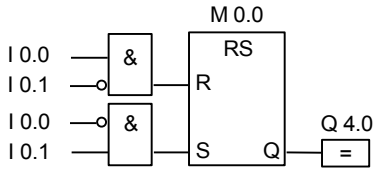
Reset_Set フリップフロップは、入力 R の信号状態が 1 で、入力 S の信号状態が 0 の場合にリセットされます。入力 R が 0 で入力 S が 1 の場合、フリップフロップは設定されます。また、両方の入力の RLO がともに 1 の場合も、フリップフロップは設定されます。

Reset_Set フリップフロップ命令は、マスタコントロールリレー(MCR)の影響を受けます。MCR ファンクションの詳細については、MCR のオン/オフを参照してください。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例

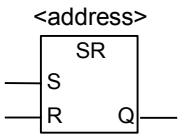


I0.0 が 1 で I0.1 が 0 の場合、メモリビット M0.0 はリセットされ、出力 Q4.0 は 0 になります。I0.0 が 0 で I0.1 が 1 の場合、メモリビット M0.0 は設定され、出力 Q4.0 は 1 になります。

両方の信号状態が 0 の場合、変化はありません。信号状態がともに 1 の場合は、命令の順序により設定命令が優先されます。このため、M 0.0 が設定され、Q 4.0 は 1 になります。

1.13 SR : Set_Reset フリップフロップ

シンボル



パラメータ	データタイプ	メモリ領域	説明
<address>	BOOL	I、Q、M、D、L	アドレスは、設定またはリセットするビットを表す。
S	BOOL	I、Q、M、D、L、T、C	設定命令の有効化
R	BOOL	I、Q、M、D、L、T、C	リセット命令の有効化
Q	BOOL	I、Q、M、D、L	<address>の信号状態

説明

Set_Reset フリップフロップ命令は、RLO が 1 の場合にだけ、設定(S)またはリセット(R)命令を実行します。RLO が 0 の場合は、これらの命令に影響を与えず、命令で指定されたアドレスは変更されません。

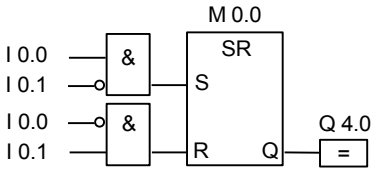
Set_Reset フリップフロップは、入力 S の信号状態が 1 で、入力 R の信号状態が 0 の場合に設定されます。入力 S が 0 で入力 R が 1 の場合、フリップフロップはリセットされます。また、両方の入力の RLO がともに 1 の場合も、フリップフロップはリセットされます。

Set_Reset フリップフロップ命令は、マスタコントロールリレー(MCR)の影響を受けます。MCR ファンクションの詳細については、MCR のオン/オフを参照してください。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例

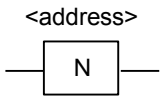


I0.0 が 1 で I0.1 が 0 の場合、メモリビット M0.0 が設定され Q4.0 が 1 になります。I0.0 が 0 で I0.1 が 1 の場合、メモリビット M0.0 がリセットされ、Q4.0 が 0 になります。

両方の信号状態が 0 の場合、変化はありません。信号状態がともに 1 の場合は、命令の順序によりリセット命令が優先されます。このため、M0.0 がリセットされ、Q 4.0 は 0 になります。

1.14 N : 立ち下がり RLO エッジの検出

シンボル



パラメータ	データタイプ	メモリ領域	説明
<address>	BOOL	I、Q、M、D、L	アドレスは、前回のチェック時の RLO の状態が格納されるエッジメモリビットを表す。

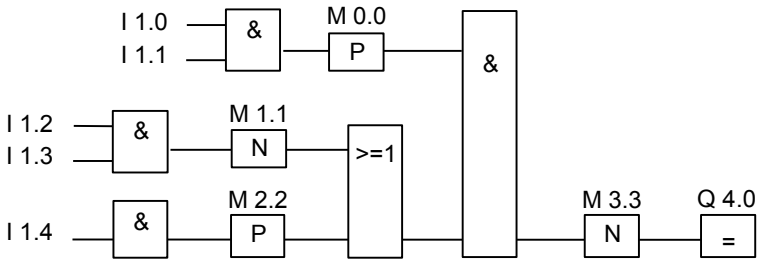
説明

立ち下がり RLO エッジの検出命令は、指定されているアドレスで 1 から 0 の状態遷移(信号立ち下がり)を監視し、この状態が検出された場合、RLO の状態を 1 に設定します。RLO の信号状態は、アドレスの信号状態（エッジメモリビット）と比較されます。この命令が実行される前にアドレスの信号状態が 1 で RLO が 0 であれば、命令が実行された後、RLO は 1(パルス)になり、それ以外の場合はすべて RLO は 0 になります。命令が実行される前の RLO は、このアドレスに保存されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	X	X	1

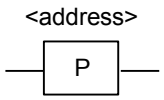
例



エッジメモリビット M3.3 は、前の RLO の信号状態を保存します。

1.15 P : 立ち上がり RLO エッジの検出

シンボル



パラメータ	データタイプ	メモリ領域	説明
<address>	BOOL	I、Q、M、D、L	アドレスは、前回のチェック時の RLO の状態が格納されるエッジメモリビットを表す。

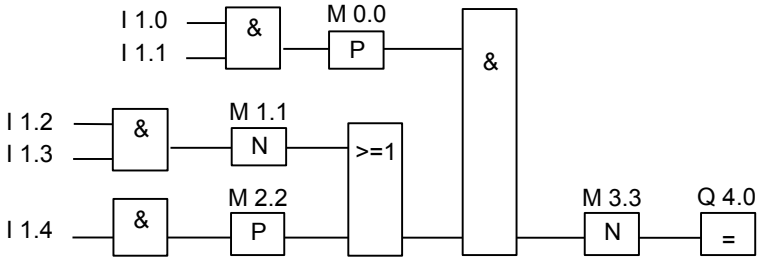
説明

立ち上がり RLO エッジの検出命令は、指定されているアドレスで 0 から 1 の状態遷移(信号立ち上がり)を監視し、この状態が検出された場合、RLO の状態を 1 に設定します。RLO の信号状態は、アドレスの信号状態（エッジメモリビット）と比較されます。この命令が実行される前にアドレスの信号状態が 0 で RLO が 1 であれば、命令が実行された後、RLO は 1(パルス)になり、それ以外の場合はすべて RLO は 0 になります。命令が実行される前の RLO は、このアドレスに保存されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	X	X	1

例



エッジメモリビット M3.3 は、前の RLO の信号状態を保存します。

1.16 SAVE : RLO の BR メモリへの保存

シンボル

説明

RLO の BR メモリへの保存命令は、ステータスワードの BR ビットに RLO を保存します。最初のチェックビット FC は、リセットされません。

このため、次のネットワークに AND 論理演算がある場合、BR ビットの状態を論理演算に取り込むことができます。

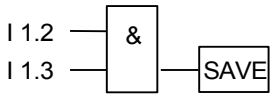
SAVE 命令(LAD、FBD、STL)については、マニュアルやオンラインヘルプで規定されている推奨の使用法ではなく、次のことが適用されます。
SAVE を使用した後に、同じブロックまたは従属ブロック内で BR ビットをチェックすることはお勧めできません。これは、中間で発生する多くの命令によって、BR ビットが修正される可能性があるためです。ブロックを終了するときには、その前に SAVE 命令を使用してください。これは、ENO 出力(= BR ビット)が、RLO ビットの値に設定されるため、ブロックにエラーがあるかどうかチェックできるからです。

RLO の BR メモリへの保存命令では、ネットワークの RLO が従属ブロック内の論理演算の一部を形成することができます。呼び出し元ブロックの CALL 命令で、最初のチェックビットがリセットされます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	-	-	-	-

例

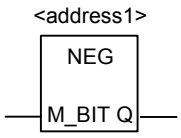


論理演算結果（RLO）は、BR ビットへ書き込まれます。

BR バイナリリザルト(ステータスワード、ビット 8)

1.17 NEG：アドレスにおける立ち下がりエッジの検出

シンボル



パラメータ	データタイプ	メモリ領域	説明
<address1>	BOOL	I、Q、M、D、L	立ち下がりエッジがチェックされる信号
M_BIT	BOOL	Q, M, D	M_BIT アドレスは、前回のチェック時の NEG の信号状態が格納されるエッジメモリビットを表す。M_BIT アドレスにプロセス入力イメージ I メモリ領域を使用できるのは、このアドレスをすでに使用している入力モジュールがない場合だけである。
Q	BOOL	I、Q、M、D、L	1 回限りの出力

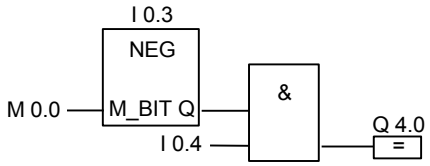
説明

アドレスにおける立ち下がりエッジの検出命令は、<address1>現在の信号状態と M_BIT パラメータに格納されている前回のチェック時の信号状態を比較します。1 から 0 に状態が遷移している場合、出力 Q に値 1 が設定され、それ以外の場合はすべて値 0 が設定されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	1	X	1

例

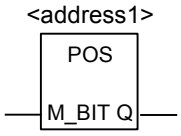


次の場合に、出力 Q4.0 は 1 になります。

- 入力 I0.3 に信号立ち下がりがある
- 入力 I0.4 の信号状態が 1

1.18 POS : アドレスにおける立ち上がりエッジの検出

シンボル



パラメータ	データタイプ	メモリ領域	説明
<address1>	BOOL	I、Q、M、D、L	信号立ち上がりをチェックする信号。
M_BIT	BOOL	Q, M, D	M_BIT アドレスは、POS の前の信号状態を保存するために使用されるエッジメモリビットを示します。このアドレスを使用している入力モジュールがない場合、M_BIT には、プロセスイメージ入力領域 I だけを使用します。
Q	BOOL	I、Q、M、D、L	1 回限りの出力

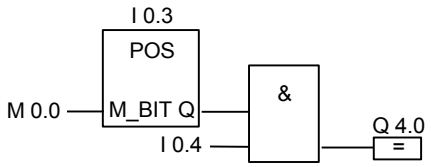
説明

アドレスにおける立ち上がりエッジの検出命令は、<address1>の信号状態とパラメータ M_BIT に格納されている前回のチェック時の信号状態を比較します。0 から 1 への変化があった場合、出力 Q の値は 1 になります。その他の場合、値は 0 です。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	1	X	1

例



次の場合に、出力 Q4.0 は 1 になります。

入力 I0.3 に信号立ち上がりがある

入力 I0.4 の信号状態が 1

1.18 POS : アドレスにおける立ち上がりエッジの検出

2 比較命令

2.1 比較命令の概要

説明

IN1 と IN2 は、選択した比較タイプに従って比較されます。

== IN1 は IN2 と等しい
<> IN1 は IN2 と等しくない
> IN1 は IN2 より大きい
< IN1 は IN2 より小さい
>= IN1 は IN2 より大きいか等しい
<= IN1 は IN2 より小さいか等しい

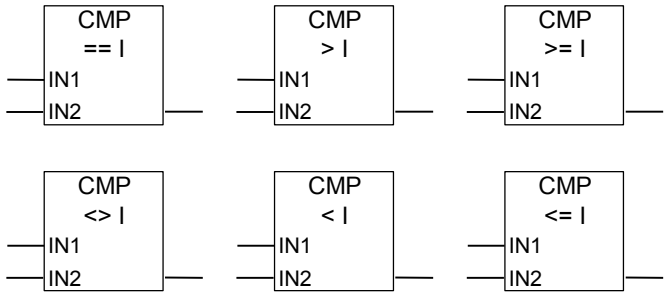
比較結果が true の場合、ファンクションの RLO は"1"になります。そうでない場合、RLO は 0 になります。比較結果自体を否定することはできませんが、逆の比較ファンクションを使用することで、否定と同じ効果を得ることができます。

使用可能な比較命令を次に示します。

- CMP ? I : 正数の比較
- CMP ? D : 倍長整数の比較
- CMP ? R : 実数の比較

2.2 CMP ? I : 正数の比較

シンボル



パラメータ	データタイプ	メモリ領域	説明
IN1	INT	I、Q、M、D、L または定数	比較する最初の値
IN2	INT	I、Q、M、D、L または定数	比較する 2 番目の値
Box output	BOOL	I、Q、M、D、L	比較の結果

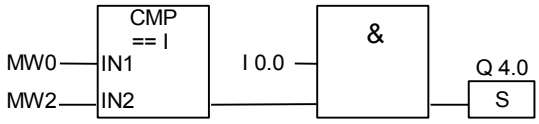
説明

[**整数の比較**]命令では、16 ビットの浮動小数点数を基にして 2 つの値を比較します。この命令では、リストボックスで選択した 比較タイプに従って、入力 IN1 と IN2 を比較します。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	0	-	0	X	X	1

例

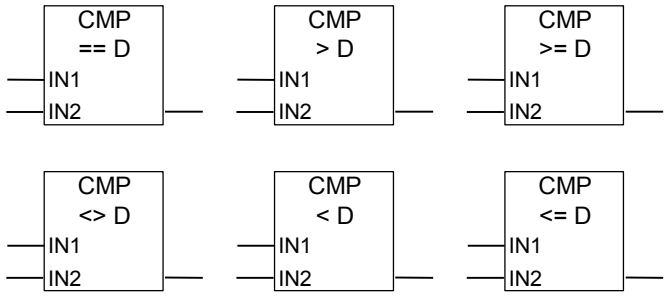


Q 4.0 は次のような場合に設定されます

- MW0 が MW2 と等しい
- 入力 I0.0 の信号状態が 1

2.3 CMP ? D : 倍長整数の比較

シンボル



パラメータ	データタイプ	メモリ領域	説明
IN1	DINT	I、Q、M、D、L または定数	比較する最初の値
IN2	DINT	I, Q, M, D,L,または定数	比較する 2 番目の値
Box output	BOOL	I、Q、M、D、L	比較の結果

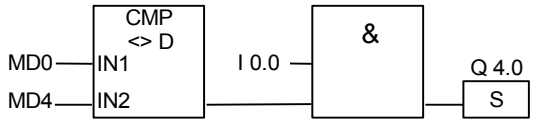
説明

倍長整数の比較命令では、32 ビットの浮動小数点数を基に 2 つの値を比較します。この命令では、リストボックスで選択した 比較タイプに従って、入力 IN1 と IN2 を比較します。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	X	X	0	-	0	X	X	1

例

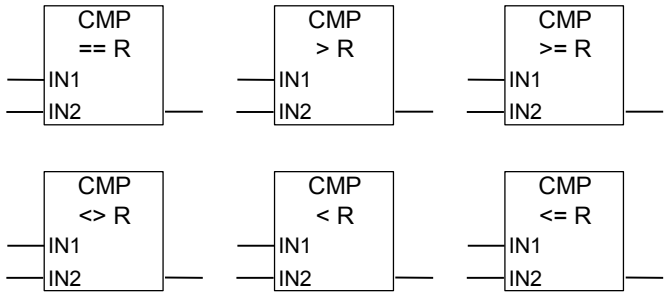


Q 4.0 は次のような場合に設定されます

- MD0 が MD4 に等しくない
- なおかつ入力 I 0.0 の信号状態が 1 である

2.4 CMP ? R : 実数の比較

シンボル



パラメータ	データタイプ	メモリ領域	説明
IN1	REAL	I、Q、M、D、L または定数	比較する最初の値
IN2	REAL	I、Q、M、D、L または定数	比較する 2 番目の値
Box output	BOOL	I、Q、M、D、L	比較の結果

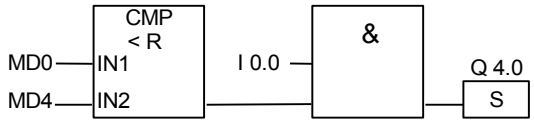
説明

実数の比較命令では、実数を基に 2 つの値を比較します。この命令では、リストボックスで選択した比較タイプに従って、入力 IN1 と IN2 を比較します。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	X	X	X	X	0	X	X	1

例



Q 4.0 は次のような場合に設定されます

- MD0 が MD4 より小さい
- なおかつ入力 I 0.0 の信号状態が 1 である

3 変換命令

3.1 変換命令の概要

説明

以下の命令を使用すれば、２進化 10 進数と整数を他のタイプの数に変換できます。

- BCD_I : BCD から整数への変換
- I_BCD : 整数の BCD への変換
- BCD_DI : BCD の倍長整数への変換
- I_DI : 整数の倍長整数への変換
- DI_BCD : 倍長整数の BCD への変換
- DI_R : 倍長整数の実数への変換

以下の命令を使用すれば、整数の補数を生成したり、浮動小数点数の符号を反転したりできます。

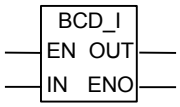
- INV_I : 整数の 1 の補数
- INV_DI : 倍長整数の 1 の補数
- NEG_I : 整数の 2 の補数
- NEG_DI : 倍長整数の 2 の補数
- NEG_R : 実数の否定

以下の命令を使用すれば、アキュムレータ 1 の 32 ビット浮動小数点数を 32 ビット整数(倍長整数)に変換できます。個々の命令で、丸め方法が異なります。

- ROUND : 倍長整数の丸め
- TRUNC : 倍長整数部の切り捨て
- CEIL : 切り上げ
- FLOOR : 切り捨て

3.2 BCD_I : BCD から整数への変換

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	WORD	I、Q、M、D、L または定数	BCD フォーマットの数値
OUT	INT	I、Q、M、D、L	整数変換後の値
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

BCD から整数への変換命令は、入力パラメータ IN の内容を 2 進化 10 進数フォーマットの 3 桁の数(BCD、±999)として読み取り、整数値に変換します。出力パラメータ OUT には結果が示されます。

ENO と EN の信号状態は常に同じです。

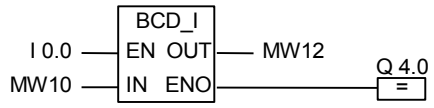
BCD の 10 進数のいずれかが 10～15 の無効な範囲にある場合、変換を実行しようとすると BCD エラーが発生し、以下が実行されます。

- CPU は STOP モードに変換されます。"BCD 変換エラー"は、イベント ID 番号 2521 で診断バッファに入力されます。
- OB121 がプログラムされている場合は呼び出されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	1	-	-	-	-	0	1	1	1

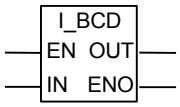
例



0 の信号状態が 1 の場合、変換が実行されます。メモリワード MW10 の内容を BCD フォーマットの 3 桁の数として読み取り、整数に変換します。この結果は、メモリワード MW12 に格納されます。変換が実行されると、出力 Q4.0 は 1 になります(ENO＝EN)。

3.3 I_BCD : 整数の BCD への変換

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	INT	I、Q、M、D、L または定数	整数
OUT	WORD	I、Q、M、D、L	BCD フォーマットの整数値
ENO	BOOL	I、Q、M、D、L	イネーブル出力

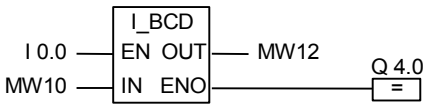
説明

整数の BCD への変換命令は、入力パラメータ IN の内容を整数値として読み取り、2 進化 10 進数形式の 3 桁の数(BCD、± 999)に変換します。出力パラメータ OUT には結果が示されます。オーバーフローが発生した場合、ENO は 0 に設定されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	-	-	X	X	0	X	X	1

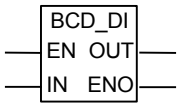
例



0 の信号状態が 1 の場合、変換が実行されます。メモリワード MW10 の内容を整数として読み取り、BCD フォーマットの 3 桁の数値に変換します。この結果は、メモリワード MW12 に格納されます。オーバーフローが発生した場合は、出力 Q4.0 の信号状態が 0 になります。入力 EN の信号状態が 0(変換が実行されないことを意味する)の場合も、出力 Q4.0 の信号状態が 0 になります。

3.4 BCD_DI : BCD の倍長整数への変換

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	DWORD	I、Q、M、D、L または定数	BCD フォーマットの数値
OUT	DINT	I、Q、M、D、L	BCD の倍長整数
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

BCD の倍長整数への変換命令は、入力パラメータ IN の内容を 2 進化 10 進数形式の 7 桁の数(BCD、±9,999,999)として読み取り、倍長整数値に変換します。出力パラメータ OUT には結果が示されます。

ENO と EN の信号状態は常に同じです。

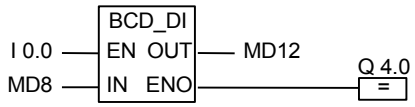
BCD の 10 進数のいずれかが 10～15 の無効な範囲にある場合、変換を実行しようとすると BCD エラーが発生し、以下が実行されます。

- CPU は STOP モードに変換されます。"BCD 変換エラー"は、イベント ID 番号 2521 で診断バッファに入力されます。
- OB121 がプログラムされている場合は呼び出されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	1	-	-	-	-	0	1	1	1

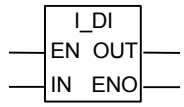
例



0 の信号状態が 1 の場合、変換が実行されます。メモリダブルワード MD8 の内容を BCD フォーマットの 7 桁の数として読み取り、倍長整数に変換します。この結果は、MD12 に格納されます。変換が実行されると、出力 Q4.0 は 1 になります(ENO＝EN)。

3.5 I_DI : 整数の倍長整数への変換

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	INT	I、Q、M、D、L または定数	または定数
OUT	DINT	I、Q、M、D、L	結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

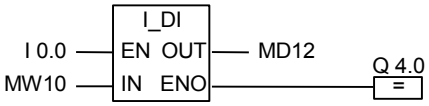
説明

整数から倍長整数への変換命令は、入力パラメータ IN の内容を整数として読み取り、倍長整数に変換します。出力パラメータ OUT には結果が示されます。ENO と EN の信号状態は常に同じです。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	1	-	-	-	-	0	1	1	1

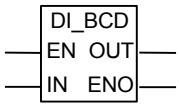
例



I0.0 の信号状態が 1 になると、変換が実行され、メモリワード MW10 に入力された整数は倍長整数へ変換されます。この結果は、メモリダブルワード MD12 に格納されます。変換が実行されると、出力 Q4.0 は 1 になります(ENO＝EN)。

3.6 DI_BCD : 倍長整数の BCD への変換

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	DINT	I、Q、M、D、L または定数	倍長整数
OUT	DWORD	I、Q、M、D、L	BCD 表記に変換後の値
ENO	BOOL	I、Q、M、D、L	イネーブル出力

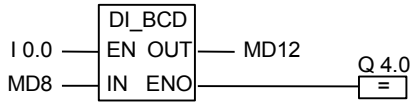
説明

倍長整数の BCD への変換命令は、入力パラメータ IN の内容を倍長整数として読み取り、BCD 形式の 7 桁の数(± 9,999,999)に変換します。出力パラメータ OUT には結果が示されます。オーバーフローが発生した場合、ENO は 0 に設定されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	-	-	X	X	0	X	X	1

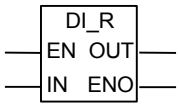
例



0 の信号状態が 1 の場合、変換が実行されます。メモリダブルワード MD8 の内容を倍長整数として読み取り、BCD フォーマットの 7 桁の数に変換します。この結果は、MD12 に格納されます。オーバーフローが発生した場合は、出力 Q4.0 の信号状態が 0 になります。入力 EN の信号状態が 0(変換が実行されないことを意味する)の場合も、出力 Q4.0 の信号状態が 0 になります。

3.7 DI_R : 倍長整数の実数への変換

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	DINT	I、Q、M、D、L または定数	または定数
OUT	REAL	I、Q、M、D、L	結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

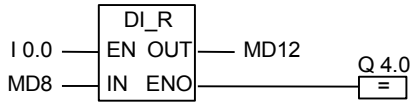
説明

倍長整数から実数への変換命令は、入力パラメータ IN の内容を倍長整数として読み取り、実数に変換します。出力パラメータ OUT には結果が示されます。ENO と EN の信号状態は常に同じです。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	1	-	-	-	-	0	1	1	1

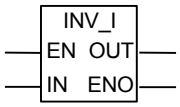
例



0 の信号状態が 1 の場合、変換が実行されます。メモリダブルワード MD8 の内容を整数として読み取り、実数に変換します。この結果は、メモリダブルワード MD12 に格納されます。変換が実行されない場合、出力 Q4.0 の信号状態は 0 です(ENO＝EN)。

3.8 INV_I : 整数の 1 の補数

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	INT	I、Q、M、D、L または定数	入力値
OUT	INT	I、Q、M、D、L	ビットが反転された整数
ENO	BOOL	I、Q、M、D、L	イネーブル出力

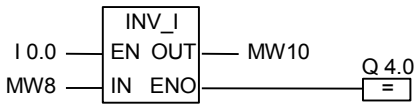
説明

整数の 1 の補数命令は、入力パラメータ IN の内容を読み取り、FFFFH でマスクされたブールワード論理命令排他的 Or ワードを実行してすべてのビットの値が逆転するようにします。出力パラメータ OUT には結果が示されます。ENO と EN の信号状態は常に同じです。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	1	-	-	-	-	0	1	1	1

例



0 の信号状態が 1 の場合、変換が実行されます。MW8 の各ビット値は、次のように逆転されます。

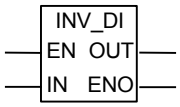
MW8 = 01000001 10000001 →

MW10 = 10111110 01111110

I0.0 の信号状態が 0 の場合、変換は実行されず、Q4.0 は 0 になります(ENO＝EN)。

3.9 INV_DI : 倍長整数の 1 の補数

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	DINT	I、Q、M、D、L または定数	入力値
OUT	DINT	I、Q、M、D、L	ビットが反転された倍長整数
ENO	BOOL	I、Q、M、D、L	イネーブル出力

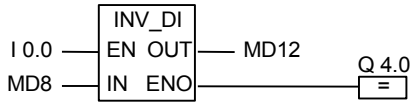
説明

倍長整数の 1 の補数命令は、入力パラメータ IN の内容を読み取り、FFFF FFFFH でマスクされたブールワード論理演算排他的 Or ワードを実行して、すべてのビットの値が逆転するようにします。出力パラメータ OUT には結果が示されます。ENO と EN の信号状態は常に同じです。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	1	-	-	-	-	0	1	1	1

例



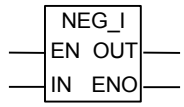
I0.0 の信号状態が 1 になると、命令が実行され、メモリダブルワード MD8 の全てのビットの値が反転します。

MD8 = F0FF FFF0 → MD12 = 0F00 000F

変換が実行されないと、Q4.0 は 0 になります(ENO = EN)。

3.10 NEG_I : 整数の 2 の補数

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	INT	I、Q、M、D、L または定数	入力値
OUT	INT	I、Q、M、D、L	整数の 2 の補数
ENO	BOOL	I、Q、M、D、L	イネーブル出力

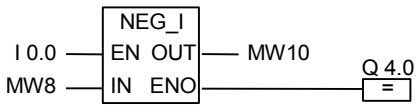
説明

整数の 2 の補数命令は、入力パラメータ IN の内容を読み取り、符号を変換します(たとえば、正数から負数へ変換)。出力パラメータ OUT には結果が示されます。EN と ENO の信号状態は常に同じですが、EN の信号状態が 1 のときにオーバーフローが発生した場合は例外です。この場合、ENO の信号状態は 0 になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

例



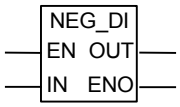
I0.0 の信号状態が 1 の場合は、変換が実行されます。メモリワード MW8 の値は、符号が逆になって 0 でメモリワード MW10 に出力されます。

MW8 = +10 → MW10 = -10

EN の信号状態が 1 でオーバーフローが発生すると、ENO は 0 になり、Q4.0 の信号状態も 0 になります。変換が実行されなかった場合、Q4.0 は 0(ENO=EN)になります。

3.11 NEG_DI : 倍長整数の 2 の補数

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	DINT	I、Q、M、D、L または定数	入力値
OUT	DINT	I、Q、M、D、L	符号が反転された倍長整数
ENO	BOOL	I、Q、M、D、L	イネーブル出力

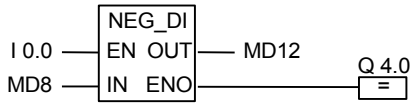
説明

倍長整数の 2 の補数命令は、入力パラメータ IN の内容を読み取り、符号を変換します(たとえば、正数から負数へ)。出力パラメータ OUT には結果が示されます。EN と ENO の信号状態は常に同じですが、EN の信号状態が 1 のときにオーバーフローが発生した場合は例外です。この場合、ENO の信号状態は 0 になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

例



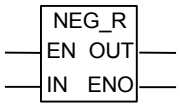
10.0 の信号状態が 1 の場合、変換が実行されます。メモリダブルワード MD8 の内容は、符号が逆になって 0 でメモリダブルワード MD10 に出力されます。

MW8 = +10 → MW10 = -10

EN の信号状態が 1 でオーバーフローが発生すると、ENO は 0 になり、Q4.0 の信号状態も 0 になります。変換が実行されなかった場合、Q4.0 は 0(ENO=EN)になります。

3.12 NEG_R : 実数の否定

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	入力値
OUT	REAL	I、Q、M、D、L	入力値の符号を反転させた値
ENO	BOOL	I、Q、M、D、L	イネーブル出力

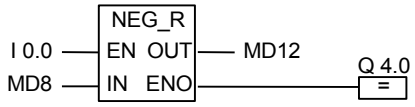
説明

実数の否定命令は、入力パラメータ IN の内容を読み取り、符号ビットを逆にします(この命令により、たとえばプラスの 0 からマイナスの 1 というように、数値の符号が変わります)。指数および仮数のビットは変わりません。出力パラメータ OUT には結果が示されます。EN と ENO の信号状態は常に同じですが、EN の信号状態が 1 のときにオーバーフローが発生した場合は例外です。この場合、ENO の信号状態は 0 になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	-	-	-	-	0	X	X	1

例



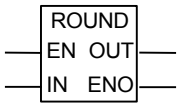
10.0 の信号状態が 1 の場合、変換が実行されます。メモリダブルワード MD8 の内容は、以下の例で示すように、符号が逆になって 0 でメモリダブルワード MD12 に出力されます。

MD8 = + 6.234 → MD12 = - 6.234

変換が実行されない場合、出力 Q4.0 の信号状態は 0 です(ENO＝EN)。

3.13 ROUND：倍長整数の丸め

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	丸められる値
OUT	DINT	I、Q、M、D、L	四捨五入後の値
ENO	BOOL	I、Q、M、D、L	イネーブル出力

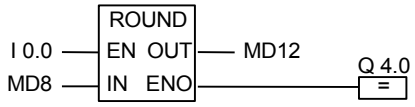
説明

倍長整数の丸め命令は、入力パラメータ IN の内容を実数として読み取り、倍長整数に変換します。結果はこの実数に最も近い整数となり、出力パラメータ OUT で出力されます。ただし、小数部が x.5 の場合は例外で、この数は偶数へ変換されます(たとえば、2.5 → 2、1.5 → 2)。オーバーフローが発生すると、ENO は 0 に設定されます。入力値が実数でない場合、OV ビットと OS ビットの値は 1 になり、ENO の値は 0 になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	-	-	X	X	0	X	X	1

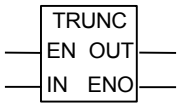
例



0 が 1 の場合、変換が実行されます。メモリダブルワード MD8 の内容を実数として読み取り、倍長整数に変換します。この近似値への丸めファンクションの結果は、メモリダブルワード MD12 に格納されます。オーバーフローが発生した場合は、出力 Q4.0 の信号状態が 0 になります。入力 EN の信号状態が 0(変換が実行されないことを意味する)の場合も、出力 Q4.0 の信号状態が 0 になります。

3.14 TRUNC：倍長整数部の切り捨て

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	または定数
OUT	DINT	I、Q、M、D、L	IN の整数部
ENO	BOOL	I、Q、M、D、L	イネーブル出力

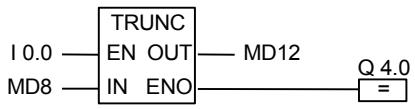
説明

倍長整数部の切り捨て命令は、入力パラメータ IN の内容を実数として読み取り、倍長整数に変換します(たとえば、1.5 は 1 になる)。結果は実数の整数部分です。)出力パラメータ OUT には結果が示されます。オーバーフローが発生すると、ENO は 0 に設定されます。入力値が実数でない場合、OV ビットと OS ビットの値は 1 になり、ENO の値は 0 になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	-	-	X	X	0	X	X	1

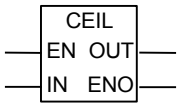
例



この命令は、I0.0 の信号状態が 1 の場合に実行されます。メモリダブルワード MD8 の内容を実数として読み取り、「0 への丸め原則」に従って倍長整数に変換します。結果として整数部分が出力され、メモリダブルワード MD12 に格納されます。オーバーフローが発生した場合は、出力 Q4.0 の信号状態が 0 になります。入力 EN の信号状態が 0(変換が実行されないことを意味する)の場合も、出力 Q4.0 の信号状態が 0 になります。

3.15 CEIL：切り上げ

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	または定数
OUT	DINT	I、Q、M、D、L	結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

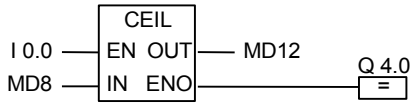
説明

切り上げ命令は、入力パラメータ IN の内容を実数として読み取り、この数値を倍長整数に変換します(例: +1.2 -> +2; -1.5 -> -1)。この結果は、指定された実数より大きいか等しい値のうち最も小さい整数になります。出力パラメータ OUT には結果が示されます。オーバーフローが発生すると、ENO は 0 になります。入力値が実数でない場合、OV ビットと OS ビットの値は 1 になり、ENO の値は 0 になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	-	-	X	X	0	X	X	1

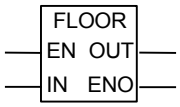
例



I0.0 が 1 になると、変換が実行されます。メモリダブルワード MD8 に入力された実数は、小数部切り上げにより、倍長整数へ変換されます。この結果は、メモリダブルワード MD12 に格納されます。オーバーフローが発生した場合は、出力 Q4.0 の信号状態が 0 になります。入力 EN の信号状態が 0(変換が実行されないことを意味する)の場合も、出力 Q4.0 の信号状態が 0 になります。

3.16 FLOOR : 切り捨て

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	または定数
OUT	DINT	I、Q、M、D、L	結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

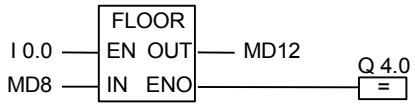
説明

切り捨て命令は、入力パラメータ IN の内容を実数として読み取り、倍長整数に変換します。この結果は、指定された実数より小さいか等しい整数のうち最も大きい整数になります。出力パラメータ OUT には結果が示されます。オーバーフローが発生すると、ENO は 0 に設定されます。入力値が実数でない場合、OV ビットと OS ビットの値は 1 になり、ENO の値は 0 になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	-	-	X	X	0	X	X	1

例



I0.0 が 1 になると変換が実行されます。メモリダブルワード MD8 の内容を実数として読み取り、次に小さい(または等しい)整数に切り下げて倍長整数に変換します。この結果は、メモリダブルワード MD12 に格納されます。オーバーフローが発生した場合は、出力 Q4.0 の信号状態が 0 になります。入力 EN の信号状態が 0(変換が実行されないことを意味する)の場合も、出力 Q4.0 の信号状態が 0 になります。

4 カウンタ命令

4.1 カウンタ命令の概要

メモリ内の領域

カウンタは、CPU のメモリ内にカウンタ用に確保された領域を持っています。このメモリ領域は、各カウンタアドレスに対して 1 つの 16 ビットワードを確保します。FBD でプログラムを行う場合、256 個のカウンタが使用できます。カウンタ命令は、カウンタメモリ領域にアクセスする唯一のファンクションです。

カウンタ値

カウンタワードのビット 0～9 には、カウント値がバイナリコードで格納されます。カウンタが設定されると、カウンタ値はカウンタワードへ転送されます。カウント値の範囲は 0～999 です。

次のカウンタ命令を使用すれば、この範囲内でカウント値を変更することができます。

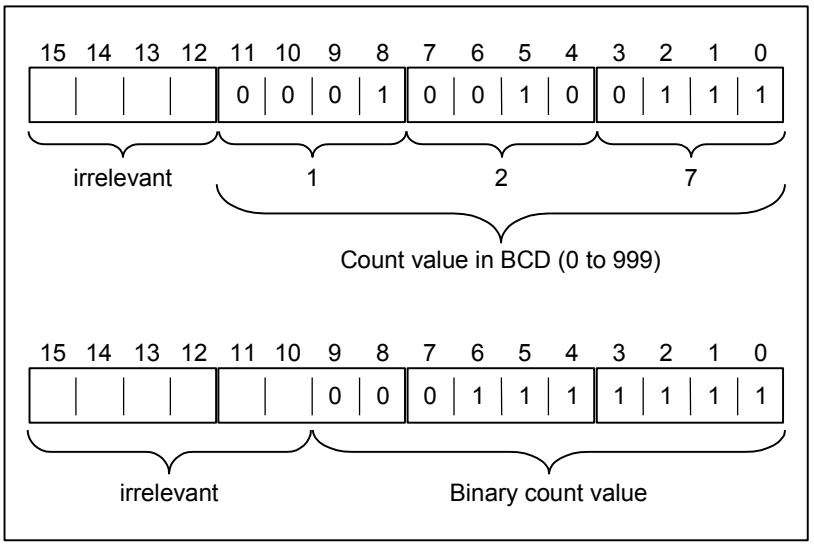
- S_CUD : パラメータおよびカウントアップ/カウントダウンの割り付け
- S_CU : パラメータおよびカウントアップの割り付け
- S_CD : パラメータおよびカウントダウンの割り付け
- SC : カウンタ値の設定
- CU : カウントアップ
- CD : カウントダウン

カウンタ内のビットコンフィグレーション

0～999 の範囲内の数値を入力することで、カウンタに事前設定値を指定できます。たとえば 127 の場合は、C#127 のように入力します。C#は 2 進化 10 進数形式を表します(BCD 形式: 4 ビットのセットごとに、10 進値の 2 進コードが 1 つ格納されています)。

0 番から 11 番のカウンタビットは、BCD 表記でカウンタ値を表します。

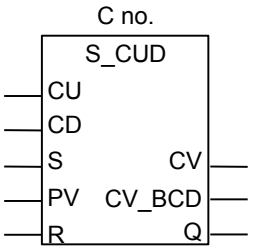
次の図に、カウント値 127 をロードした後のカウンタの内容と、カウンタが設定された後のカウンタセルの内容を示します。



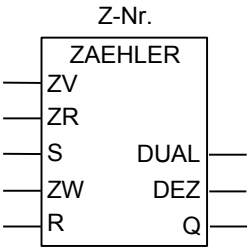
4.2 S_CUD : パラメータおよびカウントアップ/カウントダウンの割り付け

シンボル

English



German



パラメータ 英語	パラメータ ドイツ語	データタイプ	メモリ領域	説明
no.	Nr.	COUNTER	C	カウンタ番号 値の範囲は、CPU により異なる。
CU	ZV	BOOL	I、Q、M、D、L	ZV 入力:カウントアップ
CD	ZR	BOOL	I、Q、M、D、L	ZR 入力:カウントダウン
S	S	BOOL	I、Q、M、D、L、T、C	カウンタ事前設定のための設定入力
PV	ZW	WORD	I、Q、M、D、L または定数	0～999 のカウンタ値 または C#<value> として BCD フォーマットで入力されるカウンタ値
R	R	BOOL	I、Q、M、D、L、T、C	リセット入力
CV	DUAL	WORD	I、Q、M、D、L	現在のカウンタ値(16 進数)
CV_BCD	DEZ	WORD	I、Q、M、D、L	現在のカウンタ値(BCD フォーマット)
Q	Q	BOOL	I、Q、M、D、L	カウンタのステータス

説明

パラメータおよびカウントアップ/カウントダウンの割り付け命令の入力 S に信号立ち上がり(信号状態の 0 から 1 への遷移)がある場合、プリセット値(PV)入力の値でカウンタが設定されます。入力 CU の信号状態が 0 から 1 に遷移し(信号立ち上がりエッジ)、カウンタの値が 999 より小さい場合、カウンタ値は 1 ずつ増分されます。入力 CD の信号状態が 0 から 1 に遷移し(信号立ち上がりエッジ)、カウンタの値が 0 より大きい場合、カウンタ値は 1 ずつ減分されます。両方のカウンタ入力に信号立ち上がりエッジがある場合、両方の動作が実行されてカウンタ値は同じままになります。

カウンタが設定され、入力 CU/CD が RLO = 1 の場合は、信号立ち上がりから信号立ち下がりへ(またはこの逆)の変更がない場合でも、カウンタは**次**のスキャンサイクルで 1 度カウントを行います。

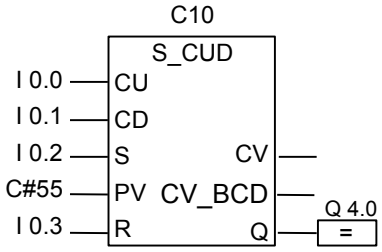
入力 R に"1"が入っている場合、カウンタはリセットされます。カウンタは、リセットするとゼロに設定されます。

カウンタ値が 1 よりも大きい場合、出力 Q の信号状態チェックで値 1 が返されます。カウンタ値が 0 の場合は、値 0 が返されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例



入力 I0.2 で、信号状態が 0 から 1 に変わると、カウンタ C10 の値が 55 になります。I0.0 の信号状態が 0 から 1 に変わると、カウンタ C10 の値が 1 ずつ増えます。ただし、C10 の値がすでに 999 のときは例外です。入力 I0.1 が 0 から 1 になると、カウンタ C10 は 1 ずつ減ります。ただし、C10 の値がすでに 0 のときは例外です。I0.3 が 0 から 1 に変わると、C10 の値が 0 に設定されます。C 10 が 0 でない場合、Q4.0 は 1 になります。

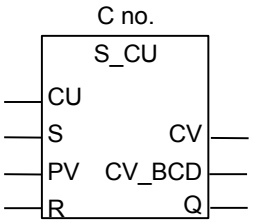
注記

複数のプログラムポイントでカウンタを使用しないでください(カウントエラーが発生する危険があります)。

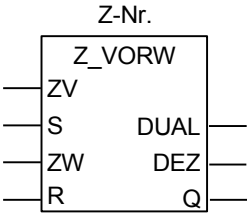
4.3 S_CU : パラメータおよびカウントアップの割り付け

シンボル

English



German



パラメータ 英語	パラメータ ドイツ語	データタイプ	メモリ領域	説明
no.	Nr.	COUNTER	C	カウンタ番号 値の範囲は、CPU により異なる。
CU	ZV	BOOL	I、Q、M、D、L	ZV 入力:カウントアップ
S	S	BOOL	I、Q、M、D、L、T、C	カウンタ事前設定のための設定入力
PV	ZW	WORD	I、Q、M、D、L または定数	0～999 のカウンタ値 または C#<value> として BCD フォーマットで入力されるカウンタ値
R	R	BOOL	I、Q、M、D、L、T、C	リセット入力
CV	DUAL	WORD	I、Q、M、D、L	現在のカウンタ値(16 進数)
CV_BCD	DEZ	WORD	I、Q、M、D、L	現在のカウンタ値(BCD フォーマット)
Q	Q	BOOL	I、Q、M、D、L	カウンタのステータス

説明

パラメータおよびカウントアップの割り付け命令の入力 S に信号立ち上がり(信号状態の 0 から 1 への遷移)がある場合、プリセット値(PV)入力の値でカウンタが設定されます。入力 CU に信号立ち上がりがある場合、カウンタ値が 999 未満であれば、カウンタ値が 1 インクリメントされます。

カウンタが設定され、入力 CU が RLO = 1 の場合は、信号立ち上がりから信号立ち下がりへ(またはこの逆)の変更がない場合でも、カウンタは**次**のスキャンサイクルで 1 度カウントを行います。

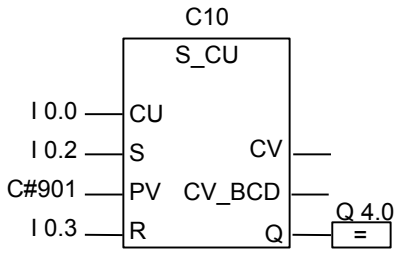
入力 R に"1"が入っている場合、カウンタはリセットされます。カウンタは、リセットするとゼロに設定されます。

カウンタ値が 1 よりも大きい場合、出力 Q の信号状態チェックで値 1 が返されます。カウンタ値が 0 の場合は、値 0 が返されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例



入力 I0.2 で、信号状態が 0 から 1 に変わると、カウンタ C10 の値が 901 になります。I0.0 の信号状態が 0 から 1 に変わると、カウンタ C10 の値が 1 ずつ増えます。ただし、C10 の値が 999 のときは例外です。I0.3 が 0 から 1 に変わると、C10 の値が 0 に設定されます。C10 が 0 以外の場合、Q4.0 は 1 になります。

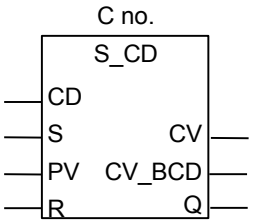
注記

複数のプログラムポイントでカウンタを使用しないでください(カウントエラーが発生する危険があります)。

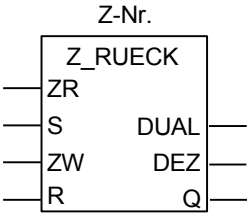
4.4 S_CD : パラメータおよびカウントダウンの割り付け

シンボル

English



German



パラメータ 英語	パラメータ ドイツ語	データタイプ	メモリ領域	説明
no.	Nr.	COUNTER	C	カウンタ番号 値の範囲は、CPU により異なる。
CD	ZR	BOOL	I、Q、M、D、L	CD 入力: カウントダウン
S	S	BOOL	I、Q、M、D、L、T、C	カウンタ事前設定のための設定入力
PV	ZW	WORD	I、Q、M、D、L または定数	0～999 のカウンタ値 または C#<value> として BCD フォーマットで入力されるカウンタ値
R	R	BOOL	I、Q、M、D、L、T、C	リセット入力
CV	DUAL	WORD	I、Q、M、D、L	現在のカウンタ値(16 進数)
CV_BCD	DEZ	WORD	I、Q、M、D、L	現在のカウンタ値(BCD フォーマット)
Q	Q	BOOL	I、Q、M、D、L	カウンタのステータス

説明

パラメータおよびカウントアップ/カウントダウンの割り付け命令の入力 S に信号立ち上がり(信号状態の 0 から 1 への遷移)がある場合、プリセット値入力(PV)でカウンタが設定されます。カウンタ値が 0 より大きい場合、入力 CD に立ち上がりパルスが発生すると、カウンタは 1 減少します。

カウンタが設定され、入力 CD が RLO = 1 の場合は、信号立ち上がりから信号立ち下がりへ(またはこの逆)の変更がない場合でも、カウンタは**次**のスキャンサイクルで 1 度カウントを行います。

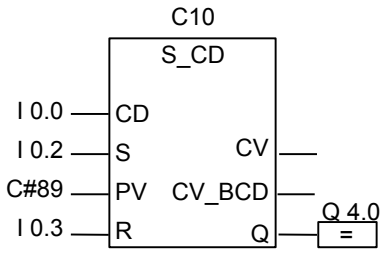
入力 R に"1"が入っている場合、カウンタはリセットされます。カウンタは、リセットするとゼロに設定されます。

カウンタ値が 1 よりも大きい場合、出力 Q の信号状態チェックで値 1 が返されます。カウンタ値が 0 の場合は、値 0 が返されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例



入力 I0.2 で、信号状態が 0 から 1 に変わると、カウンタ C10 の値が 89 になります。I0.0 の信号状態が 0 から 1 に変わると、カウンタ C10 の値が 1 ずつ増えます。ただし、C10 の値が 0 のときは例外です。C10 が 0 以外の場合、出力 Q4.0 の信号状態は 1 になります。I0.3 が 0 から 1 に変わると、C10 の値が 0 に設定されます。

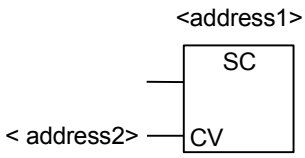
注記

複数のプログラムポイントでカウンタを使用しないでください(カウントエラーが発生する危険があります)。

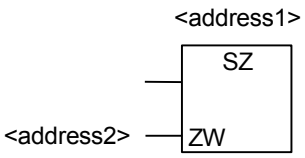
4.5 SC：カウンタ値の設定

シンボル

English



German



パラメータ 英語	パラメータ ドイツ語	データタイプ	メモリ領域	説明
カウンタ番号	カウンタ番号	COUNTER	C	アドレス 1 は、値をプリセットするカウンタの番号を表す。
CV	ZW	WORD	I、Q M、D、L、または定数	事前設定できる値(アドレス 2)の範囲は 0～999 です。定数を入力するときには、BCD フォーマットにより、指定された値の前に C#が付かなければなりません (例: C#100)。

説明

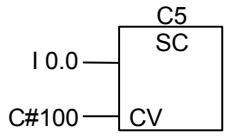
カウンタ値の設定命令により、指定したカウンタにプリセット値を割り付けることができます。この命令は、RLO に信号立ち上がり(信号状態の 0 から 1 への遷移)がある場合に限り実行されます。

カウンタ値の設定ボックスは、論理文字列の右端にしか配置することができません。**カウンタ値の設置**ボックスの番号を使用できます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	-	-	0

例



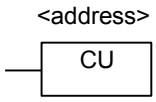
入力 I0.0 の信号状態が 0 から 1 に遷移した(RLO に信号立ち上がりがある)場合、カウンタ C5 は値 100 でプリセットされます。C#は BCD フォーマットで入力した値を示します。

信号立ち上がりがない場合、C5 のカウンタ値は変わりません。

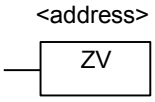
4.6 CU : カウントアップ

シンボル

English



German



パラメータ	データタイプ	メモリ領域	説明
カウンタ番号	COUNTER	C	アドレスは、インクリメントされるカウンタの番号を表す。

説明

カウントアップ命令は、RLO に信号立ち上がりエッジ(信号状態の 0 から 1 への遷移)があり、カウンタ値が 999 より小さい場合、指定したカウンタ値を 1 ずつ増分します。RLO に信号立ち上がりエッジがないか、またはカウンタ値がすでに 999 になっている場合は、増分されません。

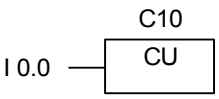
カウンタ値の設定命令は、カウンタの値を設定します。

カウントアップボックスは論理文字列の右端にしか配置することができません。**カウントアップ**ボックスの番号を使用できます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	-	-	0

例



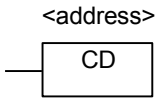
入力 I0.0 の信号状態が 0 から 1 に遷移した(RL0 に信号立ち上がりがある)場合、カウンタ C10 の値が 1 インクリメントされます(ただし、C10 の値が 999 である場合は除きます)。

立ち上がりエッジがない場合、C10 の値は変更されません。

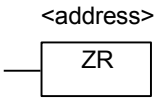
4.7 CD : カウントダウン

シンボル

English



German



パラメータ	データタイプ	メモリ領域	説明
カウンタ番号	COUNTER	C	アドレスはデクリメントされるカウンタの番号を表す。

説明

カウントダウン命令は、RLO に信号立ち上がりエッジ(信号状態の 0 から 1 への遷移)があり、カウンタ値が 0 より大きい場合、指定したカウンタ値を 1 ずつ減分します。RLO に信号立ち上がりエッジがないか、またはカウンタ値がすでに 0 になっている場合は、減分されません。

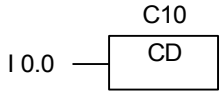
カウンタ値の設定命令は、カウンタの値を設定します。

カウントダウンボックスは論理文字列の右端にしか配置することができません。**カウントダウン**ボックスの番号を使用できます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	-	-	0

例



入力 I0.0 の信号状態が 0 から 1 に遷移した(RL0 に信号立ち上がりがある)場合、カウンタ C10 の値が 1 デクリメントされます(ただし、C10 の値が 0 である場合は除きます)。

立ち上がりエッジがない場合、C10 の値は変更されません。

5 データブロック命令

5.1 OPN：データブロックを開く

シンボル

<DB-Number> or
<DI-Number>

OPN

パラメータ	データタイプ	メモリ領域	説明
DB または DI の番号	BLOCK_DB	-	DB または DI の番号の範囲は CPU によって異なります。

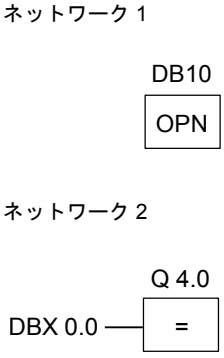
説明

データブロックを開く命令を使用すれば、既存のデータブロックを共有データブロック(DB)またはインスタンスデータブロック(DI)として開くことができます。このデータブロックの番号は、DB レジスタまたは DI レジスタへ転送されます。これに続く DB、DI コマンドが、レジスタの内容に応じて、対応するブロックにアクセスします。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	-	-	-	-

例



DB10 は現在開かれているデータブロックです。従って、DBX0.0 では、データブロック DB10 4.0 のデータバイト 0 のビット 0 がチェックされます。このビットの信号状態は、出力 Q 4.0 に割り付けられます。

6 ジャンプ命令

6.1 ジャンプ命令の概要

説明

この命令は、オーガニゼーションブロック(OB)、ファンクションブロック(FB)、ファンクション(FC)など、すべてのロジックブロックで使用できます。

使用可能なジャンプ命令を次に示します。

- JMP　ブロック内での条件なしジャンプ
- JMP　ブロック内での条件付きジャンプ
- JMPN　ジャンプイフノット

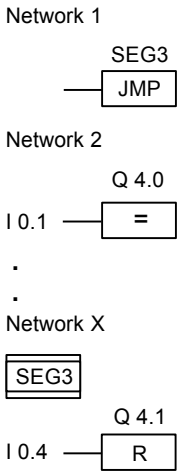
アドレスを示すジャンプラベル

ジャンプ命令のアドレスがラベルです。ジャンプラベルは、プログラムの宛先を示します。

JMP ボックスの上にラベルを入力します。ラベルには最大 4 文字が含まれます。先頭には必ず英字を使用しますが、それ以外は英字または数字を使用できます(たとえば、SEG3)。

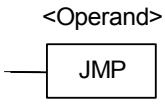
宛先を示すジャンプラベル

宛先ラベルは、ネットワークの先頭で入力する必要があります。宛先ラベルを入力するには、FBD リストボックスで LABEL を選択します。空のボックスが表示されるので、ここにラベル名を入力します。



6.2 JMP：ブロック内での条件なしジャンプ

シンボル



パラメータ	データタイプ	メモリ領域	説明
ジャンプラベルの名前	-	-	このアドレスは、プログラムの無条件ジャンプの宛先となるラベルを指定します。

説明

ブロック内での条件なしジャンプ命令は、"ラベルへの移動"命令に対応します。ジャンプ操作とラベル間の命令は一切実行されません。

この命令は、オーガニゼーションブロック(OB)、ファンクションブロック(FB)、ファンクション(FC)など、すべてのロジックブロックで使用できます。

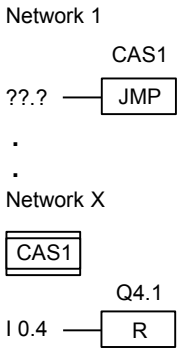
[**ブロック内での条件なしジャンプ**]ボックスの前には、どの論理演算も使用できません。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	-	-	-	-

この命令を実行しても、ステータスワードのビットは変化しません。

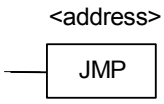
例



ジャンプは、常に実行されます。ジャンプ命令とラベルの間にある命令は実行されません。

6.3 JMP：ブロック内での条件付きジャンプ

シンボル



パラメータ	データタイプ	メモリ領域	説明
ジャンプラベルの名前	-	-	このアドレスは、RLO が 1 のときにプログラムのジャンプ先となるラベルを指定します。

説明

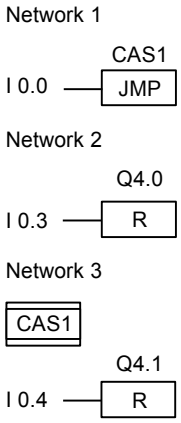
ブロック内での条件付きジャンプ命令は、RLO が 1 の場合に、「ラベルへの移動」命令に対応します。この演算にも、FBD 要素「条件なしジャンプ」が使用されますが、1 つ前の論理演算によって条件が付与されます。条件付きジャンプは、この論理演算の結果が 1 の場合にだけ実行されます。ジャンプ操作とラベルとの間では、命令は実行されません。

この命令は、オーガニゼーションブロック(OB)、ファンクションブロック(FB)、ファンクション(FC)など、すべてのロジックブロックで使用できます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	1	1	0

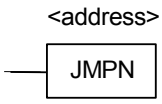
例



入力 I0.0 の信号状態が 1 になると、ラベル CAS1 へのジャンプが実行されます。入力 I0.3 の信号状態が 1 になっても、出力 Q4.0 をリセットする命令は実行されません。

6.4 JMPN：ジャンプイフノット

シンボル



パラメータ	データタイプ	メモリ領域	説明
ジャンプラベルの名前	-	-	このアドレスは、RLO が 0 のときにプログラムのジャンプ先となるラベルを指定します。

説明

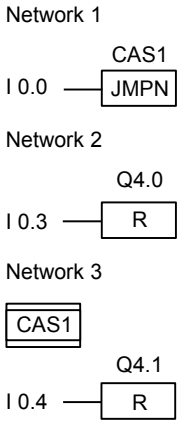
ジャンプイフノット命令は、RLO が 0 の場合に実行される"ラベルへの移動"命令に対応します。

この命令は、オーガニゼーションブロック(OB)、ファンクションブロック(FB)、ファンクション(FC)など、すべてのロジックブロックで使用できます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	1	1	0

例



入力 I0.0 の信号状態が 0 の場合、ラベル CAS1 へのジャンプが実行されます。入力 I0.3 の信号状態が 1 であっても、出力 Q4.0 をリセットする命令は実行されません。

ジャンプ操作とラベルとの間では、命令は実行されません。

6.5 LABEL：ジャンプラベル

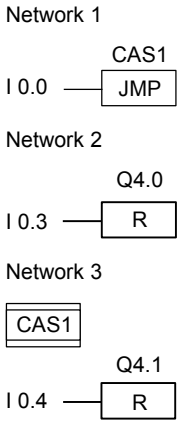
シンボル

説明

ジャンプラベルは、ジャンプ命令の宛先の識別子です。ラベルには最大 4 文字が含まれます。先頭には必ず英字を使用しますが、それ以外は英字または数字を使用できます(例:CAS1)。

ジャンプラベルは、ジャンプ命令またはジャンプイフノット命令(**JMP** または **JMPN**)ごとに必要となります。

例



I0.0 ＝1 の場合、ラベル CAS1 へのジャンプが実行されます。

ジャンプが実行されるため、I0.3 が 1 になっても Q4.0 での"出力リセット"演算は実行されません。

7 整数演算命令

7.1 整数演算命令の概要

説明

整数演算では、**2 つの整数**(16 ビットおよび 32 ビット)を使用して次の演算を実行することができます。

- ADD_I : 整数の加算
- SUB_I : 整数の減算
- MUL_I : 整数の乗算
- DIV_I : 整数の除算
- ADD_DI : 倍長整数の加算
- SUB_DI : 倍長整数の減算
- MUL_DI : 倍長整数の乗算
- DIV_DI : 倍長整数の除算
- MOD_DI : 倍長整数の剰余

関連項目: 整数演算命令によるステータスワードのビットの評価

7.2 整数演算命令によるステータスワードのビットの評価

説明

整数演算命令は、ステータスワード内の CC1 と CC0、OV と OS の各ビットに影響します。

以下の表に、整数(16 ビットと 32 ビット)を使用した命令の結果に対応するステータスワードのビットの信号状態を示します。

結果の有効範囲	CC 1	CC 0	OV	OS
0	0	0	0	*
16 ビット: -32 768 <= 結果 < 0 (負数) 32 ビット: -2 147 483 648 <=結果 < 0 (負数)	0	1	0	*
16 ビット: 32 767 >= 結果 > 0 (正数) 32 ビット: 2 147 483 647 >= 結果 > 0 (正数)	1	0	0	*

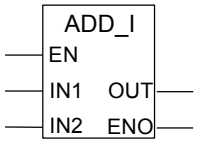
* OS ビットは、この命令結果による影響を受けません。

無効な結果範囲	A1	A0	OV	OS
アンダーフロー(加算) 16 ビット: 結果 = -65536 32 ビット: 結果 = -4 294 967 296	0	0	1	1
アンダーフロー(乗算) 16 ビット: 結果 < -32 768 (負数) 32 ビット: 結果 < -2 147 483 648 (負数)	0	1	1	1
オーバーフロー(加算、減算) 16 ビット: 結果 > 32 767 (正数) 32 ビット: 結果 > 2 147 483 647 (正数)	0	1	1	1
オーバーフロー(乗算、除算) 16 ビット: 結果 > 32 767 (正数) 32 ビット: 結果 > 2 147 483 647 (正数)	1	0	1	1
アンダーフロー(加算、減算) 16 ビット: 結果 < -32 768 (負数) 32 ビット: 結果 < -2 147 483 648 (負数)	1	0	1	1
0 による除算	1	1	1	1

操作	A1	A0	OV	OS
+D: 結果 = -4 294 967 296	0	0	1	1
/D または MOD: 0 による除算	1	1	1	1

7.3 ADD_I : 整数の加算

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	INT	I、Q、M、D、L または定数	加算の最初の値
IN2	INT	I、Q、M、D、L または定数	加算の 2 番目の値
OUT	INT	I、Q、M、D、L	加算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

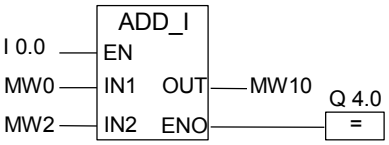
イネーブル(EN)入力の信号状態が 1 になると、**整数の加算** 命令が有効になり、この命令は、入力 IN1 と IN2 を加算します。この命令の結果は OUT で確認できます。もし結果が整数の許容範囲外にあると、ステータスワードの OV と OS ビットは 1 になり、ENO は 0 になります。

関連項目: 整数演算命令によるステータスワードのビットの評価

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

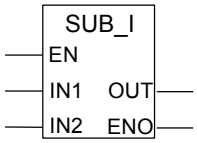
例



入力 I0.0 の信号状態が 1 になると、ADD_I ボックスが有効になります。加算 MW0+MW2 の結果は、メモリのワード MW10 に入力されます。この結果が整数の許容範囲を超えているか、入力 I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定され、命令が実行されません。

7.4 SUB_I : 整数の減算

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	INT	I、Q、M、D、L または定数	被減数(2 番目の値が引かれる値)
IN2	INT	I、Q、M、D、L または定数	減数(1 番目の値から引く値)
OUT	INT	I、Q、M、D、L	減算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

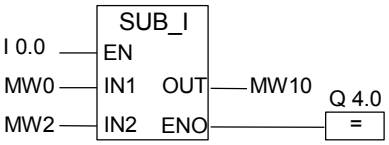
イネーブル(EN)入力の信号状態が 1 になると、**整数の減算**命令が有効になります。この命令は、入力 IN1 から IN2 を減算します。この命令の結果は OUT で確認できます。減算結果が整数の許容範囲外である場合、ステータスワードの OV ビットと OS ビットは 1 になり、ENO は 0 になります。

関連項目: 整数演算命令によるステータスワードのビットの評価

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

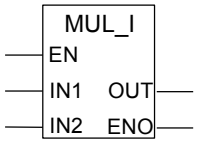
例



入力 I0.0 の信号状態が 1 になると、SUB_I ボックスが有効になります。減算 MW0 - MW2 の結果は、メモリワード MW10 に入力されます。この結果が整数の許容範囲を超えているか、入力 I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定され、命令が実行されません。

7.5 MUL_I : 整数の乗算

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	INT	I、Q、M、D、L または定数	被乗数(2 番目の値が掛けられる値)
IN2	INT	I、Q、M、D、L または定数	乗数(1 番目の値に掛ける値)
OUT	INT	I、Q、M、D、L	乗算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

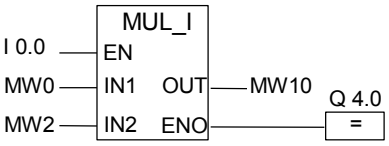
イネーブル(EN)入力の信号状態が 1 になると、**整数の乗算**命令が有効になり、この命令は、IN1 を IN2 で乗算します。この結果は 32 ビットの整数になり、OUT で確認できます。計算結果が 16 ビット整数の許容範囲外である場合、ステータスワードの OV ビットと OS ビットは 1 になり、ENO は 0 になります。

関連項目: 整数演算命令によるステータスワードのビットの評価

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

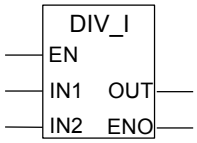
例



入力 I0.0 の信号状態が 1 になると、MUL_I ボックスが有効になります。乗算 MW0 x MW2 の結果は、メモリワード MW10 に入力されます。この結果が 16 ビット整数の許容範囲を超えているか、入力 I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定され、命令が実行されません。

7.6 DIV_I : 整数の除算

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	INT	I、Q、M、D、L または定数	被除数
IN2	INT	I、Q、M、D、L または定数	割る値
OUT	INT	I、Q、M、D、L	除算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

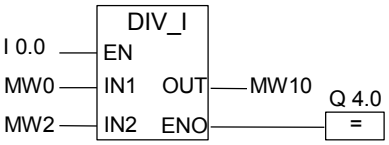
イネーブル(EN)入力の信号状態が 1 になると、**整数の除算**命令が有効になり、この命令は、IN1 を IN2 で除算します。整数の商(切り捨てられた結果)は、OUT で確認できます。商余は出力されません。商が整数の許容範囲外にある場合、ステータスワードの OV ビットと OS ビットは 1 となり、ENO は 0 です。

関連項目: 整数演算命令によるステータスワードのビットの評価

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

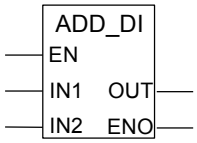
例



入力 I0.0 の信号状態が 1 になると、DIV_I ボックスが有効になります。除算 MW0÷MW2 の商は、メモリワード MW10 に入力されます。この商が整数の許容範囲を超えているか、入力 I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定され、命令が実行されません。

7.7 ADD_DI : 倍長整数の加算

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	DINT	I、Q、M、D、L または定数	加算の最初の値
IN2	DINT	I、Q、M、D、L または定数	加算の 2 番目の値
OUT	DINT	I、Q、M、D、L	加算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

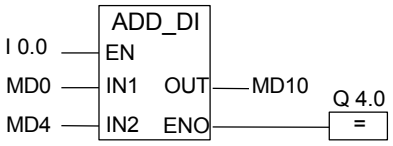
イネーブル(EN)入力の信号状態が 1 になると、**倍長整数の加算**命令が有効になり、この命令は、入力 IN1 と IN2 を加算します。この命令の結果は OUT で確認できます。加算結果が倍長整数の許容範囲外である場合、ステータスワードの OV ビットと OS ビットは 1 になり、ENO は 0 になります。

関連項目: 整数演算命令によるステータスワードのビットの評価

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

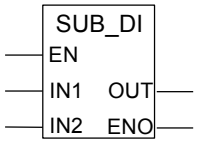
例



入力 I0.0 の信号状態が 1 になると、ADD_DI ボックスが有効になります。加算 MW0 + MD4 の結果は、メモリのダブルワード MD10 に入力されます。この結果が倍長整数の許容範囲を超えているか、入力 I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定され、命令が実行されません。

7.8 SUB_DI : 倍長整数の減算

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	DINT	I、Q、M、D、L または定数	被減数(2 番目の値が引かれる値)
IN2	DINT	I、Q、M、D、L または定数	減数(1 番目の値から引く値)
OUT	DINT	I、Q、M、D、L	減算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

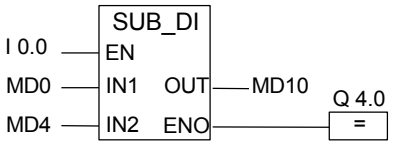
イネーブル(EN)入力の信号状態が 1 になると、**倍長整数の減算**命令が有効になり、この命令は、入力 IN1 から IN2 を減算します。この命令の結果は OUT で確認できます。結果が倍長整数の許容範囲外である場合、ステータスワードの OV ビットと OS ビットは 1 になり、ENO は 0 になります。

関連項目: 整数演算命令によるステータスワードのビットの評価

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

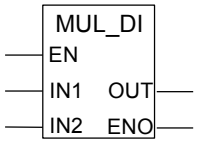
例



入力 I0.0 の信号状態が 1 になると、SUB_DI ボックスが有効になります。減算 MD0 - MD4 の結果は、メモリダブルワード MD10 に入力されます。この結果が倍長整数の許容範囲を超えているか、入力 I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定され、命令が実行されません。

7.9 MUL_DI : 倍長整数の乗算

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	DINT	I、Q、M、D、L または定数	被乗数(2 番目の値が掛けられる値)
IN2	DINT	I、Q、M、D、L または定数	乗数(1 番目の値に掛ける値)
OUT	DINT	I、Q、M、D、L	乗算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

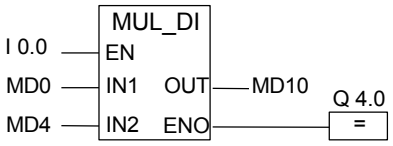
イネーブル(EN)入力の信号状態が 1 になると、**倍長整数の乗算**命令が有効になり、この命令は、入力 IN1 と IN2 を乗算します。この命令の結果は OUT で確認できます。結果が倍長整数の許容範囲外である場合、ステータスワードの OV ビットと OS ビットは 1 になり、ENO は 0 になります。

関連項目: 整数演算命令によるステータスワードのビットの評価

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

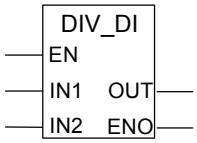
例



入力 I0.0 の信号状態が 1 になると、MUL_DI ボックスが有効になります。乗算 MD0 x MD4 の結果は、メモリダブルワード MD10 に入力されます。この結果が倍長整数の許容範囲を超えているか、入力 I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定され、命令が実行されません。

7.10 DIV_DI : 倍長整数の除算

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	DINT	I、Q、M、D、L または定数	被除数
IN2	DINT	I、Q、M、D、L または定数	割る値
OUT	DINT	I、Q、M、D、L	除算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

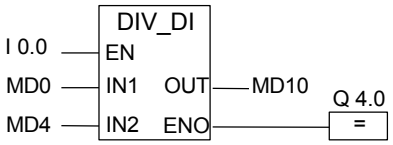
イネーブル(EN)入力の信号状態が 1 になると、**倍長整数の除算**命令が有効になり、この命令は、IN1 を IN2 で除算します。整数の商は OUT で見ることが出来ます。倍長整数の除算命令では、商は DINT フォーマットで 1 つの 32 ビット値として格納されます。なお、この命令では、剰余は生成されません。商が倍長整数の許容範囲外である場合、ステータスワードの OV ビットと OS ビットは 1 になり、ENO は 0 になります。

関連項目: 整数演算命令によるステータスワードのビットの評価

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

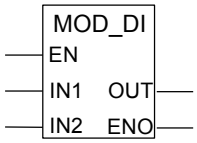
例



入力 I0.0 の信号状態が 1 になると、DIV_DI ボックスが有効になります。除算 MD0÷MD4 の結果は、メモリダブルワード MD10 に入力されます。この商が倍長整数の許容範囲を超えているか、入力 I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定され、命令が実行されません。

7.11 MOD_DI : 倍長整数の剰余

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	DINT	I、Q、M、D、L または定数	被除数
IN2	DINT	I、Q、M、D、L または定数	割る値
OUT	DINT	I、Q、M、D、L	剰余
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

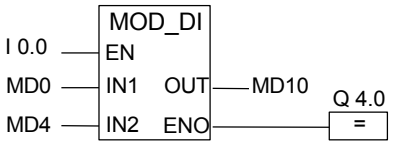
イネーブル(EN)入力の信号状態が 1 になると、**余り(倍長整数)の表示**命令が有効になります。この命令は、IN1 を IN2 で除算します。商余は OUT で表示されます。結果が倍長整数の許容範囲外である場合、ステータスワードの OV ビットと OS ビットは 1 になり、ENO は 0 になります。

関連項目: 整数演算命令によるステータスワードのビットの評価

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

例



入力 I0.0 の信号状態が 1 になると、MOD_DI ボックスが有効になります。MD0 を MD4 で除算した余り(商余)は、メモリダブルワード MD10 に格納されます。この結果が倍長整数の許容範囲を超えているか、入力 I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定され、命令が実行されません。

8 浮動小数点数値演算命令

8.1 浮動小数点数値演算命令の概要

説明

IEEE 32 ビット浮動小数点数は、REAL と呼ばれるデータタイプに属します。浮動小数点数値演算命令を使用し、32 ビット IEEE 浮動小数点数を **2** つ使用すれば、以下の数値演算命令を実行できます。

- ADD_R : 実数の加算
- SUB_R : 実数の減算
- MUL_R : 実数の乗算
- DIV_R : 実数の除算

1 つの 32 ビット IEEE 浮動小数点数を使用した場合は、次の演算を実行することができます。

- 浮動小数点数の絶対値(ABS)を求める
- 浮動小数点数の平方(SQR)または平方根(SQRT)を求める
- 浮動小数点数の自然対数(LN)を求める
- 基数 e(= 2.71828...)に対する浮動小数点数の指数値を求める
- 次の角の三角関数を 32 ビットの浮動小数点数で求める
  - サイン(SIN)およびアークサイン(ASIN)
  - コサイン(COS)およびアークコサイン(ACOS)
  - タンジェント(TAN)およびアークタンジェント(ATAN)

関連項目: 浮動小数点命令によるステータスワードのビットの評価

8.2 浮動小数点命令によるステータスワードのビットの評価

説明

浮動小数点命令は、ステータスワード内の CC 1 と CC 0、OV と OS の各ビットに影響します。

以下の表に、浮動小数点数(32 ビット)を使用した命令の結果に対応するステータスワードのビットの信号状態を示します。

結果の有効範囲	CC 1	CC 0	OV	OS
＋0, －0(ゼロ)	0	0	0	*
-3.402823E+38 < 結果 < -1.175494E-38 (負数)	0	1	0	*
+1.175494E-38 < 結果 < 3.402824E+38 (正数)	1	0	0	*

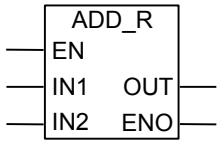
* OS ビットは、この命令結果による影響を受けません。

無効な結果範囲	CC 1	CC 0	OV	OS
アンダーフロー -1.175494E-38 < 結果 < - 1.401298E-45 (負数)	0	0	1	1
アンダーフロー +1.401298E-45 < 結果< +1.175494E-38 (正数)	0	0	1	1
オーバーフロー 結果< -3.402823E+38 (負の数)	0	1	1	1
オーバーフロー 結果> 3.402823E+38 (正数)	1	0	1	1
無効な実数または不正な命令 (有効範囲にない入力値)	1	1	1	1

8.3 基本命令

8.3.1 ADD_R : 実数の加算

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	REAL	I、Q、M、D、L または定数	加算される 1 番目の数
IN2	REAL	I、Q、M、D、L または定数	加算される 2 番目の数
OUT	REAL	I、Q、M、D、L	加算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

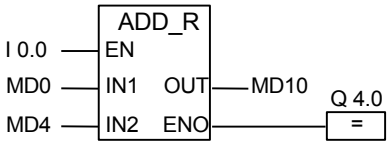
イネーブル入力(EN)の信号状態が 1 になると、**実数の加算**命令が有効になり、この命令は、入力 IN1 と IN2 を加算します。この命令の結果は出力 OUT で確認できます。入力か結果のいずれかが浮動小数点数でない場合は、OV ビットと OS ビットが 1 に設定され、ENO が 0 に設定されます。

関連項目: 浮動小数点命令によるステータスワードのビットの評価

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

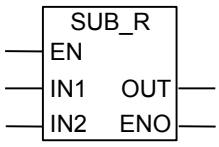
例



入力 I0.0 の信号状態が 1 になると、ADD_R ボックスが実行されます。加算 MW0 + MD4 の結果は、メモリのダブルワード MD10 に入力されます。入力か結果のいずれかが浮動小数点数でない場合で、なおかつ I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定され、命令が実行されません。

8.3.2 SUB_R : 実数の減算

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	REAL	I、Q、M、D、L または定数	被減数(この数値から 2 番目の値を引く)
IN2	REAL	I、Q、M、D、L または定数	減数(1 番目の値からこの値を引く)
OUT	REAL	I、Q、M、D、L	減算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

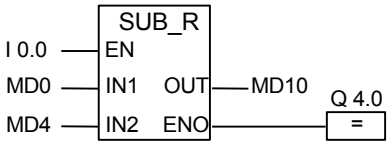
イネーブル入力(EN)の信号状態が 1 になると、**実数の減算**命令が有効になり、この命令は、入力 IN1 から IN2 を減算します。この命令の結果は出力 OUT で確認できます。入力か結果のいずれかが浮動小数点数でない場合は、OV ビットと OS ビットが 1 に設定され、ENO が 0 に設定されます。

関連項目: 浮動小数点命令によるステータスワードのビットの評価

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

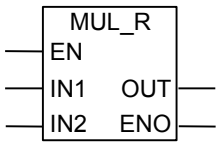
例



入力 I0.0 の信号状態が 1 になると、SUB_R ボックスが実行されます。減算 MD0 - MD4 の結果は、メモリダブルワード MD10 に入力されます。入力か結果のいずれかが浮動小数点数でない場合で、なおかつ I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定され、命令が実行されません。

8.3.3 MUL_R : 実数の乗算

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	REAL	I、Q、M、D、L または定数	被乗数(乗算される数)
IN2	REAL	I、Q、M、D、L または定数	乗数(1 番目の数に乗算する数)
OUT	REAL	I、Q、M、D、L	乗算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

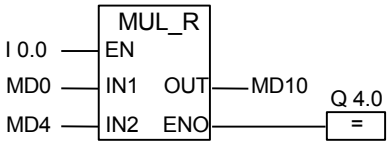
イネーブル入力(EN)の信号状態が 1 になると、**実数の乗算**命令が有効になります。この命令は、IN1 を IN2 で乗算します。この命令の結果は出力 OUT で確認できます。入力か結果のいずれかが浮動小数点数でない場合は、OV ビットと OS ビットが 1 に設定され、ENO が 0 に設定されます。

関連項目: 浮動小数点命令によるステータスワードのビットの評価

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

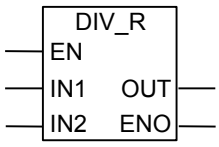
例



入力 I0.0 の信号状態が 1 になると、MUL_R ボックスが実行されます。乗算 MD0 x MD4 の結果は、メモリダブルワード MD10 に入力されます。入力か結果のいずれかが浮動小数点数でない場合で、なおかつ I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定され、命令が実行されません。

8.3.4 DIV_R : 実数の除算

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	REAL	I、Q、M、D、L または定数	被除数(2 番目の数で除算される数)
IN2	REAL	I、Q、M、D、L または定数	除数(1 番目の数を除算する数)
OUT	REAL	I、Q、M、D、L	除算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

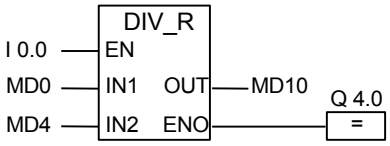
イネーブル入力(EN)の信号状態が 1 になると、**実数の除算**命令が有効になります。この命令は、IN1 を IN2 で除算します。この命令の結果は出力 OUT で確認できます。入力か結果のいずれかが浮動小数点数でない場合は、OV ビットと OS ビットが 1 に設定され、ENO が 0 に設定されます。

関連項目: 浮動小数点命令によるステータスワードのビットの評価

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

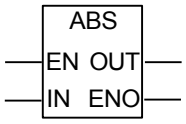
例



入力 I0.0 の信号状態が 1 になると、DIV_R ボックスが実行されます。MD0 を MD4 で割る除算の結果は、メモリダブルワード MD10 に出力されます。入力か結果のいずれかが浮動小数点数でない場合で、なおかつ I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定され、命令が実行されません。

8.3.5 ABS：浮動小数点数の絶対値を求める

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	入力値: 浮動小数点数
OUT	REAL	I、Q、M、D、L	出力値: 浮動小数点数の絶対値
ENO	BOOL	I、Q、M、D、L	イネーブル出力

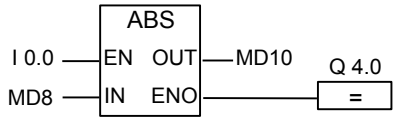
説明

浮動小数点数の絶対値を求める命令では、**浮動小数点数の絶対値を求めることができます。**

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	-	-	-	-	0	X	X	1

例



I0.0 = 1 の場合、MD8 の絶対値が MD12 に出力されます。

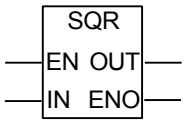
MD8 = +6.234 の結果は、MD12 = 6.234 x 1 になります。

変換が実行されないと、出力 Q4.0 は 0 になります(ENO = EN = 0)。

8.4 拡張命令

8.4.1 SQR:浮動小数点数の平方を求める

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	番号
OUT	REAL	I、Q、M、D、L	IN の値の平方
ENO	BOOL	I、Q、M、D、L	イネーブル出力

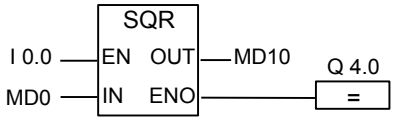
説明

浮動小数点数の平方を求める命令では、浮動小数点数の平方を求めることができます。入力か結果のいずれかが浮動小数点数でない場合は、OV ビットと OS ビットが 1 に設定され、ENO が 0 に設定されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

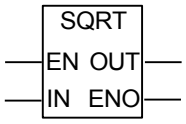
例



入力 I0.0 の信号状態が 1 になると、SQR ボックスが実行されます。結果は、メモリダブルワード MD10 に出力されます。MD0 が 0 未満の場合か、入力か結果のいずれかが浮動小数点数でない場合で、なおかつ I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定されます。

8.4.2 SQRT : 浮動小数点数の平方根を求める

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	番号
OUT	REAL	I、Q、M、D、L	数値の平方根
ENO	BOOL	I、Q、M、D、L	イネーブル出力

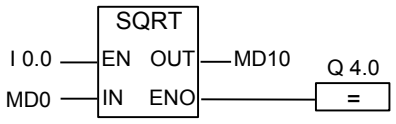
説明

浮動小数点数の平方根を求める命令では、浮動小数点数の平方根を求めることができます。アドレスの値が"0"より大きい場合、この命令は正の結果を返します。入力か結果のいずれかが浮動小数点数でない場合は、OV ビットと OS ビットが 1 に設定され、ENO が 0 に設定されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

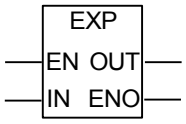
例



入力 I0.0 の信号状態が 1 になると、SQRT ボックスが実行されます。SQRT(MD0)の結果は、メモリダブルワード MD10 に出力されます。MD0 が 0 未満の場合か、入力か結果のいずれかが浮動小数点数でない場合で、なおかつ I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定されます。

8.4.3 EXP：浮動小数点数の指数値を求める

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	番号
OUT	REAL	I、Q、M、D、L	数値の指数
ENO	BOOL	I、Q、M、D、L	イネーブル出力

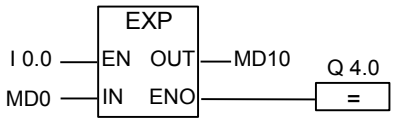
説明

浮動小数点数の指数値を求める命令では、基数 e(= 2.71828...)に対する浮動小数点数の指数値を求めることができます。入力か結果のいずれかが浮動小数点数でない場合は、OV ビットと OS ビットが 1 に設定され、ENO が 0 に設定されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

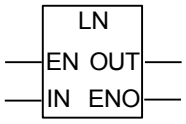
例



入力 I0.0 の信号状態が 1 になると、EXP ボックスが実行されます。EXP(MD0)の結果は、メモリダブルワード MD10 に出力されます。入力か結果のいずれかが浮動小数点数でない場合で、なおかつ I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定されます。

8.4.4 LN：浮動小数点数の自然対数を求める

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	番号
OUT	REAL	I、Q、M、D、L	自然対数
ENO	BOOL	I、Q、M、D、L	イネーブル出力

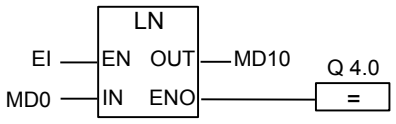
説明

浮動小数点数の自然対数を求める命令では、浮動小数点数の自然対数を求めることができます。入力か結果のいずれかが浮動小数点数でない場合は、OV ビットと OS ビットが 1 に設定され、ENO が 0 に設定されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

例



入力 I0.0 の信号状態が 1 になると、LN ボックスが実行されます。LN(MD0)の結果は、メモリダブルワード MD10 に出力されます。MD0 が 0 未満の場合か、入力か結果のいずれかが浮動小数点数でない場合で、なおかつ I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定されます。

8.4.5 角の三角関数を浮動小数点数で求める

説明

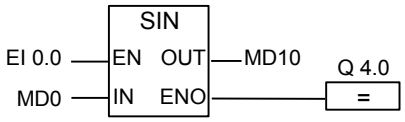
次の命令では、角の三角関数を 32 ビットの IEEE 浮動小数点数で求めることができます。

命令	意味
SIN	ラジアン単位で指定された角の浮動小数点数のサインを求める。
COS	ラジアン単位で指定された角の浮動小数点数のコサインを求める。
TAN	ラジアン単位で指定された角の浮動小数点数のタンジェントを求める。
ASIN	浮動小数点数のアークサインを求める。結果となる角はラジアン単位で指定される。値の範囲を次に示す。 -π / 2 <= arc sine <= + π / 2, where π = 3.14...
ACOS	浮動小数点数のアークコサインを求める。結果となる角はラジアン単位で指定される。値の範囲を次に示す。 0 v arc cosine <= + π, where π = 3.14...
ATAN	浮動小数点数のアークタンジェントを求める。結果となる角はラジアン単位で指定される。値の範囲を次に示す。 -π / 2 <= arc tangent <= + π / 2, where π = 3.14...

ステータスワード

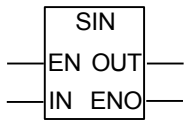
	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	X	0	X	X	1

例



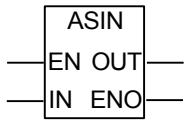
入力 I0.0 の信号状態が 1 になると、SIN ボックスが実行されます。SIN(MD0)の結果は、メモリダブルワード MD10 に出力されます。入力か結果のいずれかが浮動小数点数でない場合で、なおかつ I0.0 の信号状態が 0 の場合は、出力 Q4.0 が 0 に設定されます。

シンボル



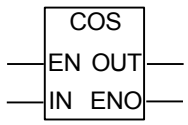
パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	番号
OUT	REAL	I、Q、M、D、L	IN の値の sin
ENO	BOOL	I、Q、M、D、L	イネーブル出力

シンボル



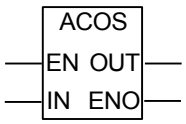
パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	番号
OUT	REAL	I、Q、M、D、L	IN のアークサイン
ENO	BOOL	I、Q、M、D、L	イネーブル出力

シンボル



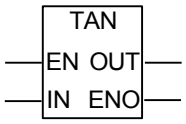
パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	番号
OUT	REAL	I、Q、M、D、L	IN の値の cos
ENO	BOOL	I、Q、M、D、L	イネーブル出力

シンボル



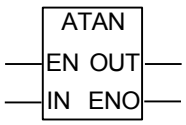
パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	番号
OUT	REAL	I、Q、M、D、L	アークコサイン
ENO	BOOL	I、Q、M、D、L	イネーブル出力

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	番号
OUT	REAL	I、Q、M、D、L	タンジェント
ENO	BOOL	I、Q、M、D、L	イネーブル出力

シンボル

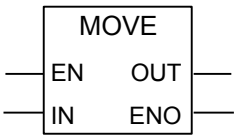


パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	REAL	I、Q、M、D、L または定数	番号
OUT	REAL	I、Q、M、D、L	IN の値の arctan
ENO	BOOL	I、Q、M、D、L	イネーブル出力

9 移動命令

9.1 MOVE：値の割り付け

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN	8、16、または 32 ビット長さのすべての基本データタイプ	I、Q、M、D、L または定数	元の値
OUT	8、16、または 32 ビット長さのすべての基本データタイプ	I、Q、M、D、L	宛先アドレス
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

値の割り付け 命令により、特定の値を変数に割り付けることができます。

IN 入力で指定した値が、出力 OUT で指定したアドレスにコピーされます。ENO は、EN と同じ信号状態になります。

ジャンプボックスでは、値の割り付け命令により、8、16、または 32 ビット長さのすべての基本データタイプをコピーできます。配列やストラクチャなどのユーザー定義データタイプは、システムファンクション"BLKMOVE"SFC 20 を使用してコピーする必要があります。

値の割り付け 命令は、マスタコントロールリレー(MCR)の影響を受けます。MCR ファンクションの使い方についての詳細は、マスタコントロールリレーのオン/オフを参照してください。

ステータスワード

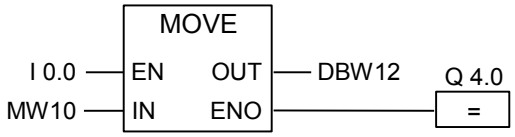
	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	1	-	-	-	-	0	1	1	1

注記

長さの異なるデータタイプに値を移動すると、必要に応じて高位値のバイトが切り捨てられるか、ゼロが充てんされます。

例: ダブルワード	**1111 1111**	**0000 1111**	**1111 0000**	**0101 0101**
移動	**結果**			
ダブルワードへの変換	1111 1111	0000 1111	1111 0000	0101 0101
バイトに移動				0101 0101
ワードへの変換			1111 0000	0101 0101
例: バイト				**1111 0000**
移動	**結果**			
バイトに移動				1111 0000
ワードへの変換			0000 0000	1111 0000
ダブルワードへの変換	0000 0000	0000 0000	0000 0000	1111 0000

例



入力 I0.0 が 1 になると、命令が実行され、MW10 の内容がオープン DB のデータワード 12(DBW12) にコピーされます。

命令が実行されると、Q4.0 は 1 に設定されます。

10 プログラム制御命令

10.1 プログラム制御命令の概要

説明

プログラム制御操作の実行に使用可能な命令を次に示します。

- CALL : パラメータなしの FC/SFC の呼び出し
- CALL_FB : FB をボックスとして呼び出す
- CALL_FC(FC をボックスとして呼び出す)
- CALL_SFB : システム FB をボックスとして呼び出す
- CALL_SFC : システム FC をボックスとして呼び出す
- 複数インスタンスの呼び出し
- ライブラリからのブロック呼び出し

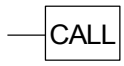
- マスタコントロールリレー命令
- MCR ファンクションの使用方法に関する重要事項
- MCR< マスタコントロールリレーのオン
- MCR> マスタコントロールリレーのオフ
- MCRA マスタコントロールリレーの開始
- MCRD マスタコントロールリレーの終了

- RET リターン

10.2 CALL : パラメータなしの FC/SFC の呼び出し

シンボル

<FC-/SFC 番号>



パラメータ	データタイプ	メモリ領域	説明
番号	BLOCK_FC	-	FC や SFC の番号(例： FC10、SFC59)を入力します。使用できる SFC は、CPU によって異なります。 条件付き呼び出しの場合、FB ではアドレスとして BLOCK_FC タイプのパラメータを入力できますが、FC では入力できません。

説明

パラメータなしの FC/SFC の呼び出し命令を使用すると、**パラメータをもたないファンクション(FC)またはシステムファンクション(SFC)** を呼び出せます。前に論理演算が置かれている場合は条件付き呼び出し、置かれていない場合は無条件呼び出しになります。

ファンクション(FC)のコードセクションでは、タイプ BLOCK_FC のパラメータを条件付き呼び出しのアドレスとして指定することはできません。ただし、ファンクションブロック(FB)では、BLOCK_FC のパラメータをアドレスとして指定できます。

条件付き呼び出しは、RLO が 1 の場合だけに実行されます。条件付き呼び出しが実行されない場合、呼び出し命令後に RLO は 0 になります。命令が実行された場合は、以下のファンクションが実行されます。

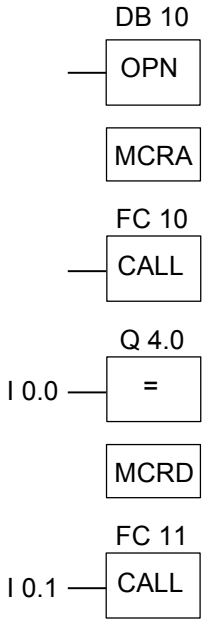
- 呼び出し側ブロックに戻る必要のあるアドレスが保存されます。
- データブロックレジスタが保存されます(データブロックおよびインスタンスデータブロック)。
- MA ビット(MCR 起動ビット)をブロックスタック(BSTACK) に書き込む。
- 呼び出された FC または SFC に、新しいローカルデータ領域が作成されます。

呼び出されたブロックで、引き続きプログラムが実行されます。

ステータスワード

		BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
条件付き呼び出し	書き込みの内容	-	-	-	-	0	0	1	1	0
無条件呼び出し	書き込みの内容	-	-	-	-	0	0	1	-	0

例



FC10 の無条件呼び出しが実行されると、CALL 命令は以下の作業を実行します。

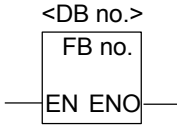
- 現在の FB に戻る必要のあるアドレスを保存します。
- DB10 と FB のインスタンスデータブロックの選択状況を保存。
- MCRA 命令で 1 に設定された MA ビットをブロックスタック(BSTACK)にプッシュし、呼び出された FC10 に対して 0 に再設定します。

これら作業が終了した後に、FC10 でプログラム処理が続行されます。FC10 で MCR ファンクションを使用したい場合は、そこでファンクションを再実行する必要があります。FC10 が完了すると、プログラムの実行は呼び出し側 FB に戻り、MA ビットが復元されます。FC10 でどの DB が使用されたかに関係なく、DB10 と、ユーザー定義 FB のインスタンスデータブロックが再び現行 DB になります。

FC10 からの戻りジャンプ後、入力 I0.0 の信号状態は出力 Q4.0 に割り付けられます。FC11 に対する呼び出しは条件付き呼び出しです。これは、入力 10.1 の信号状態が 1 の場合にだけ実行されます。呼び出しが実行された場合、このファンクションは FC10 の呼び出しと同じです。

10.3 CALL_FB : FB をボックスとして呼び出す

シンボル



シンボルは、ファンクションブロックによって異なります(パラメータが指定されているかどうか、指定されている場合はその数に影響される)。EN、ENO、および FB の名前または数は必ず指定します。

パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、L、D	イネーブル入力
ENO	BOOL	I、Q、M、L、D	イネーブル出力
FB 番号 DB 番号	BLOCK_FB BLOCK_DB	- -	FB/DB の番号。範囲は CPU により異なる

説明

CALL_FB (FB をボックスとして呼び出す)は、EN の信号状態が 1 の場合に実行されます。CALL_FB 命令呼び出しを実行すると、以下の処理が行われます。

- 呼び出し側ブロックのリターンアドレスが保存されます
- 現在開いている 2 つのデータブロック(DB およびインスタンス DB)の選択データが保存されます。
- 呼び出されたファンクションブロックに、新しいローカルデータ領域が作成されます。
- MA ビット(有効 MCR ビット)が BSTACK にプッシュされます。

ステータスワード

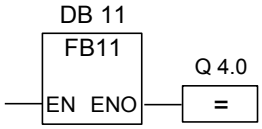
		BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
条件付き呼び出し	書き込みの内容	X	-	-	-	0	0	X	X	X
無条件呼び出し	書き込みの内容	-	-	-	-	0	0	X	X	X

例

ネットワーク 1

ネットワーク 2

ネットワーク 3



ネットワーク 4

上記に示したネットワークは、ユーザーが作成したファンクションブロックのプログラムセクションです。このブロックで DB10 が開き、MCR が有効になります。条件なしの FB11 呼び出しが実行されると、以下が実行されます。

呼び出し側ファンクションブロックのリターンアドレス、DB10 の選択データ、および呼び出し側ファンクションブロックのインスタンスデータブロックが保存されます。MCRA 命令で 1 に設定された MA ビットは BSTACK にプッシュされ、呼び出し先ファンクションブロック FB11 に対して 0 に設定されます。このプログラムは、FB11 内で継続されます。FB11 が MCR を必要とする場合は、MCR をファンクションブロックで再実行する必要があります。RLO の信号状態は、呼び出し側 FB でエラー評価を行えるように、SAVE 命令で BR ビットに保存する必要があります。FB11 が実行中の場合、プログラムは呼び出し側ファンクションブロックへ戻ります。MA ビットは復元され、ユーザーが作成したファンクションブロックのインスタンスデータブロックはオープンデータブロックに戻ります。FB11 が正常に実行されると、ENO の信号状態は 1 になり、Q4.0 の信号状態も 1 になります。

注記

FB/SFB 呼び出しを実行すると、前回開いていたデータブロックの番号は失われます。必要な DB は再度開く必要があります。

10.4 CALL_FC(FC をボックスとして呼び出す)

シンボル

シンボルは、ファンクションによって異なります(パラメータが指定されているかどうか、指定されている場合はその数に影響される)。EN、ENO、および FC の名前または番号は必ず指定します。

パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、L、D	イネーブル入力
ENO	BOOL	I、Q、M、L、D	イネーブル出力
FC 番号	BLOCK_FC	-	FC の番号。範囲は CPU により異なる

説明

CALL_FC (FC をボックスとして呼び出す)は、ファンクション(FC)を呼び出します。この呼び出しは、EN の信号状態が 1 の場合に実行されます。CALL_FC 命令を実行すると、以下の処理が行われます。

- 呼び出し側ブロックのリターンアドレスが保存されます
- 呼び出されたファンクションに、新しいローカルデータ領域が作成されます。
- MA ビット(有効 MCR ビット)が BSTACK にプッシュされます。

呼び出されたファンクションで、プログラムの実行が再開されます。

BR ビットがチェックされ、ENO が決定します。ユーザーは SAVE 命令を使って、呼び出されたブロックの BR ビットに、必要な信号状態(エラー評価)を割り付ける必要があります。

ファンクションの呼び出しを行い、呼び出されたブロックの変数宣言テーブルに IN、OUT、および IN_OUT の宣言がある場合、これらの変数は、呼び出し側ブロックのプログラムに仮パラメータリストとして追加されます。

ファンクションを呼び出す場合、その呼び出し位置で実パラメータを仮パラメータに割り付ける**必要**があります。どのような場合でも、ファンクション宣言内に初期値を指定しても、意味がありません。

ステータスワード

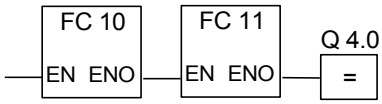
		BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
条件付き呼び出し	書き込みの内容	X	-	-	-	0	0	X	X	X
無条件呼び出し	書き込みの内容	-	-	-	-	0	0	X	X	X

例

ネットワーク 1

ネットワーク 2

ネットワーク 3



上記に示したネットワークは、ユーザーが作成したファンクションブロックのプログラムセクションです。このブロックで DB10 が開き、MCR が有効になります。条件なしの FC10 呼び出しが実行されると、以下が実行されます。

呼び出し側ファンクションブロックのリターンアドレス、DB10 の選択データ、および呼び出し側ファンクションブロックのインスタンスデータブロックが保存されます。MCRA 命令で 1 に設定された MA ビットは BSTACK にプッシュされ､呼び出し先ブロック FC10 に対して 0 に設定されます。このプログラムは、FC10 内で継続されます。FC10 が MCR を必要とする場合は、MCR を FC10 で再実行する必要があります。RLO の信号状態は、呼び出し側 FB でエラー評価を行えるように、SAVE 命令で BR ビットに保存する必要があります。FC10 が実行中の場合、プログラムは呼び出し側ファンクションブロックへ戻ります｡MA ビットが復元されます。FC10 の実行後は、ENO の信号状態に応じて、呼び出し側 FB でプログラムが再開します。

ENO = 1　FC11 が実行される

ENO = 0　次のネットワークでプログラムが開始する

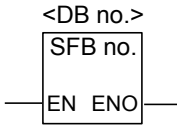
FC11 が正常に実行されると、ENO は 1 に設定され、Q4。0 も 1 に設定されます。

注記

プログラムが呼び出し側ブロックに戻った後で、前回開いていた DB が再び開くとは限りません。詳細については、Readme ファイルの注記を参照してください。

10.5 CALL_SFB : システム FB をボックスとして呼び出す

シンボル



シンボルは、システムファンクションブロックによって異なります(パラメータが指定されているかどうか、指定されている場合はその数に影響される)。EN、ENO、および SFB の名前または数は必ず指定します。

パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、L、D	イネーブル入力
ENO	BOOL	I、Q、M、L、D	イネーブル出力
SFB 番号 DB 番号	ブロック_SFB BLOCK_DB	- -	SFB/DB の番号。範囲は CPU により異なる

説明

CALL_SFB (FB をボックスとして呼び出す)は、EN の信号状態が 1 の場合に実行されます。CALL_SFB 命令を実行すると、以下の処理が行われます。

- 呼び出し側ブロックのリターンアドレスが保存されます
- 現在開いている 2 つのデータブロック(DB およびインスタンス DB)の選択データが保存されます。
- 呼び出されたシステムファンクションブロックに、新しいローカルデータ領域が作成されます。
- MA ビット(有効 MCR ビット)が BSTACK にプッシュされます。

呼び出されたシステムファンクションで、プログラムの実行が再開されます。ファンクションの呼び出しでエラーが発生しなければ ENO は 1 になります。

ステータスワード

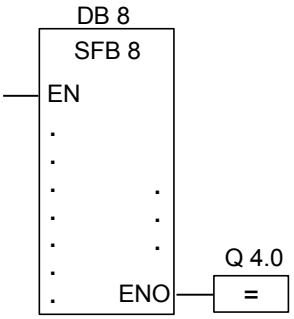
		BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
条件付き呼び出し	書き込みの内容	X	-	-	-	0	0	X	X	X
無条件呼び出し	書き込みの内容	-	-	-	-	0	0	X	X	X

例

ネットワーク 1

ネットワーク 2

ネットワーク 3



ネットワーク 4

上記に示したネットワークは、ユーザーが作成したファンクションブロックのプログラムセクションです。このブロックで DB10 が開き、MCR が有効になります。条件なしの SFB8 呼び出しが実行されると、以下が実行されます。

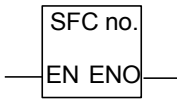
呼び出し側ファンクションブロックのリターンアドレス、DB10 の選択データ、および呼び出し側ファンクションブロックのインスタンスデータブロックが保存されます。MCRA 命令で 1 に設定された MA ビットは BSTACK にプッシュされ、呼び出し先システムファンクションブロック SFB8 に対して 0 に設定されます。このプログラムは、SFB8 内で継続されます。SFB8 が実行されている場合は、プログラムにより呼び出し元ファンクションブロックが返されます。MA ビットは復元され、ユーザーが作成したファンクションブロックのインスタンスデータブロックは現在のデータブロックに戻ります。SFB8 が正常に実行されると、ENO の信号状態は 1 になり、Q4.0 の信号状態も 1 になります。

注記

FB/SFB 呼び出しを実行すると、前回開いていたデータブロックの番号は失われます。必要な DB は再度開く必要があります。

10.6 CALL_SFC(SFC をボックスとして呼び出す)

シンボル



シンボルは、パラメータが指定されているかどうか、指定されている場合はその数によって異なります。EN、ENO、および SFC の名前または数は必ず指定します。

パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、L、D	イネーブル入力
ENO	BOOL	I、Q、M、L、D	イネーブル出力
SFC 番号	BLOCK_SFC	-	SFC の番号。範囲は CPU により異なる

説明

CALL_SFC (SFC をボックスとして呼び出す)は、システムファンクションを呼び出します。この呼び出しは、EN の信号状態が 1 の場合に実行されます。CALL_SFC 命令を実行すると、以下の処理が行われます。

- 呼び出し側ブロックのリターンアドレスが保存されます
- 呼び出されたシステムファンクションに、新しいローカルデータ領域が作成されます
- MA ビット(有効 MCR ビット)が BSTACK にプッシュされます。

呼び出されたシステムファンクションで、プログラムの実行が再開されます。ファンクションの呼び出しでエラーが発生しなければ ENO は 1 になります。

ステータスワード

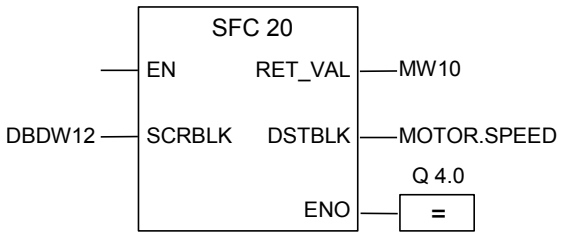
		BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
条件付き呼び出し	書き込みの内容	X	-	-	-	0	0	X	X	X
無条件呼び出し	書き込みの内容	-	-	-	-	0	0	X	X	X

例

ネットワーク 1

ネットワーク 2

ネットワーク 3



上記に示したネットワークは、ユーザーが作成したファンクションブロックのプログラムセクションです。このブロックで DB10 が開き、MCR が有効になります。条件なしの SFC20 呼び出しが実行されると、以下が実行されます。

呼び出し側ファンクションブロックのリターンアドレス、DB10 の選択データ、および呼び出し側ファンクションブロックのインスタンスデータブロックが保存されます。MCRA 命令で 1 に設定された MA ビットは BSTACK にプッシュされ、呼び出し先ブロック SFC20 に対して 0 に設定されます。このプログラムは、SFC20 内で継続されます。SFC20 が実行されている場合は、プログラムにより呼び出し元ファンクションブロックが返されます。MA ビットが復元されます。

SFC20 の実行後は、ENO の信号状態に応じて、呼び出し側 FB でプログラムが再開します。

ENO = 1　Q4.0 = 1

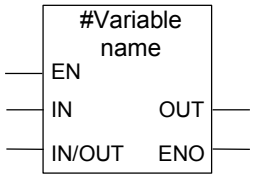
ENO = 0　Q4.0 = 0

注記

プログラムが呼び出し側ブロックに戻った後で、前回開いていた DB が再び開くとは限りません。詳細については、Readme ファイルの注記を参照してください。

10.7 複数インスタンスの呼び出し

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
ENO	BOOL	I、Q、M、D、L	イネーブル出力
#Variable name	FB/SFB	-	複数インスタンスの名前

説明

複数インスタンスは、ファンクションブロックのスタティック変数を宣言して作成します。すでに宣言されている複数インスタンスのみが、プログラムエレメントのカタログにリストされます。このインスタンスのシンボルは、パラメータが指定されているかどうか、指定されている場合はいくつ指定されているかによって異なります。EN、ENO、および変数名は常に指定されています。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	0	0	X	X	X

10.8 ライブラリからのブロック呼び出し

SIMATIC Manager で使用できるライブラリを使って、以下のブロックを選択します。

- CPU オペレーティングシステムに統合されているブロック(バージョン 3 の STEP 7 プロジェクトの場合は"標準ライブラリ"、バージョン 2 の STEP 7 プロジェクトの場合は"stdlibs(V2)")
- 繰り返し使用できるように、あらかじめライブラリに保存してあるブロック

10.9 マスタコントロールリレー命令

MCR ファンクションの使用方法に関する重要事項

マスタコントロールリレーの定義

マスタコントロールリレー(MCR、マスタコントロールリレーのオン/オフも参照)を使用して、信号の流れを有効および無効にします。信号の流れを無効にする操作は、計算値の代わりにゼロを書き込む命令シーケンス、または既存のメモリ値を変更しない命令シーケンスに相当します。以下に示す命令で実行される操作は、MCR に依存します。

割り付け命令および**中間出力**命令は、MCR が 0 の場合にメモリに 0 を書き込みます。**出力設定**命令および**出力リセット**命令は、既存の値を変更しません。

MCR 領域の影響を受ける命令を次に示します。

- # : 中間出力
- = : 割り付け
- R : 出力のリセット
- S : 出力の設定
- SR : Set_Reset フリップフロップ
- RS : Reset_Set フリップフロップ
- MOVE : 値の割り付け

MCR に依存する命令、および MCR の信号状態に対するその応答

MCR の信号状態	出力、 中間出力	出力セット、 出力リセット	移動
0 ("オフ")	0 が書き込まれる。 (電圧が失われると、オフ状態のリレーと同様に機能する)	書き込みは行われない。 (電圧が失われると、現在の状態のリレーと同様に機能する)	0 が書き込まれる。 (電圧が失われると、値 0 を生成するコンポーネントと同様に機能する)
1 ("オン")	処理を正常に実行	処理を正常に実行	処理を正常に実行

10.10 MCR ファンクションの使用方法に関する重要事項

MCRA によってマスタコントロールリレーが有効化されたブロックに関する注意

- MCR が無効化されると、マスタコントロールリレーオンとマスタコントロールリレーオフとの間のプログラムセグメントにあるすべての割り付けによって、値 0 が書き込まれます。これは、ブロックへのパラメータ転送などの割り付けが入った**すべての**ボックスに対して有効です。
- RLO がマスタコントロールリレーオン命令の前に 0 になっていると、MCR は無効になります。

危険:PLC が STOP になったり、未定義のランタイム特性が生成されます!

コンパイラは、VAR_TEMP で定義されているテンポラリ変数のローカルデータへ書き込みアクセスして、アドレスの計算を行います。すなわち、次のコマンドシーケンスを使用すると、PLC が STOP に設定されたり、ランタイム特性が未定義になったりします。

仮パラメータアクセス

- タイプ STRUCT、UDT、ARRAY、STRING の複合 FC パラメータのコンポーネントにアクセス
- マルチプルインスタンス能力を持つブロック(バージョン 2 ブロック)の IN_OUT 領域からの、STRUCT、UDT、ARRAY、STRING タイプの複合 FB パラメータで構成されるコンポーネントへのアクセス
- アドレスが 8180.0.を超える場合の、マルチプルインスタンス能力を持つファンクションブロック(バージョン 2 ブロック)のパラメータへのアクセス
- マルチプルインスタンス能力を持つファンクションブロック(バージョン 2 ブロック)の DB0 を開く BLOCK_DB パラメータへのアクセス。さらにデータアクセスが実行されると、CPU は STOP になります。T 0、C 0、FC0、または FB0 も、必ず TIMER、COUNTER、BLOCK_FC、および BLOCK_FB に使用されます。

パラメータの引き渡し

- パラメータが転送される呼び出し

LAD/FBD

- RLO = 0 で開始する Ladder または FBD の T ブランチおよび中間出力

対策

上記のコマンドを MCR から独立させます。

1st 該当するステートメントまたはネットワークの**前**で、マスタコントロールリレーの終了命令を使って、マスタコントロールリレーを**無効にします**。

2nd 該当するステートメントまたはネットワークの前にマスタコントロールリレーの開始命令を使用して、マスタコントロールリレーを**有効にします**。

10.11 MCR</MCR>：マスタコントロールリレーのオン/オフ

MCR ファンクションの使用方法に関する重要事項

シンボル

MCR をオンにする

マスタコントロールリレーオン(MCR<)命令は、RLO を MCR スタックに保存してから MCR 領域を開くオペレーションを起動します。MCR 命令に示した命令は、MCR ゾーンを開いたときに MCR スタックに保存した RLO によって変わります。

MCR スタックは、LIFO(後入れ先出し)バッファと同様に動作します。指定できるエントリは 8 つまでです。スタックがすでに一杯であると、**MCR ON** 命令により MCR スタックエラー（MCRF）が起こります。

シンボル

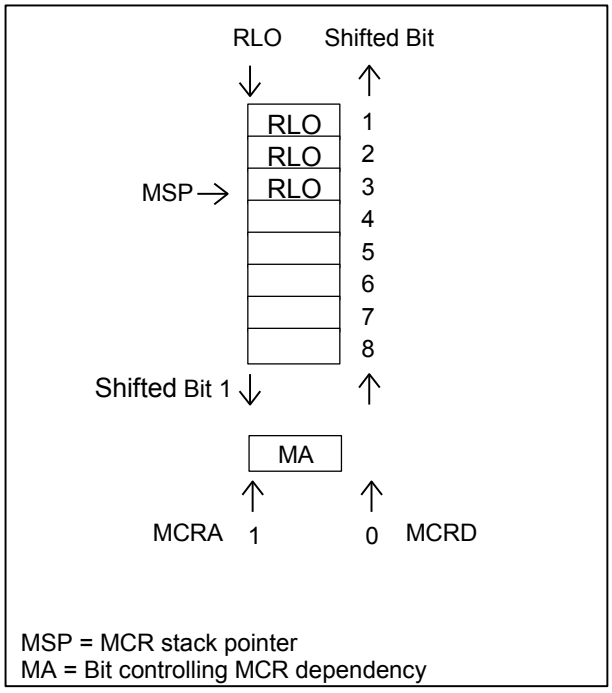
MCR をオフにする

マスタコントロールリレーオフ(MCR>)命令により、直前に開いた MCR 領域が閉じます。これは、マスタコントロールリレーオン命令によって MCR スタックに保存されていた RLO を削除することによって実行されます。**マスタコントロールリレーオン**命令によって RLO は保存されます。LIFO(ラストイン、ファーストアウト)MCR スタックの反対側でリリースされたエントリは 1 に設定します。

スタックがすでに空である場合、**MCR** 命令は MCR スタックエラー（MCRF）を発生させます。

MCR スタック

MCR は、1 ビット長で深度が 8 エントリのスタックによって制御されます。スタック内の 8 つのエントリすべてが 1 であれば、MCR が起動します。MCR< 命令は、RLO を MCR スタックにコピーします。MCR>命令は、スタックから最後のエントリを削除し、解放されたスタックアドレスを 1 に設定します。エラーが発生した場合(8 個を超える MCR>命令が連続している場合、あるいは、MCR スタックが空のときに MCR>命令を実行しようとした場合など)、MCRF エラーメッセージが表示されます。MCR スタックのモニタは、スタックポインタに基づいています(MSP: 0 = 空、1 = エントリ 1 つ、2 = エントリ 2 つ...、8 = エントリ 8 つ)。



MCR 命令は、RLO の信号状態を受け取り、これを MCR ビットにコピーします。

MCR 命令は、MCR ビットを無条件に 1 に設定します。この特性により、MCRA と MCRD 命令間にあるその他の各命令は、MCR ビットから独立して動作します。

MCR<命令と MCR>命令のネスティング

MCR<命令と **MCR>**命令はネストすることができます。すなわち、MCR> 命令の前に、最大 8 個の MCR< 命令を連続してプログラムすることができます。MCR< 命令と MCR> 命令は、同じ数ずつプログラムされなければなりません。

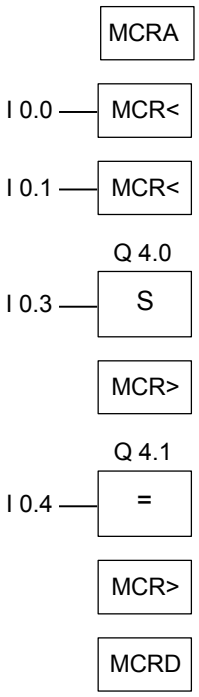
MCR< 命令がネストされると、下位のネスティングレベルの MCR ビットが形成されます。この時、MCR< 命令は、AND 真理値表に従って、RLO の信号状態を MCR ビットの信号状態と組み合わせます。

MCR>命令により 1 つのネストレベルが完了すると、上位レベルから MCR ビットが取り出されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	1	-	0

例



MCRA 命令により MCR ファンクションが有効になると、MCR 領域を 8 レベルまでネストすることができます。たとえば、MCR 領域が 2 つある場合、最初の MCR>命令は 2 番目の MCR<命令と共に動作します。2 番目の MCR ブラケット(MCR<MCR>)の間の命令はすべて、2 番目の MCR 領域に属します。オペレーションは次のように実行されます。

- I0.0 が 1 の場合、入力 I0.4 の信号状態は出力 Q4.1 に割り付けられます。
- I0.0 が 0 の場合、出力 Q4.1 の信号状態は、入力 I0.4 の信号状態にかかわらず 0 になります。入力 I0.3 の信号状態に関係なく、出力 Q4.0 の設定は変わりません。
- I0.0 と I0.1 が 1 の場合、I0.3 が 1 で Q4.1 が I0.4 であれば出力 Q4.0 は 1 に設定されます。
- I0.1 が 0 の場合、入力 I0.3 と入力 I0.0 の信号状態に関係なく出力 Q4.0 の設定は変わりません。

10.12 MCRA/MCRD：マスタコントロールリレーの有効化/無効化

MCR ファンクションの使用方法に関する重要事項

シンボル

MCR の有効化

マスタコントロールリレーの有効化 命令を使用すると、それ以降に実行されるコマンドは MCR 依存になります。このコマンドを入力すれば、命令を使って MCR ゾーンをプログラムできます(マスタコントロールリレーのオン/オフを参照)。MCR 領域が有効になると、すべての MCR アクションは、MCR スタックの内容に依存するようになります。

シンボル

MCR の無効化

マスタコントロールリレーの無効化 命令を使用すると、それ以降に実行されるコマンドは MCR に依存しなくなります。この命令の後に MCR ゾーンをプログラムすることはできません。MCR 領域が無効になると、MCR は、MCR スタックのエントリに関係なく常にオンになります。

MCR スタックとその依存性を制御するビット(MA ビット)は、個々のレベルに関係するので、ユーザーがシーケンスレベルを変えるたびに保存し、取り出さなければなりません。これらは、各シーケンスレベルの最初でプリセットされます(MCR 入力ビット 1～8 は 1 に、MCR スタックポインタは 0 に、MA ビットは 0 に設定される)。

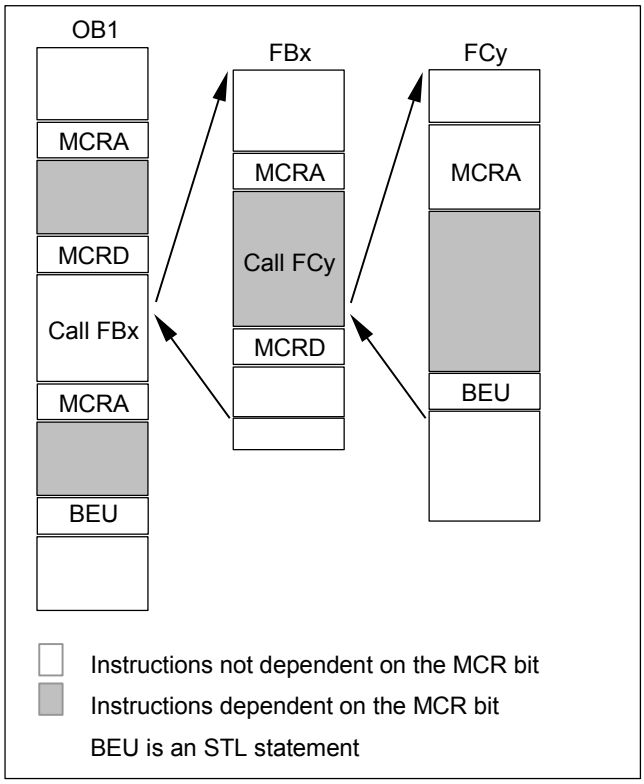
MCR スタックはブロック間で転送されます。MA ビットは、ブロックが呼び出されるたびに保存され、0 に設定されます。ブロックの終了時にフェッチバックされます。

MCR は、コード生成 CPU のランタイムを最適化できるような方法で実装されます。MCR の依存性は、ブロックによって渡されることがないため、MCR 命令で明示的に有効にする必要があります。コード生成 CPU はこの命令を認識すると、MCR 命令を認識するかブロックの最後に到達するまで、MCR スタックの評価に必要な追加コードを生成します。MCRA/MCRD の範囲外で命令を実行する場合は、実行時間が長くなるようなことはありません。

MCRA 命令と **MCRD** 命令は、プログラム内で常にペアで使用しなければなりません。

MCR ゾーンの有効化と無効化

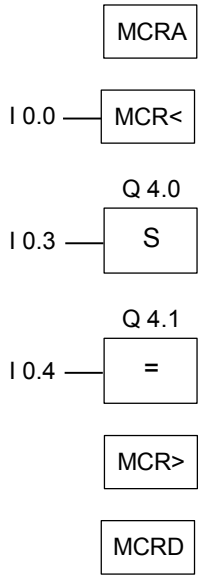
MCRA と MCRD の間でプログラミングされたオペレーションは、MCR ビットの信号状態に依存します。MCRA-MCRD シーケンス外でプログラムされた操作は、MCR ビットの信号状態には依存しません。MCRD 命令が失われると、MCRA 命令と BEU 命令の間でプログラミングされているオペレーションは MCR ビットに依存します。



ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	-	-	-	-

例



MCRA 命令により、次の MCRD まで、MCR 機能が有効になります。MCR<と MCR>の間の命令は、MA ビット(この場合は I0.0)に応じて処理されます。

- 入力 I0.0 の信号状態が 1 の場合
  - 入力 I0.3 の信号状態が 1 になると、出力 Q4.0 は 1 に設定されます。
  - 入力 I0.3 の信号状態が 0 の場合、出力 Q4.0 はそのまま変わりません。
  - 入力 I0.4 の信号状態が、出力 Q4.1 に割り付けられます。
- 入力 I0.0 の信号状態が 0 の場合
  - 入力 I0.3 の信号状態に関係なく、出力 Q4.0 の設定は変わりません。
  - 入力 I0.4 の信号状態にかかわらず、出力 Q4.1 は 0 です。

FC や FB の MCR 機能を有効にするには、ブロック内でユーザー自身がプログラムしなければなりません。ファンクションまたはファンクションブロックが MCRA/MCRD シーケンスから呼び出されると、このシーケンス内のすべての命令が自動的に MCR ビット依存になるわけではありません。MCR ビット依存にするには、呼び出されたブロックで MCRA 命令を使用します。

10.13 RET : リターン

シンボル

説明

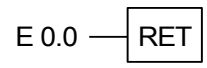
リターン命令を使用すれば、ブロックを終了することができます。この場合、条件付きで終了することができます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	*	-	-	-	0	0	1	1	0

* **RET** 操作は、"SAVE; BEC, "シーケンスで内部的に示されます。これは BR ビットにも影響します。

例



入力 I0.0 の信号状態が 1 になると、ブロックの処理は打ち切られます。

11 シフト命令および回転命令

11.1 シフト命令

11.1.1 シフト命令の概要

説明

シフト命令を使用すれば、入力 IN の内容を左または右に 1 ビットずつ移動することができます(「CPU レジスタ」も参照)。左に n ビット移動すると、入力 IN の内容が 2 の n 乗(2^n)で乗算されます。右に n ビット移動すると、入力 IN の内容が 2 の n 乗(2^n)で除算されます。たとえば、10 進数の 3 に相当するバイナリ値を左へ 3 ビット移動すると、10 進数の 24 に相当するバイナリ値が得られます。10 進数の 16 に相当するバイナリ値を右へ 2 ビット移動すると、10 進数の 4 に相当するバイナリ値が得られます。

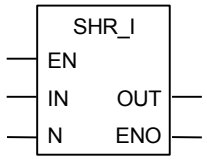
入力パラメータ N に入力する値は、シフトさせるビット数です。シフト命令により空になったビット位置には、0 が挿入されるか、符号ビット(0 は正を表し、1 は負を表す)の信号状態が挿入されます。最後に移動したビットの信号状態は、ステータスワードの CC1 ビットに入力されます。詳細については、CPU レジスタも参照してください。ステータスワードの CC および OV ビットは 0 にリセットされます。ジャンプ命令を使用して、CC ビットを評価することができます。

以下のシフト命令が使用可能です。

- SHR_I : 整数右シフト
- SHR_DI : 右シフト倍長整数
- SHL_W : ワード左シフト
- SHR_W : 右シフトワード
- SHL_DW : ダブルワード左シフト
- SHR_DW : ダブルワード右シフト

11.1.2 SHR_I : 整数右シフト

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、L、D、T、C	イネーブル入力
IN	INT	I、Q、M、L、D	シフトさせる値
N	WORD	I、Q、M、L、D	シフトさせるビット数
OUT	INT	I、Q、M、L、D	シフト命令の結果
ENO	BOOL	I、Q、M、L、D	イネーブル出力

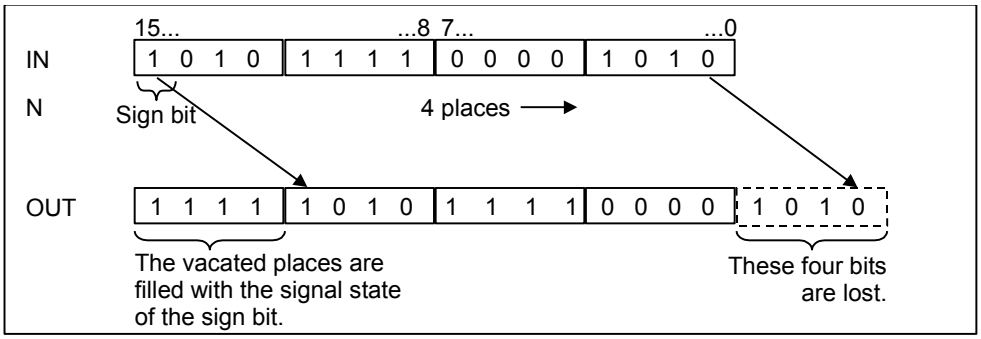
説明

イネーブル入力(EN)の信号状態が 1 になると、**整数右シフト**命令が起動します。この命令により、入力 IN のビット 0～15 が右へ 1 ビットずつ移動します。入力 N には、シフトさせるビット数を入力します。N が 16 よりも大きい場合、このコマンドは N が 16 である場合と同じように動作します。左側のビット位置は、ビット 15 の信号状態(整数の符号)に応じて埋め込まれます。正数の場合は 0 が挿入され、負数の場合は 1 が挿入されます。シフト操作の結果は出力 OUT で確認できます。

この命令で実行された操作で、N の値が 0 以外になると、ステータスワードの CC0 ビットと OV ビットは 0 にリセットされます。

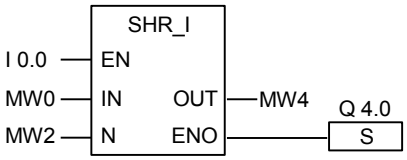
ENO は、EN と同じ信号状態になります。

ステータスワード



	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	-	X	X	X	1

例

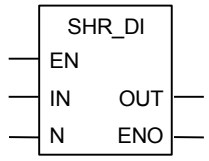


この命令は、I0.0 の信号状態が 1 の場合に実行されます。メモリワード MW0 は、メモリワード MW2 で指定されたビット数だけ右にシフトされます。

結果は、メモリワード MW4 に出力されます。出力 Q4.0 は 1 に設定されます。

11.1.3 SHR_DI : 右シフト倍長整数

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、L、D、T、C	イネーブル入力
IN	DINT	I、Q、M、L、D	シフトさせる値
N	WORD	I、Q、M、L、D	シフトさせるビット数
OUT	DINT	I、Q、M、L、D	シフト命令の結果
ENO	BOOL	I、Q、M、L、D	イネーブル出力

説明

イネーブル入力(EN)の信号状態が 1 になると、**右シフト倍長整数**命令が起動します。この命令により、入力 IN の内容全体が右へ 1 ビットずつ移動します。入力 N には、シフトさせるビット数を入力します。N が 32 よりも大きい場合、このコマンドは N が 32 である場合と同じように動作します。左側のビット位置は、ビット 31 の信号状態(倍長整数の符号)に応じて埋め込まれます。正数の場合は 0 が挿入され、負数の場合は 1 が挿入されます。シフト操作の結果は出力 OUT で確認できます。

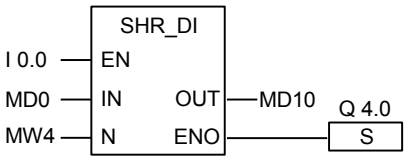
この命令で実行された操作で、N の値が 0 以外になると、ステータスワードの CC0 ビットと OV ビットは 0 にリセットされます。

ENO は、EN と同じ信号状態になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	-	X	X	X	1

例

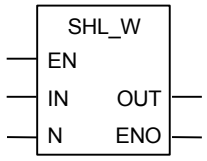


この命令は、I0.0 の信号状態が 1 の場合に実行されます。メモリダブルワード MD0 は、メモリワード MW4 で指定されたビット数だけ右にシフトされます。

この結果は、メモリダブルワード MD10 に入力されます。出力 Q4.0 は 1 に設定されます。

11.1.4 SHL_W : ワード左シフト

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、L、D、T、C	イネーブル入力
IN	WORD	I、Q、M、L、D	シフトさせる値
N	WORD	I、Q、M、L、D	シフトさせるビット数
OUT	WORD	I、Q、M、L、D	シフト命令の結果
ENO	BOOL	I、Q、M、L、D	イネーブル出力

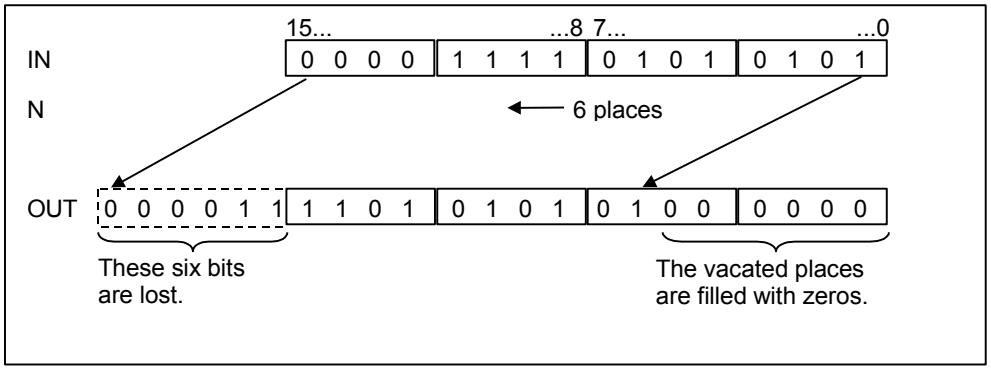
説明

イネーブル入力(EN)の信号状態が 1 になると、**ワード左シフト**命令が起動します。この命令により、入力 IN のビット 0～15 が左へ 1 ビットずつ移動します。

入力 N では、値の移動分となるビット数を指定します。N が 16 より大きい場合、出力 OUT に 0 が書き込まれ、ステータスワードの CC0 ビットと OV ビットは 0 に設定されます。右側のビット位置は 0 で埋められます。シフト操作の結果は出力 OUT で確認できます。

この命令が実行されると、ステータスワードの CC0 ビットと OV ビットは常に 0 にリセットされます。

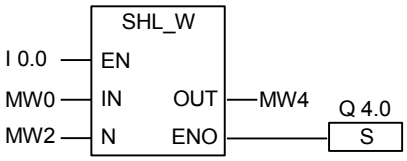
ENO は、EN と同じ信号状態になります。



ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	-	X	X	X	1

例



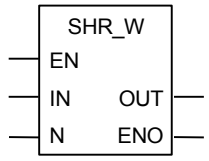
この命令は、I0.0 の信号状態が 1 の場合に実行されます。

メモリワード MW0 が、メモリワード MW2 で指定されているビット数だけ左へ移動します。

結果は、メモリワード MW4 に出力されます。出力 Q4.0 は 1 に設定されます。

11.1.5 SHR_W : 右シフトワード

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、L、D、T、C	イネーブル入力
IN	WORD	I、Q、M、L、D	シフトさせる値
N	WORD	I、Q、M、L、D	シフトさせるビット数
OUT	WORD	I、Q、M、L、D	シフト命令の結果
ENO	BOOL	I、Q、M、L、D	イネーブル出力

説明

イネーブル入力(EN)の信号状態が 1 になると、**右シフトワード**命令が起動します。この命令により、入力 IN のビット 0～15 が右へ 1 ビットずつ移動します。ビット 16～31 は変化しません。入力 N には、シフトさせるビット数を入力します。N が 16 より大きい場合、出力 OUT に 0 が書き込まれ、ステータスワードの CC0 ビットと OV ビットは 0 にリセットされます。左側に空いたビット位置は 0 で埋められます。シフト操作の結果は出力 OUT で確認できます。

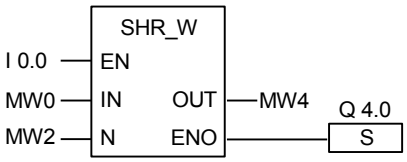
この命令で実行された操作で、N の値が 0 以外になると、ステータスワードの CC0 ビットと OV ビットは 0 にリセットされます。

ENO は、EN と同じ信号状態になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	-	X	X	X	1

例

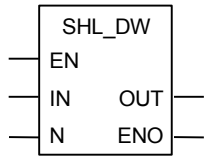


この命令は、I0.0 の信号状態が 1 の場合に実行されます。メモリワード MW0 は、メモリワード MW2 で指定されたビット数だけ右にシフトされます。

結果は、メモリワード MW4 に出力されます。出力 Q4.0 は 1 に設定されます。

11.1.6 SHL_DW：ダブルワード左シフト

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、L、D、T、C	イネーブル入力
IN	DWORD	I、Q、M、L、D	シフトさせる値
N	WORD	I、Q、M、L、D	シフトさせるビット数
OUT	DWORD	I、Q、M、L、D	シフト命令の結果
ENO	BOOL	I、Q、M、L、D	イネーブル出力

説明

イネーブル入力(EN)の信号状態が 1 になると、**ダブルワード左シフト**命令が起動します。この命令により、入力 IN のビット 0～31 が左へ 1 ビットずつ移動します。入力 N には、シフトさせるビット数を入力します。N が 32 より大きい場合、出力 OUT に 0 が書き込まれ、ステータスワードの CC0 ビットと OV ビットは 0 に設定されます。右側に空いたビット位置は 0 で埋められます。シフト操作の結果は出力 OUT で確認できます。

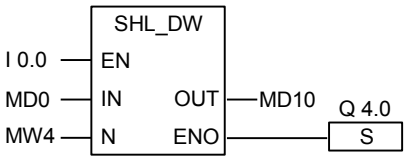
この命令が実行されると、ステータスワードの CC0 ビットと OV ビットは常に 0 にリセットされます。

ENO は、EN と同じ信号状態になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	-	X	X	X	1

例



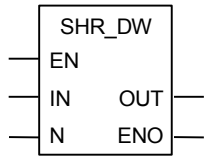
この命令は、I0.0 の信号状態が 1 の場合に実行されます。

メモリダブルワード MD0 は、メモリワード MW4 で指定されたビット数だけ左にシフトされます。

この結果は、メモリダブルワード MD10 に入力されます。出力 Q4.0 は 1 に設定されます。

11.1.7 SHR_DW : ダブルワード右シフト

シンボル



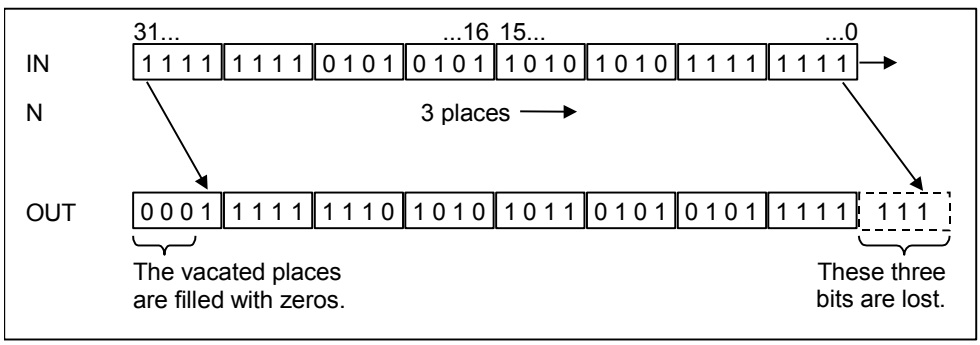
パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、L、D、T、C	イネーブル入力
IN	DWORD	I、Q、M、L、D	シフトさせる値
N	WORD	I、Q、M、L、D	シフトさせるビット数
OUT	DWORD	I、Q、M、L、D	シフト命令の結果
ENO	BOOL	I、Q、M、L、D	イネーブル出力

説明

イネーブル入力(EN)の信号状態が 1 になると、**ダブルワード右シフト**命令が起動します。この命令により、入力 IN のビット 0～31 が右へ 1 ビットずつ移動します。入力 N には、シフトさせるビット数を入力します。N が 32 より大きい場合、出力 OUT に 0 が書き込まれ、ステータスワードの CC0 ビットと OV ビットは 0 にリセットされます。左側に空いたビット位置は 0 で埋められます。シフト操作の結果は出力 OUT で確認できます。

N が 0 でない場合、この命令が実行されると、ステータスワードの CC1 ビットと OV ビットは常に 0 にリセットされます。

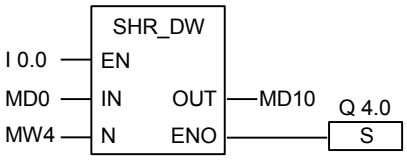
ENO は、EN と同じ信号状態になります。



ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	-	X	X	X	1

例



この命令は、I0.0 の信号状態が 1 の場合に実行されます。メモリダブルワード MD0 は、メモリワード MW4 で指定されたビット数だけ右にシフトされます。

結果は、メモリワード MD10 に出力されます。出力 Q4.0 は 1 に設定されます。

11.2 回転命令

11.2.1 回転命令の概要

説明

回転命令を使用すれば、入力 IN の内容全体を左または右に 1 ビットずつ循環させることができます(「CPU レジスタ」も参照)。空いたビット位置には、入力 IN からシフトされたビットの信号状態が入ります。

回転させるビット数は、入力パラメータ N に入力します。

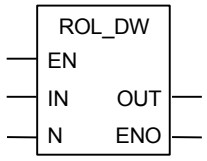
命令に応じて、ステータスワードの CC1 ビットが循環操作に使用されます。ステータスワードの CC0 ビットは 0 にリセットされます。

使用可能な回転命令を次に示します。

- ROL_DW : 左循環ダブルワード
- ROR_DW : ダブルワード右回転

11.2.2 ROL_DW : 左循環ダブルワード

シンボル



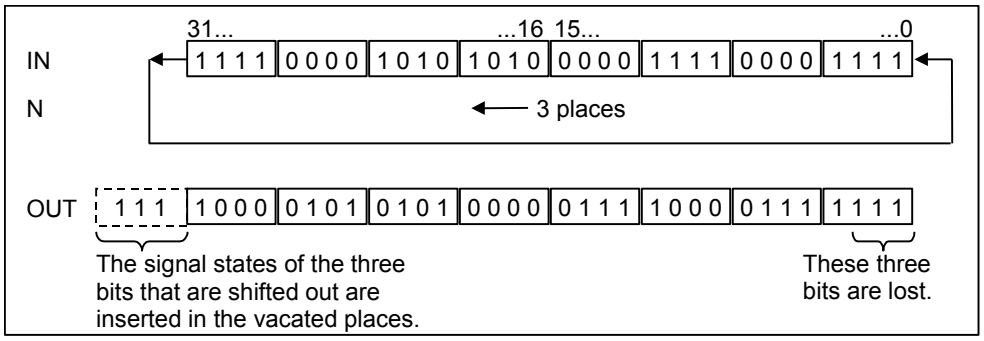
パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、L、D、T、C	イネーブル入力
IN	DWORD	I、Q、M、L、D	回転させる値
N	WORD	I、Q、M、L、D	値の移動分となるビット数
OUT	DWORD	I、Q、M、L、D	循環命令の結果
ENO	BOOL	I、Q、M、L、D	イネーブル出力

説明

イネーブル入力(EN)の信号状態が 1 になると、**左循環ダブルワード**命令が起動します。この命令では、入力 IN の内容全体が左へ 1 ビットずつ移動します。入力 N では、値の移動分となるビット数を指定します。N が 32 を超える場合は、ダブルワードの位置で移動が行われます((N-1)モジュロ 32) +1)。右側のビット位置には、移動したビットの信号状態が挿入されます。循環操作の結果は出力 OUT で確認できます。

この命令で実行された操作で、N の値が 0 以外になると、ステータスワードの CC0 ビットと OV ビットは 0 にリセットされます。

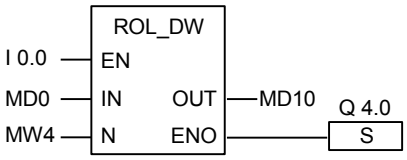
ENO は、EN と同じ信号状態になります。



ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	-	X	X	X	1

例

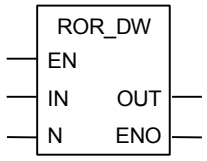


この命令は、I0.0 の信号状態が 1 の場合に実行されます。メモリダブルワード MD0 は、メモリワード MW4 で指定されたビット数だけ左に循環します。

この結果は、メモリダブルワード MD10 に入力されます。出力 Q4.0 は 1 に設定されます。

11.2.3 ROR_DW：ダブルワード右回転

シンボル



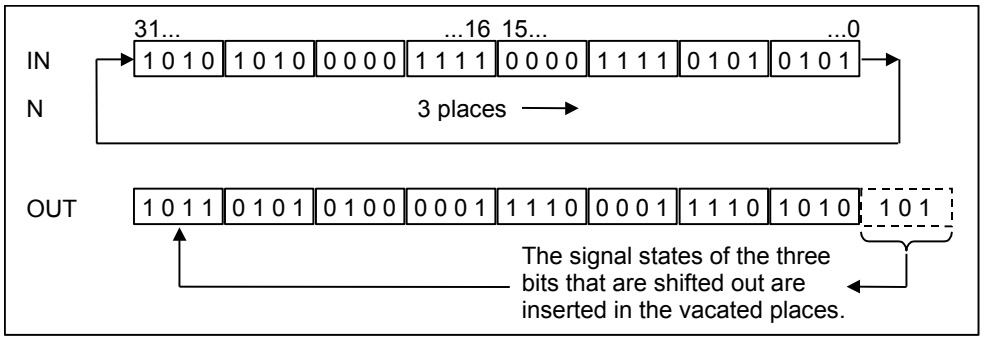
パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、L、D、T、C	イネーブル入力
IN	DWORD	I、Q、M、L、D	回転させる値
N	WORD	I、Q、M、L、D	値の移動分となるビット数
OUT	DWORD	I、Q、M、L、D	循環命令の結果
ENO	BOOL	I、Q、M、L、D	イネーブル出力

説明

イネーブル入力(EN)の信号状態が 1 になると、**ダブルワード右回転**命令が起動します。この命令では、入力 IN の内容全体が右へ 1 ビットずつ移動します。入力 N では、値の移動分となるビット数を指定します。N が 32 を超える場合は、ダブルワードの位置で移動が行われます((N-1)モジュロ 32) +1)。値 N は、0～31 まで可能です。左側のビット位置には、移動するビットの信号状態が入ります。循環操作の結果は出力 OUT で確認できます。

この命令で実行された操作で、N の値が 0 以外になると、ステータスワードの CC0 ビットと OV ビットは 0 にリセットされます。

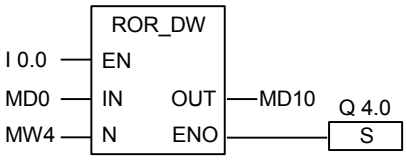
ENO は、EN と同じ信号状態になります。



ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	X	X	X	X	-	X	X	X	1

例



この命令は、I0.0 の信号状態が 1 の場合に実行されます。メモリダブルワード MD0 は、メモリワード MW4 で指定されたビット数だけ右に循環します。

この結果は、メモリダブルワード MD10 に入力されます。出力 Q4.0 は 1 に設定されます。

12 ステータスビット命令

12.1 ステータスビット命令の概要

説明

ステータスビット命令は、ステータスワードのビットと連動するビット論理命令です(「CPU レジスタ」を参照)。これらの各命令は、次の条件のいずれかに反応します。これらの条件は、ステータスワードの 1 つ以上のビットによって示されます。

- バイナリリザルトビット(BR)が設定される(信号状態が 1 になる)。
- 演算ファンクションの結果が、次のいずれかの方法で 0 に関係する。== 0, <> 0, > 0, < 0, >= 0, <= 0.
- 演算ファンクションの結果が無効(UO)である。
- 演算ファンクションによってオーバーフロー(OV)またはストアドオーバーフロー(OS)が発生した。

ステータスビット命令が連続して接続されている場合は、AND 真理値表に従って、信号状態チェックの結果と前の論理演算の結果が結合されます。ステータスビット命令が並列して接続されている場合は、OR 真理値表に従って、命令の結果と前の RLO が結合されます。

ステータスワード

ステータスワードは、CPU のメモリ内にあるレジスタで、ビットアドレスとワード論理命令で参照可能なビットが含まれています。ステータスワードのストラクチャを次に示します。

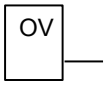
2^{15}...	...2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
		BR	CC1	CC0	OV	OS	OR	STA	RLO	/FC

ステータスワード内のビットは、次の方法で評価することができます。

- 整数演算ファンクション
- 浮動小数点ファンクション

12.2 OV : OV メモリビット

シンボル



説明

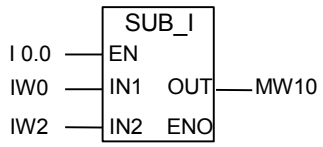
OV メモリビット命令を使用すれば、直前の数値演算にオーバーフロー(OV)が発生したかどうかを調べることができます。実行された演算ファンクションの結果が負または正の許容範囲にない場合、ステータスワードの OV ビット(「CPU レジスタ」も参照)が設定されます。この命令により、このビットの信号状態がチェックされます。数値演算にエラーがない場合、このビットはリセットされます。

ステータスワード

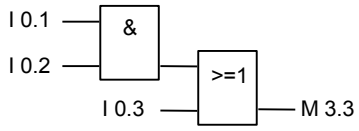
	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例

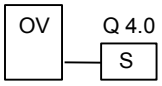
ネットワーク 1



ネットワーク 2



ネットワーク 3

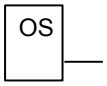


入力 I0.0 の信号状態が 1 になると、SUB_I ボックスが実行されます。入力ワード IW0－IW2 の演算結果が、整数の許容範囲外にある場合、ステータスワードの OV ビットが設定されます。OV で信号状態をチェックすると、1 になります。OV でのチェックが 1 でネットワーク 2 の RLO が 1 の場合、出力 Q4.0 が設定されます(出力 Q4.0 の前の RLO が 1 の場合)。

入力 0 の信号状態が 0(起動していない)の場合、EN と ENO の両方の信号状態は 0 になります。EN の信号状態が 1(起動している)であって、数値演算ファンクションの結果が範囲外であると、ENO の信号状態は 0 になります。

12.3 OS : 例外ビット保存オーバーフロー

シンボル



説明

論理和演算でこの命令を使用すると、AND 真理値表に従って、チェック結果と前回の論理演算結果(RLO)が結合されます。論理和演算では、OR 真理値表が使用されます。

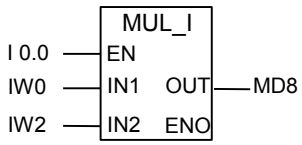
例外ビット保存オーバーフロー命令を使用すれば、数値演算で前回にオーバーフロー(保存されたオーバーフロー、OS)が発生したかどうかを調べることができます。実行された数値演算の結果が負または正の有効範囲にない場合、ステータスワードの OS ビット CPU レジスタも参照)が設定されます。この命令により、このビットの信号状態がチェックされます。OS ビットは OV(オーバーフロー)ビットとは異なり、その後の数値演算でエラーが発生しなくても設定されたままになります(OV メモリビットも参照)。

ステータスワード

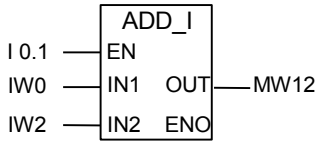
	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例

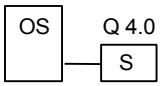
ネットワーク 1



ネットワーク 2



ネットワーク 3



入力 I0.0 の信号状態が 1 になると、MUL_I ボックスが実行されます。入力 I0.1 の信号状態が 1 になると、ADD_I ボックスが実行されます。いずれかの数値演算の結果が、整数の有効範囲にない場合、ステータスワードの OS ビットが設定されます。

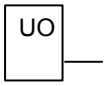
OS の信号状態チェックの結果は 1 で、出力 Q4.0 が設定されます。

ネットワーク 1:入力 10.1 の信号状態が 0(起動していない)の場合、EN と ENO の両方の信号状態は 0 になります。EN の信号状態が 0(起動している)であって、数値演算ファンクションの結果が範囲外であると、ENO の信号状態は 0 になります。

ネットワーク 2:入力 10.1 の信号状態が 0(起動していない)の場合、EN と ENO の両方の信号状態は 0 になります。EN の信号状態が 1(起動している)であって、数値演算ファンクションの結果が範囲外であると、ENO の信号状態は 0 になります。

12.4 UO : UO メモリビット

シンボル



説明

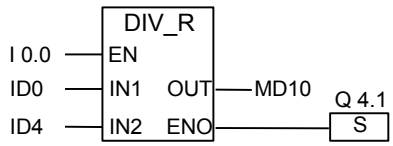
UO メモリビット命令を使用すれば、浮動小数点数値演算ファンクションの結果が無効かどうか(つまり、演算ファンクション内の値のいずれかが有効な浮動小数点数でないかどうか)をチェックすることができます。ステータスワードの条件コードビット(CC 1 と CC 0、「CPU レジスタ」を参照)が評価されます。数値演算ファンクションの結果が無効(UO)であると、信号状態チェックの結果は 1 になります CC 1 と CC 0 の組み合わせが無効でない場合、信号状態チェックの結果は 0 になります。

ステータスワード

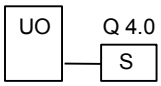
	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例

ネットワーク 1



ネットワーク 2



入力 I0.0 の信号状態が 1 になると、DIV_R ボックスが実行されます。入力ダブルワード ID0 か ID4 のいずれかの値が有効な浮動小数点数でない場合は、浮動小数点数値演算ファンクションが無効になります。
命令実行中にエラーが発生すると、ENO の検出結果は 0 になります。

ファンクション DIV_R が実行されても、演算ファンクション内の値のいずれかが有効な浮動小数点数でない場合は、Q4.0 が設定されます。入力 I0.0 の信号状態が 0 の場合、EN と ENO の信号状態は 0 です。

12.5 BR : BR メモリビット

シンボル

English	German
BR	BIE

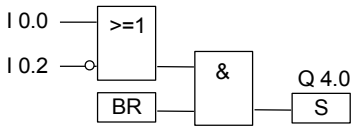
説明

BR メモリビット命令を使用すれば、ステータスワードの BR ビット(バイナリリザルト)の信号状態をチェックすることができます(「CPU レジスタ」を参照)。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

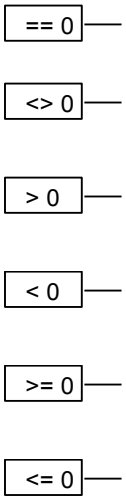
例



入力 I0.0 の信号状態が 1、または入力 I0.2 の信号状態が 0 で、この RLO に加えて、BR ビットの信号状態が 1 である場合、出力 Q4.0 が設定されます。

12.6 <> 0 : リザルトビット

シンボル



0 と等しいのリザルトビット命令は、演算命令結果が 0 と等しいかどうかを判断します。

0 と等しくないのリザルトビット命令は、演算命令結果が 0 と等しくないかどうかを判断します。

0 より大きいのリザルトビット命令は、演算命令結果が 0 より大きいかどうかを判断します。

0 より小さいのリザルトビット命令は、演算命令結果が 0 より小さいかどうかを判断します。

0 以上のリザルトビット命令は、演算命令結果が 0 以上かどうかを判断します。

0 以下のリザルトビット命令は、演算命令結果が 0 以下かどうかを判断します。

説明

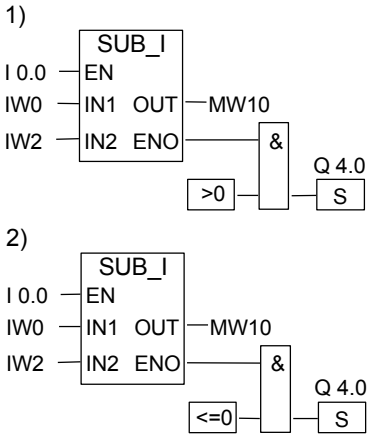
リザルトビット命令を使用すれば、演算ファンクションの結果と 0 の関係、つまり結果が==0、<>0、>0、<0、>=0、<=0 のどれかを判断することができます。ステータスワードの条件コードビット(CC 1 と CC 0、「CPU レジスタ」を参照)が評価されます。アドレスで示されている比較条件を満たしている場合、信号状態チェックの結果は 1 になります。

論理積演算でこの命令を使用すると、AND 真理値表に従って、チェック結果と前回の論理演算結果(RLO)が結合されます。論理和演算でこの命令を使用した場合は、OR 真理値表に従って、チェック結果と前回の RLO が結合されます。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例



入力 I0.0 の信号状態が 1 になると、SUB_I ボックスが実行されます。入力ワード IW0 の値が入力ワード IW2 の値よりも大きい場合、数値演算ファンクション IW0 - IW2 の結果は 0 よりも大きくなります。EN の信号状態が 1(起動している)の場合、命令が実行されている間にエラーが発生すると、ENO の信号状態は 0 になります。

- ファンクションが正しく設定され、結果が 0 以下の場合に出力 Q4.0 が設定されます。入力 I0.0 の信号状態が 0(実行されない)の場合は、EN と ENO の両方の信号状態が 0 になります。
- ファンクションが正しく設定され、結果が 0 以下の場合に出力 Q4.0 が設定されます。入力 I0.0 の信号状態が 0(実行されない)の場合は、EN と ENO の両方の信号状態が 0 になります。

13 タイマ命令

13.1 タイマ命令の概要

説明

正しい時間の設定と選択については、「メモリ内のタイマの場所およびタイマのコンポーネント」を参照してください。

使用可能なタイマ命令を次に示します。

- S_PULSE : パルスタイマパラメータの割り付けと起動
- S_PEXT : 拡張パルスタイマパラメータの割り付けと起動
- S_ODT : オンディレイタイマパラメータの割り付けと起動
- S_ODTS : 拡張オンディレイタイマパラメータの割り付けと起動
- S_OFFDT : オフディレイタイマパラメータの割り付けと起動
- SP : パルスタイマの起動
- SE : 拡張パルスタイマの起動
- SD : オンディレイタイマの起動
- SS : 拡張オンディレイタイマの起動
- SF : オフディレイタイマの起動

13.2 タイマのメモリ領域と構成要素

メモリ領域

タイムには、CPU のメモリ内に専用の領域が割り付けられています。このメモリ領域は、各タイマアドレスに対して 1 つの 16 ビットワードを確保します。FBD でプログラムを行う場合は、256 個のタイマが使用できます。使用している CPU の技術情報を参照して、使用可能なタイマワードの数をチェックしてください。

以下のファンクションが、タイマのメモリ領域へアクセスします。

- タイマ命令
- クロックタイミングによるタイマワードの更新。RUN モードの場合、この CPU ファンクションによって、時間値がゼロになるまで、タイムベースで指定されている間隔で、与えられている時間値が 1 単位ずつデクリメントされます。

時間値

タイマワードのビット 0～9 には、時間値がバイナリコードで格納されます。時間値では、多数の単位が指定されます。タイマが更新される場合、タイマ値はタイムベースで指定された間隔で 1 単位ずつ減少します。タイマ値は 0 になるまで減少し続けます。

次のいずれかのフォーマットを使用して、時間値を事前にロードすることができます。

- S5T#a**H**_b**M**_c**S**_d**MS**
  - ここで 、H = 時間、M = 分、S = 秒、MS = ミリ秒を表します。
    a、b、c、d はユーザー定義です。
  - タイムベースは自動的に選択され、タイマ値は選択されたタイムベースごとに一単位ずつ減少します。

入力できる最大タイマ値は、9,990 秒または 2H_46M_30S です。

S5TIME#4S = 4 秒
s5t#2h_15m = 2 時間 15 分
S5T#1H_12M_18S = 1 時間 12 分 18 秒

タイムベース

タイマワードのビット 12 とビット 13 には、タイムベースがバイナリコードで格納されます。タイムベースは、時間値が一単位ずつ減少する間隔を定義します。最小タイムベースは 10 ms で、最大タイムベースは 10 s です。

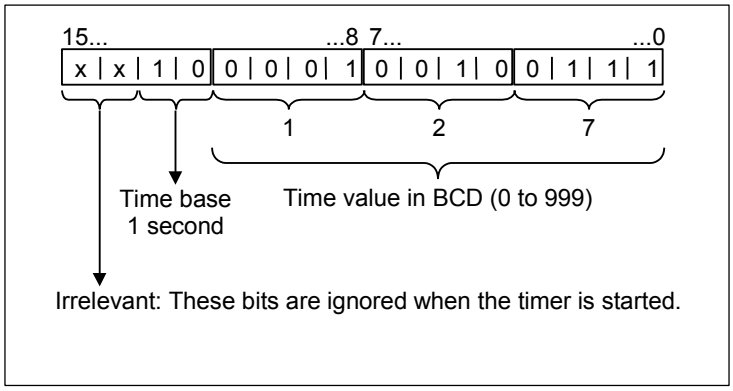
タイムベース	タイムベースのバイナリコード
10 ms	00
100 ms	01
1 秒	10
10 秒	11

タイマ値は一定の時間間隔で保存されるだけなので、この時間間隔の倍数でないタイマ値は、時間間隔の倍数に切り捨てられます。必要な範囲に対し分解能が高すぎる場合は、その範囲に合わせて値が切り捨てられ、目的の分解能に合わせて値が切り捨てられることはありません。下表は、可能な分解能とその範囲を示したものです。

分解能	タイムベース
0.01 秒	10MS ～ 9S_990MS
0.1 秒	100MS～1M_39S_900MS
1 秒	1S～16M_39S
10 秒	10S～2HR_46M_30S

タイマセルのビットコンフィグレーション

タイマを起動する場合、タイマセルの内容が時間値として使用されます。タイマセルのビット 0～11 には、2 進化 10 進フォーマットの時間値が含まれます(BCD フォーマット: 4 ビットで構成される各グループには、1 つの 10 進値のバイナリコードが含まれます)。ビット 12 とビット 13 には、タイムベースがバイナリコードで格納されます。下図は、タイマ値を 127 とし、タイムベースを 1 秒としてロードした場合のタイマセルの内容を表したものです。

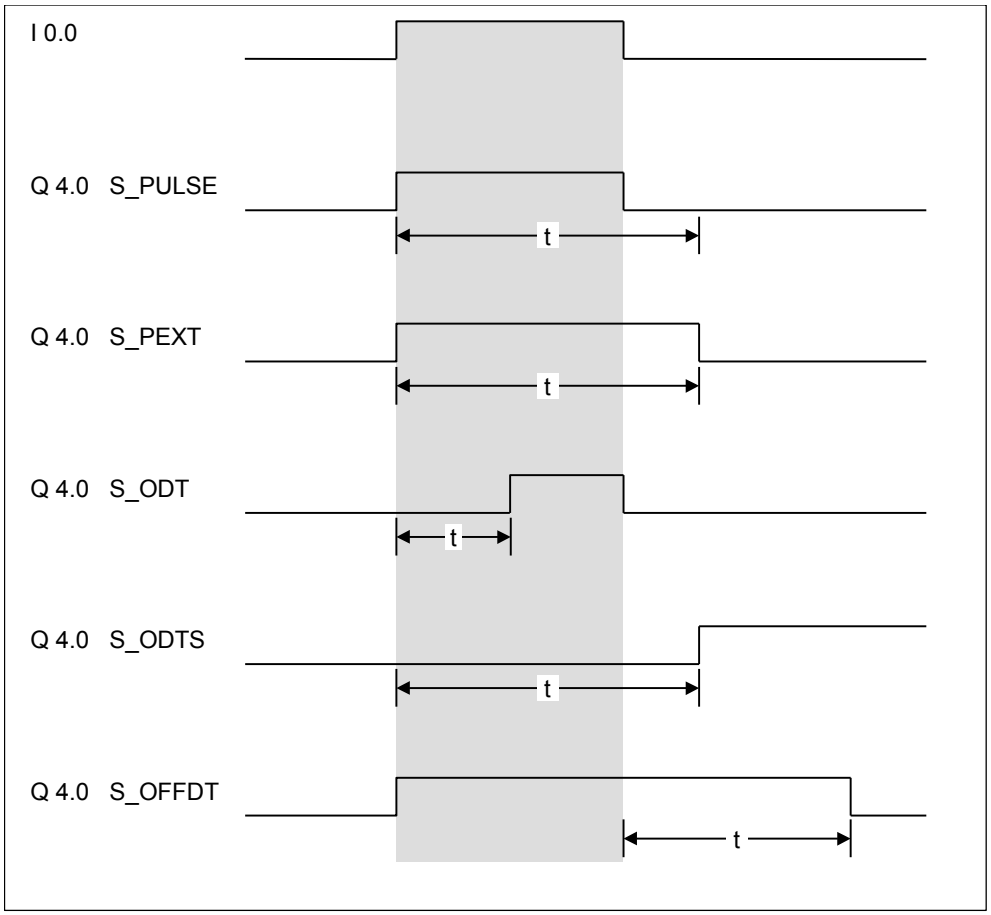


時間値とタイムベースの読み取り

1 つのタイマボックスに対し、BI と BCD の 2 つの出力があり、この 2 つの出力にワード位置を指定できます。BI 出力では、時間値がバイナリフォーマットで示され、タイムベースは表示されません。BCD 出力では、タイムベースと時間値が 2 進化 10 進(BCD)フォーマットで示されます。

適切なタイマの選択

この概要を参考にすれば、タイミングジョブに適切なタイマを選択できます。

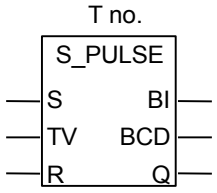


タイマ	説明
S_PULSE パルスタイマ	出力信号が 1 となる最大時間は、プログラムされたタイマ値 t と同じです。入力信号が 0 に変わる場合は、出力信号が短い間だけ 1 になります。
S_PEXT 拡張パルスタイマ	入力信号が 1 である時間に関係なくプログラムされた時間 t の間、出力信号は 1 です。
S_ODT オンディレイタイマ	プログラムされた時間 t が経過し、入力信号が 1 である場合にだけ、出力信号が 1 に変わります。
S_ODTS 拡張オンディレイタイマ	プログラムされた時間 t が経過すると、入力信号が 0 のままかどうかに関係なく、出力信号が 1 になります。
S_OFFDT オフディレイタイマ	入力信号が 1 に変わる場合、またはタイマが作動している間、出力信号は 1 になります。入力信号が 1 から 0 に変わると、タイマが開始します。

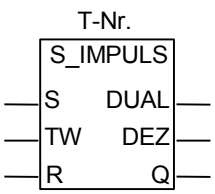
13.3 S_PULSE : パルスタイマパラメータの割り付けと起動

シンボル

English



German



パラメータ 英語	パラメータ ドイツ語	データタイプ	メモリ領域	説明
no.	Nr.	TIMER	T	タイマの番号。 値の範囲は、CPU により異なる。
S	S	BOOL	I、Q、M、D、L、T、C	起動入力
TV	TW	S5TIME	I、Q、M、D、L または 定数	タイマ値のプリセット(範囲 0～9999)
R	R	BOOL	I、Q、M、D、L、T、C	リセット入力
BI	DUAL	WORD	I、Q、M、D、L	残りのタイマ値 (整数フォーマット)
BCD	DEZ	WORD	I、Q、M、D、L	残り時間 (BCD フォーマットの値)
Q	Q	BOOL	I、Q、M、D、L	タイマのステータス

説明

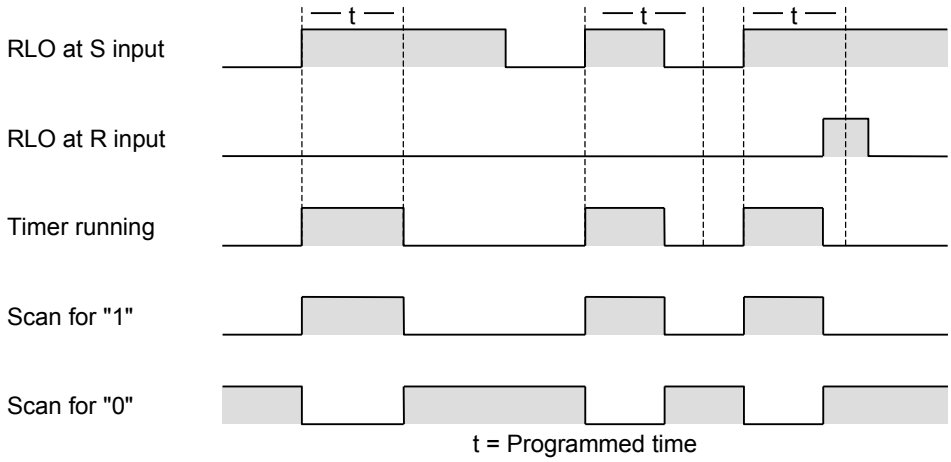
パルスタイマパラメータの割り付けと起動命令は、起動(S)入力に信号立ち上がり(信号状態の 0 から 1 への遷移)がある場合、指定されているタイマが起動します。タイマを起動するには、必ず信号の遷移が必要です。タイマは、入力 TV の信号状態が 1 の場合、プログラムされた時間が経過するまで、タイマ値(TV)入力で指定された時間、動作を継続します。タイマが動作している間、出力 Q の信号状態チェックで値 1 が返されます。指定された時間に達するまでに、S 入力で 1 から 0 への変更があると、タイマは停止します。その場合、出力 Q での信号状態チェック結果は 0 になります。

タイマの実行中にタイマのリセット(R)入力で信号状態が 0 から 1 に遷移した場合、タイマがリセットされます。また、この場合、時間値とタイムベースがゼロにリセットされます。ただし、タイマが実行中でない場合は、タイマの R 入力の信号状態が 1 であっても、影響はありません。

タイマ値は、出力 BI と BCD でモニタすることができます。BI の時間値はバイナリフォーマットで、BCD の場合は 2 進化 10 進フォーマットです。

タイムチャート

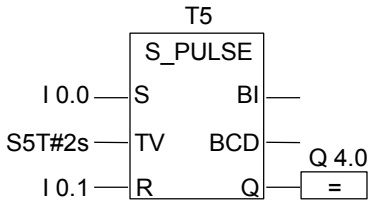
パルスタイマの特性



ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例

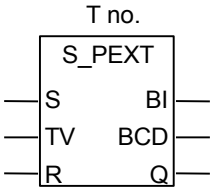


入力 I0.0 の信号状態が 0 から 1 へ変わる場合(RLO に立ち上がりエッジがある場合)、タイマ T5 は起動します。入力 10.0 が 1 である場合、タイマは指定された 2 秒間実行を継続します。時間が経過する前に、入力 I0.0 の信号状態が 1 から 0 に遷移した場合、タイマは停止されます。また、タイマの実行中に入力 I0.1 の信号状態が 0 から 1 に遷移した場合、タイマはリセットされます。タイマが実行中であれば、出力 Q4.0 の信号状態は 1 のままです。

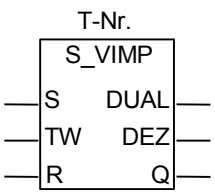
13.4 S_PEXT : 拡張パルスタイマパラメータの割り付けと起動

シンボル

English



German



パラメータ 英語	パラメータ ドイツ語	データタイプ	メモリ領域	説明
no.	Nr.	TIMER	T	タイマの番号。 値の範囲は、CPU により異なる。
S	S	BOOL	I、Q、M、D、L、T、C	起動入力
TV	TW	S5TIME	I、Q、M、D、L または 定数	タイマ値のプリセット(範囲 0～9999)
R	R	BOOL	I、Q、M、D、L、T、C	リセット入力
BI	DUAL	WORD	I、Q、M、D、L	残りのタイマ値 (整数フォーマット)
BCD	DEZ	WORD	I、Q、M、D、L	残り時間 (BCD フォーマットの値)
Q	Q	BOOL	I、Q、M、D、L	タイマのステータス

説明

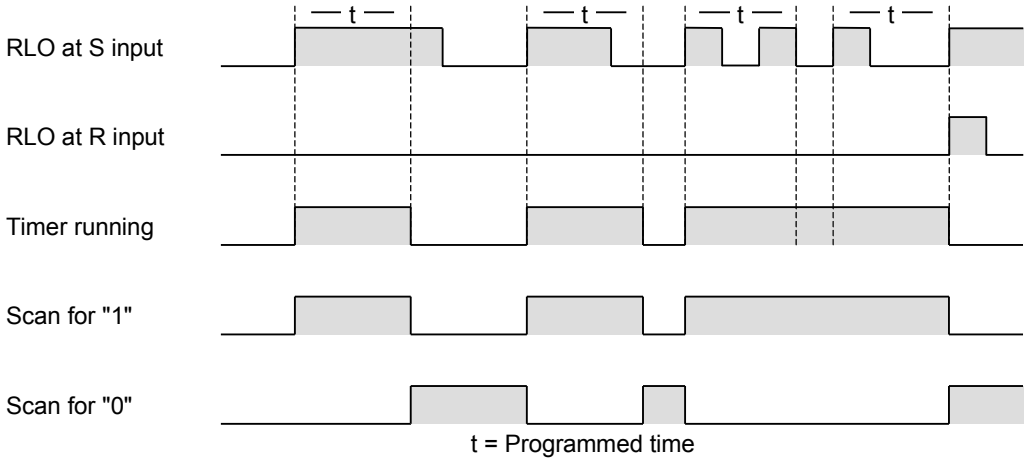
拡張パルスタイマパラメータの割り付けと起動命令は、起動(S)入力に信号立ち上がり(信号状態の 0 から 1 への遷移)がある場合、指定されているタイマが起動します。タイマを起動するには、必ず信号の遷移が必要です。指定されている時間が切れる前に S 入力の信号状態が 0 に遷移した場合でも、時間値(TV)入力に指定されている時間タイマは実行を続けます。タイマが実行中の場合、出力 Q での信号状態チェックで値 1 が返されます。また、タイマの実行中に入力 S の信号状態が 0 から 1 に遷移した場合は、タイマは再起動され、指定されている時間実行を続けます。

タイマの実行中にタイマのリセット(R)入力の信号状態が 0 から 1 に遷移した場合は、タイマがリセットされます。また、この場合、時間値とタイムベースがゼロにリセットされます。

現在の時間は、出力 BI および BCD で確認できます。BI の時間値はバイナリフォーマットで、BCD の場合は 2 進化 10 進フォーマットです。

タイムチャート

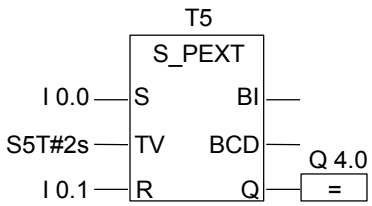
拡張パルスタイマの特性



ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例

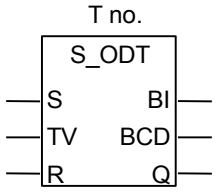


入力 I0.0 の信号状態が 0 から 1 に遷移した(RLO に信号立ち上がりがある)場合、タイマ T5 が起動されます。このタイマは、入力 S の信号立ち下がりに関係なく、2 秒(2s)の指定時間だけ実行を継続します。指定時間が切れる前に入力 I0.0 の信号状態が 0 から 1 に変わると、タイマは再起動されます。また、タイマの実行中に入力 I0.1 の信号状態が 0 から 1 に遷移した場合、タイマはリセットされます。タイマが実行中であれば、出力 Q4.0 の信号状態は 1 のままです。

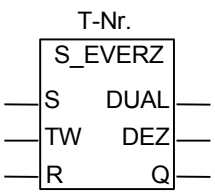
13.5 S_ODT : オンディレイタイマパラメータの割り付けと起動

シンボル

English



German



パラメータ 英語	パラメータ ドイツ語	データタイプ	メモリ領域	説明
no.	Nr.	TIMER	T	タイマの番号。 値の範囲は、CPU により異なる。
S	S	BOOL	I、Q、M、D、L、T、C	起動入力
TV	TW	S5TIME	I、Q、M、D、L または 定数	タイマ値のプリセット(範囲 0～9999)
R	R	BOOL	I、Q、M、D、L、T、C	リセット入力
BI	DUAL	WORD	I、Q、M、D、L	残りのタイマ値 (整数フォーマット)
BCD	DEZ	WORD	I、Q、M、D、L	残り時間 (BCD フォーマットの値)
Q	Q	BOOL	I、Q、M、D、L	タイマのステータス

説明

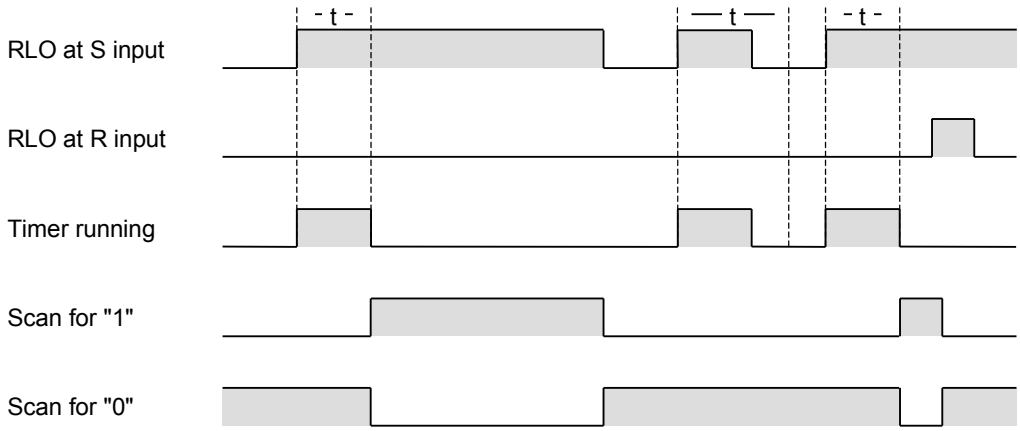
オンディレイタイマパラメータの割り付けと起動命令は、起動(S)入力に信号立ち上がり(信号状態が 0 から 1 に変わること)がある場合に、指定されたタイマを起動します。タイマを起動するには、必ず信号の遷移が必要です。タイマは、入力 S の信号状態が 1 の場合、タイマ値(TV)入力で指定された時間、動作を継続します。エラーが発生せずに時間が経過した場合、および入力 S の信号状態が 1 のままの場合、出力 Q の信号状態チェックで値 1 が返されます。タイマの実行中に入力 S の信号状態が 1 から 0 に遷移した場合、タイマは停止します。この場合、出力 Q の信号状態チェックの結果は 0 です。

タイマの実行中にタイマのリセット(R)入力の信号状態が 0 から 1 に遷移した場合は、タイマがリセットされます。また、この場合、時間値とタイムベースがゼロにリセットされます。さらに、タイマが実行中でなくても、R 入力の信号状態が 1 になると、タイマがリセットされます。

現在の時間は、出力 BI および BCD で確認できます。BI の時間値はバイナリフォーマットで、BCD の場合は 2 進化 10 進フォーマットです。

タイムチャート

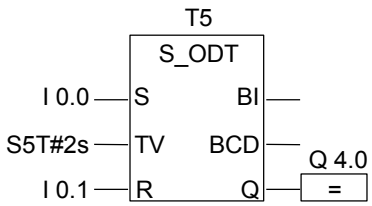
オンディレイタイマの特性



ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	X	X	X	1

例

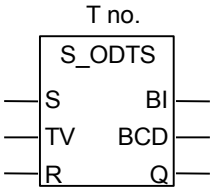


入力 I0.0 の信号状態が 0 から 1 に遷移した(RLO に信号立ち上がりがある)場合、タイマ T5 が起動されます。指定された 2 秒間(2s)が経過し、入力 10.0 の信号状態が 1 のままであれば、出力 Q4.0 の信号状態は 1 になります。入力 10.0 の信号状態が 1 から 0 に遷移した場合、タイマは停止し、出力 Q4.0 は 0 になります。タイマの実行中に入力 10.1 の信号状態が 0 から 1 に遷移した場合、タイマは再起動します。

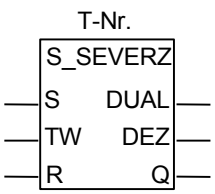
13.6 S_ODTS : 拡張オンディレイタイマパラメータの割り付けと起動

シンボル

English



German



パラメータ 英語	パラメータ ドイツ語	データタイプ	メモリ領域	説明
no.	Nr.	TIMER	T	タイマの番号。 値の範囲は、CPU により異なる。
S	S	BOOL	I、Q、M、D、L、T、C	起動入力
TV	TW	S5TIME	I、Q、M、D、L または 定数	タイマ値のプリセット(範囲 0～9999)
R	R	BOOL	I、Q、M、D、L、T、C	リセット入力
BI	DUAL	WORD	I、Q、M、D、L	残りのタイマ値 (整数フォーマット)
BCD	DEZ	WORD	I、Q、M、D、L	残り時間 (BCD フォーマットの値)
Q	Q	BOOL	I、Q、M、D、L	タイマのステータス

説明

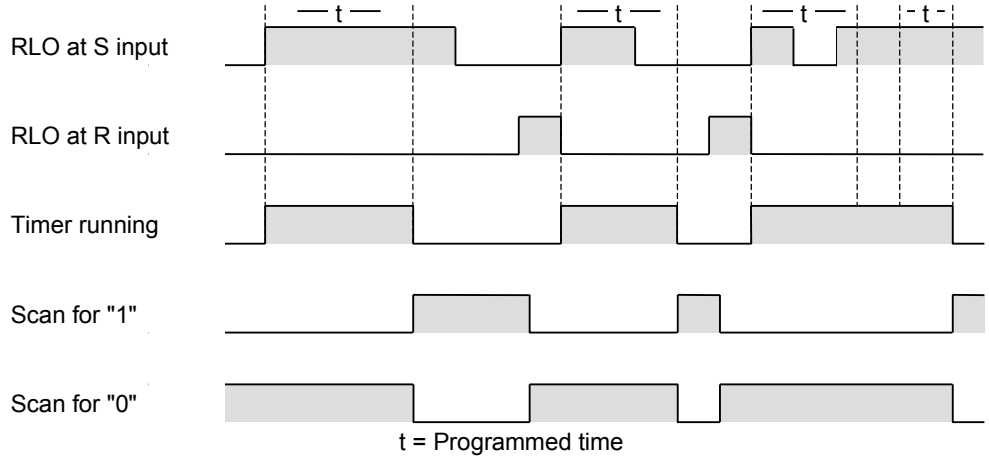
拡張オンディレイタイマパラメータの割り付けと起動命令は、起動(S)入力に信号立ち上がり(信号状態が 0 から 1 に変わること)がある場合に、指定されたタイマを起動します。タイマを起動するには、必ず信号の遷移が必要です。タイマは、入力 S の信号状態が 0 に変わっても、タイマ値(TV)入力で指定された時間が経過するまでは作動し続けます。この時間が経過すると、リセット入力(R)が 0 のままである場合の入力 S の信号状態に関係なく、出力 Q の信号状態チェックで値 1 が返されます。タイマの実行中に入力 S の信号状態が 0 から 1 に遷移した場合、タイマは指定された時間に再起動します。

タイマのリセット(R)入力で信号状態が 0 から 1 に遷移した場合は、入力 S の RLO に関係なく、タイマはリセットされます。

現在の時間は、出力 BI および BCD で確認できます。BI の時間値はバイナリフォーマットで、BCD の場合は 2 進化 10 進フォーマットです。

タイムチャート

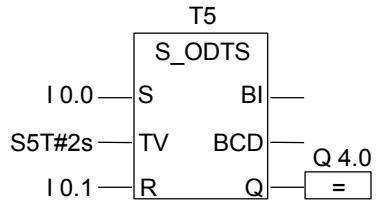
拡張オンディレイタイマの特性



ステータスワード

	BR	CC 1	CC 0	OV	OS	OR	STA	RLO	/FC
書き込みの内容:	-	-	-	-	-	x	x	x	1

例

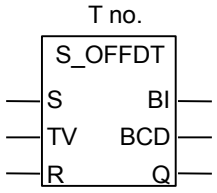


入力 I0.0 の信号状態が 0 から 1 に遷移した(RLO に信号立ち上がりがある)場合、タイマ T5 が起動されます。入力 10.0 の信号状態が 1 から 0 に遷移するかどうかにかかわらず、タイマは実行を継続します。指定された時間が経過する前に入力 10.0 の信号状態が 0 から 1 に遷移した場合、タイマは再起動されます。タイマ作動中に入力 I0.1 の信号状態が 0 から 1 へ変わると、タイマは再起動します。指定されている時間が切れ、I0.1 の信号状態が 0 のままであれば、出力 Q4.0 の信号状態は 1 になります。

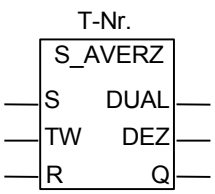
13.7 S_OFFDT : オフディレイタイマパラメータの割り付けと起動

シンボル

English



German



パラメータ 英語	パラメータ ドイツ語	データタイプ	メモリ領域	説明
no.	Nr.	TIMER	T	タイマの番号。 値の範囲は、CPU により異なる。
S	S	BOOL	I、Q、M、D、L、T、C	起動入力
TV	TW	S5TIME	I、Q、M、D、L または定数	タイマ値のプリセット(範囲 0～9999)
R	R	BOOL	I、Q、M、D、L、T、C	リセット入力
BI	DUAL	WORD	I、Q、M、D、L	残りのタイマ値 (整数フォーマット)
BCD	DEZ	WORD	I、Q、M、D、L	残り時間 (BCD フォーマットの値)
Q	Q	BOOL	I、Q、M、D、L	タイマのステータス

説明

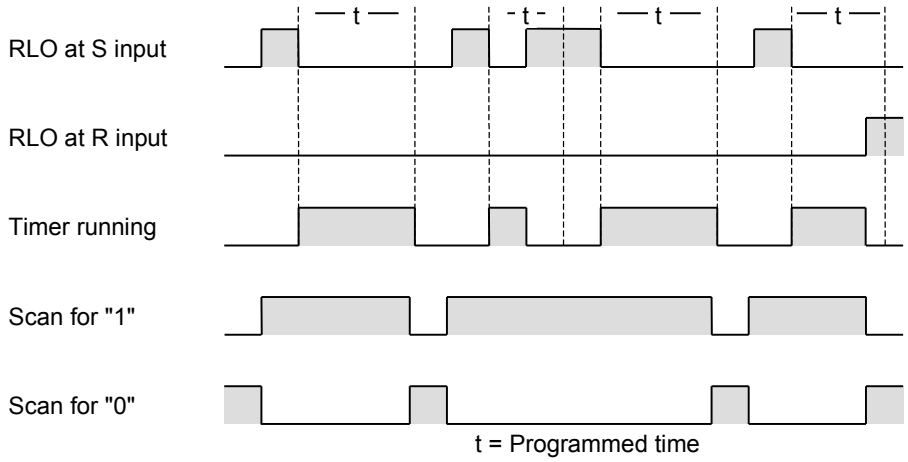
オフディレイタイマパラメータの割り付けと起動命令は、起動(S)入力に信号立ち下がり(信号状態が 1 から 0 に変わること)がある場合に、指定されたタイマを起動します。タイマを起動するには、必ず信号の遷移が必要です。入力 S の信号状態が 1 の場合またはタイマが実行中であれば、出力 Q の信号状態チェックで値 1 が返されます。タイマの実行中に、入力 S の信号状態が 0 から 1 に遷移した場合は、タイマはリセットされます。入力 S の信号状態が再度 1 から 0 に遷移するまで、タイマは再起動されません。

タイマの実行中にタイマのリセット(R)入力の信号状態が 0 から 1 に遷移した場合は、タイマがリセットされます。

タイマ値は、出力 BI と BCD でスキャンされます。BI の時間値はバイナリフォーマットで、BCD の場合は 2 進化 10 進フォーマットです。

タイムチャート

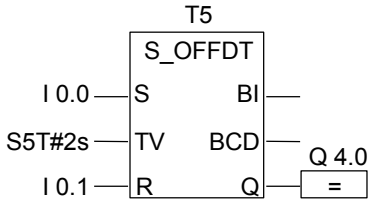
オフディレイタイマの特性



ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	x	x	x	1

例

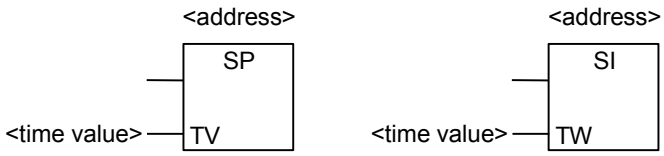


入力 I0.0 の信号状態が 1 から 0 に変わる場合、タイマは起動します。I0.0 が 1 である場合、またはタイマが作動している場合、出力 Q4.0 は 1 です。タイマの実行中に I0.1 の信号状態が 0 から 1 に遷移した場合は、タイマがリセットされます。

13.8 SP : パルスタイマの起動

シンボル

English **German**



パラメータ 英語	パラメータ ドイツ語	データタイプ	メモリ領域	説明
タイマ番号	タイマ番号	TIMER	T	アドレスは、起動するタイマの番号を表す
TV	TW	S5TIME	E、A、M、D、L または定数	タイマ値(S5TIME フォーマット)

説明

パルスタイマの起動命令は、RL0 に信号立ち上がりエッジ(信号状態が 0 から 1 への遷移)がある場合、指定されているタイマを起動します。RLO が正である限り、指定されている値を使用してタイマは実行を続けます。タイマが実行中の場合、信号状態チェックの結果は 1 になります。タイマの指定時間が切れる前に RLO の状態が 1 から 0 に遷移した場合は、タイマは停止します。この場合、信号状態チェックの結果は 0 になります。

メモリ領域とタイマの構成要素の詳細については、メモリ領域を参照してください。

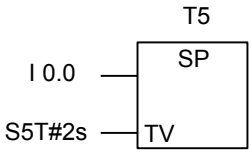
パルスタイマの起動ボックスは、論理文字列の右端にしか配置することができません。**パルスタイマの起動**ボックスの番号を使用できます。

ステータスワード

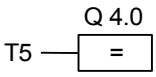
	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	-	-	0

例

ネットワーク 1



ネットワーク 2

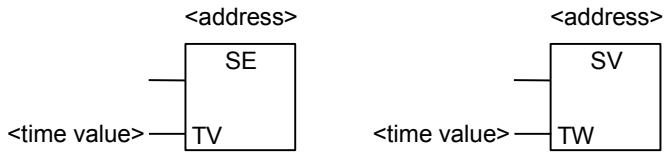


入力 I0.0 の信号状態が 0 から 1 に遷移した(RLO に信号立ち上がりがある)場合、タイマ T5 が起動されます。信号状態が 1 である限り、指定されている時間(2 秒間)タイマは実行を続けます。タイマの実行中に I0.0 の信号状態が 1 から 0 に遷移した場合、タイマは停止します。
タイマ実行時に出力 Q4.0 は 1 になります。

13.9 SE : 拡張パルスタイマの起動

シンボル

English **German**



パラメータ 英語	パラメータ ドイツ語	データタイプ	メモリ領域	説明
タイマ番号	タイマ番号	TIMER	T	アドレスは、起動するタイマの番号を表す
TV	TW	S5TIME	E、A、M、D、L または定数	タイマ値(S5TIME フォーマット)

説明

拡張パルスタイマの起動命令は、RL0 に信号立ち上がりエッジ(信号状態が 0 から 1 への遷移)がある場合、指定されているタイマを起動します。タイマの時間が切れる前に、RLO の状態が 0 に遷移した場合、指定された時間タイマは実行を続けます。タイマが実行中の場合、信号状態チェックの結果は 1 になります。タイマの実行中に RLO の状態が 0 から 1 に遷移した場合、タイマは再起動され、指定されている時間実行を続けます。

メモリ領域とタイマの構成要素の詳細については、メモリ領域を参照してください。

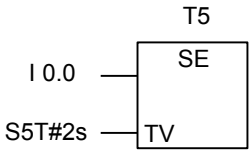
拡張パルスタイマの起動ボックスは、論理文字列の右端にしか配置することができません。**拡張パルスタイマ**ボックスの番号を使用できます。

ステータスワード

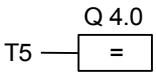
	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	-	-	0

例

ネットワーク 1



ネットワーク 2

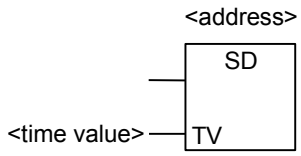


入力 I0.0 の信号状態が 0 から 1 に遷移した(RLO に信号立ち上がりがある)場合、タイマ T5 が起動されます。タイマは、RLO 内の信号立ち下がりに関係なく、実行を続けます。指定されている時間が切れる前に入力 I0.0 の信号状態が 0 から 1 に遷移した場合、タイマは再起動されます。
タイマ実行時に出力 Q4.0 は 1 になります。

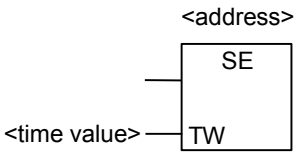
13.10 SD : オンディレイタイマの起動

シンボル

English



German



パラメータ 英語	パラメータ ドイツ語	データタイプ	メモリ領域	説明
タイマ番号	タイマ番号	TIMER	T	アドレスは、起動するタイマの番号を表す
TV	TW	S5TIME	E、A、M、D、L または定数	タイマ値(S5TIME フォーマット)

説明

オンディレイタイマの起動命令は、RL0 に信号立ち上がりエッジ(信号状態が 0 から 1 への遷移)がある場合、指定されているタイマを起動します。エラーが発生せずに指定された時間が経過し、RLO の値が 1 のままである場合、信号状態チェックの結果は 1 になります。タイマの実行中に RLO が 1 から 0 に遷移すると、タイマは停止します。この場合、信号状態チェックの結果は常に 0 になります。

メモリ領域とタイマの構成要素の詳細については、メモリ領域を参照してください。

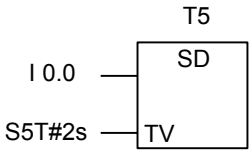
オンディレイタイマの起動ボックスは、論理文字列の右端にしか配置することができません。**オンディレイタイマ**ボックスの番号を使用できます。

ステータスワード

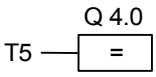
	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	-	-	0

例

ネットワーク 1



ネットワーク 2

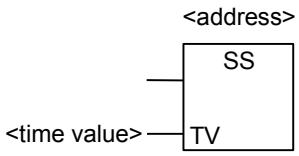


入力 I0.0 の信号状態が 0 から 1 に遷移した(RLO に信号立ち上がりがある)場合、タイマ T5 が起動されます。時間が経過しても 10.0 の信号状態が 1 のままである場合、出力 Q4.0 は 1 になります。信号状態が 1 から 0 に遷移した場合、タイマは停止します。

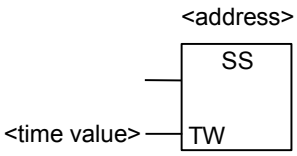
13.11 SS : 拡張オンディレイタイマの起動

シンボル

English



German



パラメータ 英語	パラメータ ドイツ語	データタイプ	メモリ領域	説明
タイマ番号	タイマ番号	TIMER	T	アドレスは、起動するタイマの番号を表す
TV	TW	S5TIME	E、A、M、D、L または定数	タイマ値(S5TIME フォーマット)

説明

拡張オンディレイタイマの起動命令は、RL0 に信号立ち上がりエッジ(信号状態の 0 から 1 への遷移)がある場合、指定されているタイマを起動します。タイマの時間が切れる前に、RLO の状態が 0 に遷移した場合、指定された時間タイマは実行を続けます。RL0 の状態に関係なくタイマの時間が切れた場合、信号状態チェックの結果は 1 になります。タイマの実行中に RLO の状態が 0 から 1 に遷移した場合、タイマは再起動され、指定されている時間実行を続けます。

メモリ領域とタイマの構成要素の詳細については、メモリ領域を参照してください。

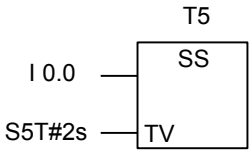
拡張オンディレイタイマの起動ボックスは、論理文字列の右端にしか配置することができません。**拡張オンディレイタイマの起動**ボックスの番号を使用できます。

ステータスワード

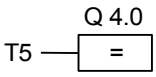
	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	-	-	0

例

ネットワーク 1



ネットワーク 2

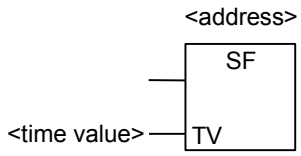


入力 I0.0 の信号状態が 0 から 1 に遷移した(RLO に信号立ち上がりがある)場合、タイマ T5 が起動されます。入力 10.0 の信号状態が 1 から 0 に遷移するかどうかにかかわらず、タイマは実行を継続します。指定された時間が経過する前に入力 10.0 の信号状態が 0 から 1 に遷移した場合、タイマは再起動されます。
タイマの時間が切れる場合、出力 Q4.0 は 1 になります。

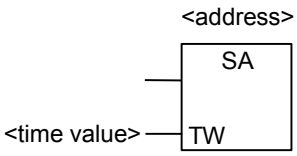
13.12　SF　オフディレイタイマの起動

シンボル

English



German



パラメータ 英語	パラメータ ドイツ語	データタイプ	メモリ領域	説明
タイマ番号	タイマ番号	TIMER	T	アドレスは、起動するタイマの番号を表す
TV	TW	S5TIME	E、A、M、D、L または定数	タイマ値(S5TIME フォーマット)

説明

オフディレイタイマの起動命令は、RL0 に信号立ち上がりエッジ(信号状態が 0 から 1 への遷移)がある場合、指定されているタイマを起動します。RLO が 1 であるか、またはタイマが実行中の場合、信号状態チェックの結果は 1 になります。タイマの実行中に、RLO の状態が 0 から 1 に遷移した場合は、タイマはリセットされます。RLO の状態が 1 から 0 に遷移した場合、タイマは再起動されます。

メモリ領域とタイマの構成要素の詳細については、メモリ領域を参照してください。

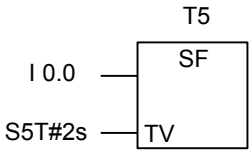
オフディレイタイマの起動ボックスは、論理文字列の右端にしか配置することができません。**オフディレイタイマ**ボックスの番号を使用できます。

ステータスワード

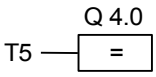
	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	-	-	-	-	-	0	-	-	0

例

ネットワーク 1



ネットワーク 2



入力 I0.0 の信号状態が 1 から 0 に変わる場合、タイマは起動します。

信号状態が 0 から 1 へ変わると、タイマはリセットされます。

入力 I0.0 の信号状態が 1 であるかまたはタイマが実行中の場合、出力 Q4.0 の信号状態は 1 になります。

14 ワード論理命令

14.1 ワード論理命令の概要

説明

ワード論理演算命令により、ワード(16 ビット) やダブルワード(32 ビット) のペアをブールロジックに従ってビットごとに比較します。

出力 OUT に示される結果は 0 との相対値になり、ステータスワードの CC1 ビットが次のように変更されます。

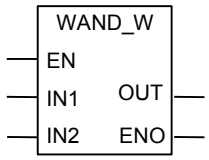
- 出力 OUT に示される結果が 0 でないと、ステータスワードの CC1 ビットは 1 に設定されます。
- 出力 OUT に示される結果が 0 であれば、ステータスワードの CC1 ビットは 0 に設定されます。

ワード論理演算には以下の命令を使用できます。

- WAND_W : AND ワード(ワード)
- WOR_W : OR ワード(ワード)
- WXOR_W : 排他的 OR ワード(ワード)
- WAND_DW : AND ダブルワード(ワード)
- WOR_DW : OR ダブルワード(ワード)
- WXOR_DW : 排他的 OR ダブルワード(ワード)

14.2 WAND_W : AND ワード(ワード)

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	WORD	I、Q、M、D、L または定数	論理演算の 1 番目の値
IN2	WORD	I、Q、M、D、L または定数	論理演算の 2 番目の値
OUT	WORD	I、Q、M、D、L	論理演算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

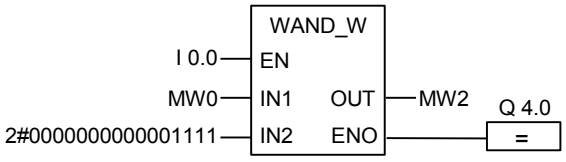
説明

(ワード)And ワード命令は、イネーブル入力(EN)の信号状態が 1 になると起動し、AND 真理値表に従って、入力 IN1 と IN2 の 2 つのデジタル値を 1 ビットずつ結合します。これらのワード値は、完全なビットパターンとして扱われます。この命令の結果は出力 OUT で確認できます。ENO は、EN と同じ信号状態になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	1	X	0	0	-	X	1	1	1

例



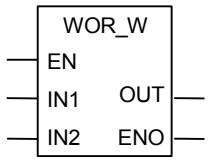
この命令は、0 の信号状態が 1 になると起動します。これに関連するビットは 0～3 だけで、MW0 の他のビットはすべてマスクされています。

IN1	=	0101010101010101
IN2	=	0000000000001111
OUT	=	0000000000000101

この命令が実行されると、Q4.0 は 1 になります。

14.3 WOR_W : OR ワード(ワード)

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	WORD	I、Q、M、D、L または定数	論理演算の 1 番目の値
IN2	WORD	I、Q、M、D、L または定数	論理演算の 2 番目の値
OUT	WORD	I、Q、M、D、L	論理演算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

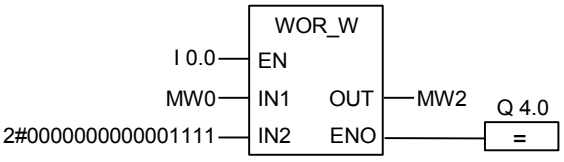
説明

(ワード)OR ワード 命令は、イネーブル入力(EN)の信号状態が 1 になると起動し、OR 真理値表に従って、入力 IN1 と IN2 の 2 つのデジタル値を 1 ビットずつ結合します。これらのワード値は、完全なビットパターンとして扱われます。この命令の結果は出力 OUT で確認できます。ENO は、EN と同じ信号状態になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	1	X	0	0	-	X	1	1	1

例



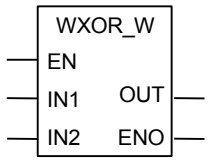
この命令は I0.0 が 1 のときに実行されます。MW0 と定数内のビットは OR 演算されてビット 0～3 が 1 に設定され、MW0 のそれ以外のビットはすべて無変化のまま MW2 に入力されます。

IN1	=	0101010101010101
IN2	=	0000000000001111
OUT	=	0101010101011111

この命令が実行されると、Q4.0 は 1 になります。

14.4 WXOR_W : 排他的 OR ワード(ワード)

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L	イネーブル入力
IN1	WORD	I、Q、M、D、L または定数	論理演算の 1 番目の値
IN2	WORD	I、Q、M、D、L または定数	論理演算の 2 番目の値
OUT	WORD	I、Q、M、D、L	論理演算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

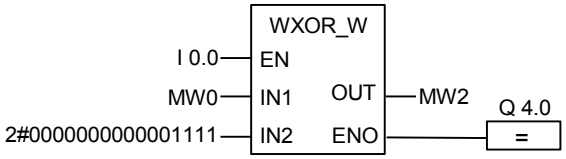
説明

(ワード)OR ワード 命令は、イネーブル入力(EN)の信号状態が 1 になると起動し、OR 真理値表に従って、入力 IN1 と IN2 の 2 つのデジタル値を 1 ビットずつ結合します。これらのワード値は、完全なビットパターンとして扱われます。この命令の結果は出力 OUT で確認できます。ENO は、EN と同じ信号状態になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	1	X	0	0	-	X	1	1	1

例



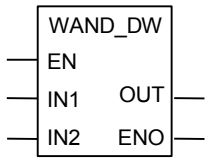
この命令は、入力 I0.0 が 1 のときに実行されます。

IN1	=	0101010101010101
IN2	=	0000000000001111
OUT	=	0101010101011010

この命令が実行されると、Q4.0 は 1 になります。

14.5 WAND_DW : AND ダブルワード(ワード)

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	DWORD	I、Q、M、D、L または定数	論理演算の 1 番目の値
IN2	DWORD	I、Q、M、D、L または定数	論理演算の 2 番目の値
OUT	DWORD	I、Q、M、D、L	論理演算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

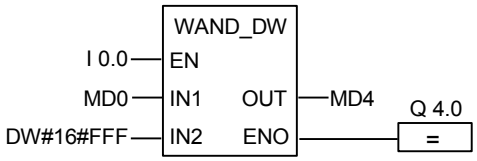
説明

(ワード)AND ダブルワード命令は、イネーブル入力(EN)の信号状態が 1 になると起動し、AND 真理値表に従って、入力 IN1 と IN2 の 2 つのデジタル値を 1 ビットずつ結合します。これらのワード値は、完全なビットパターンとして扱われます。この命令の結果は出力 OUT で確認できます。ENO は、EN と同じ信号状態になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	1	X	0	0	-	X	1	1	1

例



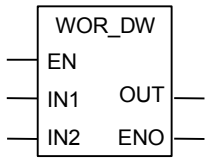
この命令は、0 のが 1 になると起動します。これに関連するビットは 0～11 だけで、MD4 の他のビットはすべてマスクされています。

IN1	=	0101010101010101	0101010101010101
IN2	=	0000000000000000	0000111111111111
OUT	=	0000000000000000	0000010101010101

この命令が実行されると、Q4.0 は 1 になります。

14.6 WOR_DW : OR ダブルワード(ワード)

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	DWORD	I、Q、M、D、L または定数	論理演算の 1 番目の値
IN2	DWORD	I、Q、M、D、L または定数	論理演算の 2 番目の値
OUT	DWORD	I、Q、M、D、L	論理演算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

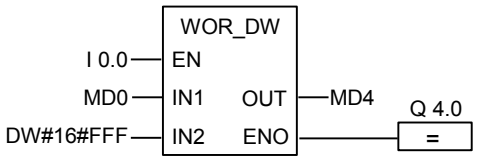
説明

(ワード)OR ダブルワード 命令は、イネーブル入力(EN)の信号状態が 1 になると起動し、OR 真理値表に従って、入力 IN1 と IN2 の 2 つのデジタル値を 1 ビットずつ結合します。これらのワード値は、完全なビットパターンとして扱われます。この命令の結果は出力 OUT で確認できます。ENO は、EN と同じ信号状態になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	1	X	0	0	-	X	1	1	1

例



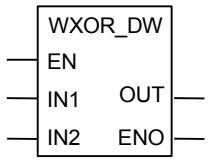
この命令は I0.0 が 1 のときに実行されます。MD0 と定数内のビットは OR 演算されてビット 0～11 が 1 に設定され、MD0 のそれ以外のビットはすべて無変化のまま MW4 に入力されます。

IN1	=	0101010101010101	0101010101010101
IN2	=	0000000000000000	0000111111111111
OUT	=	0101010101010101	0101111111111111

この命令が実行されると、Q4.0 は 1 になります。

14.7 WXOR_DW : 排他的 OR ダブルワード(ワード)

シンボル



パラメータ	データタイプ	メモリ領域	説明
EN	BOOL	I、Q、M、D、L、T、C	イネーブル入力
IN1	DWORD	I、Q、M、D、L または定数	論理演算の 1 番目の値
IN2	DWORD	I、Q、M、D、L または定数	論理演算の 2 番目の値
OUT	DWORD	I、Q、M、D、L	論理演算の結果
ENO	BOOL	I、Q、M、D、L	イネーブル出力

説明

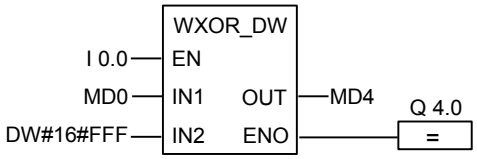
(ワード)拡張 OR ダブルワード 命令は、イネーブル入力(EN)の信号状態が 1 になると起動し、EXCLUSIVE OR 真理値表に従って、入力 IN1 と IN2 の 2 つのデジタル値を 1 ビットずつ結合します。これらのワード値は、完全なビットパターンとして扱われます。この命令の結果は出力 OUT で確認できます。

ENO は、EN と同じ信号状態になります。

ステータスワード

	BR	CC1	CC0	OV	OS	OR	STA	RLO	FC
書き込みの内容	1	X	0	0	-	X	1	1	1

例



この命令は、入力 I0.0 が 1 のときに実行されます。

IN1	=	0101010101010101	0101010101010101
IN2	=	0000000000000000	0000111111111111
OUT	=	0101010101010101	0101101010101010

この命令が実行されると、Q4.0 は 1 になります。

14.7 WXOR_DW : 排他的 OR ダブルワード(ワード)

A すべての FBD 命令の概要

A.1 ドイツ語のニーモニック(SIMATIC)に従ってソートされた FBD 命令

ドイツ語のプログラム表記法	英語のプログラム表記法	プログラムエレメントカタログ	説明
&	&	ビット論理命令	AND 論理演算
>=1	>=1	ビット論理命令	OR 論理演算
=	=	ビット論理命令	割り付け
#	#	ビット論理命令	中間出力
---|	---|	ビット論理命令	バイナリ入力の挿入
---o|	---o|	ビット論理命令	バイナリ入力の否定
==0	==0	ステータスビット	リザルトビット
>0	>0	ステータスビット	リザルトビット
>=0	>=0	ステータスビット	リザルトビット
<0	<0	ステータスビット	リザルトビット
<=0	<=0	ステータスビット	リザルトビット
<> 0	<> 0	ステータスビット	リザルトビット
ABS	ABS	浮動小数点 命令	浮動小数点数の絶対値を求める
ACOS	ACOS	浮動小数点 命令	角の三角関数を浮動小数点数で求める
ADD_DI	ADD_DI	整数の演算 命令	倍長整数の加算
ADD_I	ADD_I	整数の演算 命令	整数の加算
ADD_R	ADD_R	浮動小数点 命令	実数の加算
ASIN	ASIN	浮動小数点 命令	角の三角関数を浮動小数点数で求める
ATAN	ATAN	浮動小数点 命令	角の三角関数を浮動小数点数で求める
BCD_DI	BCD_DI	変換	BCD から倍長整数
BCD_I	BCD_I	変換	BCD から整数
BIE	BR	ステータスビット	BR メモリビット
CALL	CALL	プログラムコントロール	パラメータなしの FC/SFC の呼び出し
CALL_FB	CALL_FB	プログラムコントロール	CALL_FB(FB をボックスとして呼び出す)
CALL_FC	CALL_FC	プログラムコントロール	CALL_FC(FC をボックスとして呼び出す)
CALL_SFB	CALL_SFB	プログラムコントロール	CALL_SFB(SFB をボックスとして呼び出す)
CALL_SFC	CALL_SFC	プログラムコントロール	CALL_SFC(SFC をボックスとして呼び出す)
CEIL	CEIL	変換	切り上げ
CMP ? D	CMP ? D	比較	比較倍長整数

ドイツ語のプログラム表記法	英語のプログラム表記法	プログラムエレメントカタログ	説明
CMP ? I	CMP ? I	比較	整数の比較
CMP ? R	CMP ? R	比較	実数の比較
COS	COS	浮動小数点 命令	角の三角関数を浮動小数点数で求める
DI_BCD	DI_BCD	変換	倍長整数から BCD
DI_R	DI_R	変換	倍長整数から実数への変換
DIV_DI	DIV_DI	整数の演算 命令	倍長整数の除算
DIV_I	DIV_I	整数の演算 命令	整数の除算
DIV_R	DIV_R	浮動小数点 命令	実数の除算
EXP	EXP	浮動小数点 命令	浮動小数点数の指数値を求める
FLOOR	FLOOR	変換	切り下げ
I_BCD	I_BCD	変換	整数から BCD への変換
I_DI	I_DI	変換	整数から倍長整数への変換
INV_I	INV_I	変換	整数の 1 の補数
INV_DI	INV_DI	変換	倍長整数のビット反転
JMP	JMP	ジャンプ	ブロック内での条件なしジャンプ
JMP	JMP	ジャンプ	ブロック内での条件付きジャンプ
JMPN	JMPN	ジャンプ	ジャンプイフノット
LABEL	LABEL	ジャンプ	ジャンプラベル
LN	LN	浮動小数点 命令	浮動小数点数の自然対数を求める
MCR>	MCR>	プログラムコントロール	マスタコントロールリレーのオン/オフ
MCR<	MCR<	プログラムコントロール	マスタコントロールリレーのオン/オフ
MCRA	MCRA	プログラムコントロール	マスタコントロールリレーの有効化/無効化
MCRD	MCRD	プログラムコントロール	マスタコントロールリレーの有効化/無効化
MOD_DI	MOD_DI	整数の演算 命令	倍長整数の商余
MOVE	MOVE	移動	値の割り付け
MUL_DI	MUL_DI	整数の演算 命令	倍長整数の乗算
MUL_I	MUL_I	整数の演算 命令	整数の乗算
MUL_R	MUL_R	浮動小数点 命令	実数の乗算
N	N	ビット論理命令	立ち下がりパルス
NEG	NEG	ビット論理命令	アドレス立ち下がりパルス
NEG_DI	NEG_DI	変換	倍長整数の符号反転(NEG_DI)
NEG_I	NEG_I	変換	整数の符号反転
NEG_R	NEG_R	変換	実数の否定
OPN	OPN	DB 呼び出し	データブロックを開く
OS	OS	ステータスビット	OS メモリビット

ドイツ語のプログラム表記法	英語のプログラム表記法	プログラムエレメントカタログ	説明
OV	OV	ステータスビット	OV ビット
P	P	ビット論理命令	立ち上がり RLO 信号検出
POS	POS	ビット論理命令	アドレス立ち上がりパルス
R	R	ビット論理命令	出力のリセット
RET	RET	プログラムコントロール	リターン
ROL_DW	ROL_DW	シフト/ 循環	ダブルワード左回転
ROUND	ROUND	変換	倍長整数の丸め
ROR_DW	ROR_DW	シフト/ 循環	ダブルワード右回転
RS	RS	ビット論理命令	Reset_Set フリップフロップ
S	S	ビット論理命令	出力の設定
SA	SF	タイマ	オフディレイタイマの起動
SAVE	SAVE	ビット論理命令	RLO の BR メモリへの保存
S_AVERZ	S_OFFDT	タイマ	オフディレイタイマパラメータの割り付けと起動
SE	SD	タイマ	オフディレイタイマパラメータの割り付けと起動
S_EVERZ	S_ODT	タイマ	オフディレイタイマパラメータの割り付けと起動
SHL_DW	SHL_DW	シフト/ 循環	ダブルワード左シフト
SHL_W	SHL_W	シフト/ 循環	ワード左シフト
SHR_DI	SHR_DI	シフト/ 循環	倍長整数右シフト
SHR_DW	SHR_DW	シフト/ 循環	ダブルワード右シフト
SHR_I	SHR_I	シフト/ 循環	整数右シフト
SHR_W	SHR_W	シフト/ 循環	ワード右シフト
SI	SP	タイマ	パルスタイマの起動
S_IMPULS	S_PULSE	タイマ	パルスタイマパラメータの割り付けと起動
SIN	SIN	浮動小数点 命令	角の三角関数を浮動小数点数で求める
SQR	SQR	浮動小数点 命令	浮動小数点数の二乗(SQR)を求める
SQRT	SQRT	浮動小数点 命令	浮動小数点数の平方根(SQRT)を求める
SR	SR	ビット論理命令	フリップフロップの Set_Reset
SS	SS	タイマ	拡張オンディレイタイマの起動
S_SEVERZ	S_ODTS	タイマ	拡張オンディレイタイマパラメータの割り付けと起動
SUB_DI	SUB_DI	整数の演算 命令	倍長整数の減算
SUB_I	SUB_I	整数の演算 命令	整数の減算
SUB_R	SUB_R	浮動小数点 命令	実数の減算
SV	SE	タイマ	拡張パルスタイマの起動
S_VIMP	S_PEXT	タイマ	拡張パルスタイマパラメータの割り付けと起動
SZ	SC	カウンタ	カウンタ値の設定

ドイツ語のプログラム表記法	英語のプログラム表記法	プログラムエレメントカタログ	説明
TAN	TAN	浮動小数点 命令	角の三角関数を浮動小数点数で求める
TRUNC	TRUNC	変換	実数の倍長整数変換(小数部切り捨て)
UO	UO	ステータスビット	UO ビット
WAND_DW	WAND_DW	ワード論理命令	And ダブルワード
WAND_W	WAND_W	ワード論理命令	And ワード(ワード)
WOR_DW	WOR_DW	ワード論理命令	Or ダブルワード(ワード)
WOR_W	WOR_W	ワード論理命令	Or ワード(ワード)
WXOR_DW	WXOR_DW	ワード論理命令	排他的 Or ダブルワード(ワード)
WXOR_W	WXOR_W	ワード論理命令	排他的 Or ワード(ワード)
XOR	XOR	ビット論理命令	排他的 OR 論理演算
ZAEHLER	S_CUD	カウンタ	パラメータおよびカウントアップ/カウントダウンの割り付け
ZR	CD	カウンタ	カウントダウン
Z_RUECK	S_CD	カウンタ	パラメータおよびカウントダウンの割り付け
ZV	CU	カウンタ	カウントアップ
Z_VORW	S_CU	カウンタ	パラメータおよびカウントアップの割り付け

A.2 英語のニーモニック(インターナショナル)に従ってソートされた FBD 命令

英語のプログラム表記法	ドイツ語のプログラム表記法	プログラムエレメントカタログ	説明
&	&	ビット論理命令	AND 論理演算
>=1	>=1	ビット論理命令	OR 論理演算
=	=	ビット論理命令	割り付け
#	#	ビット論理命令	中間出力
---|	---|	ビット論理命令	バイナリ入力の挿入
---o|	---o|	ビット論理命令	バイナリ入力の否定
==0	==0	ステータスビット	リザルトビット
>0	>0	ステータスビット	リザルトビット
>=0	>=0	ステータスビット	リザルトビット
<0	<0	ステータスビット	リザルトビット
<=0	<=0	ステータスビット	リザルトビット
<> 0	<> 0	ステータスビット	リザルトビット
ABS	ABS	浮動小数点 命令	浮動小数点数の絶対値を求める
ACOS	ACOS	浮動小数点 命令	角の三角関数を浮動小数点数で求める
ADD_DI	ADD_DI	整数の演算 命令	倍長整数の加算
ADD_I	ADD_I	整数の演算 命令	倍長整数の加算
ADD_R	ADD_R	浮動小数点 命令	実数の加算
ASIN	ASIN	浮動小数点 命令	角の三角関数を浮動小数点数で求める
ATAN	ATAN	浮動小数点 命令	角の三角関数を浮動小数点数で求める
BCD_DI	BCD_DI	変換	BCD から倍長整数
BCD_I	BCD_I	変換	BCD から整数
BR	BIE	ステータスビット	BR メモリビット
CALL	CALL	プログラムコントロール	パラメータなしの FC/SFC の呼び出し
CALL_FB	CALL_FB	プログラムコントロール	CALL_FB(FB をボックスとして呼び出す)
CALL_FC	CALL_FC	プログラムコントロール	CALL_FC(FC をボックスとして呼び出す)
CALL_SFB	CALL_SFB	プログラムコントロール	CALL_SFB(SFB をボックスとして呼び出す)
CALL_SFC	CALL_SFC	プログラムコントロール	CALL_SFC(SFC をボックスとして呼び出す)
CD	ZR	カウンタ	カウントダウン
CEIL	CEIL	変換	切り上げ
CMP ? D	CMP ? D	比較	比較倍長整数

英語のプログラム表記法	ドイツ語のプログラム表記法	プログラムエレメントカタログ	説明
CMP ? I	CMP ? I	比較	整数の比較
CMP ? R	CMP ? R	比較	実数の比較
COS	COS	浮動小数点 命令	角の三角関数を浮動小数点数で求める
CU	ZV	カウンタ	カウントアップ
DI_BCD	DI_BCD	変換	倍長整数から BCD
DI_R	DI_R	変換	倍長整数から実数への変換
DIV_DI	DIV_DI	整数の演算 命令	倍長整数の除算
DIV_I	DIV_I	整数の演算 命令	整数の除算
DIV_R	DIV_R	浮動小数点 命令	実数の除算
EXP	EXP	浮動小数点 命令	浮動小数点数の指数値を求める
FLOOR	FLOOR	変換	切り下げ
I_BCD	I_BCD	変換	整数から BCD への変換
I_DI	I_DI	変換	整数から倍長整数への変換
INV_I	INV_I	変換	整数の 1 の補数
INV_DI	INV_DI	変換	倍長整数のビット反転
JMP	JMP	ジャンプ	ブロック内での条件なしジャンプ
JMP	JMP	ジャンプ	ブロック内での条件付きジャンプ
JMPN	JMPN	ジャンプ	ジャンプイフノット
LABEL	LABEL	ジャンプ	ジャンプラベル
LN	LN	浮動小数点 命令	浮動小数点数の自然対数を求める
MCR>	MCR>	プログラムコントロール	マスタコントロールリレーのオン/オフ
MCR<	MCR<	プログラムコントロール	マスタコントロールリレーのオン/オフ
MCRA	MCRA	プログラムコントロール	マスタコントロールリレーの有効化/無効化
MCRD	MCRD	プログラムコントロール	マスタコントロールリレーの有効化/無効化
MOD_DI	MOD_DI	整数の演算 命令	倍長整数の商余
MOVE	MOVE	移動	値の割り付け
MUL_DI	MUL_DI	整数の演算 命令	倍長整数の乗算
MUL_I	MUL_I	整数の演算 命令	整数の乗算
MUL_R	MUL_R	浮動小数点 命令	実数の乗算
N	N	ビット論理命令	立ち下がりパルス
NEG	NEG	ビット論理命令	アドレス立ち下がりパルス
NEG_DI	NEG_DI	変換	倍長整数の符号反転(NEG_DI)
NEG_I	NEG_I	変換	整数の符号反転
NEG_R	NEG_R	変換	実数の否定
OPN	OPN	DB 呼び出し	データブロックを開く

英語のプログラム表記法	ドイツ語のプログラム表記法	プログラムエレメントカタログ	説明
OS	OS	ステータスビット	OS メモリビット
OV	OV	ステータスビット	OV ビット
P	P	ビット論理命令	立ち上がり RLO 信号検出
POS	POS	ビット論理命令	アドレス立ち上がりパルス
R	R	ビット論理命令	出力のリセット
RET	RET	プログラムコントロール	リターン
ROL_DW	ROL_DW	シフト/ 循環	ダブルワード左回転
ROUND	ROUND	変換	倍長整数の丸め
ROR_DW	ROR_DW	シフト/ 循環	ダブルワード右回転
RS	RS	ビット論理命令	Reset_Set フリップフロップ
S	S	ビット論理命令	出力の設定
SAVE	SAVE	ビット論理命令	RLO の BR メモリへの保存
SC	SZ	カウンタ	カウンタ値の設定
S_CD	Z_RUECK	カウンタ	パラメータおよびカウントダウンの割り付け
S_CU	Z_VORW	カウンタ	パラメータおよびカウントアップの割り付け
S_CUD	ZAEHLER	カウンタ	パラメータおよびカウントアップ/カウントダウンの割り付け
SD	SE	タイマ	オンディレイタイマの起動
SE	SV	タイマ	拡張パルスタイマの起動
SF	SA	タイマ	オフディレイタイマの起動
SHL_DW	SHL_DW	シフト/ 循環	ダブルワード左シフト
SHL_W	SHL_W	シフト/ 循環	ワード左シフト
SHR_DI	SHR_DI	シフト/ 循環	倍長整数右シフト
SHR_DW	SHR_DW	シフト/ 循環	ダブルワード右シフト
SHR_I	SHR_I	シフト/ 循環	整数右シフト
SHR_W	SHR_W	シフト/ 循環	ワード右シフト
SIN	SIN	浮動小数点 命令	角の三角関数を浮動小数点数で求める
S_ODT	S_EVERZ	タイマ	オフディレイタイマパラメータの割り付けと起動
S_ODTS	S_SEVERZ	タイマ	拡張オンディレイタイマパラメータの割り付けと起動
S_OFFDT	S_AVERZ	タイマ	オフディレイタイマパラメータの割り付けと起動
SP	SI	タイマ	パルスタイマの起動
S_PEXT	S_VIMP	タイマ	拡張パルスタイマパラメータの割り付けと起動
S_PULSE	S_IMPULS	タイマ	パルスタイマパラメータの割り付けと起動
SQR	SQR	浮動小数点 命令	浮動小数点数の二乗(SQR)を求める

英語のプログラム表記法	ドイツ語のプログラム表記法	プログラムエレメントカタログ	説明
SQRT	SQRT	浮動小数点 命令	浮動小数点数の平方根(SQRT)を求める
SR	SR	ビット論理命令	フリップフロップの Set_Reset
SS	SS	タイマ	拡張オンディレイタイマの起動
SUB_DI	SUB_DI	整数の演算 命令	倍長整数の減算
SUB_I	SUB_I	整数の演算 命令	整数の減算
SUB_R	SUB_R	浮動小数点 命令	実数の減算
TAN	TAN	浮動小数点 命令	角の三角関数を浮動小数点数で求める
TRUNC	TRUNC	変換	実数の倍長整数変換(小数部切り捨て)
UO	UO	ステータスビット	UO ビット
WAND_DW	WAND_DW	ワード論理命令	And ダブルワード
WAND_W	WAND_W	ワード論理命令	And ワード(ワード)
WOR_DW	WOR_DW	ワード論理命令	Or ダブルワード(ワード)
WOR_W	WOR_W	ワード論理命令	Or ワード(ワード)
WXOR_DW	WXOR_DW	ワード論理命令	排他的 Or ダブルワード(ワード)
WXOR_W	WXOR_W	ワード論理命令	排他的 Or ワード(ワード)
XOR	XOR	ビット論理命令	排他的 OR 論理演算

B プログラミング例

B.1 プログラミングの概要例

実際の応用例

各 FBD 命令は特定の操作をトリガします。これらの命令を組み合わせて 1 つのプログラムにすると、各種のオートメーションタスクを実行できます。この章では、FBD 命令を使った以下のような実践的な応用例を説明します。

- ビット論理命令によるコンベアベルトのコントロール
- ビットロジック命令を使用した搬送機ベルトの走行方向の検知
- タイマ命令を使用したクロックパルスの生成
- カウンタ命令と比較命令を使用した格納庫の管理
- 整数演算命令による問題の解決
- オーブンの加熱時間の設定

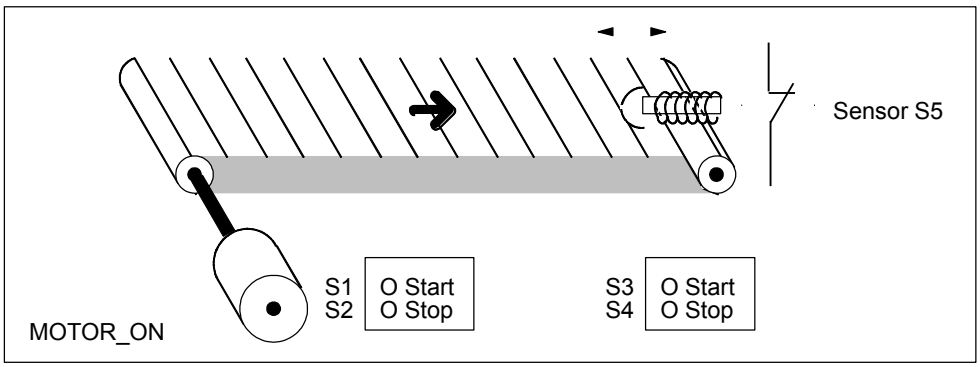
使用する命令

ニーモニック	プログラムエレメントカタログ	説明
WAND_W	ワード論理命令	ワードの論理積
WOR_W	ワード論理命令	ワードの論理和
S_CD	カウンタ	カウントダウン
S_CU	カウンタ	カウントアップ
R	ビット論理命令	出力のリセット
S	ビット論理命令	出力の設定
P	ビット論理命令	立ち上がり RLO 信号検出
ADD_I	浮動小数点命令	整数の加算
DIV_I	浮動小数点命令	整数の除算
MUL_I	浮動小数点命令	整数の乗算
CMP >=I、CMP <=I	比較	整数の比較
&	ビット論理命令	AND
>=1	ビット論理命令	OR
=	ビット論理命令	割り付け
JMPN	ジャンプ	ジャンプイフノット
RET	プログラムコントロール	リターン
MOVE	移動	値の割り付け
SE	タイマ	拡張パルスタイマ

B.2 例: ビットロジック命令

例 1: 搬送機ベルトの制御

以下の図に、電動式搬送機ベルトを示します。ベルトの始点には、START 用 S1 ボタンと STOP 用 S2 ボタンの 2 つの押しボタンがあります。ベルトの終点には、START 用 S3 スイッチと STOP 用 S4 スイッチの 2 つの押しボタンスイッチもあります。どちらの端からでもベルトを起動または停止できます。また、ベルト上の物体が終点に達すると、センサ S5 がこれを感知し、ベルトを停止させます。



絶対プログラミングとシンボルプログラミング

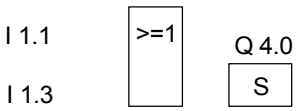
コンベアベルトをコントロールするプログラムを記述するには、コンベアシステムの各種コンポーネントを表す**絶対値**または**シンボル**を使用します。

シンボルテーブルを 1 つ作成して、選択したシンボルと絶対値を対応させる必要があります。STEP 7 オンラインヘルプを参照してください。

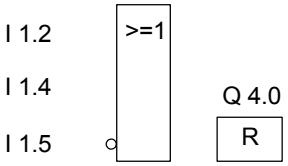
システムコンポーネント	絶対アドレス	シンボル	シンボルテーブル
START ボタン	I 1.1	S1	I 1.1 S1
STOP ボタン	I 1.2	S2	I 1.2 S2
START ボタン	I 1.3	S3	I 1.3 S3
STOP ボタン	I 1.4	S4	I 1.4 S4
センサ	I 1.5	S5	I 1.5 S5
Motor	Q 4.0	MOTOR_ON	Q 4.0 MOTOR_ON

コンベアベルトをコントロールするためのファンクションブロックダイアグラム

ネットワーク 1: いずれかの開始スイッチを押すと、モータがオンになります。

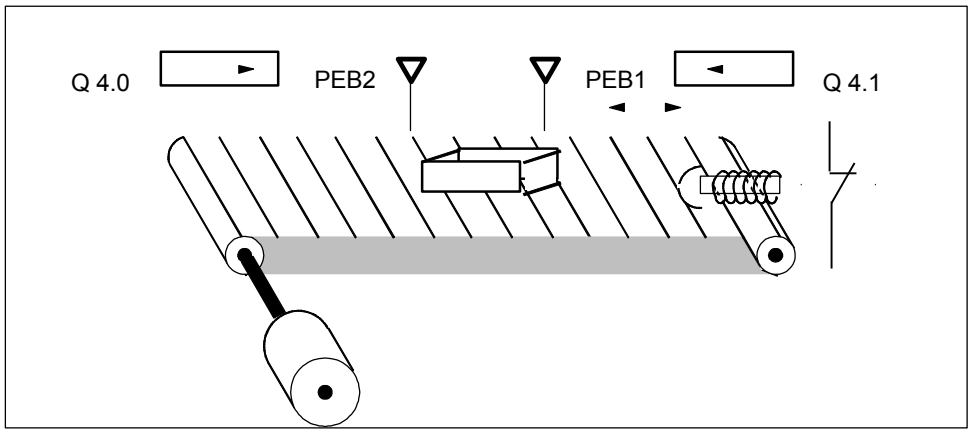


ネットワーク 2: いずれかの停止スイッチを押すか、ベルトの終端にあるセンサがモータをオフにする



例 2: コンベアベルトの方向の検出

以下の図に、光電スイッチを 2 個(PEB1 と PEB2)を装備した搬送機ベルトを示します。この 2 個の光電スイッチは、ベルト上のパッケージの移動方向を検知できます。各光電スイッチは、a 接点と同じように機能します。



絶対プログラミングとシンボルプログラミング

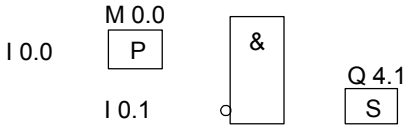
コンベアベルトシステムの方向表示を有効にするプログラムを記述するには、コンベアシステムの各種コンポーネントを表す**絶対値**または**シンボル**を使用します。

シンボルテーブルを 1 つ作成して、選択したシンボルと絶対値を対応させる必要があります。STEP 7 オンラインヘルプを参照してください。

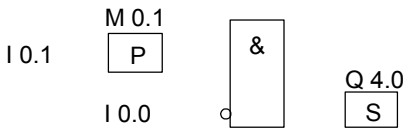
システムコンポーネント	絶対アドレス	シンボル	シンボルテーブル
光電スイッチ 1	I 0.0	PEB1	I 0.0 PEB1
光電スイッチ 2	I 0.1	PEB2	I 0.1 PEB2
右移動の表示	Q 4.0	RIGHT	Q 4.0 RIGHT
左移動の表示	Q 4.1	LEFT	Q 4.1 LEFT
パルスメモリビット 1	M 0.0	PMB1	M 0.0 PMB1
パルスメモリビット 2	M 0.1	PMB2	M 0.1 PMB2

コンベアベルトの方向を検出するためのファンクションブロックダイアグラム

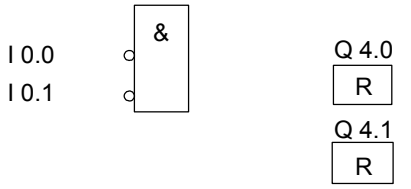
ネットワーク 1: 入力 I 0.0 の信号状態が 0 から 1(信号立ち上がり)に移行すると同時に、入力 I 0.1 の信号状態が 0 になっている場合は、ベルト上の荷物が左に移動しています。



ネットワーク 2: 入力 I 0.1 の信号状態が 0 から 1(信号立ち上がり)に移行すると同時に、入力 I 0.0 の信号状態が 0 になっている場合は、ベルト上の荷物が右に移動しています。



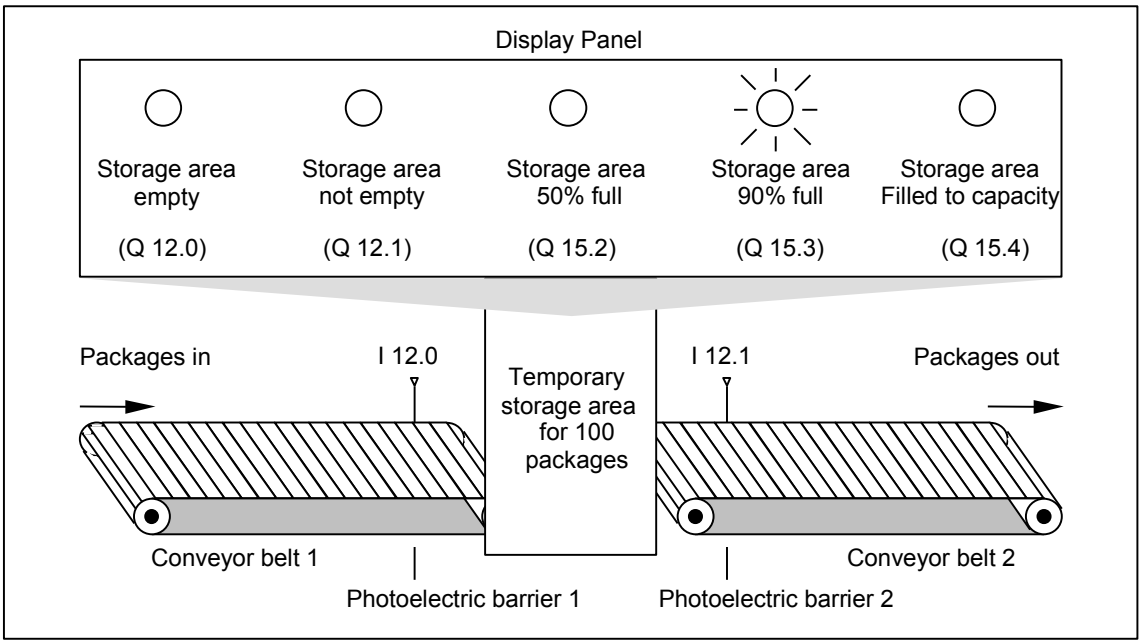
ネットワーク 3: 一方の光電バリアが遮断された場合は、バリア間に荷物があることを意味します。この場合、走行方向は表示されません。



B.3 例: カウンタ命令と比較命令

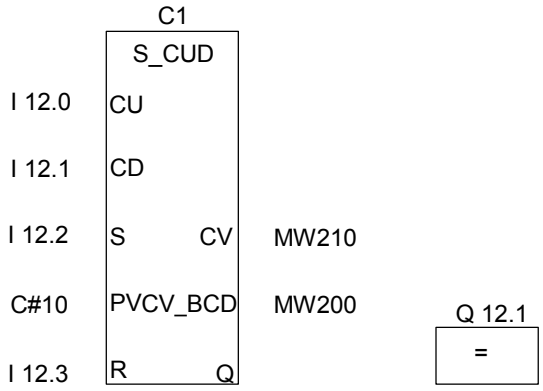
カウンタ命令と比較命令を使用した格納庫の管理

以下の図に、搬送機ベルトを 2 つ装備し、この 2 つのベルトの間にテンポラリ格納庫を装備したシステムを示します。搬送機ベルト 1 は、パッケージを格納庫まで配送します。搬送機ベルト 1 の末端、格納庫側にある光電スイッチにより、格納庫に搬入された荷物の数が確定されます。搬送機ベルト 2 は、パッケージをこのテンポラリ格納庫から積載ドックへ搬送します。パッケージは、この積載ドックからトラックで発送され、顧客に配送されます。搬送機ベルト 2 の末端、格納庫側にある光電スイッチにより、格納庫から積載ドックへ搬出された荷物の数が確定されます。テンポラリ格納庫の格納率は、表示パネルの 5 つのランプで示されます。

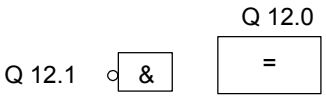


表示パネルの表示ランプを作動させるためのファンクションブロックダイアグラム

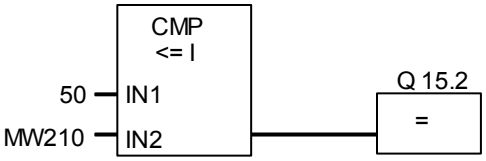
ネットワーク 1: カウンタ C1 は、入力 CU の信号状態が“0”から“1”に変わるたびにカウントを増やし、入力 CD の信号状態が“0”から“1”に変わるたびにカウントを減らします。入力 S の信号状態が“0”から”1”に変わると、カウンタ値が値 PV に設定されます。入力 R の信号状態が"0"から"1"に変わると、カウンタ値が"0"にリセットされます。MW200 には、カウンタの C1 の現在値が出力されます。Q12.1 は、“格納エリアが空でない”ことを示します。



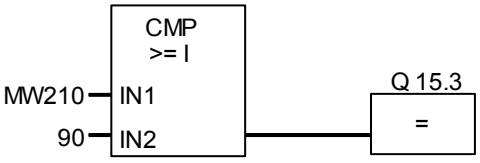
ネットワーク 2: Q12.0 は、"格納エリアが空"であることを示します。



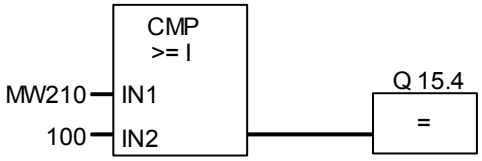
ネットワーク 3: 50 がカウンタ値よりも少ないかまたは等しい(つまり、現在のカウンタ値が 50 以上である)場合は、“格納エリア 50%充填”を示す表示ランプが点灯します。



ネットワーク 4: カウンタ値が 90 以上の場合は、“格納エリア 90%充填”を示す表示ランプが点灯します。



ネットワーク 5: カウンタ値が 100 以上の場合は、“格納エリア満杯” を示す表示ランプが点灯します。



B.4 例: タイマ命令

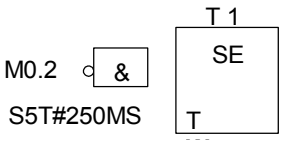
クロックパルスジェネレータ

周期反復信号の生成が必要な場合、クロックパルスジェネレータまたはフラッシャ信号を使用できます。これは、表示ランプの点滅を制御する信号システムによく使用されています。

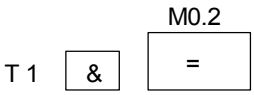
S7-300 を使用する場合､特殊なオーガニゼーションブロックでタイムドリブン処理を使用すれば、クロックパルスジェネレータファンクションを実行できます。ただし、次の FBD プログラムに示す例では、タイマファンクションを使用してクロックパルスを生成する方法を説明します。このサンプルプログラムで、タイマによるフリーホイーリングクロックパルスジェネレータの実現方法について説明します。

クロックパルスを生成するためのファンクションブロックダイアグラム(パルス効率係数 1:1)

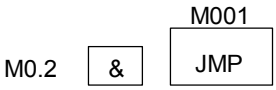
ネットワーク 1: タイマ T1 の信号状態が 0 の場合は、時間値 250 ms を T1 にロードし、T1 を拡張パルスタイマとして起動します。



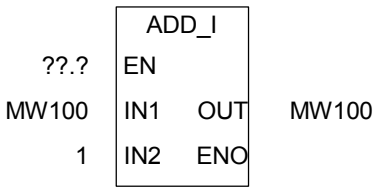
ネットワーク 2: タイマの状態は、補助メモリマーカーに一時的に保存されます。



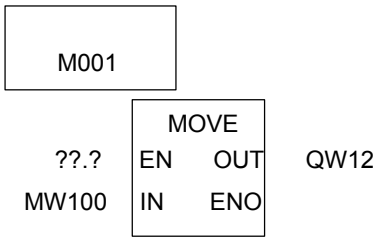
ネットワーク 3: タイマ T1 の信号状態が 1 の場合は、ジャンプラベル M001 にジャンプします。



ネットワーク 4: タイマ T1 が切れると、メモリワード 100 が 1 増えます。

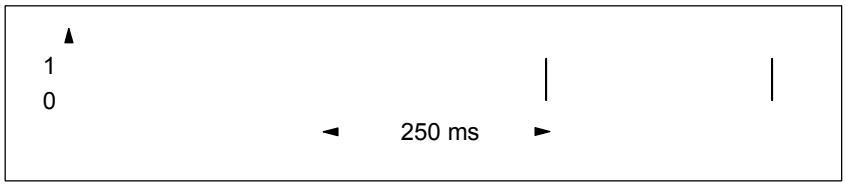


ネットワーク 5: **MOVE** 命令を使用すると、出力 Q12.0～Q13.7 で異なるクロック周波数を出力できます。



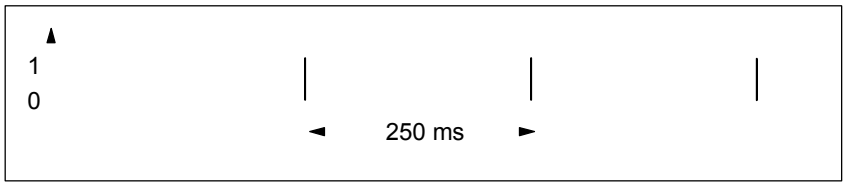
信号チェック

タイマ T1 の信号チェックは、クロックパルスの例において、AND 論理命令(M0.2)の否定入力パラメータに対して次の論理演算結果(RLO)を生成します。



タイマは満了すると直ちに再起動します。このため、信号は 1 の信号状態を短時間だけ生成します。

否定(反転)RLO:



250 ms ごとに、RLO ビットが 0 になります。ジャンプは無視され、メモリワード MW100 の値が 1 増えます。

特定周波数への到達

メモリバイト MB101 および MB100 の各ビットから、次の周波数を実現することができます。

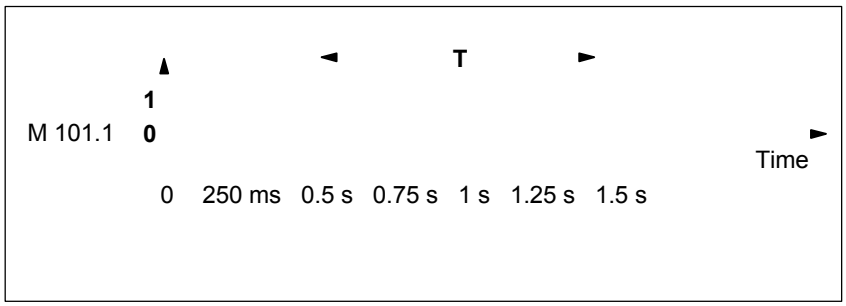
MB101/MB100 のビット	周波数(Hz)	パルス幅
M 101.0	2.0	0.5 s (250 ms オン/ 250 ms オフ)
M 101.1	1.0	1 s (0.5 s オン/ 0.5 s オフ)
M 101.2	0.5	2 s (1 s オン/ 1 s オフ)
M 101.3	0.25	4 s (2 s オン/ 2 s オフ)
M 101.4	0.125	8 s (4 s オン/ 4 s オフ)
M 101.5	0.0625	16 s (8 s オン/ 8 s オフ)
M 101.6	0.03125	32 s (16 s オン/ 16 s オフ)
M 101.7	0.015625	64 s (32 s オン/ 32 s オフ)
M 100.0	0.0078125	128 s (64 s オン/64 s オフ)
M 100.1	0.0039062	256 s (128 s オン/128 s オフ)
M 100.2	0.0019531	512 s (256 s オン/256 s オフ)
M 100.3	0.0009765	1024 s (512 s オン/512 s オフ)
M 100.4	0.0004882	2048 s (1024 s オン/1024 s オフ)
M 100.5	0.0002441	4096 s (2048 s オン/2048 s オフ)
M 100.6	0.000122	8192 s (4096 s オン/4096 s オフ)
M 100.7	0.000061	16384 s (8192 s オン/8192 s オフ)

メモリ MB101 の各ビットの信号状態

スキャンサイクル	ビット 7	ビット 6	ビット 5	ビット 4	ビット 3	ビット 2	ビット 1	ビット 0	時間値 (単位: ms)
0	0	0	0	0	0	0	**0**	0	250
1	0	0	0	0	0	0	**0**	1	250
2	0	0	0	0	0	0	1	0	250
3	0	0	0	0	0	0	1	1	250
4	0	0	0	0	0	1	**0**	0	250
5	0	0	0	0	0	1	**0**	1	250
6	0	0	0	0	0	1	1	0	250
7	0	0	0	0	0	1	1	1	250
8	0	0	0	0	1	0	**0**	0	250
9	0	0	0	0	1	0	**0**	1	250
10	0	0	0	0	1	0	1	0	250
11	0	0	0	0	1	0	1	1	250
12	0	0	0	0	1	1	**0**	0	250

MB 101 (M 101.1)のビット 1 の信号状態

周波数= 1/T = 1/1 s = 1 Hz



B.5 例: 整数値演算命令

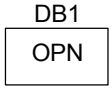
演算問題の解決

このサンプルプログラムでは、3 つの整数値演算命令を使用して、以下の方程式と同じ結果を求める方法を示します。

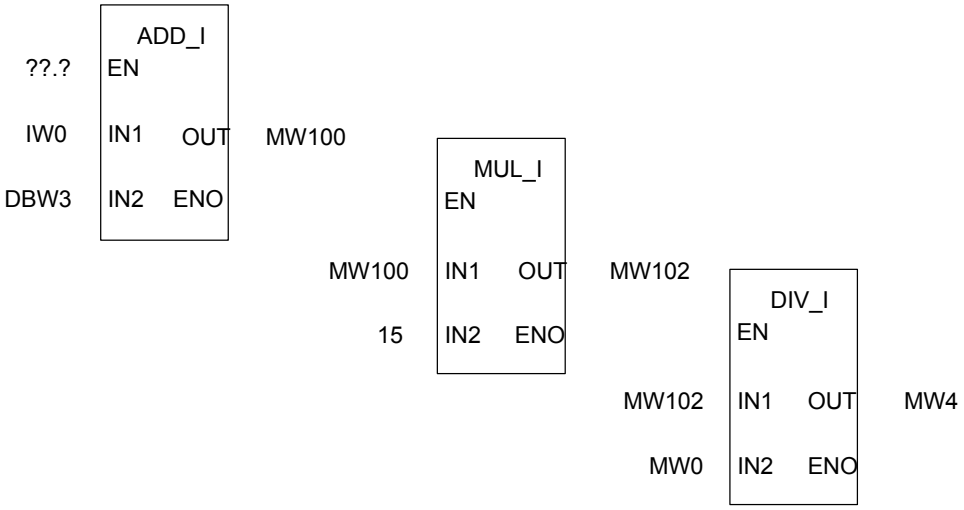
MW4 = ((IW0 + DBW3) x 15) / MW0

ファンクションブロックダイアグラム

ネットワーク 1: データブロック DB1 を開きます。



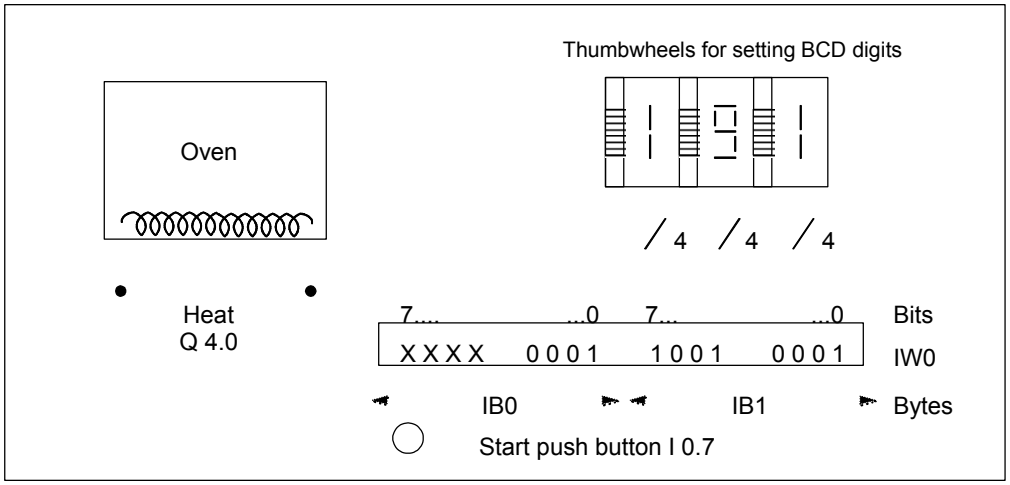
ネットワーク 2: 入力ワード IW0 が共有データワード DBW3 に加算され(データブロックが定義済みで開いていること)、合計がメモリワード MW100 にロードされます。次に、MW100 に 15 が掛けられて、その積がメモリダブルワード MW102 にセーブされます。MW102 は MW0 で割られて、その商は MW4 にセーブされます。



B.6 例: ワードロジック命令

オーブンの加熱

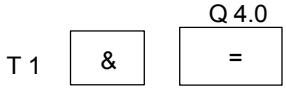
オペレータが START 押しボタンを押すと、オーブンは加熱を開始します。オペレータは、図に示されているサムホールスイッチを使用すれば、加熱時間を設定できます。設定した値は、2 進化 10 進数(BCD)フォーマットの秒数で示されます。



システムコンポーネント	絶対アドレス
START 押しボタン	I 0.7
1 の位のサムロータリースイッチ	I 1.0 ～ I 1.3
10 の位のサムホイール	I 1.4 ～ I 1.7
100 の位のサムロータリースイッチ	I 0.0 ～ I 0.3
加熱開始	Q 4.0

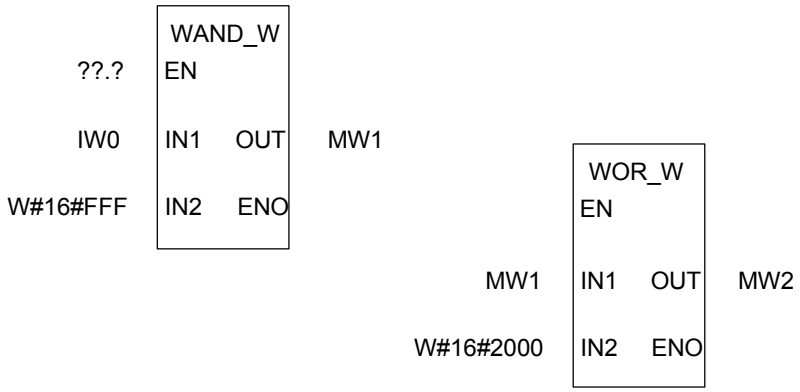
ファンクションブロックダイアグラム

ネットワーク 1: タイマが作動している場合は、ヒーターがオンになります。



ネットワーク 2: タイマが作動している場合は、**Return** 命令によりここで処理が終了します。

ネットワーク 3: 入力ビット I 0.4～I 0.7 をマスキングします(つまり、これらを 0 にリセットします)。サムホイール入力のこれらのビットは使用されません。サムホイール入力の 16 ビットは、**WAND_W** 命令に従って W#16#0FFF と組み合わされます。この結果は、メモリワード MW1 に記録されます。秒単位のタイムベースと設定するために、設定値は **WOR_W** 命令に従って W#16#2000 と組み合わされ、ビット 13 が 1 に、ビット 12 が 0 に設定されます。



ネットワーク 4: 開始押しボタンが押された場合は、タイマ T1 を拡張パルスタイマとして起動し、事前設定値として(上記の論理命令から導出された)メモリワード MW2 をロードします。

C ファンクションブロックダイアグラムでの作業

C.1 ブロックのタイプ

ブロックタイプの種類

プログラムは様々なブロックから構成され、各種のブロックを使用することで、プログラムを構築できます。ブロックはプログラムの独立した部分で、それぞれが特定の機能をもっています。STEP 7では、ブロックは大きくロジックブロックとデータブロックの2種類に分類されます。ロジックブロック(OB、FB、SFB、FC、および SFC)には、信号を処理するための命令が格納されます。これに対し、データブロック(DB、DI)は、データを格納するために使用されます。

ロジックブロック

論理ブロックは、プログラムやサブルーチンのソフトウェアモジュールとして使用されます。基本は OB1(オーガニゼーションブロック 1)です。あるブロックを別のブロック内で呼び出す場合、相互にパラメータを交換するように、ブロックの変数宣言テーブルをプログラムできます。たとえば、呼び出されたブロックは、呼び出し側のブロックから様々な入力/出力パラメータを受け取るようにプログラムできます。これによって、呼び出されたブロックで、入力/出力パラメータの入力値が処理され、結果が生成されます。その後、プログラムの結果および制御が、呼び出し側のブロックに返されます。

命令

CALL FC1 命令(ファンクションの呼び出し用)と CALL FB1, DB2 命令(ファンクションブロックの呼び出し用)の違いは、FB 呼び出しにはデータブロック(インスタンス DB)が割り付けられている点にあります。この DB には、ファンクションブロックの変数宣言テーブルに従い、ローカルブロック変数が格納されます。これに対し、ファンクションの場合は、ファンクションの実行中および削除中に別のブロックが呼び出されると、呼び出しパラメータと割り付け情報がローカルメモリ内に格納されます。

データブロック

データブロックには次の 2 種類があります。

- 作成した構造内のユーザーデータが格納されるデータブロック。このデータブロックは、読み書き操作のために、ロジックブロックで開くことができます。
- ファンクションブロックのインスタンス(インスタンス DB)の実行に使用される呼び出しパラメータとスタティックローカルデータが格納されるデータブロック。

データブロックおよびロジックブロックは、ラダーエディタ、FBD エディタ、または STL エディタで作成します。Siemens の STEP 7 パッケージでは、様々な標準ブロックが提供されています。

C.2 EN/ENO 機構

FBD/LAD ボックスのイネーブル(EN)およびイネーブル出力(ENO)は、BR ビットによって取得されます。

EN と ENO が接続される場合、次のことが適用されます。

ENO = EN AND NOT (ボックスエラー)

エラーは発生しない(ボックスエラー = 0)の場合、ENO = EN。

EN/ENO メカニズムは以下のものに対して使用されます。

- 数値演算命令
- 転送命令および変換命令
- シフト命令および循環命令
- ブロックの呼び出し

このメカニズムは、以下のものに使用**されません**。

- 比較
- カウンタ
- タイマ

ボックスにおける実際の命令では、EN/ENO メカニズムに対して追加の STL 命令が生成されます。これは、既存の前の論理操作およびその後の論理操作に依存します。下記に、考えられる 4 つの加算機の使用例を示しています。

1. EN と ENO が接続されている加算器
2. EN を接続し ENO を接続しない加算機
3. EN を接続しないで ENO を接続した加算機
4. EN も ENO も接続されていない加算器

ブロックの作成に関する注記

FBD または LAD に呼び出すブロックをプログラムする場合、そのブロックを終了する時点で BR ビットを必ず設定する必要があります。4 番目の例は、自動的に実行される事例ではないことを示しています。BR は常に EN/ENO メカニズムによって上書きされるので、BR をメモリビットとして使用できません。代わりに、テンポラリ変数を使用します。エラーが発生すると必ず、このエラーをテンポラリ変数に保存します。この変数を 0 に初期化します。命令が失敗するとブロック全体のエラーとなるポイントがある場合、ブロック内のこうしたポイントごとに、EN/ENO によってこの変数を設定します。この目的には、NOT および SET コイルで十分です。このブロックプログラムが終わると、次のネットワークが処理されます。

```
end:   AN error
       SAVE
```

どの場合も必ずこのネットワークが処理されます。この結果、ブロック内では BEC を使用できないため、このネットワークをスキップします。

C.2.1 EN と ENO が接続されている加算器

加算機に EN と ENO が接続されている場合、次の STL 命令がトリガされます。

```
1     A    I    0.0          // EN 接続
2     JNB _001               // RLO を BR にシフトし、RLO = 0 でジャンプ
3     L      in1             // ボックスパラメータ
4     L      in2             // ボックスパラメータ
5     +I                     // 実際の追加分
6     T      out             // ボックスパラメータ
7     AN     OV              // エラーの認識
8     SAVE                   // エラーを BR に保存する
9     CLR                    // ファーストチェック
10 _001:   A      BR         // BR を RLO もシフト
11    =    Q    4.0
```

行 1 に続く RLO には、前の論理演算結果が含まれます。JNB 命令は、RLO を BR ビットにコピーして、最初のチェクビットを設定します。

- RLO = 0 の場合、プログラムは行 10 にジャンプし、A BR を再開します。加算は実行されません。行 10 で、BR が RLO に再度コピーされ、0 が出力に割り付けられます。
- RLO が 1 のとき、プログラムはジャンプしません。すなわち、加算が実行されます。行 7 で、プログラムは、加算中にエラーが発生したかどうか評価します。その後、行 8 で、この結果を BR に保存します。行 9 に、最初のチェックビットが設定されます。行 10 で、BR ビットが RLO にコピーされ、出力に加算演算が正常に終了したかどうか示されます。
  行 10 と 11 では BR ビットは変更されません。このため、加算が成功したかどうかも示されます。

C.2.2 EN を接続し ENO を接続しない加算機

加算機に EN は接続されているが ENO は接続されていない場合、次の STL 命令がトリガされます。

```
1     A     I      0.0   // EN 接続
2     JNB _001           // RLO を BR にシフトし、RLO = 0 でジャンプ
3     L     in1          // ボックスパラメータ
4     L     in2          // ボックスパラメータ
5     +I                   // 実際の追加分
6     T     out          // ボックスパラメータ
7  _001:     NOP   0
```

行 1 に続く RLO には、前の論理演算結果が含まれます。JNB 命令は、RLO を BR ビットにコピーして、最初のチェクビットを設定します。

- RLO が 0 のとき、プログラムは行 7 にジャンプするため、加算は実行されません。RLO と BR は 0 です。
- RLO が 1 であった場合、プログラムはジャンプしないので、加算が実行されることを意味します。このプログラムは加算演算中にエラーが発生したかどうかを評価しません。RLO および BR は 1 になります。

C.2.3 EN を接続しないで ENO を接続した加算機

加算機に EN を接続しないが ENO を接続した場合、以下の STL 命令がトリガされます。

```
1     L     in1              // ボックスパラメータ
2           L     in2        // ボックスパラメータ
3     +I                     // 実際の追加分
4           T     out        // ボックスパラメータ
5     AN    OV               // エラーの認識
6           SAVE             // エラーを BR に保存する
7     CLR                    // ファーストチェック
8           A     BR         // BR を RLO もシフト
9     = Q 4.0
```

すべての場合、加算演算が実行されます。行 5 で、プログラムは加算演算中にエラーが発生したかどうかを評価し、行 6 でその結果を BR に保存します。行 7 は最初のチェックビットを設定します。行 8 で、BR ビットが RLO にコピーされ、出力に加算演算が正常に終了したかどうか示されます。

行 8 と 9 では BR ビットは変更されません。このため、加算が成功したかどうかも示されます。

C.2.4 EN も ENO も接続されていない加算器

加算機に EN も ENO も接続されていない場合、次の STL 命令がトリガされます。

```
1     L     in1          // ボックスパラメータ
2     L     in2          // ボックスパラメータ
3     +I                 // 実際の追加分
4     T     out          // ボックスパラメータ
5     NOP 0
```

加算が実行されます。RLO と BR ビットは変更されません。

C.3 パラメータ転送

ブロックのパラメータは、値として転送されます。ファンクションブロックでは、インスタンスデータブロックの実パラメータは、呼び出されたブロックの中で使用されます。ファンクションでは、現在値のコピーがローカルデータスタックに入れられます。ポインタはコピーされません。呼び出し前に、入力値はインスタンス DB または L スタックにコピーされます。呼び出し後に、出力値が変数にコピーされます。呼び出されたブロック内で操作対象となれるコピーは 1 つだけです。この操作に必要な STL 命令は、呼び出し側ブロック内にあるため、ユーザーから隠されていることに変わりはありません。

注記

メモリビット、入力、出力、またはペリフェラル I/O がファンクションの実アドレスとして使用されている場合は、他のアドレスとは異なる方法で扱われます。ここでは、更新は直接ではなく、L-Stack により実行されます。

例外:

対応する仮パラメータのデータタイプ BOOL の入力パラメータである場合、この現在のパラメータは L スタックを通して更新されます。

注意

呼び出されたブロックのプログラミング時には、出力として宣言されたパラメータも作成されていることを確認してください。作成されていない場合、出力の値は任意になります! ファンクションブロックでは、最後の呼び出しが指定したインスタンス DB からの値となり、ファンクションでは、L スタックに格納されている値になります。

以下の点を注意してください。

- 可能なら OUTPUT パラメータをすべて初期化します。
- Set および Reset の各命令を使用しないようにします。こうした命令は、RLO によって違ってきます。RLO の値が 0 の場合、任意の値が保たれます。
- ブロック内でジャンプする場合、出力パラメータが作成されたどの位置もスキップしないでください。BEC および MCR 命令の結果を忘れないでください。

索引

き

し

す

せ

た

ち

て

と

は

ひ

ふ

へ

ま

ら

り

れ

わ