

# LD Cv!/LD Cv2! 仮想PLC仕様書(第3版)

(C) 2020 タカミコムボード

## ■はじめに

本文書は、ラダー図開発ツールLD Cv2! (<https://www.takami.com/board/>)のターゲットマシンである仮想PLCの仕様書です。

LD Cv2!による開発では、本仕様書で規定した仮想的なPLC(Programmable Logic Controller)をターゲットマシンとして、ラダー図で制御プログラムを記述します。

この開発手法では、外部I/Oと仮想PLCのMおよびWの番号との対応付け、タイマの基準クロックを何にするか、の2点のみ決定すれば良く、その他の条件(マイコンが何であるか、OSが何であるかなど)を意識せずプログラミング作業が進められます。

仮想PLCは規模(機能、RAM容量)により次の3つのモデルがあり「Smallモデル < 標準モデル < Largeモデル」の関係にあります。

表0 仮想PLCのモデル

モデル名	LD Cv2!での指定方法	I/O メモリ	ワード メモリ	微分出力命令 ワード演算命令	特徴
Small	変換言語に「C言語(Smallモデル)」を選択	128点	なし	なし	RAM容量の少ないマイコン向け
標準	変換言語に「C言語」を選択	2048点			旧ソフトLD Cv!の仮想plcと互換
Large	変換言語に「C言語(Largeモデル)」を選択			2048ワード	あり

以下、仮想PLCの仕様について解説します。

## ■仮想PLC仕様

仮想PLCの仕様を表1に示します。

表1 仮想PLC仕様

項目	仕様
データメモリ容量	I/Oメモリ(128点または2048点) (*1)、特殊メモリ(128点)、タイマ(128点)、カウンタ(128点) (Largeモデルの場合のみ) ワードメモリ(2048ワード)
ラダーシンボル	A接点、B接点、コイル、カウンタクリア、セット、リセット、マスタコントロールセット、マスタコントロールリセット、エンド
命令語	LD、LDNOT、AND、ANDNOT、OR、ORNOT、ANDSTACK、ORSTACK、OUT、OUTT、OUTC、CLRC、SET、RES 、MCS、MCR、END
タイマ	オンディレイアップタイマ、点数128点、設定値0～65535 S0(タイマ基準クロック)の変化(↑および↓)で経過値が+1増加
カウンタ	アップカウンタ、点数128点、設定値0～65535 カウンタコイルへの入力変化(↑)で経過値が+1増加、リセット方法は同じカウンタ番号の[CLR]コイルをON
特殊メモリ	タイマ基準クロック、1、10、100、1000、10000スキャン反転、運転1スキャンON、常時ON (Largeモデルの場合のみ) 微分出力命令(S20)、ワード演算命令(S30～S3F)、コンディションフラグ(S2D～S2F) (*2)

(\*1)Smallモデルの場合のみ128点

(\*2)特殊メモリSの一部を、微分出力命令、ワード演算命令、コンディションフラグに割り当てています。

## ■仮想PLCのI/O種類

仮想PLCで利用できるI/O(実体はメモリ)は表2の通りです。

各I/Oは機能によって5つのI/O種別に分類され、先頭にI/O種別をあらわす記号を付けた16進数で表現します。

表2 I/O種類

I/O種類	点数	I/O番号	説明
I/Oメモリ(M)	128点 (Smallモデル)	M0～M7F	1ビットのリード・ライト可能なメモリ
	2048点 (標準、Largeモデル)	M0～M7FF	
特殊メモリ(S)	128点	S0～S7F	それぞれ特殊な動作を行うメモリ
タイマ(T)	128点	T0～T7F	指定時間以上のON入力の継続でONとなるコイル
カウンタ(C)	128点	C0～C7F	指定回数以上のパルス入力でONとなるコイル
ワードメモリ(W)	なし (Small、標準モデル)	-	少なくとも16ビットの記憶サイズ(*1)を持つリード・ライト可能なメモリ(*1)
	2048ワード (Largeモデル)	W0～W7FF	

(\*1)記憶サイズは適用するCコンパイラの「unsigned int型」の記憶サイズによって決定されます。

### (1)I/Oメモリ(M)

I/Oメモリは、ラダー図で接点およびコイルとして記述できる1ビットのリード・ライト可能なメモリ(RAM)です。

LD Cx2!にとってI/Oメモリは、ラダー図のプログラムにしたがってリード・ライトされるメモリにすぎません。しかし、I/Oメモリ(M)と開発対象であるマイコンのI/Oポートとの間でデータの転送(I/Oリフレッシュ)を定期的におこなうことで、I/Oメモリは開発対象のI/Oポートと連動します。

具体的には、毎スキャン実行の直前に「入力ポート→M転送」をおこなったMは、ラダー図上では入力ポートのON/OFF状態が反映された接点Mとして使えます。また、毎スキャン実行の直後に「M→出力ポート転送」をおこなったMは、ラダー図上では出力ポートのON/OFF状態を制御するコイルMとして使えます。 どのI/Oメモリ(M)をどのI/Oポートに割り当てるかのルールは無く、自由に決めてかまいません。

### (2)特殊メモリ(S)

特殊メモリは、あらかじめ番号によって機能が決められている1ビットのメモリです。

特殊メモリは表3の通りです。S0はコイルとしてラダー図中に1回のみ記述可能です。S10～S14、S1E、S1Fは接点としてラダー図中に任意の数だけ記述可能です。S10～S14とS1Eの変化はスキャンとスキャンの区切りでおこなわれるため、同一スキャン中は値が変化しないことが保証されています。

なお、Largeモデルの場合、表3以外にも拡張機能が特殊メモリに割り当てられています。これに関しては「Largeモデルの拡張機能」(p9～)をご覧ください。

表3 特殊メモリ (拡張機能を除く)

I/O番号	記述の可否		名称	説明
	接点(*1)	コイル		
S0	×	○	タイマ基準クロック	このS0が0→1、または1→0に変化する毎に、タイマの経過値が+1増加します。
S10	○	×	1スキャン反転	1スキャン毎に値が反転します。
S11	○	×	10スキャン反転	10スキャン毎に値が反転します。
S12	○	×	100スキャン反転	100スキャン毎に値が反転します。
S13	○	×	1000スキャン反転	1000スキャン毎に値が反転します。
S14	○	×	10000スキャン反転	10000スキャン毎に値が反転します。
S1E	○	×	運転1スキャンON	運転開始後、1スキャン目は値1を、2スキャン目以降は値0をとります。
S1F	○	×	常時ON	運転開始後、常に値1をとります。

(\*1)ラダープログラム中に複数記述可能です。

図1にS12の動作を示します。

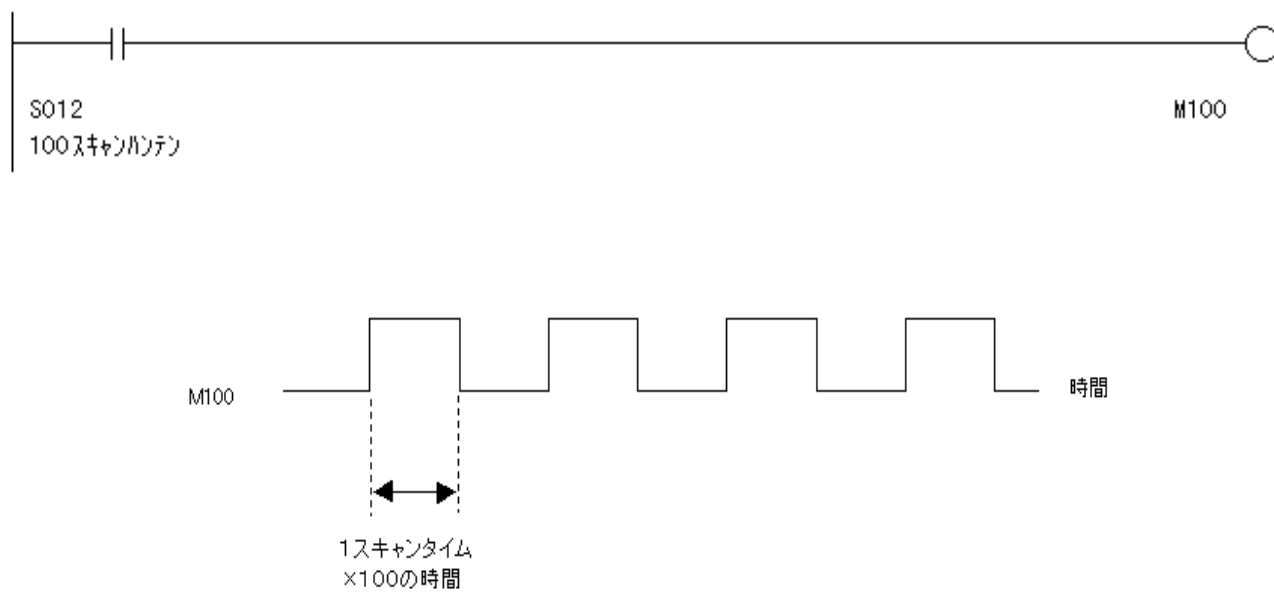


図1 S12動作(100スキャン反転)

S12は100スキャン毎にON/OFFが反転するので、ラダー図上でS12の接点でコイルM100に出力するとM100も100スキャン毎にON/OFFが反転します。

S0(タイマ基準クロック)の使用例については、図6(P6)を参照ください。

## ■ラダーシンボルの意味

ここではラダーシンボル(記号)の意味について説明します。

基本的な考え方として、左の縦線(母線)からスタートして導通した接点で右端のコイル(丸い記号)まで到達すれば、コイルの機能に応じた動作を行います。

### (1)A接点

A接点は、そのメモリの値がON(1)のとき導通となり、OFF(0)のとき非導通になる接点です。

それぞれのI/O番号について、A接点はラダー図の中にいくあってもかまいません。

A接点とコイルが接続された回路を図2に示します。

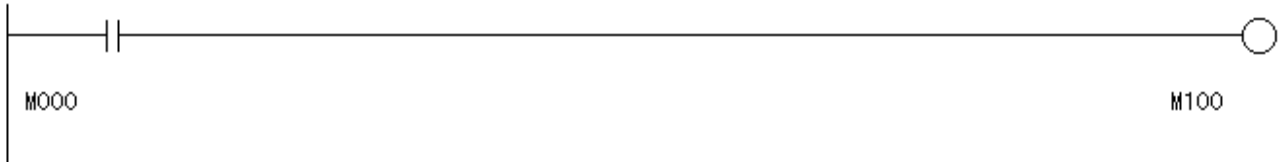


図2 A接点とコイル

この回路を式で表せば

$$M100 \leftarrow [M0]$$

となります。なお、ここで  $[M0]$  はM0の記憶内容を意味します。

### (2)B接点

B接点は、そのI/Oメモリの値がON(1)のとき非導通となり、OFF(0)のとき導通になる接点です。論理回路でのNOTに相当します。

それぞれのI/O番号について、B接点はラダー図の中にいくあってもかまいません。

B接点とコイルが接続された回路を図3に示します。

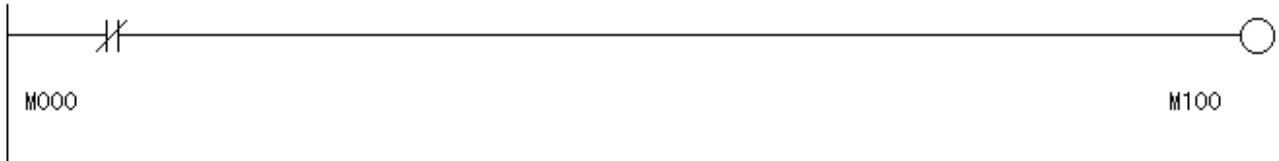


図3 B接点とコイル

この回路を式で表せば

$$M100 \leftarrow [\overline{M0}]$$

となります。なお、ここで  $[\overline{M0}]$  はM0の記憶内容のビット反転(NOT)を意味します。

### (3)コイル

コイルは、励磁(左母線と導通)されたときそのメモリの値がON(1)になり、励磁されないとそのメモリの値がOFF(0)になる素子です。

コイルがONになると、そのコイルと同じI/O番号のA接点は導通、B接点は非導通となります。

それぞれのI/O番号について、コイルはラダー図の中に1つだけしか許されません。同じI/O番号のコイルが複数あると、ラダー図チェック時または変換実行時にエラーと判断されます。

5個の接点とコイルが接続された回路を図4に示します。

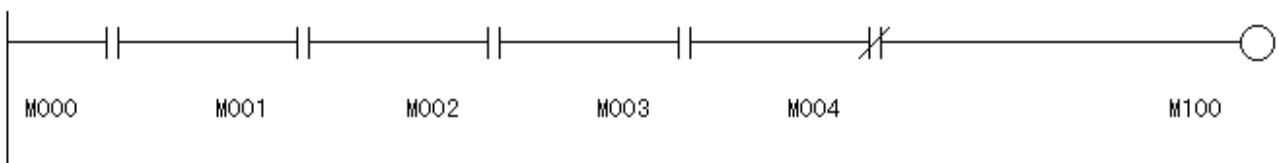


図4 5個の接点とコイル

この回路を電気回路的にみれば、M0からM3までがON(1)かつM4がOFF(0)のとき、コイルM100は励磁(左母線と導通)されON(1)になります。これ以外の場合、コイルM100は励磁されずOFF(0)になります。

この回路を式で表せば

M100 ← [M0] AND [M1] AND [M2] AND [M3] AND  $\overline{M4}$

となります。

#### (4)タイマ(T)

タイマは時間遅れ動作(オンディレイタイマ)をおこないます。

タイマはON/OFFを記憶するメモリ以外に、経過値という数値0～65535を記憶するメモリ(初期値0)を持ちます。

タイマコイルへの入力がON(1)になっても直ちにONにはならず、タイムベース(基準時間)を経過する毎にタイマの経過値が+1されます。タイマの経過値が設定値に到達(すなわち、設定値×タイムベース時間が経過)するとタイマのコイルはONになります。

タイマコイルへの入力がOFF(0)になると、ただちにそのタイマの接点はOFF(0)になり、経過値も0になります。

ラダー図上でタイマコイルを記述する場合、タイマアドレスと設定値(0～65535)を指定します。書式は次のとおりです。

タイマアドレス(T 000～T 07 F) + 空白1文字 + 設定値(0～65535)

タイマコイルとその接点の動作を図5に示します。ここで t はタイムベースを表します。

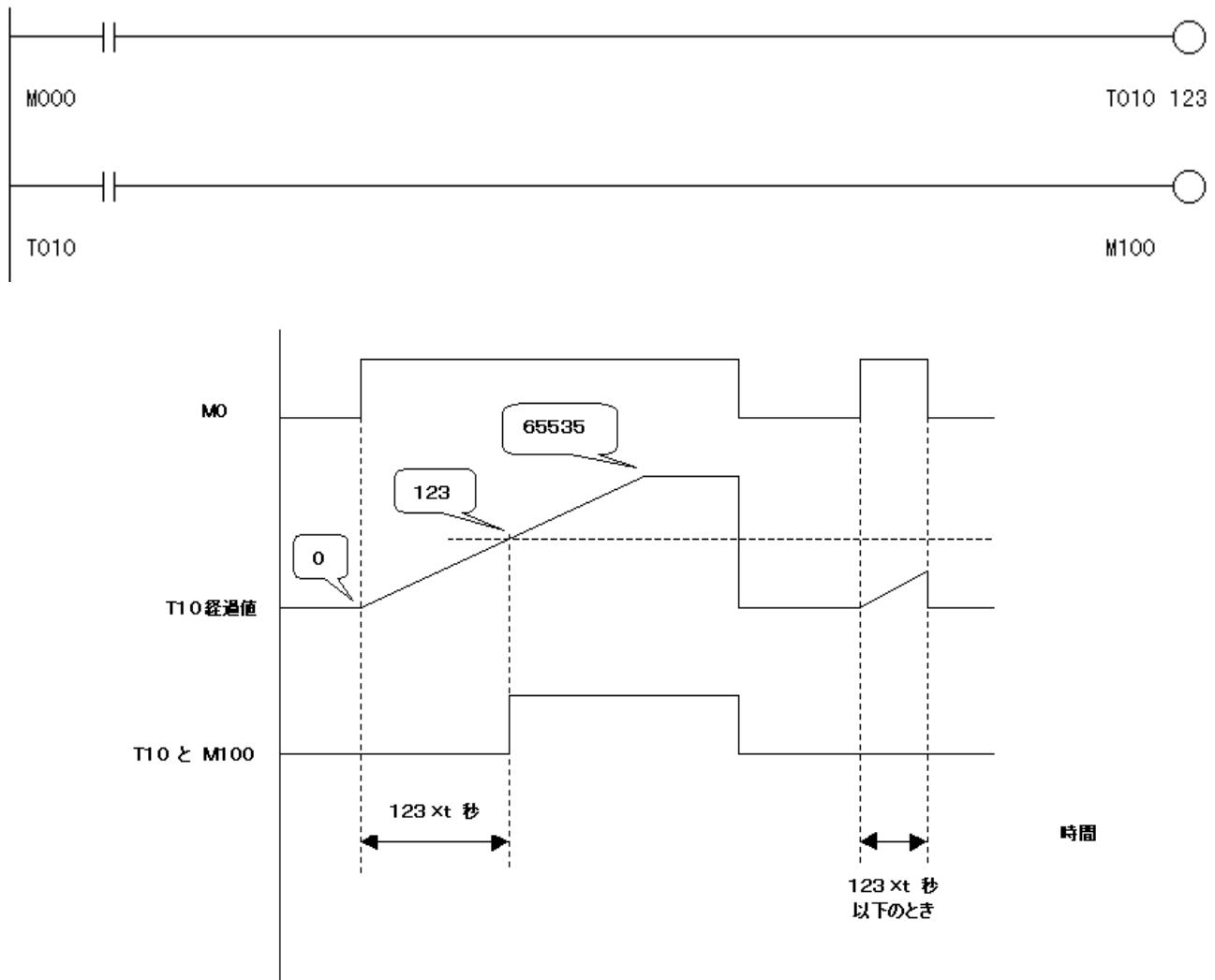


図5 タイマ動作(タイムベースt秒のとき)

タイムベースの設定には特殊メモリS0コイルを使用します。

S0のコイルに200ms周期のクロックを与えれば、100ms毎に変化するためタイムベースは0.1秒(←0.2秒でないことに注意)となり、ラダー図中のすべてのタイマは0.0秒～6553.5秒までの時間を設定できることになります。

正確な時間が必要な場合は、外部信号で毎スキャン毎に(M経由で)S0を更新しますが、図6のように簡易的にS10～S14(1スキャン反転～10000スキャン反転)のいずれかをS0のコイルに与えることでタイムベースを設定することも可能です。



図6 タイムベースの設定にS11を使用する回路

#### (5)カウンタ(C)、CLRコイル

カウンタは計数動作をおこないます。

・カウンタはON/OFFを記憶するメモリ以外に、経過値という数値0～65535を記憶するメモリ(初期値0)を持ちます。

カウンタコイルへの入力に立ち上がり入力(OFFからONへの変化)を与える毎にカウンタの経過値が+1(インクリメント)されます。カウンタの経過値が設定値に到達するとカウンタのコイルはONになります。

ラダー図上でカウンタコイルを記述する場合、カウンタアドレスと設定値(0～65535)を指定します。書式は次のとおりです。

カウンタアドレス(C 000～C 07 F)+空白1文字+設定値(0～65535)

カウンタにはカウンタクリアコイル([CLR])が用意されています。カウンタクリアコイルがON(1)になると(厳密には、ON(1)になった後に同じカウンタ番号のコイルを初めて実行したとき)、そのカウンタの接点はOFF(0)になり、経過値もゼロになります。

カウンタとカウンタクリアコイルはペアで使用しますが、ラダー図中に並べて記述する必要はありません。

カウンタコイルとその接点の動作を図7に示します。

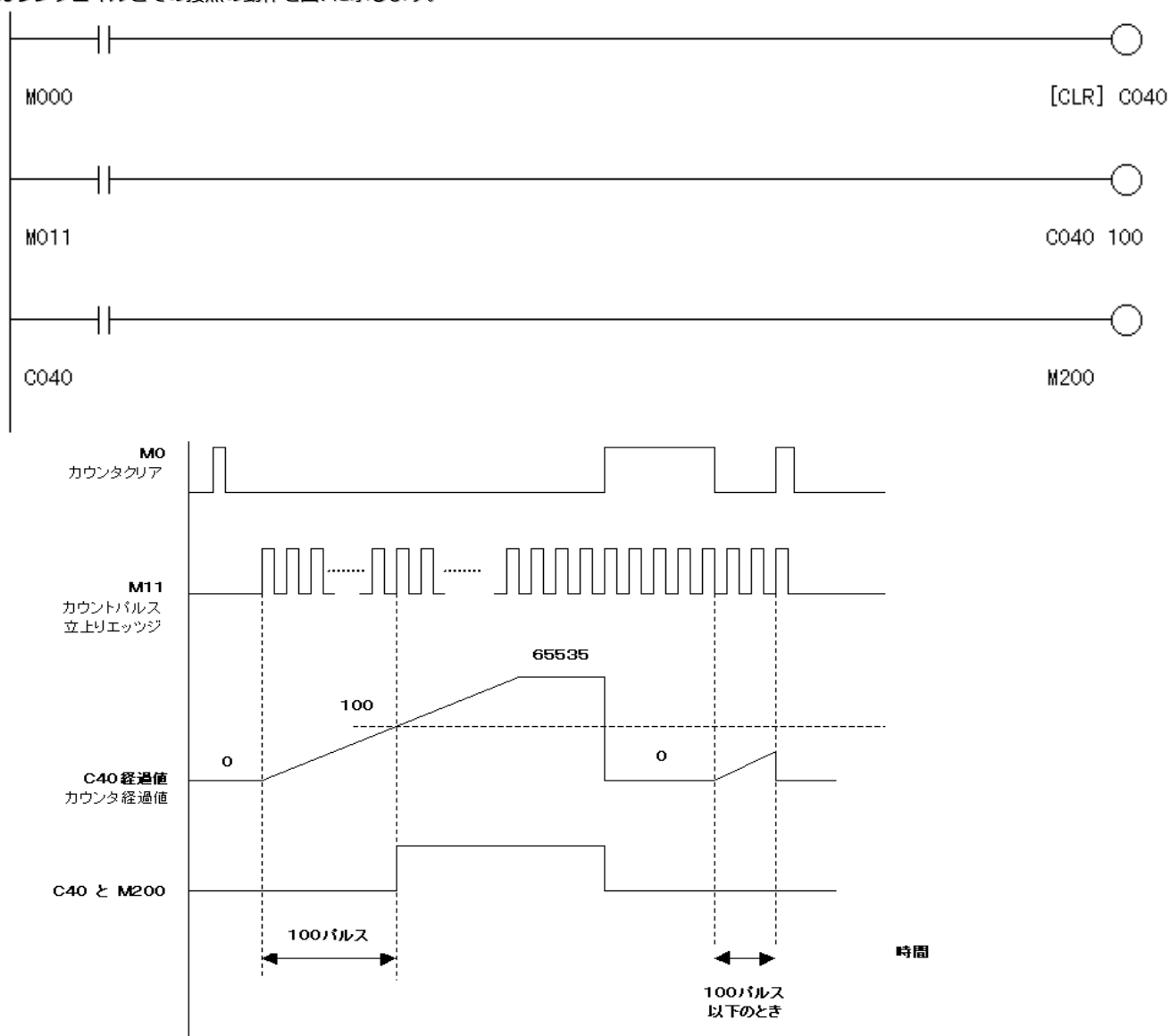


図7 カウンタ動作

#### (7)SET、RESコイル

SET(セット)コイルは、励磁(左母線と導通)したときそのI/O番号の値がON(1)になり、励磁されないとそのI/O番号の値が変化しない(以前の値を保持する)コイルです。

RES(リセット)コイルは、励磁(左母線と導通)したときそのI/O番号の値がOFF(0)になり、励磁されないとそのI/O番号の値が変化しない(以前の値を保持する)コイルです。

同一I/O番号に対してSETとRESの両方を使用して、セット入力、リセット入力のある記憶回路を構成することができます。セット、リセットは工程歩進制御の状態を記憶するのによく使われます。

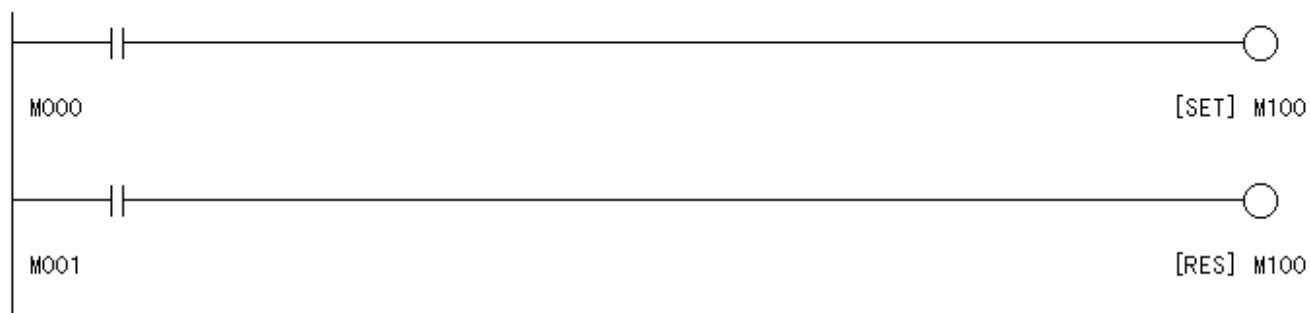


図8 SET、RESコイル

図8の回路では、M0がONのときM100がONになります。また、M1がONのときM100がOFFになります。M0とM1の両方がOFFのときはM100は以前の値を保持(記憶)したままです。

#### (7)MCS、MCR

[MCS]マスタコントロールセットは、ラダー図が連続した複数の回路にわたり共通の接点を使用しているときに、この共通の接点を1つにまとめて記述するために使用します。この共通の接点と[MCS]を接続した回路を記述すると、以降の回路ではこの共通の接点を省略して記述できます。この影響は[MCR](マスタコントロールリセット)が出現するまで続きます。

共通の接点(M0、M1、M2の直列接続)を3回路にわたり使用している回路を図9に示します。

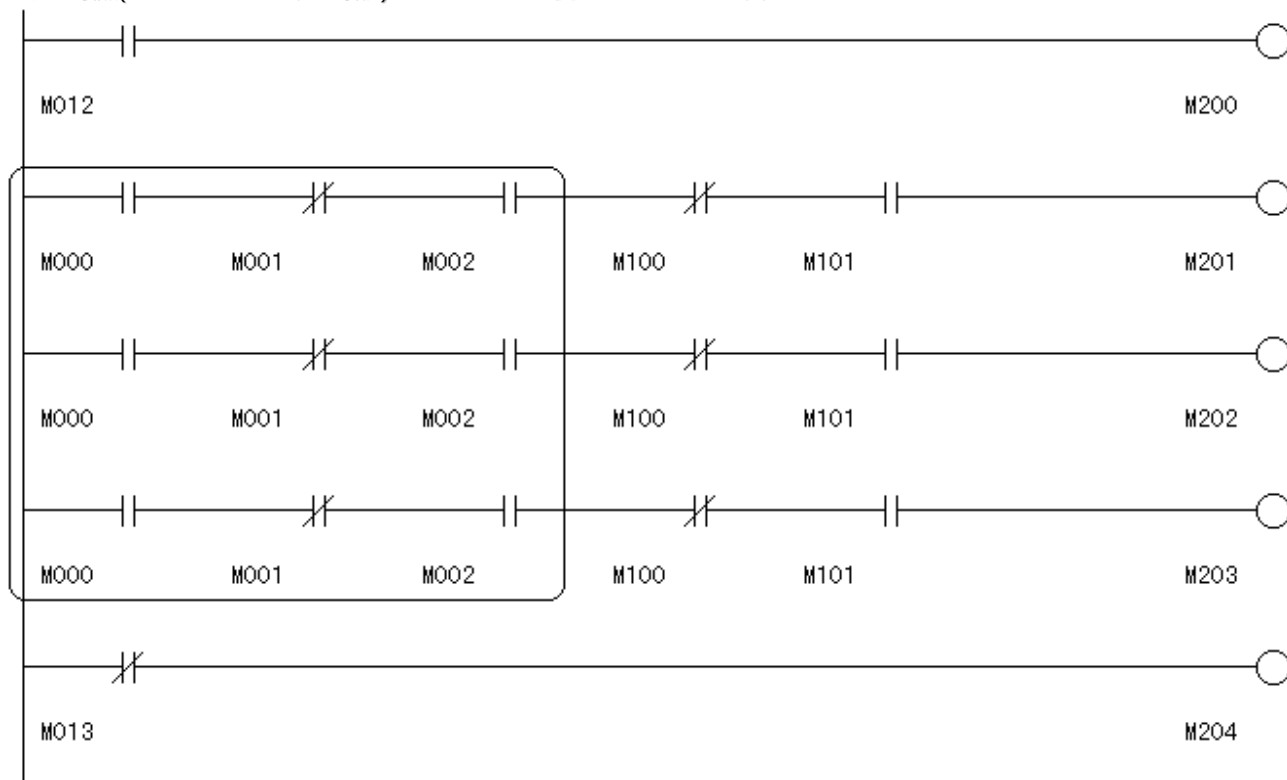


図9 マスタコントロールの説明(共通の接点部分がある)

図9の回路の枠内の共通部分は、図10の枠内のように[MCS]を使って1回路で記述できます。

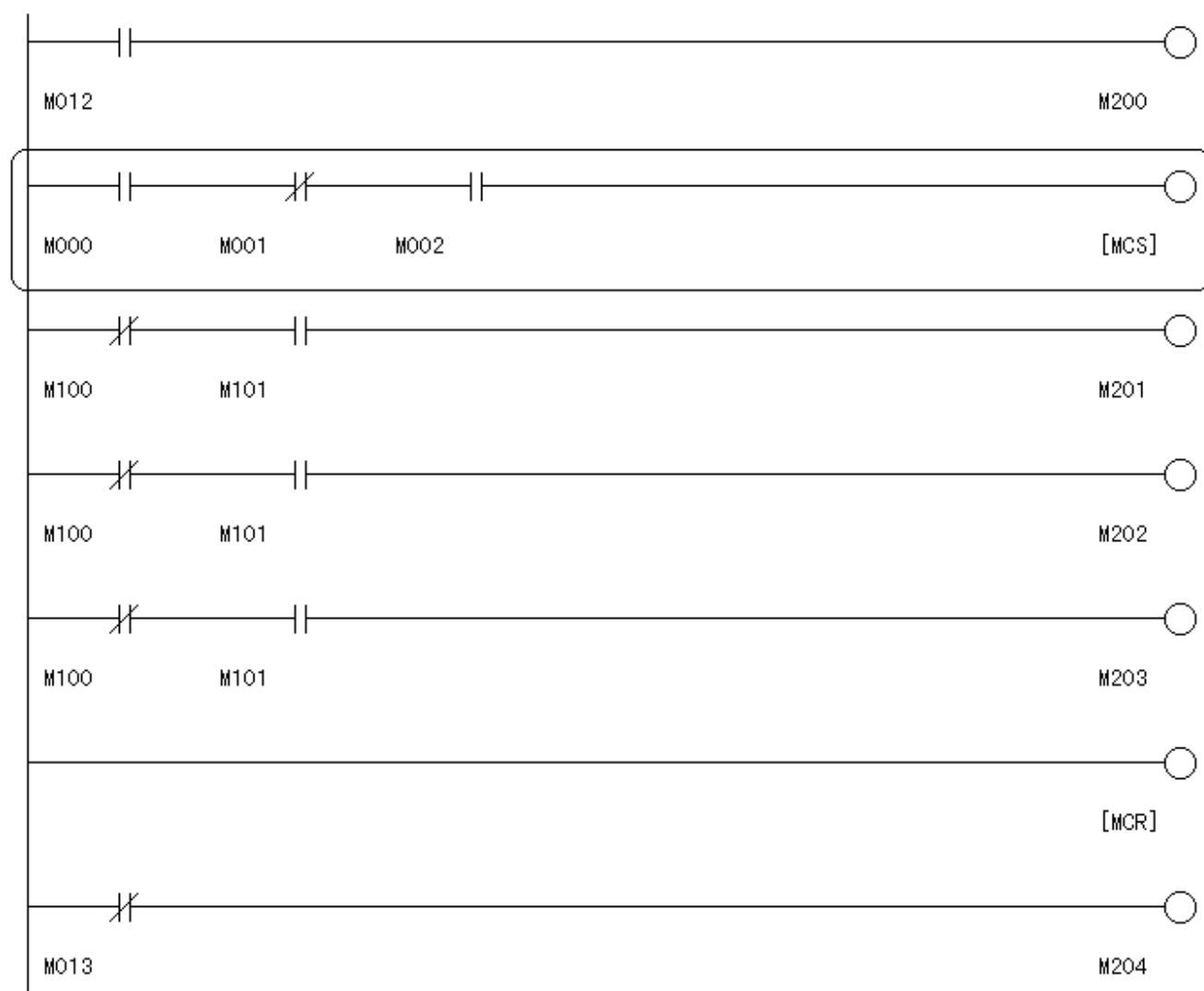


図10 マスタコントロールの説明(共通の接点部分をMCSを使って記述)

ラダー図上で、[MCS]と[MCR]は対で使用されます。[MCS]と[MCR]が1:1に対応していなかったり、対応が入れ子構造になっていると、ラダー図チェック時または変換実行時にエラーと判断されます。[MCR]は、左母線から(途中で接点を入れずに)直結してコイルを記述します。

[MCS]と[MCR]は、非常停止や手動・自動切替など、主にラダー図全体に関わる共通の条件を省略して記述するのによく使われます。

#### (8)END

ラダー図の末尾に、1スキャン終了を示す[END](エンド)を必ず記述します。ラダー図の末尾に[END]がないと、ラダー図チェック時または変換実行時にエラーと判断されます。

END回路を図11に示します。



図11 END回路



■Largeモデルの拡張機能

LD Cv2! Ver2.0から、仮想PLCに「Largeモデル」を選択することができます。  
Largeモデルでは、標準モデルに対し、ワードメモリ、レジスタ、微分出力命令、ワード演算命令が追加されています。

(1)ワードメモリ(W0～W7FF)

ワードメモリ(W)は、少なくとも16ビットの記憶サイズを持つリード・ライト可能なメモリで、2048ワードあります。

W0	16ビット	W0	32ビット
W1	16ビット	W1	32ビット
...	...	...	...
W7FF	16ビット	W7FF	32ビット

ワードメモリ(16ビットの例)

ワードメモリ(32ビットの例)

ワードメモリと開発対象であるマイコンのI/Oポートとの間でデータの転送(I/Oリフレッシュ)を定期的におこなうことで、I/Oメモリは発対象のI/Oポートと連動します。ワードデータの I/Oリフレッシュには、LD Cv2!が生成する関数plcのワードWriteコマンド(w)、ワードReadコマンド(r)を使用します。(小文字であることに注意ください)

(2)レジスタ(GRA、GRB)

仮想PLC内部に、少なくとも16ビットの記憶サイズを持つ汎用のレジスタGRAとGRBがあります。  
これらのレジスタは、ワードメモリ(W0～W7FF)とのデータ転送や、レジスタ(GRA、GRB)間の演算・転送に使用します。

GRA	16ビット	GRA	32ビット
GRB	16ビット	GRB	32ビット

レジスタ(16ビットの例)

レジスタ(32ビットの例)

ワードメモリ(W0～W7FF)、レジスタ(GRA、GRB)の実体は、Cコードで定義された「unsigned int型」の配列、変数です。つまり 仮想PLCにおけるワードメモリ、レジスタの記憶サイズは、使用するCコンパイラがunsigned int型を何ビットとして扱うかによって決まります。  
16ビットの記憶サイズを超える四則演算や、論理演算では16ビットと32ビットで演算結果が異なる場合がありますので注意が必要です。  
(例1)WADD命令で65535+1を計算すると、16ビットの場合0、32ビットの場合65536となります。  
(例2)WNOT命令でHFFFFをビット反転すると、16ビットの場合H0000、32ビットの場合HFFFF0000となります。

(3)微分出力命令(S20)

コイルS20の駆動信号がOFFからONに変化したとき、S20の引数で指定したI/Oメモリ(M)を1 スキャンだけONさせる命令です。(表4)  
S20の引数で指定したI/Oメモリ(M)の変化を検出するのではなく、S20の駆動信号の変化を検出します。  
S20の引数(微分出力先Mの番号)が異なれば、S20のコイルをラダー図中に複数個記述が可能で、それぞれ独立して動作します。

表4 微分出力命令

名称	コイル 番号	引数 (*1)(*2)	I/Oコメント (*3)	動作	備考
微分出力	S20	Mhhh	DIF (M)	コイルS20への入力信号がOFFからONに変化した時、その直後から1 スキャンの期間だけMhhhがON(微分信号)となります。 このMhhhは、接点としてのみ記述することが可能です。	運転開始直後からコイルS20への入力信号がONだった場合、「OFFからONへの変化」とは扱いません。

(\*1)コイル番号とコイル引数の間には、空白1文字を記述します。

- (\*2) MhhhはI/Oメモリ(M0～M7FF)を意味します。
- (\*3)I/Oコメントは命令の動作を解りやすくするためのデフォルトの文字列であり、書き換えても命令動作には影響を与えません。

図12にS20の動作を示します。

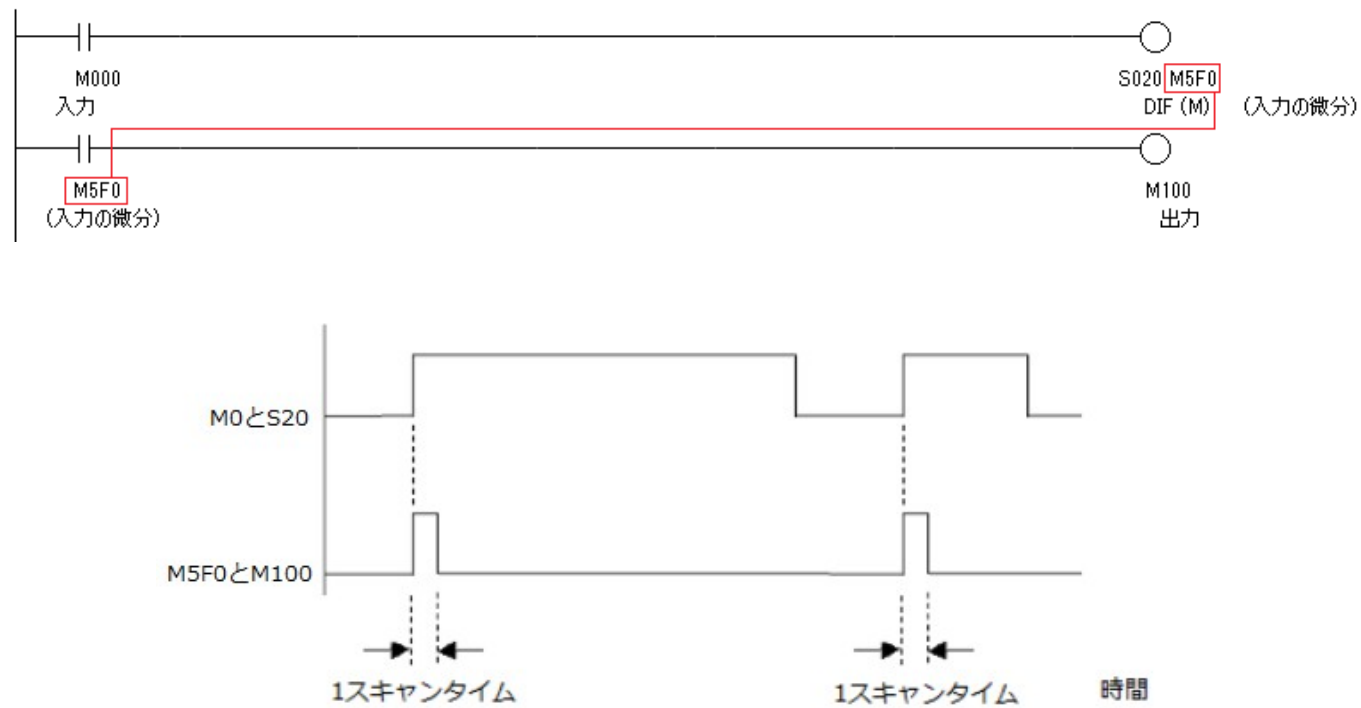


図12 S20動作(微分出力)

- (4)ワード演算命令(S30～S3F)、コンディションフラグ(S2D～S2F)
- ワード演算命令は、ワードメモリ(W0～W7FF)、レジスタ(GRA、GRB)を対象とする命令の総称です。
- ワード演算命令は、特殊メモリ(S)へのコイル出力として記述します。(接点として記述不可)
- なお、プログラムを読み易くするため、便宜的にデフォルトのI/Oコメントを ワード演算命令のニーモニックにしています。

ワード演算命令(S□□)は、最初に(通常のコイルと同じく)コイルS□□へ書込動作を行います。次に、コイルS□□がONの時に限り、Sの番号□□に応じたワード演算を実行します。コイルS□□がOFFの場合はワード演算を実行しません。

Sの番号□□の後に引数が付く命令の場合、その引数は演算対象の「定数」または「ワードメモリ(W)」を意味します。

- 図13にワード演算命令を使った簡単なプログラムを示します。
- このプログラムは接点M0がONのときのみ、ワード演算を実行します。
- 最初のS31コイルではレジスタGRAに数値12345をロードし、次のS34コイルではレジスタGRAの値(ここでは10進数12345)をワードメモリW5FFにストアしています。

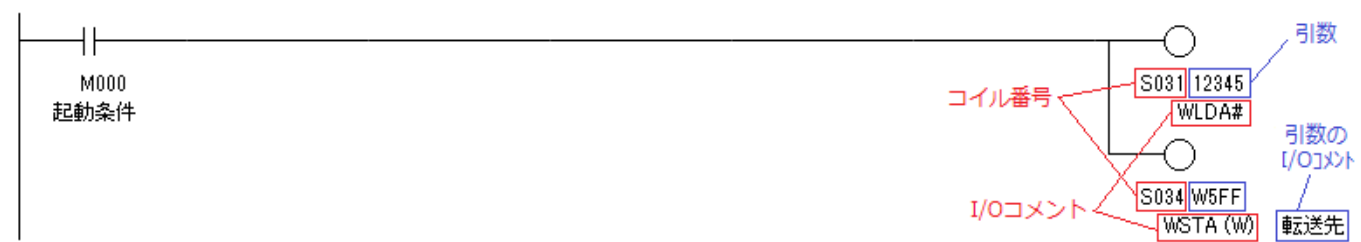


図13 ワード演算命令を使った簡単なプログラム

ワード演算は複数個の命令で使うことが多く、一般に多重出力回路(1回路に複数個のコイルがある回路)となります。

表6にワード演算命令を示します。

表6 ワード演算命令

CF1～CF3欄の記号の意味：変化なし(-)、実行結果により変化(0/1)

名称	コイル 番号	引数 (*1)	I/Oコメント (*2)	コイルON時の動作				備考
				演算動作	CF1	CF2	CF3	
ワード ロード	S30	Whhh	WLDA (W)	GRA ← Whhhの値	-	-	-	(例) S30 W45E ワードメモリW45Eの値(16ビット)をGRA にロードします。
	S31	dddd または \$hhhh	WLDA#	GRBA← 定数dddd または GRA ← 16進定数hhhh	-	-	-	(例) S31 12345 10進定数12345をGRAにロードします。
	S32	Whhh	WLDB (W)	GRB ← Whhhの値	-	-	-	
	S33	dddd または \$hhhh	WLDB#	GRB ← 10進定数dddd または GRB ← 16進定数hhhh	-	-	-	(例) S33 \$FE00 16進定数FE00をGRBにロードします。
ワード ストア	S34	Whhh	WSTA (W)	Whhh ← GRA	-	-	-	(例)S31 W7FF GRAの値をW7FFにストアします。
	S35	Whhh	WSTB (W)	Whhh ← GRB	-	-	-	
ワード 加算	S36	なし	WADD	GRA ← GRA+GRB	-	-	-	GRAの値にGRBの値を加算してGRAにス トアします。 演算の結果GRAがオーバーフローした場合、 オーバーフロー部分は無視されます。
ワード 減算	S37	なし	WSUB	GRA ← GRA-GRB	-	-	-	演算の結果GRAがオーバーフローした場合、 オーバーフロー部分は無視されます。
ワード AND	S38	なし	WAND	GRA ← GRA and GRB	-	-	-	レジスタGRAとGRBを各ビット間でAND 演算してGRAにストアします。
ワード OR	S39	なし	WOR	GRA ← GRA or GRB	-	-	-	
ワード NOT	S3A	なし	WNOT	GRA ← GRAの各ビット反転	-	-	-	
ワード 比較	S3B	なし	WCMP	S2D～S2F ← GRAとGRBの比較結果	0/1	0/1	0/1	GRAとGRBを符号なし整数として比較し、 表7の通りCF1～CF3が変化します。
ワード 交換	S3C	なし	WSWAP	GRA とGRB のデータを交換	-	-	-	
ワード 補数	S3D	なし	WNEG	GRA ← GRAの2の補数	-	-	-	GRAを符号あり整数とみなした場合 GRA ← -GRA に相当します。
ワード 乗算	S3E	なし	WMUL	GRA ← GRA×GRB	-	-	-	符号なし整数の乗算です。 演算の結果GRAがオーバーフローした場合、 オーバーフロー部分は無視されます。

ワード 除算	S3F	なし	WDIV	$GRA \leftarrow GRA \div GRB$ $GRB \leftarrow \text{余り}$ $CF3 \leftarrow \text{ゼロ割の有無}$	-	-	0/1	符号なし整数の除数です。 除数( $GRB \neq 0$ )の場合、 $CF3=0$ となり除算 を実行します。 除数( $GRB=0$ )の場合、 $CF3=1$ となり除算 を実行しません。
-----------	-----	----	------	---	---	---	-----	--

(\*1)コイル番号とコイル引数の間には、空白 1 文字を記述します。  
 ddddは10進数定数(0～65535)、\$hhhhは16進数定数(0000～FFFF)、Whhhはワードメモリ(W0～W7FF)を意味します。  
 (\*2)I/Oコメントは命令の動作を解りやすくするためのデフォルトの文字列であり、書き換えても命令動作には影響を与えません。

表7 ワード比較命令 (WCMP)実行直後のコンディションフラグ

接点番号	I/Oコメント	コンディションフラグの値		
		GRA<GRB のとき	GRA>GRB のとき	GRA=GRB のとき
S2D	CF1(<)	ON	OFF	OFF
S2E	CF2(>)	OFF	ON	OFF
S2F	CF3(=)	OFF	OFF	ON

以上

来歴(LD Cv1/LD Cv2! 仮想PLC仕様書)  
 2020/01/03 新規作成、第1版とする。  
 2020/11/01 Largeモデル追加に伴い記述を追加・変更、第2版とする。  
 2020/11/29 ワードに関する記述を変更、ワード命令を追加、他、第3版とする。