TB3218

CCLでの開始に際して

序説

著者: Cristian Pop, Microchip Technology Inc.

構成設定可能な注文論理回路(CCL:Configurable Custom Logic)はMicrochip tinyAVR® 0及び1系とmegaAVR® 0系のﾃﾞﾊﾞｲｽに対するﾁｯﾌﾟ上論理回路機能作成を許す設定可能な論理回路周辺機能です。CCLはCPU実行と無関係に動作する設定可能な組み合わせ順次論理回路を提供します。これはﾃﾞﾊﾞｲｽ ﾋﾟﾝ、事象、または他の内部周辺機能のような内部と外部の広い範囲の入力に接続することができ、ﾃﾞﾊﾞｲｽの周辺機能と外部装置間の”接続論理回路”として扱うことができます。この技術概要は以下の機能を実装するためにCCLをどう使うかを説明します。

- **ANDｹﾞｰﾄ論理回路**:
  単純なANDｹﾞｰﾄ論理回路を実装するのにCCLを使います。
- **状態解読器**:
  外部信号の特定状況を検出するのにCCL組み合わせ論理回路をどう使うかを示します。
- **SRﾗｯﾁ**:
  SRﾗｯﾁを作成するのに内部CCL順次論理回路を使います。
- **ﾏﾝﾁｪｽﾀ符号器**:
  ﾏﾝﾁｪｽﾀ符号器を実装するのに他の周辺機能との組み合わでCCLをどう使うかを実演します。

注: ｺｰﾄﾞ例はATmega4809 Xplained Pro基板 (ATMEGA4809-XPRO)を使って開発されました。

本書は一般の方々の便宜のため有志により作成されたもので、Microchip社とは無関係であることを御承知ください。しおりの[はじめに]での内容にご注意ください。

目次

© 2019 Microchip Technology Inc.

1. 関連ﾃﾞﾊﾞｲｽ

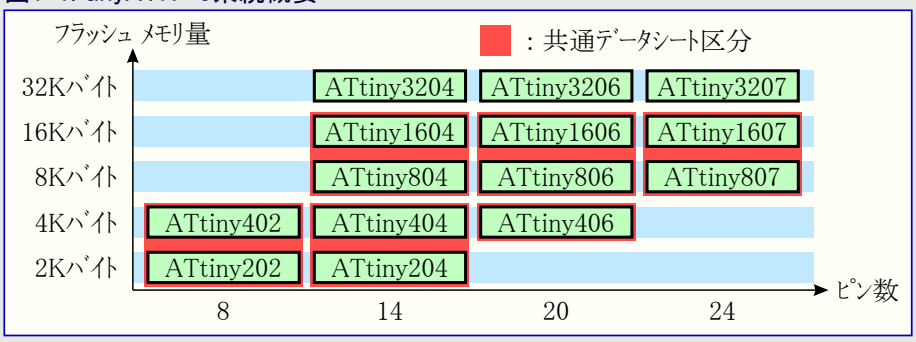
本章はこの資料に関連するﾃﾞﾊﾞｲｽを一覧にします。

1.1. tinyAVR® 0系統

下図はﾋﾟﾝ数の変種とﾒﾓﾘ量を展開してtinyAVR® 0系統ﾃﾞﾊﾞｲｽを示します。

- これらのﾃﾞﾊﾞｲｽが完全にﾋﾟﾝと機能が互換のため、垂直方向移植はｺｰﾄﾞ変更なしで可能です。
- 左への水平方向移植はﾋﾟﾝ数、従って利用可能な機能を減らします。

図1-1. tinyAVR® 0系統概要



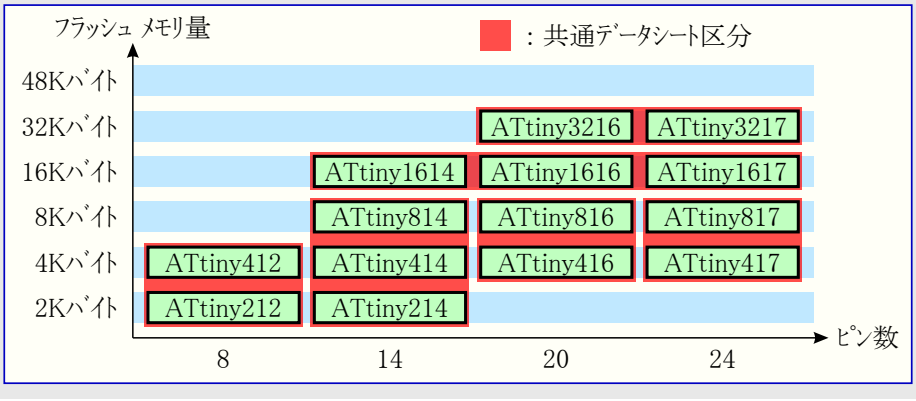
異なるﾌﾗｯｼｭ ﾒﾓﾘ量を持つﾃﾞﾊﾞｲｽは一般的に異なるSRAMとEEPROMの量を持ちます。

1.2. tinyAVR® 1系統

下図はﾋﾟﾝ配置変種とﾒﾓﾘ量を展開してtinyAVR® 1系統ﾃﾞﾊﾞｲｽを示します。

- これらのﾃﾞﾊﾞｲｽがﾋﾟﾝ互換で同じまたはより多くの機能を提供するため、垂直上方向移植はｺｰﾄﾞ変更なしに可能です。下方向移植はより少ない利用可能ないくつかの周辺機能の実体のためにｺｰﾄﾞ変更が必要かもしれません。
- 左への水平方向移植はﾋﾟﾝ数、従って利用可能な機能を減らします。

図1-2. tinyAVR® 1系統概要



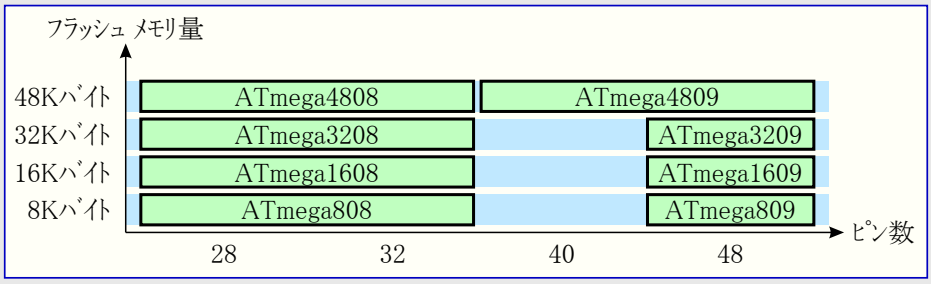
異なるﾌﾗｯｼｭ ﾒﾓﾘ量を持つﾃﾞﾊﾞｲｽは一般的に異なるSRAMとEEPROMの量を持ちます。

1.3. megaAVR® 0系統

下図はﾋﾟﾝ配置変種とﾒﾓﾘ量を展開してmegaAVR® 0系統ﾃﾞﾊﾞｲｽを示します。

- これらのﾃﾞﾊﾞｲｽが完全にﾋﾟﾝと機能が互換のため、垂直方向移植はｺｰﾄﾞ変更なしで可能です。
- 左への水平方向移植はﾋﾟﾝ数、従って利用可能な機能を減らします。

図1-3. megaAVR® 0系統概要



異なるﾌﾗｯｼｭ ﾒﾓﾘ量を持つﾃﾞﾊﾞｲｽは一般的に異なるSRAMとEEPROMの量を持ちます。

© 2019 Microchip Technology Inc.

2. 概要

CCL周辺機能は1つの参照表(LUT:Lok-Up Table)対を持ちます。各LUTは真理値表、同期部、濾波部、端検出部と共に3つの入力から成ります。各LUTは3つの入力を持つ使用者設定可能な論理式として出力を生成することができ、CCLを持つどのﾃﾞﾊﾞｲｽも最低2つの利用可能なLUTを持ちます。これらの入力は個別に遮蔽することができます。出力は入力組み合わせから生成して尖頭雑音を除去するために濾波することができます。任意選択の順次論理回路単位部を許可することができます。順次単位部への入力は2つの独立した入力、隣接LUT出力(LUT0/LUT1)によって個別に制御され、複雑な波形生成を許します。

真理値表

これは単純な論理部(AND、OR、NAND、NOR、XOR)、またはLUTの各々の3つまでの入力の真理値表を使う独自のものを作成することが可能です。3つを超える入力が必要とされる時は論理ｹﾞｰﾄを作成するのに複数接続されたLUTが使われます。

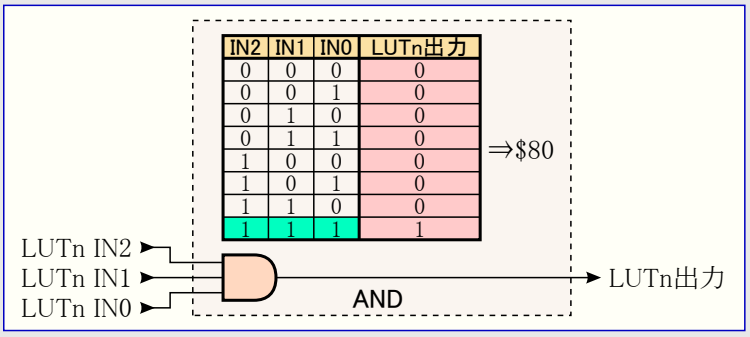
特定組み合わせ論理機能を定義するのにCCL単位部は真理値表を使います。真理値表は論理回路が3入力の様々な組み合わせにどう応じるかを示します。入力ﾋﾞｯﾄ(IN2～0)の各組み合わせは各々のTRUTHnﾚｼﾞｽﾀの1ﾋﾞｯﾄに対応します。

下は3つの入力を使っていくつかの一般的な論理ｹﾞｰﾄを作成する方法のいくつかの例です。

ANDｹﾞｰﾄからHigh(1)出力を得るには全ての入力がHigh(1)でなければなりません。右の真理値表を見ると、TRUTH7は3つ全ての入力が使われる場合にだけこの必要条件を満たします。これはTRUTH7がHigh(1)で残りがLow(0)でなければならないことを意味し、TRUTHnﾚｼﾞｽﾀで使われるべき16進値$80に帰着します。

```
CCL.TRUTHn = 0x80;
```

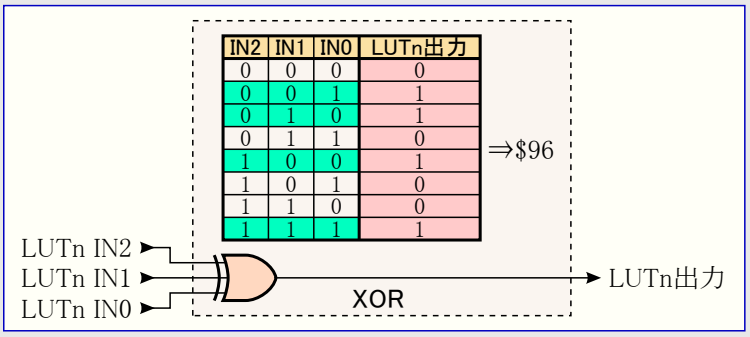
図2-1. AND論理回路



XORｹﾞｰﾄからHigh(1)出力を得るにはHigh(1)入力数が奇数でなければなりません。右の真理値表を見ると、TRUTH7,4,2,1がこの必要条件を満たします。これはこれらのﾋﾞｯﾄがHigh(1)で残りがLow(0)でなければならないことを意味し、TRUTHnﾚｼﾞｽﾀで使われるべき16進値$96に帰着します。

```
CCL.TRUTHn = 0x96;
```

図2-2. XORｹﾞｰﾄ



3つの入力のどれかが必要とされない時に、未使用入力は遮蔽(Low接続)することができます。望む論理を得るためにﾋﾞｯﾄがどのように設定されるかを判断するために真理値表を見る時に、遮蔽された入力のTRUTHﾋﾞｯﾄは'0'が使われ得るだけです。下は様々な入力が遮蔽されたいくつかの例です。

図2-3. IN0遮蔽の2入力ANDｹﾞｰﾄ

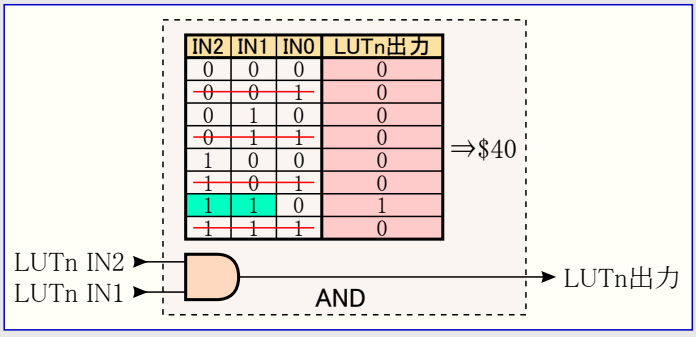
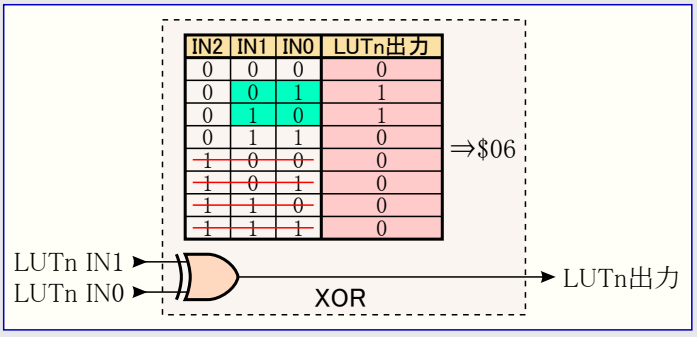


図2-4. IN2遮蔽の2入力XORｹﾞｰﾄ



いくつかの応用は3つを超える論理入力を必要とします。CCL単位部は次のLUTの出力を直接LUT入力へ内部的繋げる任意選択を提供します。例えば、LUT0とLUT1が論理機能作成に使われる場合、LUT1の出力をLUT0の入力へ内部的に接続することができます。(**訳補**:最大番号のLUT(この例ではLUT1)でのこの連結動作はLUT0の出力がこのLUTの入力になります。)

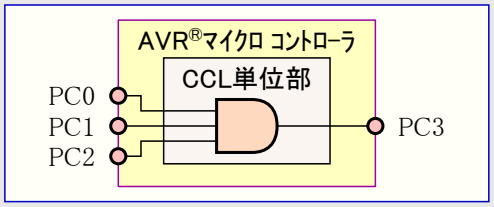
CCLの使用は外部論理回路に対する必要性をなくし、部品費用を減らしてもっと効率的に応用の時間に際どい部分を処理することをCPUに許します。

© 2019 Microchip Technology Inc.

3. ANDｹﾞｰﾄ論理回路

CCL単位部は3入力までの論理回路ｹﾞｰﾄを実装するのに使うことができます。以下の例はANDｹﾞｰﾄを実装するのにCCLのLUT1をどう構成設定して使うかを示します。

図3-1. ANDｹﾞｰﾄ論理回路としてCCLを使用



最初の段階では、LUT制御(LUTnCTRLBとLUTnCTRLC)ﾚｼﾞｽﾀでINSELx3～0ﾋﾞｯﾄを使って入出力ﾋﾟﾝが入力として選ばれます。

図3-2. LUTn制御B(LUTnCTRLB)ﾚｼﾞｽﾀ

ﾋﾞｯﾄ	7	6	5	4	3	2	1	0
	INSEL13～0				INSEL03～0			
ｱｸｾｽ種別	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ﾘｾｯﾄ値	0	0	0	0	0	0	0	0

図3-3. LUTn制御C(LUTnCTRLC)ﾚｼﾞｽﾀ

ﾋﾞｯﾄ	7	6	5	4	3	2	1	0
					INSEL23～0			
ｱｸｾｽ種別	R	R	R	R	R/W	R/W	R/W	R/W
ﾘｾｯﾄ値	0	0	0	0	0	0	0	0

下表は全ての入力に対するINSEn3～0任意選択を要約します。

図3-4. CCL入力選択任意選択

値	入力元	INSEL0	INSEL1	INSEL2
0000 ($0)	MASK	なし		
0001 ($1)	FEEDBACK	LUTn		
0010 ($2)	LINK	LUT(n+1)		
0011 ($3)	EVENT0	事象入力元0		
0100 ($4)	EVENT1	事象入力元1		
0101 ($5)	IO	IN0	IN1	IN2
0110 ($6)	AC	AC0 OUT		
0111 ($7)	-			
1000 ($8)	USART	USART0 TXD	USART1 TXD	USART2 TXD
1001 ($9)	SPI	SPI0 MOSI	SPI0 MOSI	SPI0 SCK
1010 ($A)	TCA0	WO0	WO1	WO2
1011 ($B)	-			
1100 ($C)	TCB	TCB0 WO	TCB1 WO	TCB2 WO
その他	-			

これは以下のｺｰﾄﾞに変換できます。

```
CCL.LUT1CTRLB = CCL_INSEL0_IO_gc | CCL_INSEL1_IO_gc;
CCL.LUT1CTRLC = CCL_INSEL2_IO_gc;
```

次の段階は選んだﾋﾟﾝでANDｹﾞｰﾄを実装するために正しい組み合わせ論理を生成するようにLUT1用真理値表を構成設定することです。従って、真理値表は0x80の値を持ちます。

```
CCL.TRUTH1 = 0x80;
```

© 2019 Microchip Technology Inc.

次の段階は解読部の出力、特にこの例ではPC3入出力ポート ピンを構成設定することです。これはLUT0CTRLAレジスタのOUTENビットを設定(1)することによって行われます。

図3-5. LUTn制御A(LUTnCTRLA)レジスタ

ビット	7	6	5	4	3	2	1	0
	EDGEDET	OUTEN	FILTSEL1,0			CLKSRC2~0		ENABLE
アクセス種別	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

これは以下のコードに変換できます。

```
CCL.LUT1CTRLA = CCL_OUTEN_bm;
```

入出力ピンでLUTn出力を許可することにより、対応するピンに対する設定は上書きされます。入力系列の解読を許可するにはCCLと使われるLUTが許可されることが必要です。それはLUT0CTRLAレジスタのENABLEビットを使って行われます

```
CCL.LUT1CTRLA |= CCL_ENABLE_bm;
```

準備を完了するにはCTRLAレジスタのCCL全体許可ビットを使ってCCL単位部も許可されるべきです。

図3-6. CCL制御A(CTRLA)レジスタ

ビット	7	6	5	4	3	2	1	0
		RUNSTDBY						ENABLE
アクセス種別	R	R/W	R	R	R	R	R	R/W
リセット値	0	0	0	0	0	0	0	0

```
CCL.CTRLA = CCL_ENABLE_bm;
```

GitHubでコード例を見てください。
貯蔵庫を閲覧するにはクリックしてください。

助言: 完全なコード例は「追補」章でも利用可能です。

4. 状態解読器

応用はピンに信号の特定の組み合わせ(型)が現れる時を検出することが必要かもしれません。結合論理ゲートにより、CPUを必要とすることなく、外部信号に対する簡単な状態解読器を実装することができます。

(訳注) 右図は原書の本図が本文及びソフトウェアと矛盾するため、本書では整合するようにこの図を修正しています。

この例では、入力ピンで'10110'型の存在を解読するのにCCL単位部が使われます。対応する入力ピンに接続されたLUT0とLUT1が使われます。

以下の図で示されるように、各種入力任意選択からの入力選択はLUT制御(LUTnCTRLBとLUTnCTRLC)レジスタでINSELx3～0ビットを使って行われます。

図4-1. 状態解読器としてAVRマイクロ コントローラ使用

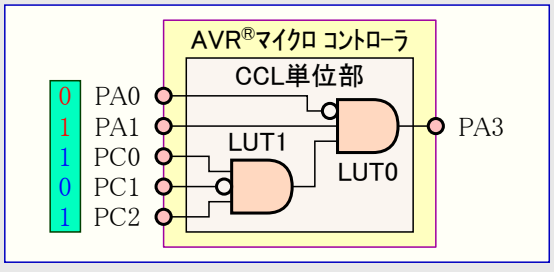


図4-2. LUTn制御B(LUTnCTRLB)レジスタ

ビット	7	6	5	4	3	2	1	0
	INSEL13~0				INSEL03~0			
アクセス種別	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

図4-3. LUTn制御C(LUTnCTRLC)レジスタ

ビット	7	6	5	4	3	2	1	0
					INSEL23~0			
アクセス種別	R	R	R	R	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

© 2019 Microchip Technology Inc.

下表は全ての入力に対するINSEn3～0任意選択を要約します。

図4-4. CCL入力選択任意選択

値	入力元	INSEL0	INSEL1	INSEL2
0000 ($0)	MASK	なし		
0001 ($1)	FEEDBACK	LUTn		
0010 ($2)	LINK	LUT(n+1)		
0011 ($3)	EVENT0	事象入力元0		
0100 ($4)	EVENT1	事象入力元1		
0101 ($5)	IO	IN0	IN1	IN2
0110 ($6)	AC	AC0 OUT		
0111 ($7)	-			
1000 ($8)	USART	USART0 TXD	USART1 TXD	USART2 TXD
1001 ($9)	SPI	SPI0 MOSI	SPI0 MOSI	SPI0 SCK
1010 ($A)	TCA0	WO0	WO1	WO2
1011 ($B)	-			
1100 ($C)	TCB	TCB0 WO	TCB1 WO	TCB2 WO
その他	-			

この例について、LUT0入力に接続されたLUT1の出力(繋がり)を持つ2つの隣接LUT(LUT0とLUT1)が使われます。

```
CCL.LUT0CTRLC = CCL_INSEL2_LINK_gc;
```

LUT0の他の2つの入力とLUT1の3つ全ての入力が入出力ﾋﾟﾝに接続されます。

```
CCL.LUT0CTRLB = CCL_INSEL0_IO_gc | CCL_INSEL1_IO_gc;
CCL.LUT1CTRLB = CCL_INSEL0_IO_gc | CCL_INSEL1_IO_gc;
CCL.LUT1CTRLC = CCL_INSEL2_IO_gc;
```

以下の段階は選んだﾋﾟﾝで'10110'を検出するように正しい組み合わせ論理を生成するため、LUT0とLUT1用の真理値表を構成設定することです。TRUTH1表は上位3ﾋﾞｯﾄ('10110')に対する型('101')を解読するのに使われます。

(**訳補**:LUTの真理値表の構成表)

IN2	0	0	0	0	1	1	1	1
IN1	0	0	1	1	0	0	1	1
IN0	0	1	0	1	0	1	0	1
OUT	TRUTH0	TRUTH1	TRUTH2	TRUTH3	TRUTH4	TRUTH5	TRUTH6	TRUTH7

```
CCL.TRUTH1 = 0x20;
```

LUT0は入力型('10110')からの下位2ﾋﾞｯﾄと第3入力(IN2)でのLUT1の解読出力を入力として持ち、2進値系列'110'の解読に帰着します。この場合の真理値表の値は0x40です。

```
CCL.TRUTH0 = 0x40;
```

次の段階は解読部の出力、特にこの例では入出力ﾎﾟｰﾄのPA3ﾋﾟﾝを構成設定することです。これはLUT0CTRLAﾚｼﾞｽﾀのOUTENﾋﾞｯﾄを設定(1)することによって行われます。

図4-5. LUTn制御A(LUTnCTRLA)ﾚｼﾞｽﾀ

ﾋﾞｯﾄ	7	6	5	4	3	2	1	0
	EDGEDET	OUTEN	FILTSEL1,0		CLKSRC2~0			ENABLE
ｱｸｾｽ種別	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ﾘｾｯﾄ値	0	0	0	0	0	0	0	0

これは以下のｺｰﾄﾞに変換できます。

```
CCL.LUT0CTRLA = CCL_OUTEN_bm;
```

入出力ﾋﾟﾝでLUTn出力を許可することにより、対応するﾋﾟﾝに対する設定が上書きされます。準備を完了して入力系列の解読を開始するにはCCLと使われるLUTが許可されることが必要です。それはLUTnCTRLAﾚｼﾞｽﾀのENABLEﾋﾞｯﾄを使って行われます

```
CCL.LUT1CTRLA = CCL_ENABLE_bm;
CCL.LUT0CTRLA |= CCL_ENABLE_bm;
```

© 2019 Microchip Technology Inc.

準備を完了するにはCTRLAレジスタのCCL全体許可ビットを使ってCCL単位部も許可されるべきです。

図4-6. CCL制御A(CTRLA)レジスタ

ビット	7	6	5	4	3	2	1	0
		RUNSTDBY						ENABLE
アクセス種別	R	R/W	R	R	R	R	R	R/W
リセット値	0	0	0	0	0	0	0	0

```
CCL.CTRLA = CCL_ENABLE_bm;
```

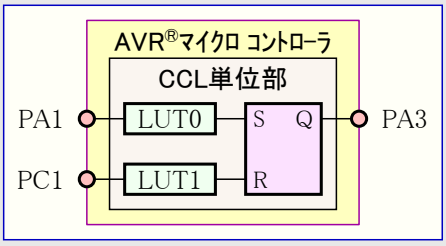
GitHubでコード例を見てください。
貯蔵庫を閲覧するにはクリックしてください。

助言: 完全なコード例は「追補」章でも利用可能です。

5. SRラッチ

本章はSRラッチを実装するのにCCLの組み合わせと順次論理回路を使う応用例を記述します。この機能は順次論理回路部を通して接続された2つの隣接LUT(LUT0とLUT1)を使って作成することができます。

図5-1. SRラッチ実装にCCLを使用



セットとリセットの信号についてはLUT用入力として(入出力ポートのPA1とPC1のピン)の2つのピンが使われます。これは以下のコードに変換できます。

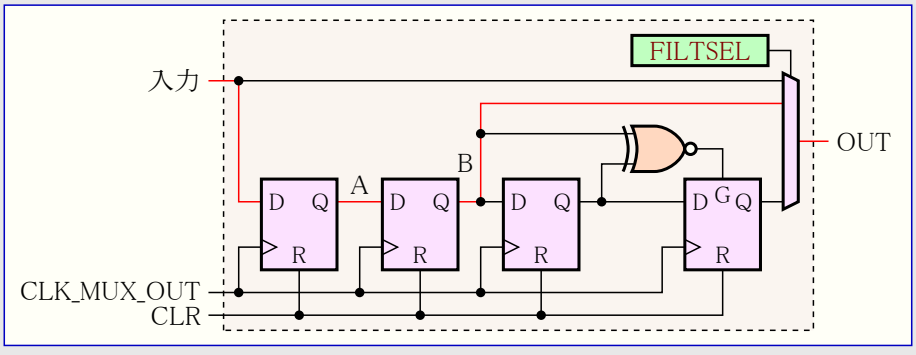
```
CCL.LUT0CTRLB = CCL_INSEL0_MASK_gc | CCL_INSEL1_IO_gc;
CCL.LUT0CTRLC = CCL_INSEL2_MASK_gc;
CCL.LUT1CTRLB = CCL_INSEL0_MASK_gc | CCL_INSEL1_IO_gc;
CCL.LUT1CTRLC = CCL_INSEL2_MASK_gc;
```

この場合、入力信号に選ばれた入力だけが各LUTに対して真理値表を構成設定する時に考慮されることが必要です。例えば、信号がHigh活性でLUTn_IN1で利用可能な場合、真理値表は0x02に設定されるべきです。多くの評価キットに対する場合である入力信号がLow活性の場合、真理値表は0x01に設定されるべきです。選んだ例について、入力信号はLow活性で、故に真理値表は両LUTに対して0x01に設定されます。(訳注:脚注参照)

```
CCL.TRUTH0 = 0x01;
CCL.TRUTH1 = 0x01;
```

真理値表出力は入力の組み合わせ関数です。これは入力が値を変える時に或る短い誤信号を発生するかもしれません。これらの誤信号はどんな問題も起こさないかもしれませんが、LUT出力が計時器や同類での入力として使われる事象を起動するように設定される場合、望まれない誤信号が望まれない事象と周辺機能活動を起動するかもしれません。濾波部を通すクロック駆動によってこれらの誤信号除去で、使用者は意図した出力だけを得ます。CCLの各参照表(LUT)はLUT出力を同期するまたは濾波するのに使うことができる濾波部を含みます。

図5-2. CCL濾波部



(訳注) 原書でのLow活性の真理値表=$00は誤っており、その場合のLUT出力は常に0で無意味です。Low活性にする場合は内部が正論理なのでLUTで反転器(NOT)を構成しなければならず、本例の場合はIN2とIN0が遮蔽で常に0なので取り得る入力値と出力値の関係は'000'⇒1と'010'⇒0になり、故に真理値表値はビット0だけが1である$01でなければなりません。

濾波部任意選択の選択はLUTnCTRLAレジスタのFILTSEL1,0ビットを使うことによって行われます。

図5-3. CCL濾波部任意選択

値	00	01	10	11
名称	DISABLE	SYNCH	FILTER	-
説明	濾波器禁止	同期化器許可	濾波器許可	(予約)

```
CCL.LUT0CTRLA = CCL_FILTSEL_FILTER_gc;
CCL.LUT1CTRLA = CCL_FILTSEL_FILTER_gc;
```

次の段階はSRラッチ機能を作成するために順次論理回路を通してLUTに接続することです。順次制御0(SEQCTRL0)レジスタのSEQSELn3～0ビットはLUT1とLUT0に対する順次構成設定を選びます。

図5-4. 順次制御0(SEQCTRL0)レジスタ

ビット	7	6	5	4	3	2	1	0
	SEQSEL13~0				SEQSEL03~0			
アクセス種別	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
リセット値	0	0	0	0	0	0	0	0

● **ビット7～4,3～0 – SEQSELn3~0 : 順次回路選択** (Sequential Selection)

SEQSELnのビットはLUT2n+1とLUT2nに対する順次(論理回路)構成設定を選びます。

値	000	001	010	011	100	101	110	111
名称	DISABLE	DFF	JK	DLATCH	RS	-		
説明	順次論理回路禁止	Dフリップフロップ	JKフリップフロップ	Dラッチ	RSラッチ	(予約)		

これは以下のコードに変換できます。

```
CCL.SEQCTRL0 = CCL_SEQSEL0_RS_gc;
```

準備を完了してLUT0 OUT(PA3)ピンでのLUT0出力を許可するには使ったLUTとCCLの許可が必要です。

```
CCL.LUT1CTRLA |= CCL_ENABLE_bm;
CCL.LUT0CTRLA |= CCL_ENABLE_bm | CCL_OUTEN_bm;
CCL.CTRLA = CCL_ENABLE_bm;
```

GitHubでコード例を見てください。
貯蔵庫を閲覧するにはクリックしてください。

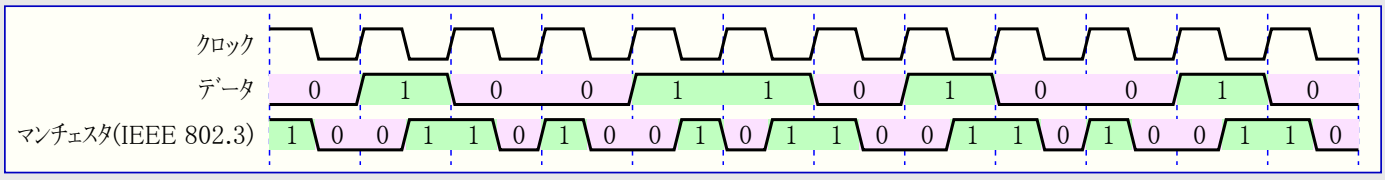
助言: 完全なコード例は「**追補**」章でも利用可能です。

6. マンチェスタ符号器

本章ではマンチェスタ符号化部を構築してマンチェスタ符号化した信号を送るのにCCLとSPIが使われます。これはSPIに入力データを設定するために非常に僅かなCPU周期を必要とし、作業の残りはCCLとSPIの単位部によって実行されます。

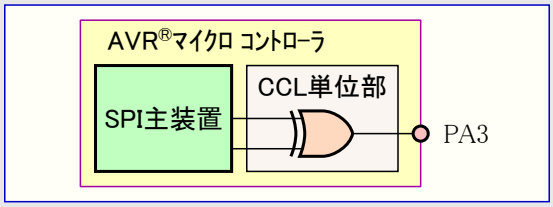
マンチェスタ符号は一定のDC成分を持つ伝送符号形式です。マンチェスタ符号には2つの版があり、この資料はIEEE 802.3規格で定義された版を元にします。マンチェスタ符号はデータとクロックの1つのクロック周期がマンチェスタ ビット周期である単一信号に結合します。常に遷移がビット周期の中央で起きます。データ=0はビット周期の中央での下降端(HighからLowへの遷移)によって表され、データ=1はビット周期の中央での上昇端(LowからHighへの遷移)によって表されます。例が下に示されます。

図6-1. マンチェスタ符号化データ



マンチェスタ符号化したデータを得る1つの方法はクロックとデータの間でXOR関数を使うことです。現在の例では符号化部に対するクロックとデータを生成するのにSPI単位部が使われます。

図6-2. マンチェスタ符号化部としてCCLとSPIを使用



SPI単位部は出力で信号を生成するように主装置として構成設定されるべきです。

```
void SPI0_init()
{
    SPI0.CTRLA = SPI_ENABLE_bm | SPI_MASTER_bm;
}
```

その後にLUT0は入力としてSPIのMOSIとSCKの信号を使うように構成設定されます。使わないLUT0入力は遮蔽されます。

```
CCL.LUT0CTRLB = CCL_INSEL0_MASK_gc | CCL_INSEL1_SPI0_gc;
CCL.LUT0CTRLC = CCL_INSEL2_SPI0_gc;
```

LUT0のIN1とIN2間で論理XOR機能を作成するために真理値表の値が0x14であるべきです。

```
CCL.TRUTH0 = 0x14;
```

準備を完了してLUT0 OUT(PA3)ピンでLUT0出力を許可するにはCCLとLUT0の許可が必要です。

```
CCL.LUT0CTRLA = CCL_ENABLE_bm | CCL_OUTEN_bm;
CCL.CTRLA = CCL_ENABLE_bm;
```

GitHubでコード例を見てください。
貯蔵庫を閲覧するにはクリックしてください。

助言: 完全なコード例は「**追補**」章でも利用可能です。

7. 参照

1. ATmega4809製品頁：https://www.microchip.com/wwwproducts/en/ATMEGA4809
2. megaAVR® 0系手引書 (DS40002015)
3. AVR®のコアから独立した周辺機能での開始に際して (DS40002451)
4. ATmega4809 Xplained Proウェブ頁：https://www.microchip.com/developmenttools/ProductDetails/atmega4809-xpro

8. 追補

例8-1. 論理ANDゲート コード例

```
#include <avr/io.h>

void PORT0_init (void);
void CCL0_init(void);

/**
 * ¥brief ポート初期化
 */
void PORT0_init (void)
{
    PORTC.DIR &= ~PIN0_bm;                    // PC0 - LUT1 IN0
    PORTC.DIR &= ~PIN1_bm;                    // PC1 - LUT1 IN1
    PORTC.DIR &= ~PIN2_bm;                    // PC2 - LUT1 IN2

    PORTC.DIR |= PIN3_bm;                     // PC3 - LUT1出力
}

/**
 * ¥brief CCL周辺機能初期化
```

例8-1 (続き). 論理ANDｹﾞｰﾄ ｺｰﾄﾞ例

```
 */
void CCL0_init(void)
{
    // 使うLUT入力を構成設定
    CCL.LUT1CTRLB = CCL_INSEL0_IO_gc           /* LUT1-IN0入力元はI/Oﾋﾟﾝ */
                  | CCL_INSEL1_IO_gc;          /* LUT1-IN1入力元はI/Oﾋﾟﾝ */
    CCL.LUT1CTRLC = CCL_INSEL2_IO_gc;          /* LUT1-IN2入力元はI/Oﾋﾟﾝ */

    // 真理値表構成設定
    CCL.TRUTH1 = 0x80;                         /* 真理値表1: 128 */

    // I/OﾋﾟﾝでのLUT1出力許可
    CCL.LUT1CTRLA = CCL_OUTEN_bm;              /* 出力許可: 許可 */

    // LUT許可
    CCL.LUT1CTRLA |= CCL_ENABLE_bm;            /* LUT1許可: 許可 */

    // CCL単位部許可
    CCL.CTRLA = CCL_ENABLE_bm;                 /* 許可: 許可 */
}

int main(void)
{
    PORT0_init();
    CCL0_init();
    while (1)
    {
        ;
    }
}
```

例8-2. 状態解読器ｺｰﾄﾞ例

```
#include <avr/io.h>

void PORT0_init (void);
void CCL0_init(void);

/**
 * ¥brief ﾎﾟｰﾄ初期化
 */
void PORT0_init (void)
{
    PORTA.DIR &= ~PIN0_bm;                     // PA0 - LUT0 IN0
    PORTA.DIR &= ~PIN1_bm;                     // PA1 - LUT0 IN1
    PORTC.DIR &= ~PIN0_bm;                     // PC0 - LUT1 IN0
    PORTC.DIR &= ~PIN1_bm;                     // PC0 - LUT1 IN1
    PORTC.DIR &= ~PIN2_bm;                     // PC0 - LUT1 IN2

    PORTA.DIR |= PIN3_bm;                      // PA3 - LUT0出力
}

/**
 * ¥brief CCL周辺機能初期化
 */
void CCL0_init(void)
{
    // 使うLUT入力を構成設定
    CCL.LUT0CTRLB = CCL_INSEL0_IO_gc           /* LUT0-IN0入力元はI/Oﾋﾟﾝ */
                  | CCL_INSEL1_IO_gc;          /* LUT0-IN1入力元はI/Oﾋﾟﾝ */
```

例8-2 (続き). 状態解読器ｺｰﾄﾞ例

```
    CCL.LUT0CTRLC = CCL_INSEL2_LINK_gc;          /* LUT0-IN2入力元は連結 */

    CCL.LUT1CTRLB = CCL_INSEL0_IO_gc             /* LUT1-IN0入力元はI/Oﾋﾟﾝ */
                  | CCL_INSEL1_IO_gc;            /* LUT1-IN1入力元はI/Oﾋﾟﾝ */
    CCL.LUT1CTRLC = CCL_INSEL2_IO_gc;            /* LUT1-IN3入力元はI/Oﾋﾟﾝ */

    // 真理値表構成設定
    CCL.TRUTH0 = 0x40;                           /* 真理値表0: 64 */
    CCL.TRUTH1 = 0x20;                           /* 真理値表1: 32 */

    // I/OﾋﾟﾝでのLUT0出力許可
    CCL.LUT0CTRLA = CCL_OUTEN_bm;                /* 出力許可: 許可 */

    // LUT許可
    CCL.LUT0CTRLA |= CCL_ENABLE_bm;              /* LUT0許可: 許可 */
    CCL.LUT1CTRLA = CCL_ENABLE_bm;               /* LUT1許可: 許可 */

    // CCL単位部許可
    CCL.CTRLA = CCL_ENABLE_bm;                   /* 許可: 許可 */
}

int main(void)
{
    PORT0_init();
    CCL0_init();
    while (1)
    {
        ;
    }
}
```

例8-3. SRﾗｯﾁ ｺｰﾄﾞ例

```
#include <avr/io.h>

void PORT0_init (void);
void CCL0_init(void);

/**
 * ¥brief ﾎﾟｰﾄ初期化
 */
void PORT0_init (void)
{
    PORTA.DIR &= ~PIN1_bm;                       // PA1 - LUT0 IN1
    PORTC.DIR &= ~PIN1_bm;                       // PC1 - LUT1 IN1

    PORTA.DIR |= PIN3_bm;                        // PA3 - LUT0出力
}

/**
 * ¥brief CCL周辺機能初期化
 */
void CCL0_init(void)
{
    // 使うLUT入力を構成設定
    CCL.LUT0CTRLB = CCL_INSEL0_MASK_gc           /* LUT0-IN0入力元は遮蔽 */
                  | CCL_INSEL1_IO_gc;            /* LUT0-IN1入力元はI/Oﾋﾟﾝ */
    CCL.LUT0CTRLC = CCL_INSEL2_MASK_gc;          /* LUT0-IN2入力元は遮蔽 */

    CCL.LUT1CTRLB = CCL_INSEL0_MASK_gc           /* LUT1-IN0入力元は遮蔽 */
```

例8-3 (続き). SRラッチ コード例

```
                        | CCL_INSEL1_IO_gc;        /* LUT1-IN1入力元はI/Oピン */
    CCL.LUT1CTRLC = CCL_INSEL2_MASK_gc;        /* LUT1-IN2入力元は遮蔽 */

    // 真理値表構成設定
    CCL.TRUTH0 = 0x01;                         /* 真理値表0: 1 */
    CCL.TRUTH1 = 0x01;                         /* 真理値表1: 1 */

    // 濾波部構成設定
    CCL.LUT0CTRLA = CCL_FILTSEL_FILTER_gc;     /* 濾波部許可 */
    CCL.LUT1CTRLA = CCL_FILTSEL_FILTER_gc;     /* 濾波部許可 */

    // LUT0とLUT1用順次論理回路許可
    CCL.SEQCTRL0 = CCL_SEQSEL0_RS_gc;

    // I/OピンでのLUT0出力許可
    CCL.LUT0CTRLA |= CCL_OUTEN_bm;             /* 出力許可: 許可 */

    // LUT許可
    CCL.LUT0CTRLA |= CCL_ENABLE_bm;            /* LUT0許可: 許可 */
    CCL.LUT1CTRLA |= CCL_ENABLE_bm;            /* LUT1許可: 許可 */

    // CCL単位部許可
    CCL.CTRLA = CCL_ENABLE_bm;                 /* 許可: 許可 */
}

int main(void)
{
    PORT0_init();
    CCL0_init();
    /* あなたの応用コードで置き換えてください。 */
    while (1)
    {
        ;
    }
}
```

例8-4. マンチェスタ符号器コード例

```
#include <avr/io.h>

void PORT0_init (void);
void CCL0_init(void);

void SPI0_init(void);
void slaveSelect(void);
void slaveDeselect(void);
uint8_t SPI0_exchangeData(uint8_t data);

void SPI0_init(void)
{
    SPI0.CTRLA = SPI_CLK2X_bm                  /* 倍速許可 */
                 | SPI_DORD_bm                 /* LSBが先に送信されます。 */
                 | SPI_ENABLE_bm               /* 単位部許可 */
                 | SPI_MASTER_bm               /* SPI単位部主装置動作 */
                 | SPI_PRESC_DIV16_gc;         /* システム クロック16分周 */
}

uint8_t SPI0_exchangeData(uint8_t data)
{
    SPI0.DATA = data;
```

例8-4 (続き). マンチェスタ符号器コード例

```
    while (!(SPI0.INTFLAGS & SPI_IF_bm))        /* データが交換されるまで待機 */
    {
        ;
    }
    return SPI0.DATA;
}

/**
 * ¥brief ポート初期化
 */
void PORT0_init (void)
{
    PORTA.DIR |= PIN4_bm;                       /* MOSIピン方向を出力に設定 */
    PORTA.DIR &= ~PIN5_bm;                      /* MISOピン方向を入力に設定 */
    PORTA.DIR |= PIN6_bm;                       /* SCKピン方向を出力に設定 */

    PORTA.DIR |= PIN3_bm;                       /* PA3をLUT0出力として設定 */
}

/**
 * ¥brief CCL周辺機能初期化
 */
void CCL0_init(void)
{
    // 使うLUT入力を構成設定
    CCL.LUT0CTRLB = CCL_INSEL0_MASK_gc          /* LUT0-IN0入力元は遮蔽 */
                  | CCL_INSEL1_SPI0_gc;         /* LUT0-IN1入力元はSPI0 MOSI */
    CCL.LUT0CTRLC = CCL_INSEL2_SPI0_gc;         /* LUT0-IN2入力元はSPI0 SCK */

    // 真理値表構成設定
    CCL.TRUTH0 = 0x14;                          /* 真理値表0: 20 */

    // I/OピンでのLUT0出力許可
    CCL.LUT0CTRLA = CCL_OUTEN_bm;               /* 出力許可: 許可 */

    // LUT許可
    CCL.LUT0CTRLA |= CCL_ENABLE_bm;             /* LUT0許可: 許可 */

    // CCL単位部許可
    CCL.CTRLA = CCL_ENABLE_bm;                  /* 許可: 許可 */
}

int main(void)
{
    PORT0_init();
    SPI0_init();
    CCL0_init();
    while (1)
    {
        SPI0_exchangeData(0xA5);
    }
}
```

Microchipｳｪﾌﾞ ｻｲﾄ

Microchipはhttp://www.microchip.com/で当社のｳｪﾌﾞ ｻｲﾄ経由でのｵﾝﾗｲﾝ支援を提供します。このｳｪﾌﾞ ｻｲﾄはお客様がﾌｧｲﾙや情報を容易に利用可能にする手段として使われます。お気に入りのｲﾝﾀｰﾈｯﾄ ﾌﾞﾗｳｻﾞを用いてｱｸｾｽすることができ、ｳｪﾌﾞ ｻｲﾄは以下の情報を含みます。

- **製品支援** – ﾃﾞｰﾀｼｰﾄと障害情報、応用記述と試供ﾌﾟﾛｸﾞﾗﾑ、設計資源、使用者の手引きとﾊｰﾄﾞｳｪｱ支援資料、最新ｿﾌﾄｳｪｱ配布と保管されたｿﾌﾄｳｪｱ
- **全般的な技術支援** – 良くある質問(FAQ)、技術支援要求、ｵﾝﾗｲﾝ検討ｸﾞﾙｰﾌﾟ、Microchip相談役ﾌﾟﾛｸﾞﾗﾑ員一覧
- **Microshipの事業** – 製品選択器と注文の手引き、最新Microchip報道発表、ｾﾐﾅｰとｲﾍﾞﾝﾄの一覧、Microchip営業所の一覧、代理店と代表する工場

お客様への変更通知ｻｰﾋﾞｽ

Microchipのお客様通知ｻｰﾋﾞｽはMicrochip製品を最新に保つのに役立ちます。加入者は指定した製品系統や興味のある開発ﾂｰﾙに関連する変更、更新、改訂、障害情報がある場合に必ず電子ﾒｰﾙ通知を受け取ります。

登録するにはhttp://www.microchip.com/でMicrochipのｳｪﾌﾞ ｻｲﾄをｱｸｾｽしてください。”Support”下で”Customer Change Notification”をｸﾘｯｸして登録指示に従ってください。

お客様支援

Microchip製品の使用者は以下のいくつかのﾁｬﾈﾙを通して支援を受け取ることができます。

- 代理店または販売会社
- 最寄りの営業所
- 現場応用技術者(FAE:Field Aplication Engineer)
- 技術支援

お客様は支援に関してこれらの代理店、販売会社、または現場応用技術者(FAE)に連絡を取るべきです。最寄りの営業所もお客様の手助けに利用できます。営業所と位置の一覧はこの資料の後ろに含まれます。

技術支援はhttp://www.microchip.com/supportでのｳｪﾌﾞ ｻｲﾄを通して利用できます。

Microchipﾃﾞﾊﾞｲｽ ｺｰﾄﾞ保護機能

Microchipﾃﾞﾊﾞｲｽでの以下のｺｰﾄﾞ保護機能の詳細に注意してください。

- Microchip製品はそれら特定のMicrochipﾃﾞｰﾀｼｰﾄに含まれる仕様に合致します。
- Microchipは意図した方法と通常条件下で使われる時に、その製品系統が今日の市場でその種類の最も安全な系統の1つであると考えます。
- ｺｰﾄﾞ保護機能を破るのに使われる不正でおそらく違法な方法があります。当社の知る限りこれらの方法の全てはMicrochipのﾃﾞｰﾀｼｰﾄに含まれた動作仕様外の方法でMicrochip製品を使うことが必要です。おそらく、それを行う人は知的財産の窃盗に関与しています。
- Microchipはそれらのｺｰﾄﾞの完全性について心配されているお客様と共に働きたいと思います。
- Microchipや他のどの半導体製造業者もそれらのｺｰﾄﾞの安全を保証することはできません。ｺｰﾄﾞ保護は当社が製品を”破ることができない”として保証すると言うことを意味しません。

ｺｰﾄﾞ保護は常に進化しています。Microchipは当社製品のｺｰﾄﾞ保護機能を継続的に改善することを約束します。Microchipのｺｰﾄﾞ保護機能を破る試みはﾃﾞｼﾞﾀﾙ ﾐﾚﾆｱﾑ著作権法に違反するかもしれません。そのような行為があなたのｿﾌﾄｳｪｱや他の著作物に不正なｱｸｾｽを許す場合、その法律下の救済のために訴権を持つかもしれません。

法的通知

ﾃﾞﾊﾞｲｽ応用などに関してこの刊行物に含まれる情報は皆さまの便宜のためにだけ提供され、更新によって取り換えられるかもしれません。皆さまの応用が皆さまの仕様に合致するのを保証するのは皆さまの責任です。**Microchipはその条件、品質、性能、商品性、目的適合性を含め、明示的にも黙示的にもその情報に関連して書面または表記された書面または黙示の如何なる表明や保証もしません。**Microchipはこの情報とそれの使用から生じる全責任を否認します。生命維持や安全応用でのMicrochipﾃﾞﾊﾞｲｽの使用は完全に購入者の危険性で、購入者はそのような使用に起因する全ての損害、請求、訴訟、費用からMicrochipを擁護し、補償し、免責にすることに同意します。他に言及されない限り、Microchipのどの知的財産権下でも暗黙的または違う方法で許認可は譲渡されません。

商標

Microchipの名前とﾛｺﾞ、Mcicrochipﾛｺﾞ、AnyRate、AVR、AVRﾛｺﾞ、AVR Freaks、BitCloud、chipKIT、chipKITﾛｺﾞ、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、Heldo、JukeBlox、KeeLoq、KeeLoqﾛｺﾞ、Kleer、LANCheck、LINK MD、maXStylus、maXTouch、MediaLB、megaAVR、MOST、MOSTﾛｺﾞ、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32ﾛｺﾞ、Prochip Designer、QTouch、SAM-BA、SpyNIC、SST、SSTﾛｺﾞ、SuperFlash、tinyAVR、UNI/O、XMEGAは米国と他の国に於けるMicrochip Technology Incorporatedの登録商標です。

ClockWorks、The Embedded Control Solutions Company、EtherSynch、Hyper Speed Control、HyperLight Load、IntelliMOS、mTouch、Precision Edge、Quiet-Wireは米国に於けるMicrochip Technology Incorporatedの登録商標です。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BodyCom、CodeGuard、CryptoAuthentication、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNetﾛｺﾞ、memBrain、Mindi、MiWi、motorBench、MPASM、MPF、MPLAB Certifiedﾛｺﾞ、MPLAB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REAL ICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、View Sense、WiperLock、Wireless DNA、ZENAは米国と他の国に於けるMicrochip Technology Incorporatedの商標です。

SQTPは米国に於けるMicrochip Technology Incorporatedの役務標章です。

Silicon Storage Technologyは他の国に於けるMicrochip Technology Inc.の登録商標です。

GestICは他の国に於けるMicrochip Technology Inc.の子会社であるMicrochip Technology Germany II GmbH & Co. KGの登録商標です。

ここで言及した以外の全ての商標はそれら各々の会社の所有物です。

© 2019年、Microchip Technology Incorporated、米国印刷、不許複製

DNVによって認証された品質管理ｼｽﾃﾑ

ISO/TS 16949

Microchipはその世界的な本社、ｱﾘｿﾞﾅ州のﾁｬﾝﾄﾞﾗｰとﾃﾝﾍﾟ、ｵﾚｺﾞﾝ州ｸﾞﾗｼｬﾑの設計とｳｪﾊｰ製造設備とｶﾘﾌｫﾙﾆｱとｲﾝﾄﾞの設計ｾﾝﾀｰに対してISO/TS-16949:2009認証を取得しました。当社の品質ｼｽﾃﾑの処理と手続きはPIC® MCUとdsPIC® DSC、KEELOQ符号飛び回りﾃﾞﾊﾞｲｽ、直列EEPROM、ﾏｲｸﾛ周辺機能、不揮発性ﾒﾓﾘ、ｱﾅﾛｸﾞ製品用です。加えて、開発ｼｽﾃﾑの設計と製造のためのMicrochipの品質ｼｽﾃﾑはISO 9001:2000認証取得です。

日本語© *HERO* 2020.

本技術概説はMicrochipのTB3218技術概説(DS90003218A-2019年1月)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意訳されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には( )内に英語表記や略称などを残す形で表記しています。

青字の部分はﾘﾝｸとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。

世界的な販売とサービス

米国	亜細亜/太平洋	亜細亜/太平洋	欧州
本社 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 技術支援: http://www.microchip.com/support ｳｪﾌﾞｱﾄﾞﾚｽ: www.microchip.com **ｱﾄﾗﾝﾀ** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **ｵｰｽﾁﾝ** TX Tel: 512-257-3370 **ﾎﾞｽﾄﾝ** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **ｼｶｺﾞ** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **ﾀﾞﾗｽ** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **ﾃﾞﾄﾛｲﾄ** Novi, MI Tel: 248-848-4000 **ﾋｭｰｽﾄﾝ** TX Tel: 281-894-5983 **ｲﾝﾃﾞｱﾅﾎﾟﾘｽ** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **ﾛｻﾝｾﾞﾙｽ** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **ﾛｰﾘｰ** NC Tel: 919-844-7510 **ﾆｭｰﾖｰｸ** NY Tel: 631-435-6000 **ｻﾝﾎｾ** CA Tel: 408-735-9110 Tel: 408-436-4270 **ｶﾅﾀﾞ - ﾄﾛﾝﾄ** Tel: 905-695-1980 Fax: 905-695-2078	**ｵｰｽﾄﾗﾘｱ - ｼﾄﾞﾆｰ** Tel: 61-2-9868-6733 **中国 - 北京** Tel: 86-10-8569-7000 **中国 - 成都** Tel: 86-28-8665-5511 **中国 - 重慶** Tel: 86-23-8980-9588 **中国 - 東莞** Tel: 86-769-8702-9880 **中国 - 広州** Tel: 86-20-8755-8029 **中国 - 杭州** Tel: 86-571-8792-8115 **中国 - 香港特別行政区** Tel: 852-2943-5100 **中国 - 南京** Tel: 86-25-8473-2460 **中国 - 青島** Tel: 86-532-8502-7355 **中国 - 上海** Tel: 86-21-3326-8000 **中国 - 瀋陽** Tel: 86-24-2334-2829 **中国 - 深圳** Tel: 86-755-8864-2200 **中国 - 蘇州** Tel: 86-186-6233-1526 **中国 - 武漢** Tel: 86-27-5980-5300 **中国 - 西安** Tel: 86-29-8833-7252 **中国 - 廈門** Tel: 86-592-2388138 **中国 - 珠海** Tel: 86-756-3210040	**ｲﾝﾄﾞ - ﾊﾝｶﾞﾛｰﾙ** Tel: 91-80-3090-4444 **ｲﾝﾄﾞ - ﾆｭｰﾃﾞﾘｰ** Tel: 91-11-4160-8631 **ｲﾝﾄﾞ - ﾌﾟﾈｰ** Tel: 91-20-4121-0141 **日本 - 大阪** Tel: 81-6-6152-7160 **日本 - 東京** Tel: 81-3-6880-3770 **韓国 - 大邱** Tel: 82-53-744-4301 **韓国 - ｿｳﾙ** Tel: 82-2-554-7200 **ﾏﾚｰｼｱ - ｸｱﾗﾙﾝﾌﾟｰﾙ** Tel: 60-3-7651-7906 **ﾏﾚｰｼｱ - ﾍﾟﾅﾝ** Tel: 60-4-227-8870 **ﾌｨﾘﾋﾟﾝ - ﾏﾆﾗ** Tel: 63-2-634-9065 **ｼﾝｶﾞﾎﾟｰﾙ** Tel: 65-6334-8870 **台湾 - 新竹** Tel: 886-3-577-8366 **台湾 - 高雄** Tel: 886-7-213-7830 **台湾 - 台北** Tel: 886-2-2508-8600 **ﾀｲ - ﾊﾞﾝｺｸ** Tel: 66-2-694-1351 **ﾍﾞﾄﾅﾑ - ﾎｰﾁﾐﾝ** Tel: 84-28-5448-2100	**ｵｰｽﾄﾘｱ - ｳﾞｪﾙｽ** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **ﾃﾞﾝﾏｰｸ - ｺﾍﾟﾝﾊｰｹﾞﾝ** Tel: 45-4450-2828 Fax: 45-4485-2829 **ﾌｨﾝﾗﾝﾄﾞ - ｴｽﾎﾟｰ** Tel: 358-9-4520-820 **ﾌﾗﾝｽ - ﾊﾟﾘ** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **ﾄﾞｲﾂ - ｶﾞﾙﾋﾝｸﾞ** Tel: 49-8931-9700 **ﾄﾞｲﾂ - ﾊｰﾝ** Tel: 49-2129-3766400 **ﾄﾞｲﾂ - ﾊｲﾙﾌﾞﾛﾝ** Tel: 49-7131-67-3636 **ﾄﾞｲﾂ - ｶｰﾙｽﾙｰｴ** Tel: 49-721-625370 **ﾄﾞｲﾂ - ﾐｭﾝﾍﾝ** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **ﾄﾞｲﾂ - ﾛｰｾﾞﾝﾊｲﾑ** Tel: 49-8031-354-560 **ｲｽﾗｴﾙ - ﾗｰﾅﾅ** Tel: 972-9-744-7705 **ｲﾀﾘｱ - ﾐﾗﾉ** Tel: 39-0331-742611 Fax: 39-0331-466781 **ｲﾀﾘｱ - ﾊﾟﾄﾞﾊﾞ** Tel: 39-049-7625286 **ｵﾗﾝﾀﾞ - ﾃﾞﾙｰﾈﾝ** Tel: 31-416-690399 Fax: 31-416-690340 **ﾉﾙｳｪｰ - ﾄﾛﾝﾊｲﾑ** Tel: 47-72884388 **ﾎﾟｰﾗﾝﾄﾞ - ﾜﾙｼｬﾜ** Tel: 48-22-3325737 **ﾙｰﾏﾆｱ - ﾌﾞｶﾚｽﾄ** Tel: 40-21-407-87-50 **ｽﾍﾟｲﾝ - ﾏﾄﾞﾘｰﾄﾞ** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **ｽｳｪｰﾃﾞﾝ - ｲｪｰﾃﾎﾞﾘ** Tel: 46-31-704-60-40 **ｽｳｪｰﾃﾞﾝ - ｽﾄｯｸﾎﾙﾑ** Tel: 46-8-5090-4654 **ｲｷﾞﾘｽ - ｳｫｰｷﾝｶﾞﾑ** Tel: 44-118-921-5800 Fax: 44-118-921-5820