

お客様各位

カタログ等資料中の旧社名の扱いについて

2010年4月1日を以ってNECエレクトロニクス株式会社及び株式会社ルネサステクノロジが合併し、両社の全ての事業が当社に承継されております。従いまして、本資料中には旧社名での表記が残っておりますが、当社の資料として有効ですので、ご理解の程宜しくお願ひ申し上げます。

ルネサスエレクトロニクス ホームページ (<http://www.renesas.com>)

2010年4月1日

ルネサスエレクトロニクス株式会社

【発行】ルネサスエレクトロニクス株式会社 (<http://www.renesas.com>)

【問い合わせ先】 <http://japan.renesas.com/inquiry>

ご注意書き

1. 本資料に記載されている内容は本資料発行時点のものであり、予告なく変更することがあります。当社製品のご購入およびご使用にあたりましては、事前に当社営業窓口で最新の情報をご確認いただきますとともに、当社ホームページなどを通じて公開される情報に常にご注意ください。
2. 本資料に記載された当社製品および技術情報の使用に関連し発生した第三者の特許権、著作権その他の知的財産権の侵害等に関し、当社は、一切その責任を負いません。当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
3. 当社製品を改造、改変、複製等しないでください。
4. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。お客様の機器の設計において、回路、ソフトウェアおよびこれらに関連する情報を使用する場合には、お客様の責任において行ってください。これらの使用に起因しお客様または第三者に生じた損害に関し、当社は、一切その責任を負いません。
5. 輸出に際しては、「外国為替及び外国貿易法」その他輸出関連法令を遵守し、かかる法令の定めるところにより必要な手続を行ってください。本資料に記載されている当社製品および技術を大量破壊兵器の開発等の目的、軍事利用の目的その他軍事用途の目的で使用しないでください。また、当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器に使用することができません。
6. 本資料に記載されている情報は、正確を期すため慎重に作成したのですが、誤りがないことを保証するものではありません。万一、本資料に記載されている情報の誤りに起因する損害がお客様に生じた場合においても、当社は、一切その責任を負いません。
7. 当社は、当社製品の品質水準を「標準水準」、「高品質水準」および「特定水準」に分類しております。また、各品質水準は、以下に示す用途に製品が使われることを意図しておりますので、当社製品の品質水準をご確認ください。お客様は、当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途に当社製品を使用することができません。また、お客様は、当社の文書による事前の承諾を得ることなく、意図されていない用途に当社製品を使用することができません。当社の文書による事前の承諾を得ることなく、「特定水準」に分類された用途または意図されていない用途に当社製品を使用したことによりお客様または第三者に生じた損害等に関し、当社は、一切その責任を負いません。なお、当社製品のデータ・シート、データ・ブック等の資料で特に品質水準の表示がない場合は、標準水準製品であることを表します。
標準水準： コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット
高品質水準： 輸送機器（自動車、電車、船舶等）、交通用信号機器、防災・防犯装置、各種安全装置、生命維持を目的として設計されていない医療機器（厚生労働省定義の管理医療機器に相当）
特定水準： 航空機器、航空宇宙機器、海底中継機器、原子力制御システム、生命維持のための医療機器（生命維持装置、人体に埋め込み使用するもの、治療行為（患部切り出し等）を行うもの、その他直接人命に影響を与えるもの）（厚生労働省定義の高度管理医療機器に相当）またはシステム等
8. 本資料に記載された当社製品のご使用につき、特に、最大定格、動作電源電圧範囲、放熱特性、実装条件その他諸条件につきましては、当社保証範囲内でご使用ください。当社保証範囲を超えて当社製品をご使用された場合の故障および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めておりますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は耐放射線設計については行っておりません。当社製品の故障または誤動作が生じた場合も、人身事故、火災事故、社会的損害などを生じさせないようお客様の責任において冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、機器またはシステムとしての出荷保証をお願いいたします。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様が製造された最終の機器・システムとしての安全検証をお願いいたします。
10. 当社製品の環境適合性等、詳細につきましては製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。お客様がかかる法令を遵守しないことにより生じた損害に関し、当社は、一切その責任を負いません。
11. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを固くお断りいたします。
12. 本資料に関する詳細についてのお問い合わせその他お気付きの点等がございましたら当社営業窓口までご照会ください。

注 1. 本資料において使用されている「当社」とは、ルネサスエレクトロニクス株式会社およびルネサスエレクトロニクス株式会社とその総株主の議決権の過半数を直接または間接に保有する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

M16C/26A,29 グループ

CRC 演算回路

1. 要約

この資料は、M16C/26A、M16C/29 グループの CRC 演算回路の使用方法を紹介し、通信エラーを検出する方法について説明します。

2. はじめに

この資料で説明する応用例は、次のマイコンでの利用に適用されます。

- ・マイコン : M16C/26A グループ
M16C/29 グループ

M16C/26A,29 グループと同様の SFR(周辺装置制御レジスタ)を持つ他の M16C ファミリでも本プログラムを使用することができます。ただし、一部の機能を機能追加等で変更している場合がありますのでマニュアルで確認してください。このアプリケーションノートをご使用に際しては十分な評価を行ってください。

3. 応用例の説明

CRC(Cyclic Redundancy Check)は、CRC コードを用いるブロックチェックで、通信エラーの検出に使用します。M16C/26A,M16C/29 グループでは、CRC 演算回路を使用して、CRC コードを生成することができます。

3.1 CRC 演算と CRC コード

CRC 演算では通信データを数値として扱い、これをメッセージ多項式と呼びます。

CRC 演算は、メッセージ多項式を生成多項式と呼ばれる定数で除算(モジュロ 2 除算)する、演算方式です。生成多項式には次の 2 つ式のどちらかを使用することができます。

- ・ CRC-CCITT $X^{16}+X^{12}+X^5+1$
- ・ CRC-16 $X^{16}+X^{15}+X^2+1$

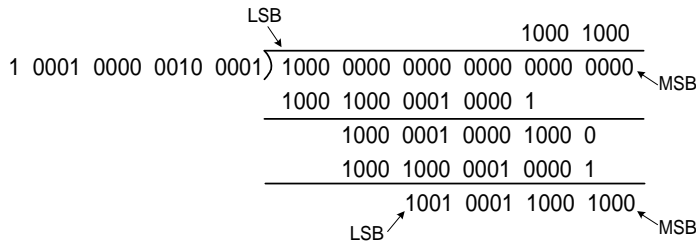
この演算の結果、得られる「余り」を CRC コードと呼びます。

LSBファーストで01Hを送受信する場合のCRC演算

生成多項式 : 1 0001 0000 0010 0001 ($X^{16}+X^{12}+X^5+1$)
 メッセージ多項式 : 1000 0000 (01HをLSBファーストで送受信)



CRCコード : 1001 0001 1000 1000 (LSBファーストなので、=1189H)



モジュロ2除算
 0+0 = 0
 0+1 = 1
 1+0 = 1
 1+1 = 0
 -1 = 1

生成多項式 : 1 1000 0000 0000 0101 ($X^{16}+X^{15}+X^2+1$)
 メッセージ多項式 : 1000 0000 (01HをLSBファーストで送受信)



CRCコード : 1000 0011 0000 0011 (LSBファーストなので、C0C1H)

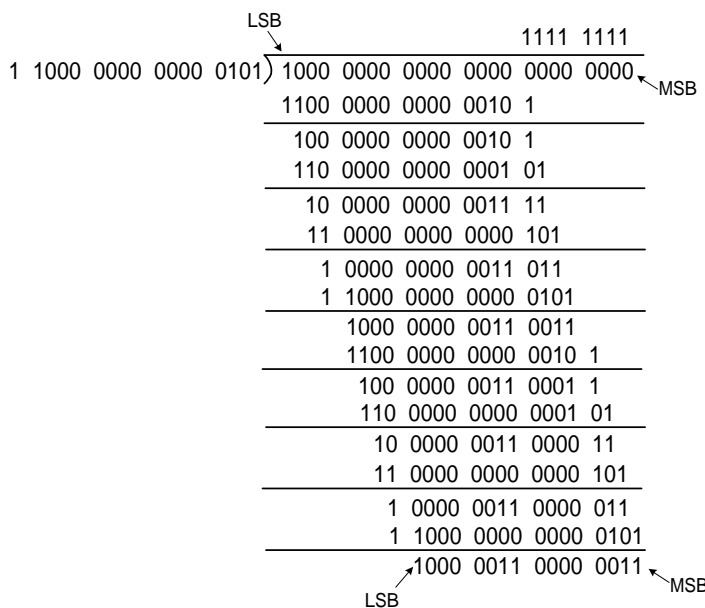


図 1 CRC 演算例

3.2 CRC 演算回路のハードウェア構成

CRC 演算の原理は除算ですが、ハードウェアではシフトレジスタと EX-OR を使用して、CRC コードを生成します。

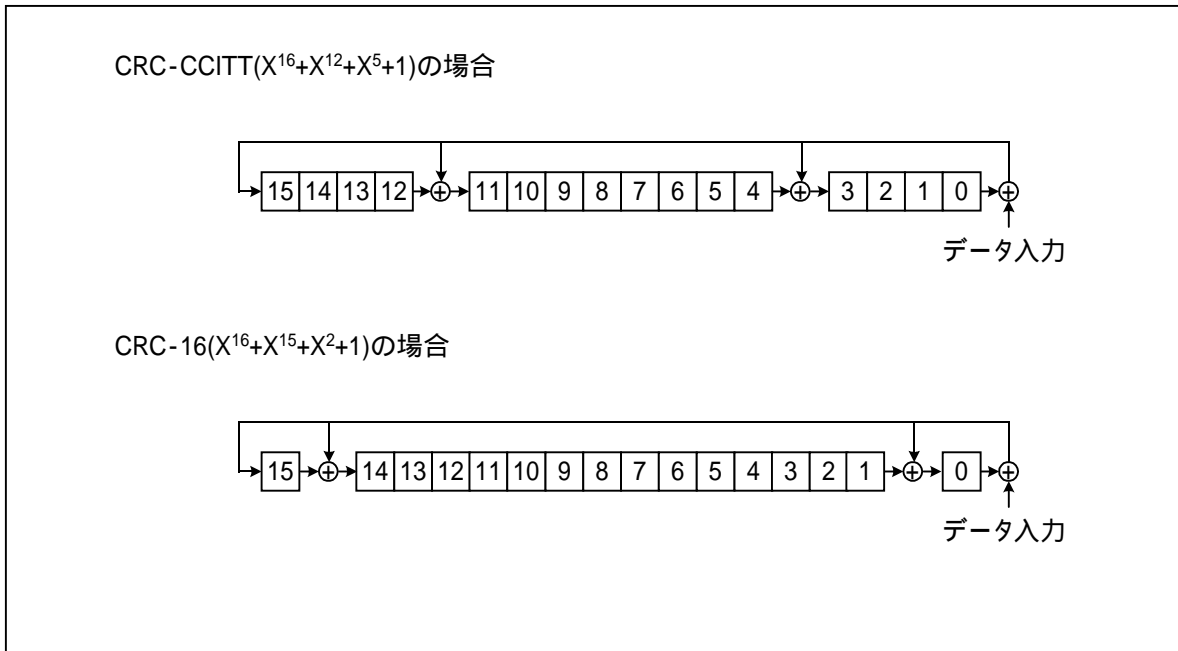


図 2 CRC 演算回路例

3.3 CRC 演算時の内部動作

M16C/26A, M16C/29 グループでは、CRC インプットレジスタに 1 バイトのデータを書き込むと、そのデータと CRC データレジスタの内容に基づいて CRC コードが生成され、CRC データレジスタに格納されます。したがって、複数バイトのデータを送受信する際、送受信データを順次、CRC インプットレジスタに書き込むと、最終的な CRC データレジスタの内容を、CRC コードとしてブロックチェックに使用することが出来ます。

また、SFR 監視アドレスレジスタに、UART の送信バッファレジスタまたは受信バッファレジスタのアドレスと、書き込み監視許可ビットまたは読み出し監視許可ビットを設定することによって、UART の送信バッファレジスタまたは受信バッファレジスタへのアクセスと並行して、送受信データを CRC インプットレジスタへ書き込む必要なく、CRC 演算が行われます。この機能は DMA 転送によって UART の送受信バッファレジスタへアクセスする場合にも、使用できます。

以下に 12345678H(001 0010 0011 0100 0101 0110 0111 1000B)を LSB ファーストで送受信する場合の CRC 演算動作を示します。

3.3.1 CRC-CCITT の場合

1. CRCデータレジスタに初期値“0000H”を書き込みます。
2. CRCインプットレジスタに最初に送受信したデータ“78H”を書き込みます。
3. CRCデータレジスタにCRCインプットレジスタのLSBより、1ビットずつ入力されて、以下に示すように演算が行われます。この結果、CRCデータレジスタの内容は“FFCFH”になります。

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LSB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
1	1	1	0	0	0	1	1	0	0	0	0	0	1	1	0	0
1	1	1	1	0	0	1	1	1	0	0	0	0	1	1	1	0
1	1	1	1	1	0	1	1	1	1	0	0	0	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1

4. CRCデータレジスタの内容を保持したままで、CRCインプットレジスタに次に送受信したデータ“56H”を書き込みます。この結果、CRCデータレジスタの内容は“09B7H”になります。

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1
1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1
1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1
0	0	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1
1	1	0	0	1	1	0	1	1	0	1	1	1	0	1	0	1
0	0	1	0	0	1	1	0	1	1	0	1	1	1	0	1	0
0	0	0	1	0	0	1	1	0	1	1	0	1	1	1	0	1
1	0	0	0	1	0	0	1	1	0	1	1	0	1	1	1	0
0	0	0	0	0	1	0	0	1	1	0	1	1	0	1	1	1

5. CRCデータレジスタの内容を保持したままで、CRCインプットレジスタに次に送受信したデータ“34H”を書き込みます。この結果、CRCデータレジスタの内容は“B69AH”になります。

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	1	0	0	1	1	0	1	1	0	1	1	1
0	1	0	0	0	0	0	0	0	1	1	0	1	0	0	1	1
0	1	1	0	0	0	1	0	0	0	1	1	0	0	0	0	1
1	0	1	1	0	0	0	1	0	0	0	1	1	0	0	0	0
0	0	0	1	1	0	0	0	1	0	0	0	1	1	0	0	0
1	1	0	0	1	1	1	0	0	1	0	0	0	0	1	0	0
1	1	1	0	0	1	0	1	0	0	1	0	0	1	0	1	0
0	0	1	1	0	0	1	0	1	0	0	1	0	0	1	0	1
0	1	0	1	1	0	1	1	0	1	0	0	1	1	0	1	0

6. CRCデータレジスタの内容を保持したままで、CRCインプットレジスタに最後に送受信したデータ“12H”を書き込みます。この結果、CRCデータレジスタの内容は“08F6H”になります。

3.3.2 CRC-16 の場合

1. CRCデータレジスタに初期値“0000H”を書き込みます。
2. CRCインプットレジスタに最初に送受信したデータ“78H”を書き込みます。
3. CRCデータレジスタにCRCインプットレジスタのLSBより、1ビットずつ入力されて、以下に示すように演算が行われます。この結果、CRCデータレジスタの内容は“2200H”になります。

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
LSB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0

4. CRCデータレジスタの内容を保持したままで、CRCインプットレジスタに次に送受信したデータ“56H”を書き込みます。この結果、CRCデータレジスタの内容は“3EA2H”になります。

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
1	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	1
1	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	0	0
1	1	0	1	1	0	1	0	1	0	0	0	1	0	0	0	1
0	1	1	1	1	1	0	1	0	1	0	0	0	1	0	0	1
1	0	1	1	1	1	1	0	1	0	1	0	0	0	1	0	0
0	0	0	1	1	1	1	1	0	1	0	1	0	0	0	1	0

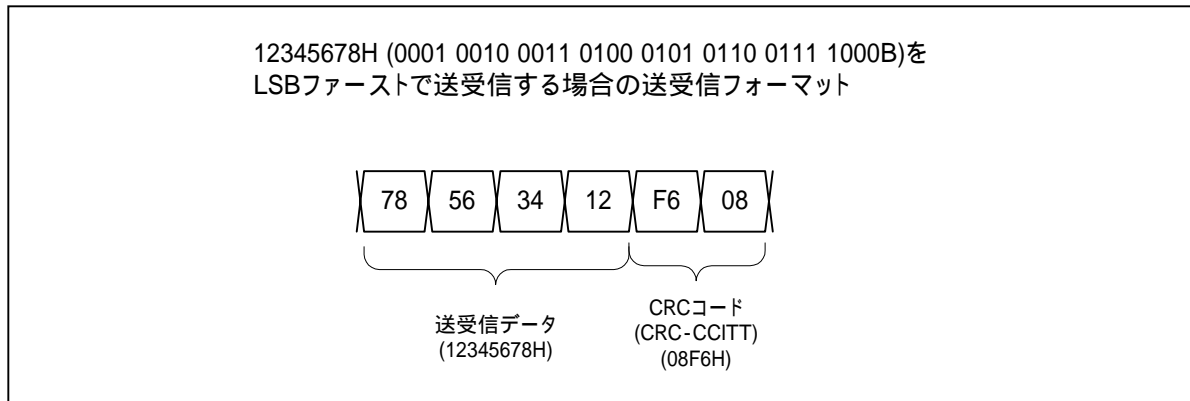
5. CRCデータレジスタの内容を保持したままで、CRCインプットレジスタに次に送受信したデータ“34H”を書き込みます。この結果、CRCデータレジスタの内容は“6EBEH”になります。

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	1	1	1	1	1	0	1	0	1	0	0	0	1	0
0	0	0	0	1	1	1	1	1	0	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1	1	1	0	1	0	1	0	0	1
0	0	1	0	1	0	1	1	1	1	1	0	1	0	1	0	0
0	0	0	1	0	1	0	1	1	1	1	1	0	1	0	1	0
1	1	0	1	1	0	1	0	1	1	1	1	1	0	1	0	0
1	1	1	1	1	1	0	1	0	1	1	1	1	1	0	1	1
0	1	1	0	1	1	1	0	1	0	1	1	1	1	1	0	0
0	0	1	1	0	1	1	1	0	1	0	1	1	1	1	1	0

6. CRCデータレジスタの内容を保持したままで、CRCインプットレジスタに最後に送受信したデータ“12H”を書き込みます。この結果、CRCデータレジスタの内容は“7D6EH”になります。

3.4 CRC の使用方法

複数バイトのデータを送受信する際、送受信データとともに、CRC コードを送受信すると通信エラーを容易に検出することができます。



3.4.1 送信側の設定手順

- (1) CRC データレジスタに、初期値“0000H”を書き込みます。CRC モードレジスタに、CRC 多項式の設定と、CRC モードを設定します。
- (2) SFR 監視アドレスレジスタに、送信バッファレジスタのアドレスをセットするとともに、書き込み監視許可ビットに“1”を設定します。
- (3) 送信データを送信バッファへ順次書き込みます。
- (4) すべてのデータを送信した後、CRC データレジスタの内容(CRC コード)を読み出し、送信バッファレジスタに書き込み送信します。

3.4.2 受信側の設定手順

- (1) CRC データレジスタに、初期値“0000H”を書き込みます。CRC モードレジスタに、CRC 多項式の設定と、CRC モードを設定します。
- (2) SFR 監視アドレスレジスタに、受信バッファレジスタのアドレスをセットするとともに、読み出し監視許可ビットに“1”を設定します。
- (3) 受信完了の度に、受信データを受信バッファレジスタから読み出します。すべての受信データを読み出した時点で、受信側の CRC データレジスタの内容と送信側の CRC レジスタの内容は同一になります。
- (4) 送信側からは CRC コードが送信されますので受信データと同様に、受信完了の度に受信バッファレジスタから読み出します。
- (5) CRC コードを含めて、すべての受信が完了した時点で、CRC データレジスタの内容が“0000H”であれば、正常受信です。CRC データレジスタの内容が“0000H”以外の値の場合は、通信エラーですので、送信側へ再送要求をしてください。

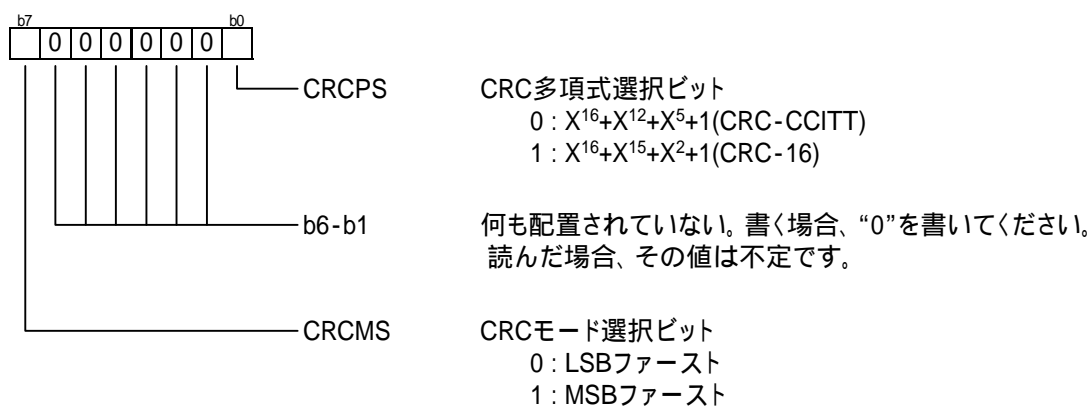
3.5 設定方法

「3. 応用例の説明」を実現するための設定手順と設定値を示します。各レジスタの詳細は M16C/26A ハードウェアマニュアル、または M16C/29 ハードウェアマニュアルを参照ください。

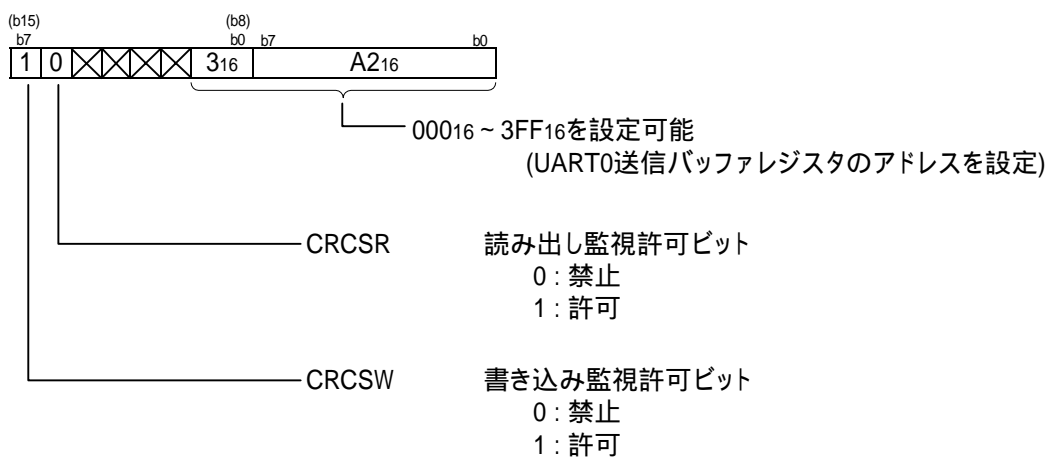
また、送受信の設定は、それぞれ関連するハードウェアマニュアル、アプリケーションノートを参照してください。

3.5.1 送信側

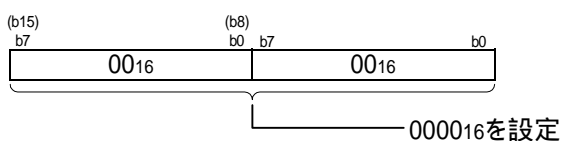
(1) CRCモードレジスタの設定



(2) SFR監視アドレスレジスタの設定

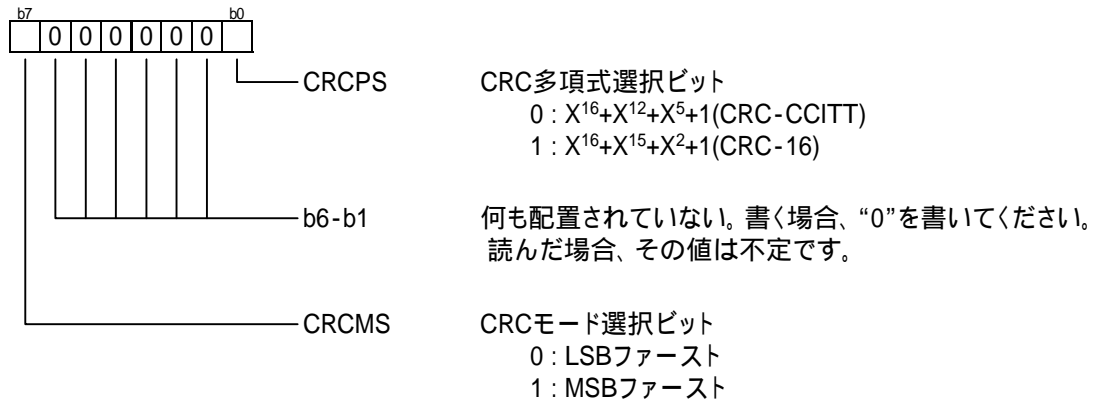


(3) CRCデータレジスタの設定

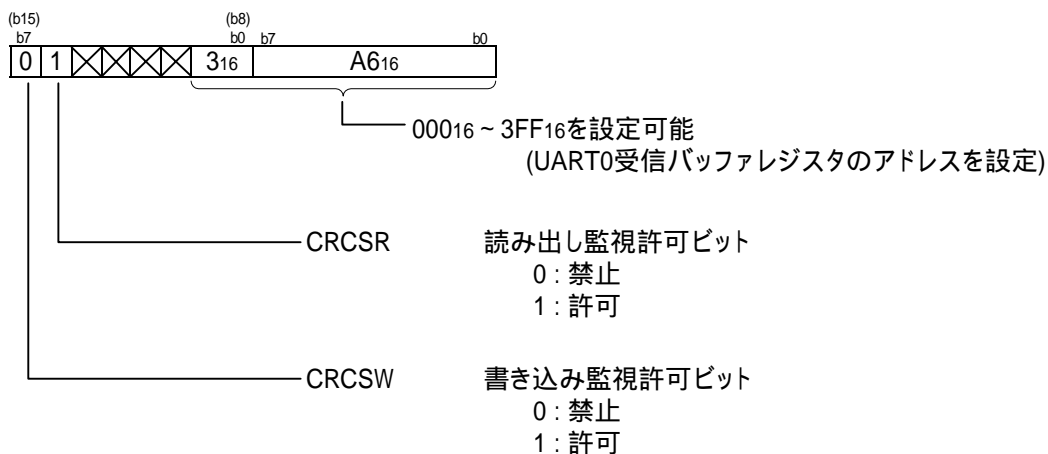


3.5.2 受信側

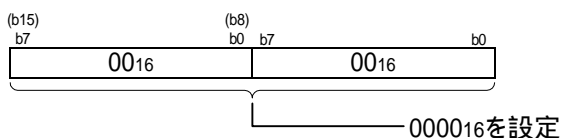
(1) CRCモードレジスタの設定



(2) SFR監視アドレスレジスタの設定



(3) CRCデータレジスタの設定



4. サンプルプログラム

クロック非同期形シリアル I/O モードで 4 バイトデータを送受信した後、通信データの CRC コードを転送する例を示します。

4.1 送信側

```

/*****
 *
 * FILE NAME :
 * CPU      : M16C/Tiny series
 * Function  : Operation of CRC
 *            (Clock asynchronous serial I/O transfer)
 *
 * Version   : 1.00
 *
 * Copyright (C)2004, Renesas Technology Corp.
 * Copyright (C)2004, Renesas Solutions Corp.
 *
 *****/
/*****
 * include file
 *****/
#include "sfr29.h"

/*****
 * main
 *****/
void main(void) {

    unsigned short   trans_data[4];
    unsigned int i;
    unsigned char    crc_data_l;
    unsigned char    crc_data_h;

    u0mr = 0x45;      /* UART0 transmit/receive mode register setting
                     UART mode transfer data 8 bits long
                     Internal clock select
                     One stop bit
                     Parity enabled (odd parity)
                     */

    u0c0 = 0x00;     /* UART0 transmit/receive control register 0 setting
                     ~CTS function select
                     ~CTS/~RTS function enabled
                     TxDO pin is CMOS output
                     Transmission data is output at falling edge of transfer
                     clock and reception data is input at rising edge
                     LSB first
                     */

    ucon = 0x01;     /* UART transmit/receive control register 2 setting
                     UART0 transmit interrupt cause is selected to "Transmission completed(TXEPT=1)"
                     ~CTS/~RTS shared pin
                     */

    u0brg = 129;     /* Setting UART0 bit rate generator (Approx 9600bps @20MHz f1) */

    u0c1 = 0x01;     /* UART transmit/receive control register 1 setting
                     Transmit enabled
                     */

    trans_data[0] = 0x78;
    trans_data[1] = 0x56;

```

```

trans_data[2] = 0x34;
trans_data[3] = 0x12;

/* CRC Calcularaion circuit */

crcmr = 0;          /*      Setting CRC mode register
                   CRC mode polynomial selection bit is selected to "0" (CRC-CCITT)
                   CRC mode selection bit is selected to "0" (LSB first)
                   */

crcsarl = (char>(&u0tb); /* Setting SFR snoop address */
crcsarh = (char)((unsigned long>(&u0tb) >> 8 );

crcsw = 1;          /*      CRC snoop on write is enabled      */
crcsr = 0;          /*      CRC snoop on read is disabled      */
crcd = 0;          /*      CRC data register set to "0"      */

for (i=0;i<4;i++) {
    u0tb = trans_data[i];          /* Writing transmit data */
    while (!ti_u0c1) { /* Check & wait the status of UART0 transmit buffer empty flag */
    }

}

crc_data_l = crcdl;
crc_data_h = crcdh;

u0tb = crc_data_l;          /* Writing transmit crc data */

while (!ti_u0c1) {          /* Check & wait the status of UART0 transmit buffer empty flag */
}

u0tb = crc_data_h;          /* Writing transmit crc data */

while (!ti_u0c1) {          /* Check & wait the status of UART0 transmit buffer empty flag */
}

}

```

4.2 受信側

```

/*****
 *
 * FILE NAME :
 * CPU      : M16C/Tiny series
 * Function  : Operation of CRC
 *             (Clock asynchronous serial I/O receive)
 *
 * Version   : 1.00
 *
 * Copyright (C)2004, Renesas Technology Corp.
 * Copyright (C)2004, Renesas Solutions Corp.
 *
 *****/
/*****
 * include file
 *****/
#include "sfr29.h"

/*****
 * main
 *****/
void main(void) {

    unsigned char    trans_data[6];
    unsigned int     i;

    u0mr = 0x45;      /* UART0 transmint/receive mode register setting
                     UART mode transfer data 8 bits long
                     Internal clokc select
                     One stop bit
                     Parity enabled (odd parity)
                     */

    u0c0 = 0x00;      /* UART0 transmint/receive control register 0 setting
                     ~CTS function select
                     ~CTS/~RTS function enabled
                     TxDO pin is CMOS output
                     Transmission data is output at falling edge of transfer
                     clock and reception data is input at rising edge
                     LSB first
                     */

    ucon = 0x01;      /* UART transmint/receive control register 2 setting
                     UART0 tansmit interrupt cause is selected to "Transmission comoleted(TXEPT=1)"
                     ~CTS/~RTS shared pin
                     */

    u0brg = 129;      /* Setting UART0 bit rate generator (Approx 9600bps @20MHz f1) */

    u0c1 = 0x04;      /* UART transmint/receive control register 1 setting
                     Receive enabled
                     */

    /* CRC Calcularaion circuit */

    crcmr = 0;        /* Setting CRC mode register
                     CRC mode polynomial selection bit is selected to "0" (CRC-CCITT)
                     CRC mode selection bit is selected to "0" (LSB first)
                     */

    crcsarl = (char)(&u0rb); /* Setting SFR snoop address */
    crcsarh = (char)((unsigned long)&u0rb >> 8 );

    crcsw = 0;        /* CRC snoop on write is disabled */
}

```

```

crcsr = 1;          /* CRC snoop on read is enabled */
crcd = 0;          /* CRC data register set to "0" */
for (i=0;i<6;i++) {
    while (!ri_u0c1) { /* Check & wait the status of UART0 receive complete flag */
    }
    trans_data[i] = u0rbl; /* Reading receive data */
}
if ( crcd == 0 ) { /* CRC data check  CRCD == 0x00 ? */
    while (1) { /* Normal end loop */
    }
} else {
    while (1) { /* Error end loop */
    }
}
}

```

5. 参考ドキュメント

ハードウェアマニュアル

M16C/26A グループハードウェアマニュアル

M16C/29 グループハードウェアマニュアル

(最新版をルネサス テクノロジホームページから入手してください。)

6. ホームページとサポート窓口

ルネサス テクノロジホームページ

<http://www.renesas.com/jpn/>

M16C ファミリ MCU 技術サポート窓口

E-mail: support_apl@renesas.com

改訂記録

Rev.	発行日	ページ	改訂内容
			ポイント
1.00	2004.12.01	-	初版発行

安全設計に関するお願い

1. 弊社は品質、信頼性の向上に努めておりますが、半導体製品は故障が発生したり、誤動作する場合があります。弊社の半導体製品の故障又は誤動作によって結果として、人身事故、火災事故、社会的損害などを生じさせないような安全性を考慮した冗長設計、延焼対策設計、誤動作防止設計などの安全設計に十分ご留意ください。

本資料ご利用に際しての留意事項

1. 本資料は、お客様が用途に応じた適切なルネサス テクノロジー製品をご購入いただくための参考資料であり、本資料中に記載の技術情報についてルネサス テクノロジーが所有する知的財産権その他の権利の実施、使用を許諾するものではありません。
2. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他応用回路例の使用に起因する損害、第三者所有の権利に対する侵害に関し、ルネサス テクノロジーは責任を負いません。
3. 本資料に記載の製品データ、図、表、プログラム、アルゴリズムその他全ての情報は本資料発行時点のものであり、ルネサス テクノロジーは、予告なしに、本資料に記載した製品または仕様を変更することがあります。ルネサス テクノロジー半導体製品のご購入に当たりましては、事前にルネサス テクノロジー、ルネサス販売または特約店へ最新の情報をご確認頂きますとともに、ルネサス テクノロジーホームページ(<http://www.renesas.com>)などを通じて公開される情報に常にご注意ください。
4. 本資料に記載した情報は、正確を期すため、慎重に制作したものです。万一本資料の記述誤りに起因する損害がお客様に生じた場合には、ルネサス テクノロジーはその責任を負いません。
5. 本資料に記載の製品データ、図、表に示す技術的な内容、プログラム及びアルゴリズムを流用する場合は、技術内容、プログラム、アルゴリズム単位で評価するだけでなく、システム全体で十分に評価し、お客様の責任において適用可否を判断してください。ルネサス テクノロジーは、適用可否に対する責任を負いません。
6. 本資料に記載された製品は、人命にかかわるような状況の下で使用される機器あるいはシステムに用いられることを目的として設計、製造されたものではありません。本資料に記載の製品を運輸、移動体用、医療用、航空宇宙用、原子力制御用、海底中継用機器あるいはシステムなど、特殊用途へのご利用をご検討の際には、ルネサス テクノロジー、ルネサス販売または特約店へご照会ください。
7. 本資料の転載、複製については、文書によるルネサス テクノロジーの事前の承諾が必要です。
8. 本資料に関し詳細についてのお問い合わせ、その他お気づきの点がございましたらルネサス テクノロジー、ルネサス販売または特約店までご照会ください。