2.17 マルチプロセッサ通信

マルチプロセッサ通信	使用機能	SCI3 ： マルチプロセッサ通信機能

仕様

(1) 図1に示すようにマルチプロセッサ通信機能を使用して、受信局AにデータH'B8を受信局BにデータH'DEをそれぞれ送信します。
(2) 送信データの通信フォーマットは、データ長が8ビット、マルチプロセッサビット1ビット、ストップビット長が1ビットに設定します。
(3) ビットレートは31250（bit/s）で送信します。またデータ送信の終了時にブレークを出力します。

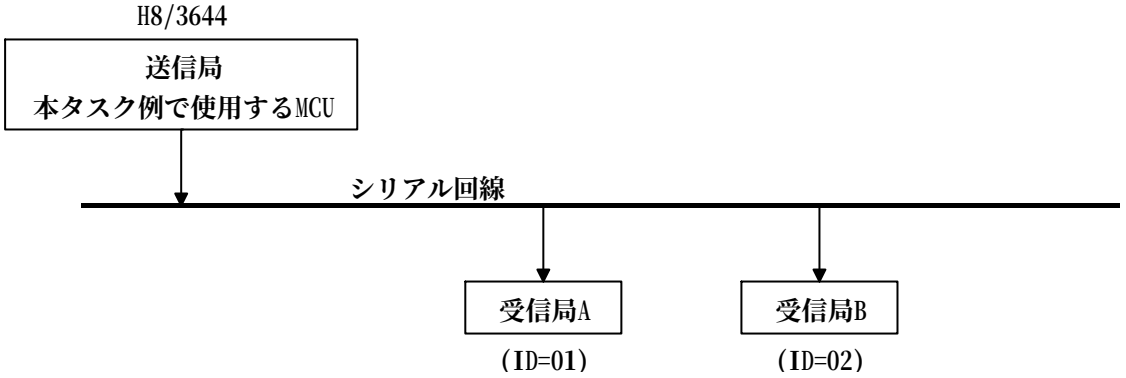


図1　マルチプロセッサ通信

使用機能説明

(1) 本タスク例では、シリアルコミュニケーションインタフェース（SCI：Serial Communication Interface）を使用して、マルチプロセッサ通信の送信を行ないます。図3にマルチプロセッサ送信のブロック図を示します。以下にマルチプロセッサ通信のブロック図について説明します。
- マルチプロセッサ通信機能とは、調歩同期式モードでマルチプロセッサビットを付加したフォーマット（マルチプロセッサフォーマット）でシリアルデータ通信を行うことにより、複数のプロセッサ間で通信回線を共有してデータの送受信を行う機能です。
- マルチプロセッサ通信を行うとき、受信局は各々固有のIDコードが割り付けられています。シリアル通信サイクルは、受信局を指定するID送信サイクルと指定された受信局へ通信データを送信するデータ送信サイクルの2つから構成されます。
- このID送信サイクルとデータ送信サイクルの区別は、マルチプロセッサビットで行います。マルチプロセッサビットが"1"のときID送信サイクル、"0"のときデータ送信サイクルとなります。
- 送信局は、まずシリアルデータ通信を行いたい受信局のIDコードに、マルチプロセッサビット"1"を付加した通信データを送信します。続いて、送信データにマルチプロセッサビット"0"を付加した通信データを送信します。
- 受信局は、マルチプロセッサビットが"1"の通信データを、自局のIDと比較し一致した場合は続いて送信される通信データを受信します。一致しなかった場合は再びマルチプロセッサビットが"1"の通信データが送信されるまで通信データを読み飛ばします。
- 送信/受信フォーマットは4種類を選択できます。マルチプロセッサフォーマットを指定した場合は、パリティビットの指定は無効です。
- 独立した送信部と受信部を備えているので、送信と受信を同時に行なうことができます。また、送信部および受信部ともにダブルバッファ構造になっているため、連続送信・連続受信ができます。
- 内蔵のボーレートジェネレータで任意のビットレートを選択可能です。
- 送受信クロックソースを内部クロック、または外部クロックから選択可能です。
- 割込み要因には送信終了、送信データエンプティ、受信データフル、オーバランエラー、フレーミングエラー、パリティエラーの6種類の割込み要因があります。
- レシーブシフトレジスタ（RSR）は、シリアルデータを受信するためのレジスタです。RSRにRXD端子から入力されたシリアルデータを、LSB（ビット0）から受信した順にセットしパラレルデータに変換します。1バイトのデータを受信すると、データは自動的にRDRへ転送されます。CPUからRSRを直接リード/ライトすることはできません。
- レシーブデータレジスタ（RDR）は、受信したシリアルデータを格納する8ビットのレジスタです。1バイトのデータの受信が終了すると、受信したデータをRSRからRDRへ転送し、受信動作を完了します。その後、RSRは受信可能となります。RSRとRDRはダブルバッファになっているため連続した受信動作が可能です。RDRは受信専用レジスタなのでCPUからライトできません。
- トランスミットシフトレジスタ（TSR）は、シリアルデータを送信するためのレジスタです。TDRから送信データをいったんTSRに転送し、LSB（ビット0）から順にTXD端子に送出することでシリアルデータ送信を行ないます。1バイトのデータを送信すると、自動的にTDRからTSRへ次の送信データを転送し、送信を開始します。ただし、TDRにデータが書き込まれていない（TDREに"1"がセットされている）場合にはTDRからTSRへのデータ転送は行ないません。CPUからTSRを直接リード/ライトすることはできません。

使用機能説明

・トランスミットデータレジスタ（TDR）は、送信データを格納する8ビットのレジスタです。TSRの"空"を検出すると、TDRに書き込まれた送信データをTSRに転送し、シリアルデータ送信を開始します。TSRのシリアルデータ送信中に、TDRに次の送信データをライトしておくと、連続送信が可能です。TDRは、常にCPUによるリード/ライトが可能です。
・シリアルモードレジスタ（SMR）は、シリアルデータ通信フォーマットの設定と、ボーレートジェネレータのクロックソースを選択するための8ビットのレジスタです。SMRは、常にCPUによるリード/ライトが可能です。
・シリアルコントロールレジスタ3（SCR3）は、送信/受信動作、調歩同期式モードでのクロック出力、割込み要求の許可/禁止、および送信/受信クロックソースの選択を行なう8ビットのレジスタです。SCR3は、常にCPUによるリード/ライトが可能です。
・図2にマルチプロセッサフォーマットを使用したシリアルデータを、表1にマルチプロセッサフォーマットを示します。

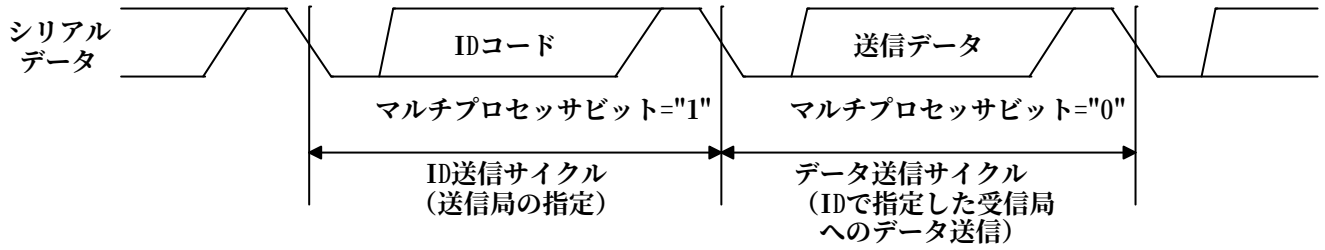


図2　マルチプロセッサビットを使用したシリアルデータ

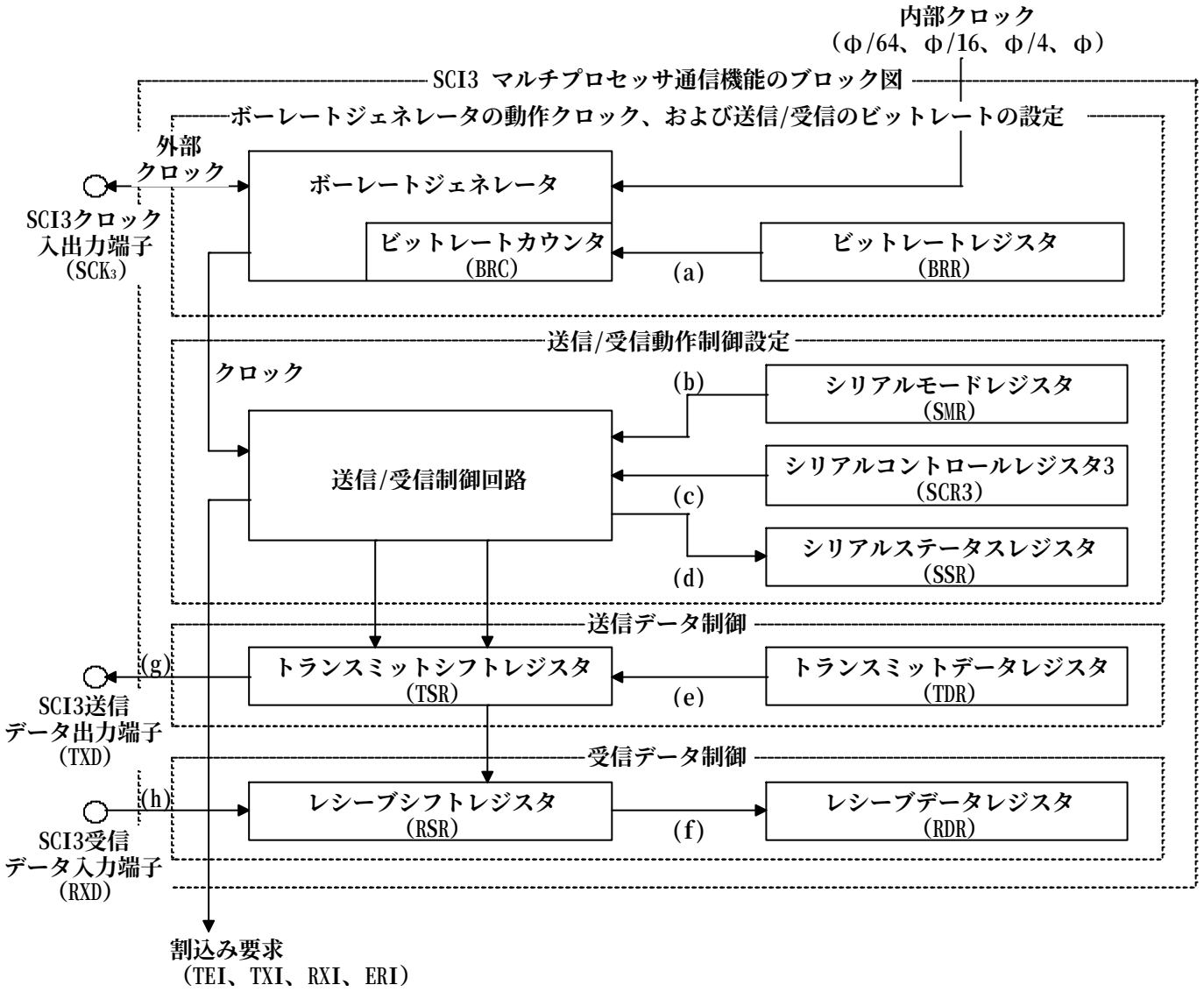
表1　マルチプロセッサフォーマット

MSR				シリアル通信フォーマットとフレーム長
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	*	1	0	S 8ビットデータ MPB STOP
0	*	1	1	S 8ビットデータ MPB STOP STOP
1	*	1	0	S 7ビットデータ MPB STOP
1	*	1	1	S 7ビットデータ MPB STOP STOP

* ： Don't care

【記号説明】
S　　： スタートビット
STOP ： ストップビット
MPB　： マルチプロセッサビット

使用機能説明



【注】(a)SMRで選択されるボーレートジェネレータの動作クロックと合わせて、送信/受信のビットレートを設定します。本タスク例では、送信のビットレートを31250（bit/s）に設定しています。
(b)シリアルデータ通信フォーマットの設定と、ボーレートジェネレータのクロックソースを選択します。本タスク例でシリアルデータ通信フォーマットは、動作モードを調歩同期式に、データ長を8ビットに、パリティビットの付加およびチェックを禁止、マルチプロセッサ通信機能を許可、ストップビット長を1ビットに、内蔵ボーレートジェネレータのクロックソースをφクロックに設定しています。
(c)送信/受信動作、調歩同期式モードでのクロック出力、割込み要求の許可/禁止を選択します。本タスク例では、調歩同期式モードでのクロック出力は、コミュニケーションモードを調歩同期式に、クロックソースを内部クロックに、SCK3端子機能をクロック出力に設定しています。また、割込み要求の許可/禁止は送信データエンプティ割込みを禁止に、受信データフル割込み要求を禁止に設定しています。
(d)ステータスフラグ（トランスミットデータレジスタエンプティ、レシーブデータレジスタフル、オーバランエラー、フレーミングエラー、パリティエラー、トランスミットエンド）によりSCI3の動作状態を示す。
(e)TSRの"空"を検出することにより、TDRに書き込まれた送信データをTSRに転送。
(f)1バイトのデータの受信が終了すると、受信したデータをRSRからRDRへ転送。
(g)送信データ。
(h)受信データ。

図3　マルチプロセッサ通信機能のブロック図

使用機能説明

・シリアルステータスレジスタ（SSR）は、SCI3の動作状態を示すステータスフラグと、マルチプロセッサビットを内蔵した8ビットのレジスタです。SSRは常にCPUからリード/ライトできます。ただし、TDRE、RDRF、OER、PER、FERへ"1"をライトすることはできません。また、これらに"0"をライトしてクリアするためには、あらかじめ"1"をリードしておく必要があります。また、TENDおよびMPBRはリード専用であり、ライトすることはできません。
・ビットレートレジスタ（BRR）は、SMRのCSK1、CKS0で選択されるボーレートジェネレータの動作クロックとあわせて、送信/受信のビットレートを設定する8ビットのレジスタです。BRRは常にCPUによるリード/ライトが可能です。
・表2に、調歩同期式モードのBRRの設定例を示します。表1はアクティブモードで、OSCが16MHzのときの値を示しています。

表2　ビットレートに対するBRRの設定例（調歩同期式モード）

Rビットレート (bit/s)	110	150	300	600	1200	2400	4800	9600	19200	31250	38400
n	3	2	2	1	1	0	0	0	0	0	0
N	70	207	103	207	103	207	103	51	25	15	12
誤差（%）	0.03	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.16	0.00	0.16

【注】 1.誤差は、1%以内となるように設定します。
2.BRRの設定値は以下の計算式で求まります。

$$N = \frac{OSC}{64 \times 2^{2n} \times B} \times 10^{6} - 1$$

B ： ビットレート（bit/s）
N ： ボーレートジェネレータのBRRの設定値（0≦N≦255）
OSC ： ϕ_{OSC}の値（MHz）= 16MHz
n ： SMRのCKS1、CKS2の設定値(0≦n≦3)　　※nとクロックの関係は表3を参照

表3　nとクロックの関係

n	クロック	SMRの設定値	
		CKS1	CKS0
0	Φ	0	0
1	Φ/4	0	1
2	Φ/16	1	0
3	Φ/64	1	1

3.表2に誤差は以下の計算式で求めた値を小数点第3位を四捨五入して表示してあります。

$$\text{誤差（%）} = \left\{ \frac{\Phi \times 10^{6}}{(N+1) \times B \times 64 \times 2^{2n-1}} - 1 \right\} \times 100$$

4.OSCが16MHzのときの最大ビットレート（調歩同期式モード）は、500000（bit/s）になります。ただし、設定値は、n=0、N=0のときです。

・調歩同期式モードは、通信開始を意味するスタートビットと通信終了を意味するストップビットとをデータに付加したキャラクタを送信/受信し、1キャラクタ単位で同期をとりながらシリアル通信を行なうモードです。
・SCI3内部では、送信部と受信部は独立しているので、全二重通信を行なうことができます。また、送信部と受信部がともにダブルバッファ構造になっているので、送信中にデータのライト、受信中にデータのリードができ、連続送信/受信が可能です。
・調歩同期式通信では、通信回線は通常マーク状態（"High"レベル）に保たれています。SCI3では通信回線を監視し、スペース（"Low"レベル）になったところをスタートビットとみなしてシリアル通信を開始します。
・通信データの1キャラクタはスタートビット（"Low"レベル）から始まり、送信/受信データ（LSBファースト：最下位ビットから）、パリティビット（"High"または"Low"レベル）、最後にストップビット（"High"レベル）の順で構成されます。
・調歩同期式モードでは、受信時にスタートビットの立ち下がりエッジで同期化を行ないます。また、データを1ビット期間の16倍の周波数のクロックの8番目でサンプリングするので、各ビットの中央で通信データを取り込みます。

使用機能説明

・SCI3クロック（SCK_3）は、SCI3のクロック入出力端子です。
・SCI3レシーブデータ入力（RXD）は、SCI3の受信データ入力端子です。
・SCI3トランスミットデータ出力（TXD）は、SCI3の送信データ出力端子です。
・SCI3の割込み要因には、送信終了、送信データエンプティ、受信データフルおよび3種類の受信エラー（オーバランエラー、フレーミングエラー、パリティエラー）の計6種類があり、共通のベクタアドレスが割り付けられています。
・各割込み要求は、SCR3のTIE、RIEで許可/禁止できます。
・SSRのTDREが"1"にセットされるとTXIが発生します。SSRのTENDが"1"にセットされると、TEIが発生します。この2つの割込みは送信時に発生します。
・SSRのTDREは初期値が"1"になっています。したがって送信データをTDRへ転送する前にSCR3のTIEを"1"にセットして送信データエンプティ割込み要求（TXI）を許可すると、送信データが準備されていなくてもTXIが発生します。
・SSRのTENDは初期値が"1"になっています。したがって、送信データをTDRへ転送する前にSCR3のTEIEを"1"にセットして送信終了割込み要求（TEI）を許可すると、送信データが送信されていなくてもTEIが発生します。
・送信データをTDRへ転送する処理を割込み処理ルーチンの中で行なうようにすることで、これらの割込みを有効に利用できます。また、これらの割込み要求（TXI、TEI）の発生を防ぐためには、送信データをTDRへ転送した後に、これらの割込み要求に対応する許可ビット（TIE、TEIE）を"1"にセットします。
・SSRのRDRFが"1"にセットされるとRXIが発生します。OER、PER、FERのいずれかが"1"にセットされるとERIが発生します。この2つの割込み要求は受信時に発生します。

(2) 表4に本タスク例の機能割付け示します。表3に示すように機能を割り付け、マルチプロセッサ通信機能による送信を行ないます。

表4　機能割付け

機能	機能割付け
TSR	シリアルデータを送信するためのレジスタ
TDR	送信データを格納するレジスタ
SMR	シリアルデータ通信フォーマット、ボーレートジェネレータのクロックソースの設定
SSR	SCI3の動作状態を示すステータスフラグ
BRR	送信/受信のビットレートを設定
PMR1	TXD出力端子設定
SCK_3	SCI3のクロック出力端子
TXD	SCI3の送信データ出力端子

マルチプロセッサ通信	使用機能	SCI3 ： マルチプロセッサ通信機能

動作原理

(1) 図4に動作原理を示します。図4に示すようなハードウェア処理、およびソフトウェア処理によりマルチプロセッサ通信機能による送信を行ないます。

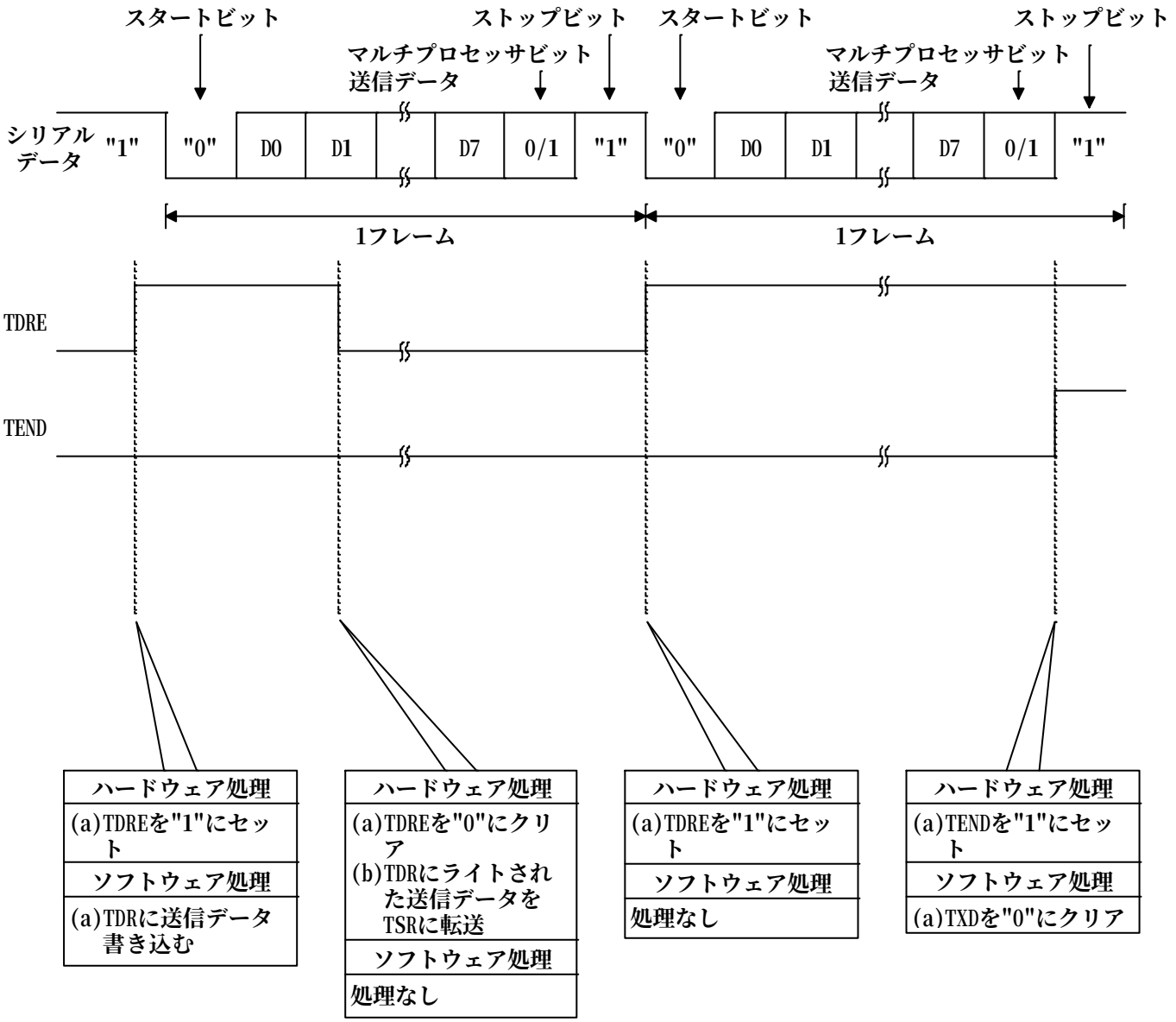


図4　マルチプロセッサ通信機能による送信の動作原理

ソフトウェア説明

(1) モジュール説明

表5に本タスク例におけるモジュール説明を示します。

表5　モジュール説明

モジュール名	ラベル名	機能
メインルーチン	main	転送データの設定、マルチプロセッサ通信機能の設定、4バイトのデータを送信したところで終了

(2) 引数の説明

表6に本タスク例で使用する引数を示します。

表6　引数の説明

引数名	機能	使用モジュール名	データ長	入出力
STD0～STD3	調歩同期式シリアル送信データ	メインルーチン	1バイト	入力

(3) 使用内部レジスタ説明

表7に本タスク例における使用内部レジスタ説明を示します。

表7　使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
SMR	COM	シリアルモードレジスタ（コミュニケーションモード） ：COM="0"のとき、コミュニケーションモードを調歩同期式モードに設定	H'FFA8 ビット7	0
	CHR	シリアルモードレジスタ（キャラクタレングス） ：CHR="0"のとき、調歩同期式モード時におけるデータ長を8ビットデータに設定	H'FFA8 ビット6	0
	PE	シリアルモードレジスタ（パリティイネーブル） ：PE="0"のとき、調歩同期式モードで、送信時にパリティビットの付加およびチェックを禁止	H'FFA8 ビット5	0
	STOP	シリアルモードレジスタ（ストップビットレングス） ：STOP="0"のとき、調歩同期式モードでのストップビットの長さを1ビットに設定	H'FFA8 ビット3	0
	MP	シリアルモードレジスタ（マルチプロセッサモード） ：MP="1"のとき、マルチプロセッサ通信機能を許可	H'FFA8 ビット2	1
	CKS1 CKS0	シリアルモードレジスタ（クロックセレクト1、0） ：CKS1="0"、CKS0="0"のとき、内蔵ボーレートジェネレータのクロックソースをφクロックに設定	H'FFA8 ビット1 ビット0	CKS1="0" CKS0="0"
BRR		ビットレートレジスタ ：BRR=H'0Fのとき、SMRのCKS1、CLS0で選択されるボーレートジェネレータの動作クロックとあわせた送信のビットレートを31250（bit/s）に設定	H'FFA9	H'0F
SCR3	TE	シリアルコントロールレジスタ3 （トランスミットイネーブル） ：TE="0"のとき、送信動作を禁止（TXD端子はトランスミットデータ端子）	H'FFAA ビット5	0
	CKE1 CKE0	シリアルコントロールレジスタ3 （クロックイネーブル1、0） ：CKE1="0"、CKE0="1"のとき、調歩同期式モードにおいてクロックソースを内部クロック、SCK_3端子機能をクロック出力に設定	H'FFAA ビット1 ビット0	CKE1="0" CKE0="1"

表7　使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
TDR		トランスミットデータレジスタ ：送信データを格納する8ビットのレジスタ	H'FFAB	-
SSR	TDRE	シリアルステータスレジスタ （トランスミットデータエンプティ） ：TDRE="0"のとき、TDRにライトされた送信データがTSRに転送されていないことを示す ：TSRE="1"のとき、TDRに送信データがライトされていな、またはTDRにライトされた送信データがTSRに転送されたことを示す	H'FFAC ビット7	1
	TEND	シリアルステータスレジスタ（トランスミットエンド） ：TEND="0"のとき、送信中であることを示す ：TEND="1"のとき、送信を終了したことを示す	H'FFAC ビット2	1
	MPBR	シリアルステータスレジスタ （マルチプロセッサビットレシーブ） ：MPBR="0"のとき、マルチプロセッサビットが"0"のデータを受信した ：MPBR="1"のとき、マルチプロセッサビットが"1"のデータを受信した	H'FFAC ビット1	0
	MPBT	シリアルステータスレジスタ （マルチプロセッサビットトランスファ） ：MPBT="0"のとき、マルチプロセッサビット"0"を送信 ：MPBT="1"のとき、マルチプロセッサビット"1"を送信	H'FFAC ビット0	0
PMR1	PMR11	ポートモードレジスタ1（$P2_2$/TXD端子機能切り替え） ：PMR11="1"のとき、$P2_2$/TXD端子をTXD端子機能に設定	H'FFE0 ビット1	1

(4) 使用RAM説明

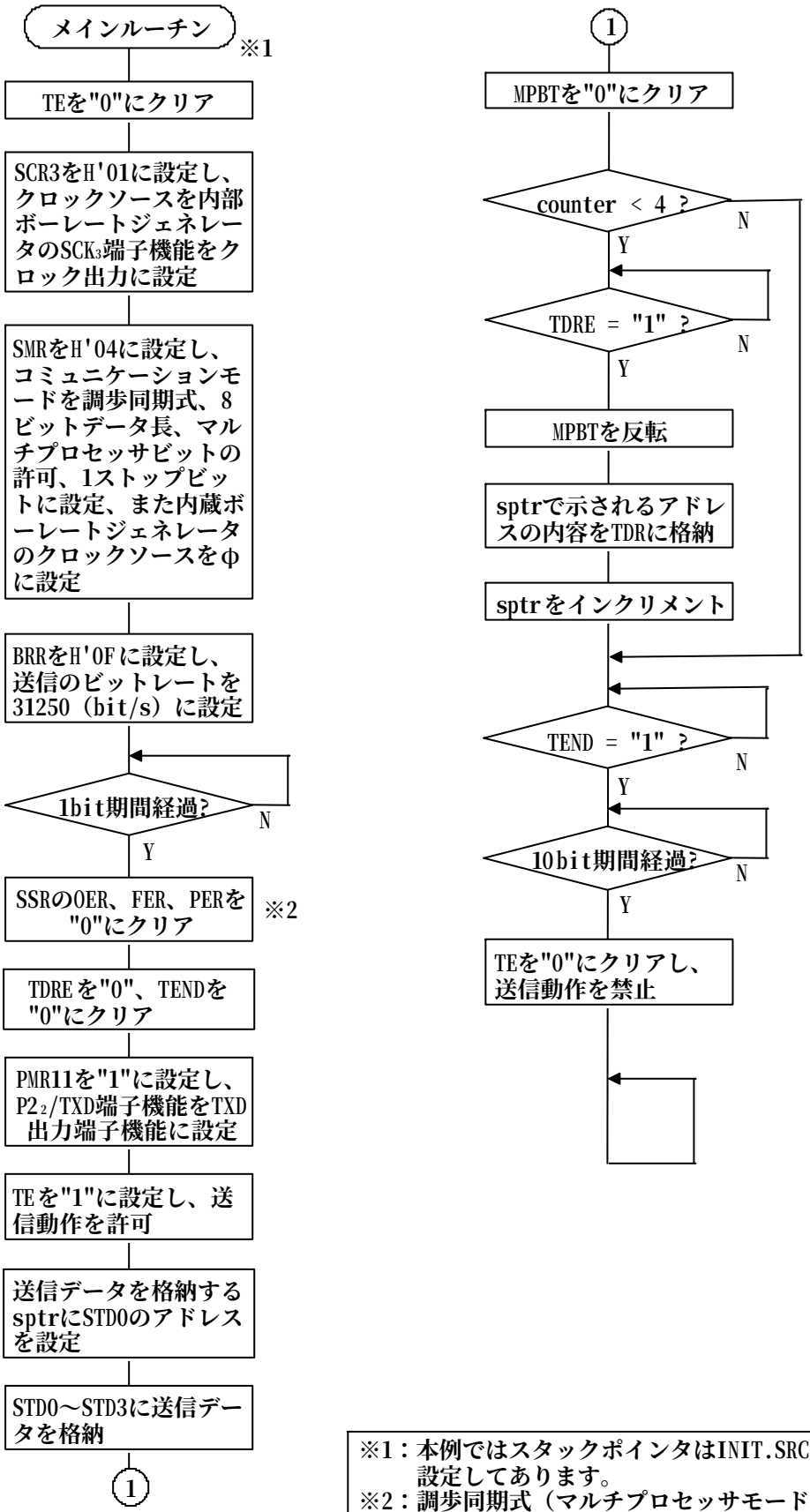
表8に本タスク例における使用RAM説明を示します。

表8　使用RAM説明

ラベル名	機能	アドレス	使用モジュール名
STD0	調歩同期式シリアル送信データの1バイト目を格納	H'FB80	メインルーチン
STD1	調歩同期式シリアル送信データの2バイト目を格納	H'FB81	メインルーチン
STD2	調歩同期式シリアル送信データの3バイト目を格納	H'FB82	メインルーチン
STD3	調歩同期式シリアル送信データの4バイト目を格納	H'FB83	メインルーチン
counter	調歩同期式シリアル送信動作を4カウントする8ビットカウンタ	H'FB84	メインルーチン

フローチャート

(a) メインルーチン



※1：本例ではスタックポインタはINIT.SRC（アセンブリ言語）で設定してあります。
※2：調歩同期式（マルチプロセッサモード）ではOER、FER、PERを"0"クリアにしておく必要がある。

プログラムリスト

INIT.SRC（プログラムリスト）

```
        .EXPORT  _INIT
        .IMPORT  _main
;
        .SECTION    P,CODE
_INIT:
        MOV.W    #H'FF80,R7
        LDC.B    #B'10000000,CCR
        JMP              @_main
;
        .END
```

```
/************************************************************/
/*                                                          */
/*  H8/300H Tiny Series -H8/3664-                           */
/*  Application Note                                        */
/*                                                          */
/*  'MultiProcessor Communications'                         */
/*                                                          */
/*  Function                                                */
/*  : Serial Communication Interface                        */
/*     -Multi-Processor Communication                       */
/*                                                          */
/*  External Clock : 16MHz                                  */
/*  Internal Clock : 16MHz                                  */
/*  Sub Clock      : 32.768kHz                              */
/*                                                          */
/************************************************************/

#include    <machine.h>

/************************************************************/
/*  Symbol Defnition                                        */
/************************************************************/
struct BIT {
    unsigned char    b7:1;      /* bit7 */
    unsigned char    b6:1;      /* bit6 */
    unsigned char    b5:1;      /* bit5 */
    unsigned char    b4:1;      /* bit4 */
    unsigned char    b3:1;      /* bit3 */
    unsigned char    b2:1;      /* bit2 */
    unsigned char    b1:1;      /* bit1 */
    unsigned char    b0:1;      /* bit0 */
};
#define    SMR      *(volatile unsigned char *)0xFFA8  /* Serial Mode Register            */
#define    SMR_BIT (*(struct BIT *)0xFFA8)             /* Serial Mode Register            */
#define    COM      SMR_BIT.b7                         /* Communication Mode              */
#define    CHR      SMR_BIT.b6                         /* Character Length                */
#define    PE       SMR_BIT.b5                         /* Parity Enable                   */
#define    PM       SMR_BIT.b4                         /* Parity Mode                     */
#define    STOP     SMR_BIT.b3                         /* Stop Bit Length                 */
#define    MP       SMR_BIT.b2                         /* Multiprocesor Mode              */
#define    CKS1     SMR_BIT.b1                         /* Clock Select 1                  */
#define    CKS0     SMR_BIT.b0                         /* Clock Select 0                  */
#define    BRR      *(volatile unsigned char *)0xFFA9  /* Bit Rate Register               */
#define    SCR3     *(volatile unsigned char *)0xFFAA  /* Serial Control Register 3       */
#define    SCR3_BIT (*(struct BIT *)0xFFAA)            /* Serial Control Register 3       */
#define    TIE      SCR3_BIT.b7                        /* Transmit Interrupt Enable       */
#define    RIE      SCR3_BIT.b6                        /* Receive Interrupt Enable        */
#define    TE       SCR3_BIT.b5                        /* Transmit Enable                 */
#define    RE       SCR3_BIT.b4                        /* Receive Enable                  */
#define    MPIE     SCR3_BIT.b3                        /* Multiprocessor Interrupt Enable */
#define    TEIE     SCR3_BIT.b2                        /* Transmit End Interrupt Enable   */
#define    CKE1     SCR3_BIT.b1                        /* Clock Enable 1                  */
#define    CKE0     SCR3_BIT.b0                        /* Clock Enable 0                  */
#define    TDR      *(volatile unsigned char *)0xFFAB  /* Transmit Data Register          */
#define    SSR      *(volatile unsigned char *)0xFFAC  /* Serial Status Register          */
#define    SSR_BIT (*(struct BIT *)0xFFAC)             /* Serial Status Register          */
```

プログラムリスト

```
#define    TDRE     SSR_BIT.b7                      /* Transmit Data Register Empty    */
#define    RDRF     SSR_BIT.b6                      /* Receive Data Register Full      */
#define    OER      SSR_BIT.b5                      /* Overrun Erorr                   */
#define    FER      SSR_BIT.b4                      /* Framing Erorr                   */
#define    PER      SSR_BIT.b3                      /* Parity Erorr                    */
#define    TEND     SSR_BIT.b2                      /* Transmit End                    */
#define    MPBR     SSR_BIT.b1                      /* Multiprocessor Bit Receive      */
#define    MPBT     SSR_BIT.b0                      /* Multiprocessor Bit Transfer     */
#define    PMR1_BIT (*(struct BIT *)0xFFE0)         /* Port Mode Register 1            */
#define    PMR11    PMR1_BIT.b1                     /* Port Mode Register 1 bit1       */
#define    RDR      *(volatile unsigned char *)0xFFAD  /* Receive data Register         */

/*****************************************************/
/*  関数定義                                         */
/*****************************************************/
extern  void    INIT( void );                       /* SP Set                  */
void    main    ( void );

/*****************************************************/
/*   RAM Allocation                                  */
/*****************************************************/
    unsigned char   STD[4];
    unsigned char   counter;

/*****************************************************/
/*  Vector Address                                   */
/*****************************************************/
#pragma section     V1                              /* VECTOR SECTOIN SET        */
void (*const VEC_TBL1[])(void) = {
/* 0x00 - 0x0f */
    INIT                                            /* 00 Reset                  */
};

#pragma section                                     /* P                         */
/*****************************************************/
/*   Main Program                                    */
/*****************************************************/
void main ( void )
{

    unsigned char   *sptr;

    TE = 0;                                         /* Clear Serial Transmitting                  */

    SCR3 = 0x01;                                    /* Initialize Serial Control Register 3       */

    SMR = 0x04;                                     /* Initialize Serial Mode Register            */

    BRR = 0x0F;                                     /* Initialize Bit Rate  Register              */

    for(counter = 0 ; counter < 1 ; counter++){     /* dummy Wait                                 */
        ;
    }

    OER = 0;                                        /* Clear OER                                  */
    FER = 0;                                        /* Clear FER                                  */
    PER = 0;                                        /* Clear PER                                  */

    TDRE = 0;                                       /* Clear TDRE                                 */
    TEND = 0;                                       /* Clear TEND                                 */

    PMR11 = 1;                                      /* Initialize Output Port TXD                 */

    TE = 1;                                         /* Start Serial Transmitting                  */

    sptr = &STD[0];                                 /* Initialize Serial Transmitting Data Address */
```

プログラムリスト

```
   STD[0] = 0x01;                          /* Set Serial Transfer Data 0                */
    STD[1] = 0xB8;                         /* Set Serial Transfer Data 1                */
    STD[2] = 0x02;                         /* Set Serial Transfer Data 2                */
    STD[3] = 0xDE;                         /* Set Serial Transfer Data 3                */

    MPBT = 0;                              /* Clear Multiprocessor Bit Transfer         */

    for(counter = 0 ; counter < 4 ; counter++){   /* Serial Transmitting Data Counter 4 Loop   */

        while(TDRE != 1){                  /* End Serial Transmitting                   */
            ;
        }

        MPBT = ~MPBT;                      /* Initialize Multiprocessor Bit Transfer    */

        TDR = *sptr;                       /* Save Serial Transmitting Data             */

        sptr++;                            /* Increment Serial Transmittimg Data Address */

    }

    while(TEND != 1){                      /* End Serial Transmitting                   */
        ;
    }

    for(counter = 0 ; counter < 10 ; counter++){  /* dummy Wait                         */
        ;
    }

    TE = 0;                                /* Initialize Transmitting Enable            */

    while(1){
       ;
    }
}
```

リンクアドレス指定

セクション名	アドレス
CV1	H'0000
P	H'0100
B	H'FB80