

## 2.1 インターバル機能による8ビットカウンタのカウントアップ

インターバル機能による 8ビットカウンタのカウントアップ	使用機能	タイマA：インターバル機能
仕様	<p>(1) タイマAのインターバル機能を内蔵した8ビットのタイマを使用して、RAMに設定した8ビットのカウンタのカウントアップを行ないます。</p> <p>(2) タイマカウンタA (TCA) のオーバーフローによりタイマA割込みを発生させ、タイマA割込み処理の中でRAMに設定したカウンタのカウントアップ、またはイニシャライズを行ないます。</p> <p>(3) RAMに設定するカウンタは、8ビットのカウンタとし、初期値のH'00からカウントアップを始めて、H'FFまでカウントを行なうと、H'00にイニシャライズして再びカウントアップを継続します。</p> <p>(4) タイマA割込みは、65.536msごとに発生するように設定します。</p>	

### 使用機能説明

- (1) 本タスク例では、タイマAインターバル機能を使用して、8ビットカウンタのカウントアップを行ないます。図1にタイマAインターバル機能のブロック図を示します。以下にタイマAインターバル機能のブロック図について説明します。
- ・システムクロック ( ) は、16MHzのOSCクロックで、CPUおよび周辺機能を動作させるための基準クロックです。
  - ・PSSは、 を入力とする13ビットのカウンタで、1サイクルごとにカウントアップします。
  - ・タイマモードレジスタA (TMA) は、8ビットのリード/ライト可能なレジスタで、プリスケアラ、入力クロックの選択を行ないます。本タスク例では、プリスケアラにPSSを、プリスケアラ分周比に4096分周を選択しています。
  - ・タイマカウンタA (TCA) は、8ビットのリード可能なアップカウンタで、入力する内部クロックによりカウントアップされます。TCAがオーバーフローすると、割込み要求レジスタ1 (IRR1) のタイマAオーバーフロー割込み要求 (IRRTA) が"1"にセットされます。
  - ・タイマAオーバーフロー割込み要求フラグ (IRRTA) は、TCAがオーバーフローすることにより"1"にセットされます。IRRTAが"1"にセットされていて、割込み許可レジスタ1 (IENR1) のタイマA割込みイネーブル (IENTA) が"1"で、かつコンディションコードレジスタ (CCR) の1ビットが"0"にクリアされている場合にタイマA割込みが受け付けられ、タイマA割込み処理を開始します。
  - ・以下に、本タスク例におけるTCAのオーバーフロー周期の計算法を示します。

$$\begin{aligned}
 \text{TCAオーバーフロー周期} &= \frac{1}{\text{システムクロック} / 4096} \times 256 \\
 &= 65.536\text{ms}
 \end{aligned}$$

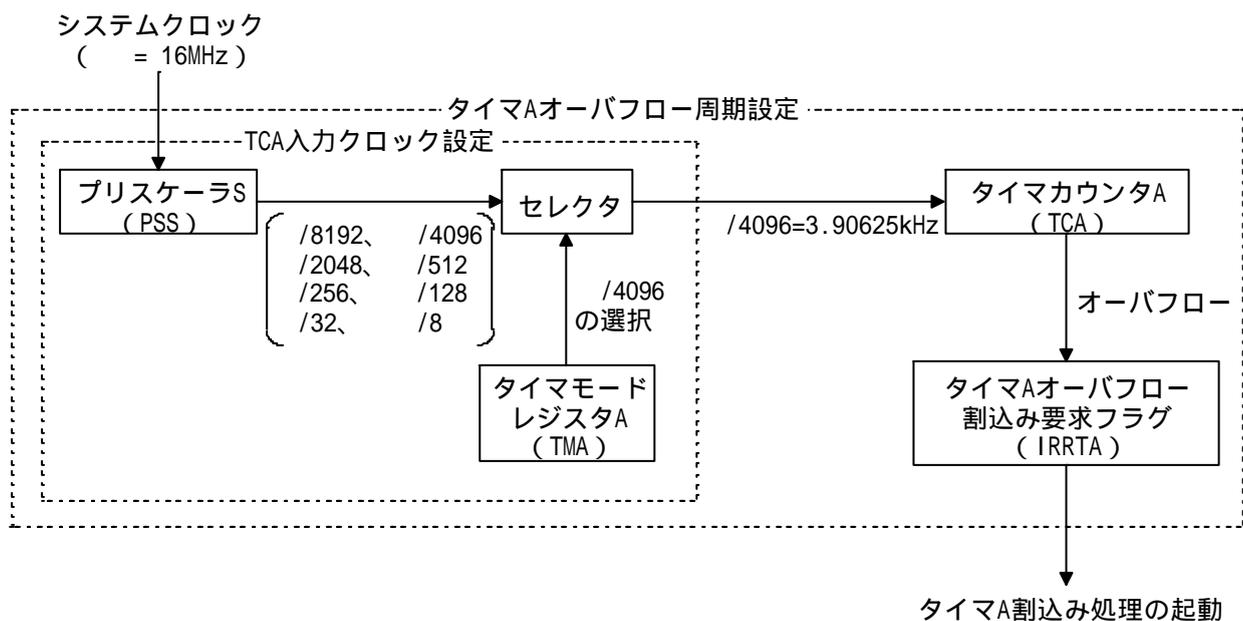


図1 タイマAインターバル機能ブロック図

**使用機能説明**

(2) 表1に本タスク例の機能割付けを示します。表1に示すように機能を割付け、タイマAインターバル機能による8ビットカウンタのカウントアップを行ないます。

表1 機能割付け

機能	機能割付け
PSS	システムクロックを入力とする13ビットのカウンタ
TCA	システムクロックを4096分周したクロックを入力とする8ビットのカウンタ
TMA	PSSの選択、およびプリスケラ分周比の設定
IRRTA	タイマA割込み要求の有無を反映

**動作原理**

(1) 図2に動作原理を示します。図2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タイマAインターバル機能による8ビットカウンタのカウントアップを行ないます。

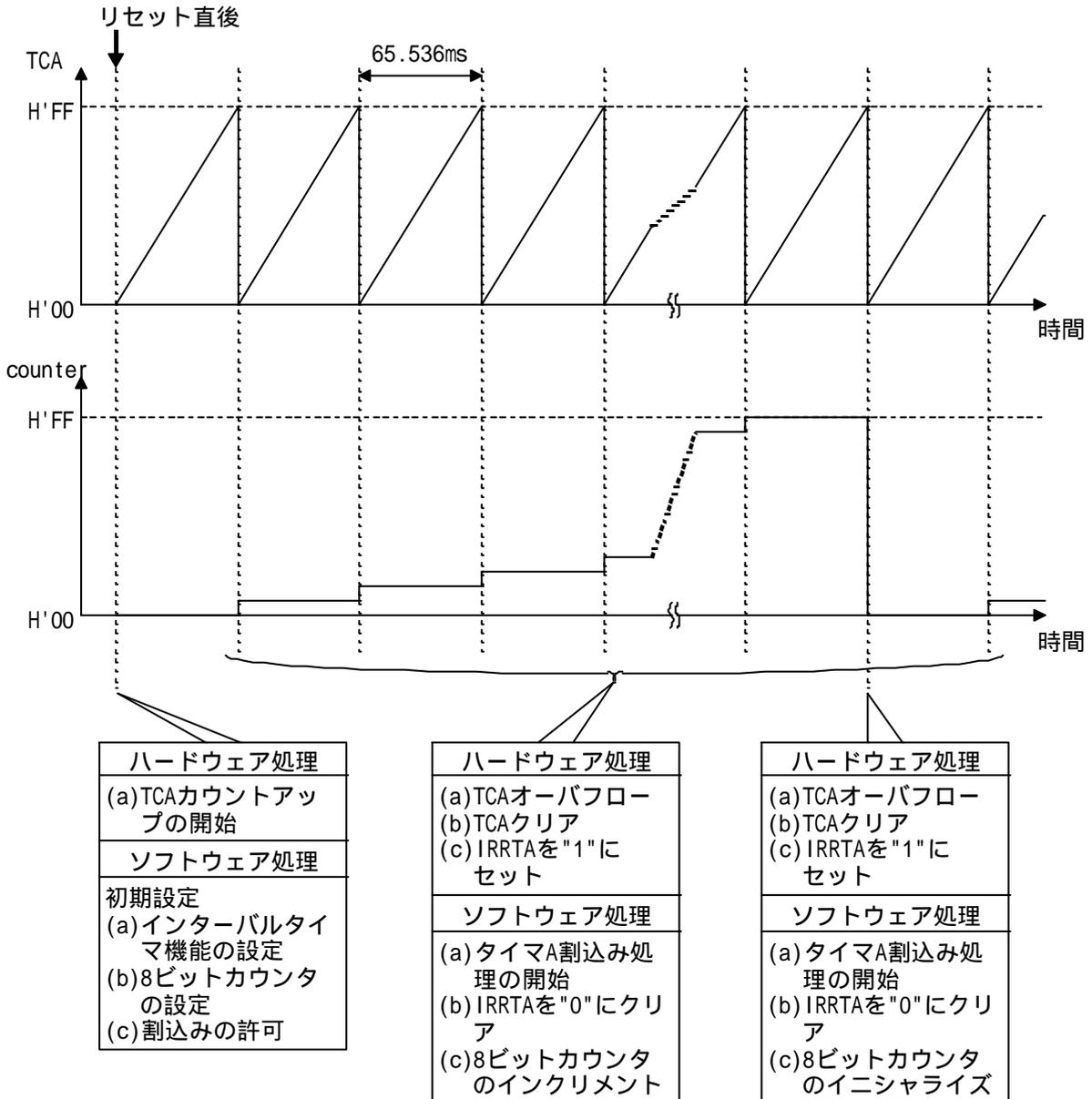


図2 タイマAインターバル機能による8ビットカウンタのカウントアップ動作の動作原理

ソフトウェア説明

(1) モジュール説明

表2に本タスク例におけるモジュール説明を示します。

表2 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	インターバルタイマの設定、8ビットカウンタの設定、割込みの許可を行なう
カウントアップ	taint	タイマA割込み処理ルーチンで、R1Lに設定した8ビットカウンタのインクリメント、またはイニシャライズを行なう

(2) 引数の説明

本タスク例では、引数は使用していません。

(3) 使用内部レジスタ説明

表3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3 使用内部レジスタ説明

レジスタ名		機能	アドレス	設定値
IRR1	IRRТА	割込み要求レジスタ1 (タイマA割込み要求フラグ) : IRRТА=0のとき、タイマA割込みが要求されていない : IRRТА=1のとき、タイマA割込みが要求されている	H'FFF6 ビット6	0
IENR1	IENТА	割込み許可レジスタ1 (タイマA割込みイネーブル) : IENТА=1のとき、タイマA割込み要求を許可	H'FFF4 ビット6	1
TMA		タイマモードレジスタA : TMA=H'11のとき、タイマA機能をインターバル機能に、TCA入力クロックソースをPSSに、プリスケラ分周比を4096分周に設定	H'FFA6	H'11
TCA		タイマカウンタA : システムクロックを4096分周したクロックを入力とする8ビットのアップカウンタ	H'FFA7	H'00

(4) 使用RAM説明

表4に本タスク例における使用RAM説明を示します。

表4 使用RAM説明

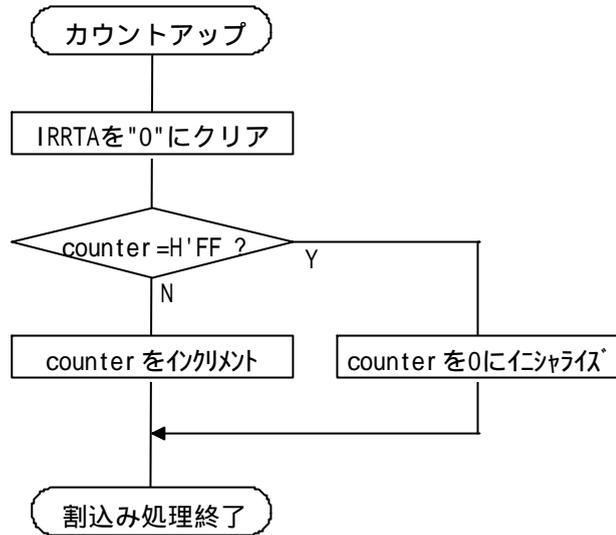
ラベル名	機能	アドレス	使用モジュール名
counter	タイマA割込みカウント用8ビットカウンタ	H'FB80	メインルーチン タイマA割込みルーチン

フローチャート

(a) メインルーチン



(b) タイマA割込み処理ルーチン



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。

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
/*:
/*: '8-bit Counter Count-Up by Interval Function'
/*:
/*: Function
/*: : Timer A Interval Timer
/*:
/*: External Clock : 16MHz
/*: Internal Clock : 16MHz
/*: Sub Clock : 32.768kHz
/*:
/*:******/

#include <machine.h>

/*:******/
/*: Symbol Definition
/*:******/
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 TMA *(volatile unsigned char *)0xFFA6 /* Timer Mode Register A */
#define TCA *(volatile unsigned char *)0xFFA7 /* Timer Counter A */
#define IENR1_BIT (*(struct BIT *)0xFFE4) /* Interrupt Enable Register 1 */
#define IENTA IENR1_BIT.b6 /* Timer A Interrupt Enable */
#define IRR1_BIT (*(struct BIT *)0xFFE6) /* Interrupt Request Register 1 */
#define IRRTA IRR1_BIT.b6 /* Timer A Interrupt Request Flag */

#pragma interrupt (taint)
/*:******/
/*: 関数定義
/*:******/
extern void INIT ( void ); /* SP Set */
void main ( void );
void taint ( void );
/*:******/
/*: RAM define
/*:******/

unsigned char counter; /* 8bit Counter */

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

```

プログラムリスト

```

INIT          /* 00 Reset          */
};
#pragma section V2          /* VECTOR SECTOIN SET  */
void (*const VEC_TBL2[])(void) = {
    taint          /* 26 Timer A Interrupt */
};
#pragma section          /* P          */
/*;*****/
/*; Main Program          */
/*;*****/
void main ( void )
{
    set_imask_ccr(1);          /* Interrupt Disable    */

    TMA = 0x11;          /* Initialize Timer A Function & TCA input clock period */

    IRRTA = 0;          /* Clear IRRTA          */
    IENTA = 1;          /* Timer A Interrupt Enable */

    counter = 0;          /* Initialize 8bit Counter */

    set_imask_ccr(0);          /* Interrupt Enable     */

    while(1) {
        ;
    }

/*;*****/
/*; Timer A Interrupt          */
/*;*****/
void taint ( void )
{
    IRRTA = 0;          /* Clear IRRTA          */

    if ( counter == 0xff ) {          /* 8bit Counter = 0xff? */
        counter = 0;          /* Clear 8bit Counter   */
    }
    else {
        counter++;          /* Increment 8bit Counter */
    }
}

```

リンクアドレス指定

セクション名	アドレス
CV1	H'0000
CV2	H'0026
P	H'0100
B	H'FB80