The following document contains information on Cypress products. The document has the series name, product name, and ordering part numbering with the prefix “MB”. However, Cypress will offer these products to new and existing customers with the series name, product name, and ordering part number with the prefix “CY”.

How to Check the Ordering Part Number

1. Go to www.cypress.com/pcn.
2. Enter the keyword (for example, ordering part number) in the **SEARCH PCNS** field and click **Apply**.
3. Click the corresponding title from the search results.
4. Download the Affected Parts List file, which has details of all changes

For More Information

Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress

Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, smart home appliances, consumer electronics and medical products. Cypress' microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first. Cypress is committed to providing customers with the best support and development resources on the planet enabling them to disrupt markets by creating new product categories in record time. To learn more, go to www.cypress.com.

AN205007

F2MC-8FX Family MB95430 Series 16-bit FRT and OCU Application Note

Associated Part Family: MB95430 Series

This application note introduces the functions of 16-bit Free-Run Timer and Output Compare Unit, and how to configure them. The related codes were also given in this application note.

Contents

1 Introduction....1
2 Feature of 16-bit OCU and FRT....1
2.1 OCU and FRT Block Diagram....1
2.2 OCU and FRT Registers....4
2.3 Pins Setting of OCU....5
2.4 Relationship with Voltage Comparator....5
2.5 Interrupt....5
3 Software Operation....6
3.1 Setting Procedure....6
3.2 Operation Mode....7
4 Influence of Some Bits....13
4.1 EOCS_BTSn....13
4.2 OCMCR_CMPMDn....14
4.3 OCMCR_FDENn (n= 0, 1)....16
5 Additional Information....17
A Appendix....18
A.1 Sample Code....18
Document History....25
Worldwide Sales and Design Support....26
Products....26
PSoC® Solutions....26
Cypress Developer Community....26
Technical Support....26

1 Introduction

This application note introduces the functions of 16-bit Free-Run Timer and Output Compare Unit, and how to configure them. The related codes were also given in this application note.

2 Feature of 16-bit OCU and FRT

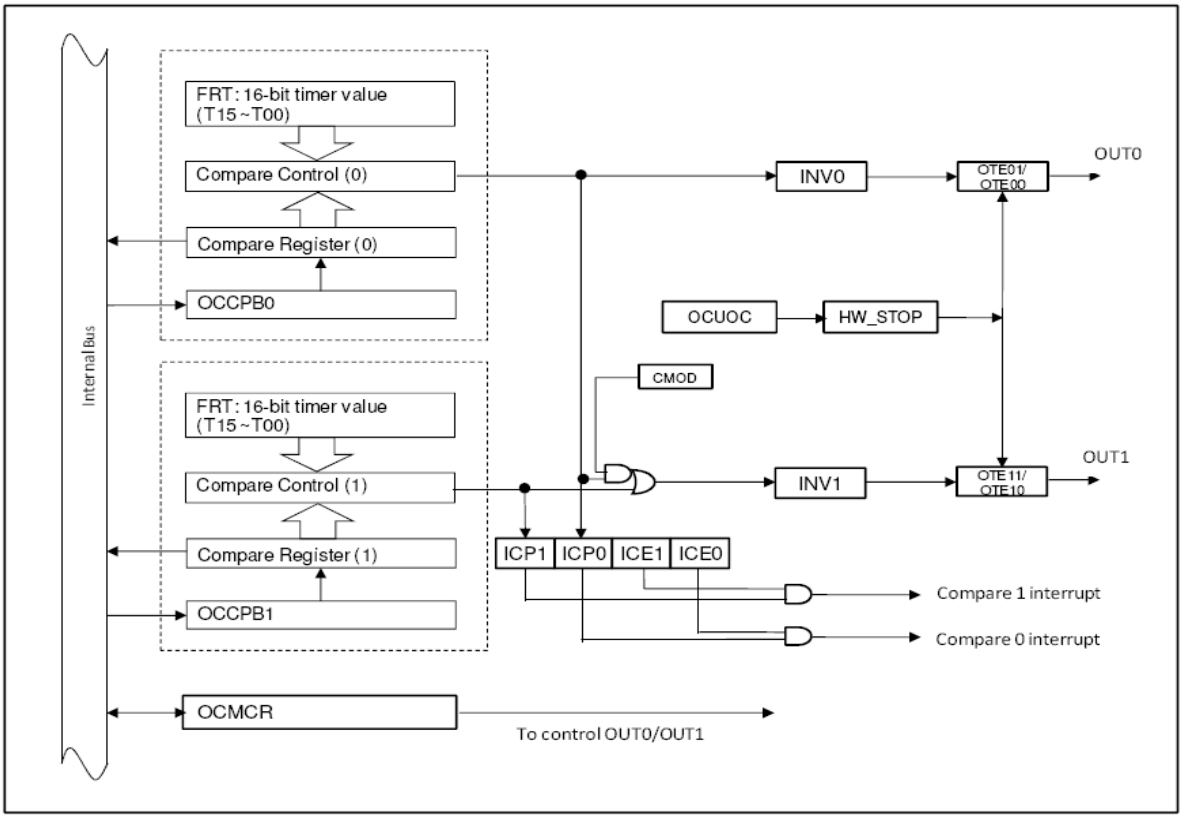
This chapter introduces the features of OCU and FRT.

2.1 OCU and FRT Block Diagram

The 16-bit Output Compare Unit is used for generation of pulse sequences. It consists of a 16-bit Free-Run Timer, two compare registers, one compare buffer register for each compare register, two compare output pins, and several control registers. If the value written to the compare register matches the 16-bit Free-Run Timer count value, the output level of the pin can be toggled and an interrupt will occur.

Figure 2-1 shows the block diagram of 16-bit OCU.

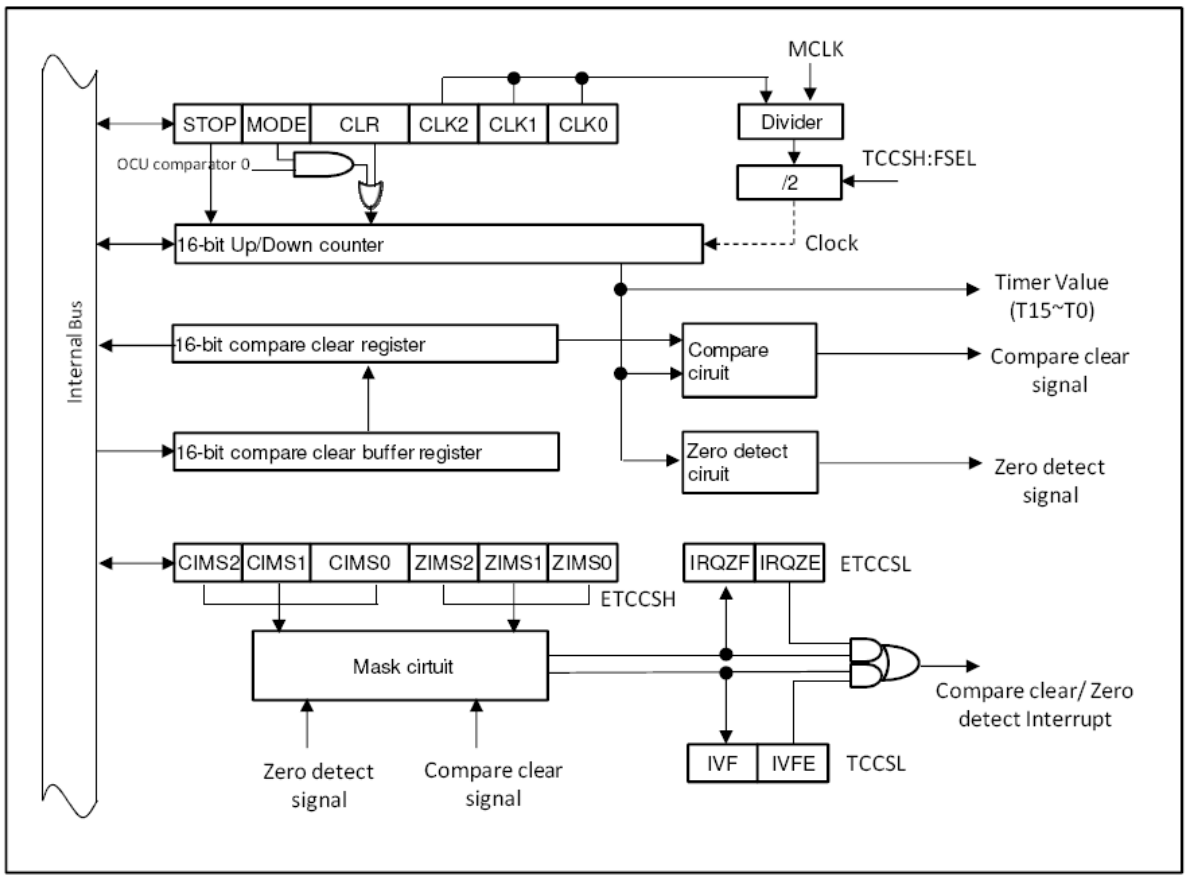
Figure 2-1: OCU Block Diagram



The 16-bit Free-Run Timer consists of a 16-bit up/down counter, compare clear buffer register and a control status register. The count values of this timer are used as the time base of the Output Compare Unit.

Figure 2-2 shows the block diagram of 16-bit Free-Run Timer.

Figure 2-2: FRT block diagram



2.2 OCU and FRT Registers

2.2.1 Registers Associated with Output Compare Unit

The OCU module has two channels—OUT0, OUT1. Each of them has a 16-bit Output Compare Register and a 16-bit Buffer Register. The Output Register and Buffer Register locate at the same address. The Output Registers are read-only, and the Buffer Registers are write-only. They can be accessed as the way mentioned in section 2.2.2.

Table 2-1 shows the Registers associated with the output compare unit.

Table 2-1: OCU Registers

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit1	Bit 0	Default Value
OCCP0/B0 H	Output Compare Register 0/ Buffer Register 0 H								b'00000000
OCCP0/B0 L	Output Compare Register 0/ Buffer Register 0 L								b'00000000
OCCP1/B1 H	Output Compare Register 1/ Buffer Register 1 H								b'00000000
OCCP1/B1 L	Output Compare Register 1/ Buffer Register 1 L								b'00000000
OCSL	ICP1	ICP0	ICE1	ICE0	--	--	CST1	CST0	b'00000000
OCSH	--	CMOD0	OTE11	OTE10	OTE01	OTE00	OTD1	OTD0	b'01000000
EOCS	HW_STOP	--	BTS1	BUF1	--	--	BTS0	BUF0	b'00000000
OCMCR	--	FDEN1	INV1	CMPMD1	--	FDEN0	INV0	CMPMD0	b'00000000
OCUOC	--	--	--	--	--	OCSTPSEL			b'00000000

2.2.2 Registers Associated with Free-Run Timer

Table 2-2 shows the Registers associated with the free-run timer.

Table 2-2: FRT Registers

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit1	Bit 0	Default Value
CPCLRH	Timer Compare Clear Register H								b'11111111
CPCLRL	Timer Compare Clear Register L								b'11111111
CPCLRBH	Timer Compare Clear Buffer Register H								b'11111111
CPCLRBL	Timer Compare Clear Buffer Register L								b'11111111
TCCSL	IVF	IVFE	STOP	MODE	CLR	CLK2	CLK1	CLK0	b'00000000
TCCSH	--	FSEL	--	--	--	--	--	--	b'01000000
ETCCSL	--	--	CNTDIR	--	BFE	IRQZF	IRQZE	CNTMD	b'00000000
ETCCSH	--	CIMS2	CIMS1	CIMS0	--	ZIMS2	ZIMS1	ZIMS0	b'00000000

The compare clear buffer registers (CPCLRBH/L) exist at the same address as the compare clear registers (CPCLRH/L). The compare clear buffer registers are write-only registers, and the compare clear registers are read-only registers. Two of them are 16-bit registers, please always use one of the following procedures to write or read them:

- Use 16-bit access instructions to write or read them (Such as "MOVW").
- Use byte access instructions to write or read upper part first, and then lower part (Such as "MOV").

2.3 Pins Setting of OCU

The OCU module has two pairs of output. The PG0 or P70 can be configured as OUT0, and the PG1 or P73 can be configured as OUT1. The configuration is as below:

SYSC2_OUTSEL0 = 0: OUT0 -- P70.

SYSC2_OUTSEL0 = 1: OUT0 -- PG0.

SYSC2_OUTSEL1 = 0: OUT1 -- P73.

SYSC2_OUTSEL1 = 1: OUT1 -- PG1.

In order to output the output compare signal, the user should enable the channels by OCSH_OTEn; otherwise the pin status was controlled by GPIO or other peripherals. It is also recommended to initialize the pin to a safe status before disable the OCU channels.

2.4 Relationship with Voltage Comparator

The OCU module can be stopped by voltage comparator output if OTE00, OTE 01(OTE10, OTE11) are set as 11. The user can configure the register OCUOC to decide which voltage comparator and what edge will stop the OCU.

2.5 Interrupt

2.5.1 Free-Run Timer interrupt

To FRT, the interrupt request will happen at the point of zero-detection or compare clear depending on the setting if the interrupts were enabled first. The user also can mask the interrupt for several matches by setting the ETCCSH register.

2.5.2 OCU interrupt

To OCU, each channel has an interrupt respectively. When output compare occurred, the interrupt flag will be set.

If an interrupt is enabled, the priority of the interrupt should be set in the vector.c file. The interrupt vectors are:

Table 2-3: Interrupt Vectors

Interrupt source	Interrupt request no.	Interrupt level setting register		Vector table address	
		Register	Setting bit	Upper	Lower
Output compare ch0 match	IRQ07	ILR07	L07[1:0]	FFECH	FFEDH
Output compare ch1 match	IRQ08	ILR08	L08[1:0]	FFEAH	FFEBH
16-bit Free-Run Timer	IRQ14	ILR14	L14[1:0]	FFDEH	FFDFH

3 Software Operation

This chapter introduces the software operation of output compare unit.

The OCU have a 16-bit free-run timer which supplies the time base. To set the free-run timer is prerequisite. In this AN, the sample code of FRT will be given. For more information about free-run timer, please refer to the chapter"16-bit Free-run Timer".

3.1 Setting Procedure

3.1.1 Free-Run Timer

Set the free-run timer count clock by TCCSL_CLK0~CLK2.

Select the count mode of FRT by ETCCSL_CNTMD.

Write the compare clear value to CPCLRH/L.

Clear the free-run timer counter by TCCSL_CLR.

Enable compare clear buffer if necessary.

Enable compare clear or zero detection interrupts if necessary.

Set compare clear or zero detection interrupts mask if necessary.

Enable the timer after OCU has been configured.

3.1.2 Output Compare Unit

Select output pin.

Write the output compare value to OCCP0 and OCCP1.

Set the compare match output by OCMCR_CMPMDn.

Enable output compare buffer if necessary.

Select the data transfer point if the buffer is enabled by EOCS_BTSn.

Define compare output mode for channel 1 by OCSH_CMOD0.

Select the Initial level of the output compare pins by OCSH_OTDn.

Select the Hardware Stop trigger source if necessary.

Enable the interrupts if necessary.

Enable the output compare pins by OCSH_OTE.

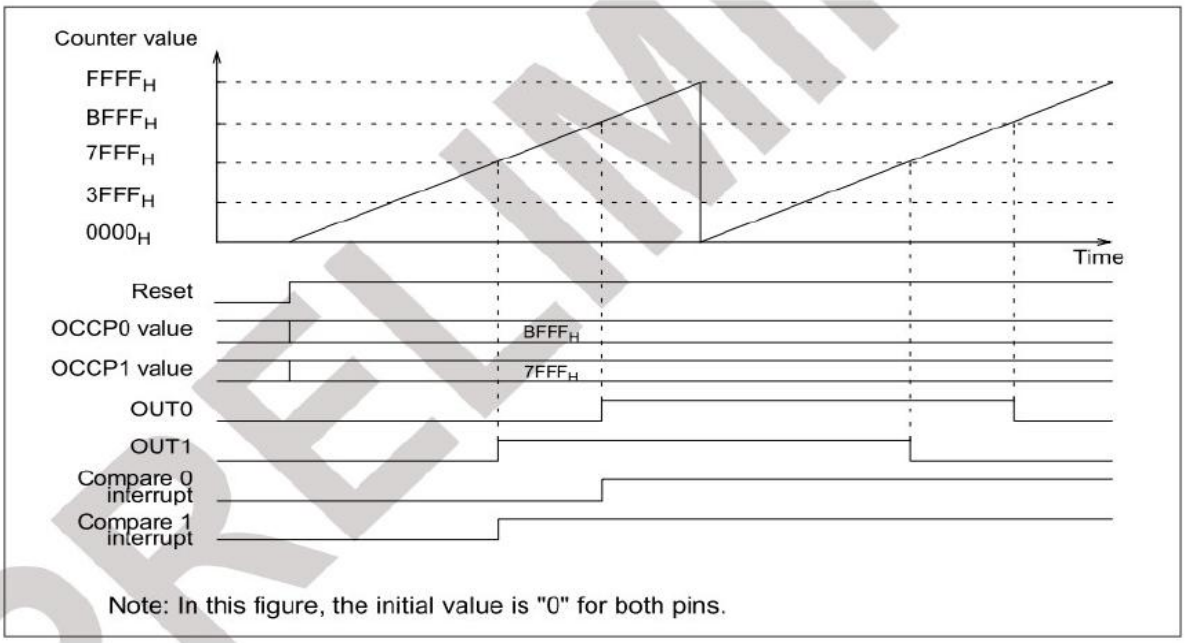
Enable the output compare operation by OCSL_CSTn.

3.2 Operation Mode

3.2.1 Output Pulse Sequences with Phase Difference

Figure 3-1 shows the pulse sequences with phase difference.

Figure 3-1: Pulse Sequence with Phase Difference



See the Figure 3-1, the OUT0 and OUT1 have the same period and duty, but the raising edge and falling edge happened at different point. The difference between the raising edges is phase difference. In order to output the desired sequences, the free-run timer operates in up-count mode. Suppose the same clock input, the compare clear register value decides the period of the pulse sequence. The user can modify the compare clear register value by writing different value to the register CPCLRH/L. The difference of OCCP0 and OCCP1 produces the phase difference of the pulse sequences.

The period (up-count mode) can be calculates by the formula below:

Period =2* (1/MCLK)*(1/Pre-scale)*CPCLRH/L. **(3-1)**

Pre-scale: free-run timer pre-scale.

So,

CPCLRH/L = Period*MCLK*Pre-scale/ 2. **(3-2)**

Suppose:

FCH = 4M, MCLK =4M, frequency = 20k, Pre-scale =1/4.

Then:

CPCLH/L = 25.

- Initialization

The code in Figure 3-2 shows the example step required to configure the FRT and OCU module. P70 and P73 are selected as the output pins. The Free-Run Timer is in up-count mode. The pre-scale is 1:4. Buffer is enabled. The Free-Run Timer and OCU function are all disabled at first in initialization code.

Note: Ensure that a value is written to the compare registers before enable the compare operation

Figure 3-2: Initialization Code

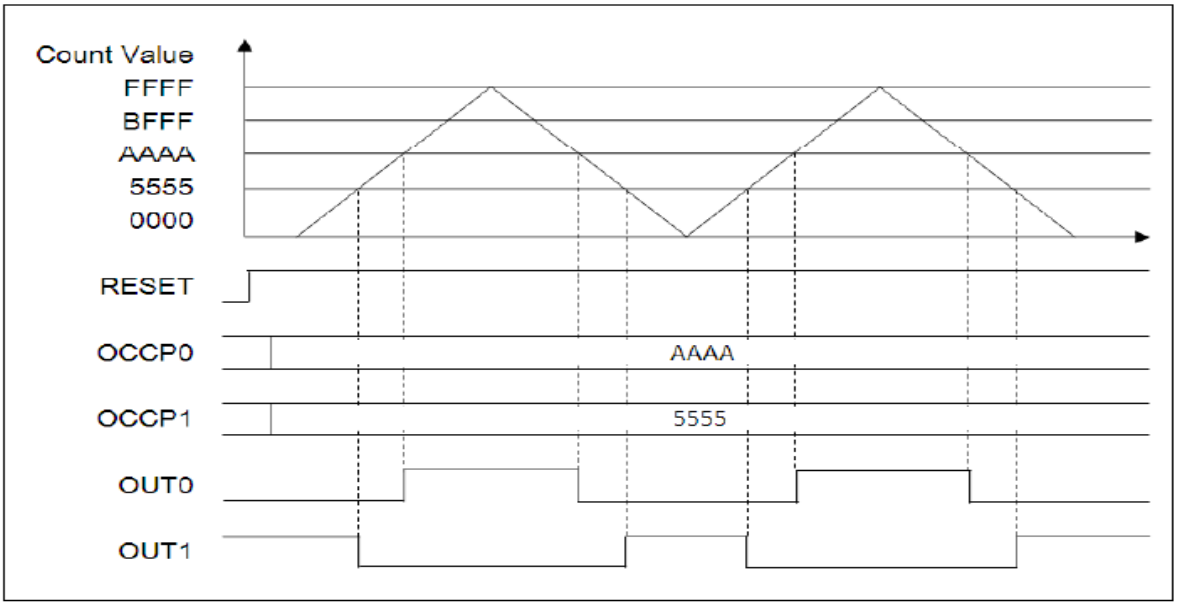
```
SYSC2_OUTSEL0 = 0;              // select OUT0 output pin
SYSC2_OUTSEL1 = 0;              // select OUT1 output pin
DDR7_P70 = 1;                   // OUT0 output
DDR7_P73 = 1;                   // OUT1 output
PDR7_P70 = 0;                   // OUT0 output 0 first
PDR7_P73 = 0;                   // OUT1 output 0 first

TCCSH = 0x40;                   // count clock have no division
TCCSL = 0x2A;                   // clear timer
                                // disable timer first
                                // count clock=1/4*MCLK
ETCCSH = 0x0;                   // no interrupt mask
ETCCSL = 0x08;                  // up count mode
                                // interrupt disabled
CPCLRH = 0xFF;                  // write upper register first
CPCLRL = 0xFF;                  // then down register

OCSL = 0x00;                    // disable compare operation first
OCSH = 0x00;                    // initialize ch0&ch1 to 0
                                // disable output
                                // ch1 reverses upon comp reg1
EOCS = 0x11;                    // buffer enabled
                                // transfer data at zero point
OCCP0H = 0xBF;                  // write upper register first
OCCP0L = 0xFF;                  // then down register

OCCP1H = 0x7F;
OCCP1L = 0xFF;
```

3.2.2 Output Pulse Sequences with Dead Time

Pulse sequence with dead time is usually used to drive H-bridge. The OCU can't output two channels of PWM signal, but it can change the frequency of the two pulse sequence easily. So, it is useful for PFM method in H-bridge driving. Figure 3-3 shows the waveform.

The CPCLRH/L is available through the formula (3-2). The duty of the pulse sequences are 50%.

Figure 3-3: Pulse Sequences with Dead Time



Dead time can be calculated like this

Dead Time = (OCCP0-OCCP1) *(1/MCLK)*(1/Pre-scale)

Changing the value of OCCP0-OCCP1 will change the dead time.

- Initialization

In this mode, the Free-Run Timer is in up/down mode. The value of CPCLRH/L is default as 0XFFFF. The OCCP0 and OCCP1 are written as 0XAAAA and 0X5555 respectively that the difference produces the dead time. Figure 3-4 shows the initialization code.

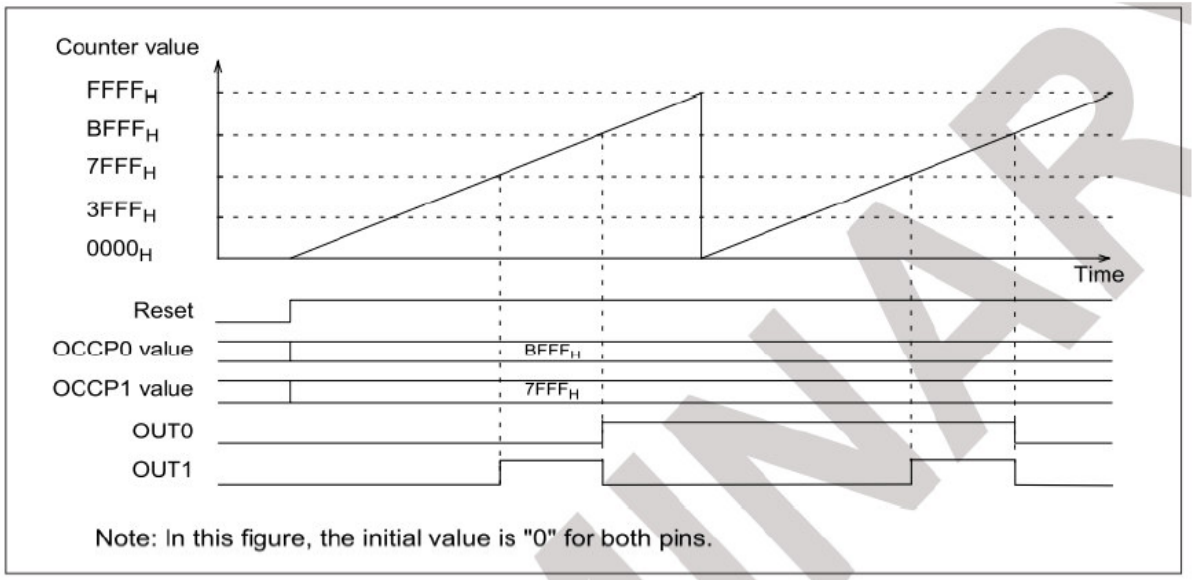
Figure 3-4: Initialization for Pulse Sequence with Dead Time

```
SYSC2_OUTSEL0 = 0;              // select OUT0 output pin
SYSC2_OUTSEL1 = 0;              // select OUT1 output pin
DDR7_P70 = 1;                   // OUT0 output
DDR7_P73 = 1;                   // OUT1 output
PDR7_P70 = 0;                   // OUT0 output 0 first
PDR7_P73 = 0;                   // OUT1 output 0 first

TCCSH = 0x40;                   // count clock have no division
TCCSL = 0x2A;                   // clear timer
                                // disable timer first
                                // count clock=1/4*MCLK
ETCCSH = 0x0;                   // no interrupt mask
ETCCSL = 0x09;                  // up/down count mode
                                // interrupt disabled
CPCLRH = 0xFF;                  // write upper register first
CPCLRL = 0xFF;                  // then down register

OCSL = 0x00;                    // disable compare operation first
OCSH = 0x00;                    // initialize ch0&ch1 to 0
                                // disable output
                                // ch1 reverses upon comp reg1
EOCS = 0x11;                    // buffer enabled
                                // transfer data at zero point
OCCP0H = 0xAA;                  // write upper register first
OCCP0L = 0xAA;                  // then down register

OCCP1H = 0x55;
OCCP1L = 0x55;
```

3.2.3 Output PWM

The OCU module can generate one channel of PWM through OUT1 when the bit OCSH_CMOD0 was set as 1. Figure 3-5 shows the PWM waveform.

Figure 3-5: PWM Waveform



The period can be calculated as:

Period = (1/MCLK)*(1/Pre-scale)*CPCLRH/L. **(3-3)**

Pre-scale: free-run timer pre-scale.

So,

CPCLRH/L = Period*MCLK*Pre-scale **(3-4)**

If the Free-Run Timer is in up/down mode, the period will be calculated by formula (3-1).

The duty will be changed when change the value of OCCP0. Keep the OCCP1 unchanged, modify OCCP0 to 0XFFFF, the duty of the OUT1 will be larger.

- Initialization

The output channel 1 was configured as reversing upon a match with compare register 0, 1 in this mode. Changing the value of the OCCP0 will change the duty of PWM waveform.

Figure 3-6 shows the initialization code for PWM.

Figure 3-6: Initialization code for PWM

```
SYSC2_OUTSEL0 = 0;              // select OUT0 output pin
SYSC2_OUTSEL1 = 0;              // select OUT1 output pin
DDR7_P70 = 1;                   // OUT0 output
DDR7_P73 = 1;                   // OUT1 output
PDR7_P70 = 0;                   // OUT0 output 0 first
PDR7_P73 = 0;                   // OUT1 output 0 first

TCCSH = 0x40;                   // count clock have no division
TCCSL = 0x2A;                   // clear timer
                                // disable timer first
                                // count clock=1/4*MCLK
ETCCSH = 0x0;                   // no interrupt mask
ETCCSL = 0x08;                  // up count mode
                                // interrupt disabled
CPCLRH = 0xFF;                  // write upper register first
CPCLRL = 0xFF;                  // then down register

OCSL = 0x00;                    // disable compare operation first
OCSH = 0x40;                    // initialize ch0&ch1 to 0
                                // disable output
                                // ch1 reverses upon comp reg0,1
EOCS = 0x11;                    // buffer enabled
                                // transfer data at zero point
OCCP0H = 0xBF;                  // write upper register first
OCCP0L = 0xFF;                  // then down register

OCCP1H = 0x7F;
OCCP1L = 0xFF;
```

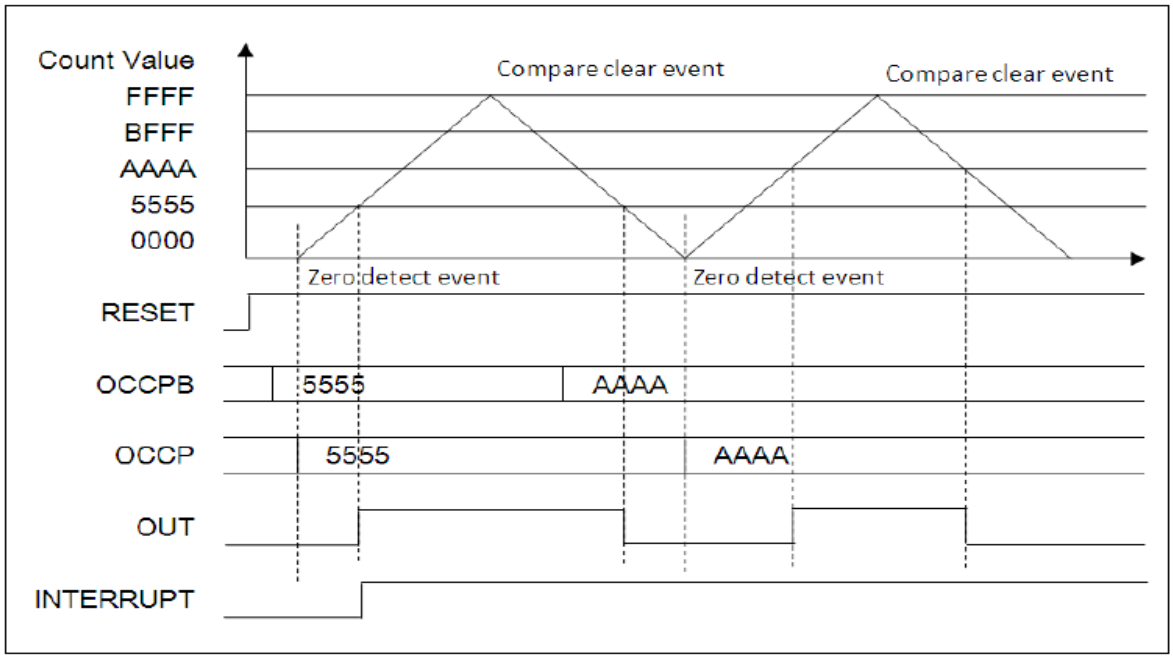
4 Influence of Some Bits

4.1 EOCS_BTSn

EOCS_BTSn decide at which point the output compare buffer register value was transferred to the output compare register when the buffer function is enabled.

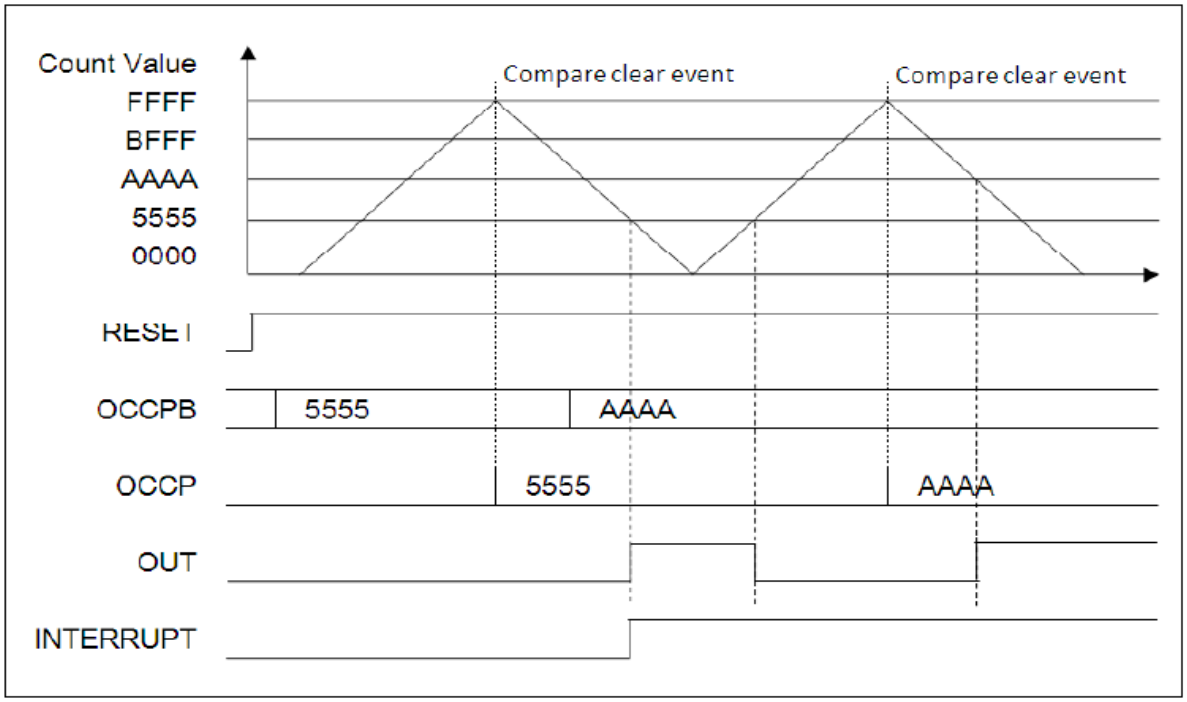
- EOCS_BTSn = 0, the transformation happened at zero-detection of FRT.

Figure 4-1: Transform at Zero-detection



EOCS_BTSn = 1, the transformation happened at compare-clear of FRT.

Figure 4-2: Transform at Compare Clear



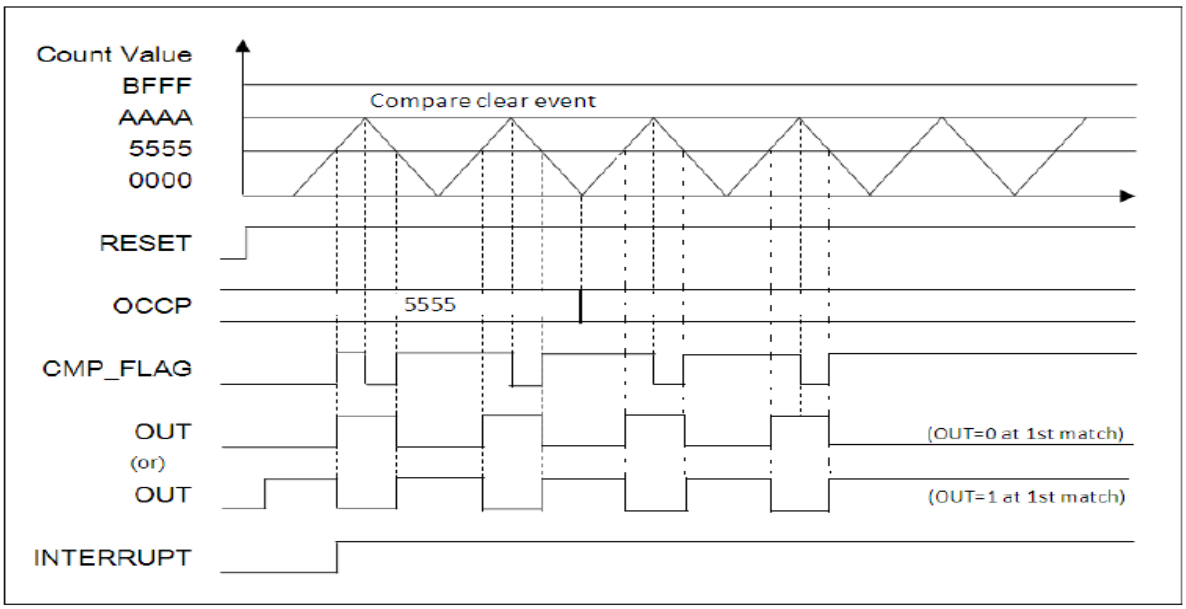
4.2 OCMCR_CMPMDn

This bit is used to change the pin output level immediately after a match occurred when pin output is enabled.

- OCMCR_CMPMDn = 0

In this mode, the pin level will reverse against previous value. So, the initial level of pin should be taken care. Figure 4-3 shows the waveform.

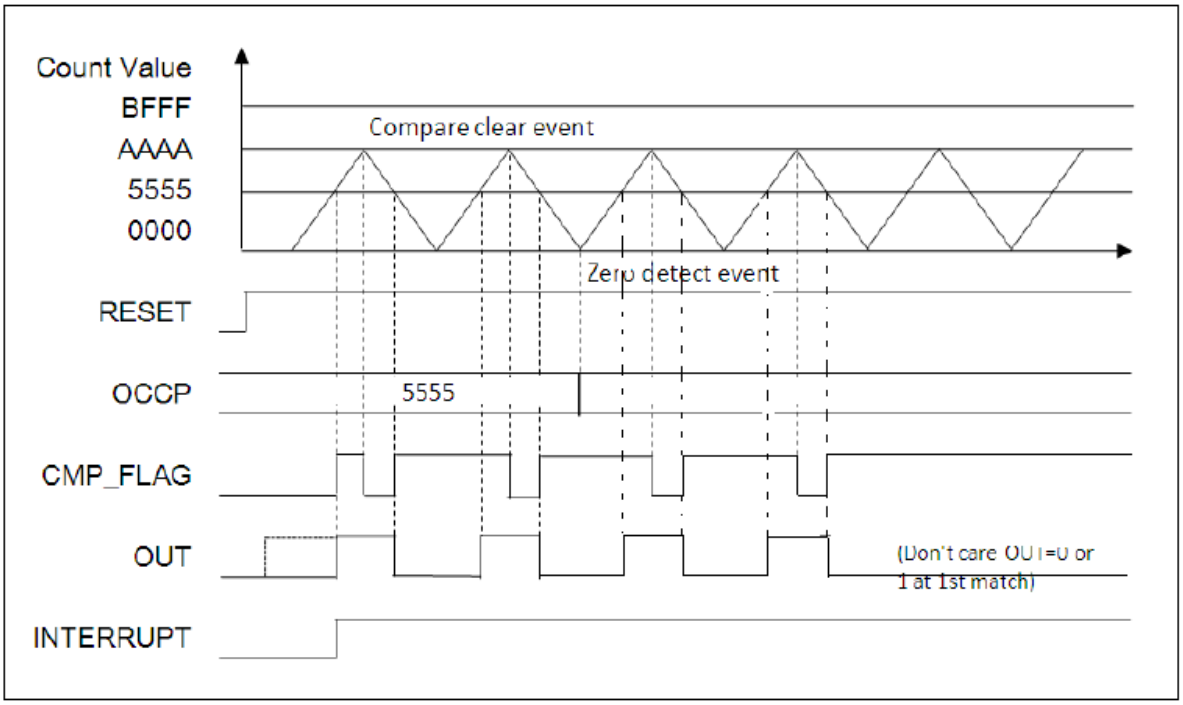
Figure 4-3: Waveform for CMPMDn=0



- OCMCR_CMPMDn = 1

In this mode, the output pin level will be set as 1 when Free-Run Timer is in the up-count, and be cleared when Free-Run Timer is in down-count. The Initial value of the output pin will not be concerned in this mode.

Figure 4-4: Waveform for CMPMDn= 1



4.3 OCMCR_FDENn (n= 0, 1)

This bit is used for “Full duty range function” for ch0 or ch1.

When FDENn = 0, the Full duty range function feature is disabled.

When FDENn =1, the Full duty range function feature is enabled (Must set OCSH_CMOD0 = 0).

If FDENn = 0.

Output level toggles when a match of the Free-Run Timer Value with the compare register (OCCPn) occurs.

If FDENn =1 and COMD0=0

When OCCPn= 0, then the output of OCU is independent of the Free-Run Timer value.

OUTn = OCMCR_INVn;

When 0<OCCP0<CPCLR

The output of OCU depends on CMPMDn bit.

When OCCP0>= CPCLR

The output of OCU depends on OCU compare flag at compare clear event.

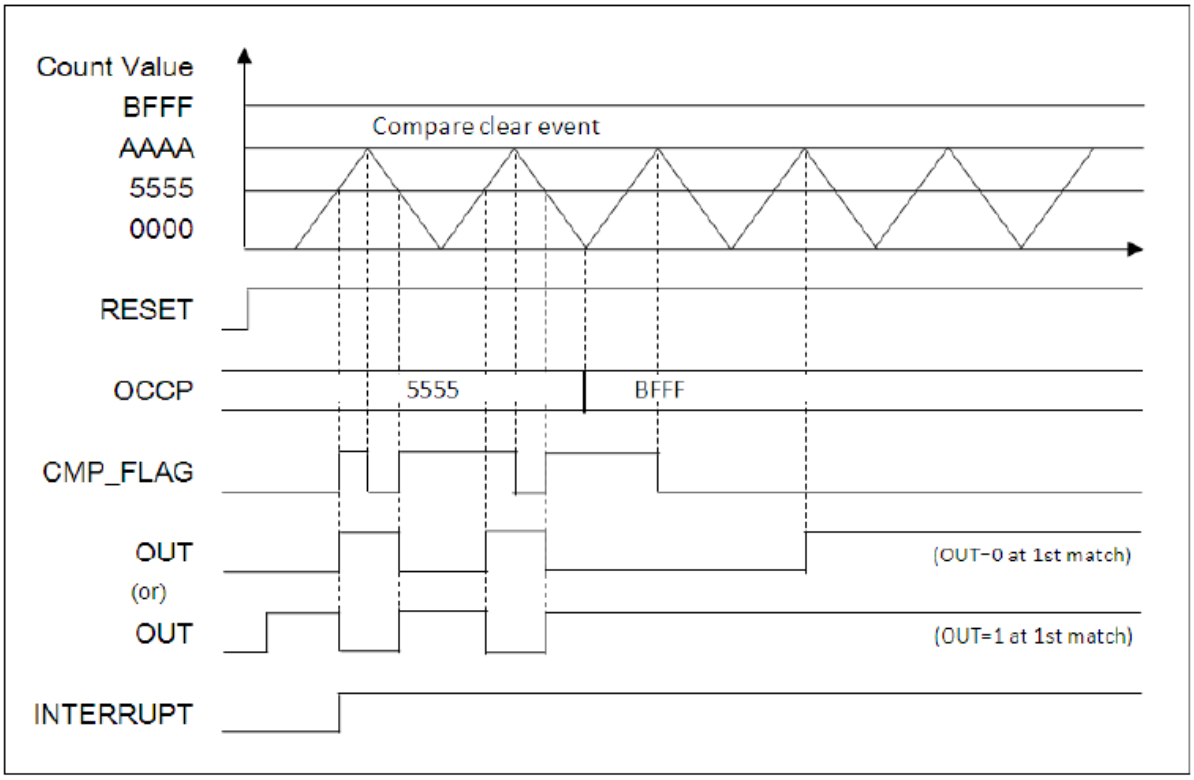
If CMP_FLAG =0, then OUTn =-OCMCR_INVn.

If CMP_FLAG =1, then OUTn is no change.

CMP_FLAG is set at compare match event and reset at compare clear event.

Figure 4-5 shows the waveform when FDEN =1, CMPMD0= 0, INV0 = 0.

Figure 4-5: Waveform of Full duty range



5 Additional Information

For more information on Cypress products, please visit the following websites:

http://www.cypress.com/cypress-microcontrollers

A Appendix

A.1 Sample Code

```
Project1: Pulse sequences with phase difference

#include "mb95430.h"
/*-------------------------------------------------------------------------------
Name        : InitFRTandOCU ()
Function    : Initialize the Free-Run Timer and OCU module
Input       : void
Output            : void
-------------------------------------------------------------------------------*/
void InitFRTandOCU (void)
{
      SYSC2_OUTSEL0 = 0;                  // select OUT0 output pin
      SYSC2_OUTSEL1 = 0;                  // select OUT1 output pin
      DDR7_P70 = 1;                       // OUT0 output
      DDR7_P73 = 1;                       // OUT1 output
      PDR7_P70 = 0;                       // OUT0 output 0 first
      PDR7_P73 = 0;                       // OUT1 output 0 first

      TCCSH = 0x40;                       // count clock have no division
      TCCSL = 0x2A;                       // clear timer
                                          // disable timer first
                                          // count clock=1/4*MCLK

      ETCCSH = 0x0;                       // no interrupt mask
      ETCCSL = 0x08;                      // up count mode
                                          // interrupt disabled
      CPCLRH = 0xFF;                      // write upper register first
      CPCLRL = 0xFF;                      // then down register

      OCSL = 0x00;                              // disable compare operation first
      OCSH = 0x00;                              // initialize ch0&ch1 to 0
                                          // disable output

                                          // ch1 reverses upon comp reg1
      EOCS = 0x11;                              // buffer enabled
                                          // transfer data at zero point
      OCCP0H = 0xBF;                      // write upper register first
      OCCP0L = 0xFF;                      // then down register

      OCCP1H = 0x7F;
      OCCP1L = 0xFF;
}
/*-------------------------------------------------------------------------------
Name        : EnableFRTandOCU ()
Function    : Enable the Free-Run Timer and OCU module
Input       : void
Output            : void
-------------------------------------------------------------------------------*/
void EnableFRTandOCU (void)
{
      TCCSL_CLR = 1;                      // clear FRT counter
```

```
      /*
      take care here, strongly recommend to write 0 to TCDTH/L !!!!
      */
      TCDTH = 0;
      TCDTL = 0;

      OCSH = 0x28;                              // output channels are enabled
                                                // if HW_STOP=0
      OCSL_CST0 = 1;                            // enable output compare0 operation
      OCSL_CST1 = 1;                            // enable output compare1 operation

      TCCSL_STOP = 1;                           // start free-run timer counting
}
/*------------------------------------------------------------------------------
Name        : DisableFRTandOCU ()
Function    : Disable the Free-Run Timer and OCU module
Input       : void
Output            : void
------------------------------------------------------------------------------*/
void DisableFRTandOCU(void)
{
      OCSH = 0;                                 // output channels are disabled
                                                // the pin are controlled by GPIO
      TCCSL_STOP = 0;                           // stop free-run timer counting

      OCSL_CST0 = 0;                            // disable operation
      OCSL_CST1 = 0;
}
/* delay */
void Delayms (unsigned char cnt_Dly)
{
      unsigned char i;
      for(; cnt_Dly > 0 ; cnt_Dly-- )
      {
            for(i = 250 ; i > 0 ; i-- )
            {
                  __wait_nop ();
                  __wait_nop ();
                  __wait_nop ();
                  __wait_nop ();
                  __wait_nop ();
                  __wait_nop ();
            }
      }
}

main()
{
      __DI();                                   // disable interrupt
      SYCC = 0x01;                              // MCLK = (1/4)*8M
      InitFRTandOCU ();                         // enable interrupt
      __EI();
```

```
      while(1)
      {
            EnableFRTandOCU ();
            Delayms(100);
            DisableFRTandOCU ();
            Delayms(100);
      }
}
```

Project 2: Output Pulse Sequences with Dead Time

```
#include "mb95430.h"
/*-------------------------------------------------------------------------------
Name        : InitFRTandOCU ()
Function    : Initialize the Free-Run Timer and OCU module
Input       : void
Output            : void
-------------------------------------------------------------------------------*/
void InitFRTandOCU (void)
{
      SYSC2_OUTSEL0 = 0;                  // select OUT0 output pin
      SYSC2_OUTSEL1 = 0;                  // select OUT1 output pin
      DDR7_P70 = 1;                       // OUT0 output
      DDR7_P73 = 1;                       // OUT1 output
      PDR7_P70 = 0;                       // OUT0 output 0 first
      PDR7_P73 = 0;                       // OUT1 output 0 first

      TCCSH = 0x40;                       // count clock have no division
      TCCSL = 0x2A;                       // clear timer
                                          // disable timer first
                                          // count clock=1/4*MCLK

      ETCCSH = 0x0;                       // no interrupt mask
      ETCCSL = 0x08;                      // up count mode
                                          // interrupt disabled
      CPCLRH = 0xFF;                      // write upper register first
      CPCLRL = 0xFF;                      // then down register

      OCSL = 0x00;                              // disable compare operation first
```

```
      OCSH = 0x40;                              // initialize ch0&ch1 to 0
                                          // disable output

                                          // ch1 reverses upon comp reg0,1
      EOCS = 0x11;                              // buffer enabled
                                          // transfer data at zero point
      OCCP0H = 0xBF;                      // write upper register first
      OCCP0L = 0xFF;                      // then down register

      OCCP1H = 0x7F;
      OCCP1L = 0xFF;
}
/*-------------------------------------------------------------------------------
Name        : EnableFRTandOCU ()
Function    : Enable the Free-Run Timer and OCU module
Input       : void
Output            : void
-------------------------------------------------------------------------------*/
void EnableFRTandOCU(void)
{
      TCCSL_CLR = 1;                      // clear FRT counter
      /*
      take care here, strongly recommend to write 0 to TCDTH/L !!!!
      */
      TCDTH = 0;
      TCDTL = 0;

      OCSH = 0x68;                              // output channels are enabled
                                          // if HW_STOP=0
      OCSL_CST0 = 1;                      // enable output compare0 operation
      OCSL_CST1 = 1;                      // enable output compare1 operation

      TCCSL_STOP = 1;                     // start free-run timer counting
}
/*-------------------------------------------------------------------------------
Name        : DisableFRTandOCU ()
Function    : Disable the Free-Run Timer and OCU module
Input       : void
Output            : void
-------------------------------------------------------------------------------*/
void DisableFRTandOCU (void)
{
      TCCSL_STOP = 0;                     // stop free-run timer counting

      OCSH = 0x80;                              // output channels are disabled
                                          // the pin are controlled by GPIO

      OCSL_CST0 = 0;                      // disable operation
      OCSL_CST1 = 0;
}
/*delay*/
void Delayms (unsigned char cnt_Dly)
{
      unsigned char i;
```

```
      for(; cnt_Dly > 0 ; cnt_Dly-- )
      {
            for(i = 250 ; i > 0 ; i-- )
            {
                  __wait_nop ();
                  __wait_nop ();
                  __wait_nop ();
                  __wait_nop ();
                  __wait_nop ();
                  __wait_nop ();
            }
      }
}
main()
{
      __DI();                                   // disable interrupt
      SYCC = 0x01;                              // MCLK = (1/4)*8M
      InitFRTandOCU ();                   // enable interrupt
      __EI();

      while(1)
      {
            EnableFRTandOCU ();
            Delayms(100);
            DisableFRTandOCU ();
            Delayms(100);
      }
}
```

Project 3: Output PWM

```
#include "mb95430.h"
/*-------------------------------------------------------------------------------
Name        : InitFRTandOCU ()
Function    : Initialize the Free-Run Timer and OCU module
Input       : void
Output            : void
-------------------------------------------------------------------------------*/
void InitFRTandOCU(void)
{
      SYSC2_OUTSEL0 = 0;                  // select OUT0 output pin
      SYSC2_OUTSEL1 = 0;                  // select OUT1 output pin
      DDR7_P70 = 1;                       // OUT0 output
      DDR7_P73 = 1;                       // OUT1 output
      PDR7_P70 = 0;                       // OUT0 output 0 first
      PDR7_P73 = 0;                       // OUT1 output 0 first

      TCCSH = 0x40;                       // count clock have no division
      TCCSL = 0x2A;                       // clear timer
                                          // disable timer first
                                          // count clock=1/4*MCLK

      ETCCSH = 0x0;                       // no interrupt mask
      ETCCSL = 0x08;                      // up count mode
```

```
                                        // interrupt disabled
      CPCLRH = 0xFF;                    // write upper register first
      CPCLRL = 0xFF;                    // then down register

      OCSL = 0x00;                            // disable compare operation first
      OCSH = 0x40;                            // initialize ch0&ch1 to 0
                                        // disable output

                                        // ch1 reverses upon comp reg0,1
      EOCS = 0x11;                            // buffer enabled
                                        // transfer data at zero point
      OCCP0H = 0xBF;                    // write upper register first
      OCCP0L = 0xFF;                    // then down register

      OCCP1H = 0x7F;
      OCCP1L = 0xFF;
}
/*------------------------------------------------------------------------------
Name        : EnableFRTandOCU ()
Function    : Enable the Free-Run Timer and OCU module
Input       : void
Output            : void
------------------------------------------------------------------------------*/
void EnableFRTandOCU(void)
{
      TCCSL_CLR = 1;                    // clear FRT counter
      /*
      take care here, strongly recommend to write 0 to TCDTH/L !!!!
      */
      TCDTH = 0;
      TCDTL = 0;

      OCSH = 0x68;                            // output channels are enabled
                                        // if HW_STOP=0
      OCSL_CST0 = 1;                    // enable output compare0 operation
      OCSL_CST1 = 1;                    // enable output compare1 operation

      TCCSL_STOP = 1;                   // start free-run timer counting
}
/*------------------------------------------------------------------------------
Name        : DisableFRTandOCU ()
Function    : Disable the Free-Run Timer and OCU module
Input       : void
Output            : void
------------------------------------------------------------------------------*/
void DisableFRTandOCU(void)
{
      TCCSL_STOP = 0;                   // stop free-run timer counting

      OCSH = 0x80;                            // output channels are disabled
                                        // the pin are controlled by GPIO

      OCSL_CST0 = 0;                    // disable operation
      OCSL_CST1 = 0;
```

```
}
/*delay*/
void Delayms (unsigned char cnt_Dly)
{
      unsigned char i;
      for(; cnt_Dly > 0 ; cnt_Dly-- )
      {
            for(i = 250 ; i > 0 ; i-- )
            {
                  __wait_nop ();
                  __wait_nop ();
                  __wait_nop ();
                  __wait_nop ();
                  __wait_nop ();
                  __wait_nop ();
            }
      }
}
main()
{
      __DI();                                   // disable interrupt
      SYCC = 0x01;                              // MCLK = (1/4)*8M
      InitFRTandOCU ();                   // enable interrupt
      __EI();

      while(1)
      {
            EnableFRTandOCU ();
            Delayms(100);
            DisableFRTandOCU ();
            Delayms(100);
      }
}
```

Document History

Document Title: AN205007 - F2MC-8FX Family MB95430 Series 16-bit FRT and OCU Application Note

Document Number: 002-05007

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	—	HUAL	03/12/2010	V1.0, First draft
			04/12/2010	V1.1, Update by review
			09/27/2010	V1.2, Update the source code
*A	5264237	HUAL	05/09/2016	Migrated Spansion Application Note “MCU-AN-500077-E-12” to Cypress format.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Lighting & Power Control	cypress.com/powerpsoc
Memory	cypress.com/memory
PSoC	cypress.com/psoc
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless/RF	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corporation. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2009-2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.