11 Timer_A

This section describes the basic functions of a the general purpose 16-bit Timer_A in MSP430 based system.

Topic **Page**

11.1 Operation of Timer_A 11-3
11.2 Registers of Timer_A 11-17
11.3 Timer_A in Applications 11-28
11.4 Timer_A special conditions 11-38

11.1 Operation of Timer_A

The major blocks of the 16-bit Timer_A are:

- a timer which can count continuously up to a predefined value, count up to a predefined value and down back to zero; the timer can also be stopped
- the clock source of the timer can be selected by software
- the selected clock source can be divided by one, two, four or eight
- five capture/compare registers, each with an individual capture event: two capture signals controlled by hardware or SW
- five output modules, supporting pulse-width modulation requirements.

The Timer_A is configured by means of the bits in the timer control register TACTL. This register defines the basic operation of the 16-bit timer. The input clock source - with its original frequency or pre-divided - and four different operating modes can be selected. Additionally, a clear function and the timer overflow interrupt control bits are included. A timer overflow is defined if the timer counts towards 0000h. This definition is independent of whether the timer counts up or down.

The five capture/compare registers operate identically, and are individually configurable with their control registers.

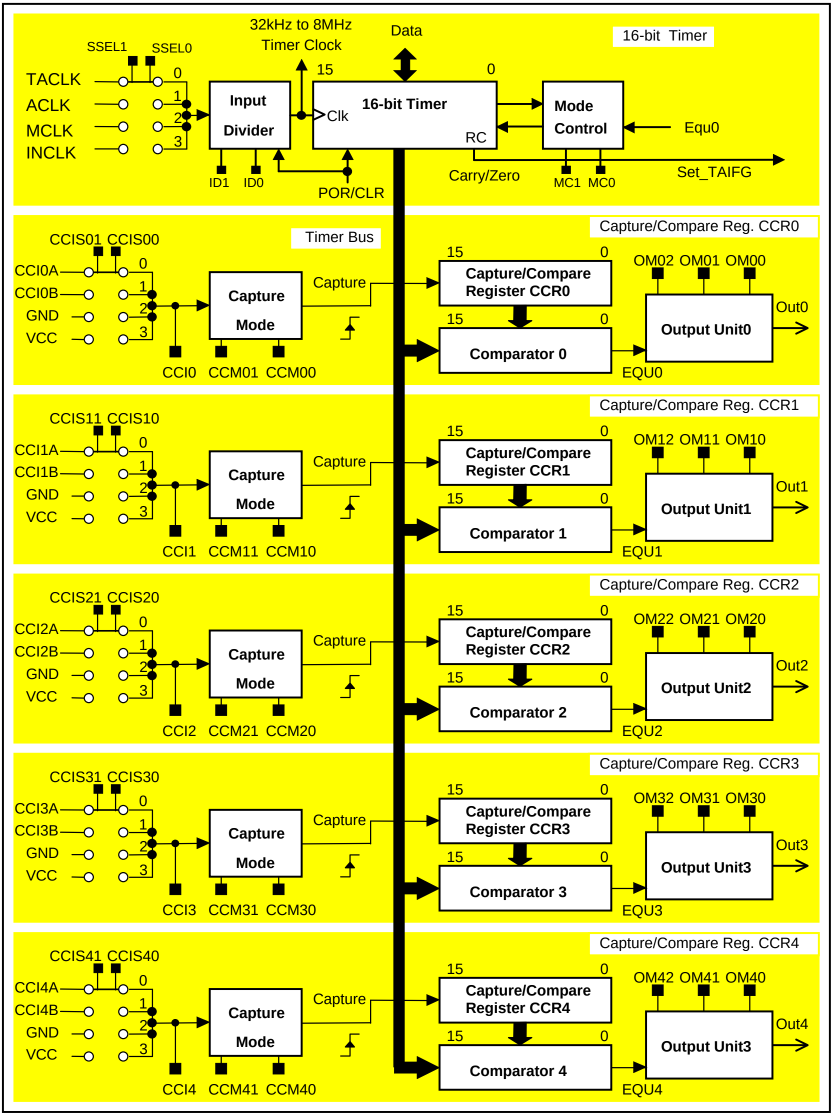


Figure 11.1: Schematic of Timer_A

11.1.1 Timer Operation

Four modes are provided to run the 16-bit timer and are defined with two control bits, MC1 and MC0, in the control register TACTL, plus the signal EQU0 which is the output of the comparator in the capture/compare 0 block. The clock source of the timer is selected via two bits - SSEL1 and SSEL0 - in the control register TACTL. The selected clock source is passed directly to the 16-bit timer or divided by 2, 4 or 8. The source signal can be supplied from internal clocks, or from outside.

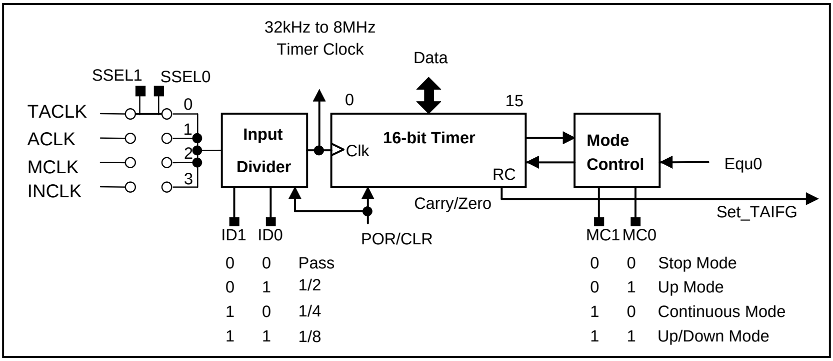


Figure 11.2: Schematic of 16-bit Timer

Clock Source Select and Input Divider

The clock source is selected by two control bits, SSEL0 and SSEL1. The output of the multiplexer directly proceeds from the previous selected signal and its level, to the new selected signal and its level. Short intermediate states of the two control bits can select any of the sources applied to the multiplexer. The input divider can receive additional clocks when the clock source is changed. The input divider is reset with POR signal - when VCC is applied or a reset condition at RST/NMI pin is detected - or when the timer is reset via bit CLR. The CLR bit is located in the timer control register TACTL. The input divider remains in its existing state when the timer is modified - even if zero is written to the timer. In normal operation, the existing state of the input divider is not visible for software.

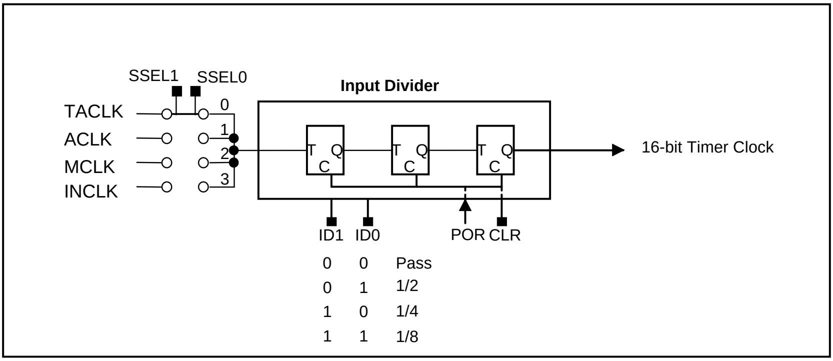


Figure 11.3: Schematic of Clock Source Select and Input Divider

Mode Control and 16-bit Timer

The 16-bit timer is incremented or decremented with each rising clock signal. It can be read and written directly from the software, via standard access to peripheral modules. The different modes are selected with bits MC1 and MC0.

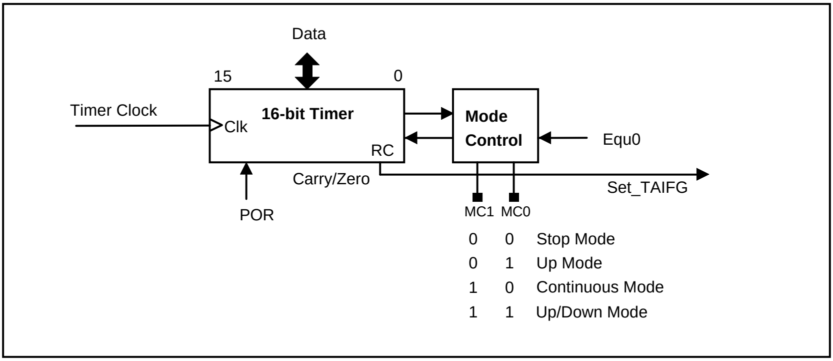


Figure 11.4: Schematic of Timer

During the low state of the timer clock, all operations are prepared which are executed with the following positive edge of the timer clock. Most of the special conditions that are discussed separately are based on this situation. An example of this feature is that a compare fails, if the counter has been already counted the value X and later the capture/compare register is also loaded with this data X.

Four timer operating modes are provided:

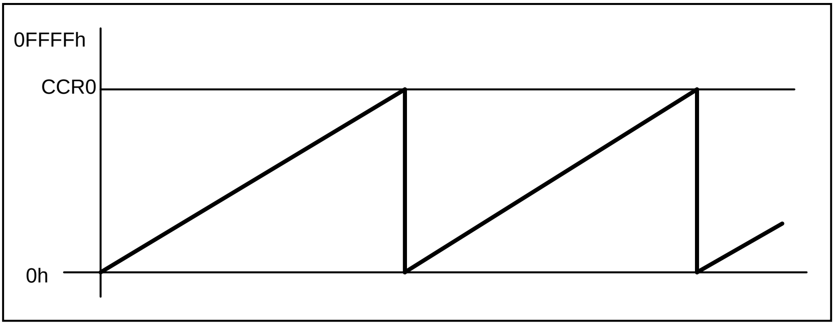
Mode Control:		**Mode**	**Description**
MC1	MC0		
0	0	Stop	Timer is halted
0	1	Up	Timer counts upwards until value is equal to value of Compare Register 0
1	0	Continuous	Timer counts upwards continuously
1	1	Up/Down	Timer counts up until the timer's value is equal to Compare Reg. 0 and then down to zero

Stop Mode

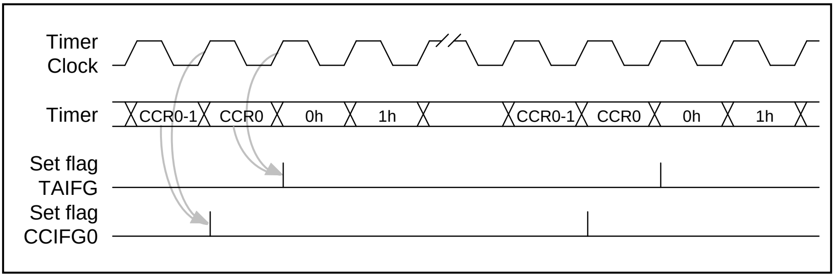
The timer is halted. When released, it counts according to the selected mode, starting from the actual content. The count direction is the same as when stopped. Nothing is reset, the present contents of all registers being used.

UP Mode

The counter counts up to the content of the compare register CCR0. The timer starts counting from the existing value in the timer register. When the timer value and the value of the compare register CCR0 are equal, the timer is reset and restarts counting from zero.



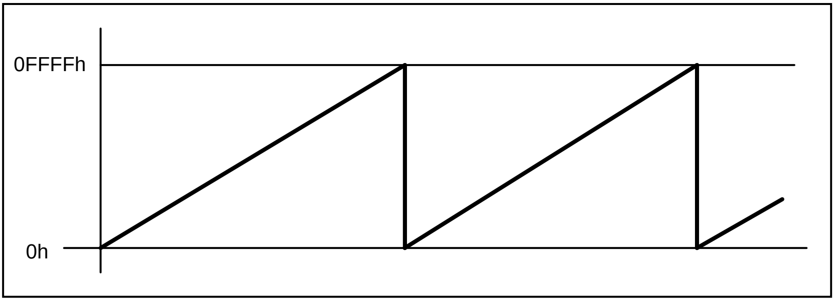
The compare register CCR0 works as the period register in the 'Up Mode'. The counter returns to zero with the next clock when the timer data are equal or greater than the CCR0 data.



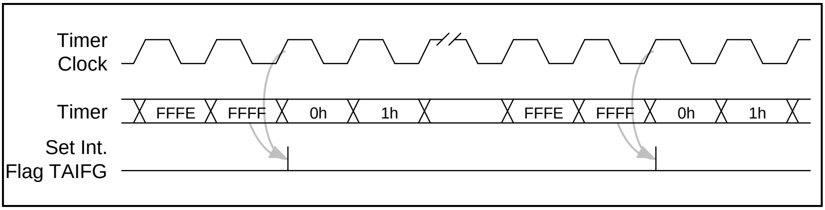
The flag CCIFG0 is set when the timer becomes equal to the CCR0 value. The TAIFG flag is set when the timer counts from CCR0 to zero. All interrupt flags are set independently of the corresponding interrupt enable bit.
An interrupt is requested if the corresponding interrupt enable bit is set and the general interrupt enable bit is set.

Continuous Mode

The timer starts counting from the present value in the timer register. The counter counts up to 0FFFFh and restarts counting from zero.



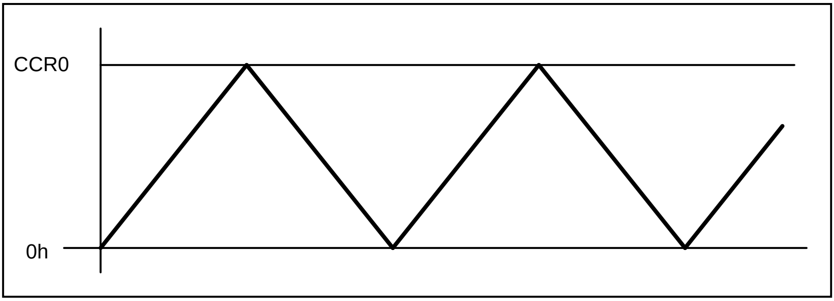
The Continuous Mode is used if more than one timing is needed. The interrupt handler adds to the corresponding compare register CCRx, the time difference from the present time (corresponding data in CCRx), to the time the next interrupt is needed.



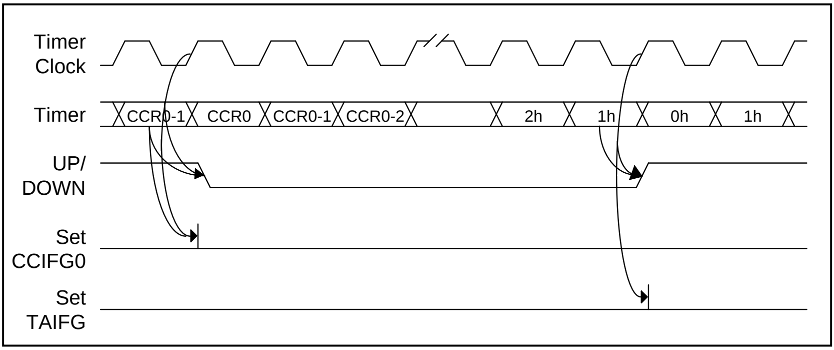
The TAIFG flag is set when the timer counts from 0FFFFh to zero. The interrupt flag is set independently of the corresponding interrupt enable bit. An interrupt is requested if the corresponding interrupt enable bit is set, and the general interrupt enable bit is set.
The capture/compare register CCR0 works the same way as the other compare registers in 'Continuous Mode'.

UP/DOWN Mode

The timer counts up to the content of the compare register CCR0. Then the count direction is reversed, and the timer counts down to zero.



The count direction is latched in a flip-flop FF. The FF is set at 0000h to have the UP condition for the timer, and is reset when the timer value is equal to CCR0, to have the DOWN condition for the timer latched.
The period is defined by the compare register CCR0, and is twice the value in the CCR0 register.



The interrupt flag CCIFG0 is set when the timer has counted up from 'CCR-1' to 'CCR0'.
The interrupt flag TAIFG is set when the timer has counted down from 0001h to 0000h.

The Capture/Compare Block

Five identical blocks provide flexible control of real time processing. Any one of the block registers may be used to capture the timer data at the applied event, or for the generation of time intervals. Each time a capture is done or a time interval is completed, interrupts are generated from the appropriate capture/compare block - if the corresponding interrupt is enabled. The mode bit CAPx in the control word CCTLx selects compare (CAPx is reset) or capture (CAPx is set) operation. The capture mode bits CCMx1 and CCMx0 in the control word CCTLx define under which conditions the capture function is performed - if no capture, capture on the leading edge, the trailing edge or at both edges is executed.

Both the interrupt enable bit CCIEx and the interrupt flag CCIFGx are used for capture and compare modes. The CCIFG is set on a capture or compare event. The control bit CAPx defines if it is used for capture or compare.
The capture inputs CCIxA and CCIxB are connected to external pins or internal signals. Different MSP430 devices will have different signals connected to CCIxA and CCIxB.

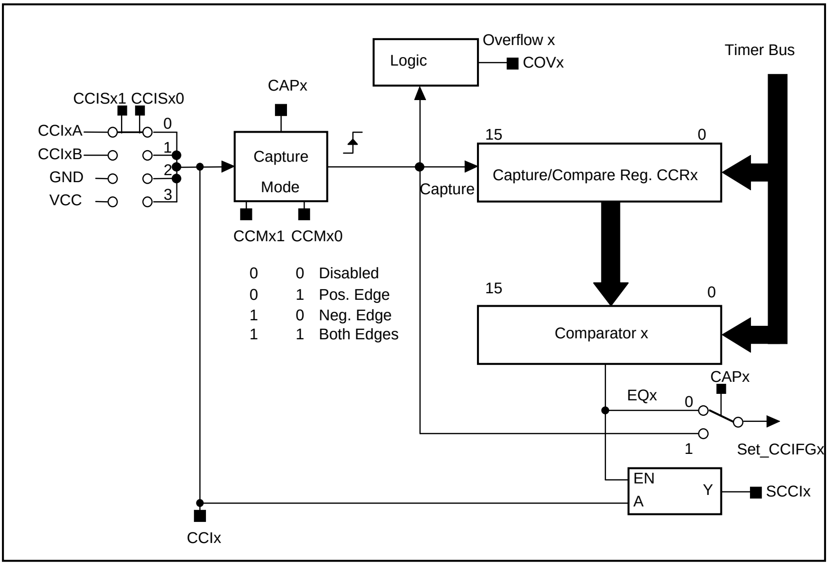
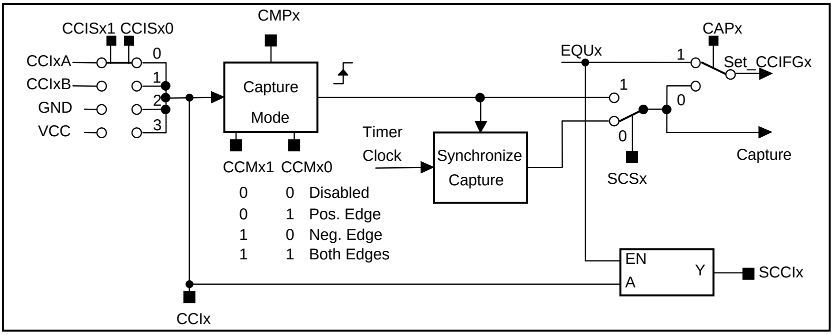
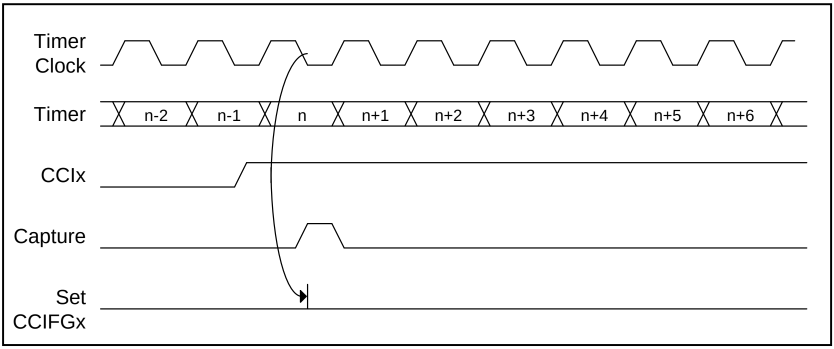


Figure 11.5: Capture/Compare Block

The source of the input signal to the capture logic can be selected by two control bits CCISx1 and CCISx0. It can be read directly by the software via bit CCIx or synchronized with the compare signal EQUx. The synchronized bit SCCIx supports serial protocol software handlers. The capture signal can be asynchronous related to the timer clock. Different application situations are supported by the possibility of using the non-synchronized or the synchronized capture signals.



The capture signal that sets the capture/compare interrupt flag, and stores the timer value into the capture register, is synchronized with the timer clock. It is synchronized to avoid race conditions between the timer data and the capture signal. The synchronized capture signal bit SCSx in the capture/compare control register CCTLx selects the mode of the capture signal.



Applications with slow timer clock are supported using the non-synchronized capture signal. A capture event can have race conditions versus the timer clock, and this results in invalid capture data. The software validates the data and corrects it.

```
; Software example for the handling of asynchronous capture signals
;
; The data of the capture/compare register CCRx are taken by the software
; in the according interrupt routine - they are taken only after a CCRIFG
; was set. The timer clock is much slower than the system clock MCLK
;
CCRx_Int_hand ...                   ; Start of interrupt handler
              ...
              ...
              CMP    &CCRx,&TAR     ; Test if the data CCRX = TAR
              JEQ    Data_Valid
              MOV    &TAR,&CCRx     ; The data in CCRx is wrong,
                                    ; use the timer data
Data_Valid    ...    ...            ; The data in CCRx are valid
              ...
              ...
              RETI
;
```

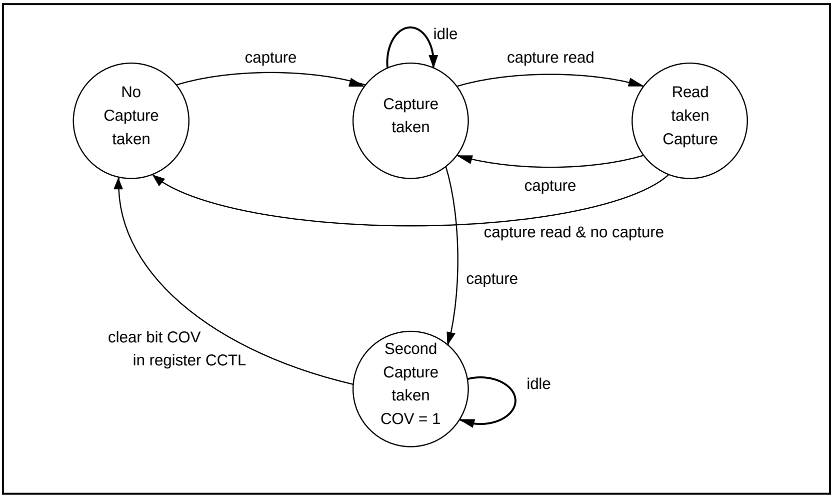
11.1.2 The Capture Mode

The capture mode is selected if the mode bit CAPx - located in the control word CCTLx - is set. The capture mode is used for the accurate fixing of time events. This may be used for speed computations or time measurements. The timer value is copied into the capture register CCRx if the selected edge (positive, negative or both) of the input signal occurs at the selected input pin. Three individual sources can be selected - CCIxA, CCIxB or from the CPU/software via the bits CCISx1/CCISx0 in the capture/compare control register CCTLx.

If a capture was done:

- the interrupt flag CCIFGx - located in the control word CCTLx - is set
- an interrupt is requested, if both interrupt enable bits CCIEx and GIE are set

The capture/compare register CCRx should be accessed with word instructions. It holds the last timer value that was copied to it. An overflow logic is provided. It indicates with its reset state that the capture data were taken before another sub-sequential capture was done. The overflow bit COVx in the register CCTLx is set when a second capture data is latched before the capture/compare register was read successfully. This allows activities for getting back into the lost synchronization.
The capture taken event is reset only if the captured data are completely read before another capture occurred. The overflow bit is set if the read operation is not completed.



The overflow bit COVx needs to be reset by software.

```
; Software example for the handling of captured data looking for overflow
; condition
;
; The data of the capture/compare register CCRx are taken by the software
; and immediately with the next instruction the overflow bit is tested
; and a decision is made to proceed regularly or with an error handler
;
CCRx_Int_hand ...                 ; Start of handler  Interrupt
              ...
              ...
              MOV    &CCRx,RAM_Buffer
              BIT    #COV,&CCTLx
              JNZ    Overflow_Hand
              ...
              ...
              ...
              RETI
Overflow_Hand BIC    #COV,&CCTLx   ; reset capture overflow flag
                                  ; get back to lost synchronization
              ...
              ...
;             RETI
```

Note: Capture with Timer halted

Capture should be stopped when the timer is halted. The sequence should be: stop capturing, and then stop the timer. When the capture function is restarted the sequence should be: start capturing, and then start the timer.

11.1.3 The Compare Mode

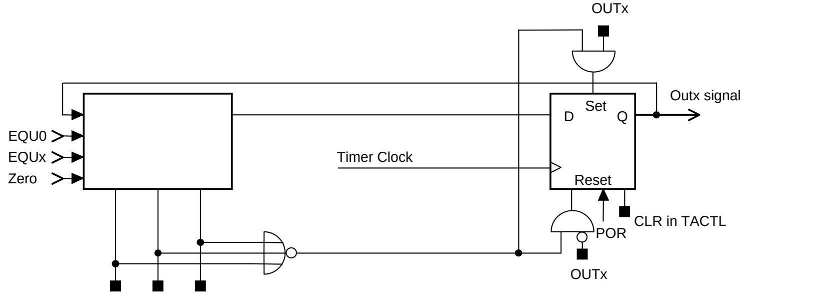
The compare mode is selected if bit CAPx is reset. The bit CAPx is located in the control word CCTLx. All circuitry of the capture hardware is inactive. If the timer becomes equal to the value in the Compare Register x then:

- the Interrupt Flag CCIFGx located in the control word CCTLx is set
- interrupt is requested if the Interrupt Enable Bit CCIEx and GIE bit are set
- the signal EQUx is output to the output unit OUx. Depending on the selected output mode this signal sets, resets or toggles the output OUTx (if OUTMODx > 0).

The capture/compare register CCRx should be accessed with word instructions. It holds the compare value that was written to it. The overflow logic provided for capture mode is inactive.
The EQU0 signal is true when the timer value is greater or equal to the CCR0 value. The EQU1 to EQU4 signals are true when the timer value is equal to the corresponding CCR1 to CCR4 value.

11.1.4 The Output Unit

The output unit supports applications that uses PWM or Digital-to-Analog conversion (DAC). The outputs EQU0 and EQUx of the capture/compare registers control the output logic according to the selected function by three control bits. Five output units OU0 to OU4 - one for each capture/compare block - are implemented. The control bits OMx0, OMx1 and OMx2 are located in the Control Register CCTLx.



OMx2	OMx1	OMx0	
0	0	0	Output Mode, Outx signal is set according to Outx bit
0	0	1	Set, EQUx sets Outx signal clock synchron with timer clock
0	1	0	PWM Toggle/Reset, EQUx toggles Outx signal, reset with EQU0, clock sync. with timer clock
0	1	1	PWM Set/Reset, EQUx sets Outx signal, reset with EQU0, clock synchron with timer clock
1	0	0	Toggle, EQUx toggles Outx signal, clock synchron with timer clock
1	0	1	Reset, EQUx resets Outx signal clock synchron with timer clock
1	1	0	PWM Toggle/Reset, EQUx toggles Outx signal, set with EQU0, clock synchron with timer clock
1	1	1	PWM set/Reset, EQUx reset Outx signal, set with EQU0, clock synchron with timer clock

Figure 11.6: Output Unit

The control bit OUTx determines the Outx signal if the output mode 0 is selected by OMx0, OMx1 and OMx2. The output signal starts with the actual level independent of the selected mode.

UP Mode

The Outx signal is changed when the timer counts up to CCRx, and when the timer counts from CCR0 to zero. The Outx signal is modified according to the selected output mode.

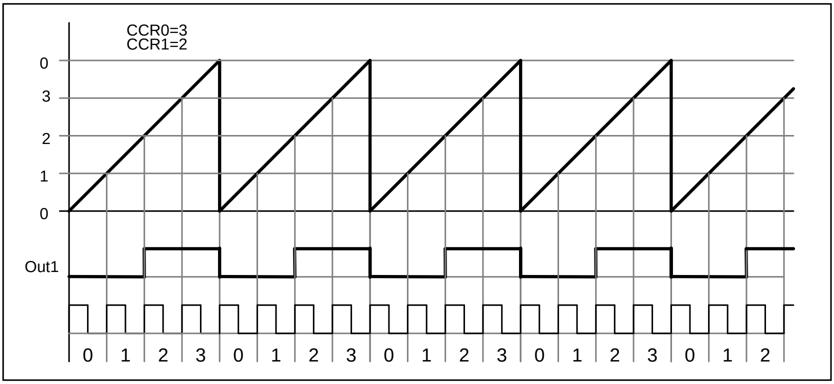


Figure 11.7: Output Unit: Example Up-Mode and Output Mode 3

Continuous Mode

The Outx signal is changed when the timer counts up to CCRx and when the timer counts up to CCR0. The Outx signal is modified according to the selected output mode.

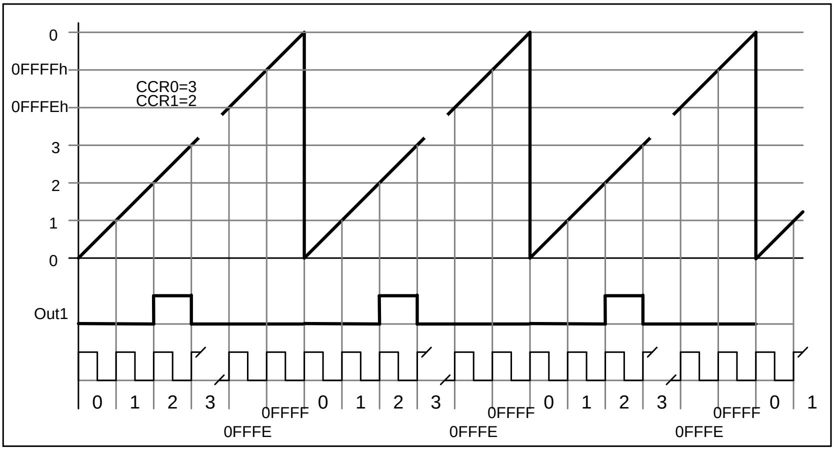


Figure 11.8: Output Unit: Example Continuous Mode and Output Mode 3

UP/DOWN Mode

The Outx signal is changed when the timer counts up to CCRx, and when the timer counts down to CCRx. The Outx signal is modified according to the selected output mode.

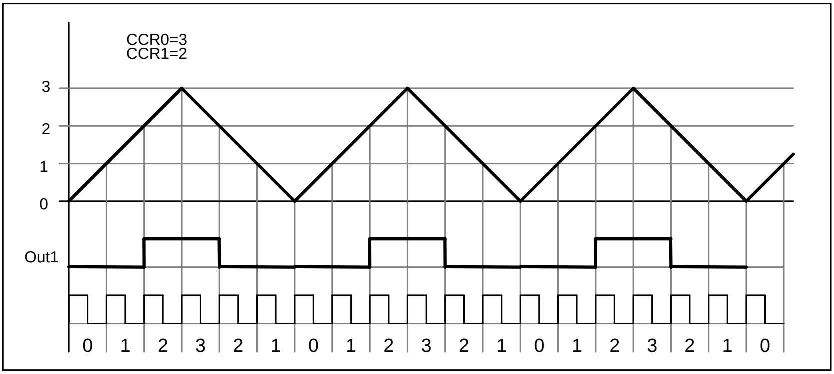


Figure 11.9: Output Unit: Example Up/Down Mode and Output Mode 4

11.2 Registers of Timer_A

The 16-bit Timer_A module hardware is word structured and should be accessed by word processing instructions.

Register	short form	Register type	Address	Initial state
● Timer_A control register	TACTL	Type of read/write	160h	POR reset
● Timer_A register	TAR	Type of read/write	170h	POR reset
● Cap/Com control register0	CCTL0	Type of read/write	162h	POR reset
● Capture/Compare register0	CCR0	Type of read/write	172h	POR reset
● Cap/Com control register1	CCTL1	Type of read/write	164h	POR reset
● Capture/Compare register1	CCR1	Type of read/write	174h	POR reset
● Cap/Com control register2	CCTL2	Type of read/write	166h	POR reset
● Capture/Compare register2	CCR2	Type of read/write	176h	POR reset
● Cap/Com control register3	CCTL3	Type of read/write	168h	POR reset
● Capture/Compare register3	CCR3	Type of read/write	178h	POR reset
● Cap/Com control register4	CCTL4	Type of read/write	16Ah	POR reset
● Capture/Compare register4	CCR4	Type of read/write	17Ah	POR reset
● Interrupt Vector register	TAIV	Type of read	12Eh	(POR reset)

The addresses 16Ch, 16Eh, 17Ch and 17Eh are reserved for future extensions.

11.2.1 Timer_A Control Register TACTL

All control bits regarding the timer and its operation are located in the timer control register TACTL. All control bits are reset automatically by the POR signal, but PUC does not affect them. The control register should be accessed with word instructions.

TACTL

160h

15															0
unused					Input Select			Input Divider		Mode Control		un-used	CLR	TA-IE	TA-IFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	(w)-(0)	rw-(0)	rw-(0)

Bit 0: TAIFG: This flag indicates a timer overflow event.

UP mode: TAIFG is set if the timer counts from CCR0 value to 0000h.

Continuous mode: TAIFG is set if the timer counts from 0FFFFh to 0000h.

UP/DOWN mode: TAIFG is set if the timer counts down to 0000h.

Bit 1: Timer Overflow Interrupt Enable TAIE bit. An interrupt request from the timer overflow bit is enabled if set, and it is disabled if reset.

Bit 2: Timer Clear CLR bit. The timer and the input divider are reset after POR, or if bit CLR is set. The CLR bit is automatically reset by the hardware and always read as zero. The timer starts operation with the next valid input edge. The timer starts in an upward direction if it is not halted by cleared mode control bits.

Bit 3: Not used

Bit 4 to 5: Mode Control Description

MC1	MC0	Count Mode	Comment, Timer ...
0	0	Stop	is halted
0	1	Up to CCR0	counts up to CCR0 and restarts at 0
1	0	Cont. Up	counts up cont. all 65536 steps
1	1	Up/Down	counts up to CCR0, down to 0,.....

Bit 6 to 7: Input Divider control bits

ID1	ID0	Operation	Comment
0	0	Pass	Input signal is passed to the timer
0	1	/2	Input signal is divided by two
1	0	/4	Input signal is divided by four
1	1	/8	Input signal is divided by eight

Bit 8 to 10: Select source of timer input clock signal - preprocessed in the Input Divider

SSEL2	SSEL1	SSEL0	O/P signal	Comment
0	0	1	TACLK	The signal at dedicated ext. pin is used
0	0	1	ACLK	Auxillary clock ACLK is used
0	1	0	MCLK	System clock MCLK is used
0	1	1	INCLK	See device description
1	X	X	-----	Reserved

Bit 11 to 15: Unused

Note: Modify Timer_A

Any write to the timer register TAR when it is operating and ACLK or external clock TACLK is selected can result in unpredictable results. The asynchronous clocks - MCLK used by the CPU and the timer clock can have critical race conditions.

Note: Changing of Timer_A Control bits

If the operation of the timer is modified by the control bits in the TACTL control register, the timer should be halted during this modification. The critical modifications are the input select bits, the input divider bits, and the timer clear bit. Asynchronous input clock situations and system clock (used by the software) can get into race conditions were the timer reacts falsely.
The recommended instruction flow is:

1. Modify the control register and stop the timer.
2. Start the timer operation.

```
E.G.:  MOV   #0286h,&TACTL        ; ACLK/8, timer stopped, timer cleared
       BIS   #10h,&TACTL          ; Start timer with continuous up mode
```

11.2.2 Capture/Compare Control Register CCTL

Each Capture/Compare block has its own control word CCTLx.

	15															0
CCTLx	CAPTURE MODE		INPUT SELECT		SCS	SCCI	un-used	CAP	OUTMODx			CCIE	CCI	OUT	COV	CCIFG
162h to 16Eh	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r	rw-(0)	rw-(0)	rw-(0)

POR resets all bits of CCTLx, PUC does not affect them.

Bit 0: Capture/compare interrupt flag CCIFGx.

Capture Mode: If set, it indicates a timer value was captured in the register CCRx.

Compare Mode:If set, it indicates timer value was equal to the data in the compare register CCRx.

CCIFG0 flag:

CCIFG0 is automatically reset when the interrupt request was accepted according to the interrupt scheme of the MSP430 family.

CCIFG1 to CCIFG4 flags:

The flag which determines the actual interrupt vector word is automatically reset after the TAIV word is read. No vector word is generated if the interrupt enable bit is reset but the flag may be set independently. The flags CCIFG1 to CCIFG4 need to be reset by software.

Bit 1: Capture overflow flag COV.

Compare mode selected, CAP = 0:

The capture signal generation is reset. No capture event will set COV bit.

Capture mode selected, CAP = 1:

The overflow flag COV is set if a second capture is done before the capture register is read. The overflow bit supports software to detect a second capture operation before the previous data are read from capture register. The overflow flag is not reset by reading the capture register.

Bit 2: The OUTx bit is at the corresponding output if OUTMODx is 0 (output only mode).

Bit3: Capture/Compare Input Signal CCIx:
Capture Mode: The selected input signal (CCIxA, CCIxB, VCC or GND) can be read.
Compare Mode: CCI is reset

Bit 4: Interrupt Enable CCIEx: Enables or disables the interrupt request signal of capture/compare block x. Interrupt request is active if enable bit is set, the flag CCIFGx is set and GIE is set.
0: Interrupt disabled 1: Interrupt enabled

Bit 5 to 7:

Output Mode		Description
0	Output only	Data of OUTx bit determines Outx signal.
1	Set	Comp. signal EQUx sets Outx signal
2	PWM Toggle/Reset	Comp. signal EQUx toggles Outx signal, EQU0 resets Outx signal
3	PWM Set/Reset	Comp. signal EQUx sets Outx signal, EQU0 resets Outx signal
4	Toggle	Comp. signal EQUx toggles Outx signal
5	Reset	Comp. signal EQUx resets Outx signal
6	PWM Toggle/Set	Comp. signal EQUx toggles Outx signal, EQU0 sets Outx signal
7	PWM Reset/Set	Comp. signal EQUx resets Outx signal, EQU0 sets Outx signal

Bit 8: CAP: Defines if the capture/compare block and associated interrupt block acts in capture or compare function.
0: Compare Mode 1: Capture Mode

Bit 9: read only, always read as 0.

Bit 10: Capture/Compare Input Signal SCCIx, synchronized with the compare output EQUx:
The selected input signal (CCIxA, CCIxB, VCC or GND) is stored into a transparent latch with the comparator's equal signal EQUx and can be read.

Bit 11: The capture/compare signal can used in asynchronous mode or synchronized to the timer clock.
The asynchronous mode (SCS is reset) allows to set the CCIFG immediately on request and also capture the timer data immediately. It will be useful if the period of the capture source is far slower than the timer clock. The data in the capture register may be wrong if race conditions of timer clock and capture source occur.
The synchronous mode (SCS is set) is normally used and the capture data are always valid.
0: asynchronous capture 1: synchronous capture

Bit 12 to 13: Input Select, CCIS1 and CCIS0.

These two bits define the source which provides the capture event in capture mode. During compare mode there is no use of these control bits.

0	Input CCIxA is selected
1	Input CCIxB is selected
2	GND, Low
3	VCC, High

Bit 14 to 15:

Capture Mode		Description
0	Disabled	The capture mode is disabled
1	Pos. Edge	Capture is done with rising edge
2	Neg. Edge	Capture is done with falling edge
3	Both Edges	Capture is done with rising and falling edge

Note: Simultaneous capture and capture mode selection

If the operation of the capture/compare block is modified by the capture/compare bit CAP in the CCRx register from compare to capture mode, no capture should be done simultaneously. The result in the capture/compare register is unpredictable.
The recommended instruction flow is:

1. Modify the control register to switch from compare to capture.
2. Capture.

```
E.G.:  BIS    #CAP,&CCTL2      ; Select capture with register CCR2

       XOR    #CCIS1,&CCTL2    ; Software capture:  CCIS0 = 0

                               ;                    Capture Mode = 3
```

11.2.3 Timer_A Interrupt Vector Register

Two interrupt vectors are associated with the 16-bit Timer_A module:

- The vector for the capture/compare register CCR0 has the highest priority of all Timer_A interrupts. The capture/compare register CCR0 can be used to define the period during the UP-Mode and the UP/DOWN-Mode. It therefore needs the fastest service.
- The multiplexed vector for the other capture/compare registers. A 16-bit vector word TAIV indicates the currently highest interrupt.

CCR0 Interrupt vector
The interrupt flag associated with the capture/compare register CCR0 is set if the timer value is equal to the compare register's value.

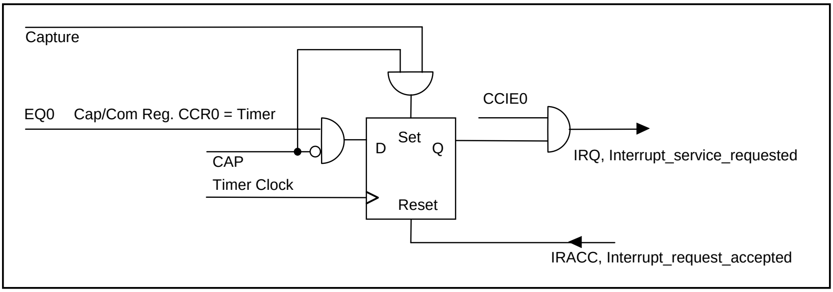

Capture
EQ0
Cap/Com Reg. CCR0 = Timer
CAP
Timer Clock
D
Set
Q
Reset
CCIE0
IRQ, Interrupt_service_requested
IRACC, Interrupt_request_accepted

Figure 11.10: Capture/Compare Interrupt Flag

The capture/compare register 0 has the highest interrupt priority, and uses its own interrupt vector to speed up the real time processing.

Vector word, TAIFG, CCIFG1 to CCIFG4 flags
A vector word is associated with the TAIFG flag and each of the other four capture/compare registers CCR1 to CCR4, and is additionally combined with a priority scheme: the flag CCIFGx with the highest priority generates a number from 0 (no flag set) to 12. This encoded number can be added to the program counter to enter the associated software according to the corresponding interrupt. The vector word TAIV is a 16-bit word to be added to the program counter (see also SW example).

Reading the actual vector word TAIV from the vector word register resets the flag CCIFGx that defines the current vector word.

	15															0
TAIV 12Eh	0	0	0	0	0	0	0	0	0	0	0	Interrupt vector				0
	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r-(0)	r-(0)	r-(0)	r-(0)	r0

Interrupt Priority	Interrupt Source	Short form	Vector Address	Vector Register TAIV Contents
Highest	Capture/Compare 0	CCIFG0	X	N.A.
	Capture/Compare 1	CCIFG1	Y	2
	Capture/Compare 2	CCIFG2	Y	4
	Capture/Compare 3	CCIFG3	Y	6
	Capture/Compare 4	CCIFG4	Y	8
	Timer Overflow	TAIFG	Y	10
Lowest	Reserved		Y	12
	No interrupt pending		Y	0

An interrupt from the timer is requested by setting of CCIFGx or TAIFG, if CCIEx or TAIE is set, and the general interrupt enable bit GIE is set. The bit with the highest priority is requesting the service. When the timer vector word TAIV was accessed the interrupt service requesting bit (CCIFGx or TAIFG) is reset automatically. The bit with the next lower priority now defines the timer vector word TAIV. An interrupt is also requested immediately if any interrupt enable bit (CCIEx or TAIE) is set and the corresponding interrupt flag was already set.

All interrupt flags CCIFGx and TAIFG are featured with full access by the CPU.

Note: Writing to read only register TAIV

When a write to the vector word register TAIV is done the actual interrupt flag that determines the vector word is reset. The requesting interrupt event is missed for later software handling. Additionally, writing to this read only register results in an increased current consumption as long as the write is active.

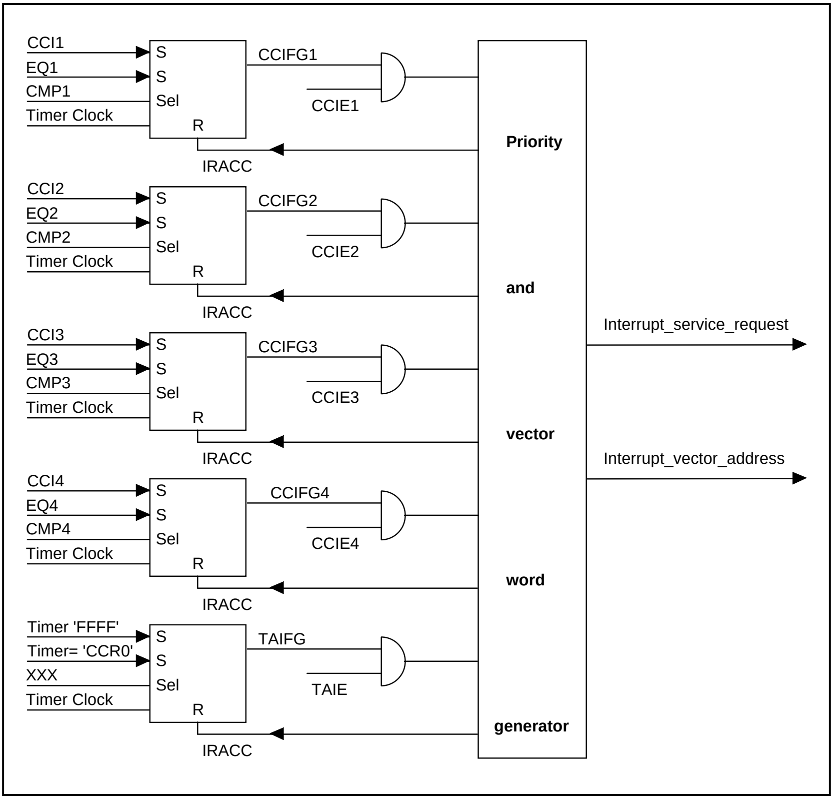


Figure 11.11: Schematic of Capture/Compare Interrupt Vector Word

Timer Interrupt Vector Register, Software Example

The software example shows the use of the vector word TAIV and the overhead of the handling. The numbers at the right margin show the necessary cycles for every instruction. The example is written for continuous mode: the time difference to the next interrupt is added to the corresponding compare register.

```
; Software example for the interrupt part                      Cycles
;
; Interrupt handler for Capture/Compare Module 0.
; The interrupt flag CCIFG0 is reset automatically
;
TIMMOD0  ...                ; Start of handler  Interrupt latency   6
             RETI                                                   5
```

```
;
; Interrupt handler for Capture/Compare Modules 1 to 4.
; The interrupt flags CCIFGx and TAIFG are reset by hardware
; Only the flag with the highest priority responsible for the
; interrupt vector word is reset.
TIM_HND       $                          ; Interrupt latency       6
              ADD    &TAIV,PC            ; Add offset to Jump table 3
              RETI                       ; Vector 0: No interrupt   5
              JMP    TIMMOD1             ; Vector 2: Module 1       2
              JMP    TIMMOD2             ; Vector 4: Module 2       2
              JMP    TIMMOD3             ; Vector 6: Module 3       2
              JMP    TIMMOD4             ; Vector 8: Module 4       2
;
; Module 5. Timer Overflow Handler: the Timer Register is
; expanded into the RAM location TIMEXT (MSBs)
;
TIMOVH                                   ; Vector 12: TIMOV Flag
              INC    TIMEXT       ; Handle Timer Overflow           4
              RETI                                                  5
;
TIMMOD2                                  ; Vector 4: Module 2
              ADD    #NN,&CCR2           ; Add time difference      5
              ...                        ; Task starts here
              RETI                       ; Back to main program     5
;
;
TIMMOD1                                  ; Vector 2: Module 1
              ADD    #MM,&CCR1           ; Add time difference      5
              ...                        ; Task starts here
              RETI                       ; Back to main program     5

; The Module 3 handler shows a way to look if any other interrupt is
; pending: 5 cycles have to be spent, but 9 cycles may be saved if
; another interrupt is pending
;
TIMMOD3                                  ; Vector 6: Module 3
              ADD    #PP,&CCR3           ; Add time difference      5
              ...                        ; Task starts here
              JMP    TIM_HND             ; Look for pending intrpts 2
;
              .SECT  "VECTORS",0FFF0h    ; Interrupt Vectors

              .WORD  TIM_HND      ; Vector for Capture/Compare Module 1..4
                                  ; and timer overflow TAIFG
              .WORD  TIMMOD0      ; Vector for Capture/Compare Module 0
```

If the FLL was turned off, then 2 additional cycles need to be added for synchronous start of CPU system and system clock MCLK.
The software overhead for the different interrupt sources includes the interrupt latency and return-from-interrupt cycles (but not the task handling itself):

• Capture/Compare block CCR0	11 cycles
• Capture/Compare blocks CCR1 to CCR4	16 cycles
• Timer Overflow TAIFG	14 cycles

Timing Limits

With the TAIV register and the above software, the shortest repetitive time distance t_{CRmin} between two events using a Compare Register is:

$$t_{CRmin} = t_{taskmax} + 16 \text{ x } t_{cycle}$$

with: $t_{taskmax}$ Maximum (worst case) time for the task to be done during the interrupt routine (e.g. incrementing of a counter)

t_{tcycle} Cycle time of the used system frequency MCLK

The shortest repetitive time distance t_{CLmin} between two events using a capture register is:

$$t_{CLmin} = t_{taskmax} + 16 \text{ x } t_{cycle}$$

11.3 Timer_A in Applications

11.3.1 Timer_A - Use of the UP-Mode

The UP-Mode is used if the period of the timer should be different to 65,536 clock cycles, which is the period in continuous mode. The capture/compare register CCR0 data is used to define the period of the timer.

Capabilities of output unit OU0
The output unit OU0 works usefully with four modes since CCR0 is also used to define the period of the timer. The four modes are output mode 0, output mode1, output mode 4 and output mode 5. The other four modes can not be used, since they use the EQU0 signal simultaneously in different ways.

Capabilities of output units OU1 to OU4
The output units OU1 to OU4 and its driving circuits are fully identical - all four have the same characteristics. Each can operate in the same or a different way.

The mix - to generate signals or to capture timer data - is selected and controlled by the application software. Examples of the different output mode basic functions are illustrated in the figure. The examples use output OUT1 for demonstration purpose.

Timer: The timer repeatedly runs from 0 up to the value of CCR0.

Output mode 0: The output signal OUTx is defined by the OUTx bit in the control register CCTLx of each capture/compare block, independently of any timing function and completely under software control.

Output mode 1: The output is set when the timer value becomes equal to the capture/compare data CCR1. The interrupt caused by the EQU0 signal (CCIFG0) may be used for modifications of the Compare Registers x.

Output mode 2: The output is toggled when the timer value becomes equal to the capture/compare data CCR1. It is reset when timer value is equal to CCR0 - timer is reset too. This is basically used for PWM functions or together with other outputs to generate phase relations.

Output mode 3: The output is set when the timer value becomes equal to the capture/compare data CCR1. It is reset when timer value is equal to CCR0 - timer is reset too. This is basically used for PWM functions or together with other outputs to generate phase relations.

Output mode 4: The output is toggled when the timer value becomes equal to the capture/compare data CCR1. The output period is double the period of the timer’s period. The phase relation to any other output is determined by selecting the CCRx data.

Output mode 5: The output is reset when the timer value becomes equal to the capture/compare data CCR1. The interrupt caused by the EQU0 signal (CINT0) may be used for modifications of the Compare Registers x.

Output mode 6: The output is toggled when the timer value becomes equal to the capture/compare data CCR1. It is set when timer value becomes equal to CCR0. This is basically used for PWM functions or together with other outputs to generate phase relations.

Output mode 7: The output is reset when the timer value becomes equal to the capture/compare data CCR1. It is set when timer value becomes equal to CCR0 - timer is reset. This is basically used for PWM functions, or together with other outputs to generate phase relations.

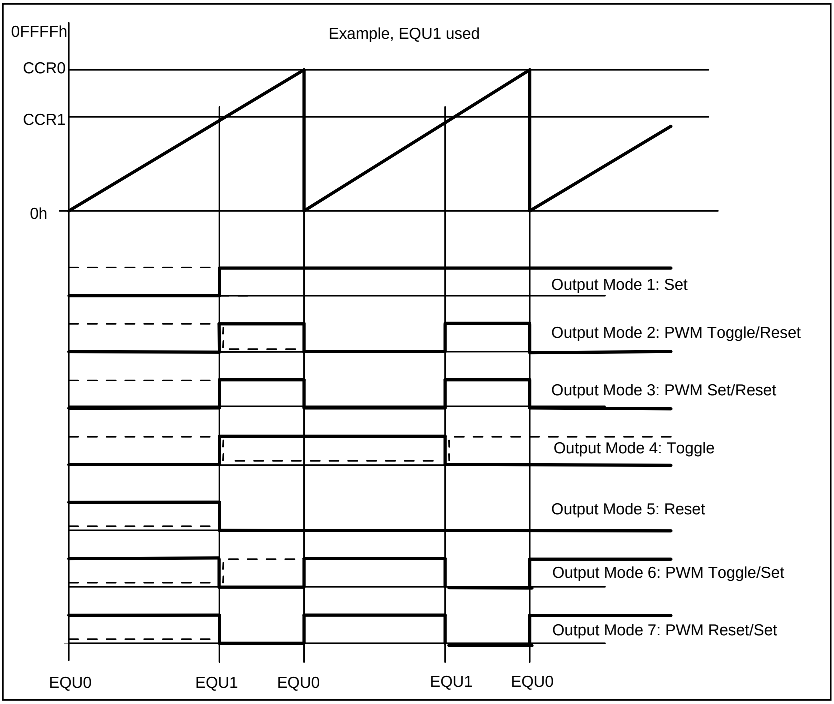

0FFFFh
Example, EQU1 used
CCR0
CCR1
0h
Output Mode 1: Set
Output Mode 2: PWM Toggle/Reset
Output Mode 3: PWM Set/Reset
Output Mode 4: Toggle
Output Mode 5: Reset
Output Mode 6: PWM Toggle/Set
Output Mode 7: PWM Reset/Set
EQU0
EQU1
EQU0
EQU1
EQU0

Figure 11.12: Output Unit in Up Mode

11.3.2 Timer_A - Use of the Continuous Mode

The continuous mode is used if the period of the timer of 65,536 clock cycles is insignificant for the application. A main application of the continuous mode is the generation oft independent software timings. The capture/compare register CCR0 data is used the same way like the other four capture/compare registers CCRx.

All output modes will be useful for various kinds of applications. The feasible output signals for the output modes are chosen by the output mode bits OMx2 to OMx0 in the CCTLx register.

The mix - to generate signals or to capture timer data - is selected and controlled by the application software. Examples of the different output mode basic functions are illustrated in the succeeding figure. The outputs OUT0 and OUT1 are used for demonstration purposes only. The data in CCR0 are greater than the data in CCR1.

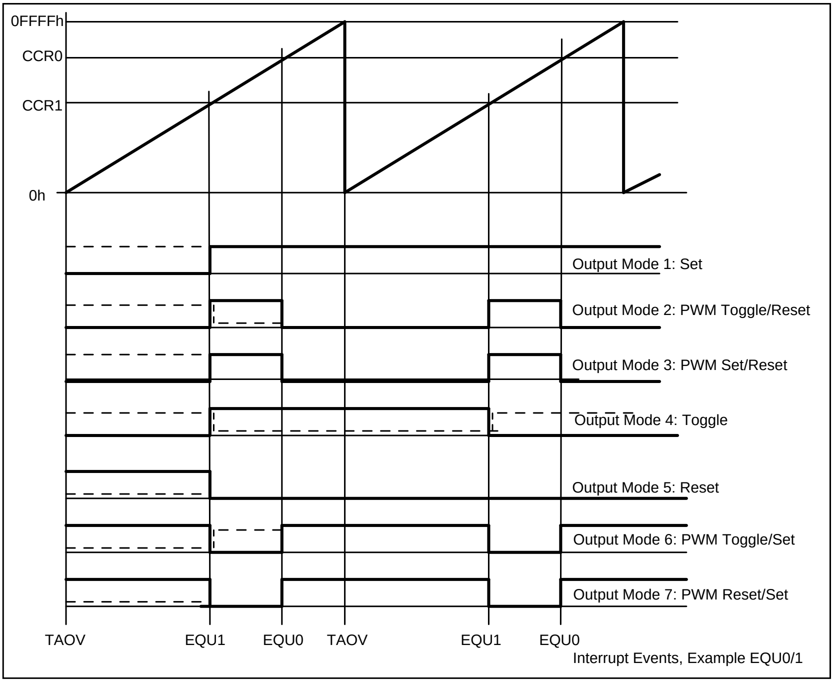

0FFFFh
CCR0
CCR1
0h
Output Mode 1: Set
Output Mode 2: PWM Toggle/Reset
Output Mode 3: PWM Set/Reset
Output Mode 4: Toggle
Output Mode 5: Reset
Output Mode 6: PWM Toggle/Set
Output Mode 7: PWM Reset/Set
TAOV
EQU1
EQU0
TAOV
EQU1
EQU0
Interrupt Events, Example EQU0/1

Figure 11.13: Output Unit in Continuous Mode

Timer: The timer repeatedly runs from 0 up to FFFF.

Output mode 0: The output signal OUTx is defined by the OUTx bit in the control register CCTLx of each capture/compare block, independently of any timing function, and completely under software control.

Output mode 1: The output is set when the timer value becomes equal to the capture/compare data CCR1. The interrupt caused by the EQU0 signal (CCIFG0) may be used for modifications of the Compare Registers x.

Output mode 2: The output is toggled when the timer value becomes equal to the capture/compare data CCR1. It is reset when the timer value is equal to CCR0. This is basically used for pulse generation.

Output mode 3: The output is set when the timer value becomes equal to the capture/compare data CCR1. It is reset when the timer value is equal to CCR0. This is basically used for pulse generation.

Output mode 4: The output is toggled when the timer value becomes equal to the capture/compare data CCR1. The output period is double the period of the timer's period. The phase relation to any other output is determined by selecting the CCRx data.

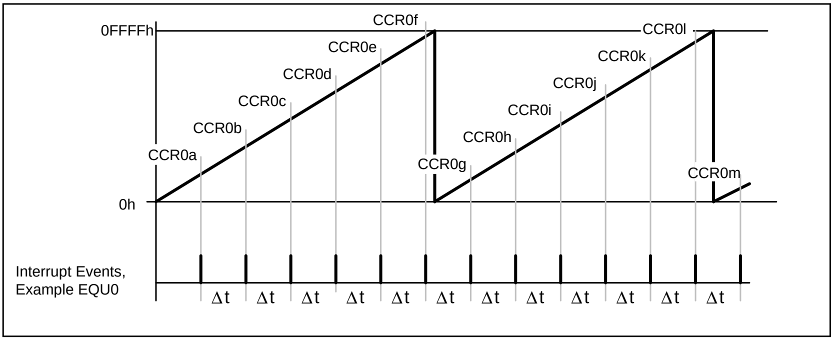
Output mode 5: The output is reset when the timer value becomes equal to the capture/compare data CCR1. The interrupt (CCIFG0) caused by the EQU0 signal may be used for modifications of the Compare Registers x.

Output mode 6: The output is toggled when the timer value becomes equal to the capture/compare data CCR1. It is set when the timer value is equal to CCR0. This is basically used for pulse generation.

Output mode 7: The output is reset when the timer value becomes equal to the capture/compare data CCR1. It is set when the timer value is equal to CCR0. This is basically used for pulse generation.

Continuous Mode - used for time intervals

The continuous mode can be used to generate easily time intervals for the application software. Each time the interval is completed, an interrupt is generated if enabled. In the interrupt routine of this event, the time distance to the next event is added to the capture/compare register CCRx used for this function. Up to five completely independent time events can be generated using all five capture/compare blocks.



Time intervals can be done also with the other modes were CCR0 is used as the period register. There handling is more complex since the sum of the old CCRx data and the new period can be higher than the CCR0 register. When the sum CCRxold plus Δt is greater than CRR0 data, the sum must be reduced by CCR0 data for correct time interval.

11.3.3 Timer_A - Use of the UP/DOWN Mode

The UP/DOWN mode is used if the period of the timer should be different to 65,536 clock cycles and symmetrical pulse waveform generation is needed. The capture/compare register CCR0 data is used to define the period of the timer. The period of the timer is twice the data contained in the CCR0.

Capabilities of output unit OU0
The capture/compare register is used to define the period of the timer. The output unit OU0 only operates effectively in the output mode 0, 1, 4 and 5. All other modes fail, since the timer is already controlled by the CCR0 equal signal EQU0.

Capabilities of output units OU1 to OU4
The output units OU1 to OU4 and its driving circuits are fully identical - all four have the same functions and can operate in different modes.

The mix - to generate signals or to capture timer data - is selected and controlled by the application software. Examples of the different output mode basic functions are illustrated in the succeeding figure. Output OUT3 is used for demonstration purpose only.

Two interrupts are generated during continuous running in the UP/DOWN mode - the interrupt from the capture/compare block CCR0 and the interrupt from the timer, when timer is in down phase and reaches zero. Both interrupts can be used to run proper output pulse modification.

11

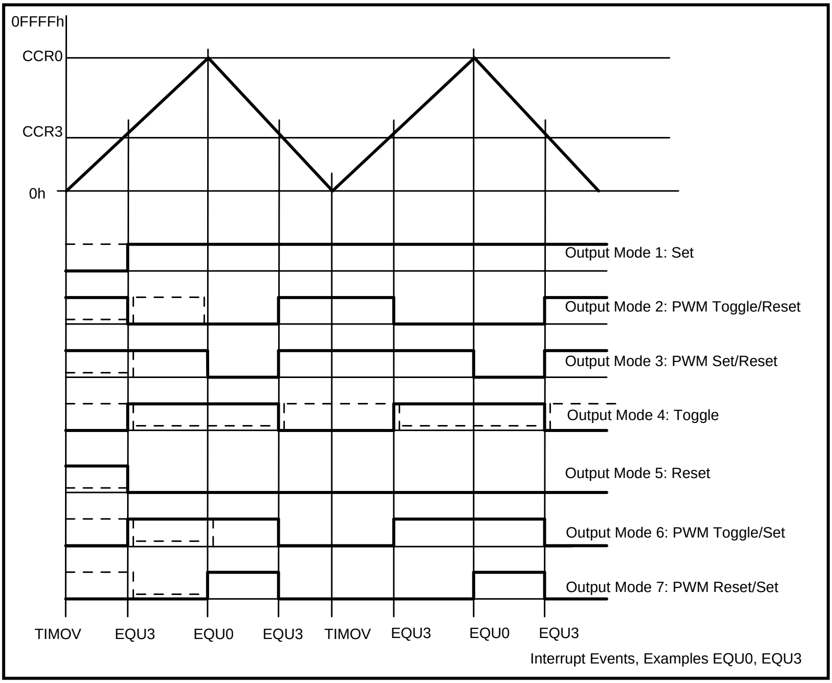

0FFFFh
CCR0
CCR3
0h
Output Mode 1: Set
Output Mode 2: PWM Toggle/Reset
Output Mode 3: PWM Set/Reset
Output Mode 4: Toggle
Output Mode 5: Reset
Output Mode 6: PWM Toggle/Set
Output Mode 7: PWM Reset/Set
TIMOV
EQU3
EQU0
EQU3
TIMOV
EQU3
EQU0
EQU3
Interrupt Events, Examples EQU0, EQU3

Figure 11.14: Output Unit in UP/DOWN Mode(I)

The UP/DOWN mode makes applications possible that enforce the use of "Dead Times" between the output signals. For example, two outputs driving an H-bridge must never be high simultaneously to avoid overload conditions. For a short programmable time - the dead time - both outputs are switched to low. Also the reverse situation is applicable - if necessary the two outputs may be programmed to be never low simultaneously. In the example the t_{dead} is:

$$t_{dead} = t_{timer} \times (CCR1 - CCR3)$$

with:

with:	t_{dead}	Time that both outputs need to be low
	t_{timer}	Cycle length of the Timer Register input frequency
	CCRx	Content of the Compare Register x

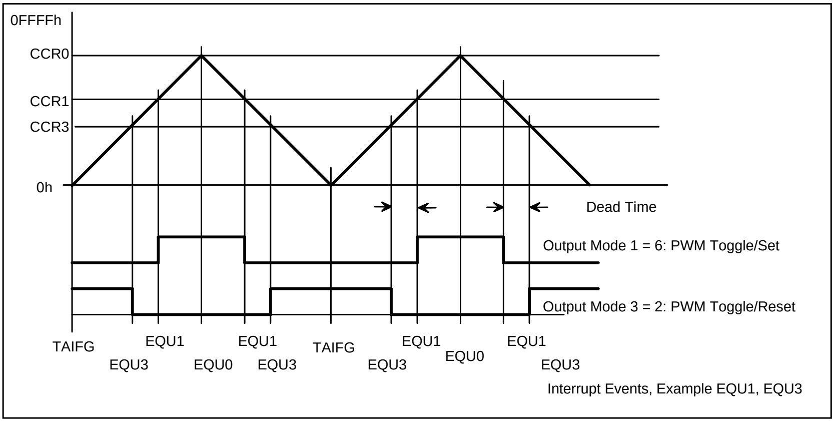

0FFFFh
CCR0
CCR1
CCR3
0h
Dead Time
Output Mode 1 = 6: PWM Toggle/Set
Output Mode 3 = 2: PWM Toggle/Reset
TAIFG
EQU3
EQU1
EQU0
EQU1
EQU3
TAIFG
EQU3
EQU1
EQU0
EQU1
EQU3
Interrupt Events, Example EQU1, EQU3

Figure 11.15: Output Unit in UP/DOWN Mode (II)

11.3.4 Timer_A - Capture via Software

Each of the capture/compare registers can be used by the software to get a time stamp. It can be used for various purposes:

- measure time used by software routines
- measure time between hardware events
- measure the system frequency
- ....

The two bits CCISx1 and CCISx0 and the capture mode selected by the two bits CCMx1 and CCMx0 are use to realize the capture performed by software. The capture mode can be selected to act on the positive edge, negative edge or both edges of the capture signal CCIx. The simplest realization is done when the capture mode is selected to capture on both edges. The capture input signal is selected to be VCC/high or GND/Low. The bit CCISx1 is set and with the bit CCISx0 the capture signal VCC/high or GND/Low is selected.

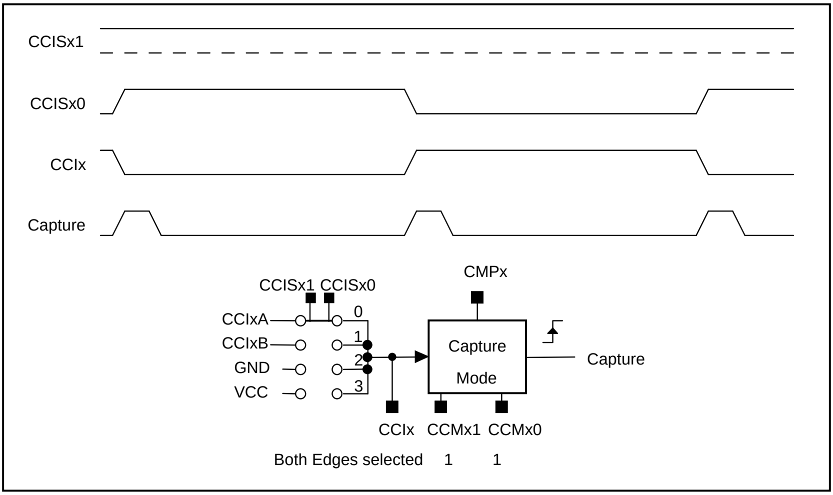


Figure 11.16: Software Capture Example

```
; Software example to capture data performed by software
;
; The data of the capture/compare register CCRx are taken by the software
; It is assumed that CCMx1, CCMxO and CCISx1 bits are set.
; The bit CCISx0 selects the CCIx signal to be high or low
;
;
              ...
              ...
              XOR    #CCISx0,&CCTLx
              ...
              ...
              ...
```

11.3.5 Timer_A - Handle asynchronous serial protocol

The serial asynchronous protocol transmits and receives the data with a defined baudrate. A few different baudrates are defined in the industry. The receive uses the same or another baudrate as that in transmit. The receive starts with a negative edge of the signal. The receiver synchronizes itself with this negative edge, and the following bits are of the selected baudrate.

The transmit feature can be realized by using one compare function to shift data via the output unit to the selected pin. The baudrate is ensured by reconfiguring the compare data along with each interrupt. The output unit sets or resets the pin using the mode 1 for set and mode 5 for reset.

The receive feature can be realized by using one capture/compare function to shift data applied to a pin via the control register’s bit SCCIx into a memory. The receive start time is recognized by capturing the timer data with the negative edge of the receive signal.

The same capture/compare block is then selected to compare. The data for compare is the captured time plus half bit time determined by the baudrate. The first bit is latched with the first compare event EQUx. The scanning of the following bits is done the same way with a timing accordingly to the selected baud rate. The interrupt routine associated with the bit scanning collects all bits of one character for later processing by software.

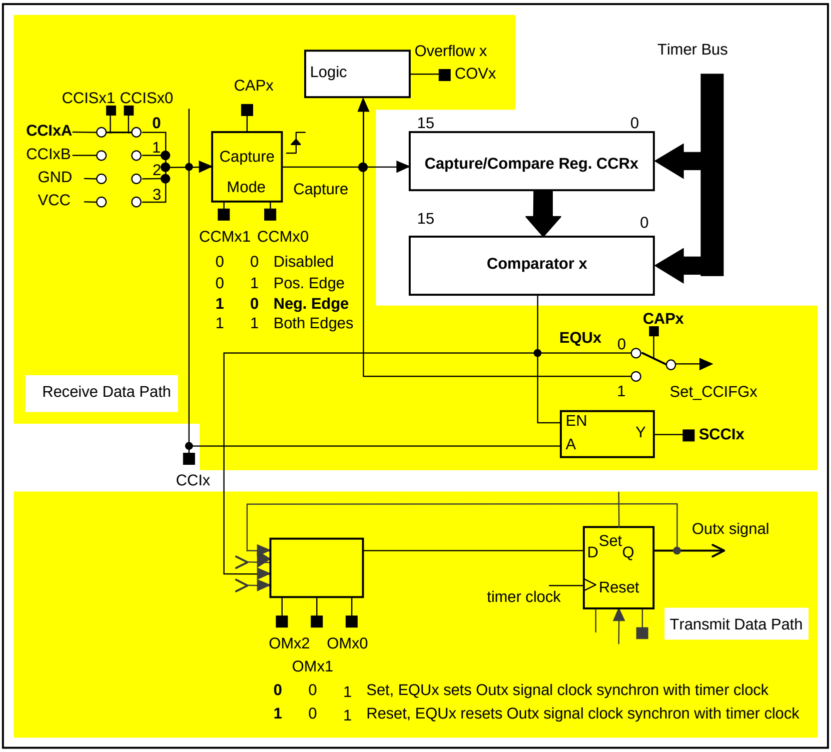


Figure 11.17: Timer_A used to handle asynchronous protocol

One capture/compare block is used when half duplex communication is selected. Two capture compare blocks are used to perform full duplex mode. In half duplex mode, receive and transmit should be sequential and use only one data line. In full duplex mode receive and transmit can be executed in parallel.

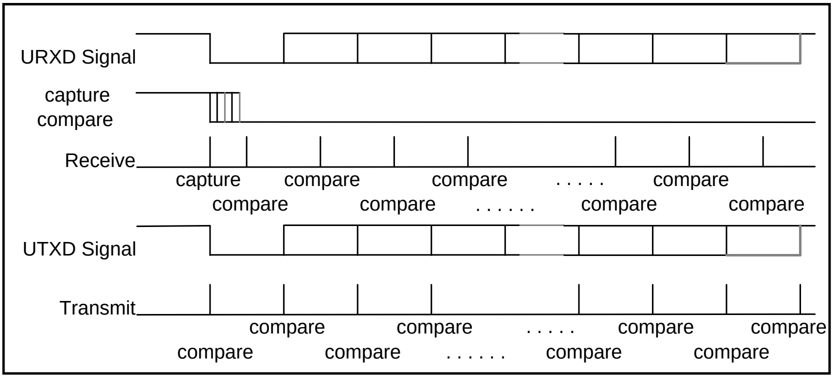


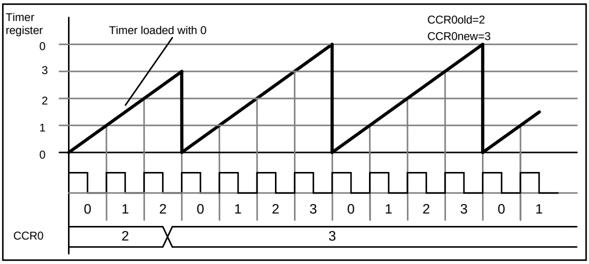
Figure 11.18: Timer_A, timing for asynchronous protocol handling

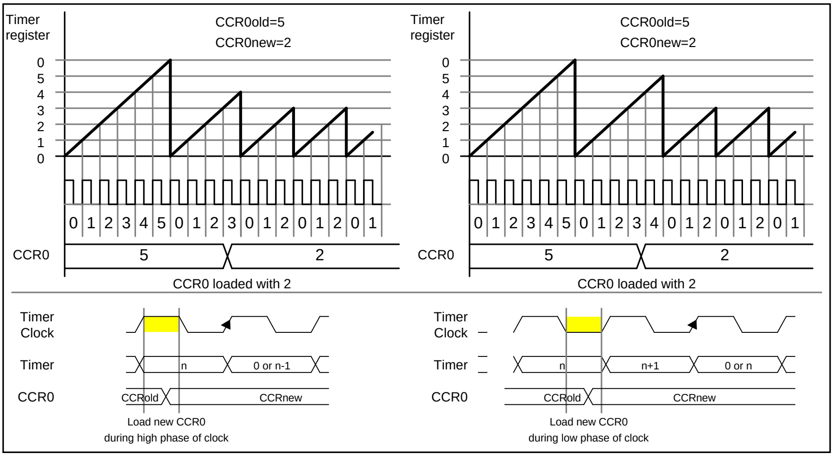
11.4 Timer_A special conditions

There are some special conditions possible, and these will be discussed in this section. A basic principle that follows all the timer and compare functions, is that increment or decrement from the timer register (by a timer clock) is needed to execute the selected function.

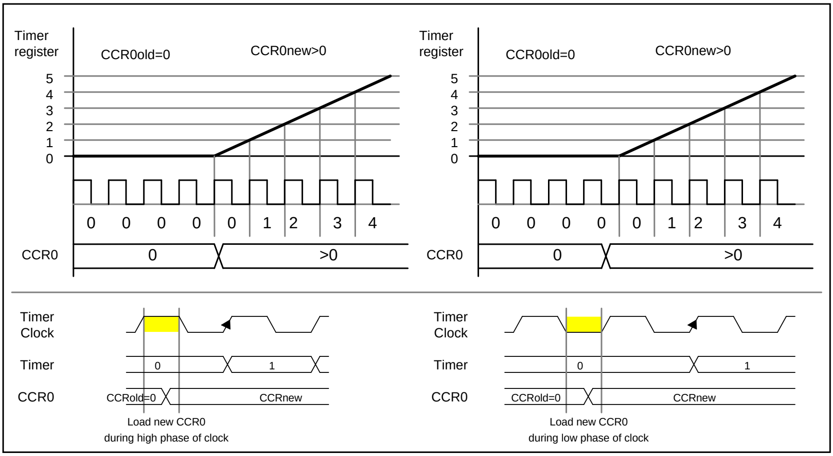
11.4.1 CCR0, used for period register

The compare registers are used for matching with the timer register 180^{o} before the timer register increments. When the CCR0 is used as a period register, and a new period is the same as or greater than the old period, the timer runs up to the new data and needs no special attention. When the CCR0 is used as the period register, and a new period is less than the old period, the timer is affected with the next positive edge if the new data was written to the CCR0 during the high phase of the timer clock. The timer continues to increment for one further leading edge of the timer clock, and is affected with the second leading timer clock edge if the CCR0 data was written during the low phase of the timer clock.





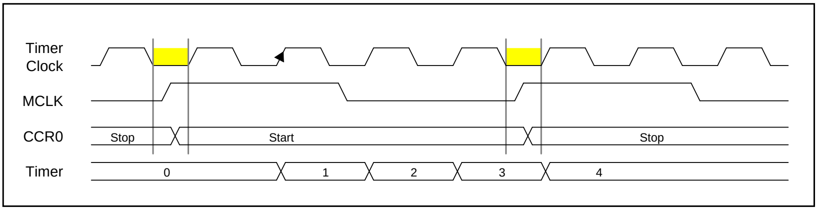
The previous examples demonstrate the different situations in the UP-Mode. The same reaction happens in the UP/DOWN-Mode when the timer operates in up-direction. The timer decrements continuously towards 0 if the period register CCR0 is altered when direction down is active.

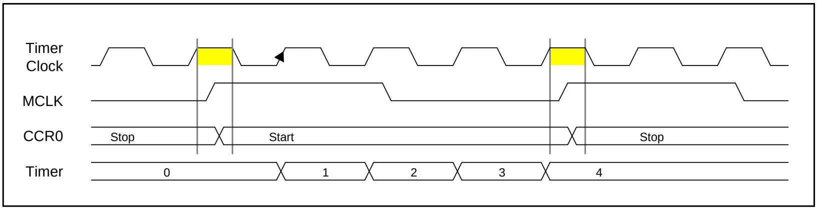


The counter starts this way in Up-Mode and UP/DOWN-Mode.

11.4.2 Start/Stop of the Timer Register

The start of the timer register, and also the stop of the timer register, follow the same basic rules as the period register CCR0.

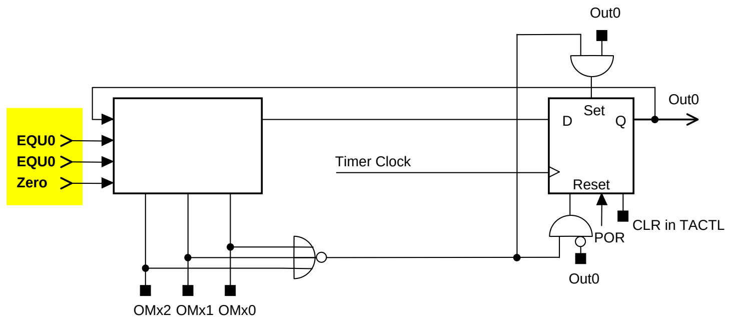




The selected count mode is loaded during the trailing edge of the timer clock. The following leading edge increments the timer register, if one of the three run modes is selected. The following leading edge does not further increment the timer register if the timer register is stopped.

11.4.3 Output Unit0

All output units have identical structures. The inputs use various control signals to define the specific operation. Two of the control signals are the comparator output timer-equal-compare register of the related module x (CCRx), and the comparator output timer-equal-compare register of the module 0 (CCR0). When the module x is the output unit 0, then not all of the possible operating conditions should be used:



OMx2	OMx1	OMx0	
0	0	0	Output Mode, Out0 signal is set according to Out0 bit
0	0	1	Set, EQU0 sets Out0 signal clock synchron with timer clock
0	1	0	>> uses for toggle and reset the EQU0 signal
0	1	1	>> uses for set and reset the EQU0 signal
1	0	0	Toggle, EQU0 toggles Out0 signal, clock synchron with timer clock
1	0	1	Reset, EQU0 resets Out0 signal clock synchron with timer clock
1	1	0	>> uses for toggle and set the EQU0 signal
1	1	1	>> uses for set and reset the EQU0 signal

The modes 0, 1, 4 and 5 are recommended.