14 The Capture / Compare Units

The C167 provides two almost identical Capture/Compare (CAPCOM) units which only differ in the way they are connected to the C167's IO pins. They provide 32 channels which interact with 4 timers. The CAPCOM units can **capture** the contents of a timer on specific internal or external events, or they can **compare** a timer's content with given values and modify output signals in case of a match. With this mechanism they support generation and control of timing sequences on up to 16 channels per unit with a minimum of software intervention.

From the programmer's point of view, the term 'CAPCOM unit' refers to a set of SFRs which are associated with this peripheral, including the port pins which may be used for alternate input/output functions including their direction control bits.

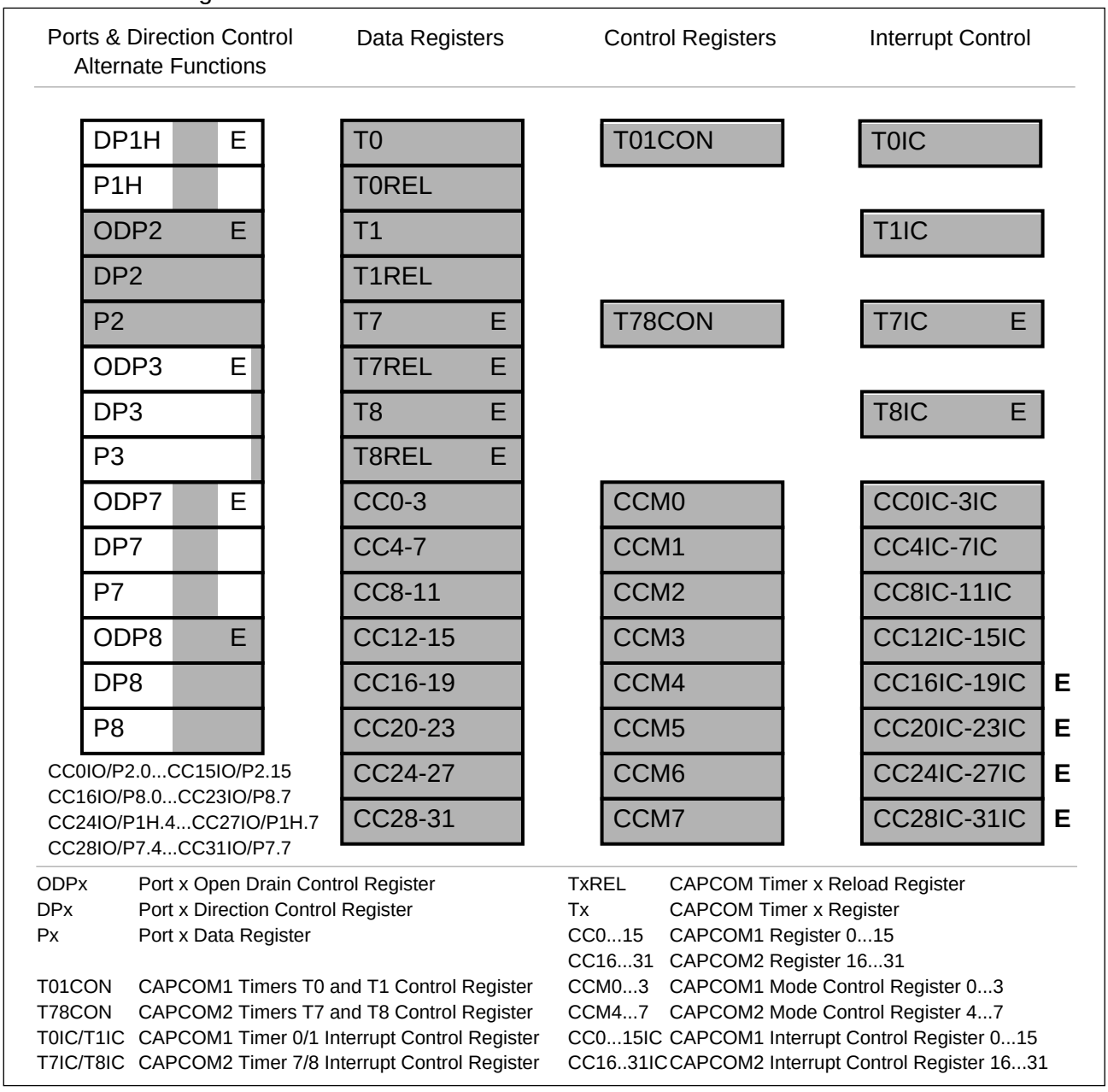

Figure 14-1
SFRs and Port Pins associated with the CAPCOM Units

A CAPCOM unit is typically used to handle high speed IO tasks such as pulse and waveform generation, pulse width modulation, or recording of the time at which specific events occur. It also allows the implementation of up to 16 software timers. The maximum resolution of the CAPCOM units is 400 ns (@ 20 MHz CPU clock).

Each CAPCOM unit consists of two 16-bit timers (T0 / T1 in CAPCOM1, T7 / T8 in CAPCOM2), each with its own reload register (TxREL), and a bank of sixteen dual purpose 16-bit capture/ compare registers (CC0 through CC15 in CAPCOM1, CC16 through CC31 in CAPCOM2).

The input clock for the CAPCOM timers is programmable to several prescaled values of the CPU clock, or it can be derived from an overflow/underflow of timer T6 in block GPT2. T0 and T7 may also operate in counter mode (from an external input) where they can be clocked by external events.

Each capture/compare register may be programmed individually for capture or compare function, and each register may be allocated to either timer of the associated unit. Each capture/compare register has one port pin associated with it which serves as an input pin for the capture function or as an output pin for the compare function (except for CC27...CC24 on P1H.7...P1H.4, which only provide the capture function). The capture function causes the current timer contents to be latched into the respective capture/compare register triggered by an event (transition) on its associated port pin. The compare function may cause an output signal transition on that port pin whose associated capture/compare register matches the current timer contents. Specific interrupt requests are generated upon each capture/compare event or upon timer overflow.

The figure below shows the basic structure of the two CAPCOM units.

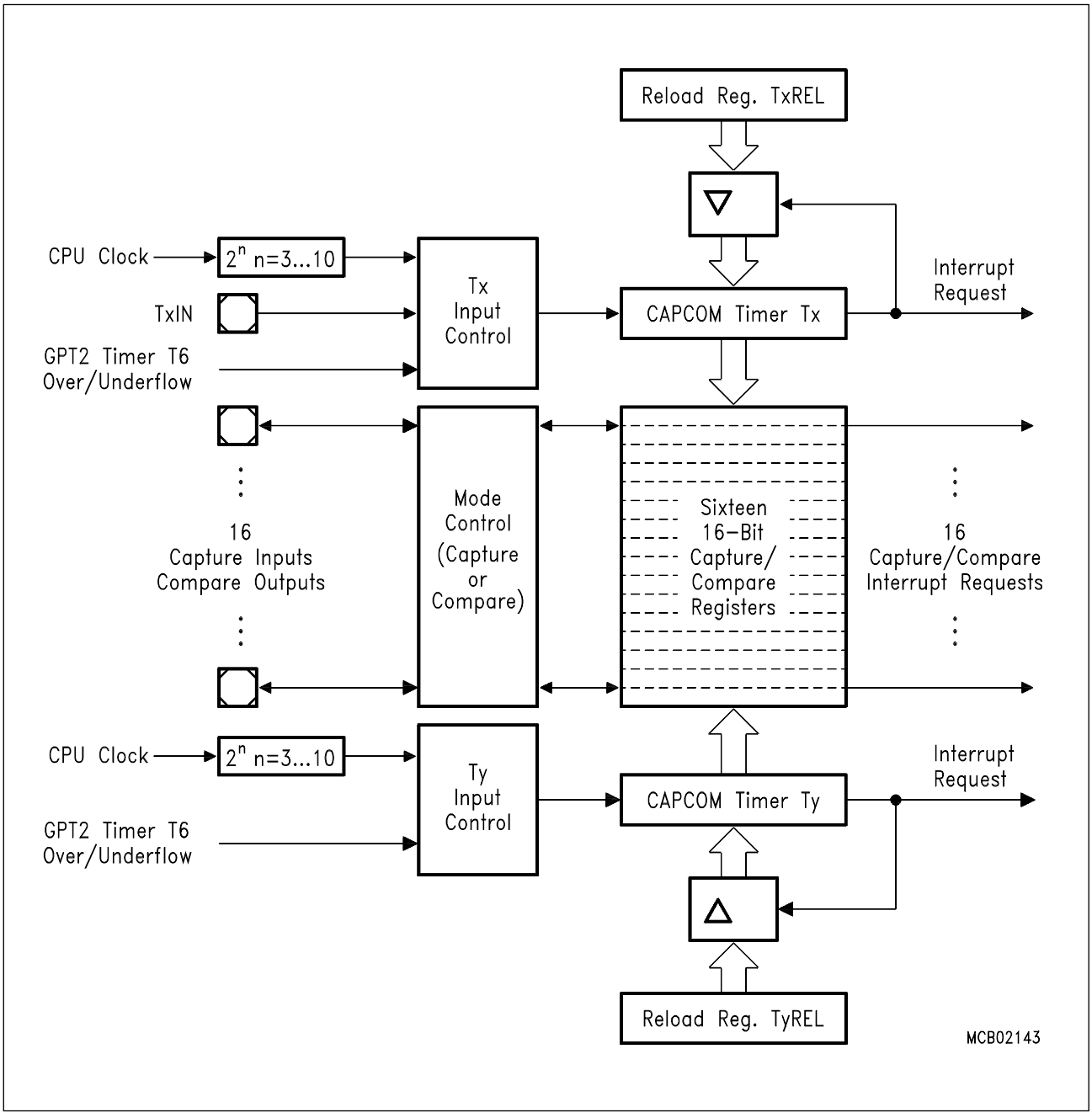

Figure 14-2
CAPCOM Unit Block Diagram

Note: The CAPCOM2 unit provides 16 capture inputs, but only 12 compare outputs.

14.1 The CAPCOM Timers

The primary use of the timers T0 / T1 and T7 / T8 is to provide two independent time bases (400 ns maximum resolution @ 20 MHz CPU clock) for the capture/compare registers of each unit, but they may also be used independent of the capture/compare registers.
The basic structure of the four timers is identical, while the selection of input signals is different for timers T0/T7 and timers T1/T8 (see figures below).

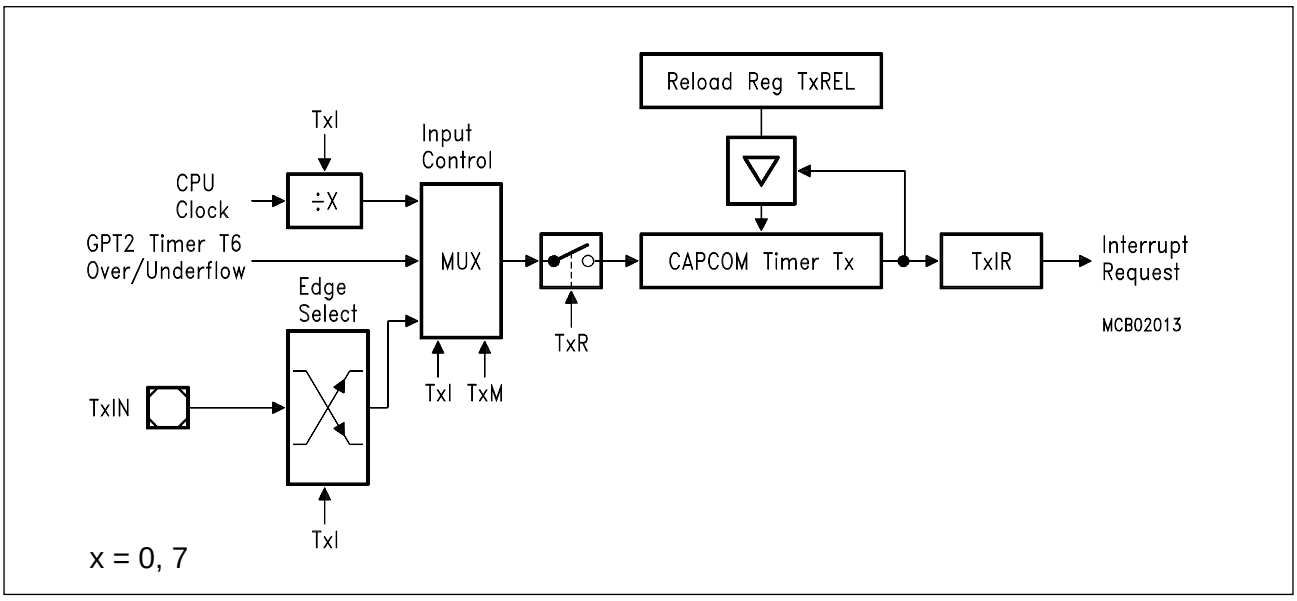

Figure 14-3
Block Diagram of CAPCOM Timers T0 and T7

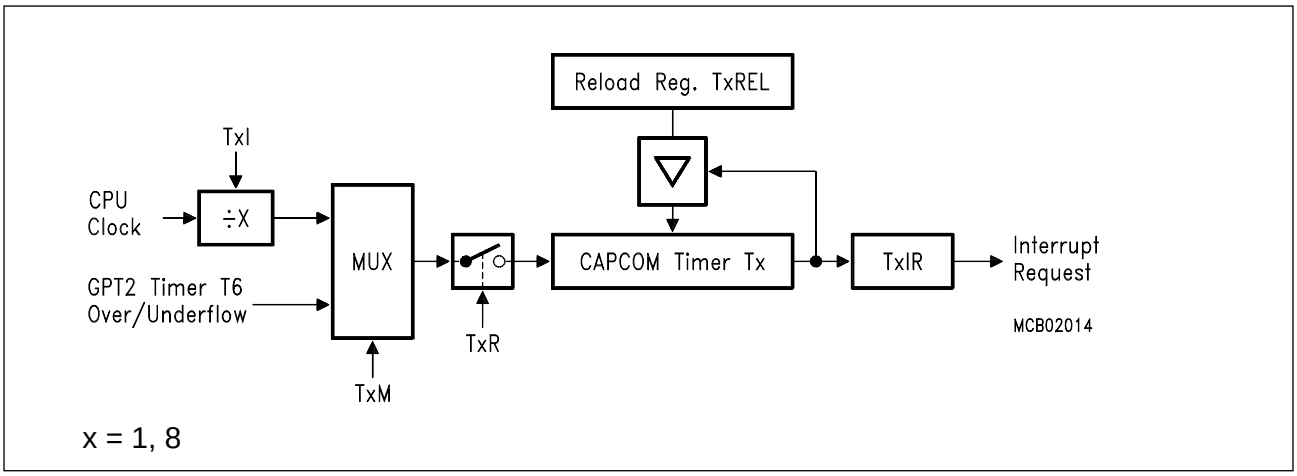

Figure 14-4
Block Diagram of CAPCOM Timers T1 and T8

Note: When an external input signal is connected to the input lines of both T0 and T7, these timers count the input signal synchronously. Thus the two timers can be regarded as one timer whose contents can be compared with 32 capture registers.

The functions of the CAPCOM timers are controlled via the bitaddressable 16-bit control registers T01CON and T78CON. The high-byte of T01CON controls T1, the low-byte of T01CON controls T0, the high-byte of T78CON controls T8, the low-byte of T78CON controls T7. The control options are identical for all four timers (except for external input).

T01CON ($FF50_H$ / $A8_H$) SFR **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	**T1R**	-	-	**T1M**		**T1I**		-	**T0R**	-	-	**T0M**		**T0I**	
-	rw	-	-	rw		rw		-	rw	-	-	rw		rw	

T78CON ($FF20_H$ / 90_H) SFR **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	**T8R**	-	-	**T8M**		**T8I**		-	**T7R**	-	-	**T7M**		**T7I**	
-	rw	-	-	rw		rw		-	rw	-	-	rw		rw	

Bit	Function
TxI	**Timer / Counter x Input Selection** Timer Mode (TxM='0') Input Frequency = f_{CPU} / $2^{(<TxI>+3)}$ See also table below for examples. Counter Mode (TxM='1'): X00 Overflow/Underflow of GPT2 Timer 6 X01 Positive (rising) edge on pin TxIN *) X10 Negative (falling) edge on pin TxIN *) X11 Any edge (rising and falling) on pin TxIN *)
TxM	**Timer / Counter x Mode Selection** '0': Timer Mode (Input derived from internal clock) '1': Counter Mode (Input from External Input or T6)
TxR	**Timer / Counter x Run Control** '0': Timer/Counter x is disabled '1': Timer/Counter x is enabled

*) This selection is available for timers T0 and T7. Timers T1 and T8 will stop at this selection!

The timer run flags T0R, T1R, T7R and T8R allow for enabling and disabling the timers. The following description of the timer modes and operation always applies to the enabled state of the timers, ie. the respective run flag is assumed to be set to '1'.

In all modes, the timers are always counting upward. The current timer values are accessible for the CPU in the timer registers Tx, which are non-bitaddressable SFRs. When the CPU writes to a

register Tx in the state immediately before the respective timer increment or reload is to be performed, the CPU write operation has priority and the increment or reload is disabled to guarantee correct timer operation.

Timer Mode

The bits TxM in SFRs T01CON and T78CON select between timer or counter mode for the respective timer. In timer mode (TxM='0'), the input clock for a timer is derived from the internal CPU clock divided by a programmable prescaler. The different options for the prescaler are selected separately for each timer by the bit fields TxI.

The input frequencies f_{Tx} for Tx are determined as a function of the CPU clock as follows, where <TxI> represents the contents of the bit field TxI:

$$f_{Tx} = \frac{f_{CPU}}{2^{(<TxI>+3)}}$$

When a timer overflows from $FFFF_H$ to 0000_H it is reloaded with the value stored in its respective reload register TxREL. The reload value determines the period P_{Tx} between two consecutive overflows of Tx as follows:

$$P_{Tx} = \frac{(2^{16} - <TxREL>) * 2^{(<TxI>+3)}}{f_{CPU}}$$

The timer input frequencies, resolution and periods which result from the selected prescaler option in TxI when using a 20 MHz CPU clock are listed in the table below. The numbers for the timer periods are based on a reload value of 0000_H. Note that some numbers may be rounded to 3 significant digits.

f_{CPU} = 20 MHz	Timer Input Selection TxI							
	000_B	001_B	010_B	011_B	100_B	101_B	110_B	111_B
Prescaler for f_{CPU}	8	16	32	64	128	256	512	1024
Input Frequency	2.5 MHz	1.25 MHz	625 kHz	312.5 kHz	156.25 kHz	78.125 kHz	39.06 kHz	19.53 kHz
Resolution	400 ns	800 ns	1.6 µs	3.2 µs	6.4 µs	12.8 µs	25.6 µs	51.2 µs
Period	26 ms	52.5ms	105 ms	210 ms	420 ms	840 ms	1.68 s	3.36 s

After a timer has been started by setting its run flag (TxR) to '1', the first increment will occur within the time interval which is defined by the selected timer resolution. All further increments occur exactly after the time defined by the timer resolution.
When both timers of a CAPCOM unit are to be incremented or reloaded at the same time T0 is always serviced one CPU clock before T1, T7 before T8, respectively.

Counter Mode

The bits TxM in SFRs T01CON and T78CON select between timer or counter mode for the respective timer. In Counter mode (TxM='1') the input clock for a timer can be derived from the overflows/underflows of timer T6 in block GPT2. In addition, timers T0 and T7 can be clocked by external events. Either a positive, a negative, or both a positive and a negative transition at pin T0IN (alternate input function of port pin P3.0) or T7IN (alternate input function of port pin P2.15), respectively, can be selected to cause an increment of T0/T7.

When T1 or T8 is programmed to run in counter mode, bit field TxI is used to enable the overflows/ underflows of timer T6 as the count source. This is the only option for T1 and T8, and it is selected by the combination TxI=$X00_B$. When bit field TxI is programmed to any other combination, the respective timer (T1 or T8) will stop.

When T0 or T7 is programmed to run in counter mode, bit field TxI is used to select the count source and transition (if the source is the input pin) which should cause a count trigger (see description of TxyCON for the possible selections).

Note: In order to use pin T0IN or T7IN as external count input pin, the respective port pin must be configured as input, ie., the corresponding direction control bit (DP3.0 or DP2.15) must be cleared ('0').
If the respective port pin is configured as output, the associated timer may be clocked by modifying the port output latches P3.0 or P2.15 via software, eg. for testing purposes.

The maximum external input frequency to T0 or T7 in counter mode is f_{CPU}/16 (1.25 MHz @ 20 MHz f_{CPU}). To ensure that a signal transition is properly recognized at the timer input, an external count input signal should be held for at least 8 CPU clock cycles before it changes its level again. The incremented count value appears in SFR T0/T7 within 8 CPU clock cycles after the signal transition at pin TxIN.

Reload

A reload of a timer with the 16-bit value stored in its associated reload register in both modes is performed each time a timer would overflow from $FFFF_H$ to 0000_H. In this case the timer does not wrap around to 0000_H, but rather is reloaded with the contents of the respective reload register TxREL. The timer then resumes incrementing starting from the reloaded value.

The reload registers TxREL are not bitaddressable.

14.2 CAPCOM Unit Timer Interrupts

Upon a timer overflow the corresponding timer interrupt request flag TxIR for the respective timer will be set. This flag can be used to generate an interrupt or trigger a PEC service request, when enabled by the respective interrupt enable bit TxIE.

Each timer has its own bitaddressable interrupt control register (TxIC) and its own interrupt vector (TxINT). The organization of the interrupt control registers TxIC is identical with the other interrupt control registers.

T0IC ($FF9C_H$ / CE_H) SFR **Reset Value: - - 00_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								T0IR	T0IE	ILVL				GLVL	
-	-	-	-	-	-	-	-	rw	rw	rw				rw	

T1IC ($FF9E_H$ / CF_H) SFR **Reset Value: - - 00_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								T1IR	T1IE	ILVL				GLVL	
-	-	-	-	-	-	-	-	rw	rw	rw				rw	

T7IC ($F17A_H$ / BE_H) ESFR **Reset Value: - - 00_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								T7IR	T7IE	ILVL				GLVL	
-	-	-	-	-	-	-	-	rw	rw	rw				rw	

T8IC ($F17C_H$ / BF_H) ESFR **Reset Value: - - 00_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								T8IR	T8IE	ILVL				GLVL	
-	-	-	-	-	-	-	-	rw	rw	rw				rw	

Note: Please refer to the general Interrupt Control Register description for an explanation of the control fields.

14.3 Capture/Compare Registers

The 16-bit capture/compare registers CC0 through CC31 are used as data registers for capture or compare operations with respect to timers T0/T1 and T7/T8. The capture/ compare registers are not bitaddressable.

Each of the registers CC0...CC31 may be individually programmed for capture mode or one of 4 different compare modes (except for CC24...CC27), and may be allocated individually to one of the two timers of the respective CAPCOM unit (T0 or T1, and T7 or T8, respectively). A special combination of compare modes additionally allows the implementation of a 'double-register' compare mode. When capture or compare operation is disabled for one of the CCx registers, it may be used for general purpose variable storage.

The functions of the 32 capture/compare registers are controlled by 8 bitaddressable 16-bit mode control registers named CCM0...CCM7 which are all organized identically (see description below). Each register contains bits for mode selection and timer allocation of four capture/compare registers.

Capture/Compare Mode Registers for the CAPCOM1 Unit (CC0...CC15)

CCM0 ($FF52_H$ / $A9_H$) SFR **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC3	CCMOD3			ACC2	CCMOD2			ACC1	CCMOD1			ACC0	CCMOD0		
rw	rw			rw	rw			rw	rw			rw	rw		

CCM1 ($FF54_H$ / AA_H) SFR **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC7	CCMOD7			ACC6	CCMOD6			ACC5	CCMOD5			ACC4	CCMOD4		
rw	rw			rw	rw			rw	rw			rw	rw		

CCM2 ($FF56_H$ / AB_H) SFR **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC 11	CCMOD11			ACC 10	CCMOD10			ACC9	CCMOD9			ACC8	CCMOD8		
rw	rw			rw	rw			rw	rw			rw	rw		

CCM3 ($FF58_H$ / AC_H) SFR **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC 15	CCMOD15			ACC 14	CCMOD14			ACC 13	CCMOD13			ACC 12	CCMOD12		
rw	rw			rw	rw			rw	rw			rw	rw		

Capture/Compare Mode Registers for the CAPCOM2 Unit (CC16...CC32)

CCM4 ($FF22_H$ / 91_H) SFR **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC 19	CCMOD19			ACC 18	CCMOD18			ACC 17	CCMOD17			ACC 16	CCMOD16		
rw	rw			rw	rw			rw	rw			rw	rw		

CCM5 ($FF24_H$ / 92_H) SFR **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC 23	CCMOD23			ACC 22	CCMOD22			ACC 21	CCMOD21			ACC 20	CCMOD20		
rw	rw			rw	rw			rw	rw			rw	rw		

CCM6 ($FF26_H$ / 93_H) SFR **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC 27	CCMOD27			ACC 26	CCMOD26			ACC 25	CCMOD25			ACC 24	CCMOD24		
rw	rw			rw	rw			rw	rw			rw	rw		

CCM7 ($FF28_H$ / 94_H) SFR **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACC 31	CCMOD31			ACC 30	CCMOD30			ACC 29	CCMOD29			ACC 28	CCMOD28		
rw	rw			rw	rw			rw	rw			rw	rw		

Bit	Function
CCMODx	**Mode Selection for Capture/Compare Register CCx** The available capture/compare modes are listed in the table below.
ACCx	**Allocation Bit for Capture/Compare Register CCx** ‘0’: CCx allocated to Timer T0 (CAPCOM1) / Timer T7 (CAPCOM2) ‘1’: CCx allocated to Timer T1 (CAPCOM1) / Timer T8 (CAPCOM2)

Selection of Capture Modes and Compare Modes

CCMODx	Selected Operating Mode
0 0 0	**Disable Capture and Compare Modes** The respective CAPCOM register may be used for general variable storage.
0 0 1	**Capture on Positive Transition (Rising Edge) at Pin CCxIO**
0 1 0	**Capture on Negative Transition (Falling Edge) at Pin CCxIO**
0 1 1	**Capture on Positive and Negative Transition (Both Edges) at Pin CCxIO**
1 0 0	**Compare Mode 0: Interrupt Only** Several interrupts per timer period; Enables double-register compare mode for registers CC8...CC15 and CC24...CC31.
1 0 1	**Compare Mode 1: Toggle Output Pin on each Match** Several compare events per timer period; This mode is required for double-register compare mode for registers CC0...CC7 and CC16...CC23.
1 1 0	**Compare Mode 2: Interrupt Only** Only one interrupt per timer period.
1 1 1	**Compare Mode 3: Set Output Pin on each Match** Reset output pin on each timer overflow; Only one interrupt per timer period.

The detailed discussion of the capture and compare modes is valid for all the capture/compare channels, so registers, bits and pins are only referenced by the placeholder ‘x’.

Note: Capture/compare channels 24...27 generate an interrupt request but do not provide an output signal. The resulting exceptions are indicated in the following subsections.
A capture or compare event on channel 31 may be used to trigger a channel injection on the C167’s A/D converter if enabled.

14.4 Capture Mode

In response to an external event the content of the associated timer (T0/T1 or T7/T8, depending on the used CAPCOM unit and the state of the allocation control bit ACCx) is latched into the respective capture register CCx . The external event causing a capture can be programmed to be either a positive, a negative, or both a positive or a negative transition at the respective external input pin CCxIO.

The triggering transition is selected by the mode bits CCMODx in the respective CAPCOM mode control register. In any case, the event causing a capture will also set the respective interrupt request flag CCxIR, which can cause an interrupt or a PEC service request, when enabled.

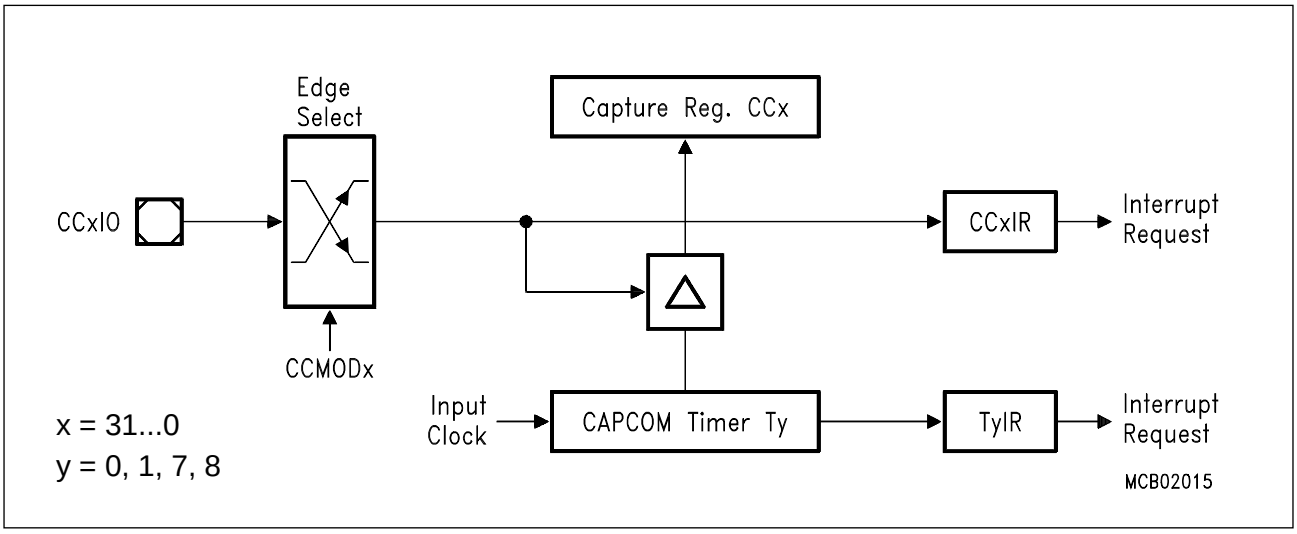

Figure 14-5
Capture Mode Block Diagram

In order to use the respective port pin as external capture input pin CCxIO for capture register CCx, this port pin must be configured as input, ie. the corresponding direction control bit must be set to ‘0’. To ensure that a signal transition is properly recognized, an external capture input signal should be held for at least 8 CPU clock cycles before it changes its level.

During these 8 CPU clock cycles the capture input signals are scanned sequentially. When a timer is modified or incremented during this process, the new timer contents will already be captured for the remaining capture registers within the current scanning sequence.

If pin CCxIO is configured as output, the capture function may be triggered by modifying the corresponding port output latch via software, eg. for testing purposes.

14.5 Compare Modes,

The compare modes allow triggering of events (interrupts and/or output signal transitions) with minimum software overhead. In all compare modes, the 16-bit value stored in compare register CCx (in the following also referred to as 'compare value') is continuously compared with the contents of the allocated timer (T0/T1 or T7/T8). If the current timer contents match the compare value, an appropriate output signal, which is based on the selected compare mode, can be generated at the corresponding output pin CCxIO (except for CC24IO...CC27IO) and the associated interrupt request flag CCxIR is set, which can generate an interrupt request (if enabled).

As for capture mode, the compare registers are also processed sequentially during compare mode. When any two compare registers are programmed to the same compare value, their corresponding interrupt request flags will be set to '1' and the selected output signals will be generated within 8 CPU clock cycles after the allocated timer is incremented to the compare value. Further compare events on the same compare value are disabled until the timer is incremented again or written to by software. After a reset, compare events for register CCx will only become enabled, if the allocated timer has been incremented or written to by software and one of the compare modes described in the following has been selected for this register.

The different compare modes which can be programmed for a given compare register CCx are selected by the mode control field CCMODx in the associated capture/compare mode control register. In the following, each of the compare modes, including the special 'double-register' mode, is discussed in detail.

Compare Mode 0

This is an interrupt-only mode which can be used for software timing purposes. Compare mode 0 is selected for a given compare register CCx by setting bit field CCMODx of the corresponding mode control register to '100_B'.

In this mode, the interrupt request flag CCxIR is set each time a match is detected between the content of compare register CCx and the allocated timer. Several of these compare events are possible within a single timer period, when the compare value in register CCx is updated during the timer period. The corresponding port pin CCxIO is not affected by compare events in this mode and can be used as general purpose IO pin.

If compare mode 0 is programmed for one of the registers CC8...CC15 or CC24...CC31, the double-register compare mode becomes enabled for this register if the corresponding bank 1 register is programmed to compare mode 1 (see section “Double- Register Compare Mode”).

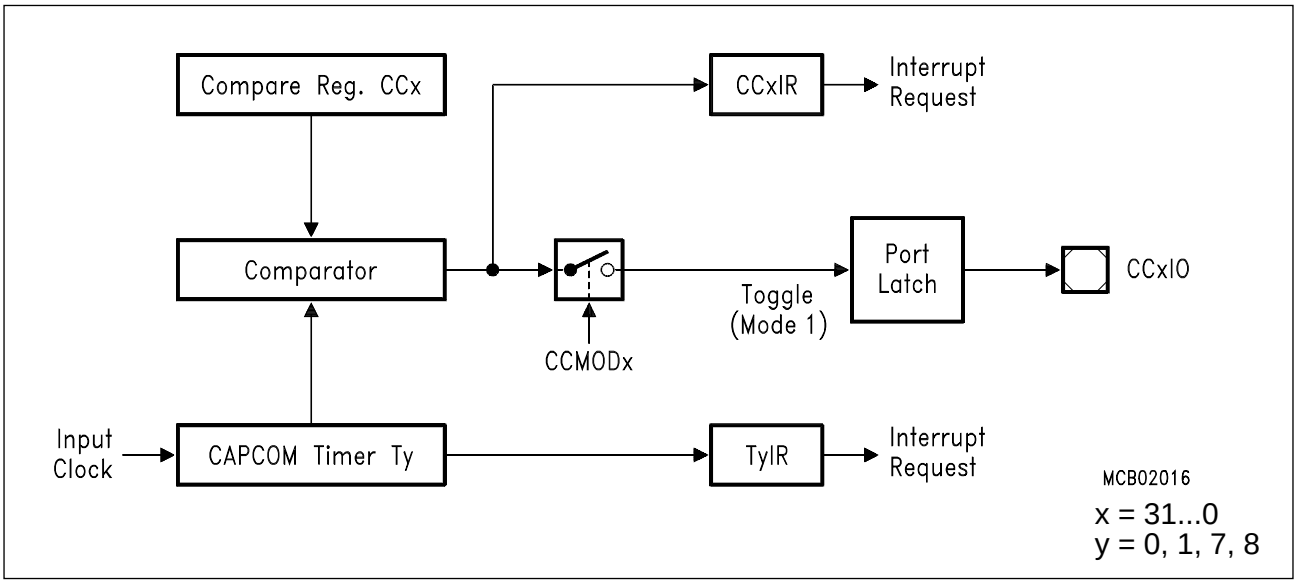

Figure 14-6
Compare Mode 0 and 1 Block Diagram

Note: The port latch and pin remain unaffected in compare mode 0.

In the example below, the compare value in register CCx is modified from cv1 to cv2 after compare events #1 and #3, and from cv2 to cv1 after events #2 and #4, etc. This results in periodic interrupt requests from timer Ty, and in interrupt requests from register CCx which occur at the time specified by the user through cv1 and cv2.

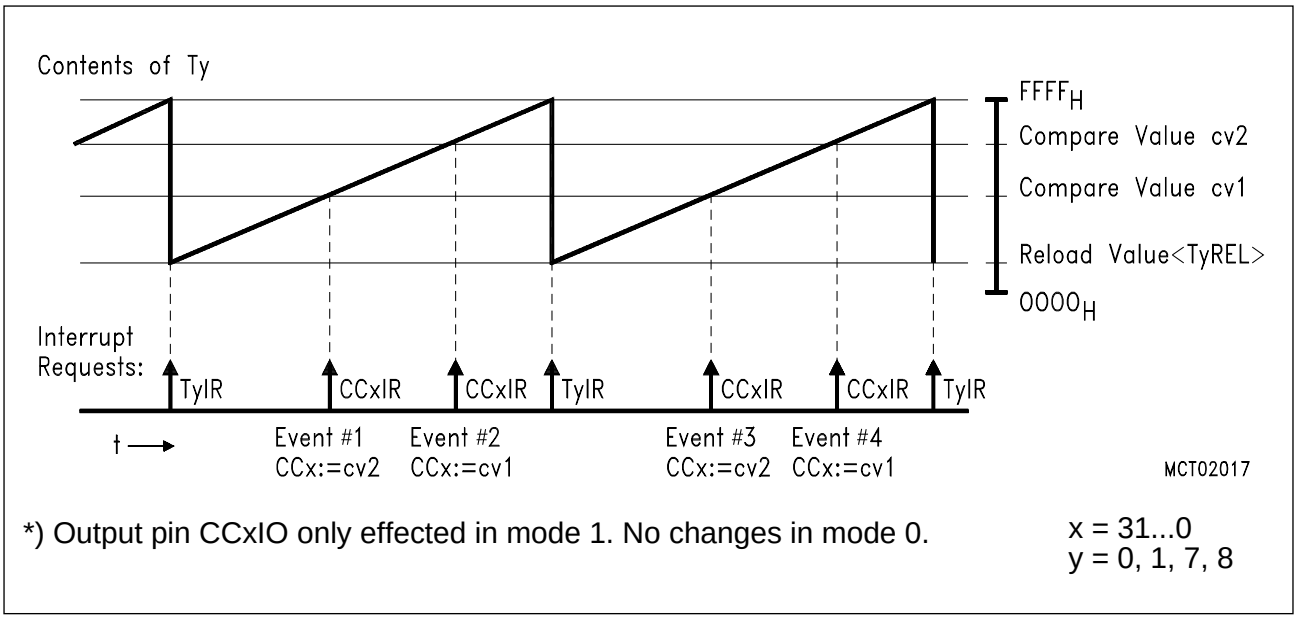

Figure 14-7
Timing Example for Compare Modes 0 and 1

Compare Mode 1

Compare mode 1 is selected for register CCx by setting bit field CCMODx of the corresponding mode control register to '101_B'.

When a match between the content of the allocated timer and the compare value in register CCx is detected in this mode, interrupt request flag CCxIR is set to '1', and in addition the corresponding output pin CCxIO (alternate port output function) is toggled. For this purpose, the state of the respective port output latch (not the pin) is read, inverted, and then written back to the output latch.

Compare mode 1 allows several compare events within a single timer period. An overflow of the allocated timer has no effect on the output pin, nor does it disable or enable further compare events.

In order to use the respective port pin as compare signal output pin CCxIO for compare register CCx in compare mode 1, this port pin must be configured as output, ie. the corresponding direction control bit must be set to '1'. With this configuration, the initial state of the output signal can be programmed or its state can be modified at any time by writing to the port output latch.

In compare mode 1 the port latch is toggled upon each compare event (see Timing Example above).

Note: If the port output latch is written to by software at the same time it would be altered by a compare event, the software write will have priority. In this case the hardware-triggered change will not become effective.

If compare mode 1 is programmed for one of the registers CC0...CC7 or CC16...CC23 the double-register compare mode becomes enabled for this register if the corresponding bank 1 register is programmed to compare mode 0 (see section "Double-Register Compare Mode").

Note: If the port output latch is written to by software at the same time it would be altered by a compare event, the software write will have priority. In this case the hardware-triggered change will not become effective.
On channels 24...27 compare mode 1 will generate interrupt requests but no output function is provided.

Compare Mode 2

Compare mode 2 is an interrupt-only mode similar to compare mode 0, but only one interrupt request per timer period will be generated. Compare mode 2 is selected for register CCx by setting bit field CCMODx of the corresponding mode control register to '110_B'.

When a match is detected in compare mode 2 for the first time within a timer period, the interrupt request flag CCxIR is set to '1'. The corresponding port 2 pin is not affected and can be used for general purpose IO. However, after the first match has been detected in this mode, all further compare events within the same timer period are disabled for compare register CCx until the allocated timer overflows. This means, that after the first match, even when the compare register is reloaded with a value higher than the current timer value, no compare event will occur until the next timer period.

In the example below, the compare value in register CCx is modified from cv1 to cv2 after compare event #1. Compare event #2, however, will not occur until the next period of timer Ty.

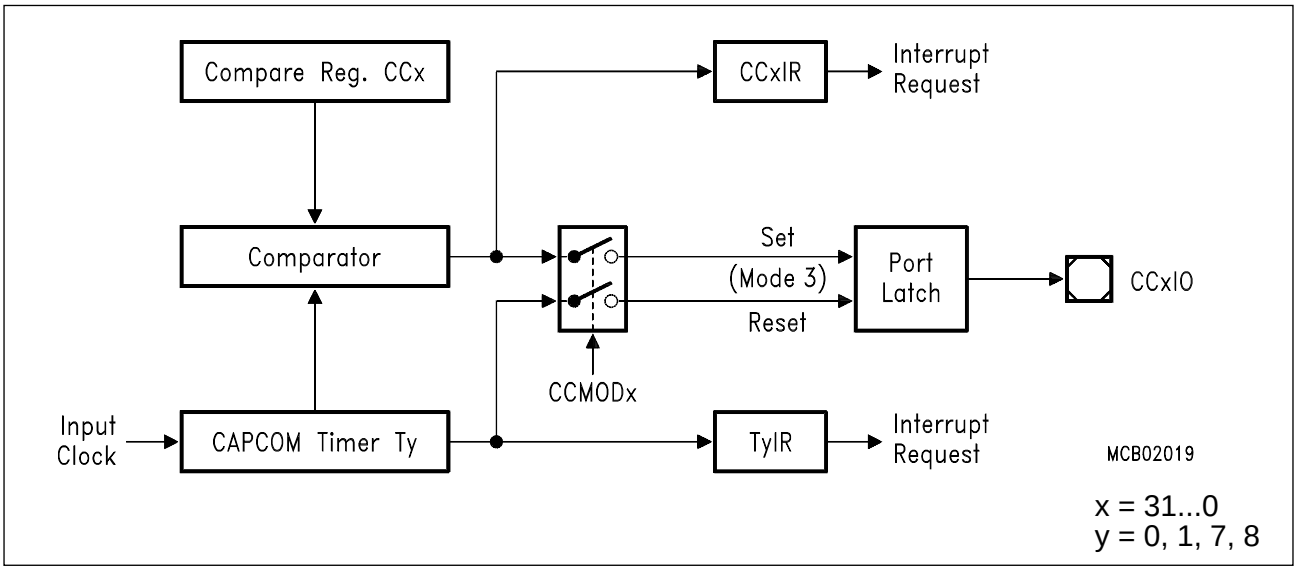

Figure 14-8
Compare Mode 2 and 3 Block Diagram

Note: The port latch and pin remain unaffected in compare mode 2.

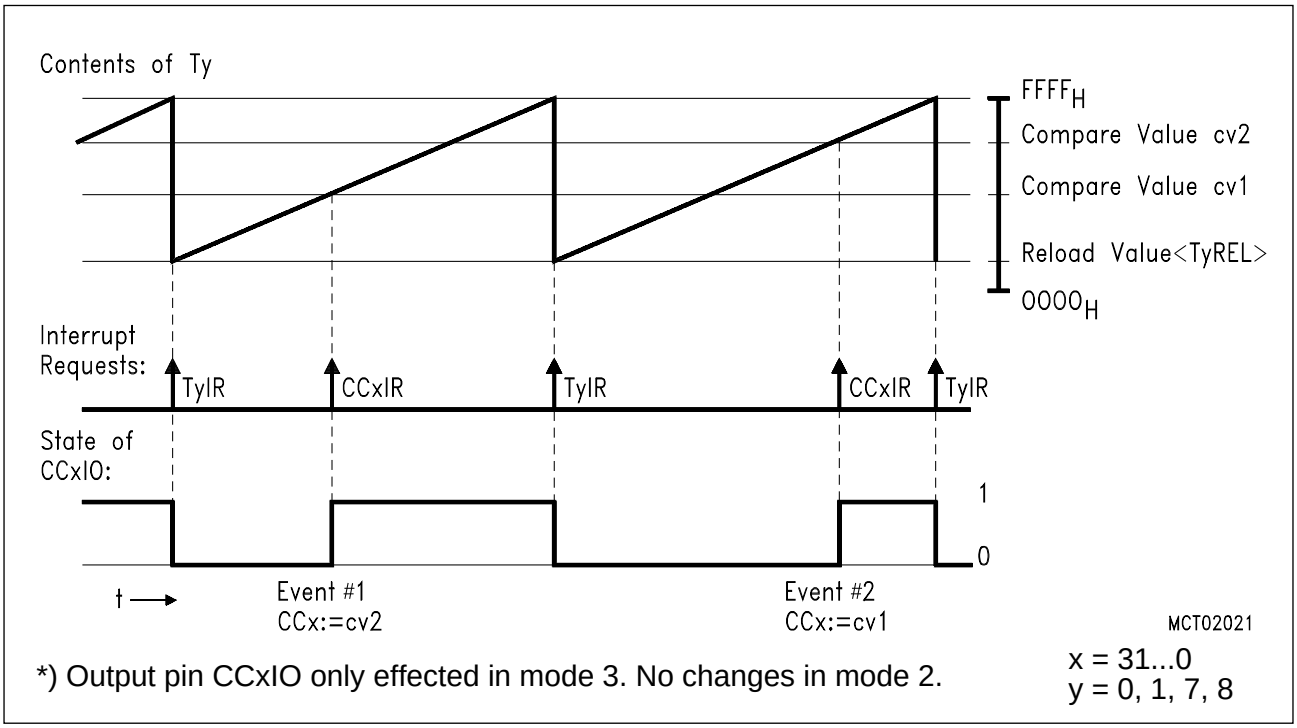

Figure 14-9
Timing Example for Compare Modes 2 and 3

Compare Mode 3

Compare mode 3 is selected for register CCx by setting bit field CCMODx of the corresponding mode control register to '111_B'. In compare mode 3 only one compare event will be generated per timer period.

When the first match within the timer period is detected the interrupt request flag CCxIR is set to '1' and also the output pin CCxIO (alternate port function) will be set to '1'. The pin will be reset to '0', when the allocated timer overflows.

If a match was found for register CCx in this mode, all further compare events during the current timer period are disabled for CCx until the corresponding timer overflows. If, after a match was detected, the compare register is reloaded with a new value, this value will not become effective until the next timer period.

In order to use the respective port pin as compare signal output pin CCxIO for compare register CCx in compare mode 3 this port pin must be configured as output, ie. the corresponding direction control bit must be set to '1'. With this configuration, the initial state of the output signal can be programmed or its state can be modified at any time by writing to the port output latch.

In compare mode 3 the port latch is set upon a compare event and and cleared upon a timer overflow (see Timing Example above).

However, when compare value and reload value for a channel are equal the respective interrupt requests will be generated, only the output signal is not changed (set and clear would coincide in this case).

Note: If the port output latch is written to by software at the same time it would be altered by a compare event, the software write will have priority. In this case the hardware-triggered change will not become effective.
On channels 24...27 compare mode 1 will generate interrupt requests but no output function is provided.

Double-Register Compare Mode

In double-register compare mode two compare registers work together to control one output pin. This mode is selected by a special combination of modes for these two registers.

For double-register mode the 16 capture/compare registers of each CAPCOM unit are regarded as two banks of 8 registers each. Registers CC0...CC7 and CC16...CC23 form bank 1 while registers CC8...CC15 and CC24...CC31 form bank 2 (respectively). For double-register mode a bank 1 register and a bank 2 register form a register pair. Both registers of this register pair operate on the pin associated with the bank 1 register (pins CC0IO...CC7IO and CC16IO...CC23IO).

The relationship between the bank 1 and bank 2 register of a pair and the effected output pins for double-register compare mode is listed in the table below.

Register Pairs for Double-Register Compare Mode

CAPCOM1 Unit			CAPCOM2 Unit		
Register Pair		Associated Output Pin	Register Pair		Associated Output Pin
Bank 1	Bank 2		Bank 1	Bank 2	
CC0	CC8	CC0IO	CC16	CC24	CC16IO
CC1	CC9	CC1IO	CC17	CC25	CC17IO
CC2	CC10	CC2IO	CC18	CC26	CC18IO
CC3	CC11	CC3IO	CC19	CC27	CC19IO
CC4	CC12	CC4IO	CC20	CC28	CC20IO
CC5	CC13	CC5IO	CC21	CC29	CC21IO
CC6	CC14	CC6IO	CC22	CC30	CC22IO
CC7	CC15	CC7IO	CC23	CC31	CC23IO

The double-register compare mode can be programmed individually for each register pair. In order to enable double-register mode the respective bank 1 register (see table) must be programmed to compare mode 1 and the corresponding bank 2 register (see table) must be programmed to compare mode 0.
If the respective bank 1 compare register is disabled or programmed for a mode other than mode 1 the corresponding bank 2 register will operate in compare mode 0 (interrupt-only mode).

In the following, a bank 2 register (programmed to compare mode 0) will be referred to as CCz while the corresponding bank 1 register (programmed to compare mode 1) will be referred to as CCx.

When a match is detected for one of the two registers in a register pair (CCx or CCz) the associated interrupt request flag (CCxIR or CCzIR) is set to ‘1’ and pin CCxIO corresponding to bank 1 register CCx is toggled. The generated interrupt always corresponds to the register that caused the match.

Note: If a match occurs simultaneously for both register CCx and register CCz of the register pair pin CCxIO will be toggled only once but two separate compare interrupt requests will be generated, one for vector CCxINT and one for vector CCzINT.

In order to use the respective port pin as compare signal output pin CCxIO for compare register CCx in double-register compare mode, this port pin must be configured as output, ie. the corresponding direction control bit must be set to ‘1’. With this configuration, the output pin has the same characteristics as in compare mode 1.

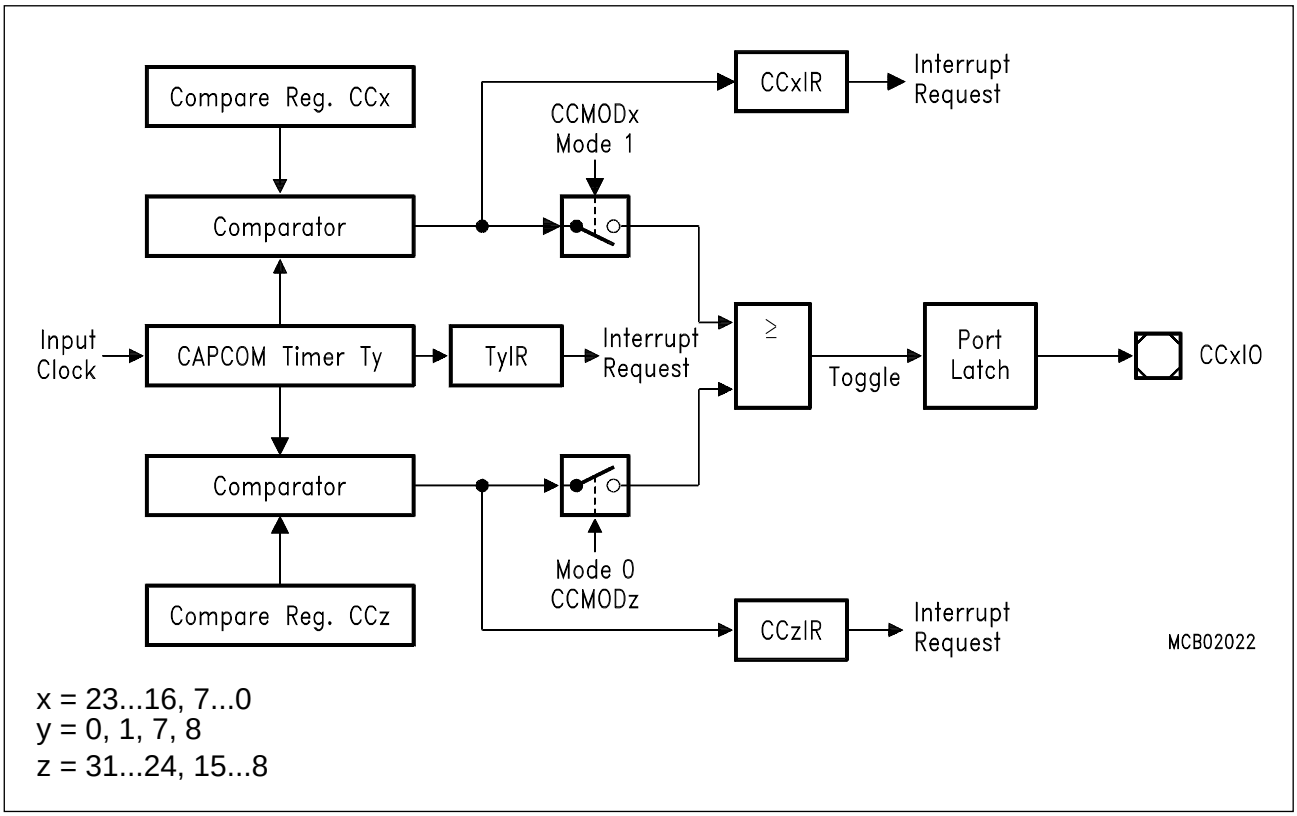

Figure 14-10
Double-Register Compare Mode Block Diagram

In this configuration example, the same timer allocation was chosen for both compare registers, but each register may also be individually allocated to one of the two timers of the respective CAPCOM unit. In the timing example for this compare mode (below) the compare values in registers CCx and CCz are not modified.

Note: The pins CCzIO (which do not serve for double-register compare mode) may be used for general purpose IO.

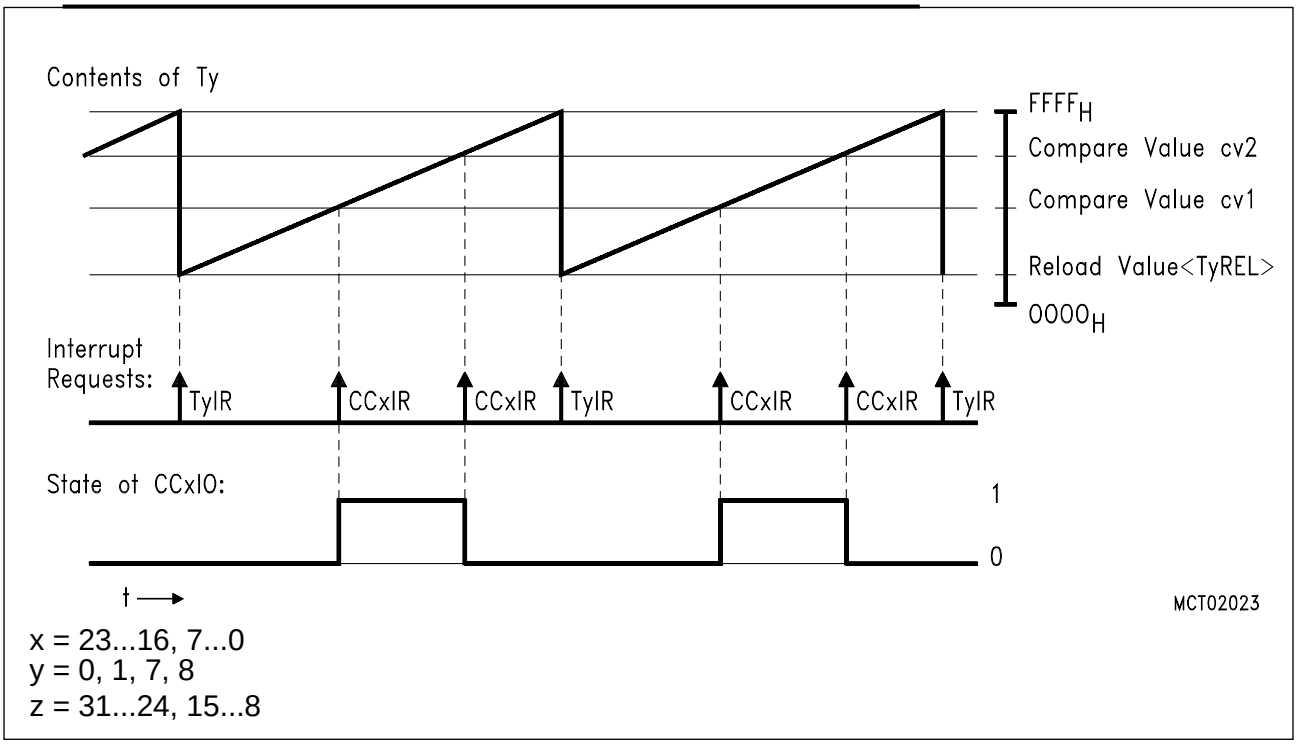

Figure 14-11
Timing Example for Double-Register Compare Mode

14.6 Capture/Compare Interrupts

Upon a capture or compare event, the interrupt request flag CCxIR for the respective capture/ compare register CCx is set to ‘1’. This flag can be used to generate an interrupt or trigger a PEC service request when enabled by the interrupt enable bit CCxIE.

Capture interrupts can be regarded as external interrupt requests with the additional feature of recording the time at which the triggering event occurred (see also section “External Interrupts”).

Each of the 32 capture/compare registers (CC0...CC31) has its own bitaddressable interrupt control register (CC0IC...CC31IC) and its own interrupt vector (CC0INT...CC31INT). These registers are organized the same way as all other interrupt control registers. The figure below shows the basic register layout, and the table lists the associated addresses.

CCxIC (See Table) SFR**/ESFR** **Reset Value: - - 00$_H$**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
								CCx IR	**CCx IE**	**ILVL**				**GLVL**	
-	-	-	-	-	-	-	-	rw	rw	rw				rw	

Note: Please refer to the general Interrupt Control Register description for an explanation of the control fields.

CAPCOM Unit Interrupt Control Register Addresses

CAPCOM1 Unit			CAPCOM2 Unit		
Register	**Address**	**Reg. Space**	**Register**	**Address**	**Reg. Space**
CC0IC	$FF78_H$ / BC_H	SFR	CC16IC	$F160_H$ / $B0_H$	**ESFR**
CC1IC	$FF7A_H$ / BD_H	SFR	CC17IC	$F162_H$ / $B1_H$	**ESFR**
CC2IC	$FF7C_H$ / BE_H	SFR	CC18IC	$F164_H$ / $B2_H$	**ESFR**
CC3IC	$FF7E_H$ / BF_H	SFR	CC19IC	$F166_H$ / $B3_H$	**ESFR**
CC4IC	$FF80_H$ / $C0_H$	SFR	CC20IC	$F168_H$ / $B4_H$	**ESFR**
CC5IC	$FF82_H$ / $C1_H$	SFR	CC21IC	$F16A_H$ / $B5_H$	**ESFR**
CC6IC	$FF84_H$ / $C2_H$	SFR	CC22IC	$F16C_H$ / $B6_H$	**ESFR**
CC7IC	$FF86_H$ / $C3_H$	SFR	CC23IC	$F16E_H$ / $B7_H$	**ESFR**
CC8IC	$FF88_H$ / $C4_H$	SFR	CC24IC	$F170_H$ / $B8_H$	**ESFR**
CC9IC	$FF8A_H$ / $C5_H$	SFR	CC25IC	$F172_H$ / $B9_H$	**ESFR**
CC10IC	$FF8C_H$ / $C6_H$	SFR	CC26IC	$F174_H$ / BA_H	**ESFR**
CC11IC	$FF8E_H$ / $C7_H$	SFR	CC27IC	$F176_H$ / BB_H	**ESFR**
CC12IC	$FF90_H$ / $C8_H$	SFR	CC28IC	$F178_H$ / BC_H	**ESFR**
CC13IC	$FF92_H$ / $C9_H$	SFR	CC29IC	$F184_H$ / $C2_H$	**ESFR**
CC14IC	$FF94_H$ / CA_H	SFR	CC30IC	$F18C_H$ / $C6_H$	**ESFR**
CC15IC	$FF96_H$ / CB_H	SFR	CC31IC	$F194_H$ / CA_H	**ESFR**