8051 Timers

Dinesh Sharma

Department Of Electrical Engineering
Indian Institute Of Technology, Bombay

February 8, 2021

Counter/Timers

- Counters are required in many applications. For example, a counter can be used with a light beam interrupter to count the number of objects on a conveyer belt.
- A counter which counts the number of clock cycles from a precise clock can be used as a timer. Timers are also required in many application – For example, we may want to sample a signal through an Analog to Digital converter at precise time intervals.
- Dedicated counter/timer chips like 8254 have been widely used for these applications.
- Most micro-controllers include one or more timer/counters on chip. The basic 8051 has two counter/timers T0 and T1 which may be configured and used individually.
- The 8052 has an additional Timer T2.

Counter/Timers

- The Counter/Timers included in 8051 count in the up direction only.
- The count is incremented at negative transitions on their inputs.
- Each counter has a 16 bit count register in the SFR area. The low and high bytes can be accessed as separate bytes.
- Registers TH1 and TL1 at direct addresses 8DH and 8BH hold the count for counter/timer T1. Similarly, TH2 and TL2 at direct addresses 8CH and 8AH hold the count for T0.
- The counting is enabled by the run flags TR1 and TR0 which are bits in the upper nibble of special function register TCON at direct address 88H.
- When their count rolls over from the maximum count to 0000, they set the corresponding timer flag (TF1 or TF0) in TCON special function register.

Counter/Timers

SPECIAL FUNCTION REGISTERS

ADDR	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2			CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

Counter/Timers

The upper nibble of the special function register TCON contains the bits TR1, TR0, TF1, TF0.

TCON register at BYTE address 88H								
Bit No.	7	6	5	4	3	2	1	0
Bit Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Bit Addr	8F	8E	8D	8C	8B	8A	89	88

The register is bit addressable. The byte address for TCON is 88H.
Its bits are addressed from 88H to 8FH.
Only the upper nibble is relevant to timer counters – the lower nibble is used by the interrupt system.

- Programs set or clear bits TR1/TR0 to start or stop the corresponding counter/timer T1 or T0.

Counter/Timers in 8051

T1 or T0 can be used as event counters (where they count the number of negative transitions on a pin connected to some external source).

Additional functions of Port 3 lines

Port Line	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
Function	$\overline{\text{RD}}$	$\overline{\text{WR}}$	T1 in	T0 in	$\overline{\text{INT1}}$	$\overline{\text{INT0}}$	TxD	RxD

Counter inputs are taken from pins P3.5 and P3.4.

These can also be used as Timers, where they count up once every twelfth clock cycle.

Selecting Counter or Timer function in 8051

TMOD register at BYTE address 89H								
Bit No.	7	6	5	4	3	2	1	0
Timer:	T1				T0			
Bit Name	G1	C/T1	T1M1	T1M0	G0	C/T0	T0M1	T0M0

(TMOD is not bit addressable).

- Function of a counter/timer is selected by setting or clearing the corresponding C/$\overline{T}$ flags in the TMOD special function register, placed at the address 89H.
- If the flag is set, the peripheral unit acts as a counter and receives its inputs from the pins.
- If the flag is cleared, it functions as a timer. In this mode, it receives inputs from the processor clock divided by 12.

Counter function in 8051

Additional functions of Port 3 lines

Port Line	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
Function	$\overline{RD}$	$\overline{WR}$	T1 in	T0 in	$\overline{INT1}$	$\overline{INT0}$	TxD	RxD

- Lines P3.5 and P3.4 can be used as counter inputs to Timers T1 and T0 respectively.
- If the C/$\overline{T}$ flag of a timer is set in TMOD, the corresponding line is sampled once every machine cycle (12 clock cycles).
- The count is advanced when a negative step is noticed on the line: this involves sampling a high level in one cycle and a low one on the next.
- Since each machine cycle takes 12 clock cycles, the fastest event counting rate is clock frequency/24.

Gating for Counter/Timers in 8051

Gating is useful when we want an external source to start or stop the timer.

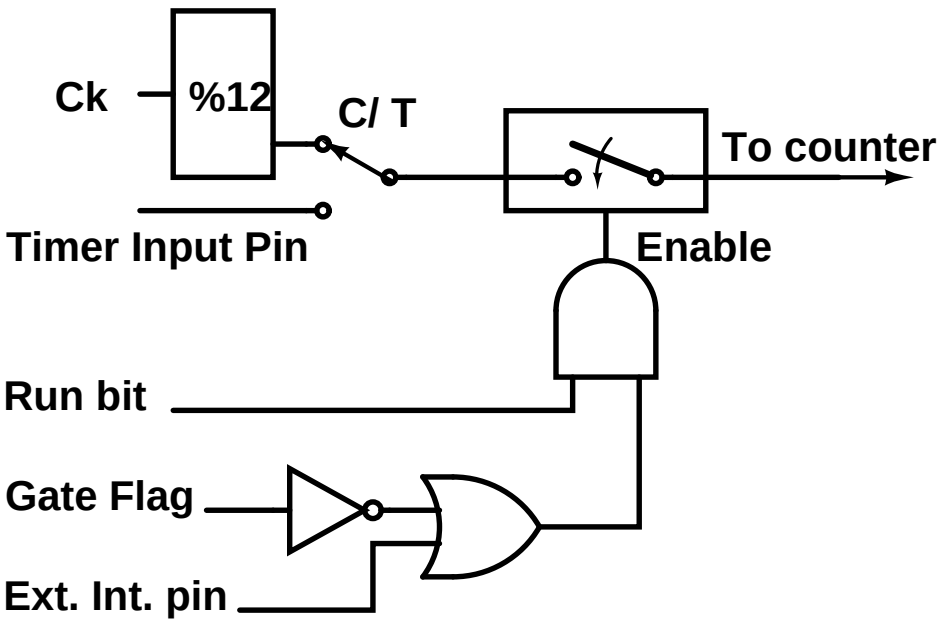
- Most timer/counters provide the facility of “gating” the clock, where an independent hardware signal enables or disables the counting process.
- In an 8051, we cannot afford to allocate a large number of pins for counter/timer functions. So gating is accomplished by control bits in the specital function register TMOD and external interrupt pins.

TMOD register at BYTE address 89H								
Bit No.	7	6	5	4	3	2	1	0
Timer:	T1				T0			
Bit Name	G1	C/T1	T1M1	T1M0	G0	C/T0	T0M1	T0M0

Bits 7 and 3 provide the gating functionality for the counter inputs.

Gating for Counter/Timers in 8051

TMOD register at BYTE address 89H								
Bit No.	7	6	5	4	3	2	1	0
Timer:	T1				T0			
Bit Name	G1	C/T1	T1M1	T1M0	G0	C/T0	T0M1	T0M0



- If the Gate Flag is cleared, the counter is enabled by the TR flag alone.
- If the Gate flag is set, counting also requires the corresponding external interrupt pin in P3 to be HIGH.
- This can be used for measuring pulse widths.

Counter/Timer modes in 8051

The timers may operate in one of four modes.
M1 and M0 bits in TMOD for the two timers select their mode of their operation.

Mode 0: In this mode, the timers act as 13 bit counters. This mode is largely meant for providing compatibility with an older microcontroller from intel (8048). This mode is practically never used in fresh designs. Except for the counter size, this mode is identical to mode 1.

Mode 1: In this mode, the timers are 16 bits in size. This is a commonly used mode. It is common to configure the timer to cause an interrupt when it overflows. The interrupt routine then reloads the timer.

Counter/Timer modes in 8051

Mode 2: This mode provides an 8 bit counter with auto-reload. Modes 0 and 1 require the software to re-load the counters with specific values every time they overflow.

If we need a free running clock generator, this would involve intervention by software for loading the delay value on a per cycle basis. This frequent software overhead is not convenient.

An auto re-load feature in hardware can free the software of this demand. We can then initialise the timer once and for all, specifyint the value to be reloaded and set the timer to run. After that, the timer runs at the set frequecy, producing periodic output which can be used by applications like baud rate generation for serial communication.

Mode 2: Auto reload mode

- This mode uses the high byte of the count register to store the count value to be re-loaded and the low byte as the actual counter.
- The 8 bit counter using TL is automatically re-loaded from TH when it overflows. Thus, there is no software overhead for re-loading the registers.
- This is convenient for generating baud rates etc.
- However, the timing resolution is much lower in this mode (only 8 bits). Therefore crystal frequencies have to be carefully chosen to generate accurate baud rates.
- Crystals of 11.059 MHz are often used rather than 12 MHz for this reason.

Counter/Timer modes in 8051

Mode 3: In this mode Timers T0 and T1 behave quite differently.

- T0 acts as two independent 8 bit counters.
- Count register TL0 uses the resources (such as the RUN flag, overflow flag) in TCON, TMOD etc. meant for T0.
- Similarly, TH0 uses the resources meant for T1.
- Thus, TR1 will enable running the 8 bit counter made up of TH0. TF1 will be set whenever TH0 overflows.
- T1 now has no control bits at all! It can only be used for services which require no control, no gating and no interrupts.
- For example, T1 can be used for Baud rate generation, while we still have two timers available (both with 8 bit resolution).

8051 Counter/Timers in interrupt mode

- There is no output brought out directly from the counter/timers in 8051.
- However, the processor can be set to be interrupted by the timers and as a part of the interrupt service routine, the processor can perform any IO function through its ports.
- The 8051 can be set up so that an interrupt occurs whenever the counter overflows (TF1 or TF0 is set).
- When 8051 branches to the interrupt vector, it automatically clears the TF flag.

8051 Counter/Timers in interrupt mode

SFR IE at byte address A8H contains flags which allow an interrupt to occur whenever a timer overflows.

Bit No.	7	6	5	4	3	2	1	0
Bit Addr	AF	AE	AD	AC	AB	AA	A9	A8
Bit Name	EA	-	-	ES	ET1	EX1	ET0	EX0
Interrupt on	IE	U	TF2	SI	TF1	Ex1	TF0	Ex0

- The most significant bit of the register is a global interrupt enable flag. This bit must be set in order to enable any interrupt.
- Bit 3 should be set to enable interrupts from Timer 1 overflow, while Bit 1 enables interrupts from Timer 0 when it overflows.

8051 Counter/Timers in interrupt mode

When an interrupt occurs, the updated PC is pushed on the stack and is loaded with the vector address corresponding to the interrupt.
The following table gives the vector addresses for timer interrupts:

Interrupt Source	Vector address
Timer 0 Overflow	000BH
Timer 1 Overflow	001BH

Only 8 bytes are available between distinct interrupt vectors. Many simple interrupt handlers can be accommodated in this space. (For example, just re-loading of counts after timer overflow).

Otherwise, jump instructions (to actual handler locations) need to be placed at the vector addresses.

8051 Counter/Timers in interrupt mode

Thus, to enable interrupts from T0, we have to do

```
SetB EA    ;    (or SetB IE.7) to enable interrupts
SetB ET0   ;    (or SetB IE.1) to enable interrupts from T0
```

After this, whenever T0 overflows,

- TF0 will be set (in SFR TCON),
- the currently running program will be interrupted,
- its PC value will be put on the stack (PC-L first, PC-H after – because the stack grows upwards in 8051),
- and PC will be loaded with 000B H, the interrupt vector for T0.
- The interrupt handler for T0 should be placed here, and it should end with the instruction:
  RETI

Initializing Timer Counts

- Timers count up and set their flags when they go from max. count to 0000.
- Therefore, it is more convenient to think of the initial counts as negative numbers which will increment to 0.
- Suppose we want the timer to time out after 10 ms and we are using a 12 MHz crystal. Since the timer is incremented at $f_{osc}/12 = 1$ MHz, each count cycle is 1 μs in duration.
- So, we need a count time of 10,000 cycles.
- We should therefore initialise the count to -10,000.

Since 10,000 = 39*256+16 = 2710H,
-10000 = 2's complement of 2710 = D8F0.
Thus, we should set TH to D8 and TL to F0.

Initializing Timer Counts

- To generate a delay of 10 ms on an 8051 system with a 12 MHz crystal, we need to load an initial count of $-10000 = D8F0$.
- Another way to think about it is that the overflow count is actually 10000H.
- So we should subtract the required count from this and initialize the count to the difference.
- Thus, the initial count in the above example should be:

```
  10000
-  2710
_______
=  D8F0
_______
```

Either way, we get the same result.

Reading Timer Counts

- Since the current count is in two bytes, we have to read these sequentially.
- This presents a problem – because by the time we go to read the second byte, the first one may have changed.
- For example, suppose the count is 04FF. We do:
  MOV R0, TL0
  and get FF in R0. However, one machine cycle has passed in executing the MOV instruction.
- So the count reaches 0500. Therefore, if we do
  MOV R1, TH0; we shall get 05 in R1.
- Thus we can be misled into thinking that the count is actually 05FF.

Reading Timer Counts

- Reading the two byte count of a running 16 bit counter may introduce errors because the second byte may have changed by the time we read it.
- This problem had been tackled in other popular counter/timers like 8254 through the “Latch Counter” command.
- This command copies the count to a latch while the counter continues operating.
- The latched value can then be read safely.
- However, this command does not exist in an 8051. (Actually, new members of the 8051 family include an additional timer which does provide this facility).
- For T0 and T1, how do we solve this problem?

Reading Timer Counts

- One way is to read the high byte of the timer, then read the low byte, then read the high byte again.
- If the high byte read the second time is not the same as the high byte read the first time, one must repeat the cycle.

In code, this would appear as:

```
READT0:
        MOV A, TH0
        MOV R0, TL0
        CJNE A, TH0, READT0
```

Another way is to freeze the counter by clearing its RUN bit, read its value and then start it again by setting the RUN bit.
Of course, one might miss a few counts this way - but at least the timer value won't be grossly wrong.

Additional timer T2 in 8052

- 8052 and most modern micro-controllers in the 8051 include an additional 16 bit timer – called T2.
- When used as a 16 bit counter/timer, it works the same way as T0 and T1.
- Pin P1.0 now provides an alternative function as counter input for this timer.
- A new feature is that this pin can also act as an output and with appropriate setting of control bits and modes, the Timer output will appear as an output signal on this pin.
- The count for T2 is maintained in two 8 bit registers TH2 and TL2 which are located at addresses CD and CC respectively.
- The timer is controlled by bits in the registers T2CON and T2MOD which are located at addresses C8 and C9 in SFR area.
- Just like T0 and T1, bit TR2 enables T2 to run, TF2 is set on overflow of T2 count and C/$\overline{T2}$ decides whether the counter/timer will work as a counter or a timer.

New features for T2

Since this timer was added only in later members of the 8051 family, it has been designed to remove some limitations seen with T0 and T1.

- Associated with this timer are two capture registers, RCAP2H and RCAP2L.
- The count can be captured into these registers while the counter is running, without needing repeated reading and comparison.
- These two registers can also be used to store the reload value for the counter when it overflows. Thus, with proper setting of control bits, the counter will be automatically re-loaded from RCAP2H and RCAP2L when it overflows.
- Unlike mode 2 for T0 and T1, we do not sacrifice counter resolution when we implement the auto-reload feature. (In mode 2 for T0 and T1, TH was used for storing the re-load value and only TL was available for counting).
- This permits us to make a free running clock generator which needs no software intervention once it is configured and started.

New features for T2

- The free running mode with auto-reload is useful for applications like baud rate generation for serial communication.
- T2 can also be configured for down counting, which was not possible for T0 and T1.
- All the new features for T2 are controlled through additional flags in T2CON and T2MOD registers. Thus there are many more control flags for T2.
- Special function registers associated with T2 are grouped together in the SFR area at addresses C8 to CD.
- Six T2 related SFRs T2CON, T2MOD, RCAP2L, RCAP2H, TL2 and TH2 are placed at addresses C8, C9, CA, CB, CC and CD respectively.

Register T2CON

T2CON is at address C8H and is bit addressable. Most configuration flags for T2 are in this register. For use as a regular 16bit counter/timer, bits RCLK and TCLK should be cleared.

T2CON register at BYTE address C8H								
Bit No.	7	6	5	4	3	2	1	0
Bit Name	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/RL2
Bit Addr	CF	CE	CD	CC	CB	CA	C9	C8

- Setting bit T2CON.2 (TR2) permits T2 to run.
- Bit T2CON.7 (TF2) is set by hardware on overflow if RCLK=0 and TCLK=0. It should be cleared by software.
- Bit T2CON.1 (C/$\overline{T2}$) is set for counter operation in which count advances with a negative edge on pin P0.1. This bit is cleared for timer operation, where the count advances at every 12th clock.

Register T2CON

A common use for T2 is as a baud rate generator for serial port. For this, either or both of RCLK and TCLK should be set. When this happens, T2 is automatically re-loaded with the contents of RCAPH and RCAPL registers on overflow.

T2CON register at BYTE address C8H								
Bit No.	7	6	5	4	3	2	1	0
Bit Name	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/RL2
Bit Addr	CF	CE	CD	CC	CB	CA	C9	C8

- Bit T2CON.5 (RCLK) should be set to use overflow events on T2 as the Receive Clock for the serial port in mode 1 or 3. It should be cleared if the receive clock is to be generated using T1.
- Bit T2CON.4 (TCLK) should be set to use overflow events on T2 as the Transmit Clock for the serial port in mode 1 or 3. It should be cleared if the transmit clock is to be generated using T1.

Register T2CON

Bit T2CON.3 (EXEN2) is the external enable bit for T2. This bit is relevant only when T2 is NOT being used for clocking the serial port. Using this, an external signal can trigger capture/reload for the count.

T2CON register at BYTE address C8H								
Bit No.	7	6	5	4	3	2	1	0
Bit Name	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T2}$	CP/RL2
Bit Addr	CF	CE	CD	CC	CB	CA	C9	C8

- When RCLK = 0 and TCLK = 0, this bit should be set to cause a capture or re-load event when a negative transistion is detected on pin T2EX (P1.1). Which of these will occur depends on T2CON.0 (CP/RL2) bit.
- This bit should be cleared to ignore transitions on T2EX (P1.1) pin for T2 operation.

Register T2CON

Bit T2CON.0 (CP/RL2) determines the direction of data flow between the count and capture registers (whether a capture or a reload will occur). This bit is relevant only when both TCLK and RCLK are 0, and T2CON.3 (EXEN2) is set.

T2CON register at BYTE address C8H								
Bit No.	7	6	5	4	3	2	1	0
Bit Name	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/RL2
Bit Addr	CF	CE	CD	CC	CB	CA	C9	C8

Notice that if either RCLK or TCLK is 1, T2 is forced to auto reload anyway whenever its count overflows.

- If EXEN2 = 1, and this bit is set, a capture event occurs when a negative transition is detected on T2EX pin (P1.1).
  If EXEN2 = 1, and this bit is cleared, a reload event occurs when a negative transition is detected on T2EX pin (P1.1) or when T2 overflows.

Register T2CON

T2CON register at BYTE address C8H								
Bit No.	7	6	5	4	3	2	1	0
Bit Name	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T2}$	CP/RL2
Bit Addr	CF	CE	CD	CC	CB	CA	C9	C8

- If EXEN2 = 1 and a capture/reload event does occur due to a negative transition of T2EX pin (P1.1), flag T2CON.6 (EXF2) gets set.
- When set, it causes the CPU to vector to Timer 2 interrupt routine if Timer 2 interrupt is enabled.
- This flag must be cleared by software.
- EXF2 does not cause an interrupt in Up/down counter mode (i.e. when TMOD bit 0, DCEN is set).

Register T2MOD

T2MOD register at BYTE address C9H								
Bit No.	7	6	5	4	3	2	1	0
Bit Name	-	-	-	-	-	-	T2OE	DCEN

This register is not bit addressable. Only two bits of this register are used for operation of T2.

- BIT T2MOD.1 (T2OE). This is the output enable bit. When set, pin P1.0 becomes an output pin for clock output.
  When cleared, pin P1.0 is used either as a port pin or as count input to T2.
- BIT T2MOD.0 (DCEN). When set, this bit enables up/down counting on T2. When cleared, up/down counting is disabled.