NMK322 - Microcontroller

Lecture 05 – 8051 Timer

8051 Timer

- 2 x 16-bit timer registers (Timer 0 & Timer 1)
  - can also function as counters
- Timer function:
  - interval timing
  - register increments every machine cycle
  - 12 osc periods per cycle → rate = $f_{osc}/12$
  - timer flag set on FFFF→0000 transition (overflow)
- Counter function:
  - counting ‘events’
  - register increments at 1→0 transition at Tx pin
  - each level should be held at least 1 machine cycle

8051 Timer Registers

- both timers (T0 & T1) are SFRs
  - 16-bit register split into 2 x 8-bit (Tx → {THx,Tlx})
  - TH0 @ 0x8C , TL0 @ 0x8A
  - TH1 @ 0x8D , TL1 @ 0x8B
- 2 SFRs for control / configuration
  - shared by both T0 & T1
  - TCON @ 0x88 (timer control register)
  - TMOD @ 0x89 (timer mode register)

Timer Control Register (TCON)

- 8-bit control/status register (@0x88) for both T0 & T1
  - bit addressable

MSB LSB

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Bit	Name	Description
TCON.7	TF1	Timer 1 overflow flag
TCON.6	TR1	Timer 1 run-control bit. Used to turn the timer on/off
TCON.5	TF0	Timer 0 overflow flag
TCON.4	TR0	Timer 0 run-control bit.
TCON.3	IE1	External Interrupt 1 edge flag
TCON.2	IT1	External Interrupt 1 type flag
TCON.1	IE0	External Interrupt 0 edge flag
TCON.0	IT0	External Interrupt 0 type flag

Timer Mode Register (TMOD)

- 8-bit mode register (@0x89) for both T0 & T1
  - 4 configuration bits for each timer

MSB LSB

GATE	C/T	M1	M0	GATE	C/T	M1	M0
Timer 1				Timer 0			

Bit	Name	Description
7	GATE	Gate bit. If set, timer 1 will only increment while INT1 is high.
6	C/T	Counter/timer select bit 1 = event counter – external timing signal 0 = interval timer – internal timing signal
5	M1	Mode bit 1
4	M0	Mode bit 0
3	GATE	Timer 0 Gate bit
2	C/T	Timer 0 counter/timer select bit
1	M1	Timer 0 M1 bit
0	M0	Timer 0 M0 bit

Gated Timer Control

- using GATE bit in TMOD
- when enabled (GATE = 1)
  - timer register is disabled when interrupt is asserted ($\overline{INTx}$ = 0), works normally otherwise
- allows timer/counter to be controlled by external source

Timer / Counter Select

- using the C/$\overline{T}$ bit in TMOD
- timer operation (C/$\overline{T}$ = 0)
  - timer register triggered by system clock (f_{osc}/12)
  - for interval timing
- counter operation (C/$\overline{T}$ = 1)
  - timer register triggered by external source (Tx pin)
  - event counting (negative-edge)

8051 Timer Modes

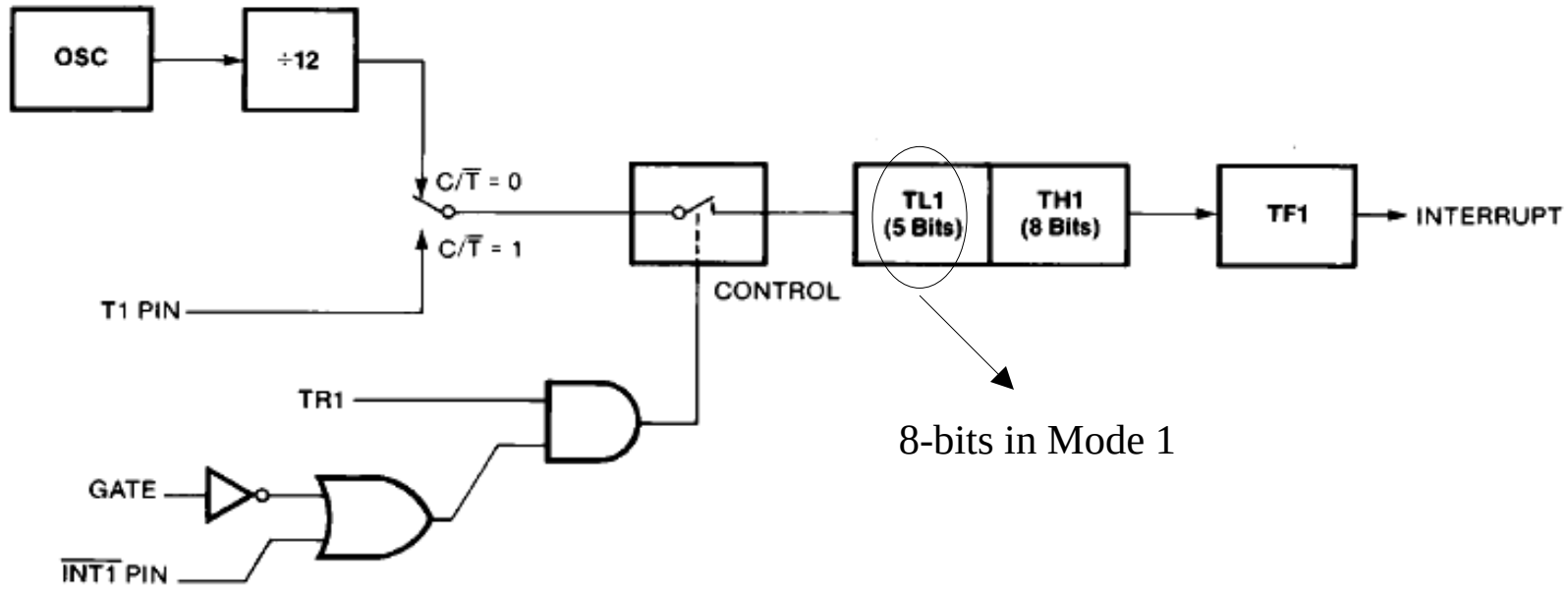
- 2-bits in TMOD {M1,M0} used to set operation mode:

M1	M0	Mode	Description
0	0	0	13-bit timer mode
0	1	1	16-bit timer mode
1	0	2	8-bit auto reload mode
1	1	3	Split timer mode: Timer 0: TL0 is an 8-bit timer controlled by timer 0 mode bits; TH0 the same except controlled by timer 1 mode bits. Timer 1: Stopped.

8051 Timer Mode 0/1

- Timer Mode 1
  - 16-bit timer/counter mode (most-used)
  - upper byte in THx , lower byte in Tlx
- Timer Mode 0
  - 13-bit timer/counter mode
  - backward compatibility to 8048
  - exactly like Mode 1, but upper 3-bits of TLx not used
- TFx flags will be set on 0xFFFF → 0x0000 transition (overflow)
  - timer does not stop here

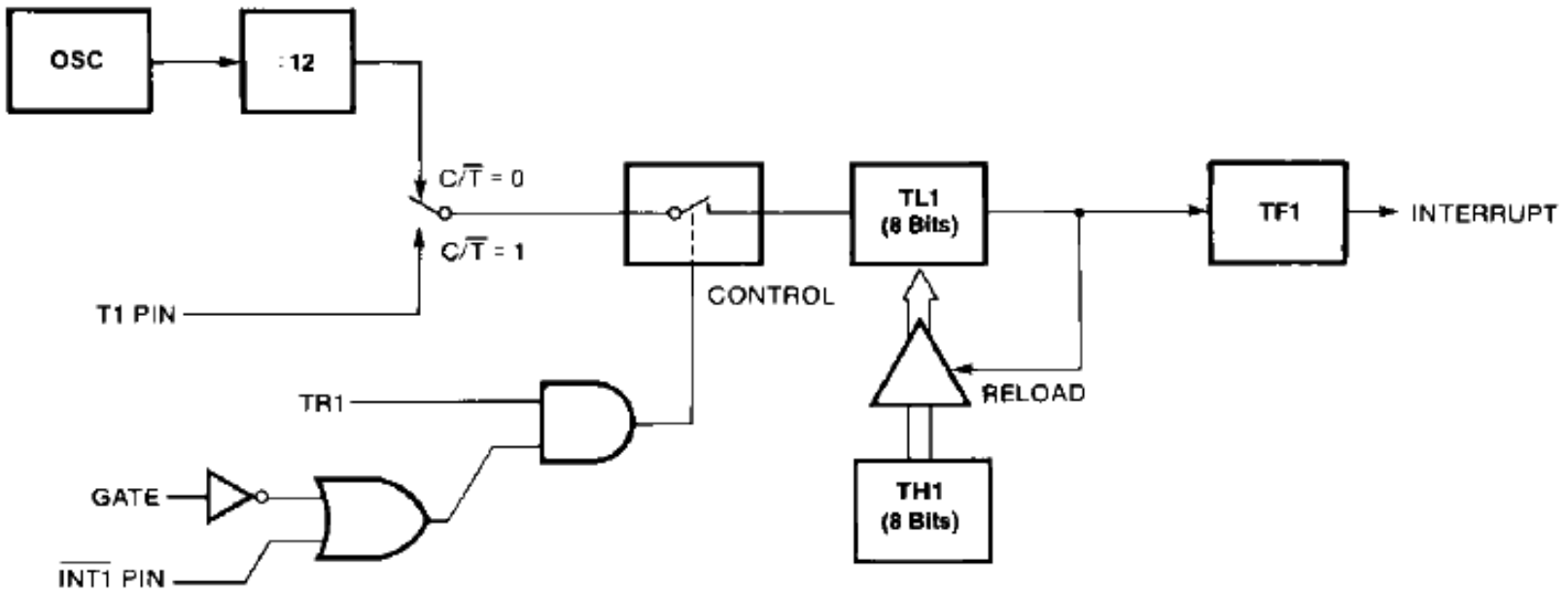
Timer Mode 0 / 1 - Diagram



8051 Timer Mode 2

- 8-bit auto-reload
  - TLx as counter register
  - THx holds reload value
  - TFx set on FF → 00 transition on TLx , TLx is then reloaded with value from Thx
  - useful for baudrate generator
- e.g. to keep counting 0x40 (64) times using T1,
  - TH1 = 0xC0 (0 - 0x40) TL1 = TH1 {SW}
  - set T1 to mode 2 {SW}
  - run T1 {SW}
  - TL1 = TH1 , repeats {HW}

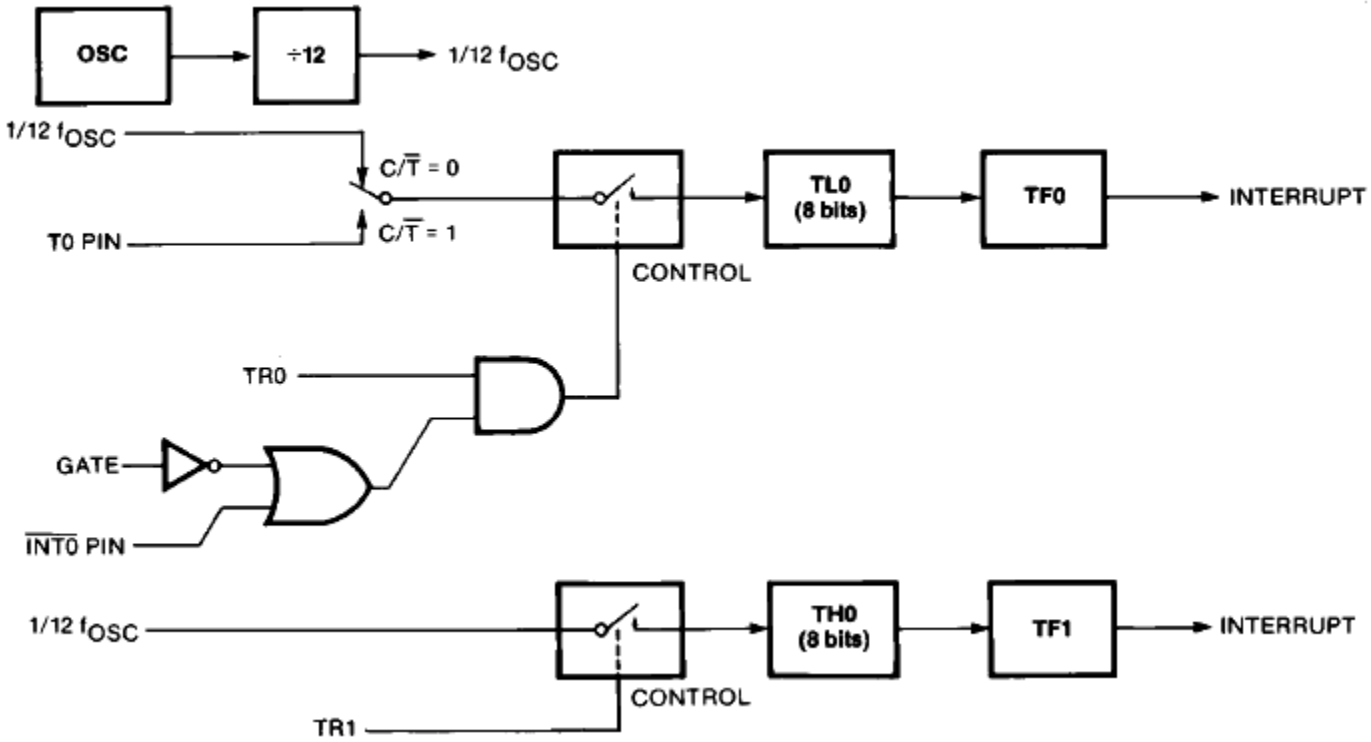
Timer Mode 2 - Diagram



8051 Timer Mode 3

- split timer
- T0 as 2 independent 8-bit timers
  - TF0 sets when TL0 overflows
  - TF1 sets when TH0 overflows
- T1 is stopped (holds value)
  - TR1 / TF1 used by TH0
  - still can switch to other modes (e.g. baudrate generator), but no control/interrupt

Timer Mode 3 - Diagram



Using 8051 Timer

- set the desired mode (e.g. timer mode 1)
- set a timer value (e.g. 100 machine cycles)
- start timer
- (optional) stop timer

```
// timer 0
//TMOD = 0x01;
TMOD &= 0xF0;
TMOD |= 0x01;
TH0 = 0xff;
TL0 = 0x9C;
TR0 = 1;
// TR0 = 0;
```

```
// timer 1
//TMOD = 0x10;
TMOD &= 0x0F;
TMOD |= 0x10;
TH1 = 0xff;
TL1 = 0x9C;
TR1 = 1;
// TR1 = 0;
```

Monitoring 8051 Timer

- can be interrupt-based (@event-based)
  - ISR (@event handler) is vectored (ref. Interrupt)
- simply monitor timer flag bit TFx
  - not cleared by hardware

```
// T0 run,wait,stop
TR0 = 1;
while (TF0!=1);
TR0 = 0;
TF0 = 0;
```

```
// T1 run,wait,stop
TR1 = 1;
while (TF1!=1);
TR1 = 0;
TF1 = 0;
```

Calc1: Timer 0 Mode 1

- need 50ms delay, f_{osc} = 11.0592MHz

Required delay, T_{delay} = 50ms

Count rate, F_{count} = 11.0592MHz/12

1-count time, T_{count} = 12/11.0592MHz

Delay count, $D_{count} = T_{delay} / T_{count}$

D_{count} = 50ms / (12 / 11.0592MHz)

= 50ms / 1.085µs

= 46082.9

≈ 46083 (require integer value)

Timer overflow, T_{ovf} = 65536

Timer value, $T_{val} = T_{ovf} - D_{count}$

T_{val} = 65536 – 46083

= 19453 @ 0x4bfd

Note: some uses T_{ovf} = 65535 (Tval = 19452 (@0x4bfc) → OK!

So TH0 = 0x4b and TL0 = 0xfd

Code1: Timer 0 Mode 1 (prev)

```
// timer-based (T0)
// 50ms delay function
void delay_50ms(void) {
  TH0 = 0x4b;
  TL0 = 0xfd;
  TR0 = 1;
  while (TF0==0);
  TR0 = 0;
  TF0 = 0;
}
```

```
// timer-based (T1?)
// 50ms delay function


    /*TRY THIS!*/
```

Calc2: Timer 0 Mode 1

- need to generate square-wave output
- 1KHz at P1.0

SQ-wave Period, T_{sqw} = 1/1KHz

= 1ms

Assume 50% D.C.

Required delay, T_{delay} = 0.5ms

F_{osc} not given, assume 12MHz

Count rate, F_{count} = 12MHz/12

= 1µs

Delay count, $D_{count} = T_{delay} / T_{count}$

D_{count} = 0.5ms / 1µs

= 500

Timer overflow, T_{ovf} = 65536

Timer value, $T_{val} = T_{ovf} - D_{count}$

T_{val} = 65536 – 500

= 65036 @ 0xfe0c

Note: some uses T_{ovf} = 65535 (Tval = 65035 (@0xfe0b) → OK!

So TH0 = 0xfe and TL0 = 0x0c

Code2: Timer 0 Mode 1 (prev)

```
#include <reg51.h>
sbit wave = P1^0;

void delay_500us(void)
{
  TH0 = 0xfe;
  TL0 = 0x0c;
  TR0 = 1;
  while (TF0==0);
  TR0 = 0;
  TF0 = 0;
}
```

```
void main (void) {
  TMOD = 0x11;
  while (1) {
    wave = ~wave;
    delay_500us();
  }
}
```

Code51 Task: Generate alternating blinking LEDs (1Hz) connected to P2

8051 Timer as Counter

- instead of time-triggered, using events
  - e.g. object sensor giving single pulse per object
  - hardwired edge-trigger
- set C/$\overline{T}$ bit to 1 to select / enable
- events from external (pins)
  - P3.4 (T0) : external pulse for Timer/Counter 0
  - P3.5 (T1) : external pulse for Timer/Counter 1

Code51 Task: Count object on conveyor system (sensor produces 1 pulse for each object)

End of Lecture05