SYMBIOSIS INTERNATIONAL (DEEMED UNIVERSITY)

॥वसुधैव कुटुम्बकम्॥

50 Years of Excellence Since 1971

Microcontrollers and Embedded Systems

Unit II: Internal Peripherals

Symbiosis Institute of Technology, Nagpur

Symbiosis Institute of Technology, Nagpur Campus (SIT-N)

- **Internal Peripherals**

Study of SFRs and Block diagram of the following peripherals: I/O Ports, Interrupts, Timers/Counters, UART. Assembly programming with 8051: I/O Programming, Timers/counters, Serial Communication, Interrupts.

8051 Timer/Counter

- Many microcontroller applications require the counting of external events, such as the frequency of a pulse train, or the generation of time delays . These task can be accomplished using software techniques
- But the software loops for counting or timing keep the processor occupied so that other functions are not done
- To relieve the processor of this burden, two 16-bit up counters/timers (T0 and T1) are provided in 8051 .
- Each timer can also act as seperate 8 bit registers.

8051 Timer/Counter

- Each counter may be programmed to count internal clock pulse (or to generate time delay), acting as a timer, or programmed to count external pulses as a counter.
- The counters are divided into two 8-bit registers called the timer low(TL0 for timer 0 and TL1 for timer 1) and high (TH0 for timer 0 and TH1 for timer 1) bytes.
- Counters always count up.
- All counter action is controlled by bit states in the timer mode control register (TMOD), and timer/counter control register (TCON), and certain instructions

8051 Timer/Counter

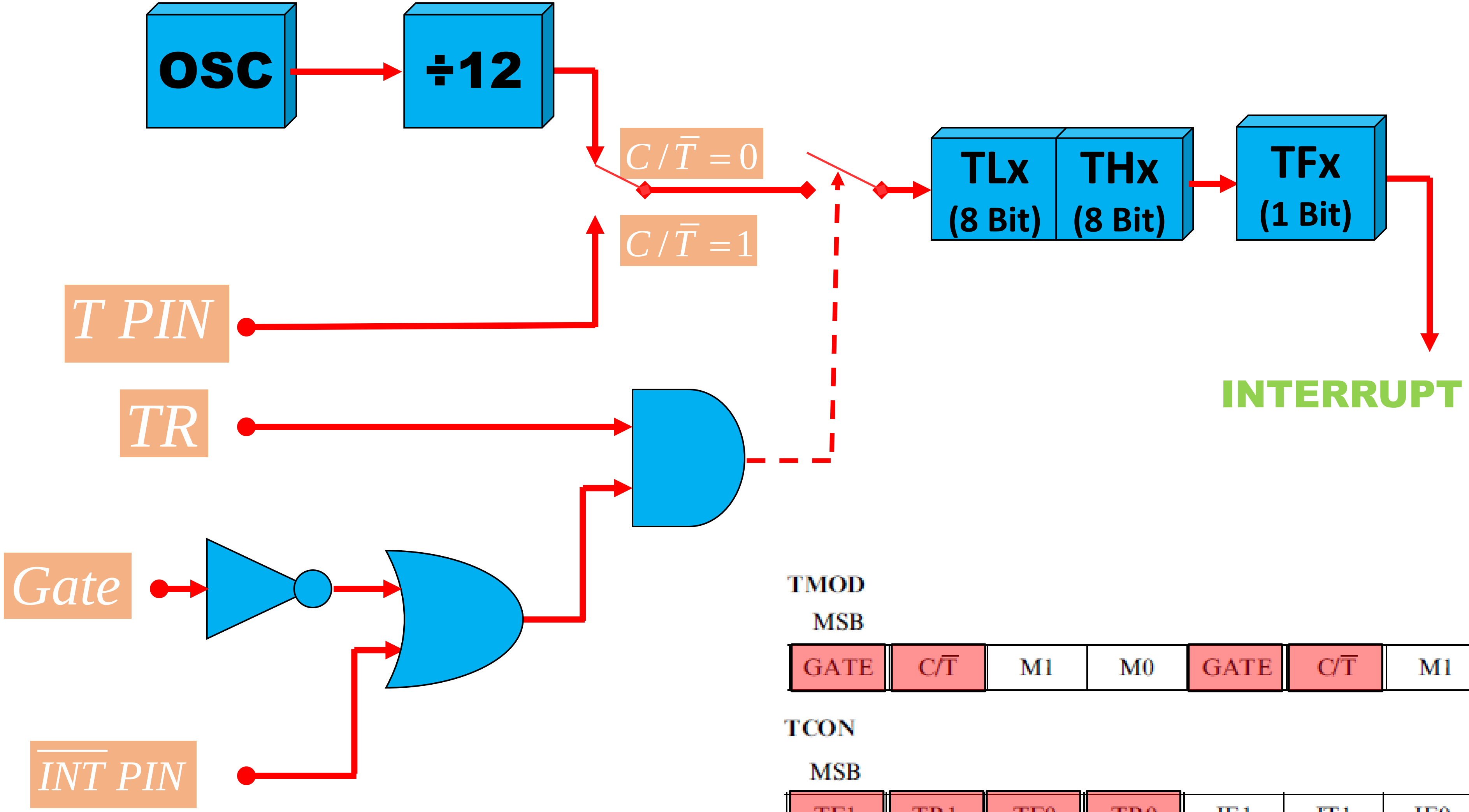
- When the program wants to count a certain number of internal pulses or external events, a number is placed in one of the counters.
- The counter increments from the initial number to the maximum and then rolls over to zero on the final pulse and set the timer flag.
- The flag condition may be tested by an instruction to tell that the count has been accomplished, or the flag may be used to interrupt the program

8051 Timer/Counter

Timer

- If the counter is programed to be a timer, it will count the internal clock frequency of the 8051-oscillator divided by 12.
- Ex: If the crystal frequency is 6 MHz, then the timer clock will have a frequency of 500 KHz.
- The oscillator clock pulse will reach the timer, if the C/T bit in the TMOD must be ‘0’, TR_X in the TCON must be 1(timer run) and the gate bit in the TMOD must be ‘0’ or external pin INT_X must be ‘1’

8051 Timer/Counter



TMOD

MSB							LSB
GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

TCON

MSB							LSB
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

8051 Timer/Counter

The Timer Mode Control (TMOD) Register

MSB LSB

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

89H

--------------- TIMER 1 --------------- --------------- TIMER 0 ---------------

GATE:

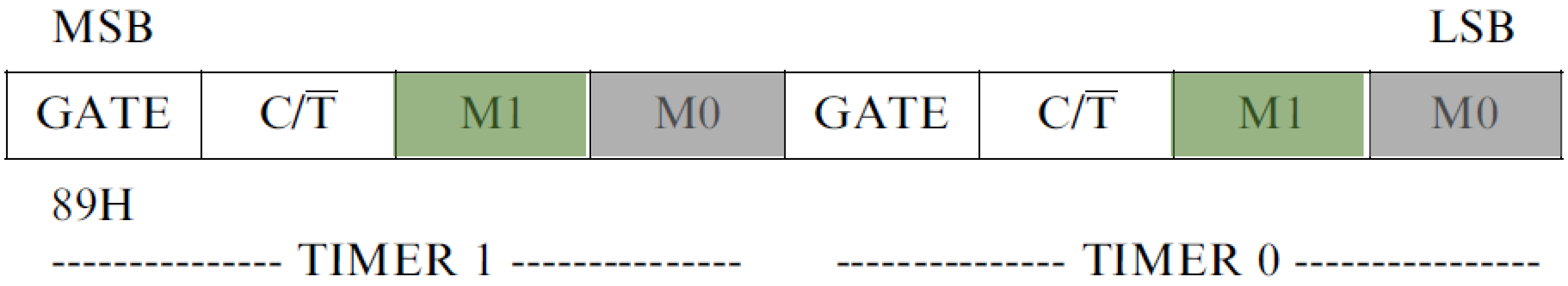
- **The bit which controls RUN/STOP of timer 0 /timer 1**
- **When set,** timer/counter x is enabled, **if INTx** pin is high and **TRx** is set.
- **When cleared,** timer/counter x is enabled, **if** TRx bit set.

C/T*:

When set, counter operation (input from Tx input pin).
When cleared, timer operation (input from internal clock).

8051 Timer/Counter

The Timer Mode Control (TMOD) Register



MSB LSB

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

89H

--------------- TIMER 1 --------------- --------------- TIMER 0 ---------------

The TMOD byte is not bit addressable.

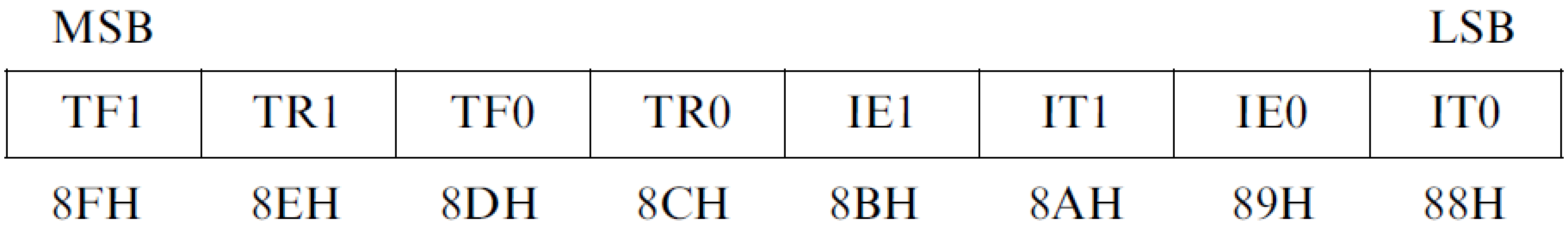
M1 , M0:

- **The timer/counter operating mode select bit. Set/Cleared by program to select mode**

M1	M0	Mode
0	0	0
0	1	1
1	0	2
1	1	3

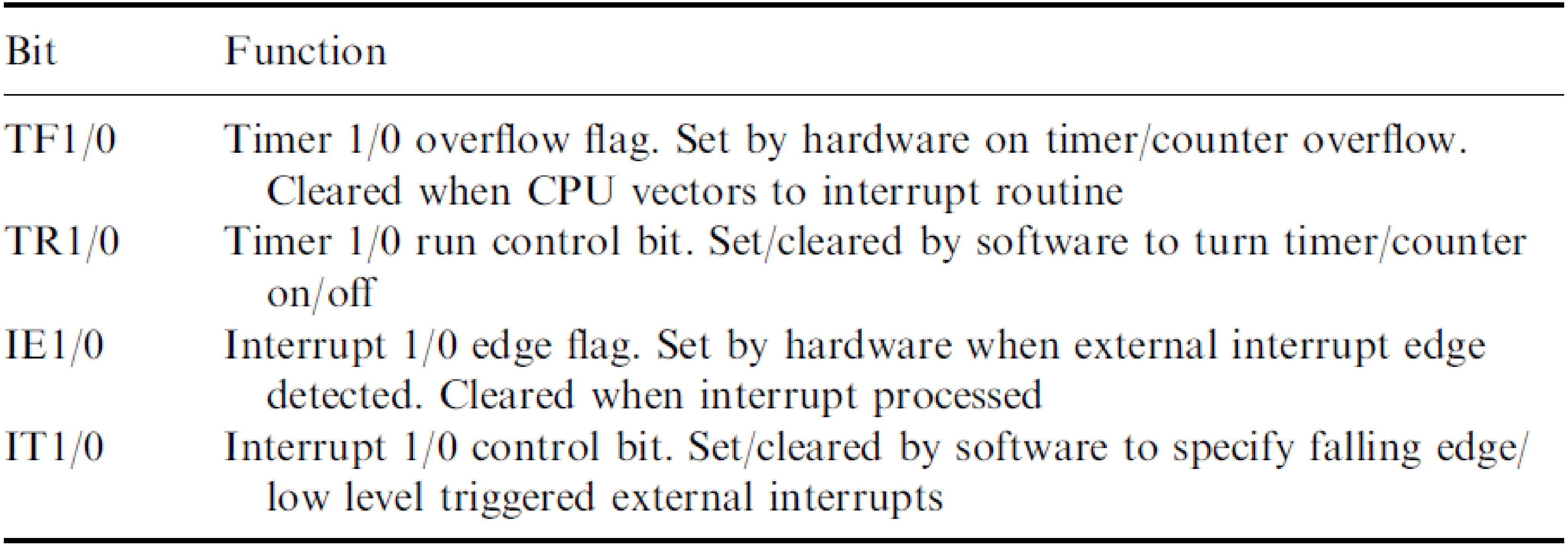
8051 Timer/Counter

The Timer Control (TCON) Register



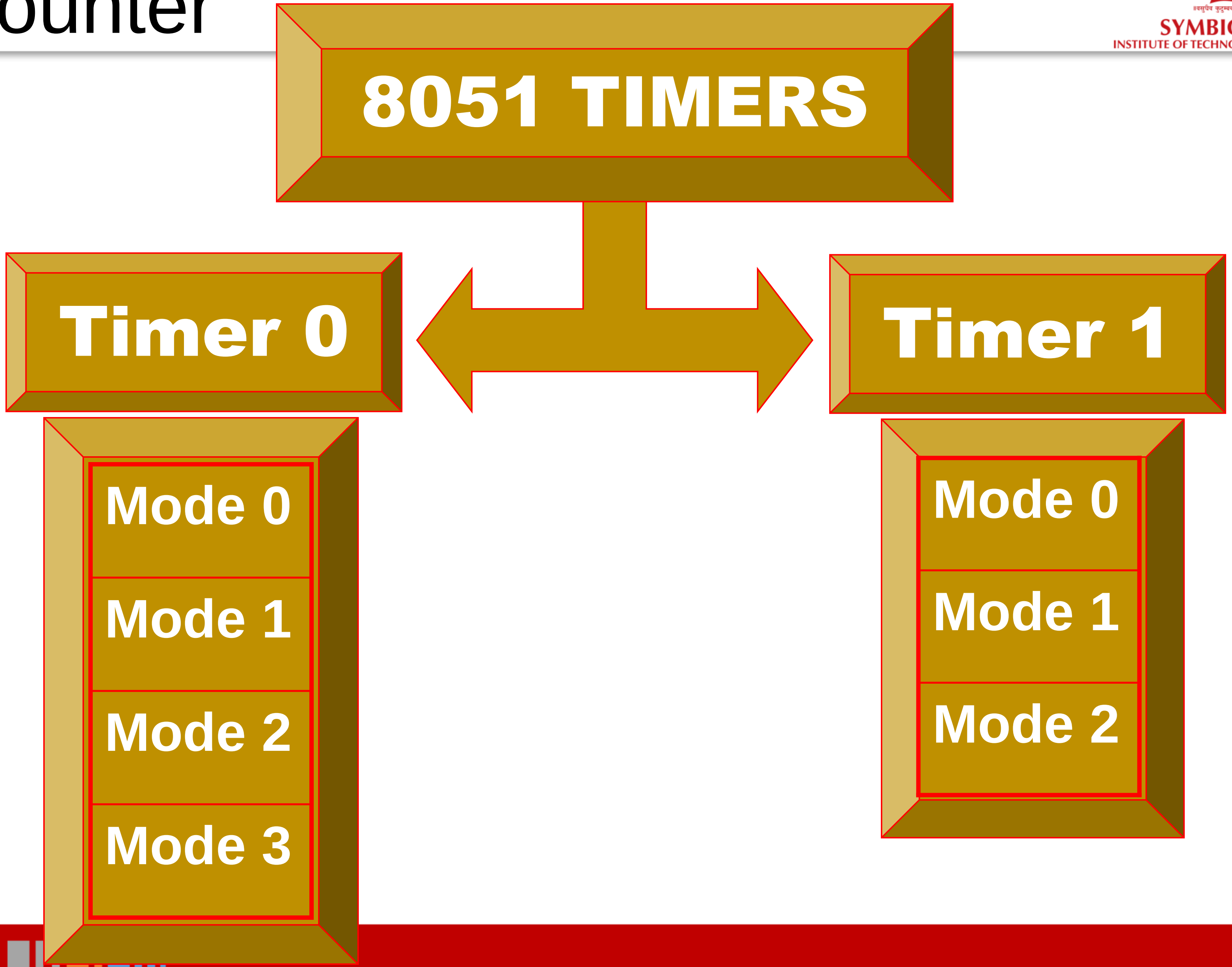
MSB							LSB
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
8FH	8EH	8DH	8CH	8BH	8AH	89H	88H

The TCON byte is bit addressable.



Bit	Function
TF1/0	Timer 1/0 overflow flag. Set by hardware on timer/counter overflow. Cleared when CPU vectors to interrupt routine
TR1/0	Timer 1/0 run control bit. Set/cleared by software to turn timer/counter on/off
IE1/0	Interrupt 1/0 edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed
IT1/0	Interrupt 1/0 control bit. Set/cleared by software to specify falling edge/ low level triggered external interrupts

8051 Timer/Counter



8051 TIMERS

Timer 0

Mode 0

Mode 1

Mode 2

Mode 3

Timer 1

Mode 0

Mode 1

Mode 2

8051 Timer/Counter

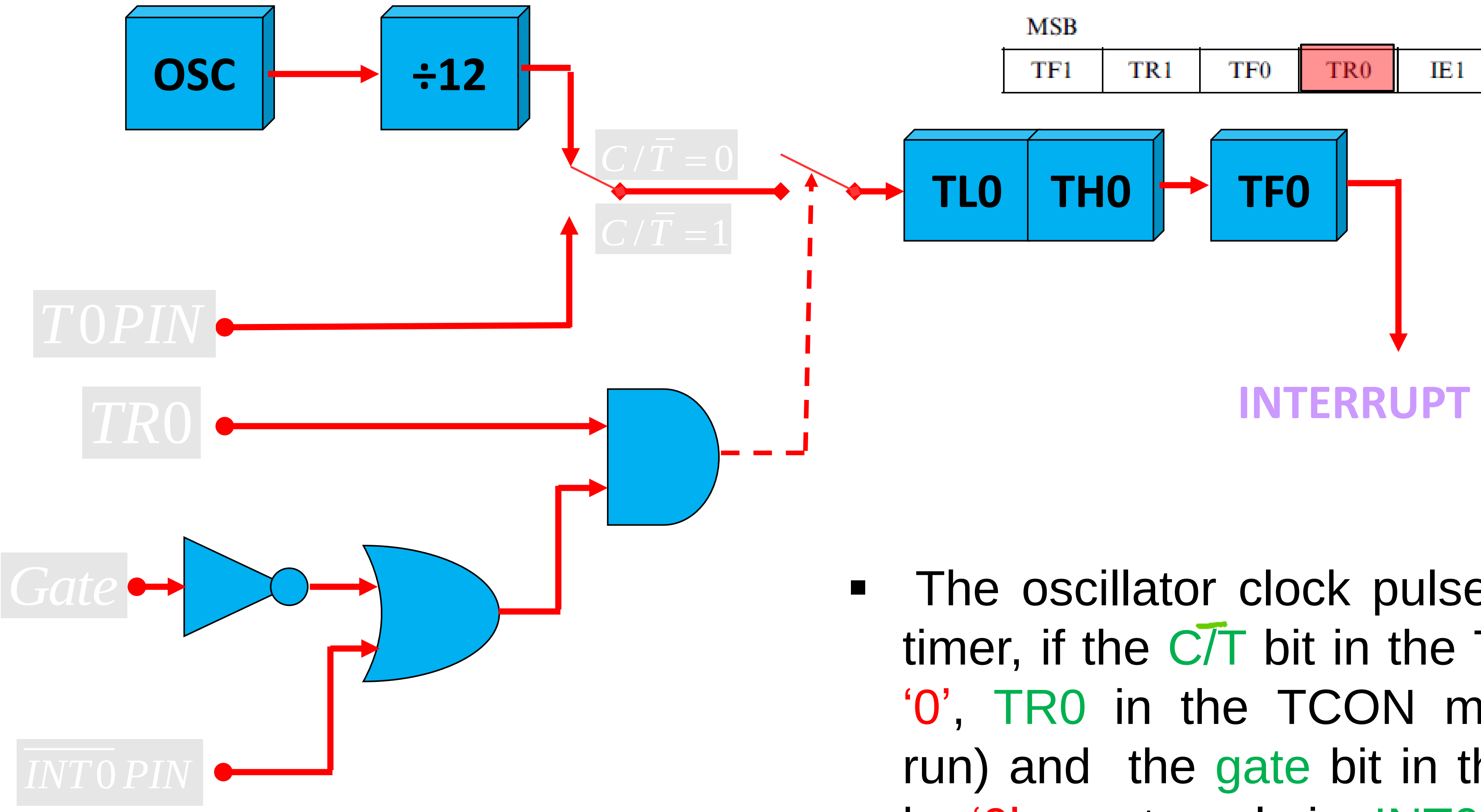
Timer 0

TMOD

MSB							LSB
GATE	$C/\overline{T}$	M1	M0	GATE	$C/\overline{T}$	M1	M0

TCON

MSB							LSB
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

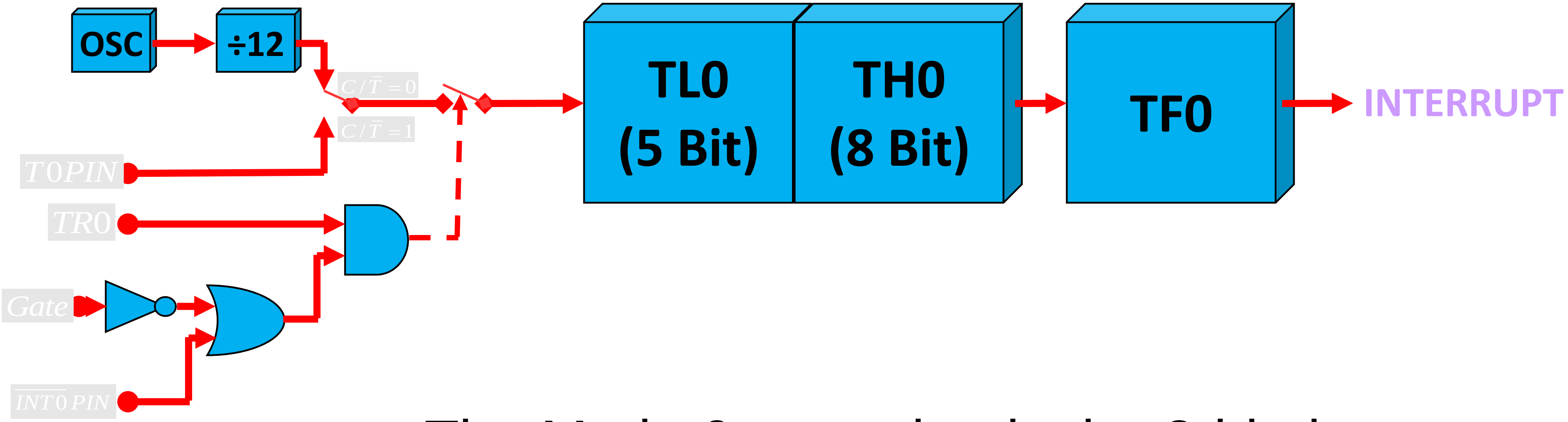


- The oscillator clock pulse will reach the timer, if the $C/\overline{T}$ bit in the TMOD must be '0', TR0 in the TCON must be 1(timer run) and the gate bit in the TMOD must be '0' or external pin INT0 must be '1

8051 Timer/Counter

Timer 0 – Mode 0

13 Bit Timer / Counter



M1	M0	Mode
0	0	0

- The Mode 0 operation is the 8-bit timer or counter with a 5-bit pre-scaler.
- So, it is a 13-bit timer/counter.
- It uses 5 bits of TL0 and all the 8-bits of TH0

Maximum Count = 1FFFh (0001111111111111)

8051 Timer/Counter

Timer 0 – Mode 0

- The pulse input is divided by 32 in TL0
- So, the TH count the original oscillator frequency reduced by 384 (12 x 32)
- If the crystal frequency is 6MHz, the final frequency to TH0 is 15625 Hz.
- Every 32 event for counter operations or 32 machine cycles for timer operation, the TH0 register will be incremented by 1.
- When the TH0 overflows from FFH to 00H, then the TF0 of TCON register will be high (set), and it stops the timer/counter.

Maximum Count = 1FFFh (0001111111111111)

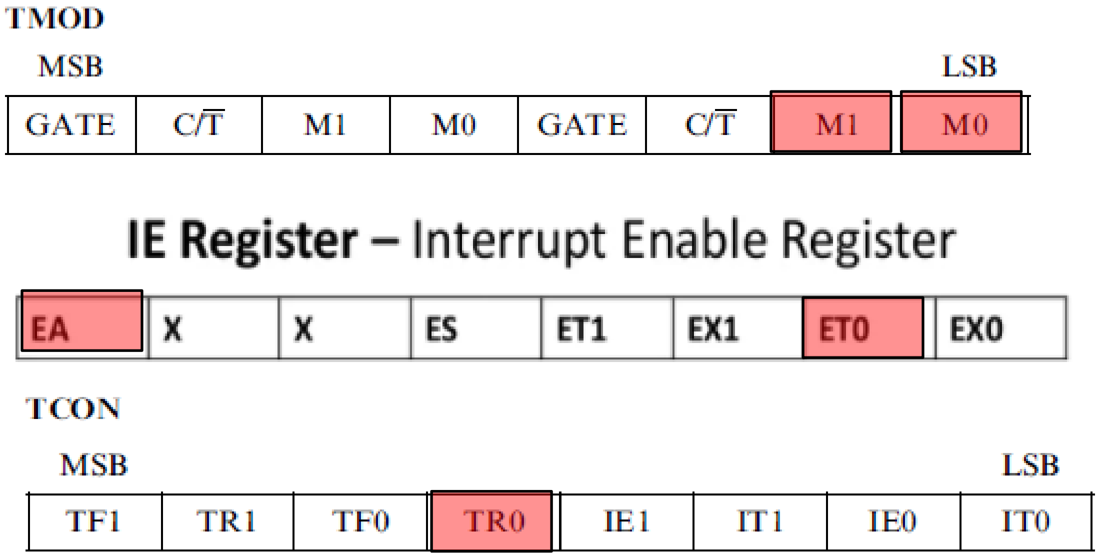
8051 Timer/Counter

Timer 0 – Mode 0

- Example:1

- If TH0 is holding F0H, and it is in timer mode, then TF0 will be high after 10H * 32 = 512 machine cycles. *10h = 16

```
MOV TMOD, #00H
MOV TH0,#0F0H
MOV IE, #82H
SETB TR0
```



TMOD

MSB							LSB
GATE	C/T̅	M1	M0	GATE	C/T̅	M1	M0

IE Register – Interrupt Enable Register

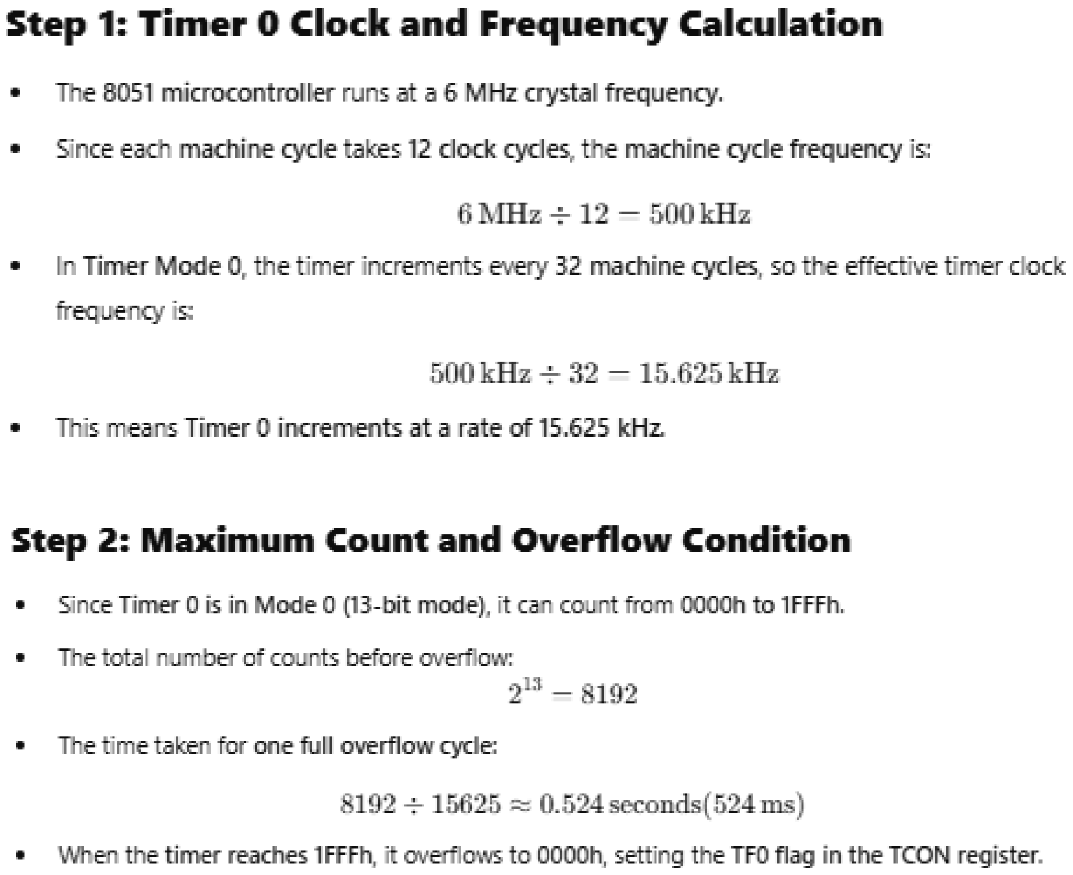
EA	X	X	ES	ET1	EX1	ET0	EX0

TCON

MSB							LSB
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

8051 Timer 0 in Mode 0 (13-bit Timer Mode)

In **Timer Mode 0**, **Timer 0** functions as a **13-bit counter**, meaning it can count from **0000h to 1FFFh** (0 to 8191 in decimal) before overflowing. The timer operates by incrementing every **32 machine cycles**.



Step 1: Timer 0 Clock and Frequency Calculation

- The 8051 microcontroller runs at a 6 MHz crystal frequency.
- Since each machine cycle takes 12 clock cycles, the machine cycle frequency is:

$$6\,\text{MHz} \div 12 = 500\,\text{kHz}$$

- In Timer Mode 0, the timer increments every 32 machine cycles, so the effective timer clock frequency is:

$$500\,\text{kHz} \div 32 = 15.625\,\text{kHz}$$

- This means Timer 0 increments at a rate of 15.625 kHz.

Step 2: Maximum Count and Overflow Condition

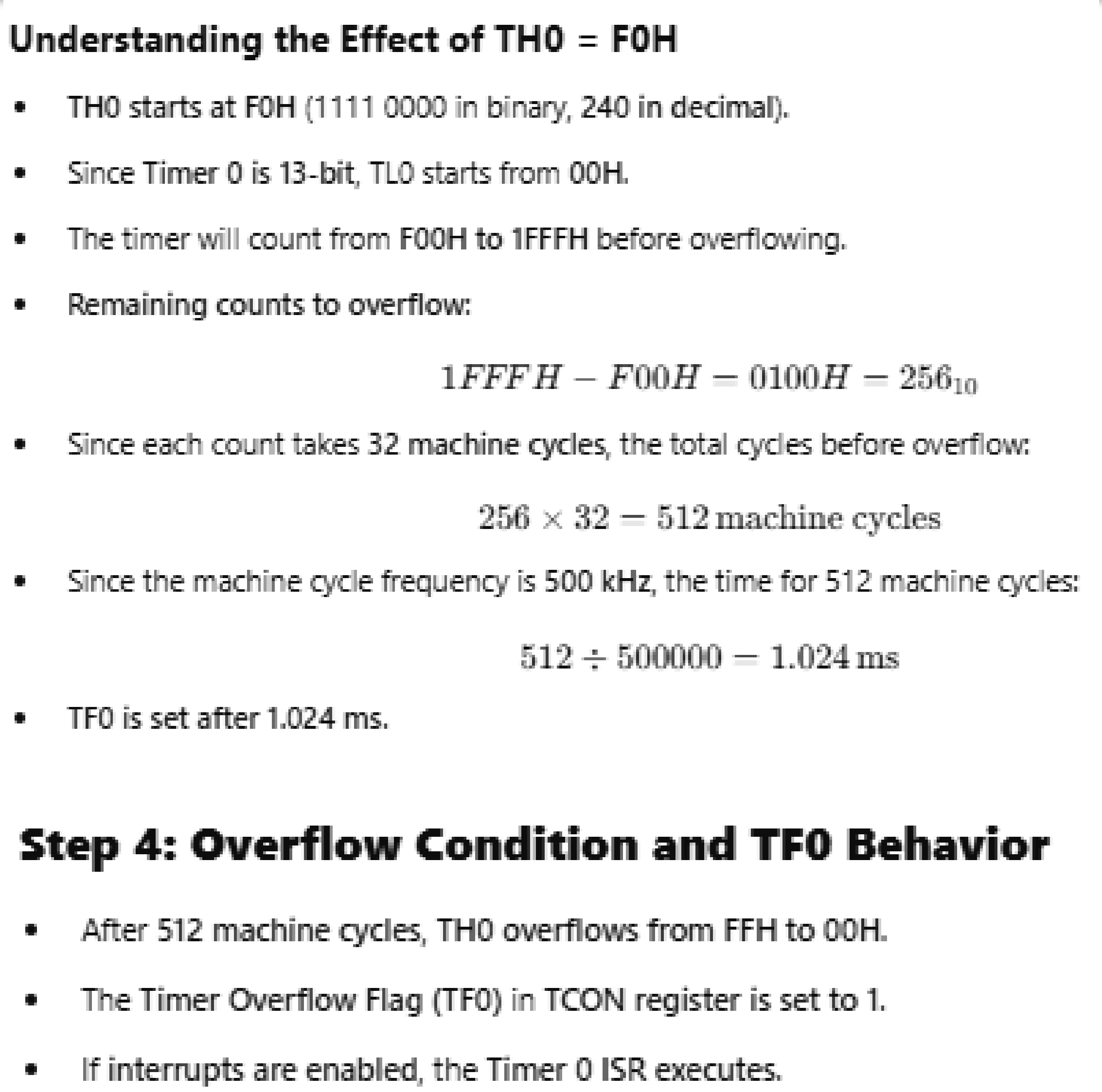
- Since Timer 0 is in Mode 0 (13-bit mode), it can count from 0000h to 1FFFh.
- The total number of counts before overflow:

$$2^{13} = 8192$$

- The time taken for one full overflow cycle:

$$8192 \div 15625 \approx 0.524\,\text{seconds}(524\,\text{ms})$$

- When the timer reaches 1FFFh, it overflows to 0000h, setting the TF0 flag in the TCON register.



Understanding the Effect of TH0 = F0H

- TH0 starts at F0H (1111 0000 in binary, 240 in decimal).
- Since Timer 0 is 13-bit, TL0 starts from 00H.
- The timer will count from F00H to 1FFFH before overflowing.
- Remaining counts to overflow:

$$1FFFH - F00H = 0100H = 256_{10}$$

- Since each count takes 32 machine cycles, the total cycles before overflow:

$$256 \times 32 = 512 \,\text{machine cycles}$$

- Since the machine cycle frequency is 500 kHz, the time for 512 machine cycles:

$$512 \div 500000 = 1.024 \,\text{ms}$$

- TF0 is set after 1.024 ms.

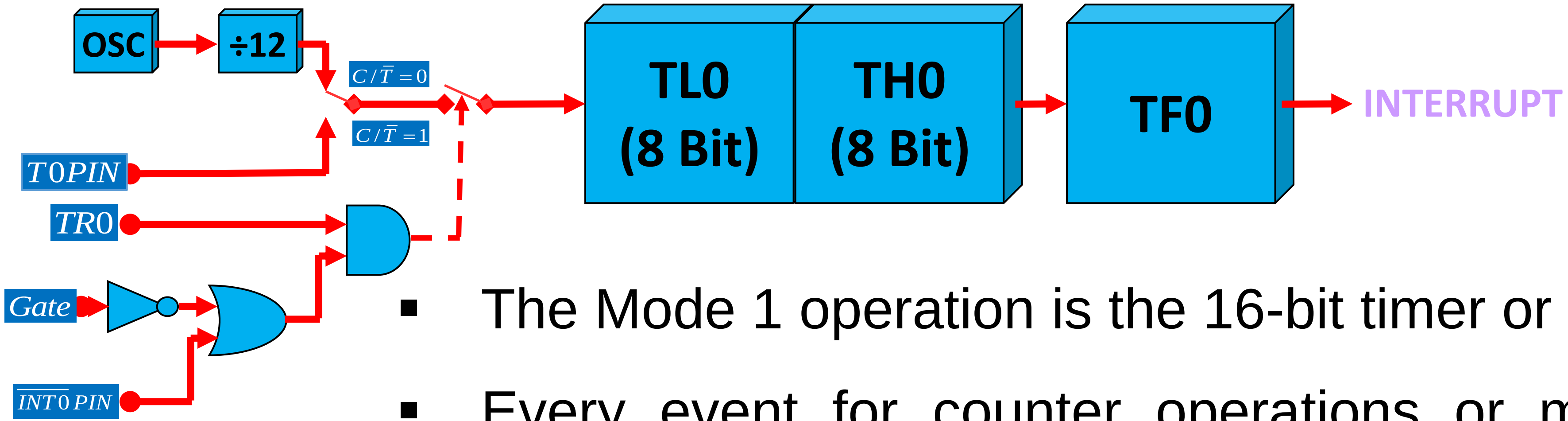
Step 4: Overflow Condition and TF0 Behavior

- After 512 machine cycles, TH0 overflows from FFH to 00H.
- The Timer Overflow Flag (TF0) in TCON register is set to 1.
- If interrupts are enabled, the Timer 0 ISR executes.

8051 Timer/Counter

Timer 0 – Mode 1

16 Bit Timer / Counter



M1	M0	Mode
0	1	1

- The Mode 1 operation is the 16-bit timer or counter.
- Every event for counter operations or machine cycles for timer operation, the TH0– TL0 register-pair will be incremented by 1.
- When the register pair overflows from FFFFH to 0000H, then the TF0 of TCON register will be high (set), and it stops the timer/counter.

8051 Timer/Counter

Timer 0 – Mode 1

- Example:1

  - If the TH0 – TL0 register pair is holding FFF0H, and it is in timer mode, then TF0 will be high after 10H = 16 machine cycles.
  - If the clock frequency is 12MHz, then the following instructions generate an interrupt 16 µs after Timer0 starts running.

```
MOV TMOD, #01H
MOV TL0, #0F0H
MOV TH0, #0FFH
MOV IE, #82H
SETB TR0
```

- Example:2

  - If TH0 – TL0 start at 0000H, then TF0 will be high (set) after 0.1311 seconds (if Crystal frequency is 6MHz)

Maximum Count = FFFFh (1111111111111111

8051 Timer 0 in Mode 1 (16-bit Timer Mode)

In **Timer Mode 1**, **Timer 0 operates as a full 16-bit timer**, meaning it counts from **0000h to FFFFh** (0 to 65535 in decimal) before overflowing. Let's analyze the scenario where **TH0 – TL0 holds FFF0H** and determine when **TF0 (Timer Overflow Flag)** is set.

Step 1: Timer Mode 1 Characteristics

- Timer 0 in Mode 1 acts as a 16-bit counter with a range 0000h to FFFFh (65536 counts)
- It increments every machine cycle (unlike Mode 0, which divides by 32).

Clock Frequency and Machine Cycle Calculation

- Given crystal frequency = 12 MHz
- Since each machine cycle = 12 clock cycles, the machine cycle frequency is:

$$12\,\mathrm{MHz} \div 12 = 1\,\mathrm{MHz}$$

- This means Timer 0 increments every 1 µs.

Step 2: Understanding TH0 = FFF0H Initialization

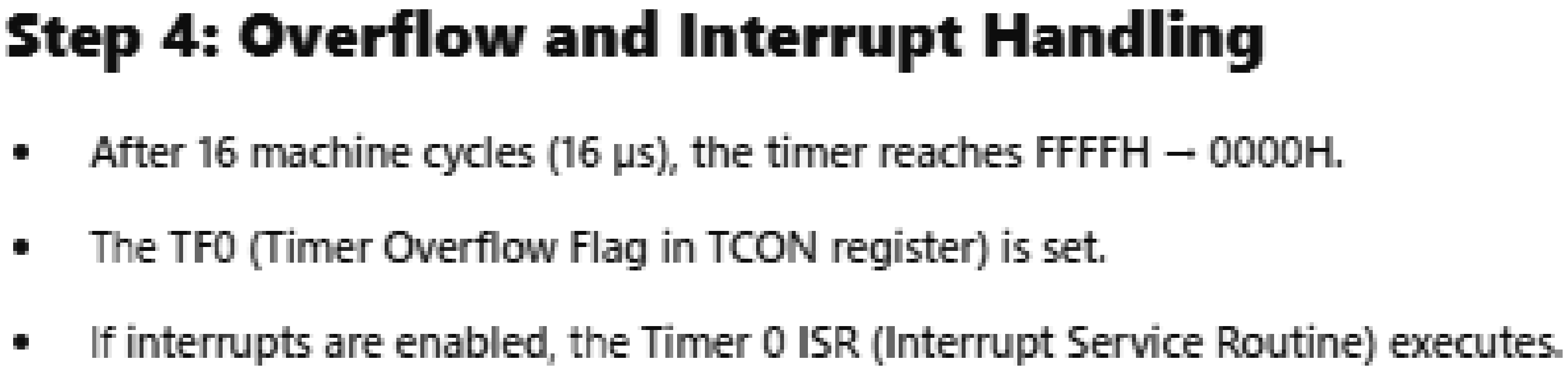
- Initial TH0 – TL0 = FFF0H (65520 in decimal).
- Final value before overflow = 10000H (65536 in decimal).
- Remaining counts before overflow:

$$10000H - FFF0H = 10H = 16_{10}$$

- Since each count takes 1 µs (from Step 1), the time to overflow is:

$$16 \times 1\,\mu\text{s} = 16\,\mu\text{s}$$

```
MOV TMOD, #01H   ; Timer 0, Mode 1 (16-bit timer mode)
MOV TH0, #0FFH   ; Load TH0 with FFH
MOV TL0, #0F0H   ; Load TL0 with F0H
MOV IE, #82H     ; Enable Timer 0 Interrupt (EA = 1, ET0 = 1)
SETB TR0         ; Start Timer 0
```

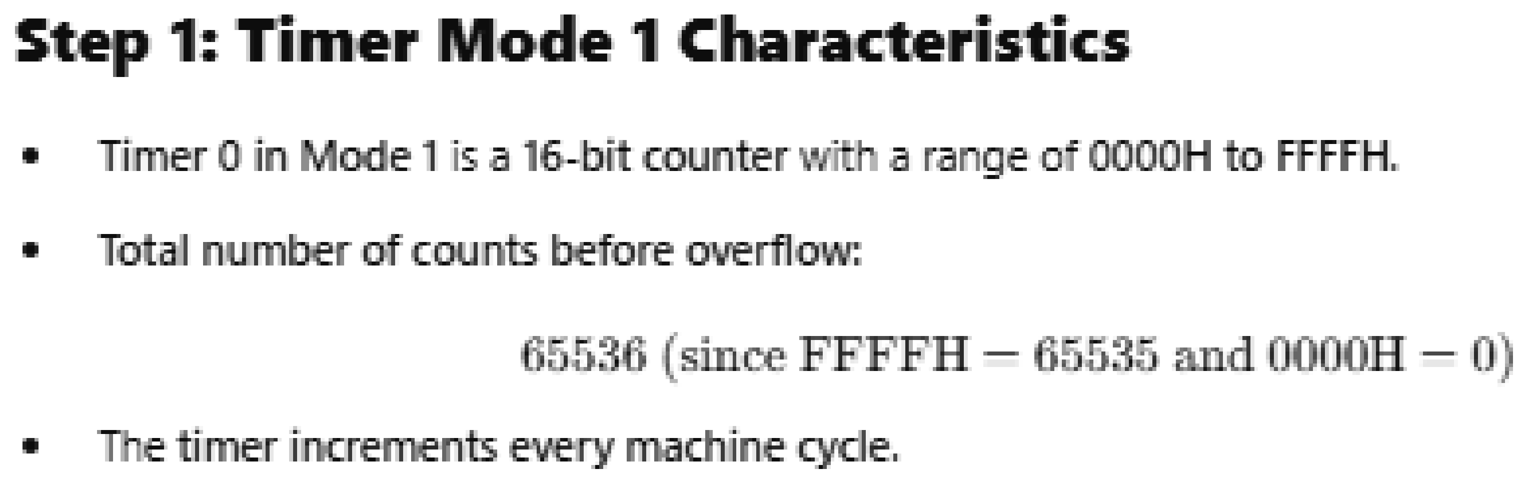


Step 4: Overflow and Interrupt Handling

- After 16 machine cycles (16 µs), the timer reaches FFFFH → 0000H.
- The TF0 (Timer Overflow Flag in TCON register) is set.
- If interrupts are enabled, the Timer 0 ISR (Interrupt Service Routine) executes.

8051 Timer 0 in Mode 1 (16-bit Timer Mode) – Example 2

In this example, **Timer 0 starts at 0000H** and runs until it overflows at **FFFFH**. We will calculate how long it takes for **TF0 (Timer Overflow Flag)** to be set, given that the **crystal frequency is 6 MHz**.



Step 1: Timer Mode 1 Characteristics

- Timer 0 in Mode 1 is a 16-bit counter with a range of 0000H to FFFFH.
- Total number of counts before overflow:

$$65536 \text{ (since FFFFH} = 65535 \text{ and 0000H} = 0)$$

- The timer increments every machine cycle.

Step 2: Clock Frequency and Machine Cycle Calculation

- Crystal frequency = 6 MHz
- Since one machine cycle = 12 clock cycles, the machine cycle frequency is:

$$6\,\text{MHz} \div 12 = 500\,\text{kHz}$$

- This means Timer 0 increments every:

$$1 \div 500000 = 2\,\mu\text{s}\ (\text{per count})$$

Step 3: Time Calculation Until Overflow

- Since the timer starts at 0000H (0 decimal) and overflows after 65536 counts, the total time taken is:

$$65536 \times 2\,\mu\text{s} = 131072\,\mu\text{s} = 0.1311\,\text{seconds}$$

```
MOV TMOD, #01H   ; Timer 0, Mode 1 (16-bit mode)
MOV TH0, #00H    ; Load TH0 with 00H
MOV TL0, #00H    ; Load TL0 with 00H
MOV IE, #82H     ; Enable Timer 0 Interrupt (EA = 1, ET0 = 1)
SETB TR0         ; Start Timer 0
```

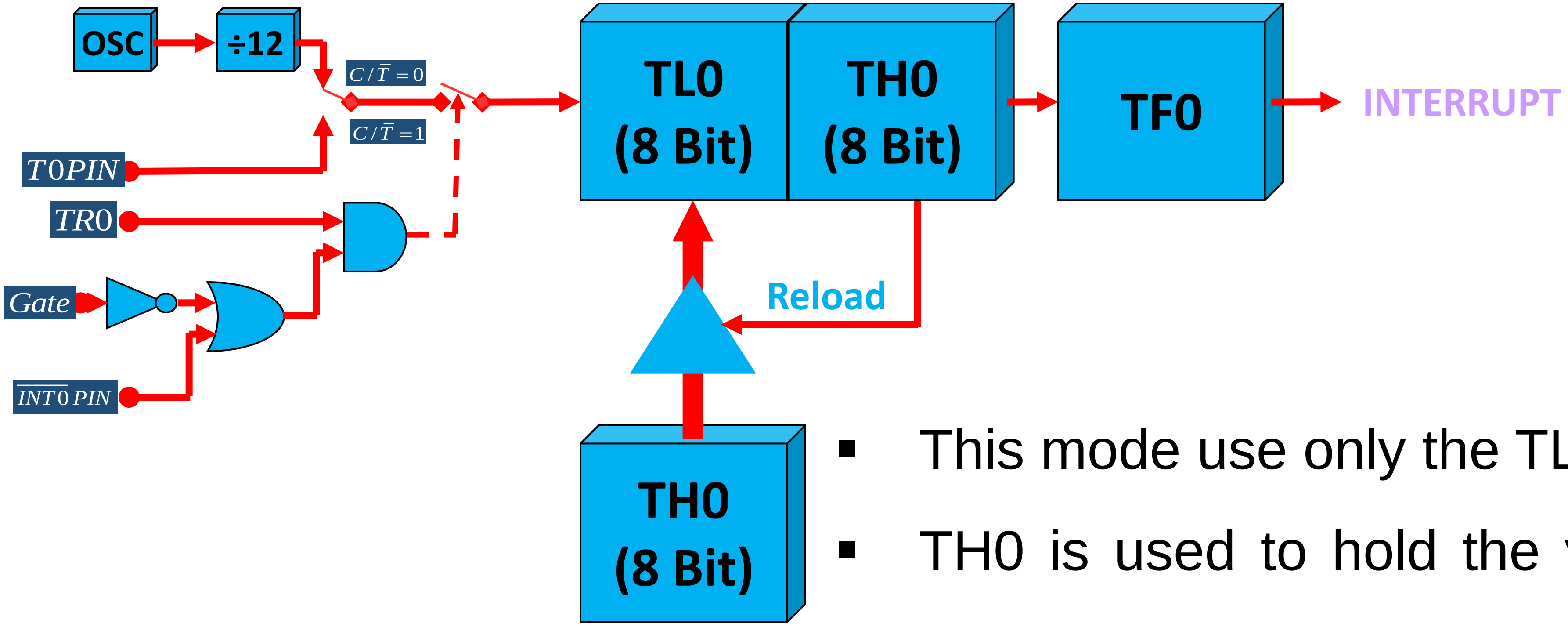
- If **interrupts are enabled**, the **Timer 0 ISR will execute**.
- **Timer 0 starts at 0000H.**
- **TF0 will be set after 0.1311 seconds (131.1 ms).**

8051 Timer/Counter

SYMBIOSIS
INSTITUTE OF TECHNOLOGY, NAGPUR

Timer 0 – Mode 2

8 Bit Timer / Counter with Auto-reload



M1	M0	Mode
1	0	2

- This mode use only the TL0 as an 8-bit counter
- TH0 is used to hold the value that is loaded into TL0 every time TL0 overflow from FFh to 00h
- The timer flag TF0 set when TL0 overflows

8051 Timer/Counter

Timer 0 – Mode 2

- In mode 2, every event for counter operations or machine cycles for timer operation, the TL0 register will be incremented by 1.
- When the register pair overflows from FFH to 00H, then the TF0 of TCON register will be high (set), also theTL0 will be reloaded with the content of TH0 and starts the operation again.
- This feature is known as auto - reload

Maximum Count = FFh (11111111)

8051 Timer/Counter

Timer 0 – Mode 2

- Example:1

  - if the TH0 and TL0 register both are holding F0H and it is in timer mode, then TF0 will be high after 10H= 16 machine cycles.
  - If the clock frequency is 12MHz this happens after 16 µs, then the following instructions generate an interrupt once every 16 µs after Timer0 starts running.

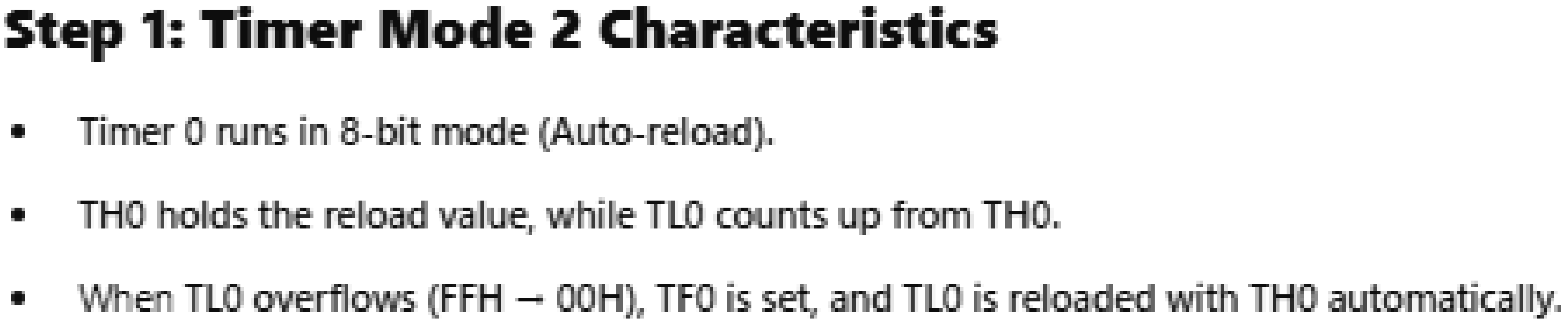
```
MOV TMOD, #02H
MOV TL0, #0F0H
MOV TH0, #0F0H
MOV IE, #82H
SETB TR0
```

- Example:2

  - Placing 9Ch in TH0 will result in a delay of 0.0002 seconds before the overflow flag is set if 6MHz crystal is used

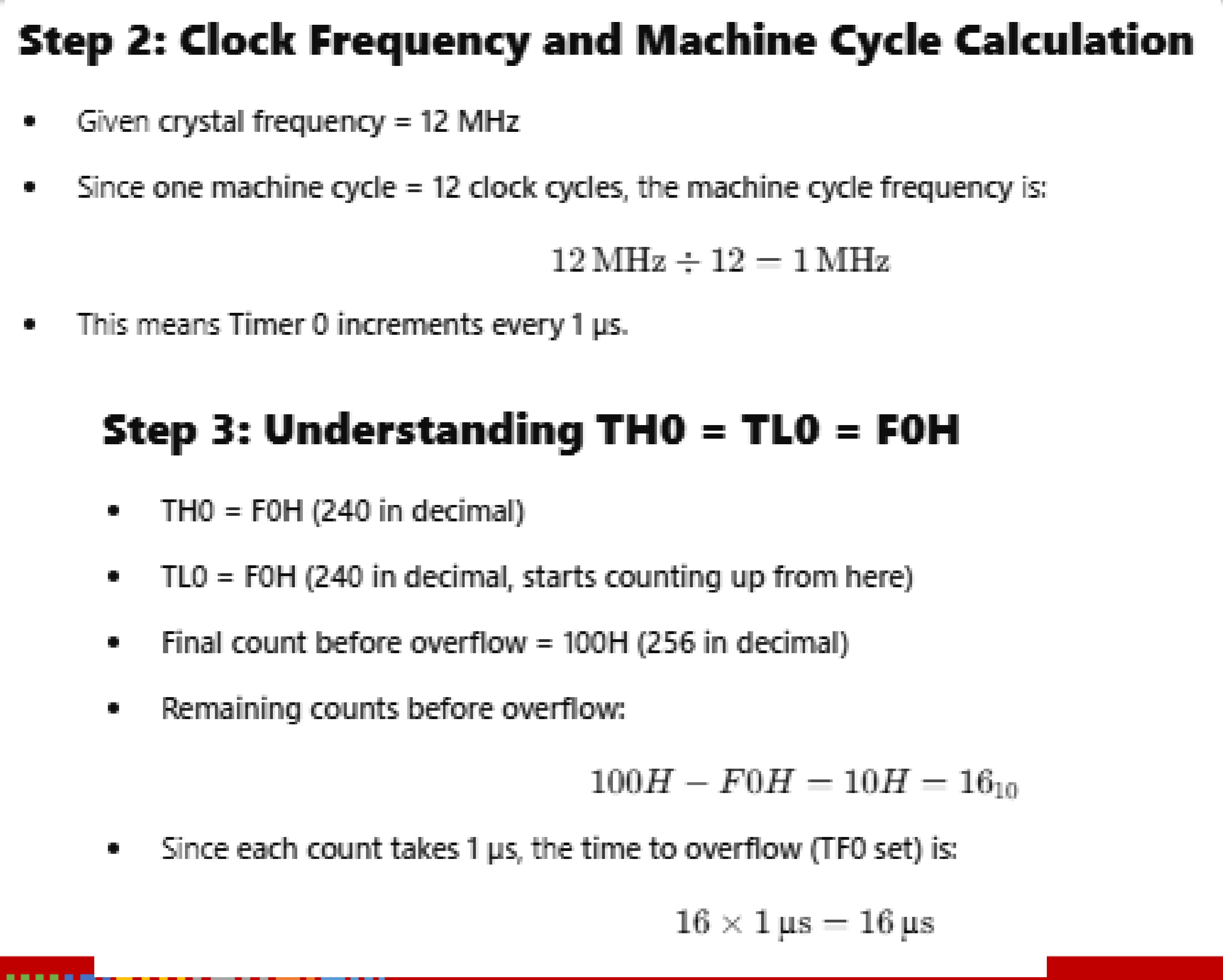
8051 Timer 0 in Mode 2 (8-bit Auto-Reload Mode) – Example 1

In **Timer Mode 2**, **Timer 0 operates as an 8-bit auto-reload timer**, meaning that **TH0 holds the reload value**, and **TL0 is incremented** until it overflows. When **TL0 overflows (reaches FFH → 00H), TF0 is set, and TL0 is automatically reloaded with TH0**.



Step 1: Timer Mode 2 Characteristics

- Timer 0 runs in 8-bit mode (Auto-reload).
- TH0 holds the reload value, while TL0 counts up from TH0.
- When TL0 overflows (FFH → 00H), TF0 is set, and TL0 is reloaded with TH0 automatically.



Step 2: Clock Frequency and Machine Cycle Calculation

- Given crystal frequency = 12 MHz
- Since one machine cycle = 12 clock cycles, the machine cycle frequency is:

$$12\,\text{MHz} \div 12 = 1\,\text{MHz}$$

- This means Timer 0 increments every 1 μs.

Step 3: Understanding TH0 = TL0 = F0H

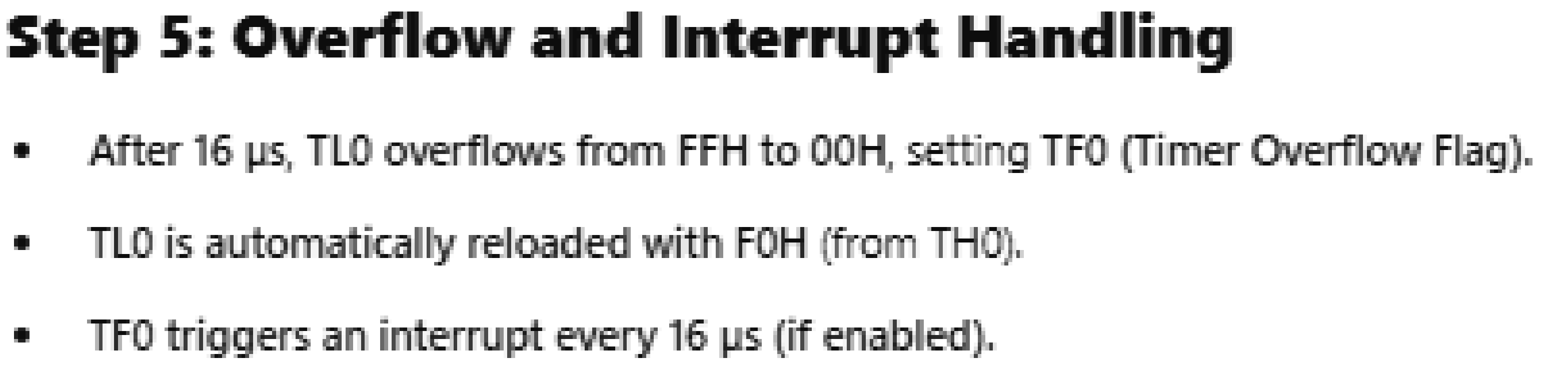
- TH0 = F0H (240 in decimal)
- TL0 = F0H (240 in decimal, starts counting up from here)
- Final count before overflow = 100H (256 in decimal)
- Remaining counts before overflow:

$$100H - F0H = 10H = 16_{10}$$

- Since each count takes 1 μs, the time to overflow (TF0 set) is:

$$16 \times 1\,\mu\text{s} = 16\,\mu\text{s}$$

```
MOV TMOD, #02H   ; Timer 0, Mode 2 (8-bit auto-reload mode)
MOV TH0, #0F0H   ; Load TH0 with F0H (reload value)
MOV TL0, #0F0H   ; Load TL0 with F0H (initial count)
MOV IE, #82H     ; Enable Timer 0 Interrupt (EA = 1, ET0 = 1)
SETB TR0         ; Start Timer 0
```

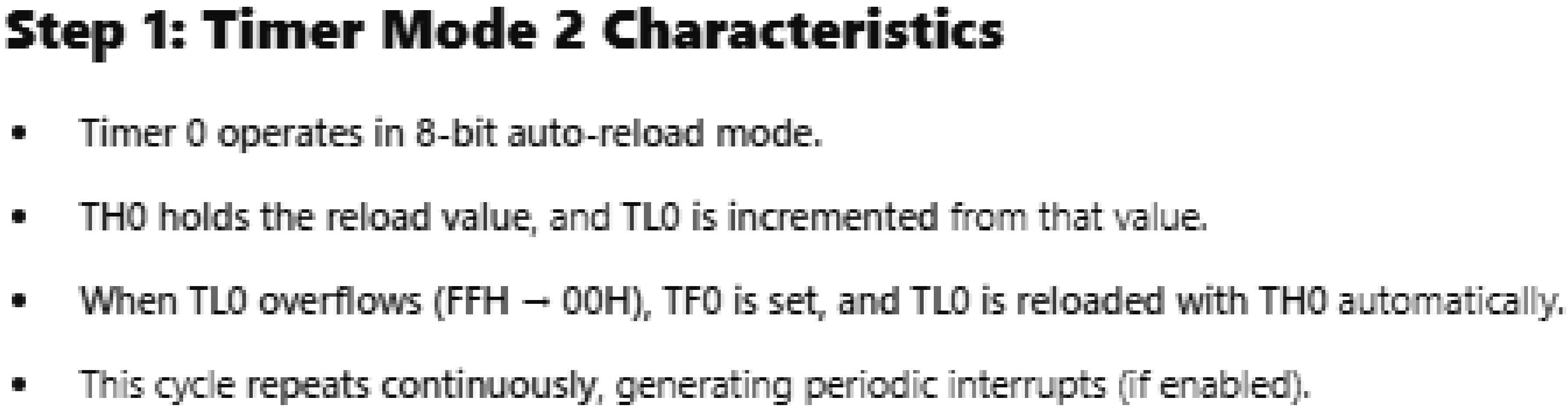


Step 5: Overflow and Interrupt Handling

- After 16 µs, TL0 overflows from FFH to 00H, setting TF0 (Timer Overflow Flag).
- TL0 is automatically reloaded with F0H (from TH0).
- TF0 triggers an interrupt every 16 µs (if enabled).

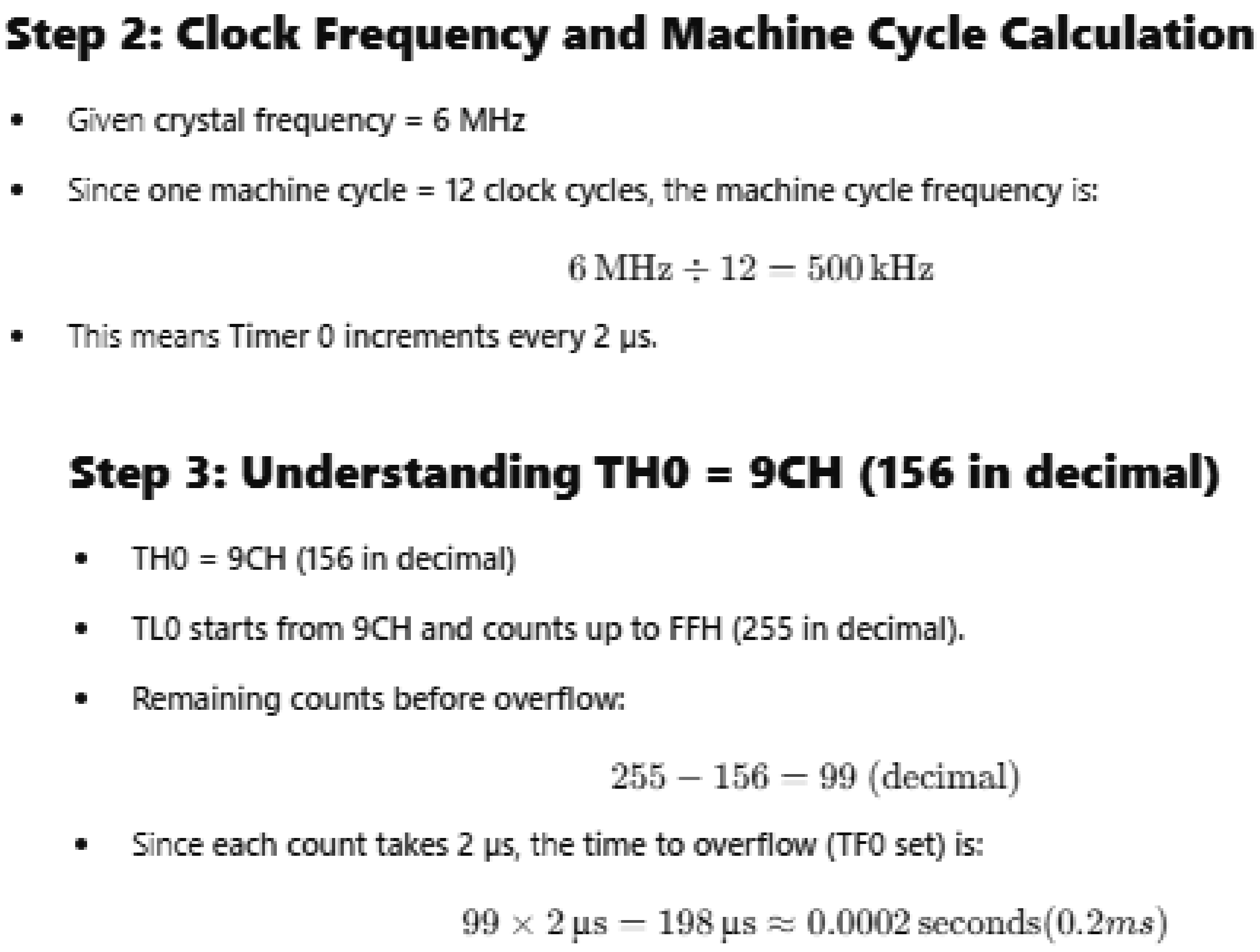
8051 Timer 0 in Mode 2 (8-bit Auto-Reload Mode) – Example 2

In **Timer Mode 2 (8-bit Auto-Reload Mode)**, **TH0 holds the reload value, and TL0 is incremented until it overflows**. Once **TL0 overflows from FFH to 00H**, the **TF0 flag is set**, and **TL0 is automatically reloaded with TH0**.



Step 1: Timer Mode 2 Characteristics

- Timer 0 operates in 8-bit auto-reload mode.
- TH0 holds the reload value, and TL0 is incremented from that value.
- When TL0 overflows (FFH → 00H), TF0 is set, and TL0 is reloaded with TH0 automatically.
- This cycle repeats continuously, generating periodic interrupts (if enabled).



Step 2: Clock Frequency and Machine Cycle Calculation

- Given crystal frequency = 6 MHz
- Since one machine cycle = 12 clock cycles, the machine cycle frequency is:

$$6\,\mathrm{MHz} \div 12 = 500\,\mathrm{kHz}$$

- This means Timer 0 increments every 2 μs.

Step 3: Understanding TH0 = 9CH (156 in decimal)

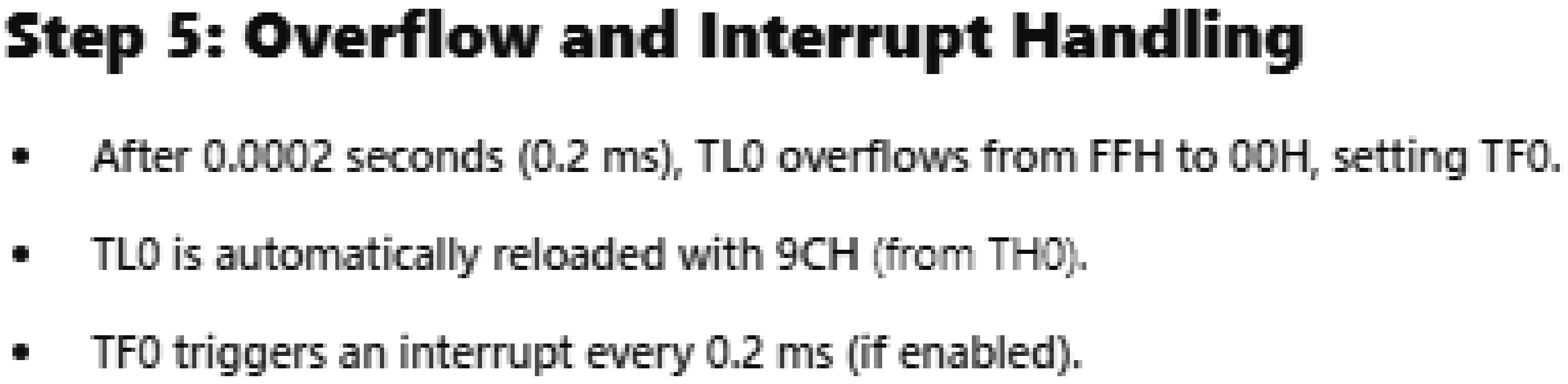
- TH0 = 9CH (156 in decimal)
- TL0 starts from 9CH and counts up to FFH (255 in decimal).
- Remaining counts before overflow:

$$255 - 156 = 99\ (\mathrm{decimal})$$

- Since each count takes 2 μs, the time to overflow (TF0 set) is:

$$99 \times 2\,\mu\mathrm{s} = 198\,\mu\mathrm{s} \approx 0.0002\,\mathrm{seconds}(0.2ms)$$

```
MOV TMOD, #02H   ; Timer 0, Mode 2 (8-bit auto-reload mode)
MOV TH0, #09CH   ; Load TH0 with 9CH (reload value)
MOV TL0, #09CH   ; Load TL0 with 9CH (initial count)
MOV IE, #82H     ; Enable Timer 0 Interrupt (EA = 1, ET0 = 1)
SETB TR0         ; Start Timer 0
```

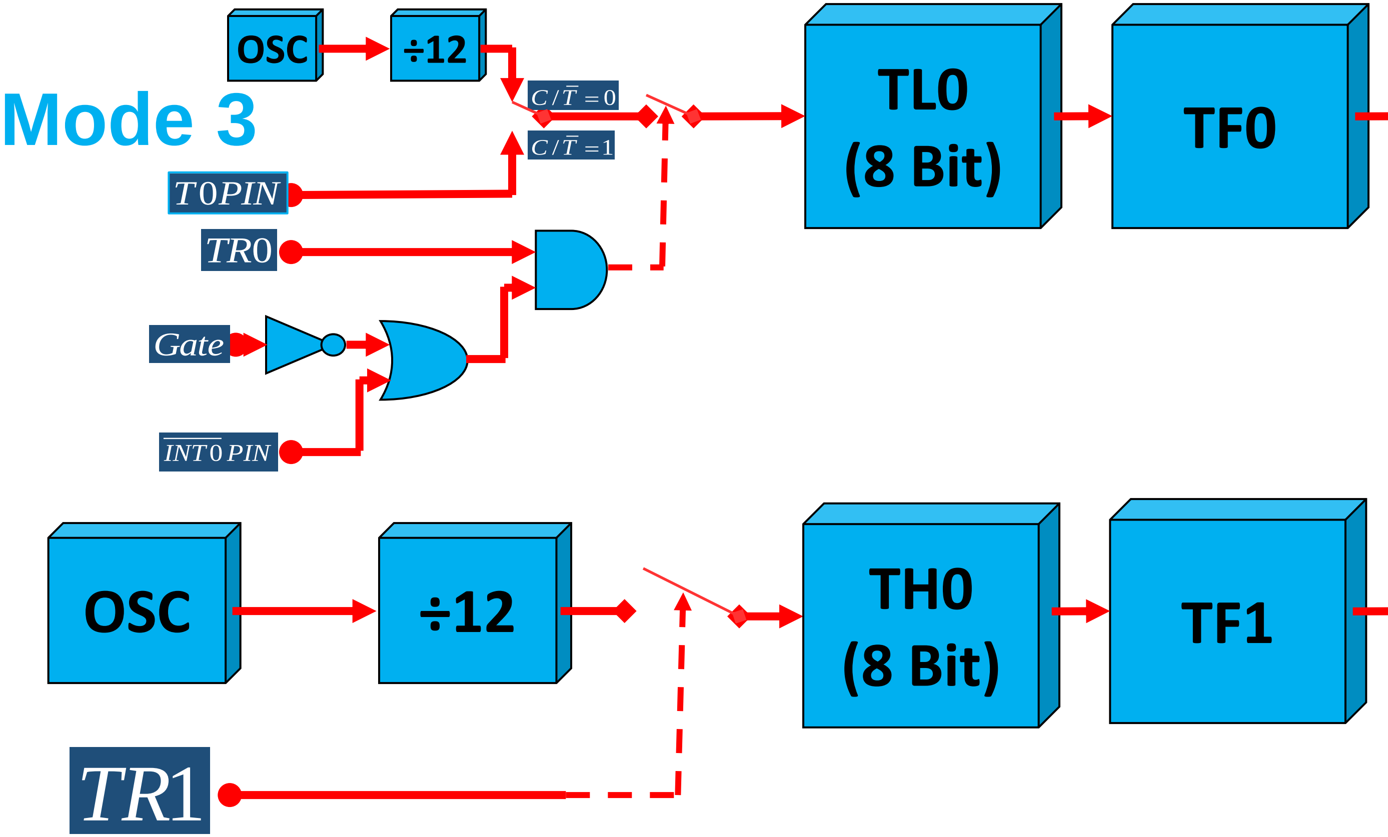


Step 5: Overflow and Interrupt Handling

- After 0.0002 seconds (0.2 ms), TL0 overflows from FFH to 00H, setting TF0.
- TL0 is automatically reloaded with 9CH (from TH0).
- TF0 triggers an interrupt every 0.2 ms (if enabled).

8051 Timer/Counter

Timer 0 – Mode 3



M1	M0	Mode
1	1	3

- Mode 3 is different for Timer0 and Timer1. When the Timer0 is working in mode 3, the TL0 will be used as an 8-bit timer/counter. It will be controlled by the standard Timer0 control bits, T0 and INT0 inputs

8051 Timer/Counter

Timer 0 – Mode 3

- When the TL0 overflows from FFH to 00H, then the TF0 of TCON register will be high (set)
- The TH0 is used as an 8-bit timer but not the counter.
- This is controlled by Timer1 Control bit TR1.
- When the TH0 overflows from FFH to 00H, then TF1 is set to 1.
- When the Timer1 is working in Mode 3, it simply holds the count but does not run.
- When Timer0 is in mode 3, the Timer1 is configured in one of the mode 0, 1 and 2.

8051 Timer/Counter

Timer 0 – Mode 3

- In this case, the Timer1 cannot interrupt the microcontroller. When the TF1 is used by TH0 timer, the Timer1 is used as Baud Rate Generator.
- If the Timer0 is in mode3, and Timer1 is working on either 0, 1 or 2, then the run control (TR1) of the Timer1 is activated when the gate bit is low or INT1 is high.
- The run control is deactivated when the gate is high and INT1 is low.

8051 Timer/Counter

Timer 1

TMOD

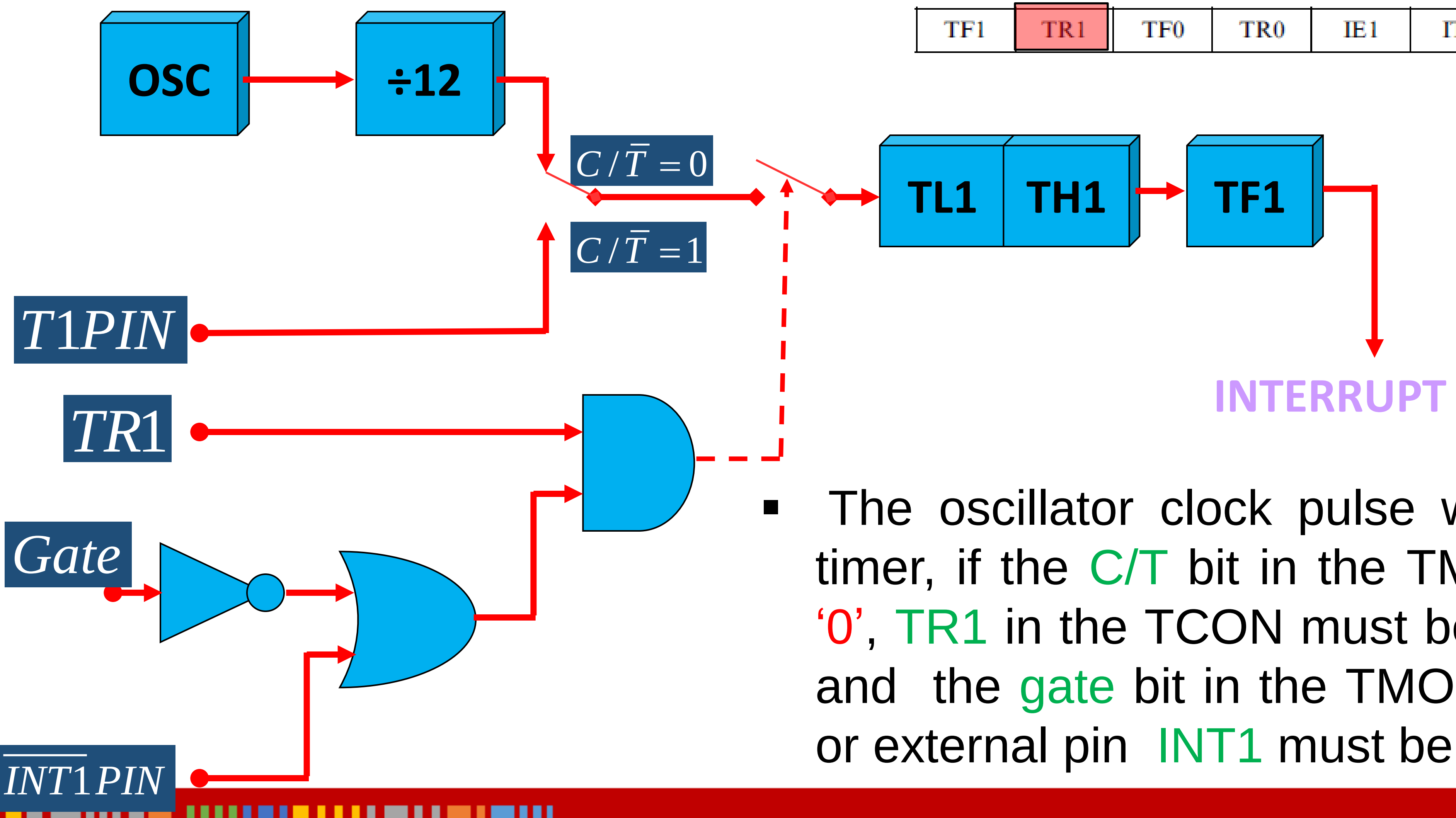
MSB … LSB

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

TCON

MSB … LSB

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

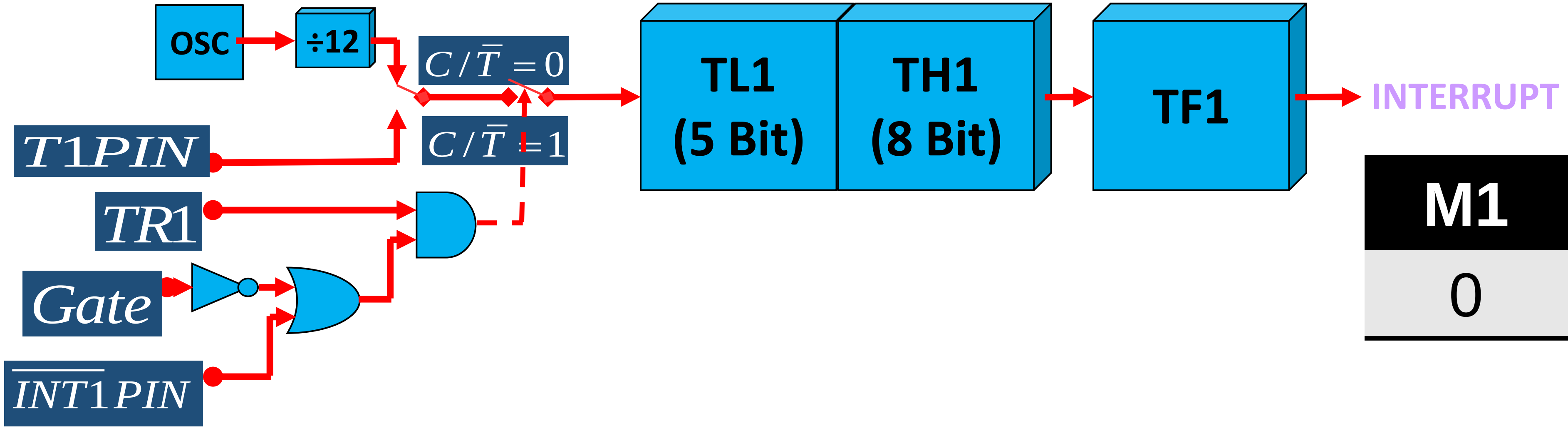


- The oscillator clock pulse will reach the timer, if the C/T bit in the TMOD must be ‘0’, TR1 in the TCON must be 1(timer run) and the gate bit in the TMOD must be ‘0’ or external pin INT1 must be ‘1

8051 Timer/Counter

Timer 1 – Mode 0

13 Bit Timer / Counter



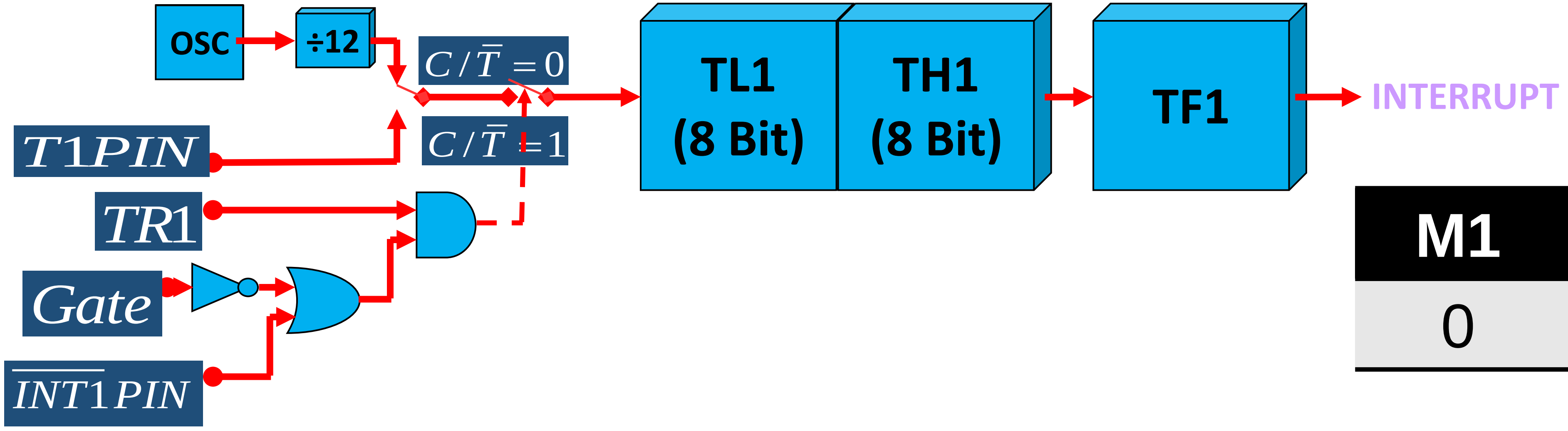
M1	M0	Mode
0	0	0

Maximum Count = 1FFFh (0001111111111111)

8051 Timer/Counter

Timer 1 – Mode 1

16 Bit Timer / Counter



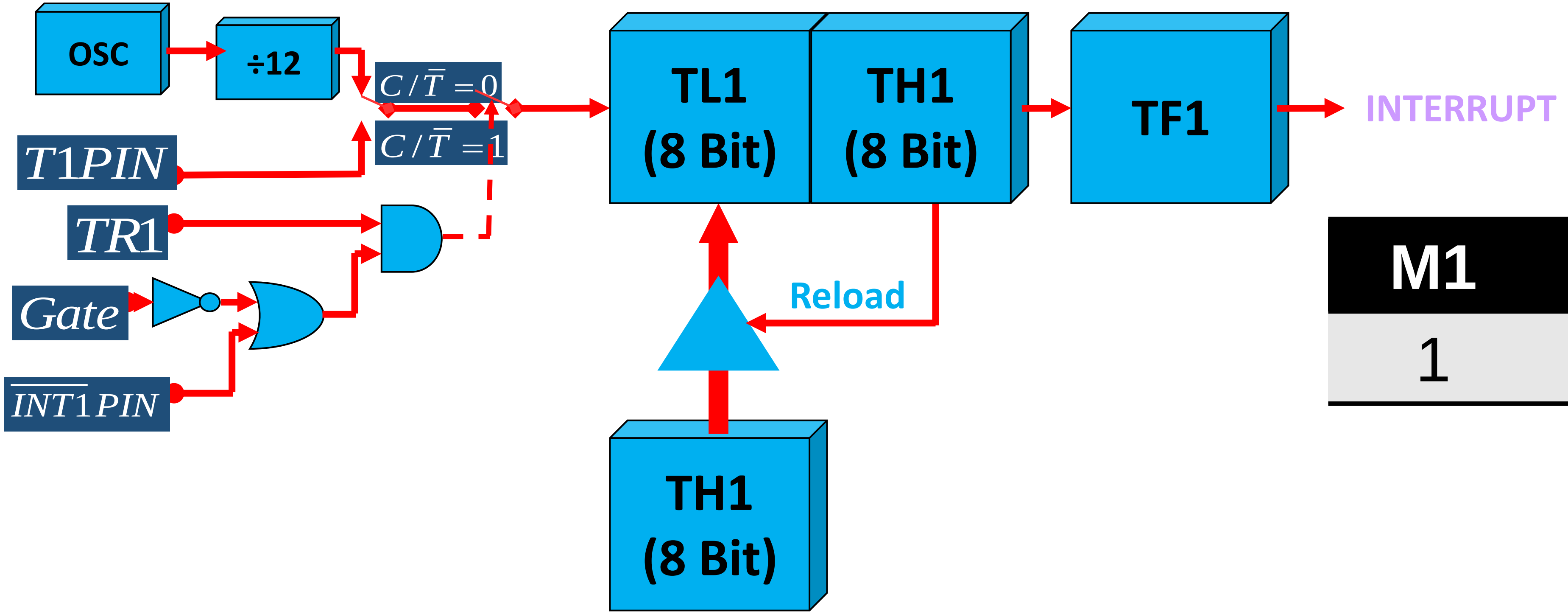
M1	M0	Mode
0	1	1

Maximum Count = FFFFh (1111111111111111)

8051 Timer/Counter

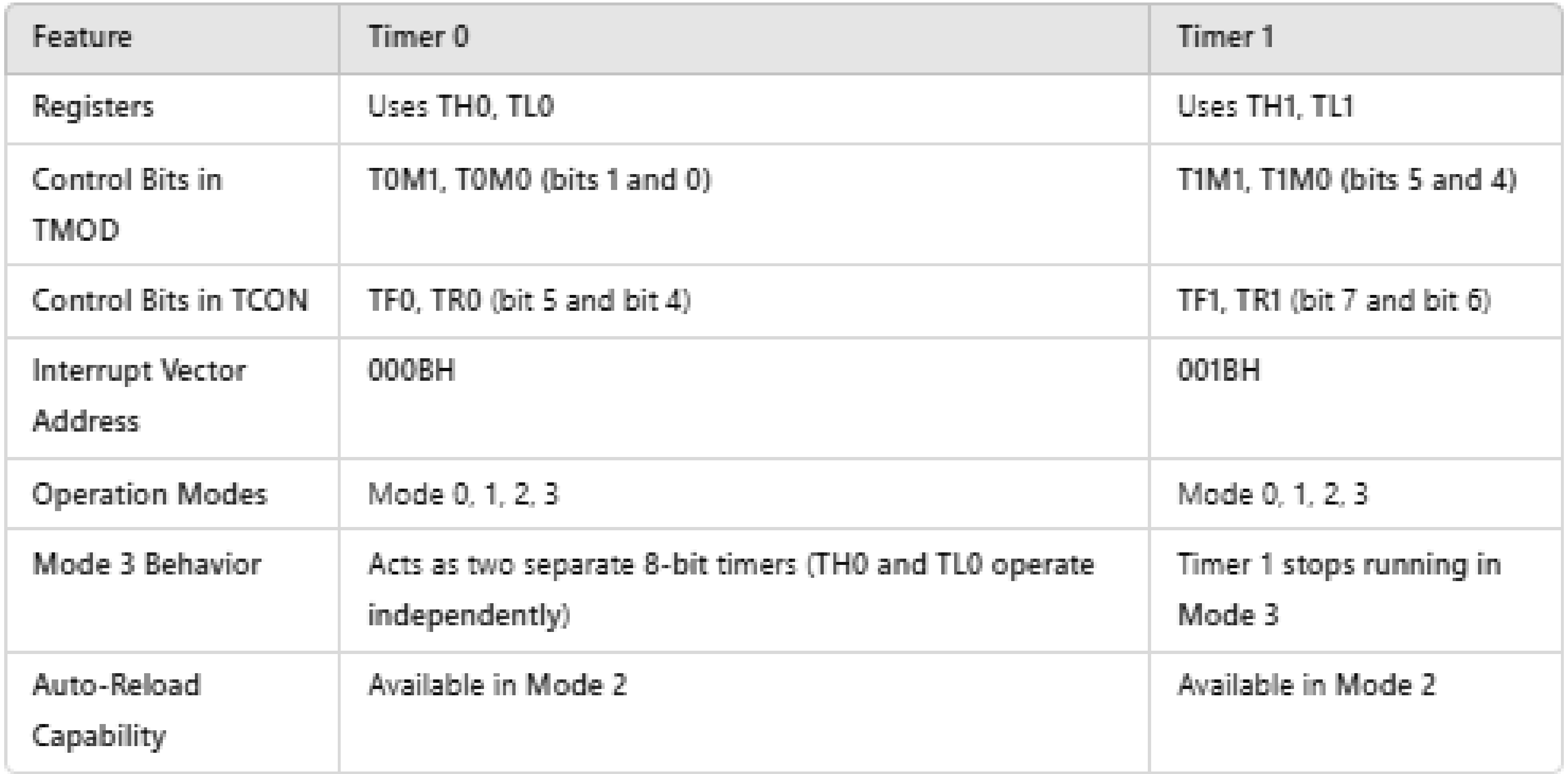
Timer 1 – Mode 2

8 Bit Timer / Counter with AUTORELOAD



M1	M0	Mode
1	0	2

Maximum Count = FFh (11111111)



Feature	Timer 0	Timer 1
Registers	Uses TH0, TL0	Uses TH1, TL1
Control Bits in TMOD	T0M1, T0M0 (bits 1 and 0)	T1M1, T1M0 (bits 5 and 4)
Control Bits in TCON	TF0, TR0 (bit 5 and bit 4)	TF1, TR1 (bit 7 and bit 6)
Interrupt Vector Address	000BH	001BH
Operation Modes	Mode 0, 1, 2, 3	Mode 0, 1, 2, 3
Mode 3 Behavior	Acts as two separate 8-bit timers (TH0 and TL0 operate independently)	Timer 1 stops running in Mode 3
Auto-Reload Capability	Available in Mode 2	Available in Mode 2

8051 Timer/Counter

Programming Timers

- **Example:** Indicate which mode and which timer are selected for each of the following.
  - **(a) MOV TMOD, #01H (b) MOV TMOD, #20H (c) MOV TMOD, #12H**

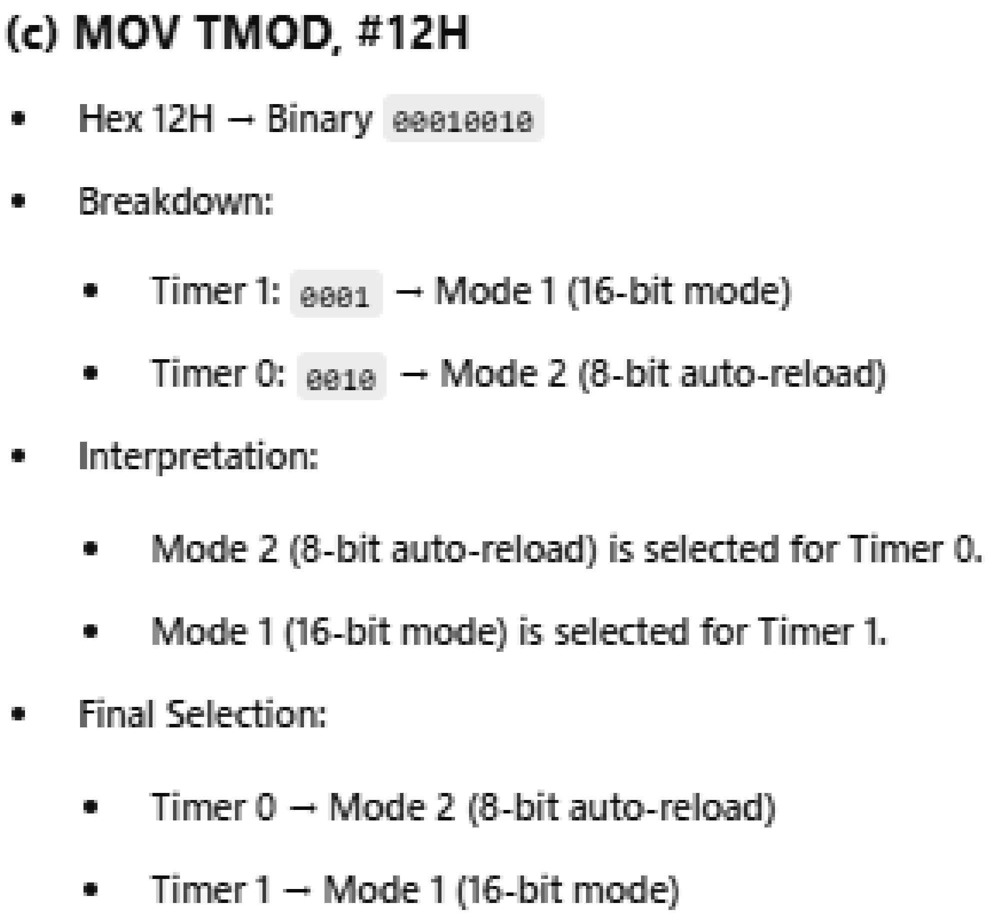
- **Solution:** We convert the value from hex to binary.
  - **(a) TMOD = 00000001, mode 1 of timer 0 is selected.**
  - **(b) TMOD = 00100000, mode 2 of timer 1 is selected.**
  - **(c) TMOD = 00010010, mode 2 of timer 0, and mode 1 of timer 1 are selected.**

(a) MOV TMOD, #01H

- Hex 01H → Binary `00000001`
- Breakdown:
  - Timer 1: `0000` → `Mode 0`
  - Timer 0: `0001` → `Mode 1`
- Interpretation:
  - Mode 1 (16-bit mode) is selected for Timer 0.
  - Timer 1 remains in Mode 0 (13-bit mode, default).
- Final Selection:
  - Timer 0 → Mode 1 (16-bit mode)
  - Timer 1 → Mode 0 (13-bit mode, default)

(b) MOV TMOD, #20H

- Hex 20H → Binary `00100000`
- Breakdown:
  - Timer 1: `0010` → Mode 2 (8-bit auto-reload)
  - Timer 0: `0000` → Mode 0 (13-bit mode, default)
- Interpretation:
  - Mode 2 (8-bit auto-reload) is selected for Timer 1.
  - Timer 0 remains in Mode 0 (default).
- Final Selection:
  - Timer 0 → Mode 0 (13-bit mode, default)
  - Timer 1 → Mode 2 (8-bit auto-reload)

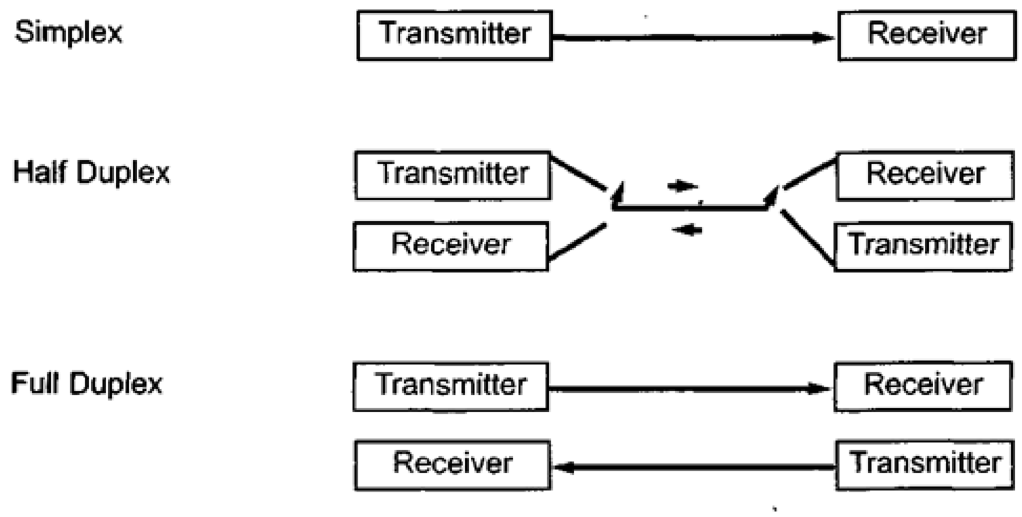


(c) MOV TMOD, #12H

- Hex 12H → Binary `00010010`
- Breakdown:
  - Timer 1: `0001` → Mode 1 (16-bit mode)
  - Timer 0: `0010` → Mode 2 (8-bit auto-reload)
- Interpretation:
  - Mode 2 (8-bit auto-reload) is selected for Timer 0.
  - Mode 1 (16-bit mode) is selected for Timer 1.
- Final Selection:
  - Timer 0 → Mode 2 (8-bit auto-reload)
  - Timer 1 → Mode 1 (16-bit mode)

8051 Serial Communication

Basics of Serial Communication

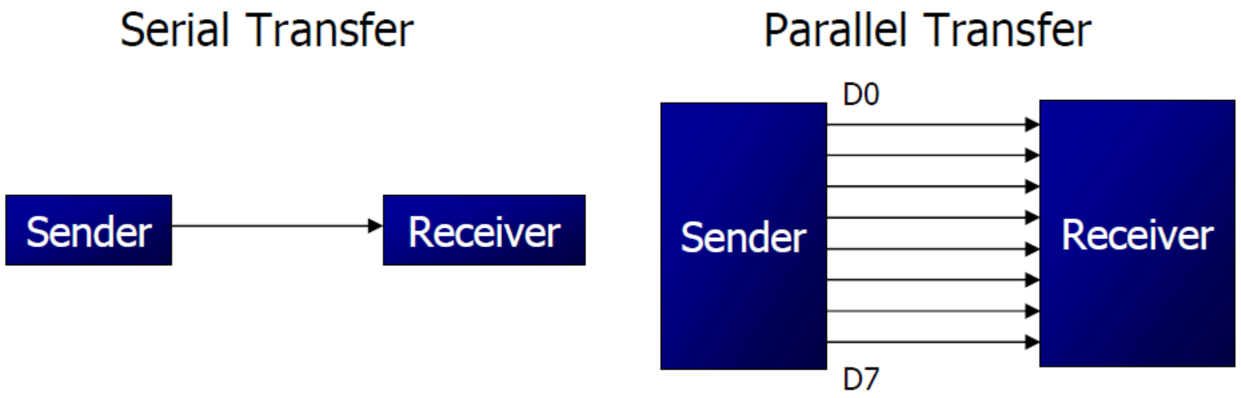


- In data transmission if the data can be transmitted and received, it is a *duplex* transmission.
- *simplex* transmissions such as with printers, in which the computer only sends data.
- Duplex transmissions can be half or full duplex, depending on whether or not the data transfer can be simultaneous.
- If data is transmitted one way at a time, it is referred to as *half duplex.*
- If the data can go both ways at the same time, it *is full duplex.*

8051 Serial Communication

Basics of Serial Communication

- Computers transfer data in **two** ways:
  - **Parallel:** Often 8 or more lines (wire conductors) are used to transfer data to a device that is only a few feet away.
  - **Serial:** To transfer to a device located many meters away, the serial method is used. The data is sent one bit at a time.



8051 Serial Communication

Basics of Serial Communication

- Serial data communication uses **two** methods
  - **Synchronous** method transfers a block of data at a time
  - **Asynchronous** method transfers a single byte at a time
- There are **special IC's** made by many manufacturers for serial communications.
  - **UART** (universal asynchronous Receiver transmitter)
  - **USART** (universal synchronous-asynchronous Receiver-transmitter)

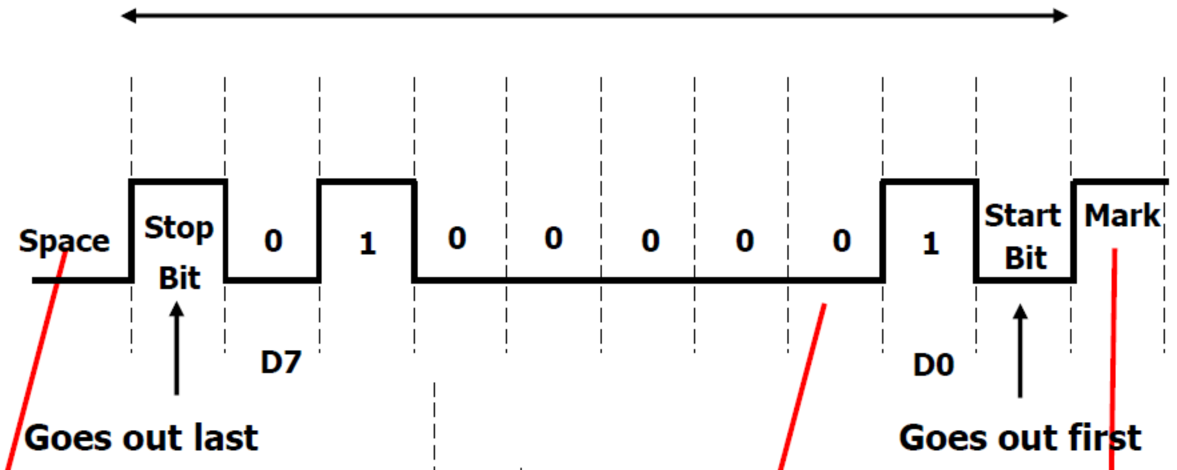
8051 Serial Communication

Basics of Serial Communication - Asynchronous

- Asynchronous serial data communication is widely used for character-oriented transmissions
  - Each character is placed in between start and stop bits, this is called framing.
  - Block-oriented data transfers use the synchronous method.
- The start bit is always one bit, but the stop bit can be one or two bits
- The start bit is always a 0 (low) and the stop bit(s) is 1 (high)

8051 Serial Communication

ASCII character “A” (8-bit binary 0100 0001)



The 0 (low) is referred to as *space*

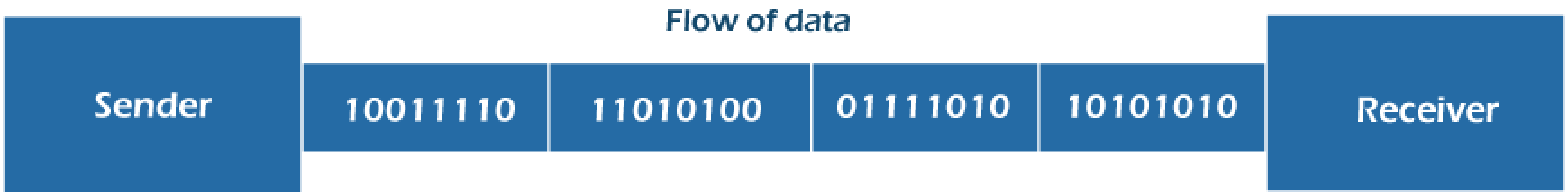
The transmission begins with a start bit followed by D0, the LSB, then the rest of the bits until MSB (D7), and finally, the one stop bit indicating the end of the character

When there is no transfer, the signal is 1 (high), which is referred to as *mark*

8051 Serial Communication

Synchronous Transmission:

- In Synchronous Transmission, data is sent in form of blocks or frames.
- This transmission is the full-duplex type. Between sender and receiver, synchronization is compulsory.
- In Synchronous transmission, There is no gap present between data.
- It is more efficient and more reliable than asynchronous transmission to transfer a large amount of data.



Synchronous Transmission

8051 Serial Communication

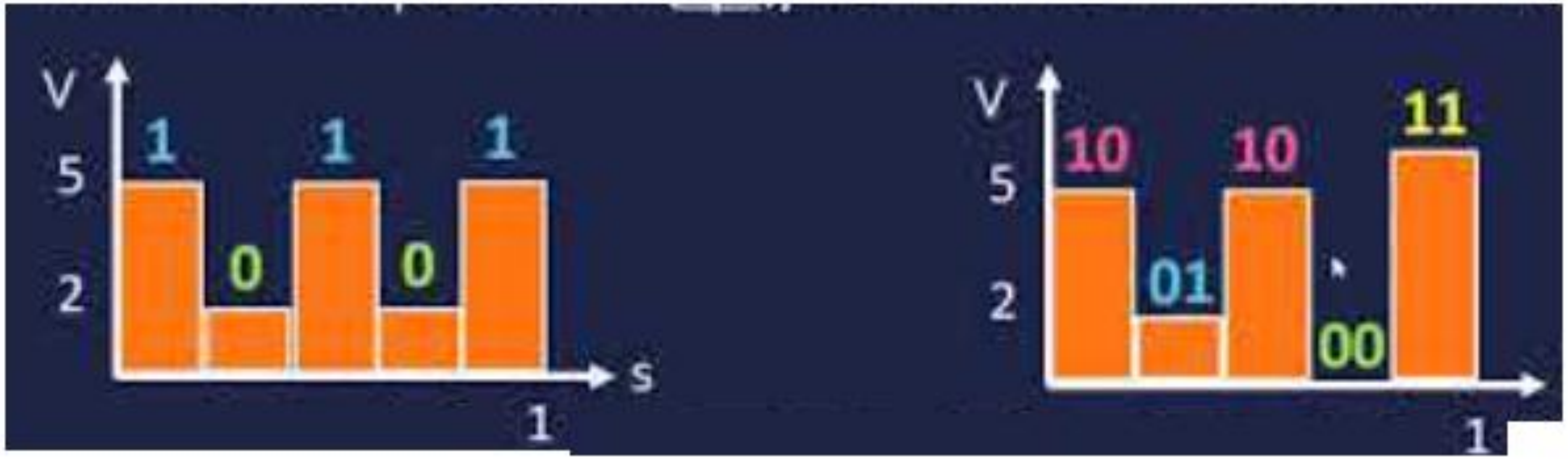
Data Transfer Rate

- The rate of data transfer in serial data communication is stated in **bps** (**bits per second**).
- Another widely used terminology for bps is **baud rate.**
  - It is modem terminology and is defined as **the number of signal changes per second**
  - Baud rate is the rate at which the number of signal elements or changes to the signal occurs per second when it passes through a transmission medium.

8051 Serial Communication

Data Transfer Rate

- The higher a baud rate is the faster the data is sent/received.
- In modems, there are occasions when a single change of signal transfers several bits of data



8051 Serial Communication

8051 Serial Port

- Synchronous and Asynchronous
- SCON Register is used to Control
- Data Transfer through TXd & RXd pins
- Some time - Clock through TXd Pin
- Four Modes of Operation:

Mode 0	**:Synchronous Serial Communication**
Mode 1	**:8-Bit UART with Timer Data Rate**
Mode 2	**:9-Bit UART with Set Data Rate**
Mode 3	**:9-Bit UART with Timer Data Rate**

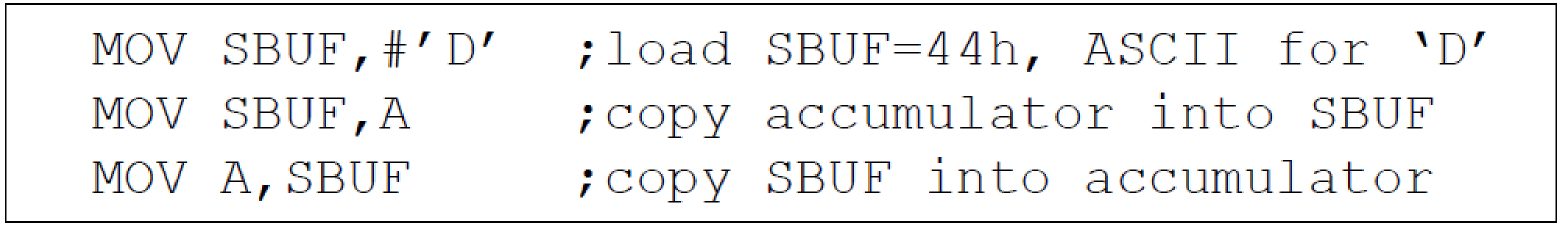
8051 Serial Communication

Registers related to Serial Communication

- SBUF Register
- SCON Register
- PCON Register

8051 Serial Communication

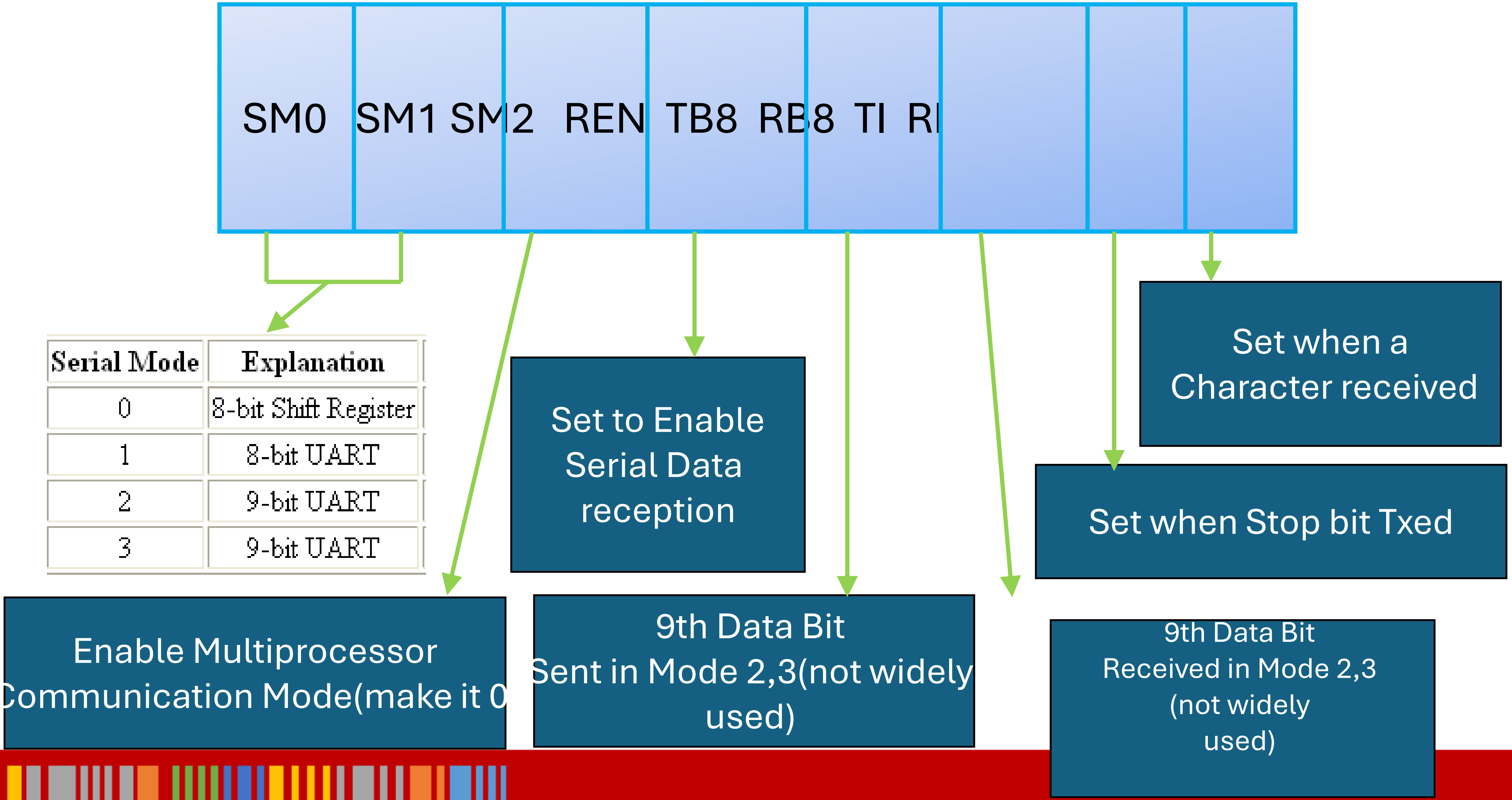
- SBUF is an 8-bit register used for serial Commn.
- For a byte data to be transferred via the TxD line, it must be placed in the SBUF register.
- The moment a byte is written into SBUF, it is framed with the start and stop bits and transferred serially via the TxD line.
- SBUF holds the byte of data when it is received by 8051 RxD line.
- When the bits are received serially via RxD, the **8051 de-frames** it by eliminating the stop and start bits, making a byte out of the data received, and then placing it in SBUF.



```
MOV  SBUF,#’D’   ;load SBUF=44h, ASCII for ‘D’
MOV  SBUF,A      ;copy accumulator into SBUF
MOV  A,SBUF      ;copy SBUF into accumulator
```

8051 Serial Communication

SCON Register



Serial Mode	Explanation
0	8-bit Shift Register
1	8-bit UART
2	9-bit UART
3	9-bit UART

Enable Multiprocessor Communication Mode(make it 0

Set to Enable Serial Data reception

9th Data Bit Sent in Mode 2,3(not widely used)

9th Data Bit Received in Mode 2,3 (not widely used)

Set when Stop bit Txed

Set when a Character received

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial port mode specifier
SM1	SCON.6	Serial port mode specifier
SM2	SCON.5	Used for multiprocessor communication. (Make it 0.)
REN	SCON.4	Set/cleared by software to enable/disable reception.
TB8	SCON.3	Not widely used.
RB8	SCON.2	Not widely used.
TI	SCON.1	Transmit interrupt flag. Set by hardware at the beginning of the stop bit in mode 1. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware halfway through the stop bit time in mode 1. Must be cleared by software.

Note: Make SM2, TB8, and RB8 = 0.

8051 Serial Communication

The Serial Port in Mode-0 has the following features:

- Serial data **enters and exits through RXD**
- **TXD** outputs the **clock**
- 8 bits are transmitted / received
- The baud rate is fixed at (1/12) of the oscillator frequency

8051 Serial Communication

The Serial Port in Mode-1 has the following features:

- Serial data enters through RXD
- Serial data exits through TXD
- On receive, the stop bit goes into RB8 in SCON
- 10 bits are transmitted / received
  - Start bit (0)
  - Data bits (8)
  - Stop Bit (1)
- Baud rate is determined by the Timer 1 overflow rate.

8051 Serial Communication

- Baud rate is determined by the Timer 1 overflow rate.
- Timer 1 is used in timer mode2 (8-bit auto reload) to generate the baud frequency

$$f_{baud} = \frac{2^{SMOD}}{32} \times \frac{Oscillator\ Frequency}{12(256 - TH1)}$$

- SMOD is the control bit in PCON and can be 0 or 1
- If Timer 1 not in mode2, then the baud rate is

$$f_{baud} = \frac{2^{SMOD}}{32} \times \text{Timer 1 overflow frequency}$$

8051 Serial Communication

- If standard baud rates are desired, then 11.0592 MHz crystal could be selected.
- To get a standard rate of 9600 hertz , the setting of TH1 can be found as (if SMOD =0)

$$TH1 \; = 256 - \frac{2^{0}}{32} \times \frac{11.0592 \times 10^{6}}{12 \times 9600} = 253 = \mathrm{0FDh}$$

8051 Serial Communication

The Serial Port in Mode-2 has the following features:

- Serial data enters through RXD
- Serial data exits through TXD
- 9th data bit (TB8) can be assign value 0 or 1
- On receive, the 9th data bit goes into RB8 in SCON
- 11 bits are transmitted / received
  - Start bit (0)
  - Data bits (9)
  - Stop Bit (1)
- Baud rate is programmable $f_{baud} = \frac{2^{SMOD}}{64} \times \text{Oscillator Frequency}$

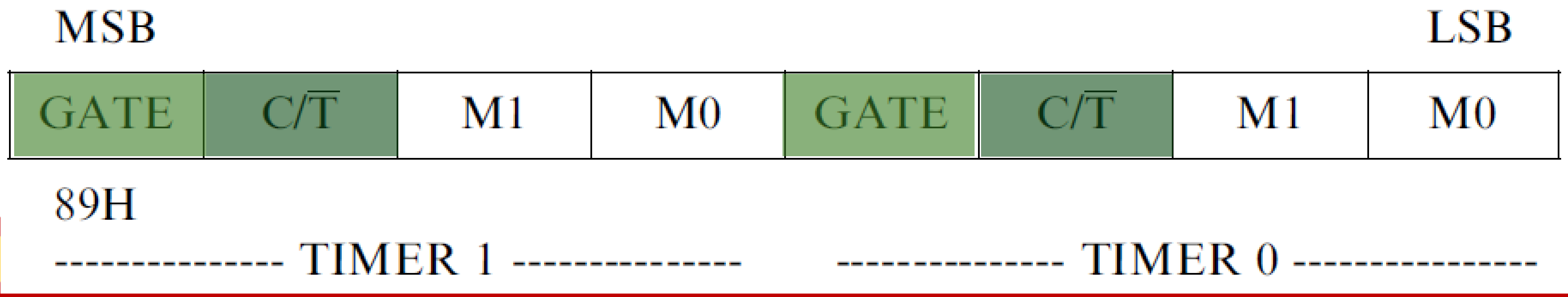
8051 Serial Communication

The Serial Port in Mode-3 has the following features:

- Serial data enters through RXD
- Serial data exits through TXD
- 9th data bit (TB8) can be assign value 0 or 1
- On receive, the 9th data bit goes into RB8 in SCON
- 11 bits are transmitted / received
  - Start bit (0)
  - Data bits (9)
  - Stop Bit (1)
- This mode is completely similar to mode 2, in mode 2 for triggering timer is used Whereas in this mode internal clock is used for triggering. It has a fixed baud rate.

Programming Serial Data Transmission

1. TMOD register is loaded with the value 20H, indicating the use of timer 1 in mode 2 (8-bit auto-reload) to set baud rate.
2. The TH1 is loaded with one of the values to set baud rate for serial data transfer.
3. The SCON register is loaded with the value 50H, indicating serial mode 1, where an 8- bit data is framed with start and stop bits.
4. TR1 is set to 1 to start timer 1
5. TI is cleared by CLR TI instruction
6. The character byte to be transferred serially is written into SBUF register.
7. The TI flag bit is monitored with the use of instruction JNB TI, xx to see if the character has been transferred completely.
8. To transfer the next byte, go to step 5



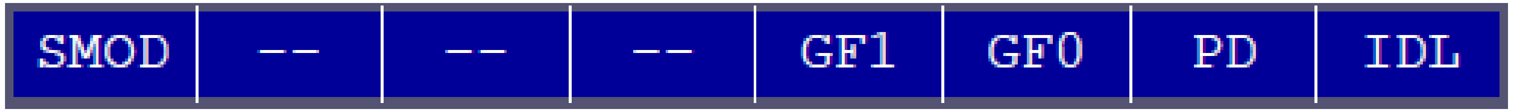
MSB							LSB
GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
89H							
-------------- TIMER 1 --------------				-------------- TIMER 0 --------------			

Programming Serial Data Reception

1. TMOD register is loaded with the value 20H, indicating the use of timer 1 in mode 2 (8-bit auto-reload) **to set baud rate.**
2. TH1 is loaded to set baud rate
3. The SCON register is loaded with the value 50H, indicating serial mode 1, where an 8-bit data is framed with start and stop bits.
4. TR1 is set to 1 to start timer 1
5. RI is cleared by CLR RI instruction
6. The RI flag bit is monitored with the use of instruction JNB RI, xx to see if an entire character has been received yet
7. **When RI is raised**, SBUF has the byte, its contents are moved into a safe place.
8. To receive the next character, **go to step 5.**

Doubling Baud Rate

- There are two ways to increase the baud rate of data transfer
  1. By using a higher frequency crystal
  2. By changing a bit in the PCON register
- PCON register is an 8-bit register.



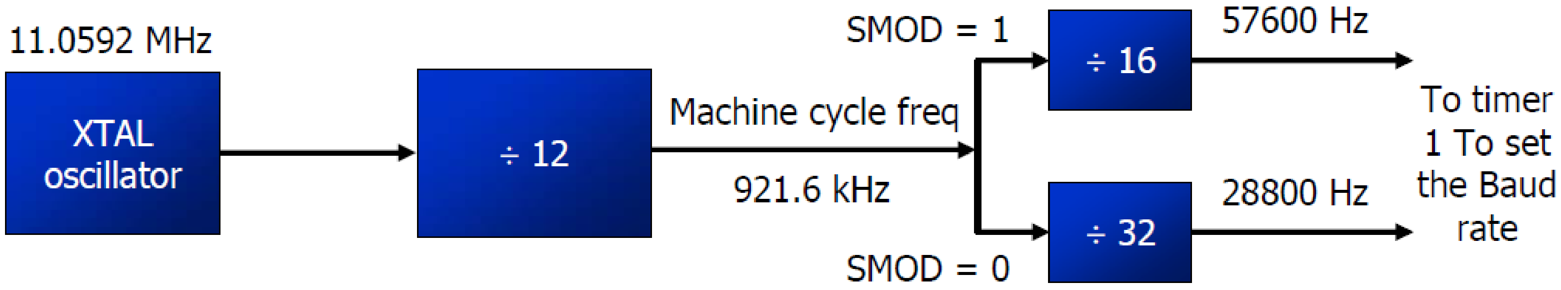
SMOD	--	--	--	GF1	GF0	PD	IDL

•When 8051 is powered up, SMOD is zero

•We can set it to high by software and thereby double the baud rate.

Doubling Baud Rate

```
MOV   A,PCON     ;place a copy of PCON in ACC
SETB  ACC.7      ;make D7=1
MOV   PCON,A     ;changing any other bits
```



Baud Rate comparison for SMOD=0 and SMOD=1

TH1	(Decimal)	(Hex)	SMOD=0	SMOD=1
	-3	FD	9600	19200
	-6	FA	4800	9600
	-12	F4	2400	4800
	-24	E8	1200	2400

8051 Connection to RS232

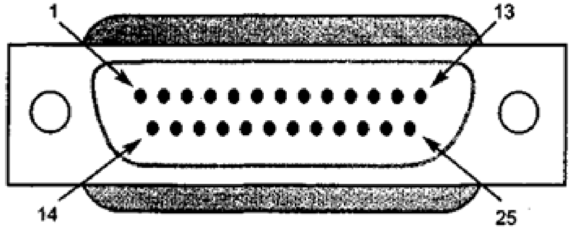
RS232 standards

- To allow compatibility among data communication equipment made by various manufacturers, an interfacing standard called RS232 was set by the Electronics Industries Association (EIA) in 1960.
- In 1963 it was modified and called RS232A. RS232B and RS232C were issued in 1965 and 1969, respectively.
- RS232 is the most widely used serial I/O interfacing standard.
- In RS232, a '1' is represented by -3 to -25 V, while a '0' bit is +3 to +25 V, making -3 to +3 undefined.
- For this reason, to connect any RS232 to a microcontroller system we must use voltage converters (MAX232) to convert the TTL logic levels to the RS232 voltage levels, and vice versa.

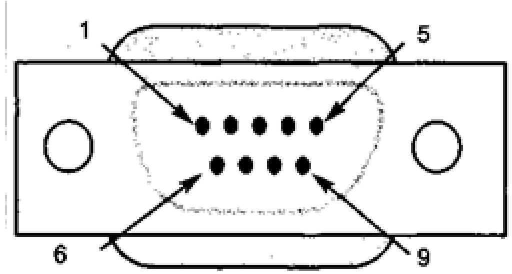
8051 Connection to RS232

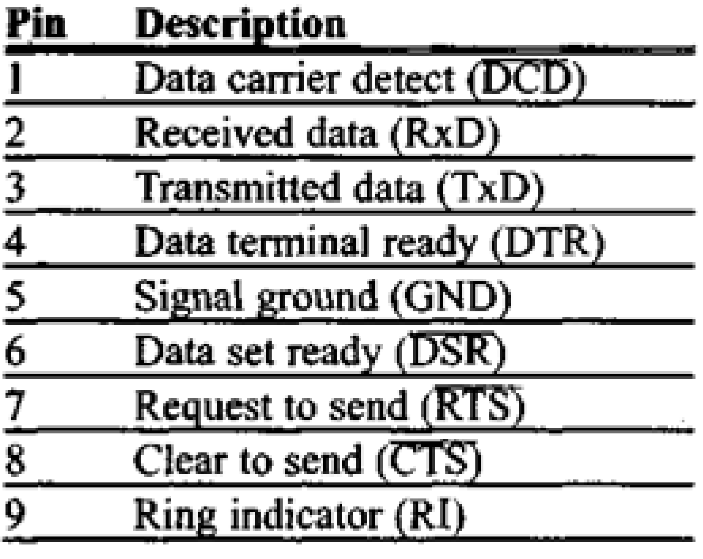
RS232 pins

- RS232 cable, commonly referred to as the DB-25 connector

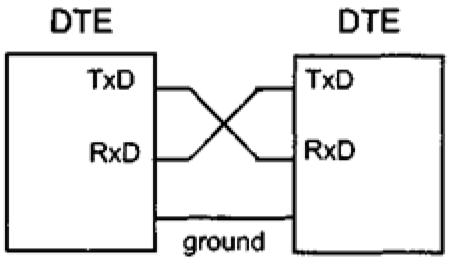


- Since not all the pins are used in PC cables, IBM introduced the DB-9 version of the serial I/O standard, which uses 9 pins only





Pin	Description
1	Data carrier detect (DCD)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready (DSR)
7	Request to send (RTS)
8	Clear to send (CTS)
9	Ring indicator (RI)



8051 connection to RS232

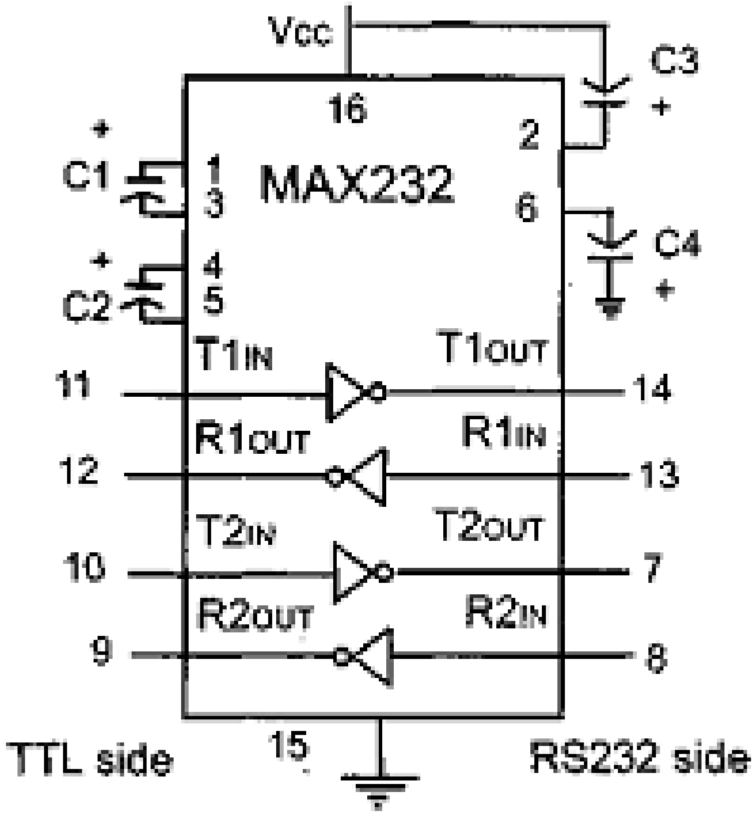
- The RS232 standard is not TTL compatible; therefore, it requires a line driver such as the MAX232 chip to convert RS232 voltage levels to TTL levels, and vice versa.

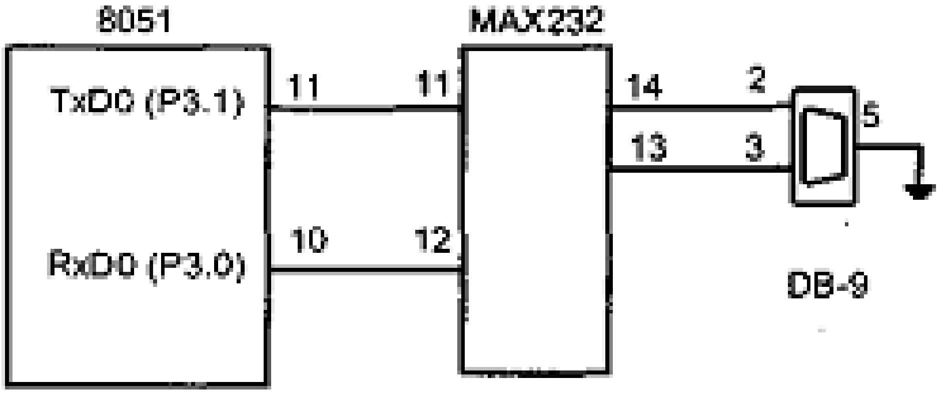
RxD and TxD pins in the 8051

- Pin 11 of the 8051 (P3.1) is assigned to TxD and pin 10 (P3.0) is designated as RxD (P3.0 and P3.1).
- These pins are TTL compatible; therefore, they require a line driver (MAX232 chip) to make them RS232 compatible.

8051 Connection to RS232

8051 connection to RS232


Vcc
C3
+
16
2
+
C1
1
3
MAX232
6
C4
+
+
C2
4
5
T1IN
T1OUT
11
14
R1OUT
R1IN
12
13
T2IN
T2OUT
10
7
R2OUT
R2IN
9
8
TTL side
15
RS232 side


8051
MAX232
TxD0 (P3.1)
11
11
14
2
5
13
3
RxD0 (P3.0)
10
12
DB-9