The 8051 Microcontroller

Prepared By,

R-THANDAIAH PRABU M.E.,

Lecturer - ECE

thandaiah@gmail.com

Disadvantages of microprocessor

- *The overall system cost is high*
- *A large sized PCB is required for assembling all the components*
- *Overall product design requires more time*
- *Physical size of the product is big*
- *A discrete components are used, the system is not reliable*

Advantages of Microcontroller based System

- *As the peripherals are integrated into a single chip, the overall system cost is very less*
- *The product is of small size compared to micro processor based system*
- *The system design now requires very little efforts*
- *As the peripherals are integrated with a microprocessor the system is more reliable*
- *Though microcontroller may have on chip ROM,RAM and I/O ports, addition ROM, RAM I/O ports may be interfaced externally if required*
- *On chip ROM provide a software security*

Three criteria in Choosing a Microcontroller

- ***meeting the computing needs of the task efficiently and cost effectively***
  - ***speed, the amount of ROM and RAM, the number of I/O ports and timers, size, packaging, power consumption***
  - ***easy to upgrade***
  - ***cost per unit***
  - ***Noise of environment***
- ***availability of software development tools***
  - ***assemblers, debuggers, C compilers, emulator, simulator, technical support***
- ***wide availability and reliable sources of the microcontrollers***

Comparison of the 8051 Family Members

- *ROM type*
  - *8031 no ROM*
  - *80xx mask ROM*
  - *87xx EPROM*
  - *89xx Flash EEPROM*
- *89xx*
  - *8951*
  - *8952*
  - *8953*
  - *8955*
  - *898252*
  - *891051*
  - *892051*
- *Example (AT89C51,AT89LV51)*
  - *AT= ATMEL(Manufacture)*
  - *C = CMOS technology*
  - *LV= Low Power(3.0v)*

Comparison some of the 8051 Family Members

	ROM	RAM	Timer
8051	4k	128	2
8031	-	128	2
8751	4k eprom	128	2
8052	8krom	256	3
8032	-	256	3
8752	8k eprom	256	3

8051 Basic Component

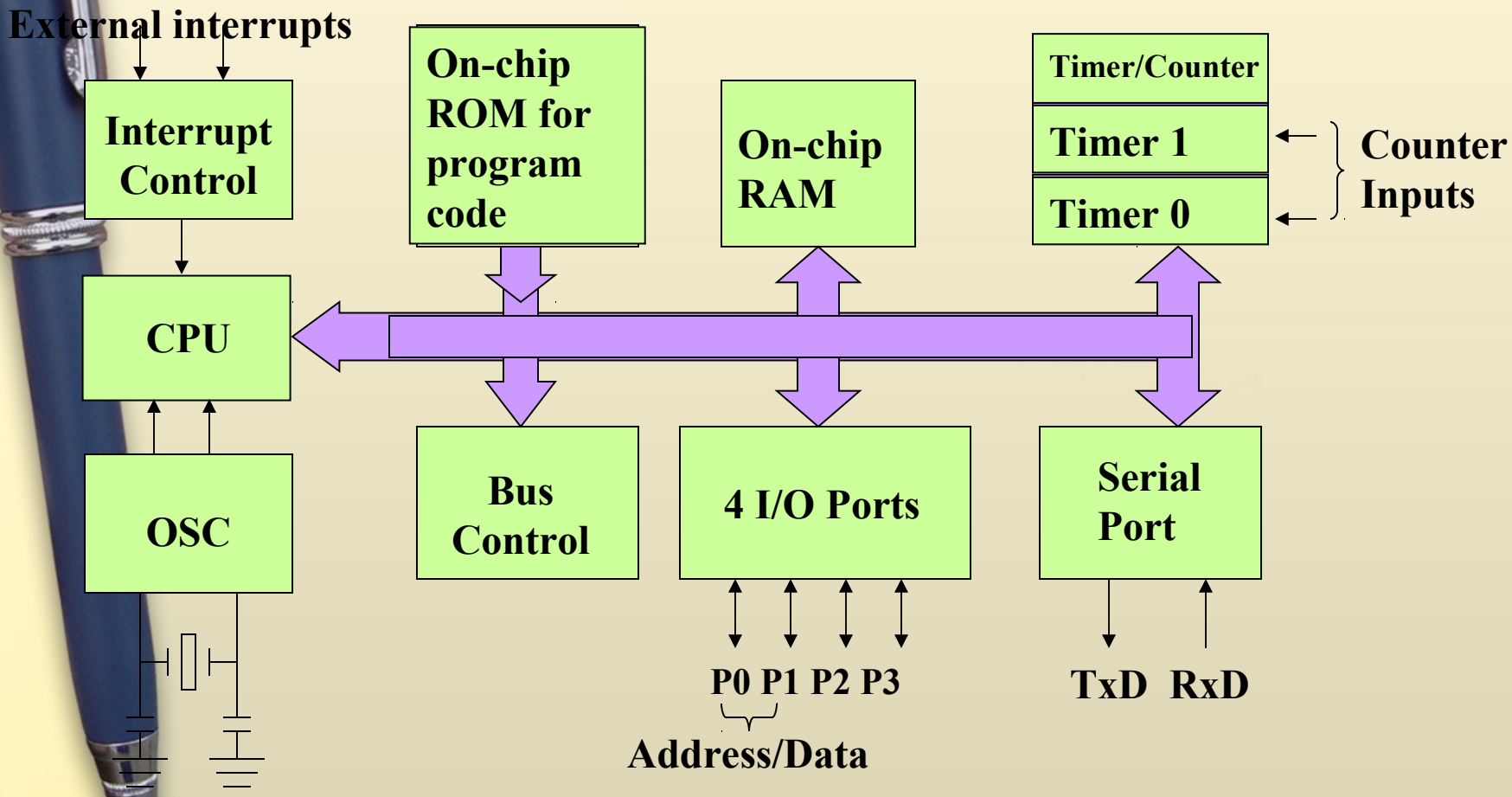
- *4K bytes internal ROM*
- *128 bytes internal RAM*
- *Four 8-bit I/O ports (P0 - P3).*
- *Two 16-bit timers/counters*
- *One serial interface*
- *64k external memory for code*
- *64k external memory for data*
- *210 bit addressable*

Microcontroller

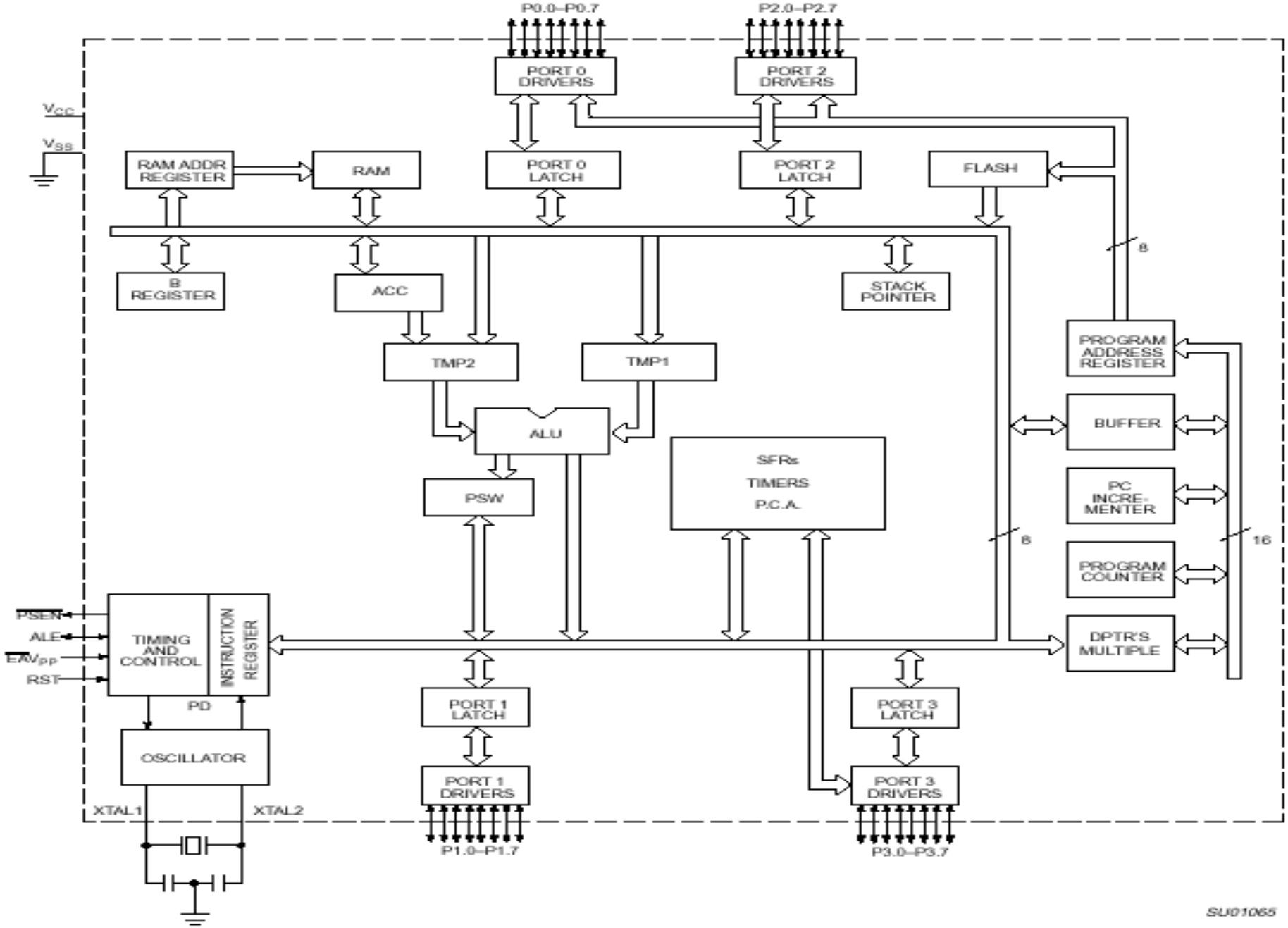
The basic 8051 Core

- 8-bit CPU optimized for control applications
- Capability for single bit Boolean operations.
- Supports up to 64K of program memory.
- Supports up to 64K of data memory.
- 4 K bytes of on-chip program memory.
    - Newer devices provide more.
- 128 or 256 bytes of on-chip data RAM
- Four 8 bit ports.
- Two 16-bit timer/counters
- UART
- Interrupts
- On-chip clock oscillator

Block Diagram



8051 Internal Block Diagram



8051 Schematic Pin out

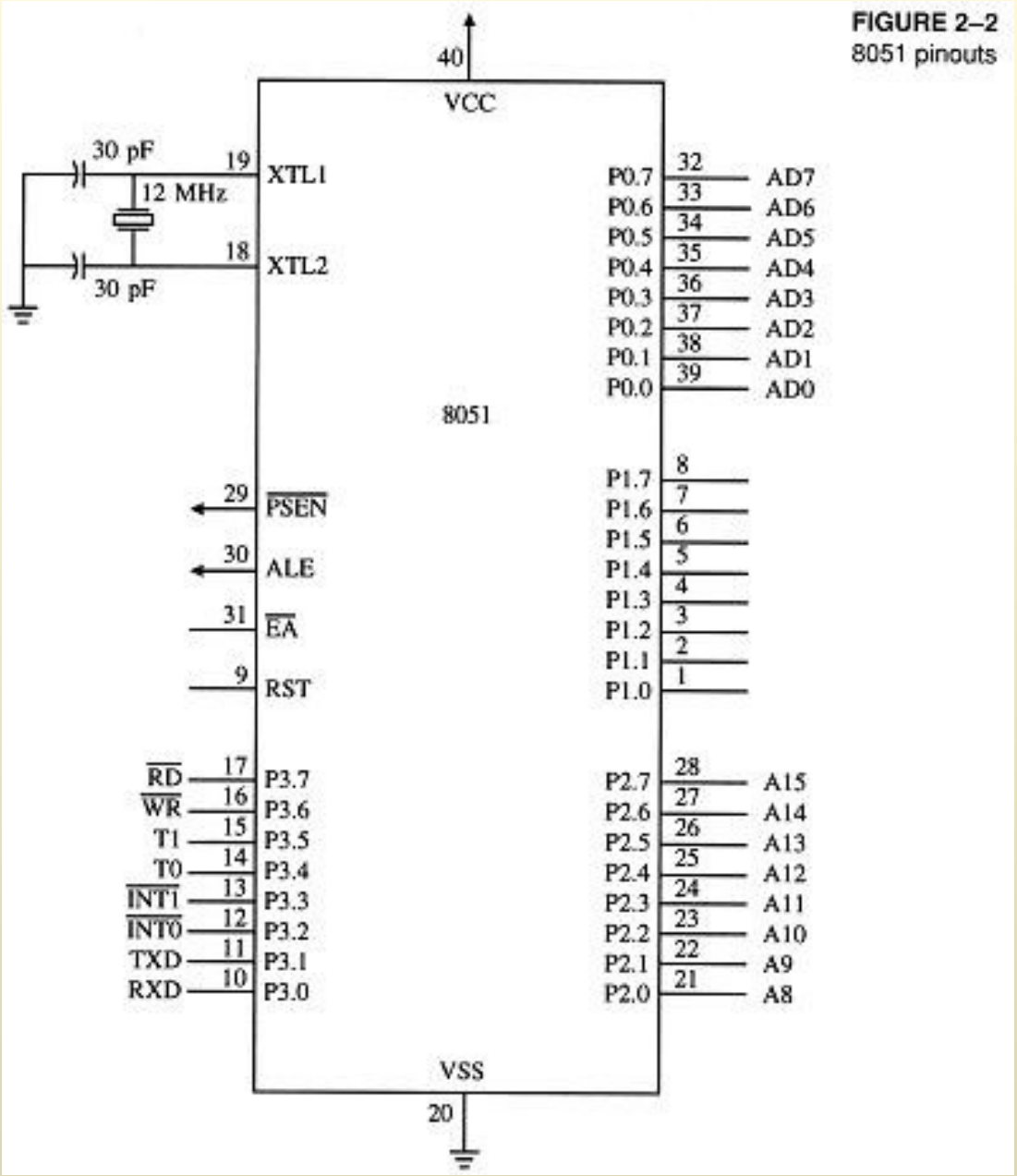
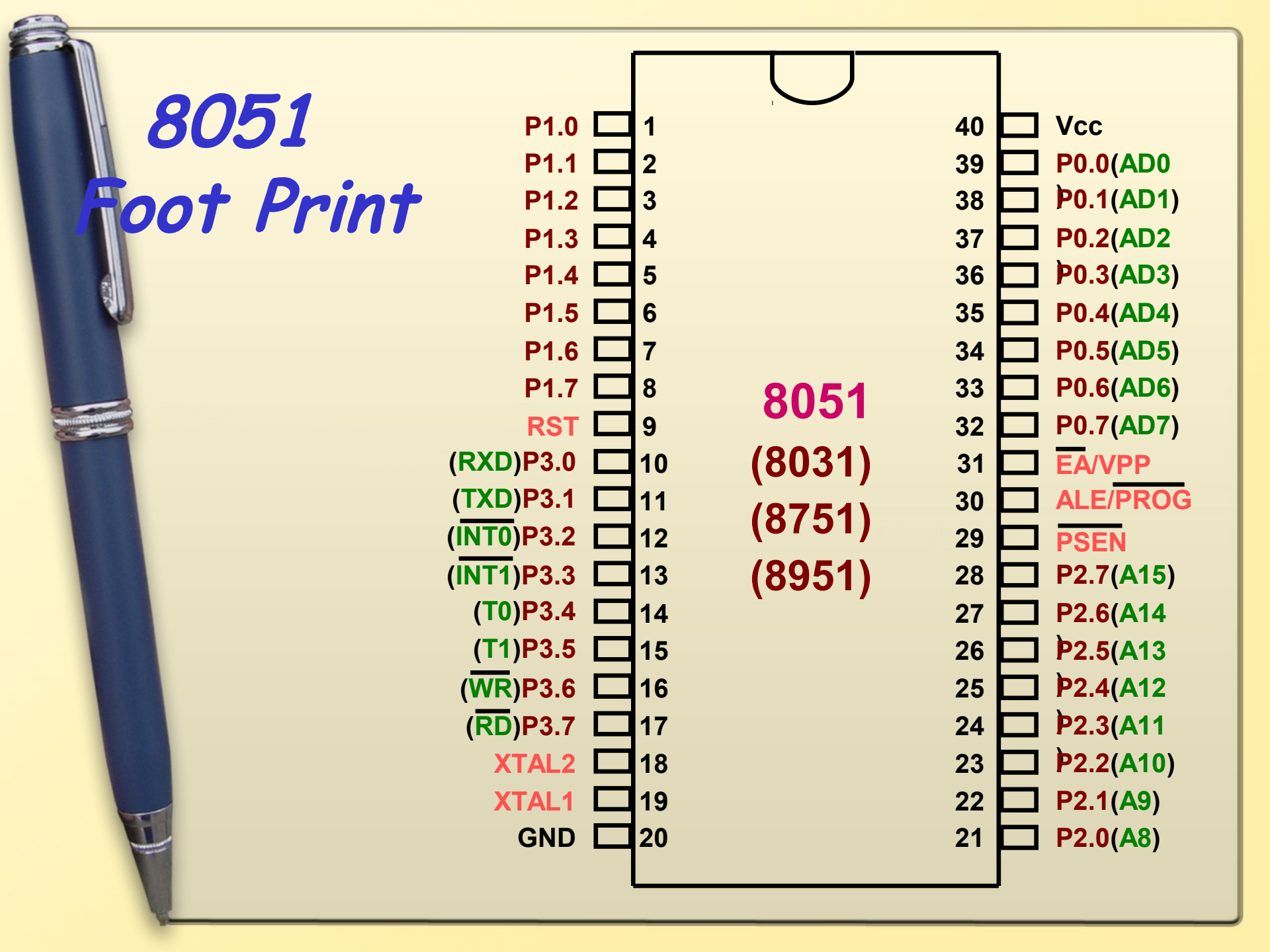
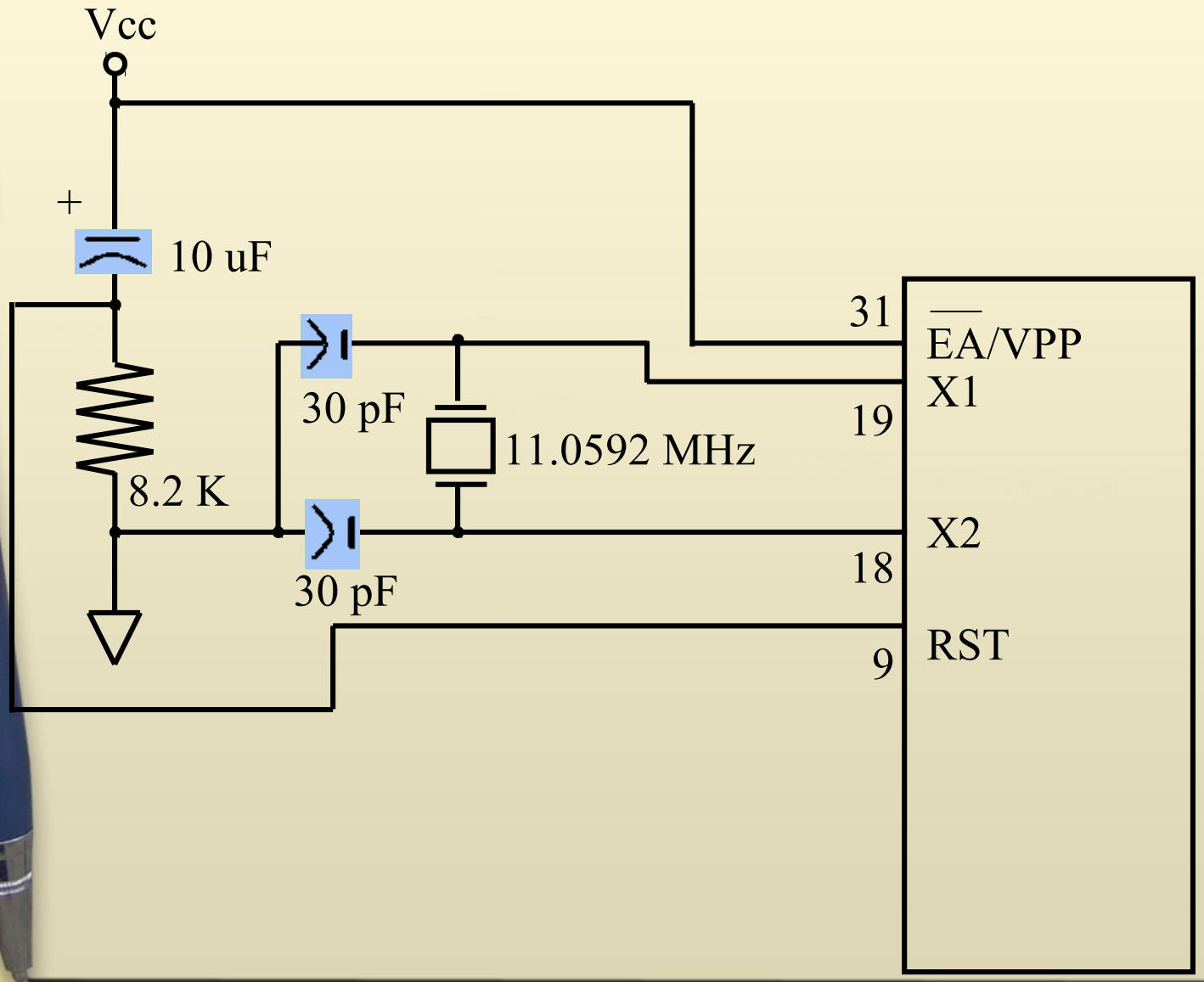


FIGURE 2–2
8051 pinouts

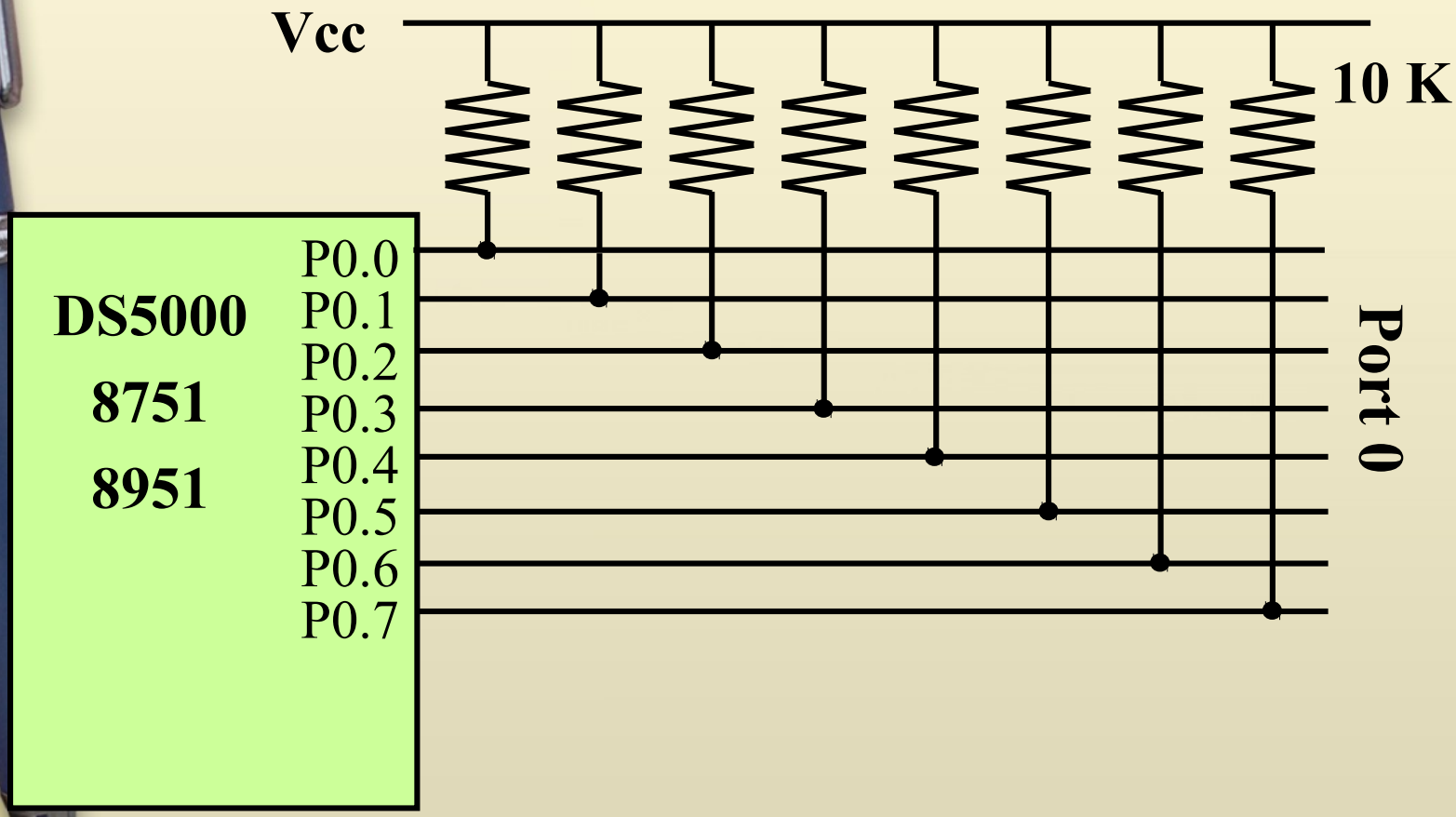
8051 Foot Print



Power-On RESET Circuit



Port 0 with Pull-Up Resistors



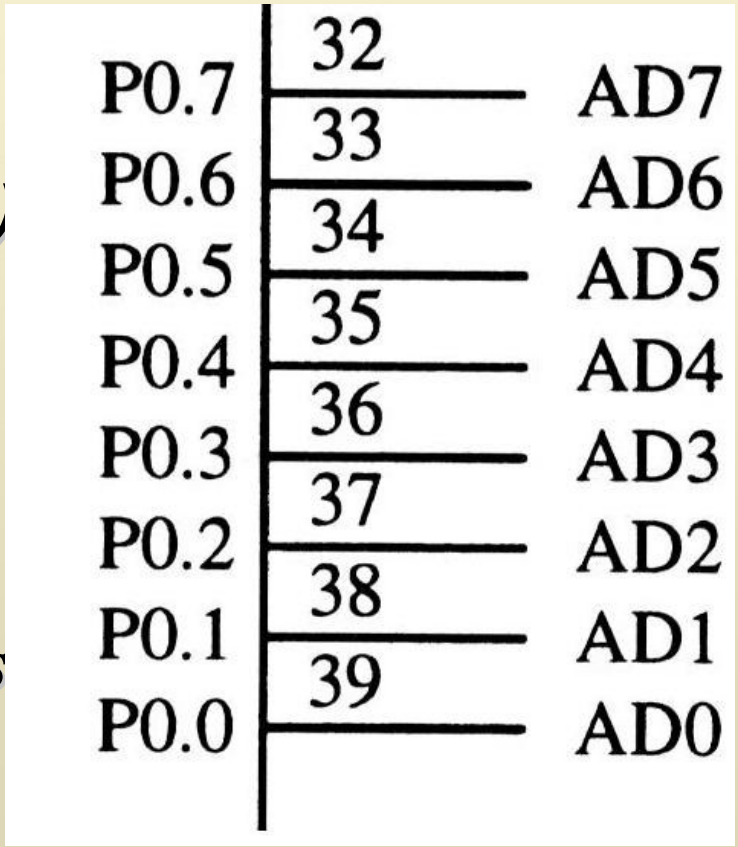
IMPORTANT PINS (IO Ports)

One of the most useful features of the 8051 is that it contains four I/O ports (P0 - P3)
Each port can be used as input or output (bi-direction)

- *Port 0*

pins 32-39 (P0.0 ~ P0.7)

  - *8-bit R/W - General Purpose I/O*
  - *Or acts as a multiplexed low byte address and data bus for external memory design*

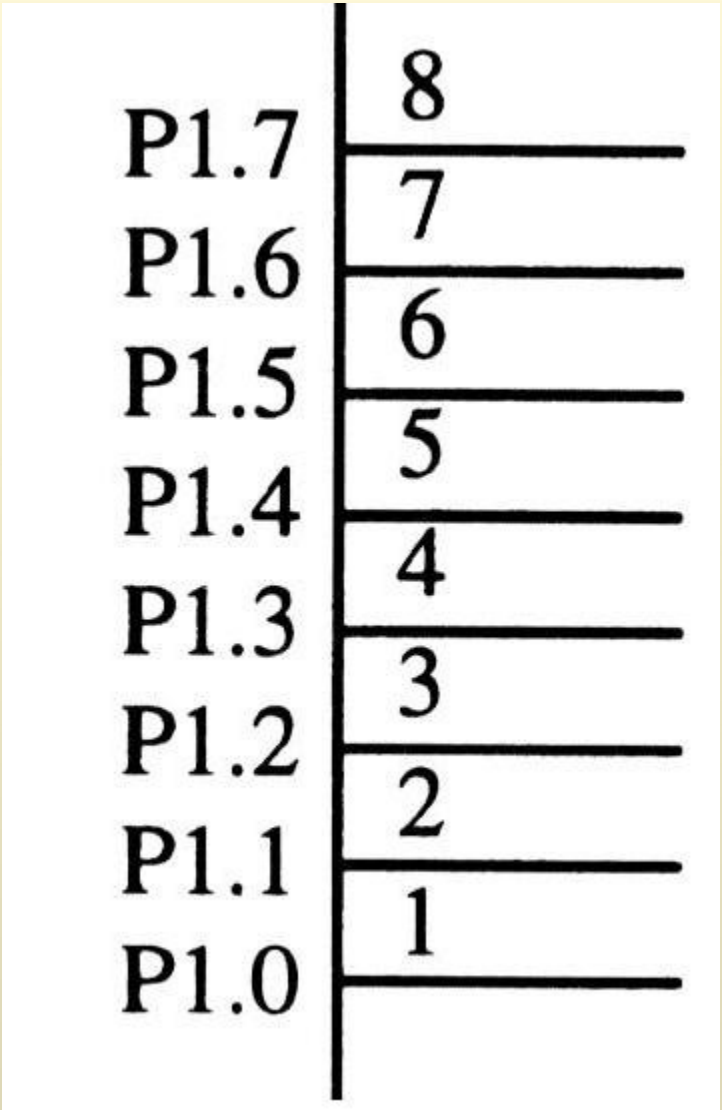


IMPORTANT PINS (IO Ports)

- *Port 1*

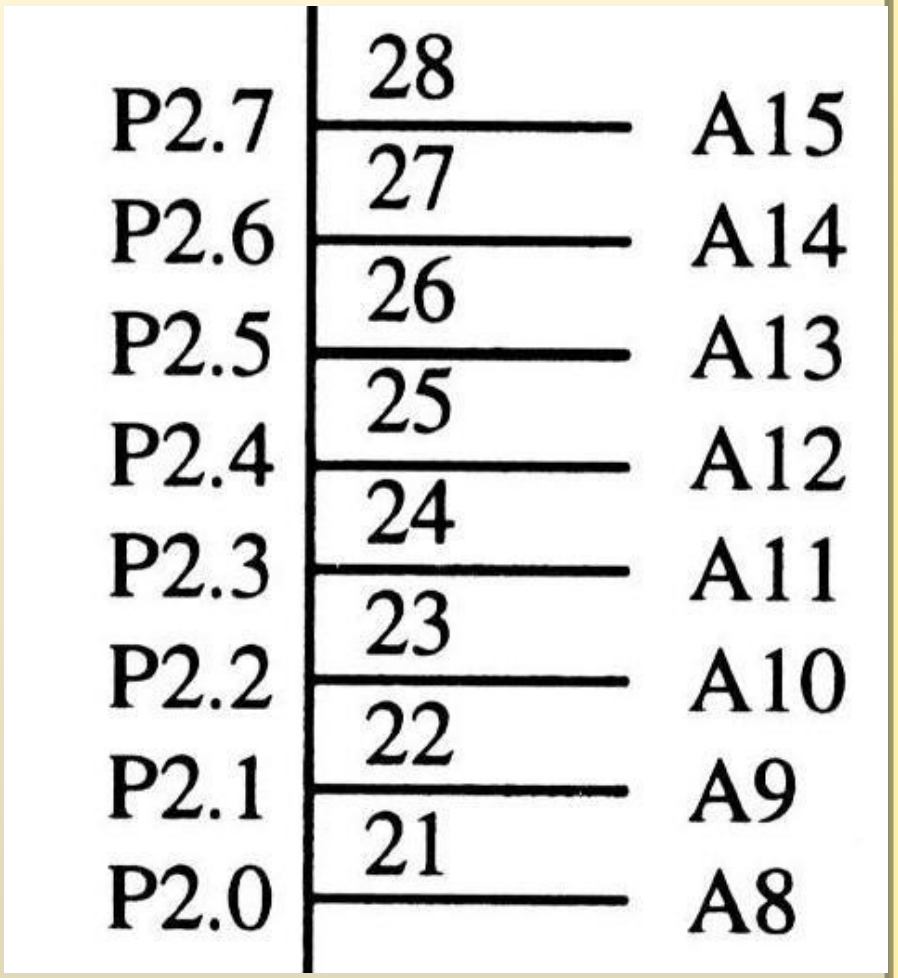
(pins 1-8) *(P1.0 ~ P1.7)*

  - *Only 8-bit R/W - General Purpose I/O*



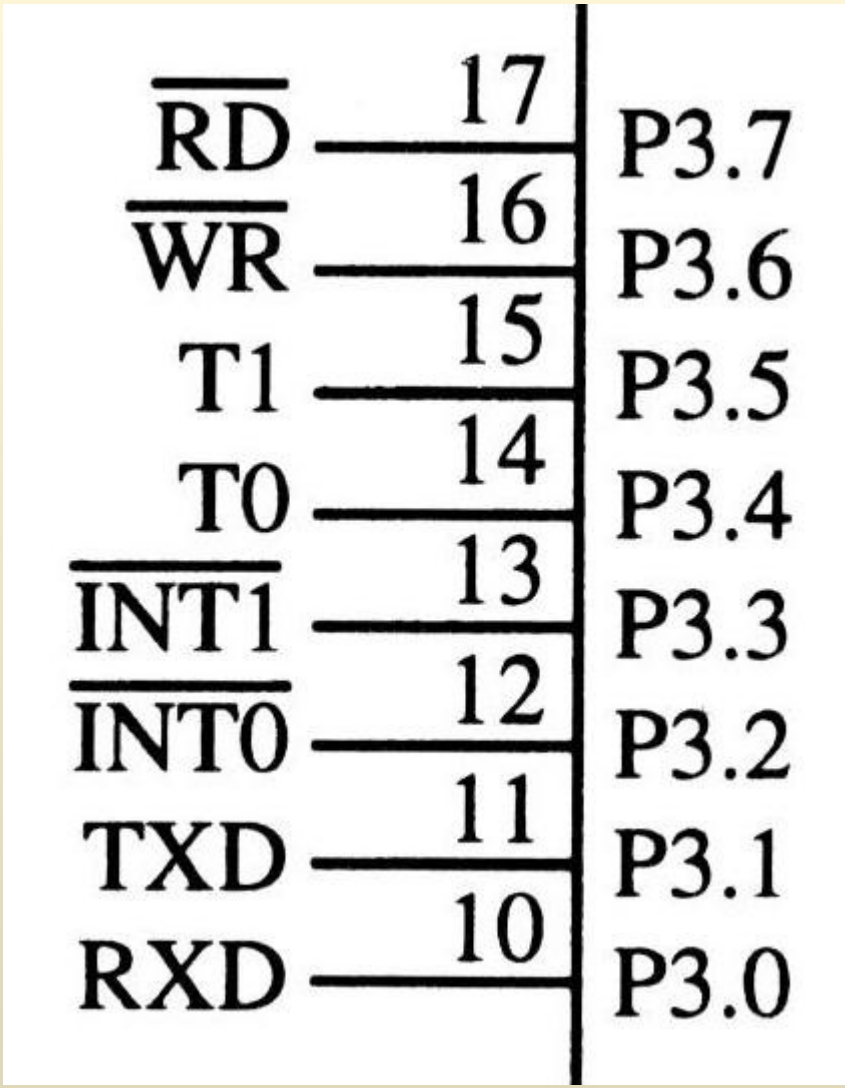
IMPORTANT PINS (IO Ports)

- *Port 2*
- *(pins 21-28 (P2.0 ~ P2.7)*
  - *8-bit R/W - General Purpose I/O*
  - *Or high byte of the address bus for external memory design*



IMPORTANT PINS (IO Ports)

- *Port 3*
- *(pins 10-17 (P3.0 ~ P3.7)*
  - *General Purpose I/O*
  - *if not using any of the internal peripherals (timers) or external interrupts.*



Port 3 Alternate Functions

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{\text{WR}}$ (external data memory write strobe)
P3.7	$\overline{\text{RD}}$ (external data memory read strobe)

- *ALE - Address latch enable to select valid address*
- *EA/V_{pp} - External access enable*

EA-0 execute program in external memory

EA-1 execute program in internal memory

Vpp it receives 21 V for on chip EPROM

PSEN Program store enable

store to read the external program memory

Registers

A
B
R0
R1
R2
R3
R4
R5
R6
R7

Some 8-bitt Registers of the 8051

DPTR

DPH	DPL

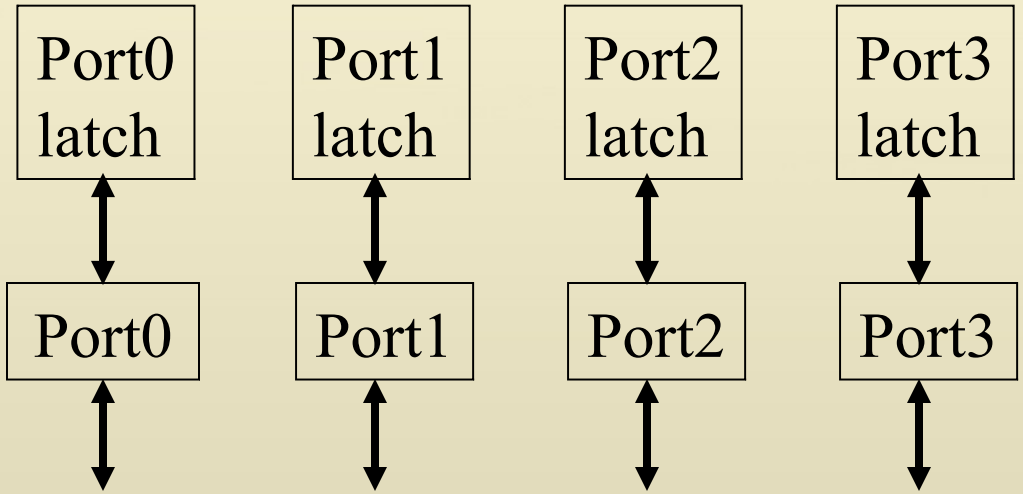
PC

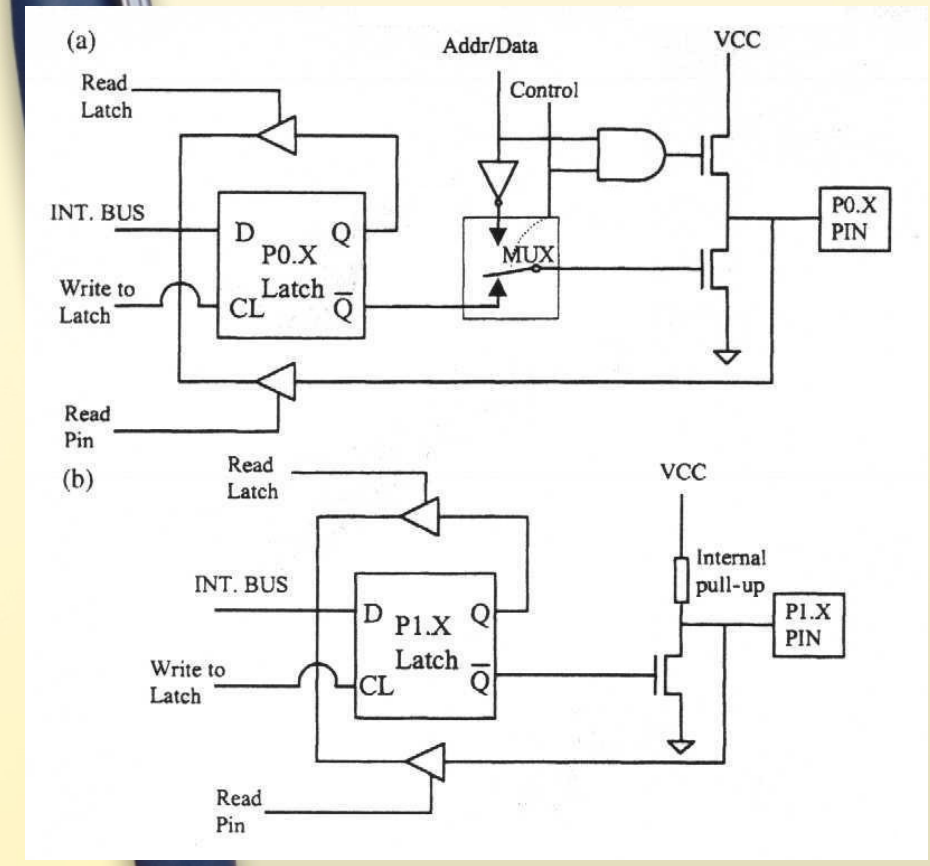
PC

Some 8051 16-bit Register

Parallel I/O Ports

- Each port can be input or output
- Direction is set in Special Function Registers





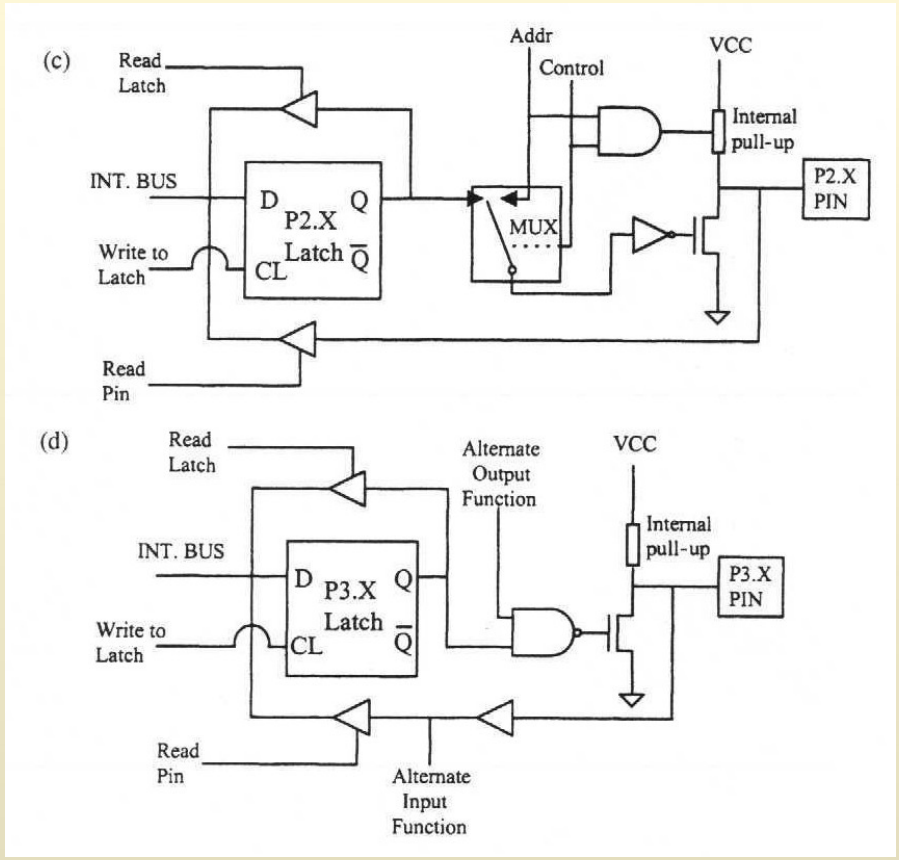


Figure 6.4 8XC51FX port bit latches and I/O buffers. (a) Port 0 bit. (b) Port 1 bit. (c) Port 2 bit. (d) Port 3 bit. (Redrawn with permission of Intel.)

DPTR

- ***The data pointer consists of a high byte(DPH) and a low byte (DPL). Its function is to hold a 16 bit address. It may be manipulated as a 16 bit data register or two independent 8 bit register. It serves as a base register in indirect jumps, lookup table instructions and external data transfer.***

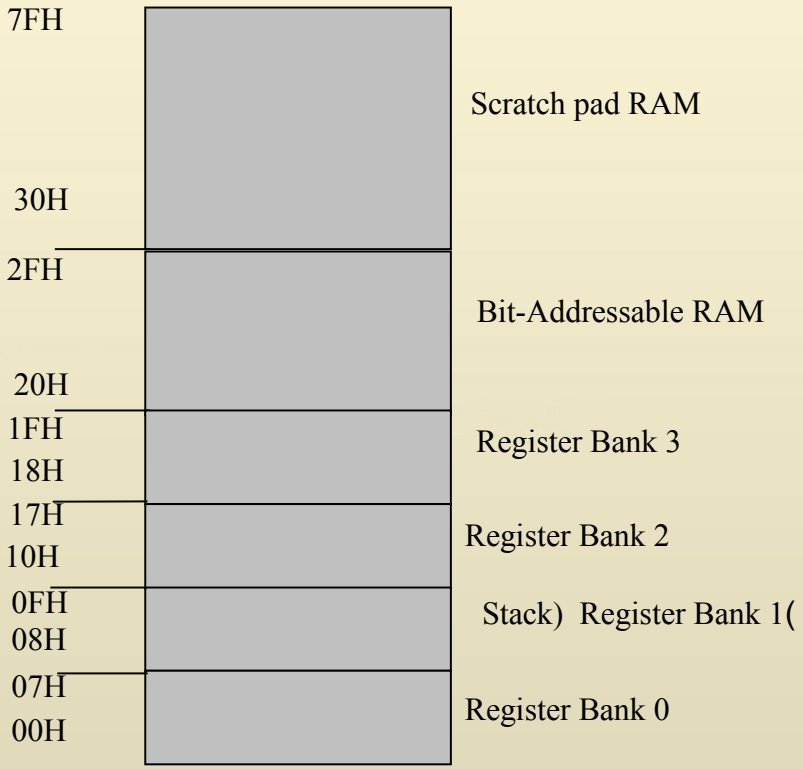
PROGRAM STATUS WORD (PSW)

CY	AC	F0	RS1	RS0	OV		P

RS0	RS1	BANK SELECTION
0	0	00H – 07H BANK0
0	1	08H – 0FH BANK 1
1	0	10H – 17H BANK2
1	1	18H – 1FH BANK 3

Stack in the 8051

- ***The register used to access the stack is called SP (stack pointer) register.***
- ***The stack pointer in the 8051 is only 8 bits wide, which means that it can take value 00 to FFH. <u>When 8051 powered up, the SP register contains value 07.</u>***



Memory Organization

- *The 8051 memory organization is rather complex.*
- *The 8051 has separate address spaces for Program Memory, Data Memory, and external RAM.*
- *This is refereed to as a Harvard architecture.*
  - *The early Mark I (1944) computer developed at Harvard was of this type of architecture.*
  - *Von Neumann at Princeton pointed out that it was not necessary to put instructions and data in separate memories.*
  - *Most machines have been Princeton architecture.*
  - *Recently Harvard architecture has been employed to help alleviate the memory bottleneck.*
- *Both program memory and external data memory are 8 bits wide and use 16 bits of address. The internal data memory is accessed using an 8-bit address.*
- *Since the same address can refer to different locations the specific location is determined by the type of instruction.*

Program or Code Memory

- *May consist of internal or external program memory. The amount of internal program memory varies depending on the device.*
  - *4K bytes typical in older devices.*
  - *The Silicon Labs C8051F310 contains 16K of flash memory for programs.*
  - *The Silicon Labs C8051F020 which is on the University Daughter Card (UDC) contains 4K bytes of program memory.*
- *The MOVC instruction can be use to read code memory.*
- *To reference code memory I will use the notation:*

CM = CM(0,…,FFFFH) = CM(0,…,FFFFH; 7,…,0)

- *This notation can be used to specify particular bits and bytes of code memory.*

For example CM(1234H) refers to the byte of code memory at address 1234H. CM(1234H;7) refers to the most significant bit in that address.

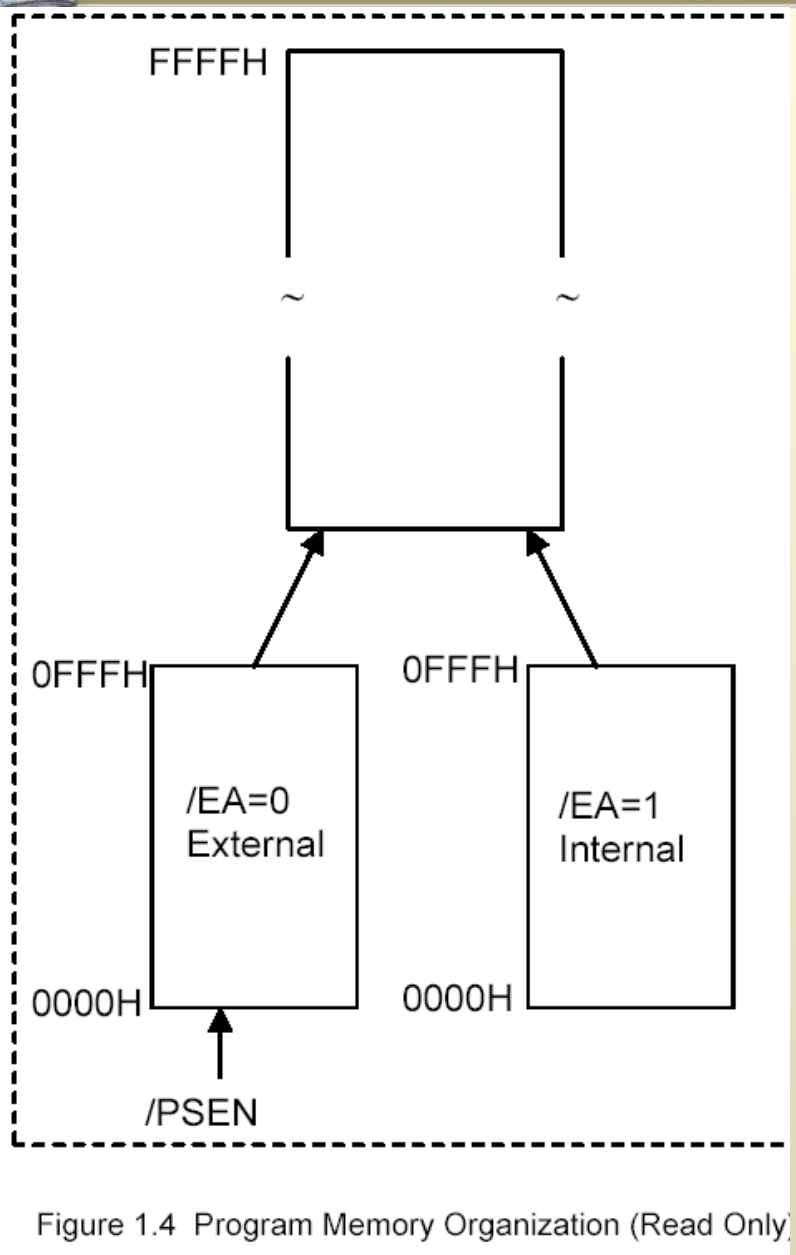


Figure 1.4 Program Memory Organization (Read Only)

MOVC A,@A + DPTR ;A ← CM(A+DPTR)
MOVC A,@A + PC ;A ← CM(A+PC)

CM

PC = PC(15..0)

DPTR = DPTR(15..0)

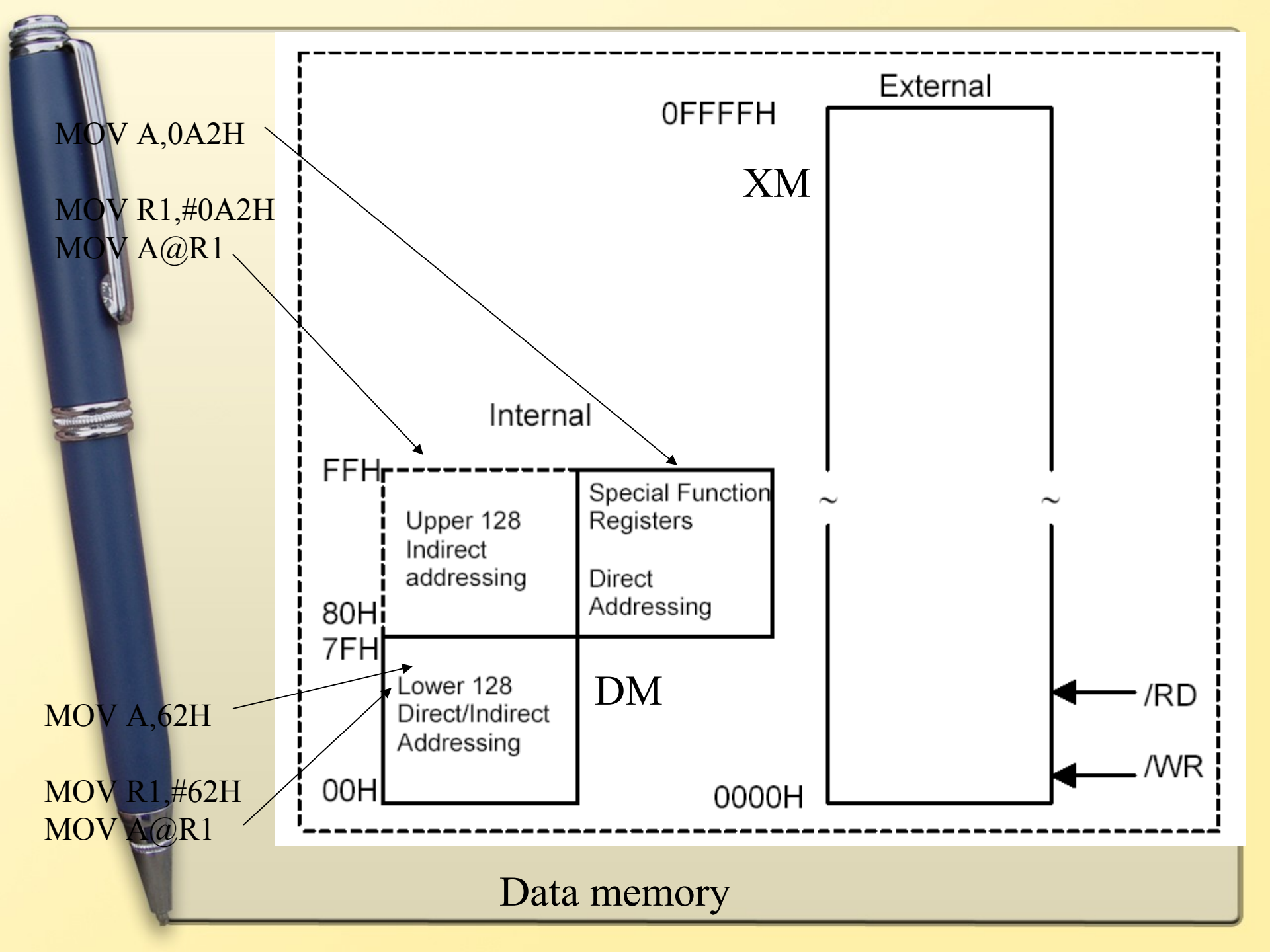
External Memory

- ***Supports up to 64K bytes external memory.***
  - ***XM(0000,…,FFFF)***
    = XM(0000,…,FFFF; 7,…,0)
  - ***Accessed by using the MOVX instruction.***
- ***On the original using external memory reduces number of available I/O ports.***
- ***On some new devices this is not the case.***
  - ***For example in C8051F020 64K bytes of external memory has been included in the chip.***
  - ***The 4 standard 8051 ports are available and three additional ports have been added.***

```
MOVX  A,@DPTR          ;A ← XM(DPTR)
MOVX   A,@Rn       ;A ← XM(P2|Rn)
MOVX    @DPTR,A        ;XM(DPTR)  ← A
MOVX   @Rn,A           ;XM(P2|Rn) ← A
```

Data Memory

- ***The original 8051 had 128 bytes of on-chip data RAM.***
  - ***This memory includes 4 banks of general purpose registers at DM(00..1F)***
  - ***Only one bank can be active at a time.***
  - ***If all four banks are used, DM(20..7F) is available for program data.***
  - ***DM(20..2F) is bit addressable as BADM(00..7F).***
- ***DM(80,…,FF) contains the special function registers such as I/O ports, timers, UART, etc.***
  - ***Some of these are bit addressable using BADM(80..FF)***
- ***On newer versions of the 8051, DM(80,…,FF) is also use as data memory. Thus, the special functions registers and data memory occupy the same address space. Which is accessed is determined by the instruction being used.***


MOV A,0A2H
MOV R1,#0A2H
MOV A@R1
MOV A,62H
MOV R1,#62H
MOV A@R1
External
0FFFFH
XM
Internal
FFH
Upper 128
Indirect
addressing
Special Function
Registers
Direct
Addressing
80H
7FH
Lower 128
Direct/Indirect
Addressing
DM
00H
0000H
/RD
/WR

Data memory

Byte Address	Bit Address							
7F	General Purpose RAM							
30								
2F	7F	7E	7D	7C	7B	7A	79	78
2E	77	76	75	74	73	72	71	70
2D	6F	6E	6D	6C	6B	6A	69	68
2C	67	66	65	64	63	62	61	60
2B	5F	5E	5D	5C	5B	5A	59	58
2A	57	56	55	54	53	52	51	50
29	4F	4E	4D	4C	4B	4A	49	48
28	47	46	45	44	43	42	41	40
27	3F	3E	3D	3C	3B	3A	39	38
26	37	36	35	34	33	32	31	30
25	2F	2E	2D	2C	2B	2A	29	28
24	27	26	25	24	23	22	21	20
23	1F	1E	1D	1C	1B	1A	19	18
22	17	16	15	14	13	12	11	10
21	0F	0E	0D	0C	0B	0A	09	08
20	07	06	05	04	03	02	01	00
1F	Bank 3							
18								
17	Bank 2							
10								
0F	Bank 1							
08								
07	Default Register Bank for R0 – R7							
00								

Bit Addressable

Byte Address	Bit Address								
FF									
F0	F7	F6	F5	F4	F3	F2	F1	F0	B
E0	E7	E6	E5	E4	E3	E2	E1	E0	ACC
D0	D7	D6	D5	D4	D3	D2	-	D0	PSW
B8	-	-	-	BC	BB	BA	B9	B8	IP
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
A8	AF	-	-	AC	AB	AA	A9	A8	IE
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
99	Not bit-addressable								SBUF
98	9F	96	95	94	93	92	91	90	SCON
90	97	96	95	94	93	92	91	90	P1
8D	Not bit-addressable								TH1
8C	Not bit-addressable								TH0
8B	Not bit-addressable								TL1
8A	Not bit-addressable								TL0
89	Not bit-addressable								TMOD
88	8F	8E	8D	8C	8B	8A	89	88	TCON
87	Not bit-addressable								PCON
83	Not bit-addressable								DPH
82	Not bit-addressable								DPL
81	Not bit-addressable								SP
80	87	86	85	84	83	82	81	80	P0

Data Memory (DM)

Table 1

Symbol	Name	Address
*ACC	Accumulator	0E0H
*B	B Register	0F0H
*PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 2 Bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
*P0	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	0A0H
*P3	Port 3	0B0H
*IP	Interrupt Priority Control	0B8H
*IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
*TCON	Timer/Counter Control	88H
*+T2CON	Timer/Counter 2 Control	0C8H
TH0	Timer/Counter 0 High Byte	8CH
TL0	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
+TH2	Timer/Counter 2 High Byte	0CDH
+TL2	Timer/Counter 2 Low Byte	0CCH
+RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
+RCAP2L	T/C 2 Capture Reg. Low Byte	0CAH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

* = Bit addressable
+ = 8052 only

Intel 8051: Timers/Counters

Timer

- ***A timer is a counter that is increased with every time an instruction is executed e.g. 8051 with 12MHz increases a counter every 1.000 µs***

- ***General 8051 has 3 timer:***
  - ***2 16-bit timer***
  - ***1 16-bit timer with extra-functionality (introduced with the 8052)***

Timer/Counter Mode Control Register TMOD

GATE	T/C	M1	M0	GATE	T/C	M1	M0
Timer/Counter1				Timer/Counter0			

Timer/Counter Control Register TCON

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Uses of Timers & Counters

- Interval Timing
- Periodic event timing
- Time base for measurements
- Event Counting
-Baud Rate Generation

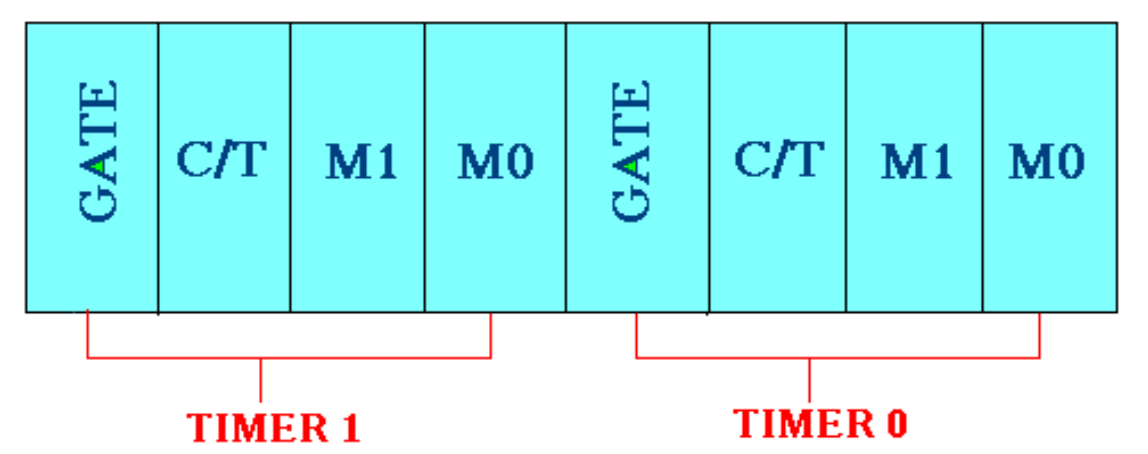
8051 Timers

- 2 timers (Timer 0 and Timer 1)
- 16-bit timers (65,535) max
- Flag is set when the timer overflows
-Timers can be based on internal clock (OSC/6) or from external source (counter mode).

TMOD - Timer/Counter mode register

TCON - Timer/Counter control register

TMOD Register:



- **Gate :** *When set, timer only runs while INT(0,1) is high.*
- **C/T :** *Counter/Timer select bit.*
- **M1 :** *Mode bit 1.*
- **M0 :** *Mode bit 0.*

M1	M0	MODE
0	0	13-bit timer mode
0	1	16-bit timer mode
1	0	8-bit auto-reload mode
1	1	split mode

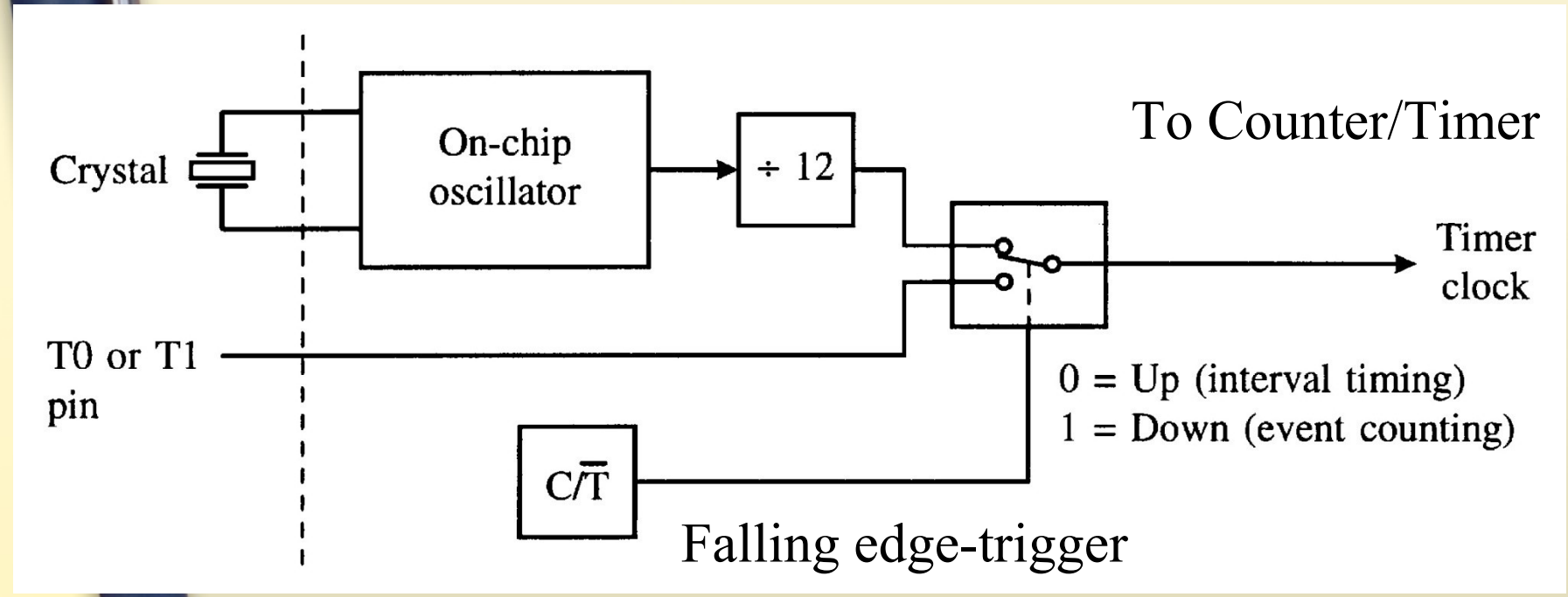
TCON Register:

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

- *TF: Overflow flag*
  - *Set by hardware on Timer/Counter overflow*
  - *Cleared by hardware when processor vectors to interrupt routine*
- *TR: Run control bit*
  - *Set/Cleared by software to turn Timer/Counter on/off*
- *IE: Interrupt Edge flag*
  - *Set by hardware when external interrupt edge detected*
  - *Cleared when interrupt processed*
- *IT: Interrupt Type control bit*
  - *Set/Cleared by software to specify falling edge/low level triggered external interrupts*

- *TF1: Timer 1 overflow flag.* *TR1: Timer 1 run control bit.*
- *TF0: Timer 0 overflag.* *TR0: Timer 0 run control bit.*
- *IE1: External interrupt 1 edge flag.* *IT1: External interrupt 1 type flag.*
- *IE0: External interrupt 0 edge flag.* *IT0: External interrupt 0 type flag.*

Internal clock



External clock

Timer Modes

- 0: 13 bit timer
- 1: 16-bit timer
- 2: 8-Bit auto reload
-3: Split timer mode

Mode 0: 13-Bit Timer

- Lower byte (TL0/TL1) + 5 bits of upper bytes (TH0/TH1).
- Backward compatible to the 8048
- Not generally used

Mode 1: 16-bit

- All 16 bits of the timer (TH0/TL0, TH1,TL1) are used.
- Maximum count is 65,536
-At 12Mhz, maximum interval is 65536 microseconds or 65.536 milliseconds
- TF0 must be reset after each overflow
- THx / TLx must be manually reloaded after each overflow.

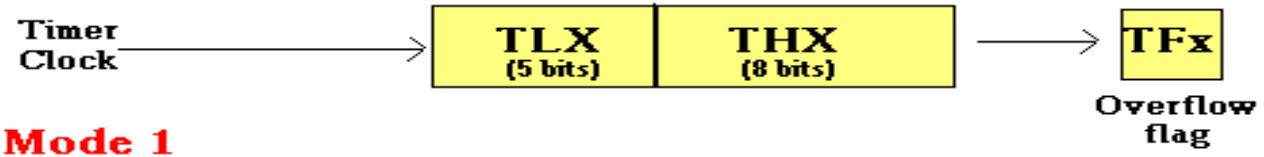
Mode 2: 8-bit Auto Reload

- Only the lower byte (TLx) is used for counting.
- Upper byte (THx) holds the value to reload into TLx after an overflow.
- TFx must be manually cleared.
- Maximum count is 256
- Maximum interval is 256 Microseconds or .256 milliseconds

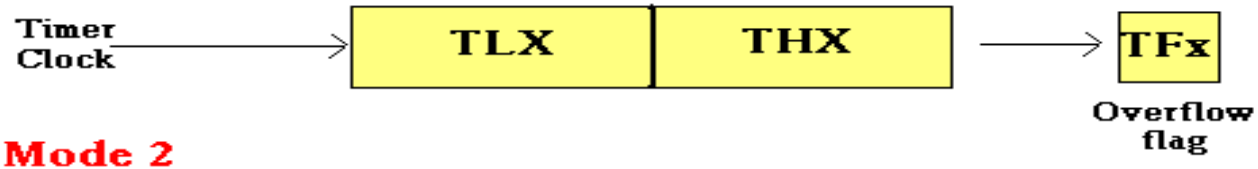
Mode 3- Split Timer

- Splits Timer 0 into two 8-bit timers
- TL0 sets TF0
- TH0 sets TF1
- Timer 1 is available for other 3 modes, but the TF1 is not available.

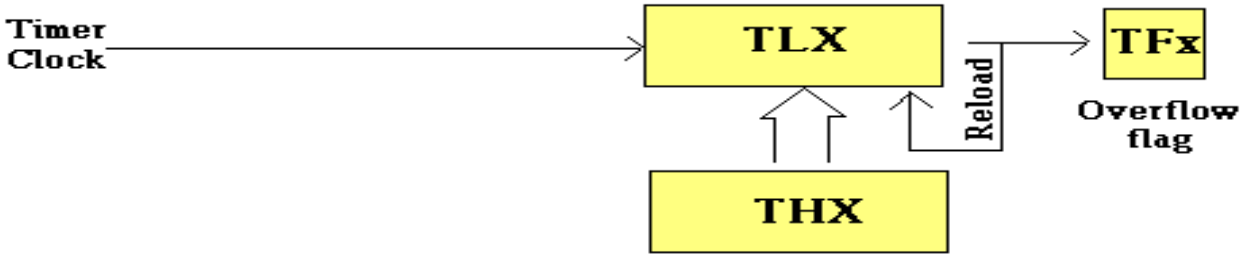
Mode 0



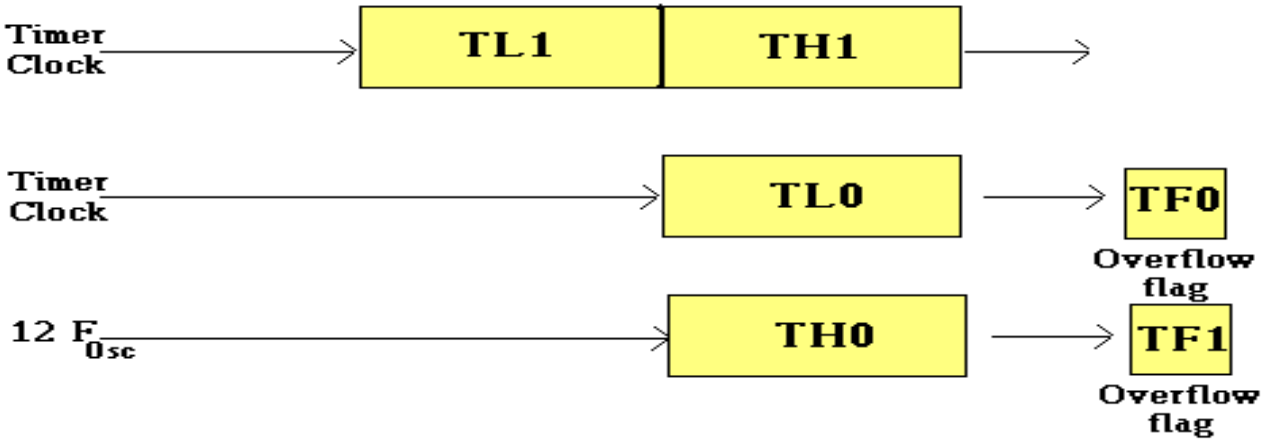
Mode 1



Mode 2



Mode 3



Timer Delay and Timer Reload Value

Timer Delay = Delay Value × Timer Clock Cycle Duration

Delay Value = how many counts before register(s) roll over

Timer Clock Cycle Duration = 6/oscillator frequency

Delay Value = Maximum Register Count – Timer Reload Value

Maximum Register Count = 65535