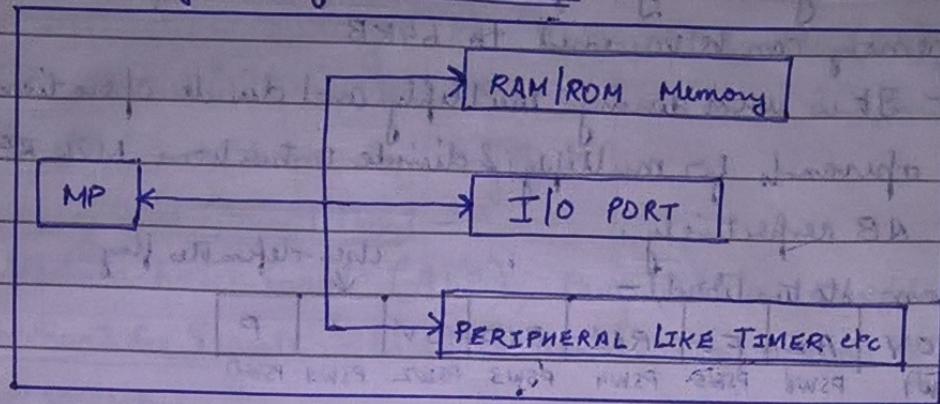


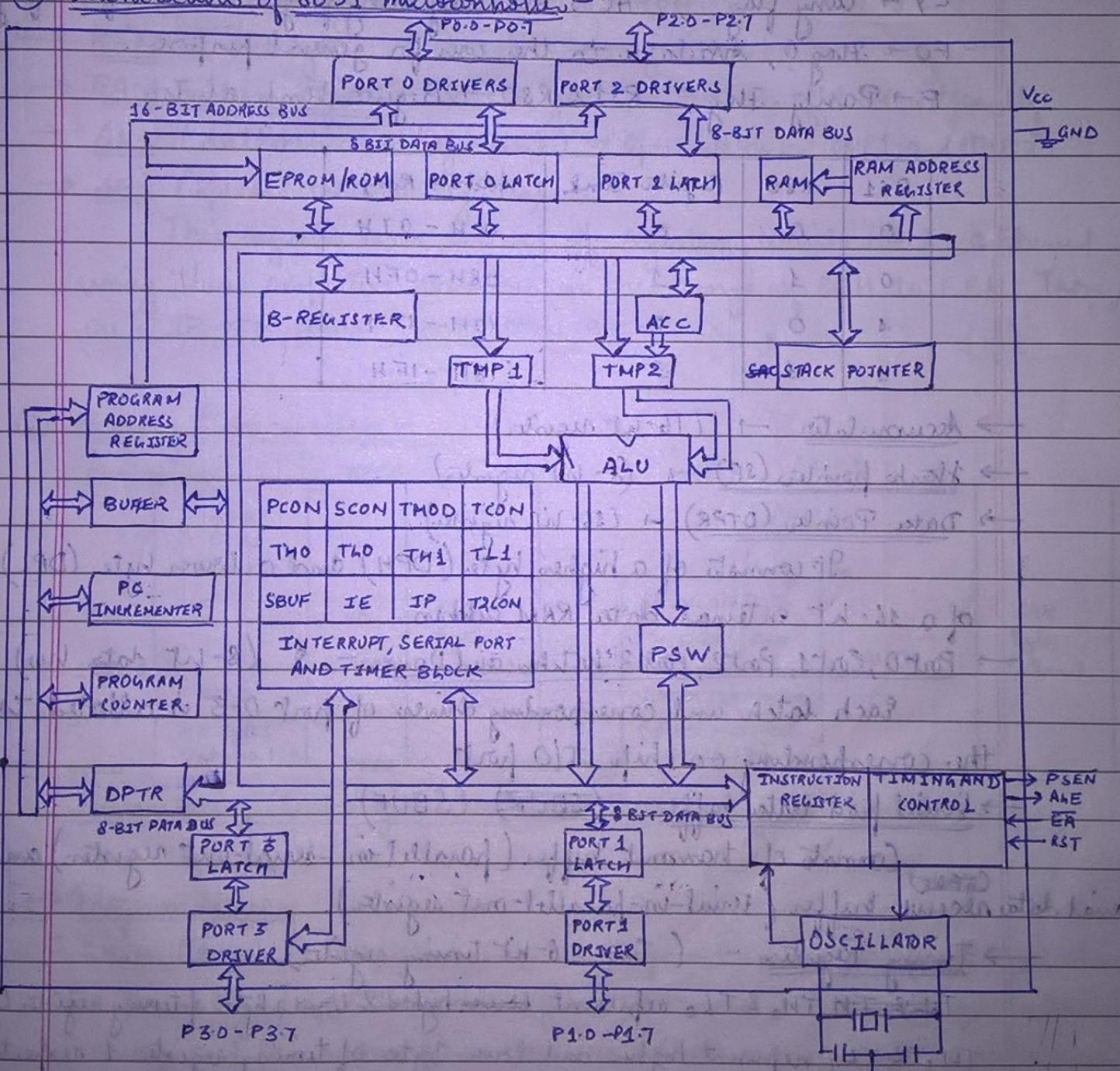


8051 MICROCONTROLLER

① Block diagram of microcontroller -



② Architecture of 8051 microcontroller -



- 4KB on-chip memory, 128 bytes of data memory.
- Program memory can be increased to 64KB.
- B Registers - It is used during multiply and divide operations to store the second operands for multiply & divide instructions MUL AB & DIV AB and DIV AB respectively.
- PSW (Program Status Word) - User-definable flag

CY	AC	FO	RS1	RS0	OV	P	
PSW7	PSW6	PSW5	PSW4	PSW3	PSW2	PSW1	PSW0

CY → Carry flag, AC → Auxiliary Carry flag, OV → Overflow flag
 FO → Flag 0, Available to the user for general purpose.
 P → Parity flag, RS1 & RS0 → Register bank selector.

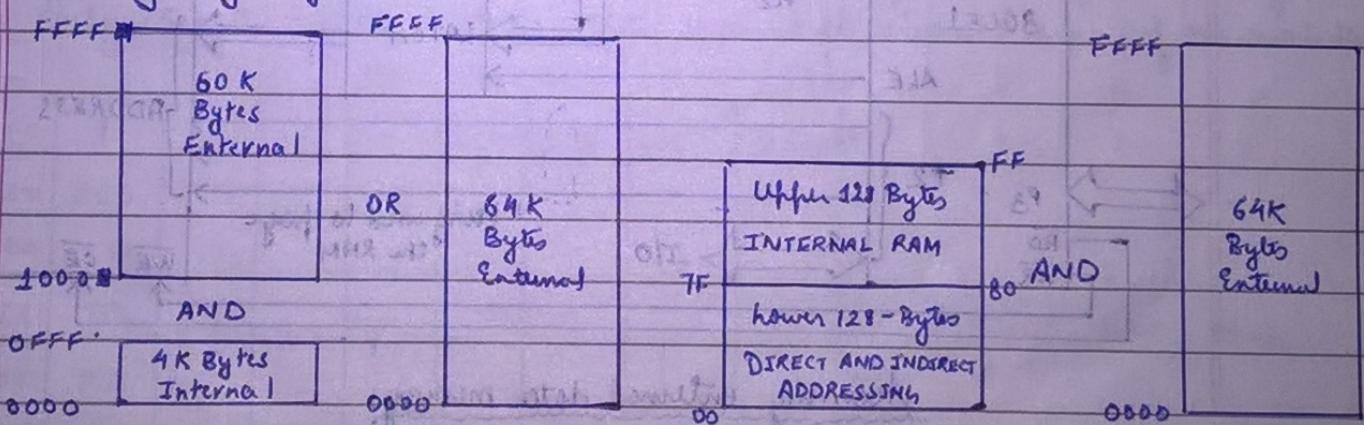
RS1	RS0	Register Bank	Address Range
0	0	0	00H - 07H
0	1	1	08H - 0FH
1	0	2	10H - 17H
1	1	3	18H - 1FH

- Accumulator → (16-bit register)
- Stack pointer (SP) → (8-bit register)
- Data Pointer (DTPR) → (16-bit register)
- It consists of a higher byte (DPH) and a lower byte (DPL) of a 16-bit external data RAM address.
- Port 0, Port 1, Port 2, Port 3 latches and Drivers - (8-bit data bus)
 Each latch and corresponding driver of port 0-3 is allotted to the corresponding on-chip I/O port.
- Serial port data buffer - (~~SBUF~~) (SBUF)
 Consists of transmit buffer (parallel-in-serial-out register) and serial data receive buffer (serial-in-parallel-out register).
- Timing Registers - (Two 16-bit timing registers)
 TH0 & TH1 TH0 & TL0 represent higher & lower byte of timing register 0.
 TH1 & TL1 represent higher and lower byte of timing register 1 respectively.

- Control Registers - (8-bit registers)
 - Instruction Priority (IP), Interrupt Enable (IE), Timer Mode (TMOD), Timer Control (TCON), Serial Port Control (SCON) and Power Control (PCON)
- Oscillator → quartz crystal
- Timing and Control Unit -
 - Address latch enable (ALE), Program store enable (PSEN), RD, WR
- Instruction Register - Used to decode the opcode of any instruction to be executed.
- Program Address Register - On-chip EEPROM
- RAM - ^{internal} 128 bytes (overall 256 bytes on-chip)
- RAM Address Register - Used to generate address of RAM internally.
- AHV (Arithmetical and logic unit) → Operands are in TMP1 and TMP2
- SFR (Special Function Registers) -

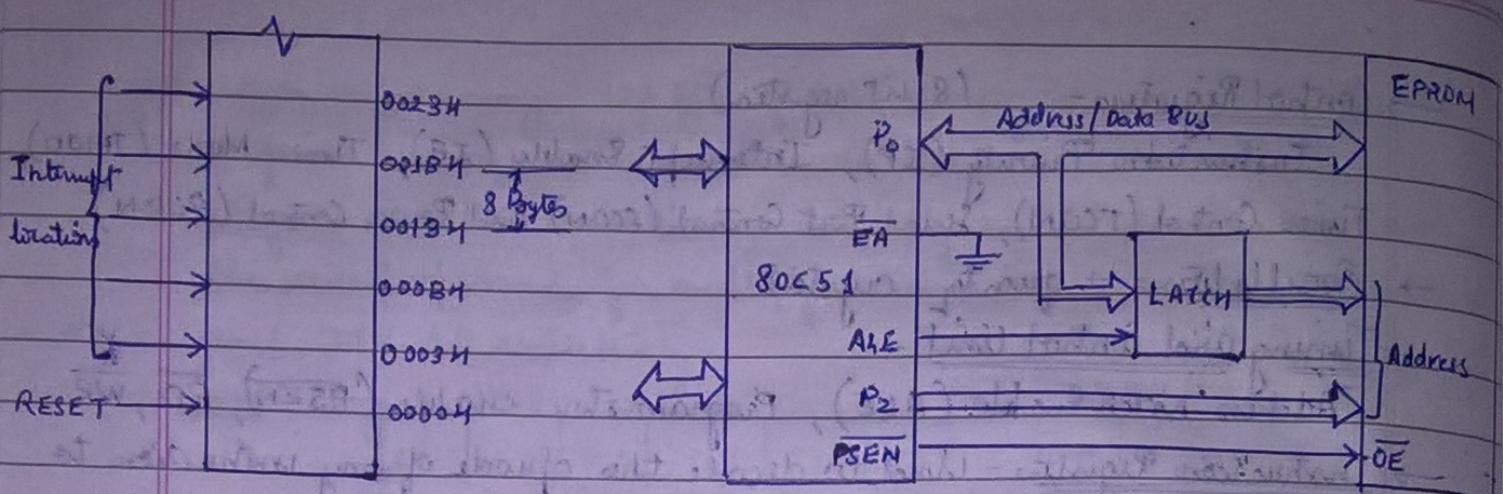
This register bank is a set of registers, which can be addressed using their respective addresses in the range of 80H to FFH. These are - IP, IE, TMOD, TCON, SCON, PCON

② Memory Organization -



→ Program Memory -

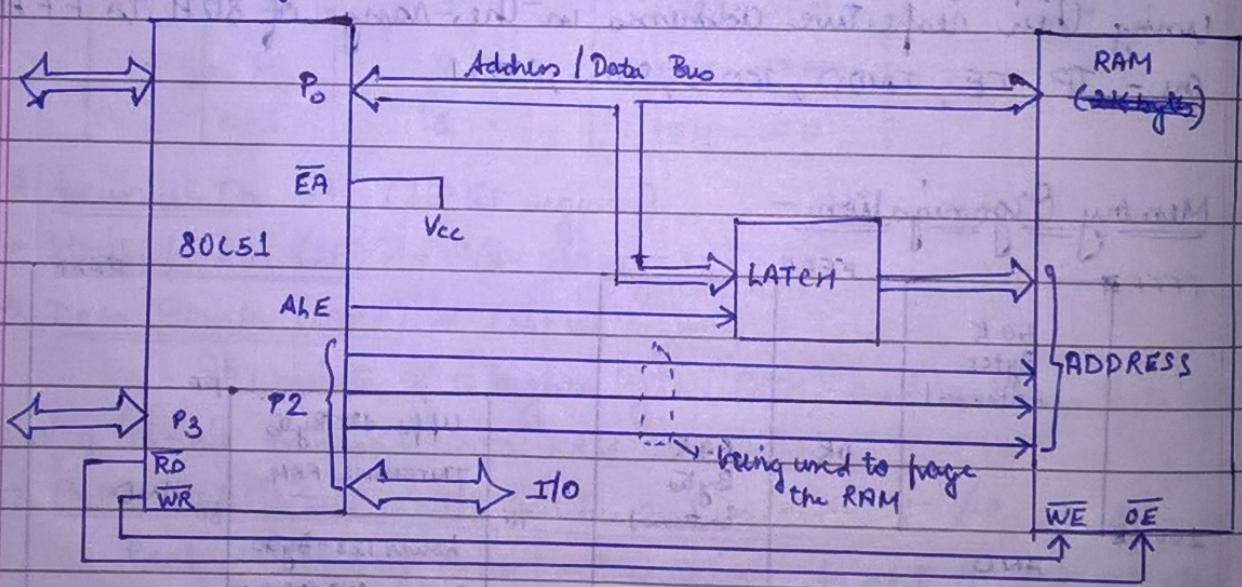
When EA pin is low / grounded, the program memory is external and when EA pin is high / Vcc, the address from 0000H to 0FFFH will refer to on-chip memory and the address from 1000H to FFFFH can refer to external memory.



8051 Program Memory

Executing memory from external program memory
When ALE is high, Port 0 is used as the low byte of the Program Counter (PC) as an address. Port 2 is used as the high byte of the Program Counter (PC). Then PSEN strobes the EPROM and the code byte is read into the microcontroller.

Data Memory



Accessing external data memory

MOVX instruction is used to access the external data memory.

However 128 bytes of RAM can be divided into three segments -

- ① Register Banks - 00H-1FH (4-Banks and each bank has 7 registers (R0-R6))
- ② Bit Addressable Segments - 20H-2FH (Addressed as either bytes or individual bits)
- ③ Scratch Pad Area - 30H-7FH one general purpose RAM.

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		→ Bit addressable RAM
2C	67	66	65	64	63	62	61	60		
2B	5F	EE	5D	5C	5B	5A	59	58		
2A	57	56	55	54	53	52	51	50	07H	R7
29	4F	4E	4D	4C	4B	4A	49	48	06H	R6
28	47	46	45	44	43	42	41	40	05H	R5
27	3F	3E	3D	3C	3B	3A	39	38	04H	R4
26	37	36	35	34	33	32	31	30	03H	R3
25	2F	2E	2D	2C	2B	2A	29	28	02H	R2
24	27	26	25	24	23	22	21	20	01H	R1
23	1F	1E	1D	1C	1B	1A	19	18	00H	R0
22	17	16	15	14	13	12	11	10		BANK-0
21	0F	0E	0D	0C	0B	0A	09	08		
20	07	06	05	04	03	02	01	00		

Upper 128 bytes of the on-chip RAM are used for special function registers -
Only 25 bytes are used. Other bytes reserved for advanced versions of microcontroller.

E8-EF									F7	
E8-EF	F0	B								
D8-DR	E0	ACC							E7	
D8-DR	00	PSW							D7	
C8-CF	B8	IP							BF	→ SFR Memory Map
CO-C7	B0	P3							B7	
A8	IE								AF	
A0	P2								A7	
98	SCON								9F	
90	P1								97	
88	TCON	TMOD	TLO	TL1	TH0	TH1			8F	
80	PO	SP	DPL	DPH				PCON	87	

④ Timers / Counters -

8051 microcontroller has two 16-bit timers / counters such as TIMER0 (T0) and TIMER1 (T1). Each timer can be programmed to count internal clock pulses of 8051 microcontroller. These timers are used for the following functions -

- Calculate time delay between two events
- Counting the number of events
- Generate baud rate for serial ports
- Frequency measurement
- Pulse width measurement

$$\text{Counting Rate} = \frac{\text{Oscillator frequency}}{12}$$

Timer 1 registers

	TH ₁								TL ₁							
	D ₁₅	D ₁₄	D ₁₃	D ₁₂	D ₁₁	D ₁₀	D ₉	D ₈	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀

Timer 0 registers

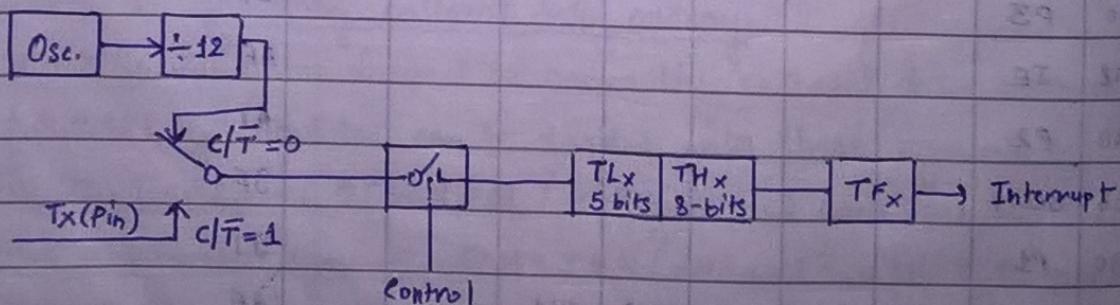
	TH ₀								TL ₀							
	D ₁₅	D ₁₄	D ₁₃	D ₁₂	D ₁₁	D ₁₀	D ₉	D ₈	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀

→ Operation modes -

M ₁	M ₀	Operating modes	Functions
0	0	Mode 0	13-bit timer mode
0	1	Mode 1	16-bit timer mode
1	0	Mode 2	8-bit timer mode
1	1	Mode 3	split timer mode

Timer mode 0 - Count value $\rightarrow 0000H - 1FFFH$.

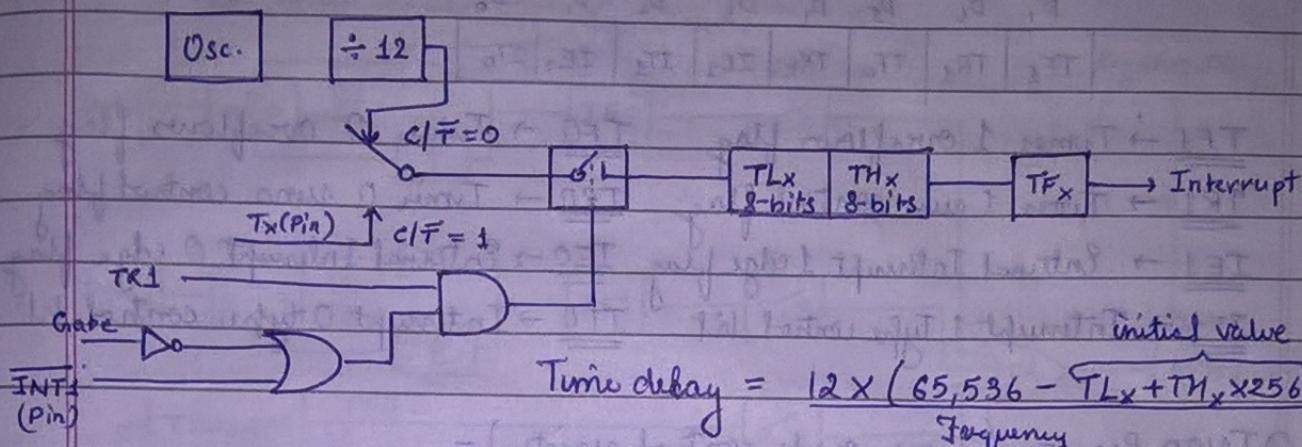
Overflow flag is set to zero after $2^5 \times 2^8 = 8192$ machine cycles.



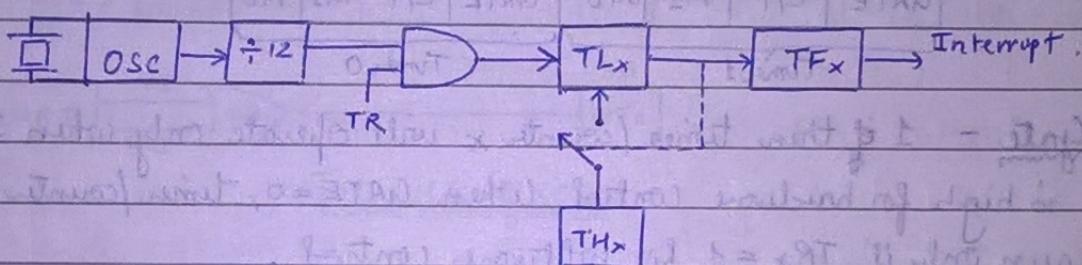
Clock frequency input to TH₀ is $\frac{\text{Oscillator frequency}}{12 \times 2^5}$

Timer mode 1 - Count value $\rightarrow 0000H - FFFFH$

Overflow flag is set to zero after $2^8 \times 2^8 = 65536$ machine cycles



Timer mode 2 - (has auto-reload feature) Count value $\rightarrow 00H - FFH$

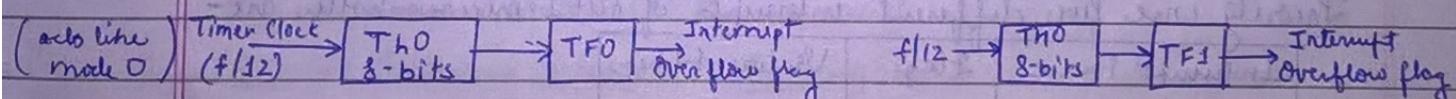


Overflow flag is set to zero after $2^8 = 256$ machine cycles.

Time delay between overflows = $\frac{12 \times (256 - \text{TH}_x)}{\text{Frequency}}$

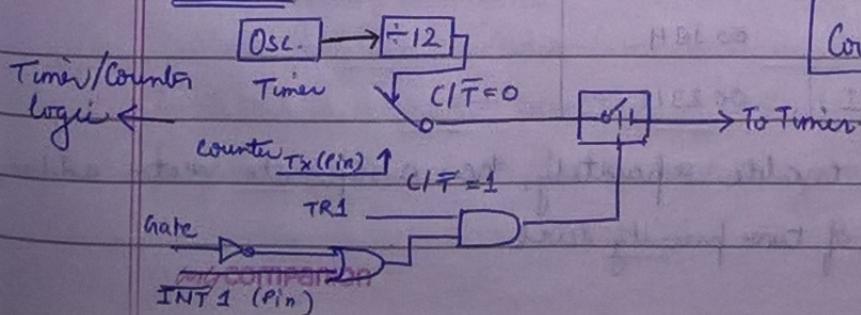
Timer mode 3

Timer 0 divides into two 8-bit counters/Timers TH0 and TH1. TH0 and TH1 are two separate timers with overflow flags TF0 and TF1 respectively.



TH0 counts CPU cycles using TR1, TF1 and Timer 1 interrupt.

→ Counters -



Source of clock pulses
 Timer → Oscillator output pulse
 Counters → Tx pin (T₀ and T₁)

→ Control Registers-

① TCON Register (Timer/Counter Control register) -

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
TF_1	TR_1	TF_0	TR_0	IE_1	IT_1	IE_0	JT_0

TF1 → Timer 1 overflow flag TFO → Timer 0 overflow flag
TR1 → Timer 1 run control flag TR0 → Timer 0 runs control flag.
IE1 → External Interrupt 1 edge flag IE0 → External Interrupt 0 edge flag
IT1 → Interrupt 1 type control bit IT0 → Interrupt 0 type control bit

③ TMOD Register (Time mode control register) -

The timing diagram illustrates the sequence of memory operations. It shows two timer periods: Timer 1 and Timer 0. The sequence of events for each timer period is:

- Timer 1:** GATE → C/F → M1 → MO
- Timer 0:** GATE → C/F → M1 → MO

Gate - 1 if then timer / counter x will operate only when INT_x pin is high for hardware control. When GATE = 0, timer / counter x will run only if TR_x = 1 for software control.

C/T (Clock/Timer) - When timer is used for time delay otherwise when timer is used as counter by counting pulse from external input pin Tx (T_1 and T_0).

M1 & M0 - Timer/Counter operating mode selector bits

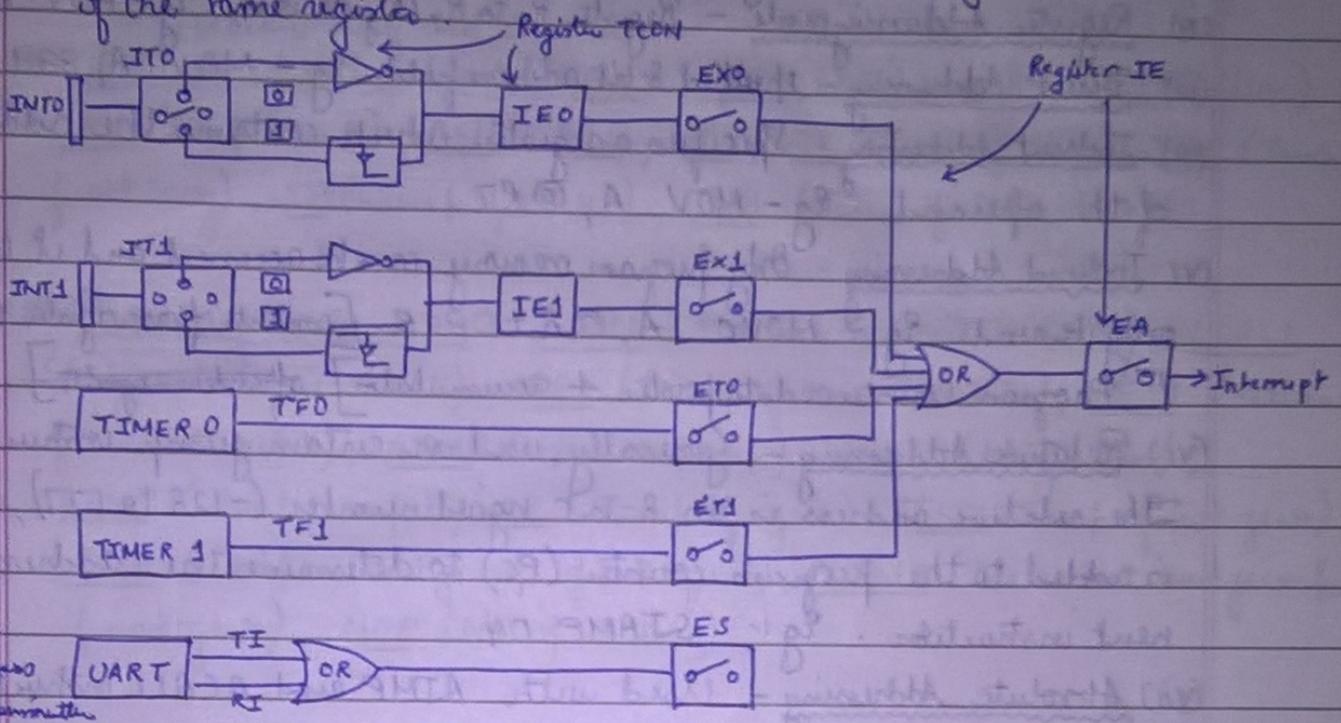
⑤ Interrupts of 8051-

Priority wise five different interrupts of 8051 microcontroller are -

Interrupt Source	Flag	Vector Address
External Interrupt 0	IE0	0003H
Timer 0	TF0	000BH
External Interrupt 1	IE1	0013H
Timer 1	TF1	001BH
Serial Port	RIRTI	0029H

→ Each interrupt can be enable separately, has a separate vector address, can be programmed to one of two priority levels.

→ External interrupts can be enabled or disabled by setting bit of IE (Interrupt enable). Whole interrupt system can be disabled by clear the EA bit of the same register.



Interrupts of 8051

→ Interrupt Control Registers -

① Interrupt Enable (IE) Register -

	7	6	5	4	3	2	1	0
EA	X	X	ES	ET1	EX1	ETO	EX0	

EA → Global interrupt enable / disable ET_x → Enable Timer _x interrupt

ES → Enable Serial Interrupt

EX_x → Enable External _x interrupt

② Interrupt Priority (IP) register -

	X	X	X	PS	PT1	PX1	PT0	PX0
D7	D6	D5	D4	D3	D2	D1	D0	

PS → Serial port interrupt priority level

PT_x → Timer _x interrupt priority level

PX_x → External interrupt _x priority level

③ Timer Control Register (TCON)

→ Execution of interrupt -

Main program execution → interrupt occurs → load vector address of interrupt (ISR)

Jump back to main ← execution of interrupt service routine → Jump to vector address ←

program execution → start executing next instruction in main program.

Indirect addressing is presented by an @ symbol before R₀ to R₇ register.

The immediate operands are preceded by # sign in assembly language.
addr 11 → 11-bit address

⑥ Addressing modes of 8051

- (i) Immediate Addressing - load data immediately Eg - MOV A, #FFH
- (ii) Register Addressing mode - Registers R₀ to R₇ (Register bank) Eg - MOV A, R₀.
- (iii) Direct Addressing - specified 8-bit address field. Eg - MOV A, 33H
- (iv) Indirect Addressing - Specifies a register which contains the address of the operand. Eg - MOV A, @R₇.
- (v) Indexed Addressing - Only program memory can be accessed and it can only be read. Eg - MOVC A, @A + D PTR [contents of base register + offset] [Program Counter or data pointer + accumulator] of instruction
- (vi) Relative Addressing - Generally, used in certain jump instructions. The relative address is an 8-bit signed number (-128 to 127), which is added to the program counter (PC) to determine the address of next instruction. Eg - SJMP 04
- (vii) Absolute Addressing - Used with AJMP and ACALL instructions. The 11 least significant bits of the destination address comes from the opcode and the upper five bits are the current upper five bits of in the program counter (PC). Eg - ACALL addr11
- (viii) Long Addressing - Used with LJMP and LCALL instructions. These instructions include a full 16-bit destination address. Eg - LJMP 9500H

⑦ 8051 Instruction set - (255 possible instructions)

(i) Arithmetic Instructions - ADD, ADDC (Add register to accumulator with carry), SUBB (Subtract from accumulator)

(ii) Arithmetic Instructions - ADD, ADDC (Add with carry), SUBB (Subtract with borrow), INC, DEC, MUL, DIV, DAA (Decimal adjustment for addition),

(3) Logical Instructions - ANL (logical AND), ORL (logical OR), XRL (logical Exclusive-OR), CLR (Clear), CPL (Complement), RL (Rotate left), RLe (Rotate left through carry), RR (Rotate Right), RRe (Rotate Right through carry), SWAP (Swap nibbles within accumulator)

$$\rightarrow (A_{3-0}) \leftrightarrow (A_{7-4})$$

Absolute → within same 2K block
long → anywhere in 64K memory
~~Absolute~~ → ~~unconditional~~
short → preceding 128 - following 127

(3) Data Transfer Instructions - MOV, MOVc (Move word byte), MOVX
(transfer data between accumulator and a byte of external memory)

(4) Boolean Operations instructions - CLR (Clear), SETB (Clear)
CPL (Complement), ANL, ORL, MOV, JC (Jump if carry), JNC (Jump if not carry), JB (Jump if bit set), JNB (Jump if bit not set)

(5) Branching Instructions - ACALL (Absolute subroutine CALL), LCALL
(long subroutine CALL), RET (Return from subroutine), RETI (Return from interrupt), ATMP (Absolute Jump), LJMP (long Jump),
SJMP (short jump), JZ (Jump if zero), JNZ (Jump if not zero)
CINE (Compare and jump if not equal), DJNZ (Decrement and jump if not zero), NOP (No operation)

(6) PUSH, POP and Exchange instructions - PUSH, POP, XCH (Exchange),
XLHD (Exchange lower-order digit).

⑧ Assembly language programs -

Eg - Add 49H, which is in the external memory location 9001H, and 56H, which is in the external memory location 9002H. The result is to be stored in the external memory location 9003H.

Sol - MOV DPTR, #9001

MOVX A, @DPTR

MOV B, A

INC DPTR

MOVX A, @DPTR

ADD A, B

INC DPTR

MOVX @DPTR, A

LJMP 0000

⑨ Applications of microcontrollers -

- (1) Stepper-motor control
- (2) traffic light control
- (3) Display (LCD's)
- (4) A/D converter interfacing
- (5) Keyboard Interface
- (6) Washing Machine Control
- (7) Time Delay