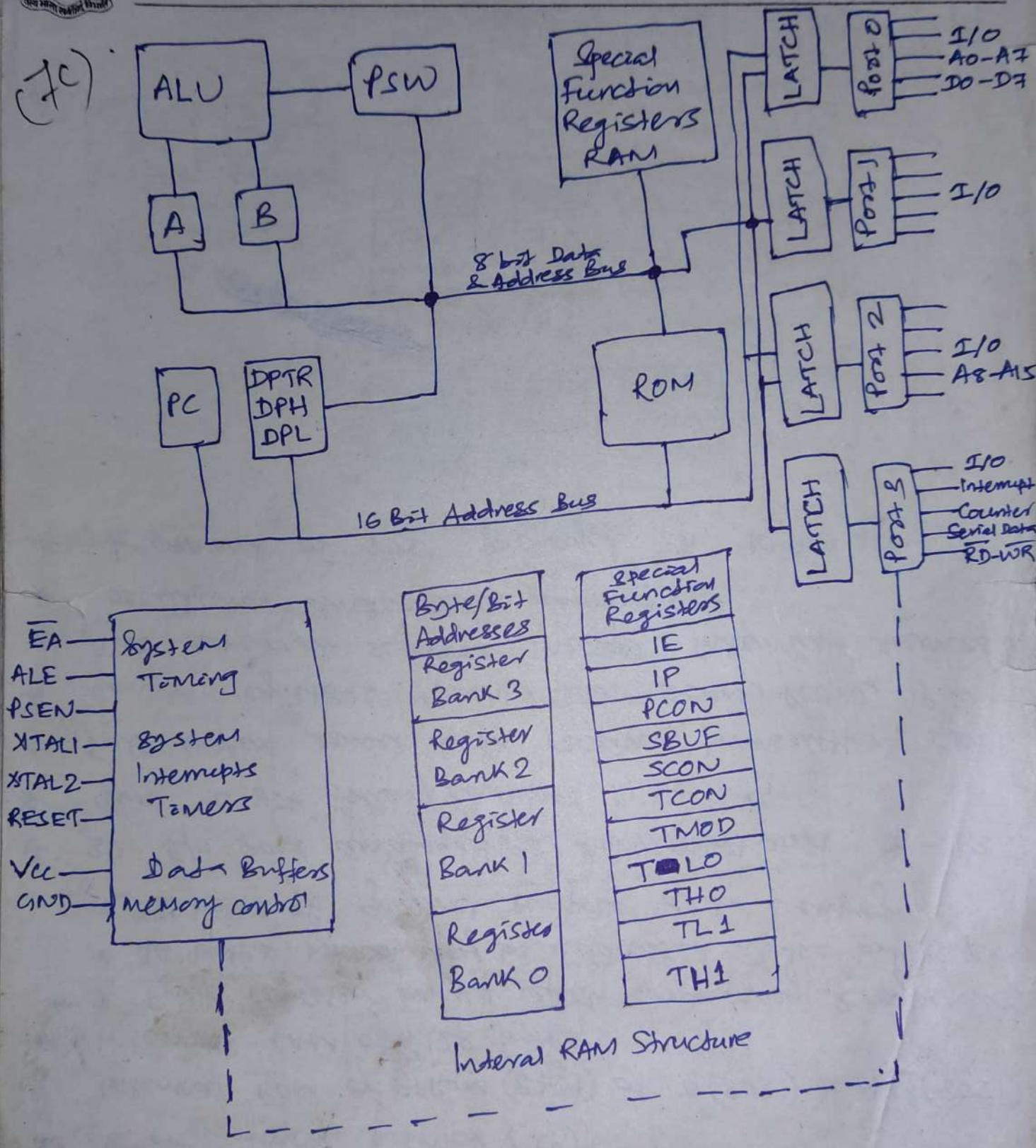
PRADEEP K.D. SAMANT
LECTURER IN ETC

MICROCONTROLLER
8051 NOTES.
4TH SEM E&TC

8051 BLOCK DIAGRAM

(7C)

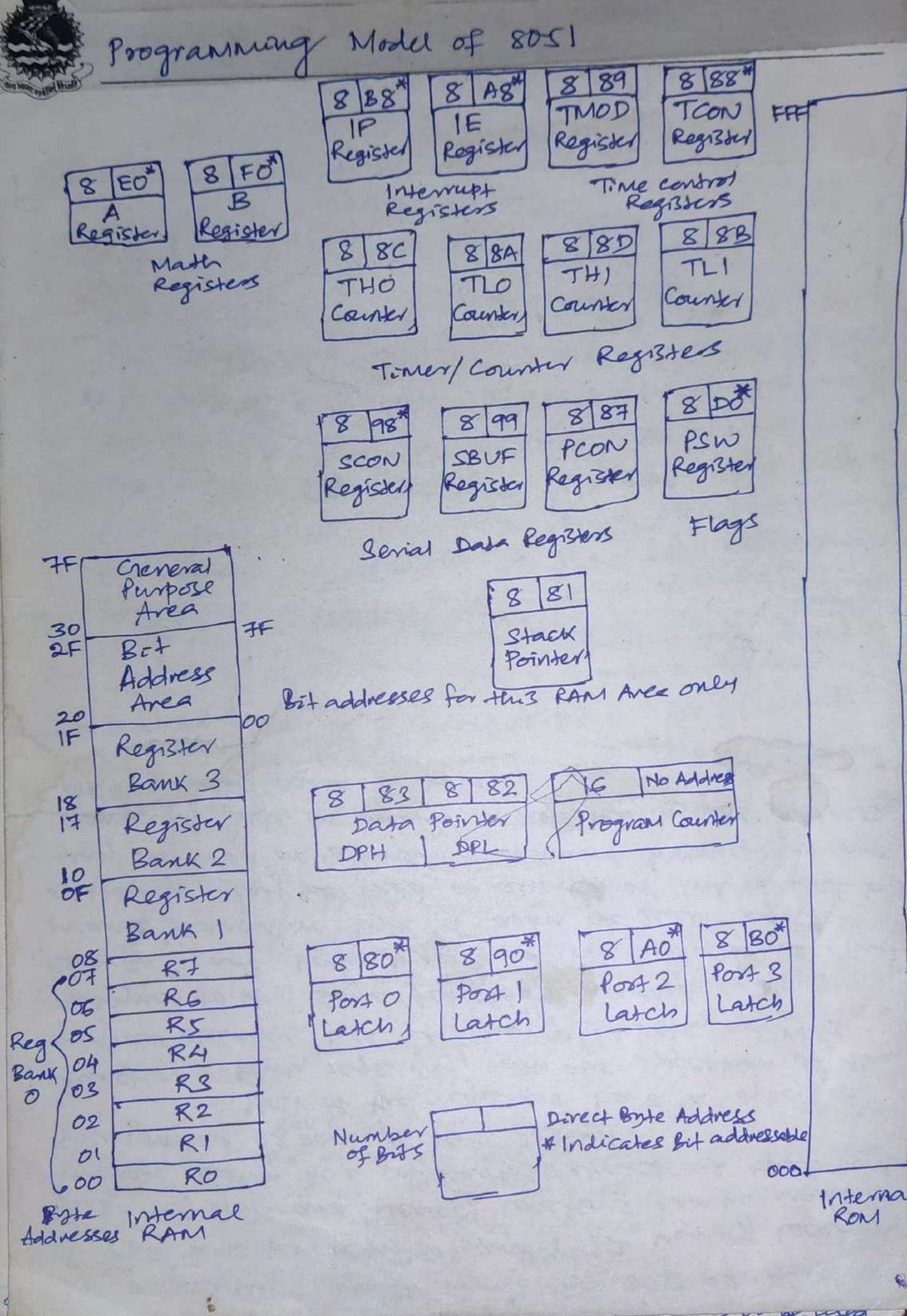


Interal RAM Structure

FEATURES OF 8051

- 8 bit CPU with registers A (the accumulator) and B.
- 16 bit program counter (PC) and data pointer (DPTR)
- 8 bit program status word (PSW)
- 8 bit Stack pointer (SP)
- Internal ROM or EPROM (8751) of 0(8031) to 4K (8051)
- Internal RAM of 128 bytes
  - Four register banks, each containing 8 registers.
  - 16 bytes, which may be addressed at the bit level.
  - 80 bytes of general purpose data memory.
- 32 I/O pins arranged as four 8 bit ports: P0 - P3.
- Two 16 bit timer/counters: T_0 & T_1
- Full duplex serial data receiver/transmitter: SBUF
- Control registers: TCON, TMOD, SCON, PCON, IP & IE
- Two external & three internal interrupt sources.
- oscillator and clock circuits.
- A pin out of 8051 packaged in 40 pin DIP.

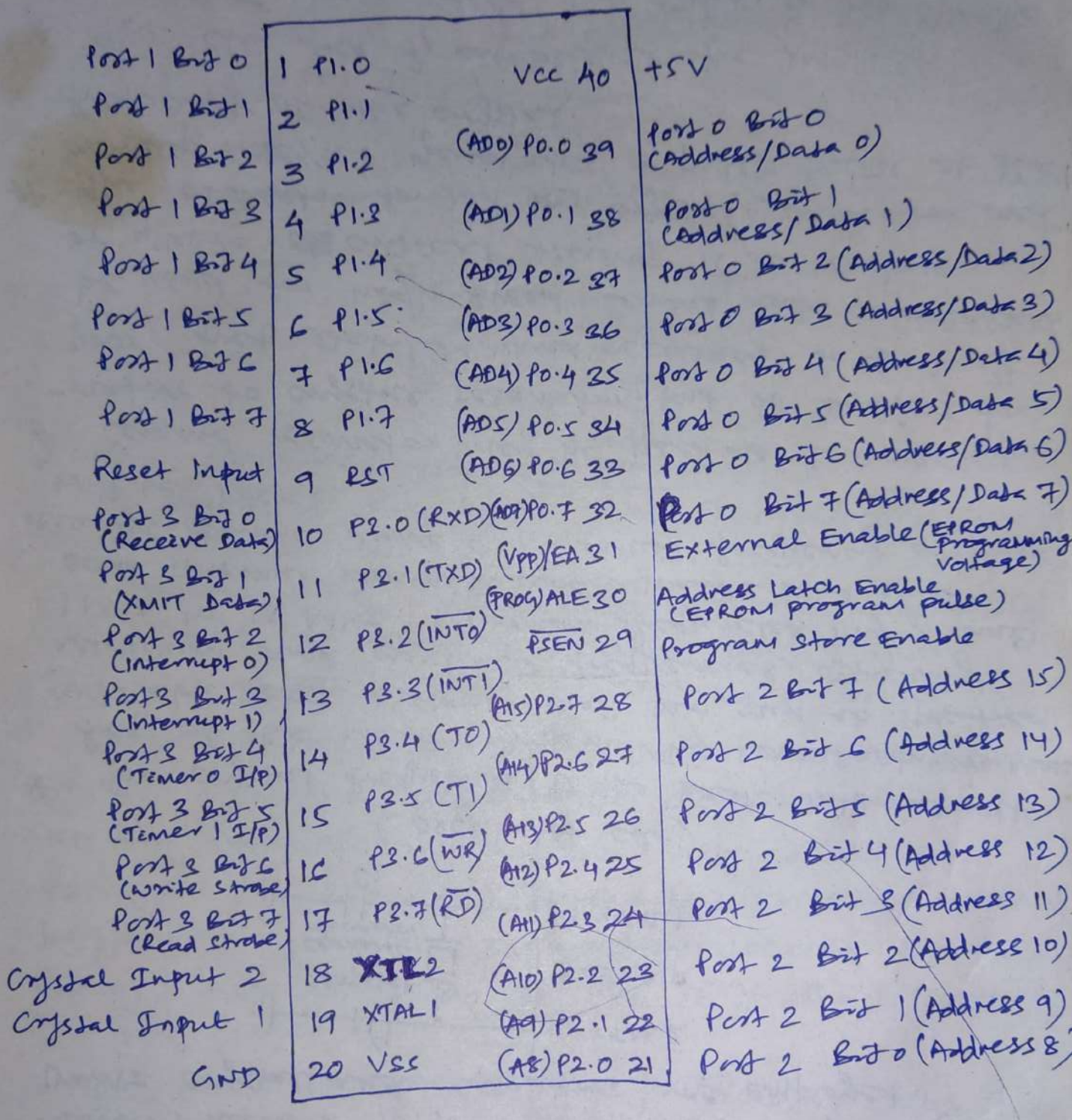
Programming Model of 8051



The programming model shows the 8051 as a collection of 8 bit & 16 bit registers and 8 bit memory locations. These registers and memory locations can be made to operate using the software instructions that are incorporated as part of the design.

Most of the registers have a specific function. Each register, with the exception of the program counter, has an internal 1-byte address assigned to it. Some registers marked with an asterisk* are both byte and bit addressable. That means the entire byte of data at such register addresses may be read or altered, or individual bits may be read or altered. Software instructions are generally able to specify a register by its address, its symbolic name or both.

8051 DIP PIN ASSIGNMENTS



Many of the pins are used for more than one function (the alternate functions are shown in parentheses). Programming instructions or physical pin connections determine the use of any multifunction pins.

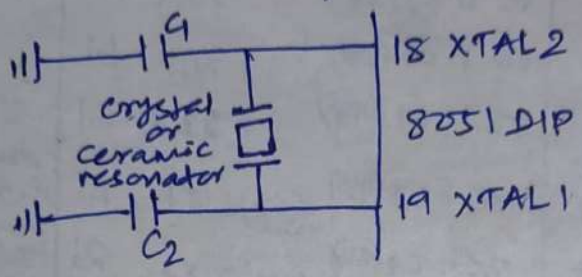
Eg a - P3.0 — General purpose I/O pin
An input (RXD) to SBUF, the serial data receiver register

The designer decides which of these two functions is to be used.

8051 Oscillator & Clock

The oscillator & clock circuitry is the heart of 8051 which generates the clk pulses for the synchronisation of all internal operations.

* Pins XTAL1 & XTAL2 are provided for connecting a resonant network to form an oscillator. Typically a quartz crystal and capacitors are employed.



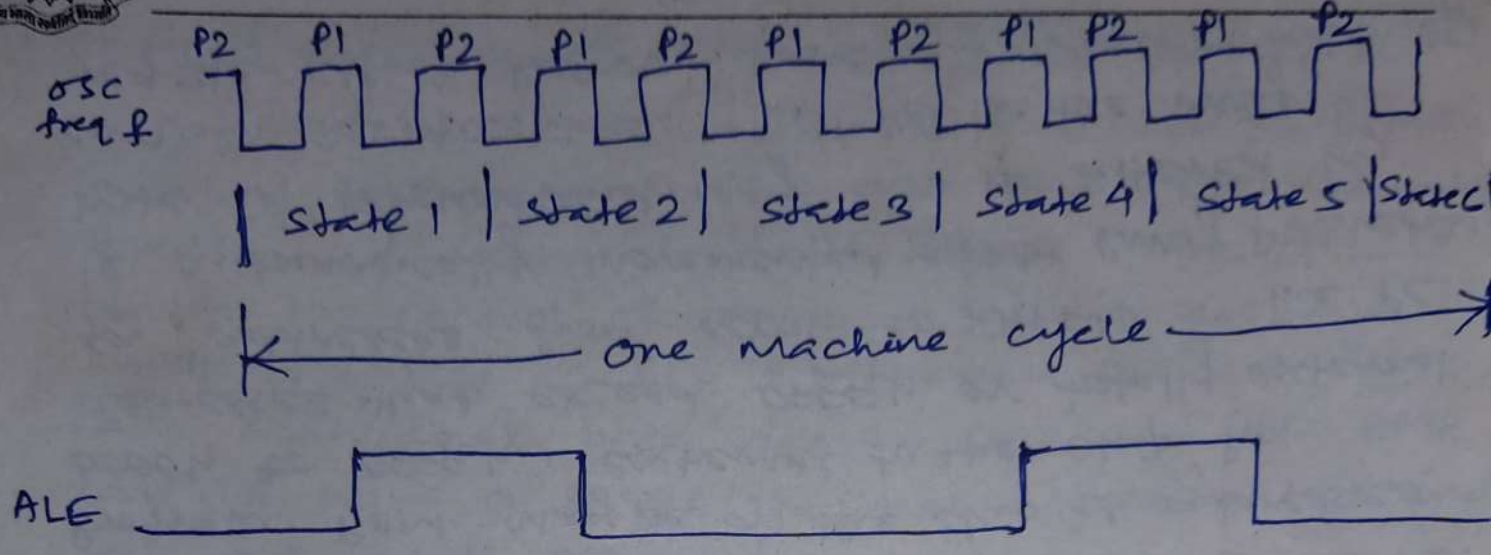
(oscillator ckt)

* The crystal frequency is the basic internal clk freq of the microcontroller. The manufacturers make available 8051 designs that can run at specified minimum and maximum frequencies, typically 1 MHz to 16 MHz. Minimum frequencies imply that some internal memories are dynamic and must always operate above a minimum frequency or data will be lost.

* Ceramic resonator may be used as low cost alternative to crystal resonator, but it suffers from poor freq stability and accuracy. So it can't be used for high speed serial data communication or where critical timing is to be done.

* The oscillator formed by crystal, capacitor and on-chip inverter generates a pulse train at the frequency of the crystal.

* The clk freq, f, establishes the smallest interval of time called the pulse, P, time. The smallest interval of time to accomplish any simple instruction or part of a complex instruction is called machine cycle.



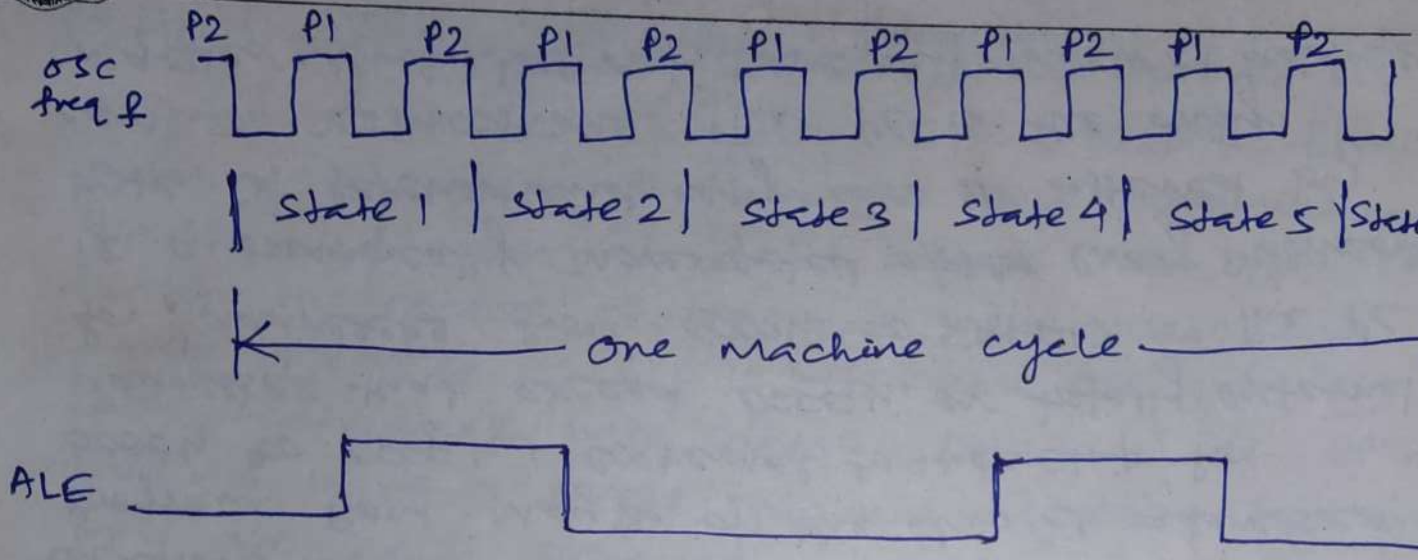
(8051 Timing)

* One machine cycle is made up (comprises) of six states. A state is the basic time interval for discrete operations of the microcontroller such as fetching an opcode byte, decoding an opcode, executing an opcode, or writing a data byte. Two oscillator pulses P_1 & P_2 define each state.

* Program instructions may require one, two or four machine cycles to be executed, depending on the type of instruction. Instructions are fetched and executed by the microcontroller automatically beginning with the instruction located at ROM memory address 0000h at the time the microcontroller is first reset.

* The time to execute any particular instruction is given by $T_{inst} = \frac{C \times 12}{\text{crystal freq.}}$

where, C = Number of cycles



(8051 Timing)

* One machine cycle is made up comprises of six states. A state is the basic time interval for discrete operations of the microcontroller such as fetching an opcode byte, decoding an opcode, executing an opcode, or writing a data byte. Two oscillator pulses P_1 & P_2 define each state.

* Program instructions may require one, two or four machine cycles to be executed, depending on the type of instruction. Instructions are fetched and executed by the microcontroller automatically beginning with the instruction located at ROM memory address 0000h at the time the microcontroller is first reset.

* The time to execute any particular instruction is given by

$$T_{inst} = \frac{C \times 12}{\text{crystal freq.}}$$

where, C = Number of cycles

* There are two ALE pulses per machine cycle. The ALE pulse which is primarily used as a timing pulse for external memory access, indicates when every instruction byte is fetched. Two bytes of a single instruction may thus be fetched and executed in one machine cycle. Single byte instructions are not executed in a half cycle, however single byte instructions "throw away" the second byte (which is the first byte of the next instruction. The next instruction is then fetched in the following cycle.

* An 11.0592 MHz crystal yields a cycle freq of 921.6 KHz which then divided by internal counter to yield standard communication baud (bps) rates of 19200, 9600, 4800, 2400, 1200 & 300 Hz.

Program Counter & Data Pointer :-

8051 contains two 16 bit registers : PC & DPTR. Each is used to hold the address of a byte in Memory.

* Program instruction bytes are fetched from locations in memory that are addressed by the PC. Program Rom may be on the chip at addresses 0000h to 0FFFh, external to the chip for addresses that exceed 0FFFh or totally external for all addresses from 0000h to FFFFh. The PC is automatically incremented after every instruction byte is fetched and may also be altered by certain instructions. The PC is the only register that does not have an internal address.

* The DPTR register is made up of two 8 bit register : DPH & DPL, which are used to furnish memory addresses for internal and external code access and external data access. The DPTR is under the control of program instructions and can be specified by its 16 bit name DPTR or by each individual byte name, DPH and DPL. DPTR does not have a single internal address; DPH and DPL are each assigned an address.

A and B CPU Registers -

The 8051 contains 34 general purpose or working registers. Two of these , registers A and B hold results of many instructions, particularly math and logical operations of the CPU. The other 32 are arranged as part of internal RAM in four banks, B0-B3, of eight registers and comprise the mathematical core.

* The register A (Accumulator) is most versatile and is used for many operations such as addition, subtraction, integer multiplication and division, and Boolean bit manipulations. The register A is also used for all data transfers between the 8051 and any external memory.

* The register B is used with the register A for multiplication and division operations and has no other function other than as a location where data may be stored.

Flags & Program status word (PSW)

* Flags are 1-bit registers provided to store the results of certain program instructions. Other instructions can test the condition of the flags and make decisions based on the flag states.

* The 8051 has four math flags that respond automatically to the outcomes of math operations and three general purpose user flags that can be set to 1 or cleared to 0 by the programmer as desired. The math flags include —

  ① Carry (C)
  ② Auxilliary carry (AC)
  ③ Overflow (OV)
  ④ Parity (P)

  The user flags include —

  ① F0
  ② GF0
  ③ GF1

  The user flags may be used by the programmer to record some event in the program.

* All of the flags can be set and cleared by the programmer at will. However the math flags are also affected by math operations.

* The flags are grouped inside the PSW and the power control (PCON) registers.

* The PSW contains the math flags, user program flag F0 and the register select bits that identify which of the four general purpose register banks is currently in use by the program.

* The remaining two user flags GF0 & GF1 are stored on PCON register.

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	—	P

(PSW Register)

Bit	Symbol	Function
PSW 7	CY	Carry flag; used in arithmetic, jump, rotate & boolean instructions.
PSW 6	AC	Auxiliary carry flag; used for BCD arithmetic.
PSW 5	F0	User flag 0
PSW 4	RS1	Register bank select bit 1
PSW 3	RS0	Register bank select bit 0
PSW 2	OV	overflow flag; used in arithmetic operation.
PSW 1	—	Reserved for future use
PSW 0	P	Parity flag; shows parity of register A. 1 – odd parity 0 – even parity.

RS1	RS0	Select register bank
0	0	0
0	1	1
1	0	2
1	1	3

Internal Memory — Internal RAM / Internal ROM

A functioning Computer must have memory for program code bytes, commonly on ROM and RAM for variable data that can be altered as the program runs.

Internal RAM –

The 128 byte internal RAM is organised into three distinct areas:

(i) 32 bytes from address 00h to 1Fh that make up 32 working registers organised as four banks of of 8 registers each. The four register banks are numbered 0 to 3 and are made up of eight registers named R0 to R7. Each register can be addressed by name (when its bank is selected) or by its RAM address (00h to 1Fh). Thus R0 of Bank 2 is R0 (if Bank 2 is currently selected) or address 10h (whether Bank 2 is selected or not). Bits RS0 and RS1 in the PSW register determine which register bank is currently in use by the program.

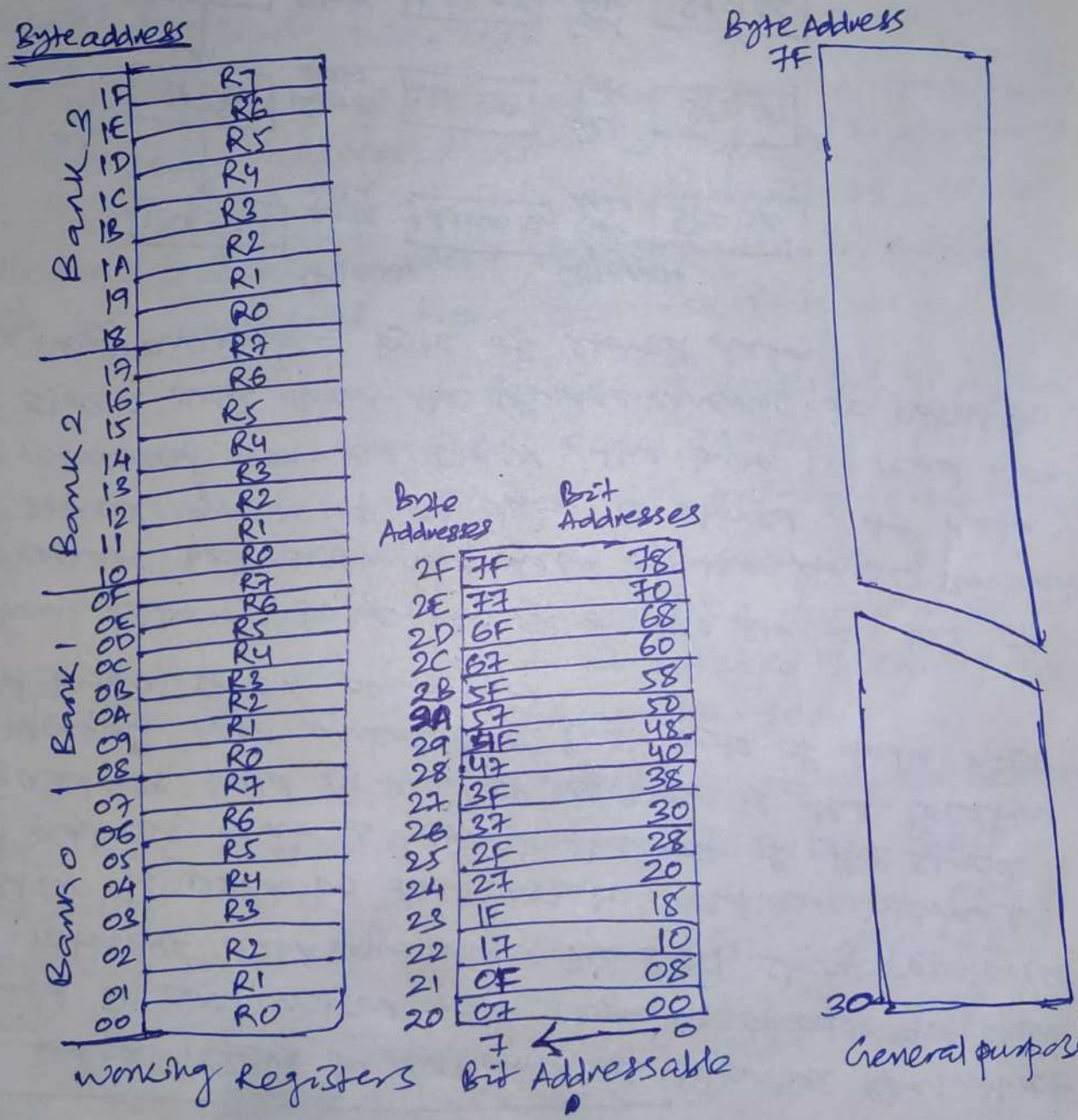
RS1	RS0	
0	0	Select B0
0	1	Select B1
1	0	Select B2
1	1	Select B3

Register banks not selected can be used as general purpose RAM. Bank 0 is selected on reset.

(2) A bit addressable area of 16 bytes occupies RAM byte addresses 20h to 2Fh, forming a total of 128 addressable bits. An addressable bit may be

Specified by its bit address of 00h to 7Fh or 8 bits may form any byte address from 20h to 2Fh Exa - bit address 4Fh is also bit 7 of byte address 29h. Addressable bits are useful when the program need only remember a binary event (switch on, light off etc.)

③ A general purpose RAM of 80 bytes occupies RAM byte addresses 30h to 7Fh

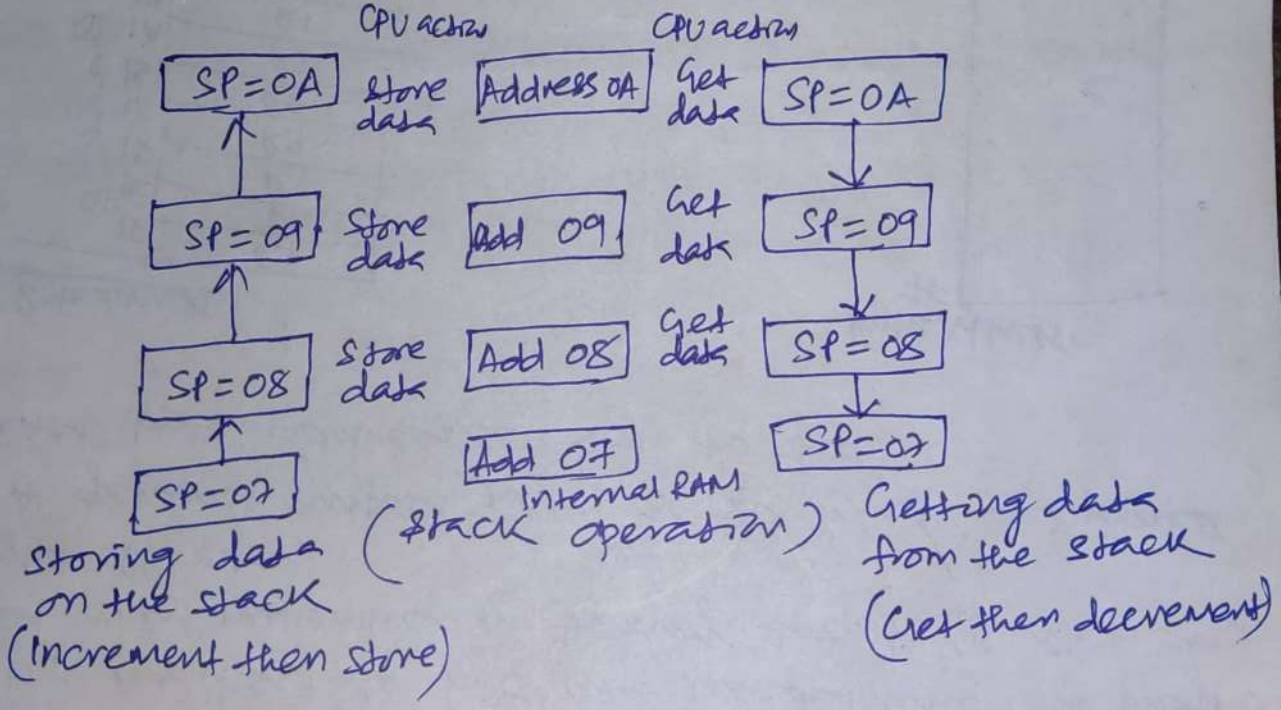


(Internal RAM organization)

The stack & the stack pointer -

The stack refers to an area of internal RAM that is used in conjunction with certain opcodes to store and retrieve data quickly. The 8 bit stack pointer (SP) register is used by the 8051 to hold an internal RAM address that is called the top of the stack. The address held in the SP register is the location in internal RAM where the last byte of data was stored by a stack operation.

* When data is to be placed on the stack, the SP increments before storing data on the stack so that the stack grows up as data is stored. As data is retrieved from the stack, the byte is read from the stack and then the SP decrements to point to the next available byte of stored data.



* The SP is set to 07h when the 8051 is reset and can be changed to any internal RAM address by the programmer using data move command.

* The stack is limited in height to the size of the internal RAM. The stack has the potential (if the programmer is not careful to limit its growth) to over write valuable data on the register banks, bit addressable RAM and scratch pad RAM areas. The programmer is responsible for making sure the stack does not grow beyond predefined bounds.

Special function Registers –

The 8051 operations that do not use the internal 128 byte RAM addresses from 00h to 7Fh are done by a group of specific internal registers, each called a special function register (SFR), which may be addressed from 80h to FFh.

* Some SFRs like A Register, B Register, IP register, IE register, TCON register, SCON register, PSW register, Port 0 – 3 latch are not only byte addressable but also bit addressable as in the case of bit area of RAM. This feature allows the programmer to change only what needs to be altered, leaving the remaining bits in that SFR unchanged.
* PC is not part of SFR and has no internal RAM address
* SFRs are named in certain opcodes by their functional names such as A or TH0 and are referenced by other opcodes by their addresses, such as 0E0h or 8Ch.
* Some SFRs have internal RAM address starting with 0. Exa – [redacted] address E0h for SFR A begins with 0. Failure to use this number convention will

result in an assembler error when the program is assembled. The SFR names and equivalent internal RAM addresses are given in the following list —

Name	Function	Internal RAM address (Hex)
A	Accumulator	OEO
B	Arithmetic	OFO
DPH	Addressing external memory	83
DPL	"	82
IE	Interrupt Enable Control	OA8
IP	Interrupt priority	OB8
PO	Input/output port Latch	80
P1	"	90
P2	"	A0
P3	"	OBO
PCON	Power control	87
PSW	Program Status Word	ODO
SCON	Serial port control	98
SBUF	Serial port data buffer	99
SP	Stack Pointer	81
TMOD	Timer/counter mode control	89
TCON	Timer/Counter Control	88
TLO	Timer 0 low byte	8A
THO	Timer 0 high byte	8B
TLI	Timer 1 low byte	8C
THI	Timer 1 high byte	8D

Internal ROM —

The 8051 is organised so that data memory and program code memory can be in two entirely different physical memory entities. Each has the same address ranges. A corresponding block of internal program code, contained on an internal ROM, occupies code address space 0000h to 0FFFh. The PC is ordinarily used to address program code bytes from addresses 0000h to FFFFh. Program addresses higher than 0FFFh which exceed the internal ROM capacity, will cause the 8051 to automatically fetch code bytes from external program memory. code bytes can also be fetched exclusively from an external memory, addresses 0000h to FFFFh, by connecting the external access pin (EA pin 31 on the DIP) to ground.

* The PC does not care where the code is; the circuit designer decides whether the code is found totally in internal ROM, totally in external ROM, or on a combination of internal and external ROM.

Input/Output Pins, Ports and circuits —

32 input/output pins are arranged as four 8 bit ports: P0, P1, P2 & P3.

P0.0 (Add/Data0)	P1.0	P2.0 (A8)	P3.0 (RXD)
P0.1 (Add/Data1)	P1.1	P2.1 (A9)	P3.1 (TXD)
P0.2 (Add/Data2)	P1.2	P2.2 (A10)	P3.2 ($\overline{INT0}$)
P0.3 (Add/Data3)	P1.3	P2.3 (A11)	P3.3 ($\overline{INT1}$)
P0.4 (Add/Data4)	P1.4	P2.4 (A12)	P3.4 (T0)
P0.5 (Add/Data5)	P1.5	P2.5 (A13)	P3.5 (T1)
P0.6 (Add/Data6)	P1.6	P2.6 (A14)	P3.6 ($\overline{WR}$)
P0.7 (Add/Data7)	P1.7	P2.7 (A15)	P3.7 ($\overline{RD}$)

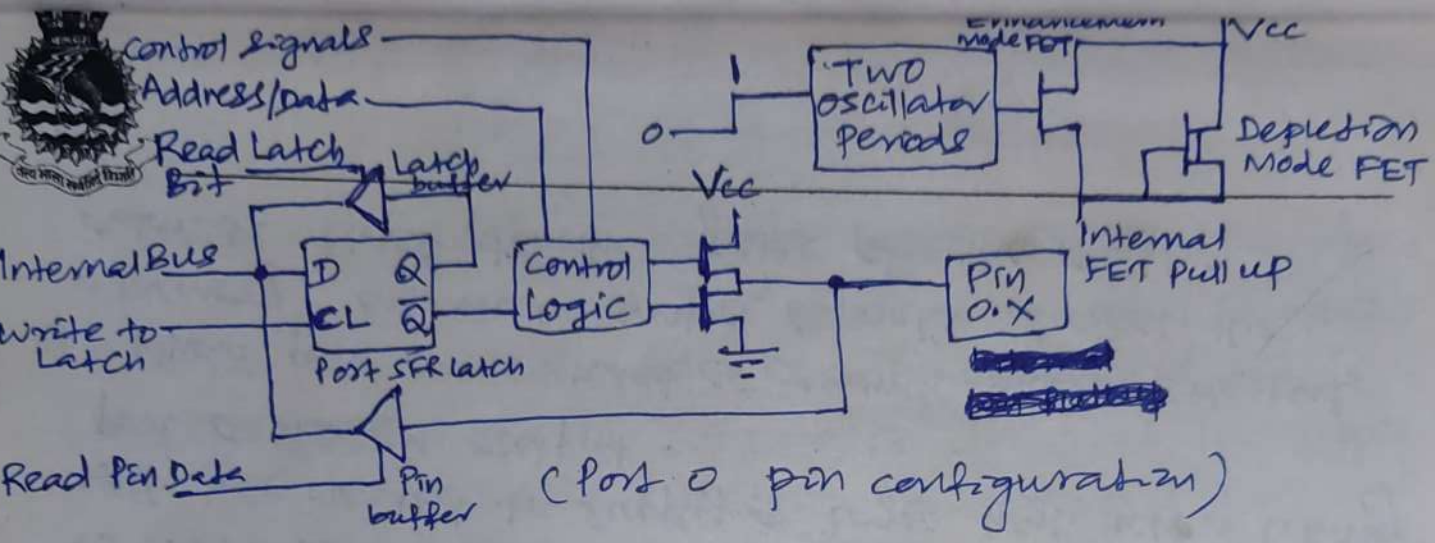
Alternate functions of port pins are shown in parentheses.

Each port has a D-type output latch for each pin. The SFR for each port is made up of these eight latches, which can be addressed at the SFR address for that port. Exa - the eight latches for port 0 are addressed at location 80h ; port 0 pin 3 is bit 2 of P0 SFR. The port latches should not be confused with the port pins ; the data on the latches does not have to be the same as that on the pins.

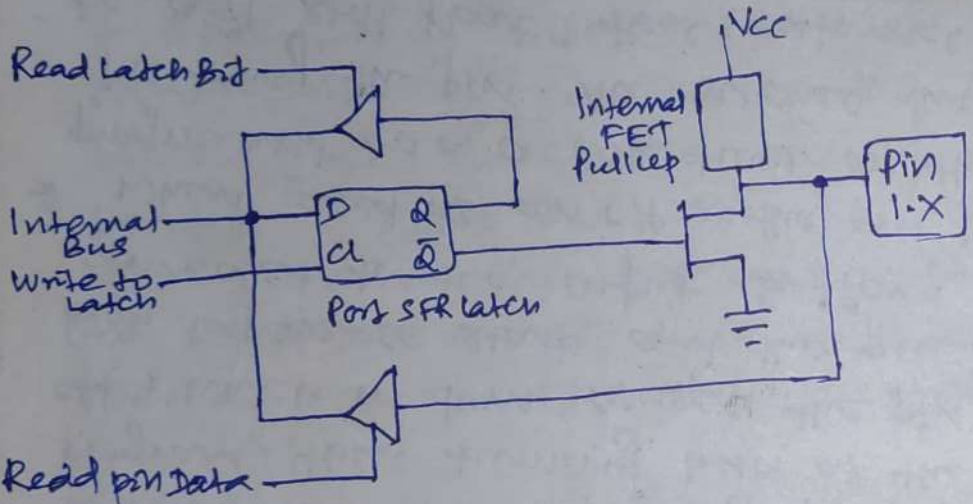
* The two data paths are shown by the [redacted] circuits that read the latch or pin data using two entirely The upper buffer is enabled when latch data is read and the lower buffer is enabled when the pin state is read. The status of each latch may be read from a latch buffer, while an input buffer is connected directly to each pin so that the pin status may be read independently of the latch state.

* Different opcodes access the latch or pin states as appropriate. Port operations are determined by the manner in which the 8051 is connected to external circuitry.

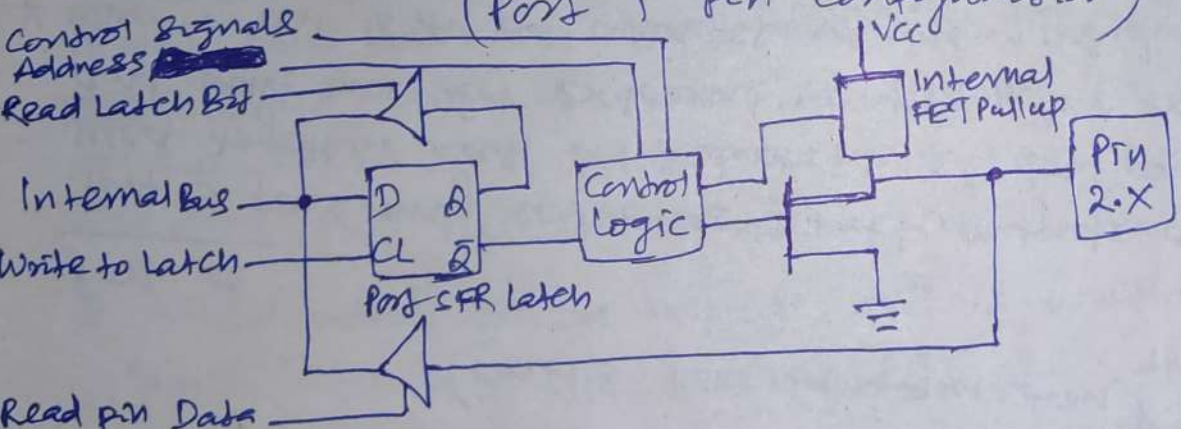
* Programmable port pins have completely different alternate functions. The configuration of the control circuitry between the output latch and the port pin determines the nature of any particular port pin function. Only port 1 can not have alternate functions ; port 0, 2 and 3 can be programmed.



(Port 0 pin configuration)

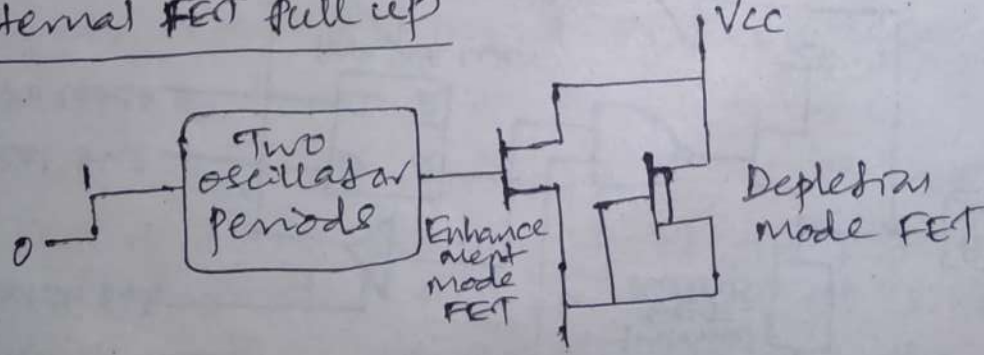


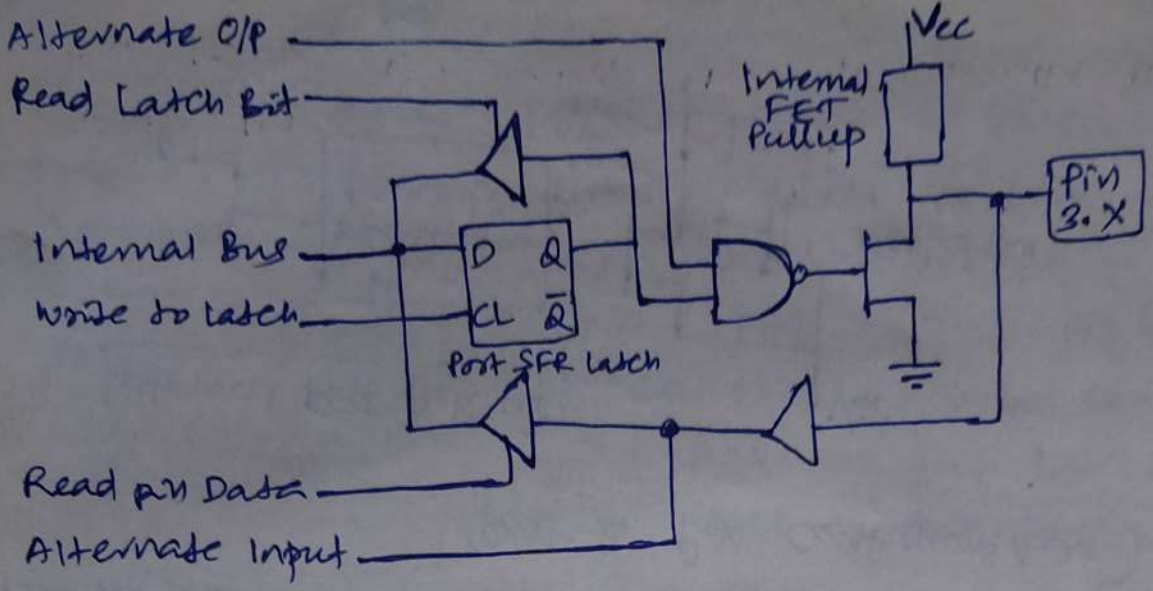
(Port 1 pin configuration)



(Port 2 pin configuration)

Internal FET Pull up





(Port 3 Pin Configuration)

Port 0

Port 0 pins may serve as inputs, outputs or when used together, as a bidirectional low order address and data bus for external memory. For example —

* When a pin is to be used as an input, a 1 must be written to the corresponding port 0 latch by the program, thus turning both of the o/p transistors, off, which in turn causes the pin to "float" in a high impedance state and the pin is essentially connected to the input buffer.
* When used as an O/P, the pin latches that are programmed to a 0 will turn on the lower FET, grounding the pin. All latches that are programmed to a 1 still float; thus external pull up resistors will be needed to supply a logic high when using port 0 as an output.
* When port 0 is used as address bus to external memory, internal control signals switch the address lines to the gates of the FETS.

When the address bit is logic 1 (high), the upper FET is turn on and the lower FET off to provide a logic high at the pin. When the address bit is logic 0 (low), the lower FET is on and the upper FET off to provide a logic low at the pin.

* After the address has been formed and latched into external circuits by the Address Latch Enable (ALE) pulse, the bus is turned around to become a data bus. Port 0 now reads data from the external memory and must be configured as an input, so a logic 1 is automatically written by internal control logic to all port 0 latches.

Port 1

Port 1 pins have no dual functions. Therefore, the output latch is connected directly to the gate of the lower FET, which has an FET circuit labeled internal FET pullup as an active pullup load.

* When used as an input, a 1 is written to the latch, turning the lower FET off; the pin and the input to the pin buffer are pulled high by the FET load. An external circuit can overcome the high impedance pull up and drive the pin low to input a 0 or leave the input high for a 1.

* When used as an output, the latches containing a 1 can drive the input of an external circuit high through the pull up. If a 0 is written to the latch, the lower FET is on, the pullup is off

and the pin can drive the input of the external circuit low.

* To aid in speeding up switching times when the pin is used as an output, the internal FET pullup has another FET in parallel with it. The second FET is turned on for two oscillator time periods during a low-to-high transition on the pin. The arrangement provides a low impedance path to +ve voltage supply to help reduce rise times in charging any parasitic capacitances in the external circuitry.

Port 2 -

Port 2 may be used as an input/output port similar in operation to port 1. The alternate use of port 2 is to supply a high order address byte in conjunction with the port 0 low order byte to address external memory.

* Port 2 pins are momentarily changed by the address control signals when supplying the high byte of a 16 bit address. Port 2 latches remain stable when external memory is addressed, as they do not have to be turned around (set to 1) for data input as is the case for port 0.

Port 3 -

Port 3 is used as an input/output port similar in operation to port 1. The input and output functions can be programmed under the control of the P3 latches or under the control of various special function registers. The alternate uses of port 3 are —

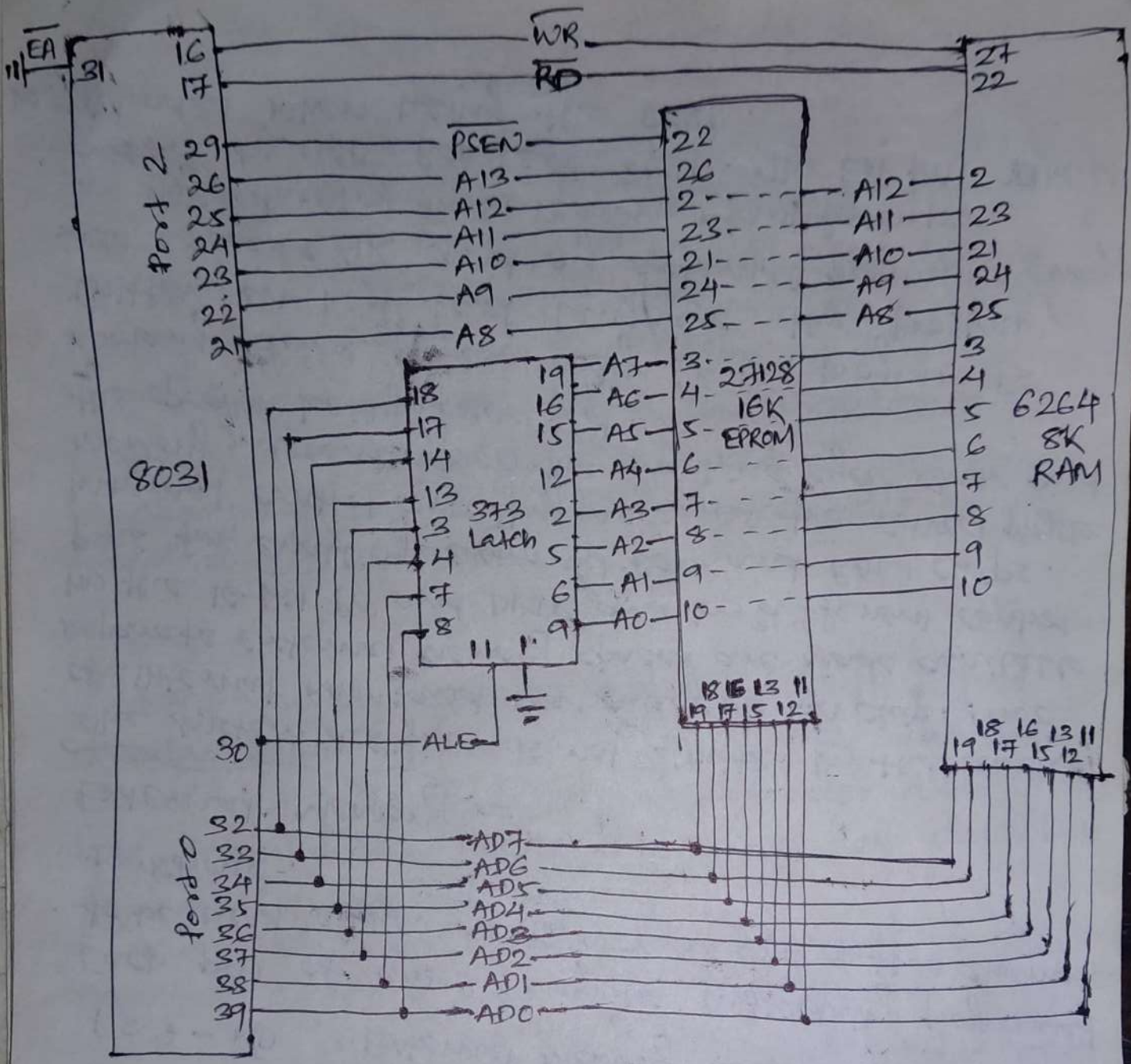
PIN	Alternate use	SFR
P3.0 – RXD	Serial data input	SBUF
P3.1 – TXD	Serial data output	SBUF
P3.2 – $\overline{INT0}$	External Interrupt 0	TCON.1
P3.3 – $\overline{INT1}$	External Interrupt 1	TCON.3
P3.4 – T0	External timer 0 input	TMOD
P3.5 – T1	External timer 1 input	TMOD
P3.6 – $\overline{WR}$	External memory write pulse	—
P3.7 – $\overline{RD}$	External Memory Read pulse	—

Each pin of port 3 may be individually programmed to be used either as I/O or as one of the alternate functions.

External Memory –

The system designer is not limited by the amount of internal RAM and ROM available on chip. Two separate external memory spaces are made available by the 16-bit PC and DPTR and by different control pins for enabling external ROM and RAM chips. Internal control circuitry accesses the correct physical memory, depending on the machine cycle state and the opcode being executed.

* Many times, particularly if the program is written on high level language, the program size exceeds 4K and an external program memory is needed. For this reason manufacturer prefers to use Romless 8031. The $\overline{EA}$ pin must be grounded when using the 8031.



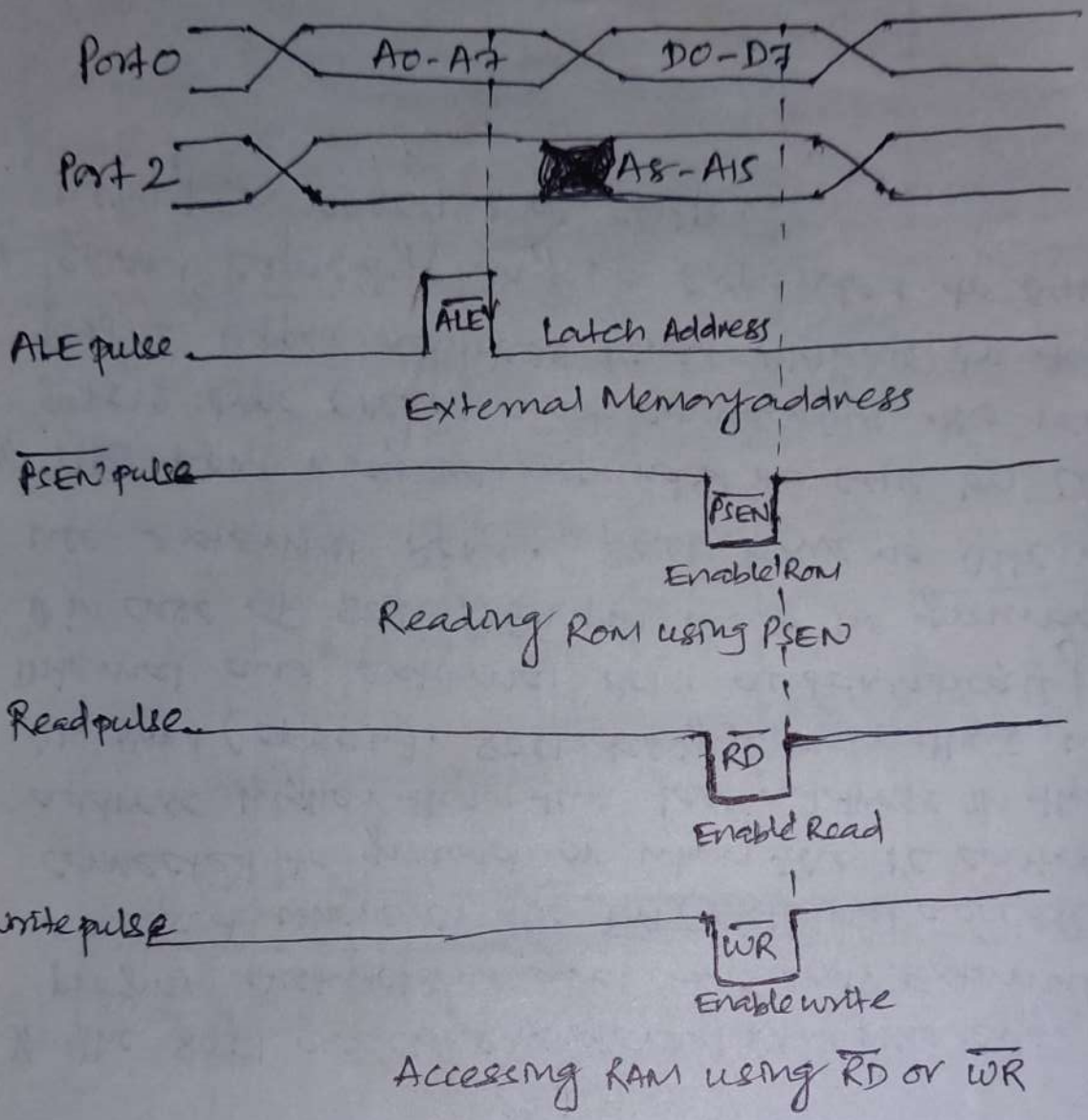
Dash lines (- - - -) show connections from EPROM pins to RAM pins. Vcc & Gnd pin connections are not shown.

[External memory connections of 8031 with 16K of EPROM and 8K RAM]

All program code is contained in an external EPROM that may be as large as 64K and that can be programmed using standard EPROM programmers.

* External RAM which is accessed by the DPTR, may also be needed when 128 bytes of internal data storage is not sufficient. External RAM upto 64K may be added to any chip of 8051 family.
* The fig shows the connection between an 8031 and an external memory configuration consisting of 16K EPROM (27128) and 8K of static RAM (6264).
* The 8051 accesses external RAM whenever certain program instructions are executed. External ROM is accessed whenever the $\overline{EA}$ (External access) pin is connected to ground or when the PC contains an address higher than the last address in the internal 4K ROM (OFFFh). 8051 designs can thus use internal and external ROM automatically whereas in case of 8031, $\overline{EA}$ pin must be grounded to use external ROM. 8031 have no internal ROM.
* The ROM may be expanded to 64K by using a 27512 type EPROM and connecting the remaining port 2 upper address lines A14-A15 to the chip.
* SRAM capacity may be expanded to 64K by using a 62864 type chip.

External Memory Timing —



The fig shows the timing associated with an external memory access cycle. During any memory access cycle, port 0 is time multiplexed. It first provides the lower byte of the 16 bit memory address (A0-A7) and then acts as a bidirectional data bus (D0-D7) to write or read a byte of memory data.

* Port 2 provides the high byte of the 16 bit memory address (A8-A15) during the entire memory read/write cycle.

* The lower address byte from port 0 must be latched into an external register to save the byte. Address byte save is accomplished by the ALE clock pulse that provides the correct timing for the 373 type data latch. The port 0 pins then become free to serve as a data bus.
* If the memory access is for a byte of program code in the ROM, the $\overline{PSEN}$ (Program Store enable) pin will go low to enable the ROM to place a byte of program code on the data bus. If the access is for a RAM byte, the $\overline{WR}$ (write) or $\overline{RD}$ (read) pins will go low, enabling data to flow between the RAM and the data bus.

* $\overline{WR}$ and $\overline{RD}$ signals are alternate uses for Port 3 pins 16 and 17. port 0 is used for the lower address byte and data; port 2 is used for the upper address bits.
* The use of external memory consumes many of the port pins, leaving only port 1 and parts of port 3 for general I/O.

Counters and Timers –

Two 16 bit up counters, namely T0 & T1, are provided for the general use of the programmer. Each counter may be programmed to count internal clock pulses, 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, TL1) and high (TH0, TH1) bytes. All counter action is controlled by bit states in the timer Mode control register (TMOD), timer/counter control register (TCON) and certain program instructions.

* TMOD is dedicated solely to the two timers and can be considered to be two duplicate 4 bit registers, each of which controls the action of one of the timers.

7	6	5	4	3	2	1	0
Gate	C/$\overline{T}$	M1	M0	Gate	C/$\overline{T}$	M1	M0
[Timer 1]				[Timer 0]			

(Bit assignments for TMOD SFR)

Bit	Symbol	Function
7/3	Gate	OR gate enable bit which controls RUN/STOP of timer 1/0. Set to 1 by program to enable timer to run if bit TR1/0 in TCON is set and signal on external interrupt $\overline{INT}$1/0 pin is high. Cleared to 0 by program to enable timer to run if bit TR1/0 in TCON is set.

Bit	Symbol	Function
6/2	C/$\overline{T}$	Set to 1 by program to make timer 1/0 act as a counter by counting pulses from external input pins 3.5(T1) or 3.4(T0). cleared to 0 by program to make timer act as a timer by counting internal freq.
5/1	M1	Timer/counter operating mode select bit 1. set/cleared by program to select mode.
4/0	M0	Timer/counter operating mode select bit 0. set/cleared by program to select mode.

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

TMOD is not bit addressable.

* TCON has control bits and flags for the timers in upper nibble, and control bits and flags for the external interrupts in the lower nibble.

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	TE0	IT0

(Bit assignments for TCON SFR)

Bit	Symbol	Function
7	TF1	Timer 1 overflow flag. Set when timer rolls from all 1s to 0. Cleared when processor vectors to execute interrupt service routine located at program address 001B h.
6	TR1	Timer 1 run control bit. set to 1 by program to enable timer to count; cleared to 0 by program to halt timer. Does not reset timer.

Bit	Symbol	Function
6/2	C/T	Set to 1 by program to make timer 1/0 act as a counter by counting pulses from external input pins 3.5(T1) or 3.4(T0). Cleared to 0 by program to make timer act as a timer by counting internal freq.
5/1	M1	Timer/counter operating mode select bit 1. set/cleared by program to select mode.
4/0	M0	Timer/counter operating mode select bit 0. set/cleared by program to select mode.

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

TMOD is not bit addressable.

* TCON has control bits and flags for the timers in upper nibble, and control bits and flags for the external interrupts in the lower nibble.

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	TE0	IT0

(Bit assignments for TCON SFR)

Bit	Symbol	Function
7	TF1	Timer 1 overflow flag. Set when timer rolls from all 1s to 0. Cleared when processor vectors to execute interrupt service routine located at program address 001B h
6	TR1	Timer 1 run control bit. set to 1 by program to enable timer to count; cleared to 0 by program to halt timer. Does not reset timer.

Bit	Symbol	Function
5	TF0	Timer 0 overflow flag. Set when timer rolls from all 1s to 0. Cleared when processor vectors to execute interrupt service routine located at program address 000Bh.
4	TR0	Timer 0 run control bit. Set to 1 by program to enable timer to count; cleared to 0 by program to halt timer. Does not reset timer.
3	IE1	External interrupt 1 Edge flag. Set to 1 when a high-to-low edge signal is received on Port 3 pin 3.3 ($\overline{INT1}$). Cleared when processor vectors to execute interrupt service routine located at program address 0013h. Not related to timer operations.
2	IT1	External interrupt 1 Signal type control bit. Set to 1 by program to enable external interrupt 1 to be triggered by a falling edge signal. Set to 0 by program to a enable a low-level signal on external interrupt 1 to generate an interrupt.
1	IE0	External interrupt 0 Edge flag. Set to 1 when a high-to-low edge signal is received on port 3 pin 3.2 ($\overline{INT0}$). Cleared when processor vectors to execute interrupt service routine located at program address 0003h. Not related to timer operations.
0	IT0	External interrupt 0 signal type control bit. Set to 1 by program to enable external interrupt 0 to be triggered by a falling edge signal. Set to 0 by program to enable a low level signal on external interrupt 0 to generate an interrupt.

* Bit addressable as TCON.0 to TCON.7

Timer Counter Interrupts –

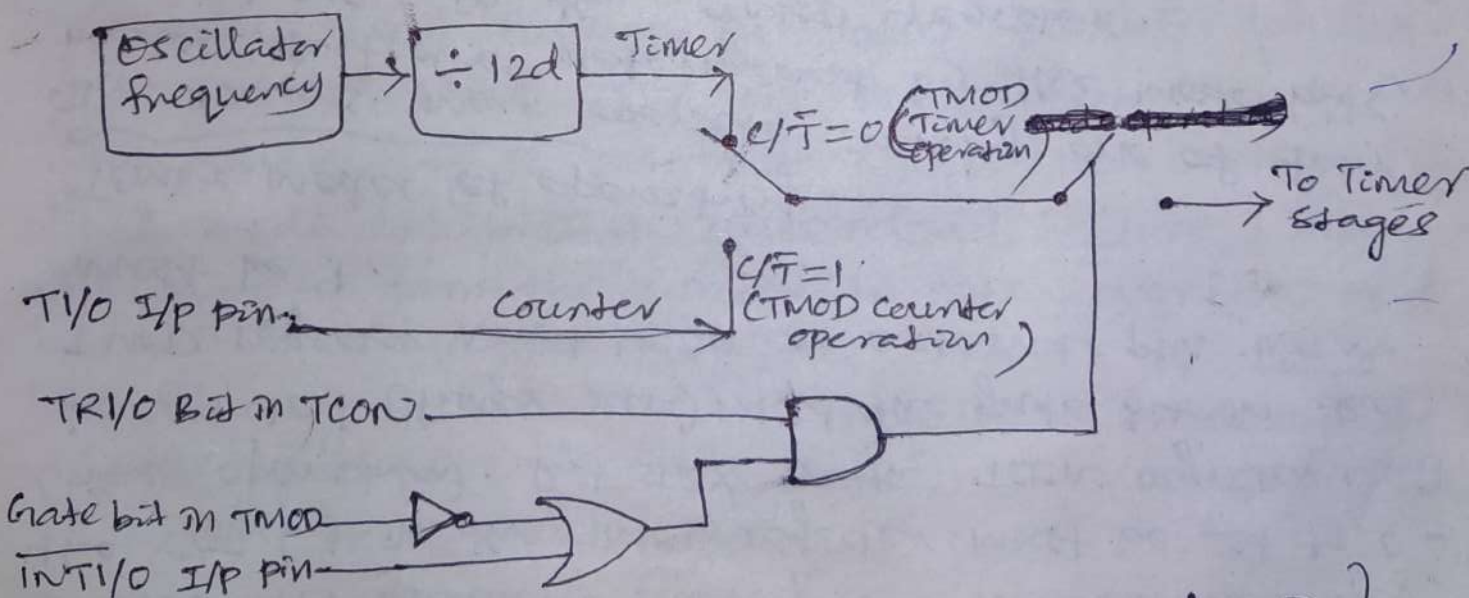
The counters have been included on the chip to relieve the processor of timing and counting chores. When the program wishes to count a certain number of internal pulses or external events, a number is placed in one of the counters. The number represents the maximum count less the desired count, plus 1. The counter increments from the initial number to the maximum and then rolls over to 0 on the final pulse and also sets a timer flag. The flag condition may be tested by an instruction to tell the program that the count has been accomplished, or the flag may be used to interrupt the program.

Timing – If a counter is programmed to be a timer, it will count the internal clock frequency of the 8051 oscillator divided by 12d. For example

If crystal freq = 6 MHz

$$\text{Timer clock freq} = \frac{6 \times 10^6}{12} = 500\ \text{KHz}.$$

The resultant timer clock is gated to the timer by means of the circuit shown below –



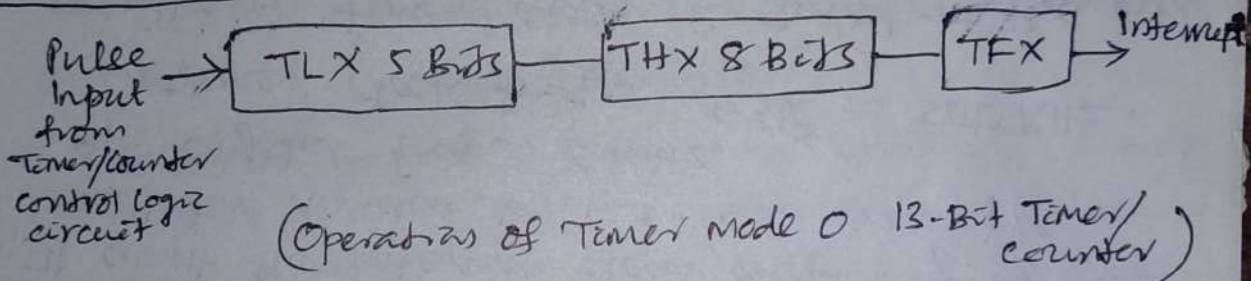
(Timer/counter control logic)

In order for oscillator clock pulses to reach the timer, the C/T bit in the TMOD register must be set to 0 - (timer operation). Bit TRX in the TCON register must be set to 1 (timer run), and the gate bit in the TMOD register must be 0, or external pin $\overline{INTX}$ must be 1.

Timer modes of operation —

The timers may operate in any one of four modes that are determined by the mode bits, M1 and M0, in the TMOD register.

① Timer Mode 0 —



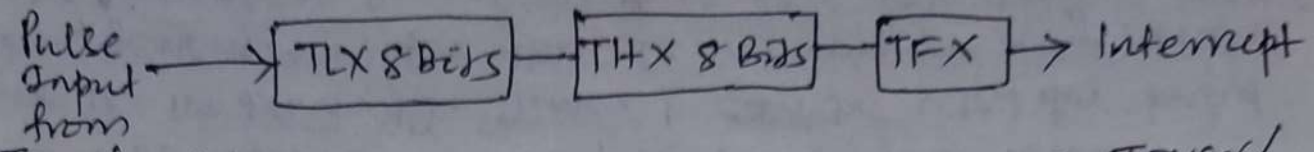
(Operation of Timer Mode 0 13-Bit Timer/counter)

Setting timer X Mode bits to 00b in the TMOD register results in using the THX register as an 8 bit counter and TLX as a 5-bit counter. Another way when M1 = 0 & M0 = 0 in the TMOD register timer mode 0 is selected.

* The pulse input is divided by 32d on TL. Hence TH counts the original oscillator frequency reduced by a total of (12 X 32) = 384d. For example 6MHz oscillator frequency would result in a final frequency to TH of 15625 Hz. The timer flag is set whenever THX goes from FFh to 00h, or in 0.0164 seconds for 6MHz crystal if THX starts at 00h.

64×10^{-5} 60.9

② Timer Mode 1 –

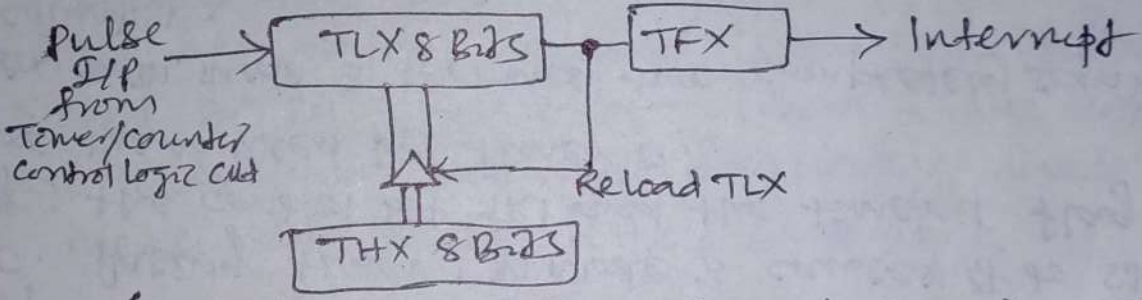


(Operation of Timer Mode 1 16-bit Timer/Counter)

Setting timer X mode bits to 01b in the TMOD register results in using TLX as an 8 bit counter and THX as an 8 bit counter. The timer flag would be set in 0.1311 seconds using a 6MHz crystal.

Timer Mode 2 –

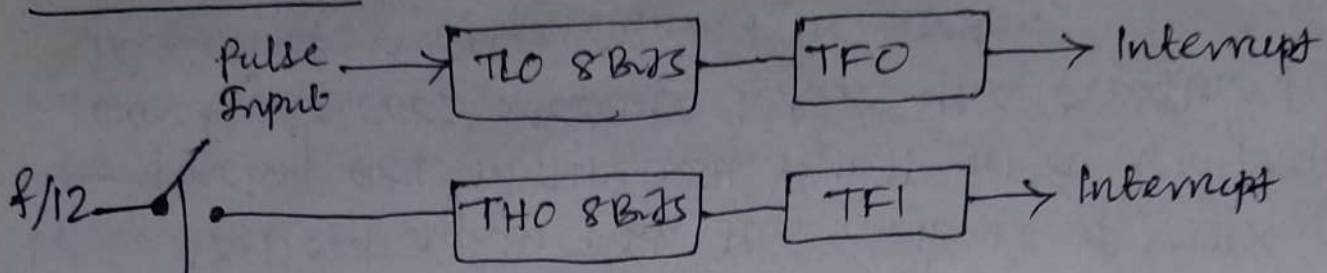
Setting the mode bits to 10b in TMOD configures the timer to use only the TLX counter as an 8 bit counter.



(Timer mode 2 Auto-Reload of TL from TH)

THX is used to hold a value that is loaded into TLX everytime TLX overflows from FFh to 00h. The timer flag is also set when TLX overflows. This mode exhibits an auto reload feature: TLX will count up from the number in THX, overflow and be initialised again with the contents of THX. Exa- placing 9Ch in THX will result in a delay of exactly 0.0002 seconds before the overflow flag is set if a 6MHz crystal is used.

Timer Mode 3



(Timer mode 3 Two 8 bit timers using Timer 0)

Timers 0 and 1 may be programmed to be in mode 0, 1 or 2 independently of a similar mode for the other timer. This is not for mode 3; the timers do not operate independently if mode 3 is chosen for timer 0. Placing timer 1 in mode 3 causes it to stop counting; the control bit TR1 and the timer 1 flag TF1 are then used by timer 0.

* Timer 0 in mode 3 becomes two completely separate 8 bit counter.
* TL0 is controlled by the gate arrangement of timer/counter control logic ckt and sets the timer flag TF0 whenever it overflows from FFh to 00h. TH0 receives the timer clock (f/12) under the control of TR1 only and sets the timer flag TF1 when it overflows.
* Timer 1 may still be used in modes 0, 1 and 2 when timer 0 is in mode 3 with one important exception: No interrupts will be generated by timer 1 while timer 0 is using the TF1 overflow flag. Switching timer 1 to mode 3 will stop it (and hold whatever count is in timer 1). Timer 1 can be used for baud rate generation for the serial port or any other mode 0, 1, or 2 function that does not depend on interrupt (or any other use of the TF1 flag) for proper operation.

Counting — The only difference between counting and timing is the source of the clock pulses to the counter. When used as a counter pin T0 (P3.4) supplies pulses to counter 0 and pin T1 (P3.5) to counter 1. The C/T bit in TMOD must be set to 1 to enable pulses from T pin to reach the control ckt shown in fig Timer/counter control logic.
The input pulse on T is sampled during P2 of state 5 every machine cycle. A change on the input from high to low between samples will increment the counter. Each high and low state of the input pulse must thus be held constant for at least one machine cycle to ensure reliable counting. Since this takes 24 pulses (one machine cycle - 12 pulses) the maximum input frequency that can be accurately counted is the oscillator freq divided by 24. For exa — when 6MHz crystal is used, maximum external frequency that can be measured is 250KHz (6M/24).

Serial Data Input/Output —

The 8051 has a serial data communication circuit that uses register SBUF to hold data, register SCON to control data communication and register PCON to control data rates. Pins RXD (P3.0) and TXD (P3.1) are connected to the serial data network to receive and send data bits respectively.

SBUF — SBUF is physically two registers. One is write only and is used to hold data to be transmitted out of the 8051 via TXD. The other is read only & is used to hold received data from external sources via RXD. Both mutually exclusive registers use address 99h.

SCON - Register SCON controls data communication. There are four programmable modes for serial data communication that are chosen by setting the SM1X bits in SCON. Baud rates are determined by the mode chosen.

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

(Serial Port control (SCON) SFR)

Bit	Symbol	Function
7	SM0	Serial port Mode bit 0. Set/cleared by program to select mode.
6	SM1	Serial port Mode bit 1. Set/cleared by program to select mode.

SM0	SM1	Mode	Description
0	0	0	Shift register; baud = f/12
0	1	1	8 bit UART; baud = variable
1	0	2	9 bit UART; baud = f/32 or f/64
1	1	3	9 bit UART; baud = variable.

5	SM2	Multiprocessor communications bit. Set/cleared by program to enable multiprocessor communications in modes 2 and 3. When set to 1 an interrupt is generated if bit 9 of the received data is a 1; no interrupt is generated if bit 9 is a 0. If set to 1 for mode 1, no interrupt will be generated unless a valid stop bit is received. Clear to 0 if mode 0 is in use.
4	REN	Receive enable bit. Set to 1 to enable reception; cleared to 0 to disable reception.
3	TB8	Transmitted bit 8. Set/cleared by program in modes 2 and 3

2	RB8	Received Bit 8. Bit 8 of received data in modes 2 and 3; stop bit in mode 1. Not used in mode 0.
1	TI	Transmit interrupt flag. Set to one at the end of bit 7 time in mode 0, and at the beginning of the stop bit for other modes. Must be cleared by the program.
0	RI	Receive interrupt flag. Set to one at the end of bit 7 time in mode 0 and halfway through the stop bit for other modes. Must be cleared by the program.

* Bit addressable as SCON.0 to SCON.7.

PCON – Register PCON controls data rates.

7	6	5	4	3	2	1	0
SMOD	–	–	–	GF1	GF0	PD	IDL

(Power mode control (PCON) SFR)

Bit	Symbol	Function
7	SMOD	Serial baud rate modify bit. Set to 1 by program to double baud rate using timer 1 for modes 1, 2 and 3. Cleared to 0 by program to use timer 1 baud rate.
6-4	–	Not Implemented.
3	GF1	General purpose user flag bit 1. Set/cleared by program.
2	GF0	General purpose user flag bit 0. Set/cleared by program.
1	PD	Power down bit. Set to 1 by program to enter power down configuration for CHMOS processors.
0	IDL	Idle mode bit. Set to 1 by program to enter idle mode configuration for CHMOS processors.

* PCON is not bit addressable.

Serial data Interrupts -

The serial data flags in SCON, TI and RI are set whenever a data byte is transmitted (TI) or received (RI). These flags are ORed together to produce an interrupt to the program. The program must read these flags to determine which caused the interrupt and then clear the flag. This is unlike the timer flags that are cleared automatically. It is the responsibility of programmer to write routines that handle the serial data flags.

* Serial data flags in SCON aid in efficient data transmission and reception.

Data transmission -

Transmission of serial data begins anytime data is written to SBUF. TI is set to 1 when the data has been transmitted and signifies that SBUF is empty (for transmission purposes) and that another data byte can be sent. If the program fails to wait for the TI flag and overwrites SBUF while a previous data byte is in the process of being transmitted, the results will be unpredictable (meaning garbage out).

Data Reception - Reception of serial data will begin

if the receive enable bit (REN) in SCON is set to 1 for all modes. In addition for Mode 0 only, RI must be cleared to 0. Receiver interrupt flag (RI) is set after data has been received in all modes setting REN is the only direct program control that limits the reception of unexpected data; the requirement that RI also be 0 for Mode 0 prevents the reception of new data until the program has

dealt with the old data and reset RI.

* Reception can begin in modes 1, 2 and 3 if RI is set when the serial stream of bits begins. RI must have been reset by the program before the last bit is received or the incoming data will be lost.
* Incoming data is not transferred to SBUF until the last bit data has been received so that the previous transmission can be read from SBUF while new data is being received.

Serial Data Transmission Modes —

There are four programmable modes for serial data communication that are chosen by setting the SMX bits in SCON. ~~Baud rates are fixed for mode 0 and variable using timer.~~

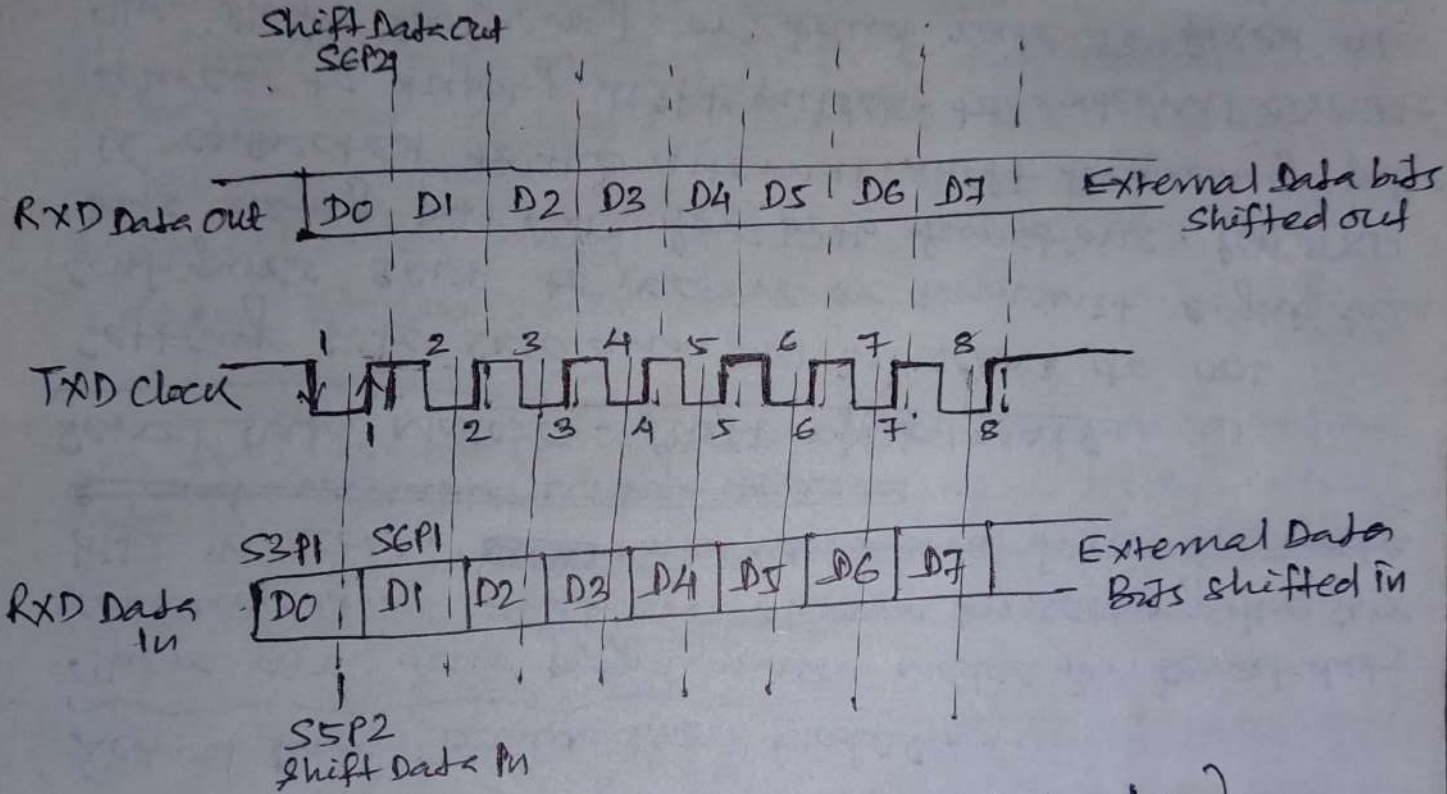
Serial Data Mode 0 - Shift register mode

Setting bits SM0 and SM1 in SCON to 00b configures SBUF to receive or transmit eight data bits using pin RXD for both functions. Pin TXD is connected to the internal shift frequency pulse source to supply shift pulses to external circuits. The shift frequency or baud rate is fixed at 1/12 of the oscillator frequency, the same rate used by the timers when in the timer configuration.

* Mode 0 is intended not for data communication between computers, but as a high speed serial data collection method using discrete logic to achieve high data rates.

* The baud rate used in mode 0 will be much higher than standard for any reasonable oscillator freq for exa - For a 6MHz crystal, the shift rate will be 500KHz.

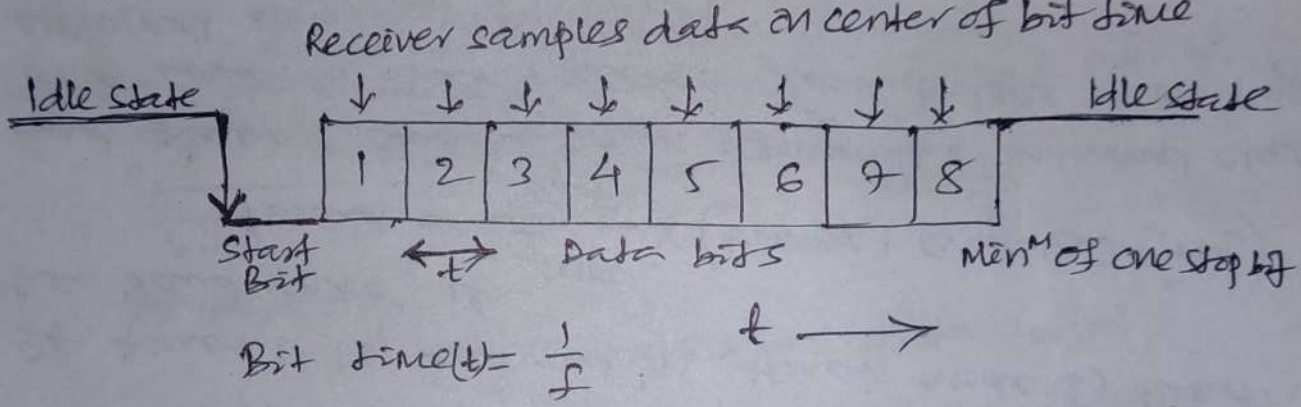
* The TXD shift clock is a square wave that is low for machine cycle states S3-S4-S5 and high for S6-S1-S2.



(Timing for Mode 0 shift register Data txn)

Serial Data Mode 1 – Standard UART –

When SM0 and SM1 in SCON are set to 01b, SBUF becomes a 10 bit full duplex receiver/transmitter that may receive and transmit data at the same time. Pin RXD receives all data and pin TXD transmits all data.



(Standard UART Data word)

* Transmitted data is sent as a start bit, eight data bits (LSB first) and a stop bit. Interrupt flag TI is set once all ten bits have been sent. Each bit time/interval is the inverse of the baud rate frequency and each bit is maintained high or low over that interval.
* Received data is obtained in the same order; reception is triggered by the falling edge of the start bit and continues if the stop bit is true (0 level) halfway through the start bit interval. This is an anti-noise measure; if the reception ckt is triggered by noise on the transmission line, the check for a low after half a bit interval should limit false data reception.

~~Mode 1 Baud rates~~

~~$f_{baud} = \frac{2}{}$~~

During reception the start bit is discarded, the eight data bits go to SBUF and the stop bit is saved in bit RB8 of SCON. RI is set to 1, indicating a new data byte has been received.

Mode 1 Baud rates — Timer 1 is used to generate the baud rate for mode 1 by using the overflow flag of the timer to determine the baud frequency. Typically timer 1 is used in timer mode 2 as an auto load 8 bit timer that generates the baud frequency:

$$f_{baud} = \frac{2^{SMOD}}{32d} \times \frac{\text{oscillator freq}}{12d \times [256d - (TH1)]}$$

where, SMOD is the control bit in PCON and can be 0 or 1, which doubles the baud rate.

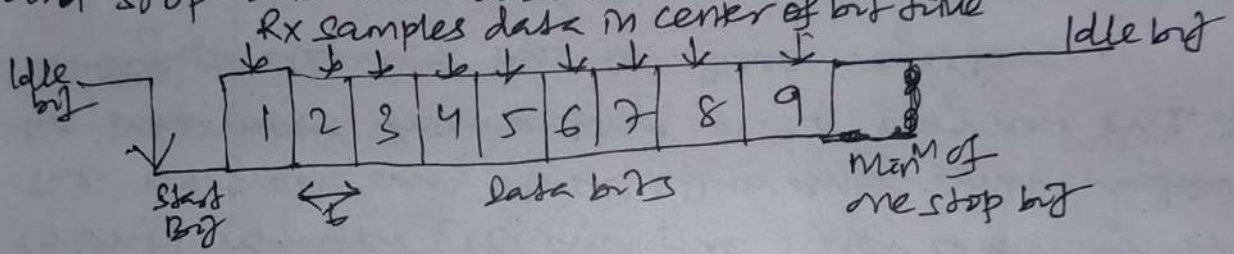
If timer 1 is not run in timer mode 2, then the baud rate is

$$f_{baud} = \frac{2^{SMOD}}{32d} \times (\text{timer 1 overflow freq})$$

and timer 1 can be run using the internal clock or a counter that receives clock pulses from any external source via pin T1.

Serial Data Mode 2 – Multiprocessor Mode

Mode 2 is similar to mode 1 except 11 bits are transmitted; a start bit, nine data bits and a stop bit. The 9th data bit is copied from bit TB8 in SCON during transmit and stored in bit RB8 of SCON when data is received. Both the start and stop bits are discarded.



Bit time $t = \frac{1}{f}$ (Multiprocessor data word)

The baud rate is programmed as follows –

$$f_{baud2} = \frac{2^{SMOD}}{64d} \times \text{Oscillator freq}$$

As in the case for mode 0, the baud rate is much higher than standard communication rates. The high data rate is needed in many multiprocessor applications. The conditions for setting RI for mode 2 is similar to mode 1. RI must be 0 before the last bit received and ninth bit must be a 1.

Serial Data Mode 3

Mode 3 is identical to mode 2 except that the baud rate is determined exactly as in mode 1, using timer 1 to generate communication frequencies.

Interrupts —

Interrupts may be generated by internal chip operation or provided by external sources. Any interrupt can cause the 8051 to perform a hardware call to an interrupt handling subroutine that is located at a predetermined (by the 8051 designers) absolute address in program memory. Interrupt force the program to call a subroutine.

* Five interrupts are provided in the 8051. Three of these are generated automatically by internal operations: Timer flag 0, Timer flag 1 and the serial port interrupt (RI or TI). Two interrupts are triggered by external signals provided by circuitry that is connected to pins $\overline{\text{INT0}}$ and $\overline{\text{INT1}}$ (port pins 3.2 and 3.3).

* All interrupt functions are under the control of the program. The programmer is able to alter control bits in the Interrupt Enable register (IE), the interrupt Priority register (IP) and the Timer control register (TCON). The program can block all or any combination of the interrupts from acting on the program by suitably setting or clearing bits in these registers.

* After the interrupt has been handled by the interrupt subroutine which is placed by the programmer at the interrupt location in program memory, the interrupted program must resume operation at the instruction where the interrupt took place. Program resumption is done by storing the interrupted PC address on the stack in RAM before changing the PC to the interrupt address in ROM. The PC

address will be restored from the stack after an RETI instruction is executed at the end of the interrupt subroutine.

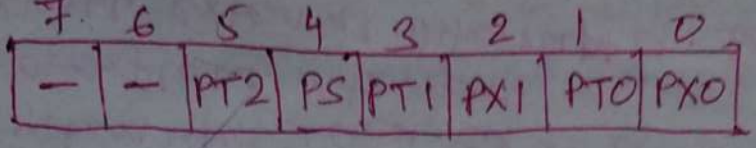
IE (Interrupt Enable) special function register –

7	6	5	4	3	2	1	0
EA	–	ET2	ES	ET1	EX1	ET0	EX0

Bit addressable as IE.0 to IE.7

Bit	Symbol	Function
7	EA	Enable interrupts bit. Set to 1 to permit individual interrupts to be enabled by their enable bits. Cleared to 0 by program to disable all interrupts. EA is a Master or global bit that can enable or disable all of the interrupts.
6	–	not implemented.
5	ET2	Reserved for feature use.
4	ES	Enable serial port interrupt. Set to 1 by program to enable serial port interrupt; cleared to 0 to disable serial port interrupt.
3	ET1	Enable timer 1 overflow interrupt. Set to 1 by program to enable timer 1 overflow interrupt; cleared to 0 to disable timer1 overflow interrupt.
2	EX1	Enable external interrupt 1. Set to 1 by program to enable $\overline{INT1}$ interrupt; cleared to 0 to disable $\overline{INT1}$ interrupt.
1	ET0	Enable timer 0 overflow interrupt. Set to 1 by program to enable timer 0 overflow interrupt; cleared to 0 to disable timer 0 overflow interrupt.
0	EX0	Enable external interrupt 0. Set to 1 by program to be $\overline{INT0}$ interrupt; cleared to 0 to disable $\overline{INT0}$ interrupt.

Interrupt Priority (IP) Special function register –



Bit addressable as IP.0 to IP.7

Bit	Symbol	Function
7	–	Not implemented
6	–	Not implemented
5	PT2	Reserved for future use
4	PS	Priority of serial port interrupt. Set/cleared by program.
3	PT1	Priority of timer 1 overflow interrupt. set/cleared by program.
2	PX1	Priority of external interrupt 1. set/cleared by program.
1	PT0	Priority of timer 0 overflow interrupt. set/cleared by program.
0	PX0	Priority of external interrupt 0. set/cleared by program.

Priority may be 1 (highest) or 0 (lowest).

Timer flag interrupt –

When a timer/counter overflows, the corresponding timer flag, TF0 or TF1, is set to 1. The flag is cleared to 0 when the resulting interrupt generates a program call to the appropriate timer subroutine in memory.

Serial Port Interrupt –

If a data byte is received, an interrupt bit, RI, is set to 1 in SCON register. When a data byte has been transmitted an interrupt bit, TI, is set to 1 in SCON register.

These are ORed together to provide a single interrupt to the processor: the serial port interrupt. These bits are not cleared when interrupt generated program call is made by the processor. The program that handles serial data communication must reset RI or TI to 0 to enable the next data communication operation.

External Interrupts.

Pins $\overline{INT0}$ and $\overline{INT1}$ are used by external circuitry. Inputs on these pins can set the interrupt flags IE0 and IE1 in the TCON register to 1 by two different methods. The IEX flags may be set when $\overline{INTX}$ pin signal reaches a low level or the flags may be set when a high-to-low transition takes place on the $\overline{INTX}$ pin. Flags IEX will be reset when a transition generated interrupt is accepted by the processor and the interrupt subroutine is accessed.

* Bit IT0 and IT1 in TCON register program the $\overline{INTX}$ pins for low level interrupt when set to 0 and program the $\overline{INTX}$ pins for transition interrupt when set to 1.

* It is the responsibility of system designer and programmer to reset any level generated external interrupts when they are serviced by the program. The external circuit must remove the low level before an RETI is executed. Failure to remove the low will result in an immediate interrupt after RETI, from the same source.

Reset -

A reset can be considered to be the ultimate interrupt because the program may not block the action of the voltage on the RST pin. This type of interrupt is often called nonmaskable, because no combination of bits in any register can stop or mask the reset action. Unlike other interrupts, the PC is not stored for later program resumption; a reset is an absolute command to jump to program address 0000h and commence running from there.

* Whenever a high level is applied to RST pin, the 8051 enters a reset condition. After RST pin is brought low, the internal registers will have the following values -

Register	Value (Hex)
PC	0000
DPTR	0000
A	00
B	00
SP	07
PSW	00
P0-3	FF
IP	XXX00000b
IE	0XX00000b
TCON	00
TMOD	00
TH0	00
TL0	00
TH1	00
TL1	00
SCON	00
SBUF	XX
PCON	0XXXXXXXb

Internal RAM contents may change during reset; also the states of the internal RAM bytes when power is first applied to 8051 are random. Register bank 0 is selected on reset as all bits in PSW are 0.

Interrupt Priority –

Register IP bits determine if any interrupt is to have a high or low priority. Bits set to 1 give the accompanying interrupt a high priority, a 0 assigns a low priority. Interrupts with a high priority can interrupt another interrupt with a lower priority; the lower priority interrupt continues after the higher is finished. If two interrupt with the same priority occur at the same time, then they have the following ranking –

1. IE0
2. TF0
3. IE1
4. TF1
5. Serial (RI or TI).

For exa– the serial interrupt could be given the highest priority by setting the PS bit in IP to 1 and all others to 0.

Interrupt Destinations –

Each interrupt source causes the program to do a hardware call to one of the dedicated addresses in program memory. It is the responsibility to place a routine at the address that will service the interrupt.

The Interrupt saves the PC of the program, which is running at the time the interrupt is serviced on the stack in internal RAM. A call is then done to appropriate memory location. The locations are-

Interrupt	Address (Hex)
IE0	0003
TF0	000B
IE1	0013
TF1	001B
Serial	0023.

A RETI instruction at the end of the routine restores the PC to its place in the interrupted program and resets the interrupt logic so that another interrupt can be serviced.

Special Function register -

Register	Bit	Primary function	Bit addressable
A	8	Math, data manipulation	Y
B	8	Math	Y
PC	8	Addressing program bytes	N
DPTR	8	Addressing code & external data	N
SP	8	Addressing internal RAM stack data	N
PSW	8	Processor status	Y
P0-P3	8	store I/O port data	Y
TH0/TL0	8/8	Timer/counter 0	N
TH1/TL1	8/8	Timer/counter 1	N
TCON	8	Timer/counter control	Y
TMOD	8	Timer/counter mode control	N
SBUF	8	Serial port data	N
SCON	8	Serial port control	Y
PCON	8	Serial port control, User flags	N
IE	8	Interrupt enable control	Y
IP	8	Interrupt priority control	Y

Software-generated Interrupts –

When any interrupt flag is set to 1 by any means, an interrupt is generated unless blocked. This means that the program itself can cause interrupts of any kind to be generated simply by setting the desired interrupt flag to 1 using a program instruction.

Addressing Modes —

Data is stored at a source address and moved (actually the data is copied) to a destination address. The ways by which these addresses are specified are called the addressing modes. Various modes are-

(1) Immediate addressing mode — Here data is part of the instruction itself. Exa- MOV A #2A
Copy data byte 2A to reg A.

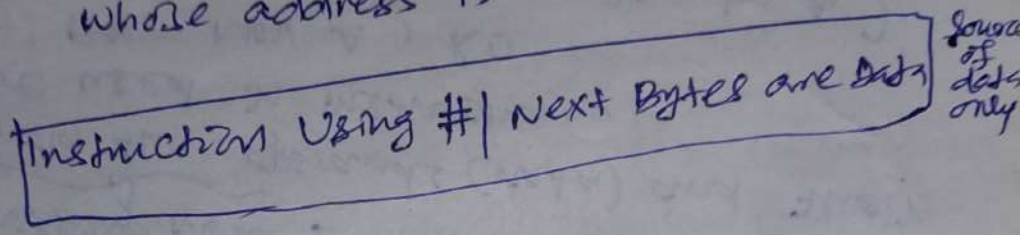
(2) Register addressing mode - In this mode register are used to hold the operands (data) and their names are used in instruction.
Exa- MOV A, R0
(copy data from R0 to A).

(3) Direct addressing mode - In this mode, address of the operand is given in the instruction. Exa- MOV A, 12h
Copy data from RAM memory location 12h to reg A.

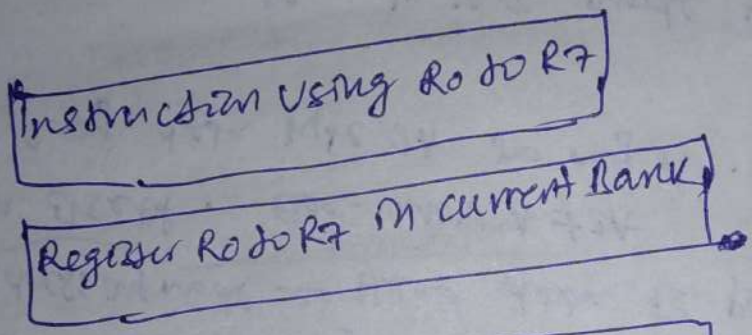
(24) Indirect addressing mode — In this mode, a register holds the actual address that will finally be used in the instruction. It means a register is used to hold the address and the register name which is mentioned in the instruction. Eg — MOV @R1, A

copy the contents of A to RAM location whose address is in R1.

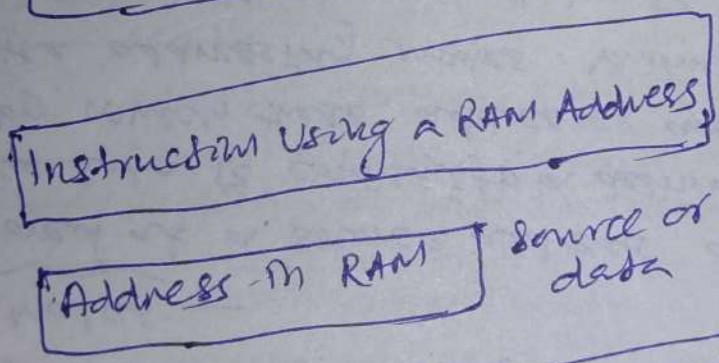
Immediate



Register



Direct



Indirect