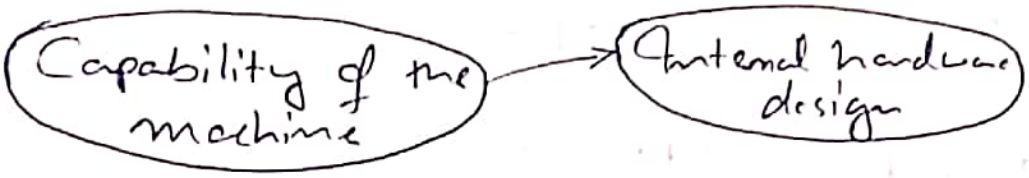
The 8051 Architecture

Introduction



Writing meaningful Programs is not possible until you have become acquainted with both the hardware and the Software of a computer.

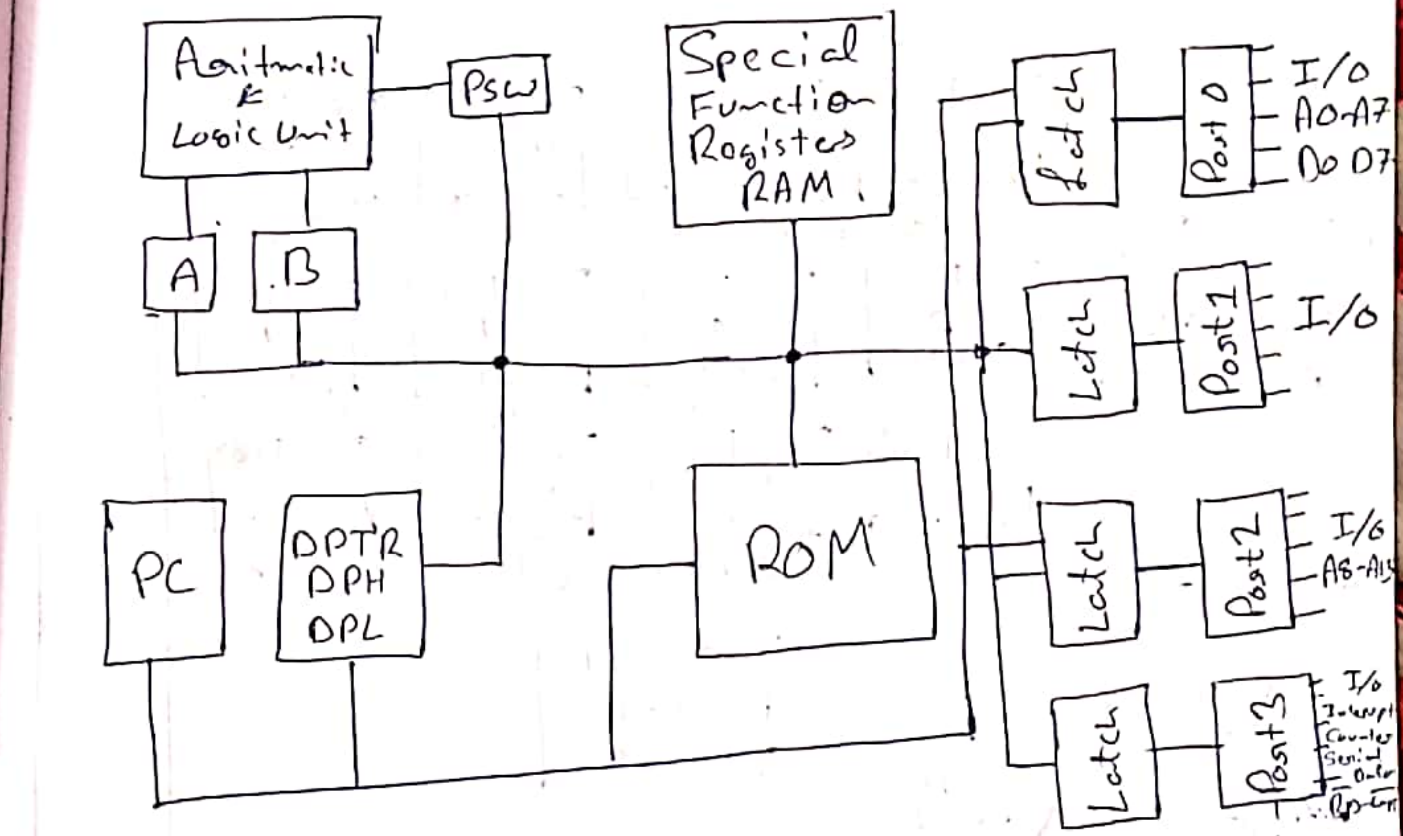
8051 Microcontroller Hardware

⇒ 8051 microcontroller actually includes a whole family of microcontrollers that have number ranging from 8031 to 8751.

→ available in
NMOS (N-Channel metal oxide Silicon)
&
CMOS (Complementary Metal oxide Silicon)

→ in a Variety of package type.

⇒ 8052 also exists with its own family of Variations and even include one member that can be programmed in BASIC.

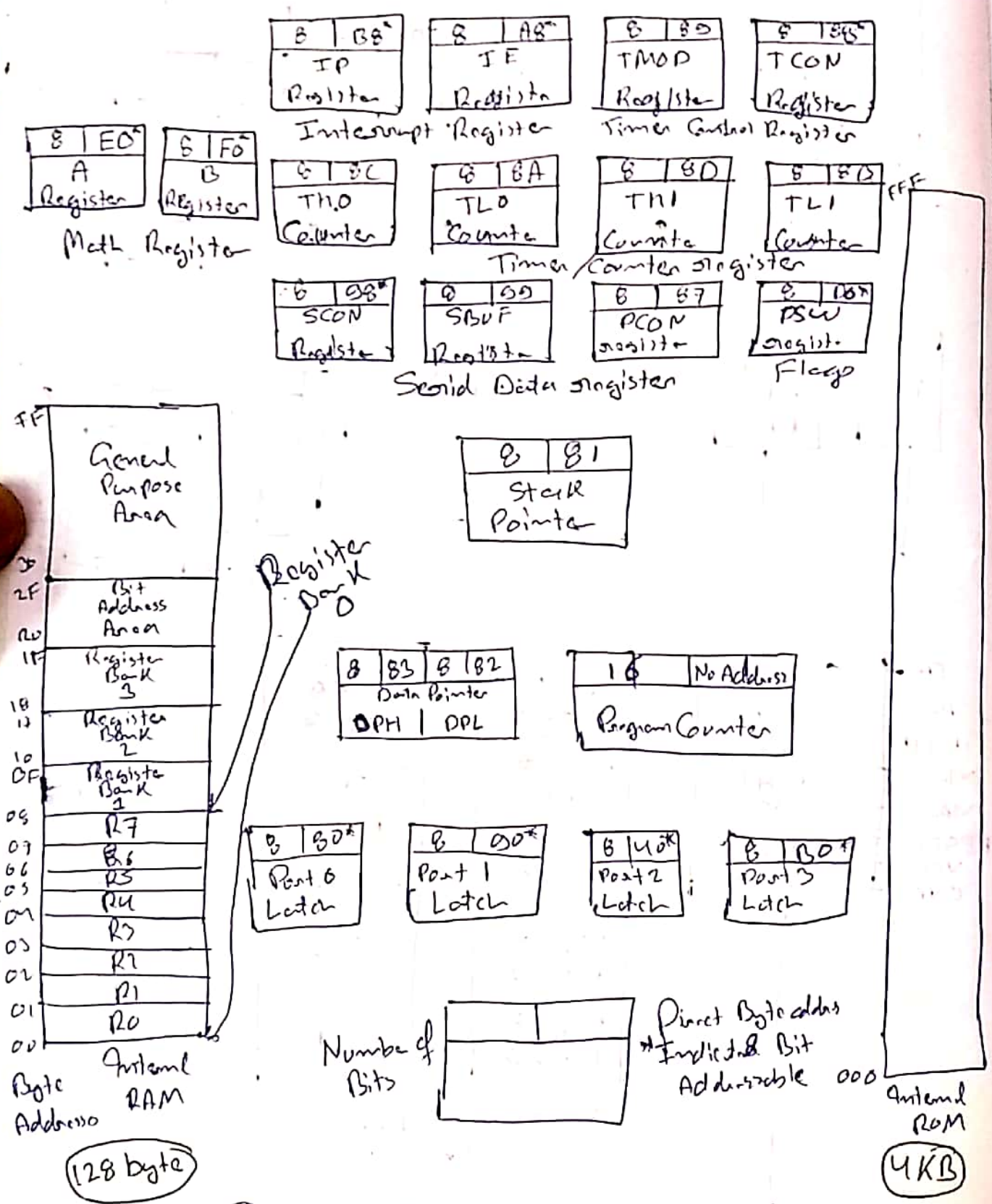


Byte/Bit Addresses
Register Bank 3
Register Bank 2
Register Bank 1
Register Bank 0

Special Function Registers
IE
IP
PCON
SBUF
TCON
TMOD
TL0
TH0
TL1
TH1

Internal RAM Structure

8051 BLOCK DIAGRAM



8051 Programing Model

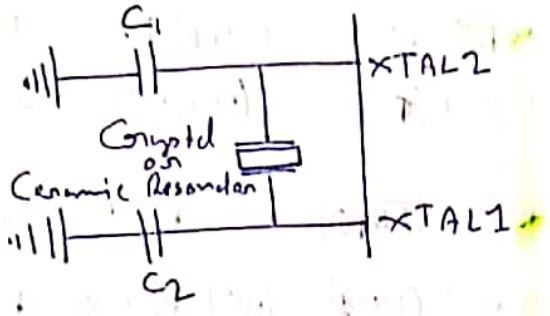
PSW = Program Status Word
On-board ROM ⇒ EPROM

⇒ Each register, with exception of the Program Counter has an internal 1-byte address assigned to it.

⇒ Software instructions generally able to specify a register by its address, its symbolic name or both.

★ The 8051 Oscillator and Clock

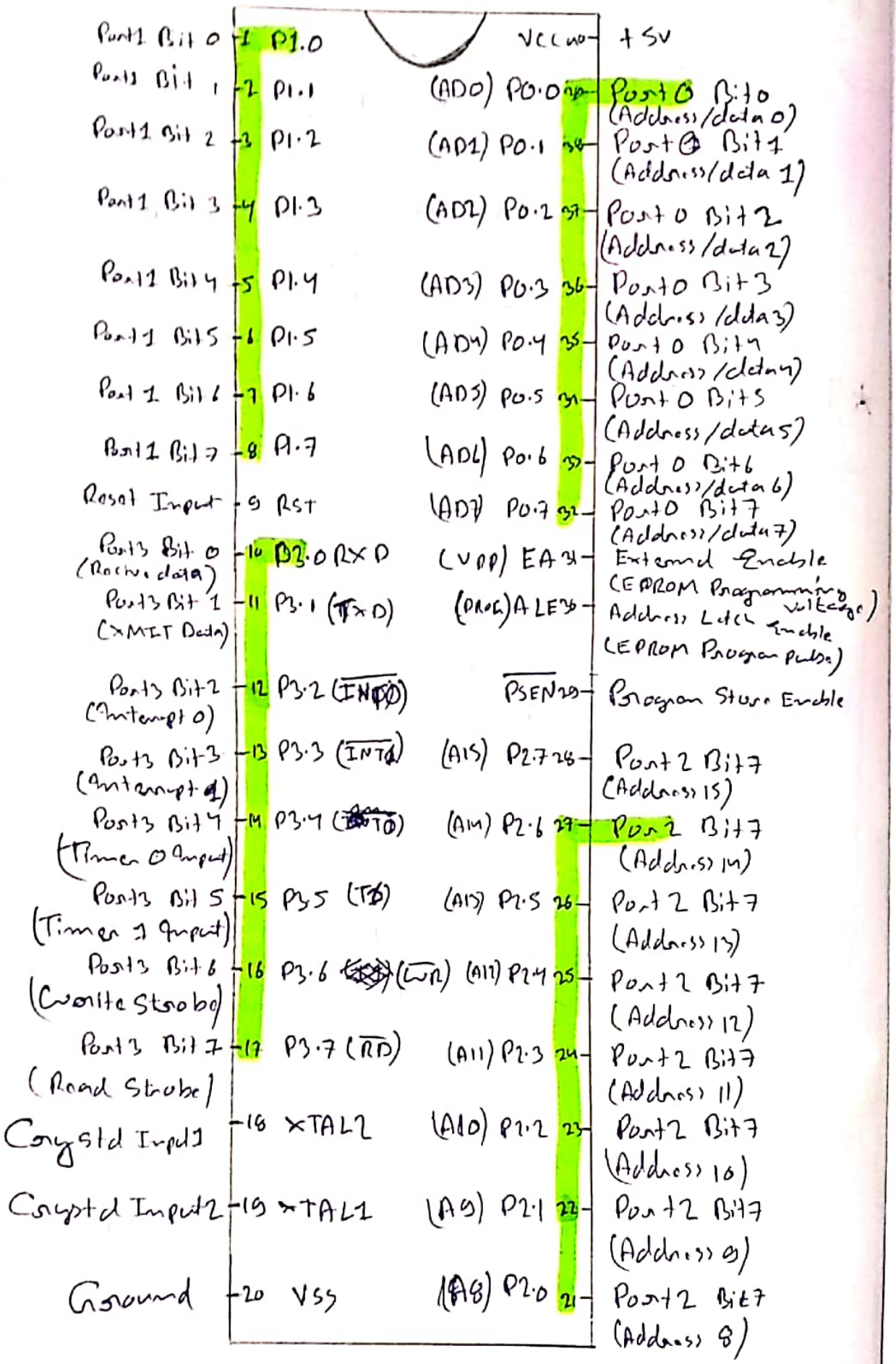
⇒ Pins XTAL1 and XTAL2 are provided for connecting a resonant network to form an oscillator.



⇒ The crystal frequency is the basic internal clock frequency of the microcontroller.

↳ Manufacturers make available 8051 designs that can run at maximum and minimum frequencies, typically 1 megahertz to 16 megahertz.

Some internal memory and dynamic are must always operate above a minimum frequency or data will be lost.



8051 DIP Pin Assignment

Ceramic resonators → Low cost Alternative of Crystal

Crystal resonator → better than Ceramic

→ Preferred if high speed data communication with other system or critical timing is to be done

⇒ The Smallest interval of time to accomplish any simple instruction or part of complex instruction is the machine cycle.

→ It is made up of 6 State.

⇒ A state is the basic time interval for discrete operations of the micro controller such as :

→ fetching an opcode byte
→ decoding an opcode
→ executing an opcode
→ writing data byte.

Two oscillating pulse define each state

⇒ A program instruction may require one, two or four machine cycle to be executed depending on the type of instruction.

C = number of cycle

$$\text{Time} = \frac{C \times 12}{\text{Crystal frequency}}$$

→ 16 MHz

A 12 megahertz Crystal yields the convenient time of one microsecond per cycle.

⇒ An 11.0592 megahertz Crystal, yields a code frequency of 921.6 KHz, which can be divided evenly by the Standard Communication baud rates of 19200, 9600, 4800, 2400, 1200 and 300 Hz.

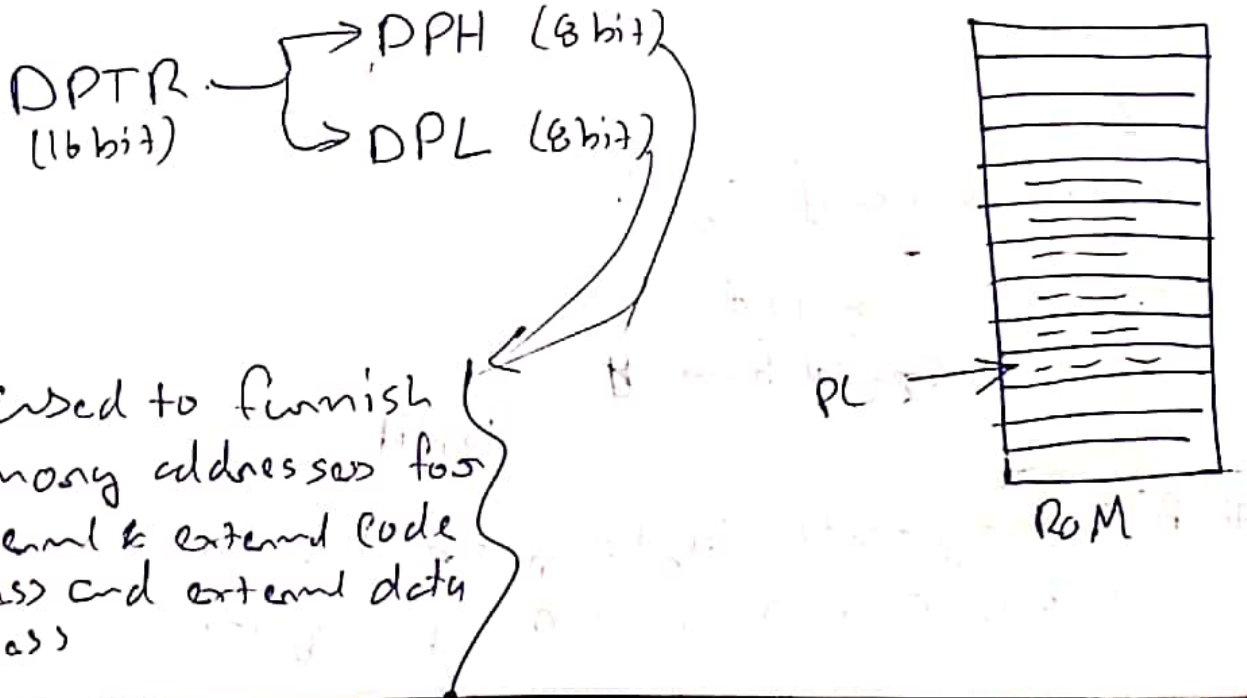
★ Program Counter and Data Pointer

⇒ 8051 Contain two 16-bit registers

- Program Counter (PC)
- Data Pointer (DPTR)

⇒ Program instruction bytes are fetched from locations in memory that are addressed by the PC.

⇒ The PC is automatically incremented after every instruction byte is fetched and may also be altered by certain instruction.



★ A and B CPU Register

8051 has 32 general purpose or working registers.

⇒ A & B, comprises the mathematical core of the 8051 CPU.

⇒ The other 32 are arranged as part of internal RAM in four banks, B0 - B3, of eight registers each, named R0 to R7.

⇒ A (Accumulator) is used in addition, subtraction, integer multiplication and division and Boolean Bit manipulator.

→ It is used for all data transfer between the 8051 and any external memory.

⇒ B is used with A for multiplication and division operations.

★ Flags and the Program Status word (PSW)

⇒ Flags are 1 byte registers provided to store the results of certain Program Instruction.

↳ In order that the flags may be conveniently addressed, they are grouped inside the Program Status word (PSW) and the Power Control registers (PCON).

⇒ 8051 has four math flag :-

(a) C ⇒ Carry
(b) AC ⇒ Auxiliary Carry
(c) OV ⇒ over flow
(d) P ⇒ Parity

⇒ Three general-purpose flag :-

(a) F0
(b) RF0
(c) RF1

{ used by the programmer to store some events in the Program }

★ Internal Memory

→ ROM : Memory for Program code bytes.

→ RAM : For Vaidble data that can be altered as the Program runs.

Additional memory can be added externally using suitable circuits.

Von Neumann architecture

Harvard architecture

⇒ In 8051 Same address in different memories for code and data can be used.

Internal RAM (128 byte)

It is organized into three distinct area :-

1) 32 bytes : from address 00h to 1Fh.

→ Organized as four banks of eight register each.

→ numbered 0 to 3 and are made up of 8 registers named R0 to R7.

2) 16 bytes: Bit-addressable area of 16 bytes occupies address 20h to 2Fh.

⇒ Address bits are useful when the program need only remember a binary event.

3) A General-purpose Ram area from address 30h to 7Fh.

★ The Stack and the Stack Pointer

⇒ Stack refers to an area of internal RAM that is used in conjunction with certain opcode to store and retrive data quickly.

⇒ (8 bit) Stak pointer register is used by the 8051 to hold an internal RAM address that is called the "top of the stak".

→ {location of last byte of data stored by a stak operation}

"When data is to be placed on the stack the SP increaments before storing data on the stake so that the Stak grows up as data is stored"

⇒ Stack is limited in height to the size of the intern RAM.

★ Special Function Registers

⇒ The 8051 operations that do not use the intern 128-byte RAM are done by a group of specific intern registers, each called a Special-function register. (addresses from 80h to FFh)

⇒ Not all of the addresses from 80h to FFh are used ∴ {attempting to use an address that is not defined, or "empty" results in unpredictable results.

⇒ Any address used in the program must start with a number, thus add E0h ⇒ 0E0h.

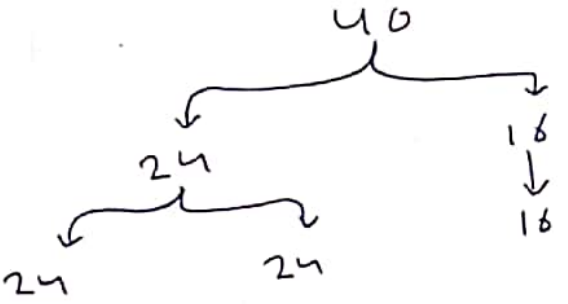
★ Intern ROM

⇒ 8051 is organized so that data memory and Program Code memory can be in two entirely different physical memory entities. Each has the same address range.

⇒ It store program and Constat data in the program permanently.

★ Input/Output Pins, Ports, and Circuit

⇒ The main constraint that limits numerous functions is the number of pins available to the 8051 circuit designers.



⇒ 64 different configuration.

⇒ The function of pin performs at any given instant depends, first, upon, what is physically connected to it and, then, upon what software command are used to program the pin.

⇒ Each port has a D-type output latch for each pin. The SRF for each port is made up of these eight latch.

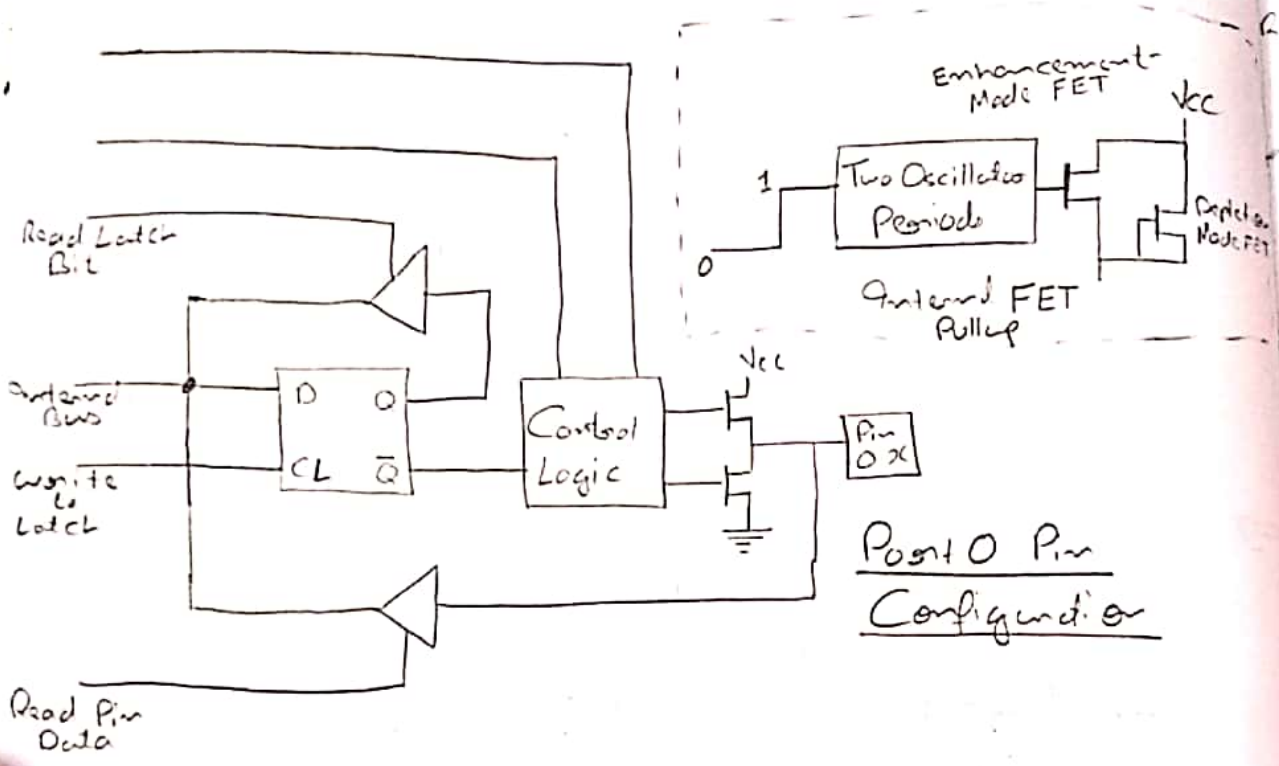
⇒ The port latches should not be confused with the Port pins; the data on the latches does not have to be the same as that on the pins.

~~⇒ Port Operations are determined~~

⇒ The Configuration of the Control Circuitry between the output latch and the Port pin determines the nature of any particular Port pin function.

⇒ The ports are not capable of driving loads that require current in the tens of mA.

Port 0



Port 0 Pin Configuration

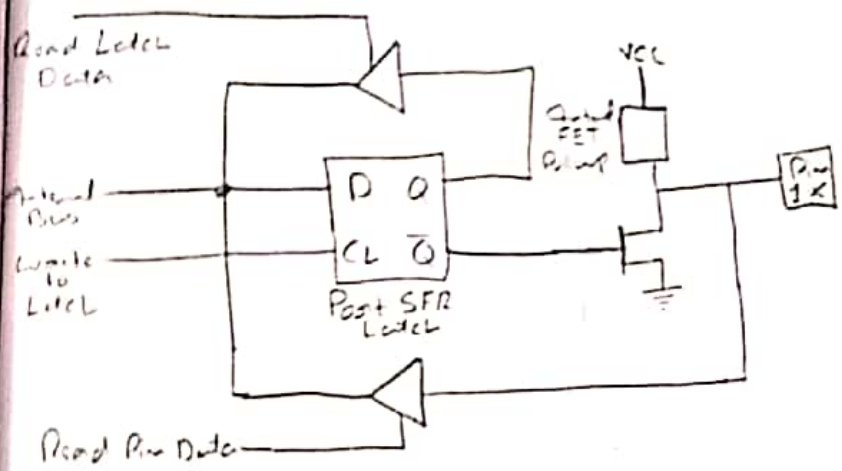
⇒ Port 0 Pins may serve as inputs, outputs or when used together as a bidirectional low-order address and data bus for extend memory.

When a pin is to be used as input, 1 must be written to the corresponding Port 0 latch by the program. Similarly for output 0.

when used as output external pullup resistors will be needed to supply a logic high.

When Port 0 is used as an address line bus to extend memory, internal Control signals switch the address lines to the gates of the (FETs).

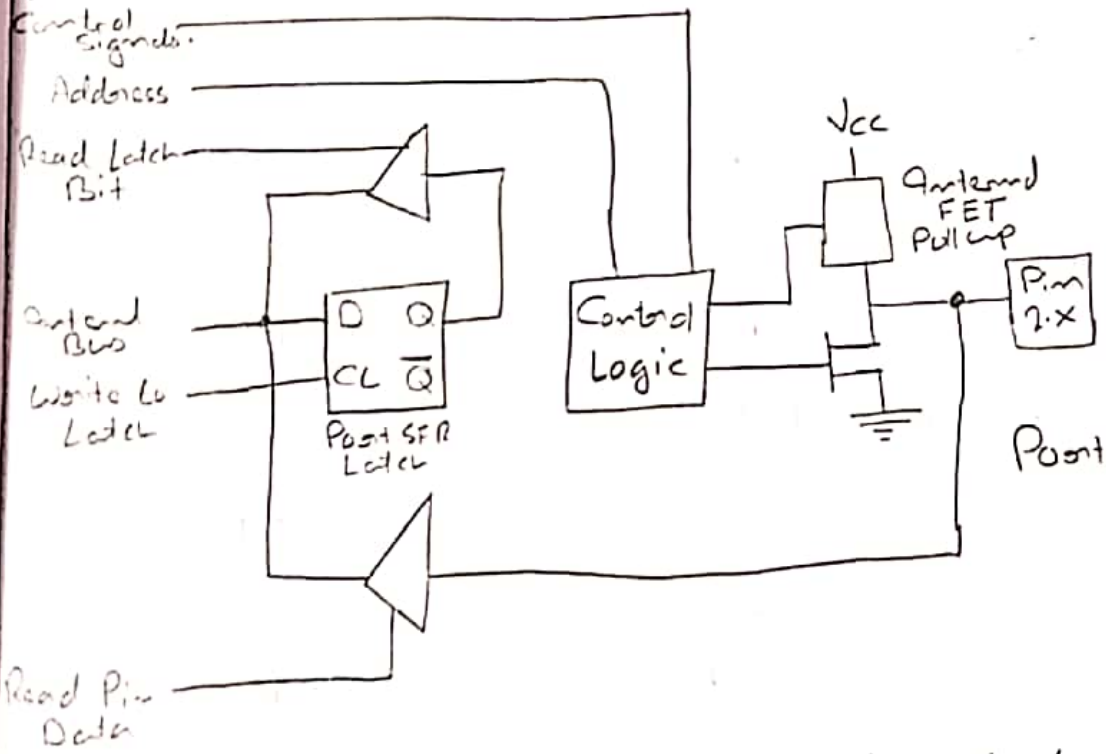
Port 1



Port 1 Pin Configuration

⇒ Port 1 pins have no dual function.

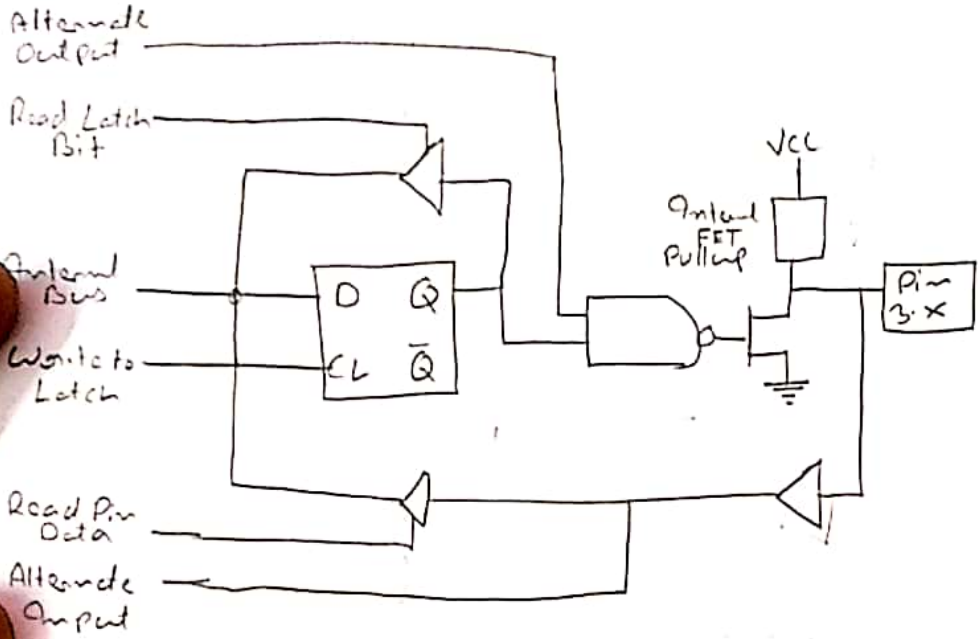
Port 2



Port 2 Pin Configuration

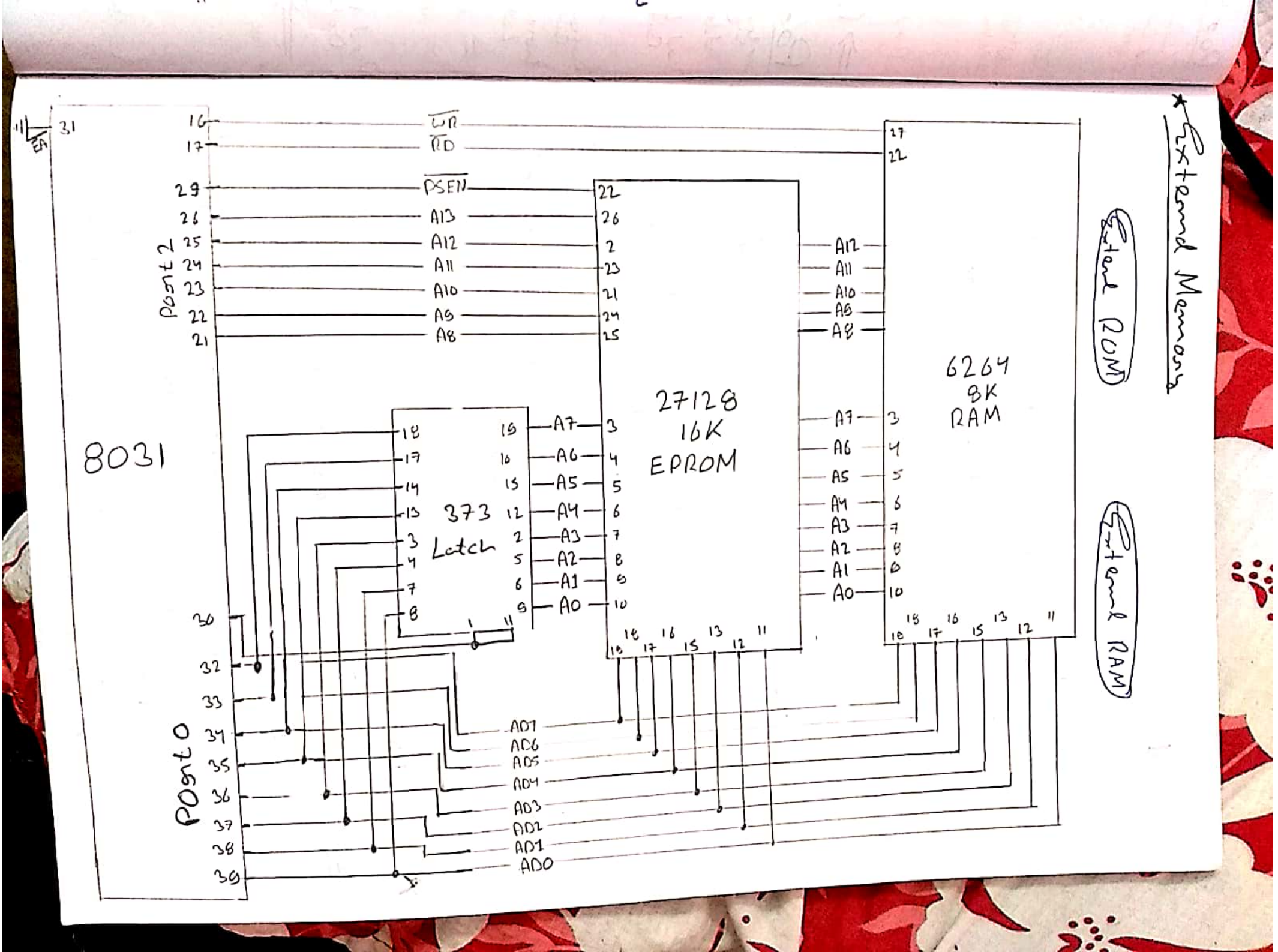
⇒ Port 2 may be used as input / output port similar to operation to Port 1. The alternative use of Port 2 is to supply a high order address byte in conjunction with the Port 0 low-order byte to address external memory.

Port 3



⇒ Port 3 is an input/output port similar to Port 1. The Port 3 alternate use are shown in the following table.

	Alternative use	SFR
P3.0	Serial data input	SBUF
P3.1	Serial data output	SBUF
P3.2	External interrupt 0	TCON.1
P3.3	External interrupt 1	TCON.3
P3.4	External timer 0 input	TMOD
P3.5	External timer 1 input	TMOP
P3.6	External memory write pulse	———
P3.7	External memory read pulse	———



⇒ 8051 accesses extend RAM whenever certain Program Instructions are executed.

⇒ Extend ROM is accessed whenever $\overline{EA}$ (External access) Pin is connected to ground or when the PC contains an address higher than the last address in the internal 4K bytes ROM.

During any memory access cycle, Port 0 is time multiplexed.

→ At first provide the low bit of 16 bit memory address, then acts as a bidirectional data bus to write/read a byte of memory data

→ The low address byte from port 0 must be latched into an external register to save the byte.

★ Counters and Timers

⇒ MicroController applications require counting and generation of time delay.

↳ Both of these tasks can be accomplished using Software technique, but software loops for counting and timing keep the Processor occupied so that other perhaps more important functions are not done

⇒ To relieve the processor of this burden, two 16-bit MP Counter named T0 and T1 are provided for the general use of the programmer.

↳ Each Clock may be programmed to count internal clock pulses or count external pulses.

⇒ 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), the timer/Counter Control register (TCON) and Certain program instruction.

TMOD

7	6	5	4	3	2	1	0
Gate	C/T	M1	M0	Gate	C/T	M1	M0
[Timer 1]				[Timer 0]			

TCON

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

★ Timer Counter Interrupt

⇒ When the Program wishes to count number of internal pulses or external events, a number is placed in one of the Counter.

★ Timing

⇒ If a Counter is programmed to be a timer, it will Count the internal Clock frequency of the 8051 Oscillator divided by 12.

{ Note, if ~~Clock~~ Crystal frequency is 12 MHz then clock frequency will be 1 MHz }

⇒ In order for oscillator clock pulses to reach the timer :

→ C/T bit in the TMOD register must be Set to 0

→ Bit TRx in TCON register must be Set to 1

→ Gate bit in the TMOD register must be 0 or external pin INTx must be 1.

Timer Modes of Operation

★ Timer Mode 0

* * *

★ Timer Mode Control (TMOD)

→ used for selecting timer or counter and mode of timer.

→ Lower 4-bits are used for Control operation of timer 0 or Counter 0 and remaining 4-bits are used for Control operation of timer 1 or counter 1.

Gate: → If gate bit is set to 0 then we can start and stop the "Software" timer in the same way.
→ If the gate is set to 1, then we can perform hardware timer.

C/T: → If C/T is 1 then it is acting as a Counter mode.
→ If C/T is 0 then it is acting as a timer mode.

M_1	M_2	Modes	
0	0	Mode 0	→ 13 bit mode
0	1	Mode 1	→ 16 bit mode
1	0	Mode 2	→ 8 bit auto reload mode.
1	1	Mode 3	→ Split timer mode.

★ Timer Control Register (TCON)

TF1: → Timer 1 flag bit.
→ Set when timer rolls from all 1s to 0.
→ Cleared when the processor vectors to execute interrupt service routine located at program address 001Bh.

TR1: → Timer 1 run Control bit.
→ Set to 1 by programmer to enable timer to count.
→ Cleared to 0 by program to halt timer

TF0 # TR0

IE1: External interrupt 1 Edge flag. Not related to timer operation.

#IT1:→ Extend interrupt 1 signal type control bit.
→ Set to 1 by Program to Enable extend interrupt 1 to be triggered by a falling edge signal.
→ Set to 0 by Program to enable a low level signal on extend interrupt 1 to generate an interrupt

#IE0: #IT0

★Counting

⇒The only difference between Counting and timing is the Source of the clock pulses to the Counters.

- When used as a timer, the clock pulses are sourced from the oscillator through the divide by 12d circuit.
- When used as Counter, Pin T0 (P3.4) Supply pulses to Counter 0 and Pin T1 (P3.5) to Counter 1.

⇒ The input pulse on Tx is sampled during P2 of state 5 every machine cycle.

- 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 pulse, the maximum input frequency that can be accurately counted is the Oscillator frequency divided by 24.

{for an 6 megahertz Crystal, the calculation yields a maximum extend frequency of 250 Kilohertz}

★ Serial data Input/Output

SCON → Register to Control data Communication

SBUF → Register to hold data

PCON → Control data Rate

⇒ Pins RXD (P3.0) and TXD (P3.1) connect to the Serial data network.

⇒ SBUF is Physically two registers ÷ ?? (99h) → Mutually Exclusive

① → Write Only
→ Used to hold data to be transmitted out

② → Read Only
→ holds ~~data~~ received data from extend Source.

⇒ There are four programmable modes of Serial data Communication.

★ Serial Data Interrupts

⇒ Serial data Communication is relatively slow Process, occupying many milliseconds per data byte to accomplish.

↳ In order not to tie up valuable processor time Serial data flags are in SCON aid in efficient data transmission and reception.

SCON

7	6	5	4	3	2	1	0
SM0	SM1	SM2	REN	TBB	RBB	TI	RI

SM0 #SM1

→ Serial port mode bits.

SM0	SM1	Mode	
0	0	0	→ Shift register, baud = $f/12$
0	1	1	→ 8bit UART; baud = Variable
1	0	2	→ 9bit UART; baud = $f/32$ or $f/64$
1	1	3	→ 9bit UART; baud = Variable

SM2

→ Multiprocessor Communication bit.

→ Set and Cleared by Program to enable multiprocessor Communications in modes 2 and 3.

↳ when Set to 1:

① : an interrupt is generated if bit 9 of the received data is a 1.

② no interrupt is generated if bit 9 is 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.

REN: Receive enable bit
- Set to 1 to enable reception.
- Cleared to 0 to disable reception.

TBB: → Transmitted bit 8.
→ Set/Cleared by program in mode 2 & 3.

RBB: → Received bit 8 of received data in mode 2 & 3.
→ Stop bit in mode 1.
→ Not used in mode 0.

TI: Transmit interrupt flag.
RI: Receive interrupt flag.

PCON

7	6	5	4	3	2	1	0
SMOD	-	-	-	GFI	GFO	PD	IDL

SMOD: Serial baud rate modify bit.
- Set to 1 by program to double baud rate using timer 1 for mode 1, 2, & 3.
- Cleared to 0 by program to use timer 1 baud rate.

GF1: # GF0
- General purpose user flag bit.
- Set/cleared by program.

PD: Power down bit
- Set to 0 to enable power down configuration.

IDL: Idle mode bit.
- Set to 1 by program to enter idle mode.

★ Data transmission

⇒ It begins anytime data is written SBUF.

↳ TI is set to 1 when data has been transmitted and signifies that SBUF is empty.

★ Data Reception

⇒ It begins if the receive enable bit (REN) in SCON is set to 1 for all modes.

↳ Incoming data is not transferred to SBUF until the last data bit has been received so that the previous transmission can be read from SBUF while new data is being received.

★ Interrupts

⇒ A Computer has two way to determine the Conditions that exist in intenal/Extenal circuit

One method uses software instructions that Jump on the states of flags and Port pins

The Second responds to Hardware signals, called Interrupts, that forces the program to call a Sub-routine.

⇒ Five interrupts are provided in the 8051.

3 Intenal

2 Extenal Interrupt

⇒ After the interrupt has been handled by the interrupt subroutine, which is placed by the programer at the instruction where the interrupt took place.

Timer flag 0	Timer flag 1	Serial port interrupt

INT0	INT1

★ 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 non-maskable interrupts.