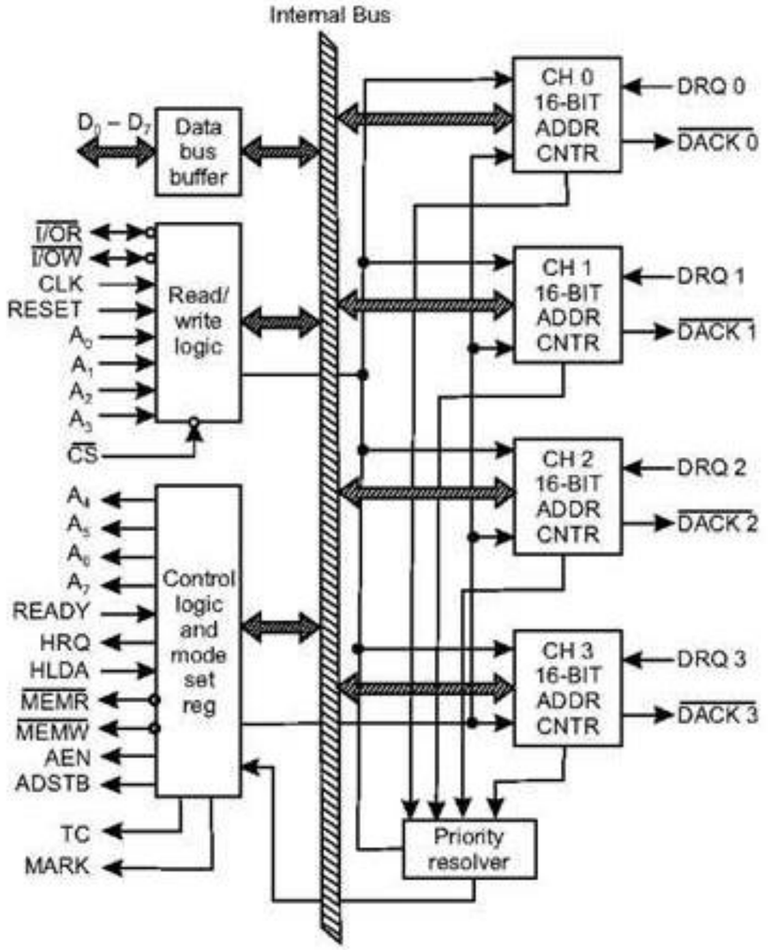
MPMC ASSIGNMENT

1. Draw and discuss the architecture of 8257

Ans: The Architecture of 8257 DMA is show below.

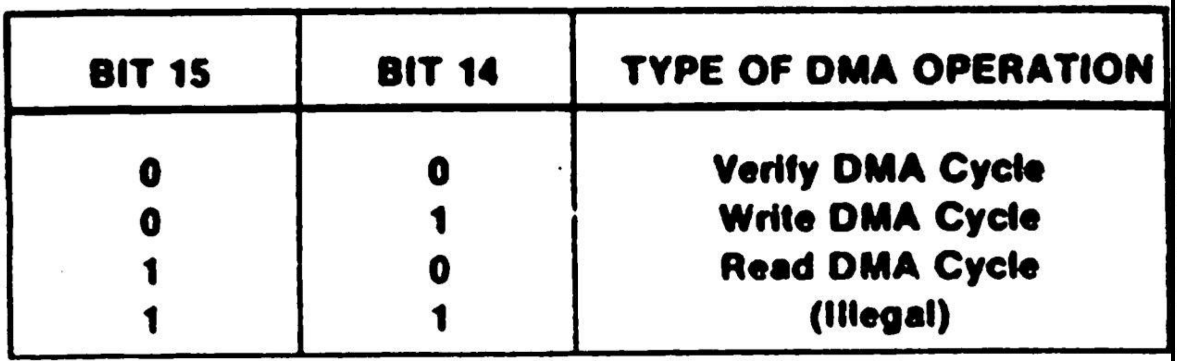


BLOCK DIAGRAM DESCRIPTION:

1. **DMA Channels:**

   The 8257 provides four separate DMA channels (labelled CH-0 to CH-3). Each channel includes two sixteen-bit registers: (1) DMA address register, and (2) terminal count register. Both registers must be initialized before channel is enabled. The DMA address register is loaded with the address of the first memory location to be accessed. The value loaded into the low-order 14-bits of the terminal count register specifies the number of DMA cycles minus one before the Terminal Count (TC) output is activated. For instance, terminal count of would cause the TC output to be active in the first DMA cycle for that channel. In general, if the number of desired DMA cycles, load the value N - 1 into the low-order 14-bits of the terminal count register. The most significant two bits of the terminal count register specify the type of DMA operation for that channel. These two bits are not modified during DMA cycle, but can be changed between DMA blocks. Each channel accepts DMA Request (DRQn) input and provides DMA Acknowledge

(DACKn) output



BIT 15	BIT 14	TYPE OF DMA OPERATION
0	0	Verify DMA Cycle
0	1	Write DMA Cycle
1	0	Read DMA Cycle
1	1	(Illegal)

(DRQ 0-DRQ 3) DMA Request: These are individual asynchronous channel request inputs used by the peripherals to obtain DMA cycle. If not in the rotating priority mode then DRQ 0 has the highest priority and DRQ has the lowest .request can be generated by raising the request line and holding it high until DMA acknowledge. For multiple DMA cycles (Burst Mode) the request line is held high until the DMA acknowledge of the last cycle arrives.

(DACK DACK 3)- DMA Acknowledge: An active low level on the acknowledge output informs the peripheral connected to that channel that it has been selected for DMA cycle. The DACK output acts as "chip select'* for the peripheral device requesting service. This line goes active (low) and inactive (high) once for each byte transferred even if burst of data is being transferred. 2. Data Bus Buffer This three-state, bi-directional, eight bit buffer interfaces the 8257 to the system data bus.

2. Data Bus Buffer

This three-state, bi-directional, eight bit buffer interfaces the 8257 to the system data bus.

(D0-07):

Data Bus Lines: These are bi-directional three-state lines. When the 8257 is being programmed by the CPU. Eight bits of data for DMA address register, terminal count register or the Mode Set register are received on the data bus. When the CPU reads DMA address register, terminal count register or the Status register, the data is sent to the CPU over the data bus. During DMA cycles (when the 8257 is the bus master), the 8257 will output the most significant eight-bits of the memory address (from one of the DMA address registers) to the 8212 latch via the data bus. These address bits will be transferred at the beginning of the DMA cycle: the bus will then be released to handle the memory data transfer during the balance of the DMA cycle.

3. Read/Write Logic

When the CPU is programming or reading one of the 8257*s registers (i.e., when the 8257 is "slave" device on the system bus), the Read/Write Logic accepts the I/O Read (USE) or I/O Write (175OT) signal, decodes the least significant four address bits, (A0-A3), and either writes the contents of the data bus into the addressed register (if I/OW is true) or places the contents of the addressed register onto the data bus (if I/OR is true). During DMA cycles (i.e., when the 8257 is the bus "master"), the Read/Write Logic generates the I/O read and memory write (DMA write cycle) or I/O Write and memory read (DMA read cycle) signals which control the data link with the peripheral that has been granted the DMA cycle. Note that during DMA transfers Non-DMA I/O devices should be de-selected (disabled) using "AEN" signal to inhibit I/O device decoding of the memory address as an erroneous device address.

(i/OR)
I/O Read: An active-low, bi-directional three-state line. In the "slave" mode, it is an input which allows the 8-bit status register or the upper/lower byte of 18-bit DMA address register or terminal count register to be read. In the "master" mode, I/OR is control output which is used to access data from peripheral during the DMA write cycle.

(I/OW)
I/O Write: An active-low, bi-directional three-state line. In the "slave" mode, it is an input which allows the contents of the data bus to be loaded into the 8-bit mode set register or the upper/lower byte of 18-bit DMA address register or terminal count register. In the "master" mode. I/OW is control output which allows data to be output to peripheral during DMA read cycle.

(**CLK**)
Clock Input: Generally from an Intel® 8224 Clock Gen erator device. (*2 TTL) or Intel® 8085A CLK output.

(RESET)
Reset: An asynchronous input (generally from an 8224 or 8085 device) which disables all DMA channels by clearing the mode register and 3-states all control lines.

4. Control Logic

This block controls the sequence of operations during all DMA cycles by generating the appropriate control signals and the 16-bit address that specifies the memory location to be accessed.

Address Lines:

These four address lines are three-state outputs which constitute bits through of the 16-bit memory address generated by the 8257 during all OMA cycles.

(READY) Ready: This asynchronous input is used to elongate the memory read and write cycles in the 8257 with wait states if the selected memory requires longer cycles. READY must conform to specified setup and hold times.

(HRQ) Hold Request: This output requests control of the system bus. In systems with only one 8257, HRQ will normally be applied to the HOLD input on the CPU. HRQ must conform to specified setup and hold times.

(HLDA) Hold Acknowledge: This input from the CPU indicates that the 8257 has acquired control of the system bus.

(MEMR) Memory Read: This active-low three-state output is used to read data from the addressed memory location during DMA Read cycles.

(MEMW) Memory Write: This active-low three-state output is used to write data into the addressed memory location during DMA Write cycles.

(AOSTB) Address Strobe: This output strobes the most significant byte of the memory address into the 8212 device from the data bus.

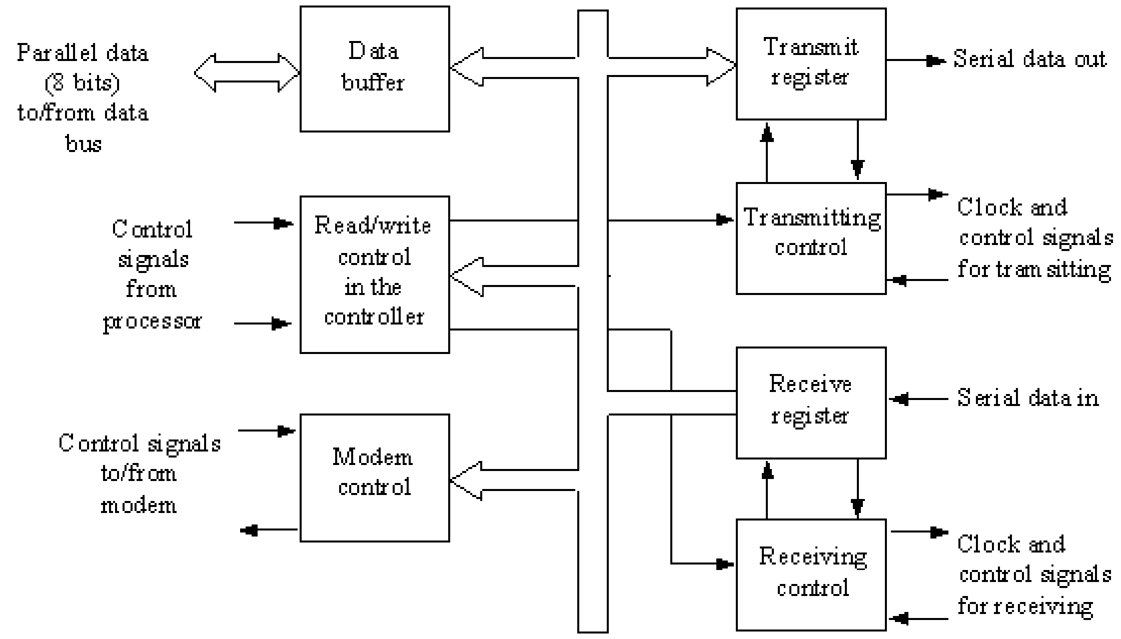
(AEN) Address Enable: This output is used to disable (float) the System Data Bus and the System Control Bus. It may also be used to disable (float) the System Address Bus by use of an enable on the Address Bus drivers in systems to inhibit non-DMA devices from responding during DMA cycles. It may be further used to isolate the 8257 data bus from the System Data Bus to facilitate the transfer of the most significant DMA address bits over the 8257 data I/O pins without subjecting the System Data Bus to any timing constraints for the transfer. When the 8257 is used in an I/O device structure (as opposed to memory mapped), this AEN output should be used to disable the selection of an I/O device whe n the DMA address is on the address bus. The I/O device selection should be determined by the DMA acknowledge outputs for the channels.

(TC) Terminal Count: This output notifies the currently selected peripheral that the present DMA cycle should be the last cycle for this data block. If the TC STOP bit in the Mode Set register is set. the selected channel will be automatically disabled at the end of that DMA cycle. TC is activated when the 14-bit value in the selected channel's terminal count register equals zero. Recall that the loworder 14-bits of the terminal count register should be loaded with the values (n-1). where the desired number of the DMA cycles.

(**MARK) Modulo 128 Mark:** This output notifies the selected peripheral that the current DMA cycle is the 128th cycle since the previous MARK output. MARK always occurs at 128 (and all multiples of 128) cycles from the end of the data block. Only if the total number of DMA cycles (n) is evenly divisable by 128 (and the terminal count register was loaded with n-1). will MARK occur at 128 (and each succeeding multiple of 128) cycles from the **beginning of the data block.**

2-a. Explain about 8251A (USART) with neat diagram.

Ans: The Architecture of 8251A is shown below.


Parallel data (8 bits) to/from data bus
Data buffer
Control signals from processor
Read/write control in the controller
Control signals to/from modem
Modem control
Transmit register
Serial data out
Transmitting control
Clock and control signals for tram sitting
Receive register
Serial data in
Receiving control
Clock and control signals for receiving

♣ 8251 is a USART (Universal Synchronous Asynchronous Receiver Transmitter) for serial data communication.

♣ Programmable peripheral designed for synchronous /asynchronous serial data communication, packaged in a 28-pin DIP.

♣ Receives parallel data from the CPU & transmits serial data after conversion.

♣ Also receives serial data from the outside & transmits parallel data to the CPU after conversion.

Sections of 8251:

1. **Data Bus Buffer**

   ¬ **D0-D7 :** 8-bit data bus used to read or write status, command word or data from or to the 8251A 2.

2. **Read/Write Control logic**

   ♣ Includes a control logic, six input signals & three buffer registers: Data register, control register & status register.

   ♣ **Control logic** : Interfaces the chip with MPU, determines the functions of the chip according to the control word in the control register & monitors the data flow.

3. **Modem Control**

   ♣ DSR – Data Set Ready : Checks if the Data Set is ready when communicating with a modem.

   ♣ DTR - Data Terminal Ready : Indicates that the device is ready to accept data when the 8251 is communicating with a modem.

   ♣ CTS - Clear to Send : If its low, the 8251A is enabled to transmit the serial data provided the enable bit in the command byte is set to '1'.

   ♣ RTS - Request to Send Data : Low signal indicates the modem that the receiver is ready to receive a data byte from the modem.

4. **Transmitter section**

   ♣ Accepts parallel data from MPU & converts them into serial data.

   ♣ Has two registers:

   | Buffer register : To hold eight bits

   | Output register : To convert eight bits into a stream of serial bits.

   ♣ The MPU writes a byte in the buffer register.

   ♣ Whenever the output register is empty; the contents of buffer register are transferred to output register.

   ♣ Transmitter section consists of three output & one input signals

   | TxD - Transmitted Data Output: Output signal to transmit the data to peripherals

   | TxC - Transmitter Clock Input: Input signal, controls the rate of transmission.

   | TxRDY - Transmitter Ready: Output signal, indicates the buffer register is empty and the USART is ready to accept the next data byte.

   | TxE - Transmitter Empty: Output signal to indicate the output register is empty and the USART is ready to accept the next data byte.

5. **Receiver Section**

   ♣ Accepts serial data on the RxD pin and converts them to parallel data.

   ♣ Has two registers:

   | Receiver input register

   | Buffer register

   ♣ When RxD goes low, the control logic assumes it is a start bit, waits for half bit time, and samples the line again. If the line is still low, the input register accepts the following data, and loads it into buffer register at the rate determined by the receiver clock.

   ♣ RxRDY - Receiver Ready Output: Output signal, goes high when the USART has a character in the buffer register & is ready to transfer it to the MPU.

♣ RxD - Receive Data Input : Bits are received serially on this line & converted into a parallel byte in the receiver input register.
♣ RxC - Receiver Clock Input : Clock signal that controls the rate at which bits are received by the USART.

Control Register:

♣ 16-bit register for a control word consist of two independent bytes namely mode word & command word.

♣ Mode word: Specifies the general characteristics of operation such as baud, parity, number of bits etc.

♣ Command word: Enables the data transmission and reception.

♣ Register can be accessed as an output port when the Control/Data pin is high.

Status register:

♣ Checks the ready status of the peripheral.

♣ Status word in the status register provides the information concerning register status and transmission errors.

Data register:

♣ Used as an input and output port when the C/D is low

2-b.explain in detail about the IEEE488 protocol

IEEE-488 refers to the Institute of Electrical and Electronics Engineers (IEEE) Standard number 488. This standard was first established in 1978, 13 years after Hewlett-Packard (HP) of Palo Alto, CA, began work to enable its broad range of instruments to communicate with one another and with “host” computers.

The IEEE-488 interface, sometimes called the General Purpose Interface Bus (GPIB), is a general purpose digital interface system that can be used to transfer data between two or more devices. It is particularly wellsuited for interconnecting computers and instruments. Some of its key features are:

->Up to 15 devices may be connected to one bus

->Total bus length may be up to 20 m and the distance between devices may be up to 2 m

->Communication is digital (as opposed to analog) and messages are sent one byte (8 bits) at a time

->Message transactions are hardware handshaked

->Data rates may be up to 1 Mbyte/sec

3. Define Addressing Mode. Explain various addressing modes of 8051 wit examples.

Addressing Mode: Addressing modes are an aspect of the instruction set architecture in most central processing unit (CPU) designs. The various **addressing modes** that are defined in a given instruction set architecture define how machine language instructions in that architecture identify the operand(s) of each instruction.

Addressing modes of 8051

1. Immediate addressing mode:

In this type, the operand is specified in the instruction along with the opcode. In simple way, it means data is provided in instruction itself.

Ex: MOV A, #05H -> Where MOV stands for move, # represents immediate data. 05h is the data. It means the immediate date 05h provided in instruction is moved into A register.

2. Register addressing mode:

Here the operand in contained in the specific register of microcontroller. The user must provide the name of register from where the operand/data need to be fetched. The permitted registers are A, R7-R0 of each register bank. Ex: MOV A, R0-> content of R0 register is copied into Accumulator.

3**. Direct addressing mode**:

In this mode the direct address of memory location is provided in instruction to fetch the operand. Only internal RAM and SFR's address can be used in this type of instruction. Ex: MOV A, 30H => Content of RAM address 30H is copied into Accumulator.

4. **Register In direct addressing mode**:

Here the address of memory location is indirectly provided by a register. The '@' sign indicates that the register holds the address of memory location i.e. fetch the content of memory location whose address is provided in register.
Ex: MOV A,@R0 => Copy the content of memory location whose address is given in R0 register.

5**. Indexed Addressing mode:**

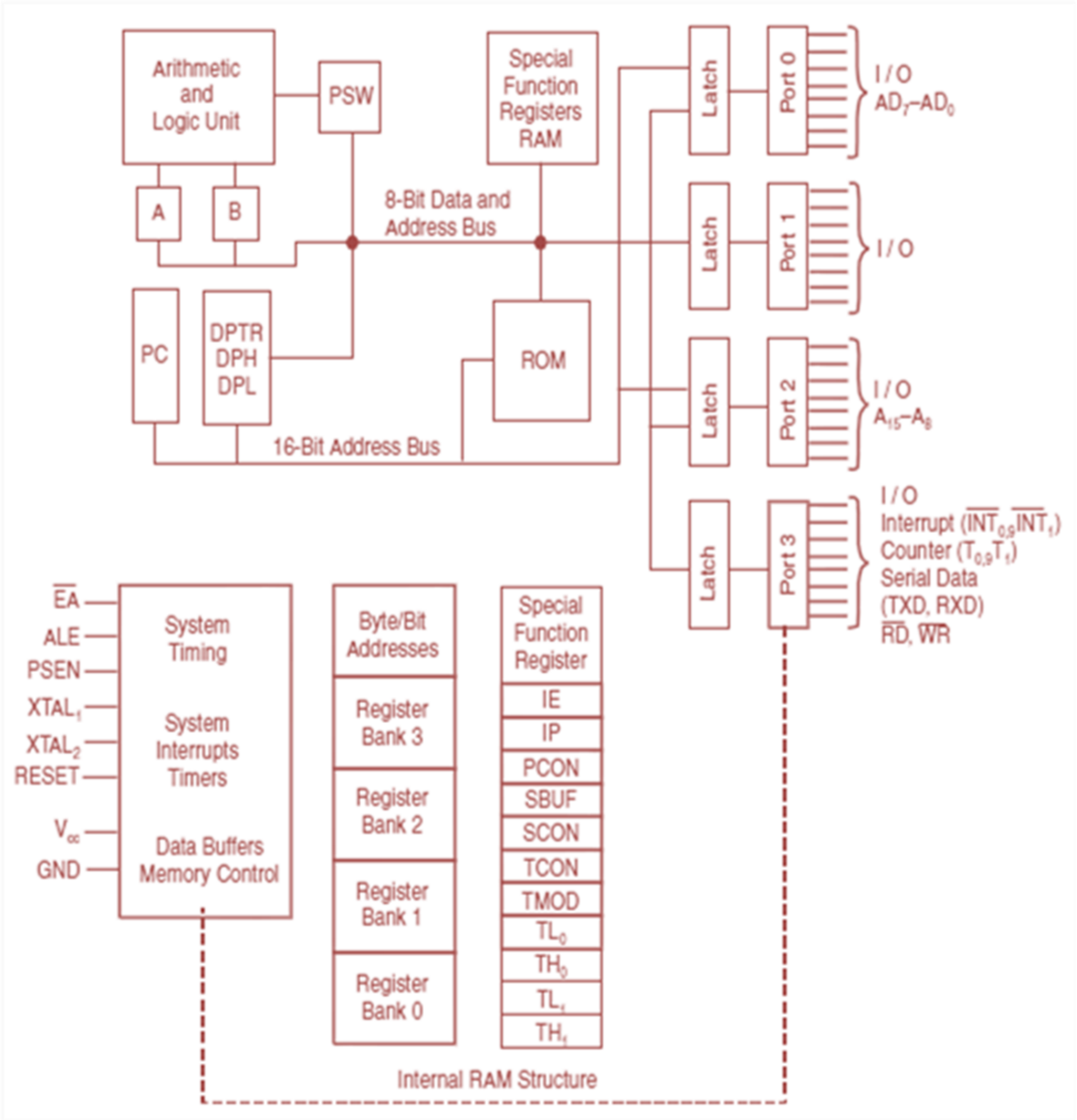
This addressing mode is basically used for accessing data from look up table. Here the address of memory is indexed i.e. added to form the actual address of memory.

Ex: MOVC A,@A+DPTR => here 'C' means Code. Here the content of A register is added with content of DPTR and the resultant is the address of memory location from where the data is copied to A register.

4. Draw the Architectural diagram of 8051 and explain each block.

Ans: The 8051 Microcontroller is a Microcontroller designed by Intel in 1980's. It was based on Harvard Architecture and developed primarily for use in Embedded Systems. Originally it was developed using NMOS technology but as those requires more power to operate there fore Intel redesigned Microcontroller 8051 using CMOS technology and later versions came with a letter 'C' in their name, for example: 80C51. These latest Microcontrollers requires less power to operate as compared to their predecessors.

The Architecture of 8051 Microcontroller is shown below



1. Oscillator and clock generator:

All operations in a microcontroller are synchronized by the help of an oscillator clock. The oscillator clock generates the clock pulses by which all internal operations are synchronized. A resonant network connected through pins XTAL1 and XTAL2 forms up an oscillator. For this purpose a quartz crystal and capacitors are employed. The crystal run at specified maximum and minimum frequencies typically at 1 MHz to 16 MHz.

2. ALU:

It is 8 bit unit. It performs arithmetic operation as addition, subtraction, multiplication, division, increment and decrement. It performs logical operations like AND, OR and EX-OR. It manipulates 8 bit and 16 bit data. It calculates address of jump locatio ns in relative branch instruction. It performs compare, rotate and compliment operations. It consists of Boolean processor which performs bit, set, test, clear and compliment. 8051 micro controller contains 34 general purpose registers or working registers.2 of them are called math registers A & B and 32 are bank of registers.

a. Accumulator(A-reg):

It is 8 bit register. Its address is E0H and it is bit and byte accessible. Result of arithmetic & logic operations performed by ALU is accumulated by this register. Therefore it is called accumulator register. It is used to store 8 bit data and to hold one of operand of ALU units during arithmetical and logical operations. Most of the instructions are carried out on accumulator data. It is most versatile of 2 CPU registers.

b. B-register:

It is special 8 bit math register. It is bit and byte accessible. It is used in conjunction with A register as I/P operand for ALU. It is used as general purpose register to store 8 bit data.

c. PSW:

It is 8 bit register. Its address is D0H and It is bit and byte accessible. It has 4 conditional flags or math flags which sets or resets according to condition of result. It has 3 control flags, by setting or resetting bit required operation or function can be achieved. The format of flag register is as shown below:

i. MATH FLAG:

1. Carry Flag (CY): During addition and subtraction any carry or borrow is generated then carry flag is set otherwise carry flag resets. It is used in arithmetic, logical, jump, rotate and Boolean operations.
2. Auxiliary carry flag (AC): If during addition and subtraction any carry or borrow is generated from lower 4 bit to higher 4 bit then AC sets else it resets. It is used in BCD arithmetic operations.
3. Overflow flag (OV): If in signed arithmetic operations result exceeds more than 7 bit than OV flag sets else resets .It is used in signed arithmetic operations only.
4. Parity flag (P): If in result, even no. Of ones "1" are present than it is called even parity and parity flag sets. In result odd no. Of ones "1"are present than it is called odd parity and parity flag resets.

ii. CONTROL FLAGS:

1. FO: It is user defined flag. The user defines the function of this flag. The user can set test n clear this flag through software.
2. RS1 and RS0: These flags are used to select bank of register by resetting those flags
3. Program counter (PC): The Program Counter (PC) is a 2-byte address which tells the 8051 where the next instruction to execute is found in memory. It is used to hold 16 bit address of internal RAM, external RAM or external ROM locations. When the 8051 is initialized PC always starts at 0000h and is incremented each time an instruction is

executed. It is important to note that PC isnt always incremented by one and never decremented.

4. Data pointer register (DTPR): It is a 16 bit register used to hold address of external or internal RAM where data is stored or result is to be stored. It is used to store 16 bit data. It is divided into2- 8bit registers, DPH-data pointer higher order (83H) and DPL-data pointer lower order (82H). Each register can be used as general purpose register to store 8 bit data and can also be used as memory location. DPTR does not have single internal address. It functions as Base register in base relative addressing mode and in-direct jump.

5. Stack pointer(SP): It is 8-bit register. It is byte addressable. Its address is 81H. It is used to hold the internal RAM memory location addresses which are used as stack memory. When the data is to be placed on stack by push instruction, the content of stack pointer is incremented by 1, and when data is retrieved from stack, content of stack of stack pointer is decremented by 1.

iii. Special function Registers(SFR): The 8051 microcontroller has 11 SFR divided in 4 groups:

A. Timer/Counter register: 8051 microcontroller has 2-16 bit Timer/counter registers called Timer-reg-T0 And Timer/counter Reg-T1.Each register is 16 bit register divide into lower and higher byte register as shown below: These register are used to hold initial no. of count. All of the 4 register are byte addressable.

1. Timer control register: 8051 microcontroller has two 8-bit timer control register i.e. TMOD and TCON register. TMOD Register: it is 8-bit register. Its address is 89H. It is byte addressable. It used to select mode and control operation of time by writing control word.

2. TCON register: It is 8-bit register. Its address is 88H. It is byte addressable. Its MSB 4-bit are used to control operation of timer/ counter and LSB 4-bit are used for external interrupt control.

B. Serial data register: 8051 micro controller has 2 serial data register viz. SBUF and SCON.

1. Serial buffer register (SBUF): it is 8-bit register. It is byte addressable .Its address is 99H. It is used to hold data which is to be transferred serially.

2. Serial control register (SCON): it is 8-bit register. It is bit/byte addressable. Its address is 98H. The 8-bit loaded into this register controls the operation of serial communication.

C. Interrupt register: 8051 µC has 2 8-bit interrupt register.

1. Interrupt enable register (IE): it is 8-bit register. It is bit/byte addressable. Its address is A8H.it is used to enable and disable function of interrupt.

2. Interrupt priority register (IP): It is 8-bit register. It is bit/byte addressable. Its address is B8H. it is used to select low or high level priority of each individual interrupts.

D. Power control register (PCON): it is 8-bit register. It is byte addressable .Its address is 87H. its bits are used to control mode of power saving circuit, either idle or power down mode and also one bit is used to modify baud rate of serial communication.

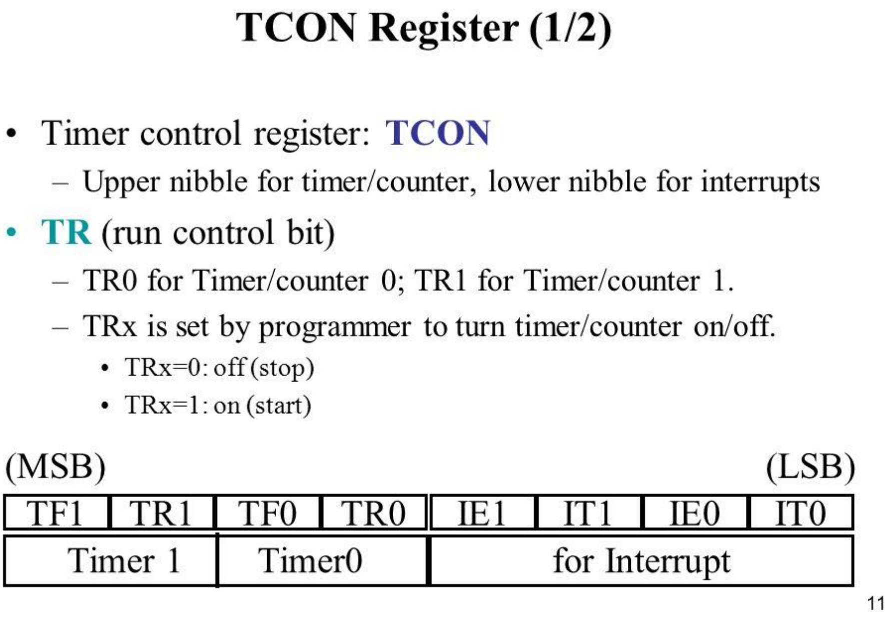
Internal RAM

Internal RAM has memory 128-byte. See 8051 hardware for further internal RAM design. Internal RAM is organized into three distinct areas: 32 bytes working registers from address 00h to 1Fh 16 bytes bit addressable occupies RAM byte address 20h to 2Fh, altogether 128 addressable bits General purpose RAM from 30h to 7Fh.

Internal ROM

Data memory and program code memory both are in different physical memory but both have the same addresses. An internal ROM occupied addresses from 0000h to 0FFFh. PC addresses program codes from 0000h to 0FFFh. Program addresses higher than 0FFFh that exceed the internal ROM capacity will cause 8051 architecture to fetch codes bytes from external program memory.

5-a. Draw the register formats of TMOD and TCON.

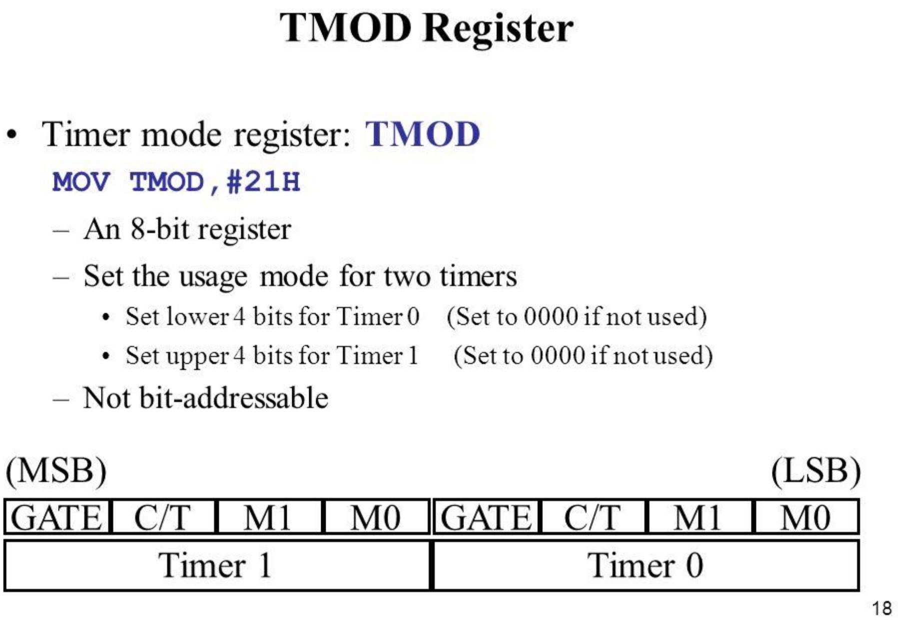


TCON Register (1/2)

- Timer control register: **TCON**
  - Upper nibble for timer/counter, lower nibble for interrupts
- **TR** (run control bit)
  - TR0 for Timer/counter 0; TR1 for Timer/counter 1.
  - TRx is set by programmer to turn timer/counter on/off.
    - TRx=0: off (stop)
    - TRx=1: on (start)

(MSB) (LSB)

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Timer 1		Timer0		for Interrupt			



TMOD Register

- Timer mode register: **TMOD**

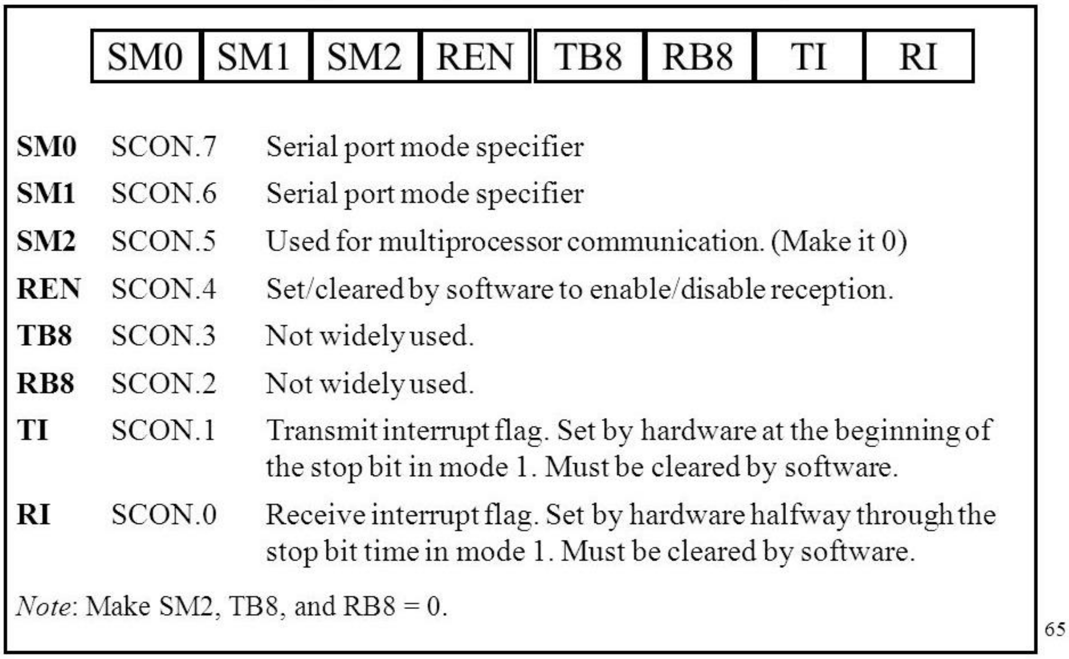
```
MOV TMOD,#21H
```

  - An 8-bit register
  - Set the usage mode for two timers
    - Set lower 4 bits for Timer 0 (Set to 0000 if not used)
    - Set upper 4 bits for Timer 1 (Set to 0000 if not used)
  - Not bit-addressable

(MSB) (LSB)

GATE	C/T	M1	M0	GATE	C/T	M1	M0
Timer 1				Timer 0			

6-a. Draw the register formats of SCON and PCON.

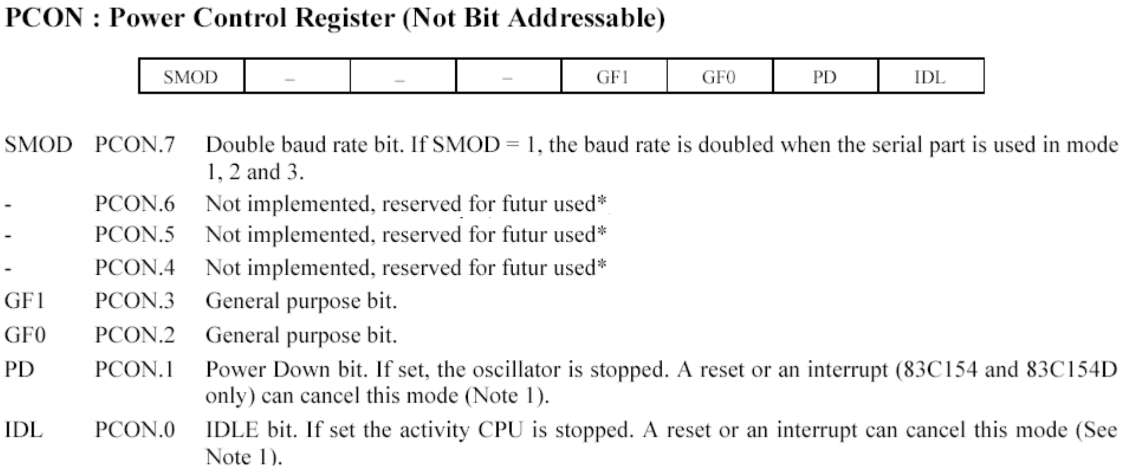


SM0	SM1	SM2	REN	TB8	RB8	TI	RI

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

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

65



PCON : Power Control Register (Not Bit Addressable)

SMOD	–	–	–	GF1	GF0	PD	IDL

SMOD	PCON.7	Double baud rate bit. If SMOD = 1, the baud rate is doubled when the serial part is used in mode 1, 2 and 3.
-	PCON.6	Not implemented, reserved for futur used*
-	PCON.5	Not implemented, reserved for futur used*
-	PCON.4	Not implemented, reserved for futur used*
GF1	PCON.3	General purpose bit.
GF0	PCON.2	General purpose bit.
PD	PCON.1	Power Down bit. If set, the oscillator is stopped. A reset or an interrupt (83C154 and 83C154D only) can cancel this mode (Note 1).
IDL	PCON.0	IDLE bit. If set the activity CPU is stopped. A reset or an interrupt can cancel this mode (See Note 1).

6-b.Write an ALP to send the string “MPMC LAB” serially with a baud rate of 9600.

Program

```
MOV TMOD , # 30h

MOV SCON , # 50h

MOV TH1 , # 08H

SETB  TR1

HARI: MOV A , # ‘M’

        ACALL  SERIAL
```

```
MOV A, # 'M'

ACALL SERIAL

MOV A , # 'L'

ACALL SERIAL

MOV A, # 'A'

ACALL SERIAL

MOV A, '  '

ACALL SERIAL

SJMP HARI

SERIAL: MOV SBUF, A

HERE: JNB TI, HERE

      CLR TI

      RET

      END
```

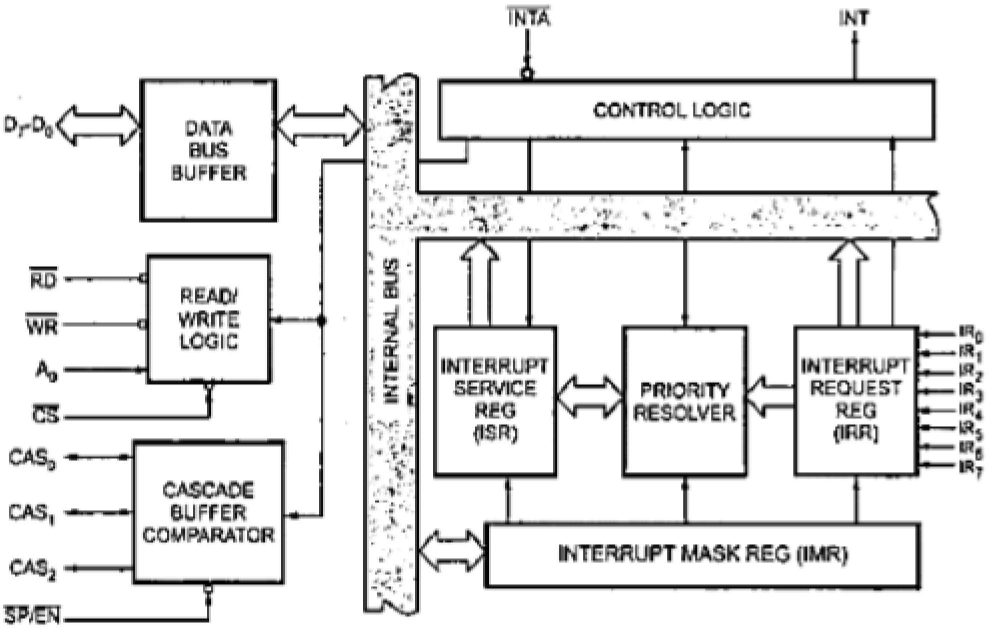
7. Draw the architecture of 8259, explain each block.

1. Fig below shows the internal block diagram of the 8259A.
2. It includes eight blocks: data bus buffer, read/write logic, control logic, three registers (IRR, ISR and IMR), priority resolver, and cascade buffer.

1. Data Bus Buffer: The data bus buffer allows the 8085 to send control words to the 8259A and read a status word from the 8259A. The 8•bit data bus buffer also allows the 8259A to send interrupt opcode and address of the interrupt service subroutine to the 8085.

2. Read/Write Logic: The RD and WR inputs control the data flow on the data bus when the device is selected by asserting its chip select (CS) input low.

3. **Control Logic:** This block has an input and an output line. If the 8259A is properly enabled, the interrupt request will cause the 8259A to assert its INT output pin high. If this pin is connected to the INTR pin of an 8085 and if the 8085 Interrupt Enable (IE) flag is set, then this high signal will cause the 8085 to respond INTR as explained earlier.



Block Diagram of 8259A

4. Interrupt Request Register (IRR): The IRR is used to store all the interrupt levels which are requesting the service. The eight interrupt inputs set corresponding bits of the Interrupt Request Register upon service request.

5. Interrupt Service Register (ISR): The Interrupt Service Register (ISR) stores all the levels that are currently being serviced.

6. Interrupt Mask Register (IMR): Interrupt Mask Register (IMR) stores the masking bits of the interrupt lines to be masked. This register can be programmed by an Operation Command Word (OCW). An interrupt which is masked by software will not be recognized and serviced even if it sets the corresponding bits in the IRR.

7. Priority Resolver: The priority resolver determines the priorities of the bits set in the IRR. The bit corresponding to the highest priority interrupt input is set in the ISR during the MITA input.

8. Cascade Buffer Comparator: This section generates control signals necessary for cascade operations. It also generates Buffer-Enable signals. As stated earlier, the 8259 can be cascaded with other 825% in order to expand the interrupt handling capacity to sixty-four levels. In such a case, the former is called a master, and the latter are called slaves. The 8259 can be set up as a master or a slave by the SP/ER pin.

9. CAS_0 - CAS_2: For a master 8259, the CAS_0 - CAS_2pins are output pins, and for slave 8259s, these are input pins. When the 8259 is a master (that is, when it accepts interrupt requests from other 8259s), the CALL opcode is generated by the Master in response to the first INTA. The vector address must be released by the slave 8259. The master sends an identification code of three-bits to select one out of the eight possible slave

8259s on the CAS_0 - CAS_2lines. The slave 8259s accept these three signals as inputs (on their CAS_0 - CAS_2 pins) and compare the code sent by the master with the codes assigned to them during initialization. The slave thus selected (which had originally placed an interrupt request to the master 8259) then puts the address of the interrupt service routine during the second and third INTA pulses from the CPU.

10. SP ER (Slave Program /Enable Buffer): The SP/EN signal is tied high for the master. However it is grounded for the slave. In large systems where buffers are used to drive the data bus, the data sent by the 8259 in response to INTA cannot be accessed by the CPU (due to the data bus buffer being disabled). If an 8259 is used in the buffered mode (buffered or non-buffered modes of operation can be specified at the time of initializing the 8259), the SP/ER pin is used as an output which can be used to enable the system data bus buffer whenever the data bus outputs of 8259 are enabled (i.e. when it is ready to send data). Thus, in non-buffered mode, the SP/EN pin of an 8259 is used to specify whether the 8259 is to operate as a master or as a slave, and in the buffered mode, the SP / EN pin is used as an output to enable the data bus buffer of the system.

9. Explain internal memory organization of (RAM,ROM) of 8051 microcontroller.

Ans: **8051 Memory Organisation**

Internal ROM

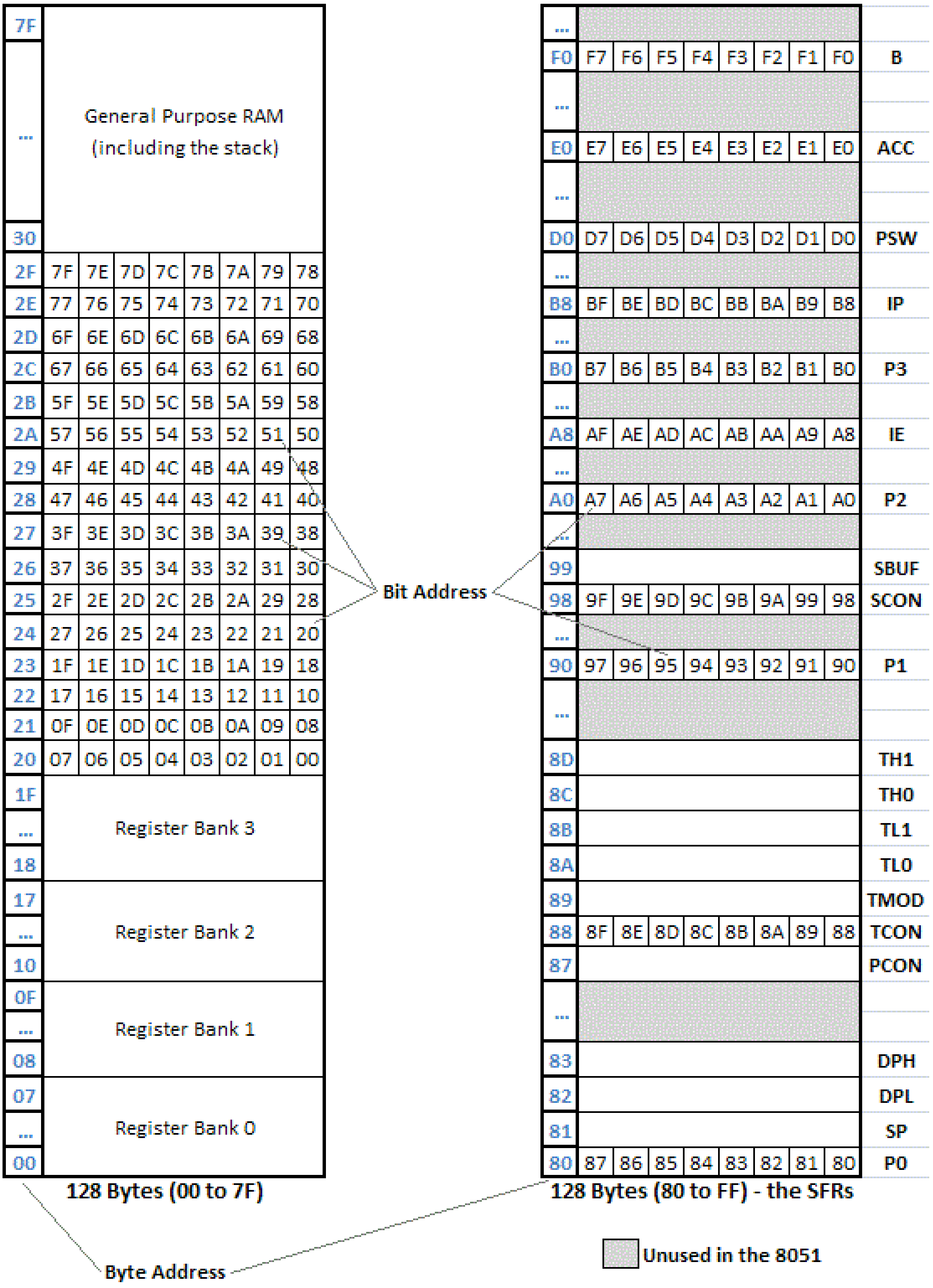
The 8051 has 4K (4096 locations) of on-chip ROM. This is used for storing the system program. $2^{12} = 4096$, therefore the internal ROM address bus is 12 bits wide and internal ROM locations go from 000H to FFFH.

Internal RAM

There are 256 bytes of internal RAM on the 8051. $2^{8} = 256$, therefore the internal RAM address bus is 8 bits wide and internal RAM locations go from 00H to FFH.

The first 128 locations (00H to 7FH) of internal RAM are used by the programmer for storing data while the second 128 locations (80H to FFH) are the Special Function Registers (SFRs) which we will deal with later.

The diagram below is a summary of the 8051 on-chip RAM.



7F	General Purpose RAM (including the stack)							
...								
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	Register Bank 3							
...								
18								
17	Register Bank 2							
...								
10								
0F	Register Bank 1							
...								
08								
07	Register Bank 0							
...								
00								

128 Bytes (00 to 7F)

...	▨								
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	D1	D0	PSW
...	▨								
B8	BF	BE	BD	BC	BB	BA	B9	B8	IP
...	▨								
B0	B7	B6	B5	B4	B3	B2	B1	B0	P3
...	▨								
A8	AF	AE	AD	AC	AB	AA	A9	A8	IE
...	▨								
A0	A7	A6	A5	A4	A3	A2	A1	A0	P2
...	▨								
99									SBUF
98	9F	9E	9D	9C	9B	9A	99	98	SCON
...	▨								
90	97	96	95	94	93	92	91	90	P1
...	▨								
8D									TH1
8C									TH0
8B									TL1
8A									TL0
89									TMOD
88	8F	8E	8D	8C	8B	8A	89	88	TCON
87									PCON
...	▨								
83									DPH
82									DPL
81									SP
80	87	86	85	84	83	82	81	80	P0

128 Bytes (80 to FF) - the SFRs

▨ Unused in the 8051

Register Banks

There are four register banks from 00H to 1FH. On power-up, registers R0 to R7 are located at 00H to 07H. However, this can be changed so that the register set points to any of the other three banks (if you change to Bank 2, for example, R0 to R7 is now located at 10H to 17H).

Bit-addressable Locations

The 8051 contains 210 bit-addressable locations of which 128 are at locations 20H to 2FH while the rest are in the SFRs. Each of the 128 bits from 20H to 2FH have a unique number (address) attached to them, as shown in the table above. The 8051 instruction set allows you to set or reset any single bit in this section of RAM.

With the general purpose RAM from 30H to 7FH and the register banks from 00H to 1FH, you may only read or write a full byte (8 bits) at these locations.

However, with bit-addressable RAM (20H to 2FH) you can read or write any single bit in this region by using the unique address for that bit. We will later see that this is a very powerful feature.

Special Function Registers (SFRs)

Locations 80H to FFH contain the special function registers. As you can see from the diagram above, not all locations are used by the 8051 (eleven locations are blank). These extra locations are used by other family members (8052, etc.) for the extra features these microcontrollers possess.

Also note that not all SFRs are bit-addressable. Those that are have a unique address for each bit.

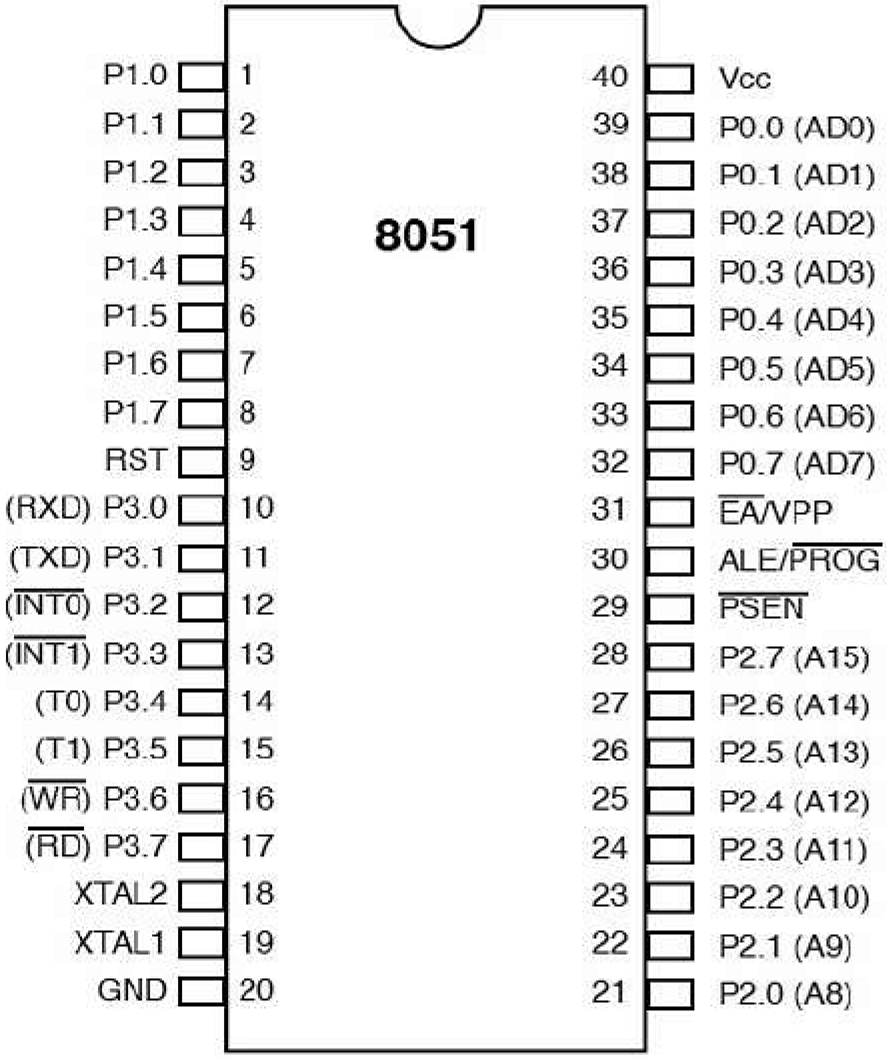
We will deal with each of the SFRs as we progress through the course, but for the moment you should take note of the accumulator (ACC) at address E0H and the four port registers at addresses 80H for P0, 90h for P1, A0 for P2 and B0 for P3.

We will later see how easy this makes reading from and writing to any of the four ports.

The Missing Registers

The two registers not shown in the table above are the instruction register and the program counter. These two very important registers are not placed in RAM because it is not necessary to make them directly accessible to the programmer. The instruction register gets its data from the execution cycle while the program counter is manipulated by the operation of these instructions.

10-a. Draw the pin diagram of 8051.



PORT 0:

The structure of a Port-0 pin is shown in fig 6.It has 8 pins (P0.0-P0.7).

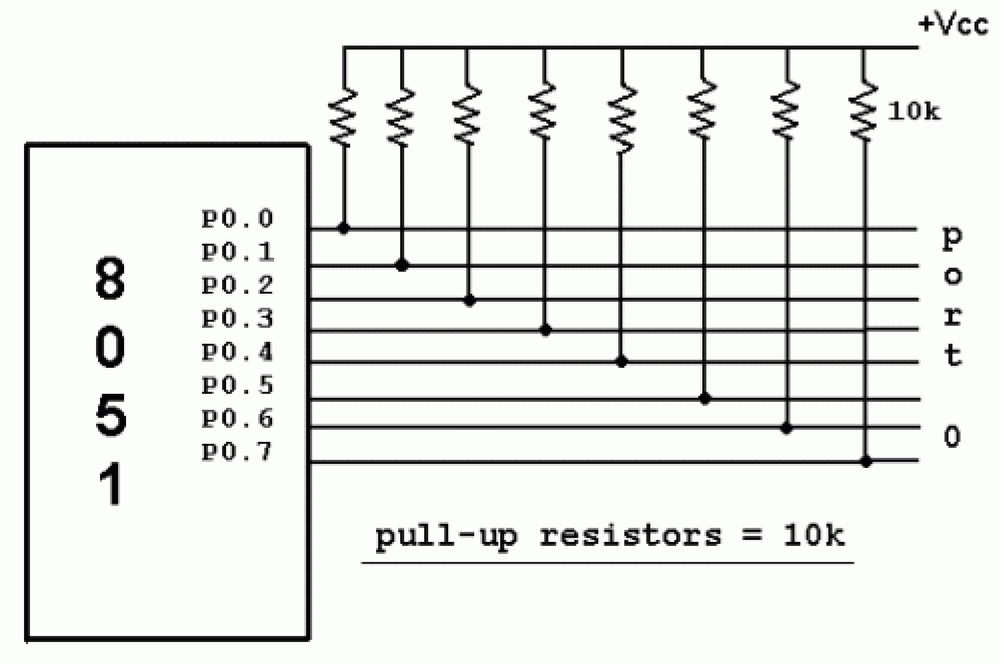
Port-0 can be used as a normal bidirectional I/O port or it can be used for address/data interfacing for accessing external memory. When control is '1', the port is used for address/data interfacing. When the control is '0', the port can be used as a bidirectional I/O port.

PORT 0 as an Input Port

Let us assume that control is '0'. When the port is used as an input port, '1' is written to the latch. In this situation both the output MOSFETs are 'off'. Hence the output pin have floats hence whatever data written on pin is directly read by read pin.

PORT 0 as an Output Port

Suppose we want to write 1 on pin of Port 0, a '1' written to the latch which turns 'off' the lower FET while due to '0' control signal upper FET also turns off as shown in fig. above. Here we wants logic '1' on pin but we getting floating value so to convert that floating value into logic '1' we need to connect the pull up resistor parallel to upper FET . This is the reason why we needed to connect pull up resistor to port 0 when we want to initialize port 0 as an output port .



If we want to write '0' on pin of port 0, when '0' is written to the latch, the pin is pulled down by the lower FET. Hence the output becomes zero.

When the control is '1', address/data bus controls the output driver FETs. If the address/data bus (internal) is '0', the upper FET is 'off' and the lower FET is 'on'. The output becomes '0'. If the address/data bus is '1', the upper FET is 'on' and the lower FET is 'off'. Hence the output is '1'. Hence for normal address/data interfacing (for external memory access) no pull-up resistors are required.Port-0 latch is written to with 1's when used for external memory access.

PORT 1:

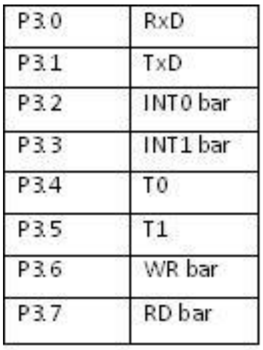
Port-1 dedicated only for I/O interfacing. When used as output port, not needed to connect additional pull-up resistor like port 0. It have provided internally pull-up resistor as shown in fig. below. The pin is pulled up or down through internal pull-up when we want to initialize as an output port. To use port-1 as input port, '1' has to be written to the latch. In this input mode when '1' is written to the pin by the external device then it read fine. But when '0' is written to the pin by the external device then the external source must sink current due to internal pull-up. If the external device is not able to sink the current the pin voltage may rise, leading to a possible wrong reading.

PORT 2:

Port-2 we use for higher external address byte or a normal input/output port. The I/O operation is similar to Port-1. Port-2 latch remains stable when Port-2 pin are used for external memory access. Here again due to internal pull-up there is limited current driving capability.

PORT 3:

Following are the alternate functions of port 3:



P3.0	RxD
P3.1	TxD
P3.2	INT0 bar
P3.3	INT1 bar
P3.4	T0
P3.5	T1
P3.6	WR bar
P3.7	RD bar

It work as an IO port same like Port 2, Only alternate function of port 3 makes its architecture different than other ports.