RSET
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

Module 5 – Microcontrollers

100003/CS500D

Microprocessors and Microcontrollers

Syllabus

- 8051 Architecture- Register Organization- Memory and I/O addressing- Interrupts and Stack
- 8051 Addressing Modes
- 8051 Instruction Set- data transfer instructions, arithmetic instructions, logical instructions, Boolean instructions, control transfer instructions
- Simple programs

Course Outcomes

- **CO1:** Illustrate the architecture, modes of operation and addressing modes of microprocessors.
- **CO2:** Develop 8086 assembly language programs.
- **CO3:** Demonstrate interrupts, its handling and programming in 8086.
- **CO4:** Illustrate how different peripherals (8255,8254,8257) and memory are interfaced with microprocessors.
- **CO5: Outline features of microcontrollers and develop low level programs.**

Microcontrollers

- Microcontroller is a single chip micro computer made through VLSI fabrication
- A microcontroller is also called an embedded controller because the microcontroller and its support circuits are often built into, or embedded in, the devices they control
- A microcontroller is available in different word lengths like microprocessors. 4 bit, 8 bit, 16 bit, 32 bit, 64 bit and 128 bit microcontrollers are available today

Microcontroller vs Microprocessors

Microprocessors	Microcontroller
Only have CPU in the chip.	Have RAM, ROM and other peripherals along with the CPU or processor.
Used in an application where the task is not predefined	Designed for a specific task and once the program is embed on MCU chip, it can't be altered easily
Used where intensive processing is required like laptops, computers, mobiles etc.	Used in many electronic appliances like washing machine, microwave oven, timer, etc.
The microprocessors are run at higher clock speeds.	High clock speed is not required.
Amount of memory required for the microprocessor is very large.	Amount of memory required is quite less compared to microprocessors.
The cost of the microprocessor is high compared to the microcontroller.	It is cheaper.
The power consumption for the microprocessor is high.	The power consumption for the microcontroller is less.
Since memory and IO has to be connected externally the circuit will be large.	Since memory and IO is present internally the circuit will be small.

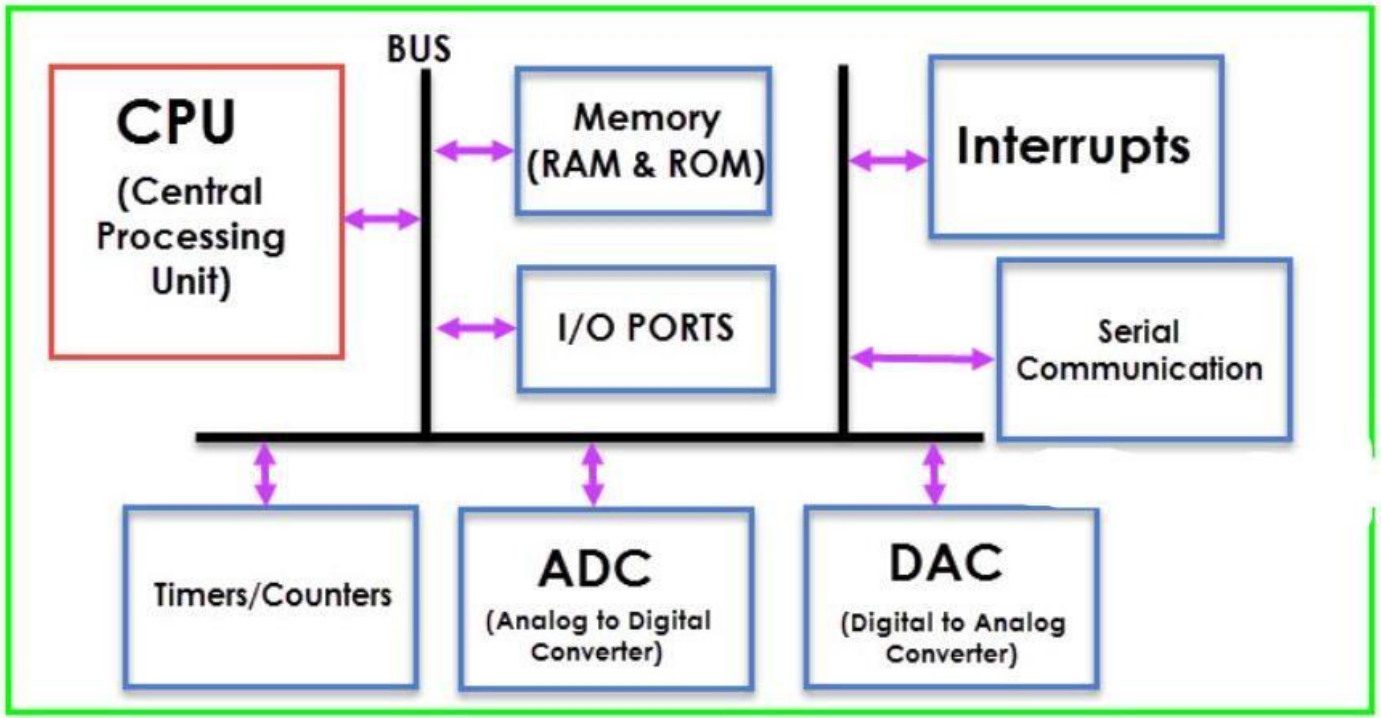
Advantages of Microcontrollers

- Due to the integration of all functional blocks in a single chip microcontroller IC, the size of the control board and power consumption are reduced, system reliability increased and provides more flexibility.
- The microcontroller is able to interface additional RAM, ROM and I/O ports for additional peripherals & memory and better software security.
- Once microcontrollers are programmed then they can be reprogrammed with some difficulty, so reusability exist.
- At the same time many tasks can be performed, so human effort can be saved.
- Easy trouble shooting & maintenance.
- Integrated circuits, such as the microcontroller, are much more dependable than relays. Before microcontrollers, control circuitry relied on many electromechanical relays and timers to control the system.

Disadvantages of Microcontrollers

- The microcontroller cannot interface high power devices directly.
- It has more complex structure as compared to microprocessor.
- It only performed limited number of executions simultaneously.
- It is generally used in micro equipment.

Elements of a Microcontroller (MCU)



Elements of a Microcontroller (MCU)

- Central processing unit (CPU)
- Random Access Memory (RAM)
- Read Only Memory (ROM)
- Input/output ports
- Timers and Counters
- Interrupt Controls
- Analog to digital converters
- Digital to analog converters
- Serial interfacing ports
- Oscillatory circuits

Applications of Microcontrollers

- Automatic Process Control
- Instrumentation Applications
- Embedded devices
- Application of Microcontroller in Day to Day Life Devices:
- Light sensing & controlling devices
- Temperature sensing and controlling devices
- Fire detection & safety devices

8051 Resources

- 8 bit microcontroller originally developed by Intel in 1980.
- High-performance CMOS Technology.
- Contains Total 40 pins.

8051 Resources

- Address bus is of 16 bit & data bus is of 8 bit.
- 4K bytes internal ROM (program).
- 128 bytes internal RAM (data).
- Four 8-bit I/O ports.
- Two 16-bit timers.
- Serial interface Communication.

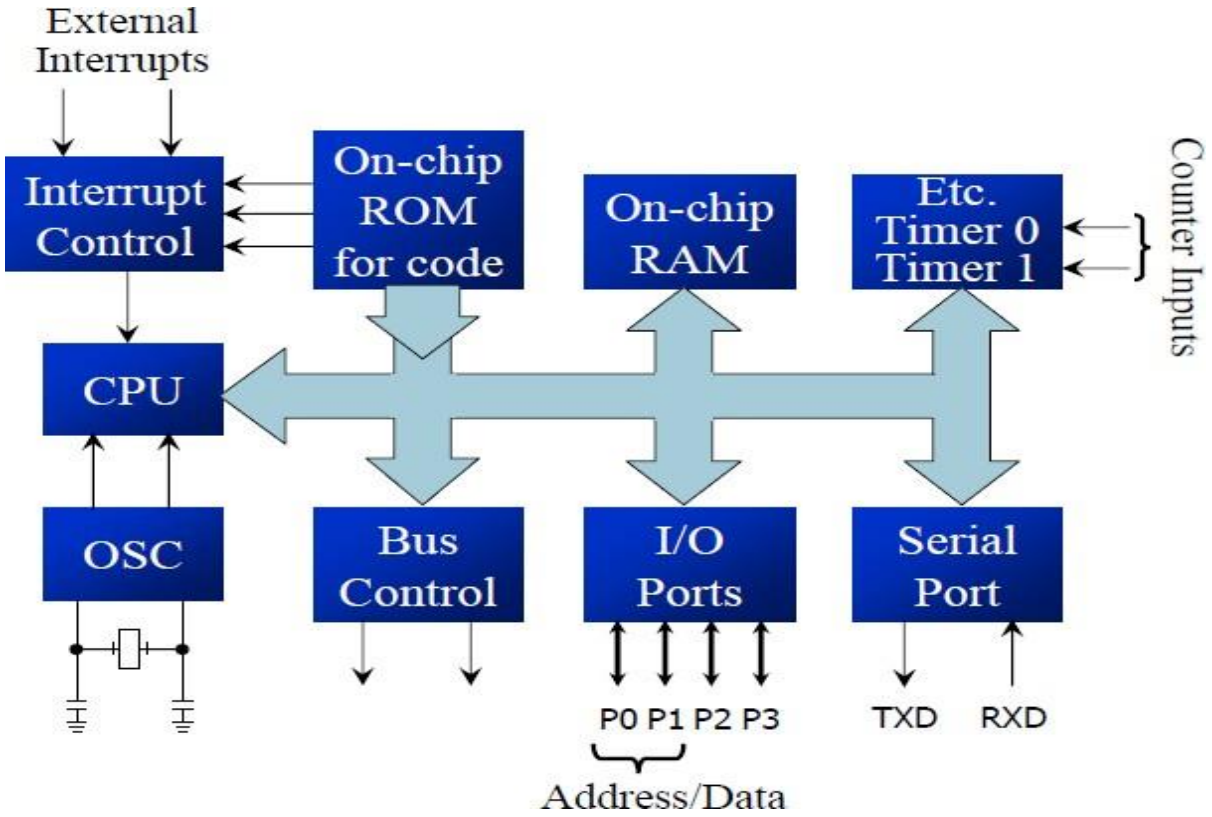
8051 Resources

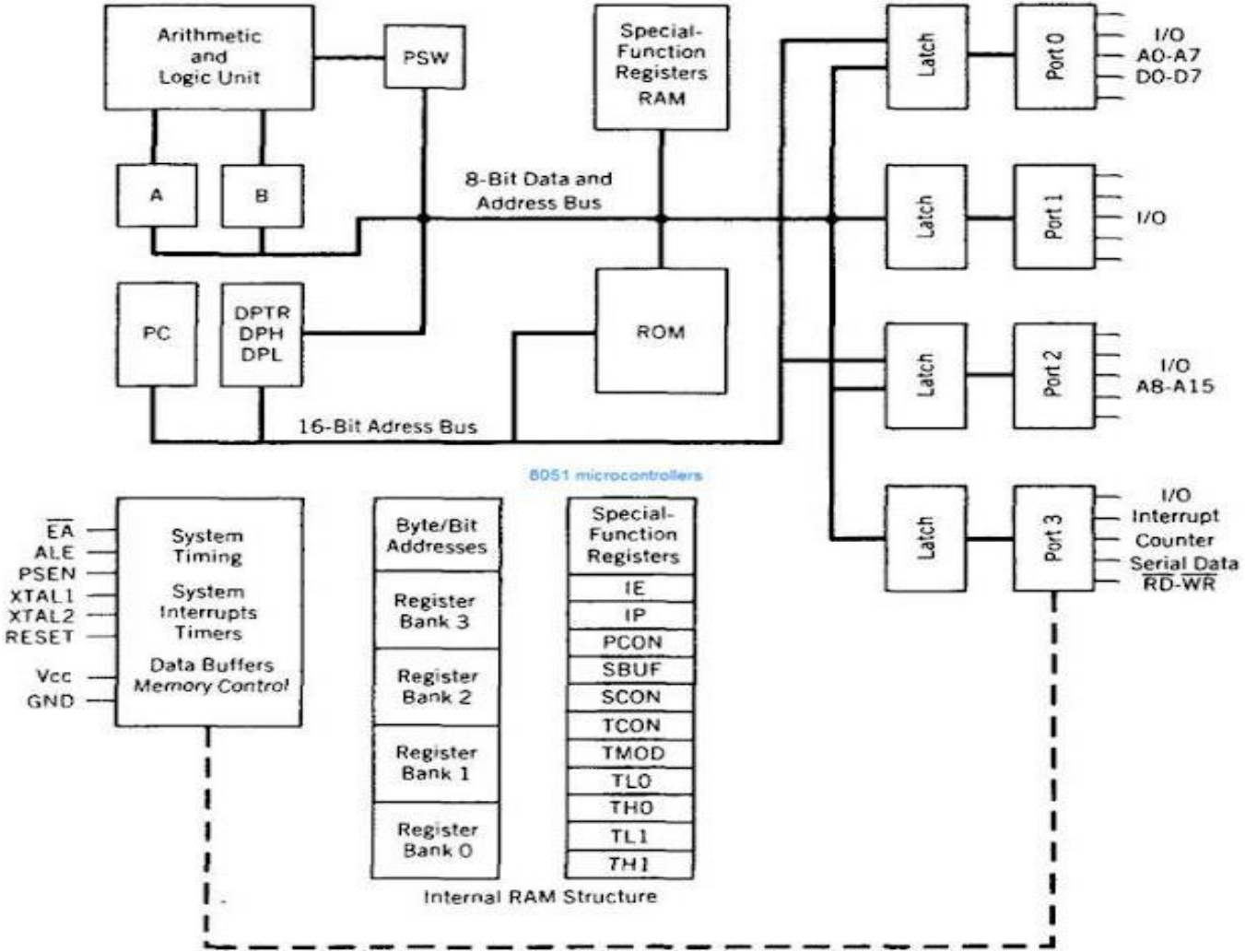
- 64K external code & data memory space.
- 128 bit-addressable locations.
- Internal memory consists of on-chip ROM and on-chip data RAM.
- 8051 implements a separate memory space for programs (code) and data.
- Operating frequency is 24MHz-33MHz.

8051 Resources

- +5V Regulated DC power supply is required to operate
- RAM, ROM, I/O ports, one serial port and timers are all on-chip.
- 6-interrupts (2 are external with 2 priority levels).
- Low-power Idle and Power-down Modes.
- Full duplex UART.
- 8051 has 21 special function registers (SFRs).

8051 Block Diagram

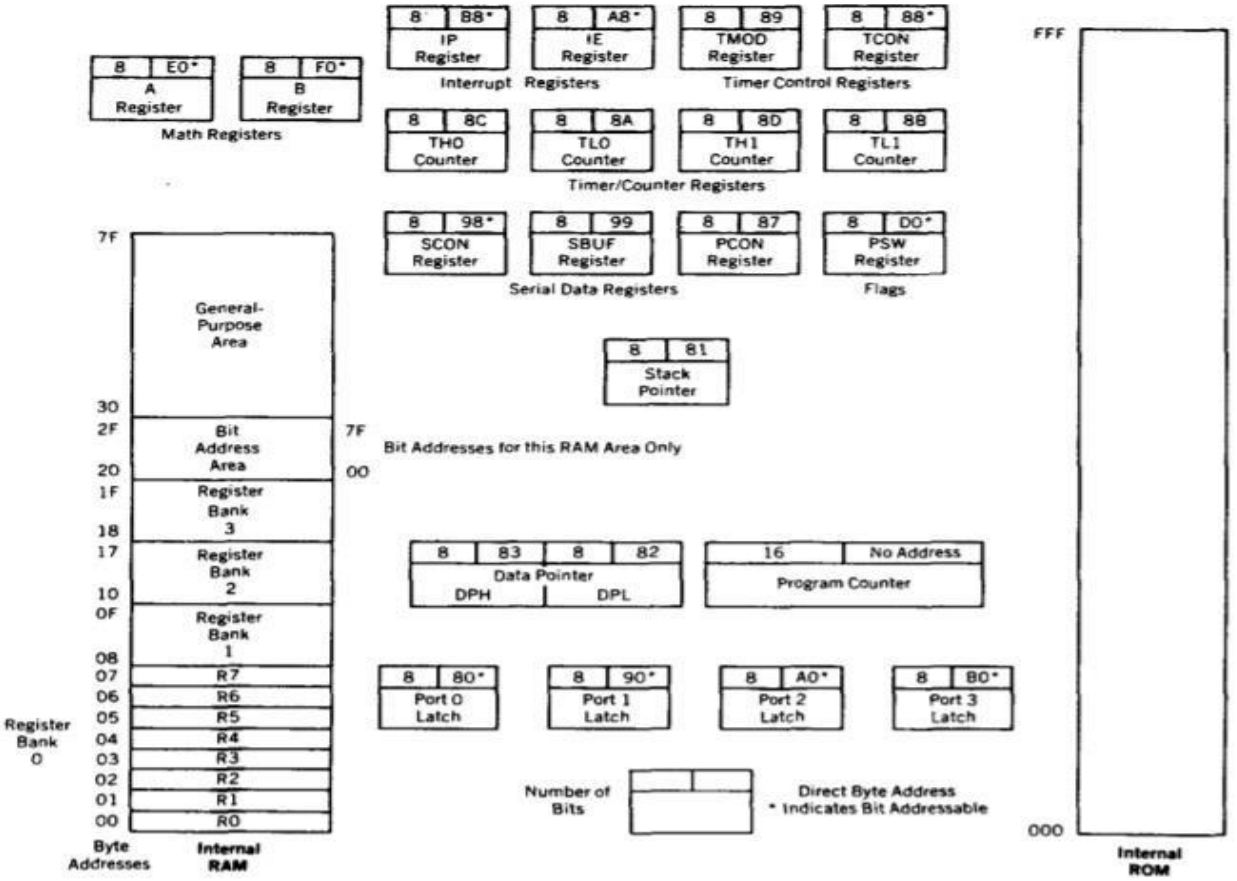



Arithmetic and Logic Unit
PSW
A
B
PC
DPTR
DPH
DPL
Special-Function Registers
RAM
ROM
8-Bit Data and Address Bus
16-Bit Adress Bus
Latch
Port 0
I/O
A0-A7
D0-D7
Port 1
I/O
Port 2
I/O
A8-A15
Port 3
I/O
Interrupt
Counter
Serial Data
RD-WR
8051 microcontrollers
EA
ALE
PSEN
XTAL1
XTAL2
RESET
Vcc
GND
System Timing
System Interrupts Timers
Data Buffers
Memory Control
Byte/Bit Addresses
Register Bank 3
Register Bank 2
Register Bank 1
Register Bank 0
Special-Function Registers
IE
IP
PCON
SBUF
SCON
TCON
TMOD
TL0
TH0
TL1
TH1
Internal RAM Structure

8051 Architecture consists of:

- 8 bit CPU with registers A (accumulators) 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 EPRROM(4K)
- Internal RAM of 128 bytes:
  - 4 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: T0 and T1
- Full duplex serial data receiver/transmitter: SBUF
- Control registers: TCON, TMOD, SCON, PCON, IP and IE
- 2 external and 3 internal interrupt sources
- Oscillator and Clock circuits

8051 Programming Model



8051 Oscillator Circuit

- The circuitry that generates clock pulses by which all internal operations are synchronized.
- The 8051 has an on-chip oscillator circuit usually runs around 12MHz.
- Pins XTAL1 & XTAL2 have been used to connect to the external oscillator.
- It employs a quartz crystal and capacitors.
- The frequency of the crystal oscillator is the basic internal clock frequency of the microcontroller.

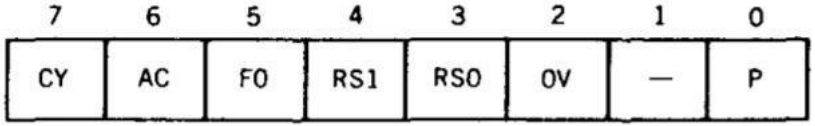
Program Counter and Data Pointer

- They are both 16 bitregisters.
- Each is to hold the address of a byte in memory
- PC contains the address of the next instruction to be executed.
- DPTR is made up of two 8 bit register DPH and DPL.
- DPTR contains the address of data that has to be accessed.

A and B registers

- A register is used for many operations such as addition, subtraction, integer multiplication and division, and Boolean bit manipulations.
- A register is also used for all data transfers between 8051 and any external memory.
- B register or accumulator B is used along with the accumulator A for multiply and divide operations.
- MUL A,B : multiplies 8 bit unsigned values in A and B and leaves the 16 bit result in A (low byte) and B (high byte).
- DIV A,B : divides A by B, leaving the integer result in A and remainder in B.
- B register is bit-addressable.

Program Status Word (PSW)



THE PROGRAM STATUS WORD (PSW) SPECIAL FUNCTION REGISTER

Bit	Symbol	Function
7	CY	Carry flag; used in arithmetic, JUMP, ROTATE, and BOOLEAN instructions
6	AC	Auxilliary carry flag; used for BCD arithmetic
5	F0	User flag 0
4	RS1	Register bank select bit 1
3	RS0	Register bank select bit 0

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

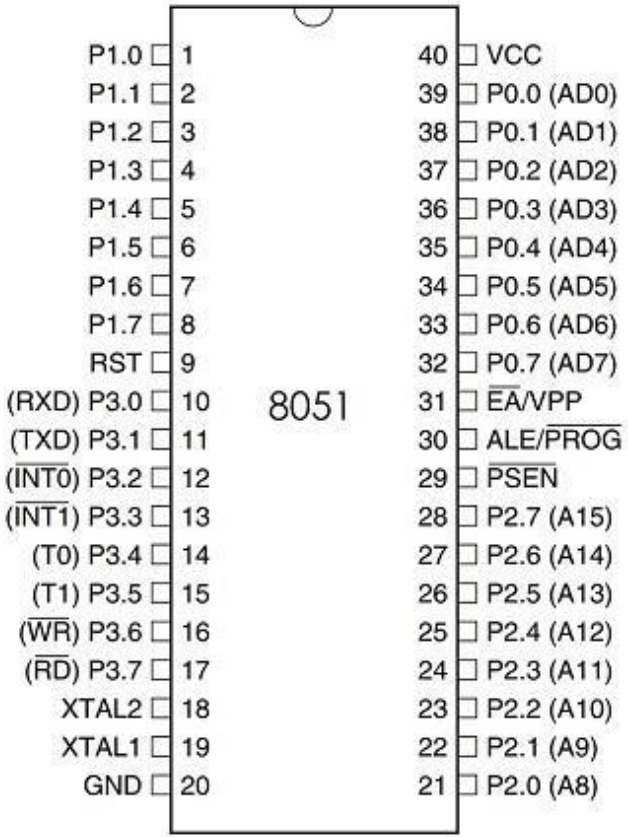
2	OV	Overflow flag; used in arithmetic instructions
1	—	Reserved for future use
0	P	Parity flag; shows parity of register A: 1 = Odd Parity

Bit addressable as PSW.0 to PSW.7

Stack Pointer

- Stack pointer (SP) is an 8-bit register at address 81H.
- It contains the address of the data item currently on top of the stack.
- Stack operations include pushing data on the stack and popping data off the stack.
- Pushing increments SP before writing the data
- Popping from the stack reads the data and decrements the SP
- 8051 stack is kept in the internal RAM
- Depending on the initial value of the SP, stack can have different sizes
- Example: MOV SP,#5FH
- On 8051 this would limit the stack to 32 bytes since the uppermost address of on chip RAM is 7FH.

PIN Description of 8051



8051 PIN Layout

- The 8051 is a 40 pin, Dual In Line(DIP) package IC.
- Out of these 40 pins, 32 are used for I/O. (Four 8-bit ports: P0, P1, P2, and P4).
- 24 of these are dual purpose, i.e. they can operate as I/O or a control line or as part of address or data bus.
- Remaining 8 pins are Vcc, GND, XTAL1, XTAL2, RST, PSEN, EA, ALE

PINs of 8051

Pin	Description
VCC (Pin 40)	Vcc provides supply voltage to the chip. The voltage source is +5V.
GND (Pin 20)	Ground
XTAL1 and XTAL2（Pins 19,18）	These 2 pins provide external clock
RST (Pin 9): reset	It is an input pin and is active high. Upon applying a high pulse to RST, the microcontroller will reset and all values in registers will be lost.
ALE (pin 30): Address Latch Enable	It is an output pin and is active high. 8051 port 0 provides both address and data. The ALE pin is used for de-multiplexing the address and data by connecting to the G pin of the 74LS373 latch.

PINs of 8051

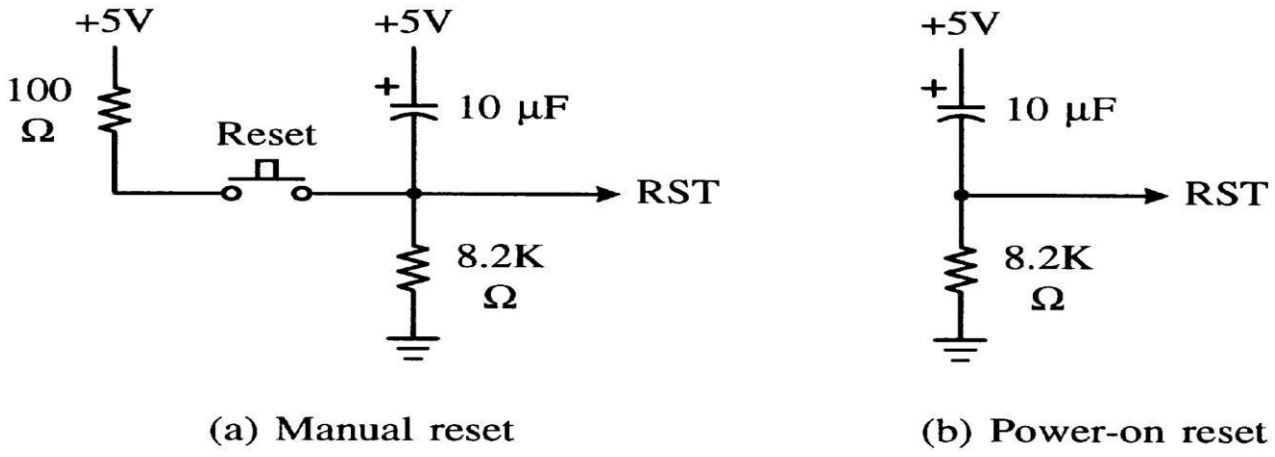
Pin	Description
$\overline{\text{PSEN}}$ (Pin 29: Program Store Enable)	For enabling the external program memory. It will be connected to the external ROM memory chip. This pin usually connects to an EPROM's Output Enable (OE) pin. This is a logic low pin and during a fetch stage involving an instruction stored in external memory, the pin will be pulsed 'LOW'.
EA (Pin 31: External Access)	If you need to connect to external ROM then this pin must be tied LOW (0V). This pin must be tied high (+5V) if the programs executes from internal ROM EA pin is for forcing the controller to use the external program memory. When it is connected to ground , controller will fetch data from external memory.

PINs of 8051

Pin	Description
I/O Ports	The four ports P0, P1, P2, and P3. Each port uses 8 pins. All I/O pins are bi-directional.
Port 0 - P0 (pins 32-39)	P0.0 (LSB) ~ P0.7 (MSB) - Also labelled as AD0 – AD7 Dual Purpose Port. Works as I/O or address lines Lower order address lines / Data bus
Port 1 - P1 (pins 1-8)	P1.0 (LSB) ~ P1.7 (MSB) – Dedicated I/O For interfacing with external devices likes switches, LED
Port 2 – P2 (pins 21-28)	P2.0 (LSB) ~ P2.7 (MSB) – Also labelled as A8 – A15 Dual Purpose Port. Works as I/O or address lines Higher order address lines
Port 3 – P3 (pins 10-17)	P3.0 (LSB) ~ P3.7 (MSB) Dual purpose, rarely used for I/O. Used for functions related to special features of 8051

Reset Operation of 8051

- To reset the 8051, the RST pin must be held high for at least 2 machine cycles.
- This can be achieved upon power–up using an RC network.
- This is pin 9 of the IC and is used as the master reset for the 8051.
- In order for the 8051 to recognize that a reset has occurred, this pin must be brought HIGH for at least two machine cycles.
- During normal operation, this pin must be at logic LOW.



(a) Manual reset

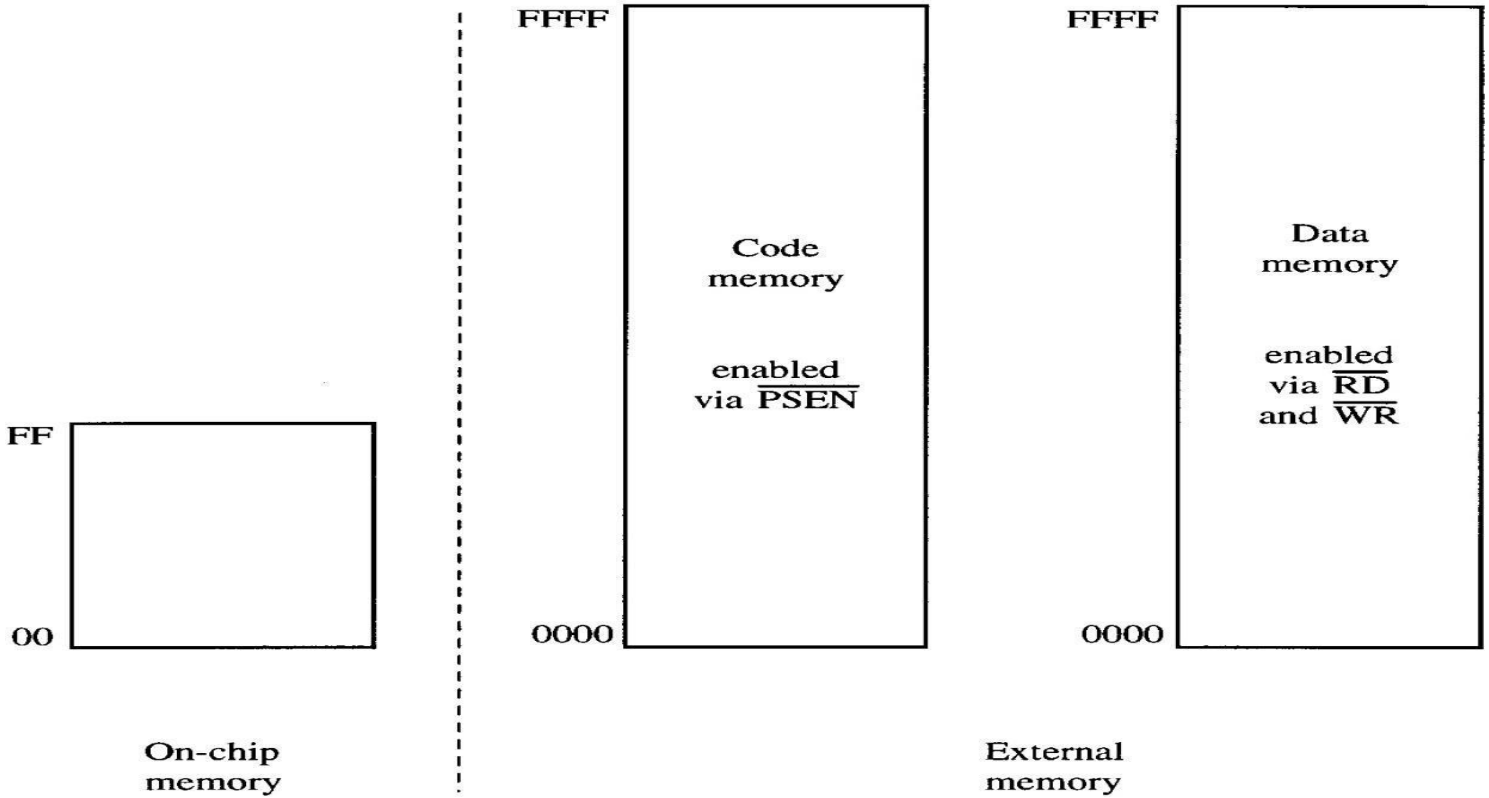
(b) Power-on reset

8051 Memory Structure

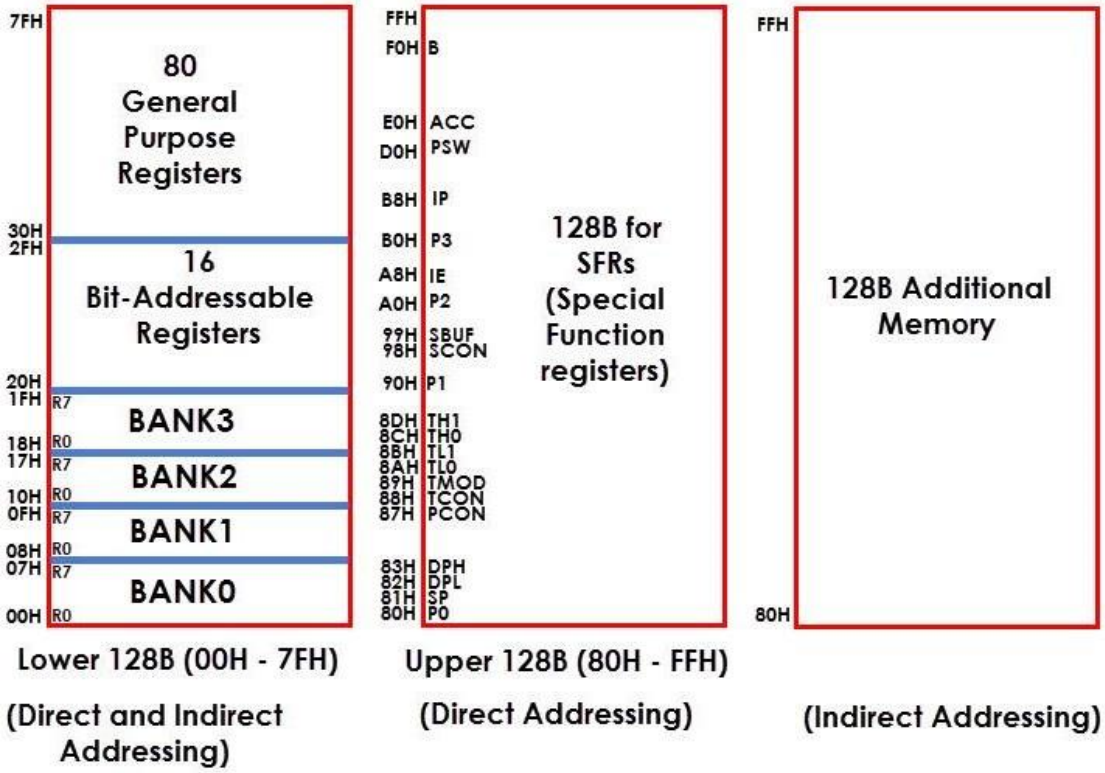
- While most microprocessors implement a shared memory space for data and code (programs), microcontrollers has limited memory and the program is usually stored in ROM.
- In the 8051, both code and data may be internal but they are stored in separate memories,

  namely the internal ROM and RAM.
- Expandable to a max of 64K code memory and 64K data memory using external memory.
- **Internal Memory**
  - Consists of on-chip ROM and on-chip data RAM.
  - On-chip RAM contains a rich arrangement of general purpose storage, bit addressable storage, register banks, and special function registers.
  - In the 8051, the registers and input/output ports are memory mapped and accessible like any other memory location.
  - In the 8051, the stack resides within the internal RAM, rather than in external RAM.

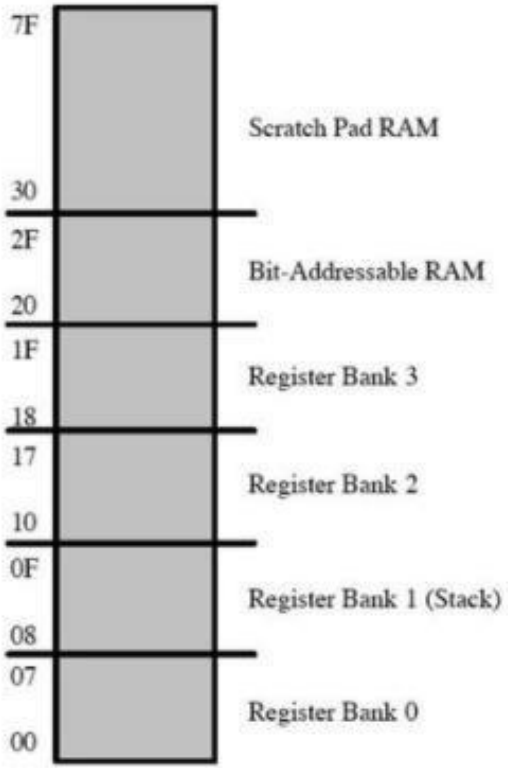
8051 Memory Organization



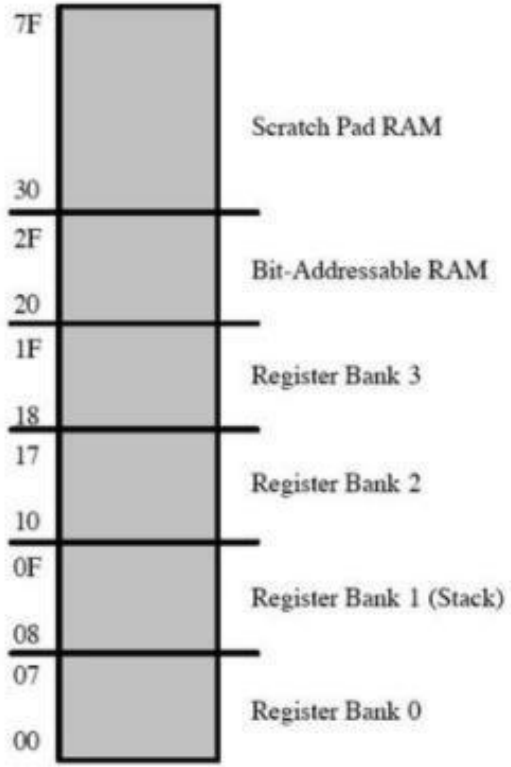
8051 on chip Memory Allocation



RAM Memory Allocation

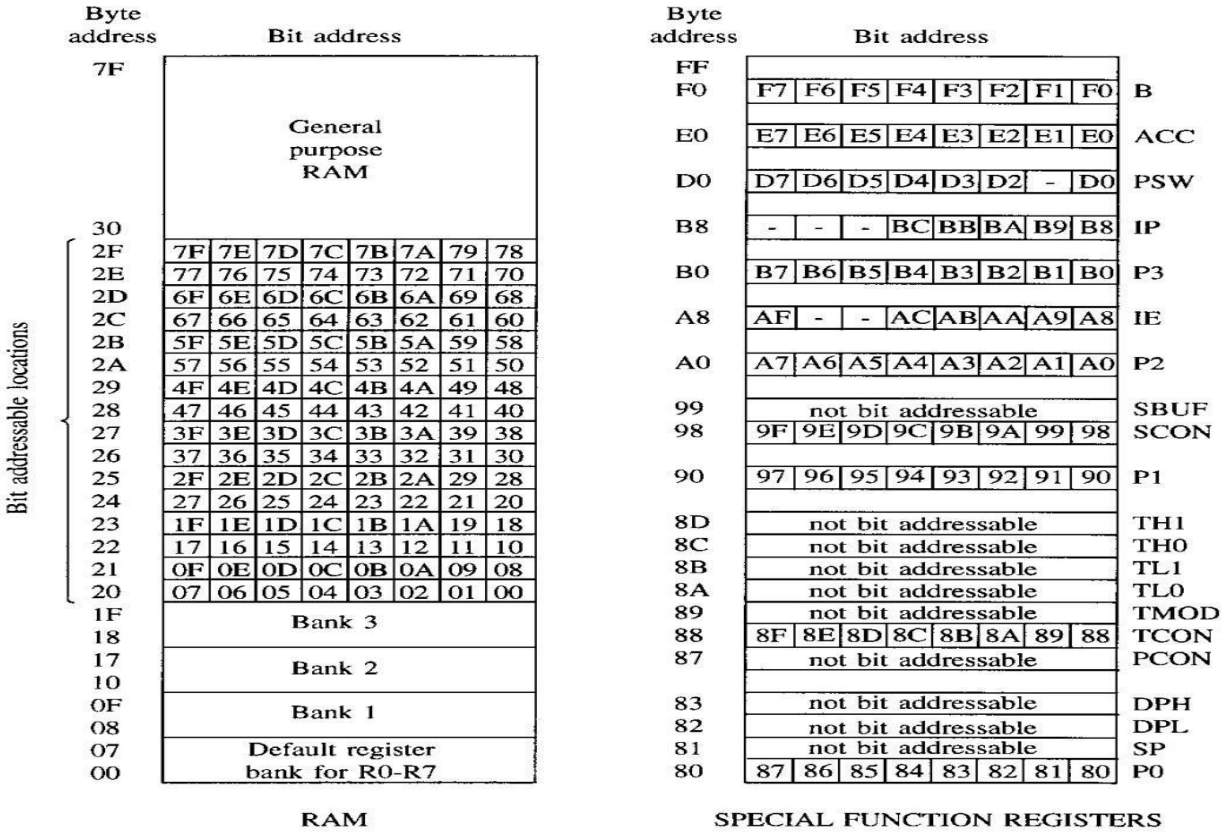


Stack in 8051



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

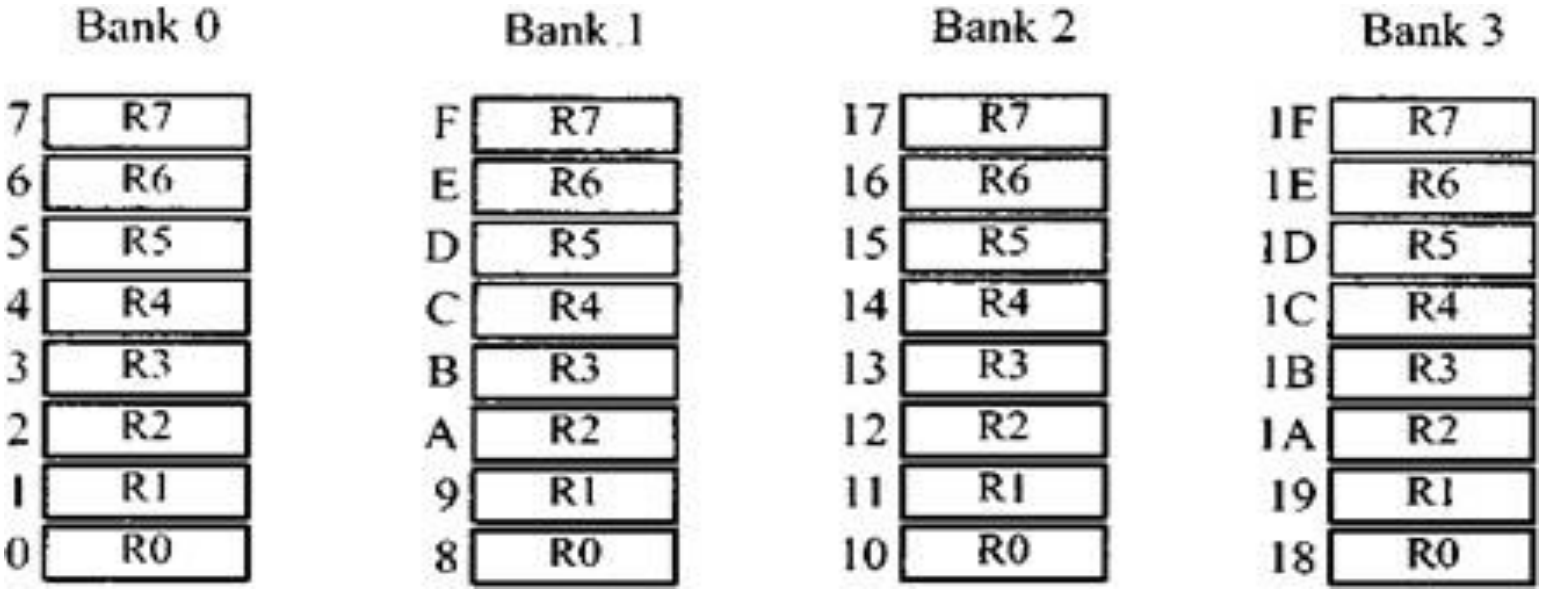
Summary of 8051 on chip data memory



Register Banks in 8051

- 32 bytes of RAM are set aside for the register banks and stack.
- These 32 bytes are divided into 4 banks of registers in which each bank has 8 registers, RO – R7.
- RAM locations from 0 to 7 are set aside for bank 0 of R0 – R7 where R0 is RAM location 00H, R1 is RAM location 01H, R2 is location 02H, and so on, until memory location 07H, which belongs to R7 of bank 0.
- The second bank of registers R0 – R7 starts at RAM location 08H and goes to location 0FH.
- The third bank of R0 – R7 starts at memory location 10H and goes to location 17H.
- Finally, RAM locations 18H to 1FH are set aside for the fourth bank of R0 – R7.

Register Banks in 8051 and their RAM Address



Special Function Registers in 8051

- 8051 has 21 special function registers (SFRs) at the top of internal RAM from address 80H to FFH.
- Most of the addresses from 80H to FFH are not defined, except for 21 of them.
- Some SFR's are both bit-addressable and byte addressable, depending on the instruction accessing the register.
- This area consists of a series of memory-mapped ports and registers.
- All 8051 CPU registers, I/O ports, timers and other architecture components are accessible in 8051 through SFRs

Special Function Registers in 8051

Name	Function
A	Accumulator Register
B	Arithmetic
DPH	Data Pointer High
DPL	Data Pointer Low
IE	Interrupt Enable
IP	Interrupt Priority
P0	I/O Port Latch
P1	I/O Port Latch
P2	I/O Port Latch
P3	I/O Port Latch
PCON	Power Control
PSW	Program Status Word

Name	Function
SCON	Serial Port Control
SBUF	Serial Port data buffer
SP	Stack Pointer
TMOD	Timer/Counter mode control
TCON	Timer/Counter control
TL0	Timer0 lower byte
TH0	Timer0 higher byte
TL1	Timer1 lower byte
TH1	Timer1 higher byte

Interrupts of 8051

- The 8051 microcontroller can recognize five different events that cause the main program to
  interrupt fromthe normal execution.

- These five sources of interrupts in 8051are:

  - **Timer 0 overflow interrupt- TF0**
  - **Timer 1 overflow interrupt- TF1**
  - **External hardware interrupt- INT0**
  - **External hardware interrupt- INT1**
  - **Serial communication interrupt- RI/TI**

- The Timer and Serial interrupts are internally generated by the microcontroller, whereas the external interrupts are generated by additional interfacing devices or switches that are externally connected to the microcontroller.

- These external interrupts can be edge triggered or level triggered.

- When an interrupt occurs, the microcontroller executes the interrupt service routine so that
  memory location corresponds to the interrupt that enables it.

- The Interrupt corresponding to the memory location is given in the interrupt vector table.

Interrupt Vector Table of 8051

Interrupt	Source	Vector Address
System Reset	RST	0000H
External 0	IE0	0003H
Timer 0	TF0	000BH
External 1	IE1	0013H
Timer 1	TF1	001BH
Serial Port	RI or TI	0023H

- **Reset:**
  - When the reset pin is activated, the 8051 jumps to address location 0000.
  - This is the power-up reset.
- **Timer Interrupts T0 and T1:**
  - Two interrupts are set aside for the timers: one for Timer 0 and one for Timer 1.
  - Memory locations 000BH and 001BH in the interrupt vector table belong to Timer 0 and Timer 1, respectively.

- **External hardware interrupts:**
  - Two interrupts are set aside for hardware external hardware interrupts.
  - Pin numbers 12 (P3.2) and 13 (P3.3) in port 3 are for the external hardware interrupts INT0 and INT1, respectively.
  - These external interrupts are also referred to as EX1 and EX2. Memory locations 0003H and 0013H in the interrupt vector table are assigned to INT0 and INT1, respectively.
- **Serial Communication Interrupt:**
  - Serial communication has a single interrupt that belongs to both receive and transmit.
  - The interrupt vector table location 0023H belongs to this interrupt.

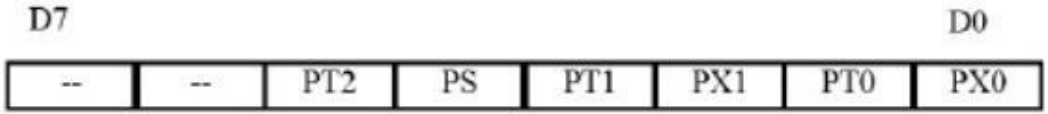
- **Interrupt Enable (IE) Register:**
  - This register is responsible for enabling and disabling the interrupt.
  - It is a bit addressable register in which bit D7 (EA) must be set to 1 for enabling interrupts.
  - If it is set to 0 all interrupts will be disabled.
  - The corresponding bit in this register enables particular interrupt like timer, external and serial inputs.
  - In the below IE register, bit corresponding to 1 activates the interrupt and 0 disables the interrupt.

IE Register – Interrupt Enable Register

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

- EX0- Enable/Disable – External Interrupt 0
- ET0- Enable/Disable – Timer 0 overflow Interrupt
- EX1- Enable/Disable – External Interrupt 1
- ET1- Enable/Disable – Timer 1 overflow Interrupt
- ES -Enable/Disable – Serial port Interrupt
- EA - Enable/Disable – All interrupts

- **Interrupt Priority Register (IP):**
  - It is also possible to change the priority levels of the interrupts by setting or clearing the corresponding bit in the Interrupt priority (IP) register as shown in the figure.
  - This allows the high priority interrupt to interrupt the low-priority interrupt, but prohibits the interruption by another low-priority interrupt.
  - Similarly, the high-priority interrupt cannot be interrupted.
  - If these interrupt priorities are not programmed, the microcontroller executes in predefined manner and its order is INT0, TF0, INT1, TF1, and SI.



Priority bit = 1 assigns high priority. Priority bit = 0 assigns low priority.

--	IP.7	Reserved
--	IP.6	Reserved
PT2	IP.5	Timer 2 interrupt priority bit (8052 only)
PS	IP.4	Serial port interrupt priority bit
PT1	IP.3	Timer 1 interrupt priority bit
PX1	IP.2	External interrupt 1 priority bit
PT0	IP.1	Timer 0 interrupt priority bit
PX0	IP.0	External interrupt 0 priority bit

- **TCON Register:**
  - TCON register specifies the type of external interrupt to the 8051 microcontroller, as shown in the figure.
  - The two external interrupts, whether edge or level triggered, specify by this register by setting or clearing appropriate bits in it.
  - A bit addressable register

TCON : Timer/Counter Control Register (Bit Addressable)

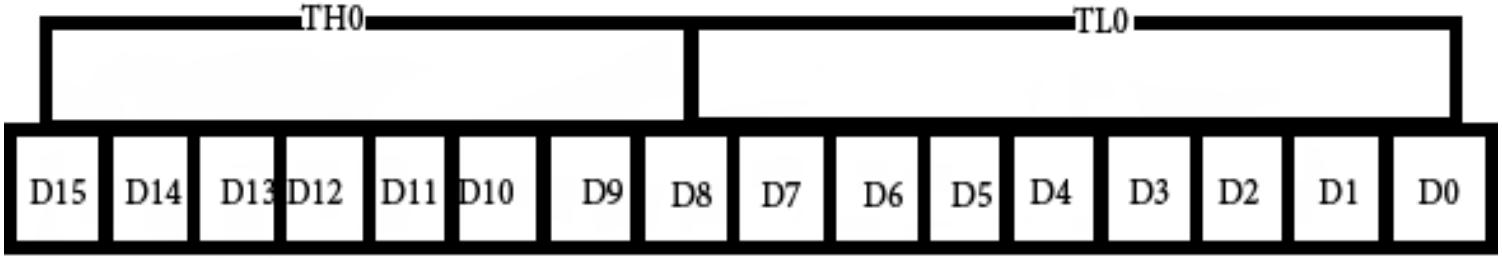
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter ON/OFF.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when External interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/flow level triggered External Interrupt.
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

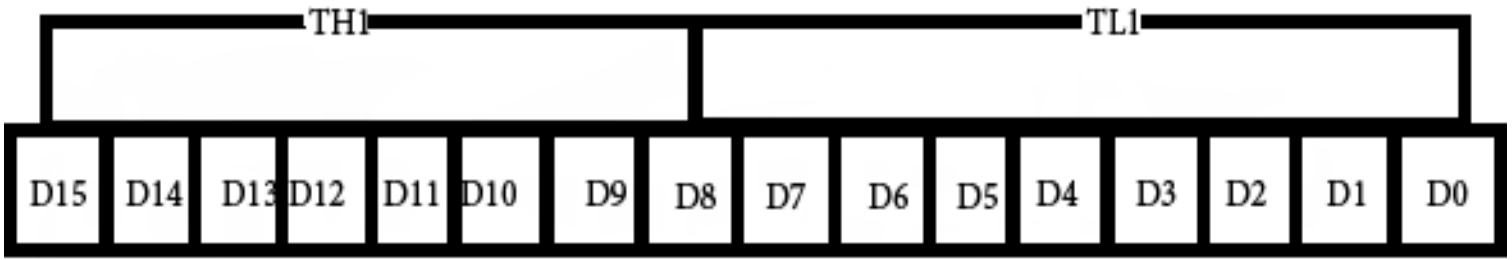
8051 Timers/Counters

- The 8051 has two timers: Timer0 and Timer1.
- They can be used either as timers or as counters.
- They can be used to:
  - Calculate time delays
  - Counting number of events
  - Frequency measurements
  - Pulse width measurements
- Both timers are 16 bits wide.
- Since the 8051 has an 8-bit architecture, each 16-bit is accessed as two separate registers of low byte and high byte.

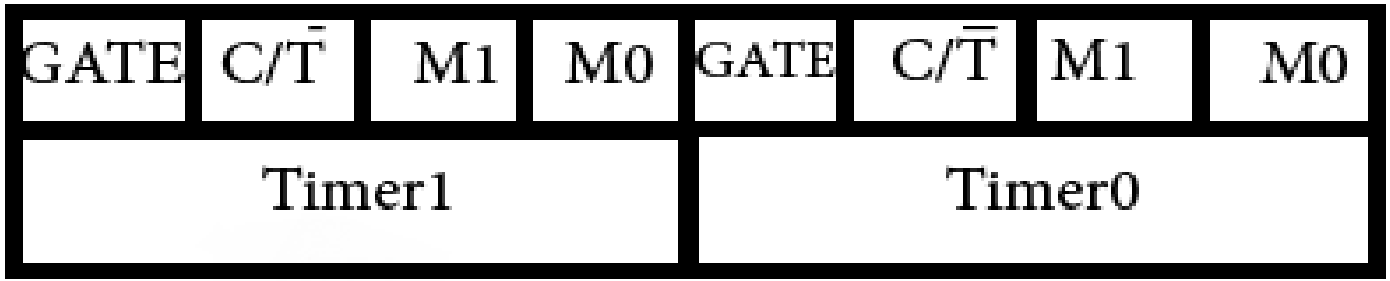
- **Timer0 Register:**
  - 16 bit register and accessed as low byte and high byte.
  - The low byte is referred as a TL0 and the high byte is referred as TH0.
  - These registers can be accessed like any other registers.



- **Timer1 Register:**
  - 16 bit register
  - Split into two bytes, referred to as TL1 and TH1.



- **TMOD (timer mode)Register:**
  - This is an 8-bit register which is used by both timers 0 and 1 to set the various timer modes.
  - In this TMOD register, lower 4 bits are set aside for timer0 and the upper 4 bits are set aside for timer1.
  - In each case, the lower 2 bits are used to set the timer mode and upper 2 bits to specify the operation.



- In upper or lower 4 bits, first bit is a **GATE** bit.
- Every timer has a means of starting and stopping.
- Some timers do this by software, some by hardware, and some have both software and hardware controls.
- The hardware way of starting and stopping the timer by an external source is achieved by making GATE=1 in the TMOD register.
- When GATE=0, no external hardware is needed to start and stop the timers.
- The second bit is **C/T** bit and is used to decide whether a timer is used as a time delay generator or an event counter.
- If this bit is 0 then it is used as a timer and if it is 1 then it is used as a counter.

- In upper or lower 4 bits, the last bits third and fourth are known as M1 and M0 respectively. These are used to select the timer mode.

- **Mode 1-** It is a 16-bit timer; therefore it allows values from 0000 to FFFFH to be loaded into the timer's registers TL and TH. After TH and TL are loaded with a 16-bit initial value, the timer must be started.

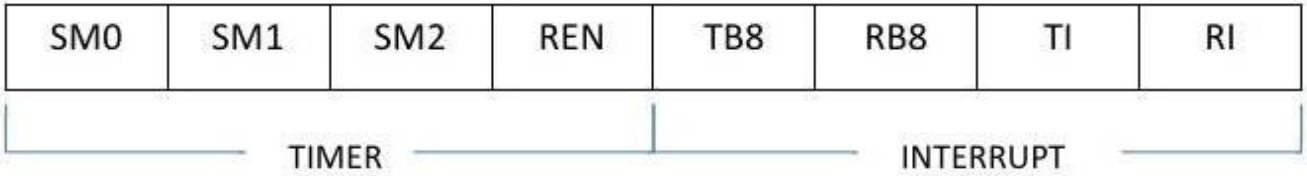
- **Mode0-** Mode 0 is exactly same like mode 1 except that it is a 13-bit timer instead of 16-bit.

- **Mode 2-** It is an 8 bit timer that allows only values of 00 to FFH to be loaded into the timer's register TH. After TH is loaded with 8 bit value, the 8051 gives a copy of it to TL.

- **Mode3-** Mode 3 is also known as a split timer mode. It is different for Timer0 and Timer1. When the Timer0 is working in mode 3, the TL0 will be used as an 8-bit timer/counter. It will be controlled by the standard Timer0 control bits. The TH0 is used as an 8-bit timer but not the counter. This is controlled by Timer1 Control bits. Timer1 is off.

8051 Serial Control Register (SCON)

- The Serial Control or SCON SFR is used to control the 8051 Microcontroller's Serial Port.
- It is located as an address of 98H.
- Using SCON, you can control the Operation Modes of the Serial Port, Baud Rate of the Serial Port and Send or Receive Data using Serial Port.
- SCON Register also consists of bits that are automatically SET when a byte of data is transmitted or received.



SM0/SCON.7 = SERIAL port mode specifier
SM1/SCON.6 = SERIAL port mode specifier
SM2/SCON.5 = For multiprocessor communication
REN/SCON.4 = Receive Enable
TB8/SCON.3 = Transfer bit 8
RB8/SCON.2 = Receive bit 8
TI/SCON.1 = Transmit Interrupt
RI/SCON.0 = Receive Interrupt

8051 Serial Data Buffer (SBUF)

- The Serial Buffer or SBUF register is used to hold the serial data while transmission or reception.

8051 Addressing Modes

- Register addressing
- Direct addressing
- Indirect addressing
- Immediate constant addressing
- Relative addressing
- Absolute addressing
- Long addressing
- Indexed addressing

Register Addressing Mode

- The register addressing instruction involves information transfer between registers
- Example: MOV R0, A
- The instruction transfers the accumulator content into the R0 register.
- The register bank (Bank 0, 1, 2 or 3) must be specified prior to this instruction.

Direct Addressing Mode

- This mode allows you to specify the operand by giving its actual memory address (typically specified in hexadecimal format) or by giving its abbreviated name (e.g. P3)

- Note: Abbreviated SFR names are defined in the “C8051F020.inc” header file

- **Examples:**

- MOV A, P3 *Transfer the contents of Port 3 to the accumulator*

- MOV A, 020H *Transfer the contents of RAM location 20H to the accumulator*

Indirect Addressing Mode

- This mode uses a pointer to hold the effective address of the operand
- Only registers R0, R1 and DPTR can be used as the pointer registers
- The R0 and R1 registers can hold an 8-bit address, whereas DPTR can hold a 16-bit address
- Examples:
- MOV @R0,A *Store the content of accumulator into the memory location pointed to by register R0. R0 could have an 8-bit address, such as 60H.*
- MOVX A,@DPTR *Transfer the contents from the memory location pointed to by DPTR into the accumulator. DPTR could have a 16-bit address, such as 1234H.*

Immediate Constant Addressing Mode

- This mode of addressing uses either an 8- or 16-bit constant value as the source operand
- This constant is specified in the instruction, rather than in a register or a memory location
- The destination register should hold the same data size which is specified by the source operand
- Examples:
- ADD A, #030H *Add 8-bit value of 30H to the 8 bit accumulator register.*
- MOV DPTR, #0FE00H *Move 16-bit data constant FE00H into the 16-bit Data Pointer Register.*

Relative Addressing Mode

- This mode of addressing is used with some type of jump instructions, like SJMP (short jump) and conditional jumps like JNZ
- These instructions transfer control from one part of a program to another ‘
- The destination address must be within -128 and +127 bytes from the current instruction address because an 8-bit offset is used ($2^8 = 256$)
- Example:

```
GoBack: DEC A       Decrement A
JNZ GoBack          If A is not zero, loop back
```

Absolute Addressing Mode

- Two instructions associated with this mode of addressing are ACALL and AJMP instructions

- These are 2-byte instructions where the 11-bit absolute address is specified as the operand

- The upper 5 bits of the 16-bit PC address are not modified.

- The lower 11 bits are loaded from this instruction. So, the branch address must be within the current 2K byte page of program memory (2^{11} = 2048)

- Example:
- ACALL PORT_INIT *PORT_INIT should be located within 2k bytes.*
- PORT_INIT: MOV P0, #0FH *PORT_INIT subroutine*

Long Addressing Mode

- This mode of addressing is used with the LCALL and LJMP instructions
- It is a 3-byte instruction and the last 2 bytes specify a 16-bit destination location where the program branches
- It allows use of the full 64 K code space
- The program will always branch to the same location no matter where the program was previously
- Example: LCALL TIMER_INIT *TIMER_INIT address (16-bits long) is specified as the operand; In C, this will be a function call: Timer_Init().*
- TIMER_INIT: ORL TMOD,#01H *TIMER_INIT subroutine*

Indexed Addressing Mode

- The Indexed addressing is useful when there is a need to retrieve data from a look-up table
- A 16-bit register (data pointer) holds the base address and the accumulator holds an 8-bit displacement or index value
- The sum of these two registers forms the effective address for a JMP or MOVC instruction

Indexed Addressing Mode

- Example:

- MOV A,#08H — *Offset from table start*
- MOV DPTR,#01F00H — *Table start address*
- MOVC A,@A+DPTR — *Gets target value from the table start address + offset and puts it in A.*

- After the execution of the above instructions, the program will branch to address 1F08H (1F00H+08H) and transfer into the accumulator the data byte retrieved from that location (from the look-up table)

8051 Instruction Set

- Divided into 5 functional groups:
- Arithmetic operations
- Logical operations
- Data transfer operations
- Boolean variable operations
- Program branching operations