

UNIT -- IV

Introduction to 8051 Microcontroller

Introduction:

- Microcontrollers are processor chips that generally have memory, input ports, and output ports within the chip itself.
- Therefore, they can also be called single-chip computers, computer-on-a-chip, or system-on-a-chip.
- Microcontrollers are used in machine control applications, where there is no need to change the program.
- Equipments that use microcontrollers include computer printers, plotters, fax machines, Xerox machines, telephones, automotive engine control mechanisms, and electronic instruments such as oscilloscopes, multimeters, IC testers, etc.

The main features -8051 are:

- 8 bit CPU,
 - Operating frequency is 12MHz,
 - 4Kbytes of on chip Program memory,
 - 128 bytes of on chip data RAM,
 - 4 ports of 8bit each,
 - Two 16 bit timers,
 - Full duplex serial port and
 - On-chip clock oscillator.
- In addition to the above features, the 8051 provide Boolean processing; six interrupt capabilities and full-fledged CPU for control applications.

Intel 8051 Architecture:

The various functional blocks of 8051 microcontroller are

- Arithmetic and Logic Unit
- Timing and Control Unit
- Data memory(RAM)/Program Memory(ROM)
- Registers
- Latches and Drives for P0,P1,P2 and P3

Arithmetic and Logic Unit:

- Arithmetic and logic unit performs arithmetic and logical operations on 8 bit operands.
- Accumulator is the register which gets the output of the ALU in most of the arithmetic and logical operations.
- Apart from addition and subtraction operations, it performs multiplication and division operations.
- Logical operations like AND, OR, XOR and NOT are possible.

Registers:

- The general purpose or working registers, stack pointer, program counter and in addition to these CPU registers, there are 21 special function registers (SFRs).
- The working registers are accumulator, B register and four register banks. Each register bank has 8 registers R0 through R7.

Accumulator:

It is an 8 bit register. It is referred as register 'A' It is used by the all the arithmetic and logical instructions. Accumulator has a special importance, that is one of the operands is stored in it before the execution of an instructions and it also stores the result after the execution of an instruction.

B Register:

It is an 8 bit register. B register is used as a general purpose register when it is not being used by the multiplication and division instructions. While multiplying, it holds one of the 8 bit operands and after the execution of the multiplication, it stores the higher byte of the result. While dividing, it holds an 8 bit divisor and after the execution, the remainder is stored in B register.

Registers R0 through R7:

These are used as scratch pad registers. There are four register banks each containing R0 through R7. At a time only one bank can be selected by RS0 and RS1 bits in the PSW.

Stack Pointer:

Stack pointer is an 8 bit register. it is incremented during push or call operations and is decremented during pop or return instructions.

Program Counter:

It is a 16 bit register. PC always points to the instruction to be fetched and it is automatically incremented after fetching the instruction.

Special function registers:

The RAM locations from 80H to FFH are reserved for the special functions and therefore these are called as Special function registers (SFRs). The SFRs are used for the special functions and cannot be used for any other purpose. These SFRs are used for control or show the status of various functions done by microcontroller. All SFRs are directly addressable and can be read or written to as well.

Program Status Word:

PSW is an 8 bit register. it consists of carry, auxiliary carry, overflow, and parity flags, there are RS1 and RS0 for register bank selection. It has a general purpose flag F0, which cab used by the programmer to store 0 or 1.

Data Pointer:

It is a 16 bit register, contains two bytes. The higher byte is called DPH and lower byte is called DPL. It is used for addressing the off-chip data and program with the MOVX and MOVC commands respectively.

Timer registers:

Register pairs (TH0, TL0), (TH1, TL1) from two 16 bit timers in 8051. The operation may be timing or counting. There are various modes in which timers can be configured. For this TCON and TMOD registers are used.

Ports 0 to 3:

P0, P1, P2 and P3 are the SFRs corresponding to four I/O ports respectively. Each of these ports is bit as well as byte addressable.

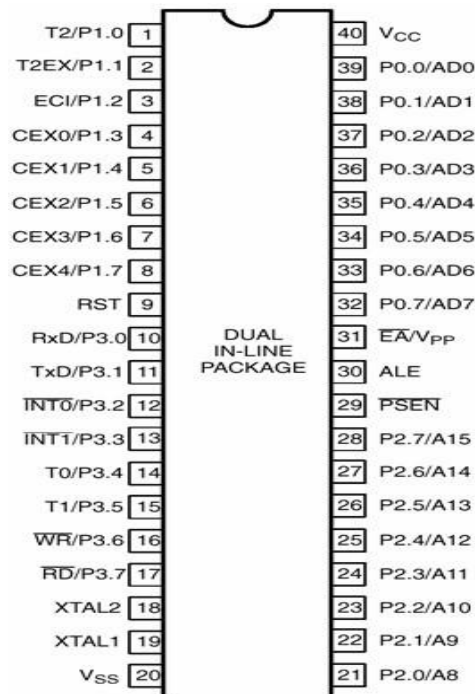
Control Registers:

TCON, TMOD, IE, IP, SCON, PCON contain the control and status for interrupts, serial I/O and timers/counters.

Oscillator:

8051 operates at 12MHz frequency.. The quartz crystal has to be connected externally and all the other oscillator circuit is on chip.

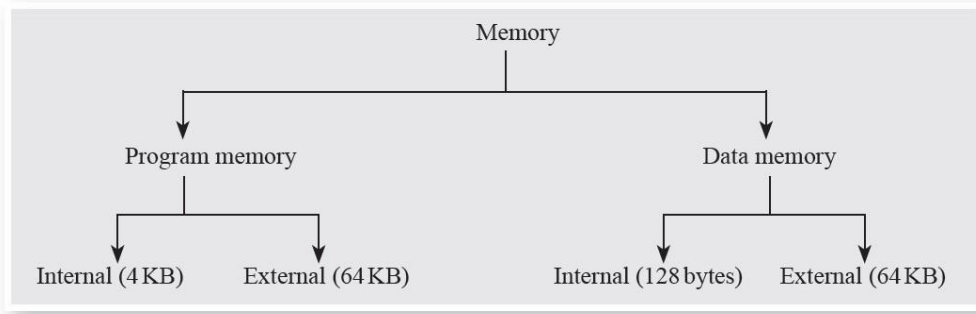
8051 Pin details of 8051 DIP IC:



- The 8051 is a 40-pin chip.
- The power supply +VCC and VSS takes two pins and the built-in clock oscillator requires two pins (– XTAL1 and XTAL2) for connecting the crystal.
- The four control signals pins of 8051 are PSEN, ALE, EA and RST.
- RST is an active high reset signal to restart the controller chip
- 8051 responds to a RST high input only if the RST is held high for at least two machine cycles.
- A machine cycle is the period taken by any processor to fetch and execute one instruction.
- In 8051, the maximum number of clock cycles taken for a machine cycle is 12.
- So, the RST pin must be high for at least 24 clock periods.
- PSEN, ALE, EA are the signals used in conjunction with the external memory access of the 8051

MEMORY ORGANISATION:

- In the 8051, the memory is organized logically into program memory and data memory separately.
- The program memory is read-only type; the data memory is organized as read–write memory.
- Again, both program and data memories can be within the chip or outside.
- The Intel 8051 has 128 bytes of RAM and 4 KB of ROM within the chip.
- The address bus of the 8051 is 16 bits wide. So it can access 64 KB of memory.



- As the memory is organized separately as program memory and data memory, the 8051 microcontroller can access 64 KB of program memory and 64 KB of data memory.
- The user can configure the entire program memory outside the chip or use 4 KB inside and 60 KB outside the chip.
- The internal data memory is accessed with 8-bit addresses and the external data memory with 16-bit addresses.
- So the maximum data memory that can be connected to the 8051 system is 64 KB.

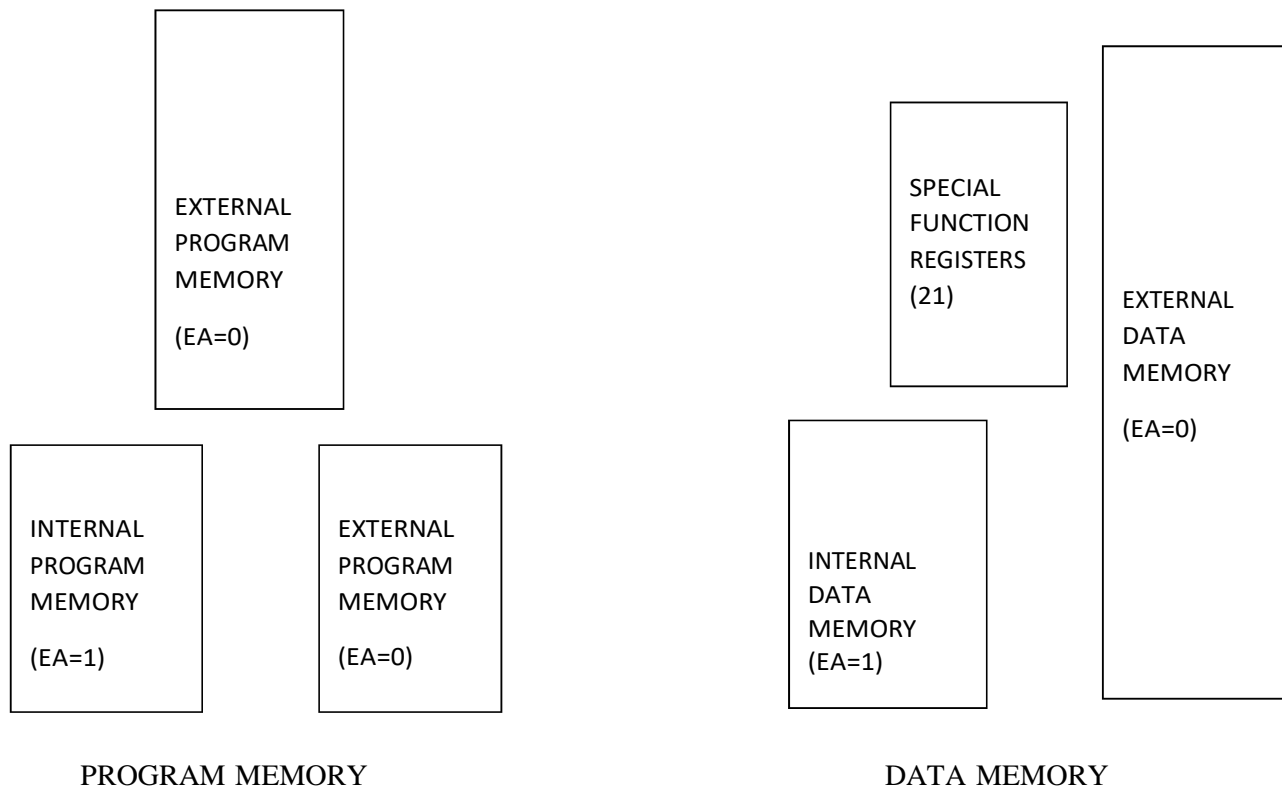
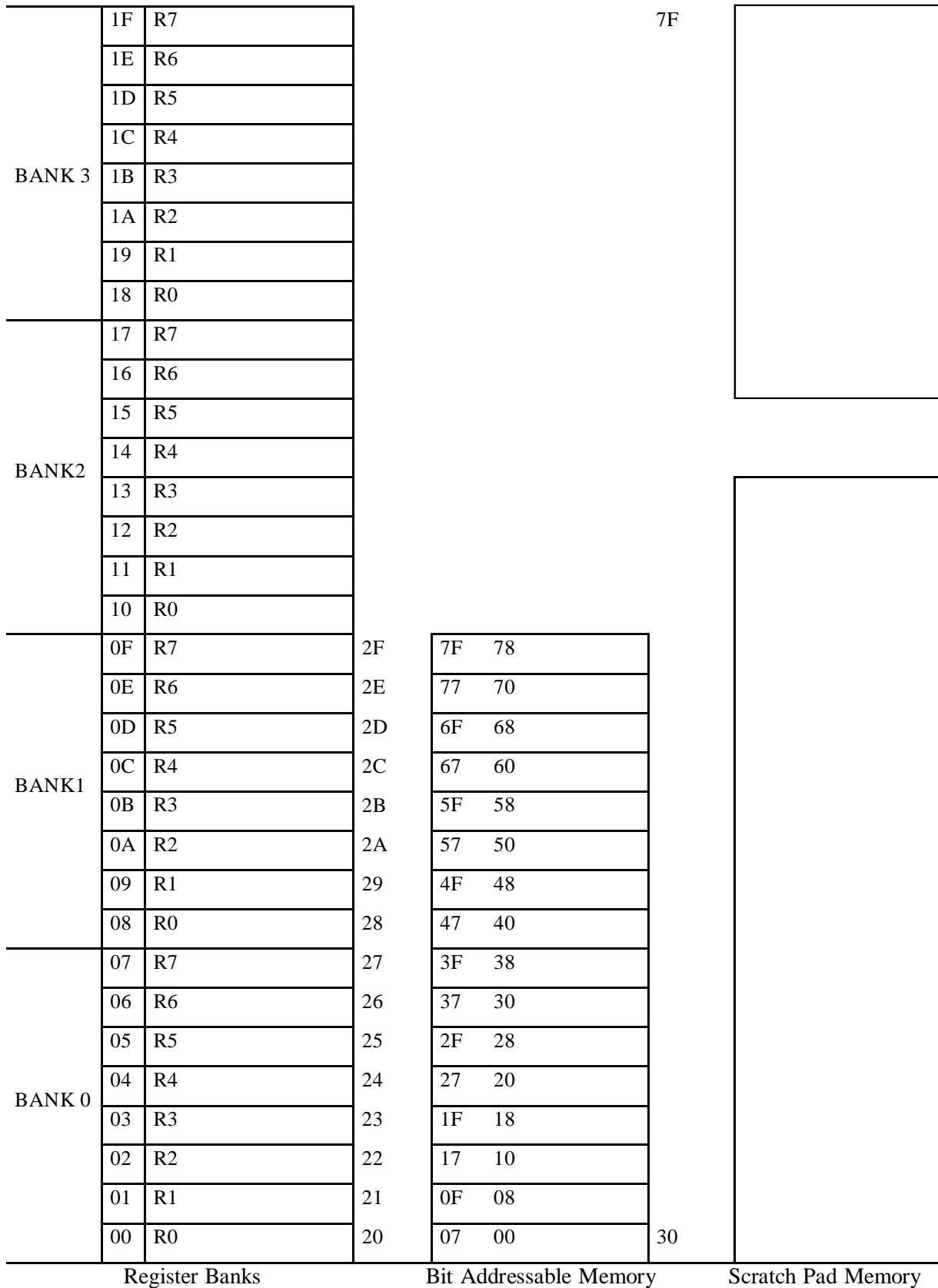


Fig: Memory organization in 8051

INTERNAL RAM STRUCTURE:

- The 8051 has 128 bytes of internal data RAM, which is accessible as bytes or sometimes as bits.
- The address of the internal RAM starts at 00H and occupies space up to 7FH. The RAM space is divided into three blocks—the register banks, the bit-addressable memory, and the scratch pad memory.
- The 8051 has four register banks of eight registers each, with addresses from 00H to 1FH. In assembly language, they are addressed by the names R0–R7.
- The register banks are identified with 2 bits in the processor status word.
- The PSW has two bits for identifying the register bank, i.e., 00 represents bank 0, 01 represents bank 1, 10 represents bank 2, and 11 represents bank 3.
- In the 8051, bitwise operations are also possible with special instructions using the bit addresses. The bit-addressable memory is both bit-addressable (from 00H to 7FH) and byte-addressable (from 20H to 2FH). Bit operations are helpful in many control algorithms.

Using general-purpose scratch pad memory, programmers can read and write data at any time for any purpose. This memory ranges from the byte address 30H to the address 7FH.



SPECIAL FUNCTION REGISTERS

- SFR, which occupies upper 128 bytes of internal memory are the registers, that control the entire processor
- They can be accessed by DIRECT addressing.
- The registers available in the 8051 are as follows :
 - Accumulators - A and B
 - Process Status Word - PSW
 - I/O port registers - P0, P1, P2, P3
 - Data pointers - DPH and DPL
 - Serial data buffer register - SBUF
 - Stack pointer - SP
 - Timer registers - TH0, TH1 and TL0, TL1
 - Timer Control Registers - TCON, TMOD
 - Power and Port control - PCON, SCON
 - Interrupt Control Registers - IP, IE.
- Programmers should not use the addresses in the range 80H to FFH (other than SFR) as it is used by INTEL CORPORATION for expansion functions of 8051.
- The 8051 has two accumulators -A register and B register.
- The register B forms the accumulator for multiplication and division instructions and for other instructions it can be accessed as a general purpose register.
- The stack in the 8051 is organized within the internal RAM area.
- The stack pointer is eight bits wide and has to be initialized with an address in the RAM area.
- When the 8051 is reset, the stack pointer is by default set to 07H.
- The stack pointer is incremented before storing a data in the stack.
- Similarly, while reading data from the stack, the data is read first and then the stack pointer is decremented.

Special Function Registers of 8051:

<u>Direct</u> <u>Memory Address (SFR)</u>	<u>Addressed</u> <u>Memory Address (SFR)</u>	<u>Special Function</u> <u>Register</u>
80		P0
81		SP
82		DPL
83		DPH
87		TCON
88		TMOD

89	TL0
8A	TL1
8B	TH0
8C	TH1
90	P1
98	SCON
99	SBUF
A0	P2
A8	IE
B0	P3
B8	IP
D0	PSW
E0	ACC
F0	B

Processor Status Word

- The PSW contains all the flags of the 8051 and is eight bits wide.
- The PSW is accessible fully as an 8-bit register, with the address D0H.
- The bit pattern of this flag register is

PSW	CY	AC	F0	RS1	RS0	OV	-	P
Bit Address	D7	D6	D5	D4	D3	D2	D1	D0

-
- **Parity bit (P)**
 - It is set to 1 if the accumulator contains an odd number of 1s, after an arithmetic or logical operation.
- **Overflow flag (OV)**
 - This flag is set during ALU operations, to indicate overflow in the result. It is set to 1 if there is a carry out of either the D7 bit or the D6 bit of the accumulator. Overflow flag is set when arithmetic operations such as add and subtract result in sign conflict.
- **General-purpose flag (F0)**
 - This is a user-programmable flag; the user can program and store any bit of his/her choice in this flag, using the bit address.
- **Register bank selection (RS1 and RS0)**
 - These bits are user-programmable.
 - The register bank selection in the programs can be changed using these two bits.

RS1	RS0	Selected Bank	Address Range
0	0	Bank 0	00h to 07h
0	1	Bank 1	08h to 0Fh
1	0	Bank 2	10h to 17h
1	1	Bank 3	18h to 1Fh

- **Auxiliary carry flag (AC)**

- It is used in association with BCD arithmetic. This flag is set when there is a carry out of the D3 bit of the accumulator.

- **Carry flag (CY)**

- This flag is used to indicate the carry generated after arithmetic operations. It can also be used as an accumulator, to store one of the data bits for bit-related Boolean instructions.

The 8051 supports bit manipulation instructions.

- This means that in addition to the byte operations, bit operations can also be done using bit data.
- For this purpose, the contents of the PSW are bit-addressable.

Bit addressable registers:

- 8051 supports bit manipulation instructions. -bit operations can also be done using bit data.
- Similarly, the accumulator and B register contents -bit addressable.
- The bit addresses of all the bits of the accumulator and B registers are given as

ACCUMULATOR:

Accumulator bits	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
Bit Address	E7	E6	E5	E4	E3	E2	E1	E0

B REGISTER:

B register bits	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
Bit Address	F7	F6	F5	F4	F3	F2	F1	F0

POWER CONTROL IN 8051:

- The 8051 has various power control modes, which are used to control the power consumed by the microcontroller chip.
- Some of these modes let the microcontroller go into a ‘sleep’ mode, which makes it consume lesser power than during normal operation.
- The power control modes are selected through the Special Function Register PCON.

<u>Bit</u>	<u>Name</u>	<u>Explanation of Function</u>
7	SMOD	Serial port Baud rate set bit
6	-	Reserved
5	-	Reserved
4	-	Reserved
3	GF1	General Purpose Flag 1
2	GF0	General Purpose Flag 0
1	PD	Power Down mode set bit
0	IPL	Idle Mode Set bit

IDLE MODE:

- The micro-controller enters the idle mode whenever the PCON.0 bit is set to 1.
- In the idle mode, the clock pulses applied to the CPU are masked while all other units like interrupt controller etc., will be kept active.
- The contents of the CPU are not affected in this idle mode.
- The processor can be revoked -idle mode -hardware interrupt or by giving a hardware reset signal.
- These two actions - reset PCON.0 and the processor execution -resumed to the instruction following the instruction that set idle mode.

POWER DOWN MODE:

- The Power down mode is initiated by making PCON.1 bit to 1.
- In this mode, the clock generator -switched off and only the internal memory is active.
- The supply voltage Vcc can be reduced to 2V and the power consumption –reduced.
- Only way to revoke the processor from power down mode -reset the system.

THE STACK OPERATION:

- In the 8051, the stack is configured as a series of memory locations following the Last-In First-Out (LIFO) pattern.
- In general, the stack is initialized in the internal RAM area. Any 8-bit data can be stored and retrieved from the stack using PUSH and POP instructions, with the help of the stack pointer.
- The stack pointer (SP) is an 8-bit register within the SFR area, with the address 81H.
- This register can hold one 8-bit address at a time, which is actually the memory location at top of the stack.
- A push operation in the 8051 is used to store an 8-bit data in the stack.
- The PUSH instruction first increments the value of SP and then stores the data mentioned in the instruction in the memory location pointed to by SP.
- Similarly, the POP instruction stores the value from the top of the stack in the register mentioned in the instruction and then decrements the value of SP.

- The stack pointer is initialized to the value 07H when the 8051 microcontroller is reset.
- The other instructions of the 8051 that affect the stack and the stack pointer are ACALL, LCALL, RET, and RETI.
- The stack pointer can be initialized to any internal RAM address by the programmer, by writing the required address in the SP SFR address 81H.

ADDRESSING MODES OF 8051

- The way by which a data is specified in an instruction is called as addressing mode.
- The data fetched for execution depends upon the addressing mode.
- The instruction set of 8051 supports 5 addressing modes.

Immediate Addressing Mode:

- The data to be manipulated is directly given in the instruction itself.
- The data is preceded by a # symbol.
- E.g. ADD A, #80h. ; adds the data 80h to the contents of the accumulator and the result is stored in the accumulator itself.

Register Direct Addressing:

- The register, that contains the data to be manipulated, is specified in the instruction.
E.g. ADD A, R0 ; adds the contents stored in register R0 with the accumulator contents and store the result in accumulator.

Memory Direct Addressing:

- The memory address that contains the data to be operated is specified here in the instruction.
- E.g. ADD A, 74h; adds the data in accumulator with that stored in memory address 74h.

Memory Indirect Addressing:

- The register, which contains the actual memory address of the data, is specified in the instruction.
- The register specified is preceded by @ symbol in assembly language format.
E.g. ADD A, @R0 ; The value stored in the register R0 is now the address of the memory location of the data to be fetched.

Indexed Addressing:

- In this type of addressing, the instruction consists of two parts - a base address and an offset.
- This type of addressing is useful in relative memory accessing and relative jumping.
- The base address is stored in data pointer (DPTR) or any other register.
- The offset value is stored in Accumulator e.g. MOVC A, @A+DPTR.

INSTRUCTION SET OF 8051

- Instruction supported by 8051 can be classified into different types depending upon their operational functions.
- The instruction set classification is as followed.

Data Transfer Instructions:

- As the name indicates, instructions in this set are used to transfer data.
- The data can be transferred from or to external RAM or within the internal memory itself.
- The instruction MOV is used to transfer the data between internal registers/memory.
- The general format is
 - MOV Reg destination, Reg source.

Mnemonic	Operation	Addressing Modes			
		Direct	Indirect	Register	Immediate
MOV A, <src>	A = <src>	√	√	√	√
MOV <dest>, A	<dest> = A	√	√	√	
MOV <dest>, <src>	<dest> = <src>	√	√	√	√
MOV DPTR, # data 16	DPTR = 16-bit immediate data				√
PUSH <src>	INC SP: MOV “@SP”, <scr>	√			
POP <dest>	MOV <dest>, “@SP”: DEC SP	√			
XCH A, <byte>	ACC and <byte> Exchange Data	√	√	√	
XCHD A, @Ri	ACC and @ Ri exchange low nibbles		√		

MOVX A, @Ri	Copy 8 bit data from the external RAM location pointed to by Ri to register A	Only Indirect Addressing mode
MOVX @ Ri, A	Copy 8-bit data from register A to the external RAM location pointed to by Ri	Only Indirect Addressing mode
MOVX A, @ DPTR	Copy 8-bit data from the external RAM location pointed to by the 16-bit DPTR to register A	Only Indirect Addressing mode
MOVX @ DPTR, A	Copy 8-bit data from register A to the external RAM location pointed to by the 16-bit DPTR	Only Indirect Addressing mode
MOVC A, @A + DPTR	Read Program Memory at (A + DPTR)	Only Indirect Addressing mode
MOVC A, @A + PC	Read Program Memory at (A + PC)	Only Indirect Addressing mode

- The instructions with the mnemonic MOVX is used to access data from external memory locations using indirect addressing only. –
- MOVX instruction must use Accumulator (A) register as -destination or source and the other is indirectly accessed external memory.
- MOVX can be used -8 bit external memory address and 16 bit external memory address. It can be noted that the external memory -interfaced with 8051 with either 8 bit address or 16 bit address.
- If the 8 bit address is used-internal register (any location in Internal RAM) -hold the address of the memory. If 16 bit address is used-Data Pointer (DPTR) is used to hold the address.
- The instructions MOVC A,@A+DPTR and MOVC A,@A+PC are the two instructions meaning MOVE CODE MEMORY and are used to transfer data from program memory using indexed addressing

- The program memory addressing using MOVC instruction needs 16 bit address. So, the Data Pointer register (DPTR) and Program Counter (PC) -base registers -instructions.
- Data can only be read from the program memory and not written into because the program memory is generally ROM.
- PUSH instruction is used to copy data in any internal RAM location to the stack
- The POP instruction is used to copy data from the top of the stack to the RAM location specified in the instruction.
- XCHD is used to transfer only the lower-order nibble between the accumulator and the indirectly addressed internal RAM.
- XCH is used to exchange the contents of the accumulator and a register or the internal memory of the 8051.

Arithmetic Instructions:

- These instructions are used to do arithmetic operations.
- The common arithmetic operations like addition, subtraction, multiplication and division are possible with 8051.
- All the data used in arithmetic instructions must be available inside the controller i.e. in the internal RAM area only.
- ADD instruction is used to add any 8 bit data with Accumulator and the result is stored in Accumulator (A) register. The carry generated if any is stored in Carry flag of the processor status word.
- The ADC instruction is also used to add any 8 bit data with Accumulator along with Carry bit.
- The SUBB instruction -subtract contents of a register from the Accumulator content and during this subtraction, the Carry bit is also subtracted from the accumulator.
- For ADD and SUBB instructions, one of the data must be in Accumulator and the other data - in any direct addressed or indirect addressed internal memory location or can be an immediate data.
- In addition to - ADD, ADC and SUBB instructions in 8085, -have instructions MUL and DIV.
- The register B is exclusively used for these two instructions. The operands should be stored in the registers A and B for the MUL and DIV instructions.
- The MUL instruction multiplies the contents of A and B registers and stores the 16 bit result in the combined A and B registers.
- The lower order byte -result is stored in A register and the higher order byte - stored in B register.
- The DIV instruction upon execution will divide the contents of A register by the contents of B register.
- The quotient of the result - stored in A register and the remainder is stored in B register.
- A division by 0 i.e. 0 in the B register before executing DIV AB will result in the overflow flag (OV) set to 1.
- DA A instruction -to convert binary sum obtained after adding two BCD numbers into BCD number.

Mnemonic	Operation	Addressing Modes			
		Direct	Indirect	Register	Immediate
ADD <byte> A,	$A = A + \text{<byte>}$	√	√	√	√
ADDC A,<byte>	$A = A + \text{<byte>} + C$	√	√	√	√
SUBB <byte> A,	$A = A - \text{<byte>} - C$	√	√	√	√
INC A	$A = A + 1$	Accumulator Only			
DEC A	$A = A - 1$	Accumulator Only			
DEC <byte>	$\text{<byte>} = \text{<byte>} - 1$	√	√	√	√
MUL AB	$B:A = B \times A$	Accumulator Only			
DIV AB	$A = \text{Int } [A/B]$ $B = \text{Mod } [A/B]$	Accumulator Only			
DA A	Decimal Adjust Accumulator	Accumulator Only			

Logical Instructions:

- In addition to logical AND, OR and XRL operation, 8051 has additional instructions - CLR, CPL. All the data for the logical instructions -available in the internal RAM only.
- The instruction CLR A -to clear the contents of A register, CPL is used to complement or logically invert the contents of the A register and SWAP - to swap the nibbles of A register.
- 8051 supports four rotate operations with the options –rotating left or right and rotating through carry or not.

Mnemonic	Operation	Addressing Modes			
		Direct	Indirect	Register	Immediate
ANL A, <byte>	$A = A \text{ AND } \text{<byte>}$	√	√	√	√
ANL <byte>, A	$\text{<byte>} = \text{<byte>} \text{ AND } A$	√			
ANL <byte>, # data	$\text{<byte>} = \text{<byte>} \text{ AND } \# \text{ data}$	√			
ORL A, <byte>	$A = A \text{ OR } \text{<byte>}$	√	√	√	√
ORL <byte>, A	$\text{<byte>} = \text{<byte>} \text{ OR } A$	√			
ORL <byte>, # data	$\text{<byte>} = \text{<byte>} \text{ OR } \# \text{ data}$	√			

XRL A, <byte>	A = A XOR <byte>	√	√	√	√
XRL <byte>, A	<byte> = <byte> XOR A	√			
XRL <byte>, # data	<byte> = <byte> XOR # data	√			
CLR A	A = 00H	Accumulator only			
CLP A	A = NOT A	Accumulator only			
RL A	Rotate ACC Left 1 bit	Accumulator only			
RLC A	Rotate Left through Carry	Accumulator only			
RR A	Rotate ACC Right 1 bit	Accumulator only			
RRC A	Rotate Right through Carry	Accumulator only			

Branching Instructions:

- 8051 supports unconditional jumping and subroutine calling in three different ways.
- They are Absolute jump AJMP, ACALL, long jump LJMP, LCALL, and short jump SJMP.

Mnemonics	Operation
SJMP rel_addr	Jump to (PC) + 8-bit rel_addr.
AJMP 11-bit addr	Jump to PC:addr.
LJMP addr	Jump to addr.
JMP @A + DPTR	Jump to A + DPTR.
ACALL 11-bit addr	Call subroutine at PC:addr.
LCALL addr	Call subroutine at addr.
RET	Return from subroutine.
RETI	Return from interrupt.
NOP	No operation

Conditional Branching Instructions:

Mnemonic	Operation	Addressing Modes			
		Direct	Indirect	Register	Immediate
CJNE A,<byte>,rel	Jump if A ≠ <byte>	√			√
CJNE <byte>,#data,rel	Jump if <byte> = #data		√	√	
JZ rel	Jump if A = 0	Accumulator only			
JNZ rel	Jump if A ≠ 0	Accumulator only			

Bit Manipulation Instructions:

- The special feature of the 8051 micro controller is that it can handle bit data also like that of byte data.
- The internal data memory map of 8051 has a bit- addressable area also.
- The special function registers that have the address with 0 or 8 as last digit in their hex address, are also bit addressable.
- The bit manipulation instructions include logical instructions and conditional branching.

Mnemonic	Operation
ANL C,bit	C = C AND bit
ANL C,/bit	C = C AND (NOT bit)
ORL C,bit	C = C OR bit
ORL C,/bit	C = C OR (NOT bit)
MOV C,bit	C = bit
MOV bit,C	bit = C
CLR C	C = 0
CLR bit	bit = 0
SETB C	C = 1
SETB bit	bit = 1
CPL C	C = NOT C
CPL bit	bit = NOT bit
JC rel	Jump if C = 1
JNC rel	Jump if C = 0
JB bit,rel	Jump if bit = 1
JNB bit,rel	Jump if bit = 0
JBC bit,rel	Jump if bit = 1 ; CLR bit

- The logical instructions - ANL and ORL. Conditional branching - JC, JNC, JB, JNB, JBC.
- The other instructions available -CLR, SETB, CPL, and MOV.
- There are no instructions for halting the machine execution.
- Figure 10.6 shows the flag bits affected by the various instructions.
- Increment and decrement instructions do not affect the flag register.

Assembler Directives:

The assembler directives are special instruction to the assembler program and are used to define specific operations. These directives are not part of the executable program.

Some of the most frequently assembler directives are listed as follows:

ORG - OriGinate, defines the starting address for the program in program (code) memory

EQU - EQUate, assigns a numeric value to a symbol identifier so as to make the program more readable.

DB - Define a Byte, puts a byte (8-bit number) number constant at the memory location specified.

DW - Define a Word, puts a word (16-bit number) number constant at the memory location specified.

DBIT - Define a Bit, defines a bit constant, which is stored in the bit addressable section of the Internal RAM.

END - This is the last statement in the source file to advise the assembler to stop the assembly process.

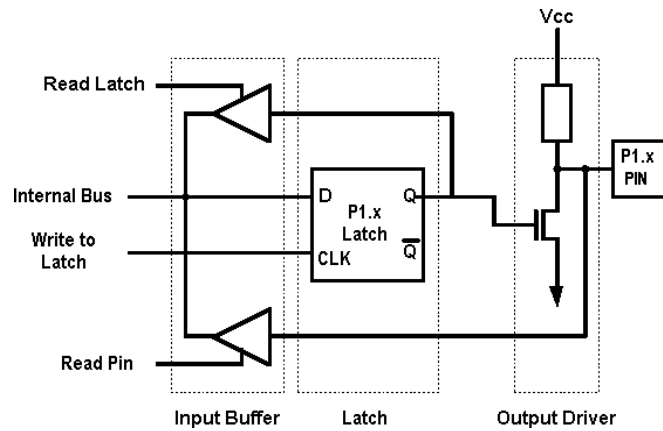
Hardware Features of 8051

Parallel Ports of 8051

- 8051 has 4 I/O ports namely Port0, Port1, Port2 and Port3. The major constraint -the number of pin count. To reduce - number of pins of the microcontroller ICs, the pins allotted for the parallel ports - alternate functions also.
- Out of the available four parallel ports of 8051, Port 1 is used exclusively for input and output functions alone.
- The other port pins have -distinct function in addition to - used for input and output functions. So, all the 24 pins of Port 0, port 2, port 3 -two different functions based on the commands or programs running.
- All the four ports are bi-directional -programmed to have input or output operation. All the 8 port pins -connected through 8 D type port latches. One D type latch connects the data in it to a port pin when the port is used as output port.
- The user can access all the four ports using -addresses mapped in the special function register area.
- Note that it is possible to address individual bits of all the four ports by - bit addresses. Using this bit address, individual bits can be read in or changed.

Structure of Port 1

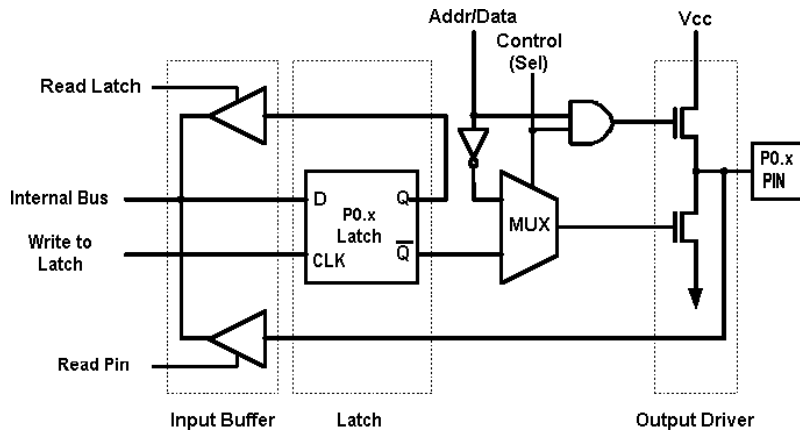
- Port 1 -only port in 8051 -used exclusively for input and output operation.
- The output of the port latch is connected to the port pin through a transistor driver with internal pull up resistor. The port can be operated as an input after writing 1 to all the bits of port 1 latch.
- 8051 ports are organized such that most instructions read the data from the pin for read operation and some instructions read the data from the latch.
- So, the input buffer consists of the select logic and the related control signals – ‘Read Latch’, ‘Read Pin’ for discriminating this.



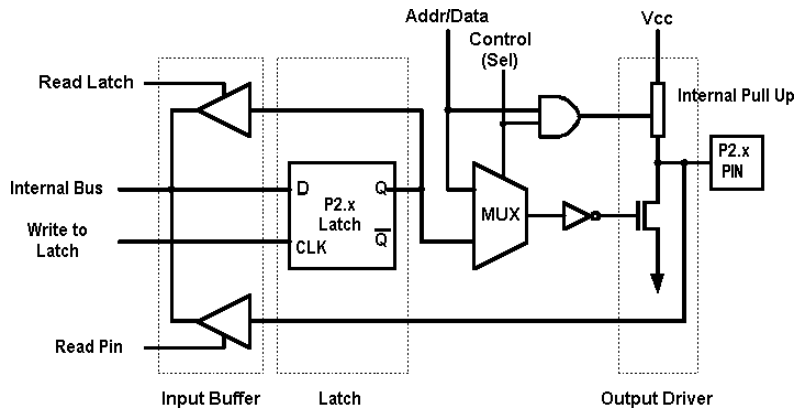
Internal Structure of Port 1

Structures of Port 0 and Port 2

- Pins of ports 0 and 2 can be used as input port pins if a 1 is written to the corresponding port latches by the programmer.
- For using Pins of ports 0 and 2 as output pins, a pull up resistor may have to be connected to the corresponding port pins.
- Ports 0 and 2 have an alternate function in addition to being used as input and output ports.
- The Ports 0 and 2 are used as address/ data bus when external memory or I/O devices are accessed. Port 0 -low order address bus and the port 2 -higher order address bus.
- The drivers of Port0 and Port2 have an internal multiplexer to serve this purpose as shown in figures.



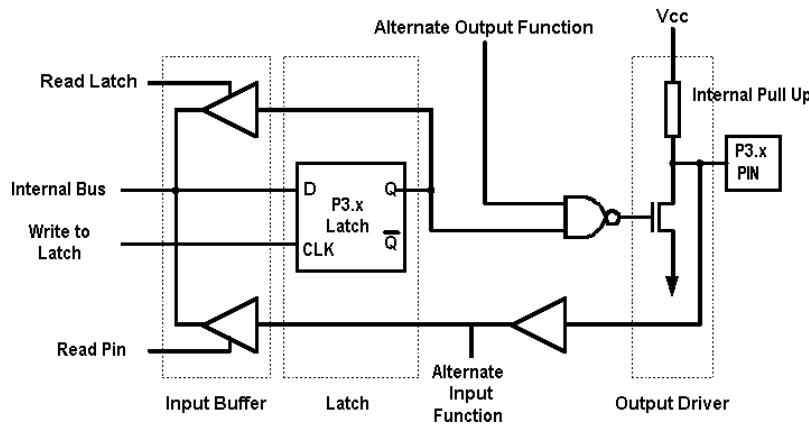
Internal Structure of Port 0



Internal Structure of Port 2

Structure of Port 3:

- Port 3 is different from the other ports in the aspect that individual port pins can be programmed for input and output operation.
- Each pin of port 3 can be programmed - for input, output operation or alternate functions.
- All the Port3 pins -serve an alternate function according to the hardware signals and interfacing.
- The alternate functions can be activated only if the port3 bits are written with 1s in their position.



Internal Structure of Port 3

Alternate functions of Port 3

Regarding the port access-two possibilities for read operation. The read instruction for a port can either read the port latch or the port pins.

This difference is made in the internal hardware of 8051 in order to avoid misinterpretation of the voltage level at the pins.

The instructions, which have the operations – Read, Modify and write to the port – read the Port latches and not the pins.

8051 Timers

- The 8051 comes -with two 16 bit timers, both of which may be controlled, set, read, and configured individually. The 8051 timers have three general functions:
 - Programming predefined length of time, and then issuing an interrupt request.
 - Counting the transitions on an external pin,
 - Generating baud rates for the serial port.
- If the timer registers are incremented by the internal clock pulses from the microcontroller, then the operation is termed as ‘Timing’ operation.
- Meanwhile if the timer registers get their clock pulses from an external device through the port 3 pins of 8051, then the operation is termed as ‘Counting’.
- Timer 0 external input pin P3.4 (T0) is used give clock input to timer 0 to act as counter.
- Timer 1 external input pin P3.5 (T1) is used give clock input to timer 1.
- One timer is TIMER0 and the other is TIMER1. Each timer also has two 8 bit SFRs namely TH0 and TL0 forming the higher and lower order bytes of Timer0 and TH1 and TL1 forming the higher and lower bytes of Timer1.

TMOD:

The TMOD SFR -used to control the mode of operation of both timers.

The higher order four bits (bits 4 through 7) relate to Timer 1 whereas the low four bits (bits 0 through 3) perform the same functions for timer 0.

Bit	Name	Explanation of Function	Timer
D7	GATE1	When this bit is set the timer will only run when INT1 (P3.3) is high. When this bit is clear the timer will run regardless of the state of INT1.	1
D6	C/T1	When this bit is set the timer will count events on T1 (P3.5). When this bit is clear the timer will be incremented every machine cycle.	1
D5	T1M1	Timer mode bit (see below)	1
D4	T1M0	Timer mode bit (see below)	1
D3	GATE0	When this bit is set the timer will only run when INT0 (P3.2) is high. When this bit is clear the timer will run regardless of the state of INT0.	0
D2	C/T0	When this bit is set the timer will count events on T0 (P3.4). When this bit is clear the timer will be incremented every machine cycle.	0
D1	T0M1	Timer mode bit (see below)	0
D0	T0M0	Timer mode bit (see below)	0

Two bits are used for each timer to specify a mode of operation. So, each timer can be operated in any one of four modes.

TxM1	TxM0	Timer mode	Description of mode
0	0	0	13-bit timer
0	1	1	16-bit timer
1	0	2	8-bit auto reload
1	1	3	Split timer mode

TCON:

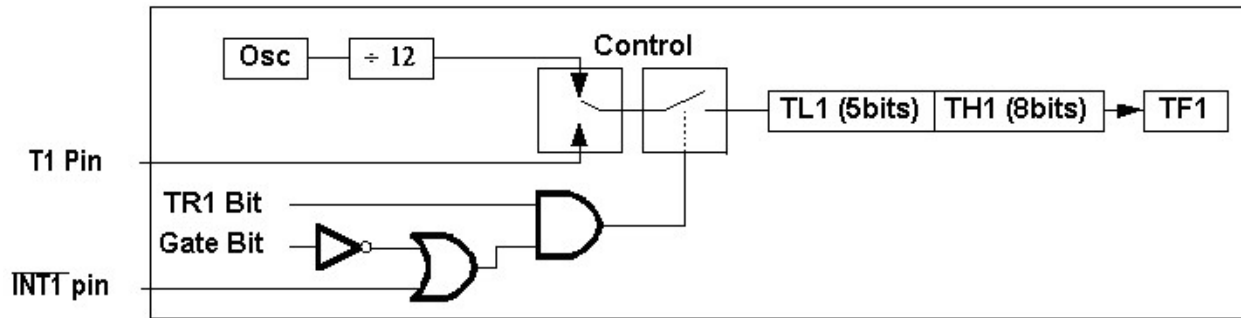
Bit position	D7	D6	D5	D4	D3	D2	D1	D0
Name	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
Explanation	Timer 1 Overflow flag	Timer 1 run control bit. Set to 1 by software to run.	Timer 0 Overflow flag	Timer 0 run control bit. Set to 1 by software to run.	External Interrupt 1 edge detect bit	Time 1 Interrupt. Set to 1 by software for edge triggering and cleared for level triggering	External Interrupt 0 edge detect bit	Timer 0 Interrupt 0. Set to 1 by software for edge triggering and cleared for level triggering

Timer Operating Modes

The two 16 bit timers of 8051 can be operated in any one of the four modes. The mode selection can be done by the setting the proper bits in the TMOD SFR.

Mode 0 - 13-bit Timer Mode

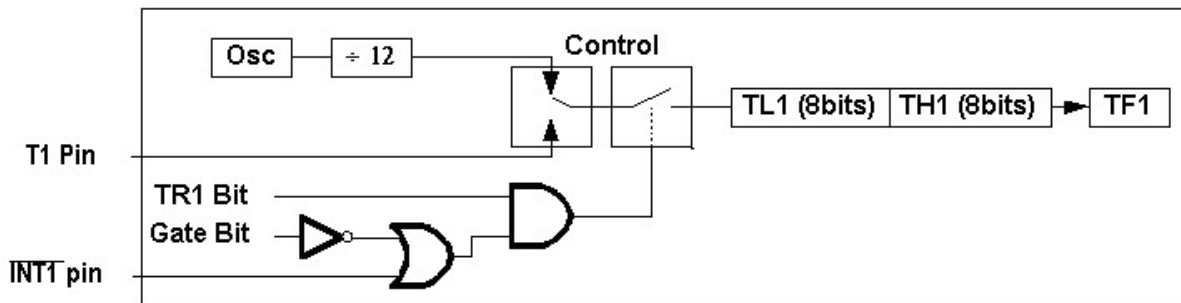
- Timer mode "0" is a 13-bit timer. Out of 16 bits of Timers, only 13bits are used. The 5 bits of lower order byte is used and 8 bits of the higher order byte of the timers are used in Mode0. Lower order byte TL0/1 will count from 0 to 31.
- When TL0/1 is incremented from 31, it will "reset" to 0 and increment TH0/1. So, the timer can only contain 8192 values from 0 to 8192.
- The timer can be operated as timer with internal clock pulses or as a counter with external clock pulses.
- This selection is done by D2 bits of TMOD for Timer 0 and D6 bit of TMOD for Timer1
- The clock pulses selected by D2 and D6 bits of TMOD is then controlled by programmer setting and connected to the Timer registers. The control is by three different means.
- First is the Timer Run control bits D4 and D6 of TCON register. The timer will run only when Timer run control bits are set to 1.
- The other controls for the timers are through the GATE control bits D4 and D7 of TMOD and the External inputs for timer. Setting GATE to 1 allows the timer to count only if the external control input INTO or INT1 is set to 1. Setting Gate to 0 will disable the corresponding external timer control inputs INTO and INT1.
- Setting Timer to mode 0 will overflow back to zero after 8192 counts. This will set the TF1 and TF0 bits for timer 1 and timer 0 respectively.



Mode 0 Operation of Timer 1 of 8051

Mode 1 – 16-bit Timer Mode

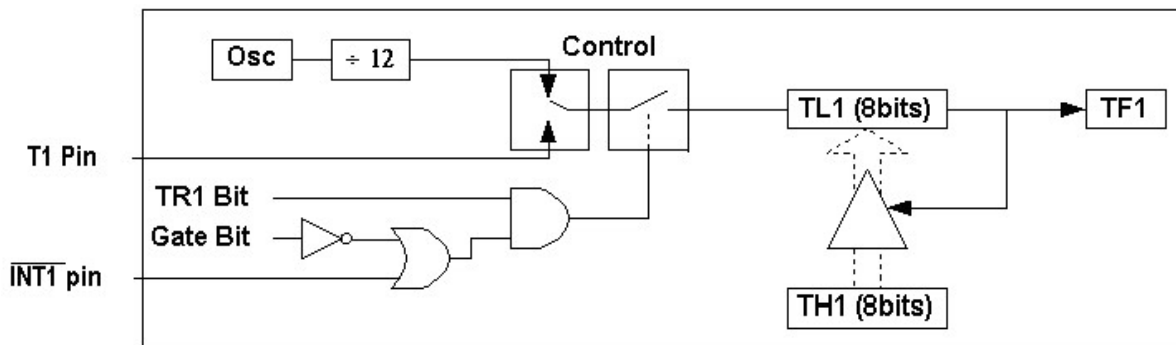
- In timer Mode 1 of 8051, each timer is operated as a 16-bit timer. It functions just like mode 0 except that all 16 bits are used.
- TL0/1 bits are incremented from 0 to 255. When TL0/1 is incremented from 255, it resets to 0 and causes TH0/1 to be incremented by 1. Since this is a full 16-bit timer, the timer may contain -65536 distinct values.
- The control of gating and running the timer in mode 1 is similar to that of mode 0



Mode 1 Operation of Timer 1 of 8051

Mode 2 – 8-bit Timer Auto Reload Mode

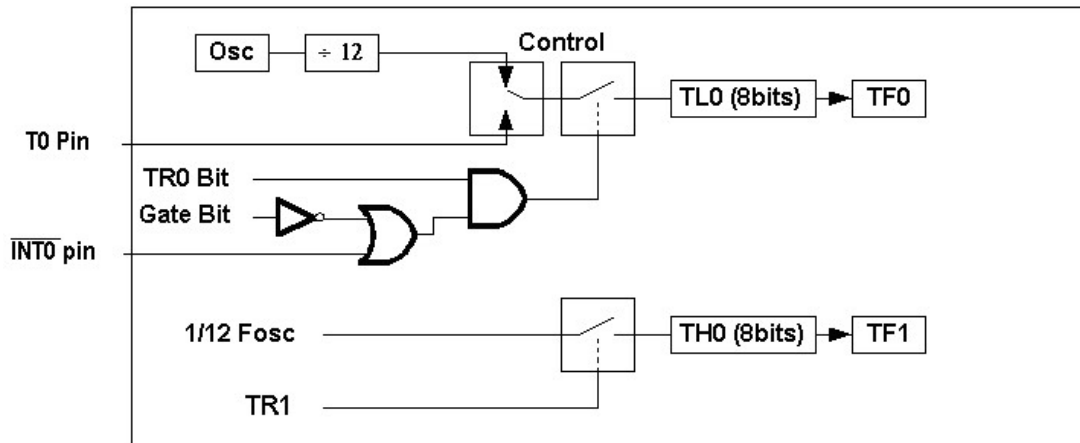
- Only 4 of the 8 bits of the TCON SFR is defined. the other 4 bits of the SFR don't have anything to do with timers. They are related with Interrupts and they will be discussed in the chapter that addresses interrupts.
- Note that the individual bits of TCON register can be addressed separately by their bit addresses. This allows the programmer to run the timers using bit addressable instructions and check the overflow independently.



Mode 2 Operating of Timer 1 of 8051

Mode 3 – Split Timer Mode

- Mode "3" of 8051 timer is a split-timer mode and is applicable only for Timer 0. When Timer 0 is placed in mode 3, it essentially becomes two separate 8-bit timers. That is, Timer 0 is TL0 and Timer 1 is TH0.
- Both timers count from 0 to 255 and overflow back to 0.
- In mode 3, all the bits that are related to Real Timer 1 will simply hold its count and will not run and the situation is similar to keeping TR1=0.
- In Split Timer mode of Timer 0, the real Timer 1 (i.e. TH1 and TL1) can not be started or stopped since the bits that do that are now linked to TH0. The real timer 1, in this case, will be incremented every machine cycle no matter what.
- When two separate timers in addition to a baud rate generator is required in an application, then real Timer 1 can be used as a baud rate generator and TH0/TL0 can be used as two separate timers in mode 3.



Mode 3 Operation of Timer 0 of 8051

8051 Interrupts

- 8051 basically has following five interrupt sources so that any of the following events will make 8051 to execute an interrupt service routine.
 - Timer 0 Overflow.
 - Timer 1 Overflow.
 - Reception/Transmission of Serial Character.
 - External hardware interrupt 0.
 - External hardware interrupt 1.

Enabling and Disabling Interrupts:

IE register:

Bit position	D7	D6	D5	D4	D3	D2	D1	D0
Bit Address	AF			AC	AB	AA	A9	A8
Name	EA	-	-	ES	ET1	EX1	ET0	EX0
Explanation	Global Interrupt Enable/Disable	Undefined	Undefined	Enable Serial Interrupt	Enable Timer 1 Interrupt	Enable External 1 Interrupt	Enable Timer 0 Interrupt	Enable External 0 Interrupt

Bit Patterns for the IE

Interrupt Priorities and Polling Sequence

- The 8051 automatically evaluates whether an interrupt occurs after every instruction. When checking for interrupt conditions, it checks them in the following order:

Priority of Interrupts:

- External 0 Interrupt
 - Timer 0 Interrupt
 - External 1 Interrupt
 - Timer 1 Interrupt
 - Serial Interrupt
- So, whenever the External 0 interrupt and Timer 1 interrupt occurs at the same instant, then 8051 microcontroller executes the interrupt service routine corresponding to External 0 interrupt first.
 - Then 8051 microcontroller will return to the main program, execute one instruction and then execute the interrupt service routine corresponding to Timer 1 Interrupt.
 - Interrupt priorities are controlled by the IP SFR (B8h). For example, if the Serial Interrupt is much more important than the Timer 0 interrupt, then the Interrupt Priority register IP SFR at the address B8h can be properly programmed to set the priority.

- This is done by assigning a high priority to the Serial Interrupt and a low priority to the Timer 0 Interrupt. By setting the D4 bit to 1, the serial interrupt will be set to higher priority and making D1 bit to 0, the Timer 0 interrupt will be set to lower priority.
- For example, the timer 0 interrupt priority can be made high by setting the D1 bit of IP SFR. So, the following instructions can be used for the same.
- SETB PT0 (or) SETB B9H (or) MOV IP, #82H

IP register:

Bit position	D7	D6	D5	D4	D3	D2	D1	D0
Bit Address				BC	BB	BA	B9	B8
Name	EA	-	-	PS	PT1	PX1	PT0	PX0
Explanation	Enable Interrupts - Made 0 to disable all interrupts	Undefined	Undefined	Serial Interrupt Priority	Timer 1 Interrupt Priority	External 1 Interrupt Priority	Timer 0 Interrupt Priority	External 0 Interrupt Priority

8051 Serial Ports:

- One of the 8051's many powerful features is its integrated Universal Asynchronous Receiver Transmitter (UART), otherwise known as a serial port. With integrated serial port of 8051, data can be transmitted and received easily by reading and writing the data to the serial port registers.

The features of the 8051 serial ports are

- Full duplex operation.
- Receive Buffered.
- Access using single double buffered Register SBUF.
- Four different modes of operation.
- Option to use fixed baud rate or programmable baud rate.
- Full Duplex serial port means that it can transmit and receive data simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the SBUF register.
- However, if the first byte still hasn't been read by the time reception of the second byte is complete, one of the bytes will be lost. The serial port receive and transmit registers are both accessed at Special Function Register SBUF.

SERIAL PORT CONTROL SFRS

- The serial port of 8051 is controlled by two registers in SFR area of 8051 as shown in Table
- The two registers are Serial Port control registers; SCON and serial port buffer register SBUF.

SFR Name	Description	SFR Address
SCON	Serial port control register	98h
SBUF	Serial port buffer register	99h

- In addition to the above two registers, the MSB of PCON register named as SMOD bit is used to double the baud rate of serial transmission and reception.
- If SMOD bit is set to 1, then the baud rate is doubled.

SCON:

Bit	Name	Explanation of Function
D7	SM0	Serial port mode select bits
D6	SM1	
D5	SM2	Multiprocessor Communications Enable bit
D4	REN	Receiver Enable. This bit must be set in order to receive characters.
D3	TB8	Transmit bit 8. The 9th bit to transmit in mode 2 and 3.
D2	RB8	Receive bit 8. The 9th bit received in mode 2 and 3.
D1	TI	Transmit Interrupt Flag. Set when a byte has been completely transmitted.
D0	RI	Receive Interrupt Flag. Set when a byte has been completely received.

Definition of Bits SM0 and SM1 in SCON SFR

SM0	SM1	Serial Mode	Explanation	Baud Rate
0	0	0	8-bit Shift Register	FOSC / 12
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	FOSC / 64 Or FOSC/32
1	1	3	9-bit UART	Variable

OPERATING MODES

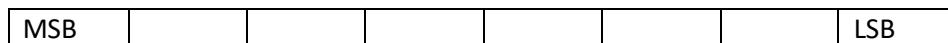
- Bits SM0 and SM1 are used to set the serial mode to a value between 0 and 3.
- Selecting the Serial Mode selects the mode of operation (8-bit/9-bit, UART or Shift Register) and also determines how the baud rate will be calculated.
- In modes 0 and 2 the baud rate is fixed based on the oscillator’s frequency. In modes 1 and 3 the baud rate is variable based on Timer 1 overflows.

Mode 0 (8 bit shift register):

- In this mode serial data is entering and exiting through RxD pin.
- So, in mode 0, Full duplex is not possible – meaning that both transmission and reception cannot take place simultaneously.
- TxD pin outputs the shift clock. 8 bits are transmitted/received (LSB first).

$$\text{Baud rate} = \text{Fosc}/12 \quad \text{where Fosc is the oscillator frequency}$$

- Transmission is started by writing a data byte to the SBUF register and once the transmission is complete, TI flag is set.
- Reception is started by placing a data byte to the SBUF register and once the reception is complete, RI flag is set.

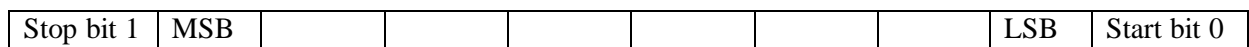


8 bits transmitted in mode 0 of serial transmission with LSB first

Mode 1 (8 bit UART):

- In this mode, 10 bits are transmitted through TxD and simultaneously 10 bits can be received through RxD.
- The 10 bits are made up of a start bit (0), 8 data bits (LSB first), and a stop bit (1).
- On completion of reception, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable and is set by the Timer 1 overflow rate. The baud rate for mode 1 is fixed at the following rate.

$$\text{Baud rate} = 2^{\text{SMOD}}/32 \times (\text{Timer 1 overflow rate})$$

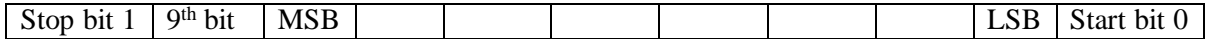


10 bits transmitted in mode 1 of serial transmission

Mode 2 (9 bit UART):

- In this mode, 11 bits are transmitted through TxD or received through RxD.
- The 11 bits are made up of one start bit (always 0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (always 1).
- The 9th data bit transmitted is same as TB8 bit in SCON special function register. It can be assigned the value of 0 or 1 by the programmer.
- On reception, the 9th data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored.
- The baud rate is programmable to either 1/32 or 1/64 of the oscillator clock frequency.

$$\text{Baud rate} = 2^{\text{SMOD}}/64 \times (\text{Oscillator Frequency})$$



11

bits transmitted in mode 2 of serial transmission

Mode 3 (9 bit UART):

- In this mode, 11 bits are transmitted through TxD and simultaneously 11 bits are received through RxD.
- The 11bits are made up of a start bit (0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except baud rate.
- The baud rate in Mode 3 is variable. The baud rate for mode 3 is fixed at the following rate similar to mode1.

$$\text{Baud rate} = 2^{\text{SMOD}}/64 \times (\text{Timer 1 overflow rate})$$



11 bits transmitted in mode 3 of serial transmission

SBUF

- SBUF is physically two registers with the same address.
- When data to be transmitted is written to the SBUF register, then it will be shifted bit by bit into the TXD line of 8051. The port 3 pin 3.1 acts as the TXD line. The shifting is done by the transmit clock which determines the baud rate.
- Similarly, when the data bits are received on the RXD line (Pin 3.0 of port 3), the bits are shifted serially into the shift register according to the Receive clock.
- After the reception is complete, the data received will be placed on SBUF from where it can be read by the programmer through the internal bus.

