MODULE 6 – 8051 ADDRESSING MODES & INSTRUCTION SETS

8051 Addressing Modes

- It is the way in which the instruction is specified or different ways that are used by the processor to access or store data.
- 8051 provides 5 addressing modes.

1. Immediate
2. Direct
3. Register
4. Register Indirect
5. Indexed

1. Immediate Addressing Mode

- Here the source operand is constant.
- Immediate data is preceded by the sign ' # '
- This addressing mode can be used to load information in to any of the registers.
- Eg: MOV A, #25H; (25H is load to register A)
  MOV R4, #64H;
  MOV DPTR, #40H

2. Direct Addressing Mode

- In this mode the direct address of memory location is provided in instruction to fetch the operand.
- Only the internal RAM and SFR's address can be used in this type of instruction.
- Eg: MOV A, 30H; (content of RAM address 30H is copied in to Accumulator)

  MOV 56H, A; (save the content of A into RAM location 56H)

  MOV R6, R2; (it is invalid)

3. Register Addressing Mode

- Here the operand is 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.
- Eg: MOV A, R0 ; (content of R0 register is copied in to Accumulator)

  MOV DPTR, A (Invalid)

4. Register Indirect Addressing Mode

- Here the address of memory location is indirectly provided by a register
- The " @" sign indicates that the register hold the address of memory location , ie. Fetch the content of the memory location whose address is provided in register.
- Eg: 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 located in program code ROM of 8051
- Here the address of memory is indexed, ie, added to form the actual address of memory.
- Eg: MOVC A, @A+ DPTR (Here C means code. The content of A register is added with content of DPTR and the result is copied to A register)
- Because the data elements are stored in the program space ROM of the 8051, it uses the instruction MOVC instead of MOV

8051 instruction sets

DATA TRANSFER	ARITHMETIC	LOGICAL	BOOLEAN	PROGRAM BRANCHING
MOV	ADD	ANL	CLR	LJMP
MOVC	ADDC	ORL	SETB	AJMP
MOVX	SUBB	XRL	MOV	SJMP
PUSH	INC	CLR	JC	JZ
POP	DEC	CPL	JNC	JNZ
XCH	MUL	RL	JB	CJNE
XCHD	DIV	RLC	JNB	DJNZ
	DA A	RR	JBC	NOP
		RRC	ANL	LCALL
		SWAP	ORL	ACALL
			CPL	RET

7

- An 8051 Instruction consists of an Opcode followed by Operand(s) of size Zero Byte, One Byte or Two Bytes.
- The Op-Code part of the instruction contains the Mnemonic, which specifies the type of operation to be performed. All Mnemonics or the Opcode part of the instruction are of One Byte size.
- Coming to the Operand part of the instruction, it defines the data being processed by the instructions. The operand can be any of the following:

• Data value
• I/O Port
• Memory Location
• CPU register

8

- There can multiple operands and the format of instruction is as follows:

  MNEMONIC DESTINATION OPERAND, SOURCE OPERAND

- Based on the operation they perform, all the instructions in the 8051 Microcontroller Instruction Set are divided into five groups. They are:

1. **Data Transfer Instructions**
2. **Arithmetic Instructions**
3. **Logical Instructions**
4. **Boolean or Bit Manipulation Instructions**
5. **Program Branching Instructions**

1. Data Transfer Instructions

- Data transfer instructions are associated with transfer of data between registers or external program memory or external data memory

Sl. No	Instruction Format	Function Performed
1.	MOV dest, src	Copy the Content of Source to Destination
2.	MOVX dest, src	Used to move to/from external data memory only
3.	MOVC dest, src	Used to move from program memory (ROM) Only
4.	PUSH src	Copies one byte from source to stack top
5.	POP dest	Copies one byte from stack top to destination
6.	XCH	Exchanges data between two sources
7.	SWAP A	Exchanges upper and lower nibbles of A

2. Arithmetic Instructions

- It performs several basic operations such as addition, subtraction, division, multiplication etc. After execution , the result is stored in the first operand.

Sl. No	Instruction Format	Function Performed
1.	ADD A ,Rn	Adds the register to the accumulator
2.	ADDC A ,Rn	Adds the register to the accumulator with a carry flag
3.	SUBB A ,Rn	Subtracts the register from the accumulator with a borrow
4.	INC A	Increments the accumulator by 1
5.	DEC A	Decrements the accumulator by 1
6.	MUL AB	Multiplies A and B
7.	DIV AB	Divides A by B
8.	DA A	Decimal adjustment of the accumulator according to BCD code
9.	CLR A	Clear the value in A(a=0)
10.	CJNE dest,src,target	compare the Source and Destination, and jump to target if they are not equal

3. Logical Instructions

- Logical Instructions perform logical operations

Sl. No	Instruction Format	Function Performed
1.	ANL dest,src	Logically AND the source and the Destination
2.	ORL dest,src	Logically OR the source and the Destination
3.	CPL dest	Complement- Logically NOT the destination
4.	XRL dest,src	Logically XOR the source and the Destination

4. Boolean or Bit Manipulation Instructions

- Boolean or Bit Manipulation Instructions will deal with bit variables.
- There is a special bit-addressable area in the RAM and some of the Special Function Registers (SFRs) are also bit addressable.

Boolean or Bit manipulation instructions are:

a) Data manipulation instructions
b) Port instructions
c) Complement instructions
d) Rotate instructions

a) Data Manipulation Instructions

a. Data Manipulation Instructions

Sl. No	Instruction Format	Function Performed
1.	SETB Bit	Set the indicated Bit
2.	CLR Bit	Clear the indicated Bit
3.	CPL Bit	Complement the Indicated Bit
4.	MOV C, Bit	Move the indicated bit to C(Carry Flag)
5.	MOV Bit, C	Move the C(Carry Flag) to indicated bit
6.	ANL C, Bit	Move to C(Carry Flag) the logical AND of C and the indicated bit
7.	ANL Bit, C	Move to the bit , the logical AND of C and the indicated bit
8.	ORL C, Bit	Move to C(Carry Flag) the logical OR of C and the indicated bit
9.	ORL Bit, C	Move to the bit , the logical OR of C and the indicated bit

b) Port Instructions

- Ports can only take data in (from outside) or put data out.
- There are no special instructions for ports in 8051, but it uses MOV instructions for this.

Ex: **MOV A, P_1** (copy the content of port1 to accumulator)

MOV P_2, R_1 (copy the content of Register R1 to port2)

C) Complement Instructions

- Here only A register and direct RAM address can be used as the destination.

Ex: **CPL A** (Complement the contents of A)

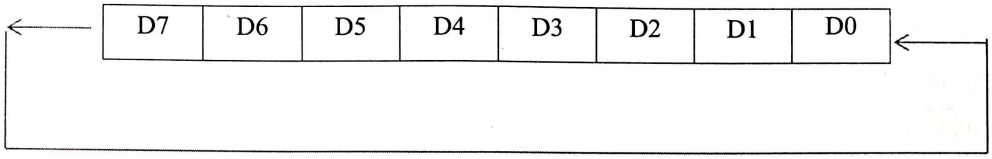
CPL 43H (Complement the contents of RAM address 43H)

d) Rotate Instructions

- There are 4 rotate and 1 swap instructions for the chip.
- Only the A register can be used as the destination

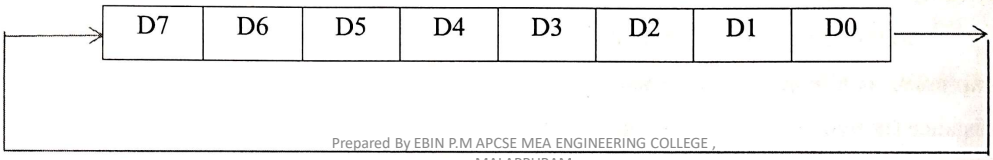
i) **RL A ; Rotate left the content of A**

The A register is rotated left and D7 appears in the D0 position after rotation.



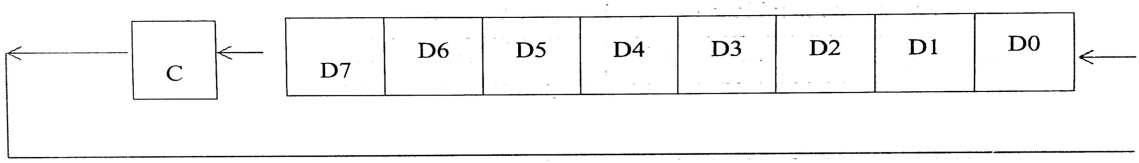
ii) **RR A ; Rotate right the content of A**

The A register is rotated right and the bit D0 is moved to D7



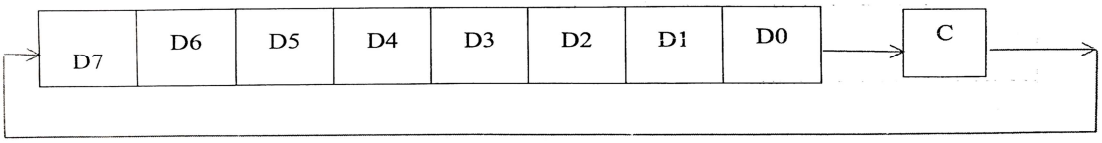
iii) RLC A ; Rotate left through carry

The A register is shifted left and bit D7 is moved to carry, while the carry bit is moved to D0 of A

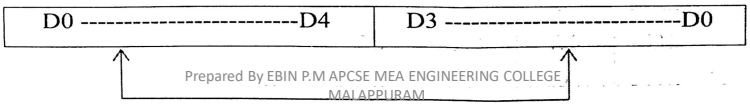


iv) RRC A ; Rotate RIGHT through carry

The A register is shifted right and bit is moved to D7, while the D0 moves into the carry.



v) SWAP A ; This is a special instruction where the lower and upper nibbles of A are swapped.



5. Program Branching Instructions

- Flow control instructions are used to divert the flow of the program
- These instructions are used to implement loops and subroutine calls
- Branch instructions change the flow of a program by modifying the program counter (PC)

❖ Unconditional Branch Instructions:

1) SJMP Target

- SJMP stands for Short Jump and it is a relative jump
- Here the destination of jumping is expressed as a relative number and it will be short.ie, 8 bits only
- This 8 bit number represents distance between current PC value and target address

- If this number is negative it is backward and the maximum range of which is **-128**
- If it is a forward jumping the number is positive with maximum range of **+127**

2) LJMP Target

- This is a Long Jump instruction and is not relative
- It is a three byte instruction
- When this instruction is executed, the current PC value is simply replaced by the 16 bit number in the instruction, which may vary from 0000 to FFFF.

3) AJMP Target

- This is an Absolute Jump instruction and also a relative jump.
- Here the range of jump is 2K and 11 bits specify the destination range.

❖ Conditional Branch Instructions :

- These are the instructions which make the programs really useful.
- Computers are used for repetitive and conditional tasks and conditional branching is the method for it.
- **All conditional jumps are short jumps**

Sl. No	Instruction Format	Function Performed
1.	JC target	Jump if CY=1
2.	JNC target	Jump if CY=0
3.	JZ target	Jump if the register A=0
4.	JNZ target	Jump if the register A is not zero
5.	JB bit, target	Jump if the bit=1
6.	JNB bit, target	Jump if the bit=0
7.	JBC bit, target	Jump if the bit=1. Then clear bit
8.	DJNZ byte, target	Decrement the byte, and jump if the byte is zero

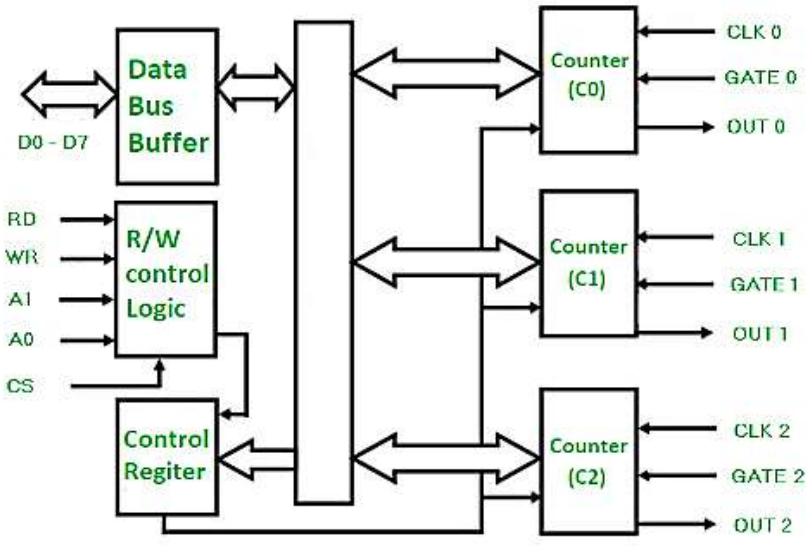
Peripheral chips for timing control – 8254/8253

- The Intel 8253 and 8254 are programmable Interval Timers (PTIs) designed for microprocessors to perform timing and counting functions using three 16 bit registers.
- Each counter has 2 input pins, i.e, Clock & Gate, and one pin for "OUT" output.
- To operate a counter , a 16-bit count is loaded in its register.
- On command, it begins to decrement the count until it reaches 0, then it generates a pulse that can be used to interrupt the CPU

8254 programmable interval timer

- 8254 is a device designed to solve the timing control problems in a microprocessor.
- It has 3 independent counters, each capable of handling clock inputs up to 10 MHz and size of each counter is 16 bit.
- It operates in +5V regulated power supply and has 24 pin signals.
- All modes are software programmable.
- The 8254 is an advanced version of 8253 which did not offered the feature of read back command.

• 8254 Architecture



Difference between 8253 and 8254

8253	8254
• Its operating frequency is 0 - 2.6 MHz	Its operating frequency is 0 - 10 MHz
• It uses N-MOS technology	It uses H-MOS technology
• Read-Back command is not available	Read-Back command is available
• Reads and writes of the same counter cannot be interleaved.	Reads and writes of the same counter can be interleaved.

MODES OF OPERATION

- 8253/8254 can be operated in 6 different modes.

1) Mode 0 – Interrupt on Terminal Count

- It is used to generate an interrupt to the microprocessor after a certain interval.
- Mode 0 is typically used for event counting.
- After the Control Word is written, OUT is initially low, and will remain low until the counter reaches zero
- it is decremented by 1 after every clock cycle. OUT then goes high and remains high until a new count or a new Mode 0 Control Word is written into the counter.
- GATE = 1 enables counting, GATE = 0 disables counting.

2) Mode 1 - Programmable one shot mode

- It can be used as a mono stable multi-vibrator.
- The gate input is used as a trigger input in this mode
- The output remains high until the count is loaded and a trigger is applied

3) Mode 2- Rate Generator

- Initially value of OUT is low.
- When counting is enabled, it becomes high and this process repeats periodically.
- Value of count = Input Frequency / Output Frequency.
- This mode works as a frequency divider.

4) Mode 3 – Square Wave Generator

- Counting is enabled when GATE = 1 and disabled when GATE = 0.
- This mode is used to generate square waveform and time period (equal to count) is generated.
- If N is count and is even then ontime of wave = N/2 and offtime = N/2
- If N is odd the ontime = (N + 1) / 2 and offtime = (N – 1) / 2

5) Mode 4- Software Triggered Mode

- In this mode counting is enabled by using GATE = 1 and disabled by GATE = 0.
- Initially value of OUT is high and becomes low when value of count is at last stage. Count is reloaded again for subsequent clock pulse.

6) Mod 5- Hardware Triggered Mode

- OUT will initially be high.
- Counting is triggered by a rising edge of GATE.
- When the initial count has expired, OUT will go low for one clock pulse and then go high again.
- After writing the Control Word and initial count, the counter will not be loaded until the clock pulse after a trigger.