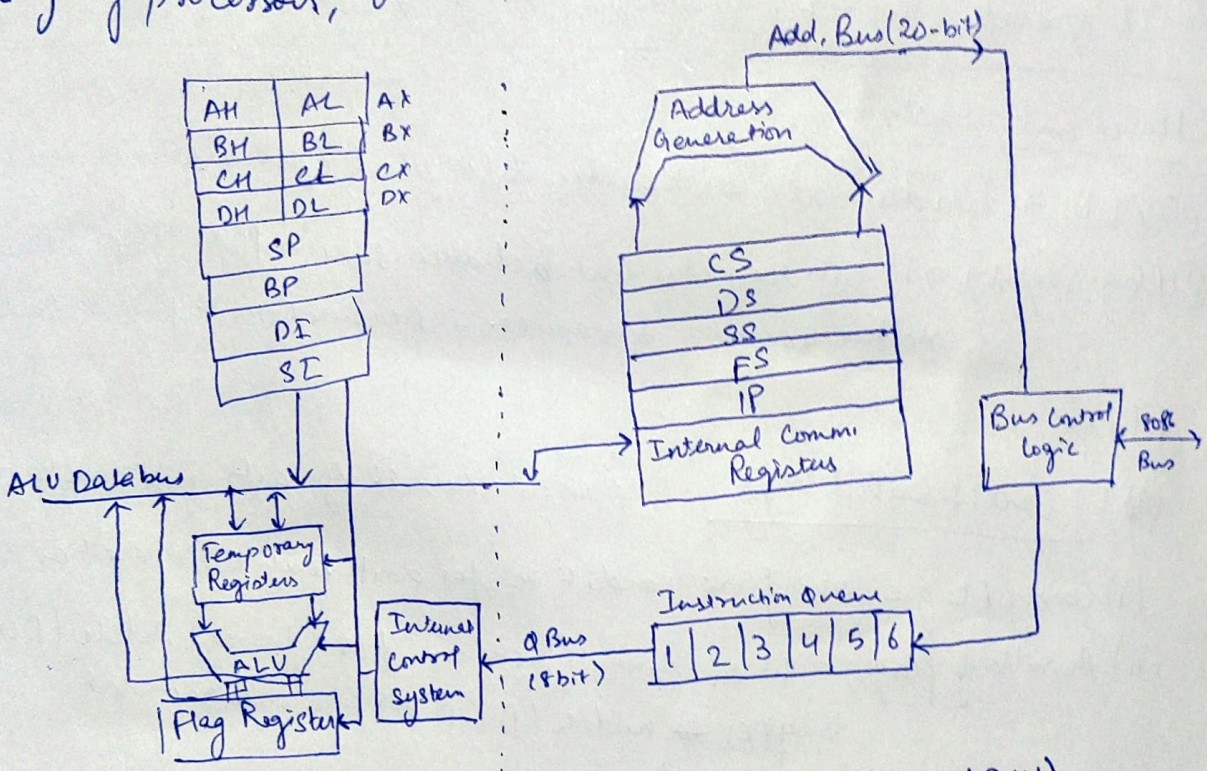
21BCB0069

BECE204L Digital Assignment-1: MPMC

① Elaborate 8086 Architecture, addressing modes, Programming with DOS and BIOS function calls, min & max mode configuration

8086 Architecture.

The 8086 microprocessor is a 16 bit microprocessor that was introduced by Intel in 1978. It is the first member of x86 family of processor, which is in use till date.



EU executes instructions that have already been fetched by the BIU.

BIU fetches instructions, reads data from memory and I/O ports, writes data to memory & I/O ports.

BIU & EU functions seperately.

Execution unit - The main components of EU are General purpose registers, the ALU, Special purpose registers, the IR's and Ins. decoder and the flag/status Reg.

Function of EU:

① Fetches instructions from the queue in BIU, decodes, and executes arithmetic and logic operations using the ALU.

② Sends control signals for internal data transfer operations within the microprocessor (control unit)

③ ~~Sends control signals for internal data transfer operations within the microprocessor (Control unit).~~

④ Sends requests signals to BIU to access the external modules

⑤ It operates w.r.t T-states (clock cycles) & not MC.

Functional parts

① ALU → handles all arithmetic & logical operations.

② Flag register → 16 bit register that behaves like a flip-flop, changes status according to result in Accumulator 9 flags - Conditional Co

Flags

I) Conditional Flags - represent result of last instruction

(i) Carry Flag - overflow condition for arithmetic operation

(ii) Auxiliary flag - when carry / borrow from lower nibble (D0-D3) to upper nibble (D4-D7), this flag is set.

(Carry given by D3 to D4).

(iii) Parity flag - lower order 8 bits of result contains even no. of 1's → Flag set
odd no. of 1's → Parity flag is cleared

(iv) Zero flag - Set to 1 when result of operation is 0.

(v) Sign flag - Holds sign of result - i.e. -ve value set to 1, else 0

(vi) Overflow flag - Represents the result when system capacity is exceeded.

Reg no. 21BCB0069 Name. VISHAL AGRAWAL

II) Control Flags - controls the operations of execution unit.

(i) Trap flag - It is used for single step control and allows the user to execute one instruction at a time for debugging. If it is set, then the program can be run in single step mode.

(ii) Interrupt flag - Interrupt enable = 1, disable = 0. allows interrupt.

(iii) Direction flag - Used in string operation, 1 = String byte are accessed from higher memory address to lower memory add.

Source Index (SI) and Destination Index (DI)

- used in indexed addressing

Instructions that process data strings use the SI and DI registers together with DS and ES respectively in order to differentiate between source & destination address

General Purpose registers

- 8 general purpose regs. AH, AL ; BH, BL ; CH, CL ; DH, DL
  - AX → Accumulator register (stores operands)
  - BX → base register (storing base address)
  - CX → counter
  - DX → hold I/O port address for I/O instruction

Stack Pointer Reg - holds address from the start of the segment

Bus Interface Unit (BIU) → It provides the interface of 8086 to external memory & I/O devices via the system bus. It performs various machine cycles such as memory read, I/O read, etc to transfer data between memory & I/O.

functions → generates 20-bit physical address for memory access.
→ fetches instructions from memory
→ transfers data to and from memory and I/O.
→ maintains 6 byte for fetch instruction queue.

Reg no. 21BCB0069

Name - VISHAL ANBARASU

BIU →

- **Instruction queue** – BIU contains instruction queue, gets upto 6 bytes of next inst. & stores in queue. Fetching the next inst. while current execution executes is Pipelining.
- **Segment registers** – BIU has 4 segment buses. It holds the addresses of inst. and data in memory, used by the processor to access memory locations

CS – Code segment
DS – Data segment – (data used by program)
SS – Stack segment (store data and address
ES – Extra segment.

- **Instruction pointer** – hold the address of the next inst. to be execute

---

Addressing modes →

Way of specifying data to be operated by an instruction.

① **Register** – name of register in instruction MOV CL, DH

② **Immediate** – 8bit/16bit data is specified in instruction
MOV DL, 08H

③ **Direct Addressing** – effective address of memory location at which data operand is stored.
eg → MOV BX, [1354H]

④ **Register Addressing** (Indirect) – Content of DS reg used for base add. calculation
MA = BA + EA (BA = DS × 16_{10})

⑤ **Based Addressing** – BX or BP used to hold effective address & signed 8-bit / unsigned 16bit.

EA = BX + 0008H
BA = DS × 16_{10}
MA = BA + EA.

Reg. no. 21BBS0069 Name – VISHAL AGRAWAL

6) Indexed addressing - SI or DI reg used to hold an index value for memory data and a signed 8/16 bit displacement will be specified in the inst.

MOV CX, [SI + 0A2H]

EA = SI + FFA2
BA = (DS) X 16_{10}
MA = BA + EA

7) Based Index Addressing → EA = BX + SI + 0D0AH
BA = DS X 16_{10}
MA = BA + EA.

8) String Addressing → to operate on string data

9) Direct I/O port Addressing → eg IN AL, [09H]

10) Indirect I/O port Addressing →

11) Relative Addressing - EA is relative to IP

12) Implied addressing - no operands, instruction itself specify

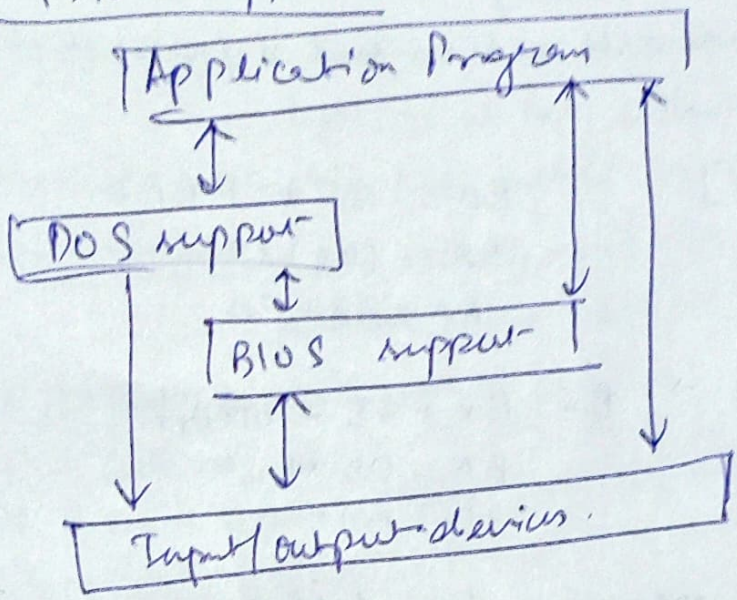
---

Min Mode - Microprocessor do not associate with any co-processors & cannot be used for multiprocessor system.

signals on Pins 24-31 ; MN/$\overline{MX}$ to logic high.
simple ckt, lower performance

Max Mode - 8086 can work in multi-processor or co-processor config ; MN/$\overline{MX}$ to ground.
Pins 24-31 reassigned.
Complex ckt, high performance.

DOS & BIOS support



Data input & output through the keyboard are most commonly used functions.

(DOS) — Disk Operating System Interrupts. INT 21H.

(BIOS) → INT 10H subroutines are burned into the ROM BIOS of 80% based and compatibles and are used to communicate with computer like screen video. Much of the manipulation of screen text or graphics is done through INT 10H.

Question 2) Assume data in Accumulator. Check MSB → if 1, transfer data in ROM starting from 200H to 20FH to internal RAM starting from 50H to 5FH.

```
        ORG  0000H
        ADD  A, #80H        ; adds 80H, if data has MSB 1,
        JNC  HERE           ; generates carry & executes.
        MOV  DPTR, #0200H
        MOV  R0, #0FH
        MOV  R1, #50H
LOOP:   CLR A               ; Loop for data transfer.
        MOVC A, @A+DPTR
        MOV  @R1, A
        INC  DPTR
        INC  R1
        DJNZ R0, LOOP
HERE:   SJMP HERE
        ORG 0200H
        DB "HELLO MY NAME IS "; data to be transffered
        END
```

Question 3) If ALE freq. = 11.0592 MHz.

```
DELAY:  MOV R0, #200        (1)
AGAIN:  MOV R1, #250        (1)
HERE:   NOP                 (1)
        NOP                 (1)
        DJNZ R1, HERE       (2)
        DJNZ R0, AGAIN      (2)
        RET                 (2)
```

$$\left(\{[4\times250]+1+2\}\times200\right)+1+2$$

$$=(1003\times200)+1+2$$

$$=200603$$

Time period for 1 m.c $= \frac{12}{11.0592\times10^6}$

$= 1.0854\,\mu s.$

Tot. time delay

$= 0.218\,s.$

Question 4)

```
XX:  MOV  R0, # 50H
     MOV  50H, #25
     MOV  A, @R0
     JZ   XX
     INC  R0
     MOV  A, @R0
     END
```

R0 : ~~50H~~ 51H
[50H] : 25
~~R0~~ [51H] : undefined.

Question 5) Assume string of data stored in code space starting at address 200H, transfer string in reverse order to RAM location starting 40H.

```
     ORG  0000H
     MOV  DPTR, #0200H
     MOV  R0, #40H
     MOV  R1, #0EH
LOOP: CLR  A
     ADD  A, R1
     SUBB A, #01H
     MOVC A, @A+DPTR
     MOV  @R0, A
     INC  R0
     DJNZ R1, LOOP
HERE: SJMP HERE
     ORG 0200H
     DB "VIT UNIVERSITY"
     END
```

Question 6) Write an 8051 ALP to generate 60% duty cycle on P1.2

```
        ORG 0000H
        CLR P1.2
        MOV TMOD, #01H
HERE:   SETB P1.2
        ACALL DELAY
        ACALL DELAY
        ACALL DELAY
        SJMP HERE
DELAY:  CLR P1.2
        ACALL DELAY
        ACALL DELAY
        SJMP HERE
DELAY:  MOV TH0, #0FFH
        MOV TL0, #0F0H
        SETB TR0
BACK:   JNB TF0, BACK
        CLR TR0
        CLR TF0
        RET
        END
```

Question 7) Switch is connected to port P2.3. Let port P1 & P0 are each connected with 8 LEDs. WAP 8051 ALP

(a) Switch high → turn ON LED on Port 0 & OFF LEDs on Port 1

(b) else turn OFF on Port 0 & ON LEDs Port 1

```
SW1   EQU P2.3
      ORG 0000H
HERE: JB  SW1, NEXT
      MOV A, #0FFH
      MOV P1, A
      MOV A, #00H
      MOV P0, A
      SJMP HERE
NEXT: MOV A, #0FFH
      MOV P0, A
      MOV A, #00H
      MOV P1, A
      SJMP HERE
      END
```


Name - VISHAL AGRAWAL

Question 8) Write an ALP to Get Data (Rgno. 21BCB00069) from Port P1 & send it to Port P2

P1: Input ; P2: Output Port.

Normal →

```
        ORG   0000H
        MOV   A, #0FFH
HERE:   MOV   P1, A
        MOV   A, P1
        MOV   P2, A
        SJMP  HERE
        END
```

Serial Comm →

```
        ORG   0000H
        MOV   TMOD, #20H
        MOV   TH1, #-3
        MOV   SCON, #50H
        SETB  TR1
        MOV   A, #0FFH
        MOV   P1, A
HERE:   JNB   RI, HERE
        MOV   A, SBUF
        MOV   P2, A
        MOV   A, P1
        MOV   P2, A
        CLR   RI
        SJMP  HERE
        END
```

Question 9) WAP to read the temp. through any port and test it for value 75. Accordig to the test results, place the temprature value into the registers indicated by the following.

If T = 75 then A = 75H

T < 75 then R1 = T

T > 75 then R2 = T

```
        ORG  0000H
        MOV  A,P1
        MOV  R0,A
        SUBB A,#4BH
        JZ   EQUAL
        JNC  MORE
        MOV  A,R0
        MOV  R1,A
        MOV  A,#00H
        MOV  R0,A
        MOV  R2,A
        SJMP HERE
MORE:   MOV  A,R0
        MOV  R2,A
        MOV  A,#00H
        MOV  R0,A
        MOV  R1,A
        SJMP HERE
EQUAL:  MOV  A,R0
        MOV  R0,#00H
        MOV  R1,#00H
        MOV  R2,#00H
        SJMP HERE
HERE:   SJMP HERE
        END
```

Question 10) Draw Memory Interface to 8086 and explain.

Design – 8086 in min mode with 8MHz
- 64KB EPROM using 32KB EPROM
- 128 KB RAM using 64KB RAM.

64KB EPROM with 32 KB EPROM

- no. of chips = 2 chips → one chip for 32KB for even, another for odd.
- Address lines ⇒ for 64KB = $2^{10} \times 2^6 = 2^{16}$ – 16 lines.
- Data lines ⇒ for 64KB = 8 (for byte, it is 8bits)
- Control lines ⇒ for 64KB EPROM = Memory read.

Reg no. 21BCB0069

Name – VISHAL AGRAWAL

→ 128 KB RAM using 64KB RAM

- No. of chips = 2 chips of 64KB RAM (1-even, 1-odd)
- Add. lines for 128 KB Add = $2^{10} \times 2^7 = 2^{17}$ – 17 add. lines
- Data lines for 128 KB = 8 (for byte, it is 8 bits)
- Control lines for 128 KB RAM = 2 (Memory Read / write)

Chip Select-

Memory IC	A19	A18	A17	A16	A15	..... A10	..... A2	A1	A0	Address
EPROM 1 [Lower Byte]	1	1	1	1	0	0 0 . . . 0	0 . . . 0	0	0	F 0000 H
	1	1	1	1	1	1 . . . 1	1 . . . 1	1	0	F FFFEH
EPROM 2 [Higher Byte]	1	1	1	1	0	0 . . . 0	0 . . . 0	0	1	F000 1H
	1	1	1	1	1	1 . . . 1	1 . . . 1	1	1	FFFFH
RAM 1 [Lower Byte]	0	0	0	0	0	0 . . . 0	0 . . . 0	0	0	0000 0H
	0	0	0	1	1	1 . . . 1	 1	1	0	1FFFEH
RAM 2 [Higher byte]	0	0	0	0	0	0 . . . 0	0 . . . 0	0	1	0000 1H
	0	0	0	1	1	1 . . . 1	1 . . . 1	1	1	1FFFFH