www.studentsuvidha.in

B.E.

Fifth Semester Examination, May-2009

Micro Processor (EE-309-C)

Note : Attempt any five questions. All questions carry equal marks.

Q. 1. (a) Draw and explain the architecture of 8085 microprocessor.

Ans. 8085 Microprocessor :

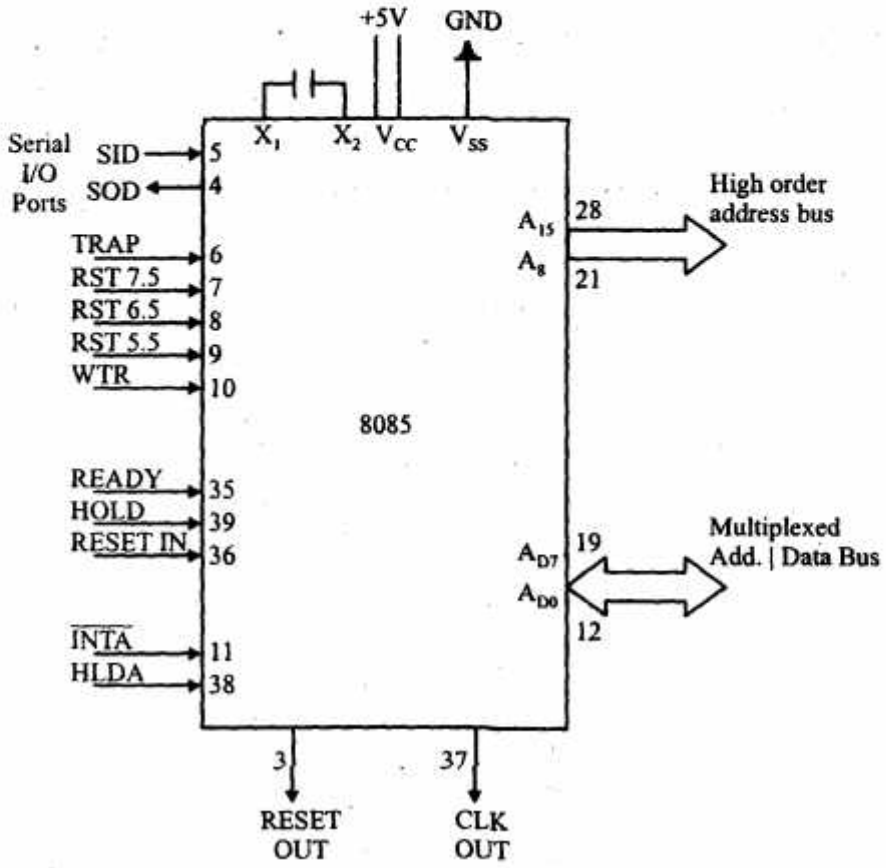
(i) 8085A/8085 is an 8 bit general purpose microprocessor capable of addressing 64 K of memory. The device has 40 pms, requires +5V single power supply, and can operate with a 3 MHz single phase clock.

(ii) 8085 has 16 signal lines/pins that are used as address bus, as $A_{15} - A_8$ and $A_{D_7} - A_{D_0}$

$V_{CC} = +5V$

Δ_{SS} = Ground reference

$CLK(OUT)$ = Clock output



www.studentsuvidha.in

Control & Status Signals :

ALE : — Address latch enable
— +ve going purse
— Generated every time 8085 begins an operation.

$\overline{RD}$: — Read
— Read control signal
— Active low

$\overline{WR}$: — Write active low

IO/$\overline{M}$: — Status signal
— Differentiates between output and memory operations.

S_1, S_0 : — Status signals
— Identify various operations.
— Rarely used in small systems.

Interrupts :

INTR (input)	—	Interrupt request
$\overline{INTA}$ (output)	—	Interrupt acknowledge
RST 7-5	—	Restart interrupts
RST 6-5	—	..... do .....
RST 5-5	—	.....do .....
TRAP (input)	—	Non markable interrupt (highest priority)
HOLD (input)	—	DMA controller access indicator

Q. 1. (b) Write an assembly language program using 8085 microprocessor to find the number of 1's and number of 0's.

Ans

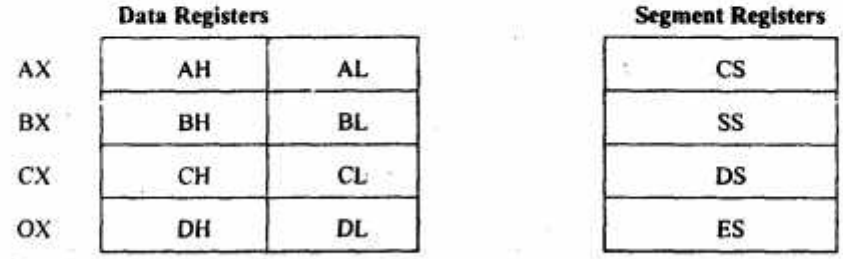
Instructions	Memory address	Hexadecimal code	
LDA 2051H	2030	3A	copy the first byte
	2031	51	
	2032	20	
NOV B, A	2033	47	
LDA 2052 H	2034	3A	
	2035	52	
	2036	20	
SUB B	2037	90	
STA 2053H	2038	32	saving result
	2039	55	

www.studentsuvidha.in

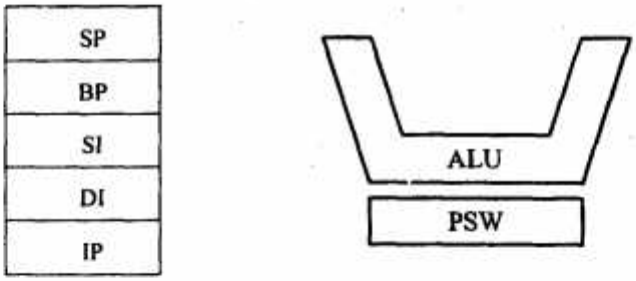
	203A	20	
HLT or RST7	203B	76	FF EOP
	2051	49	(These data types must be manually loaded, they are not part of writing the program)
	2052	9F	
	2053	0D	

Q. 2. (a) Explain the register structure of 8086 microprocessor.

Ans.



Pointers :



Address / data = 20 pins
Control / = 16 pins

The data group consists of AX, BX, CX and DX registers. These registers can be used to store both operands and results and each of them can be accessed as a whole, or the upper and lower bytes can be accessed separately.

BX — May be used as base register in address calculations.

CX — Used as implied counter.

DX — Used to hold I/O address.

8086	8080
AL	A
BH	H
BL	L
CH	B

www.studentsuvidha.in

CL	C
DH	D
DL	E

341B Effective address
+ 1230A Beginning segment address
157BB Physical address of instruction.

(i) Segment registers allow the memory capacity to be 1MB even though the addresses association with the individual instructions are only 10 bits wide.

(ii) They permit a program and/or its adata to be put into different areas of memory each time the program is executed.

Q. 2. (b) Draw and explain timing diagram for 8086 memory read bus cycle for minimum mode of operation.

Ans. The execution time of an instruction can be determined by multiplying the number of clock cycles needed to execute the instruction by the clock's period.

It can be expressed as the sum of a basic execution time plus the time required to calculate the effective addressing as memory operand is involved.

The basic execution time assumes that the instruction to be executed hs already been prefetched and stored in the instruction queue; otherwise any additional clock cycles necessary to fetch the instruction must be added.

Examples of Execution Times :

ADD or SUB

R to R	3	0
M to R	9 + EA	1
R to M	16 + EA	2
I to R	4	0
I to M	17 + EA	2

MOV :

A to M	10	1
M to A	10	1
R to R	2	0
M to R	8 + EA	1
I to R	9 + EA	1
I to M	4	0
R to SR	10 + EA	1
M to SR	8 + EA	0
SR to R	2	1
SR to M	9 + EA	0

www.studentsuvidha.in

Some instructions have variable basic execution times which are data dependent typical examples are multiply divide, shift and rotate instructions.

Q. 3. (a) Using 8086 microprocessor write an assembly language program to find the smallest number.

Ans.

	Mneumonics	Machine Code	Memory Address
1 Start :	INF1H	DB	2000
		F1	2001
2	MOV B, A	78 1	2002
3	INF2H	DB	2003
		F2	2004
4	AMI 80H	E6	2005
		80	2006
5	MOV GA	4F	2007
6	MOV A, B	78	2008
7	AMI 80H	E6 2	2009
8	ANA C	A1	200A
9	JN2 SHTDWN	C2 3	200B
		20	200C
10	MOV A, B	14	200D
11	AMI 1FH	78	200E
		E6	200F
12	OUT F3H	IF	2010
		D3	2011
13	JMP START	F3	2012
14	SHTDWN MVI A, 40H	C3 4	2013
		3E	2014
15	OUT F3H	4D	2015
16	HLT	D3 5	2016
		76	2017

Q. 3. (b) Explain the branch addressing modes of 8086 microprocessor.

Ans. Immediate : The datum is either 8 bits or 16 bits long and is part of the instruction.

Op code	Operand	.....	Operand

General instruction format

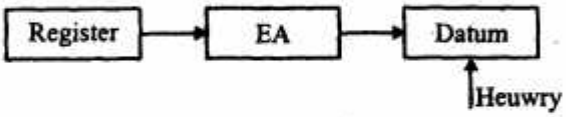
www.studentsuvidha.in

Direct : The 16 bit effective address of the datum is part of instruction.

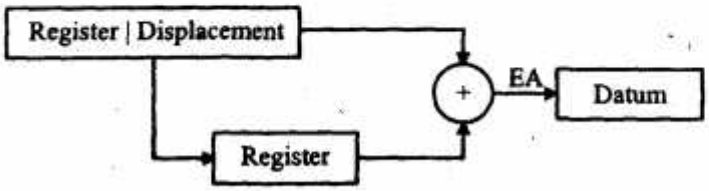
Register : The datum is in the register that is specified by the instruction.

Register Indirect :

$$EA = \begin{Bmatrix} BX \\ DI \\ SI \end{Bmatrix}$$



Register Relative :



$$EA = \begin{Bmatrix} BX \\ BP \\ SI \\ DI \end{Bmatrix} + \begin{Bmatrix} \text{8 bit displacement} \\ \text{(sign exreuded 16 bit)} \end{Bmatrix}$$

Based Indexed :

$$EA = \begin{Bmatrix} BX \\ BP \end{Bmatrix} + \begin{Bmatrix} SI \\ DI \end{Bmatrix}$$

Q. 4. Draw the interfacing diagram and address mapping for 8086 system in minimum mode with following specifications :

(a) 16KBRAM (b) 32KBROM (iii) 8255 PPI in I/O address space.

Ans. One byte instruction—Implied operand (s)

OP	CODE

One byte instruction—Register mode

OP CODE	REG

Register to Register

OP CODE	11	REG	RIM

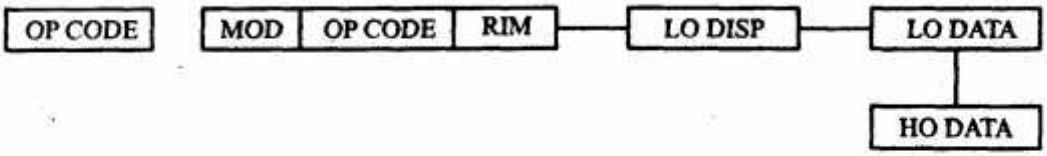
Registrer to from memory with no displacement

OP CODE	MODE	REG	RIM

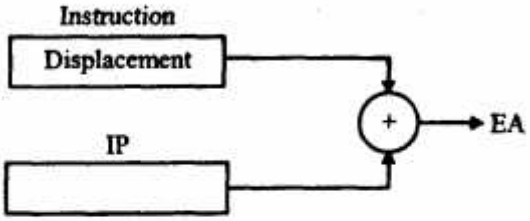
www.studentsuvidha.in

With displacement

16 bit displacement



Register address	Registers w = 1	Registers w = 0	Register Address	Segment Register
000	AX	AL	00	ES
001	CX	CL	01	CS
010	DX	DL	10	SS
011	BX	BL	11	DS
100	SP	AH		
101	BP	CH		
110	SI	DH		
111	DI	BH		



Interactive direct

MOD / RIM	00	01	10	11 w = 0	11 w = 1
000	BX + SI DS	BX + SI + DB DS	BX + SI + D16 DS	AL	AX
001	BX + DI DS	BX + DI + DB DS	BX + DI + D16 DS	CL	CX

www.studentsuvidha.in

010	BP + SI	BP + DI + DB	BP + SI + D16	DL	DX
	SS	SS	SS		
011	SI	SI + DB	BP + DI + D16	BL	BX
	DS	DS	SS		
100	DI	DI + DB	SI + D16	AH	SP
	DS	DS	DS		
101	DI	BP + DS	DI + D16	CH	BP
	DS	SS	DS		
110	D16	BX + DB	BP + D16	DH	SI
	DS	DS	SS		

The is register is always used as the segment register when computing the address of the next instruction to be executed.

When SP is used, SS is always the segment register.

For string operations are ES register is always used as segment register for the destination operand.

Q. 5. (a) Calculate the opcode of following instructions :

(i) M00 CX, [0401] in the formate of 1000 10dw mod reg r/m.

(ii) ADDCX, 1435 in the formate of 1000 00sw 000 r/m, data.

(iii) MUL CX in the formate of 1111011w mod 1000 r/m

(iv) RCLAX, 01 in the formate of 1101 00cw mod 010 r/m

Ans. (i) MOV :

MOV	CC	Bytes
Accumulator to memory	10	3
Memory to accumulator	10	3
Register to register	02	2
Memory to register	8 + EA	2-4
Immediate to memory	9 + EA	2-4

(ii) ADD :

Addition	3	2
Regsiter to register	9 + EA	2-4
Memory to register	16 + EA	2-4
Register to memory	4	3-4

www.studentsuvidha.in

(iii) MUL Unsigned Multi-location :

8 bit register	70-77	2
16 bit register	118-133	2
8 bit memory	76-83 + EA	2-4
16 bit memory	124 - 139 + EA	2-4

(iv) RCL :

Rotate flag onto stock	2	1
Left through carry	8 + 4 bit	2
Register, single shift variable shift	15 + EA	2-4
Memory, single shift variable shift	20 + EA + 4 bit	2-4

Q. 5. (b) Explain the function of following instructions with one example :
(i) LDS (ii) AAA (iii) Out using variable port addressing (iv) LODSW (v) REPNZ

Ans. (i) LDS :

LDS — Load pointer using DS

Clock cycles 4

Number of Bytes : (2-4)

Example :

```
MOV     DST, SRC
LEA     REG, SRC
LDS     REG, SRC
LES     REG, SRC
XCHG OPR1, OPR2
```

(ii) AAA : ASCII adjust for addition

CC = 4

Bytes = 1

Example :

```
MOV     AL, UP1
ADD     AL, UP2
AAA
MOV     DL, AL
MOV     AL, UP1 +1
ADC     AL, UP2 +1
AAA
XCHG    AL, DL
```

www.studentsuvidha.in

(iii) Out using Variable Port Addressing :

Output to I/O port	CC	Bytes
Fixed port	10	2
Variable port	8	1

Example :

```
OUT     PORT, AL
OUT     PORT, AX
OUT     DX, AL
OUT     DX, AX
```

(iv) LODSW :

Load word string CC
NOV Repeated 12
Repeated 9 + 13/rep.

(v) REPNZ : Repeat operation while MOV zero

REP string primitive CX = 0
REPE string primitive CX = 0, ZF = 0
or
REPZ
REPME/REPM2 string primitive CX = 0 or ZF = 1

Q. 6. Explain the various types of assembler directives. Write a sample program using assembler directives.

Ans. Assembler Directives : Assembler instructions are translated into machine language and correspond to executable statements in high level language programs.

Data Defination & Storage Allocation Statement :

Variable Mnemonic Operand,....., operand Comments

DB (Define byte)

DW (Define word)

DD (Define double world)

Alignment Directives : There are 2 directives which are used for alignment. The directive.

Forces the address of the address of the next byte to be even. It has been noted that for the 8086 words can be accessed in less time if they begin at even address.

Note that for the 8088, since each reference to a word operand register 2 bus cycles regardless of the operand address there is no advantage in using even address alignment.

The directive ORG constant operation/expression causes the next byte to be $(n+1)1-h$ byte within a segment, where n is the value of the constant expression.

Just as an END statement is needed to signal the end of a high level language program, an END directive of the form END label.

is needed to indicate the end of a set of assembler language code.

www.studentsuvidha.in

Segment Defination : Segment name SEGMENT storage defination, allocation & alignment directives.

Segment name ENDS

Example :

```
Order 1     DW              12
OPER2       DW              230
RESULT      DW              ?
ASSUME      CS : EXAMPLE    DS : EXAMPLE
START :     MOV             AX, CS
            MOV             DS, AX
            MOV             AX, OPER1
            ADD             AX, OPER2
            JGE             STORE
            MEG             AX
STORE :     MOV             RESULT, AX
            HIT
EXAMPLE     EMDS
            END             START
```

Q. 7. (a) Explain the initialization command words of 8259.

Ans. Initialization Command Words :

```
DI
MVI A, 76H      :   ICW1
OUT 80H         :   Intialize 8259A
MVI 20H         :   ICW2
OUT 81H         :   Initialize 8259A
```

The DI instruction disables the interrupts so that the initializations process will not be interrupted.

The command word 76H specifies the following parameters :

Low order byte of the IR_0 call address

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
0	1	1	0	0	0	0	0 = 60H

www.studentsuvidha.in

The address bits A_4-A_0 are supplied by the 8259A. The subsequent addresses are four locations apart

Example : $IR_1 = 64H$

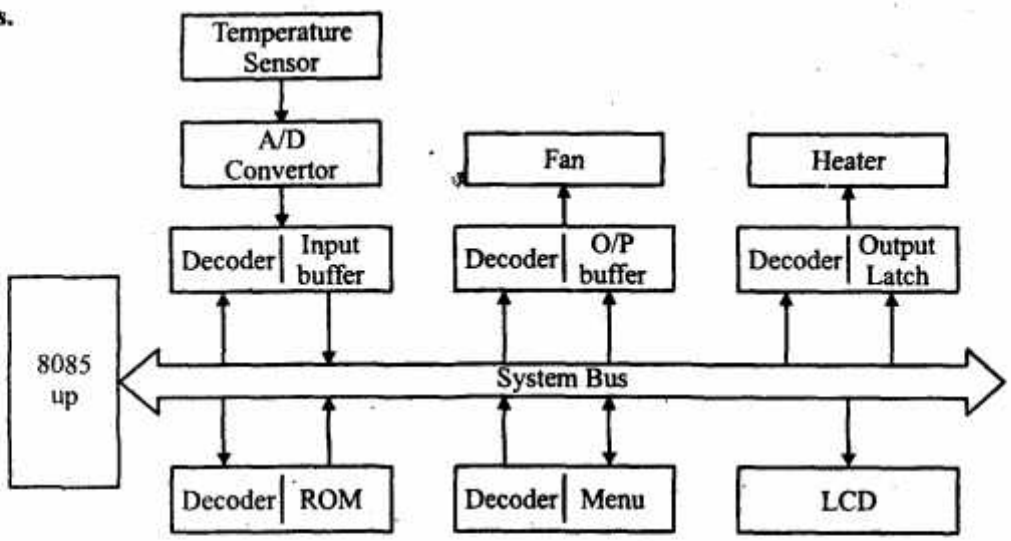
The port address of the 8259A for ICW_1 is 80H; A_0 should be at logic 0, and the other bits are determined by the decoder.

Command word ICW_2 is 20H, which specifies the high order byte of the call address.

The port ddress of ICW_2 is 81H; A_0 should be at logic 1.

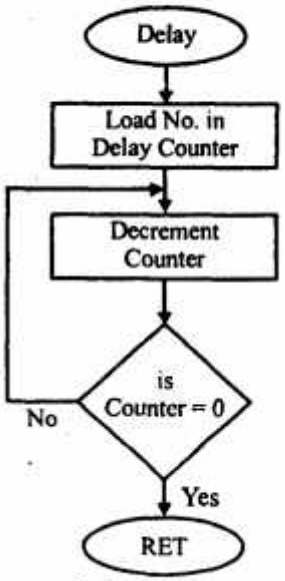
Q. 7. (b) Interface stepper motor with 8085/8086 microprocessor. Also write subroutine to rotate the motor in forward direction using full step mode.

Ans.



Example : Traffic Signal Controller

Lights	Data Bits	On time
Green	D_0	15 sec.
Yellow	D_2	5 sec.
Red	D_4	20 sec.
Walk	D_6	15 sec.
Don't walk	D_7	25 sec.



Subroutined

www.studentsuvidha.in

Address	Code	Mnemonics
XX00	31	
01	99	1X1, SP, XX99
02	XX	
03	3E	START : MUI, A, 41H
04	41	
05	DE	OUT PORT #
06	PORT #	
07	06	MVI B, OPH
08	OF	
09	CD	CALL DELA4
0A	50	
0B	XX	
0C	3E	MVI A, 84H
0D	84	
0E	D3	OUT PORT #
0F	PORT #	
10	06	MVI, B, 05
11	05	
12	CD	CALL DEAL
13	50	
14	XX	
15	3E	MVI A, 90H
16	90	
17	D3	OUT PORT #
18	PORT #	
19	06	MVI B, 14H
1A	14	
1B	CD	CALL DELA4
1C	50	
1D	XX	
1E	C3	JMP START
1F	D3	
20	XX	

www.studentsuvidha.in

Q. 8. (a) Explain the control register and status registe. of 8257.

Ans.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

BCD

0	Binary counter 16 bits
1	Binary coded decimal (BCD) 4 decades

SC	Select	Counter
SC1	SC0	
0	0	SC0
0	1	SC1
1	0	SC2
1	1	Read back command

M-Made :

M_2	M_1	M_0	
0	0	0	Mode 0
X	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

RW = Read/write :

RW1	RW0	
0	0	Counter latch command
0	1	Read/write lsb only
1	0	Read/write msb only
1	1	Read/write LSB then MSB