Name: Soham Pawe-

MPL assignment 2

Q1 List and explain the string instruction of 8086. Write 8086 Assembly Language program to display the given string in reverse order.

Ans = 1) REP / REPE / REPNE / REPNZ instructions

Syntax : REP <other string instruction>

REP is a prefix which is written before one of the string instruction of 8086. These instruction repeat until specified condition exists

instruction code	condition for Exit
REP	CX = 0
REPE / REPZ	CX = 0 or ZF = 0
REPNE / REPNZ	CX = 0 or ZF = 0

Examples

```
REPZ CMPSB :    ; compare string bytes until
                ; CX = 0 or until string
                ; byte not equal
```

2) MOVS / MOVSB / MOVSW instruction:

- This instruction copies a byte or word from a location in the data segment to a location in the extra segment.
- the offset of the source byte or word in the data segment must be in SI register.
- the offset of the destination in extra segment must be contained in DI register.

- For multiple byte or multiple word moves the number of elements to be moved input in the C register so that it can function as a counter
- After the byte word SI and DI are automatically adjusted to point to the next source and the next destination
- If the direction flag is 0, then SI and DI will be incremented by after a word move
- MOVS affects no flags
- The way to tell the assembler wether to code the instruction for a byte or word move is to add a 'B' or a 'W' to the ~~to~~ movs mnemonic
- MOV SB for example, says move a string as byte
- MOVSW says move a strings as word

Example : REP MOV SB.

3) CMPS I CMPSB I CMPSW instruction
- A string instruction in 8086 is a series of the same type of data items in sequential memory locations
- The CMPS instruction can be used to compare a byte in one string with byte in another String or to compare a word in ~~est~~ one string with a word in another string
- SI used to hold the offset of a byte or word in the source string and DI is used to hold offset of a byte or a word in the another string

- The comparision done by subtraction the byte or word pointer by DI from byte or word pointer to by SI
- The AF, CF, OF, PF, SF and ZF flags are affected by the comparsion, but neither operand is affected
- After the comparision SI and DI will be automatically incremented or decremented
- According to direction flag to point to next element in the two strings (if DF=0), SI and DI (x function as a counter which is decremented after each comparison
- this will go on cx=0

Example : REPE CMPS

Syntax : <REP instruction> CMPSB.

9) SCAS / SCASB / SCASW transcation:
- SCAS compares a string byte with a byte in AL or a string word with word in AX
- the instruction affects the operand in AL (Ax) or the operand in the string instruction in 8086
- the string to be scanned must be in the extra segment and DI will be automatically incremented and decremented according to direction flag. to point to the next element in the two strings (IF DF=0, SI and DI ↑) Cx functions as a counter which is decremented after each comparision

This will go on until CX = 0
- SCAS affects the AF, CF, OF, PF, SF, and ZF flags.

Syntax: <REP instruction> SCASB

Example: REPNE SCASB

5) LODS / LODSB / LODSW instructions.
- This instruction copies a byte from AL or a word from AX memory location in the extra segment.
- DI is used to hold the offset of the memory location in the extra segment.
After the copy DI is automatically incremented or decremented to point the next string element in memory.
- If the direction flag DF is clear then DI will be automatically be incremented by one for byte string or incremented by two for a word string instruction in 8086
- If the direction flag is set, DI will be automatically decremented by one for byte string or decremented by two for a word string.
- STOS does not affect and flags
- STOSB copies bytes and STOSW copies a word

Syntax: STOS <String>

Example: STOS ES:D-String.

Program for reversing a given string:

Code:

```
Assume DS: data CS: code

Data segment
m1 db 0AH ,0DH , 'Enter string : $'
m2 db 0AH ,0DH , 'Reverse of string : $'
buff db 80 H
     db 00 H
     db 80 H dup ( 00 H)

data ends

Print MACRO MSG
MOV AH , 09 H
LEA DX , MSG
INT 21 H

ENDM

Code segment
start : MOV AX , Data
        MOV DS, AX
        Print m1
        MOV AH ,0AH
        LEA DX, buff
        int 21 H
        Print m2
```

09

```
LEA Bx, buf1
INC Bx
MOV CH, 00H
MOV CL, buff+1
MOV DI, CX

Basic: MOV DL [Bx+DI]
       MOV AH, 02H
       INT 21H
       DEC DI
       JNZ BACK
MOV AH, 4CH
INT 21H
Code ENDS
End starts.
```

Q2 Explain pin diagram of ADC 0808/0809 and Design its interfacing to 8086 with suitable example.

Ans

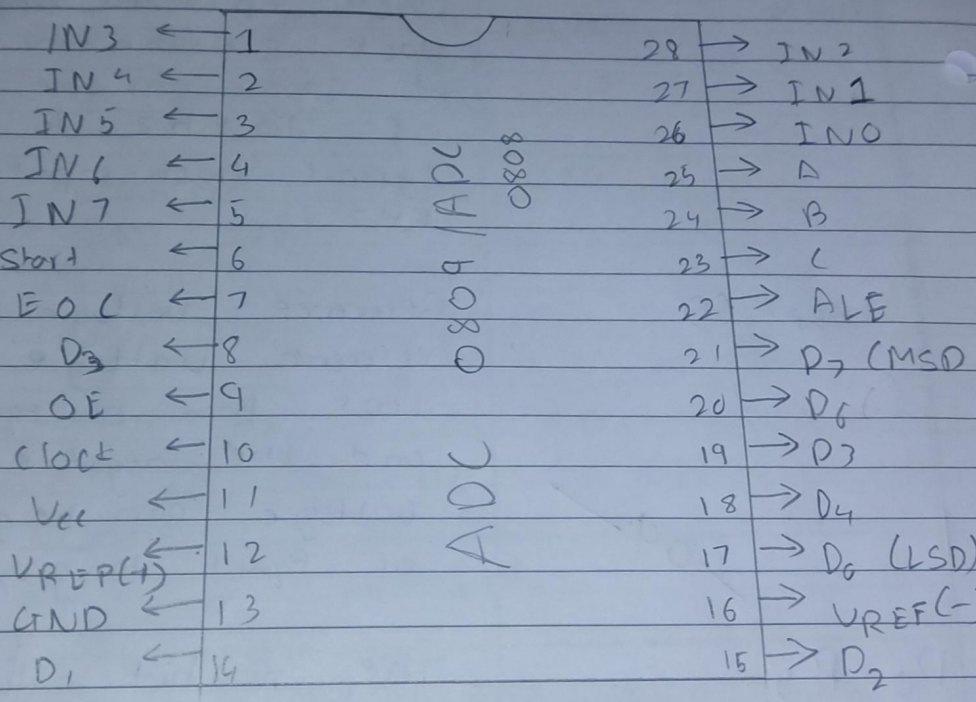
Signals	Description
IN0 - IN7	Eight single ended analog input to ~~select~~ ADC
A, B, C	3 bit binary input to select one of the eight analog signals for conversion at any one time.
ALE	Address Latch enable used to latch the 3 bit address input to an internal Register

Start	Start of conversion at pulse input to start APC process this signal should be asserted high and then low signal should remain high atleast for 100 ns
Clock	Clock input and frequency of clock can be in the range of clock can be in the range of 10 KHZ to 1280 KHZ. Typical clock input is 640 KHZ
V_{REF} (+)	Reference voltage input positive reference voltage can be less than or equal to V_{CC} and the negative reference voltage can be greater than or equal to ground
$D_0 - D_7$	The 8-bit digital output the reference voltage will decide the mapping of analog input to digital data
EOC	End of conversion
OE	output buffer Enable the signal is used to read the digital data from output buffer after a valid EOC

V_{cc} Power supply +5 V

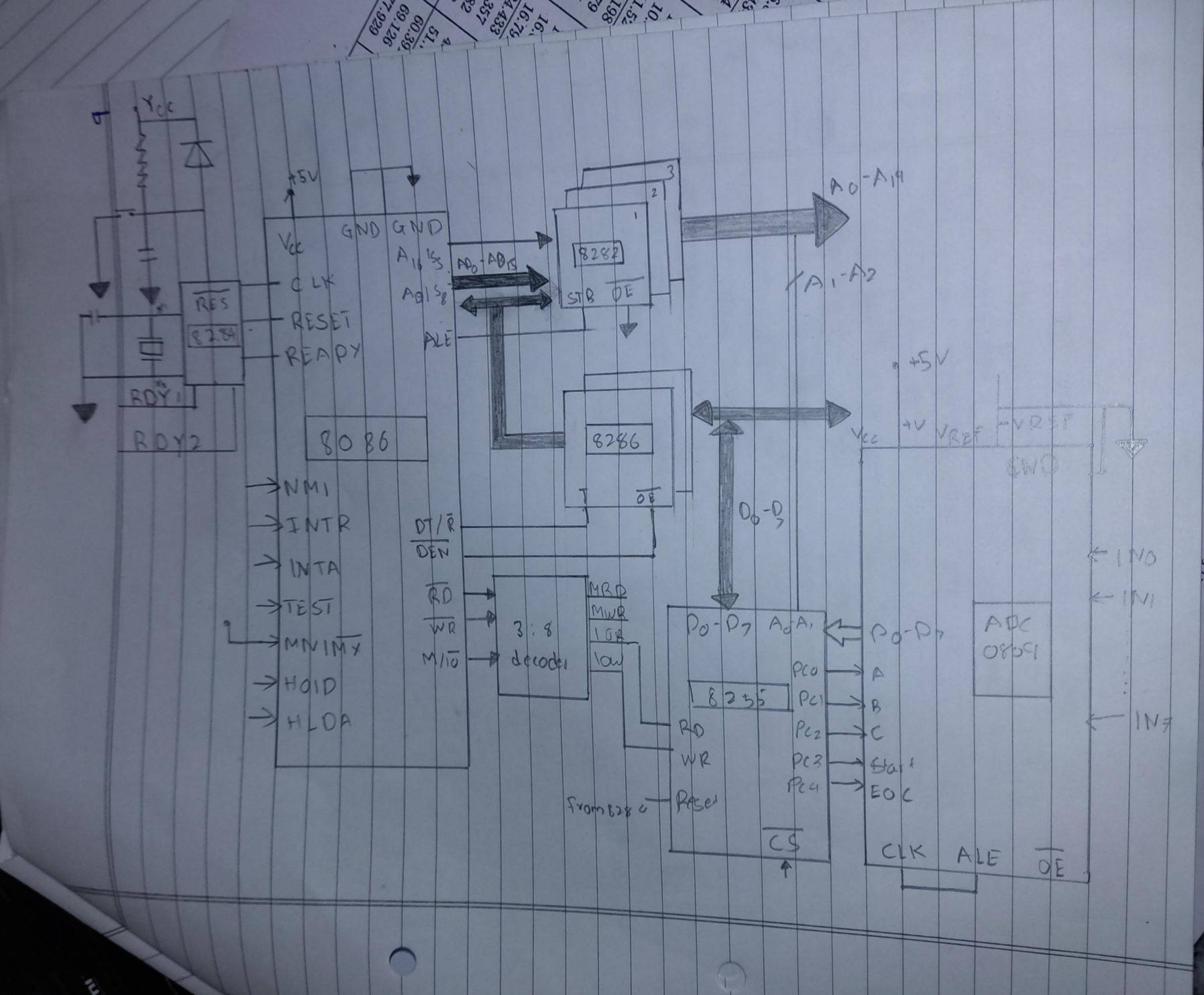
GND Power supply ground, 0V

Pin Diagram



LSD - Least significant Digit

MSD - Most significant Digit


Vcc
+5V
GND GND
Vcc
CLK
RESET
READY
RES
8284
RDY1
RDY2
8086
NMI
INTR
INTA
TEST
MN/MX
HOLD
HLDA
A16/S3
AD8-AD15
AD1/S8
ALE
8282
STB OE
A0-A19
A1-A2
8286
OE
+5V
Vcc
+V
VREF
-VREF
GND
D0-D7
DT/R
DEN
RD
WR
M/IO
3:8 decoder
MRD
MWR
IOR
IOW
8255
PA0-PA7
PB0-PB7
PC0
PC1
PC2
PC3
PC4
A
B
C
Start
EOC
RD
WR
Reset
from 8284
CS
ADC 0809
IN0
IN1
IN7
CLK ALE OE