

## Assignment 01

Q1) Classify & discuss 8086 instruction set.

→ An instruction set is a binary pattern designed inside a microprocessor to perform a specific function.

Classifications of instruction set of 8086

- 1) Data transfer instructions
- 2) Arithmetic instructions
- 3) Bit manipulations instructions
- 4) String instructions
- 5) Program execution & transfer instruction
- 6) Processor control instructions
- 7) Iteration control instructions
- 8) Interrupt control instruction
- 9) External hardware synchronization instruction
- 10) Shift & rotate instructions

### 1) Data transfer instruction:

These instruction are used to transfer the data from the source operand to the destination operand.

Instruction to transfer a word:

- MOV - USED to copy the byte or word from the provide source to the provided destination
- PUSH - Used to put a word at top of stack
- POP - Used to get a word from the top of stack to provided location.
- PUSHA - Used to put all registers into the stack
- POPA - Used to get word from stack to all registers

- XCHG - Used to exchange the data from two locations
- XLAT - Used to translate a byte in AL using a table in the memory.

Instruction for input & output port transfer

- IN - Used to read a byte or word from the provided port to the accumulator
- OUT - Used to send out a byte or word from the accumulator to provided port.

Instructions to transfer the address

- LEA - Used to load the address of operand into the provided register.
- LDS - Used to load DS register & other provided register from the memory
- LES - Used to load ES register & other provided register from the memory.

Instruction to transfer flag registers :

- LHMF - Used to flag load AL with the low byte of the flag register
- SHAF - Used to store AL register to low byte of flag register.
- PUSHF - Used to copy the flag register at the top of stack
- POPF - Used to copy a word at the top of stack to the flag register

## 2) Arithmetic Instruction:

These instruction are used to perform arithmetic operation like addition,

subtraction, multiplication, division etc.

Instruction to perform addition:

ADD, ADC, INC, AAA, DAA

Instruction to perform subtraction:

SUB, SBB, DEC, NEG, CMP, AAS, DAS

Instruction to perform multiplications:

MUL, IMUL, AAM

Instruction to perform division:

DIV, IDIV, AAD, CBW, CWD

3) Bit Manipulation Instructions:-

- These instruction are used to perform operation where data bits are involved i.e operation like logical, shift, etc. following is list of instruction under the group:-

NOT, AND, OR, XOR, TEST

Instruction to perform shift operation:-

SHL / SAL, SHR, SAR

Instruction to perform rotate operation:-

ROL, RDR, RCR, RCL

4) String Instruction:

String is a group of bytes / words & their memory is always allocated in sequential order.

REP - Used to repeat the given instruction till CX ≠ 0

REPE / REPZ, REPNE / REPNZ, MOVS  
MOUSB / MOUSW, COMS / COMPSB / COMPSW  
/ INS / INSB / NSW; OUTS / OUTSB /  
OWTSW, SCAS, / SCASB / SCASW, LODS  
/ LODSB / LODSW

5) Program Execution Transfer Instructions:

These instructions are used to transfer / branch the instruction are used to transfer If branch the instruction during an execution.

If includes the following instruction:

CALL, RET, JMP, JA / JNB E,  
JAE / jNB, JBE / JNA, JC, JE / JZ, JA / JNL  
JGE / JNL, JL / JNGE, JE / JNG, JNC,  
JNE / JNZ, JNO, JNP / JPO, JNS, JO  
JP / JPE, JS.

6) Process Control Instruction:

These instructions are used to control the processor action by setting / resetting the flag values

STC, CIC, CMC, STD, CLD, STI, CLI

7) Iteration Control Instructions

These instructions are used to execute the given instructions for number of times.

LOOP, LOOPE / LOOPZ, LOOPNE, LOOPNZ  
JCXZ

### 8) Iteration Control Instructions:

These instructions are used to execute the given instruction for number of times.

LOOP - Used to loop a group of instruction until condition satisfies CX=0  
LOOPE / LOOPZ, LOOPNE / LOOPNZ,  
JCXZ

### 9) Interrupt Instructions:

These to call the interrupt during program execution

INT, INTO, IRET

### Q2) State & explain assembler directives

- Assembler directives are the instructions used by the assembler at the time of assembling a source program more specifically are used assembler directives are the commands or instructions that control the operation of the assembler
- Assembler directives are the instructions provided to the assembler, not the processor as the processor has nothing to do with these instructions. These instructions are also known as pseudo-

opcode:

These assembler directives are specifically used as follow:-

Assume :- shows the segment name to assembler

DD :- Define Double word.

DQ :- Define Quad word.

DT :- Define Ten bytes

DUP : Duplication

DWORD : Double word

PROC : Procedure

FAR : This directive is type specifies that is used by assembler directive intersegment call

NEAR : This is used for intrasegment call

ENDP : End of procedure

SEGMENT : Beginning of memory segment

ENDS : End of segment

EVEN : It is used to inform the assembler to align the data beginning from an even address

PTR : Pointer

PUBLIC : This directive used to provide a declaration to variable that are common different program modules

STACK : This directive shows the presence of a stack segment

SHORT : This is used to reference to jump instruction in order to assign a displacement

THIS : It is used along with EQU

Daryl Funandes 2020012004 (72)

(7)

directive for setting label to either byte  
word or double-word.