CS332 : MICROPROCESSOR LAB

CYCLE 1

Assembly Language Programming Experiments using 8086 Trainer kit

1) Arithmetic operations implementation using
   a) 8 bit operands b) 16 bit operands
   (addition ,subtraction ,multiplication and division operations)
2) Implement code for finding the factorial of a given number
3) Implement code for checking whether the given number is even or odd
4) Implement code for checking whether the given number is prime or not
5) Implement code for linear search algorithm
6) Implement code for bubble sort algorithm

Assignment Works

1) Implement code for finding the sum of a given list of elements
2) Separate even and odd numbers from a given list and store them separately in different locations
3) Implement code for finding the largest among the given list of numbers

CYCLE 2

Assembly language programming experiments using MASM

1) Implement the code for addition of two 32 bit numbers
2) Implement the code for checking whether the given string is palindrome or not
3) Implement code for searching algorithm
4) Implement code for finding HCF of two 16 bit numbers
5) Implement MASM program to copy the content of one file to another

Assignment Works

1) Implement code for bubble sorting
2) Implement code to check whether the given string is palindrom or not

Record format

Right hand side page

1)Date and program number on the top corner
2)Aim:
3)Algorithm
4)

ADDRESS	OPCODE	MNEMONICS	COMMENT

5)Result

Left hand side page

OUTPUT

Input : address location data

Output: address location result

VIVA topics :

- 8086 architecture –units ,registers ,flags ,memory segments ,physical address calculations
- Addressing modes of 8086
- Instruction sets of 8086

Instructions :

- prepare cycle 1 –first programs for next lab session
- prepare for VIVA
- Write aim ,algorithm , MNEMONICS and comments on to record

Sample Programs in CYCLE 1

Assembly Language Programming Experiments using 8086 Trainer kit

Sample Program :

Aim: Addition of two 16 bit numbers

Algorithm

Step 1: read two 16 bit numbers

Step 2: clear the location to store carry

Step3: perform addition operation

Step 4 : if no carry occurs go to step 6

Step 5: store carry value to a location

Step 6: store sum value to a location

Step 7: stop

ADDRESS	OPCODE	MNEMONICS	COMMENT
		MOV AX, [2000]	Move first operand from location 2000H to AX
		MOV BX, [2002]	Move second operand from location 2002H to BX
		MOV [2004],0000	Clear location 2004H
		ADD AX, BX	Perform addition of AX with BX and sum available IN AX
		JNC L1	Check if carry occurs or not
		MOV [2004],0001	If carry occurs store carry value in location 2004H
		L1:	
		MOV [2006], AX	Store sum value at 2006H
		HLT	Halt the processor