BECE204P-Microprocessors & Microcontrollers Lab

LAB-3

SUM & AVERAGE OF "N" 8-BIT NUMBERS, FACTORIAL, CELSIUS TO FAHRENHEIT

INSTRUCTION REQUIRED

OPCODE	OPERAND	EXPLANATION	EXAMPLE
ADC	D, S	D = D + S + CF	ADD AX, BX
INC	D	D=D+1	INC AX
DEC	D	D=D-1	DEC [0500H]
JNZ	Address	Jump if not zero	JNZ 1020
CMP	D, S	Compare D with S (S-D). Doesn't store result but it affect ONLY flag register	CMP AX, BX

Here CF stands for Carry Flag, D stands for destination and S stands for source.
D and S can either be register, data or memory address.

LAB TASK-1

SUM OF "N" NUMBERS

- Write 8086 Assembly language program to perform sum of "N" 8-bit numbers stored in array of memory locations

Algorithm:

1. Initialize SI with memory location 0500H as a starting a address to hold the value of the count "N" and store array of numbers to be added in the consecutive memory locations
2. Initialize DI with 0600H as a starting a address to hold the result of sum of "N" numbers
3. Initialize AX register 0000H to eliminate error in the result by adding previous value of AX
4. Assign CL register with the value of the count "N" by loading it from 0500h memory location using SI
5. Increment the content of SI to point the first element of the array
6. Perform addition on the value of AL and first element of the array (Result will be stored again in AL)
7. Perform addition of AH+00+C bit to hold result when Lower byte addition resulted any carry
8. Increment the content of SI to point the next element of the array
9. Decrement CL value by 1 and check whether count is reached 0
10. If not zero, repeat steps 6 to 9
11. Otherwise store the final result of sum of all numbers in the array held by AX into the memory location pointed by DI register
12. Stop the execution

LAB TASK-1

SUM OF "N" NUMBERS

ADDRESS	MEMONICS	COMMENTS
1000	MOV SI, 0500H	Load 0500H into Source Index Register
	MOV DI, 0600H	Load 0600H into Destination Index Register
	MOV AX, 0000	Clear AX register
	MOV CL, [SI]	Load the block size (Value of N)
	INC SI	Increment SI to point next memory location
BACK:	ADD AL, [SI]	Add AL and data pointed by SI register
	ADC AH, 00	Add AH and 00H along with Carry
	INC SI	Increment SI to point next memory location
	DEC CL	Decrement CL value by 1
	JNZ BACK	If Z=0, jump to BACK label
	MOV [DI], AX	Store the result of the division into memory location pointed by DI
	HLT	Stop the execution

Input:

ADDRESS	VALUE
0500H	05H
0501H	04H
0502H	02H
0503H	09H
0504H	01H
0505H	05H

Output:

ADDRESS	VALUE
0600H	15H
0601H	00H

LAB TASK-2

AVERAGE OF "N" NUMBERS

- Write 8086 Assembly language program to perform sum of "N" 8-bit numbers stored in array of memory locations

Algorithm:

1. Initialize SI with memory location 0500H as a starting a address to hold the value of the count "N" and store array of numbers to be added in the consecutive memory locations
2. Initialize DI with 0600H as a starting a address to hold the result of sum of "N" numbers
3. Initialize AX register 0000H to eliminate error in the result by adding previous value of AX
4. Assign CL register with the value of the count "N" by loading it from 0500h memory location using SI
5. Also copy the count "N" value into BL register to act as devisor in the division operation
6. Increment the content of SI to point the first element of the array
7. Perform addition on the value of AL and first element of the array (Result will be stored again in AL)
8. Perform addition of AH+00+C bit to hold result when Lower byte addition resulted any carry
9. Increment the content of SI to point the next element of the array
10. Decrement CL value by 1 and check whether count is reached 0
11. If not zero, repeat steps 6 to 9
12. Otherwise perform division operation on the final result of sum of all numbers in the array held by AX by BL register count
13. Store the Quotient in reminder result in the memory location pointed by DI register
14. Stop the execution

LAB TASK-2

AVERAGE OF "N" NUMBERS

ADDRESS	MEMONICS	COMMENTS
1000	MOV SI, 0500H	Load 0500H into Source Index Register
	MOV DI, 0600H	Load 0600H into Destination Index Register
	MOV AX, 0000H	Clear AX register
	MOV CL, [SI]	Load the block size (Value of N)
	MOV BL, CL	Also store N into BL
	INC SI	Increment SI to point next memory location
BACK	ADD AL, [SI]	Add AL and data pointed by SI register
	ADC AH, 00H	Add AH and 00H along with Carry
	INC SI	Increment SI to point next memory location
	DEC CL	Decrement CL value by 1
	JNZ BACK	If Z=0, jump to BACK label
	DIV BL	Store the result of the division into memory location pointed by DI
	MOV [DI], AX	Otherwise divide it with BL
	HLT	Stop the execution

Input:

ADDRESS	VALUE
0500H	05H
0501H	04H
0502H	02H
0503H	09H
0504H	01H
0505H	05H

Output:

ADDRESS	VALUE
0600H	04H (Q)
0601H	01H (R)

LAB TASK-3

FACTORIAL

- Write 8086 Assembly language program to calculate factorial of given number

Algorithm:

1. Input the Number whose factorial is to be find and Store that Number in CX Register
2. Insert 0001 in AX(Condition for MUL Instruction) and copy AX to BX register
3. Increment BX by 1
4. Multiply AX and BX (Results stored in AX)
5. Repeat step 3 and 4 until BX repeated for CX number using CMP and JNZ instructions
6. Use CMP and JNZ instruction to CX with AX until CX become Zero(0) using LOOP Instruction
7. Copy the final content of AX to memory location 4000H
8. Stop Execution

LAB TASK-3

FACTORIAL

ADDRESS	MEMONICS	COMMENTS
1000	MOV CX, 04H	Load Number whose factorial is to be find in CX
	MOV AX, 0001H	Load AX with 0001H
	MOV BX,AX	COPY AX into BX
BACK:	INC BX	Increment BX value by 1
	MUL BX	Multiply AX*BX = DX:AX
	CMP BX,CX	Compare BX with CX i.e BX-CX
	JNZ BACK	If Z is not Zero, jump to BACK label
	MOV [4000H], AX	Store AX register content to memory location 4000H
	HLT	Stop the execution

Input:
CX =04H

Output:

ADDRESS	VALUE
4000H	18H
4001H	00H

LAB TASK-4

CONVERT CELSIUS TO FAHRENHEIT (Exercise)

- Write 8086 Assembly language program to Convert the given temperature in Celsius (C) scale to Fahrenheit (F) scale. Assume the value of temperature in Celsius as 25°C (19H).

$$F = (9C/5)+32$$

Algorithm:

1. Input the Celsius (C) value in CL Register
2. Store AL with constant value 09H
3. Perform 09 * C using MUL instruction
4. Store DL with constant value 05H
5. Perform (09 * C)/5 using DIV instruction
6. Add 20H (32) with the Quotient result from previous operation
7. Copy the final content of AL to memory location 2000H
8. Stop Execution

LAB TASK-4

CONVERT CELSIUS TO FAHRENHEIT (Exercise)

ADDRESS	MEMONICS	COMMENTS
1000	MOV CL, 19H	Load 25°C (19H) Celsius value into CL register
	MOV AL, 09H	Load AL with 09H
	MUL CL	Multiply AL with 25°C Celsius value and store the result in AX
	MOV DL, 05H	Move 05H to DL register
	DIV DL	Divide AX by DL. Quotient in AL and Reminder in AH.
	ADD AL, 20H	Add 32 (20H) along with AL (Quotient)
	MOV [2000H], AL	Store the final result available in AL into memory location 2000H
	HLT	Stop the execution

Input:
CL =19H

Output:

ADDRESS	VALUE
2000H	4DH