KERNEL MASTERS Lab Assignment

|  |  |
| --- | --- |
| NAME: VIGNESH.B | KMID: KM40BESD01 |

**ALP LAB Assignments:**

1. **To find Odd No or Even No:**

Write an ALP to find given number is odd or even and load number in 4000H memory location the result stored in 6000H. If even, store 22H at memory location 6000 otherwise store 11H at memory location 6000.

**STEP 1: Pseudo Code**

* + Load data from the memory location 4000h to microprocessor
  + Check the loaded data Even or Odd.
  + LSB bit indicates the data odd or even.
  + So perform AND operation with the data.
  + If the result is 00 then its EVEN else its ODD.
  + If its even then store 22h data to the 6000h memory location.
  + And if the data is odd then store 11H data to the 6000h memory location.

**STEP 2: Flow Chart**

Store 11h to 6000h memory location

AND 01H with

A REGISTOR

A🡨[4000H]

ODD EVEN

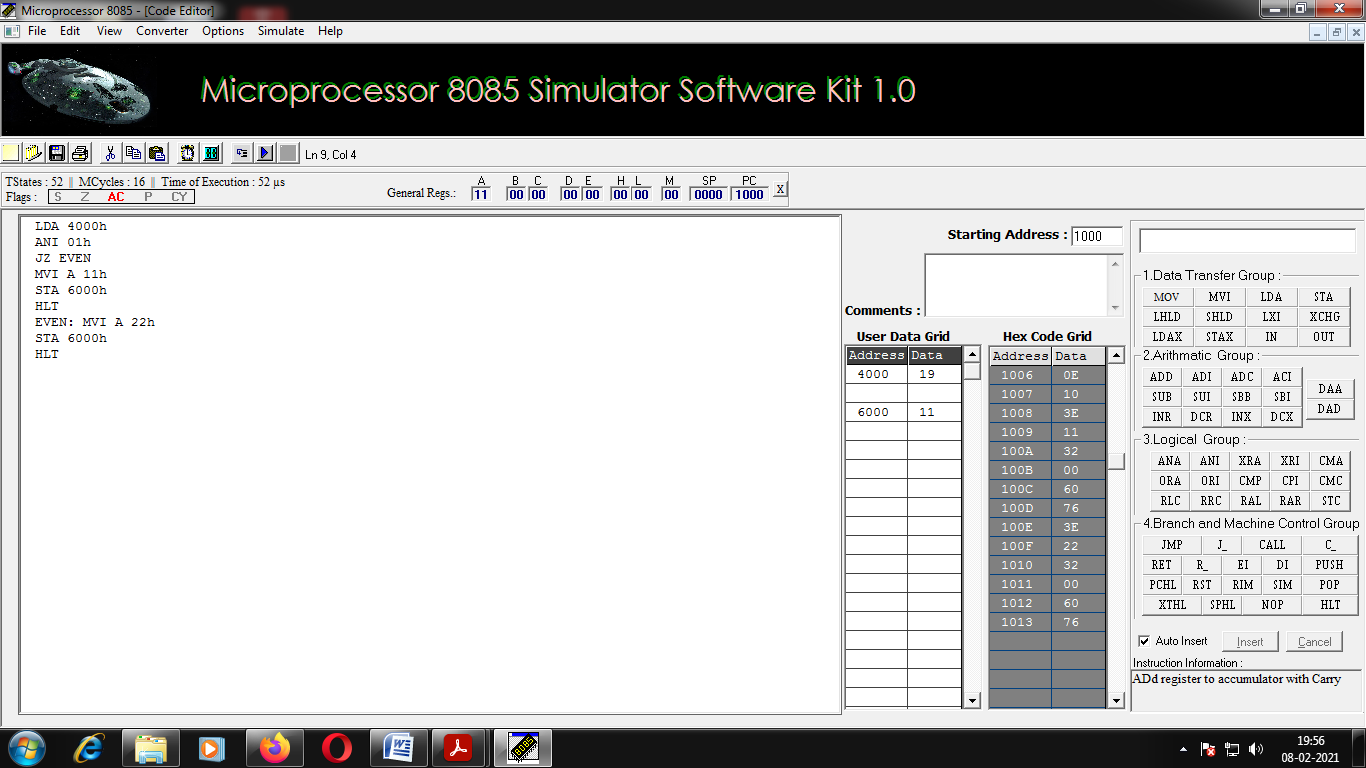
Store 22h to 6000h memory loaction

**STEP 3: ALP PROGRAM**

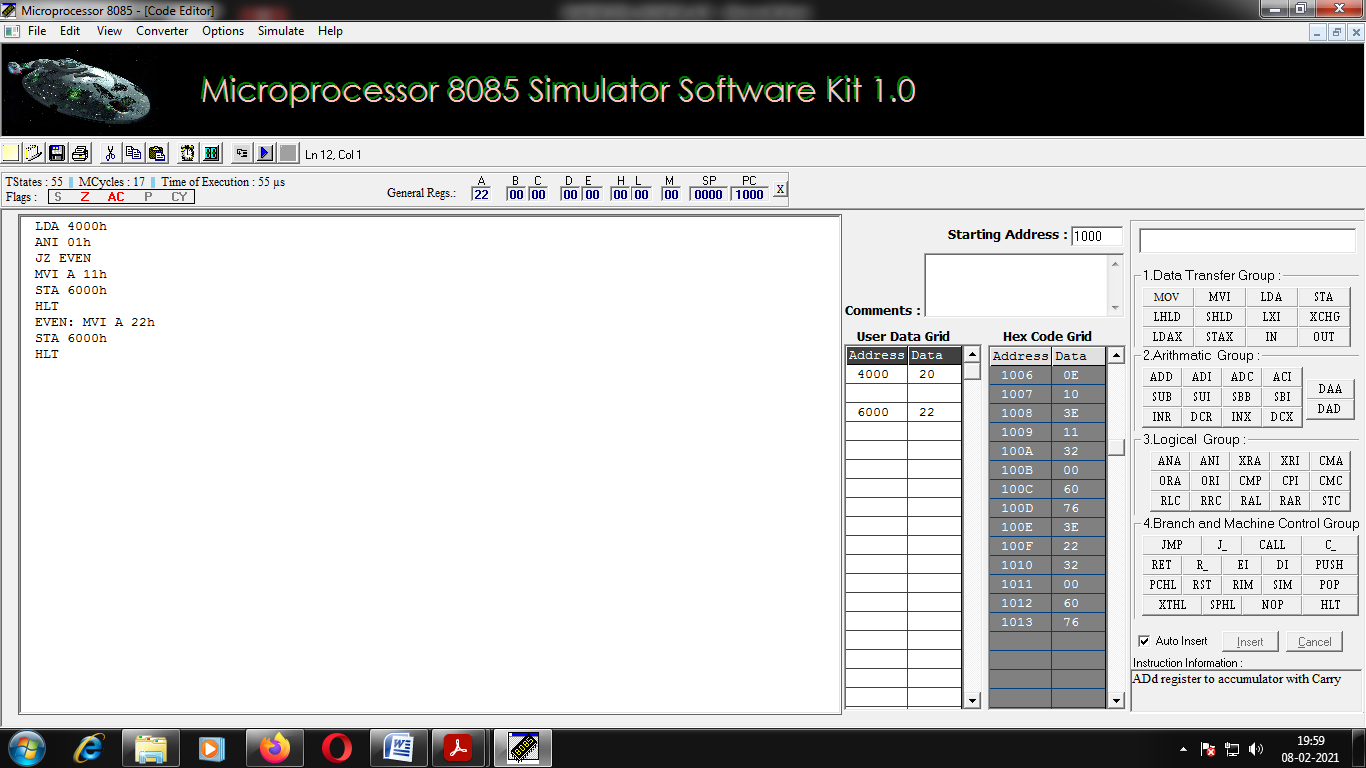
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Memory address | Hexa code | Label | Opcode | Operand | Comments |
| 1000h | XX |  | LDA | 4000H | A🡨[4000H] |
| 1001h | 00H |  |  |  |  |
| 1002h | 40H |  |  |  |  |
| 1003h | XX |  | ANI | 01H | A🡨A (AND) 01H |
| 1004h | 01H |  | JZ | EVEN | Jump to EVEN label if zero flag set Zf=1 |
| 1005h | XX |  | MVI | A,11H | A🡨11H |
| 1007H | 22H |  |  |  |  |
| 1008H | XX |  | STA | 6000H | [6000H]🡨A |
| 1009H | 00H |  |  |  |  |
| 100AH | 60H |  |  |  |  |
| 100BH | XX |  | HLT |  | EXIT |
| 100CH | XX | EVEN: | MVI | A,22H | A🡨22H |
| 100DH | 11H |  |  |  |  |
| 100EH | XX |  | STA | 6000H | [6000H]🡨A |
| 100FH | 00H |  |  |  |  |
| 1010H | 60H |  |  |  |  |
| 1011H | XX |  | HLT |  | EXIT |

**STEP 4: EXECUTION IN SIMULATOR**

**IF ODD NUMBER ENTERED:**

****

**IF EVEN NUMBER ENTERED:**

****

**2. Data transfer memory to memory:**

Write an ALP to 16 bytes of data stored in memory locations at 2000H to 200FH. Transfer the entire block of data to new memory locations starting at 4000H.

STEP 1: Pseudo code

STEP 1: Pseudo code

* initialize the counter
* Move the source and destination memory address as data to two register pair.
* Move the content of the one data pair to another data pair
* increment both the register pair address
* decrement the counter
* repeat the step 3 until the counter becomes zero

**STEP 2: FLOW CHART**

Initialize the counter value.

Initialize the source address to HL register pair

Initialize the destination address to DE register pair

LOAD DATA TO ACCUMULATOR FROM SOURCE ADDRESS

STORE ACCUMULATOR DATA TO DESTINATION ADDRESS

INCREMENT THE SOURCE AND DESTINATION ADDRESS

DECREMENT THE COUNTER

**Zero flag**

**ZF=0**

**ZF=1**

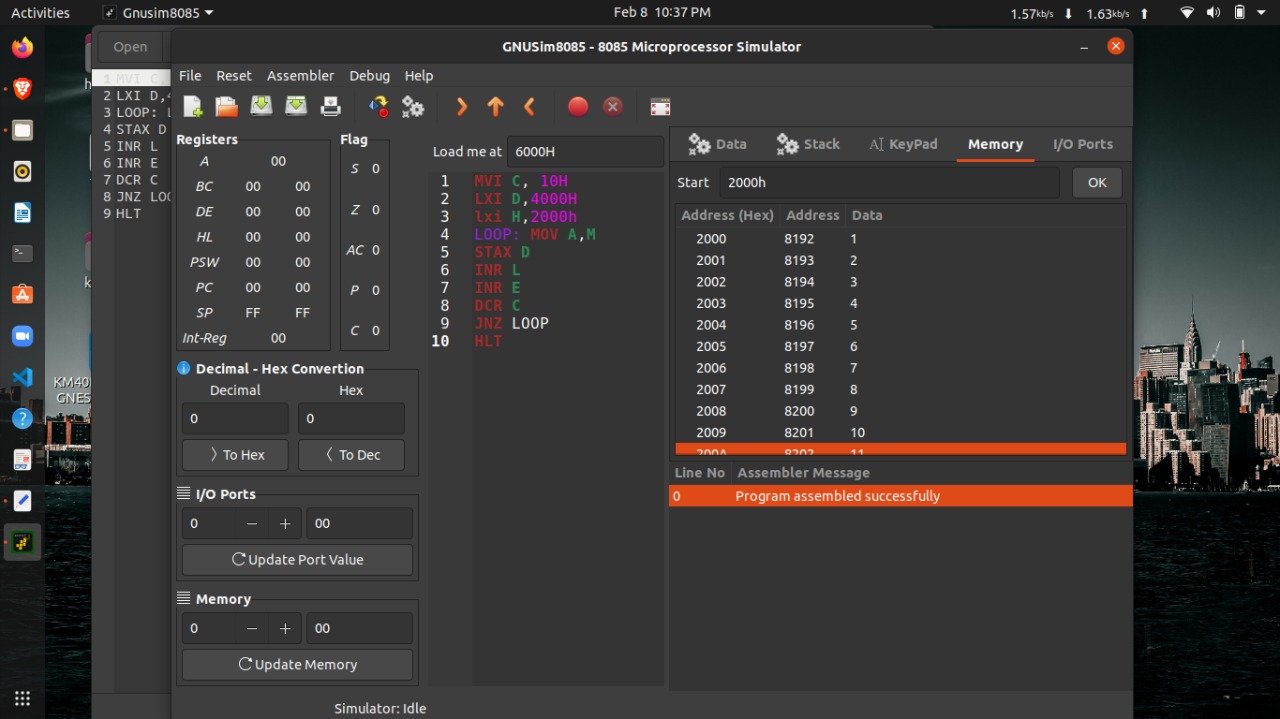
**STEP 3: ALP**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Memory location** | **Hexa code** | **label** | **Opcode** | **operand** | **comment** |
| 1001H | XX |  | MVI | C,10h | C<-- 16 |
| 1002H | 10H |  |  |  |  |
| 1003H | XX |  | LXI | D,4000h | D<-- 4000H |
| 1004H | 00H |  |  |  |  |
| 1005H | 40H |  |  |  |  |
| 1006H | XX |  | LXI | H,2000H | H<-- 2000H |
| 1007H | 00H |  |  |  |  |
| 1008H | 20H |  |  |  |  |
| 1009H | XX | LOOP: | MOV | H,M | A<--[HL] |
| 100AH | XX |  | STAX | D | [DE]<--A |
| 100BH | XX |  | INR | L | INCREMENT SOURCE ADDRESS |
| 100CH | XX |  | INR | E | INCREMENT THE DESTINATION ADDRESS |
| 100DH | XX |  | DCR | C | DECREMENT COUNTER |
| 100EH | XX |  | JNZ | LOOP | GOTO LOOP LABEL IF C=00H |
| 100FH | XX |  | HLT |  | EXIT |

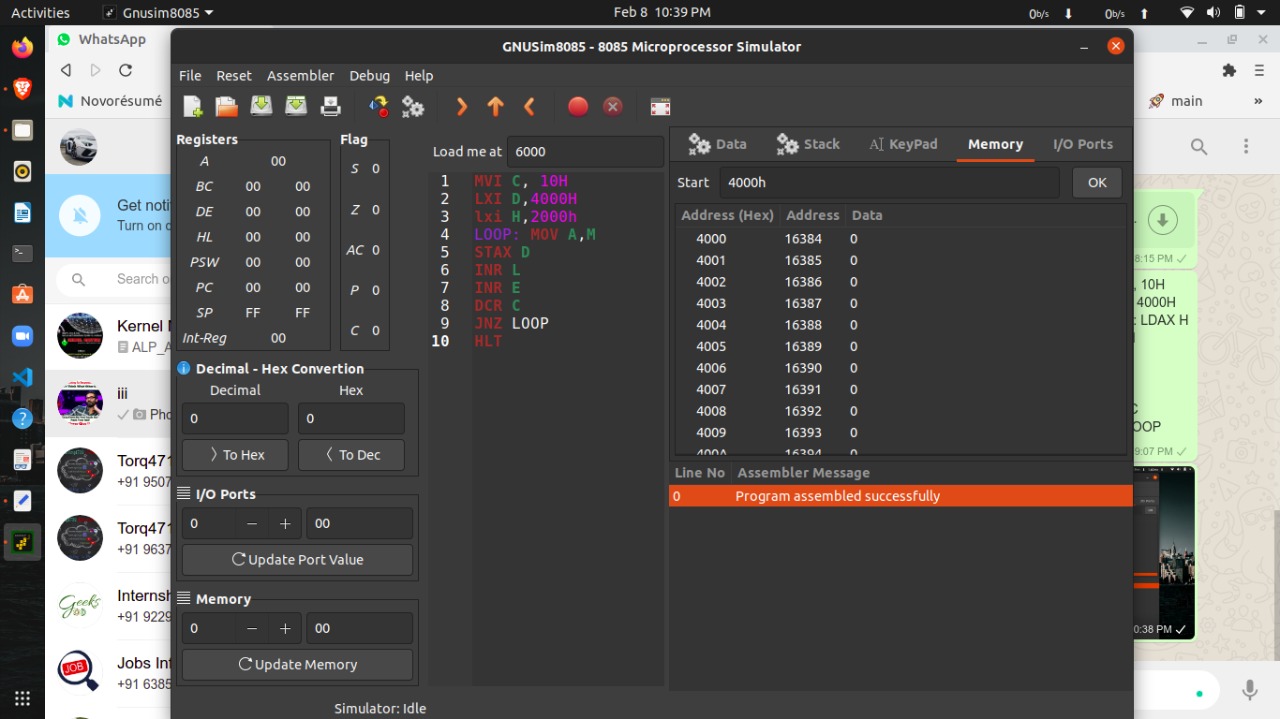
**STEP 4: EXECUTION IN SIMULATION**

**BEFORE PROGRAM EXECUTION :**

**STARTING MEMORY ADDRESS: (2000H)**

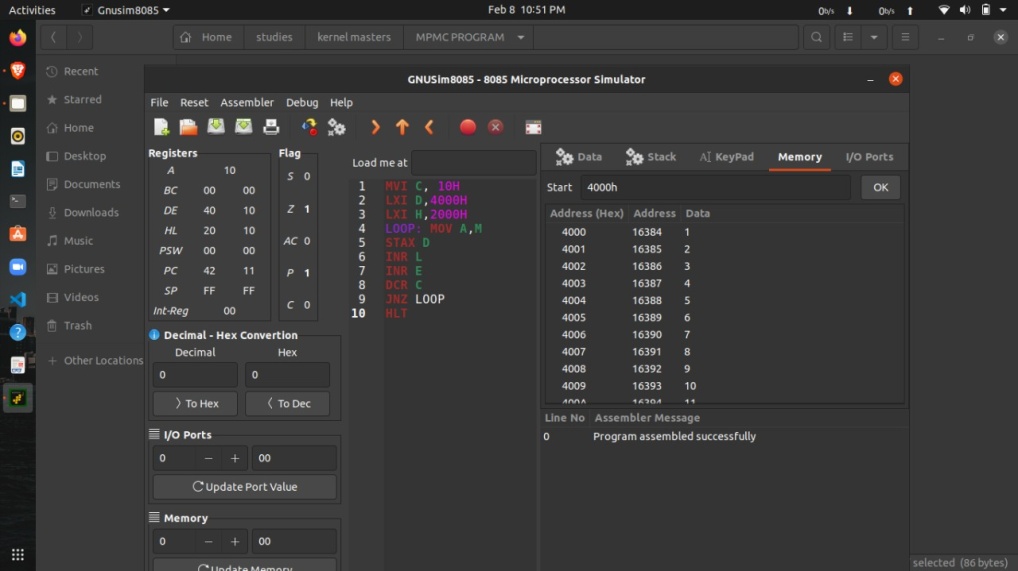
****

**DESTINATION ADDRESS: (4000H)**

****

**AFTER EXECUTING PROGRAM:**

**DESTINATION ADDRESS: (4000H)**

****

**3. To Perform Multiplication without using MUL instruction:**

Write an ALP to perform multiplication of two numbers without using MUL instruction first & second number stored in 4000H & 4001H memory locations respectively and the result stored in 6000H?

**STEP 1: Pseudo code**

1. Load the content of the 4001h to accumulator.
2. Move the accumulator value to counter.
3. Load the content of the 4000h to accumulator.
4. Add the accumulator value with SAME DATA store the value in the accumulator.
5. Decrement counter value
6. Repeat step 4&5 until counter become zero.
7. Store the result in 6000h.

**STEP 2: FLOW CHART**

A🡸[4001H]

C🡸A

A🡨[4000H]

B🡨A

A🡨00H

ZERO FLAG

STORE RESULT VALUE TO 6000H

A🡨A+B

DECREMENT COUNTER

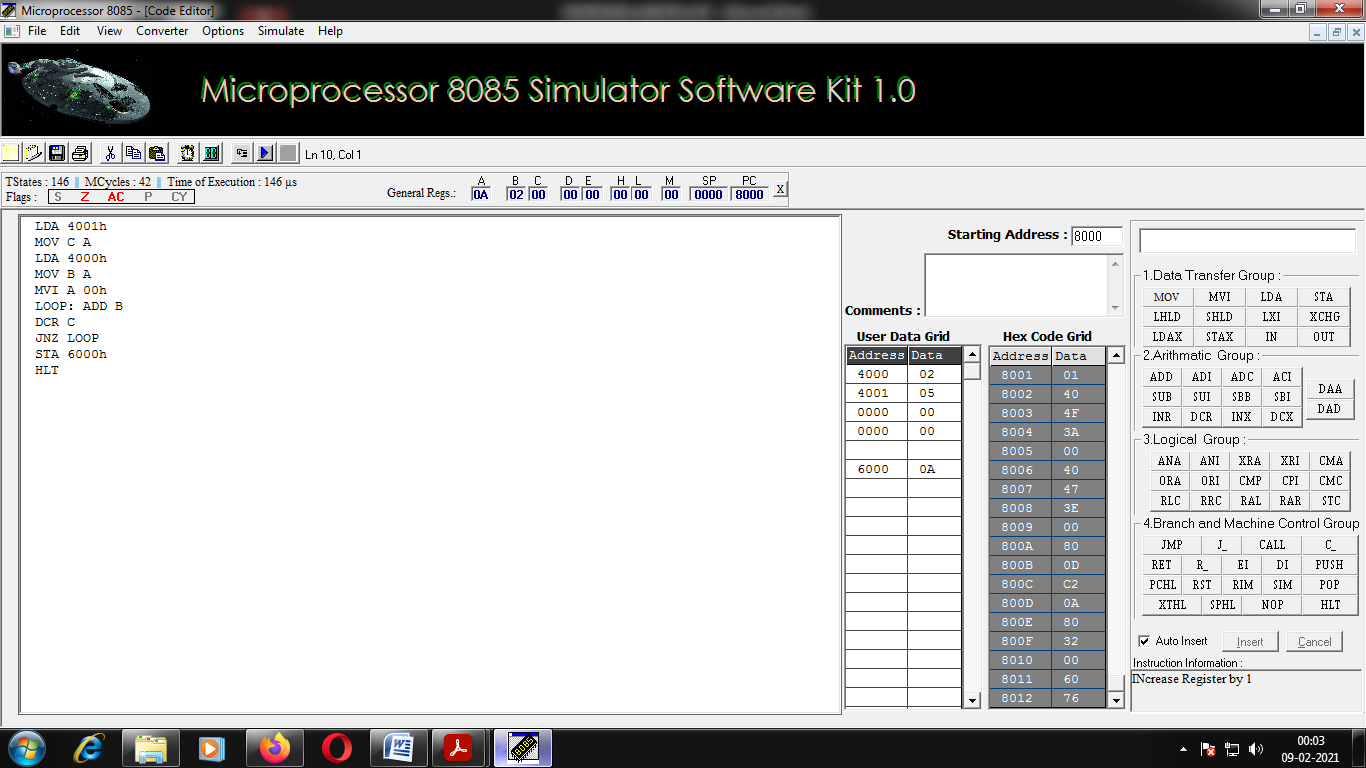
ZF=0

ZF=1

**STEP 3:ALP**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MEMORY ADDRESS | HEXA CODE | LABEL | OPCODE | OPERAND | COMMENT |
| 1000H | XX |  | LDA | 4001H | A🡨[4001] |
| 1001H | 01H |  |  |  |  |
| 1002H | 40H |  |  |  |  |
| 1003H | XX |  | MOV | C, A | C🡨A |
| 1004H | XX |  | LDA | 4000H | A🡨[4000H] |
| 1005H | 00H |  |  |  |  |
| 1006H | 40H |  |  |  |  |
| 1007H | XX |  | MOV | B, A | B🡨A |
| 1008H | XX |  | MVI | A,00H | A🡨00H |
| 1009H | 00H |  |  |  |  |
| 100AH | XX | LOOP: | ADD | B | A🡨B |
| 100BH | XX |  | DCR | C | C🡨C-1 |
| 100CH | XX |  | JNZ | LOOP | JUMP TO LOOP LABEL IF NO ZERO FLAG=0, ELSE EXIT |
| 100DH | XX |  | STA | 6000H | [6000H]🡨A |
| 100EH | 00H |  |  |  |  |
| 100FH | 60H |  |  |  |  |
| 1010H | XX |  | HLT |  | EXIT |

**STEP 4 : EXECUTION**



**4. To Perform Division without using DIV instruction:**

Write an ALP to perform division of two numbers without using DIV instruction first & second number stored in 4000H & 4001H memory locations respectively and the result stored in 6000H?

STEP 1:Pseudo code

1. Load the dividend to the accumulator 4001h
2. Store dividend in any register
3. Load divisor to accumulator 4000h
4. Initialize counter
5. Increment counter
6. Subtract dividend by divisor and store result in accumulator
7. Goto step 4 until the value become negative
8. Store the counter in accumulatror
9. Store the no of times loop run in 6000h

STEP2 :FLOW CHART

SIGN FLAG

INCREMENT COUNTER

A🡸A-B (NO OF TIMES LOOP RUNS)

A🡨[4001H] DIVIDEND

B🡨A

A🡨[4000H] DIVISOR

INITIALIZE COUNTER

SF=0

SF=1

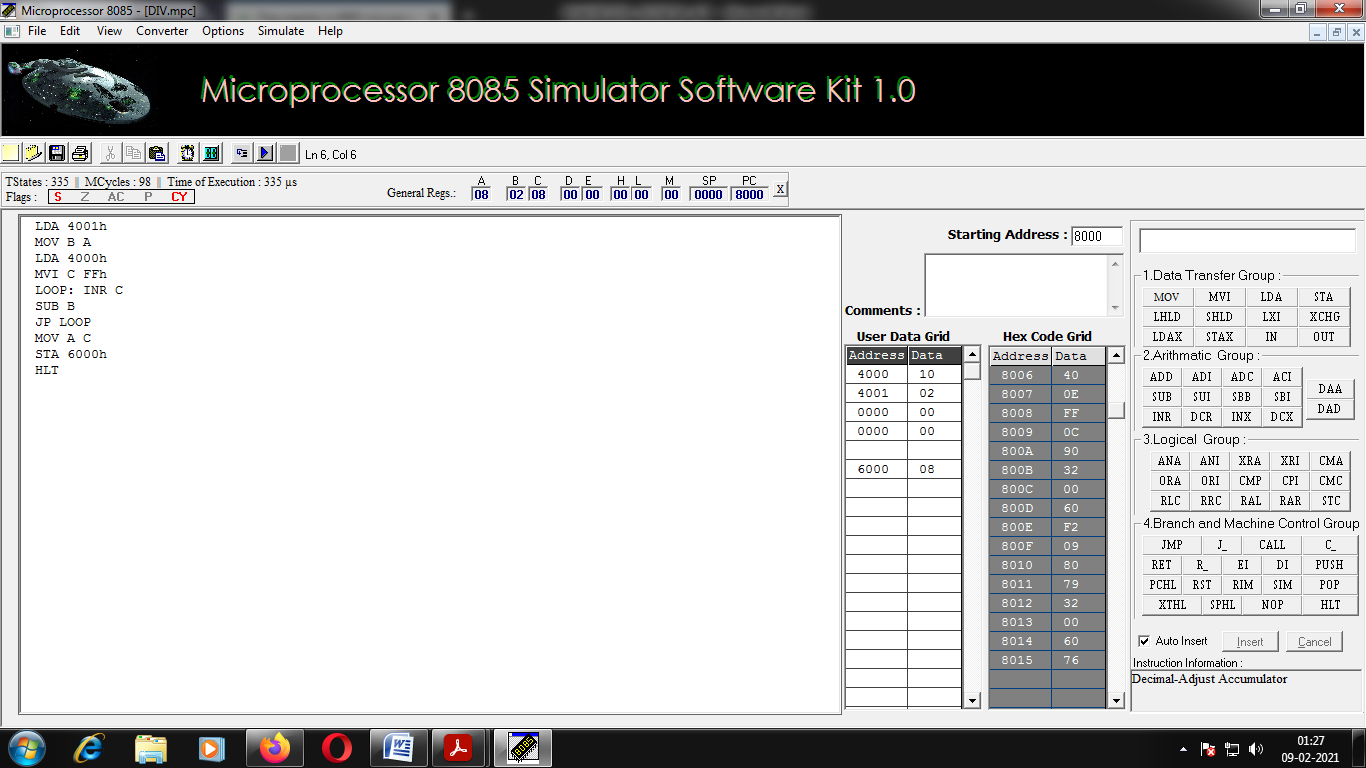
A🡸C (NO OF TIMES LOOP RUNS)

STORE THE RESULT IN 6000H

STEP3: ALP

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MEMORY ADDRESS | HEXA CODE | LABEL | OPCODE | OPERAND | COMMENT |
| 1000H | XX |  | LDA | 4001H | A🡨[4001] |
| 1001H | 01H |  |  |  |  |
| 1002H | XX |  | MOV | B, A | B🡨A |
| 1003H | XX |  | LDA | 4000H | A🡨[4000H] |
| 1004H | 00H |  |  |  |  |
| 1005H | 40H |  |  |  |  |
| 1006H | XX |  | MVI | C, FFH | C🡨FFH |
| 1007H | FFH |  |  |  |  |
| 1008H | XX | LOOP: | INR | C | C=C+1 |
| 1009H | XX |  | SUB | B | A🡸A-B |
| 100AH | XX |  | JP | LOOP | GOTO LOOP LABEL IF THE OUTPUT IS POSITIVE |
| 100BH | XX |  | MOV | A, C | A🡨C |
| 100CH | XX |  | STA | 6000H | [6000]🡨A |
| 100DH | 00H |  |  |  |  |
| 100EH | 60H |  |  |  |  |
| 100FH | XX |  | HLT |  | EXIT |

STEP 4: EXECUTION IN SIMULATION



**5. To find average of 5 Numbers:**

Write an ALP average of 5 numbers takes the numbers from 4000H to 4004H location and store the result in 4005H?

**Step 1: Pseudo code**

1. Initialize HL register with 4000h ,accumulator and counter
2. Add content pointing by the HL pair register with accumulator
3. Increment HL pair
4. Goto step 2 for 5 times to load data
5. Initialize counter
6. Increment counter
7. A🡸A-05
8. Incase sign flag active exit, else goto step 6 until it became negative.
9. Store the no of times loops runs to the 4005H memory location.

**Step 2: flow chart**

A🡸A+[HL]

L=L+1

C=C-1

HL🡨4000H

A🡨00H

C🡨05H

LOAD

ZERO FLAG

ZF=1 ZF=0

C=C+1

A🡸A-5

INITIALIZE COUNTER

SIGN FLAG

SF=0 AVERAGE

SF=1

**STEP 3: ALP**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MEMORY ADDRESS | HEXA CODE | LABEL | OPCODE | OPERAND | COMMENT |
| 1000H | XX 00H 40H |  | LXI | H, 4000H | H🡨[4000H] |
| 1003H | XX 00H |  | MVI | A, 00H | A🡨00H |
| 1005H | XX 05H |  | MVI | C, 05H | C🡨05H |
| 1007H | XX | LOAD: | ADD | M | A🡸A+[HL] |
| 1008H | XX |  | INR | L | INCREMENT MEMORY ADDRESS |
| 1009H | XX |  | DCR | C | C=C-1 |
| 100AH | XX |  | JNZ | LOAD | JUMP TO LOAD IF COUNTER NOT ZERO ELSE EXIT |
| 100BH | XX FFH |  | MVI | C, FFH | COUNTER INITIALIZE |
| 100DH | XX | AVERAGE: | INR | C | C=C+1 |
| 100EH | XX |  | SUB | L | A🡸A-05 |
| 100FH | XX |  | JP | AVERAGE |  |
| 1010H | XX |  | MOV | M, C | STORE THE NO OF TIMES EXECUTED IN THE 4005H |
| 1011H | XX |  | HLT |  |  |

**STEP 5: EXECUTION IN SIMULATION**

