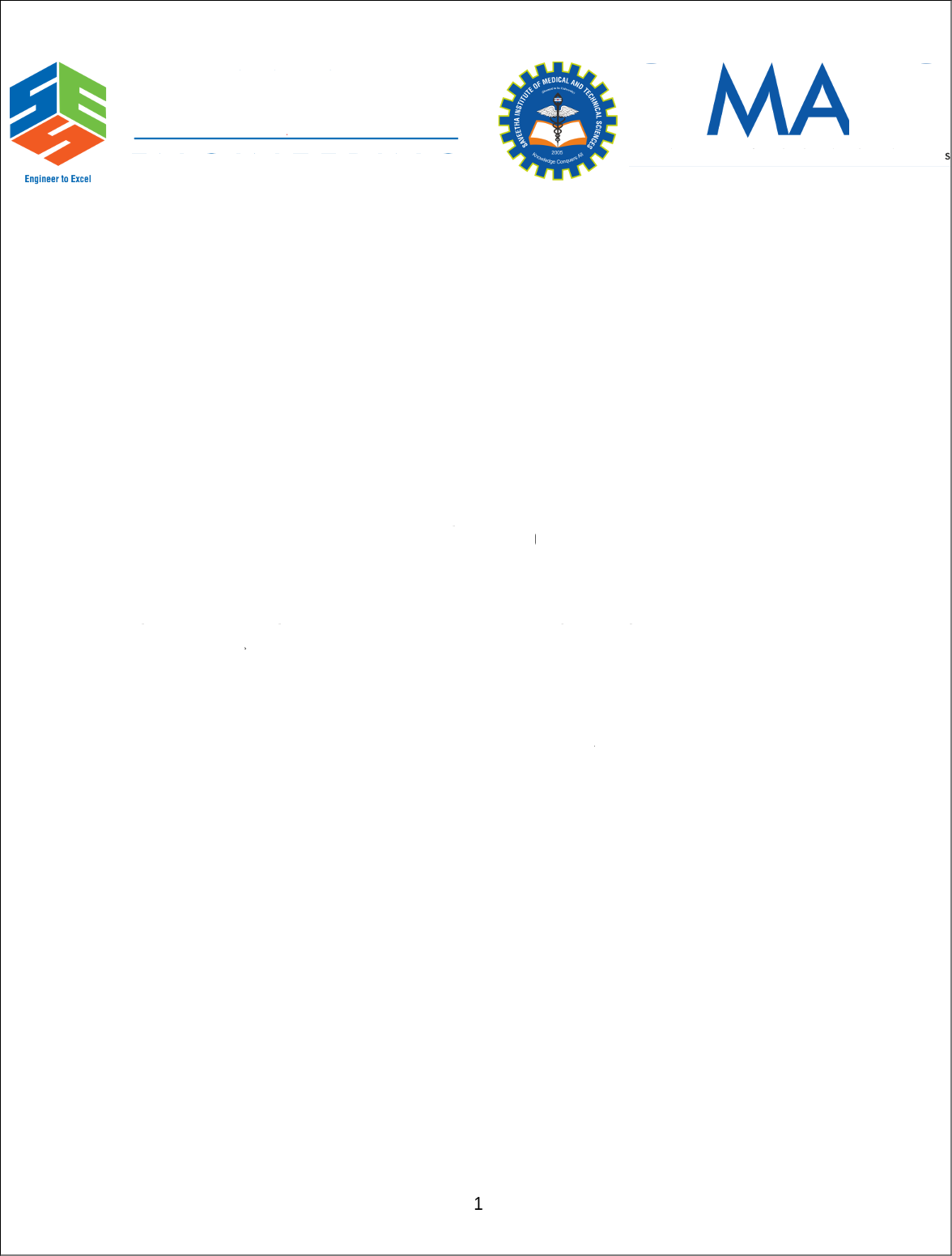
S IM AT S



SI 1S

goveetho Institute of Medical And Technical Science

E N G IN E E RI N G

(0ecl8red as Deemed to be university under Section 3 of UGC Act 1956)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

&A8MANUAL

## INDEX

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.No | Name of the Experiment | | | | | Page No. |
|  | 8085 Microprocessor | | | | |  |
| 1. | 8-bit Addition using direct addressing with carry in 8085 Microprocessor | | | | |  |
| 2. | 16-Bit addition with carry | | | | |  |
| 3. | 8-bit Subtraction with borrow in 8085 Microprocessor | | | | |  |
| 4. | 8-bit Multiplication operation using 8085 Microprocessor | | | | |  |
| 5. | 8-bit Division operation using 8085 Microprocessor | | | | |  |
| 6. | ASCII to HEX (Binary) Conversion using 8085 Microprocessor | | | | |  |
| 7. | Ascending Order of an Unsorted Array using 8085 Microprocessor | | | | |  |
| 8. | Descending Order of an Unsorted Array using 8085 Microprocessor | | | | |  |
| 9. | Searching the largest of given array of numbers | | | | |  |
| lo. | Searching the smallest of given array of numbers | | | | |  |
| 11. | Perform Masking and Setting of lower nibbles on given data using 8085 Microprocessor | | | | |  |
| 12. | Find One’s and Two’s complement of given data using 8085 Microprocessor | | | | |  |
|  | 8086 Microprocessor | | | | |  |
| 13. | 16-bit Addition using indirect addressing with carry in 8086 Microprocessor | | | | |  |
| 14. | 16-bit Subtraction with borrow in 8086 Microprocessor | | | | |  |
| 15. | 16-bit Multiplication operation using 8086 Microprocessor | | | | |  |
| 16. | 16-bit Division operation using 8086 Microprocessor | | | | |  |
| 17. | String Operation using 8086 Microprocessors  (Move a block of data from source to destination) | | | | |  |
| 18. | Logical operations in 8086 Microprocessor   1. Masking of bits using AND operation. 2. Setting of bits using OR operation. | | | | |  |
| 19. | Sum of N Numbers in a word array using 8086 Microprocessor | | | | |  |
|  | Interfacing with 8085/8086 | | | | |  |
| 20. | Interface Stepper Motor with 8085 Microprocessor  Anticlockwise direction | to | run | in | Clockwise and |  |
| 21. | Interface 8279 with 8086 fOl"  [a]. Character display using Keyboard Display Controller [b]. Rolling display using Keyboard Display Controller |  |  | the | following |  |
| 22. | Interface switches with 8086 via 8255 Programmable Peripheral Interface | | | | |  |
| 23. | Interface the Timer 8253 with 8085 Microprocessor to generate square wave | | | | |  |
|  | 8051 Microcontroller | | | | |  |
| 24. | Addition operation using 8051 Microcontroller | | | | |  |
| 25. | Subtraction operation using 8051 Microcontroller | | | | |  |
| 26. | Multiplication operation using 80S1 Microcontroller | | | | |  |
| 27. | Division operation using 8051 Microcontroller | | | | |  |
| 28. | Setting and Masking of bits using 8051 Microcontroller | | | | |  |

### 8-BIT ADDITION WITH CARRY USING DIRECT ADDRESSING

AIM:

To write an assembly language program to add tow numbers of 8-bit data stored in memory

locations 4200H and 420IH and store the result in 4202H and 4203H with carry using direct addressing.

APPARATUS REQUIRED

1. 8085 microprocessor kit ———————————-— 1

1. Power card ————1
2. Keyboard 1
3. 8085 Simulator and a PC

ALGORITHM:

* 1. Clear C reg to count carry
  2. Load the first data from memory to accumulator and move it to B register.
  3. Load the second data from memory to accumulator.
  4. Add the content of B — register to accumulator
  5. If Carry flag = 0 then jump to step 7

##### Increment C register to count the carry

* 1. Store the sum in memory.

###### Move the carry to accumulator and store in memory.

* 1. Stop.

PROGRAM TO ADD TWO 8-BIT DATA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Memory  address | Label | Instruction | Opcode | Comments |
|  |  | MVI C,00 |  | Clear C reg to count carry |
|  |  | LDA 4200H |  | Get 1st data in A and save in B. |
|  |  | MOV B, A |  |  |
|  |  | LDA 420IH |  | Get 2nd data in A-register |
|  |  | ADD B |  | Get the sum in A register |
|  |  | JNC SKI P |  | If CY=0 Then skip next step |
|  |  | INR C |  | Increment C register to count the  carry |
|  | SKIP | STA 4202H |  | Store the sum in memory |
|  |  | MOV A,C |  | Move the carry to accumulator and  store in memory |
|  |  | STA 4203H |  |  |
|  |  | HLT |  | Stop the Execution |

|  |  |  |  |
| --- | --- | --- | --- |
| Input | | Output | |
| Address | Data | Address | Data |
| 4200 |  | 4202 | (Sum) |
| 4201 |  | 4203 | (Carry) |

RESULT:

Thus, an assembly language program for addition of given two 8-bit number with carry was written, execute and Verified the Result successfully using 8085 kit.

AIM:

2. 16-BIT ADDITION WITH CARRY

To write an assembly language program to add two numbers of 16-bit data stored in memory

locations 4200H, 4201H, 4202H & 4203H and store the results in 4204H, 4205H & 4206H with carry.

APPARATUS REQUIRED

1. 8085 microprocessor kit 1
2. Power card 1
3. Keyboard 1
4. 8085 Simulator and a PC

ALGORITHM:

* 1. Clear B reg to count carry
  2. Load the first data from memory to HL pair and move it to DE pair.

###### Load the second data from memory to HL pair.

* 1. Add the content of BC Pair to HL pair
  2. If Carry flag = 0 then jump to step 6

##### Increment D register to count the carry

* 1. Store the sum in memory.

##### Move the carry to accumulator and store in memory.

###### Stop.

PROGRAM TO ADD TWO 8-BIT DATA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Memory  address | Label | Instruction | Opcode | Comments |
|  |  | MVI B,00 |  | Clear B reg to count carry |
|  |  | LHLD 4200H |  | Get 1st data in HL pair and save in DE. |
|  |  | XCHG |  |  |
|  |  | LHLD 4202H |  | Get 2nd data in HL-register |
|  |  | DAD D |  | ADD HL with DE and Get the sum in  HL register pair |
|  |  | JNC SKIP |  | If CY=0 Then skip next step |
|  |  | INR B |  | Increment C register to count the carry |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | SKIP | SHLD 4204H |  | Store the sum in memory |
|  |  | MOV A, B |  | Move the carry to accumulator and store  in memory |
|  |  | STA 4206H |  |  |
|  |  | HLT |  | Stop the Execution |

|  |  |  |  |
| --- | --- | --- | --- |
| Input | | Output | |
| Address | Data | Address | Data |
| 4200 |  | 4204 | (Sum) |
| 4202 |  | 4206 | (Carry) |

RESULT:

Thus, an assembly language program for addition of given two 16-bit number with carry was written, executeiJ and Verified the Result successfully using 8085 kit and Simulator.

### 3. 8-BIT SUBTRACTION WITH BORROW USING DIRECT ADDRESSING

AIM:

To write an assembly language program to subtract tow numbers of 8-bit data stored in memory

locations 4200H and 4201H and store the result in 4202H and 4203H with borrow using direct addressing. APPARATUS REQUIRED

1. 8085 microprocessor kit 1
2. Power card 1
3. Keyboard 1
4. 8085 Simulator and a PC

ALGORITHM:

1. Load the second data from memory to accumulator and move it to B register.

###### Load the first data from memory to accumulator.

1. Subtract the content of B — register from accumulator

###### If Carry flag = 0 then jump to step 5 & 6

1. Increment C register to count the borrow

###### Take two’s complement of the difference

1. Store the Difference in memory.

###### Move the borrow to accumulator and store in memory.

1. Stop.

PROGRAM TO SUBTRACT TWO 8-BIT DATA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Memory  address | Label | Instruction | Opcode | Comments |
|  |  | LDA 420IH |  | Get 2nd data in A and save in B. |
|  |  | MOV B, A |  |  |
|  |  | LDA 4200H |  | Get 1st data in A-register |
|  |  | SUB B |  | Subtract B-Reg from A register |
|  |  | JNC SKIP |  | If CY=0 Then skip next two steps |
|  |  | INR C |  | Increment C register to count the carry |
|  |  | CMA |  | Take two’s complement of difference |
|  |  | INR A |  |  |
|  | SKIP | STA 4202H |  | Store the Difference in memory |
|  |  | MOV A,C |  | Move the Borrow to accumulator and store  in memory |
|  |  | STA 4203H |  |  |
|  |  | HLT |  | Stop the Execution |

SAMPLE DATA:

|  |  |  |  |
| --- | --- | --- | --- |
| Input | | Output | |
| Address | Data | Address | Data |
| 4200 |  | 4202 | (Sum) |
| 4201 |  | 4203 | (Borrow) |

RESULT:

Thus, an assembly language program for subtraction of given two 8-bit number with borrow was writter , executed and Verified the Result successfully using 8085 kit.

1. 8-BIT MULTIPLICATION OPERATIONS USING 8085 MICROPROCESSOR

## AIM:

To write an assembly language program to multiply two numbers of 8-bit data stored in memory 4200H and 420IH and store the product in 4202H and 4203H.

**APPARATUS REQUIRED:**

1. 8085 microprocessor kit ———————————-— 1

Z. Power card ————1

* 1. Keyboard 1
  2. 8085 Simulator and a PC

**ALGORITHM:**

¥ Load the first data in ACC and move to E.

6 Load the second data ACC and move to B (count)

6 Clear HL pair (Initial sum)

###### 6 Clear D for overflow (carry)

6 Add the content of DE to HL

##### 6 Decrement the count.

6 Check whether count has reached zero.

+ Check the zero flag. If ZF = 0, repeat addition or If ZF = 1, go to next step

##### 6 Store the content of HL in memory. (Least significant 16 bits of the product)

6 Stop.

**PROGRAM TO MULTIPLY** TWO **NUMBERS OF 8-BIT** DATA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Memory  address | Label | Instruction | | Opcode | Comments |
|  |  | LDA | 4200H |  | ;Get 1 st data in A |
|  |  | MOV | E, A |  | ;Save Ist data in E |
|  |  | LDA | 4201H |  | ;Get 2nd data in A |
|  |  | MOV | B, A |  | ;save 2nd data in B |
|  |  | LXI | H,0000H |  | ;CIear HL pair(initial sum=0) |
|  |  | MVI | D,00H |  | ;CIear E for accounting overflow. |
|  | NEXT: | DAD | D |  | ;Add the content of DE to sum(HL) |
|  |  | DCR | B |  | Decrement data 2 for every addition |
|  |  | JNZ | NEXT |  | ;Repeat Addition until count is zero. |
|  |  | SHLD | 4202H |  | ;Store the product in memory |
|  |  | HLT |  |  | Stop the Execution |

**SAMPLE DATA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Address | Input Data | Address | Output Data | |
| 4200 | (Data-1) | 4202 |  | Lower byte of product) |
| 4201 | (Data-2) | 4203 |  | Higher byte of product) |

## RESULT:

Thus, an assembly language program to multiply two numbers of 8-bit data was written, executed an

Verified the Result successfully using 8085 kit.

* 1. 8-BIT DIVISION OPERATIONS USING 8085 MICROPROCESSOR

## AIM:

To write an ALP to perform division of two 8 bit numbers Stored in memory location 4200H,

4201H and Store the remainder in 4202H and the quotient in 4203H.

APPARATUS REQUIRED

1. 8085 microprocessor kit 1

2. - Power card 1

1. Keyboard 1
2. 8085 Simulator and aPC

ALGORITHM:

* 1. Load the divisor in accumulator and move if to B-register
  2. Load the dividend in accumulator.

###### Clear C-register to account for quotient

* 1. Check whether divisor is less than dividend
  2. If divisor is less than dividend, go to step 9, otherwise go to next step

##### Subtract the content of B-register (quotient)

###### Increment the content of C-register (quotient)

* 1. Go to step 4

##### Store the content of accumulator (remainder) in memory.

###### Move the content of C-register (quotient) to accumulator and store in memory

* 1. Stop.

PROGRAM TO DIVIDE TWONUMBERS OF 8-BIT DATA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Memory  address | Label | Instruction | | Comments |
|  |  | LDA | 4201H |  |
|  |  | MOV | B,A | ;Get the divisor in B register |
|  |  | LDA | 4200H | ;Get the dividend in A register |
|  |  | MVI | C,00H | ;CIear C register for quotient |
|  | AGAIN: | CMP | B |  |
|  |  | JC | STORE | ;If divisor is less than dividend go to store |
|  |  | SUB | B | ;Subtract divisor from dividend. Increment |
|  |  | INR | C | ;quotient by one for each subtraction. |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | JMP AGAIN |  |
|  | STORE: | STA 4203H | ;Store the remainder in memory |
|  |  | MOV A,C |  |
|  |  | STA 4202H | ;Stare the quotient in memory |
|  |  | HLT | Stop the Execution |

Sample data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Address | Input Data | | Address | Output Data |
| 4200 |  | Dividend) | 4202 | (Quotient) |
| 4201 |  | Divisor) | 4203 | (Remainder) |

RESULT:

##### Thus, an assembly language program to Divide two numbers of 8-bit data was written, eKecuted an I Verified the Result successfully using 8085 kit.

AIM:

6. SORT AN ARRAY OF DATA IN ASCENDING ORDER

To write an assembly language program to sort an array of data in ascending order. The array is storeiJ

in memory starting form 4200H. The first element of the array gives the count value for the number of elements in the array.

**APPARATUS REQUIRED:**

1. 8085 microprocessor kit 1

2. - Power card 1

1. Keyboard 1
2. 8085 Simulator and a PC

**ALGORITHM:**

* 1. Load the count value from memory to A-register and save it in B-register

##### Decrement B-register (B is a count for (N-1) repetitions)

###### Set HL pair as data address pointer

* 1. Set C-register as counter for (N-1) comparisons.
  2. Load a data of the array in accumulator using the data address pointer

##### Increment the HL pair (data address pointer)

###### Compare the data pointed by HL wait accumulator

* 1. If carry flag is set (If the content of accumulator is smaller than memory) then go to step 10, otherwis › go to next step
  2. Exchange the content of memory pointed by HL and the accumulator

#### Decrement C-register. If zero flag is reset go to step 6 otherwise go to next step

* 1. Decrement B-register. If zero flag is reset go the step 3 otherwise go to next step

##### Stop.

PROGRAM TO SORT AN ARRAY OF DATA IN ASCENDING ORDER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Memory Label Instruction Opcode address | | | | Comments |
|  | LDA | | 4200H | ;Load the count value |
|  | MOV | | B,A | ;Set counter for (N-1)  repetitions |
|  | DCR | | B | ;of (N-1) comparisons |
|  | LOOP 2 | LXI | H,4200H | ;Set pointer for array |
|  | MOV | | C,M | ;Set count for (N-1)  comparisons |
|  | DCR | | C |  |
|  | INX | | H | ;Increment pointer |
|  | LOOP I | MOV | A,M | ;Get one data of array in A |
|  | INX | | H |  |
|  | CMP | | M | ;Compare next data with A  register |
|  | JC | | AHEAD | ;If content of A is less than  memory then go to AHEAD |
|  | MOV | | D,M | ;If the content of A is greater  than |
|  | MOV | | M,A | ;then content of memory |
|  | DCX | | H | ;pointed by HL and previous  location |
|  | MOV | | M,D |  |
|  | INX | | H |  |
|  | AHEAD | DCR | C | ;Repeat comparisons until C  count is zero |
|  | JNZ | | LOOP I |  |
|  | DCR | | B | ;Repeat until B count is zero |
|  | JNZ | | LOOP 2 |  |
|  | HLT | |  | Stop the Execution |

Sample Data

|  |  |  |  |
| --- | --- | --- | --- |
| Address | Data Array (Before sorting) | Address | Data Array (After sorting) |
| 4200 | 07 (Count) | 4200 | 07 (Count) |
| 4201 | AB (Data -1) | 4201 | 34 (Data -1) |
| 4202 | 92 (Data -2) | 4202 | 4F (Data -2) |
| 4203 | 84 (Data -3) | 4203 | 69 (Data -3) |
| 4204 | 4F (Data -4) | 4204 | 84 (Data -4) |
| 4205 | 69 (Data -5) | 4205 | 92 (Data -5) |
| 4206 | F2 (Data -6) | 4206 | AB (Data -6) |
| 4207 | 34 (Data -7) | 4207 | F2 (Data -7) |

RESULT:

Thus, an assembly language program for sorting in Ascending order of an unsorted array of given 8-bit number was written, executed and Verified the Result successfully using 8085 kit.

7. SORT AN ARRAY OF DATA IN DESCENDING ORDER

AIM:

##### To write an assembly language program to sort the array of data in descending order. The array is stored i › memory stored in memory starting from 4200H. The first element of the array gives the count value for th › number of elements in the array.

APPARATUS REQUIRED:

1. 8085 microprocessor kit 1

2. - Power card 1

1. Keyboard 1
2. 8085 Simulator and a PC

ALGORITHM:

The algorithm is same as algorithm of example program IS except step 8.

Step 8: If carry flag is reset (If content of accumulator is larger than memory) then go to step 10, otherwise g › to next step

PROGRAM TO SORT AN ARRAY OF DATA IN DESCENDING ORDER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Memory  address | Label | Instruction Opcode | | Comments |
| 4100 | LDA | | 4200H | ;Load the count value |
| 4103 | MOV | | B,A | ;Set counter for (N-I) repetitions |
| 4104 | DCR | | B | ;of (N-I) comparisons |
| 4105 | LOOP 2 LXI | | H,4200H | ;Set pointer for array |
| 4108 | MOV | | C,M | ;Set count for (N-1) comparisons |
| 4109 | DCR | | C |  |
| 410A | INX | | H | ; Increment pointer |
| 410B | LOOP 1 MOV | | A,M | ;Get one data of array in A |
| 410C | INX | | H |  |
| 410D | CMP | | M | ;Compare next data with A register |
| 410E | JNC | | AHEAD | ;If content of A is less than memory then go  to AHEAD |
| 4111 | MOV | | D,M | ;If the content of A is greater than |
| 4112 | MOV | | M,A | ;then content of memory |
| 4113 | DCX | | H | ;pointed by HL and previous location |
| 4114 | MOV | | M,D |  |
| 4115 | INX | | H |  |
| 4116 | AHEAD | DCR | C | ;Repeat comparisons until C count is zero |

|  |  |  |  |
| --- | --- | --- | --- |
| 4117 | JNZ | LOOP I |  |
| 411A | DCR | B | ;Repeat until B count is zero |
| 41IB | JNZ | LOOP 2 |  |
| 41IE | HLT |  | Stop the Execution |

**SAMPLE DATA:**

|  |  |  |  |
| --- | --- | --- | --- |
| Address | Data Array  (Before sorting) | Address | Data Array  (After sorting) |
| 4200 | 07 (Count) | 4200 | 07 (Count) |
| 4201 | AB (Data -1) | 4201 | F2 (Data -7) |
| 4202 | 92 (Data -2) | 4202 | AB (Data -6) |
| 4203 | 84 (Data -3) | 4203 | 92 (Data -5) |
| 4204 | 4F (Data -4) | 4204 | 84 (Data -4) |
| 420S | 69 (Data -5) | 4205 | 69 (Data -3) |
| 4206 | F2 (Data -6) | 4206 | 4F (Data -2) |
| 4207 | 34 (Data -7) | 4207 | 34 (Data -1) |

**RESULT:**

##### Thus, an assembly language program for sorting in descending order of an unsorted array of given 8-bit number was written, executed and Verified the Result successfully using 8085 kit.

AIM:

8. SEARCH THE SMALLEST NUMBER FROM AN ARRAY

Write an assembly language program to search the smallest data in an array of N data stored in memory from

4200H to (4200H + N). The first element of the array gives the number of data in the array. Store the smallest data in 4300H. APPARATUS REQUIRED:

1. 8085 microprocessor kit 1

2. - Power card 1

1. Keyboard 1
2. 8085 Simulator and a PC

# ALGORITHM

* 1. Load the address of the first element of the array in HL register pair. (Pointer)
  2. Move the count to B-register
  3. Increment the pointer
  4. Get the first data in accumulator.
  5. Decrement the count
  6. Increment the pointer
  7. Compare the content of memory addressed by HL pair with that of accumulator
  8. If carry = 1, go to step 10 or if carry = 0, go to step 9
  9. Move the content memory addressed HL to accumulator.
  10. Decrement the count.
  11. Check for zero of the count. If ZF = 0, Go to step 6, or If ZF = 1 go to next step
  12. Store the smallest data in memory.
  13. Stop.

PROGRAM TO SEARCH SMALLEST DATA IN AN ARRAY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MEMORY  ADDRESS | LABEL | INSTRUCTION | | COMMENTS |
|  |  | ORG | 4100H | ;AssembIer directive |
|  |  | LXI | H,4200H | ;set pointer for array |
|  |  | MCV | B, M | ;set count for number of elements in array |
|  |  | INX | H |  |
|  |  | MCV | A, M | ;Set 1st element of array as smartest data |
|  |  | DCR | B | ;Decrement the count. |
|  | LOOP | INX | H | ;Compare on element of array |
|  |  | CMP | M | ;with current smallest data |
|  |  | JC | AHEAD | ;If CY = 1, go to AHEAD |
|  |  | MOV | A,M | ;If CY = 0 then content of memory |
|  |  |  | | :is smaller than A. Hence if CY = 0, |
|  |  |  | | ;Make memory as smallest by moving to A |
|  | AHEAD | DCR | B |  |
|  |  | JNZ | LOOP | , Repeat Comparison until count is zero |
|  |  | STA | 4300H | ;Store the smallest data in memory. |
|  |  | HLT | | Stop the Execution |

Sample data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Address | Input Data | | Address | Output Data |
| 4200 | 07 (Count) | | 4300 | 1 C (Smallest data in the array) |
| 4201 | 42 (Data -1) | |  |  |
| 4202 | 3A (Data -2 |  |  |  |
| 4203 | 1C (Data -3 |  |  |  |
| 4204 | 24 (Data -4 |  |  |  |
| 4205 | B4 (Data -5 |  |  |  |
| 4206 | 25 (Data -6) | |  |  |

RESULT:

Thus, an assembly language program for searching a smallest number from an array of given 8-bit number wa›

written, executed and Verified the Result successfully using 8085 kit and simulator.

AIM:

9. SEARCH THE LARGEST NUMBER FROM AN ARRAY

Write an assembly language program to search the largest data in an array of N data stored in memory from

4200H to (4200H + N). The first element of the array gives the number of data in the array. Store the smallest data in 4300H.

###### APPARATUS REQUIRED:

1. 8085 microprocessor kit 1

###### 2. - Power card 1

1. Keyboard 1
2. 8085 Simulator and a PC

# ALGORITHM

* 1. Load the address of the first element of the array in HL register pair. (Pointer)
  2. Move the count to B-register
  3. Increment the pointer
  4. Get the first data in accumulator.
  5. Decrement the count
  6. Increment the pointer
  7. Compare the content of memory addressed by HL pair with that of accumulator
  8. If carry = 1, go to step 10 or if carry = 0, go to step 9
  9. Move the content memory addressed HL to accumulator.
  10. Decrement the count.
  11. Check for zero of the count. If ZF = 0, Go to step 6, or If ZF = 1 go to next step
  12. Store the smallest data in memory.
  13. Stop.

PROGRAM TO SEARCH SMAI I FST DATA IN AN ARRAY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MEMORY  ADDRESS | LABEL | INSTRUCTION | | COMMENTS |
|  |  | ORG | 4100H | ;AssembIer directive |
|  |  | LXI | H,4200H | ;set pointer for array |
|  |  | MCV | B,M | ;set count for number of elements in array |
|  |  | INX | H |  |
|  |  | MCV | A,M | ;Set 1st element of array as smartest data |
|  |  | DCR | B | ;Decrement the count. |
|  | LOOP | INX | H | ;Compare on element of array |
|  |  | CMP | M | ;with current smallest data |

14

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | JNC | AHEAD | ;If CY = 0, go to AHEAD |
|  |  | MOV | A,M | ;If CY = 1 then content of memory |
|  |  |  | | :is largest than A. Hence if CY = 1, |
|  |  |  | | ;Make memory as largest by moving to A |
|  | AHEAD | DCR | B |  |
|  |  | JNZ | LOOP | ; Repeat Comparison until count is zero |
|  |  | STA | 4300H | ;Store the smallest data in memory. |
|  |  | HLT | | Stop the Execution |

Sample data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Address | Input Data | | Address | Output Data |
| 4200 | 07 (Count) | | 4300 | B4 (Largest data in the array) |
| 4201 | 42 (Data -1) | |  |  |
| 4202 | 3A (Data -2 |  |  |  |
| 4203 | 1C (Data -3 |  |  |  |
| 4204 | 24 (Data -4 |  |  |  |
| 4205 | B4 (Data -5 |  |  |  |
| 4206 | 25 (Data -6) | |  |  |

RESULT:

Thus, an assembly language program for searching a smallest number from an array of given 8-bit number wa› written, executed and Verified the Result successfully using 8085 kit and simulator.

10.ASCII CODE TO HEX CODE CONVERSION

AIM:

To write an assembly language program to convert an array of ASCI I codes to corresponding binar /

(Hex) value in 8085 Microprocessor

APPARATUS REQUIRED:

1. 8085 microprocessor kit 1
2. Power card 1
3. Keyboard 1
4. 8085 Simulator and aPC

ALGORITHM:

1. Get the ASCII data in A register from 4200H
2. Subtract 30H from A-register

###### Compare the content of A-register with OAH

1. If CY = 1 go to step 6. If CY = 0, go to next step

##### Subtract 07H from A-register

1. Store the HEX into 4202H
2. Stop the program

##### PROGRAM TO CONVERT ASCII CODE TO BINARY VALUE

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Memory  address | Label | Instruction | | oP Code | Comments |
|  |  | LDA | 4200 |  | Get the ASCII date to A register |
|  |  | SUI | 30H |  | ;Subtract 30h from the data |
|  |  | CPI | OAH |  | Compare the result with 0A |
|  |  | JC | STORE |  | ;If CY = 1, Store the result |
|  |  | SUI | 07H |  | ;EIse then subtract 07H |
|  | STORE | STA | 4201 |  | Store the result |
|  |  | HLT | |  | Stop the program |

SAMPLE DATA:

|  |  |  |  |
| --- | --- | --- | --- |
| ASCII Input | | Hex Output | |
| 4200 | 41 | 4201 | 0A |

RESULT:

Thus, an assembly language program for converting ASCII to HEX of given 8-bit number was written, executeiJ and Verified the Result successfully using 8085 kit.

##### 11.MASKING AND SETTING OF LOWER NIBBLES ON GIVEN DATA

AIM:

##### To write and execute an assembly language program for performing Masking, Setting, One’s and Two’s Complement of given data of 8-bit numbers using 8085 Microprocessor.

APPARATUS REQUIRED:

###### 8085 microprocessor kit 1

###### Power card 1

###### Keyboard 1

* 1. 8085 Simulator and aPC

MASKING OF BITS

ALGORITHM:

1. Load the Data in A-register.
2. Logically AND the content of A with OFH.
3. Store the result in memory location.
4. Stop the program

PROGRAM:

1. By using 8086 kit:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | LABEL | MNEMONICS | OPCODE | COMMENTS |
| 4100 |  | LDA 4200 |  | Load A-register with Data |
| 4103 |  | AN I , OFH |  | AND the content of A with OFH |
| 4105 |  | STA 4201 |  | Store the Result |
| 4108 |  | HLT |  | Stop the program |

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Address | Data | Address | Data |
| 4200H | 4A | 4201H | 0A |

SETTING OF BITS

ALGORITHM:

1. Load the Data in A-register.

###### Logically ORI the content of A with OFH..

1. Store the result in memory location.

###### Stop the program

PROGRAM:

1. By using 8086 kit:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | OPCODE | LABEL | MNEMONICS | COMMENTS |
| 4100 |  |  | LDA 4200 | Load A-reg ister with Data |
| 4103 |  |  | ORI, OFH | OR the content of A with OFH |
| 4105 |  |  | STA 4201 | Store the Result |
| 4108 | P4 |  | HLT | Stop the program |

OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Address | Data | Address | Data |
| 4200H | cs | 4201H | CF |

RESULT:

8085 kit.

Thus, an assembly language program for performing logical Masking and Setting of bits were executed using

AIM:

12.ONE’S AND TWO’S COMPLEMENT

##### To write and execute an assembly language program for performing One’s and Two’s Complement of given 8- bit numbers using 8085 Microprocessor.

APPARATUS:

1. 8085 microprocessor kit ———————————- 1

###### Power card 1

1. Keyboard———— 1

MASKING OF BITS ALGORITHM:

1. Load the Data in A-register.
2. Logically NOT the content of A.
3. Store the One’s complement in memory location.
4. Increment the content of A.
5. Store the Two’s complement in memory location.
6. Stop the program
7. By using 8086 kit:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ADDRESS | OPCODE | | LABEL | MNEMONICS | | COMMENTS | |
| 4100 |  | |  | LDA 4200 | | Load AL-register with 1“ Data | |
| 4103 |  | |  | CMA | | NOT the content of AX | |
| 4104 |  | |  | STA 4201 | | Store the One’s complement in  memory location. | |
| 4107 |  | |  | INR A | | Increment the content of AX. | |
| 4108 |  | |  | STA 4202 | | Store the Two’s complement in  memory location | |
| 410B | P4 | |  | HLT | | Stop the program | |
| OUTPUT: | | | | | | | |
| INPUT | | | | | OUTPUT | | |
| Address | | Data | | | Address | | Data |
| 4200H | | AB | | | 4201H | | 54 |
|  | |  | | | 4202H | | 55 |

RESULT:

Thus, an assembly language program for performing One’s and Two’s Complement of bits were executed using

8085 kit.

13. ADDITION OF 16 BIT NUMBERS WITH CARRY

AIM:

To write and execute an assembly language program to add two 16-bit unsigned numbers with carry in 8086 kit an‹ Emulator8086.

APPARATUS:

1. 8086 microprocessor kit 1
2. [Power card———— 1](#_TOC_250000)
3. Keyboard 1
4. Emulator8086 and PC

ALGORITHM:

1. Load the First Data in AX-register.
2. Load the First Data in BX-register.
3. Add the two data and get the sum in AX-register.
4. If C=0 then skip next step.

###### Increment CX Reg for carry

1. Store the sum in memory locations.

###### Store the Carry in memory location.

1. Stop the program.

P 7OGRAMM

ADDRESS LABEL

MNEMONICS

MOV CX , 000 OE

OPCODE

MOV AX, [1200] | MOV BX, [1202 ] |

D AX , BX

|WC I•1

|INC MOV MOV

!+++

CX

[1206] ,CX

[1204] ,AX

|

|Ll

COMMENT

|InitiaIize counter CX

|Get the first data in AX register.

|Get the second data in BX register.

Add the contents of both the register AX & BX

|Check for carry

|If carry exists, increment the CX

|Store the carry

|Store the sum

!Stoptheprogram

OUTPUT FOR ADDITION:

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Address | Data | Address | Data |
| 1200 |  | 1204 |  |
| 1202 |  | 1206 |  |

RESULT

Thus, an assembly language program for addition with carry of given 16-bit numbers was written, executed an 1 Verified the Result successfully using 8086 kit and Emulator8086.

AIM

14.SUBTRACTION OF 16 BIT NUMBERS WITH BORROW

To write and execute an assembly language program to subtract two 16-bit unsigned numbers with borrow in 8086

kit and Emulator8086.

APPARATUS:

###### 8086 microprocessor kit 1

###### Power card 1

###### Keyboard 1

1. Ernulator8086 and PC

ALGORITHM:

1. Load the second data from memory to accumulator and move it to B register.
2. Load the first data from memory to accumulator.
3. Subtract the content of B — register from accumulator

###### If Carry flag = 0 then jump to step S & 6

1. Increment C register to count the borrow
2. Take two’s complement of the difference

##### Store the Difference in memory.

1. Move the borrow to accumulator and store in memory.
2. Stop.

PROGRAMM

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ADDRESS | LABEL | MNEMONICS | | | | | OPCODE | COMMENT |
|  |  | MOV CX, OOOOH | | | | |  | Initialize counter CX |
|  |  | MOV AX, | | 1300 | |  |  | Get the first data in AX register |
|  |  | MOV BX, | | 1302 | |  |  | Get the second data in BX register. |
|  |  | SUB AX, BX | | | | |  | Subtract the contents of both the register  AX & BX |
|  |  | JNC SKIP | | | | |  | Check the Borrow. |
|  |  | INC CX | | | | |  | If carry exists, increment the CX |
|  |  | NEG AX | | | | |  | Take two’s complement of the  difference |
|  | SKIP | MOV | 1306 | | ,CX | |  | Store the Borrow. |
|  |  | MOV | 1304 | | ,AX | |  | Store the difference. |
|  |  | HLT | | | | |  | Stop the program |

OUTPUT FOR SUBTRACTION:

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | |  | |
| Address | Data | Address | Data |
| 1200 |  | 1204 |  |
| 1202 |  | 1206 |  |

RESULT

Thus, an assembly language program for subtraction with borrow of given 16-bit numbers was written, execute J and Verified the Result successfully using 8086 kit and Emulator8086.

AIM

15.MULTIPLICATION OF 16 BIT NUMBERS

1. To write and execute an assembly language program to Multiply two 16-bit unsigned numbers in 8086 kit and Ernulator8086.

APPARATUS:

1. 8086 microprocessor kit———————————-— 1

1. Power card 1
2. Keyboard ——— 1
3. Ernulator8086 and PC

ALGORITHM:

##### Load the multiplier from memory to accumulator.

###### Load the Multiplicand from memory to BX Reg .

1. Multiply AX with BX.
2. Store the Lower word in memory from AX.

##### Store the Higher word in memory from DX.

1. Stop.

PROGRAMM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | LABEL | MNEMONIC | OPCODE | COMMENTS |
|  |  | MOV AX, [1200] |  | Load AX-register with 1s’ data |
|  |  | MOV BX,[1202] |  | Load BX-register with 2" data |
|  |  | MUL BX |  | Multiply the contents of AX with  BX-register |
|  |  | MOV [1204],AX |  | Store the Lower word |
|  |  | MOV [1206],DX |  | Store the Higher word |
|  |  | HLT |  | Stop the program |



|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Address | Data | Address | Data |
| 1200 |  | 1204 |  |
| 1202 |  | 1206 |  |

RESULT

* 1. Thus, an assembly language program for multiplication of given 16-bit numbers was written, eKecuted and Verified the Result successfully using 8086 kit and Emulator8086.

AIM

**16.DIVISION OF 32BIT BY 16 BIT NUMBER**

To write and execute an assembly language program to Divide two 16-bit unsigned numbers in 8086 kit and

Emulaotr8086.

APPARATUS:

1. 8086 microprocessor kit———————————-— 1

* 1. Power card 1

###### Keyboard ——— 1

* 1. Ernulator8086 and PC

ALGORITHM:

##### Load the Divisor from memory to accumulator.

1. Load the Divisor from memory to BX Reg .
2. Divide DXAX by BX.
3. Store the \tuotient in memory from AX.

##### Store the Reminder in memory from DX.

1. Stop.

**PROGRAMM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ADDRESS** | **LABEL** | **MNEMONICS** | **OPCODE** | **COMMENTS** |
|  |  | MOV DX,[1200H] |  | Initialize DX-register with Lsb of  Dividend |
|  |  | MOV AX,[1202] |  | Load AX-register with Dividend |
|  |  | MOV BX, [1204] |  | Load BX-register with Divisor |
|  |  | DIV CX |  | Divide AX by BX-register |
|  |  | MOV [1206],AX |  | Store the Quotient |
|  |  | MOV [1209],DX |  | Store the Remainder |
|  |  | HLT |  | Stop the program |

OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| **INPUT** | | **OUTPUT** | |
| Address | Data | Address | **Data** |
| **1200** |  | 1206 |  |
| 1202 |  | 1208 |  |
| 1204 |  |  |  |

**RESULT**

Thus, an assembly language program for Division of given 16-bit numbers was written, executed and Verifie J

the Result successfully using 8086 kit.

###### 17.LOGICAL OPERATION

AIM:

1. To write and execute an assembly language program for performing Masking, Setting, One’s and Two’s Complement of given 16-bit numbers using 8086 Microprocessor kit and Emulator8086.

APPARATUS:

1. 8086 microprocessor kit ———————————-— 1

###### Power card ————1

1. Keyboard 1

###### EmuIator8086 and PC

MASKING OF BITS

ALGORITHM:

1. Load the Data in AX-register.

##### Logically AND the content of AX with OFOFH.

1. Store the result in memory location.

##### Stop the program

PROGRAM:

1. By using 8086 kit:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | OPCODE | LABEL | MNEMONICS | COMMENTS |
|  |  |  | MOV AX,[1200] | Load AL-register with 1“ Data |
|  |  |  | AND AX, OFOFH | AND the content of AX with OFOFH |
|  |  |  | MOV [1202],AX | Store the Result |
|  |  |  | HLT | Stop the program |

OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Address | Data | Address | Data |
| 1200H |  | 1202H |  |

SETTING OF BITS

ALGORITHM:

1. Load the Data in AX-register.
2. Logically OR the content of AX with OFOFH.
3. Store the result in memory location.
4. Stop the program

PROGRAM:

1. By using 8086 kit:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | OPCODE | LABEL | MNEMONICS | COMMENTS |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | MOV AX,[1200] | Load AL-register with 1“ Data |
|  |  |  | OR AX, OFOFH | AND the content of AX with OFOFH |
|  |  |  | MOV [1202],AX | Store the Result |
|  | F4 |  | HLT | Stop the program |

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Address | Data | Address | Data |
| 1200H |  | 1202H |  |

RESULT:

8086 kit.

Thus, an assembly language program for performing logical Masking and Setting of bits were executed using

AIM:

18.MOVE A DATA BLOCK WITHOUT OVERLAP

To write and execute an assembly language program for transferring data from one block to another block without overlapping using 8086 kit and Emulator8086.

APPARATUS:

###### 8086 microprocessor kit ———————————- 1

1. Power card ————1

###### Keyboard 1

1. Ernulator8086 and PC

ALGORITHM:

1. Initialize counter.
2. Initialize source block pointer.
3. Initialize destination block pointer.
4. Get the byte from source block.
5. Store the byte in destination block.
6. Increment source, destination pointers and decrement counter.
7. Repeat steps 4, 5 and 6 until counter equal to zero.
8. Stop.

PROGRAM:

1. By using 8086 kit:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | OPCODE | LABEL | MNEMONICS | COMMENTS |
|  | C7 C6 0012 |  | MOV SI, 1150H | Initialize the source address. |
|  | C7 C7 0013 |  | MOV DI,1250H | Initialize the destination  address. |
|  | C7 C1 0600 |  | MOV CX,0006 H | Initialize count value to the  count register. |
|  | FC | REPEAT: | OLD | Clear the direction flag. |
|  | A4 |  | MOVSB | Move the string byte. |
|  | E2, F3 |  | LOOP REPEAT | Unconditional loop to address  specified by the label REPEAT. |
|  | P4 |  | HLT | Stop the program |

OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| INPU I | | OU i i U i | |
| Address | Data | Address | Data |
| 1150. | 52. | 1250. | 52. |
| 1151. | 53. | 1251. | 53. |
| 1152. | 54. | 1252. | 54. |
| 1153. | 55. | 1253. | 5s. |
| 1154. | 56. | 1254. | 56. |

|  |  |  |  |
| --- | --- | --- | --- |
| 1155. | 57. | 1255. | 57. |

VIVA OUESTIONS:

* 1. What is the fabrication technology used for 8086?
  2. What are the functional units available in 8086 architecture?
  3. Write the flags of 8086.
  4. What are control bits?
  5. What are the flag manipulation instructions of 8086?
  6. What is Macro?
  7. Which bus controller used in maximum mode of 8086?
  8. What is the size of data bus and address bus in 8086?
  9. What are the various segment registers in 8086?
  10. What is the maximum memory addressing capability of 8086?

###### Thus, an assembly language program for transferring data from one block to another block without overlapping was executed using 8086 kit.

19.SUM OF N NUMBERS IN A WORD ARRAY

AIM:

###### To write and execute an assembly language program for adding N Numbers in a word array using 8086 kit and Emulator8086

APPARATUS:

###### 8086 microprocessor kit 1

1. Power card 1

###### Keyboard———— 1 4.ErnuIator8086 and PC

ALGORITHM:

1. Initialize counter.
2. Initialize source block pointer.
3. Initialize destination block pointer.
4. Get the byte from source block.
5. Store the byte in destination block.
6. Increment source, destination pointers and decrement counter.
7. Repeat steps 4, 5 and 6 until counter equal to zero.
8. Stop.

PROGRAM:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | LABEL | MNEMONICS | OPCODE | COMMENTS |
|  |  | MOV DX,0H |  | Clear DX |
|  |  | MOV SI,1250H |  |  |
|  |  | MOV CX,03H |  |  |
|  |  | MOV AX, [SI] |  |  |
|  | A1: | INC SI |  |  |
|  |  | INC SI |  |  |
|  |  | ADD AX, [SI] |  |  |
|  |  | JNC NEXT |  |  |
|  |  | INC DX |  |  |
|  | NEXT: | LOOP A1 |  |  |
|  |  | MOV [1300H], AX |  |  |
|  |  | MOV [1302H], DX |  |  |
|  |  | HLT |  |  |

OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Address | Data | Address | Data |
| 1250 |  | 1300 |  |
| 1252 |  | 1302 |  |
| 1254 |  |  |  |
| 1256 |  |  |  |

VIVA C,tUESTIONS:

* 1. What is the fabrication technology used for 8086?
  2. What are the functional units available in 8086 architecture?
  3. Write the flags of 8086.
  4. What are control bits?

s What are the flag manipulation instructions of 8086?

1. What is Macro?
2. Which bus controller used in maximum mode of 8086?
3. What is the size of data bus and address bus in 8086?
4. What are the various segment registers in 8086?
5. What is the maximum memory addressing capability of 8086?

RESULT:

Thus, an assembly language program for transferring data from one block to another block without overlapping was executed using 8086 kit .

20.STEPPER MOTOR INTERFACING

AIM:

To write and execute an assembly language Program to run a stepper motor at different speed, and to control its

###### direction using 8085 Microprocessor

APPARATUS:

###### 8085 microprocessor kit 1

###### Stepper Motor ————1

###### Stepper Motor Interface board 1

###### Power card 1

###### Keyboard ————1

PROGRAM:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | LABEL | MNEMONICS | OPCODE | COMMENTS |
| 4100 | START | LXI H, 4200 |  | Initialize HL with 4200H |
| 4103 |  | MVI C, 04 |  | Copy the value 04 to C- register |
| 4105 | NEXT | MOV A, M |  | Copy the content M to A-register |
| 4106 |  | OUT C0 |  | The content of A is moved to Out port |
| 4108 |  | LXI D, 1010 |  | Copy the data 1010 to DE-reg Pair |
| 410B | loop | DCX D |  | Decrement DE-register |
| 410C |  | MOV A,E |  |  |
| 410D |  | ORA D |  | Check DE = 0000 |
| 410E |  | JNZ loop |  | Jump on no zero to loop |
| 4111 |  | INX H |  | Increment HL -register Pair |
| 4112 |  | DCR C |  | Decrement the count |
| 4113 |  | JNZ NEXT |  | Jump to NEXT if Z flag is zero |
| 4115 |  | JMP START |  | Jump to label START |
| 4118 |  | HLT |  | Stop the program. |
| 4200 | TABLE | 09 05 06 0A |  | clockwise direction |
| 4200 | TABLE | 0A06 05 09 |  | Counter clockwise direction |

OUTPUT

|  |  |  |
| --- | --- | --- |
| INPUT | | OUTPUT |
| Address | Data | Motor Rotates on |
| 4200 | 09 05 06 OA | clockwise direction |
| 4200 | 0A 06 05 09 | Counter clockwise direction |

RESULT:

Thus an assembly language Program to run the stepper motor in both forward and reverse direction with

delay was executed and its output was verified.

21.KEYBOARD AND DISPLAY

AIM:

To write and execute an assembly language Program to display a character “7” and the rolling message “HELP

US” in the display.

APPARATUS:

###### 1. 8086 microprocessor kit 1

###### Z. 8ZT9 Interface board 1

###### Power card 1

###### Keyboard 1

ROLLING MESSAG E “H ELP US”

ALGORITHM:

Display of rolling message “HELF Uñ “

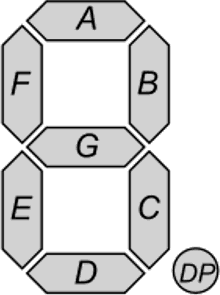
1. Initialize the counter
2. Set 8279 for 8 digit character display, right entry
3. Set 8279 for clearing the display
4. Write the command to display
5. Load the character into accumulator and display it
6. Introduce the delay
7. Repeat from step 1.

PROGRAM:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | LABEL | PROGRAM | OPCODE | COMMENTS |
| 1100 | START | MOV SI,1200H |  | Initialize array |
| 1104 |  | MOV CX,000FH |  | Initialize array size |
| 1108 |  | MOV AL,10 |  | Store the control word for display mode |
| 110B |  | OUT C2,AL |  | Send through output port |
| 110D |  | MOV AL,0CC |  | Store the control word to clear display |
| 1110 |  | OUT C2,AL |  | Send through output port |
| 1112 |  | MOV AL,90 |  | Store the control word to write display |
| 1115 |  | OUT C2,AL |  | Send through output port |
| 1117 | NEXT | MOV AL,[SI] |  | Get the first data |
| 1119 |  | OUT C0,AL |  | Send through output port |
| 111B | DELAY | MOV DX,0FFFFH |  | Store 16bit count value |
| 111F | LOOP1 | DEC DX |  | Decrement count value |
| 1120 |  | JNZ LOOP1 |  | Loop until count values becomes zero |
| 1122 |  | INC SI |  | Go & get next data |
| 1123 |  | LOOP NEXT |  | Loop until all the data has been taken |
| 1125 |  | JMP START |  | Go to starting location |
| 1127 |  | HLT |  |  |

LOOK-UP TABLE:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1200 | 98 | 68 | 7C | C8 |
| 1204 | FF | IC | 29 | FF |



ON — 0 OFF - 1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MEMORY  LOCATION | Message | 7-SEGMENT LED FORMAT | | | | | | | | HEX CODE |
| D | C | B | A | DP | G | F | E |
| 1200H | W | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 98 |
| 1201W | E | 0 | I | I | 0 | 1 | 0 | 0 | 0 | 68 |
| l202lH | L | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 7C |
| 1203W | P | 1 | I | 0 | 0 | 1 | 0 | 0 | 0 | C8 |
| 1204W |  | 1 | I | I | 1 | 1 | 1 | 1 | 1 | FF |
| 1205H | U | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1C |
| 1206H | S | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 29 |
| 1207W |  | 1 | I | I | 1 | 1 | 1 | 1 | 1 | FF |

DISPLAY THE CHARACTER “3” ALGORITHM:

###### Set 8279 for 8-digit character display, right entry

1. Set 8279 for clearing the display

###### Write the command to display

1. Load the character into accumulator and display it

###### Repeat from step 1.

PROGRAM:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | LABEL | PROGRAM | OPCODE | COMMENTS |
| 1100 |  | MOV AL,00 | C6 C0 00 | Store the control word for display mode |
| 1103 |  | OUT C2,AL | E6 C2 | Send through output port |
| 1105 |  | MOV AL,0CC | C6 C0 CC | Store the control word to clear display |
| 1108 |  | OUT C2,AL | E6 C2 | Send through output port |
| 110A |  | MOV AL,90 | C6 C0 90 | Store the control word to write display |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 110D |  | OUT C2,AL | E6 C2 | Send through output port |
| 110F |  | MOV AL,8F | C6C08F | Get the first data |
| 1112 |  | OUT C0,AL | E6 C0 | Send through output port |
| 1114 |  | HLT | F4 | Stop the program |

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Address | Data | Address | Data |
| 1111 | 0B | Display | 3 |

VIVA C,tUESTIONS:

1. What are the types of interfacing?
2. Compare memory interfacing and IO interfacing.
3. What is the difference between memory mapped IO and IO mapped IO interfacing?
4. What IC 8279 is>
5. What are the tasks involved in keyboard interface?
6. What is scanning in keyboard and what is scan time?
7. What is the difference between 2-key and n-key roIIover+
8. What is the control registers available in 8279?
9. What is key debouncing?
10. What are the command words available in 8279+



Thus, the rolling message “HELP US” and the character “3” are displayed using 8279 interface kit with 8086

Microprocessor

22.INTERFACE SWITCHES WITH 8086 THROUGH 8255

AIM:

To write and execute an assembly language Program to Interface 8 switches with 8086 Microprocessor through 8Z55 PPI.

APPARATUS:

1. 8086 microprocessor kit ———————-—————1

###### Z. 8Z55 Interface board 1

1. Power card 1

###### Keyboard ————1

ALGORITHM:

1. Configure the 8255 port A as input port with the control reg value as “90H”

###### Read the port A switch status through C0.

1. Store the output in 1250.

###### Stop

PROGRAM:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | LABEL | PROGRAM | OPCODE | COMMENTS |
| 1100 |  | MOVAL,90 |  | Load the AL with control word |
| 1103 |  | OUT C2,AL |  | Send the control word to control reg of 8255 |
| 1105 |  | IN AL,C0 |  | Read port A |
| 1108 |  | MOV [1250],AL |  | Store the result on memory |
| 1114 |  | HLT | F4 | Stop the program |

|  |  |  |
| --- | --- | --- |
| INPUT | OUT  PUT | |
| VARY THE SWITCH POSITIONS  ON OFF ON ON OFF ON OFF ON | Address | Data |
| 1250 |  |

RESULT

Thus, an assembly language program for Interfacing of switches with 8086 through 8255 PPI was writter , executed and Verified the Result successfully.

AIM:

23.TRAFFICLIGHT INTERFACE

To write and execute an assembly language program for traffic light interfacing to handle the traffic using 8085 Microprocessor.

APPARATUS RE§fUIRED:

1. 8085 microprocessor kit —————————————1

1. Traffic light Interface board 1
2. Power card 1
3. Keyboard ————1

ALGORITHM:



PROGRAM: SQUARE WAVE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | LABEL | PROGRAM | OPCODE | COMMENTS |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

OUTPUT:

RESULT:

Thus, an assembly language program for Interfacing of Timer 8253/54 with 8085 was written, executed and Verified the Result successfully.

24.ADDITION OPERATION USING 8051 MICROCONTROLLER

AIM:

To write and execute an assembly language program to Add of two 8-bit numbers using 8051 and Keil software.

APPARATUS:

1. 8051 microcontroller kit 1
2. Power card 1
3. Keyboard ——— 1
4. Keil and PC

ALGORITHM:

1. Load the First Data in A-register.
2. Load the Second Data in B-register.
3. Add the two data with carry.

###### Store the sum in memory location.

1. Stop the program.

PROGRAM:

ADDITION

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | OPCODE | LABEL | PROGRAM | COMMENTS |
| 4100 | 74,05 |  | MOVA,#data | Load data 1 in accumulator. |
| 4102 | 75,F0,05 |  | MOV B,#data | Load data 2 in B-register |
| 4105 | 35,F0 |  | ADDC A,B | Add the contents of accumulator and B-reg  with carry. |
| 4107 | 90,11,00 |  | MOV DPTR,#4500H | Initialize DPTR with address 4500a |
| 410A | F0 |  | MOVX @ DPTR,A | Store the Sum in 4500a |
| 410B | 80, FE | STOP: | SJMP STOP | Stop the program |

OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | |  | |
| Register | Data | Address | Data |
| 4101 |  | 4500 |  |
| 4104 |  |  |  |

RESULT:

Thus, an assembly language program for addition of given two 8-bit number was written. executed and Verified the Result successfully using 8051 kit

* 1. **SUBTRACTION OPERATION USING 8051 MICROCONTROLLER**

AIM:

To write and execute an assembly language program to subtract two 8-bit numbers using 8051.

APPARATUS:

1. 8051 microcontroller kit 1

###### Power card 1

1. Keyboard ——— 1

###### Keil and PC

**SUBTRACTION**

ALGORITHM:

1. Load the First Data in A-register.
2. Load the Second Data in B-register.

###### Subtract the two data with borrow.

1. Store the sum in memory location.

###### Stop the program.

**PROGRAM:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ADDRESS** | **OPCODE** | **LABEL** | **PROGRAM** | COMMENTS |
| 4100 | 74,05 |  | MOV A,#data | Load data 1 in accumulator. |
| 4102 | 75,F0,04 |  | MOV B,#data | Load data 2 in B-register |
| 4105 | 95,F0 |  | SUBB A,B | Subtract the contents of B-  reg from accumulator with borrow. |
| 4107 | 901100 |  | MOV DPTR,#4500H | Initialize DPTR with address 4500a |
| 410A | F0 |  | MOVX @ DPTR,A | Store the difference in 4500H |
| 410B | 80, FE | STOP: | SJMP STOP | Stop the program |

**OUTPUT:**

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Register | Data | Address | **Data** |
| 4101 |  | **4500** |  |
| 4104 |  |  |  |

**RESULT:**

Thus, an assembly language program for subtraction of given two 8-bit number was written, executed and Verified the Result successfully using 8051 kit

AIM:

* 1. MULTIPLICATION OPERATION USING 8051 MICROCONTROLLER

To write and execute an assembly language program to multiply two 8-bit numbers using 8051.

APPARATUS:

* + 1. 8051 microcontroller kit 1

###### Power card 1

* + 1. Keyboard ——— 1

###### Keil and PC

MULTIPLICATION

ALGORITHM:

1. Get the multiplier in the accumulator.

###### Get the multiplicand in the B register.

1. Multiply A with B.
2. Store the product in memory locations.
3. Stop the program.

PROGRAM:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | OPCODE | LABEL | PROGRAM | COMMENTS |
| 4100 | 74,05 |  | MOVA,#data | Load data 1 in accumulator. |
| 4102 | 75, F0,05 |  | MOV B,#data | Load data 2 in B-register |
| 4105 | A4 |  | MUL AB | A\*B, Higher byte of result in B and lower byte of result in A. |
| 4106 | 90,11,00 |  | MOV DPTR,#4500H | Initialize DPTR with address 1100a |
| 4109 | F0 |  | MOVX @ DPTR,A | Store the LSB in 4500a |
| 410A | A3 |  | INC DPTR | Increment Data pointer |
| 410B | E5,F0 |  | MOV A,B | Copy the content of B-reg to A-register. |
| 410D | F0 |  | MOVX @ DPTR,A | Store the MSB in 4501H |
| 410E | 80, FE | STOP: | SJMP STOP | Stop the program |



|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| REGISTER | DATA | ADDRESS | DATA |
| 4101 |  | 4500 |  |
| 4104 |  | 4501 |  |

RESULT:

Thus, an assembly language program for multiplication of given two 8-bit number was written, executed and Verified the Result successfully using 8051 kit

AIM:

###### DIVISION OPERATION USING 8051 MICROCONTROLLER

To write and execute an assembly language program to divide two 8-bit numbers using 8051.

APPARATUS:

* + 1. 8051 microcontroller kit 1
    2. Power card 1
    3. Keyboard ——— 1

###### Keil and PC

DIVISION

ALGORITHM:

1. Get the Dividend in the accumulator.
2. Get the Divisor in the B register.
3. Divide A by B.
4. Store the Quotient and Remainder in memory.
5. Stop the program.

PROGRAM:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | OPCODE | LABEL | PROGRAM | COMMENTS |
| 4100 | 74,data1 |  | MOV A,#CF | Load data 1 in accumulator. |
| 4102 | 75,data2 |  | MOV B,#21 | Load data 2 in B-register |
| 4104 | 84 |  | DIV AB | Divide. Remainder in A and quotient in B |
| 4105 | 90,11,00 |  | MOV DPTR,#4500H | Initialize DPTR with address 1100H |
| 4108 | F0 |  | MOVX @ DPTR,A | Store the quotient in 4500H |
| 4109 | A3 |  | INC DPTR | Increment Data pointer |
| 410A | E5,F0 |  | MOV A,B | Copy the content of B-reg to A-register. |
| 410C | F0 |  | MOVX @ DPTR,A | Store the Remainder in 4501a |
| 410D | 80, FE | STOP: | SJMP STOP | Stop the program |

OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| REGISTER | DATA | ADDRESS | DATA |
| 4101 |  | 4500 | (quotient) |
| 4104 |  | 4501 | (remainder) |

RESULT:

Thus, an assembly language program for Division of given two 8-bit number was written, executed and Verified the Result successfully using 8051 kit

###### 28.LOGICAL OPERATIONS USING 8051

AIM:

To write and execute an assembly language program for Setting and Masking of given 8-bit number using 8051.

APPARATUS REQUIRED:

###### 8051 microcontroller kit 1

1. Power card 1

###### Keyboard ——— 1

1. Keil and PC

###### Load the Data in A-register.

1. Load OF to set the lower nibble in B-register.

###### 8. Perform OR operation with B-register.

1. Store the Result in memory location.

###### Stop the program.

PROGRAM:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | OPCODE | LABEL | PROGRAM | COMMENTS |
| 4100 | 74,05 |  | MOVA,#C3 | Load data 1 in accumulator. |
| 4102 | 75,F0,05 |  | MOV B,#0F | Load data 2 in B-register |
| 4105 | 35,F0 |  | ORL A,B | OR the contents of accumulator and B-reg. |
| 4107 | 90,11,00 |  | MOV DPTR,#4500H | Initialize DPTR with address 4500a |
| 410A | F0 |  | MOVX @ DPTR,A | Store the Result in 4500H |
| 410B | 80, FE | STOP: | SJMP STOP | Stop the program |

OUTPUT:

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | |  | |
| Register | Data | Address | Data |
| 4101 | C3 | 4500 | CF |

MASKING OF BITS

ALGORITHM:

1. Load the Data in A-register.

###### Load OF to mask the higher nibble in B-register.

1. Perform AND operation with B-register.

###### Store the Result in memory location.

1. Stop the program.

PROGRAM:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADDRESS | OPCODE | LABEL | PROGRAM | COMMENTS |
| 4100 | 74,05 |  | MOVA,#4D | Load data 1 in accumulator. |
| 4102 | 75,F0,05 |  | MOV B,#0F | Load data 2 in B-register |
| 4105 | 35,F0 |  | ANL A, B | AND the contents of accumulator and B-  reg. |
| 4107 | 90,11,00 |  | MOV DPTR,#4500H | Initialize DPTR with address 4500a |
| 410A | F0 |  | MOVX @ DPTR,A | Store the Result in 4500H |
| 410B | 80, FE | STOP: | SJMP STOP | Stop the program |

|  |  |  |  |
| --- | --- | --- | --- |
| INPUT | | OUTPUT | |
| Register | Data | Address | Data |
| 4101 | 4D | 4500 | OD |

RESULT:

Thus, an assembly language program for Setting and Masking of 8-bit numbers using 8051 were

performed and its outputs were verified.