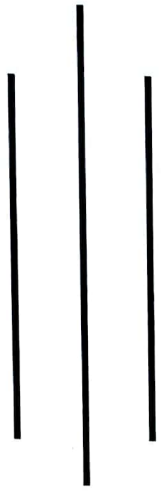
TRIBHUVAN UNIVERSITY

PATAN MULTIPLE CAMPUS

PATAN DHOKA, LALITPUR



MICROPROCESSOR AND COMPUTER ARCHITECTURE (BIT 151)

SUBMITTED BY

NAME: DOLINDRA BAHADUR RAUT

CLASS: BIT – I/II

ROLL NO: 24

DATE: 2081/04/11

SUBMITTED TO

JYOTI PRAKASH CHAUDHARY

.....

CHECKED BY

8.1. Write an assembly language to add two 8-bit numbers without carry.

* Solution :-

① objective :- To write a program (in assembly language) to add two 8-bit numbers.

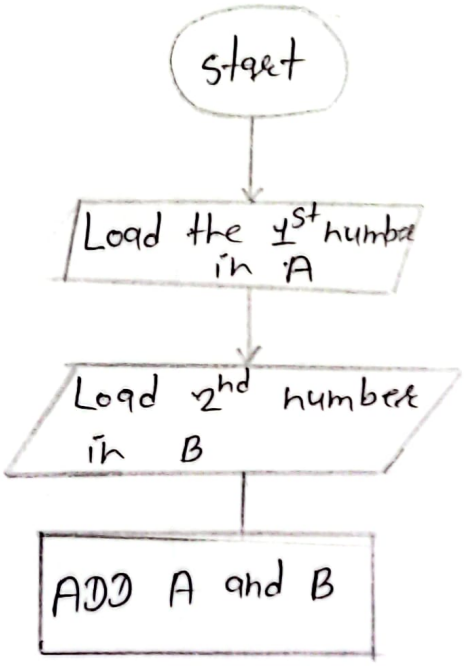
② Requirements :-

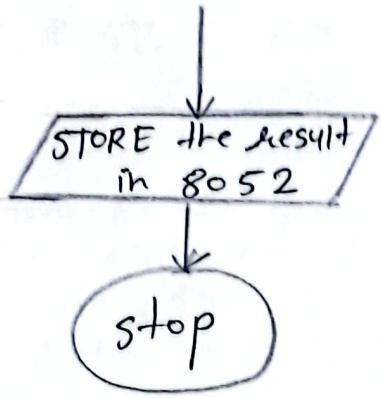
i) 8085 Microprocessor kit / 8085 simulator
ii) (0-5V) DC Battery

③ ~~Flowchart :-~~ Algorithm :-

step 1 : Start the microprocessor.
step 2 : Load the first 8-bit data into accumulator.
step 3 : Load the second 8-bit data into B-register.
step 4 : Add the two 8-bit data.
step 5 : Store the result from accumulator to the specified memory location.
step 6 : Stop the program execution :-

④ Flowchart :-





⑤ Main program:-

```
MVI   A, AB
MVI   B, FF
ADD   B
STA   8052
HLT
```

⑥ Result:-

Result after Assembling:

Address	op code
8000	3E
8001	AB
8002	06
8003	FF
8004	80
8005	32
8006	52
8007	80
8008	76
~~8009~~	

Final result : 8052 = AA

⑦ Conclusion :-

Hence, the two 8-bit data is added using 8085 microprocessor and result was verified. The final result i.e. sum of two numbers is stored in 8052 memory address.

08/04/11

8.2. Write an assembly language to subtract 2 8-bit numbers without borrow.

Solution:

① objective: To write a assembly program that subtract two 8-bit numbers.

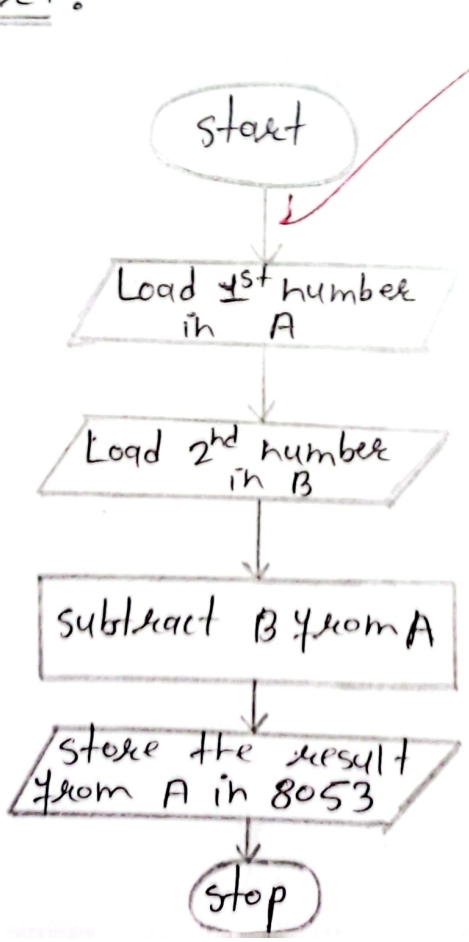
② Requirements:

i) 8085 Microprocessor kit / 8085 simulator
ii) DC Battery of (0-5V)

③ Algorithm:

step1 : start the program
step2 : Load the first 8-bit data into accumulator.
step3 : Load the second 8-bit data into B-register.
step4 : subtract the two 8-bit data.
step5 : store the result from accumulator to the specified memory location.
step 6 : stop the program execution.

④ Flowchart:



⑤ Main program :-

```
MVI  A, 08
MVI  B, 02
SUB  B
STA  8053
HLT
```

⑥ Result :-

Result after Assembling :-

Address	op code
8000	3E
8001	08
8002	06
8003	02
8004	90
8005	32
8006	53
8007	80
8008	76
~~8009~~	

Final result : 8053 = 06

⑦ conclusion :-

Hence, the two 8-bit data was being subtracted using 8085 microprocessor. We subtracted 02 from 08 and then stored the result in 8053 memory location and result was verified.

08/04/11

Q.3. Write an assembly language program to add two data stored in memory location 8050 and 8051 and stored the sum in memory location 8052 and carry (if any) in memory location 8053.

Solution:

① Objective :- To write an assembly programm to add 2 data stored in memory location 8050 and 8051 ~~and store the~~ :- to store the result in memory location 8052 and carry (if any) in memory location.

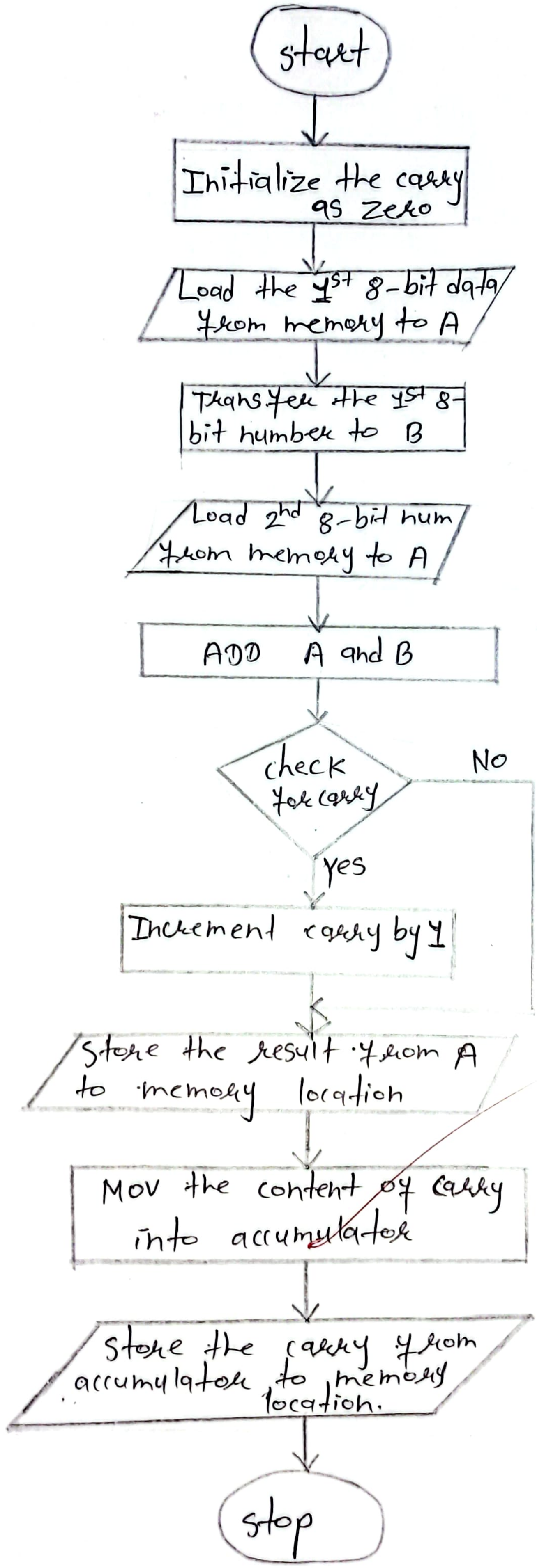
② Requirements :-

i) 8050 Microprocessor kit / 8085 simulator
ii) DC Battery of (0–5V).

③ Algorithm :-

step1 : Start the microprocessor.
step2 : ~~Load the first 8-bit data into~~ Initialize carry as zero (0).
step3 : Load the first 8-bit data from memory location into accumulator.
step4 : copy content of accumulator to register B.
step5 : Load the second 8-bit data into accumulator from memory location.
step6 : Add the 2 8-bit data and check for carry.
step7 : Jump if no carry.
step8 : Increment carry if it is there.
step9 : store the result of sum from accumulator to the specified memory location.
step10 : store the value of carry from accumulator to memory location.
step11 : stop the program execution.

(4) Flowchart:-



⑤ Main program :-

```
MVI   C, 00
LDA   8050
MOV   B, A
LDA   8051
ADD   B
JNC   SKIP
INR   C

SKIP:
STA   8052
MOV   A, C
STA   8053
HLT
```

⑥ Result :-

Address	Op code
8000	0E
8001	00
8002	3A
8003	50
8004	80
8005	47
8006	3A
8007	51
8008	80
8009	80

⑦ Conclusion :-

Hence, the two 8-bit data was being added using 8085 microprocessor and result was verified with carry also.

08/04/11

8. 4. Write an assembly language program to subtract 2 8-bit values stored in memory location 8050 and 8051 and store the result of subtraction in memory location 8052 and borrow (if any) in memory location 8053.

solution:-

① objective :-

: To write an assembly program that subtract two 8-bit numbers with borrow.

② Requirements :-

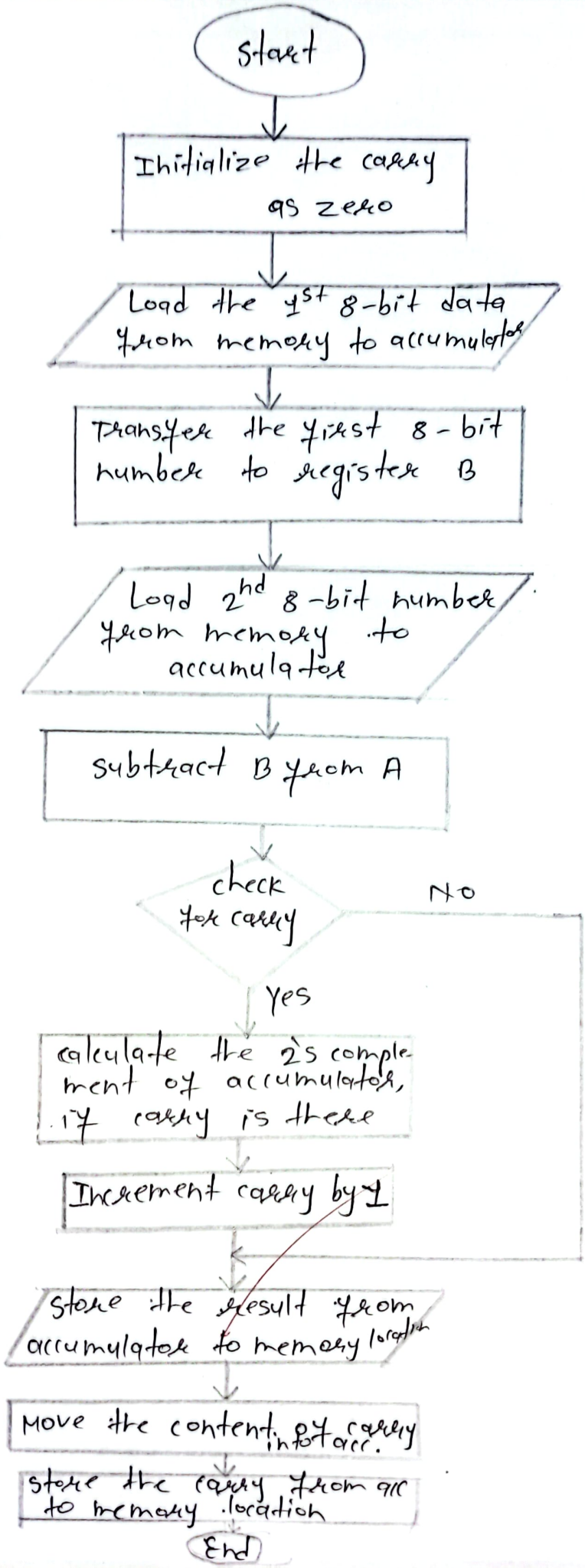
i) 8085 Microprocessor kit / 8085 simulator

ii) (0-5V) DC Battery

③ Algorithm :-

step1 : start the microprocessor.
step2 : Initialize C register as zero to store borrow.
step3 : Load the first 8-bit data from memory location on into accumulator.
step4 : copy content of accumulator to register B.
step5 : Load the second 8-bit data into accumulator from memory location.
step6 : subtract the two 8-bit data and check for carry.
step7 : Jump to step 10, if no carry.
step8 : calculate the 2's complement of the result if carry is there.
step9 : Increment carry if it is there.
step10 : store the result of subtraction from accumulator to specified memory location.
step11 : store the value of carry from accumulator to memory location.
step12 : Stop the program execution.

④ Flowchart :-


Start
Initialize the carry as zero
Load the 1st 8-bit data from memory to accumulator
Transfer the first 8-bit number to register B
Load 2nd 8-bit number from memory to accumulator
Subtract B from A
check for carry
NO
Yes
calculate the 2's complement of accumulator, if carry is there
Increment carry by 1
Store the result from accumulator to memory location
Move the content of carry into acc.
Store the carry from acc to memory location
End

3) Explain the dynamic memory allocation with new and delete in C++.

⑤ Main program :-

```
MVI   C, 00
LDA   8050
MOV   B, A
LDA   8051
SUB   B
JNC   Loop
CMA
INR   A
INC   C
Loop:
STA   8052
MOV   A, C
STA   8053
HLT
```

⑥ Result :-

Address	opcode
8000	0E
8001	00
8002	3A
8003	50
8004	80
8005	47
8006	3A
8007	51
8008	80
8009	90
800A	D2

Address	opcode
800B	10
800C	80
800D	2F
800E	3C
800F	0C
8010	32
8011	52
8012	80
8013	79
8014	32
8015	53
8016	80
8017	76

08/04/24

final result :-

8050 = FE , 8051 = F0, 8052 = 01, 8053 = 1

⑦ conclusion :-

Hence, successfully subtracted two 8-bit data with consideration of borrow.

Name: Dolindra Bahadur

Q.no. 5 : Write an assembly language program to add two 16-bit data values using DAD instruction.

⇒ Solution:

a. Objective :-

- To write an assembly language program which adds two 16-bit numbers.

b. Requirements :

- 8085 simulator / 8085 Microprocessor kit
- DC Battery of (0-5)V.

c. Algorithm :

step 1: Load both the lower and the higher bits of first number at once.

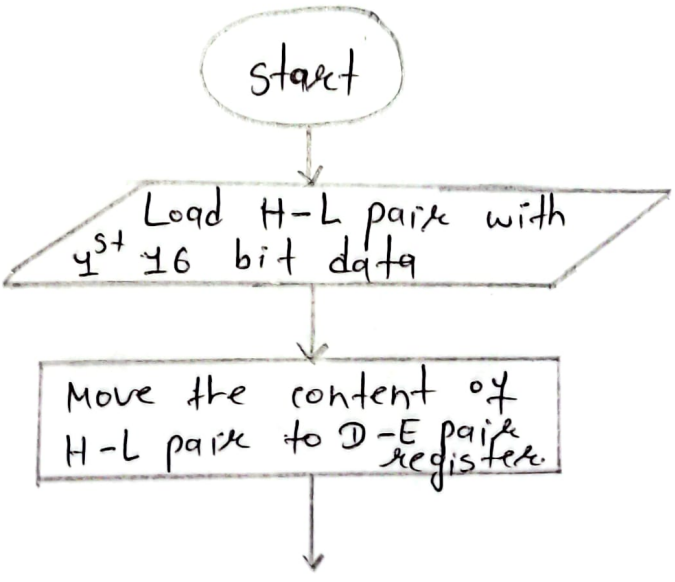
step 2: Move the first data to another register pair.

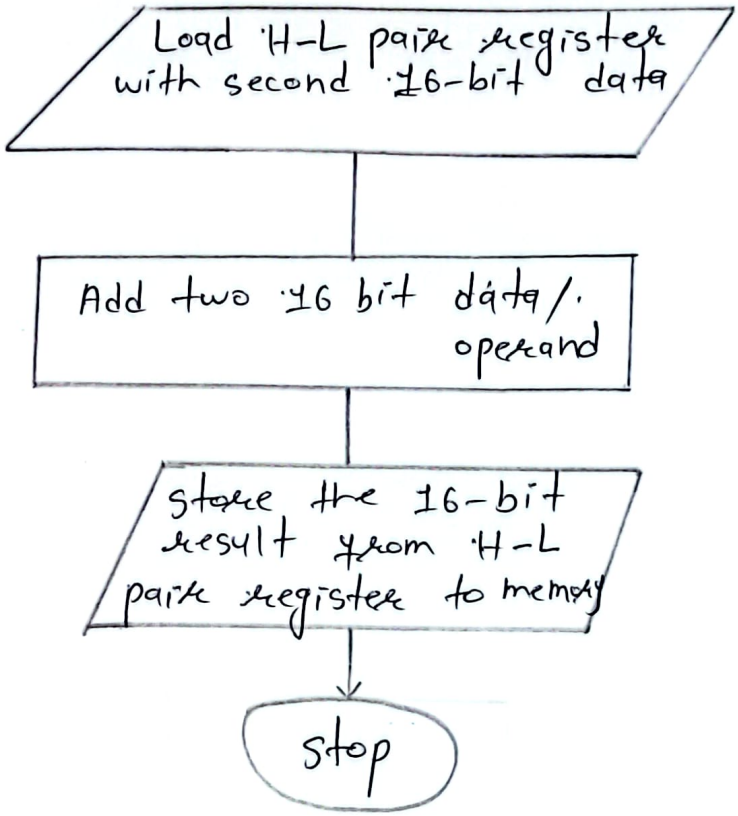
step 3: Load both the lower and the higher bits of second data/number at once.

step 4: Add both the register pairs.

step 5: Store the result in a specific memory location.

d. Flowchart :-





e. program:

```
LHLD 8050
XCHG
LHLD 8052
DAD D
SHLD 8068
HLT
```

f. Result after assembling:

Address	opcode	Address	opcode
8000	2A	8005	52
8001	50	8006	80
8002	80	8007	19
8003	EB	8008	22
8004	2A	8009	68

Address	opcode	Address	opcode
800A	80	8050	02
800B	76	8051	AA
~~800C~~		8052	F1
		8053	2B

Final result: 8068 = F3

8069 = D5

g. conclusion:

∴ Hence, we successfully added two 16-bit numbers using DAD instruction. The result given in the 8068 memory location shows the lower bits and in 8069 memory location shows the higher bits of the result.

08/10/32

Q. 6) write an assembly language program to add two 16 bit data and store the result in memory location along with carry without using DAD instruction.

solution :-

① objective :-

→ To write an assembly language program in 8085 to add two 16 bit data.

② Requirements :-

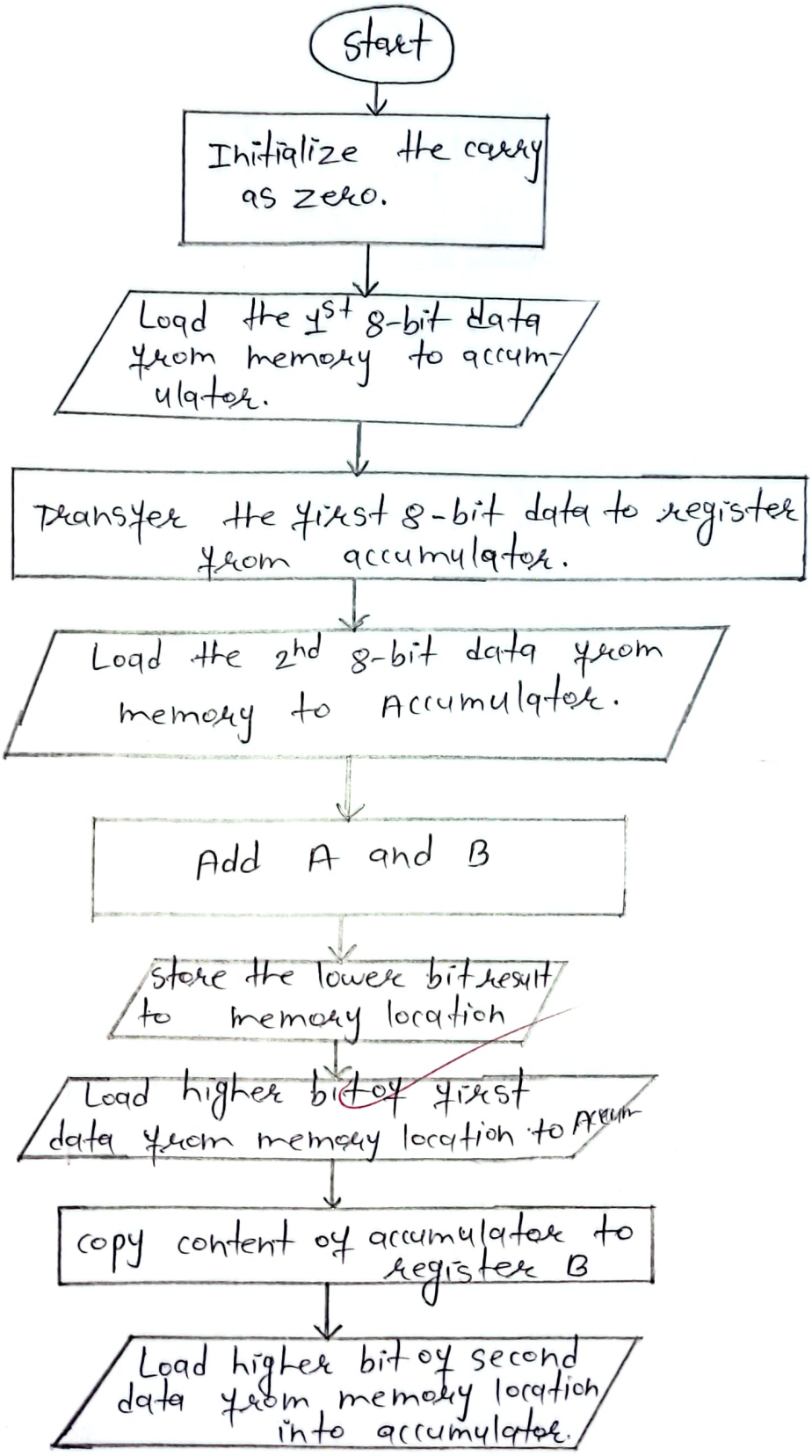
→ 8085 Microprocessor kit / 8085 simulator.
→ (0-5V) DC Battery.

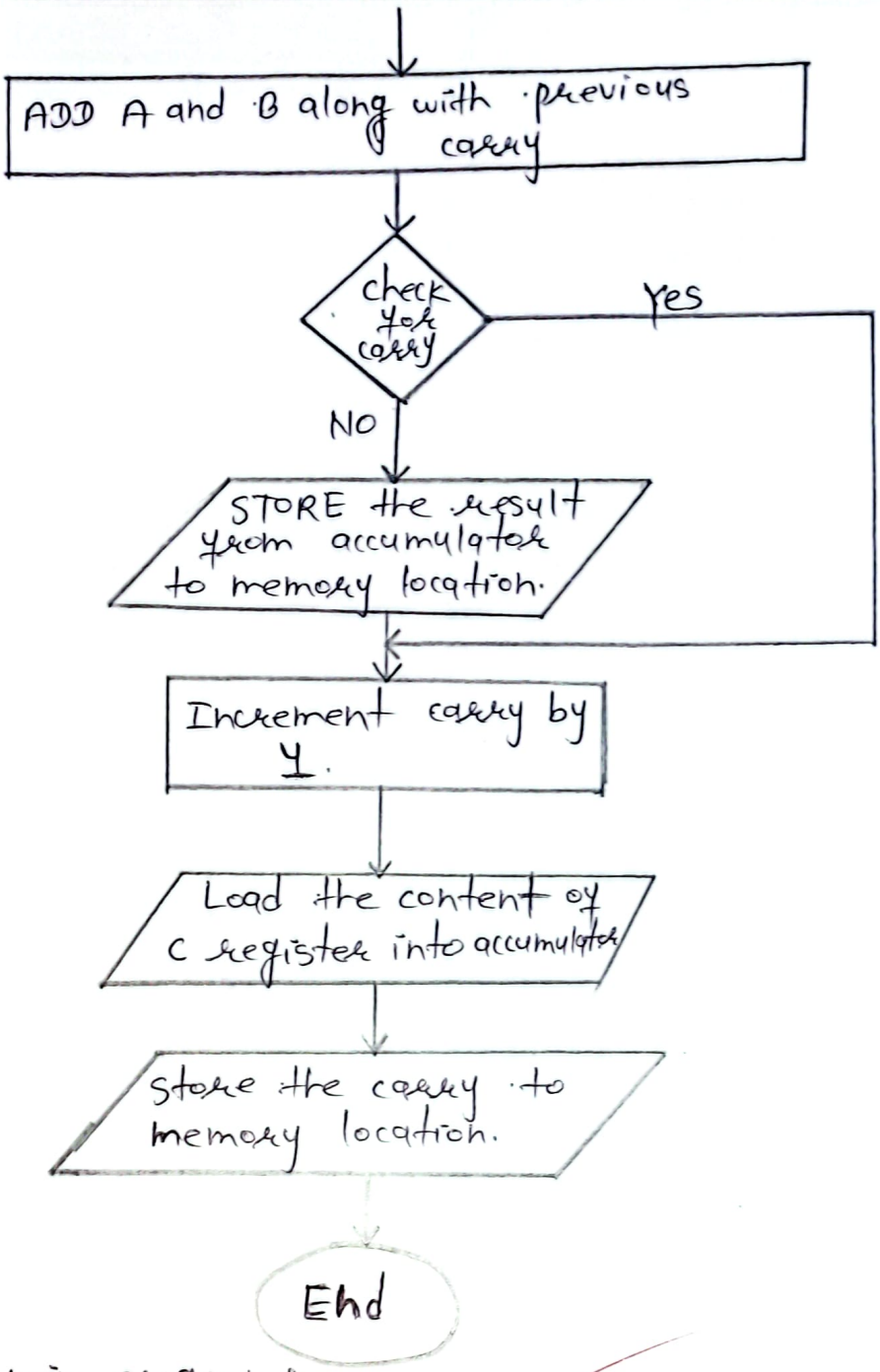
③ Algorithm :-

step 1 : start the Microprocessor.
step 2 : Initialize C register as zero to store carry.
step 3 : Load the first 8-bit data into accumulator from memory.
step 4 : copy the content of accumulator to register B.
step 5 : Load the second 8-bit data into accumulator from memory location.
step 6 : Add the two 8-bit data and check for carry.
step 7 : store the lower bit result to memory location.
step 8 : Load higher bit of first data from memory location into accumulator.
step 9 : copy content of accumulator to register B.
step 10 : Load higher bit of second data from memory location into accumulator.
step 11 : Add the two data along with previous carry.
step 12 : check for carry and store the se result to memory location, if no carry.

step13 : Increment the C register, if there is carry.
step14 : Load the content of C register to accumulator.
step15 : Store the carry to memory location.
step16 : Stop the execution.

(d) Flowchart :-





⑤ Main program:-

```
MVI  C,00
LDA  8059
MOV  B,A
LDA  8064
ADD  B
STA  8063
LDA  8060
MOV  B,A
LDA  8062
ADC  B
JNC  SKIP
INR  C
SKIP:
STA  8064
MOV  A,C
STA  8065
HLT
```

⑥ Result after ~~ass~~ assembling :-

Address	opcode	Address	opcode.
8000	0E	8011	3A
8001	00	8013	80
8002	3A	8014	88
8003	50	8015	80 D2
8004	60	8016	19
8005	47	8017	80
8006	3A	8018	0C
8007	32	8019	32
8008	80	801A	55
8009	80	801B	60
800A	32	801C	79
800B	54	801D	32
800C	80	801E	56
800D	3A	801F	80
800E	51	8020	76
800F	80		
8010	47		

Final result :-

8050 = 22
8051 = 33
8052 = 33
8053 = 22
8054 = 55
8055 = 55
8056 = 00

⑦ Conclusion :-

Hence, The two 16-bit data ~~was~~ is added along with carry. ~~wi~~ The result is also verified and the sum is stored in 8063 and 8064 memory location and carry is in 8065 memory location.

08/10/28

.7) Write an assembly language program to subtract wo 16-bit data values stored in memory location.

olution:-

1) objective:-

→ To subtract write an assembly language program that subtracts the two 16-bit data.

2) Requirements:-

→ 8085 Microprocessor kit/8085 simulator.
→ DC Battery of (0-5V).

3) Algorithm:-

step1: start the 8085 Microprocessor.
step2: Load 1st 8-bit number stored in 8050 and 2nd number stored in 8051 with HL pair register.
step3: Exchange the value of HL pair with DE pair register.
step4: Load 1st 8-bit number stored in 8052 and 2nd number stored in 8053 with HL pair register.
step5: value of 8050 is loaded into accumulator.
step6: Lower byte of 2nd 16-bit number is subtracted
step7: Moving the result of subtraction of lower bytes from accumulator to L register.
step8: Value of 8051 is loaded into accumulator.
step9: Moving the result of subtraction of higher bytes from accumulator to H register.
step10: store the result of subtraction in 8054 location and borrow if any in 8055.
step11: stop the execution.

⑤ Main program :

```
LHLD 8050
XCHG
LHLD 8052
LDA 8050
SUB L
MOV L, A
LDA 8051
SBB H
MOV H, A
SHLD 8054
HLT
```

⑥ Result after assembling :

Address	opcode	Address	opcode
8000	2A	800E	80
8001	50	800F	9C
8002	80	8010	87
8003	EB	8011	22
8004	2A	8012	54
8005	52	8013	80
8006	80	8014	76
8007	3A		
8008	50		
8009	80		
800A	95		
800B	BF		
800C	3A		
800D	51		

Final result :-

8050 = 0A
8051 = 32
8052 = E2
8053 = 04
8054 = 28
8055 = 31

08/01/28

⑦ conclusion :-

∴ Hence, two 16-bit data stored in memory location were subtracted using assembly language programming in 8085 microprocessor.

Q.8) Write an assembly language program for multiplying two 8-bit numbers by using 8085 microprocessor kit / simulator.

Solution :-

① objective :-

→ To write an assembly language program for multiplying two 8-bit numbers.

② Requirements :-

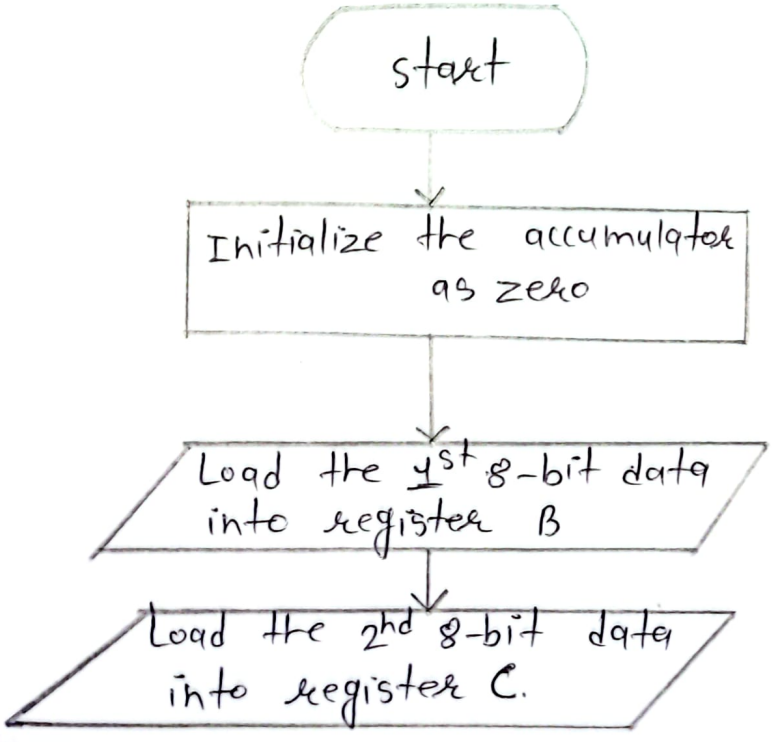
→ 8085 Microprocessor kit / 8085 simulator

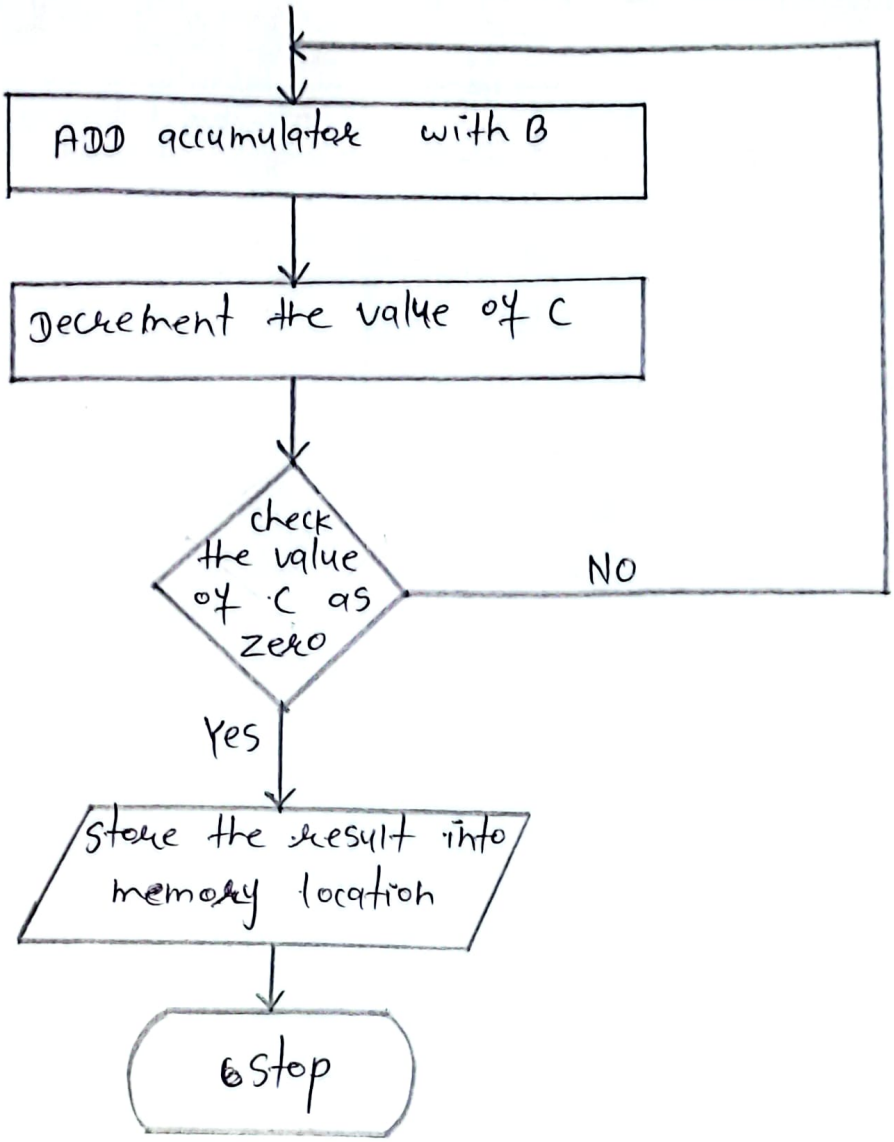
→ (0-5V) DC Battery.

③ Algorithm :-

step1 : Initialize the value of accumulator as zero.
step2 : Load the 1st data into B.
step3 : Load the 2nd data into C.
step4 : Add accumulator with B.
step5 : Decrement the value of C.
step6 : check the value of C, until it is decrement to zero, Go to step 4.
step7 : store the value of accumulator into memory location.
step8 : Halt the program.

④ Flowchart :-





⑤ Main program :-

```
MVI  A,00
MVI  B,08
MVI  C,06
loop:
ADD  B
DCR  C
JNZ  loop
STA  8050
HLT
```

⑥ Result after assembling:-

Address	opcode	Address	opcode
8000	3E	8008	C2
8001	D0	8009	06
8002	06	800A	80
8003	08	800B	32
8004	0E	800C	50
8005	06	800D	80
8006	80	800E	76
8007	0D		

Final result:-

8050 :- 30

⑦ conclusion:-

∴ Hence, two 8-bit numbers were multiplied using assembly language programming and result was ~~was~~ verified.

08/05/28

8.9) Write an assembly language program to perform 8-bit division of given two numbers.

Solution:-

① objective:-

→ To write an assembly language program to perform division of 8-bit given numbers.

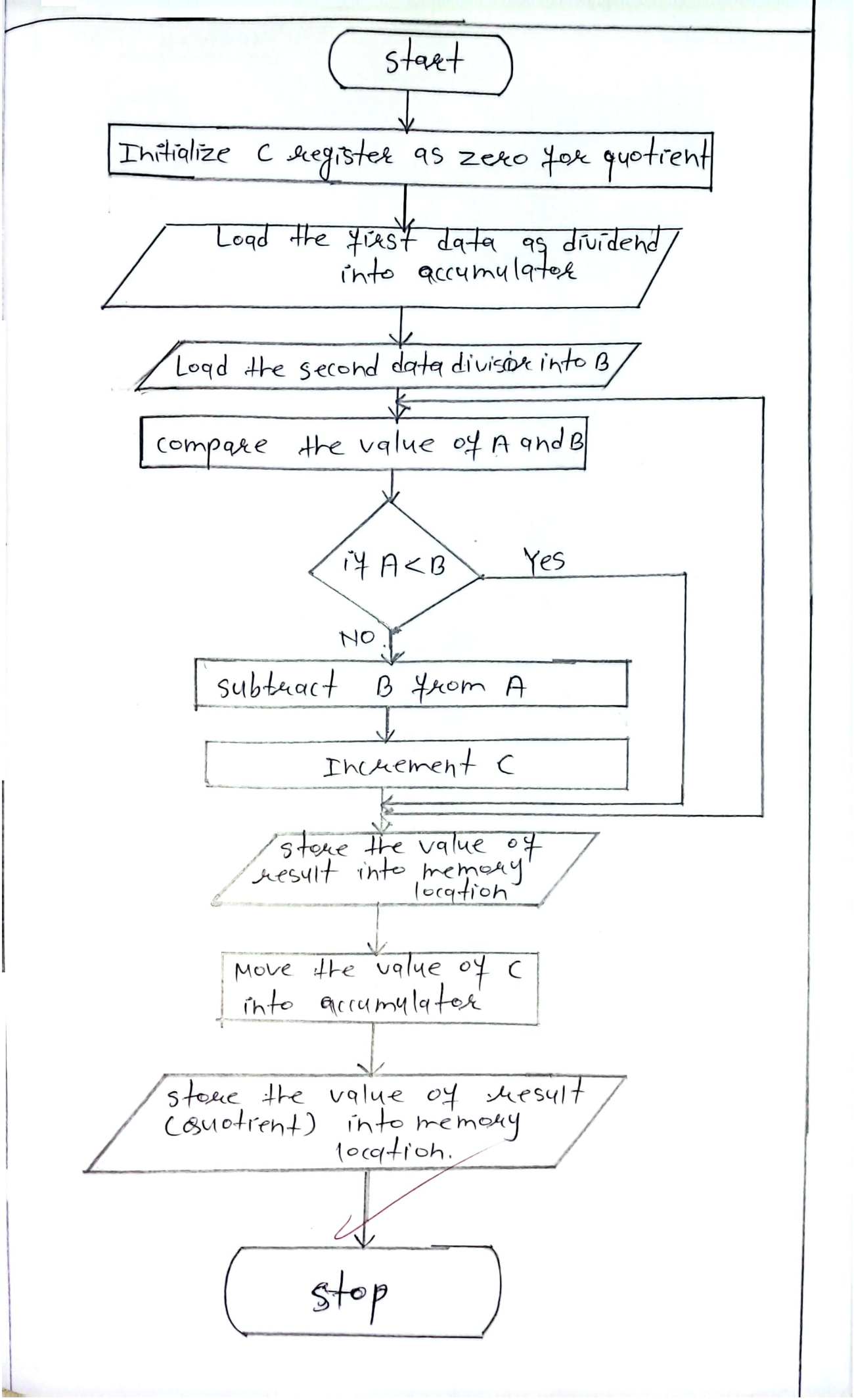
② Requirements:-

→ 8085 Microprocessor kit / 8085 simulator.
→ (0-5V) DC Battery.

③ Algorithm:-

step1: start the microprocessor.
step2: Initialize C ~~register~~ register as zero.
step3: Load the 1st 8-bit data divedent into accumulator.
step4: Load the 2nd 8-bit data divisor into B register.
step5: compare the value of A and B.
step6: If A<B, carry flag is set and move to step 9.
step7: If A>B, subtract B from A.
step8: Increment C, jump to step 5.
step9: store the value of result (Remainder) into memory location.
step10: Move the value of C into accumulator.
step11: store the value of accumulator (quotient) into memory location.
step12: stop the execution.

④ Flowchart:-



⑤ Main program :-

```
MVI C, 00
MVI A, 09
MVI B, 02
Loop 2 :
CMP B
JC Loop1
SUB B
INR C
JMP Loop2
Loop1 :
STA 8050
MOV A, C
STA 8051
HLT
```

⑥ Result after assembling :-

Address	opcode	Address	opcode
8000	0E	800D	06
8001	00	800E	80
8002	3E	800F	32
8003	09	8010	50
8004	06	8011	80
8005	02	8012	79
8006	B8	8013	32
8007	DA	8014	51
8008	0F	8015	80
8009	80	8016	76
800A	90		
800B	0C		
800C	CE		

Final result :- 80.

04/05/23

⑦ conclusion :-

: Hence, The two - 8 bit given numbers were divided i.e. 09 is divided by 02 using 8085 microprocessor with assembly language.

Q.40) Write an assembly language program to find the largest element of an array and store it in memory.

Solution:-

① Objective:-

→ To write a program in assembly language to find the largest element of an array.

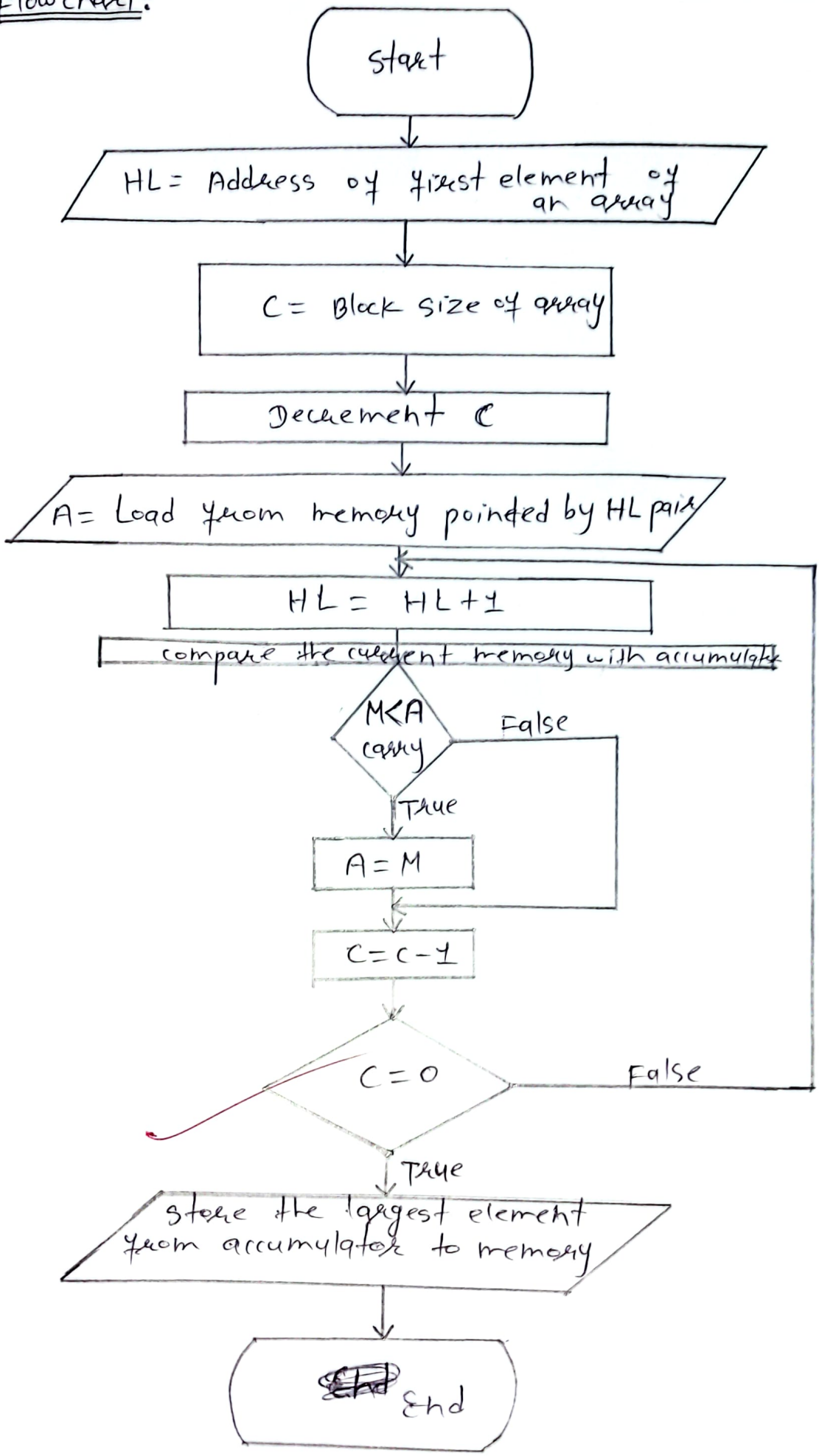
② Requirements:-

i. 8085 Microprocessor kit / 8085 simulator.
ii. (0-5V) DC Battery.

③ Algorithm:-

step1: start the Microprocessor.
step2: store the address of the first element in HL register pair.
step3: store the block size to C.
step4: Decrement C.
step5: Load the first number from memory.
step6: Increment the address of HL pair.
step7: compare the incremented memory location to accumulator.
step8: check for carry.
step9: If carry is there, move the memory to accumulator.
step10: If no carry, decrement C.
step11: check C for zero.
step12: If C is not zero, repeat from step6.
step13: If C is zero, store the largest number from accumulator to memory location.
step14: stop the execution.

(1) Flowchart :



⑤ Main program :

```
LXI H, 8050
MVI C, 05
DCR C
MOV A, M
REPEAT : INX H
CMP M
JNC SKIP
MOV A, M
SKIP : DCR C
JNZ REPEAT
STA 8060
HLT
```

⑥ Result after assembling :

Address	opcode	Address	opcode
8000	21	8010	80
8001	50	8011	32
8002	80	8012	60
8003	0E	8013	80
8004	05	8014	76
8005	0D	8015	91
8006	7E	8016	76
8007	23	80	
8008	BE	Input :	
8009	D2	8050	01
800A	0D	8051	03
800B	80	8052	02
800C	7E	8053	05
800D	0D	8054	04
800E	C2	805	
800F	07		

Final result :

8060 = 5

⑦ Conclusion :

: Hence, we have completed the program to find the largest number in an array and the result is also verified.

08/05/23

8.11) Write a program to find smallest element of an array and store it into memory location.

Solution :-

① Objective :-

→ To write an assembly language program to find the smallest element of an array and store it into specific memory location.

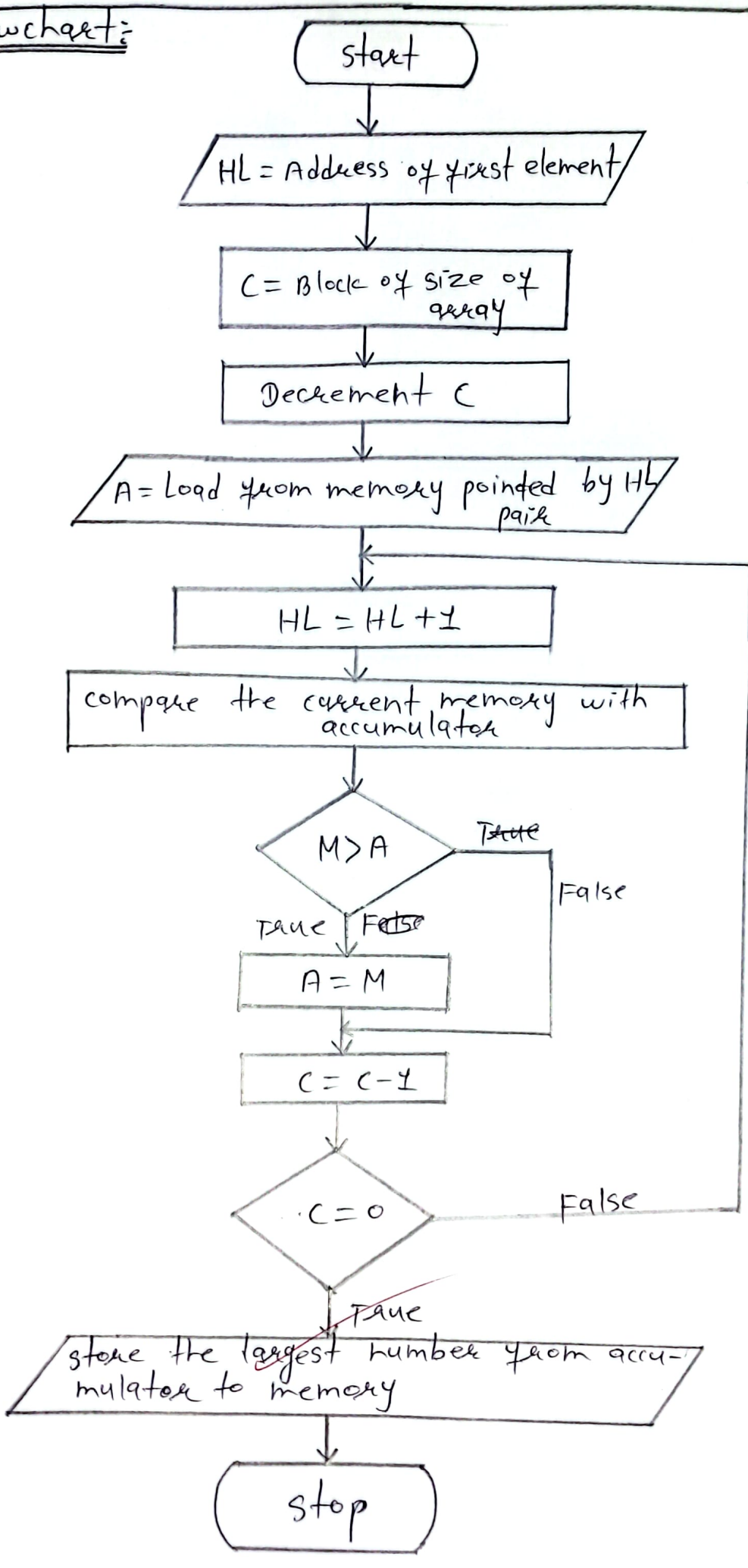
② Requirements :-

→ 8085 Microprocessor kit / 8085 simulator.
→ (0-5V) DC Battery.

③ Algorithm :-

step1 : start the microprocessor.
step2 : store the address of first element to HL register pair.
step3 : store the block size to C.
step4 : Decrement C.
step5 : Load the first number from memory.
step6 : Increment the address of HL pair.
step7 : Compare the incremented memory location to accumulator.
step8 : check for carry.
step9 : If carry is not there, move the memory to accumulator.
step10 : If carry is there, decrement C.
step11 : check C for zero.
step12 : If C is not zero repeat from step 6.
step13 : If C is zero, store the smallest number from accumulator to memory.
step14 : stop the execution.

(4) Flowchart:



⑤ Main program :

```
LXI H, 8050
MVI C, 05
DCR C
MOV A, M
REPEAT : INX H
CMP M
~~JNZ~~ JC SKIP
MOV A, M
SKIP : DCR C
JNZ REPEAT
STA 8055
HLT
```

⑥ Result after assembling :

Address	opcode	Address	opcode
8000	21	800D	0D
8001	50	800E	C2
8002	80	800F	07
8003	0E	8010	80
8004	05	8011	32
8005	0D	8012	55
8006	FE	8013	80
8007	23	8014	76
8008	BE	Input:	
8009	DA	8050	02
800A	0A	8051	10
800B	80	8052	05
800C	7E	8053	06
		8054	04

Final result:

8055 = 02

08/05/28

⑦ Conclusion :

Hence, we the smallest element of an array of size 5 ~~was~~ is stored at the memory location 8060. so, smallest element is found and result is also verified.

Q.42) Write a program in assembly language to short given 'n' number in an array in ascending order.

Solution :-

① Objective :-

→ To write a program (in assembly language) to short given 'n' number in an array.

② Requirements :-

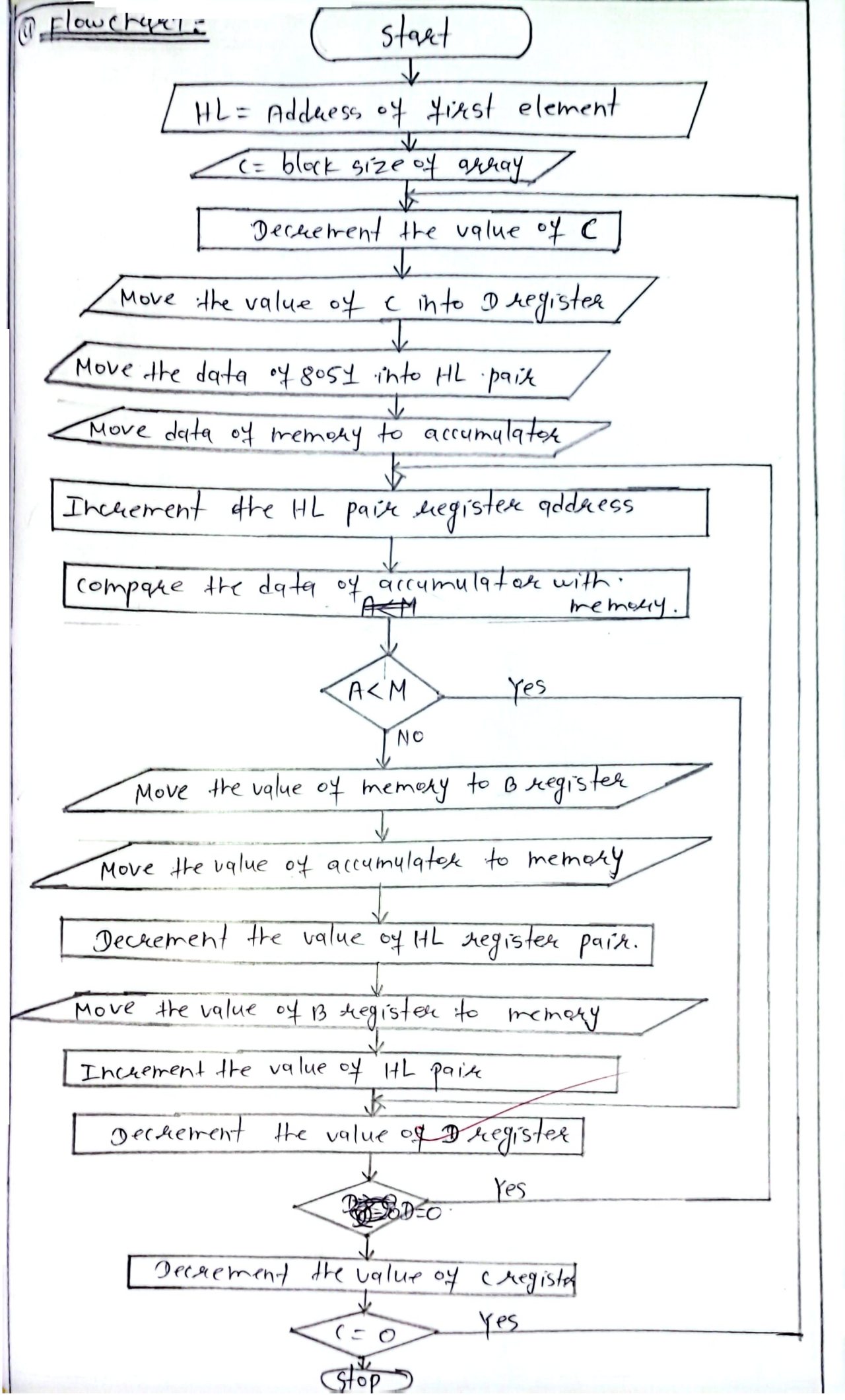
→ 8085 Microprocessor kit/ simulator.

→ DC Battery of (0-5V).

③ Algorithm :-

step1 : Start the Microprocessor.
step2 : Store the address of first element of HL pair.
step3 : store the block size to register C.
step4 : Decrement C.
step5 : Move the value of c into D register.
~~step6 : Move data of memory to accumulator.~~
step6 : Move data the of 8051 to HL pair.
step7 : Move data of memory to accumulator.
step8 : Increment the HL pair register address.
step9 : compare the data of accumulator with memory.
step10 : check for carry, if carry is there then go to step-16.
step11 : Move the value of memory to B register.
step12 : Move the value of accumulator to memory.
step13 : Decrement the value of HL register pair.
step14 : Move the value of B register to ~~memory~~ memory.
step15 : Increment the value of HL pair.
step16 : Decrement the value of D register.
step17 : if D is not zero then go to step-8.
step18 : Decrement the value of C.
step19 : if the value of C is not zero, go to step-4.
step20 : stop.

(1) Flowchart:



⑤ Main program :

```
LXI H, 8050
MOV C, M
DCR C
REPEAT :
MOV D, C
LXI H, 8051
Loop :
MOV A, M
INX H
CMP M
JC SKIP
MOV B, M
MOV M, A
DCX H
MOV M, B
INX H
SKIP :
DCR D
JNZ Loop
DCR
JNZ REPEAT
HLT
```

⑥ Result after assembling :

Address	opcode	Address	opcode
8000	21	800D	10
8001	50	800E	80
8002	80	800F	46
8003	4C	8010	77
8004	0D	8011	2B
8005	4A	8012	70
8006	21	8013	23
8007	51	8014	15
8008	80	8015	C2
8009	7E	8016	09
800A	23	8017	80
800B	BE	8018	0D
800C	DA	8019	C2
		801A	05
		801B	80
		801C	76

~~Input result~~:

Input:

8050 = 3
8051 = 5
8052 = 7
8053 = 2
8054 = 6

output:

8050 = 2
8051 = 3
8052 = 5
8053 = 6
8054 = 7

(7) Conclusion:

: Hence, we have completed the program to short given 'n' number in ascending order array using assembly language in 8085 Microprocessor. we also verified the result. Here, we shorted the element in ascending order that is from ~~low to~~ small to large.

08/01/28

8.13) Write an assembly language program to short given 'n' numbers in an array in descending order.

Solution:-

① objective:-

→ To write an assembly language program to short given 'n' numbers in an array in descending order.

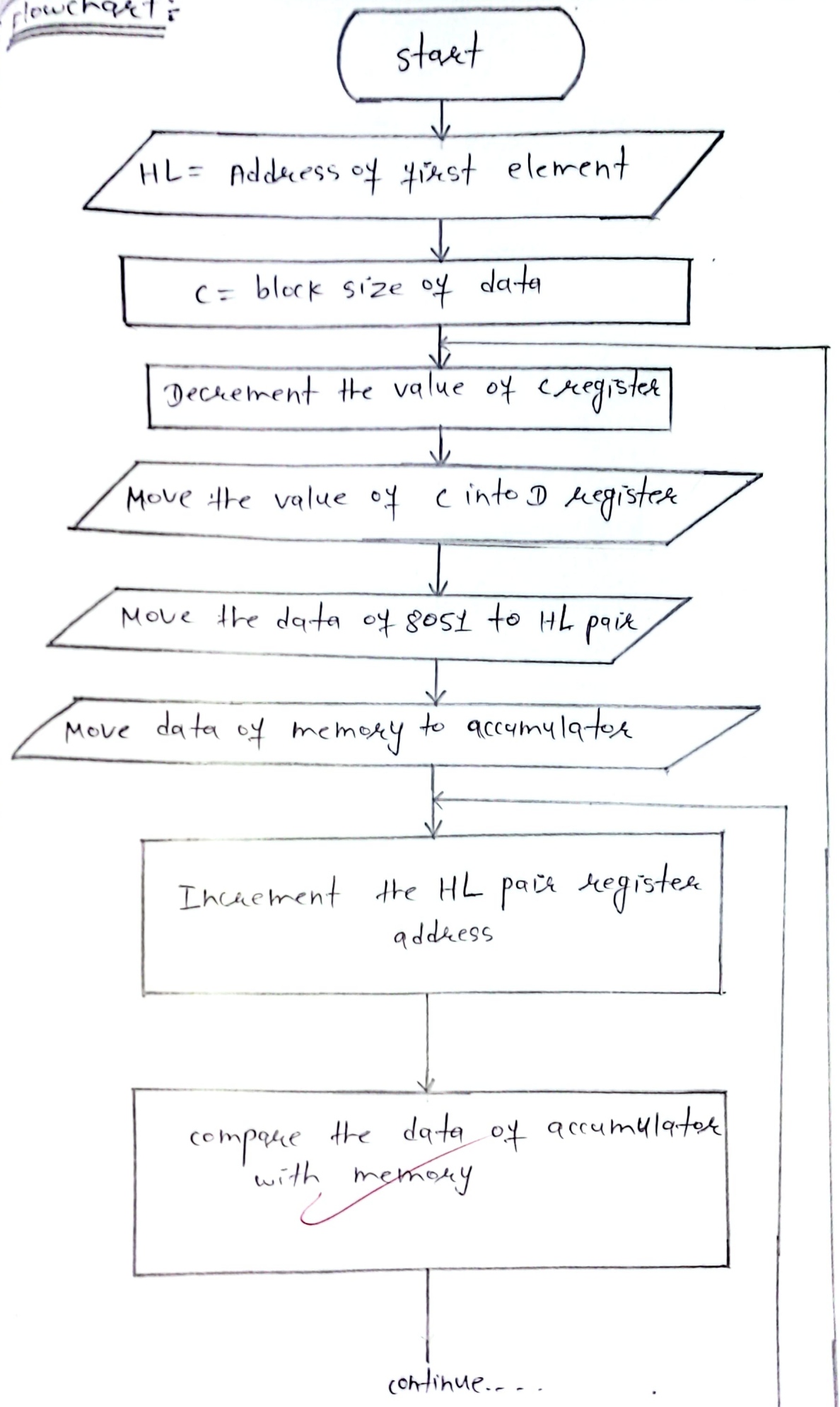
② Requirements:-

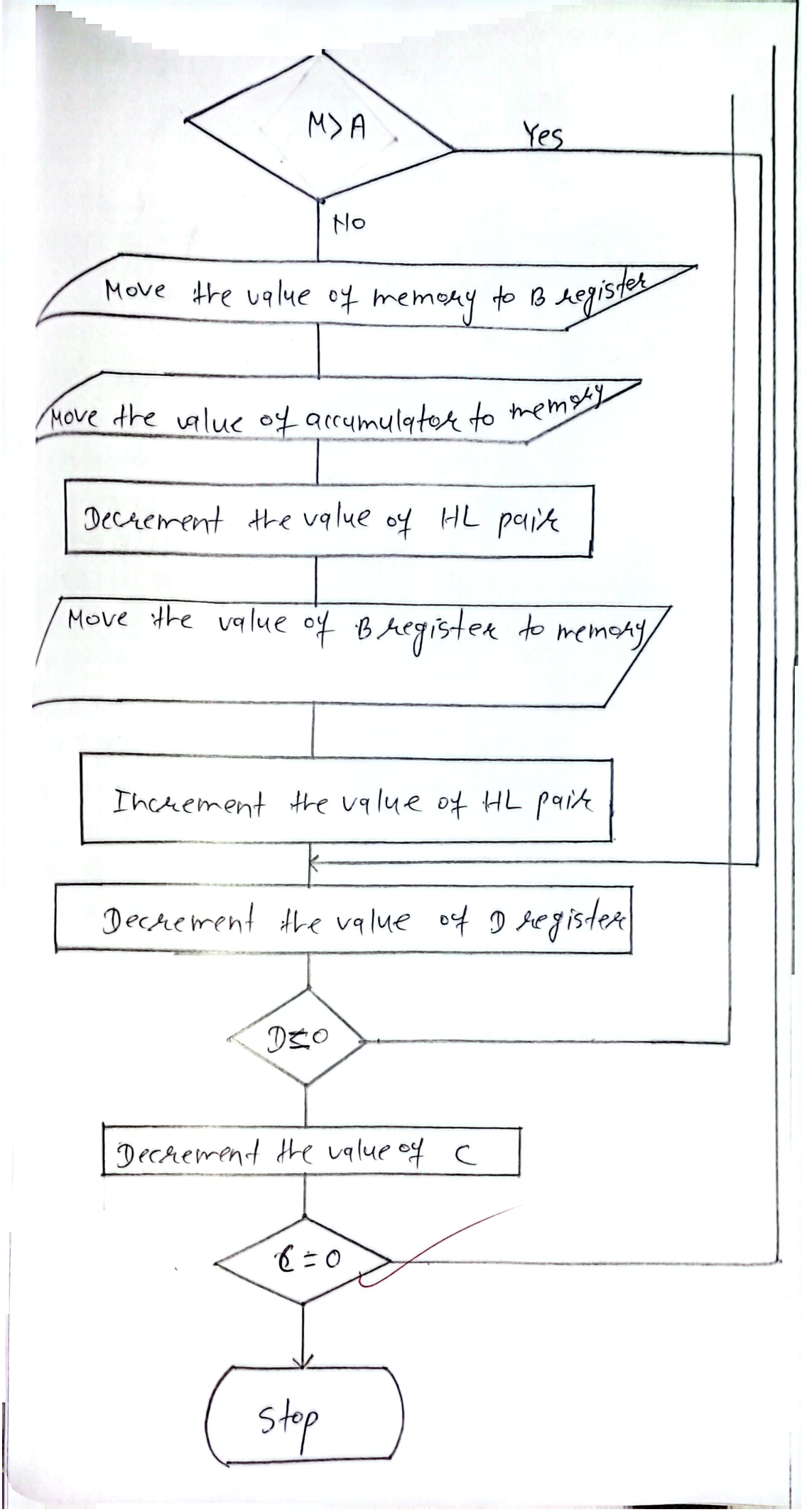
i. 8085 Microprocessor kit/ 8085 simulator.
ii. (0-5V) DC Battery.

③ Algorithm:-

step1 : start the Microprocessor.
step2 : store the address of first element to HL pair register.
step3 : store the block of size to c.
step4 : Decrement c.
step5 : Move the content of c into D register.
step6 : Move data of ~~memory~~ 8051 to ~~accumulator~~ HL pair.
step7 : Move data of memory to accumulator.
step8 : Increment the HL pair register address.
step9 : Compare the data of accumulator with memory.
step10 : check for carry, if carry is not there, then go to step-16.
step11 : Move the value of memory to B register.
step12 : Move the value of accumulator to memory.
step13 : Decrement the value of HL register pair.
step14 : Move the value of B register to memory.
step15 : Increment the value of HL register pair.
step16 : Decrement the value of D register.
step17 : if D is not zero, then go to step 8 & 4.
step 20 : stop.

Flowchart :-





⑤ Main program:

```
LXI H, 8050
MOV C, M
DCR C
REPEAT :
 MOV D, C
 LXI H, 8051
 Loop:
 MOV A, M
 INX H
 CMP M
JNC SKIP
MOV B, M
MOV M, A
DCX H
MOV M, B
 INX H
SKIP:
DCR D
JNZ Loop
 DCR C
JNZ REPEAT
 HLT
```

⑥ Result after assembling:-

Address	opcode	Address	opcode
8000	21	800E	80
8001	50	800F	46
8002	80	8010	77
8003	4E	8011	28
8004	0D	8012	70
8005	4A	8013	23
8006	21	8014	15
8007	50	8015	C2
8008	80	8016	09
8009	7E	8017	80
800A	23	8018	0D
800B	BE	8019	C2
800C	D2	801A	05
800D	10	801B	80
		801C	76

~~Final result:~~ Input:

8050 = 2
8051 = 3
8052 = 5
8053 = 2
8054 = 1

Final result:

8050 = 5
8051 = 3
8052 = 2
8053 = 2
8054 = 1

⑧ conclusion:

: Hence, given n numbers in an array were sorted in descending order using ~~using~~ 8085 Microprocessor.

6.14) write an ALP to generate ~~0~~ 10 odd numbers.

=> solution:

① objective:

→ To write an assembly language to generate ten odd numbers.

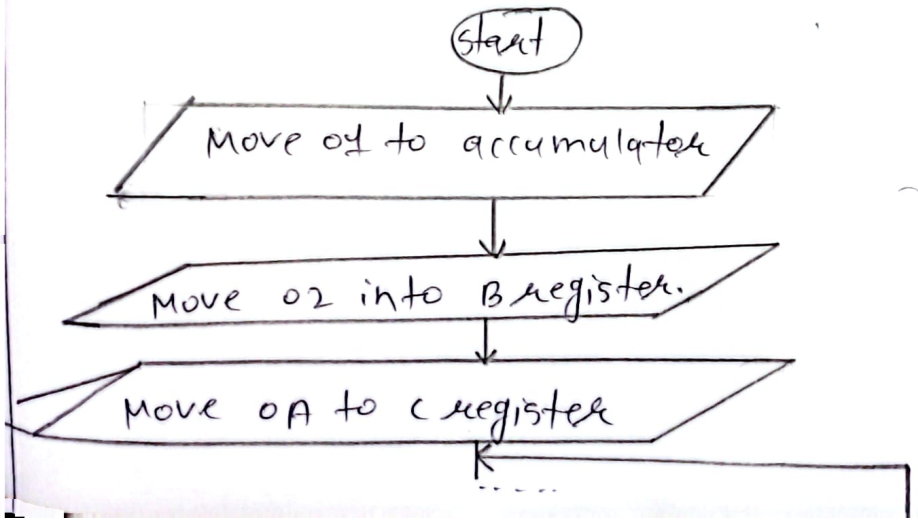
② Requirements:

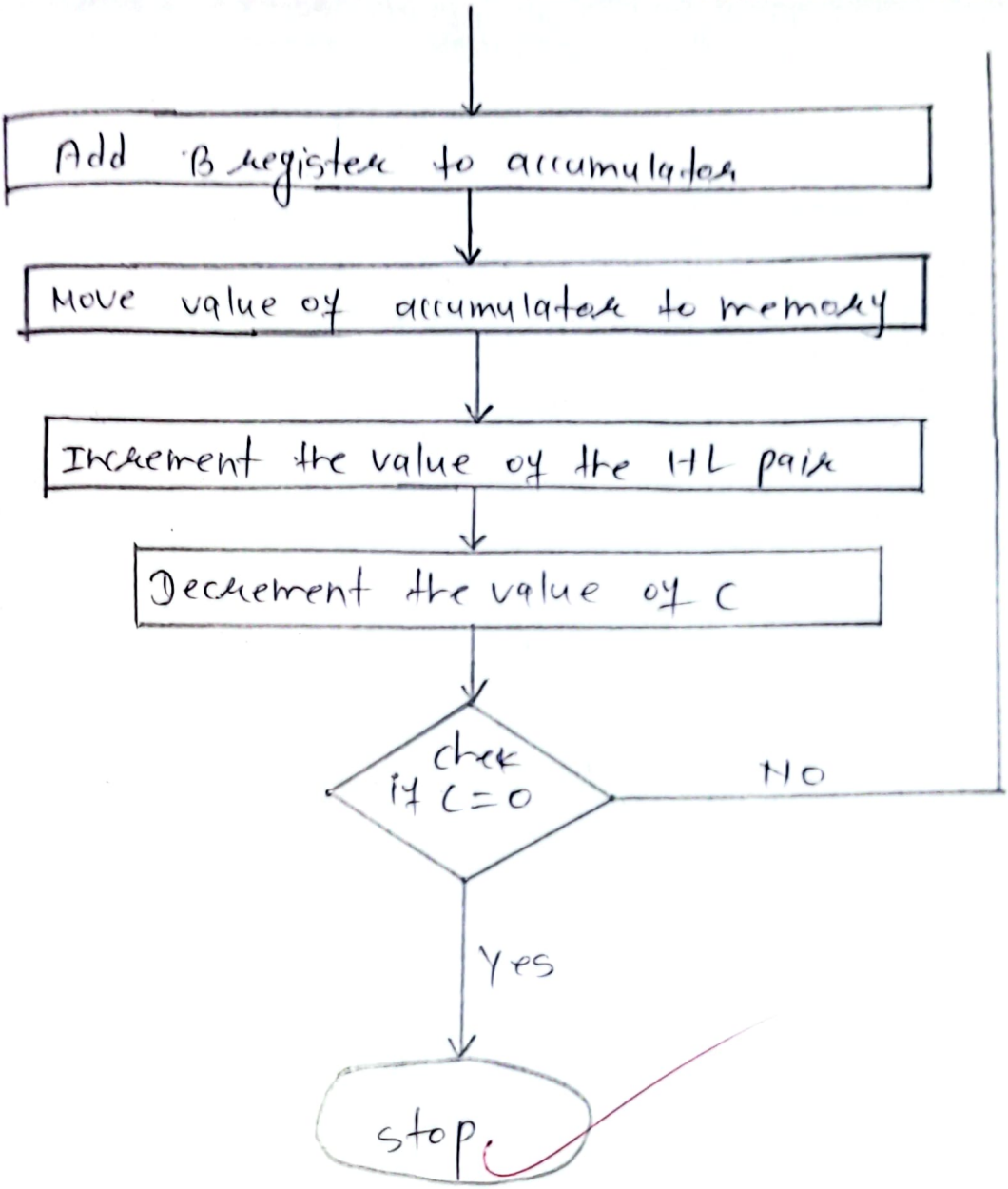
i. 8085 Microprocessor kit/simulator
ii. (0-5V) DC Battery.

③ Algorithm:

step-1: start
step-2: Move 01 to accumulator.
step-3: Move 02 to register B.
step-4: Move 0A to C register.
step-5: Move data of 8050 into HL pair register.
step-6: Add B register to accumulator.
step-7: Move value of accumulator to memory.
step-8: Increment the value of HL pair.
step-9: Decrement the value of C register.
step-10: if C is not zero; go to step-6.
step-11: stop

④ Flowchart:





⑤ Main program:

```
MVI A,05
MVI B,02
MVI C,0A
LXI H,8050
Loop:
MOV M,A
ADD B
INX H
DCR C
JNZ Loop
HLT
```

⑥ Result after assembling:-

Address	opcode	Address	opcode
8000	3E	8008	80
8001	01	8009	77
8002	06	800A	80
8003	02	800B	23
8004	0E	800C	0D
8005	0A	800D	C2
8006	21	800E	09
8007	50	800F	80
		8010	76

Final result:

8050 = 01
8051 = 03
8052 = 05
8053 = 07
8054 = 09
8055 = 0B
8056 = 0D
8057 = 0F
8058 = 11
8059 = 13

08/12/23

⑦ conclusion:

Hence, 10 odd numbers were generated using 8085 Microprocessor.

Q.15) Write an ALP to generate 10 even numbers.

solution:

① objective:

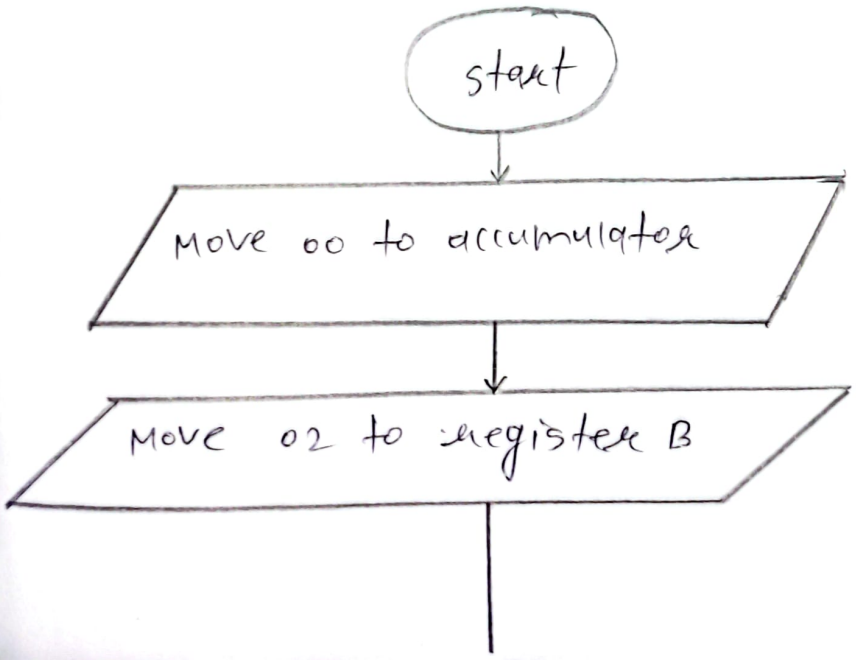
→ To generate 10 even numbers in assembly language.

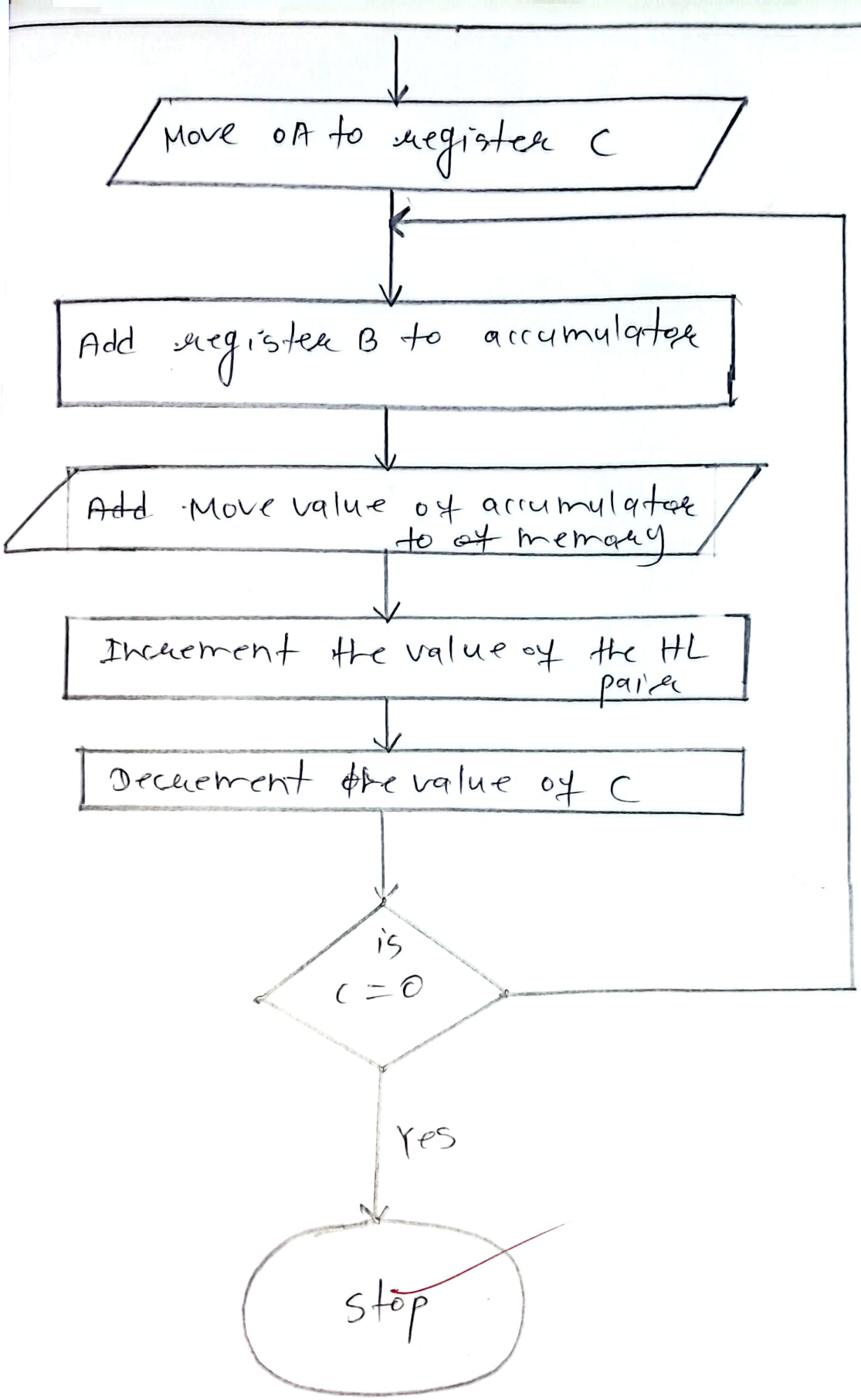
② Requirements:- i. 8085 Microprocessor kit
ii. 0-5V DC Battery

③ Algorithm:

step1 : start
step2 : Move 00 to accumulator
step3 : Move 02 to B register
step4 : Move 0A to C register.
step5 : Move data of 8050 to HL pair
step6 : Add B register to accumulator.
step7 : Move value of accumulator to memory.
step8 : Increment the value of HL pair.
step9 : Decrement the value of C register.
step10 : If C isnot zero, go to step-6.
step-11 : HALT

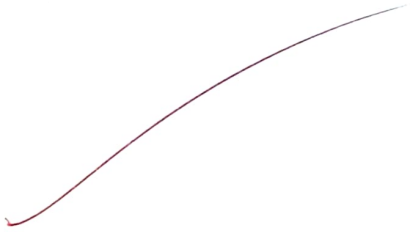
④ Flowchart:




Move 0A to register C
Add register B to accumulator
Add Move value of accumulator to of memory
Increment the value of the HL pair
Decrement the value of C
is C=0
Yes
stop

(5) Main program:

```
MVI   A,00
MVI   B,02
MVI   C,0A
LXI   H,8050
Loop:
MOV   M,A
ADD   B
INX   H
DCR   C
JNZ   Loop
HLT
```



Result after assembling :-

Address	opcode	Address	opcode
8000	3E	800A	80
8001	00	800B	23
8002	06	800C	0D
8003	02	800D	C2
8004	0E	800E	09
8005	0A	800F	80
8006	21	~~8010~~	
8007	50	8010	76
8008	80		
8009	77		

final result:

8050 = 00
8051 = 02
8052 = 04
8053 = 06
8054 = 08
8055 = 0A
8056 = 0C
8057 = 0E
8058 = 10
8059 = 12

⑦ conclusion:

Hence, 10 even numbers were generated using 8085 microprocessor.

08/05/23