Index

S. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
1	Addition of 2 16bit number	2	2/6/2021		
2	Subtraction of 2 16 bit number	4	2/6/2021		
3	Multiplication of 2 16 bit number	6	7/6/2021		
4	Division of 2 16bit number.	7	7/6/21		
5	Sum of elements of an array	8	9/6/21		
6	larger of 2 bit numbers	10	14/6/21		
7	ADD two BCD data	11	14/6/21		
8	Subtract two BCD data	13	14/6/21		
9	Search for largest data	15	20/6/2021		
10	Convert BCD to binary	17	25/6/21		
11	Binary to BCD Conversion	19	25/6/21		
12	Search for a given data	20	5/7/2021		
13	2's Complement	22	5/7/2021		
14	Seperation of +ve number	25	14/7/2021		

Index

S. No.	Name of the Experiment	Page No.	Date of Experiment	Date of Submission	Remarks
15	Sepration of even number	26	2[illegible]/7/2021		
16	Sorting on array in ascending order	28	26/7/21		
17	Sorting in descending order	30	29/7/21		
18	Implementation of expression Y=AB+CD	32	8/9/21		
19	Implementation of expression Y=(A+B)(C+D)	34	8/9/21		
			8		
20	Moment of string	36	8/9/21		
21	Filling array with given data	38	9/9/21		

Algorithm

1. Set SI - register as pointer for data.
2. Get the first data in Ax - register
3. Get the second data in Bx - register.
4. ~~Clear~~. Clear CL - register.
5. Get the Sum in Ax - register.
6. Store the Sum in memory.
7. Check for Carry. If Carry flag is set then go to next step otherwise go to step 9.
8. Increment CL - register.
9. Store the Carry in Memory
10. Stop.

Date

Expt. No. 9/6/2021

Page No. 2

Program - 1

Addition of two 16-bit numbers.

AIM: Write an assembly language program to add the two numbers of 16-bit data.

Address	label	Mnemonics	Comments
0100		MOV SI, 1100_H	Set SI - register as pointer for data.
0103		MOV Ax, [SI]	Get the first data in Ax - register.
0105		MOV Bx, [SI+2]	Get the Second data in Bx - register.
0108		MOV CL, 00_H	Clear the CL-register for Carry
010A		ADD Ax, Bx	Add the two data Sum will be in Ax-register.
010C		MOV [SI+4], Ax	Store the Sum in memory location
010F		JNC AHEAD	Check the status of Carry flag if Carry flag is not set then go the specified step.
0111		INC CL	Increment the CL-register.
0113	AHEAD	MOV [SI+6], CL	Move the Carry in Memory location
0116		HLT	End.

Teacher's Signature

Input (with Carry)

Memory address	Content
1100	b2
1101	23
1102	15
1103	f2

Output

Memory address	Content
1104	C7
1105	15
1106	01
1107	00

Input (without Carry)

Memory Address	Content
1100	32
1101	34
1102	56
1103	45

Memory address	Content
1104	88
1105	79
1106	00
1107	00

Expt. No. _____ Date _____

Page No. 3

Result: Executed the program for addition of two 16 bit number and the result is verified.

Teacher's Signature _____

Algorithm

1. Set SI-register as pointer for data.
2. Get the minuend in AX-register.
3. Get the subtrahend in BX-register.
4. Clear CL-register to account for sign
5. Subtract the content of BX from AX, the difference will be in AX
6. Check for Carry. If Carry flag is set then go to next step otherwise go to step 9
7. Increment CL-register by one
8. Take 2's Complement of the difference in AX-register. (For this Complement AX and add one)
9. Store the magnitude of difference in memory.
10. Store the sign bit in memory.
11. Stop.

Date

Expt. No. 2/6/2021 Page No. 4

Program-2

Subtraction of two 16-bit numbers.

AIM : Write an assembly language program to Subtract two numbers of 16-bit data.

Addresses	label	Mnemonics	Comments.
0100		MOV SI, 1100H	load the address of data in SI - register.
0103		MOV AX, [SI]	Get the minuend in Ax - register.
0105		MOV BX, [SI+2]	Get the subtrahend in Bx - register.
0108		MOV CL, ~~BX~~00H	Clear CL-register.
010A		SUB AX, BX	Get the difference in Ax register
010C		JNC ~~store~~ STORE	Check the status of Carry flag if it is not set then go to specified step.
010E		INC CL	Increment CL register
0110		NOT AX	then take 2's Complement of the difference
0112		ADD AX, 0001H	
0115	STORE	MOV [SI+4], AX	Store difference in memory location

Teacher's Signature

Input (without Carry)

Memory address	Content
1100	0E
1101	F2
1102	1C
1103	93

Output

Memory Address	Content
1104	F2
1105	CE
1106	00
1107	00.

Date

Expt. No.

Page No. 5

		0118	MOV [SI+6], CL	Store Sign bit in memory location
		011B	HLT	End

Result: Executed the program for subtraction of two 16 bit number the result is verified.

Teacher's Signature

Algorithm

1. Load the address of data in SI - register.
2. Get the first data in Ax - register.
3. Get the second data in Bx - register.
4. Multiply the content of Ax and Bx. The product will be in Ax and Dx
5. Save The product (Ax and Dx) in memory
6. Stop

Input

Memory address	Content
1100	02
1101	0L
110L	03
1103	08

output

Memory addrs	Content
1104	06
1105	0C
1106	06
1107	00.

Date

Expt. No. 7/6/2021 Page No. 6

Program - 3

Multiplication of two 16-bit numbers.

AIM: Write an assembly language program to multiply two 16-bit numbers?

Addresses	Label	Memonics.	Comments.
0100		MOV SI, 1100_H	load the address of data in SI register.
0103		MOV Ax, [SI]	Mov the first data in Ax-register.
0105		MOV Bx, [SI+2]	Move the Second data in Bx-register.
0108		MUL Bx	Multiply Ax and Bx
010A		Mov [SI+4], Ax	Save the lower 16-bit of Product in memory
010D		Mov [SI+6], Dx	Save the Upper 16-bit of product in memory.
0110		HLT	End.

Result: Executed the program for multiplication of two 16 bit numbers and the result is verified.

Teacher's Signature

Algorithm

1. get the dividend in Ax - register.
2. get the divisor in Dx - register.
3. Divide the content of AX by Bx
4. Transfer the quotient to the memory location pointer by SI + 4
5. Transfer the remainder to the memory location pointed by SI + ~~by~~ 6
6. STOP

~~output~~ Input

Memory address	Content
1100	0C
1101	00
1102	02
1103	02

output

Memory Address	Content
1104	05
1105	00
1106	02
1107	00

Date

Expt. No. 7/6/2021

Page No. 7

Program 4

Division of two - 16-bit data.

AIM : Write a assembly language program to divide two -16 bit data.

Addresses	label	Mnemonics	Comments
0100		MOV SI, 1100H	Load the address of data to SI, register.
0103		MOV AX[SI+2]	Get the dividend in Ax-register.
0105		MOV Bx, [SI+2]	Get the divisor in Bx - register.
0108		DIV BX	Divide the content of Ax by Bx
0104		MOV[SI+4] AX	Transfer the Quotient to the memory location pointed by SI+4
010D		MOV [SI+6], Dx	Transfer the remainder to the memory location pointed by SI+6
0110		HLT	End.

Result: Executed the program for division of two bit number and the result is verified.

Teacher's Signature

Algorithm

1. Load the address of the array in the SI-register.
2. Load the address of the result in the DI-register.
3. Load the Count Value in CL-register.
4. Initialize the Count of Ax be Sum and keep initial Sum as Zero.
5. Add a byte of array to Sum
6. Check for Carry if Carry flag is set then go to next step otherwise go to step 8.
7. Increment the array pointer.
8. Decrement the Count (CL-register)
9. If Count (CL) is Zero go to next step, otherwise go to Step 5
10. Store the 16-bit Sum in memory.
11. Stop.

Date

Expt. No. 9/6/2021

Page No. 8

Program - 5

Sum of elements of an array

AIM: Write an assembly language program to determine the Sum of elements in an array

Address	label	Mnemonics	Comments
0100		MOV SI, 1100_H	Set SI-register as pointer for array.
0103		MOV DI, 1200_H	set DI-register as pointer for array
0106		MOV CL, [SI]	set CL as Count for number of bytes in array
0108		INC SI	Set SI to point to I^{st} byte of array.
0109		MOV AX, 0000H	set initial Sum as Zero
010C	AGAIN	ADD AL, [SI]	Add a byte of array to Sum
010E		JNC AHEAD	Check for Carry flag.
0110		INC AH	if Carry flag is set then Increment AH

Teacher's Signature

Input

Memory Address	Content
1100	05 (Count)
1101	2
1102	4
1103	5
1104	7
1105	3

output

Memory Address	Content
1200	15
1201	00
1202	00
1203	00.

Expt. No. _____ Date _____

Page No. 9

0112	AHEAD	INC SI	increment array. flag.
0113		LOOP AGAIN	Repeat addition. until Count is Zero.
0115		MOV [DI], AX	Store Sum in memory.
0117		HLT	End.

Result : Executed the program. For Sum of elements in an array and the result is verified.

Teacher's Signature _____

Algorithm

1. Move the first number to register AL
2. Move the second number to register BL
3. Compare the contents of register AL and BL. If not Carry go to step 5 else go to step 4
4. Move the Content of register BL to the register AL
5. Move the Content of register AL to the memory location

Input

Memory data Address	Content
1100	06
1101	08

Output

Memory Address	Content
1200	08
1201	00

Date

Expt. No. 9/6/2021 Page No. 10

Program - 6

Larger of two 8-bit numbers

AIM: Write a assembly language program to find the larger of two 8bit numbers.

Address	label	Mnemonics	Comments
0100		MOV SI, 1100H	Set SI as pointer for data.
0103		MOV DI, 1200H	Set DI to point the result
0106		MOV AL, [SI]	Get the first data in register AL
0108		MOV BL, [SI+1]	Get the second data in register BL
010B		CMP AL, BL	Compare two datas
010D		JNC AHEAD	if the data in reg AL is larger the go to the specified location
010F		MOV AL, BL	Transfer the Content of reg BL to Reg AL
0111	AHEAD	MOV [DI], AL	Display the Content of reg. AL to the location pointed by DI
0113		HLT	End.

Result: Execute the program for largest of two 8-bit number and the Result is verified

Teacher's Signature

Algorithm

1. load the address of data in SI-register
2. Clear CL- register to account for Carry.
3. load the first data in Ax-register and second data in Bx register
4. perform binary addition of two byte of data to get the Sum in AL-register
5. Adjust the sum of lows bytes to BCD
6. Save the Sum of lows bytes in DL-register.
7. Get the high byte of first Data in AL-register.
8. Add the high byte of Second data in previous Carry to. AL-register. Now the sum of high bytes will be in AL-register.
9. Adjust the Sum of high bytes to BCD.
10. Save the Sum of high bytes to DH-register.
11. Check for Carry. If Carry flag is set then go to next step, otherwise go to step 13
12. Increment CL-register
13. Save the Sum (Dx-register) in memory.
14. Save the Carry (CL-register) in memory.
15. Stop.

Date

Expt. No. 14/6/2021 Page No. 11

Program-7

Program to add two BCD data.

AIM: Write an assembly language program to add two numbers of BCD data.

Address	label	M	Comments
0100		MOV SI, 1100H	Set SI-register as pointer for data
0103		MOV CL, 00H	Clear CL-register
0105		MOV AX, [SI]	Get first data in AX-register
0107		MOV BX[SI+2]	Get Second data in Bx-register
010A		ADD AL, BL	Get Sum of two low bytes in AL-register
010C		DAA	Adjust the Sum to BCD
010D		MOV DL, AL	Save Sum of lows bytes in DL-register
010F		MOV AL, AH	Move high byte of first data to AL
0111		ADC AL, BH	Get Sum of high bytes in AL-register
0113		DAA	Adjust the Sum to BCD.

Teacher's Signature

Input

Memory Address	Content
1100	23
1101	36
1102	47
1103	92

output

Memory Address.	Content
1104	70
1105	28
1106	01
1101	00

Expt. No. _____ Date _____

Page No. 12

0114		MOV DH, AL	Save Sum of high bytes in DH - register.
0116		JNC AHEAD	Checks for Carry flag.
0118		INC CL	~~Check for Carry flag.~~ if Carry flag is set then increment CL
011A	AHEAD	MOV [SI+4], DX	store the sum in memory.
011D		MOV [SI+6], CL	store the carry in memory.
0120		HLT	End.

Result : Executed the program for addl two BCD data and the Result is verified

Teacher's Signature _____

Algorithm

1. Load the address of data in SI-register.
2. Clear CL-register to account for borrow
3. Load the minued in Ax-register.
4. Get the Subtrahend in Bx-register.
5. Subtract BL from AL to get the difference of low bytes in AL
6. Adjust the difference of low bytes to BCD and then save it in DL-register
7. Get the high byte of minuend in AL-register.
8. Subtract BH and the previous borrow from AL to get the difference of high bytes in AL-register.
9. Adjust the difference of high bytes to BCD and save it in DH-register.
10. Check for Carry. If Carry flag is set then go to next step, otherwise go to step 12
11. Increment CL-register
12. Save the difference (Dx-register) and the borrow (CL-register) in memory.
13. Stop.

Date

Expt. No. 14/6/2021 Page No. 13

Program - 8.

Program to Subtract two BCD Data

AIM: Write an assembly language program to Subtract two numbers of BCD data.

Address	label		Comments
0100		MOV SI, 1100_H	Set SI-register as pointer for data
0103		MOV CL, 00_H	Clear CL-register
0105		Mov Ax, [SI]	Get minuend in Ax-register.
0107		Mov Bx, [SI+2]	Get difference of low byte Subtrahend in Bx-register.
010A		SUB AL, BL	Get difference of low bytes in AL-register.
010C		DAS	Adjust the difference to BCD.
010D		MOV DL, AL	Save difference of high/low bytes in DL-register.
010F		Mov AL, AH	Move high bytes of minued to AL-register.
0111		SBB AL, BH	Get difference of high bytes in AL-register.
0113		DAS	Adjust the difference to BCD.

Teacher's Signature

Input

Memory Address	Content
1100	72
1101	95
1102	93
1103	47

output

Memory Address	Content
1200	79
1201	47
1202	00
1203	00

Date

Expt. No.

Page No. 14

0118 4		MOV DH, AL	Save difference of high bytes in DH - register.
0118 6		JNC AHEAD	Check for Carry flag.
0118		INC CL	If Carry flag is set the increment CL.
011A	AHEAD	MOV [SI+4], DX	store the difference in memory.
011D		MOV [SI+6], CL	store the borrows in memory.
0120		HLT	Halt End.

Result : Executed the program for Subtract two BCD data and Result is verified.

Teacher's Signature

Algorithm

1. load the starting address of the array in SI - register.
2. load the address of result in DI - register.
3. load the number of bytes in the array in CL-register.
4. Increment the array pointer (SI - register).
5. Get the first byte of the array in AL-register.
6. Decrement the byte Count (CL - register)
7. Increment the array pointer (SI - register)
8. Get next byte of the array in BL - register.
9. Compare Current largest (AL) and next byte (BL) of the array.
10. Check Carry flag. If Carry flag is reset then go to step 12. Otherwise go to next step
11. Move BL to AL
12. Decrement byte Count (CL - register)
13. Check Zero flag. If Zero flag is reset then go to step 7 Otherwise go to next step.
14. Store the largest data in memory pointed by DI
15. Stop

Date

Expt. No. 20/6/2021 Page No. 15

Program - 9

Search for largest Data.

AIM: Write an assembly language program to search the largest data in an array.

Addresses	label	Mnemonics	Comments
0100		MOV SI, 1100	Set SI - register as points to array.
0103		MOV DI, 1200	Set DI - register as pointer for result
0106		MOV CL, [SI]	Set CL Count for elements in the array.
0108		INC SI	Increment the address pointer.
0109		MOV AL, [SI]	Set first data as largest
010B		DEC CL	Decrement the Count
010D	AGAIN	INC SI	Make SI to point to next data in array
010E		MOV BL, [SI]	get the next data in BL - register
0110		CMP AL, BL	Compare the Current largest data in AL BL
0112		JNC AHEAD	If Carry is not set then AL is greater than BL proceed to. AHEAD.

Teacher's Signature

Input

Memory Address.	Content
1100	05 (Count)
1101	09.
1102	08
1103	00
1104	06
1105	0C

output

Memory Address.	Content
1200	0C
1201	1C

Date

Expt. No.

Page No. 16

0114		MOV AL, BL	If Carry is set the BL as current largest
0116	AHEAD	DEC CL	Decrement the count.
0118		JNZ AGAIN	If Count is not Zero repeat search
011A		MOV [DI], AL	Store the largest data in memory
011C		HLT	End.

Result: Executed the program for search for largest data in an array and Result is varified.

Teacher's Signature

Algorithm

1. Load the address of BCD data in Bx-register
2. Get the BCD data in AL-register
3. Copy the BCD data in DL-register.
4. logically AND DL with $0F_H$ to mask upper nibble and get units digit in DL
5. logically AND AL with $F0_H$ to mask lower nibble
6. Move the Count Value for rotation in CL-register.
7. Rotate the Content of AL to move the upper nibble to lower nibble position
8. Move $0A_H$ to DH-register.
9. Multiply AL with DH-register. The product will be in AL-register.
10. Add the units digit in DL-register to product in AL-register.
11. Save the binary data (AL) in memory.
12. Stop.

Expt. No. 25/6/2021 Date

Page No. 17

Program-10

Program to Convert a BCD data to binary data.

AIM: Write an assembly language program to Convert a BCD data (2 digit | 8 bit) to binary.

Address	label	Mnemonics	Comments
0100		MOV Bx, 1100_H	load the address of the data in Bx-register.
0103		MOV AL, [Bx]	Get the BCD data in AL-register
0105		MOV DL, AL	Copy the data in DL-register
0107		AND DL, 0FH	Mask upper nibble (tens digit)
010A		AND AL, 0F0H	Mask lower nibble (units digit)
010C		MOV CL, 4	Rotate the upper nibble to lower nibble position.
0106		ROR AL, CL	
0110		MOV DH, 0AH	Set Multiplier as 0AH
0112		MUL DH	Multiply tens digit by 0AH
~~0114~~			The product will be in AL
0114		ADD AL, DL	Get Sum of unit's digit and product in AL
~~0116~~			

Teacher's Signature

Input-

Memory Address	Content
1100	23

Output

Memory Address	Content
1101	17

Date

Expt. No. _____ Page No. 18

0116		MOV [BX+1],AL	Save the binary data is memory.
0119		HLT	End.

Result: Executed the program for BCD to binary conversion the result is Verified.

Teacher's Signature _____

Algorithm

1. Load the address of data in Bx - register.
2. Get the binary data in AL - register.
3. Clear Dx - register for storing hundreds and tens
4. Compare AL with 64_H (100_{10})
5. Check Carry flag. If Carry flag is set then go to step 9. Otherwise go to next step.
6. Substract 64_H (100_{10}) from AL-register.
7. Increment hundreds register (DL)
8. Go to step 4.
9. Compare AL with $0A_H$ (10_{10}).
10. Check Carry flag. If Carry flag is Set then go to step 14, Otherwise go to next step.
11. Substract $0A_H$ (10_{10}) from AL-register
12. Increment tens register (DH)
13. Go to step 9
14. Move the Count value 04_H for rotation in CL-register
15. Rotate the Content of DH four times
16. Add DH to AL to Combine tens and units as

Date

Expt. No. 25/6/2021 Page No. 19

Program - 11

Binary to BCD Conversion

Write an assembly language program to Convert 8-bit binary data to bcd.

Address	label	Mnemonics	Comments
0100		MOV BX, 1100H	- load the address of the data in Bx - register.
0103		MOV AL, [BX]	- Get the binary data in AL - register.
0105.		MOV DX, 0000H	- Clear Dx for Storing hundreds and tens
010B.	HUND	CMP AL, 64H	- Compare whether data is less than 100 (or 64H)
010A		JC TEN	- If the data is less than 100 then jump to TEN
010E		SUB AL, 64H	- If data greater than 100, substract hundred
0110E		INC DL	- Increment hundreds register.
0110		JMP HUND	- Repeat Substraction of hundred.

Teacher's Signature

2-digit BCD

17. Save AL and DL in memory.

18. Stop

Input

Memory Address	Content
1100	24

Output

Memory Address	Content
1101	36

Date

Expt. No.

Page No. 20

0112	TEN	CMP AL, 0AH	- Compare whether data is less than 10 (or 0AH)
0114		JC UNIT	- If data is less than 10 then jump to UNIT
0116		SUB AL, 0AH	- If data greater than 10 then, subtract then
0118.		~~JMP~~ INC DH	- Increment tens register.
011A		JMP TEN	- Repeat Subtraction of ten
011C	UNIT	MOV CL, 4	- Rotate tens digit to upper nibble position
011E		ROL DH, CL	
0120		ADD AL, DH	- Combine tens and units digit
0122.		MOV [Bx+], AL	- Save tens and units in memory
0125		MOV [Bx+2], DL	- Save hundreds in memory.
0128.		HLT	- End.

Result: Executed the program to Convert the 8 bit binary to bcd and the result is Verified.

Teacher's Signature

Algorithm

1. Set SI-register as pointer for the array.
2. Set DI-register as pointer for given data and result
3. Get the data to search in DL-register
4. Let BL-register keep track of position. Initialize the position count as one
5. Get an element of array in AL
6. Compare an element of array (AL) with given data (DL)
7. Check for the Zero flag. If Zero flag is set then go to step 14. Otherwise go to next step
8. increment the array pointer (SI) and position count (BL)
9. Get next element of array in AL-register.
10. Compare AL with end marker (20_H)
11. Check Zero flag. If zero flag is not set then go to step 6. Otherwise go to next step.
12. Clear CX-register and store CX-register in four Consecutive locations in memory after the given data

...

Date

Expt. No. 5/7/2021 Page No. 21

Program - 12

Search for a given data.

AIM: Write an assembly language program to search a given data in an array. Also determine the position and address of the data in the array.

Address	label	Mnemonics	Comments
0100	START	MOV SI, 1100_H	- set SI-register as pointer for array.
0103		MOV DI, 1200_H	- load the address of data to search in DI-register.
~~0106~~			
0106		MOV DL, [DI]	- Get the data to search in DL-register.
~~01~~			
0108.		MOV BL, 01H	- Set BL-register as position count
010A		MOV AL, [SI]	Get first element of array in AL-register.
010C	AGAIN	CMP AL, DL	- Compare an element of array with data to search.
010E		JZ AVAIL	- If data are equal then jump to AVAIL
0110		INC SI	- If data are not equal, Increment Address pointer.
0111		INC BL	- Increment position Count

Teacher's Signature

13. Jump to end (step 17)

14. Move FF_H to BH-register and store it in memory.

15. Store the position count (CL) in memory.

16. Store the address (DI) in memory.

17. Stop.

(Available data)

Input

Memory Address	Content
1100	4
1101	0
1102	7
1103	8
1104	4
1105	2
1106	1
1107	20

Memory Address	Content
1200	2 data to be searched

Output

Memory Address	Content
1201	FF
1202	06
1203	05
1204	11

Search for Non-available data.

Memory Address	Content
1200	9

Output

Memory Address	Content
1201	00
1202	00
1203	00
1204	00

Date

Expt. No.

Page No. 22

0113		MOV AL, [SI]	- Get the next element of array in AL-register.
0115		CMP AL, 20H	- check for end of array.
0117		JNZ AGAIN	- If not end, Repeat search, otherwise go to NOTAVA
0119	NOTAVA	MOV CX, 0000H	- If search data is not found, then store Zero
011C		MOV [DI+1], CX	
011F		MOV [DI+3], CX	
0122		JMP OVER.	
0124	AVAIL	MOV BH, FFH	
0126		MOV [DI+1], BH	- Store FFH to indicate avilability of data.
0129		MOV [DI+2], BL	- Store the position of data
012C		MOV [DI+3], SI	- Store the address of data
012F	OVER	HLT	End.

Result: Execute the program for Search for a given data in an array and Result is verified.

Teacher's Signature

Algorithm

1. Set SI register as pointer for data
2. Get the 16 bit number in AX register.
3. Complement the content of AX - register.
4. Add one to the 1's Complement to get 2's Complant
5. Get the result in Memory pointed by SI+2.
6. Stop

Input

Memory address	Content
1100	12
1101	32

Output

Memory Address	Content
1102	EE
1103	CD
1104	00

Date

Expt. No. 5/7/2021 Page No. 23

Program - 13

2's Complement

AIM: Write an assembly language program to find the 2's Complement

Address	Label	Mnemonics	Comments
0100		MOV SI, 1100	Set SI register pointer for data
0103		MOV AX, SI	Get the 16 bit number in Ax-reg.
0105		NOT AX	Complement the content of Ax-register.
0107		ADD AX, 0001	Add one to the 1's Complement to get 2's Complement
010A		MOV [SI+2], AX	Get the result in memory memory pointed by SI+2
010D.		HLT	End.

Result: Executed the program for find the 2's Compliment and Result is Verified.

Teacher's Signature

Algorithm

1. Load the address of data in SI register.
2. Load the address of Result in DI register.
3. Get the Count in CL register.
4. Increment array pointer.
5. Get the data in AL register.
6. Rotate the Content of Register AL to left by once
7. Checks for Carry flag and if set transfer the program to step 11. otherwise go to next step.
8. Get the data again in AL register.
9. Transfer the data into array pointed by DI
10. Increment array pointer DI
11. Increment array pointer SI
12. Decrement the Count
13. Check Zero flag if it is not Zero. repeat the next data.
14. Stop.

Date

Expt. No. 14/7/2021

Page No. 24

Program-14

Separation of positive numbers

Aim: Write an assembly language program to Separate the numbers from an array.

Address	label	Mnemonics	Comments
0100		MOV SI 1100	Set SI-register as pointer for array.
0103		MOV DI, 1200	set DI-register as pointer for result
0106		MOV CL, [SI]	set CL as Count for element in the array.
0108		INC SI	Increment the address pointer.
0109	REP	MOV AL, [SI]	Get the data in AL register.
010B		ROL AL, 1	Rotate the Content of Register AL to left by once.
010D		JC NEXT	Check for Carry flag and if set transfer the program to specified location
010F		MOV AL, [SI]	Get the data again in AL-register
0111		MOV [DI], AL	Transfer the data into array pointed by DI
0113		INC DI	Increment array pointer DI
0114	NEXT	INC SI	Increment array pointer SI

Teacher's Signature

Input

Memory Address	Content
1100	0a (count)
1101	82
1102	a8
1103	72
1104	13
1105	83
1106	25
1107	65
1108	a2
1109	39
110A	FF

output

Memory Address	Content
1200	72
1201	13
1202	25
1203	69
1204	39
1205	00

Date

Expt. No.

Page No. 25

0115		DEC CL	Decrement the Count
0117		JNZ REP.	If it is not zero, repeat the procedure for the next data
0117		HLT	End.

Result : Executed the program for. Separate two numbers from an array. and Result is Verified.

Teacher's Signature

Algorithm

1. Set SI register as pointer for data
2. Set DI register as pointer for result
3. Get the ~~Content~~ Count in CL register.
4. Increment the array pointer.
5. Get the data in AL - register.
6. Rotate the Content of Reg AL to right by one
7. Check for Carry and if it is set transfer the program to step 11 ~~others~~. otherwise

Date

Expt. No. 20/1/2021

Page No. 26

Program - 15

Separation of even number.

AIM: Write an assembly language program to separate the even number.

Address	label	Mnemonics	Comments
0100		MOV SI, 1100	Set SI register as pointer array.
0103		MOV DI, 1200	Set DI register as pointer array for result
0106		MOV CL, [SI]	Set CL as count for number of bytes in array.
0108		INC SI	Increment SI pointer
0109	REP	MOV AL, [SI]	Get an element of array to AL register
010B		ROR AL, 1	Rotate bits of bytes
010D		JC NEXT	Jump on Carry.
010F		MOV AL, [SI]	Get an element of Array in AL register.
0111		MOV [DI], AL	Store the data in AL register to DI register.
0113		INC DI	Increment DI register.
0114	NEXT	INC SI	Increment SI register.

Teacher's Signature

Input

Memory Address	Content
1100	08 (count)
1101	4
1102	3
1103	8
1104	1
1105	9
1106	5
1107	3
1108.	2

output

Memory Address	Content
1200	04
1201	08
1202	02
1203	00.

Expt. No. _____ Date _____

Page No. 27

			1
0115		DEC CL	Decrement CL register
0117		JNZ REP	Jump on non-zero
0119		HLT	End.

Result: Execute the program to find the seperation of even number and the result is ~~Ver~~ Verified.

Teacher's Signature _____

Algorithm

1. Set SI register as pointer for array.
2. Set CL register as Count for N-1 repetions
3. Initialize array pointer.
4. Set CH register as count for N-1 repetitions
5. Increment the array pointer.
6. Get an element of array in AL register
7. Increment the array pointer
8. Compare the next element of array with AL
9. Check the Carry flag. If Carry flag is set then go to step 12, otherwise go to the next step.
10. Exchange the ~~Con~~ Content of memory pointed by SI and the count of previous memory location. (for this exchange AL and memory pointed by SI and then exchange AL and memory pointed by SI-1)
11. Decrement the Count for Comparisons (CH-register)
12. Check Zero flag. If Zero flag is reset then go to step 6 otherwise go to next step.
13. Decremat the Count for repetitions (CL-register.
14. Check Zero flag. If Zero flag is reset then go

Date

Expt. No. 26/7/2021

Page No. 28

Program -16

AIM: Write an assembly language program to sorting an array in ~~Assembly~~ ascending order.

Address	label	Mnemonics	Comments
0100		MOV SI, 1100	Set SI-register as pointer for array.
0103		MOV CL, [SI]	Set CL-register as Count for N-1 repetions.
0105		DEC CL	
0107	REP	MOV SI, 1100	initialize the pointer.
010A		MOV CH, [SI]	Set CH as count for N-1 comparisons.
010C		DEC CH	
010E		INC SI	Increment the pointer.
0107	REP COM	MOV AL, [SI]	Get an elent of array in AL-register.
0111		INC SI	
0112		CMP AL, [SI]	Compare with next element of array in memory.
		.	

Teacher's Signature

to Step3. otherwise go to next step.

step 15: Stop.

Input

Memory Address	Count
1100	09
1101	5
1102	3
1103	8
1104	1
1105	7
1106	2
1107	9
1108	F
1109	C
1110	9

Output

Memory Address	Count
1100	0A
1101	01
1102	02
1103	03
1104	05
1105	07
1106	08
1107	09
1108	0A
1109	0C
1110	0F

Expt. No. _____ Date _____

Page No. 29

0114		JC AHEAD	If AL is lesser than memory then go to AHEAD
0116		XCHG AL, [SI]	if AL is less than memory then exchange the content of memory pointed by SI and the previous memory locations.
0118		XCHG AL, [SI-1]	
0118	AHEAD	DEC CH	Decrement the Count for Comparisons.
011D		JNZ REPCOM.	Repeat comparison until CH count is Zero
011F		DEC CL	Decrement the Count for repetitions.
0121		JNZ REP	Repeat N-1 Comparisons until CL count is Zero.
0123		HLT	End.

Result: Executed the program to sorting an array in ascending order and Result is verified

Teacher's Signature _____

Algorithm

1. Sed SI-register as pointer for array.
2. Set CL-register as Count for N-1 repetions.
3. Initialize array pointer.
4. Set CH as Count for N-1 comparisons
5. Increment the array pointer.
6. Get an element of array in AL register.
7. Increment the array pointer.
8. Compare the next element of array in AL register.
9. Check Carry flag if Carry flag is reset then go to step 12 - otherwise go to nosct step.
10. Exchange the ~~Content~~ Content of memory pointed by SI and the Content ~~pos~~ previous memory locations (for this, exchange AL and memory pointed by SI and then ~~get~~ exchange AL and memory pointed by SI-1)
11. Decrement the Count for Comparisons (CH-register)
12. Check Zero flag. If Zero flag is reset then go to step 6. Otherwise go to next step.

Date

Expt. No. 26/7/2021

Page No. 30

Program - 17

Sorting in descending order.

AIM: Write an assembly language program for Sorting an array in descending order.

Address	label	Mnemonics	Comments
0100		MOV SI, 1100	Set SI register as pointer for array.
0103		MOV CL, [SI]	Set CL as Count for N-1 repetitions
0105		DEC CL	
0107	REPEAT	MOV SI, 1100	initialize pointer.
010A		MOV CH, [SI]	Set CH as Count for N-1 Comparisons
010C		DEC CH	
010E		INC SI	
010F	REP COM	MOV AL, [SI]	Get an element of array in AL-register.
0111		INC SI	

Teacher's Signature

13. Decrement the Count for repetitions (CL-register)

14. Check for Zero flag. If Zero flag is reset then go to step 3. otherwise go to next step.

15. Stop.

Input

Memory Address	Content
1100	05 (Count)
1101	8
1102	7
1103	1
1104	4
1105	6

Output

Memory Address	Content
1100	05
1101	08
1102	07
1103	06
1104	04
1105	01

Date

Expt. No.

Page No. 31

		CMP AL, [SI]	Compare with next element of the array in memory.
		XCHG AL, [SI]	If AL is lower than memory then
		XCHG AL, [SI]	Exchange the content of memory pointed by SI and the previous memory location.
	AHEAD	DEC CH	Decrement the Count for Comparisons
		JNZ REPCOM	Repeat Comparisons until CH Count is zero.
		DEC CL	Decrement the Count for repetitions.
		JNZ REPEAT	Repeat N-1 Comparisons until CL Count is Zero
		HLT	End.

Result : Executed the program for Sorting an array in descending order and Result is verified.

Teacher's Signature

Algorithm

1. Set SI as the memory pointer for data
2. Set DI as the memory pointer for string result
3. Clear CL register to store carry
4. Get the first number A in AL register.
5. Get the second number B in BL register.
6. Multiply the two numbers
7. Store the product of (A*B) in Bx register.
8. Get the third number C in AL register
9. Get the fourth number D in CH register.
10. Get the product in CX register.
11. Add the Content of Ax and Bx in order to get AB + CD
12. Check Carry flag. If the Carry is not set go to specified step 14. Otherwise go to next step.
13. If Carry flag is set Increment the Count in CL register.
14. Get the result in memory location pointed by DI
15. ~~Get the Carry in location pointed by DI+2~~
16. ~~Stop~~

Date

Expt No. 29/1/2021

Page No. 32

Program-18

Implementation of Expression Y= AB+CD

Aim: Write an assembly language program to evaluate the expression Y= AB+CD where A, B, C, D is an 8 bit binary numbers

Address	Label	Mnemonics	Comments
0100		MOV SI,0100	Set SI as the memory pointer for data
0103		MOV DI, 0200	Set DI as the memory pointer for string result
0106		MOV CL,00	Clear CL register for store Array
0108		MOV AL, [SI]	Get the first number AL register
010A		MOV BL, [SI+1]	Get the second number B in BL register
010D		MUL BL	Multiply the two numbers
010F		MOV BX, AX	Add the Content of Ax and Bx in order to get AB+CD
0111		MOV AL, [SI+2]	Get the third number in register AL
0114		MOV CH, [SI+3]	Get the fourth number in register CH

Teacher's Signature

15. Get the carry in location pointed by DI+2

16. Stop.

Input

Memory Address	Content
1100	23
1101	12
1102	34
1103	67

Output

Memory Address	Content
1200	62
1201	17
1202	00

Date

Expt. No.

Page No. 33

0117		MUL CH	Get the product fo of CxD
0119		ADD Ax, Bx	Add the Content of Bx and Bx border to get AB+CD
011B		JNC LP2	Check for array IA Carry is not set, go to Specified Instruction
011D		INC CL	If Carry flag is set increment the Content of CL register.
011F	LP1	Mov [DI], Ax	Get the result in memory location pointed by DI
0121		Mov [DI+2], CL	Get the array in location pointed by DI+2.
0124		HLT	Stop

Result: Executed the program for the implementation of expression $5 \cdot AB+CD$ and the result is verified.

Teacher's Signature

Algorithm

1. Set SI register as pointer for data
2. Set DI register as pointer for result
3. Initialize AH to Zero for indicating the carry
4. Get the first number in AL register.
5. Add the second number to the content of AL to get A+B
6. Check for Carry if not set, go to the step 8. otherwise go to next step
7. If Carry is there, increment AH register.
8. Transfer the sum A+B to register Bx
9. Clear AH register.
10. Get the third data (C) to register AL
11. Add fourth data (D) with the content of AL register to get C+D
12. Check for Carry, if not set go to step 14. otherwise go to next step
13. If Carry is there, increment AH register.
14. Multiply the content of Bx register with Ax to get (A+B)(C+D)
15. Transfer the lower 16 bits of the product to the memory pointed by DI

Date

Expt. No. 8/9/2021

Page No. 34

Program-19

Implementation of the expression y = (A+B)(C+D)

AIM: Write an assembly language program to evaluate the expression y = (A+B)(C+D) where A, B, C, D is an 8 bit binary number.

Address	label	Mnemonics	Comments
0100		MOV SI, 0100	Set SI register as pointer for data
0103		MOV DI, 0200	Set DI register as pointer for result
0106		MOV AH, 00	Initialize AH to zero for indicating the Carry
0108		MOV AL, [SI]	Get the first number in AL register.
010A		ADD AL, [SI+1]	Add the second number to the content of AL to get A+B
010D		JNC LP.	Check for Carry, if not set, go to the Specified location
010F		INC AH	If Carry is there, increment AH register.
0111	LP.	MOV Bx, Ax	Transfer the sum A+B to register Bx
0113		MOV Ax, 00	Clear AH register.

Teacher's Signature

16. Transfer the higher 16 bits of the product to the memory pointed by DI+2

17. Stop

Input

Memory Address	Content
1100	25
1101	12
1102	30
1103	10

output

Memory Address	Content
1200	C0
1201	0D
1202	00

Expt. No. _____ Date _____

Page No. 25

0115		MOV AL, [SI+2]	Get the third data (C) in Register AL
0118		ADD AL, [SI+1]	Add Fourth data (D) in the Content of AL register to get C+D
		JNC LP2	check For Carry If, not set go to specified location
		INC AH	If Carry is there increment AH register.
	LP2	MUL DX	Multiply the Content of DX register with Ax to get (A+B) x (C+D)
		MOV [DI], Ax	Transfer the lower 16 bits of the product to the memory pointed.
		MOV [DI+2], DX	Transfer the higher 16 bits of the product to memory pointed DI+2
		HLT	End.

Result: Executed the program for the implementation of expression (A+B) (C+D) and the result is verified

Teacher's Signature _____

Algorithm

1. Set SI register as pointer to array.
2. Set DI register as pointer to array to store the result
3. Set CL register as two the count
4. Increment SI pointer.
5. Clear direction flag.
6. Get the byte from ~~Source~~ Source string to destination string.
7. loop
8. Stop

Input

Memory Address	Content
1100	03 (count)
1101	5
1102	23
1103	46

Output

Memory Address	Content
1200	05
1201	23
1202	46
1203	00

Date

Expt. No. 8/9/2021 Page No. 36

Program - 20.

Moment of String

AIM: Write an ALP to perform more byte from Source to destination the input stored from location 2700 and the output will be stored from 3701

Address	label	Mnemonics	Comments
0100		Mov SI, 2700	Set SI register as the pointer to array
0103		Mov CL, [SI]	Set CL register as the Count for no. of bytes in array
0105		INC SI	Increment SI pointer.
0106		Mov di, 3701	Set DI register as the pointer to array.
0109		Cld	Clear direction flag
010A	L1	Mov sb	Mov the byte from Source string to destination to destnation string loop.
010B		loop L1	loop.

Teacher's Signature

Date

Expt. No. _____ Page No. 37

010D		HLT	End.

Result : Executed the program & the moment of string.
the Result is Varified.

Teacher's Signature _____

Algorithm

1. Set SI register as pointer to array
2. Get DI register as pointer array to store the result
3. Move the Content in SI to AL register.
4. Move the Data 00FF to Cx register.
5. Clear direction flag.
6. Store bytes into register.
7. Loop
8. End.

Input

Memory Address	Content
1100	29

Output

Memory Address	Content
1200	2A
1201	2A
1202	2A
12FF	2A

Date

Expt No. 9/9/2021 Page No. 38

Program: 21

Filling Array with given data.

Aim: Write an assembly language program to fill the locations in memory with bytes to given

Address	Label	Mnemonics	Comments
0100		MOV SI, 1100	Set SI register as pointer to array.
0103		MOV DI, 1200	Set DI register as pointer. array to store the result
0105		MOV AL, [SI]	Move the Content in SI to AL register.
0108		MOV CX, 00FF	Move the data 00FF to Cx register
010B		CLD	Clear direction flag.
010C	LP,	STOSB	Store byte into register.
010D		LOOP LP,	loop.
010F		HLT	End.

Result: Executed the program for to fill the locations in memory with bytes to given and Result is verified.

Teacher's Signature