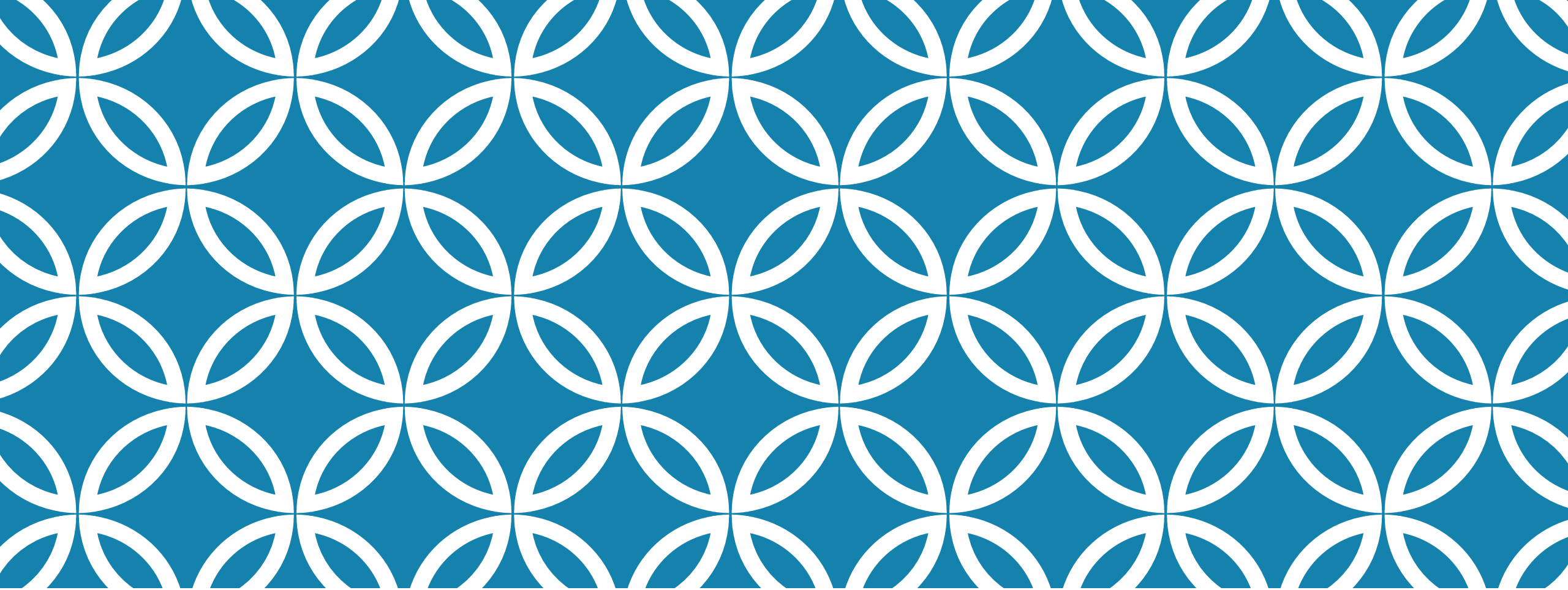

ASSEMBLY LANGUAGE PROGRAMS (ALP) OF 8085

DATA TRANSFER PROGRAMS

1. Write an ALP for loading registers A, B, C, D, E, H and L with single byte data addressing using immediate addressing

MEMORY ADDRESS	MACHINE CODES	LABEL	OPCODE	OPERAND	COMMENTS
2000	3E 01	START	MVI	A,01	Load A with 01
2002	06 02		MVI	B,02	Load B with 02
2004	0E 03		MVI	C,03	Load C with 03
2006	16 04		MVI	D,04	Load D with 04
2008	1E 05		MVI	E,05	Load E with 05
200A	26 06		MVI	H,06	Load H with 06
200C	2E 07		MVI	L,07	Load L with 07
200E	EF	END	RST	05	Return to monitor program

2. Write an ALP for loading registers B, C, D, E, H and L with same data using register addressing

MEMORY ADDRESS	MACHINE CODES	LABEL	OPCODE	OPERAND	COMMENTS
2020	3A 50 20	START	LDA	2050	Load accumulator with 2050
2023	47		MOV	B, A	Move the content of A to B
2024	4F		MOV	C, A	Move the content of A to C
2025	57		MOV	D, A	Move the content of A to D
2026	5F		MOV	E, A	Move the content of A to E
2027	67		MOV	H, A	Move the content of A to H
2028	6F		MOV	L, A	Move the content of A to L
2029	EF	END	RST	05	Return to monitor program

3. Write an ALP for loading register pairs BC, DE and HL with 16-bit data using immediate addressing

MEMORY ADDRESS	MACHINE CODE	LABEL	OPCODE	OPERAND	COMMENTS
2050	01 50 21	START	LXI	B, 2150	Load BC register pair with data 2150
2053	11 51 21		LXI	D, 2151	Load DE register pair with data 2151
2056	21 52 21		LXI	H, 2152	Load HL register pair with data 2152
2059	EF	END	RST	05	Return to monitor program

BLOCK DATA TRANSFER

4. Write an ALP to copy a block of data from 4 memory locations to another 4 memory locations using 8-bit data transfer addressing mode direct addressing.

MEMORY ADDRESS	MACHINE CODES	LABEL	OPCODE	OPERAND	COMMENTS
2060	3A 50 20	START	LDA	2250	Load accumulator with 2250
2063	32 54 22		STA	2254	Accumulator content stored in 2254
2066	3A 51 22		LDA	2251	Load data in 2251 to accumulator
2069	32 55 22		STA	2255	Accumulator data stored in 2255
206C	3A 52 22		LDA	2252	Load data in 2252 to accumulator
206F	32 56 22		STA	2256	Accumulator data stored in 2256
2072	3A 53 22		LDA	2253	Load data in 2253 to accumulator
2075	32 57 22		STA	2257	Accumulator data stored in 2257
2078	EF	END	RST	05	Return to monitor program

5. Repeat 4th ALP using 16-bit data transfer addressing mode direct addressing

MEMORY ADDRESS	MACHINE CODE	LABEL	OPCODE	OPERAND	COMMENTS
2080	2A 50 20	START	LHLD	2050	Data in 2050 to L register and data in 2051 to H register
2083	22 54 20		SHLD	2054	L register content to 2054 and H register content to 2055
2086	2A 52 20		LHLD	2052	Data in 2052 to L register and data in 2053 to H register
2089	22 56 20		SHLD	2056	L register content to 2056 and H register content to 2057
2092	EF	END	RST	05	Return to monitor program

6. Repeat 4th ALP using 16-bit data transfer addressing mode indirect addressing

MEMORY ADDRESS	MACHINE CODE	LABEL	OPCODE	OPERAND	COMMENTS
		START	LXI	H, 2050	Initialize HL pair
			LXI	B, 2051	Initialize BC pair
			LXI	D, 2055	Initialize DE pair
			MVI	M, 04	Set counter as 4
		LOOP	LDAX	B	Load content of memory location whose address is in BC pair to accumulator
			STAX	D	Store content of accumulator into memory location whose address is in DE pair
			INX	B	Increment BC pair
			INX	D	Increment DE pair
			DCR	M	Decrement count by 1
			JNX	LOOP	Jump if non zero
	EF	END	RST	05	Return to monitor program

BLOCK DATA TRANSFER

Program –

MEMORY	MNEMONICS	OPERANDS	COMMENT
2000	MVI	C, 05	[C] <- 05
2002	LXI	H, 2500	[H-L] <- 2500
2005	LXI	D, 2600	[D-E] <- 2600
2008	MOV	A, M	[A] <- [[H-L]]
2009	STAX	D	[A] -> [[D-E]]
200A	INX	H	[H-L] <- [H-L] + 1
200B	INX	D	[D-E] <- [D-E] + 1
200C	DCR	C	[C] <- [C] – 1
200D	JNZ	2008	Jump if not zero to 2008
2010	HLT		Stop

ADDITION OF TWO 8 BIT NUMBERS-SUM 8 BIT

PROGRAM

Memory address	*Machine Codes*	*Mnemonics*	*Operands*	*Comments*
2000	21, 01, 25	LXI	H, 2501 H	Get address of 1st number in H-L pair.
2003	7E	MOV	A,M	1st number in accumulator.
2004	23	INX	H	Increment content of H-L pair.
2005	86	ADD	M	Add 1st and 2nd numbers.
2006	32, 03, 25	STA	2503 H	Store sum in 2503 H.
2009	76	HLT		Stop

WRITE AN ALP TO SUBTRACT TWO 8 BIT NUMBERS, DIFFERENCE 8 BITS

ADDITION OF TWO 8 BIT NUMBERS-SUM 16 BITS

PROGRAM

Memory address	Machine Codes	Labels	Mnemonics	Operands	Comments
2000	21, 01, 25		LXI	H, 2501 H	Address of 1st number in H-L pair.
2003	0E, 00		MVI	C,00	MSBs of sum in register C. Initial value = 00.
2005	7E		MOV	A, M	1st number in accumulator.
2006	23		INX	H	Address of 2nd number 2502 in H-L pair.
2007	86		ADD	M	1st number + 2nd number.
2008	D2, 0C, 20		JNC	AHEAD	Is carry? No, go to the label AHEAD.
200B	0C		INR	C	Yes, increment C.
200C	32, 03, 25	AHEAD	STA	2503 H	LSBs of sum in 2503 H.
200F	79		MOV	A, C	MSBs of sum in accumulator.
2010	32, 04, 25		STA	2504 H	MSBs of sum in 2504 H.
2013	76		HLT		Halt

WRITE AN ALP FOR THE DECIMAL ADDITION OF TWO 8 BIT NUMBERS, SUM 16 BITS

ADDITION OF TWO 16 BIT NUMBERS-SUM 16 BIT OR MORE

PROGRAM

Address	Machine Codes	Labels	Mnemonics	Operands	Comments
2000	2A, 01, 25		LHLD	2501 H	1st 16-bit number in H-L pair.
2003	EB		XCHG		Get 1st number in D-E pair.
2004	2A, 03, 25		LHLD	2503 H	2nd 16-bit number in H-L pair.
2007	0E, 00		MVI	C, 00	MSBs of the sum in register C. Initial value = 00.
2009	19		DAD	D	1st number + 2nd number
200A	D2, 0E, 20		JNC	AHEAD	Is carry? No, go to the label AHEAD.
200D	0C		INR	C	Yes, increment C.
200E	22, 05, 25	AHEAD	SHLD	2505 H	Store LSBs of sum in 2505 and 2506 H.
2011	79		MOV	A, C	MSBs of sum in accumulator.
2012	32, 07, 25		STA	2507 H	Store MSBs of the sum in 2507 H.
2015	76		HLT		Halt.

SUM OF A SERIES OF 8 BIT NUMBERS: SUM 16 BITS

Program

Address	HEX Codes	Labels	Mnemonics	Comments
F000	21, 00, 80		LXI H,8000H	Load the address to get count of numbers
F003	4E		MOV C, M	Load C with the count value
F004	21, 10, 80		LXI H, 8010H	Load HL with the starting address
F007	AF		XRA A	Clear accumulator
F008	5F		MOV E, A	Clear the E register also
F009	86	LOOP	ADD M	Add Memory content with Accumulator
F00A	D2, 0C, F0		JNC SKIP	When Carry flag is 0, skip next task
F00D	1C		INR E	Increase E, when C flag is set
F00E	0D	SKIP	DCR C	Decrease C register by 1
F00F	23		INX H	Point to next location
F010	C2, 09, F0		JNZ LOOP	When Zero is false, go to LOOP
F013	21, 00, 90		LXI H,9000H	Load address to store result
F016	77		MOV M, A	Save accumulator content
F017	23		INX H	Increase HL pair
F018	73		MOV M, E	Store carry
F019	76		HLT	Terminate the program

WRITE AN ALP TO ADD A SERIES OF 8 BIT DECIMAL NUMBERS, SUM 16 BITS

Write an ALP to shift an 8 bit number left by 1 bit

```
LDA 2501H
ADD A
STA 2502H
RST 05
```

Write an ALP to shift an 8 bit number left by 2 bits

Write an ALP to shift an 16 bit number left by 1 bit

```
LHLD 2501H
DAD H
SHLD 2503H
RST 05
```

Write an ALP to shift an 16 bit number left by 2 bits

TWO 8 BIT MULTIPLICATION

ALGORITHM:

1) Start the program by loading HL register pair with address of memory location.
2) Move the data to a register (B register).
3) Get the second data and load into Accumulator.
4) Add the two register contents.
5) Check for carry.
6) Increment the value of carry.
7) Check whether repeated addition is over and store the value of product and carry in memory location.
8) Terminate the program.

PROGRAM:

```
        MVI     D, 00       Initialize register D to 00
        MVI     A, 00       Initialize Accumulator content to 00
        LXI     H, 4150
        MOV     B, M        Get the first number in B - reg
        INX     H
        MOV     C, M        Get the second number in C- reg.
LOOP:   ADD     B           Add content of  A - reg to register B.
        JNC     NEXT        Jump on no carry to NEXT.
        INR     D           Increment content of register D
NEXT:   DCR     C           Decrement content of register C.
        JNZ     LOOP        Jump on no zero to address
        STA     4152        Store the result in Memory
        MOV     A, D
        STA     4153        Store the MSB of result in Memory
        HLT                 Terminate the program.
```

TO FIND LARGER OF TWO NUMBERS

PROGRAM

Memory address	*Machine Codes*	*Labels*	*Mnemonics*	*Operands*	*Comments*
2000	21, 01, 25		LXI	H, 2501 H	Address of 1st number in H-L pair.
2003	7E		MOV	A, M	1st number in accumulator.
2004	23		INX	H	Address of 2nd number in H-L pair.
2005	BE		CMP	M	Compare 2nd number with 1st number. Is the 2nd number > 1st ?
2006	D2, 0A, 20		JNC	AHEAD	No, larger number is in accumulator. Go to AHEAD.
2009	7E		MOV	A, M	Yes, get 2nd number in accumulator.
200A	32, 03, 25	AHEAD	STA	2503 H	Store larger number in 2503 H.
200D	76		HLT		Stop

TO FIND SMALLER OF TWO NUMBERS

TO FIND THE LARGEST NUMBER IN AN ARRAY

PROGRAM

Memory address	*Machine Codes*	*Labels*	*Mnemonics*	*Operands*	*Comments*
2000	21, 00, 25		LXI	H, 2500 H	Address for count in H-L pair.
2003	4E		MOV	C, M	Count in register C.
2004	23		INX	H	Address of 1st number in H-L pair.
2005	7E		MOV	A, M	1st number in accumulator
2006	0D		DCR	C	Decrement count.
2007	23	LOOP	INX	H	Address of next number.
2008	BE		CMP	M	Compare next number with previous maximum. Is next number > previous maximum?
2009	D2, 0D, 20		JNC	AHEAD	No, larger number is in accumulator. Go to the lable AHEAD
200C	7E		MOV	A, M	Yes, get larger number in accumulator.
200D	0D	AHEAD	DCR	C	Decrement count.
200E	C2, 07, 20		JNZ	LOOP	
2011	32, 50, 24		STA	2450 H	Store result in 2450 H.
2014	76		HLT		Stop.

TO FIND THE SMALLEST NUMBER IN AN ARRAY

ARRANGE THE NUMBERS IN THE DESCENDING ORDER

ALGORITHM:

1. Initialize HL pair as memory pointer
2. Get the count at 4200 into C – register
3. Copy it in D – register (for bubble sort (N-1) times required)
4. Get the first value in A – register
5. Compare it with the value at next location.
6. If they are out of order, exchange the contents of A –register and Memory
7. Decrement D –register content by 1
8. Repeat steps 5 and 7 till the value in D- register become zero
9. Decrement C –register content by 1
10. Repeat steps 3 to 9 till the value in C – register becomes zero

PROGRAM:

```
        LXI     H,4200
        MOV     C,M
        DCR     C
REPEAT: MOV     D,C
        LXI     H,4201
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
```

ARRANGE THE NUMBERS IN THE ASCENDING ORDER

ALGORITHM:

1. Initialize HL pair as memory pointer
2. Get the count at 4200 into C – register
3. Copy it in D – register (for bubble sort (N-1) times required)
4. Get the first value in A – register
5. Compare it with the value at next location.
6. If they are out of order, exchange the contents of A –register and Memory
7. Decrement D –register content by 1
8. Repeat steps 5 and 7 till the value in D- register become zero
9. Decrement C –register content by 1
10. Repeat steps 3 to 9 till the value in C – register becomes zero

PROGRAM:

```
        LXI     H,4200
        MOV     C,M
        DCR     C
REPEAT: MOV     D,C
        LXI     H,4201
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     C
        JNZ     REPEAT
        HLT
```