Name : Chandrawanshi Mangesh Shivaji

Roll No. : 1801CS16

Course : CS322 (Computer Architechture Lab)

Date : 05/09/2020.

FileName : 1801CS16_Lab2.pdf

Task1 :-

P1) Program to swap two 8-bit numbers.

PROGRAM :-

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERAND	COMMENT
2000H			ORG	2000H	Program start address
			BEGIN	2000H	Start program from 2000H
2000H	3A 00 25		LDA	2500H	Load value at 2500H in A
2003H	47		MOV	B,A	Move it to B.
2004H	3A 01 25		LDA	2501H	Load value at 2501H in A.
2007H	32 00 25		STA	2500H	Store value at A in 2500H
200AH	78		MOV	A, B	Move value in B to A.
200BH	32 01 25		STA	2501H	Store value at A in 2501H
200EH	76		HLT		Halt program execution.

OBSERVATION and RESULTS :-

INPUT DATA :

Memory Address	Content
2500H	14H
2501H	24H

(Before Swap)

OUTPUT DATA :

MEMORY Address	Content.
2500H	24H
2501H	14H

(After Swap)

P2) Program to find square of an 8-bit number (only if square can be stored in 8-bit).

PROGRAM:-

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERAND	COMMENTS
2000H			ORG	2000H	Program start address.
			BEGIN	2000H	Start Program (2000H)
2000H	21 00 30		LXI	H, 3000H	Load Memory location of given no. in H-L register pair
2003H	46		MOV	B, M	Move value to B from M
2004H	4E		MOV	C, M	Move value to C from M
2005H	3E 00		MVI	A, 00H.	Initialize A to zero
2007H	80	LOOP	ADD	B	Add B(value) to A
2008H	0D		DCR	C	Decrement C (counter)
2009H	C2 07 20		JNZ	LOOP	IF C is not-zero, go to LOOP label.
200CH	23		INX	H	Increment H-L pair to next location
200DH	32 00 25		STA	2500H	Store value in A (square) to 2500H
2010H	76		HLT		HALT program execution.

OBSERVATION and RESULTS :-

INPUT DATA:

Memory Address	Content
3000H	05H

(Given Number)

OUTPUT DATA:

Memory Address	Content
2500H	19H.

(Squared value).

P3) Program to find largest of 2 8-bit numbers

PROGRAM :-

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERAND	COMMENTS
2000H			ORG1	2000H	Program start address
			BEGIN	2000H	Start Program (2000H)
2000H	21 00 30		LXI	H, 3000H	Load memory location of first no. in given H-L register pair
2003H	7E		MOV	A, M	Move value to A
2004H	23		INX	H	Increment H for accessing second no.
2005H	46		MOV	B, M	Move it to B.
2006H	B8		CMP	B	Compare B with A.
2007H	D2 0B 20		JNC	GO	If B<=A, jump to GO label.
200AH	78		MOV	A, B	Move B (value) to A
200BH	23	GO	INX	H	Increment H.
200CH	32 00 25		STA	2500H	Store largest no. at 2500H.
200FH	76		HLT		halt prog. execution.

OBSERVATION and RESULTS :-

INPUT DATA :

Memory Address	Content
3000H	05H
3001H	08H

(Two Given Numbers)

OUTPUT DATA

Memory Address	Content
2500H	08H.

(Largest Number)

P4) Program to find smallest from an array

PROGRAM :-

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERAND	COMMENTS
2000H			ORG	2000H	Program start address
2000H			BEGIN	2000H	Start Program (2000H)
2000H	21 00 30		LXI	H, 3000H	load Memory Location of size of the array in HL register Pair
2003H	4E		MOV	C, M	Move size(M) to C
2004H	23		INX	H	Increment H to access array elements.
2005H	7E		MOV	A, M	Move first no to A.
2006H	0D		DCR	C	decrement C (counter)
2007H	23	LOOP	INX	H	Increment H to access next element.
2008H	46		MOV	B, M	Move value to B.
2009H	B8		CMP	B	Comp. with A.
200AH	DA 0E 20		JC	G10	if value is small else G10 Label
200DH	78		MOV	A, B	Move it to A.
200EH	0D	G10	DCR	C	Decrement C.
200FH	C2 07 20		JNZ	LOOP	if C is not-zero go to LOOP label
2012H	23		INX	H	Increment H
2013H	32 00 25		STA	2500H	Store Smallest value (A) to 2500H.
2016H	76		HLT		

OBSERVATION and RESULTS :-

INPUT DATA :

Memory Address	Content
3000H	05H
3001H	08H
3002H	01H
3003H	12H
3004H	06H
3005H	03H

OUTPUT DATA :

Memory Address	Content
2500H	01H

P5) Program to sort the array in Descending Order (In place sorting).

PROGRAM :-

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERAND	COMMENT.
2000H			ORG	2000H	Program start address.
			BEGIN	2000H	Start Program
2000H	21 0030		LXI	H, 3000H	Load location of size in HL
2003H	4E		MOV	C, M	Move val(M) to C
2004H	21 00 30	LOOP1	LXI	H, 3000H	Load location of size in HL
2007H	56		MOV	D, M	Move val(M) to D
2008H	23		INX	H	Increment H
2009H	15		DCR	D	Decrement D
200AH	7E	LOOP2	MOV	A, M	Move val(M) to A (first element)
200BH	23		INX	H	Increment H
200CH	BE		CMP	M	Comp M with A
200DH	D2 18 20		JNC	G0	if value in M is greater or equal, keep as it is.
2010H	CA 18 20		JZ	G0	
2013H	46		MOV	B, M	else Move M to B (swap)
2014H	77		MOV	M, A	Move A to M
2015H	2B		DCX	H	Decrement H
2016H	70		MOV	M, B	Move B to M
2017H	23		INX	H	Increment H (Swap done)
2018H	15	G0	DCR	D	Decrement D
2019H	C2 0A 20		JNZ	LOOP2	if D is not zero loop again
201CH	0D		DCR	C	Decrement C
201DH	C2 04 20		JNZ	LOOP1	if C is not zero loop again
2020H	76		HLT		Halt prog. exec.

OBSERVATION AND RESULT

Input Data: Memory Address	Content		Output Data: Memory Address	Content
3000H	05H	→ size of array	3000H	05H
3001H	08H	Array elements.	3001H	12H
3002H	01H		3002H	08H
3003H	12H		3003H	06H
3004H	06H		3004H	03H
3005H	03H		3005H	01H
Unsorted array			Sorted Array (Desc.)	

P6) Program to convert HEX to BCD.

PROGRAM :-

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERAND	COMMENT
2000H			ORG	2000H	Address of Start
			BEGIN	2000H	Start prog.
2000H	21 00 30	~~LXI~~	~~H,3000H~~ LXI	H,3000H	LOAD Mem. loc. in HL pair
2003H	4E		MOV	C,M.	Move M to C
2004H	C6 01	LOOP	ADI	01	Add 01H to A.
2006H	27		DAA		Get BCD sum as outcome
2007H	D2 0B 20		JNC	G10.	if carry is set
200AH	04		INR	B.	increment B.
200BH	0D	G10	DCR	C	~~else~~ decrement C
200CH	C2 04 20	~~JNZ~~	JNZ	LOOP.	if C is not zero loop again
200FH	6F		MOV	L,A	Move A to L
2010H	60		MOV	H,B.	Move B to H
2011H	22 00 25		SHLD	2500H	Store the BCD to 2500H
2014H	76		HLT.		Halt prog. exec.

OBSERVATION and RESULT :-

INPUT DATA:	Mem. Add.	Content	Output	Mem. Add	Content
	3000H	A5H (HEX)	DATA:	2500H	65H (BCD)
				2501H	01H

P7) Write a Program to Multiply number by 7 (w/o actually adding or multiplying) (only if result is representable by 8-bits).

PROGRAM :-

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERAND	COMMENT
2000H			ORG	2000H	Address of Start
			BEGIN	2000H	Start prog.
2000H	21, 00 30		LXI	H, 3000H	Load Mem Loc. of given no. in HL
2003H	7E		MOV	A, M	Move M to A
2004H	4E		MOV	C, M	Move M to C.
2005H	07		RLC		Shift A to left bit
2006H	07		RLC		Shift A to left bit
2007H	07		RLC		Shift A to left 1 bit
2008H	91		SUB	C	Subtract C from A
2009H	32 00 25		STA	2500H	Store required val to 2500H
200CH	76		HLT.		halt prog. exec.

OBSERVATION and RESULT :-

INPUT DATA:

Mem. Add.	Content
3000H	05H

(given number)

output DATA:

Mem. Add.	Content
2500H	23H

(value on multiplying by 5)

P8) Count total odd numbers in an array.

PROGRAM :-

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERAND	COMMENT
2000H			ORG1	2000H	Address of Start
			BEGIN	2000H	Start prog.
2000H	21 00 30		LXI	H, 3000H	load Mem Loc. of size of array in HL
2003H	06 00		MVI	B, 00H	Initize all B with 0
2005H	4E		MOV	C, M	Move M to C
2006H	23		INX	H	Increment H
2007H	7E	LOOP	MOV	A, M	Move M to A
2008H	E6 01		ANI	01H	Apply AND 01H to A
200AH	CA 0E 20		JZ	GO.	if zero, label GO
200DH	04		INR	B	increment B
200EH	0D	GO	DCR	C	decrement C.
200FH	23	"	INX	H	Increment H
2010H	C2 07 20		JNZ	LOOP	if C non-zero loop again
2013H	78		MOV	A, B.	Move B to A
2014H	32 00 25		STA	2500H	Store count of odd nos. at 2500H
2017H	76		HLT		halt prog. exec.

OBSERVATION and RESULT :-

INPUT DATA

Memory Address	Content	
3000H	05H	} size
3001H	08H	} Array elements
3002H	01H	
3003H	12H	
3004H	06H	
3005H	03H	

output DATA

Memory Address	Content
2500H	02H.

No. of odd elements in array.

Pg) Calculate sum of all even numbers in the given array

PROGRAM :-

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERAND	COMMENT.
2000H			ORG	2000 H	Address of start
			BEGIN	2000 H	Start Prog.
2000H	21 00 30		LXI	H, 3000 H	load size location in HL pair
2003H	06,00		MVI	B, 00H	initialize B to 0.
2005 H	4E		MOV	C, M	Move M to C
2006H	23		INX	H	Increment H
2007 H	7E	LOOP	MOV	A, M	Move M to A
2008H	E6 01		ANI 01	01H	Apply AND to A with 01 H
200AH	C2 10 20		JNZ	G10	if A is not zero jump to G10.
200DH	78		MOV	A, B	Move B to A.
200EH	86		ADD	M	Add M to A
200FH	47		MOV	B, A	Move A to B.
2010 H	0D	G10	DCR	C	decrement C.
2011 H	23		INX	H	Increment H
2012 H	C2 07 20		JNZ	LOOP	if C is not zero jump to LOOP.
2015 H	78		MOV	A, B	Move B to A.
2016 H	32 00 25		STA	2500H	Store sum (A) to 2500H.
2019H	76		HLT		halt prog exec.

- OBSERVATIONS AND RESULTS.

IP DATA

Mem. Add.	Content.	
3000H	05H	→ size
3001 H	08H	} Array elements
3002H	01 H	
3003H	12 H	
3004H	06H	
3005H	03H	

OP DATA

Mem Add.	Content
2500H	20H

sum of even nos.

P10) Program to FIND factorial of a number (result must be representable in 8-bits).

PROGRAM :-

ADDRESS	OPCODE	LABEL	MNEMONICS	OPERAND	COMMENTS
2000H			ORG	2000H	Starting Address
			BEGIN	2000H	Start Program.
2000H	21 00 30		LXI	H, 3000H	Load Mem. Loc. of no. in HL pair
2003H	46		MOV	B, M	Move M to B
2004H	16 01		MVI	D, 01H	Initialize D to 01H
2006H	CD 13 20	FACTORIAL	CALL (MULTIPLY) →		go to label MULTIPLY
2009H	05		DCR	B	Decrement B
200AH	C2 06 20		JNZ	FACTORIAL	if B is non-zero go to factorial label.
200DH	23		INX	H	Increment H.
200EH	7A		MOV	A, D	Move D to A
200FH	32 00 25		STA	2500H	store factorial (A) to 2500H
2012H	76		HLT		halt prog. exec.
2013H	48	MULTIPLY	MOV	C, B	Move B to C.
2014H	3E 00		MVI	A, 00H	Initialize A to 00H
2016H	82	LOOP	ADD	D	Add D to A.
2017H	0D		DCR	C	Decrement C
2018H	C2 16 20		JNZ	LOOP	if C is not-zero go to LOOP label
201BH	57		MOV	D, A	Move A to D.
201CH	C9		RET		return (like a function call in high level Prog. Lang).

OBSERVATION and RESULT.

IP DATA

Mem. Add	Content
3000H	05H (given) DATA (number)

OP

Mem. Add	Content
2500H	78H (factorial value)

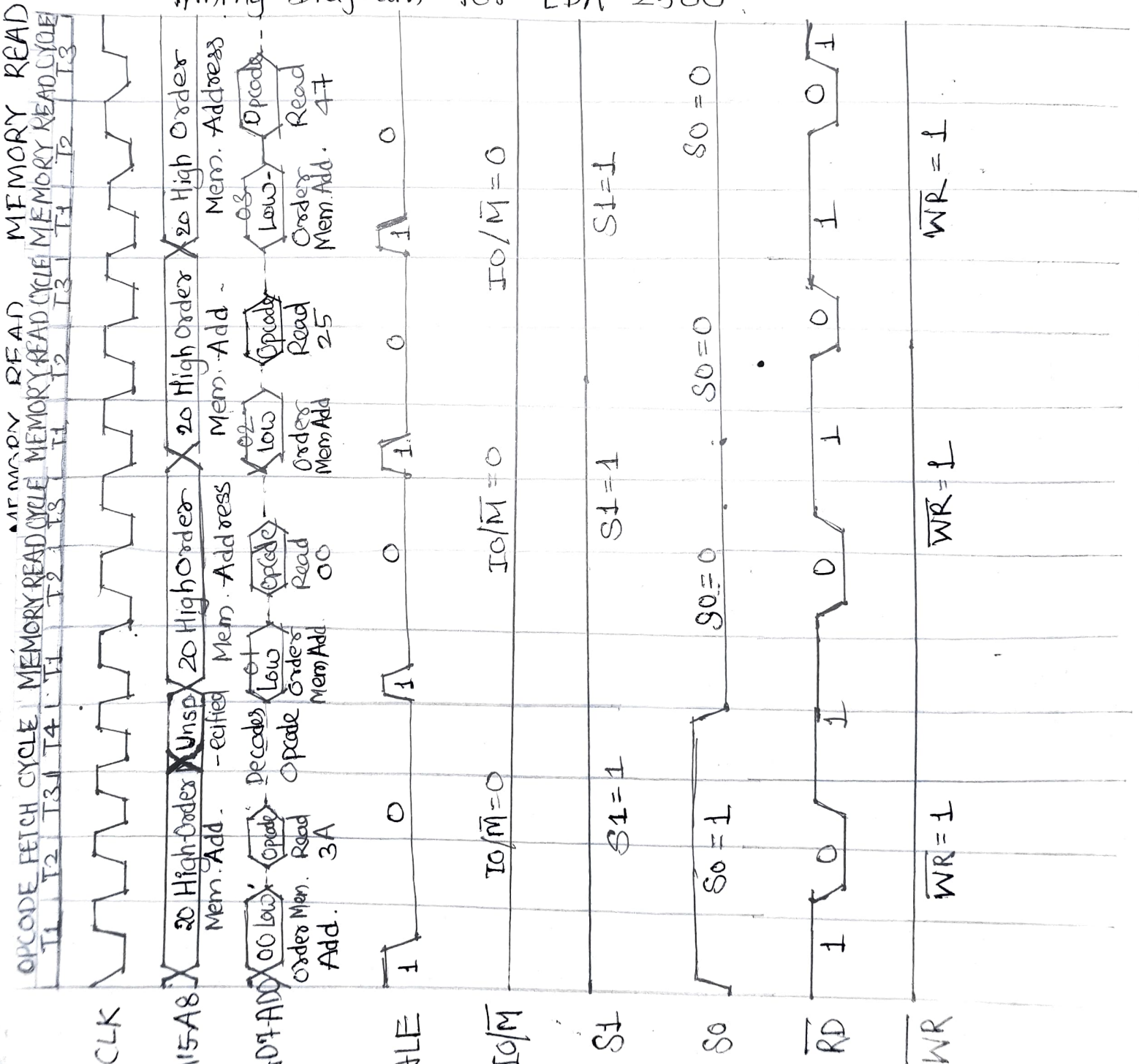
Task 2 :- Draw Timing Diagram of 10 key instructions from task 1 (All instructs should be different).

Instruction 1 : LDA 16-bit address. (3-byte instruction)

What it does : Load Accumulator with the contents from memory The accumulator will get initialized with 8-bit content from the 16-bit memory address.

1st Byte - opcode 2nd Byte and 3rd - 16-bit address

Timing Diagram for LDA 2500

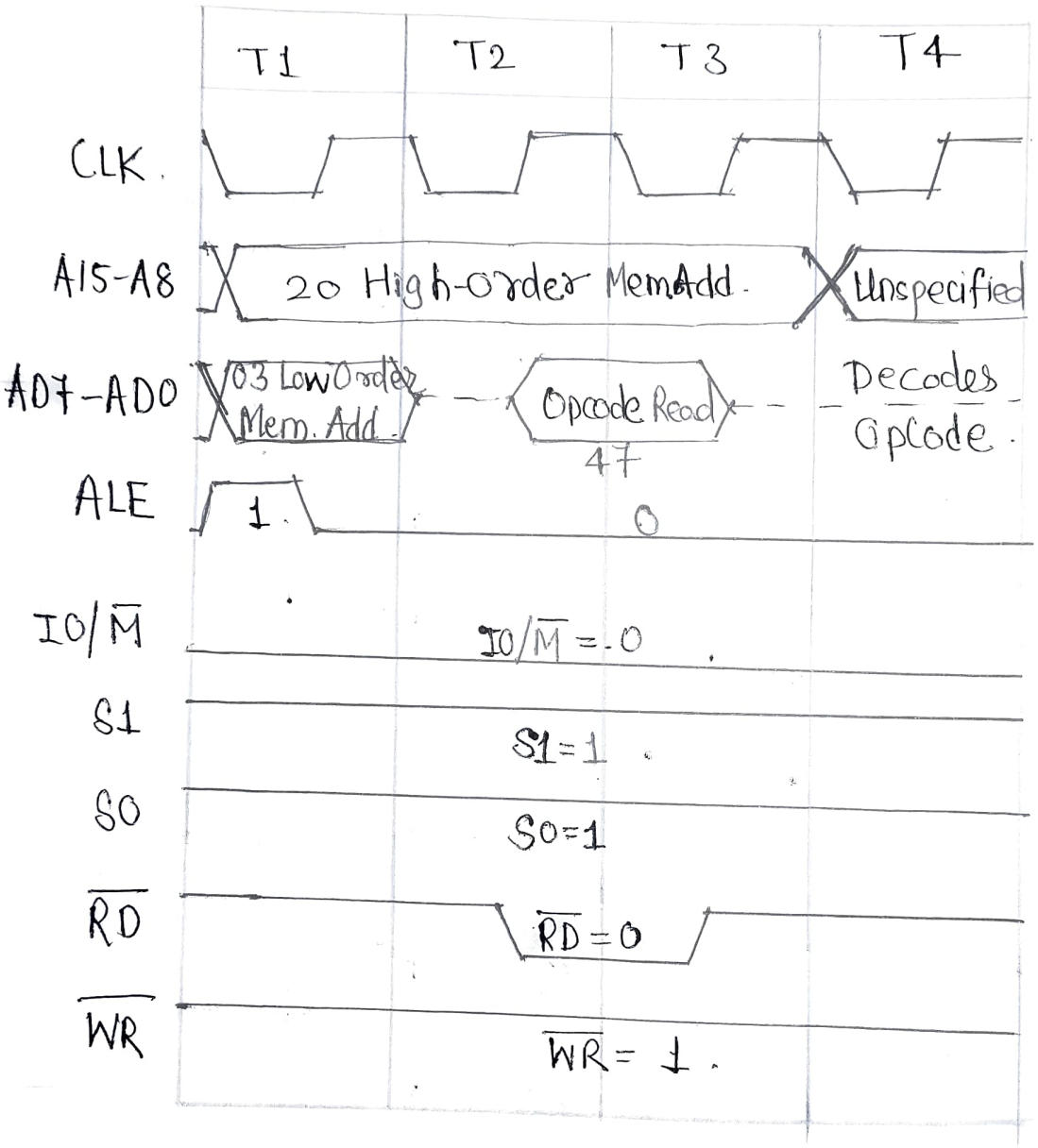


Instruction 2: MOV dest, src (1 byte Instruction)

What it does :- 8-bit data value in register src will be moved to 8-bit register dest.

Timing diagram of MOV B, A

OPCODE FETCH CYCLE



Instruction 3 :- STA 16-bit Address (3-Byte Instruction)

What it does :- Stores value in accumulator at given memory location specified by 16-bit Address.

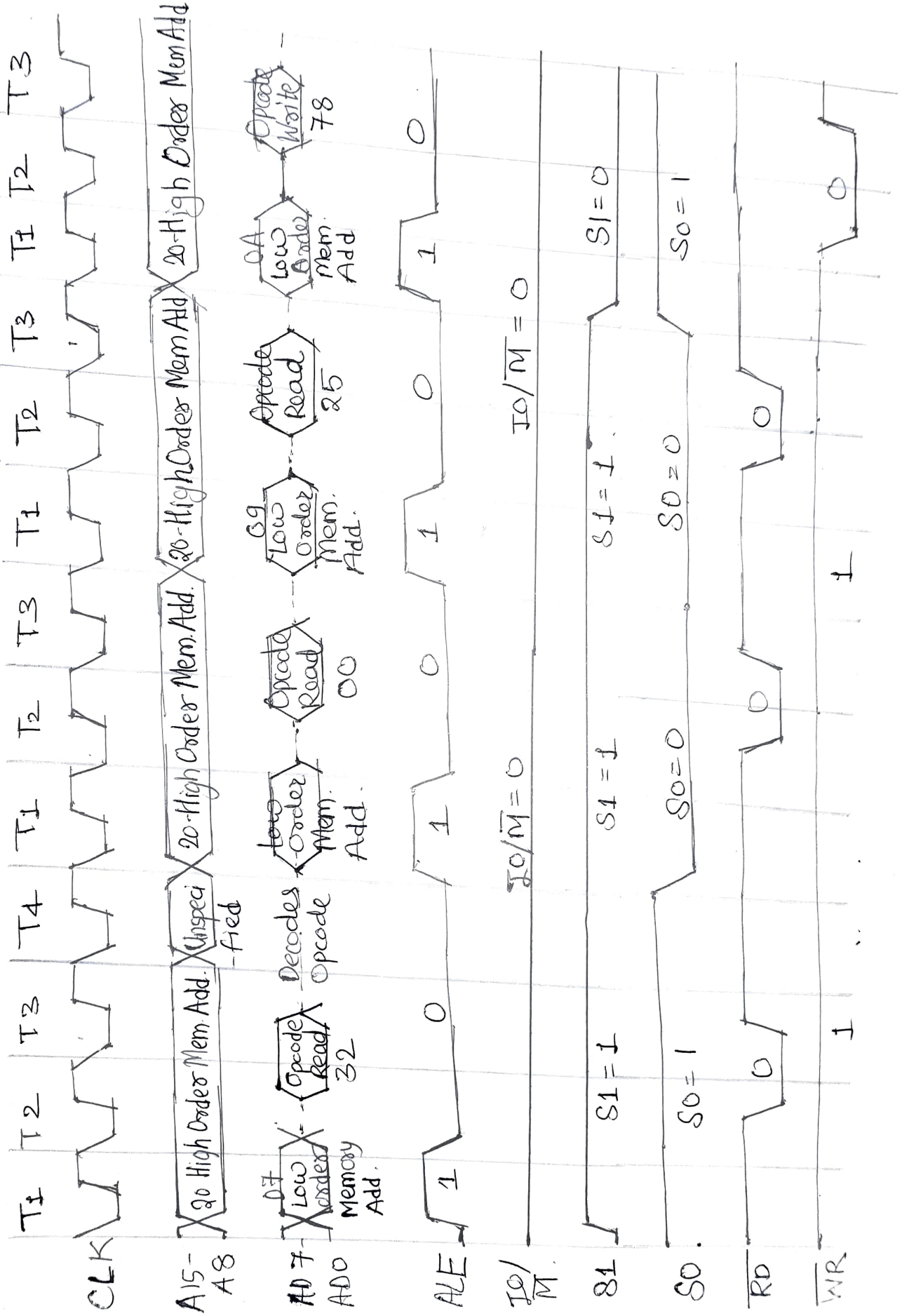
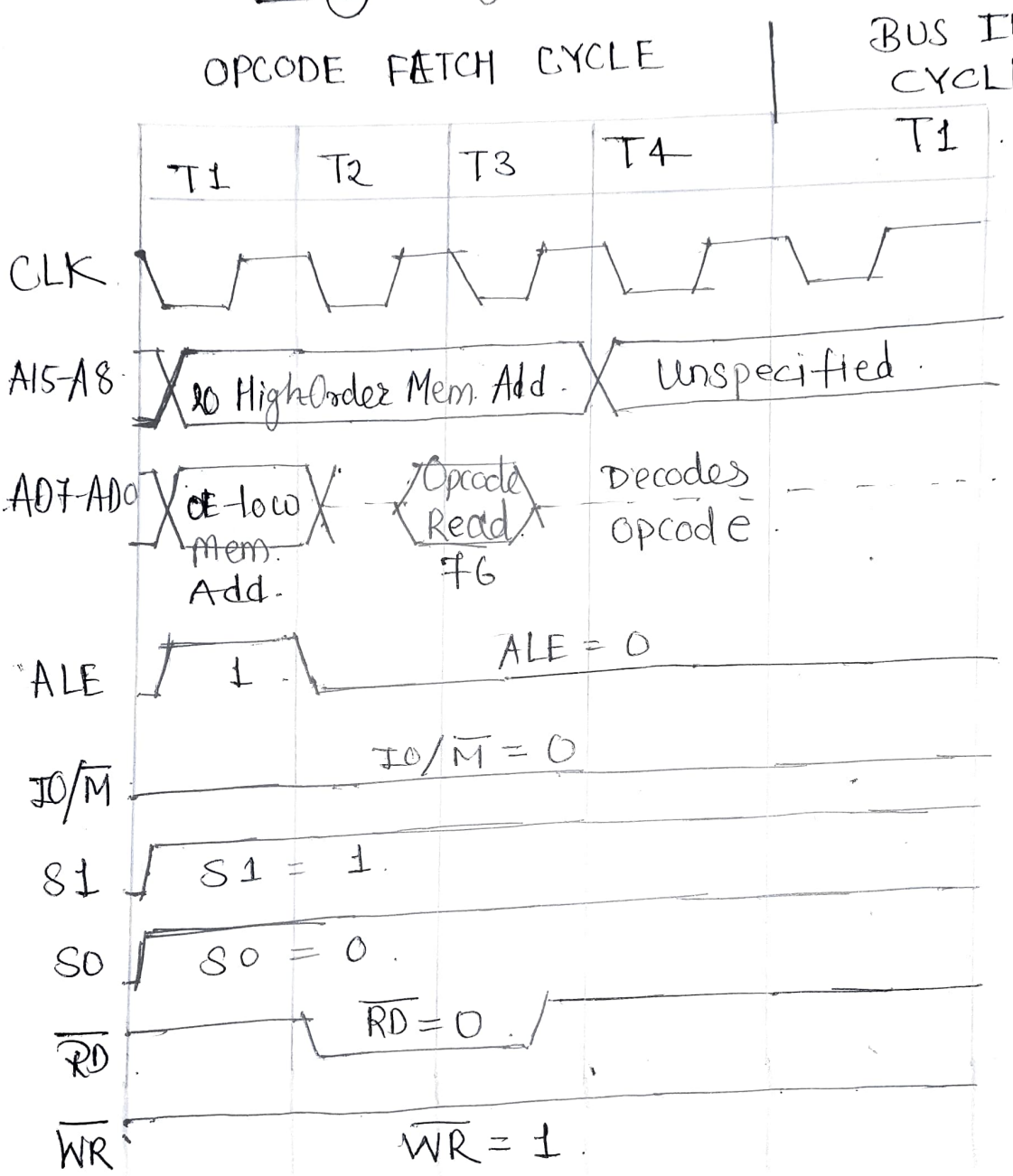


Fig. Timing diagram of STA 2500.

Instruction :- HLT (1 Byte Instruction)

WHAT it does :- STOP program execution.

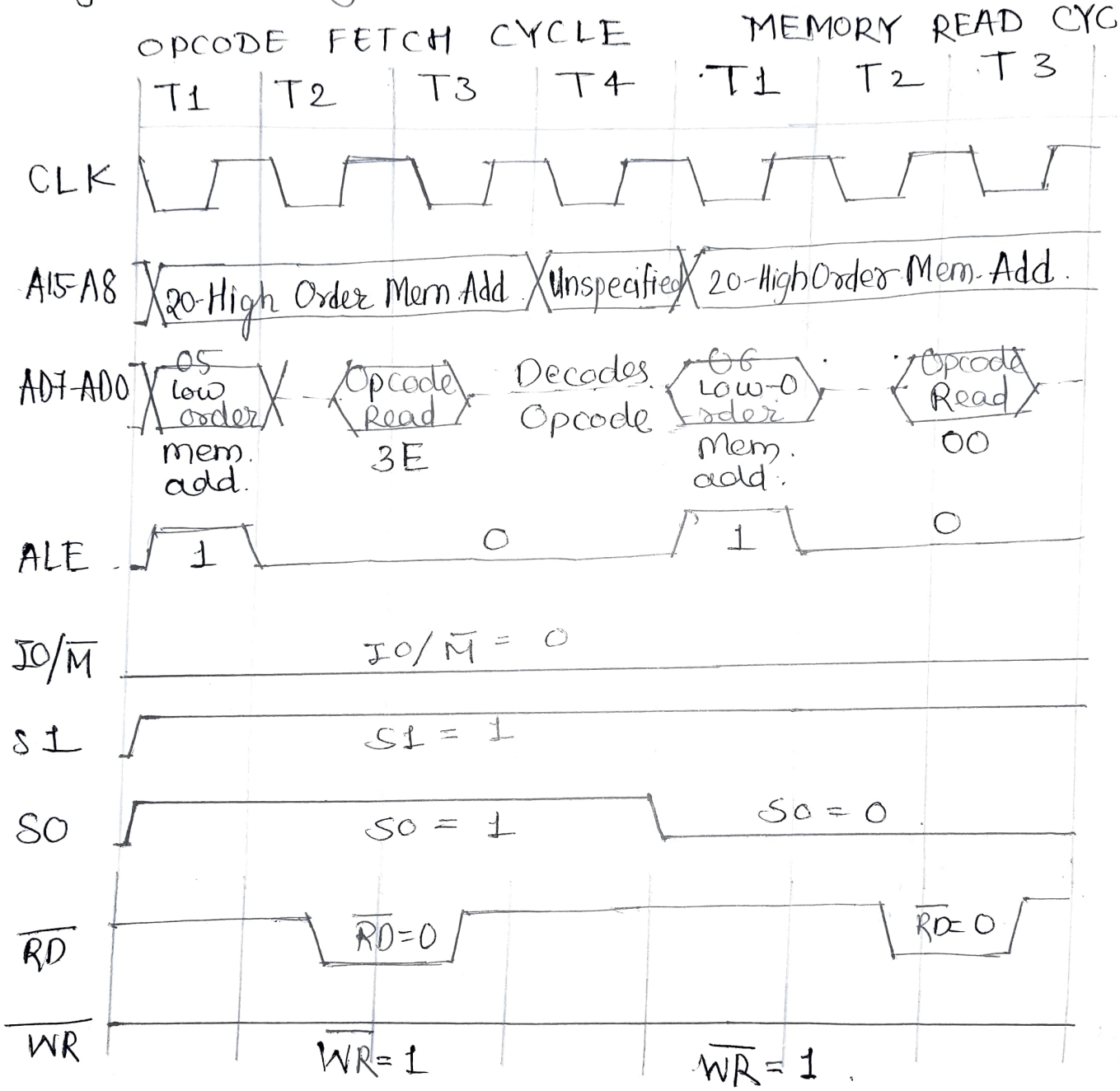
Timing Diagram of HLT.



Instruction 5 :- MVI r, d8 (8 bit data) (2-Byte Instruction)

What it does :- loads register r with 8 bit data value.

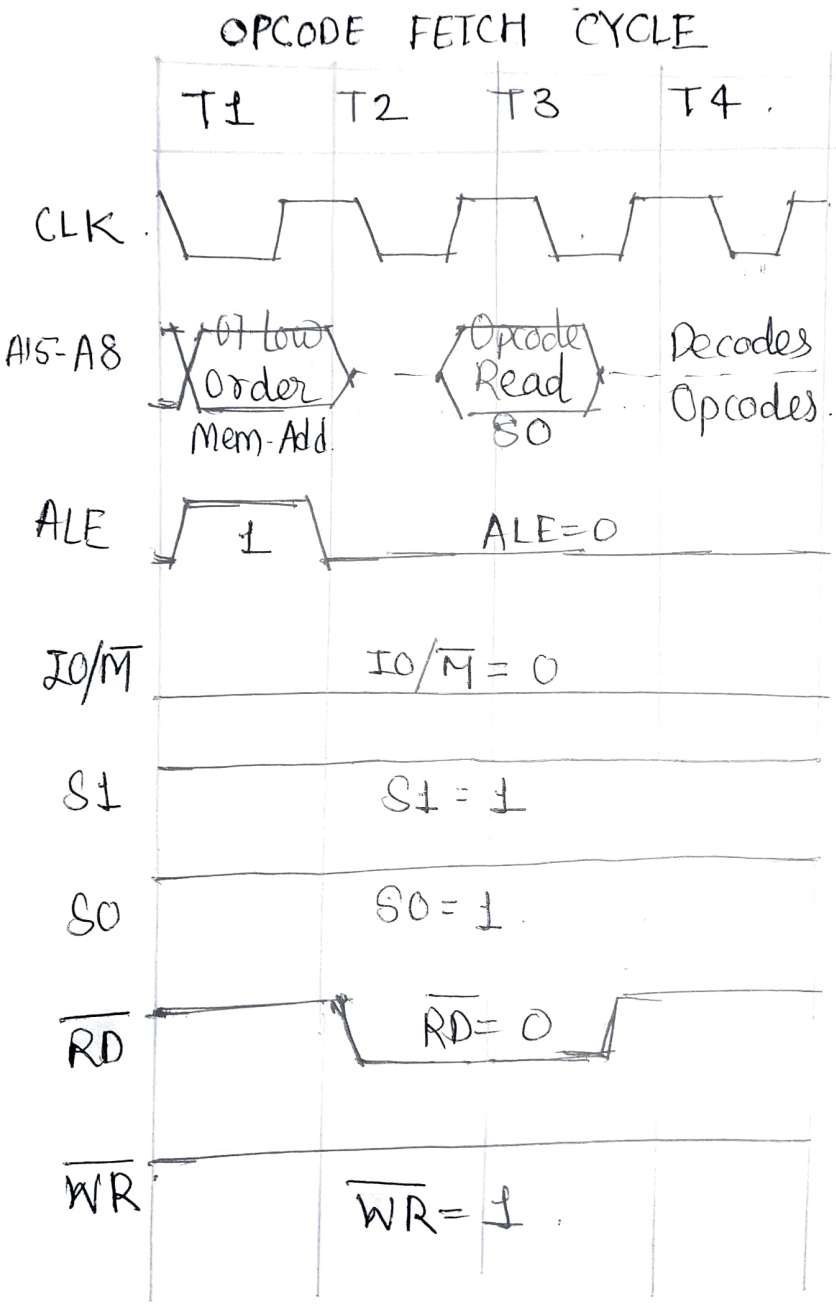
eg. Timing diagram of MVI A, 00.



Instructions :- ADD R. (1 Byte instruction).

What it does :- Add contents of R to Accumulator.

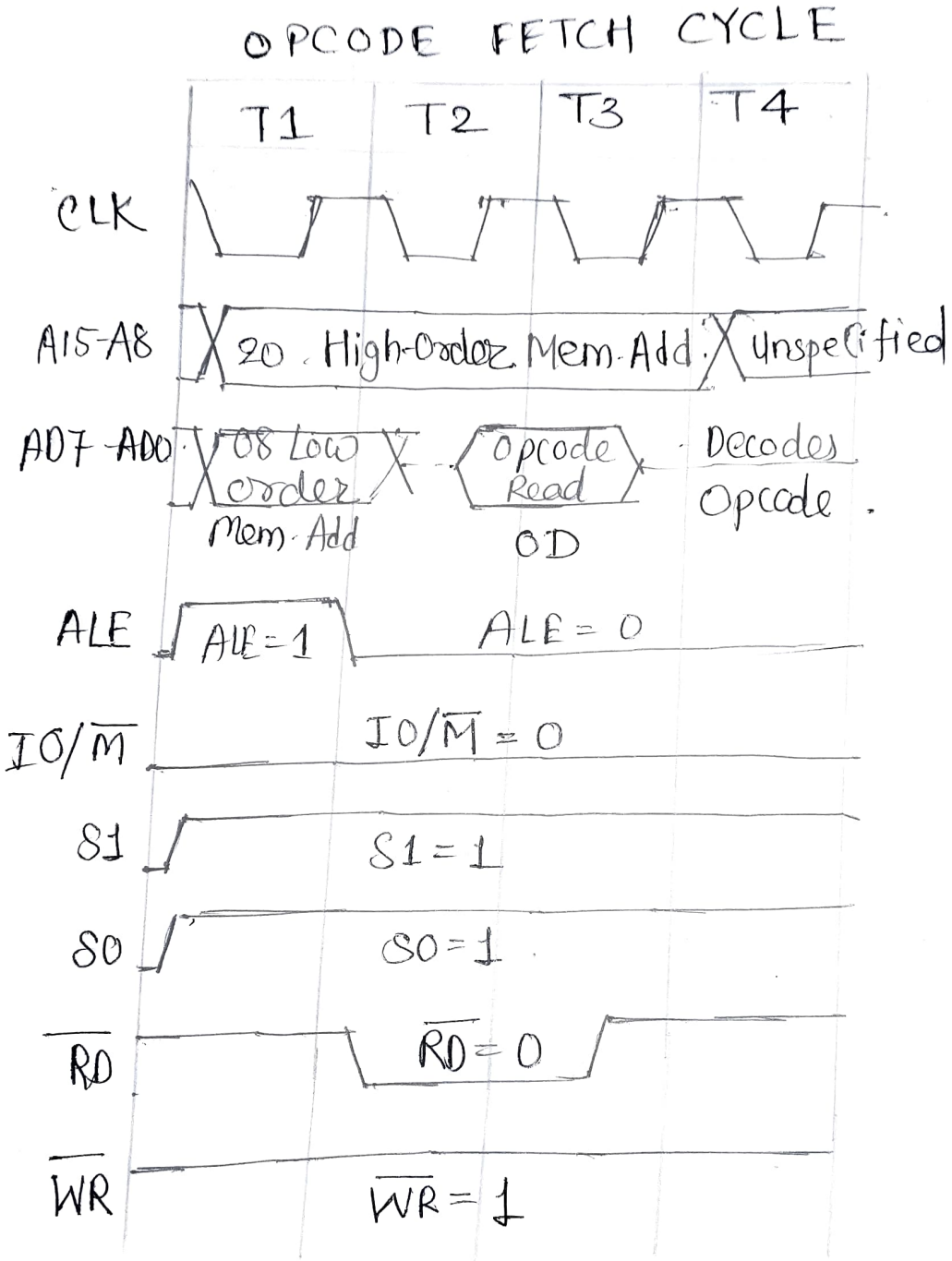
eg. Timing Diagram of ADD B.



Instruction : DCR R (1-Byte Instruction)

What it does : decrease content of R by 1.

eg. Timing diagram of DCR C.



Instruction 8 :- CMP R (1 Byte Instruction)

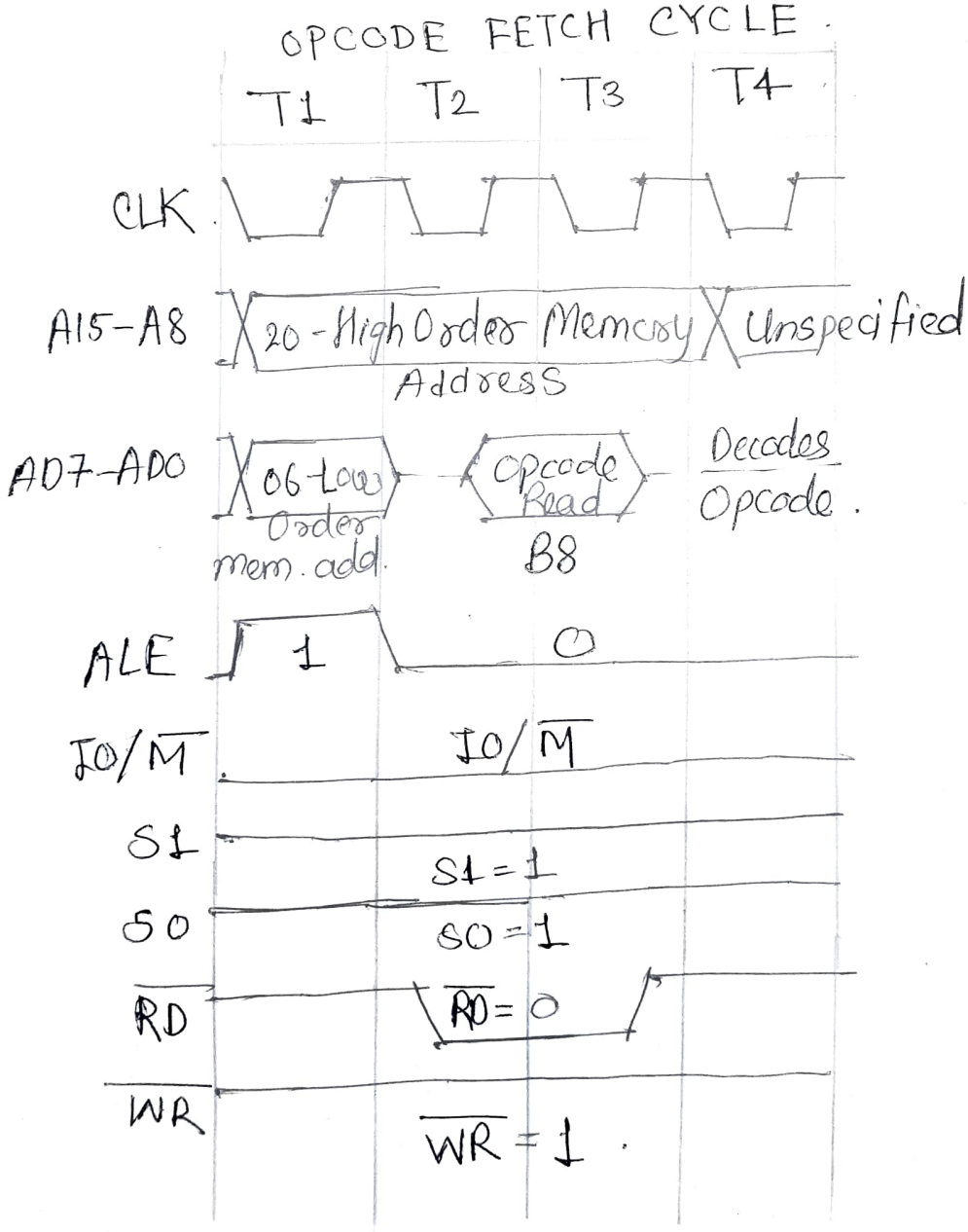
What it does :- Compare content in R with accumul
- ator A.

If A is less than R, then Cy (Carry flag) is set
and Z (zero flag) is reset.

A equals to R, then Cy (Carry flag) is reset
and Z (zero flag) is set.

A greater than R, then Cy (Carry flag) is reset
Z (zero flag) is reset.

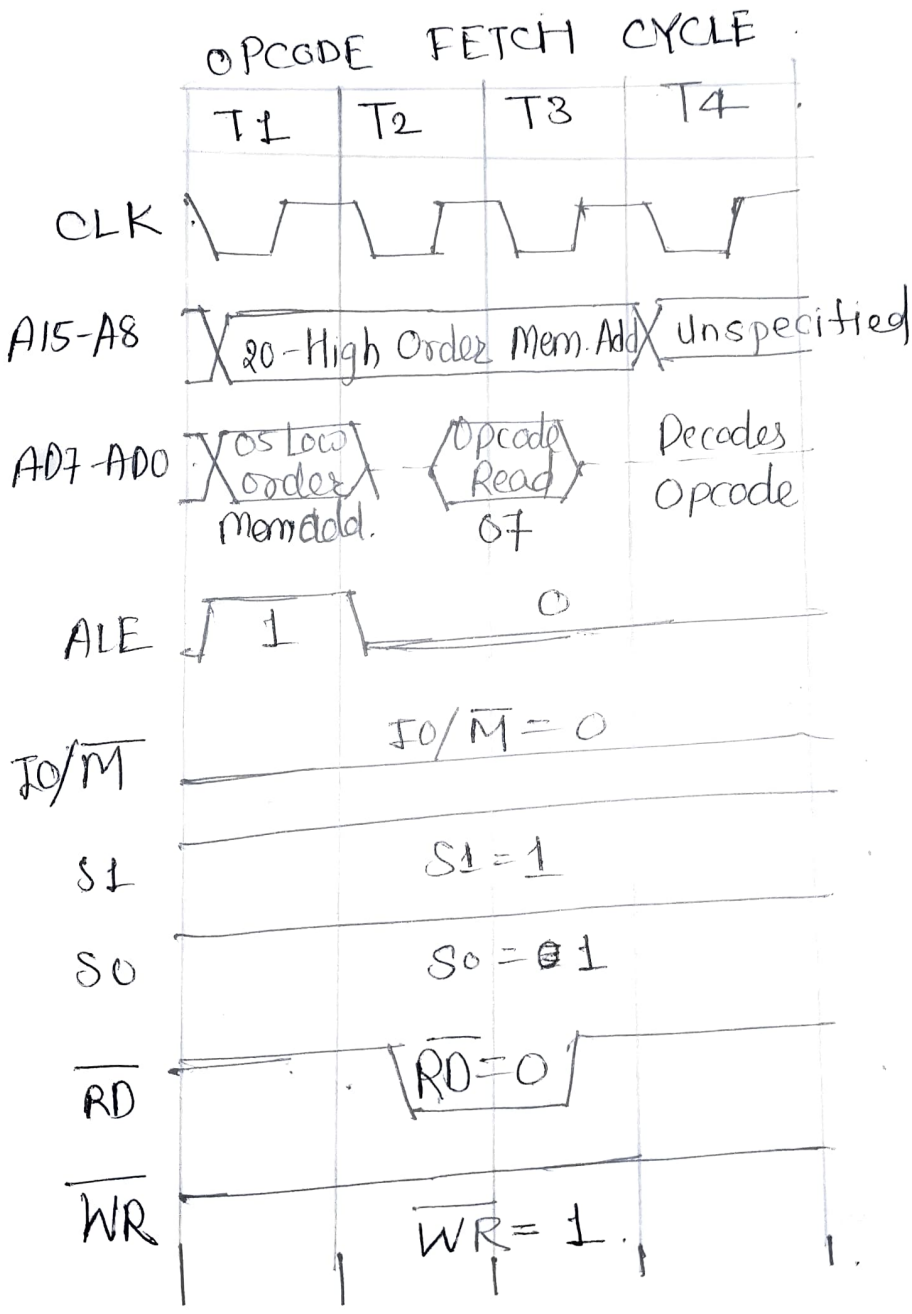
eg. Timing diagram of CMP B.



Instruction g :- RLC (1 Byte Inst.) Rotate Left Accumulator

What it does :- Rotates the accumulator contents to the left by 1-bit position.

eg. Timing Diagram of RLC



Instruction 10 :- LXI rp d16. (3-Byte Instruction)

rp ⇒ register pair d16 ⇒ 16-bit address

WHAT it does :- used to load 16-bit address (d16) into the register pair (rp).

eg. Timing diagram of LXI H,3000.