CS-322 Lab-2

Name: P.V.SRIRAM
Roll No: 1801CS37

(I)

(Q1) 8085 simulator program for Hexadecimal to Decimal number conversion.

Address	Label	Mnemonics	Hexcode - operation	Comment
F000		LXI H, 8000	21	Initialize
F001			00	memory
F002			80	Pointer
F003		MVI D, 00	16	Clear D-reg
F004			00	for MSB
F005		XRA A	AF	Clear Accumula-tor
F006		MOV C, M	4E	Get HEX data
F007	LOOP	ADI 01	C6	Increase A
F008			01	by 1
F009		DAA	27	Adjust for BCD
F00A		JNC SKIP	D2	Jump if not
F00B			0E	carry, to SKIP
F00C			F0	
F00D		INR D	14	Increase D
F00E	SKIP	DCR C	0D	Decrease C
F00F		JNZ LOOP	C2	Jump if not
F010			07	zero to Loop
F011			F0	
F012		MOV L, A	6F	Load LSB
F013		MOV H, D	62	Load MSB
F014		SHLD 8050	22	Store the BCD
F015			50	
F016			80	
F017		HLT	76	Terminate

Q1)

	Value	Addr
Input :	34	(8000)
Output :	52	(8050)

Q2) 8085 simulator program for decimal to Hexadecimal conversion

Address	Label	Mnemonics	OPERAND	OP CODE	COMMENTS
2000 2001 2002		LDA	201F	3A 1F 20	Load value from 201F to A
2003		MOV	B, A	47	Move from A to B
2004 2005		ANI	0F	E6 0F	Bitwise AND of A with 0F
2006		MOV	C, A	4F	Move A to C
2007		MOV	A, B	78	Move B to A
2008 2009		ANI	F0	E6 F0	Bitwise AND of A to F0
200A 200B 200C		JZ	SKIPMULTIPLY	CA 1A 20	Jump is zero to MULTIPLY
200D		RRC	~~0F~~	0F	Shift Right
200E		RRC		0F	
200F		RRC		0F	
2010		RRC		0F	
2011		MOV	D, A	57	Move A to D
2012		XRA	A	AF	Bitwise Reset
2013 2014		MVI	E, 0A	1E 0A	Store a value 0A in E
2015	SUM	ADD	D	82	Add D to A
2016		DCR	E	1D	Decrement E by 1
2017 2018 2019		JNZ	SUM	C2 15 20	Jump if not zero to Sum
201A	SKIP MUL	ADD	C	81	Add C to A
201B 201C 201D		STA	2020	32 20 20	Store value from A to 2020
201E		HLT		76	Terminate

Q2) Input : 72 (201F)

output : 48 (2020)

Q3) 8085 simulator program to find the sum of odd numbers in an input array.

Address	Label	Mnemonics	Operand	OPCODE	COMMENTS
0000 0001 0002		LDA	2200	3A 00 22	Load into A from 2200
0003		MOV	C, A	4F	Initialize Counter
0004 0005 0006		LXI H	H, 2201	21 01 22	Initialize pointer
0007 0008		MVI	E, 00	1E 00	Sum low=0
0009		MOV	D, E	53	Sum high = 0
000A	BACK	MOV	A, M	7E	Get num from Mem
000B 000C		ANI	01	E6 01	Mask bit 1-7
000D 000E 000F		JZ	SKIP	CA 17 00	Dont add if num is even
0010		MOV	A, E	7B	Get lower byte
0011		ADD	M	86	sum = Sum + data
0012		MOV	E, A	5F	Store result in E
0013 0014 0015		JNC	SKIP	D2 17 00	Jump if not carry to SKIP
0016		INR	D	14	Add carry to MSB
0017	SKIP	INX	H	23	Increment pointer
0018		DCR	C	0D	Decrement counter
0019 001A 001B		JNZ	BACK	C2 0A 00	Check if counter 0, repeat.
001C		MOV	A, E	7B	Move From E to A
001D 001E 001F		STA	2300	32 00 23	Store lower byte
0020		MOV	A, D	7A	Move from D to A
0021 0022 0023		STA	2301	32 01 23	Store high byte
0024		HLT		76	Terminate

Q3)

Input

Val	Addr
04	2200
9A	2201
52	2202
89	2203
3F	2204

Output

Val	Addr
C8	2300

Q4) 8085 simulator program to find the sum of even numbers in an input array.

Address	Label	Mnemonics	Operand	Opcode	Comments
0000		LDA	2200	3A	Move value from 2200 to A
0001				00	
0002				22	
0003		MOV	C, A	4F	Initialize count
0004		MVI	B, 00	06	Sum = 0
0005				00	
0006		LXI	H, 2201	21	Initialize Pointer
0007				01	
0008				22	
0009	BACK	MOV	A, M	7E	Get the no.
000A		ANI	01	E6	Mask bit 1-7
000B				01	
000C		JNZ	SKIP	C2	Dont add if num is odd
000D				12	
000E				00	
000F		MOV	A, B	78	Get the sum
0010		ADD	M	86	sum = sum + data
0011		MOV	B, A	47	store ans in B
0012	SKIP	INX	H	23	Increment Point
0013		DCR	C	0D	Decrease count
0014 - 0016		JNZ	BACK	C2, 09, 00	if count 0, repeat
0017		MOV	A, B	78	Store in A
0018 - 001A		STA	2210	32, 10, 22	Store sum
001B		HLT		76	Terminate

Q4)

Input

Val	Addr
04	2200
20	2201
15	2202
13	2203
22	2204

Output

Val	Addr
42	2210

Q5) 8085 simulator program to sort input array in ascending order.

Address	Label	Mnemonics	Operand	Opcode	Comments
0000 0001 0002		LXI	H, 4200	21 00 42	Move from Memory Pointer
0003		MOV	C, M	4E	Move from M to C
0004		DCR	C	0D	Decrement C
0005	REPEAT	MOV	D, C	51	Move C to D
0006 0007 0008		LXI	H, 4201	21 01 42	Memory pointer to 4201
0009	LOOP	MOV	A, M	7E	Move M to A
000A		INX H	H	23	Point to next loc
000B		CMP	M	BE	Compare M to A
000C 000D 000E		JC	SKIP	DA 14 00	Jump if Carry to SKIP
000F		MOV	B, M	46	Move M to B
0010		MOV	M, A	77	Move A to M
0011		PCX	H	2B	Prev location
0012		MOV	M, B	70	Move B to M
0013		INX	H	23	Back to loc
0014	SKIP	PCR	D	15	Decrement D
0015 0016 0017		JNZ	LOOP	C2 09 00	Jump if not Zero to LOOP
0018		DCR	C	0D	Decrement D
0019 001A 001B		JNZ	REPEAT	C2 05 00	Jump if not Zero to REPEAT
001C		HLT		76	Terminate

Q5)

Input

Val	Addr
06	4200
06	4201
05	4202
04	4203
01	4204
02	4205
03	4206

Output

Val	Addr
01H	4201
02	4202
03	4203
04	4204
05	4205
06	4206

Q6) 8085 simulator program for BCD to Hexadecimal number conversion.

Address	Label	Mnemonics	Operands	Opcode	Comments
0000		LXI	H, 4150	21	Memory Points to 4150
0001				50	
0002				41	
0003		MOV	A, M	7E	Move M to A
0004		ADD	A	87	MSD *2
0005		MOV	B, A	47	Move A to B
0006		ADD	A	87	MSD *4
0007		ADD	A	87	MSD *8
0008		ADD	B	80	MSD *10
0009		INX	H	23	Point to LSB
000A		ADD	M	86	Add to form HEX
000B		INX	H	23	Increment pointer
000C		MOV	M, A	77	Store result
000D		HLT		76	Terminate

Input:

Val Addr
02 (4150) MSB
09 (4151) LSB
Inp is 29H

Output: 1DH (4152)

Q7) 8085 simulator program for Hexadecimal to BCD conversion

Address	Label	Mnemonics	Operand	Opcode	Comments
0000		LXI	H, 4150	21	Memory pointer to 4150
0001				50	
0002				41	
0003		MVI	D, 00	16	Clear D-reg for MSB
0004				00	
0005		XRA	A	AF	Clear A
0006		MOV	C, M	4E	Get HEX inp
0007	LOOP2	ADI	01	C6	Increment A
0008				01	
0009		DAA		27	Adjust for BCD
000A		JNC	LOOP1	D2	Jump if not carry to Loop1
000B				0E	
000C				00	
000D		INR D	D	14	Increment D
000E	LOOP1	DCR	C	0D	Decrement C
000F		JNZ	LOOP2	C2	Jump if not zero to Loop2
0010				07	
0011				00	
0012		STA	4151	32	Store val in A to 4151
0013				51	
0014				41	
0015		MOV	A, D	7A	Move D to A
0016		STA	4152	32	Store val of A in 4152
0017				52	
0018				41	
0019		HLT		76	Terminate

Q7)

	Val	Addr	
Input :	FF	4150	
Output :	55	4151	- LSB
	02	4152	- MSB

Q8) 8085 simulator program to Transfer contents to overlapping memory blocks.

Address	Label	Mnemonics	Operand	Opcode	Comments
0000		MVI	C, FF	0E	Initialize
0001				FF	Counter
0002		LXI	H, 3005	21	Initialize
0003				05	source
0004				30	memory pointer
0005		LXI	D, 3008	11	Initialy
0006				08	destination
0007				30	memory pointer
0008	BACK	MOV	A,M	7E	Move M to A
0009		STAX	D	12	Store in dest ma
000A		DCX	H	2B	Decrement Point
000B		DCX	D	1B	Decrement Point
000C		DCR	C	0D	Decrement count
000D		JNZ	BACK	C2	If counter 0,
000E				08	repeat
000F				00	
0010		HLT		76	Terminate

Input

Val	Addr
01	3000
02	3001
03	3002
04	3003
05	3004
06	3005

Output

Val	Addr
01	3003
02	3004
03	3005
04	3006
05	3007
06	3008

Q9) 8085 program for masking of Lower, Higher nibbles of 8-bit num.

Address	Label	Mnemonics	Operand	Opcode	Comments
0000		LDA	2050	3A	Load from 2050 to A
0001				50	
0002				20	
0003		MOV	B, A	47	Move A to B
0004		ANI	0F	E6	Bitwise AND of A with 0F
0005				0F	
0006		STA	3050	32	Move A to STA
0007				50	
0008				30	
0009		MOV	A, B	78	Move B to A
000A		ANI	F0	E6	Bitwise AND of A with F0
000B				F0	
000C		RLC		07	~~Rotate~~ left
000D		RLC		07	
000E		RLC		07	
000F		RLC		07	
0010		STA	3051	32	Store A in 3051
0011				51	
0012				30	
0013		HLT		76	Terminate

Input

Val	Addr
64	(2050)

Output

Val	Addr
04H	(3050)
06H	(3051)

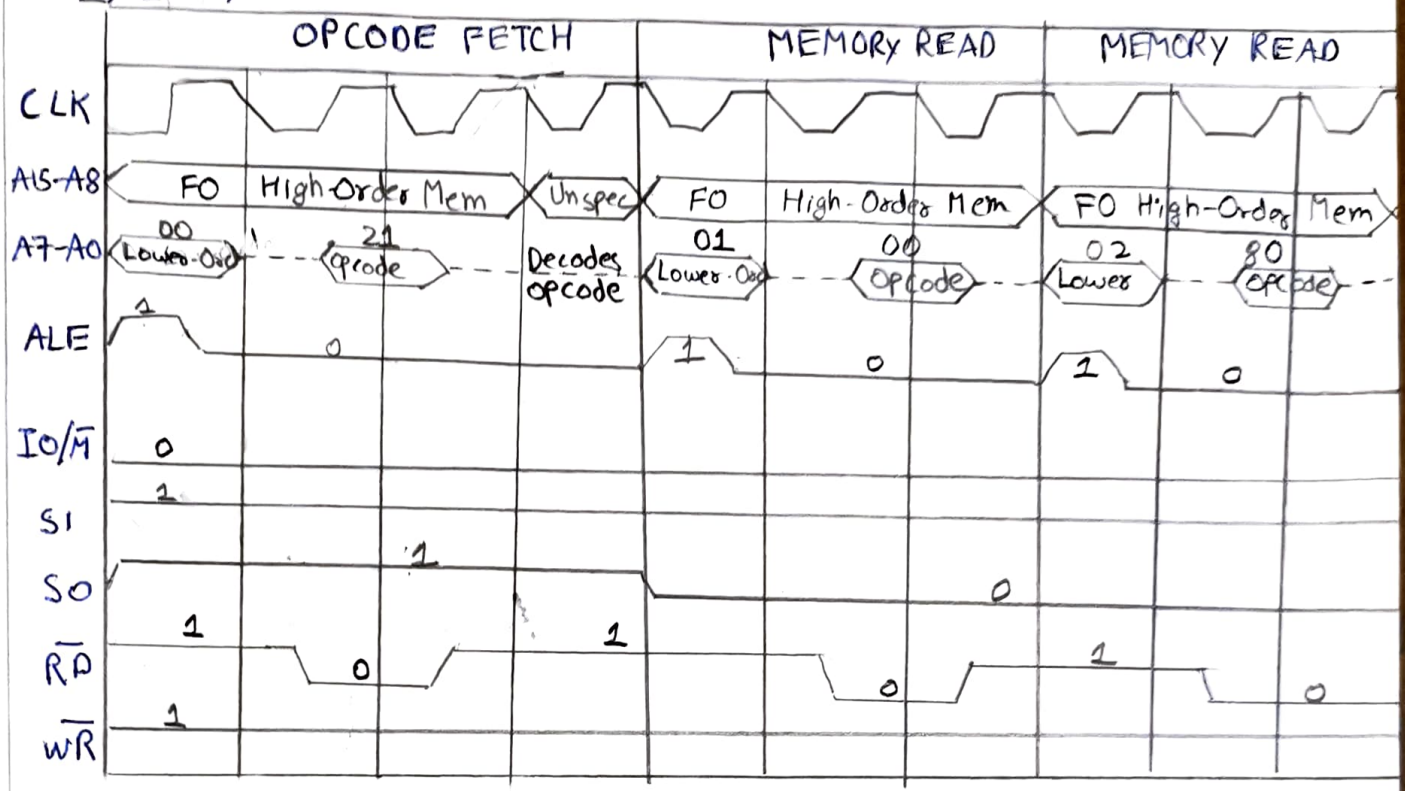
Q10) 8085 simulator program to find the sum of squares of first N natural numbers.

Address	Label	Mnemonics	Operand	Opcode	Comment
0000		LDA	201B	3A	Load val
0001				1B	from 201B to A
0002				20	
0003		MOV	B, A	47	Mov B → A
0004		ADD	B	80	Add B to A
0005		INR	A	3C	Incr A
0006		MOV	D, A	57	Move A → D
0007		MOV	A, B	78	Move B → A
0008		INR	A	3C	Incr A
0009		MOV	C, A	4F	Move A → C
000A		MVI	A, 00	3E	Init A to 0
000B				00	
000C	LOOP1	ADD	B	80	Add B to A
000D		DCR	C	0D	Dcr C
000E		JNZ	LOOP1	C2	Loop until
000F				0C	zero flag
0010				00	
0011		MOV	E, A	5F	Move A to E
0012		MVI	A, 00	3E	Init A to 0
0013				00	
0014	LOOP2	ADD	E	83	Add E to A
0015		DCR	D	15	Dcr d
0016		JNZ	LOOP2	C2	Loop until
0017				14	zero flag
0018				00	
0019		MVI	C, 06	0E	C val. for division
001A				06	
001B		MVI	B, 00	06	Init B to 0
001C				00	
001D	LOOP3	INR	B	04	Incr B
001E		SUB	C	91	Subt C from A
001F		JNZ	LOOP3	C2	Loop until
0020				1D	zero flag
0021				00	
0022		MOV	A, B	78	Move B → A
0023		STA	201C	32	store A value
0024				1C	in 201C
0025				20	
0026		HLT		76	Terminate

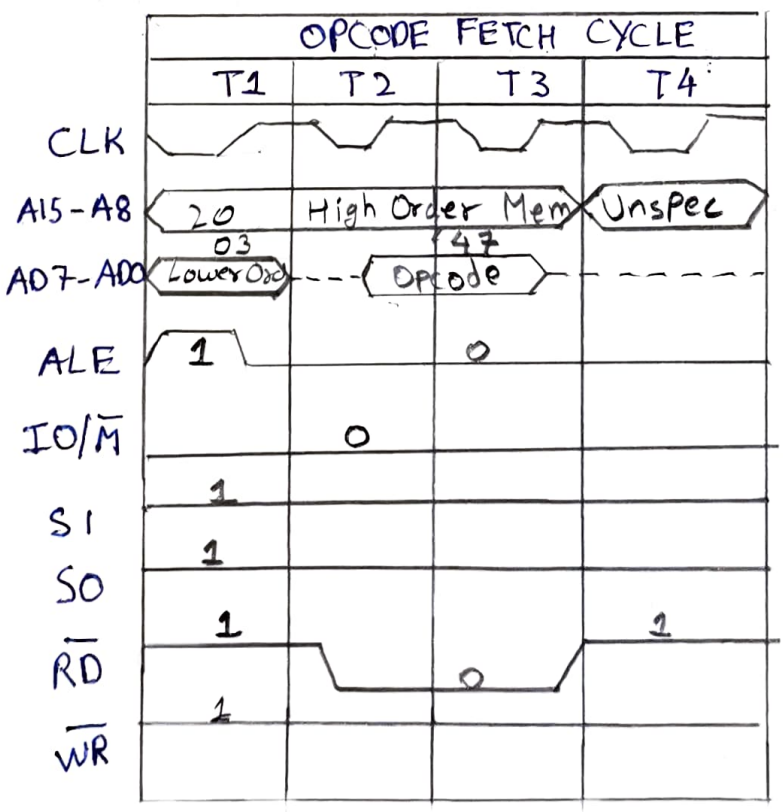
		Val	Addr
Q 10)	Input:	04	201B
	Output:	1E	201C

II.

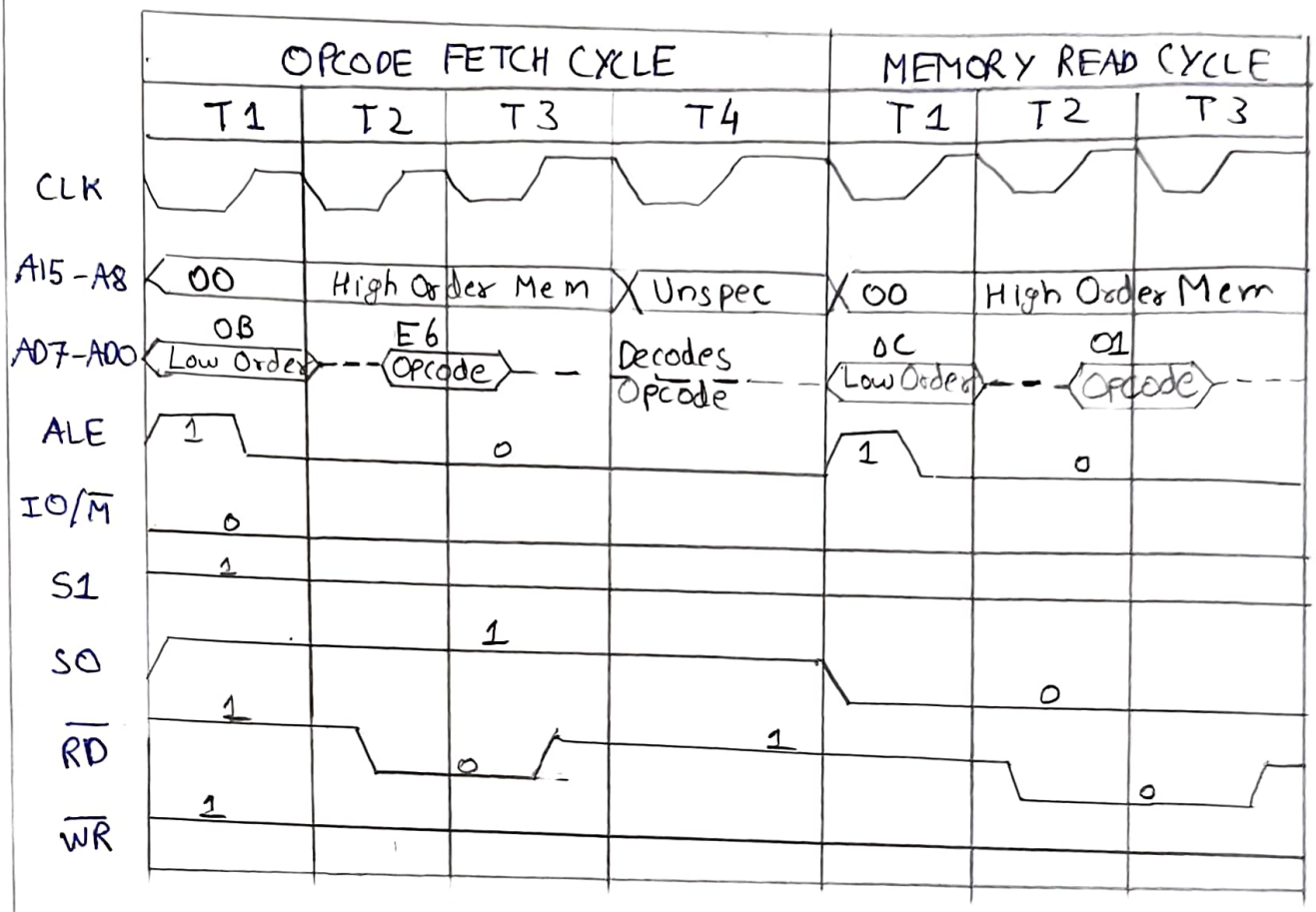
1) LXI H, 8000



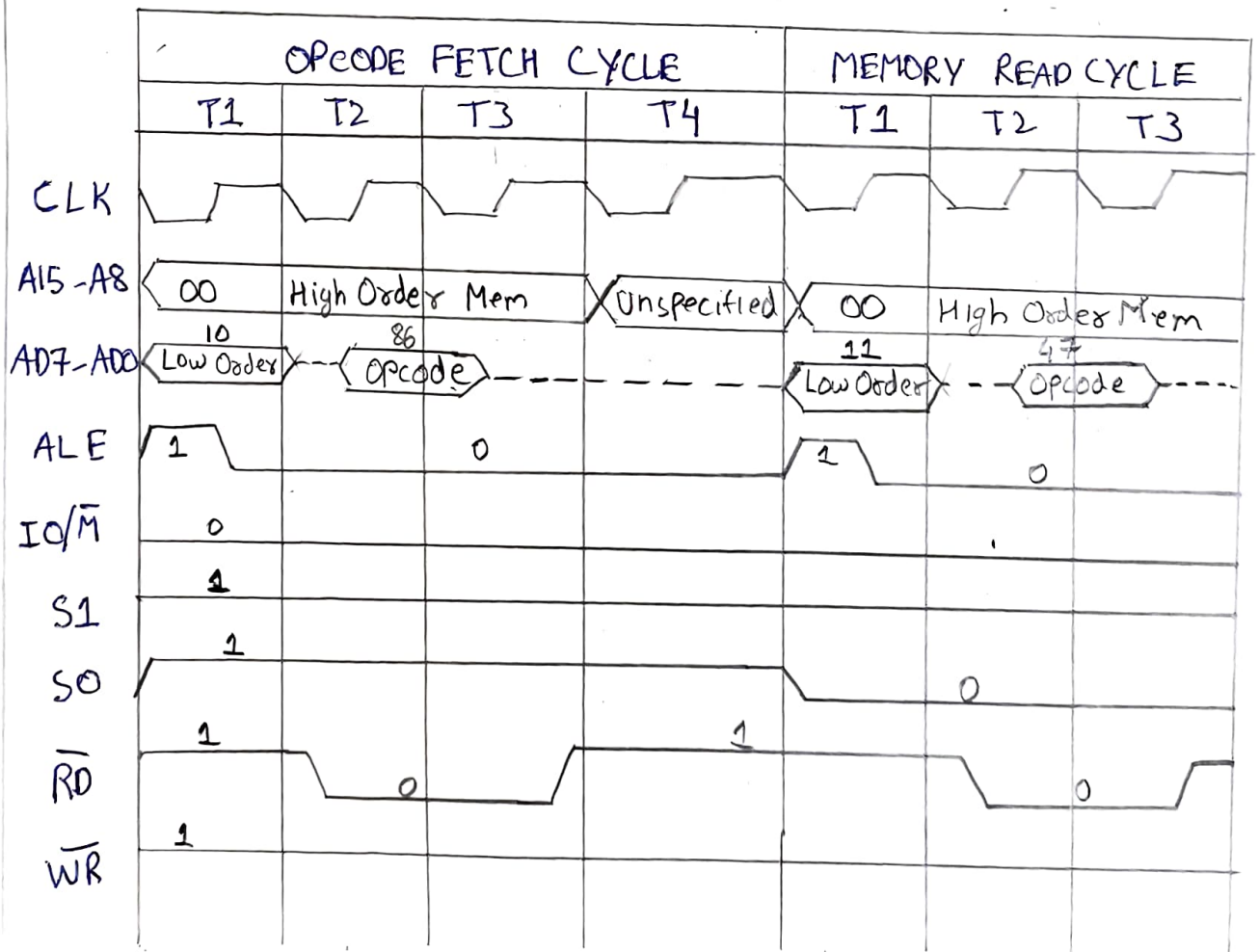
2) MOV B, A



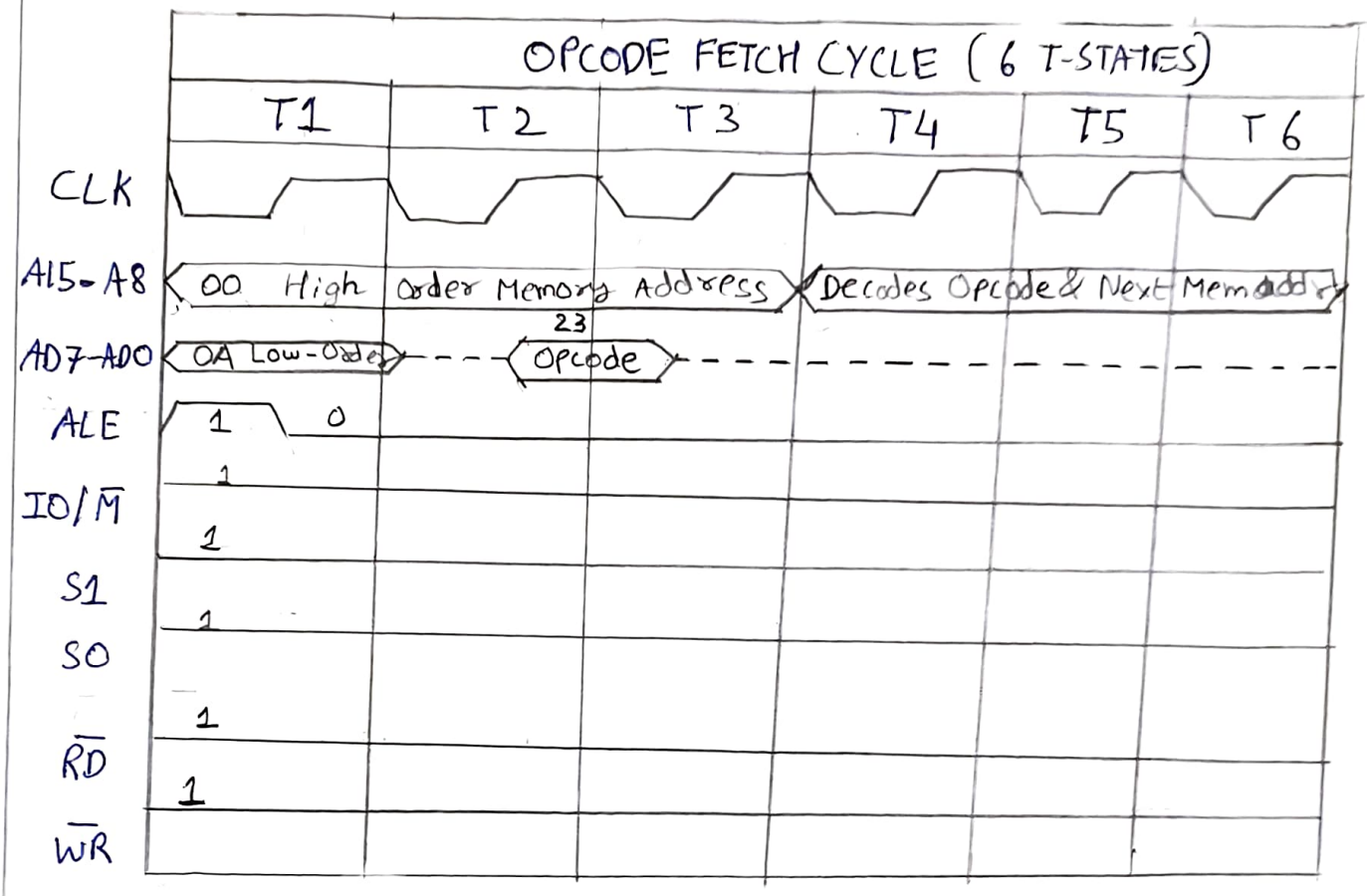
3) ANI 01



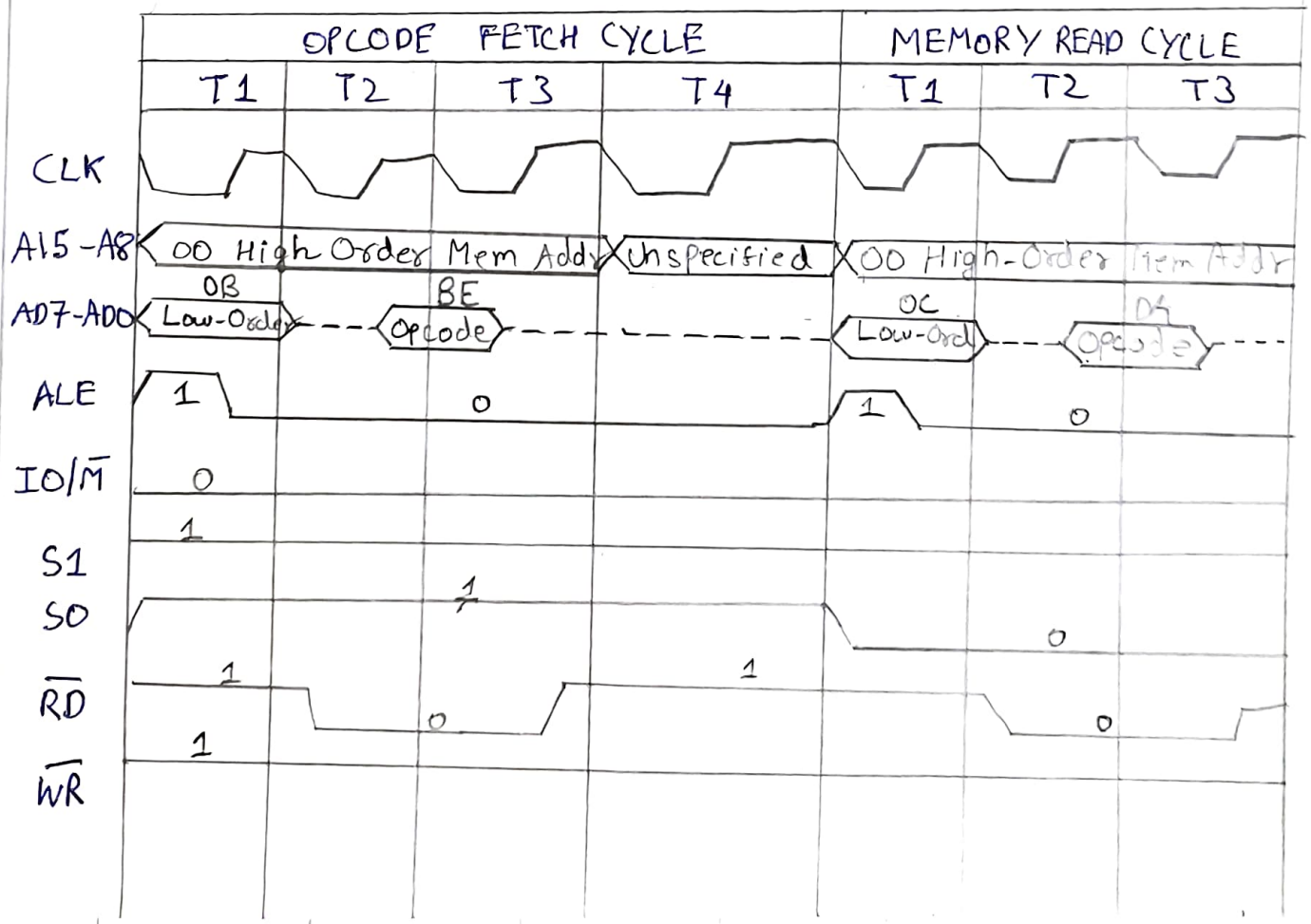
4) ADD M



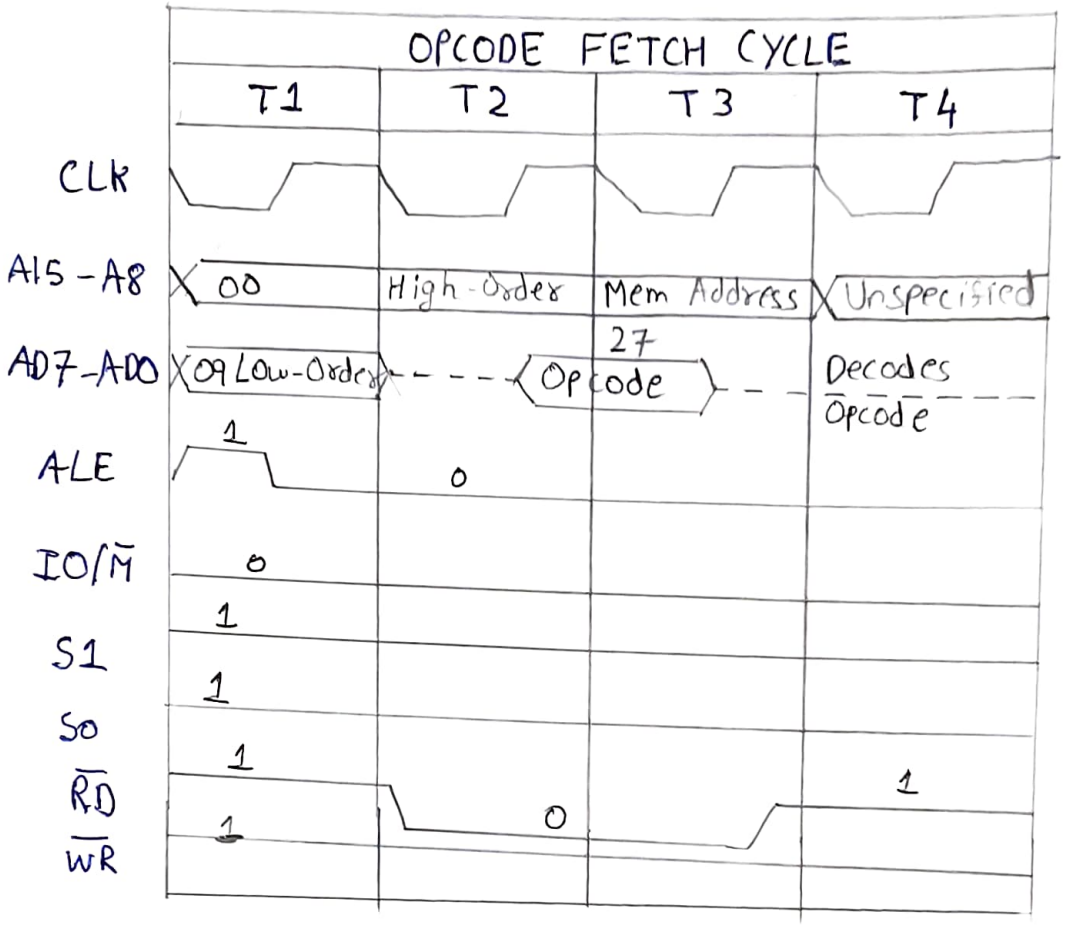
5) INX H


OPCODE FETCH CYCLE (6 T-STATES)
T1
T2
T3
T4
T5
T6
CLK
A15-A8
00 High Order Memory Address
Decodes Opcode & Next Mem addr
AD7-AD0
0A Low-Order
23
Opcode
ALE
1
0
IO/M
S1
SO
RD
WR

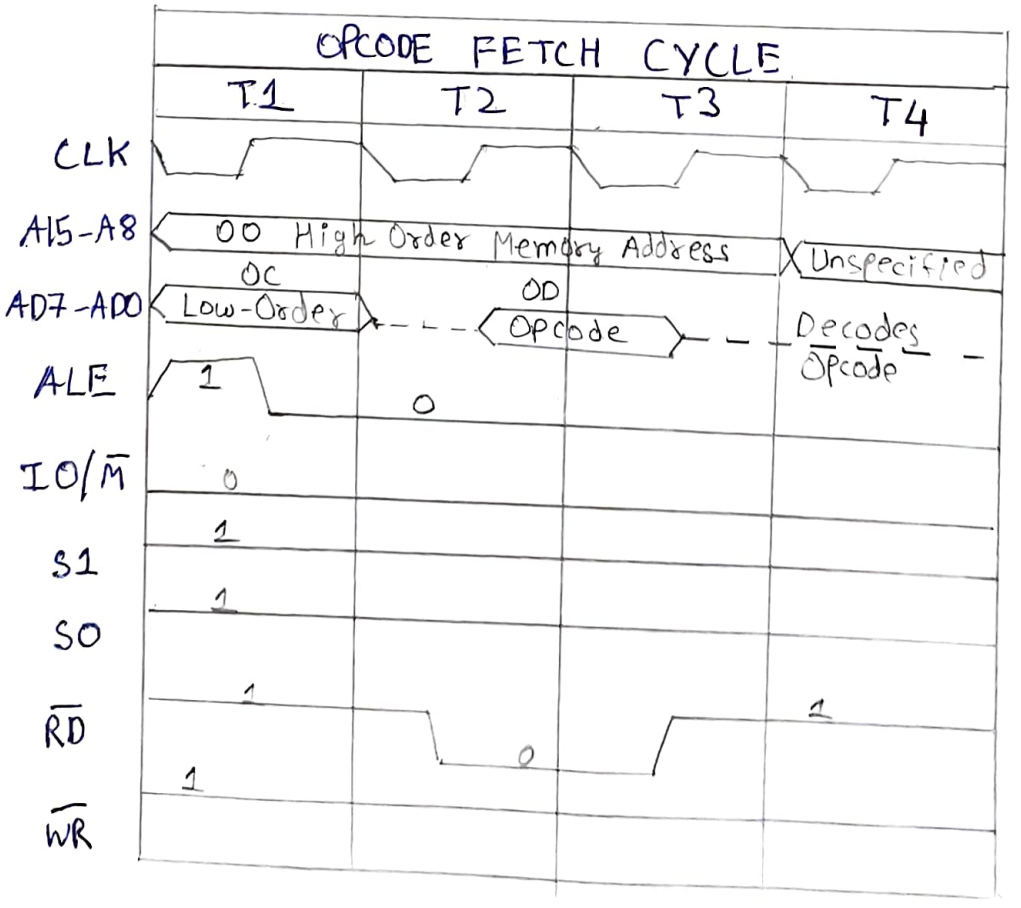
6) CMP M


OPCODE FETCH CYCLE
MEMORY READ CYCLE
T1
T2
T3
T4
T1
T2
T3
CLK
A15-A8
00 High Order Mem Addr
Unspecified
00 High-Order Mem Addr
AD7-AD0
0B
Low-Order
BE
Opcode
0C
Low-Ord
D4
ALE
IO/M
S1
SO
RD
WR

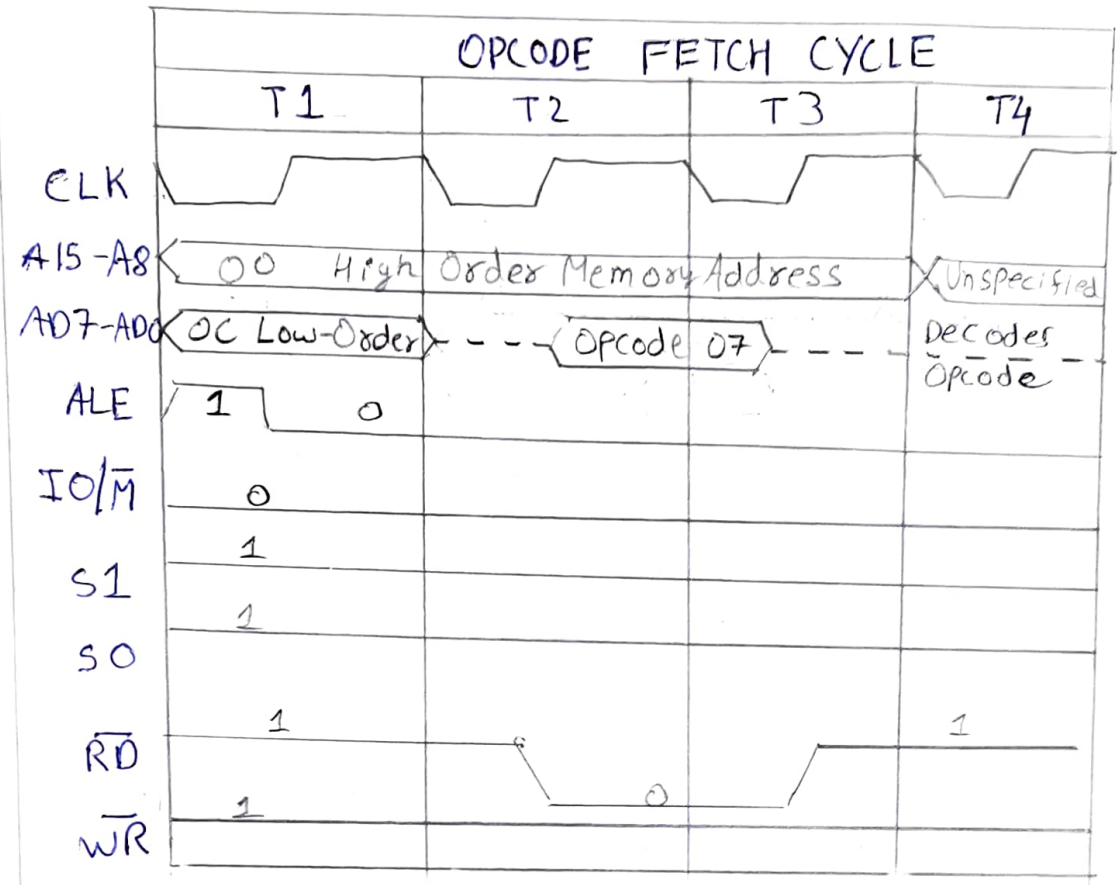
7) DAA


OPCODE FETCH CYCLE
T1
T2
T3
T4
CLK
A15 - A8
00
High-Order
Mem Address
Unspecified
AD7 - AD0
09 Low-Order
27
Opcode
Decodes
Opcode
ALE
1
0
IO/M
0
S1
1
SO
1
RD
1
0
1
WR
1

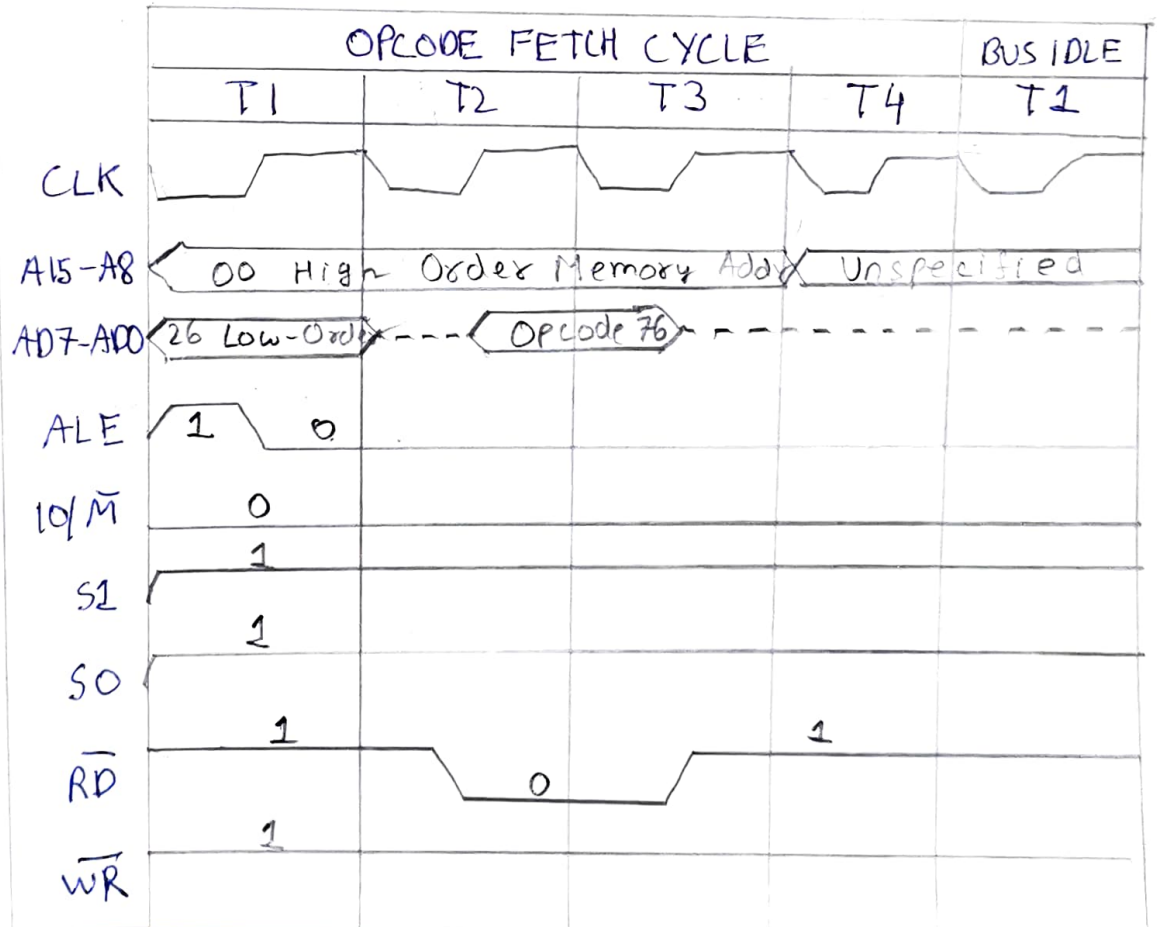
8) DCR C


OPCODE FETCH CYCLE
T1
T2
T3
T4
CLK
A15-A8
00 High Order Memory Address
Unspecified
AD7-AD0
0C
Low-Order
0D
Opcode
Decodes
Opcode
ALE
1
0
IO/M
0
S1
1
SO
1
RD
1
0
1
WR
1

9) RLC


OPCODE FETCH CYCLE
T1
T2
T3
T4
CLK
A15-A8
00 High Order Memory Address
Unspecified
AD7-AD0
0C Low-Order
Opcode 07
Decodes Opcode
ALE
1
0
IO/M
0
S1
1
S0
1
RD
1
1
0
WR
1

10) HLT


OPCODE FETCH CYCLE
BUS IDLE
T1
T2
T3
T4
T1
CLK
A15-A8
00 High Order Memory Add
Unspecified
AD7-AD0
26 Low-Ord
Opcode 76
ALE
1
0
IO/M
0
S1
1
S0
1
1
RD
0
1
WR
1