



INDEX

PARTICULARS OF THE EXPERIMENTS PERFORMED

DATE	SERIAL NO.	Experiment No.	SUBJECT / EXPERIMENT	PAGE NO.	SIGNATURE	REMARKS
18/01/18	1	1	Organization of mixed memory with latches.	1-7	S. Chakraborty 25/1/18	3/4 (3.5/4)
08/02/18	2	2	Memory testing	8-21	Smitra 15/2/18	4/4
08/03/18	3	3	Data transfer between MDR and RAM	22-23	Smitra 08/03/18	9/4
05/04/18	4	4	Swapping contents of register using micro program.	28-31	S. Chakraborty 05/4/18	3.0/4
12/04/18	5	5	Implementation of data movement instruction	35-38	S. Chakraborty 12/4/18	7/4 19/20

REPORT ON Experiment 1



REPORT ON

Experiment No. 1.....

Title ..Organization..of..mixed..memory..with..latches.....

Name. Shriti Saha, Tanisha Sarangi Date of Performance 4/01., 11/01., 18/01

Semester 4th Date of Submission 25/01/18

Branch. CST..... Roll No. 61x-23, 27..... Signature. A. A. A.,.....

Examined by S. Chakraborty 25/11/18

Computer Architecture and Organization Laboratory
Department of Computer Science and Technology, IIEST

Experiment No: 1 (Organization of $(m \times d)$ memory with latches)

Objective: To build memory module with the features of Read/Write, D_{in}/D_{out} using RS latch considering

- (a) $m = 1, d = 1$,
- (b) $m = 2, d = 1$
- (c) $m = 2, d = 2$ - using the design resulted out of (b)

Utilize the following ICs:

1. RS latch (74279),
2. Tri-state logic (74367)
3. Triple-input NAND (7410),
4. Inverter (7404)

Realize the logic circuits shown in Figure 1 and verify the designs as per the following tables.

TABLE I
VERIFICATION FOR DESIGN (a)

Sl no	Select	R/W	Data in [supply]	Data out [verify]	Activity
1	1.	0	1	-	Write 1 in Loc-1
2	1	1	-	1	Read 1 from Loc-1
3	0	x	-	-	No operation
4	1	0	0	-	write 0 in Loc-1
5	1	1	-	0	Read 0 from Loc-1

$\bar{R} \quad \bar{S} \quad Q$

TABLE II
VERIFICATION FOR DESIGN (b)

No	Select	R/W	D_{in}	D_{out}	Verify
1	1	0	1 (d_1)	-	Write 1 in Loc-1
2	0	0	1 (d_2)	-	Write 1 in Loc-0
3	1	1	-	1 (d_1)	Read 1 from Loc-1
4	0	1	-	1 (d_2)	Read 1 from Loc-0
5	1	0	0	-	Write 0 in Loc-1
6	0	0	1	-	Write 1 in Loc-0
7	1	1	-	0	Read 0 from Loc-1
8	0	1	-	1	Read 1 from Loc-0
9	1	0	1	-	Write 1 in Loc-1
10	0	0	0	-	Write 0 in Loc-0
11	1	1	-	1	Read 1 from Loc-1
12	0	1	-	0	Read 0 from Loc-0

$\bar{R} \quad \bar{S} \quad Q$

TABLE III
VERIFICATION FOR DESIGN (c)

No	Select	R/W	$D1_{in}$	$D0_{in}$	$D1_{out}$	$D0_{out}$	Verify
1	1	0	1	0	-	-	Write 10 in Loc-1
2	0	0	0	1	-	-	Write 01 in Loc-0
3	1	1	-	-	1	0	Read 10 from Loc-1
4	0	1	-	-	0	1	Read 01 from Loc-0
5	1	0	0	0	-	-	Write 00 in Loc-1
6	0	0	1	1	-	-	Write 11 in Loc-0
7	1	1	-	-	0	0	Read 00 from Loc-1
8	0	1	-	-	1	-1	Read 11 from Loc-0

M = word memory
 α = No. of bits

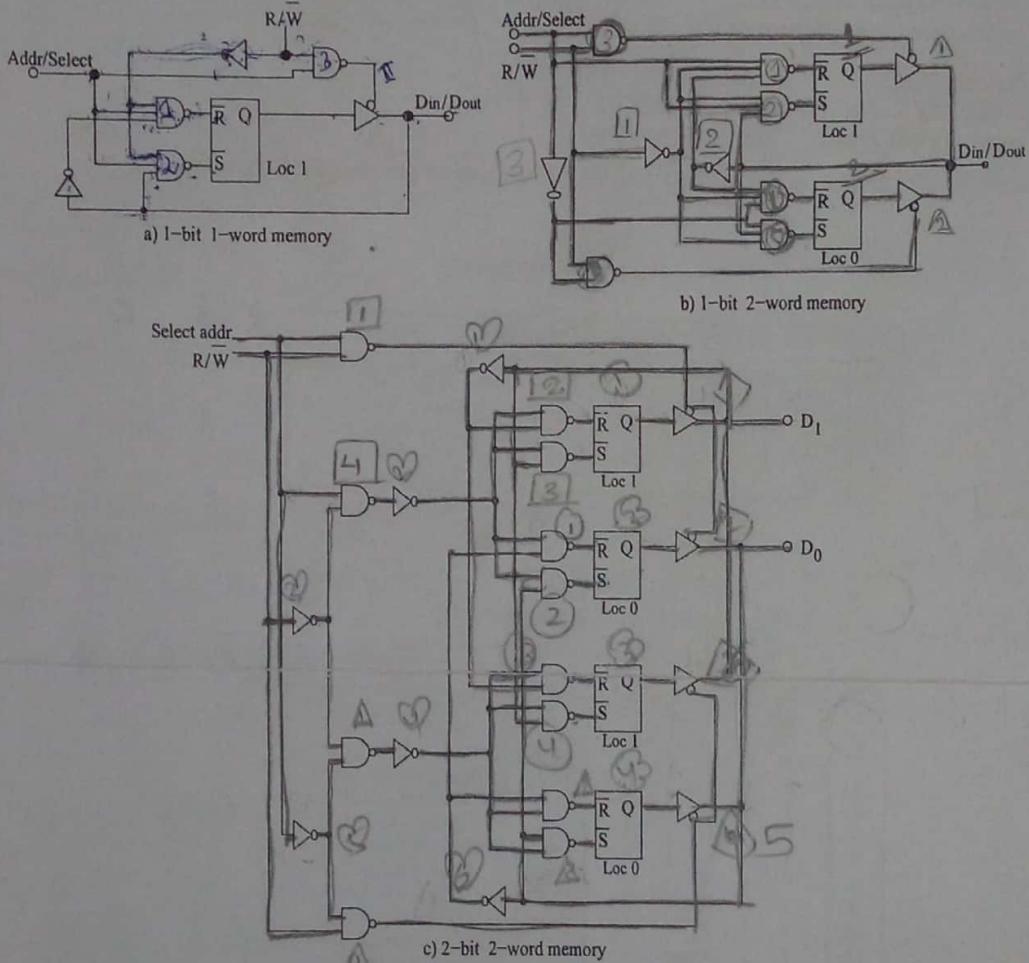


Fig. 1. Circuit diagram

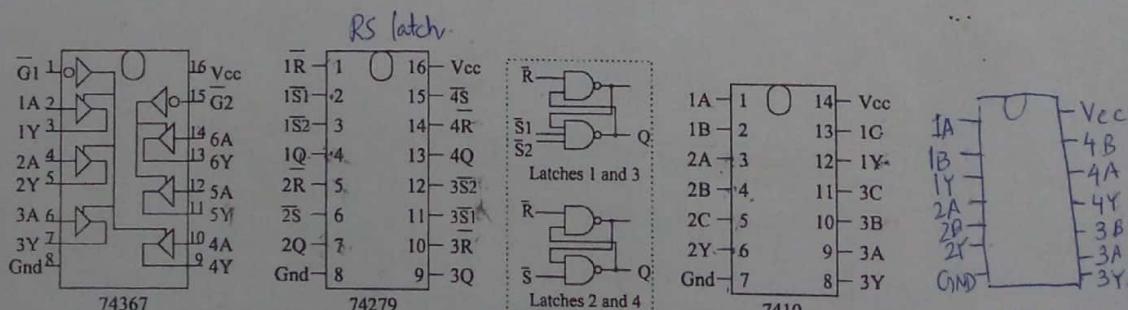


Fig. 2. Chip configurations



REPORT ON Experiment No. 1

ORGANIZATION OF mxd MEMORY WITH LATCHES

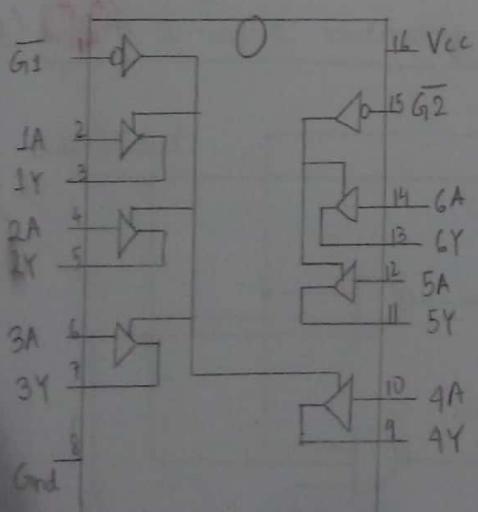
Objective: To build memory module with the features of Read/Write, Din/Dout using RS latch considering -

- (a) $m=1, d=1$
- (b) $m=2, d=1$
- (c) $m=2, d=2$ using the design resulted out of (b)

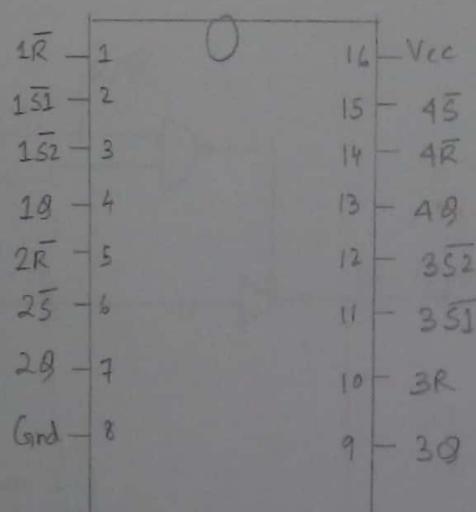
Requirements: We utilise the following ICs:

- 1. RS latch (74279)
- 2. Tri-state logic (74367)
- 3. Triple - input NAND (7410)
- 4. Inverter (7404)
- 5. Two- input NAND (7400).

PIN DIAGRAMS :



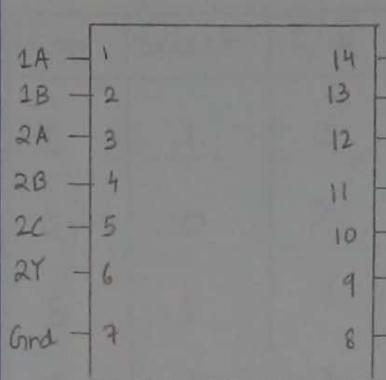
74367



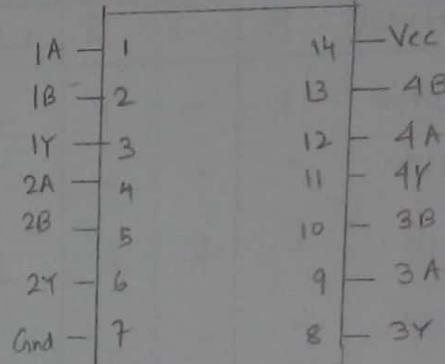
74279



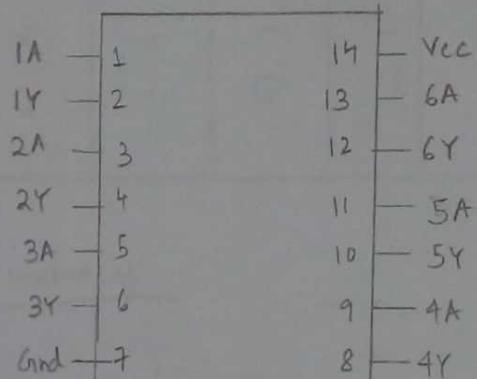
REPORT ON



7410

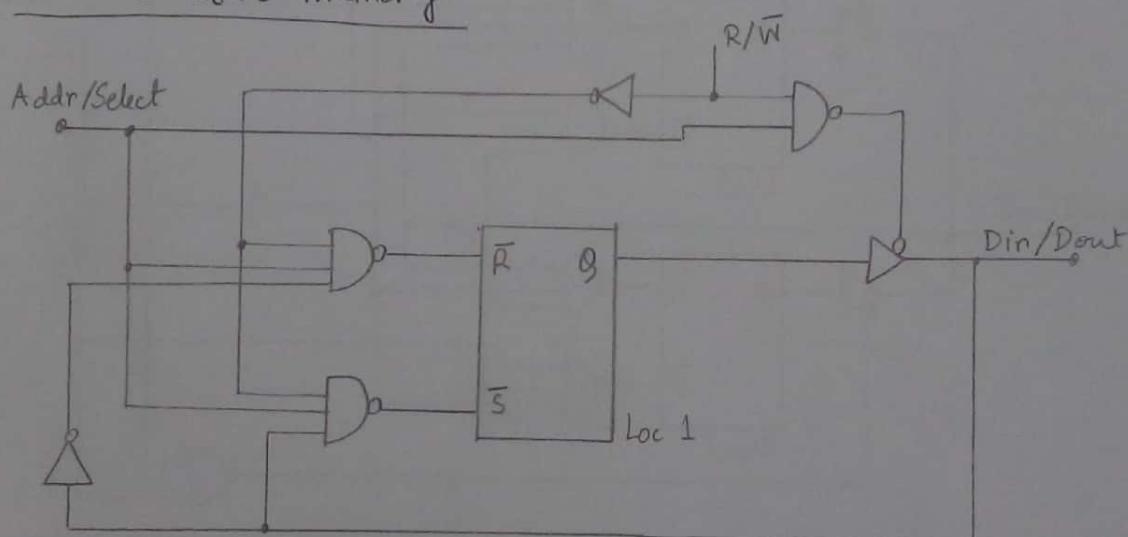


7400



7404

(a) 1-bit 1-word memory



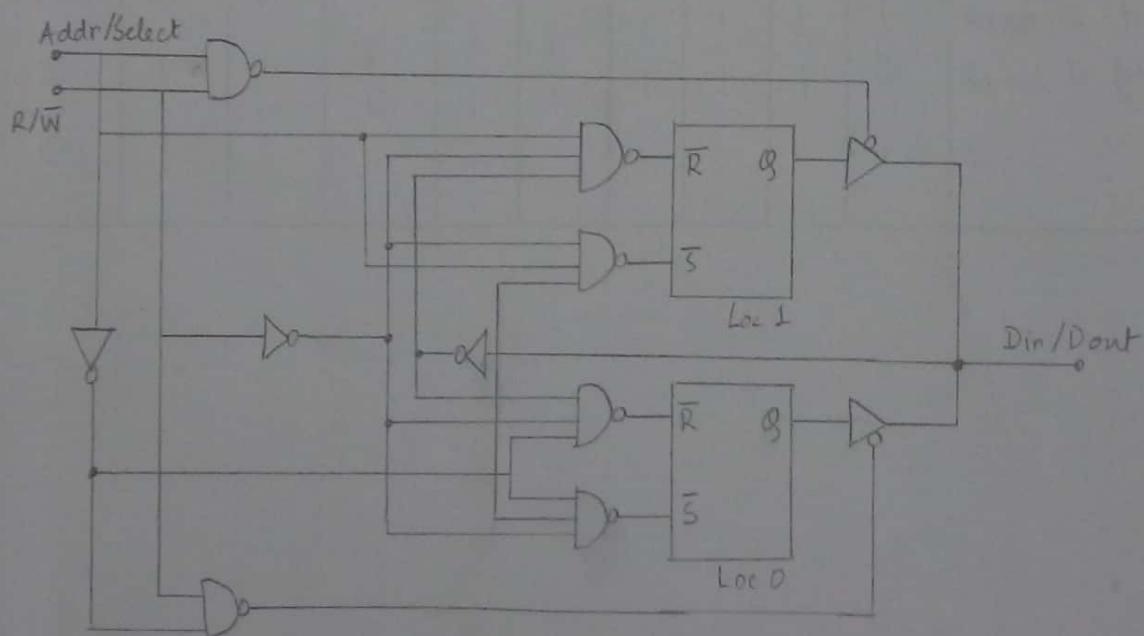


REPORT ON

Table I : Verification for design (a)

Sl.no.	Select	R/ \bar{W}	Din	Dout	\bar{R}	\bar{S}	\varnothing	Verification
1	1	0	1	-	1	0	1	Write 1
2	0	X	-	-	-	-	-	NOP
3	1	1	-	1	1	1	1	Read 1
4	0	X	-	-	-	-	-	NOP
5	1	0	0	-	0	1	0	Write 0
6	0	X	-	-	-	-	-	NOP
7	1	1	-	0	1	1	0	Read 0.

(b) 1-bit 2-word memory





REPORT ON

Table II: Verification for design (b)

Sl. No	Select	R/W	Din	Dout	Loc 1			Loc 0			Verification
					\bar{R}	\bar{S}	\bar{Q}	\bar{R}	\bar{S}	\bar{Q}	
1	1	0	1	-	1	0	1	1	1	0	Write 1 in Loc-1
2	0	0	1	-	1	1	1	1	1	1	Write 1 in Loc-0
3	1	1	-	1	1	1	1	1	1	1	Read 1 from Loc-1
4	0	1	-	1	1	1	1	1	1	1	Read 1 from Loc-0
5	1	0	0	-	0	1	0	1	1	1	Write 0 in Loc-1
6	0	0	1	-	1	1	0	1	0	1	Write 1 in Loc-0
7	1	1	-	0	1	1	0	1	1	1	Read 0 from Loc-1
8	0	1	-	1	1	1	1	1	1	1	Read 1 from Loc-0
9	1	0	1	-	1	0	1	1	1	1	Write 1 in Loc-1
10	0	0	0	-	1	1	1	0	1	0	Write 0 in Loc-0
11	1	1	-	1	1	1	1	1	1	1	Read 1 from Loc-1
12	0	1	-	0	1	1	1	1	1	0	Read 0 from Loc-0

Date 18/01.....

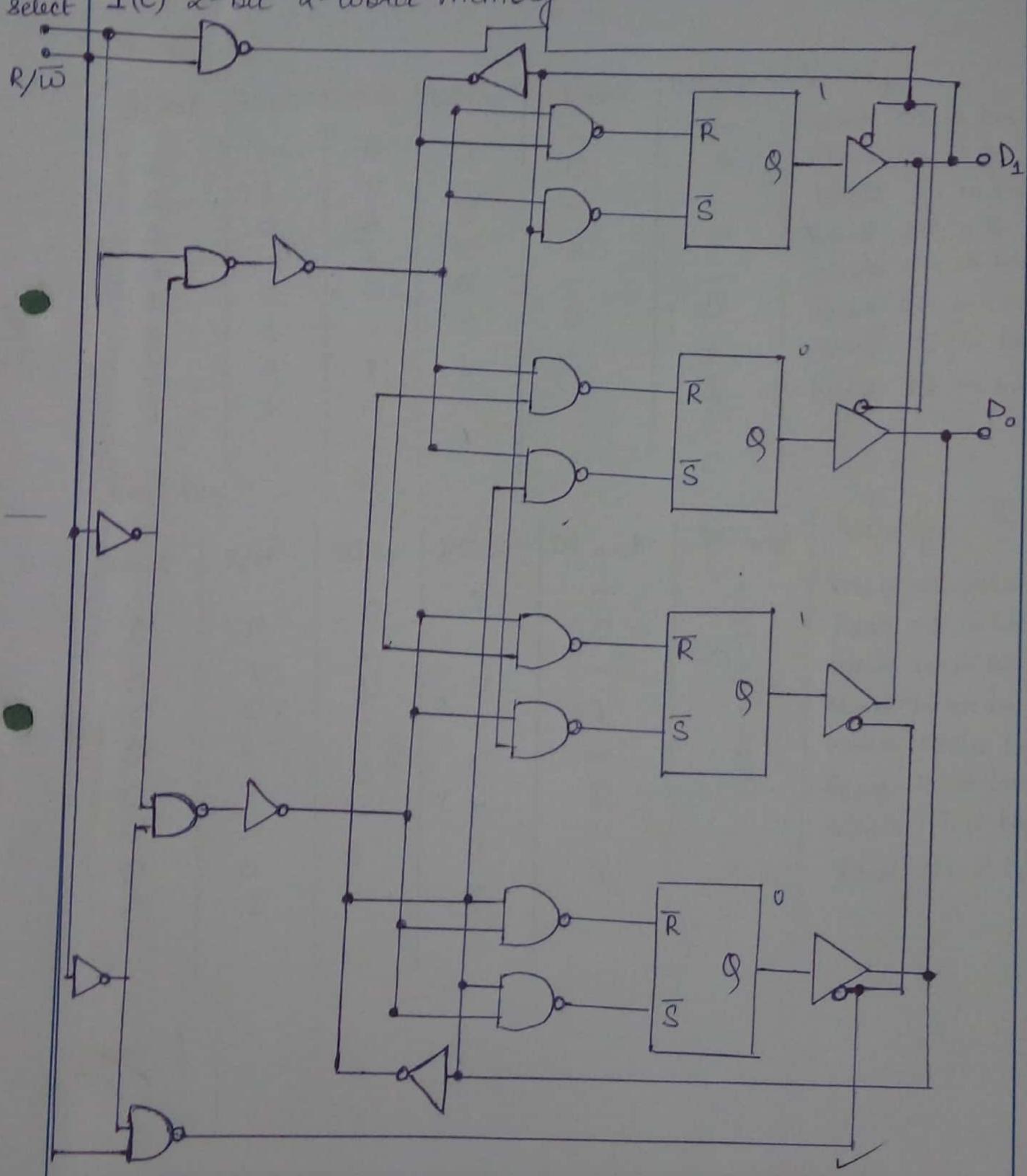
Page No. 4.....

Signature.....



REPORT ON

1(c) 2-bit 2-word memory





REPORT ON

Table III : Verification for design (c)

For Loc 1

Select	R/W	D1in	D0in	D1out	D0out	Verify
1	0	0	1	-	-	Write 01 in Loc-1
1	1	-	-	0	1	Read 01 in Loc-1
1	0	1	0	-	-	Write 10 in Loc-1
1	1	-	-	1	0	Read 10 in Loc-1
1	0	0	0	-	-	Write 00 in Loc-1
1	1	-	-	0	0	Read 00 in Loc-1
1	0	1	1	-	-	Write 11 in Loc-1
1	1	-	-	1	1	Read 11 in Loc-1

For Loc 0

Select	R/W	D1in	D0in	D1out	D0out	Verify
0	0	0	1	-	-	Write 01 in Loc-0
0	1	-	-	0	1	Read 01 in Loc-0
0	0	1	0	-	-	Write 10 in Loc-0
0	1	-	-	1	0	Read 10 in Loc-0
0	0	0	0	-	-	Write 00 in Loc-0
0	1	-	-	0	0	Read 00 in Loc-0
0	0	1	1	-	-	Write 11 in Loc-0
0	1	-	-	1	1	Read 11 in Loc-0



REPORT ON

Precautions -

- 1) The integrated circuit chips should be handled carefully in order to avoid damage.
- 2) It must be made sure that the wires do not touch each other to prevent short circuit.
- 3) All connections should be made proper and tight.

— * —

Indian Institute of Engineering Science And Technology, Shibpur
Department of ...C.S.T.....

REPORT ON Expt 2



REPORT ON

Experiment No. 2

Title ..Memory testing

Name. Dhriti Saha, Tanisha Sarawgi..... Date of Performance 25/01, 1/02, 8/02.

Semester. 4th..... Date of Submission..15./02./18.....

Branch....C.S.T..... Roll No 6ix-23, 27..... Signature #8.....

Examined by *S. M. Mitala 15/2/18*

WF CS

Computer Architecture and Organization Laboratory
Department of Computer Science and Technology, IIEST

Experiment No: 2 (Memory testing)

Objective: To become familiar with read/write operation of the memory and verification of the data written.

Utilize the following ICs:

1. RAM (6116)
2. 12-stage binary counter (CD 4040)
3. 4-bit comparator (7485).

The tasks

a) Perform data write to RAM (6116) and read data from the RAM (consider 4×2 memory as shown in Figure 1). Set address and data manually.

(i) Write to RAM:

1. Set $\overline{OE} = H$
2. Set $\overline{CS} = L$
3. Set address at A_1, A_0
4. Set data $I/O_2, I/O_1$
5. Set $\overline{WE} = L$
6. Set $\overline{WE} = H$. Go to Step 3 for write to next location.

(ii) Read from RAM:

1. Set $\overline{WE} = H$, and $\overline{CS} = L$.
2. Change address A_1, A_0 .
3. Set $\overline{OE} = L$.
4. Read data at $I/O_2, I/O_1$.

b) (i) Perform fast data write to RAM (6116) in l locations ($l < m$ of an $m \times n$ memory).

Generate memory address using counter (CD 4040). Consider 4-bit data. Data is input from the least significant 4-bit of the counter.

(ii) Perform data read from RAM. Compare the data read from memory and the data actually written (use comparator 7485) to verify the condition of memory.

c) Detect fault in RAM.

1. Perform fast data write to RAM, as in (b).
2. Insert fault to one or more locations of the RAM by force.
3. Perform data read from RAM. Compare the data read and the data actually written (use comparator 7485) to detect faulty RAM. As soon as the fault is detected at l_1 location, stop further data read.
4. Report on: A logic to start data read from the next memory location ($l_1 + 1$).

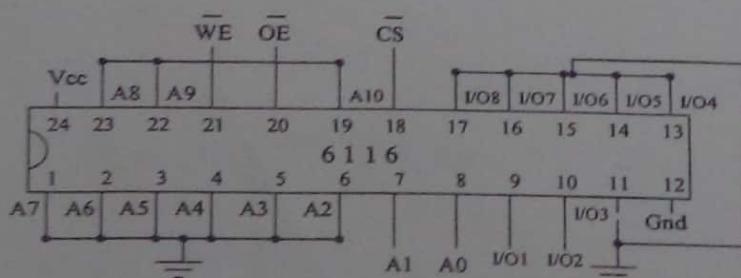


Fig. 1. Memory read/write (manual)

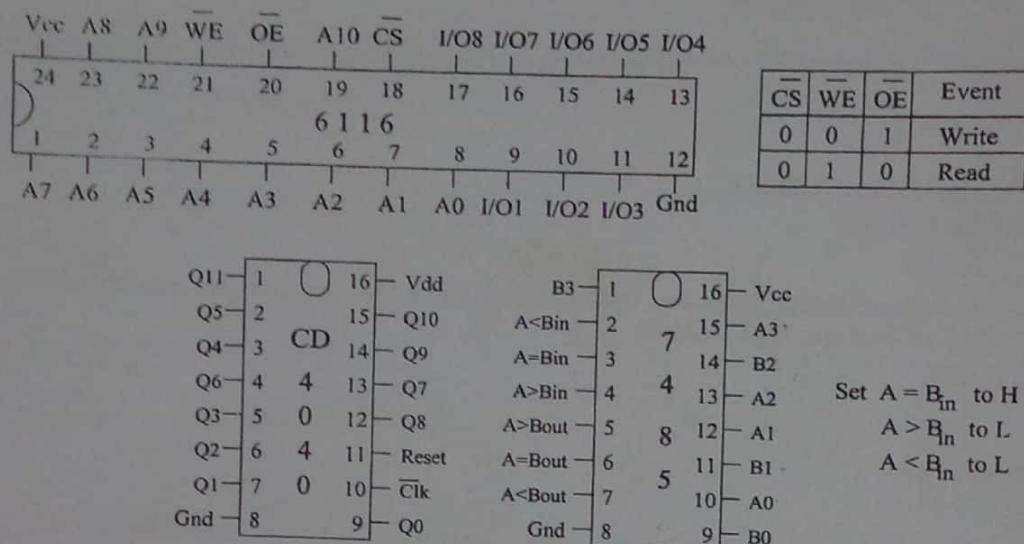


Fig. 2. The chips

TABLE I
FUNCTION TABLE OF 7485

$A_3\ B_3$	Comparing inputs			Cascading inputs			Outputs		
	$A_2\ B_2$	$A_1\ B_1$	$A_0\ B_0$	$A > B$	$A < B$	$A = B$	$A > B$	$A < B$	$A = B$
$A_3 > B_3$	X	X	X	X	X	X	H	L	L
$A_3 < B_3$	X	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X	H	L	L
$A_3 = B_3$	$A_2 < B_2$	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X	X	X	X	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	X	X	X	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	H	L	L	H	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	H	L	L	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	L	H	L	L	H



REPORT ON Experiment No. 2

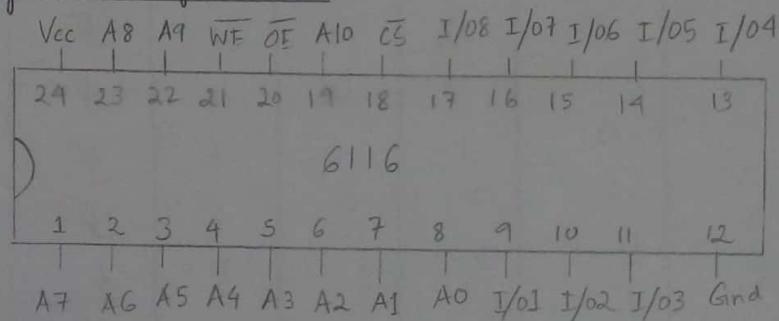
MEMORY TESTING

Objective: To become familiar with read/write operation of the memory and verification of the data written.

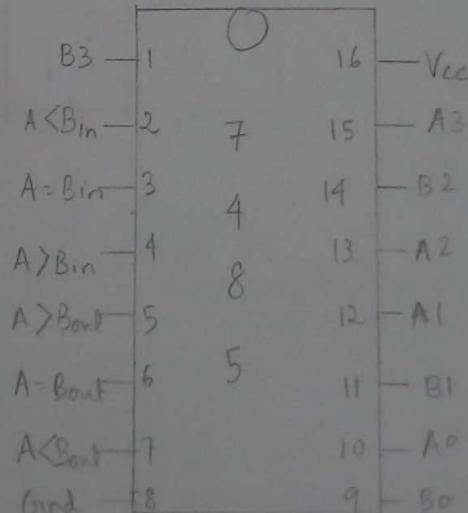
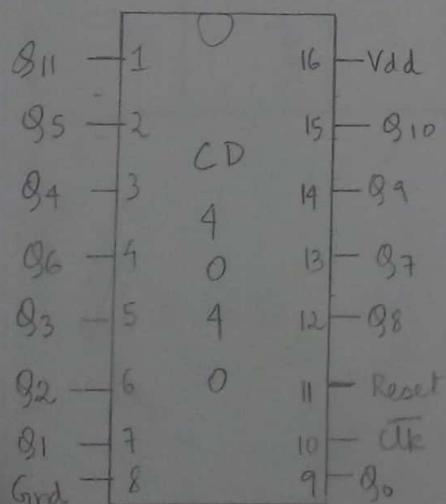
Utilize the following ICs:

1. RAM (6116)
2. 12-stage binary counter (CD 4040)
3. 4-bit comparator (7485)

Pin diagrams of ICs:



C _S	WE	OE	Funct
0	0	1	Write
0	1	0	Read



Set A = Bin to H
A > Bin to L
A < Bin to L



REPORT ON

Verification table for CD 4040

Reset	clk	Q ₃	Q ₂	Q ₁	Q ₀
open	open	0	1	0	0
1	x	0	0	0	0
0	z	0	0	0	1
0	z	0	0	1	0
0	z	0	0	1	1
0	z	0	1	0	0
0	z	0	1	0	1
0	z	0	1	1	0
0	z	0	1	1	1
0	z	1	0	0	0
0	z	1	0	0	1
0	z	1	0	1	0
0	z	1	0	1	1
0	z	1	1	0	0
0	z	1	1	0	1
0	z	1	1	1	0
0	z	1	1	1	1

It would start from 0000.



REPORT ON

Function table of 7485

Comparing inputs				Cascading inputs			Outputs		
$A_3 B_3$	$A_2 B_2$	$A_1 B_1$	$A_0 B_0$	$A > B$	$A < B$	$A = B$	$A > B$	$A < B$	$A = B$
$A_3 > B_3$	X	X	X	X	X	X	H	L	L
$A_3 < B_3$	X	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X	H	L	L
$A_3 = B_3$	$A_2 < B_2$	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X	X	X	X	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	X	X	X	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	H	L	L	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	H	L	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	L	H	L	L	H

The tasks :-

- a) Perform data write to RAM(6116) and read data from the RAM (consider 4x2 memory as shown in Fig 1), Set address and data manually.

REPORT ON

a)

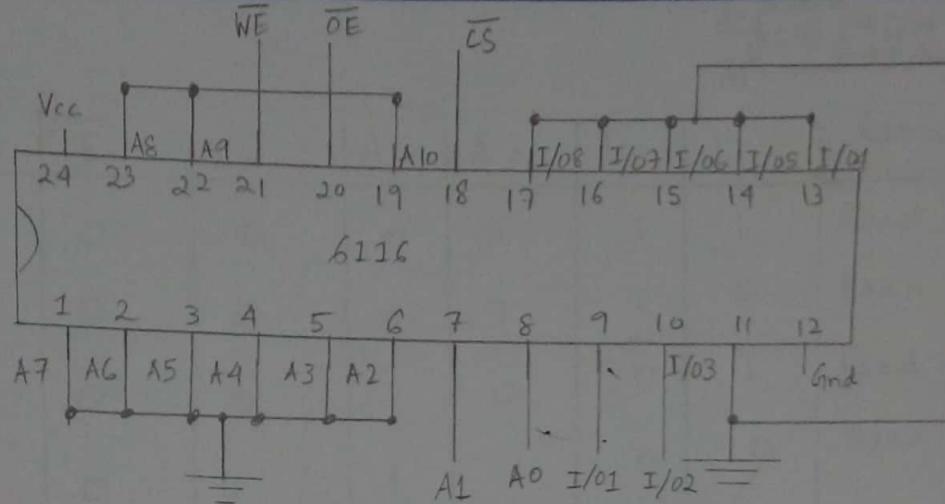
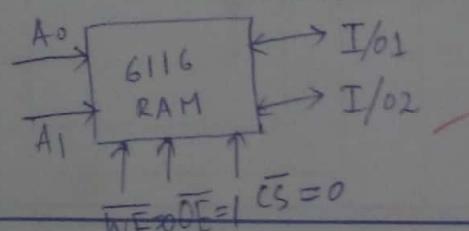


Fig 1 - Memory read/write (manual)

Write operation

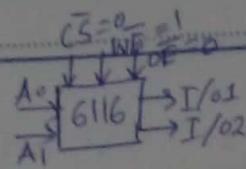
WE	OE	CS	A ₁	A ₀	I/O ₂	I/O ₁	Remarks
1	1	0	0	0	1	1	Set data and address
0	1	0	0	0	1	1	Write 11 in Loc 00
1	1	0	0	1	1	0	Set data & address
0	1	0	0	1	1	0	Write 10 in Loc 01
1	1	0	1	0	0	1	Set data & address
0	1	0	1	0	0	1	Write 01 in Loc 10
1	1	0	1	1	0	0	Set data and address
0	1	0	1	1	0	0	Write 00 in Loc 11





REPORT ON

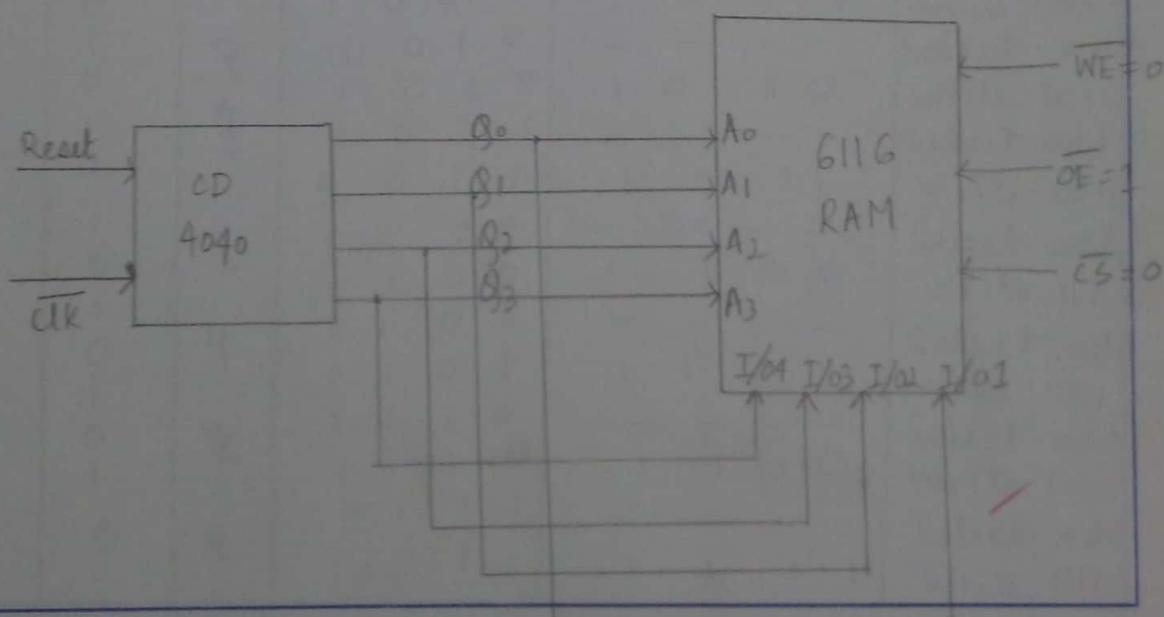
Read



WE	OE	CS	A ₁	A ₀	I ₂	I ₁	Remarks
1	0	0	0	0	1	1	Read from Loc 00
1	0	0	0	1	1	0	Read from Loc 01
1	0	0	1	0	0	1	Read from Loc 10
1	0	0	1	0	0	0	Read from Loc 11
1	0	0	1	1	0	✓	

- b)(i) Perform fast data write to RAM(6116) in l locations ($l < m$ of an $m \times n$ memory). Generate memory address using counter (CD 4040). Consider 4-bit data. Data is input from the least significant 4-bit of the counter.

Write operation





REPORT ON

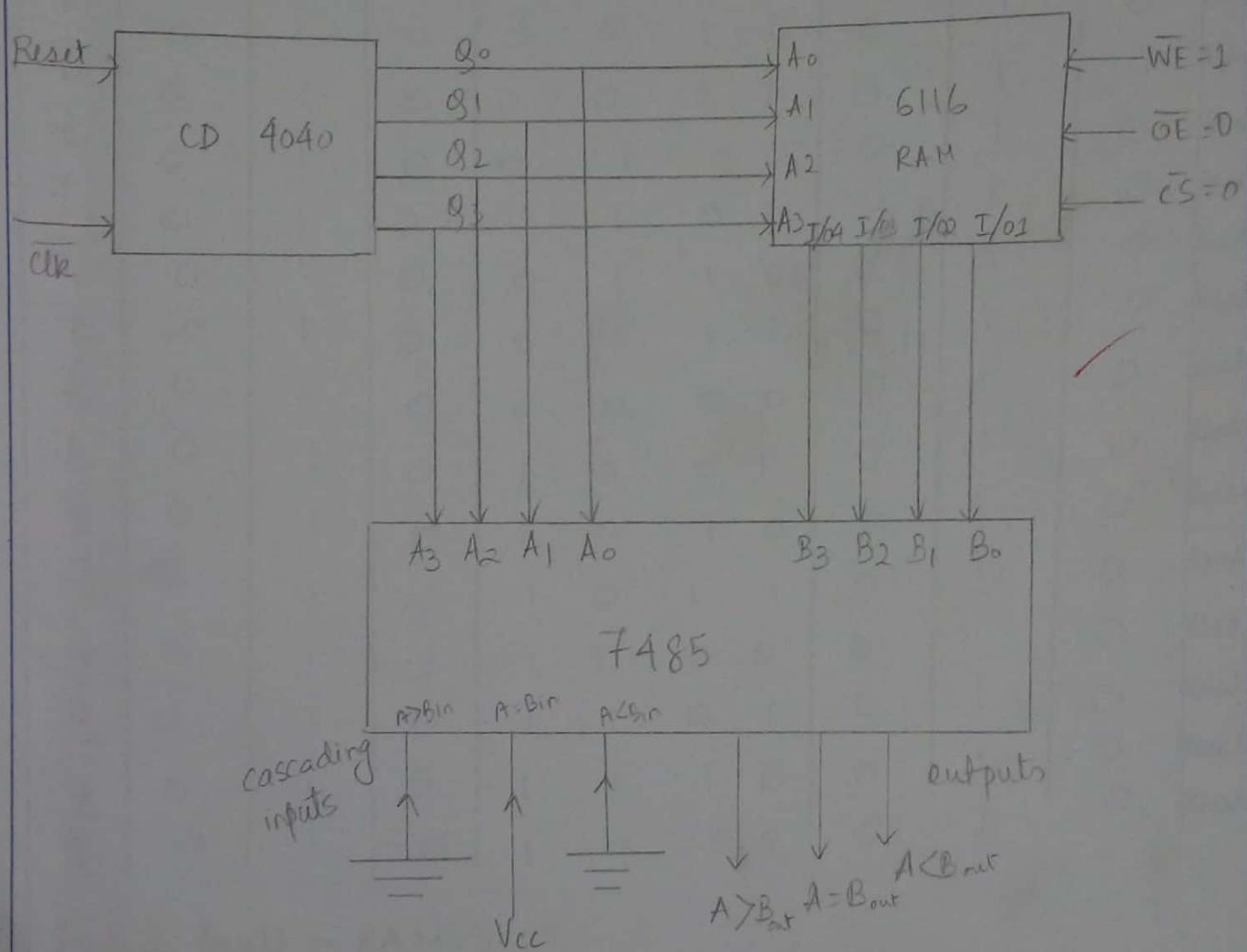
Write operation

CS	OF	WF	CP	A ₃	A ₂	A ₁	A ₀	I/bt	I/o ₃	I/o ₂	I/o ₁	Verification
0	1	1	Z	0	0	0	0	-	-	-	-	Select address
0	1	0		0	0	0	0	0	0	0	0	Write 0000
0	1	1	Z	0	0	0	1	-	-	-	-	Select address
0	1	0		0	0	0	1	0	0	0	1	Write 0001
0	1	1	Z	0	0	1	0	-	-	-	-	Select address
0	1	0		0	0	1	0	0	0	1	0	Write 0010
0	1	1	Z	0	0	1	1	-	-	-	-	Select add.
0	1	0		0	0	1	1	0	0	1	1	Write 0011
0	1	0		0	1	0	0	-	-	-	-	Select add.
0	1	1	Z	0	1	0	0	0	1	0	0	Write 0100
0	0		Z	0	1	0	0	0	1	0	1	Write 0101
0	1	1	Z	0	1	0	1	-	-	-	-	Select add
0	0		Z	0	1	0	1	0	1	0	1	Write 0110
0	1	1	Z	0	1	1	0	-	-	-	-	Select add
0	0		Z	0	1	1	0	0	1	1	0	Write 0111
0	1	1	Z	1	0	0	0	-	-	-	-	Select add
0	0		Z	1	0	0	0	1	0	0	0	Write 1000
0	1	1	Z	1	0	0	1	-	-	-	-	Select add.
0	0		Z	1	0	0	1	1	0	0	1	Write 1001
0	1	1	Z	1	0	1	0	-	-	-	-	Select add
0	0		Z	1	0	1	0	1	0	1	0	Write 1010
0	1	1	Z	1	0	1	1	-	-	-	-	Select add
0	0		Z	1	0	1	1	1	0	1	1	Write 1011
0	1	1	Z	1	1	0	0	-	-	-	-	Select add
0	0		Z	1	1	0	0	1	1	0	0	Write 1100
0	1	1	Z	1	1	0	1	-	-	-	-	Select add
0	0		Z	1	1	0	1	1	1	0	1	Write 1101
0	1	1	Z	1	1	1	0	-	-	-	-	Select add
0	0		Z	1	1	1	0	1	1	1	0	Write 1110
0	1	1	Z	1	1	1	1	-	-	-	-	Select add
0	0		Z	1	1	1	1	1	1	1	1	Write 1111



REPORT ON

- ii) Perform data read from RAM. Compare the data read from memory and the data actually written (use comparator 7485) to verify the condition of memory.



A = B_{in} to H

A > B_{in} to L

A < B_{in} to L

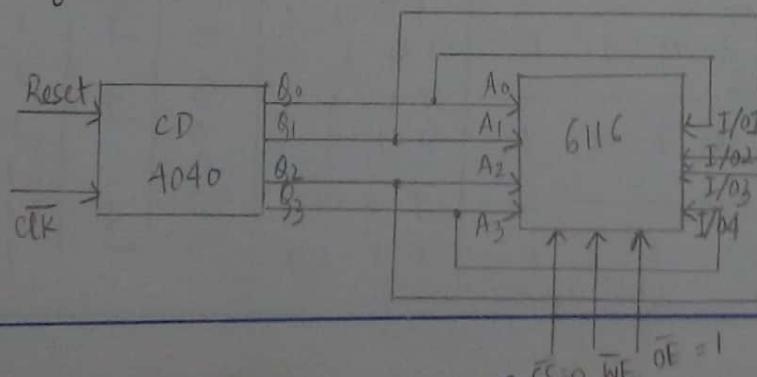


REPORT ON

Read.

CP	\bar{CS}	\bar{WE}	\bar{OE}	A_3	A_2	A_1	A_0	$A > B$	$A = B$	$A < B$	Remarks
1	0	1	0	0	0	0	0	0	1	0	Read 0000
1	0	1	0	0	0	0	1	0	1	0	Read 0001
1	0	1	0	0	0	1	0	0	1	0	Read 0010
1	0	1	0	0	0	1	0	0	1	0	Read 0011
1	0	1	0	0	0	1	1	0	1	0	Read 0100
1	0	1	0	0	1	0	0	0	1	0	Read 0101
1	0	1	0	0	1	0	1	0	0	1	Read 0110
1	0	1	0	0	1	1	0	0	1	0	Read 0111
1	0	1	0	0	1	1	1	0	1	1	Read 1000
1	0	1	0	1	0	0	0	0	1	0	Read 1001
1	0	1	0	1	0	0	1	0	1	0	Read 1010
1	0	1	0	1	0	1	1	0	1	0	Read 1011
1	0	1	0	1	0	1	1	0	1	1	Read 1100
1	0	1	0	1	1	0	0	0	1	1	Read 1101
1	0	1	0	1	1	0	1	0	1	0	Read 1110
1	0	1	0	1	1	1	0	0	1	0	Read 1111
1	0	1	0	1	1	1	1	0	1	1	Read 1111

- c) Detect fault in RAM.
i) Perform fast data write to RAM, as in (b).



Date 8/02.....

Page No. 15

Signature.....



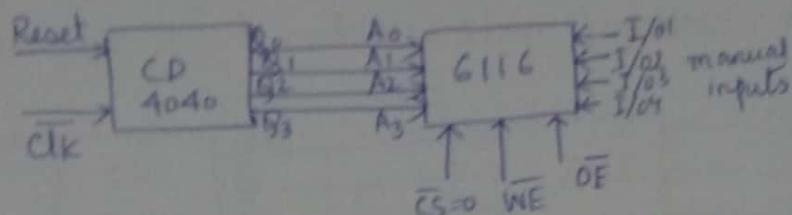
REPORT ON

CS	WE	OE	CP	Reset	A ₃	A ₂	A ₁	A ₀	I/04	I/03	I/02	I/01	Remarks
0	1	1			1	0	0	0	- - -	- - -	- - -	- - -	Set address
0	0	1			0	0	0	0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0000 written to Loc 0000
0	1	1	X		0	0	0	0	1	- - -	- - -	- - -	Set address
0	0	1			0	0	0	0	1	0 0 0 1	0 0 0 1	0 0 0 1	0001 written to Loc 0001
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	0 0 1 0	0 0 1 0	0 0 1 0	0010 at Loc 0010
0	1	1	X		0	0	0	0	1	- - -	- - -	- - -	Set address
0	0	1			0	0	0	0	1	0 0 1 0	0 0 1 0	0 0 1 0	0010 at Loc 0010
0	1	1	X		0	0	0	0	1	- - -	- - -	- - -	Set address
0	0	1			0	0	0	0	1	0 0 1 1	0 0 1 1	0 0 1 1	0011 at Loc 0011
0	1	1	X		0	0	0	0	1	- - -	- - -	- - -	Set address
0	0	1			0	0	0	0	1	0 0 1 1	0 0 1 1	0 0 1 1	0011 at Loc 0011
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	0 1 0 0	0 1 0 0	0 1 0 0	0100 at Loc 0100
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	0 1 0 0	0 1 0 0	0 1 0 0	0100 at Loc 0100
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	0 1 0 1	0 1 0 1	0 1 0 1	0101 at Loc 0101
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	0 1 0 1	0 1 0 1	0 1 0 1	0101 at Loc 0101
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	0 1 1 0	0 1 1 0	0 1 1 0	0110 at Loc 0110
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	0 1 1 0	0 1 1 0	0 1 1 0	0110 at Loc 0110
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	0 1 1 1	0 1 1 1	0 1 1 1	0111 at Loc 0111
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	0 1 1 1	0 1 1 1	0 1 1 1	0111 at Loc 0111
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	1 0 0 0	1 0 0 0	1 0 0 0	1000 at Loc 1000
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	1 0 0 1	1 0 0 1	1 0 0 1	1001 at Loc 1001
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	1 0 1 0	1 0 1 0	1 0 1 0	1010 at Loc 1010
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	1 0 1 1	1 0 1 1	1 0 1 1	1011 at Loc 1011
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	1 1 0 0	1 1 0 0	1 1 0 0	1100 at Loc 1100
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	1 1 0 0	1 1 0 0	1 1 0 0	1100 at Loc 1100
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	1 1 0 1	1 1 0 1	1 1 0 1	1101 at Loc 1101
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	1 1 0 1	1 1 0 1	1 1 0 1	1101 at Loc 1101
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	1 1 1 0	1 1 1 0	1 1 1 0	1110 at Loc 1110
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
0	0	1			0	0	0	1	0	1 1 1 1	1 1 1 1	1 1 1 1	1111 at Loc 1111
0	1	1	X		0	0	0	1	0	- - -	- - -	- - -	Set address
Date	8/62	Page No.	16	Signature	AB								



REPORT ON

- ii) Insert fault to one or more locations of the RAM by force.



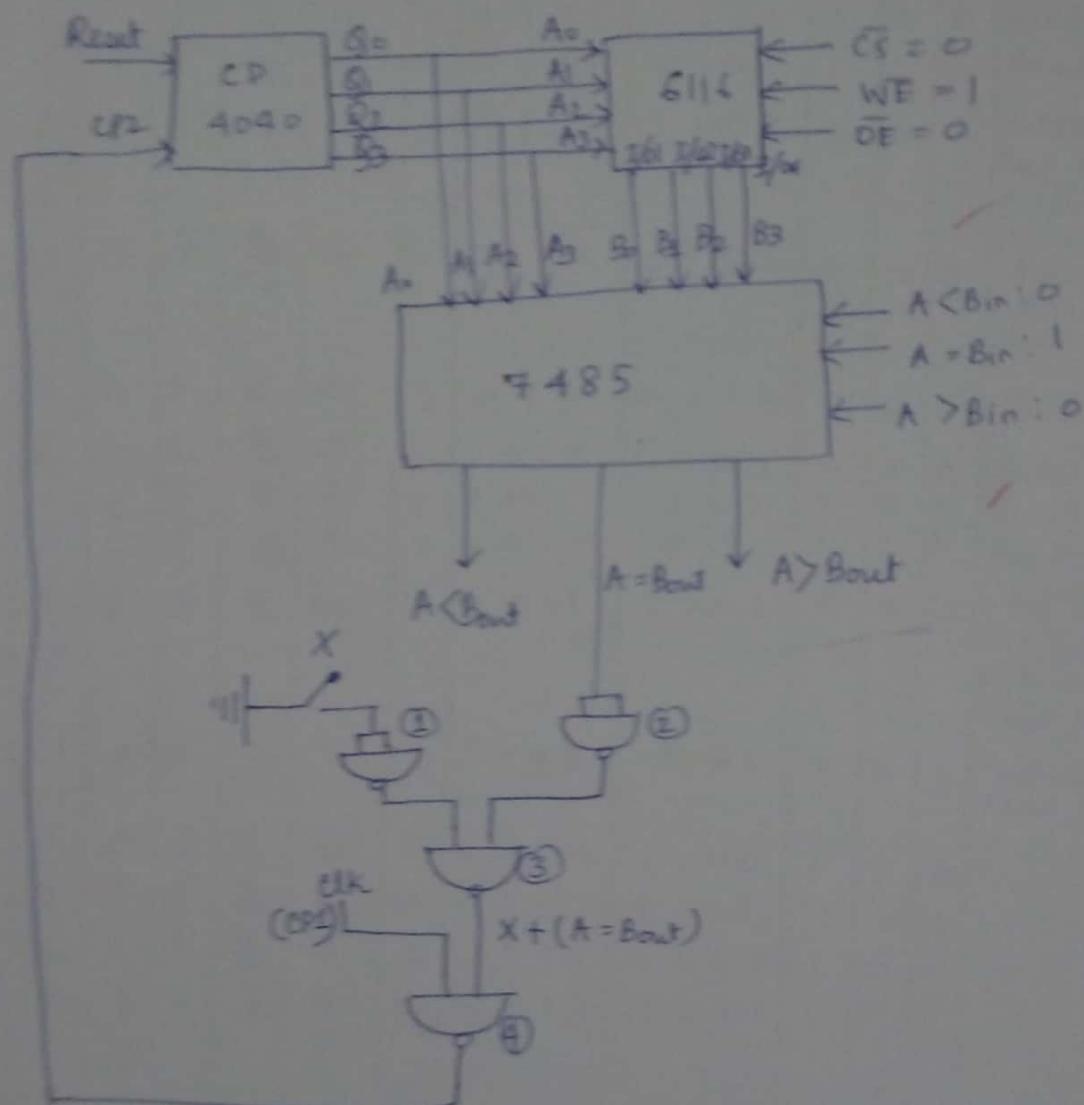
CS	WE	OE	CP	Reset	A ₃	A ₂	A ₁	A ₀	I/O4	I/O3	I/O2	I/O1	Remarks	
0	1	0		1	0	0	0	0	0	0	0	0	0	Read 0000
0	1	0	X	0	0	0	0	1	0	0	0	1		Read 0001
0	1	0	X	0	0	0	1	0	0	0	1	0		Read 0010
0	1	0	X	0	0	0	1	1	0	0	1	1		Read 0011
0	1	0	X	0	0	1	0	0	0	1	0	0		Read 0100
0	1	0	X	0	0	1	0	0	1	0	1	0		Read 0101
0	1	0	X	0	0	1	0	1	0	1	0	1		Read 0110
0	1	0	X	0	0	1	1	0	0	1	1	0		RAM deactivated
0	1	1		0	0	0	1	1	0	0	1	0		0100 written at loc 0110 (Fault inserted)
0	0	1		0	0	0	1	1	-	-	-	-		Address set to 0111
0	1	1	X	0	0	0	1	1	1	0	1	1		Read 0111
0	1	0		0	0	1	0	0	0	1	0	0		Read 1000
0	1	0	X	0	0	1	0	0	1	0	0	1		Read 1001
0	1	0	X	0	0	1	0	1	0	1	0	1		Read 1010
0	1	0	X	0	0	1	0	1	1	0	1	1		Read 1011
0	1	0	X	0	0	1	0	1	-	-	-	-		RAM deactivated
0	1	1		0	0	1	0	1	1	0	0	1		1001 written at loc 1011 (Fault inserted)
0	0	1		0	0	1	0	1	1	0	0	1		Address set to 1100
0	1	1	X	0	0	1	1	0	-	-	-	-		Read 1100
0	1	0	X	0	0	1	1	0	0	1	1	0		Read 1101
0	1	0	X	0	0	1	1	0	1	1	0	1		Read 1110
0	1	0	X	0	0	1	1	1	1	1	1	1		Read 1111

Fault inserted at location 0110 and 1011



REPORT ON

- (ii) Perform data read from RAM. Compare the data read and the data actually written (use comparator 7485) to detect faulty RAM. As soon as the fault is detected at A_1 location, stop further data read.





REPORT ON

Read operation

CS	WE	OE	CP1	CP2	Reset	X	A ₃	A ₂	A ₁	A ₀	I ₀₄	I ₀₃	I ₀₂	I ₀₁	A=B	A>B	Remarks
0	1	0				1	0	0	0	0	0000	0000	0001	1	0	0	Read 0000
0	1	0	5	7		0	0	0	0	0	0001	0001	0001	1	0	0	Read 0001
0	1	0	5	7		0	0	0	0	1	0010	0010	0010	1	0	0	Read 0010
0	1	0	5	7		0	0	0	0	1	0011	0011	0011	1	0	0	Read 0011
0	1	0	5	7		0	0	0	1	0	0100	0100	0100	1	0	0	Read 0100
0	1	0	5	7		0	0	0	1	0	0101	0101	0101	1	0	0	Read 0101
0	1	0	5	7		0	0	0	1	1	0110	0100	0100	0	0	1	D100 read at Loc 0110
0	1	0	5	7	0		0	0	0	1	1010	0100	0100	0	0	1	A = Bout, logic 0 Hence fault detected. Further data read stopped
0	1	0	5	7	0		1	0	1	1	0111	0111	0111	1	0	0	Read 0111
0	1	0	5	7	0		0	1	1	0	1000	1000	1000	1	0	0	Read 1000
0	1	0	5	7	0		0	1	1	0	1001	1001	1001	1	0	0	Read 1001
0	1	0	5	7	0		0	1	1	0	1010	1010	1010	1	0	0	Read 1010
0	1	0	5	7	0		0	0	1	0	1011	1001	1001	0	0	1	1001 read at Loc 1011 A = Bout: Logic 0 Fault detected.
0	1	0	5	7	0		0	0	1	0	1011	1001	1001	0	0	1	Further data read stopped
0	1	0	5	7	0		0	1	1	0	1100	1100	1100	1	0	0	
0	1	0	5	7	0		0	0	1	1	101	1101	1101	1	0	0	
0	1	0	5	7	0		0	0	1	1	1110	1110	1110	1	0	0	
0	1	0	5	7	0		0	0	1	1	1111	1111	1111	1	0	0	

CP1 = Clock pulse applied

CP2 = clock pulse received at CD 404.



REPORT ON

Report on : A logic to start data read from the next memory location ($l_1 + 1$).

Mechanism of operation:

- i) At normal operation, $X = 0$. Hence the 4th NAND gate is governed by the inputs Clk and $X + (\text{A} = \text{Bout}) = (\text{A} = \text{Bout})$. When $\text{A} = \text{Bout}$, the data read is correct, hence the output of 4th NAND gate is

$$\overline{(\text{Clk})(\text{A} = \text{Bout})} = \overline{\text{Clk}} \quad [\because \text{A} = \text{Bout} = \text{logic 1}]$$

Hence application of positive edge triggered clock applies a negative edge trigger on CD 4040. Hence the address changes when $\text{A} = \text{Bout}$ for the previous address.

- ii) When $\text{A} \neq \text{Bout}$, then a fault is detected in the RAM. The output of the 5th NAND gate becomes

$$\overline{(\text{Clk})(\text{A} = \text{Bout})} = \overline{0} = 1 \quad [\because \text{A} = \text{Bout} = \text{logic 0}]$$

Hence, even though we apply $\text{CP1}, \text{CP2} = 1$ (always) and so ; next address can't be obtained.

- iii) To change the address, X is momentarily changed to 1 and clock pulse is applied.

$$\text{When } X = 1, \quad X + (\text{A} = \text{Bout}) = 1$$

\therefore At NAND gate 4, $\overline{(\text{Clk})(X + (\text{A} = \text{Bout}))} = \overline{\text{Clk}}$

Thus, required negative edge trigger is applied on CD 4040 and the address is changed. X is again changed to 0. The normal operation continues.



REPORT ON

Precautions:-

- 1) All connections should be made tight.
 - 2) The IC chips should be handled carefully in order to avoid damage.
 - 3) It is advisable to verify the working of each IC chips (ie, counter CD 4010, comparator 7485), so as to avoid unnecessary wrong data.
 - 4) It should be ensured that the wires do not touch each other.
-

Indian Institute of Engineering Science and Technology, Shibpur
 Department of Computer Science

Report on Data transfer between MDR and RAM

REPORT ON



Experiment No. 3

Title Data transfer between
MDR and RAM

Name Tanusha Sarawgi, Dhriti Saha Date of Performance 15/2/18

Semester 4th Date of Submission 8/3/18

Branch CST Roll No. Gm-27,28 Signature Tanusha

Examined by Sulata Mitra 8/3/18

Computer Architecture and Organization Laboratory
Department of Computer Science and Technology, IIEST

Experiment No: 3 (Data transfer between MDR and RAM)

Objective: To become familiar with data transfer from buffer (MDR) to main memory (RAM) and from RAM to MDR. Further, to study the data access from external world (input device) to MM.

Utilize the following ICs:

1. Octal D latch (74373)
2. Tri-state noninverting buffer (74241) - 2 nos.
3. RAM (6116)

The design

a) Realize the logic circuit shown in Figure (a) for 1-bit and verify data transfer (i) from external source E to MDR, (ii) MDR to RAM, and (iii) RAM to MDR.

b) Realize the logic circuit shown in Figure (b) for 2-bit, utilizing the design implemented for 1-bit and verify data transfer (i) from MDR to RAM, and (ii) RAM to MDR.

Steps to realize data transfer

(i) E to MDR:

0. Set data to E.
1. OP/CTRL = L, then Enable (G) = H of the octal latch.
2. Mode = H, then R/ \bar{W} = L (writes E to latch -that is MDR).

(ii) MDR to RAM:

A. *Read from MDR*

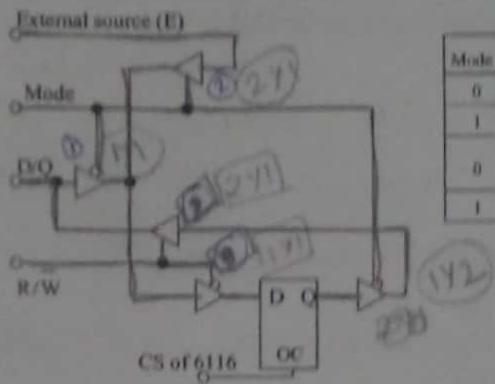
1. Enable (G) = L of the octal latch.
2. Mode = L, then R/ \bar{W} = H (reads from output of the latch).
3. Data available at the output Q.

B. *Output of MDR to RAM*

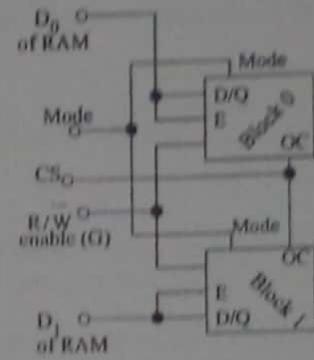
0. Data available at I/O.
1. \bar{WE} = L (write enabled), then \bar{OE} = H (output disabled) of RAM.
2. \bar{WE} = H (output of RAM are now fixed to the value Q).

(iii) RAM to MDR:

1. \bar{OE} = L (output enabled), and \bar{WE} = H.
2. Enable (G) = H of latch (enables write latch/MDR).
3. Mode = L (to prevent external data to input to latch), then R/ \bar{W} = L (writes MDR).



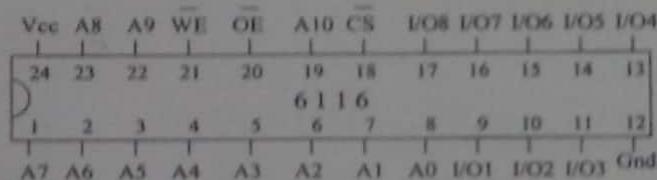
Mode	R/W	Event
0	1	Read MDR
1	1	—
0	0	Write D (from memory) to MDR
1	0	Write E to MDR



a) Internal structure of a block (1-bit MDR)

b) Two-bit MDR with RAM

Fig. 1. Circuit diagram



CS	WE	OE	Event
0	0	1	Write
0	1	0	Read

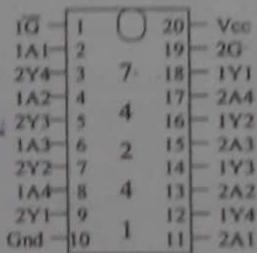
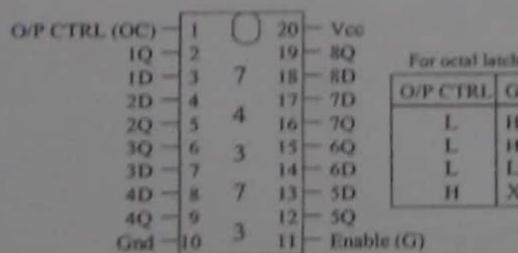


Fig. 2. The chips



Report on Data Transfer between MDR and RAM.

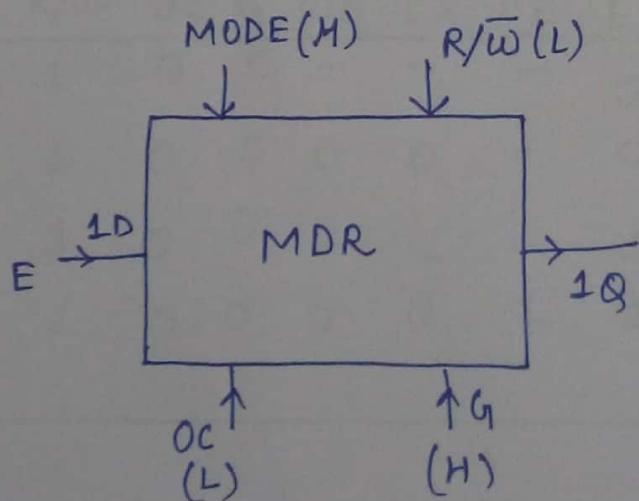
Experiment No. : 3

Objective : To become familiar with data transfer from buffer (MDR) to main memory (RAM) and from RAM to MDR. Further, to study the data access from external word (Input device) to MM.

Utilize the following ICs :

- (1) Octal D Latch (74373)
- (2) 16-bit non Inverting Buffer (74241) - 2
- (3) RAM (6116)

(a) Realise the logic circuit for:
1-bit and verify data transfer

(i) E to MDR

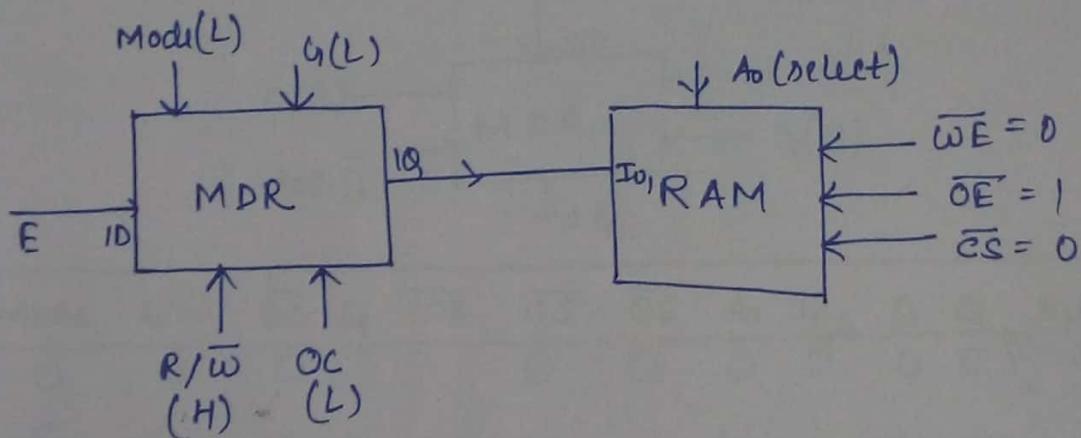
Date

Signature



Report on

Mode	R/W	E	G ₁	OC	Q	Remarks
1	0	0	1	0	0	Read 0 from external & write in MDR
1	0	1	1	0	1	Read 1 from external & write in MDR

(ii) MDR to RAM

Mode	R/W	OC	G ₁	WE	CS	OE	E	Q	A ₀	I _{O1}	Remarks
0	1	0	0	-	-	-	0	0	-	-	Read 0 from MDR
0	1	0	0	0	✓	1	0	0	0	0	write in RAM from MDR
0	1	0	0	-	-	-	1	1	-	-	Read 1 from MDR
0	1	0	0	0	0	✓	1	1	1	1	1 write in RAM from MDR

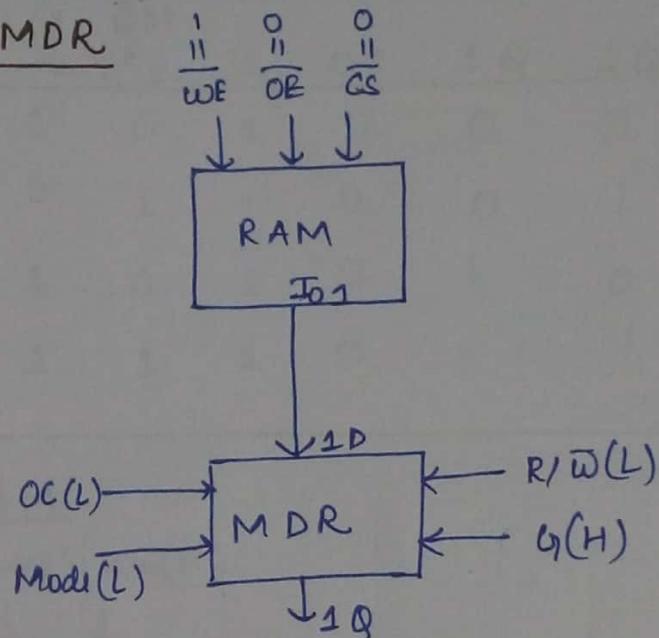
Date

Signature



Report on

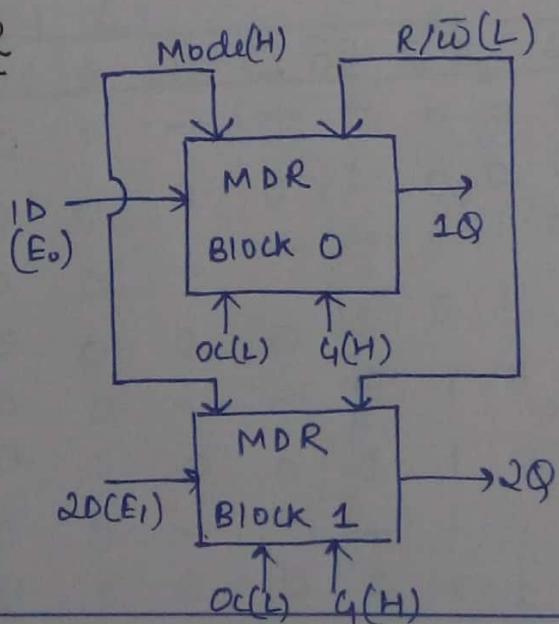
(iii) RAM to MDR



Mode	R/W	\bar{OC}	G	\bar{WE}	\bar{CS}	\bar{OE}	A_0	I_{Q_1}	D	Q	Remarks
0	0	0	1	1	0	0	0	0	0	0	Read D from RAM & write in MDR
0	0	0	1	1	0	0	1	1	1	1	Read 1 from RAM and write in MDR

• 2-bit data :

(i) E to MDR



Date

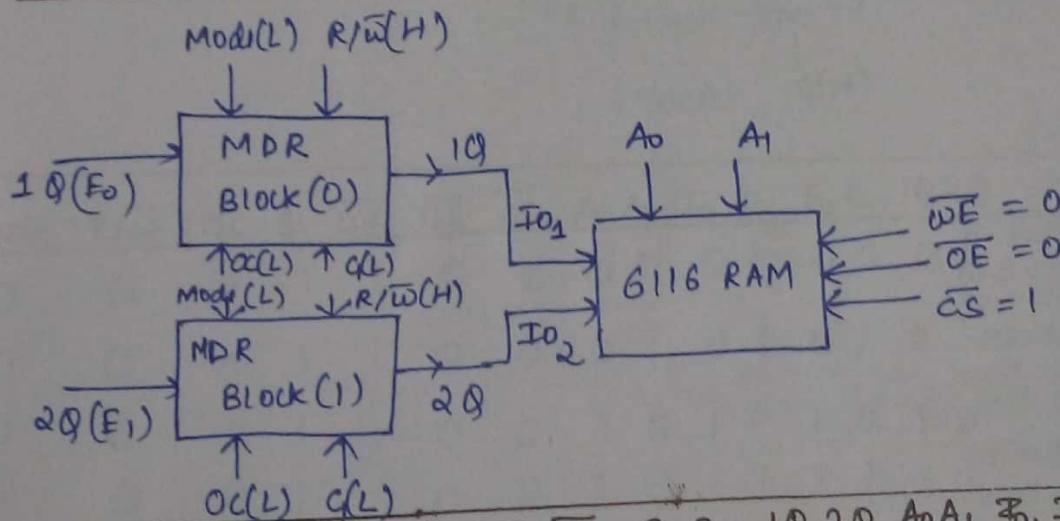
Signature



Report on

Mode	R/W	E_0	E_1	G	OC	1Q	2Q	Remarks
1	0	0	0	1	✓ 0	0	0	write 0 in block 0 and 0 in Block 1
1	0	0	1	1	✓ 0	0	1	write 0 in block 0 and 1 in block 1
1	0	1	0	1	✓ 0	1	0	write 1 in block 0 and 0 in block 1
1	0	1	1	1	✓ 0	1	1	write 1 in block 0 and 1 in block 1.

(ii) MDR to RAM



Mode	R/W	OC	G	\overline{WE}	\overline{CS}	\overline{OE}	E_0	E_1	1Q	2Q	A_0	A_1	I_{DQ_1}	I_{DQ_2}	Remarks
0	1	0	0	-	-	-	0	0	0	0	-	-	-	-	Read from block 0 and 1 from block 1
0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	Write from block 0 and 1 to RAM
0	1	0	0	-	-	-	0	1	0	1	-	-	-	-	Write from block 0 and 1 to RAM
0	1	0	0	0	0	1	0	1	0	1	-	-	-	-	Read from block 0 and 1 from block 1
0	1	0	0	-	-	-	1	0	1	0	-	-	-	-	Read from block 0 and 1 from block 1
0	1	0	0	0	0	1	1	0	1	0	1	0	1	1	Write from block 0 and 1 to RAM
0	1	0	0	-	-	-	1	1	1	1	-	-	-	-	Read from block 0 and 1 from block 1
0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	Write from block 0 and 1 to RAM

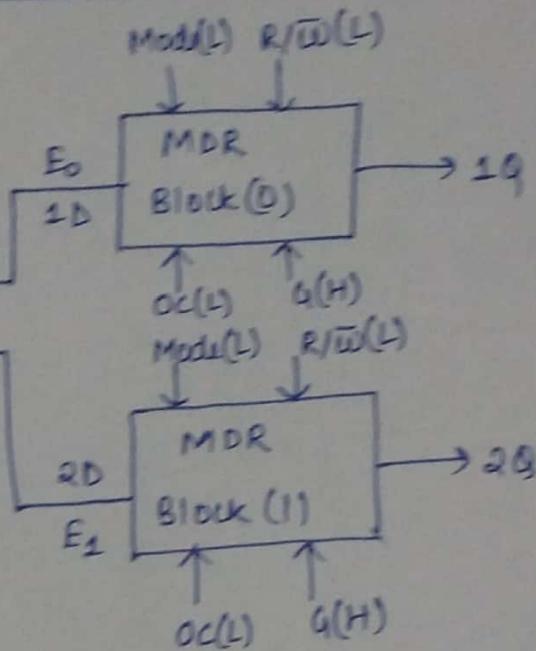
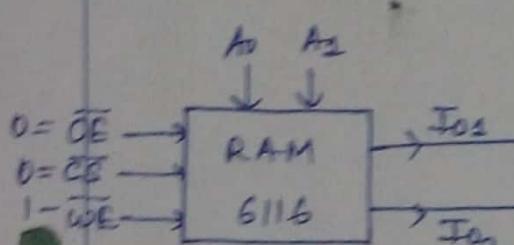
Date

Signature



Report on

(iii) RAM to MDR



Mode	R/W	OC	G	\overline{WE}	\overline{CS}	\overline{OE}	A ₀	A ₁	I _{o1}	I _{o2}	E ₀	E ₁	1Q	2Q	Remarks
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0 Read 00 from RAM and write 0 in block 0 and 0 in block 1
0	0	0	1	1	0	0	0	1	0	1	0	1	0	1	1 Read 0 from RAM and write 0 in block 0 and 1 in block 1
0	0	0	1	1	0	0	1	0	1	0	1	0	1	1	1 Read 11 from RAM and write 1 in block 0 and 1 in block 1.
0	0	0	1	1	0	0	1	1	1	1	1	1	1	1	

Date

Signature



Indian Institute of Engineering Science and Technology, Shibpur
 Department of Computer Science & Technology

Report on Experiment 4

REPORT ON



Experiment No. 4

Title Swapping contents of register
 using microprogram

Name	Tanisha Sarangi	Dwiti Saha	Date of Performance	22/3/18
Semester	4th		Date of Submission	5/4/18
Branch	CST	Roll No. Gx-23, 27	Signature	Tanisha
Examined by	S. Chakraborty 05/04/18			



REPORT ON Experiment 4

Experiment No. - 4

Name of Experiment - Swapping contents of registers using microprogram.

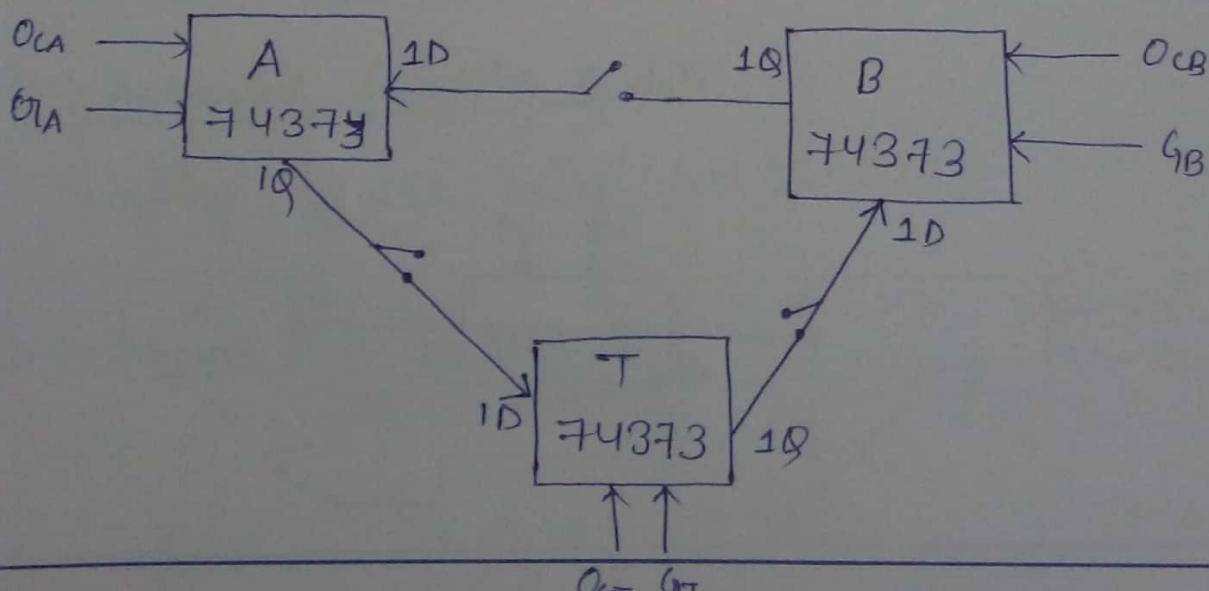
Objective - To become familiar with

- Register level data transfer through common bus
- Microprogrammatical realization of the register level data transfer.

Ic's:

- (1) Octal D latches (74373) - 3
- (2) 4-bit counter (7493)
- (3) RAM (6116)
- (4) 74245 - 2

- (i) Setting registers and control switches (manually) to realize swapping of data:





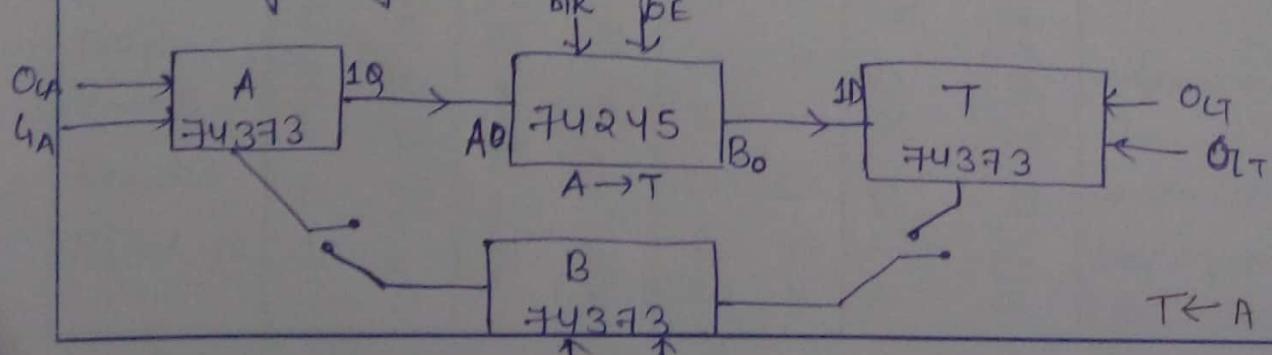
REPORT ON Expt 4

M operations	O_{CA}	G_A	O_{CT}	G_T	O_{CB}	G_B
Write A	0	1	1	1	1	1
Write B	1	1	1	1	0	1
NOP	1	1	1	1	1	1
$T \leftarrow A$	0	0	0	1	1	1
NOP	1	1	1	1	1	1
$A \leftarrow B$	0	1	1	1	1	1
NOP	1	1	1	1	0	0
$B \leftarrow T$	1	1	0	0	0	1

$O_{Ci}G_i = 11$ means that the 74373 chip isn't active $i = A/T/B$ and has no use. Actually more appropriate will be $O_{Ci}G_i = 1X$, because chip becomes inactive for $O_{Ci} = 1$ and hence G_i can either be zero or one i.e. don't care condition.

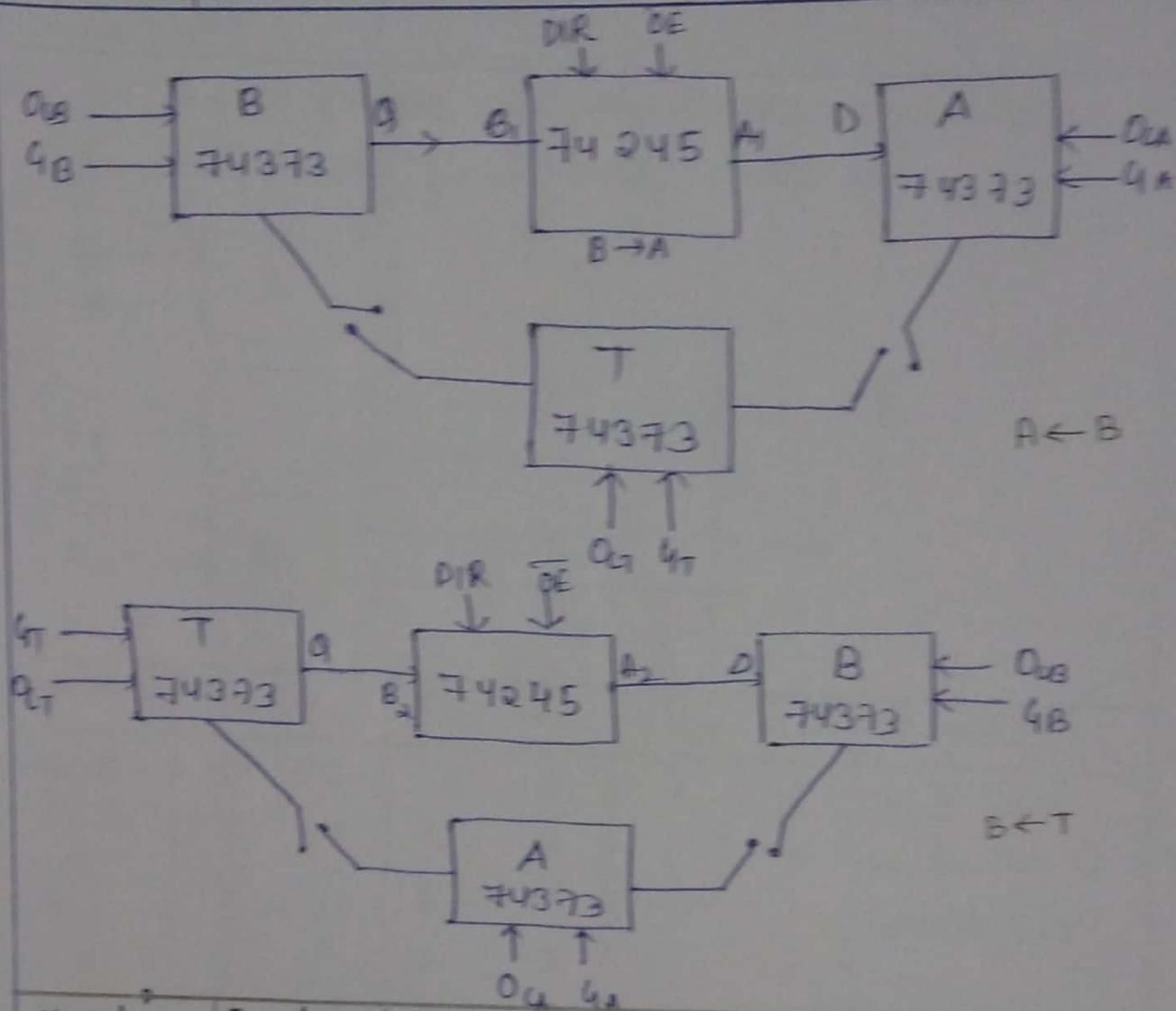
(ii) Automated Swapping of data

Swapping of data using BUS (74245):





REPORT ON EASYT 4

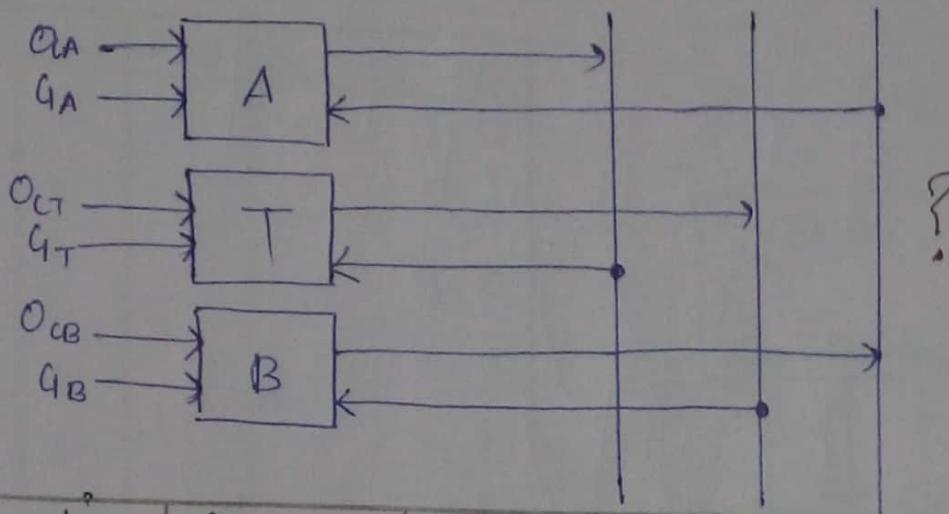


Operation	Q_A	Q_A	Q_B	Q_B	Q_T	Q_T	DIR	\overline{OE}	Remarks
write A	0	1	1	X	1	X	—	—	
write B	0	1	X	0	1	X	—	—	write 1 in A
$A \rightarrow T$	0	0	1	X	0	1	1	0	write 0 in B
NOP	0	1	X	1	X	0	X	X	write 1 in T
$B \rightarrow A$	0	1	1	X	1	X	X	X	—
NOP	1	X	0	0	1	X	0	0	write 0 in A
$T \rightarrow B$	1	X	1	X	1	X	X	X	—
Read A	0	0	0	1	0	0	0	0	write 1 in B
Read B	1	X	0	0	1	X	—	—	Read 0 from A Read 1 from B



REPORT ON Expt 4

Now we weren't provided with chip 74045 and instead use pins on the bread board as the bus.



Operation	O_{CA}	G_A	O_{CB}	G_B	O_{CT}	G_T	Remarks
write A	0	1	1	x	1	x	write 1 in A
write B	1	x	0	1	1	x	write 0 in B
$A \rightarrow T$	0	0	1	x	0	1	write 1 in T
NOP	1		0		1	x	—
$B \rightarrow A$	0	x	1	x	1	x	—
NOP	1	1	0	0	1	x	write 0 in A
$T \rightarrow B$	1	x	1	x	1	x	—
Read A	0	0	0	1	0	0	write 1 in B
Read B	1	x	0	x	1	x	Read 0 from A
			0	0	1	x	Read 1 from B

(iii) Setting up control memory

As there are 8 operations, each will have a corresponding 4-bit program. At 8 locations of RAM, write the 6-bit instruction unique to each address.

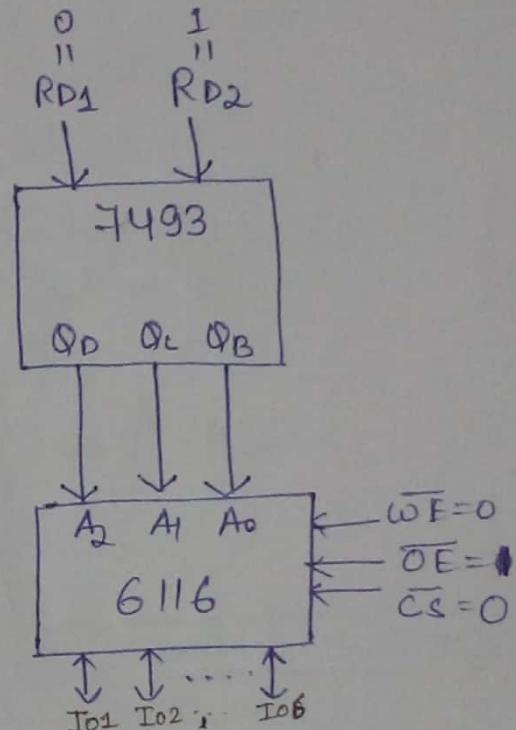
Generate address of RAM using counter 7493. Supply manual pulse to generate consecutive address.



REPORT ON Expt 4

7493 - asynchronous counter. we generate 3 bit values (Q_D, Q_C, Q_B) as we have 8 M instructions.

CLK	RD1	RD2	Q_D	Q_C	Q_B
↓	0	1	0	0	0
↓	0	1	0	0	1
↓	0	1	0	1	0
↓	0	1	0	1	1
↓	0	1	1	0	0
↓	0	1	1	0	1
↓	0	1	1	1	0
↓	0	1	1	1	1



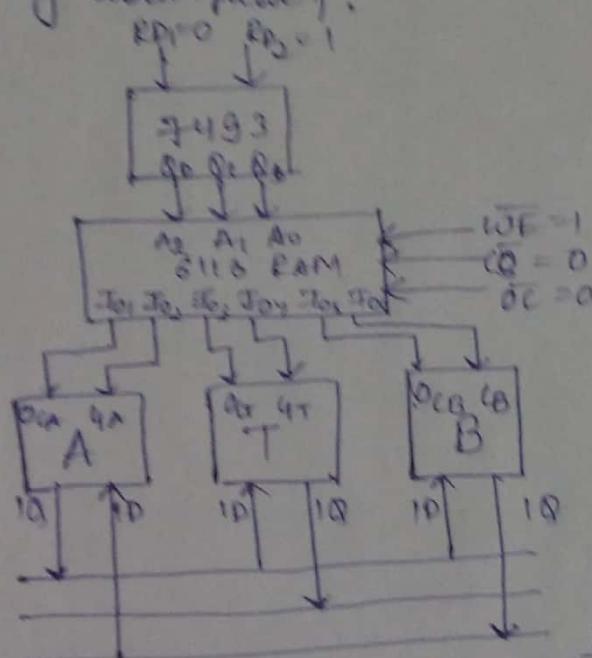
write M instructions to 8 memory locations of control memory

\bar{CS}	\bar{OE}	\bar{WE}	A_2	A_1	A_0	I_{01}	I_{02}	I_{03}	I_{04}	I_{05}	I_{06}
0	1	0	0	0	0	0	1	1	1	1	1
0	1	0	0	0	1	1	1	1	1	0	1
0	1	0	0	1	0	1	1	1	1	1	1
0	1	0	0	1	1	0	0	0	1	1	1
0	1	0	1	0	0	1	1	1	1	1	1
0	1	0	1	0	1	0	1	1	1	1	0
0	1	0	1	1	0	1	1	1	1	1	0
0	1	0	1	1	1	1	1	0	0	0	1



Report on Expt. 4

Next we connect the T_{05} to required O_{Cs} and O_s and at each clock pulse, a given instruction is executed.
(i.e. the instruction present in the address generated by clock-pulse).



$T_{01} \rightarrow O_{Cs}$
 $T_{02} \rightarrow Q_A$
 $T_{03} \rightarrow Q_T$
 $T_{04} \rightarrow 4T$
 $T_{05} \rightarrow O_{CB}$
 $T_{06} \rightarrow Q_B$

<u>CS WE OE</u>	<u>A₂ A₁ A₀</u>	<u>T₀₁ T₀₂ T₀₃ T₀₄ T₀₅ T₀₆</u>	<u>operation</u>	<u>1D1Q</u>	<u>1D1Q</u>	<u>1D1Q</u>
0 1 0	0 0 0	0 1 1 1 1 1	addition A	-	-	-
0 1 0	0 0 1	1 1 1 1 0 1	addition B	-	-	0
0 1 0	0 1 0	1 1 1 1 1 1	NOP	-	-	-
0 1 0	0 1 1	0 0 0 1 1 1	TEA	X	1	X
0 1 0	1 0 0	1 1 1 1 1 1	NOP	-	-	-
0 1 0	1 0 1	0 1 1 1 0 0	A \leftarrow B	0	X	X
0 1 0	1 1 0	1 1 1 1 1 1	NOP	-	-	-
0 1 0	1 1 1	1 1 0 0 1 1	B \leftarrow T	X	X	1

Date... 22/2/18

Signature... DS.....

Indian Institute of Engineering Science And Technology, Shibpur
Department of CST

REPORT ON Expt. 5



REPORT ON

Experiment No. 5

Title Implementation of data movement instruction

Name. Dhriti Saha, Tanisha Sarangi Date of Performance 5/4/18

Semester.... 4 Date of Submission 12/4/18

Branch.... CST Roll No 6ix-23, 27 Signature

Examined by S. Mukherjee 12/4/18

Computer Architecture and Organisation Laboratory
Department of Computer Science and Technology, IIEST

Experiment No: 5 (Implementation of data movement instructions)

Objective: To realize data movement instructions in a CPU having four 2-bit registers.

Utilise the following ICs:

1. Dual M/S flip-flops (7474) - 4 nos.
2. Dual 4 to 1 Max (74153)
3. 1 of 4 decoder (74139)

The design

There are four 2-bit registers A, B, C and D. Realize implementation of data movement instructions $MOV \text{ dest reg, source reg}$ as shown in Figure 1, where *dest* and *source* registers can be chosen from the set of registers {A, B, C, D}. Set a register address manually.

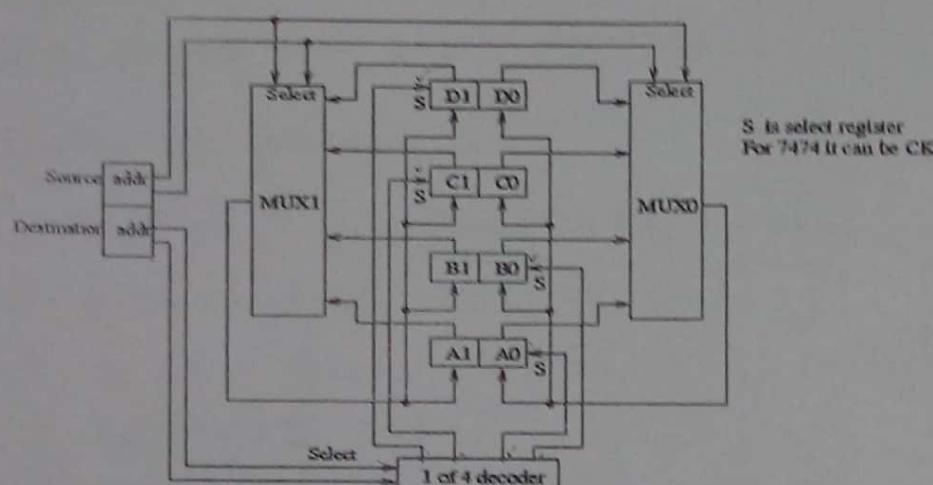


Fig. 1. Circuit diagram

1CLR	1	○	14	- Vcc
1D	2	X	13	- 2CLR
1CK	3	7	12	- 2D
1PR	4	4	11	- 2CK
1Q	5	7	10	- 2PR
1Q̄	6	9	9	- 2Q
Gnd	7	4	8	- 2D̄

Q	1	○	16	- Vcc
1A	2	X	15	- 2G
1B	3	7	14	- 2A
1Y0	4	4	13	- 2B
1Y1	5	1	12	- 2Y0
1Y2	6	3	11	- 2Y1
1Y3	7	9	10	- 2Y2
Gnd	8	9	9	- 2Y3

1G	1	○	16	- Vcc
B	2	X	15	- 2G
1C3	3	7	14	- A
1C2	4	4	13	- 2C3
1C1	5	1	12	- 2C2
1C0	6	5	11	- 2C1
1Y	7	3	10	- 2C0
Gnd	8	9	9	- 2Y

Function table 7474					
Inputs		Outputs			
Preset	Clear	Clock	D	Q	\bar{Q}
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Qn	\bar{Q}_n

Function table 74153		
Enable	Select	Outputs
G	B A	Y0 Y1 Y2 Y3
H	X X	H H H H
L	L L	L H H H
L	L H	H L H H
L	H L	H H L H
L	H H	H H H L

Function table 74139			
Select	Inputs	Stroba	Output
B A	C0 C1 C2 C3	G	Y
X X	X X X X	H	L
L L	D0 X X X	L	D0
L H	X D1 X X	L	D1
R L	X X D2 X	L	D2
H H	X X X D3	L	D3

Fig. 2. The chips

CST, IIEST, Shibpur

- 1) Write in A & B keeping MUX & decoder disable
- 2) Enable MUX to transfer src o/p to dest I/P
- 3) " Decoder Keeping MUX enable to write in dest.



REPORT ON Experiment No. 5

IMPLEMENTATION OF DATA MOVEMENT INSTRUCTIONS

Objective : To realise data movement instructions in a CPU having two 2-bit registers.

Utilise the following ICs:

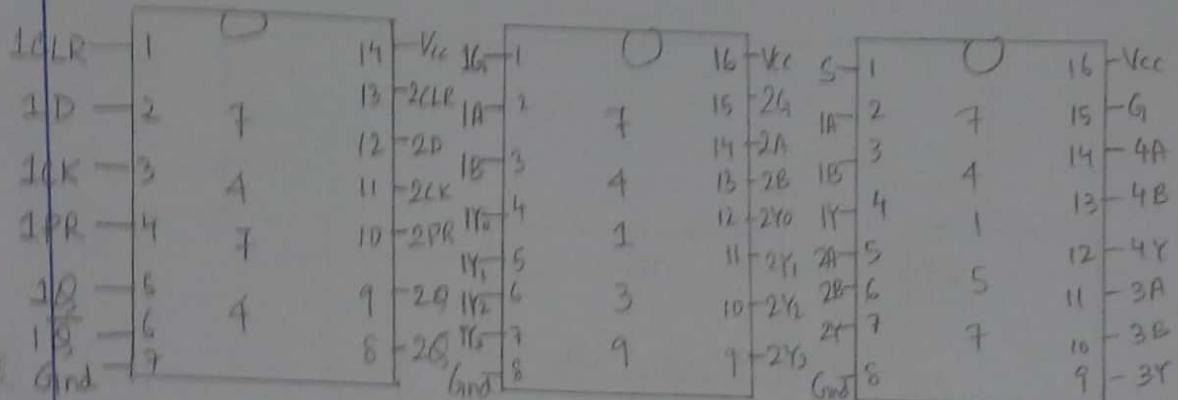
1. Dual M/S flip-flops (7474) - 2 nos
2. Dual 2 to 1 MUX (74157)
3. 2 to 4 decoder (74139)
4. NOT gates (7404)

The design

There are two 2-bit registers, namely A and B. We realize implementation of data movement instructions MOV dest reg, source reg as shown in the Figure 1, where dest and source registers can be chosen from the set of registers {A, B}. We set a register address manually.



REPORT ON Expt 5



(i)

(ii)

(iii)

Fig : Pin configurations of - (i) Dual M/S flipflop (7474)
(ii) 2 to 4 decoder (74139)
(iii) Dual 2:1 MUX (74157)

Present	Inputs		Output		
	clear	clock	D	Q	\bar{Q}
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H*	H*
H	H	F	H	H	L
H	H	F	L	L	H
H	H	L	X	\bar{Q}_0	\bar{Q}_0

Function table of 7474.

Enable	Select		Outputs			
	B	A	Y ₀	Y ₁	Y ₂	Y ₃
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

Function table of 74139.



REPORT ON Expt 5

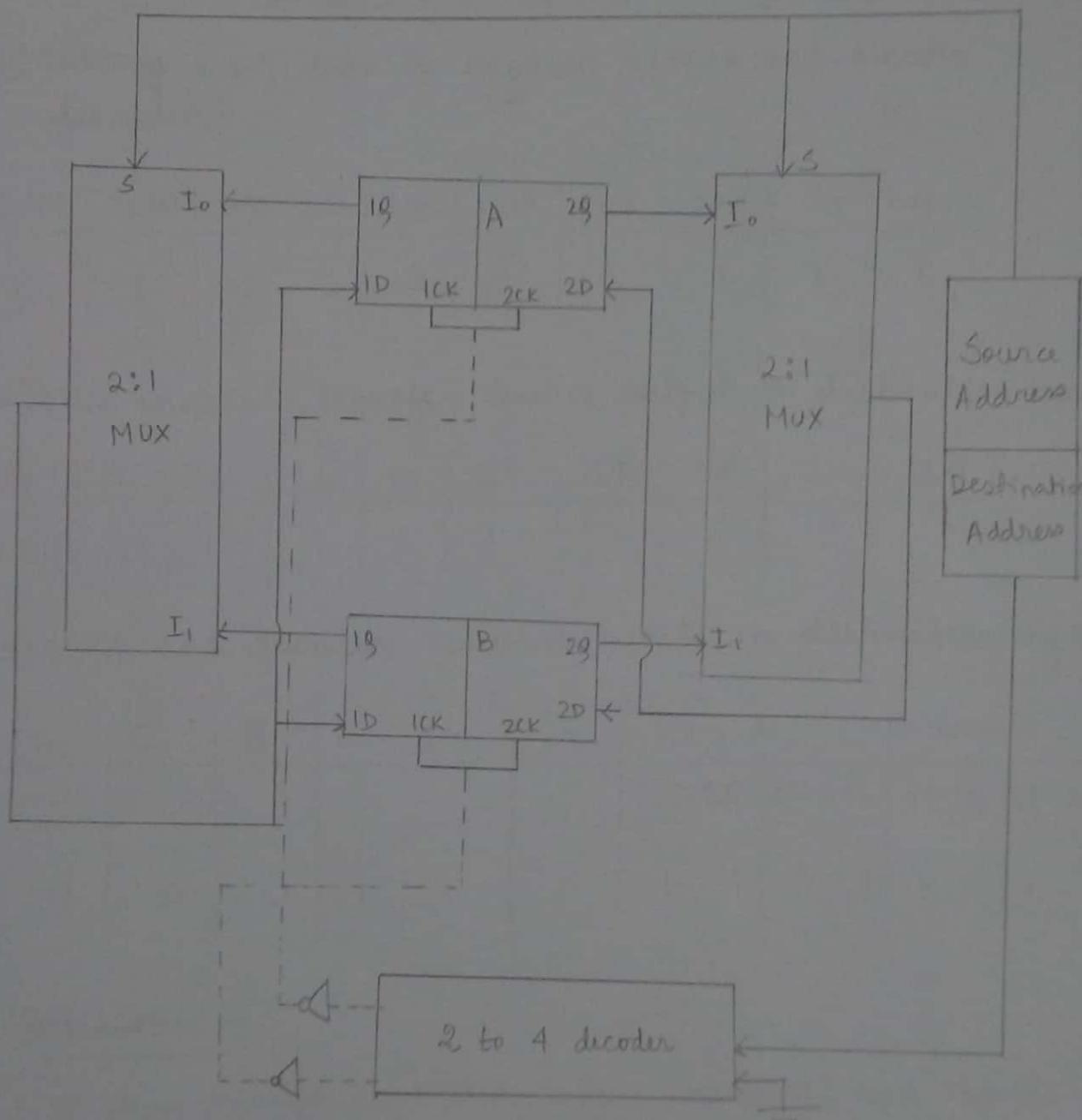


Fig: Implementation of data movement instructions



REPORT ON Expt 5

Observation table :-

Operation: MOV(B,A)

A = source

B = destination

- 1) Writing 2-bit data to register A (MUX and decoder disabled)

1CLK	1CLK	IPR	ID	1S	2CLK	2CLK	2PR	2D	2S
F	1	1	1	1	F	1	1	1	1

- 2) MUX enabled (transfer source output to destination (input))

G	S	1A	2A	1Y	2Y	ID	1S	2D	2S
0	0	1	1	1	1	1	1	1	1

- 3) MUX and decoder enabled (write in destination, reg B)

G	E	S	D	1Y	2Y	Reg A		Reg B	
						1D	1S	2D	2S
0	0	0	1	1	1	1	1	1	1

Precautions :-

- IC chips should be tested before working, and should be handled carefully.
- All connections should be tight.

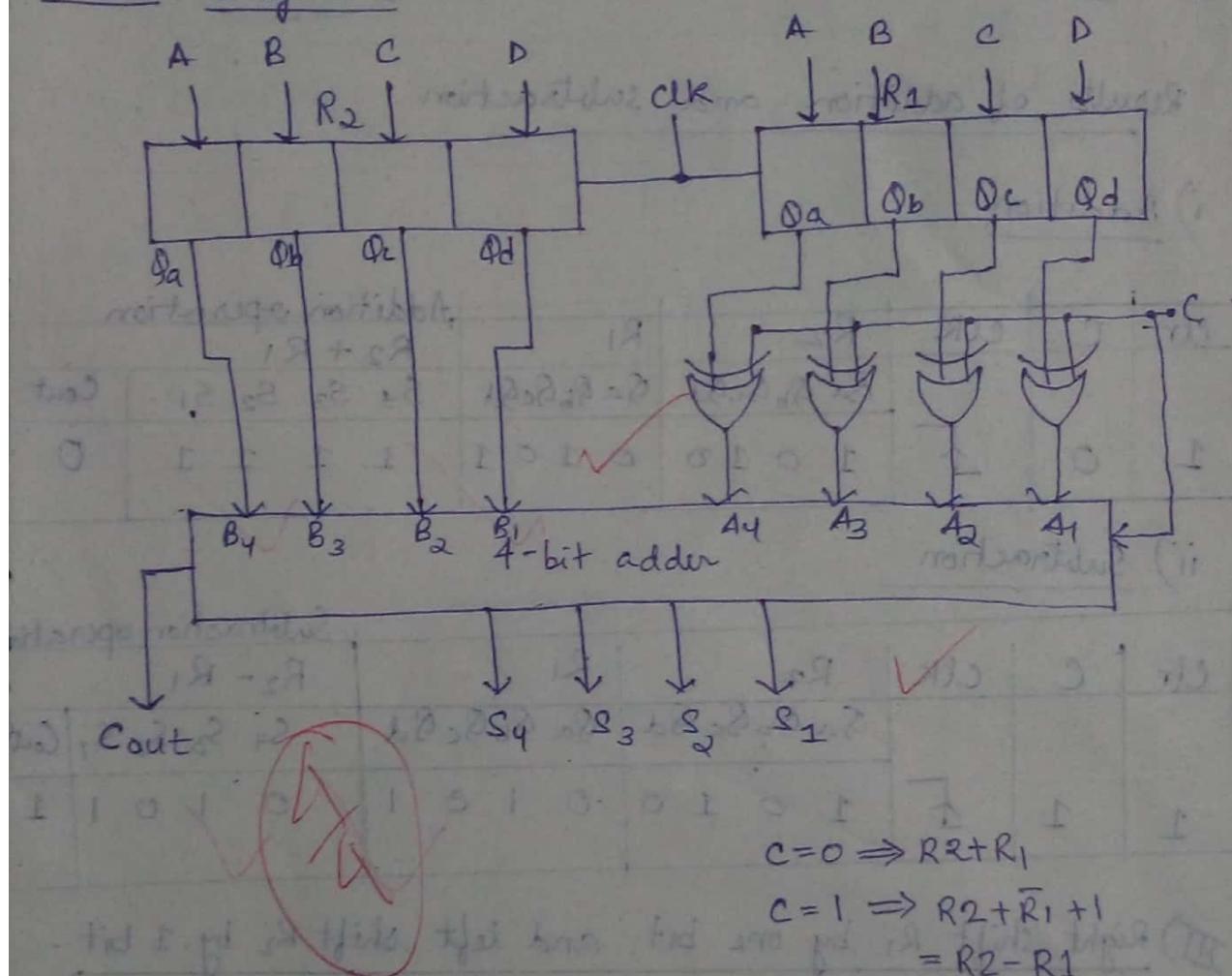
— * —

✓

Experiment No. 6

Objective : To realize addition and subtraction instruction in 2's complement.

Circuit Diagram



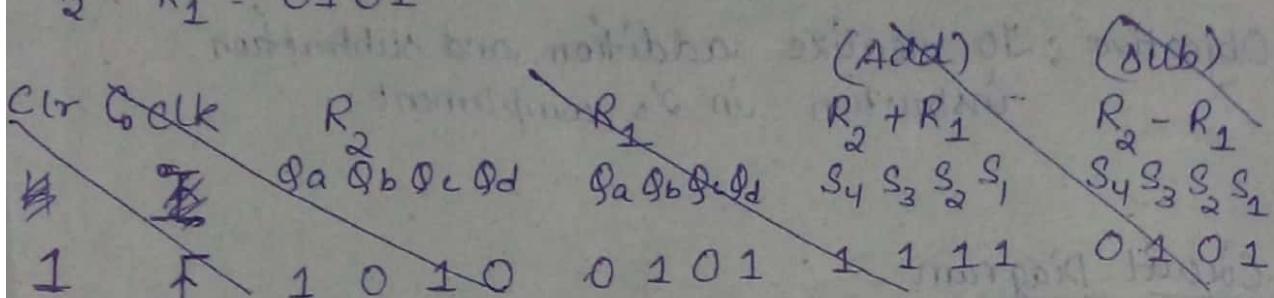
(I) Perform parallel load operation in R_1, R_2

Clr	Mode S ₁ S ₀	CLK	Parallel I/P A B C D	Output Q _a Q _b Q _c Q _d	Remarks
1	1 1	F	1 0 1 0	1 0 1 0	Load 1010 in R_2
1	1 1	F	0 1 0 1	0 1 0 1	Load 0101 in R_1

$$\text{II) } R_2 = 1010, \quad R_1 = 0101$$

$$R_2 + R_1 = 1111$$

$$R_2 - R_1 = 0101$$



Results of addition and subtraction

i) Addition

Clr	C	Clk	R_2	R_1	Addition operation	
					$R_2 + R_1$	Count
1	0	F	1 0 1 0	0 1 0 1	1 1 1 1	0

ii) Subtraction

Clr	C	Clk	R_2	R_1	Subtraction operation	
					$R_2 - R_1$	Count
1	1	F	1 0 1 0	0 1 0 1	0 1 0 1	1

III) Right shift R_1 by one bit, and left shift R_2 by 1 bit.

Clr	Mode $S_1 \ S_0$	Clk	Serial I/P S.Left-R.Right	Outputs $Q_a \ Q_b \ Q_c \ Q_d$	Remarks
1	0 1	F	X 0	0 0 1 0	Right shift R_1 by 1 bit
1	1 0	F	0 X	0 1 0 0	Left shift R_2 by 1 bit

IV Final result of R_1+R_2 and R_2-R_1

i) Addition

Clr	C	CLK	R ₂		R ₁		Addition		Cout
			Q _a	Q _b	Q _c	Q _d	Q _a	Q _b	
1	0	↑	0	1	0	0	0	0	1

ii) Subtraction

Clr	C	CLK	R ₂		R ₁		Sub		Cout
			Q _a	Q _b	Q _c	Q _d	Q _a	Q _b	
1	1	↑	0	1	0	0	0	0	1

→ * →