



211109

Computer Organisation and Design (Lovely Professional University)

computer organization and Design

Digital electronics: is a combinational circuits. word "Digital".

~~NLSM~~

Encoder, Decoder, Multiplexer, Demultiplexer

Combination circuits depends only upon the Present Input

Logic gates

Adder, XOR and XNOR, flip flops

~~Do invert~~
↓
input

- a) Combinational circuits don't have capability to store anything or state.
- b) Speed is fast
- c) easy to design
- d) Time is independent
- e) There is no feedback between input and output

Multiplexer examples:

- 1) Remote
- 2) Annocation
- 3) Broadcasts
- 4) Data communication
- 5) Computer memory

latest configuration
full adders, etc.

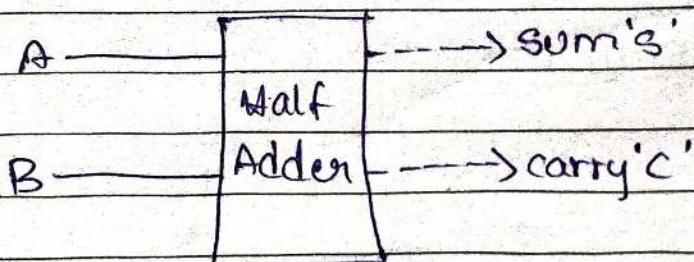
sequential circuits

flipflops, resistors, IC

- a) In this output depends upon present as well as past input.
- b) Speed is slow comparative to combinational circuits
- c) It is designed tough compare to "
- d) This is time dependent.
- e) There exists feedback b/w input and output

Half adder

Add two binary

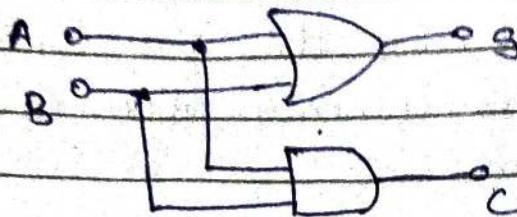


Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

If inputs are same then in XOR it
is "0".

Sum = $A \oplus B$

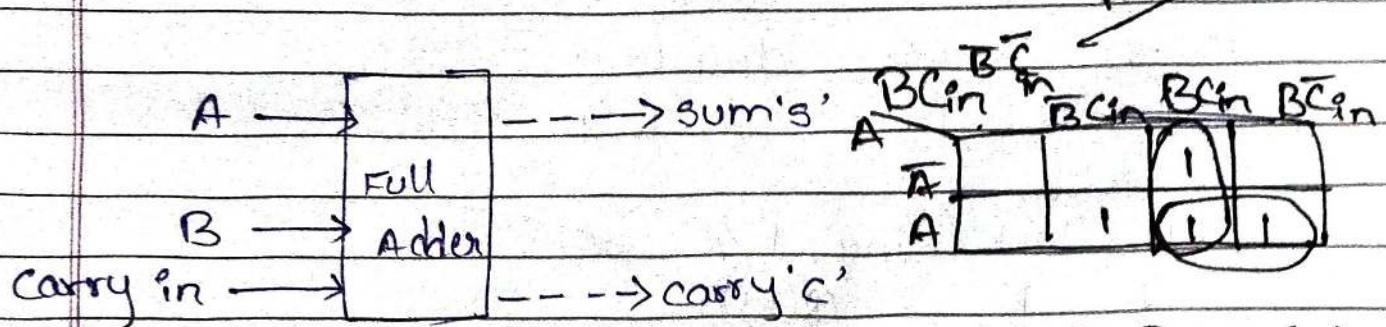
Carry = $A \wedge B$



K-map are used for expressions

A, B	A	B
A	00	01
B	00	11

Full Adder



$$C_{out} = AB + BC_{in} + C_{in}A$$

Inputs

Output

A	B	C-in	C-out	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0

flip flops

A circuit with two stable states used to store binary data (a storage element) used in computers, communications and many other types of systems.

SR (Set/Reset) F/F

S R Q(t+1)

0 0 Q(t) no change

0 1 0 clear to 0

1 0 1 set to 1

1 1 ? intermediate

D (Data) F/F

D Q(t+1)

0 0 clear to 0

1 1 set to 1

JK (Jack/King) F/F

J K Q(t+1)

0 0 Q(t) no change

0 1 0 clear to 0

1 0 1 set to 1

1 1 Q(t) complement

~~Book for
Computer architecture
and design~~

T (Toggle) F/F

T

$Q(t+1)$

O

$Q(t)$

no change

I

$Q'(t)$

complement

Master-Slave configuration in motherboard or keyboard.

Multiplexing $\rightarrow 2M$

Q: Draw the logic diagram/gates if all the 3 inputs O.

$2 \times OR$ and

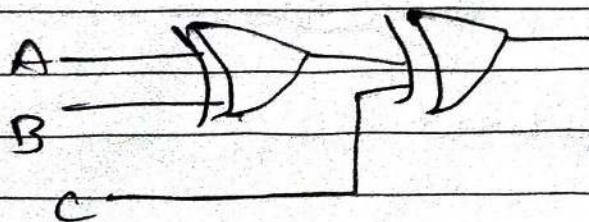
$$S = A'B'C + A'B'C' + ABC' + ABC$$

$$= A'(B'C + BC') + A(BC' + BC)$$

$$= A'(B \oplus C) + A(B \oplus C)' \quad (\because B \oplus C = P \text{ Let})$$

$$= A'P + AP'$$

$$= A' \oplus P \therefore = A \oplus B \oplus C$$

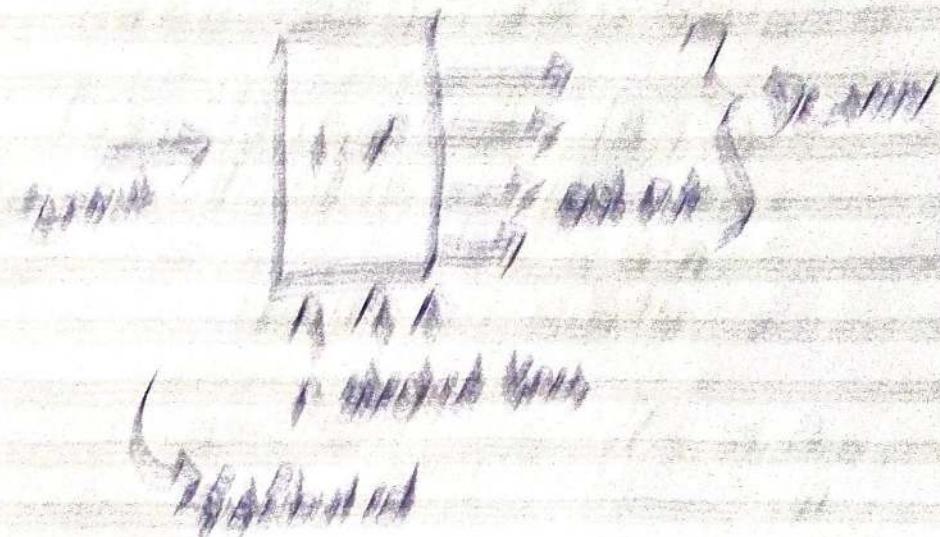


Scandinavia

W. H. C. W. H. C. W. H. C. W. H. C. W. H. C.

Minimally invasive

- 2011



Circuit \rightarrow Combinational or different devices.

Date: 1

Page: 1

MUX

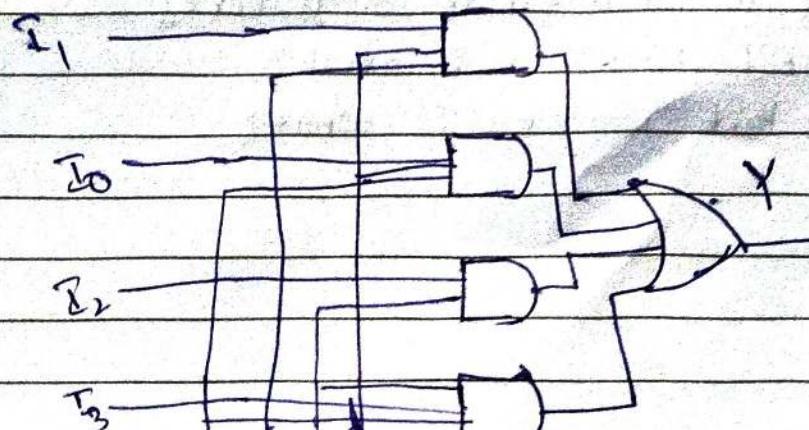
- 1) It is data selector
- 2) It has ~~or~~ input data lines
- 3) It works on many to one operational principle
- 4)
- 5) In time division multiplexing, demultiplexer used at ~~remitter~~ transmitter end

De-Mux

- 1) It is data distributor
- 2) It has only single ~~single~~ input
- 3) It works on one to many operational principle
- 4) Broadcasting
- 5) In time division multiplexing, demultiplexer used at receiver end.

DEMUX cannot be a switch

Select		OUTPUT
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



4×1

8×1

16×1

Date. /

Page. /

what is the application of Quad

Decoder \rightarrow changes binary into n output lines

Eg: $n=3$ (at input)

The output of encoder is 2^3 or 8 binary bits

$3 \times 8 \rightarrow$ decoder $\rightarrow n \times 1$

$n \times n \rightarrow$ encoder

In decoder maximum is 2^n , it can be less than too.

Register (Many flip flops)

1 flip flop = 1 bit.

Shift Registers

1 Register = many flip flops

2 two common types of registers:

1) parallel load Register

2) Shift Register

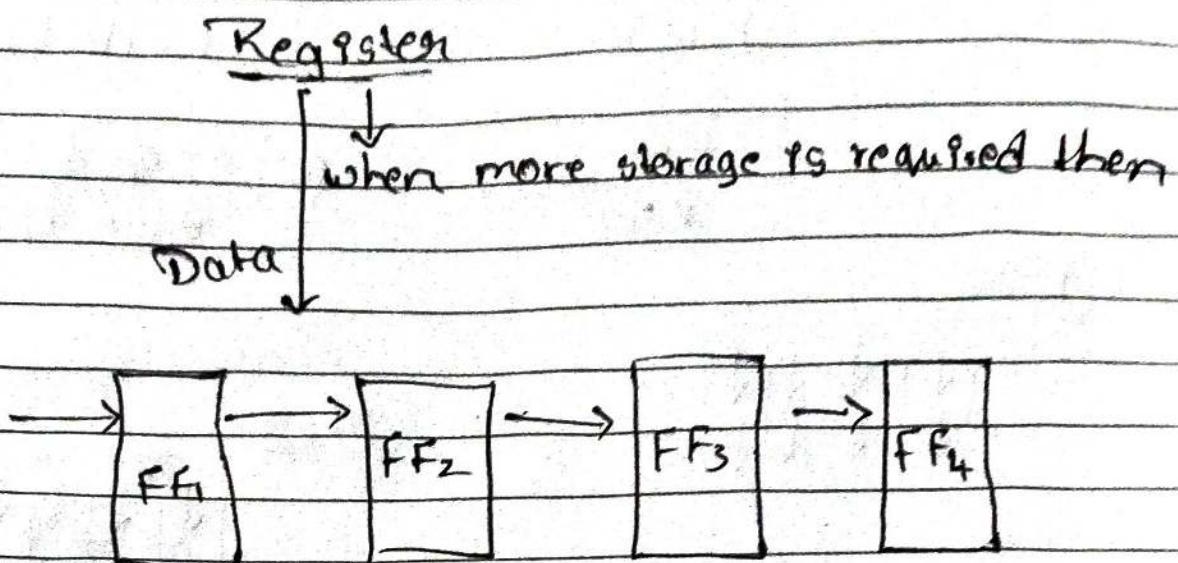
what is "1" flip flop?

why it is used in Registers?

4-bit Register is what?

8-bit Register is what?

combinational, Data stored, Data passed \rightarrow "D" flipflop



$$4 \text{ bit} = 4 \text{ FF}, \text{ Register} = \frac{4 \text{ bits}}{\text{bus}}$$

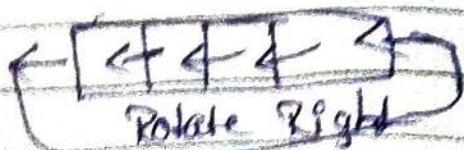
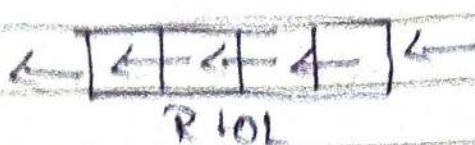
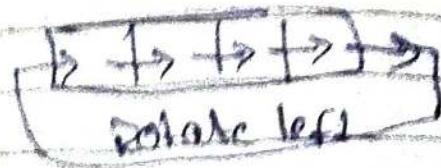
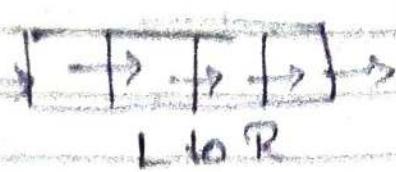
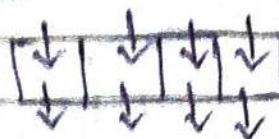
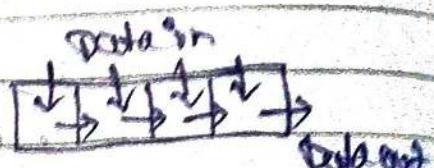
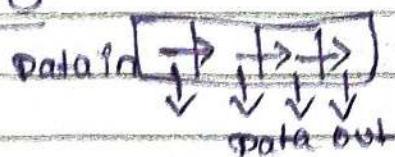
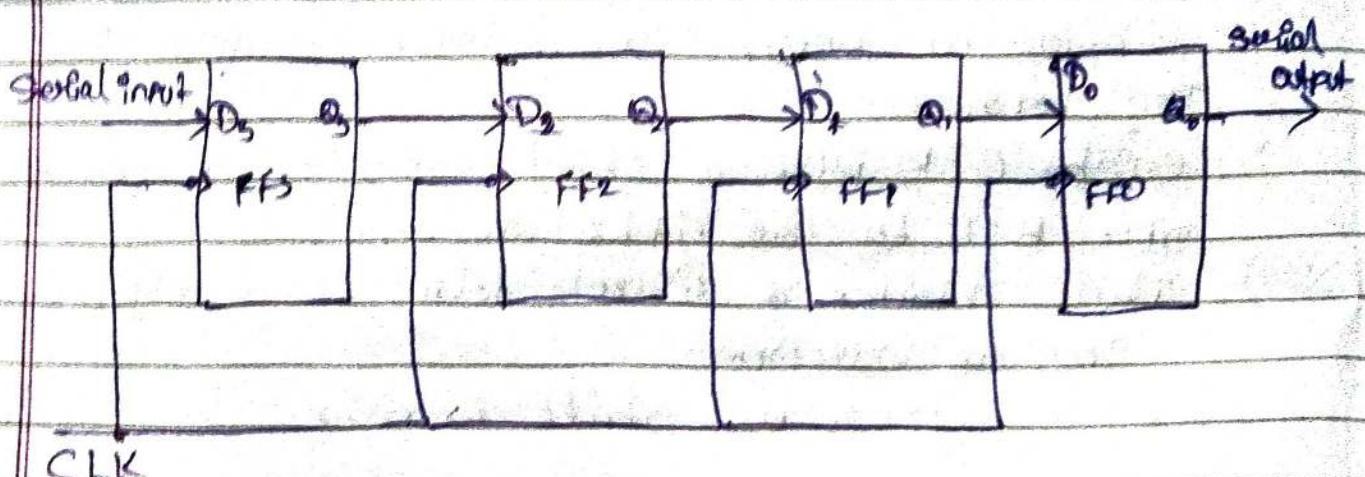
Shift Registers

- 1) used to implement arithmetic operations
- 2) within a register, data can transfer from one flipflop to another. such register is called shift registers.
- 3) each CLK PULSE shift Register content by one bit to the 'right' or 'left'.

* 4) They produce a discrete delay of a digital signal or waveform.

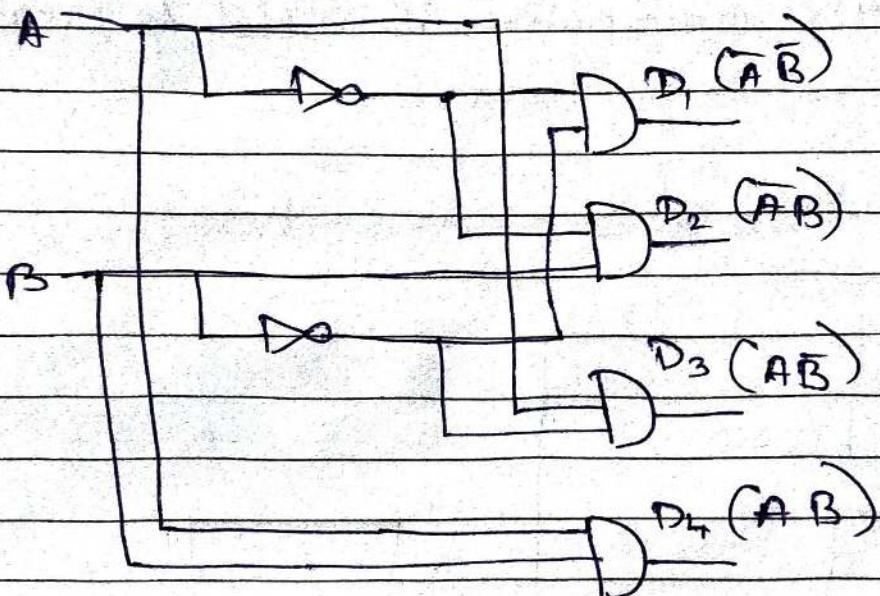
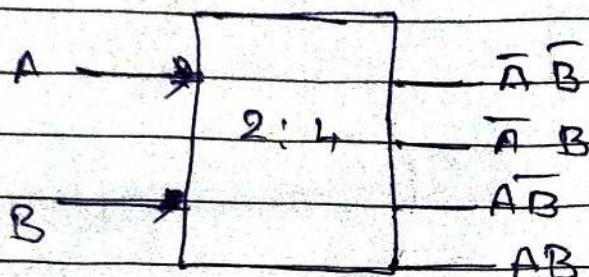
Four types of Shift Registers

- 1) Serial in serial out (SISO)
- 2) Serial in parallel out (SIPO)
- 3) Parallel in parallel out (PIPO)
- 4) Parallel in serial out (PISO)

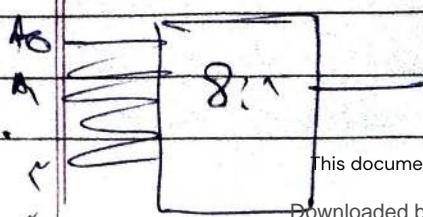
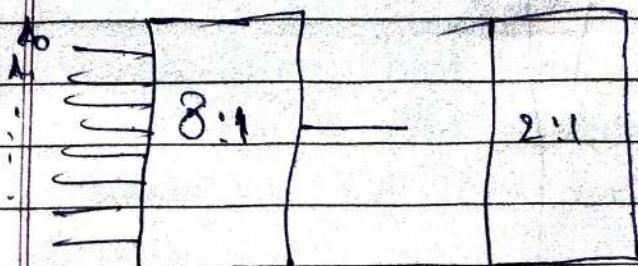
SI SOPIPOPISOSI POST SO

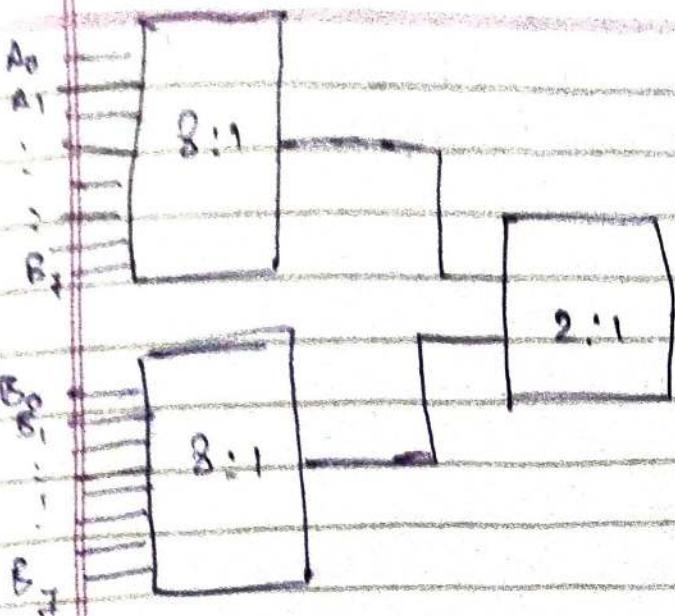
Q) Draw logic diagram of a 2 by 4 decoder
use one enable - bit / input / signal.

Ans 2 by 4 decoder

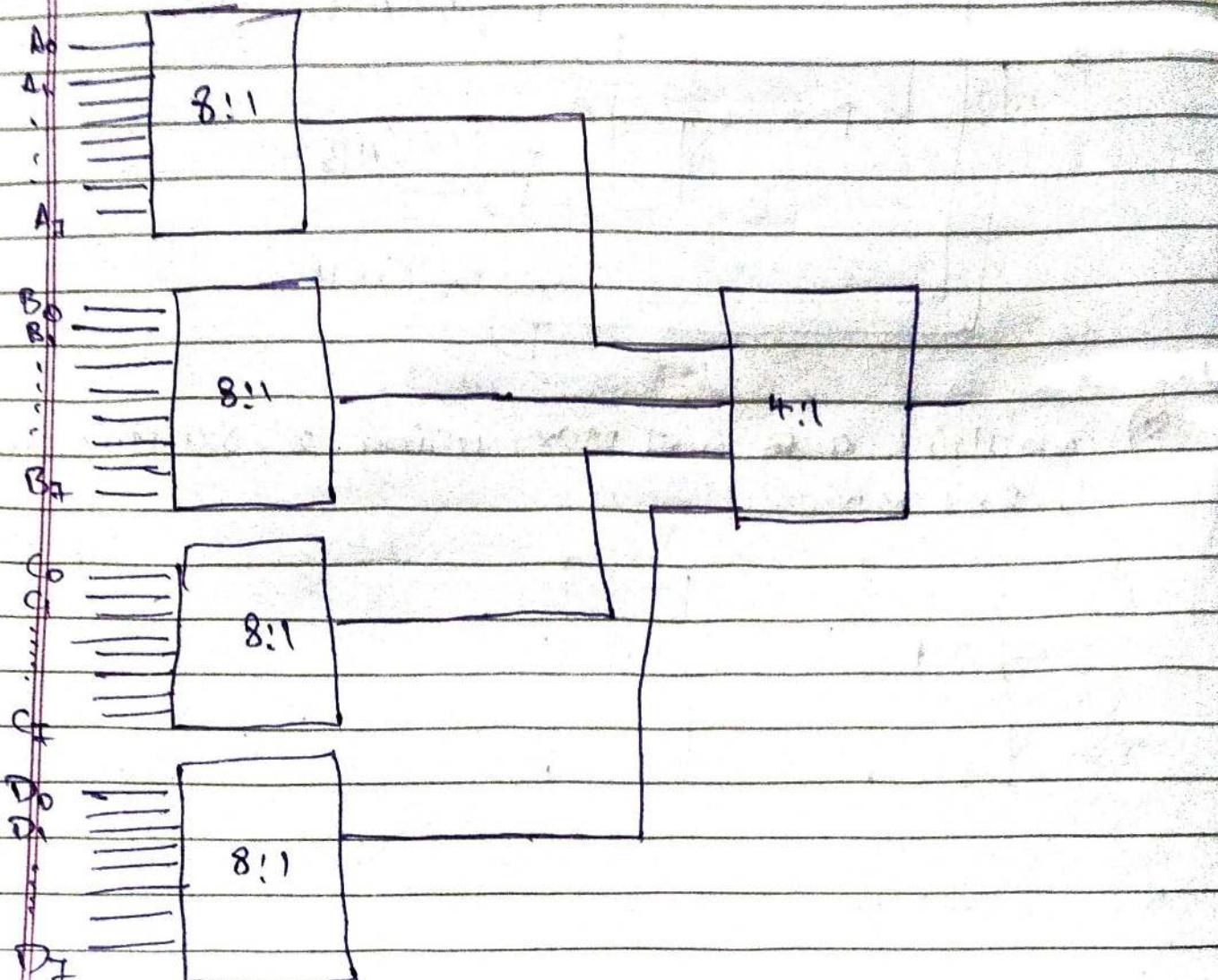


Q) construct a 16x1 MUX using 2, 8x1 MUX and 1, 2x1 mux.

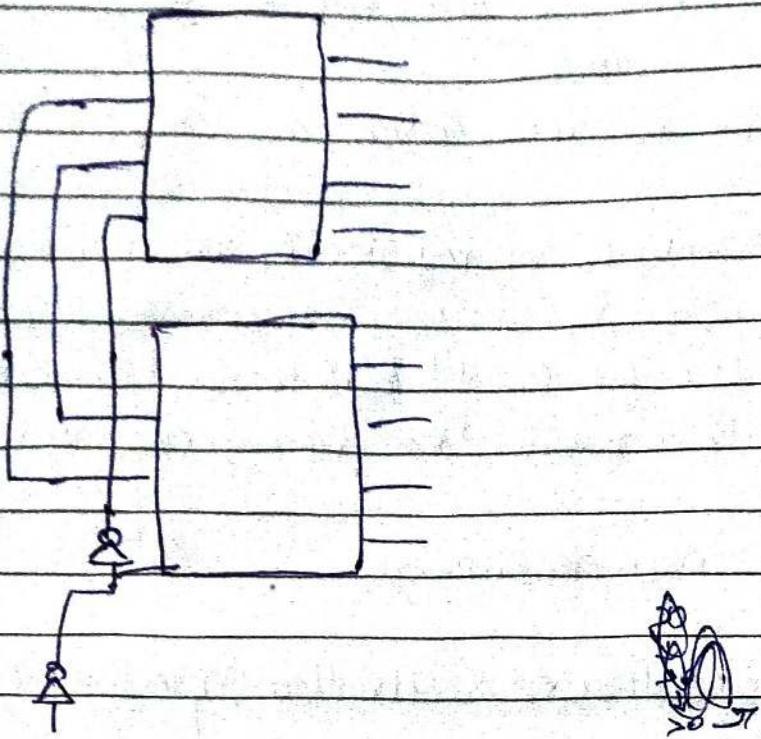




Q) Which MUX combination is useful to create 32x1 MUX.



Q) construct a 3×8 decoder using two, 2×4 decoders.



10 Marks

clocks needed for n-bit shift register

Mode	Loading	Reading	Total
SISO	n	n-1	2n-1
SIPO	n	0	n
PISO	1	n-1	n
PIPO	1	0	1

2M

Register Transfer Language

* Combination and sequential circuits can be used to create simple digital systems.

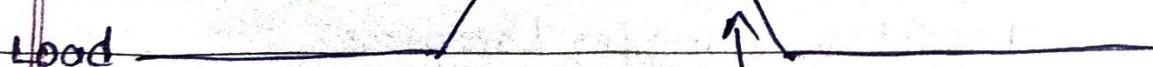
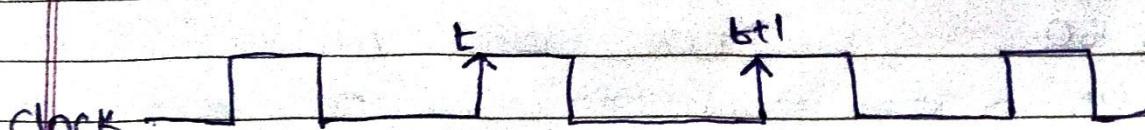
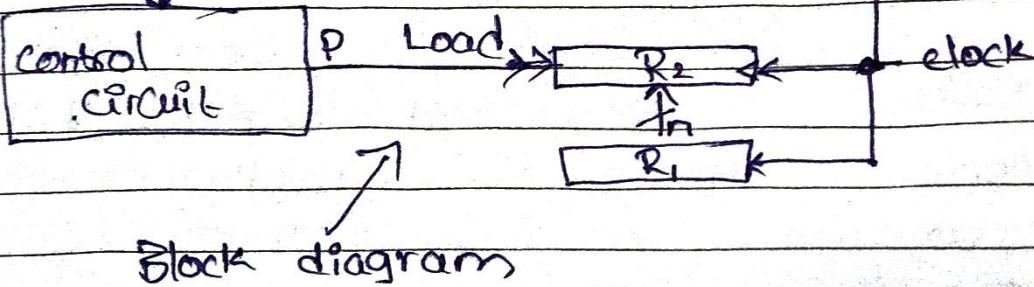
These are low-level building blocks of digital computer. Registers are used in digital systems, various micro-operation executes on registers data.

The notations used to represent such micro operation is called Register Transfer Language.

Register Transfer

- 1) copying the contents of one register to another is called Register Transfer
- 2) A register transfer is represented as $R_2 \leftarrow R_1$
- 3) In this case, the contents of register R_1 are copied (loaded) into register R_2
- 4) A simultaneous transfer of all bits from the source R_1 , to the destination register R_2 , during one clock pulse.
- 5) Note that this is non-destructive.

Hardware Implementation of controlled Transfers in RTG

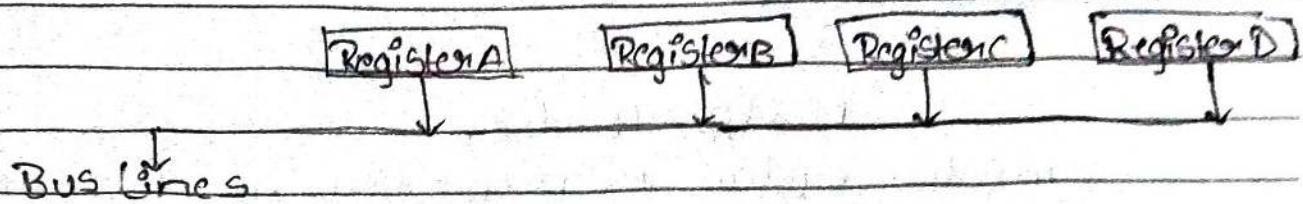


Timing diagram

connecting Registers - Bus Transfer

- consists of set of common lines, one for each bit of register through which binary information is transferred one at a time.
- Have control circuits to select which register is the source and which is the destination.

from a register to bus : BUS \leftarrow R



Common Bus system using Multiplexers:

- Q) Construct a bus system for 4 Registers.

Ans: common bus system can be constructed by using
1) Multiplexer

2) Buffer gate (Three-state Bus Buffer)

The multiplexer helps to select the source Register so based on this selection, so the binary information of selected registers will be placed on the Bus.

If we have four registers then selection of registers using multiplexer is shown in the function table:

S ₁	S ₀	(Selection of Register)
0	0	A
0	1	B
1	0	C
		D

Another method for bus structure is three-state gate. or It is also called use of high-impedance state.



$y = A$ when $C = 1$
circuit is disable if $C = 0$

A : normal input to buffer

Y : output from buffer-gate (i.e., high-impedance)

C : control input

Micro-operations

computer system micro operations are four types:

- Register Transfer micro operations
- Arithmetic Micro operations
- Logic microoperations
- Shift microoperations

(a) The basic arithmetic micro operations are

Addition

Subtraction

Increment

Decrement

(b) The additional arithmetic micro operations are

Add with carry

Subtract with borrow

Transfer/Load, etc...

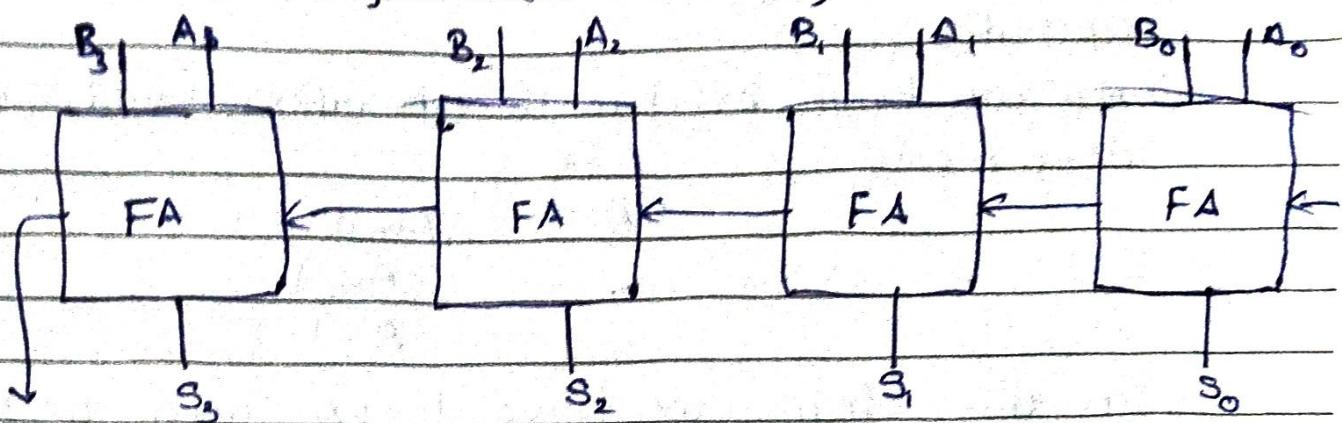
109

Topic

Binary Adder

→ 4 bit Binary Adder:

- (a) Full adder = 2-bit sum + previous carry
- (b) Binary adder = the arithmetic sum of two binary numbers of length
- (c) C_0 (input carry), C_4 (output carry)



→ As addition micro operation with carry or without carry can be implemented using binary serial adder as shown diagram.

- Here use full adders (FA) rather than half Adder(HA)
- In the case of one hardware circuit but will work for addition as well as subtraction then use full adders and XOR gates on 2nd register bits before actually passing those bits through full Adders.

Mode input M controls the operation

$M=0 \rightarrow$ Adder

$M=1 \rightarrow$ subtractor.

Half Adder (HA) \rightarrow 2 inputs

Full Adder (FA) \rightarrow 3 or more inputs.

Arithmetic Circuits

- The target of Arithmetic circuits is, 1 circuit design which can perform various Arithmetic operations.
- As we have Arithmetic operations (add, sub, incre, decrements)

$$\begin{array}{c} A+B \\ A-B \quad | \quad A+B'+1 \\ \hline A+1 \\ A-1 \end{array}$$

→ In this circuit we have 4-FA and 4 MUX
each MUX is of 4×1

→ The inputs to different MUX are required which can be 2nd register values.

→ 1st register bits can be directly given to FA's

→ As 4×1 MUX has 2 select lines.

so provide these and the side the input of 2nd register bits which will be input for FA

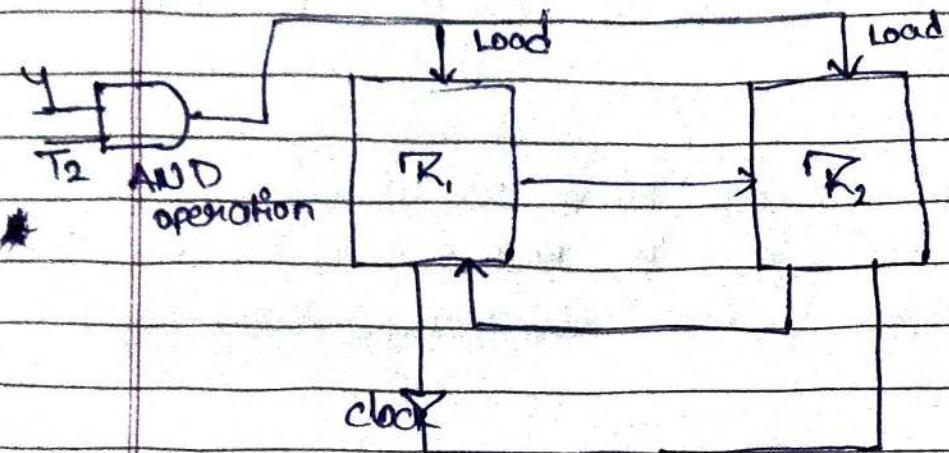
Select 1011	Input D_{in}	Micro-operation	Output
			$D = A + B + C$
0 0 0	Y	Add	$A + B$
0 0 1	B	Add with carry	$A + B + 1$
0 1 0	B'	Subtract "borrow"	$A + B'$
0 0 1 0 1	B'	Subtract	$A + B' + 1$
1 0 0	0	Transfer A	$D = A$
1 0 1	0	Increment A	$D = A + 1$
1 1 0	1	Decrement A	$D = A - 1$
1 1 1	1	Transfer A	$D = A$

Arithmetic circuit Function table.

10M Arithmetic Adder-subtractor with function table?

2M Provide the hardware implementation of following register transfer statement.

st :- $YT2 : R_2 \leftarrow R_1, R_1 \leftarrow R_2$

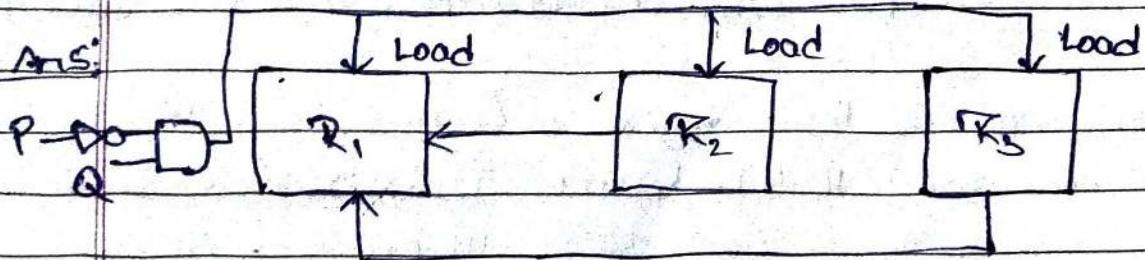


Q) Represent the following Conditional Control Statements

by 2 Registers transfer statement with control function.

if $P=1$ then $(R_1 \leftarrow R_2)$

else if $Q = 1$ then $(R_1 \leftarrow R_3)$



Representation:-

$$P: R_1 \leftarrow R_2$$

$PQ : R_1 \leftarrow R_3$

10 M

Logic Micro operations

Logic micro operations consider each bit of the register separately and treat them as binary variables

$$P: R_1 \leftarrow R_1 \oplus R_2$$

1010 content of R_1

1100 content of R_2

0110 content of R_1
after $P = 1$

- 1) There are total 16 logic Micro operations.
- 2) Take 2 Registers A and B. Both are 4-bit registers.
Let's say X and Y represents various combinations of both registers.

3)

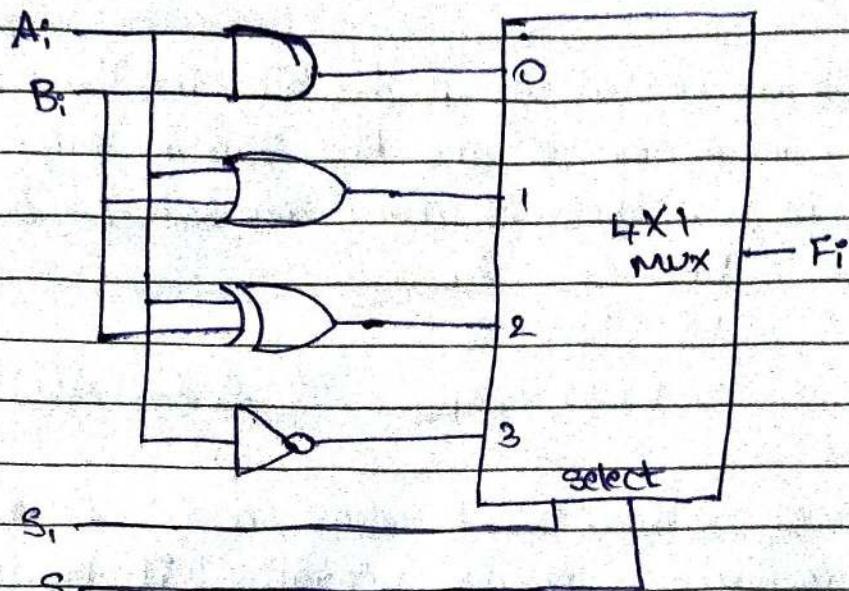
X	Y	F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	0	1	0
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

→ Truth Table for 16 functions of 2 variables

2 M

- ⑥ out of various logic micro operations, most circuits recommends only 4 which are
 - i) AND
 - ii) OR
 - iii) XOR
 - iv) Complement.

S_1	S_0	OUTPUT	M -operation
0	0	$F = A \wedge B$	AND
0	1	$F = A \vee B$	OR
1	0	$F = A \oplus B$	XOR
1	1	$F = A'$	complement



QOM Applications of Logic Microoperations

Logic microoperations can be used to manipulate individual bits for portions of a word in a register.

- 1) Selective-set: Take a register (name = A). Use this register as input data register. Now, take another register (name = B) which will be used to modify A register data.
- 2) Suppose A register has the value/bits/data/content 1100 use a bit pattern in B to set certain bits in A .

If bit in $B = 1$ then A will also set same position same bit also 1

1 bit = 1 byte

1100 A

1010 B

1110 A_{left} (A ← A + B)

If a bit in B is set to 1, that same position in A gets set to 1, otherwise that bit in A keeps its previous value.

- ② In a clear operation if the bits in the same position in A and B are the same, they are cleared in A, otherwise they are set in A

1100 A_{left}

1010 B

0110 A_{left} (A ← A ⊕ B)

109

Daniel

③

Daniel

- ③ Insert operation: It uses mask operation. After applying mask operation, it performs OR operation.

Shift Micro Operations

Symbolic destination

Description

R ← shl R

Shift Left Register R

R ← shr R

Shift Right register R

R ← cil R

Circular shift left " R

R ← cir R

Circular shift Right " R

R ← ashl R

Arithmetic shift Left Register R

R ← ashr R

" " Right " R

There are three types of shifts:

- 1) logic shift
- 2) circular shift
- 3) Arithmetic shift.

Arithmetic shift

- It shifts a signed binary number to left or right.
- 1) An arithmetic left shift multiples a signed number by two.
- 2) An arithmetic Right shift divides a signed number by two.
- Sigbit: 0 for positive and 1 for negative.

11101100

00010011

~~11111111~~

00010011

1

~~0010100~~

11101100

00010011

~~00000~~

② 11001100

Apply left shift and justify every left shift is a multiplication of every time.

Daniel

11001100

00110011

~~00000~~

00110011

+1

~~00110100~~

Kavita

1100

0011

+1

0100

-4

?? ?? ?? -54 -52

- ③ How many 128x8 RAM chips are required or needed to provide a memory capacity of 2048 bytes?

sol:

Let "n" is the
no. of chips.

$$128 \times 8 \text{ RAM} = 1024 \text{ bytes}$$

$$1 \text{ bit} = \frac{1}{8} \text{ byte}$$

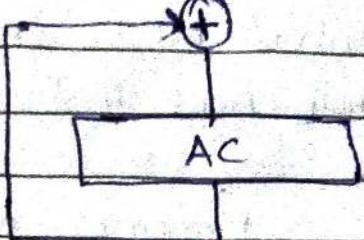
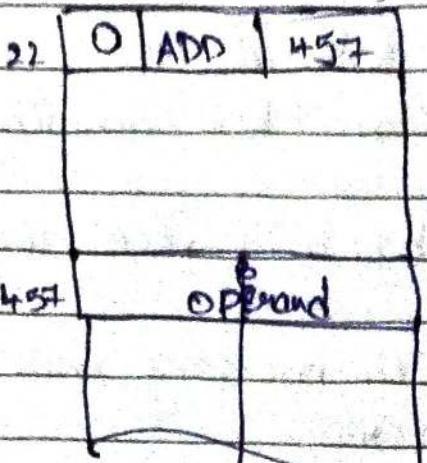
$$1024 \text{ bytes} = \frac{1024}{8} \text{ bytes}$$

$$= 128 \text{ bytes}$$

$$2048 \text{ bytes} = 128n \Rightarrow n = 16 \text{ chips}$$

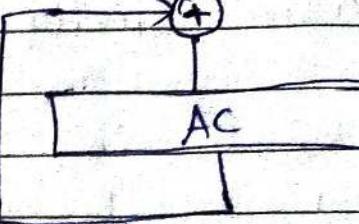
Addressing Mode:

Direct Addressing



1350

Operand



Basic computer has

- 8 registers but only 7 registers output is connected to the bus.
 - It has 1 memory unit, it has 1 arithmetic or control or logic unit.
 - There is one common bus on which data, address or control can transfer.
 - Information Transfer can be 1 register to another Register $R \rightarrow R$
 $R \rightarrow$ Memory or
 $\text{memory} \rightarrow R$
- Use one common bus rather than wires in bulk to connect individual components (different) of computer.
- As 16 registers are connected with bus when we want to transfer on bus so, use minimum 3 selected lines are required.

GU(05) 2 M

Hardware implementation of Shift Microoperation:



UNIT-3



Every processor has its own design (different registers, buses, microoperations).

(Machine) Instruction:

A group of bits that tell the computer to perform specific operation (a sequence of micro operation). The instructions of program, along with any needed data are stored in memory.

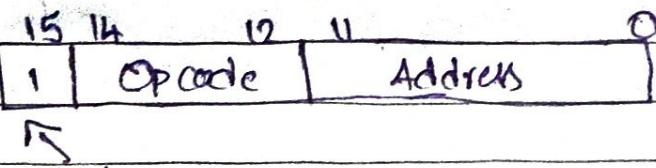
The CPU reads the next instruction of the memory. It is placed in Instruction Register (IR).

Instruction Format:

A computer instruction can be often divided into two parts:

1) An Opcode (Operation code) that specifies the operation for that instruction.

2) An Address that specifies the registers and/or locations in the memory to use that operation.



Addressing mode.

Addressing mode: 0 → Direct and
1 → Indirect.

* 12 bits are required for Address, 3 bits are for Opcode and 1 bit is required for specifying Addressing mode.

Memory Unit

- There two states (a) write (b) Read.
- When write is enable, it means bus contains are available to write in MU.
- When read is enable, then bus will read from memory unit.
- The registers which deal with memory address are 12 bit or Address registers.

- ✓ (a) In a bus system, following control inputs are active specify the RTL at their next instruction.

S ₂	S ₁	S ₀	Type of Register	Memory Address
1	1	1	IR	Read
1	1	0	PC	-
1	0	0	DR	Write
0	0	0	AC	Add

Hint:

S ₂	S ₁	S ₀	Register
0	0	0	X
0	0	1	AR
0	1	0	PC
0	1	1	DR
1	0	0	AC
1	0	1	TR
1	1	0	TR
1	1	1	Memory

If Memory Read then M at which side of RTL?

If memory write opt & M at which side of RTL?

d) Relavent to half adder

" " " 4 bit binary adder

Adder, Subtractor.

3-8 line decoder (com) 2-4 line decoder

3-8 @ decoder but 2-4 decoder, using
It can be 4x1, 8x1, 16x1 mux diagram. (5M or 10M)
4 bit shift register.

Addressive ~~mode~~ modes

Common Bus system.

Arthematic Circuits, function table.

construct 8x8 decoder using two 2x4 decoders.

Basic computer instruction formats.

15 14 12 11 0

I	· Opcode	Address	0
---	----------	---------	---

(opcode = 000 through 40)

(a) Memory - reference instruction

15 12 "

0 · 1 1 1	Register operation
-----------	--------------------

(opcode = 111, J = 0)

(b) Register - reference instruction

15 12 "

1 1 1 1	I/O operation
---------	---------------

(opcode = 111, J = 1)

(c) Input - output instruction.

MTE

Flow chart (Instruction Cycle) → △

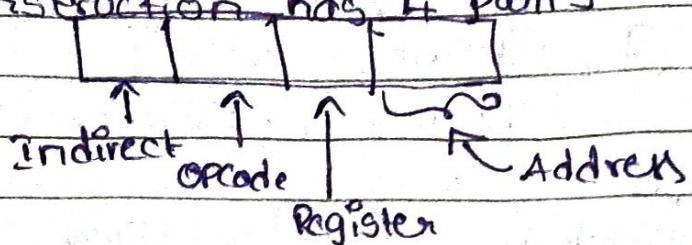
Control unit:

- 1) Hardwired control 2) Microprogrammed control
- Total timing signals will be 16 (T_0, T_1, \dots, T_{15})
- Also use one more 3×8 decoder for various operations.

There are 2 types of instructions direct
indirect

- (1) How many reference to memory are needed for each type of instruction to bring an operand into a processor register.
- (2) A computer uses a MU of ~~256~~ 256 K words and 32-bit each.

A binary instruction code is stored in 1 word memory and instruction has 4 parts



- (3a) How many bits in each → (for 64-bit Register)
- (3b) Draw instruction format for it
- (3c) How many bits for data and address inputs of the memory.

- (4) Draw a timing diagram (same like we did in Timing and control)

Assume that $SC \leftarrow$ clears to zero at T_3 and C_7 is active.

Note: C_7 is active with positive transition associated with T_1 ,

$$C_7 T_3 : SC \leftarrow 0$$



If multiple have using unique so can we job the job that is dependent of

Flags FULL

empty LIFO

↓

SP
(Stack Pointer)

Last in first out.

NOTE

(a) Stack organisation →

↓

Ans.

Point