

Visual display, audio o/p u
located easily on a home

In this, the

CD-ROM's and diskettes.

portable "

version of PC, here all

single unit, the size of

"Work station

i/o capability. It has the

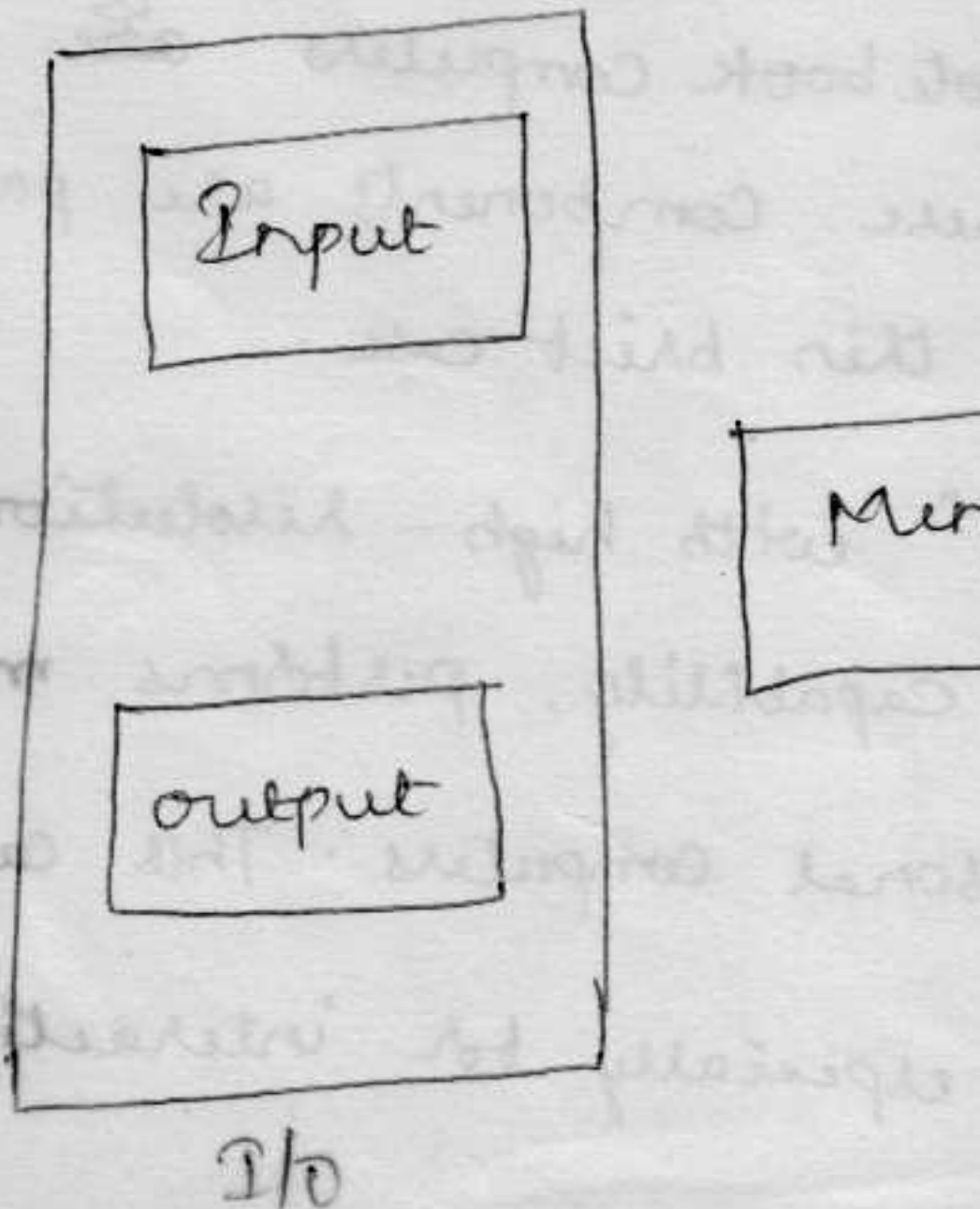
computational power than P

in engineering applications

A Computer consists of independent main parts

- 1) I/p 2) Memory 3) All

The below fig. shows



controlled by the stored program
interruption by an operator
machine.

Data are numbers
used as operands by the
operation used to any digital
a HLL source program into
constituting a machine language
program.

The source program
program which translates
language program.

units; which read the data
device is the keyboard.

Corresponding letter or digit
its corresponding binary code
either memory or the processor.

Memory unit :-

The function of the
& data. There are two
& secondary storage.

Primary storage is
electronic speeds. programs run
while they are being executed.
large numbers of semi-conductors

programs m
execution. Instructions &
or read from the memory
Memory in
a short & fixed amount of
called Random-access memory
access one word is called
The mem
-ented as a memory hierarchy
RAM units with different
RAM units is cache memory
The large
the main memory.

with all these

world (or user). The most

Some units, such as the CPU, perform both an o/p function & an

Control unit :-

The memory, arithmetic and process information & control unit is effectively the nerve center of the computer, connecting to other units & senses their

The actual operations are determined by 'timing signals' which determine when a given action

The operation of a computer follows

be used for only one transaction.
actively use the bus at any time.
used to arbitrate multiple transactions.

The main thing is
Cost & its flexibility for a system.

Systems that support
Concurrency in operations by allowing
Carried out at the same time
but at an increased cost.

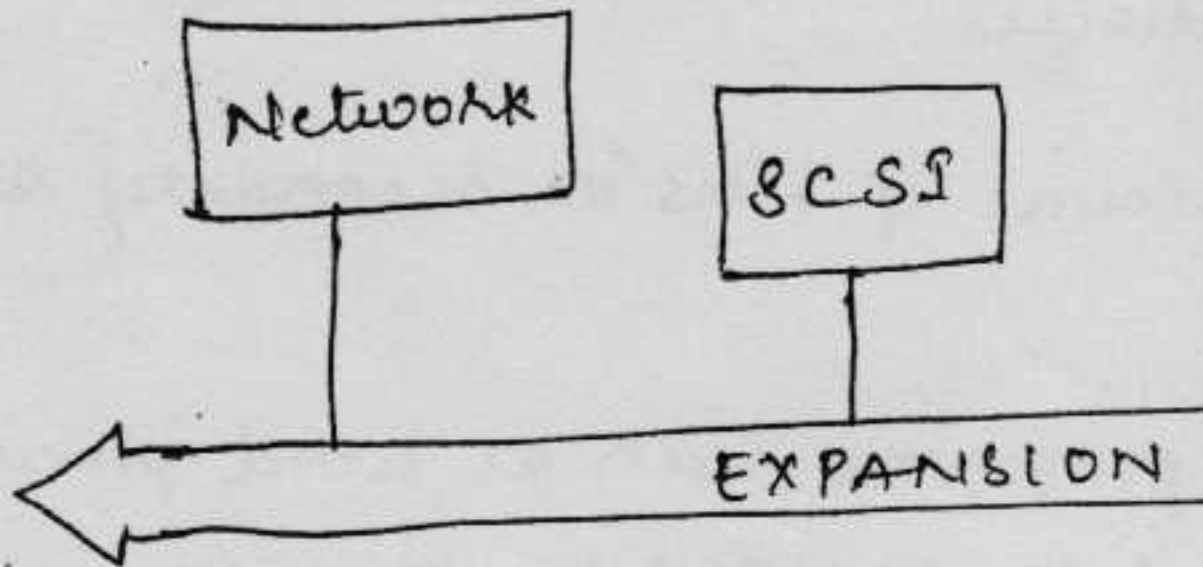
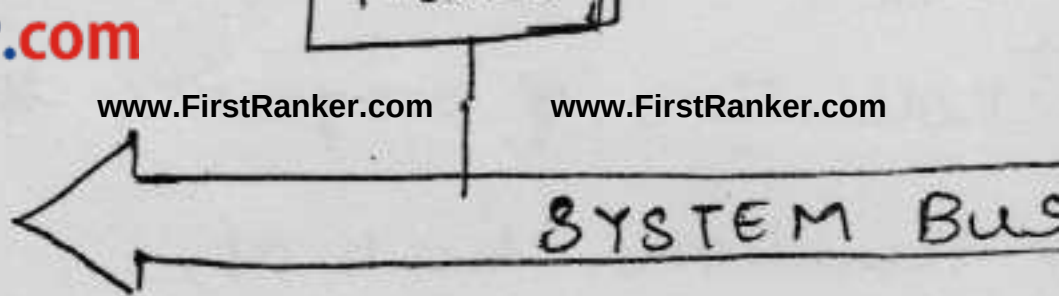
Memory & processing
speeds, making them the backbone.
These devices must communicate
efficient transfer mechanism
devices & that can be used

All processors have
such systems, & the term
systems. The high performance
increased complexity & cost
and memory units, cost is
more complex interconnection

In contrast
an interconnected group of
total computational power
only to their own memory &
executing need to communicate
exchanging messages over a

System software is executed as needed to perform

- Receiving & interpreting
- Entering & editing application files in secondary storage
- Managing the storage & retrieval of data
- Running standard applications such as spreadsheets or games, word processing, etc.
- Controlling I/O units to receive input & send results.
- Translating programs from source code to machine code
- Linking & running user-written programs



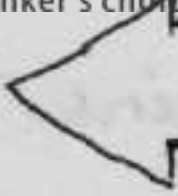
There is a
to a Cache memory and to
devices. The Cache memory
System bus also, which
memory modules.

A serial port could be
This architecture
break down as higher and
I/O devices.

In addition to,
approach taken by industry
requiring a bridge between
bus.

This arrangement is
architecture.

The fig. shows the
architecture.



Expansion

N = No. of actual mach

S = steps to be taken

R = clock rate per se

pipelining and Super scalar

The improvement
by overlapping the execution
'pipelining'.

When multiple no. of
by creating parallel paths
can be executed in every
operation is called 'SuperSc

Compiler :-

A Compiler translates
into a sequence of machine
to have a suitable machine
Compiler will provide good

Performance Measurement :-

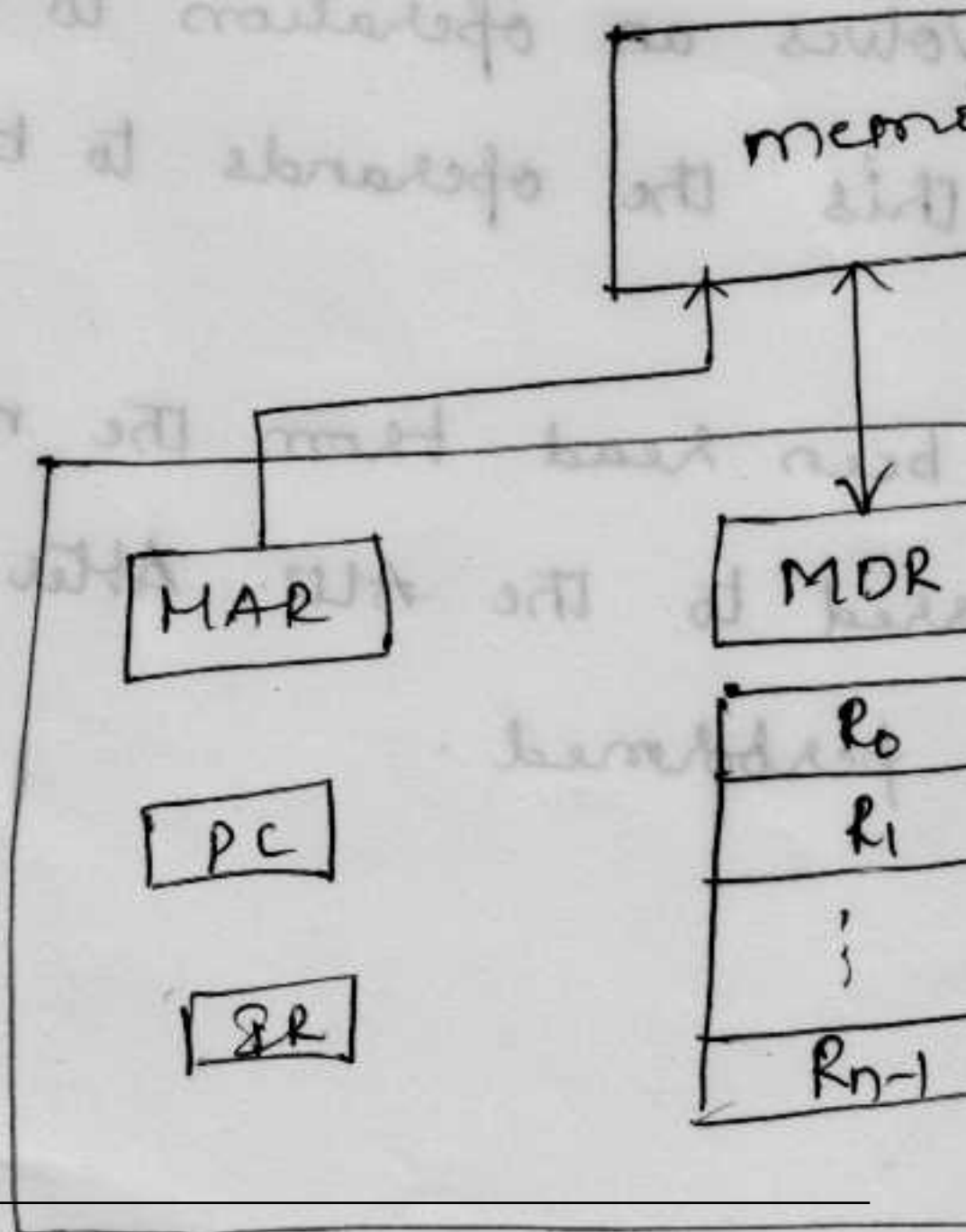
The idea of measuring
benchmark programs. A

System performance Evaluation
& publishes representative
application domains

to be accessed into the memory
appropriate control signals

The below

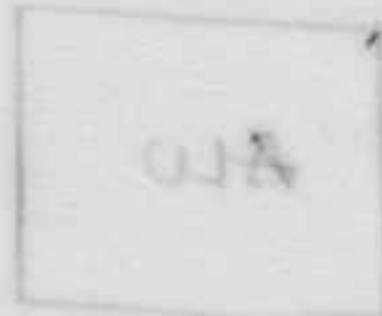
can be done between the



The contents of the PC are
a 'Read' control signal
that, the word is read out
into the MDR.

The contents of the
IR. If the instruction is
performed by the ALU, for
from memory.

When the operand has
into the MDR, then to transfer
the desired operation to



is necessary to
power of 'h' and
weighted digit

Decimal :- The decimal
10 system.

The digit 82

$$8 \times 10^2 + 2 \times 10^1 +$$

binary :- The binary n

two digits n

101101 is inte

$$1 \times 2^5 + 0 \times 2^4$$

The decimal number

$$(F3)_{16} = F \times 16^1 + \dots$$

Conversion:-

The conversion of

done by first separating
and fraction part +6875.

dividing 41 by $2=2$ to
remainder of 1. The qu
give a new quotient and
until the integer quotient
remainder will give the dec
binary number.

rep. plays an important role

$2^3 = 8$ & $2^4 = 16$. Each octal digit

hexadecimal digit represents

from binary to octal is

binary to hexadecimal is

eg.
 $\begin{array}{ccccccc} & A & & F & & 6 & \\ \hline 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & \\ \hline 1 & 2 & & 7 & & 5 & & 4 & \end{array}$

Octal
Number

Binary - code

0	000
1	001
2	010
3	011
4	100
5	

14

0001 0100

78

1111 1000

Decimal representation:-

The binary number system for a computer, but decimal system. One way

i/p decimal numbers into

<u>Decimal</u>	<u>BCD</u>
0	0000
1	0001
2	0010
3	0011

10 0001 0000

148 0010 0100 1000

a number that consists

$10^n - 1$ is a number repre

for eg: with $n=4$, we have

The 9's complement is
each digit from 9.

for eg: 9's complement

$$999999 - 5467$$

$$99999 - 1238$$

1's Complement

For binary numbers, the

of N is $(2^n - 1) - N$ for eg:

The 1's complement of a

the 9's complement.

The 2's complement

Subtraction of unsigned

• The direct method
concept

The subtraction of

$M - N$ ($N \neq 0$) in base 'r'

1. Add the minuend M

Subtrahend N . This produces

2. If $M \geq N$, the sum will

be discarded, and the

$$59282$$

$$M = 13250$$

$$10's \text{ Complement } N = \begin{array}{r} 27468 \\ \hline \end{array}$$

$$\text{Sum} = 40718$$

Answer is negative 59

Binary Numbers :-

$$X = 1010100 \quad \&$$

*

$$X = 1$$

$$2's \text{ Complement of } Y = \begin{array}{r} \\ \hline \end{array}$$

$$\text{Sum} = 10$$

$$\text{Discard and carry } 2^7 = -10$$

~~Yes~~

Integer Representation

When an int
the sign is represent
positive binary number
is negative, the sign
of the number may be
ways.

1. Signed magnitude
2. Signed-1's Complement
3. Signed-2's

The signed mag
consists of the magnit
2, we are going to

Arithmetic Addition :-

The addition of two system follows the rules.

- If the signs are the same, add the magnitudes and give the sum the same sign.
- If the signs are different, subtract the smaller magnitude from the larger and give the result the sign of the larger.

$$\begin{array}{r}
 +6 \quad 00000110 \\
 +13 \quad 00001101 \\
 \hline
 +19 \quad 00010011 \\
 \hline
 \end{array}$$

Take the 2's complement
the sign bit) and add it
sign bit). A carry out

A subtraction operation
addition operation it

This can be demonstrated

$$(\pm A) - (+B) =$$

$$(\pm A) - (-B) =$$

But changing a positive
is easily done by taking
is also true.

The detection of an overflow depends on the width of the registers. If two binary numbers depend on the sign bit, they are considered to be unsigned. When two numbers are added, an overflow occurs if the end carryout of the MSB is not equal to the left most bits. Always, if the left most bits always are in 2's complement, numbers are in 2's complement. When numbers are added, the overflow occurs if the number and the end carryout are not equal. Overflow.

An overflow doesn't occur if the number and the end carryout are equal. If the number is +ve and another is -ve, an overflow occurs.

If these two
an overflow will be de
equal to '1'.

Decimal Fixed-point

The representation
is a function of the binary
decimal digit. A 4-bit
bits for each decimal
4385 in BCD requires 16
each digit.

0100 0011 1

By representing numbers
considerable amount of

of the decimal point is
indicated decimal point

The equivalent is

The floating-point is

mx

Only the mantissa m is
represented in the register

A floating-point bin
similar manner except

Eg: $+1001.11$

Fraction

01001110

than the fixed-point number
place longer & requires

Error Detection Codes :-

An error detection code is a digital error during transmission that can't be corrected but that

parity bit :-

The most common error detection code is a parity bit. A parity bit is a binary message to maintain either odd or even.

are applied to a parity
parity adopted (odd or even)
checked parity doesn't

The parity method detects
any odd number of errors
is not detected.

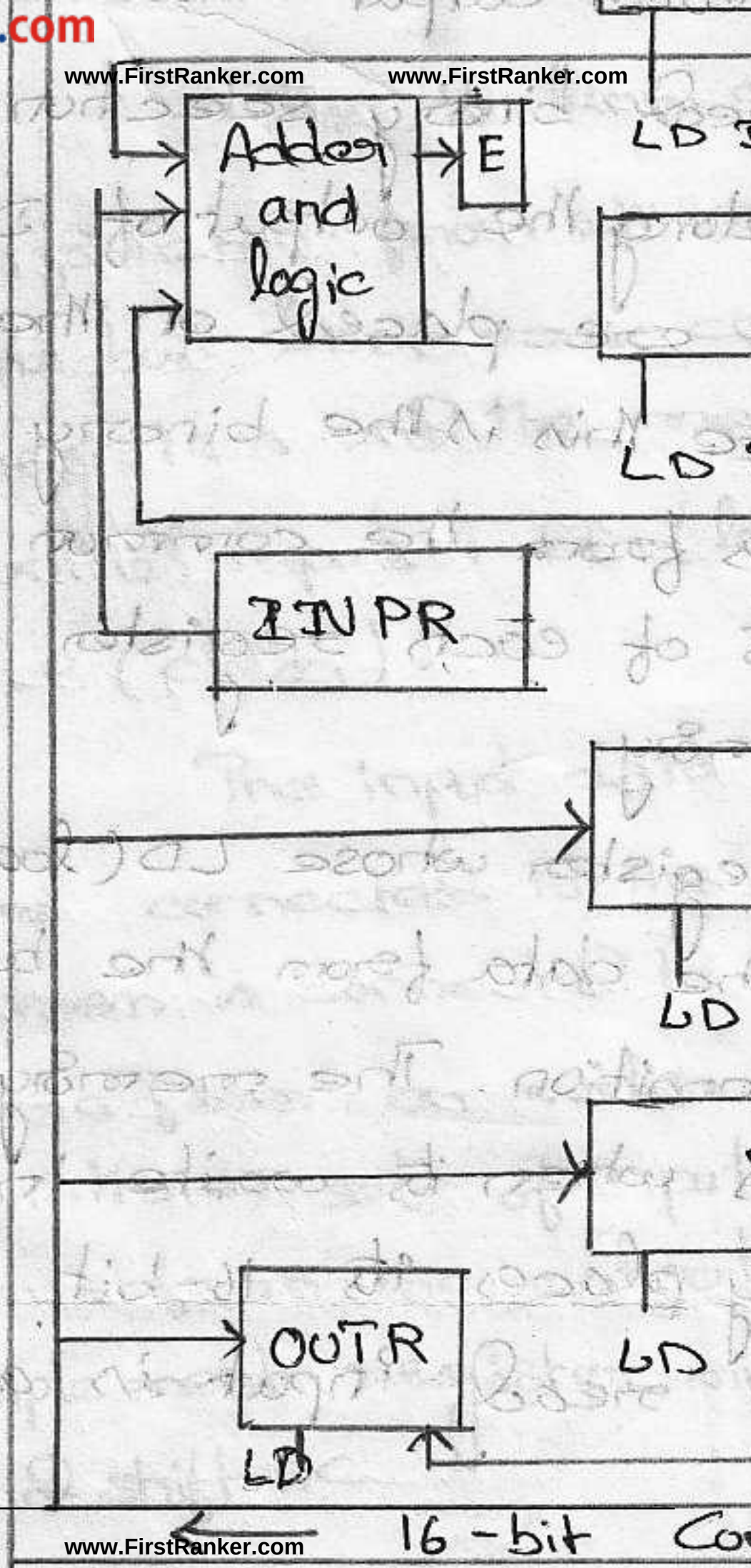
Parity generator &
with X-OR functions. An odd
value is binary 1 if an
According to this definition,
 x, y, z because it is equal to 1.
The variables are equal to 1.
Complement of the $p(\text{even})$ &

parity generator

This circuit consists of 0
is the CX-OR of x, y, z & $P(\text{odd})$

The message & the odd - p
destination where they applied
occurred during transmission it
is even, since the binary int
odd.

The o/p of the parity checker



operation is an elementary
information stored in one
the operation may be stored
may be in another register

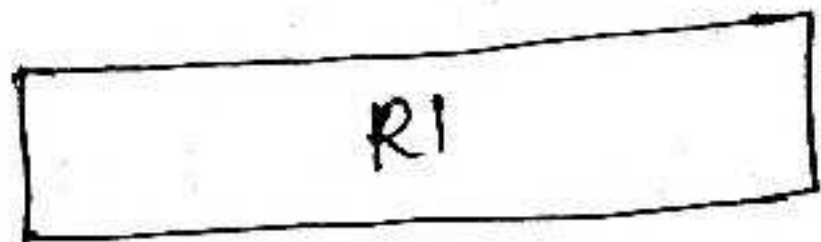
The internal
digital computer is best

- 1) The set of registers it
- 2) The sequence of microop
information stored in
- 3) The control that initi

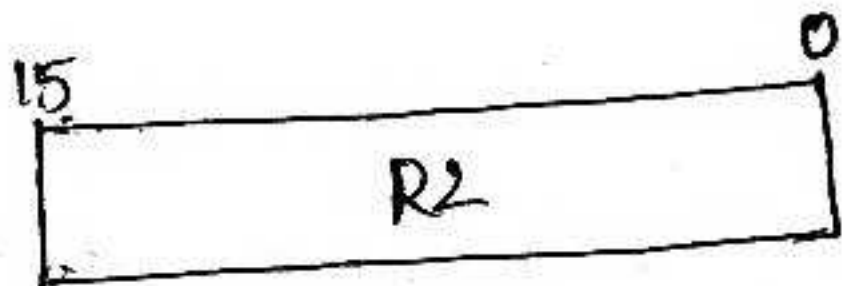
The symbol
microoperations transfers

individual bit-blobs in a
0 through $n-1$, starting
and increasing the number

In fig. shows the rep-
diagram below.



a) Register R



c) Numbering of bit

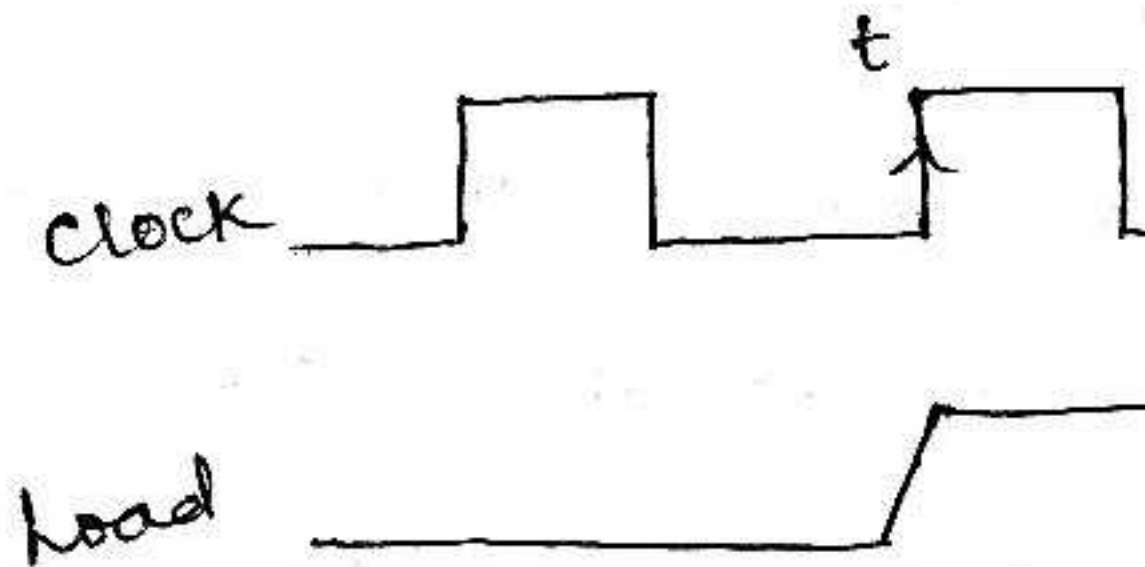
can't be changed after
A statement that is
that circuits are available
register to the inputs of the
destination register has a
want the transfer to occur
control condition. This is
if-then statement.

$$\text{If } (P=1)$$

Where 'P' is a control signal.
It is sometimes convenient
from the register transfer
function". A control function

Fig: Transfer to
The 'n' old's of register
of register R2.

It is assumed that
with the same clock as the



Then

Comma (,)

Separate

The statement $T \leftrightarrow R$ operation that exchanges during one common clock simultaneous operation have edge-triggered flip

Bus and Memory Transfer

The most efficient bus registers is a multiple common bus system. A bus common lines, one for each

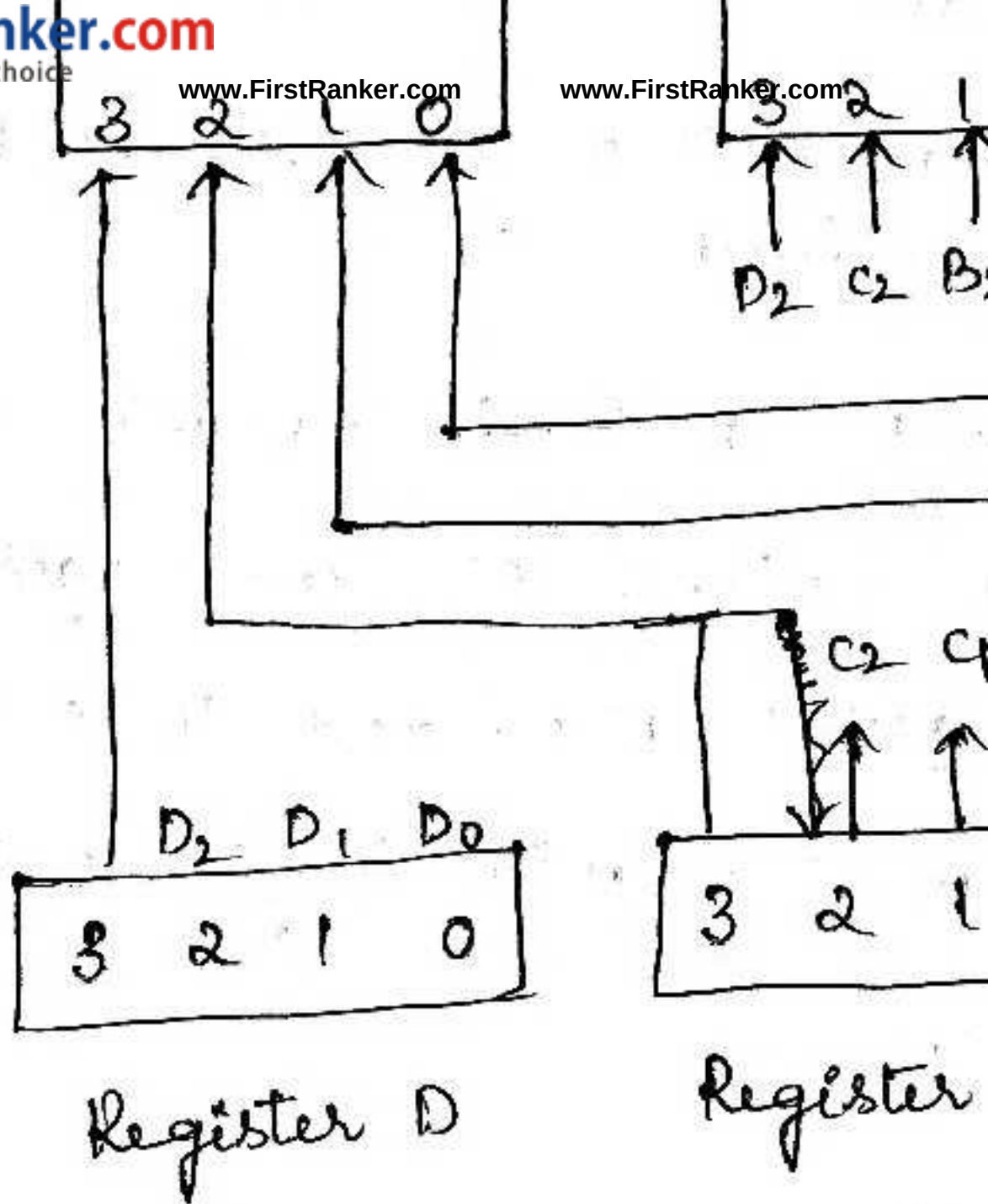


Fig : Bus sys

Each register has bou

Not too complicated the diag

We use labels to show the

Selected if $S_1 S_0 = 01$ and

The below table shows
for each of the four possi-
bles.

<u>S_1</u>	<u>S_0</u>
0	0
0	1
1	0
1	1

In general, a bus system
each to produce an n -line
multiplexers needed to cons

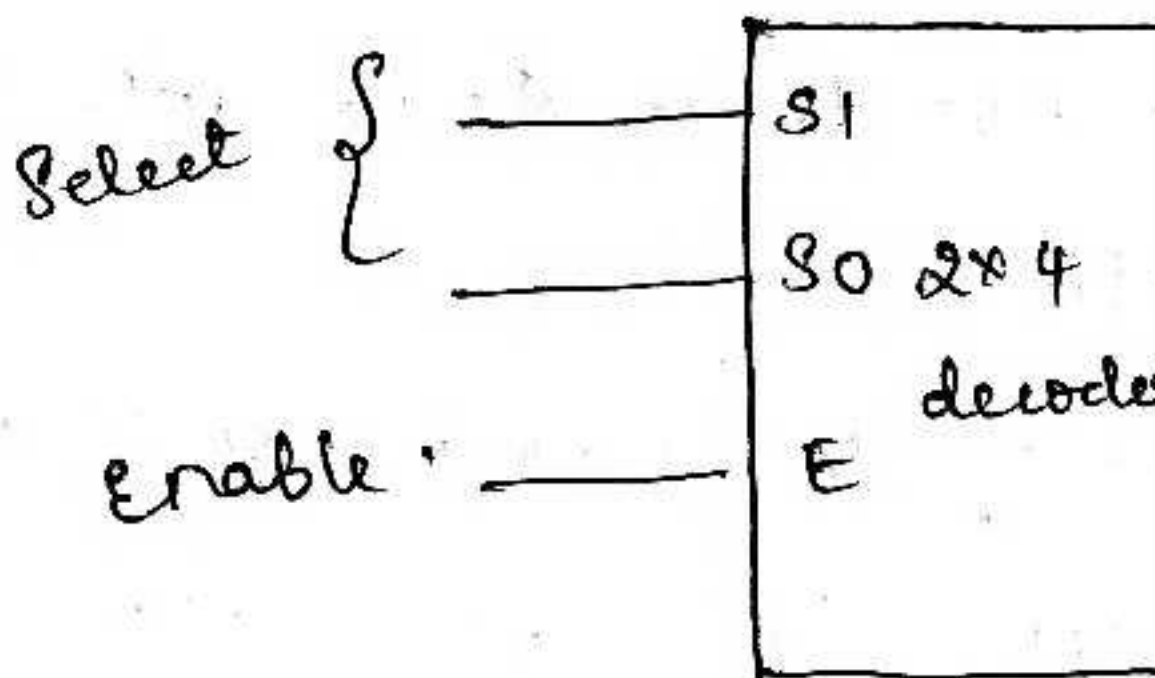
$R1 \leftarrow C$

Three - State Bus Buffers

A bus system can be implemented using 3-state buffers instead of multiplexers.

A three-state gate is a gate that can output three states. Two of the states are '1' and '0'. The third state is a high-impedance state, denoted by 'Z'. When a gate is in the high-impedance state, it does not drive the bus. This allows multiple gates to share a common bus. When a gate is in the high-impedance state, it is effectively disconnected from the bus. This allows multiple gates to perform any combination of AND, OR, and NOT operations. The one most common system is the bus buffer gate.

Other gates : The three standard
wires to form a common bus



the flip's to the decoder.
To construct a Co
bits each using three-8
with four bidders and

Memory Transfer :-

The transfer of
outside environment is

The transfer of n
is called a 'write' op

It is necessary to
writing memory transfer
with in the square brace

1) Register transfer microoperation

from one register to another

2) Arithmetic microoperation

numeric data stored in

3) Logic microoperations

on non-numeric data

4) Shift microoperations

in registers

The basic arithmetic

subtraction, increment,

The arithmetic micro

$$R3 \leftarrow R1 + R2$$

It states that the content

To implement the need the registers to hold that performs the arithmetic.

The digital circuit that takes two n -bit numbers and a previous carry bit as inputs and produces an n -bit sum and a carry out. The circuit that generates the sum of two n -bit numbers of any length is called a ripple-carry adder.

The binary adder cascade to each other.

The fig. shows of Full-adders (FA) to provide

and 'n' data bits for the 'P' and the sum can be transferred to the source registers by

Binary Adder - Subtractor

The addition & subtraction can be implemented into one common circuit with each full-adder.

be shown in below block diagram.

The Mode Input 'M' is used to select the operation. When $M=0$, the circuit is configured for addition. When $M=1$, the circuit becomes a subtractor.

When $M \geq 1$, we have
Complemented and a '1' is

The circuit performs

B. For unsigned numbers

2's complement of $(B-A)$ is

Result is $A-B$.

Binary Incrementer :-

The increment in
in a register.

Eg: $0110 + 1 = 0111$

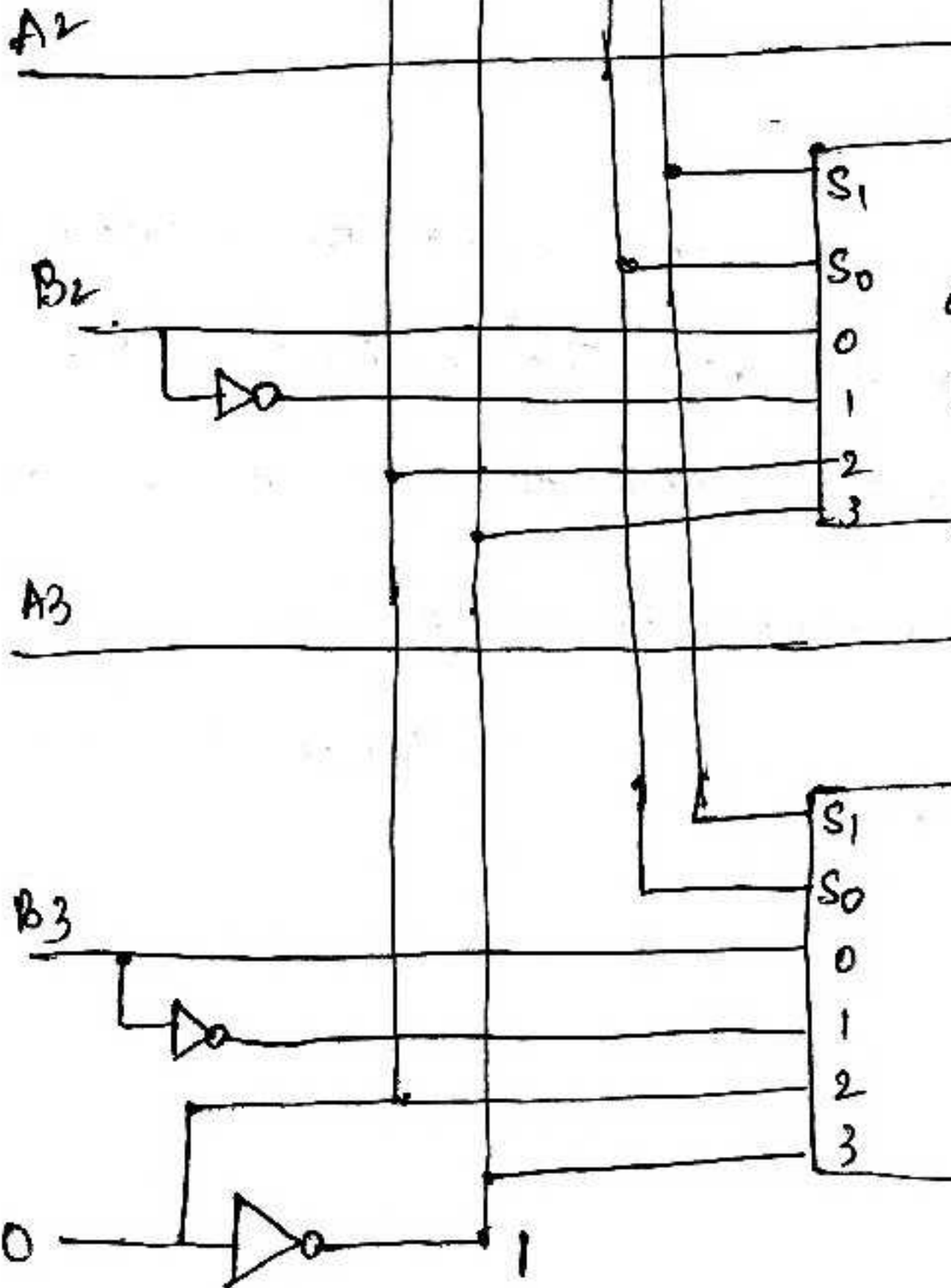
The diagram of a
is shown in fig. one of
adder (HA) is connected to

generates the incremented
carry C_4 will be 1 only
This also causes o/p's So

The circuit diagram
binary incrementer by
'n' half-adders.

Arithmetic circuit :-

The basic co
is the parallel adder.
adder, it is possible to
operations.



The 4-bit of the binary

following arithmetic sum

$$D = A + Y$$

Where 'A' is the 4-bit

Y is the 4-bit binary

adder. cin is the input

'0' or '1'. Symbol '+' is the

We can generate the

operations.

of the adder. If $C_{in} = 0$

o/p $D = A + B + 1$. Both cases with or without carry

when $S_1 S_0 = 01$, the ~~value~~ ^{complement}

adder. If $C_{in} = 1$, then

2's complement of B, which

$A - B$. When $C_{in} = 0$, then

when $S_1 S_0 = 10$, the i/p

are inserted into the 4 i/p's

gives $D = A$ when $C_{in} = 0$ and

a direct transfer from i/p

value of A is increment

List of logic micro operations

There are 16 different logic micro operations that can be performed with two binary variables.

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	f ₁₄	f ₁₅
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Each of the 16 columns represents a different logic micro operation that can be performed on the two input variables x and y.

$$F_8 = (x + y)'$$

$$F_9 = (x \oplus y)'$$

$$F_{10} = y'$$

$$F_{11} = x + y'$$

$$F_{12} = x'$$

$$F_{13} = x' + y$$

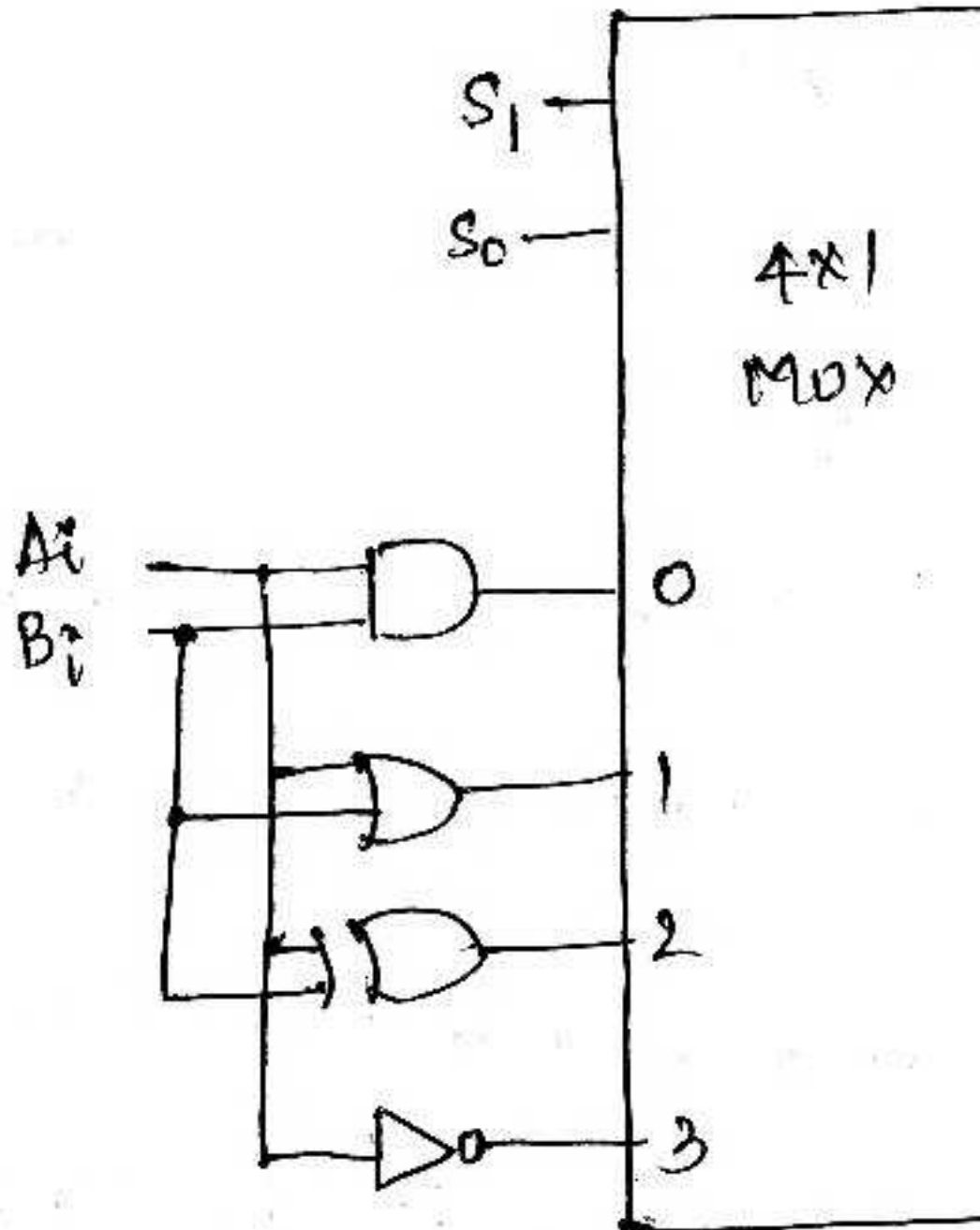
$$F_{14} = (xy)'$$

$$F_{15} = 1$$

The 16 logic
these functions by replacing

Complement.

The below fig shows
generates the four basic de



- ting individual bits or a register. They can be used to insert a group of bits, or insert some of the types

Selective-set :-

The selective-set register A where there are It does not affect bit position

Ex:

$$\begin{array}{r} 1010 \\ 1100 \\ \hline 1110 \end{array}$$

$$\begin{array}{r} 1010 \\ 1100 \\ \hline 0110 \end{array}$$

Again the two left most
corresponding bits of A
is just like an exclusive

Selective - clear :-

The selective - clear
A only where there are
for eg,

$$\begin{array}{r} 1010 \\ 1100 \\ \hline 0010 \end{array}$$

The two right most bits of B are left unchanged. The operation is more selective than the selective set operation. It uses 'AND' micro operation.

The 'insert' operation is used to insert bits into a register. This is done by performing OR with the register.

For eg, Suppose, A register contains 1010. To replace the four left most bits with 1111, first mask the four left

When A & B are equal,
both 0 or both 1

• In the case, the EX-
OR's result is then checked
are equal.

$$R1 \leftarrow \text{SHL } R1 \cdot C$$

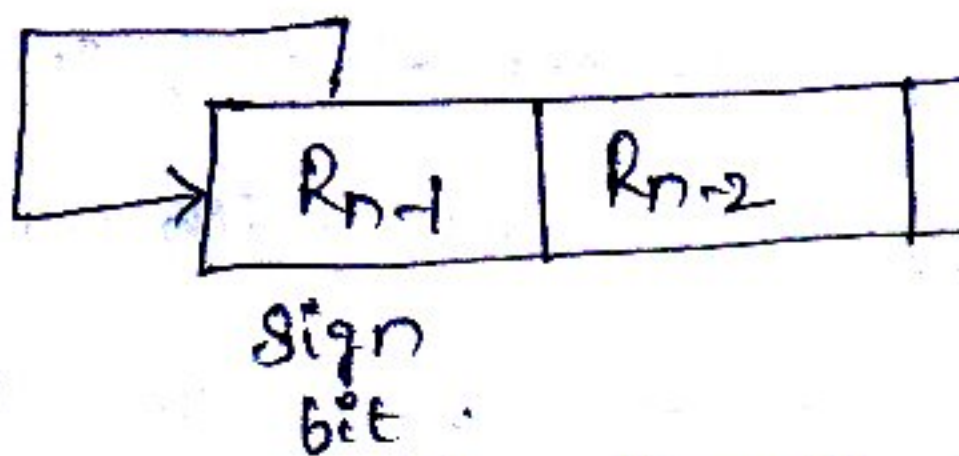
$$R2 \leftarrow \text{SHR } R2 \cdot C$$

are two micro operation
left of the content of register
right of the content of register
end position through the serial
a logical shift.

The "Circular shift"
bits of the register around
information. We use the
shift left & right respectively.

The symbolic notation
shown in below table

arithmetic sheet
because the sign of the number
multiplied or divided by
holds the sign bit, and
number. The sign bit is
-ve numbers. -ve number

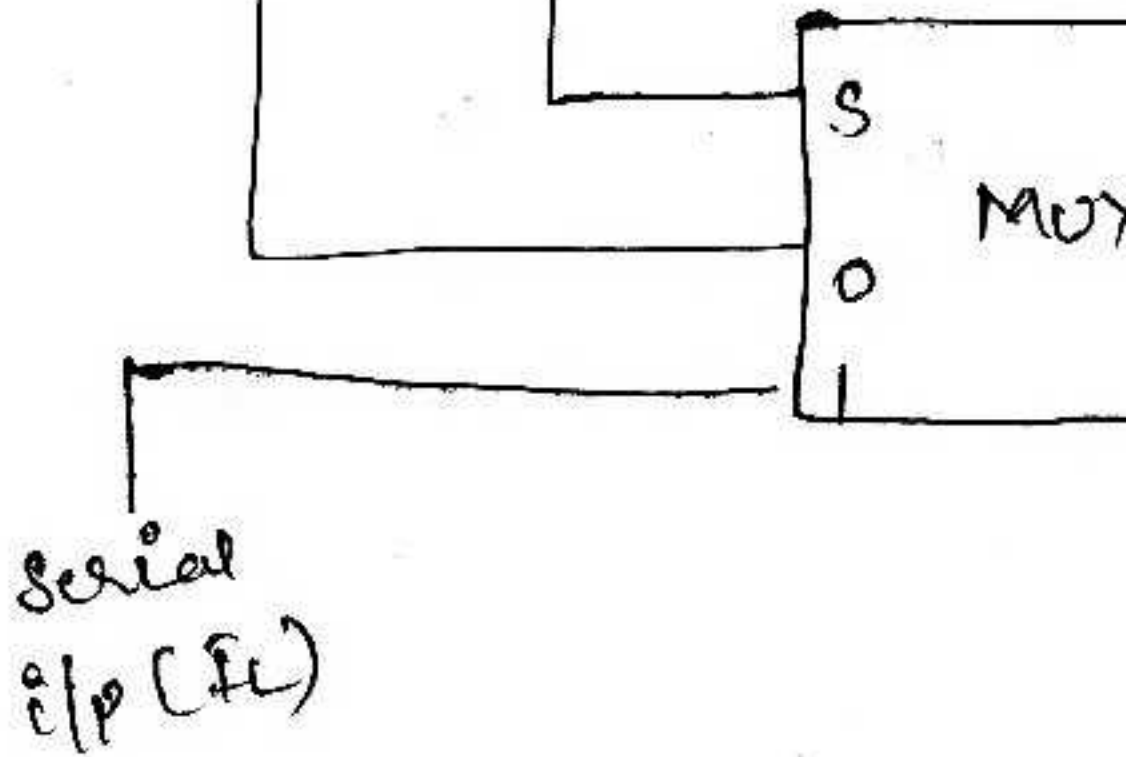


The above fig. shows
 R_{n-1} in the left most position
most significant bit and

Information can be parallel and then shifted of configuration, a clock data into the register and the shift.

A combinational circuit with multiplexers as shown.

The 4-bit shifter has four data i/p's to the i/p's, one for shift left (SL), one for shift right (SR), one for selection i/p



When $S=1$, the i/p d

A shifter with n data

The two serial i/p's can

to provide the three possible

Be

Ac

R A_{i-1}

L A_{i+1}

0	1	1	X
1	0	X	X
1	1	X	X

Instruction codes :-

A computer 'inst' specifies a sequence of m together with data are st each instruction from m register. The control then 'instruction and proceeds microoperations. The ability the stored program concep

Consider a computer with
being an ADD operation. This
with a bit configuration

When this operation is
the computer issues control
memory and add the oper

We must recognize
operation and a microoper
an instruction stored in
code that tells the computer
The control unit receives
interprets the operation code

Specifies one of 2 registers

Instruction code
designers who specify the a

Stored program organization

The simplest way
one processor register and
two parts. The first part
and the second specifies an
tells the control where to
operand is read from the
operated on together with

Register

and data in another.

A memory unit with
specify an address ($\therefore 2^{16}$
code in one 16-bit memory
code and 12 bits to specify
control reads a 16-bit
of memory. It uses the
to read a 16-bit operand
It then executes the operation

Computers that
usually assign to it the name
Ar. The operation is performed

instruction code not as an operand. When the second specifies an operand, the instruction is an immediate operand.

When the second specifies an operand, the instruction is an immediate operand.

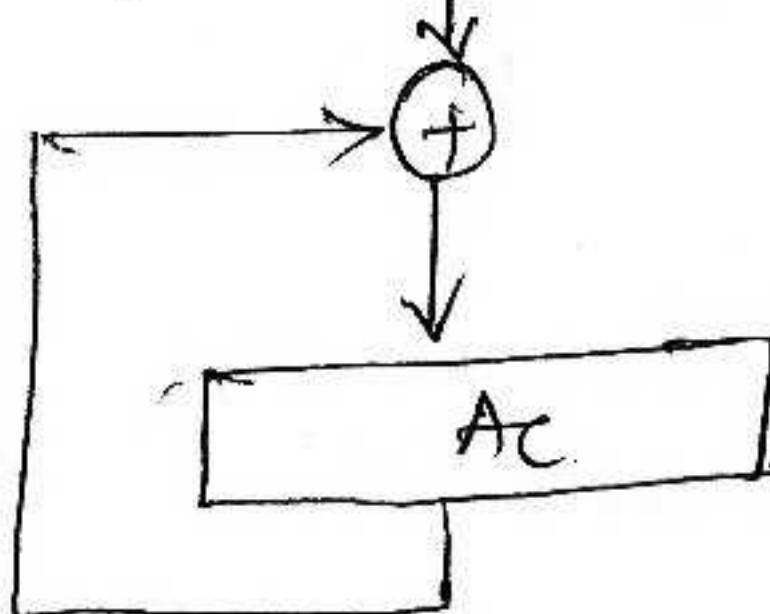
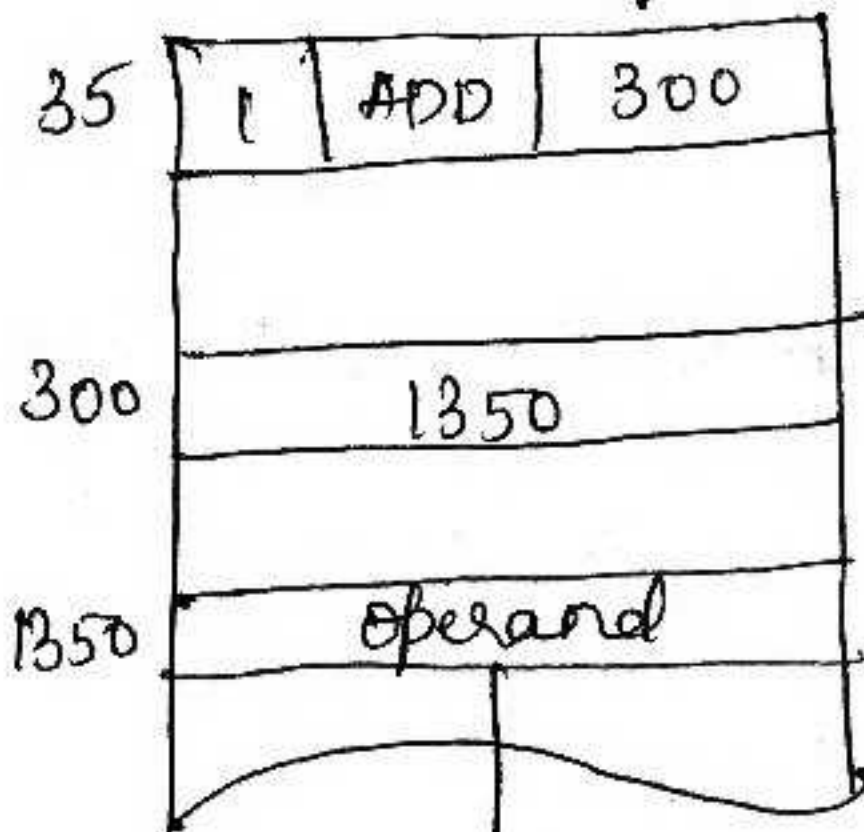
The bits in the second operand designate an address of memory. The bit of the operand is bound to the instruction. It is used to distinguish between the two.

Consider the instruction

fig

is shown in below fig.

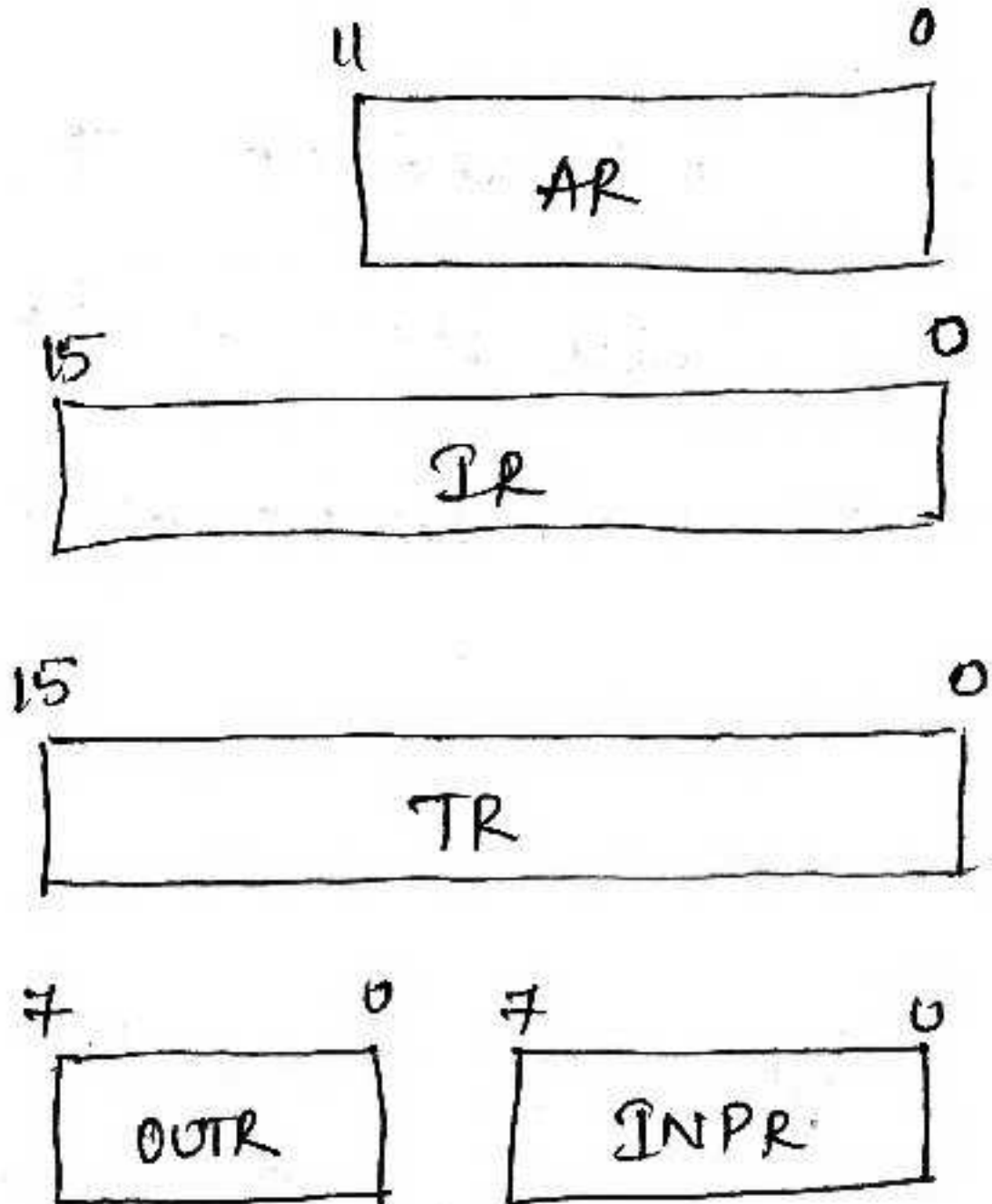
Memory



The
the a
of op
oper

nee
bete

the
is



The memory word
and each word contains
- ction word are needed

the address of the next instruction
after the current instruction.

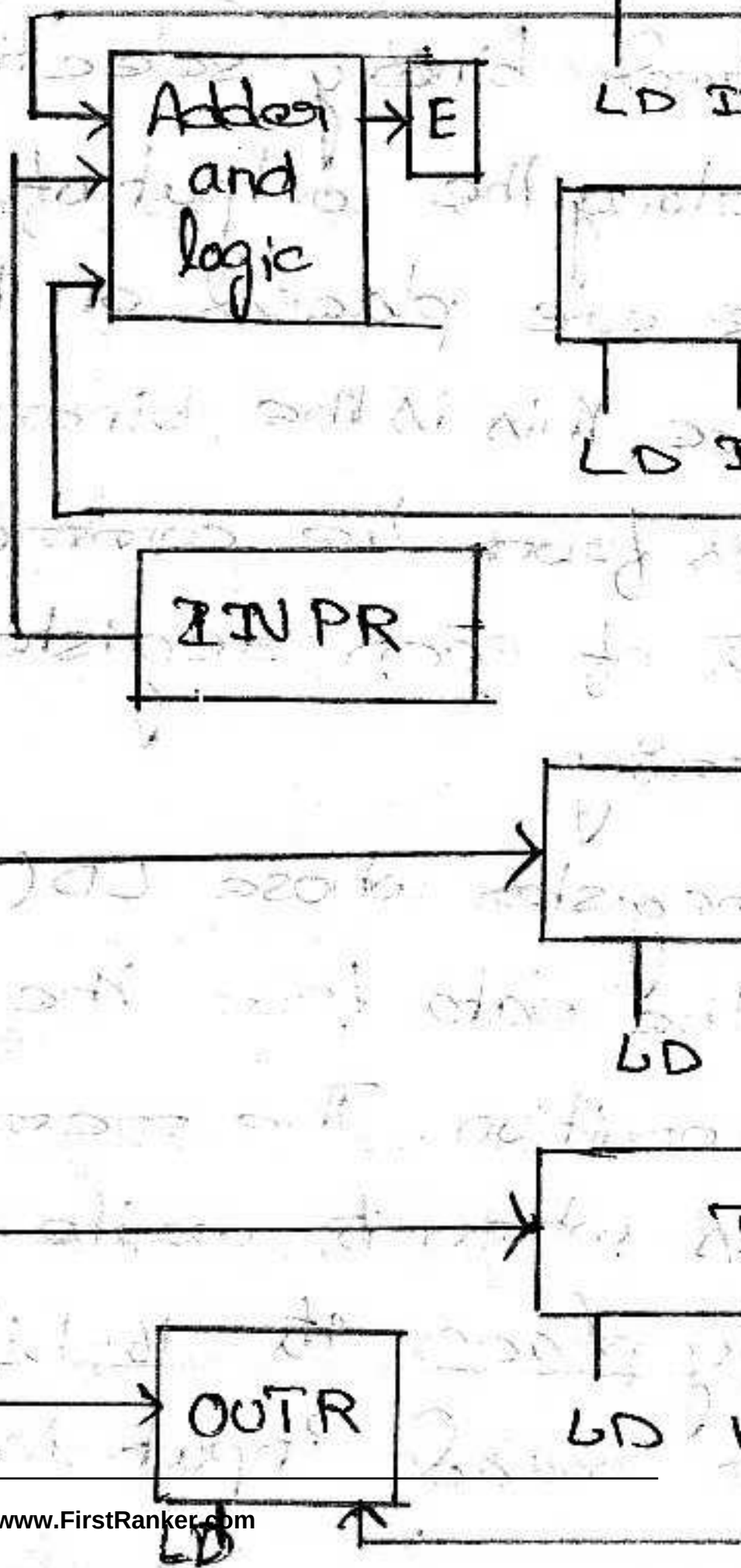
Two registers are
register (INPR) receives an

The o/p register (OUTR) holds
device.

Common Bus System :-

The basic computer
unit and a control unit
transfer information between
memory and registers.

A more efficient
information or system with many reg



The number stored in the register is the equivalent of the memory address. For example, the number 3 is stored in the register. The 16 bit outputs of the register are 011 when $S_2 S_1 S_0 = 011$ which is the binary representation of decimal 3. The register is connected to the input data inputs of the memory.

The particular memory location is enabled receiver. The next clock pulse the contents of the memory are activated. The memory outputs the data onto the bus when $S_2 S_1 S_0 = 111$ and $S_2 S_1 S_0 = 111$

There is no transfer

The 16-lines
information from six
The bus lines are connected to
registers and the control
control inputs: LD (load)
CLR (clear).

The input data
are connected to the
address is connected to
come from an adder
has three sets of input
come from the output
implement register
shift AC.

www.FirstRanker.com

reference to memory and if a 1 in the left most bit of are used to specify the type

The type of inst control from the 4 bits in instruction. If the 3 opcode not equal to 111, the inst and the bit in position 15 the 3 bit op code is equal in position 15. If the bit type.

If the bit is 1, the in position 15 of the inst not used as mode bit whe

STA

3xxx

Bxx

BUN

4xxx

Cxx

BSA

5xxx

Dxx

ISZ

6xxx

Exx

CLA

≠ 800

CLE

≠ 400

CMA

≠ 200

CME

≠ 100

CIR

≠ 080

CIL

≠ 040

INC

≠ 020

SPA

≠ 010

SNA

≠ 008

SZA

≠ 004

SZG

≠ 002

HLT

≠ 001

Instruction Cycle :-

A computer consists of instructions residing in memory. The processor goes through a cycle for each instruction, which is divided into sub-cycles.

- ① fetching.
- ② Decoding.
- ③ Read the effective address.
- ④ Execution.

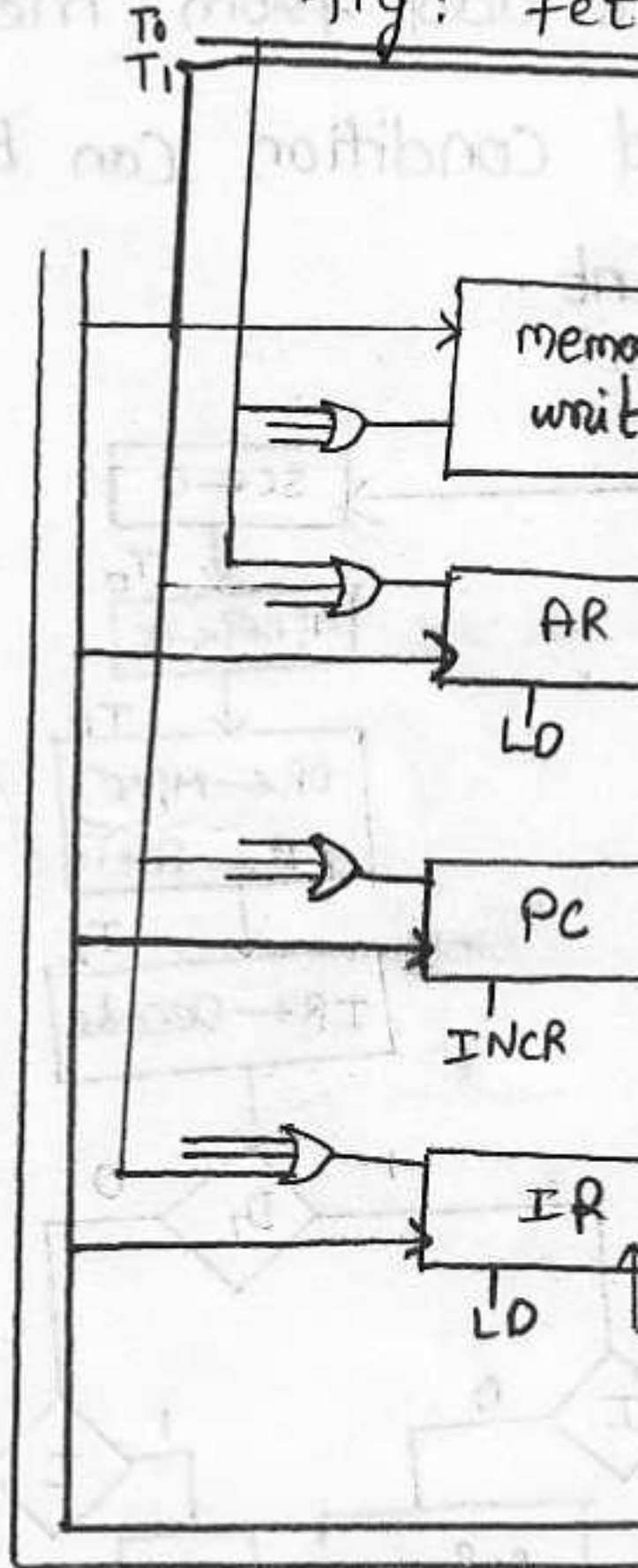
Then the control unit fetches the next inst'n - This process continues until the next inst'n is encountered.

The micro operations for
specified by the above tran

Since only
of memory it is necessary
during the clock transition
The instr'n read from mem
with the clock transition as
the same time Pc is ~~A~~ incr
add of the next instr'n in p

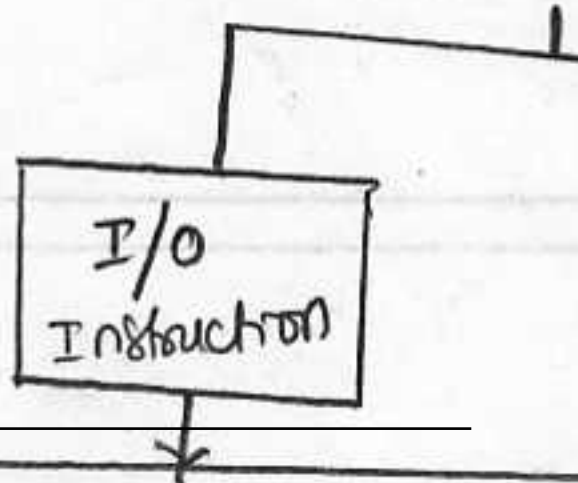
At time T_2
bit is transferred to ff I,
transferred to AR. Sc is
to produce the sequence

fig: fet



have a memory reference. It is necessary to read the effective operation for the indirect the register transfer stat

fig:



- Memory ref inst'n with necessary to do anything.
- $D_7' T_3 = 1$ so that the can be Continued with
- A seq reference / $Ip - 0$ clock associated by T_2
- After the inst'n is a fetch phase with $T_0 = 1$.
- ← Sc is either $\uparrow$ ed on Sc
- ← if Sc is incremented C'
 $Sc \leftarrow Sc + 1$ but it w
next timing signal in

SPA	$\pi B_4 :$	if	LA
SNA	$\pi B_3 :$	if	LA
SZA	$\pi B_2 :$	if	LA
SZG	$\pi B_1 :$	if	LA
HLT	$\pi B_0 :$	$S \leftarrow 0$	

Memory Reference Instr

The decoded o/p of the operation decoder that belongs to the table.

The effective address of the instruction was placed there during the timing signal T_3 when $I=1$. The instruction starts with timing signal T_0 .

The symbolic representation of the table in terms of the

on pairs of bits in AC of effective address. The result is stored in AC. The micro operations to

$D_0 T_4 : I$

$D_0 T_5 : A$

ADD to AC :

This inst'n adds the value specified by the effective address to the sum in AC. The sum is transformed into AC. The micro operations are:

$D_1 T_4 : D$

$D_1 T_5 : A$

BUN : Branch unconditionally

$D_4 T_4 : PC$

BSA : Branch & Save return

$M[AR] \leftarrow PC$

20	0	BSA	135
PC=21	Next Instruction		
AR			
135			
136	sub routine ↓		
	1	BUN	135

20
21
135
PC=136

Isz : Increment and skip if

$D_6 T_4 : D$

$D_6 T_5 : D$

$D_6 T_6 : M$

I/P - O/P
terminal

Serial
Communication



Fig1 :- Input- output

- * Initially the **ilp flag** **FG**
- * When a key is struck in **INPR** and the flag is set, the info by striking another key.
- * The computer checks the **FG** **INPR** is transferred in. Once the flag is cleared into **INPR** by striking

Program Interrupt

* The process of communication between the computer and the input-output device is called Programmed Control transfer.

* The computer keeps checking the input-output device and initiates an information transfer.

* The difference of information between the computer and the input-output device is called Programmed Control transfer.

* The computer is wasting time in checking the input-output device and some other useful processing.

- * Initially the flip flag FG
- * When a key is struck in is shifted into INPR and the flag is set, the info by striking another key.
- * The computer checks the flag INPR is transferred in once the flag is cleared into INPR by striking

The interrupt cycle is a branch and save return address available in pc is stored in a found later when the program it was interrupted. This location is a specific memory location at address 0 as the place for saving the address 1 into pc and no interruptions can occur until the interrupt has been serviced.

An example that shows
interrupt cycle is shown in

Suppose that an internal
control is executing the instruction

return address 256 is in PC.

ilp - o/p service program in

A RIN 119n instruction. It

In the main program is a b
part of 0. This instruction is plo
After this instruction is read
control goes to the indirect phase
address. The effective address
address that was stored there
The execution of the indirect
the return address from location

always available for head

the word that is inserted

→ The two operations of deletion of items.

→ The operation of insertion can be thought of as pushing on top.

→ The operation of deletion can be thought of as the result of popping the stack.

→ However, nothing is pushed.

→ These operations are decrementing the stack.

- The stack pointer register whose value is equal to currently on the top of
- The stack contains the
- Item C is on the top of
Sp is now 3.
- To remove the top item, the memory word at address
- Item B is now on top of
- To insert a new item, Sp and writing a word to stack. Note that item C is physically removed.

not full (if full = 0), a
operation. The push oper
following sequence of

$$SP \leftarrow SP + 1$$

$$M[SP] \leftarrow DR$$

if (sp = 0) then (full $\leftarrow$ 1)

$$EMPTY \leftarrow 0$$

→ The sp is incremented
of the next - higher

→ A memory write opera
into the top of the s

DR $\leftarrow$ M[SP]

SP $\leftarrow$ SP - 1

if (SP = 0) then (ENTRY $\leftarrow$

full $\leftarrow$ 0

→ The top item is read
SP is then decremented

→ If its value reaches
ENTRY is set to 1. This
item read was in location

→ Once this item is read on
the value 0, which is

→ Note that if a pop operation
& then SP is decremented, SP
to decimal 63.

- The Stack pointer sp
- The three registers are either one can provide
 - Pc is used during the
 - AR is used during the ex
 - Sp is used to push or
 - from the figure, the initial grows with decreasing add
 - Thus the first item stored is the second item is stored address that can be used
 - No provisions are available
 - We assume that the is with a data register.

Stack pointer is de
address of the next wor

→ A memory write operat
into the top of the stack

→ A new item is deleted

$DR \leftarrow H[$

$Sp \leftarrow Sp$

→ The top item is read

→ The sp is then incremented
item in the stack.

→ Most Computers do not
for stack overflow or

→ The stack limits can
registers: one to hold the
other to hold the

next empty location above

→ In this case, the sequence is interchanged.

→ A stack pointer is loaded

→ This initial value must be assigned to the stack in memory.

→ Henceforth, SP is automatically

updated in every push or pop operation.

→ The advantage of a stack is that it can refer to it without the need for an explicit address since the address is updated in the stack pointer.

Product to obtain
 $(A * B) CD * +$

→ where $(A * B)$ is a Product. The next operator two operands are C & D an expression with two

→ The next operator is $(A * B)$ are the two products, so obtain the result.

→ The Conversion from notation must take into hierarchy adopted for infix

→ This hierarchy dictates that
Inside the inner Parentheses

→ The converted expres

$$AB + DE +$$

→ Proceeding from left to
add D & E . At this po

$$(A+B) (D+E) C$$

where $(A+B)$ & $(D+E)$ are
from the sum.

→ The two operands to

→ These two numbers
added to F .

→ The final $*$ causes $+$
terms.

2.

An address field that
Processor register.

3.

A mode field that is
effective address is

→ The operation code field
bits that define various
subtract, complement &

→ The bits that define
code specify a variety
operands from the given

→ The various addressing
for digital computers
field of an instruction
of including multiple

a Computer depends on Registers.

- Most Computers fall into three categories:
1. Single accumulator
 2. General registers
 3. Stack organisation

→ All operations are performed on a single register. The instruction uses one address.

→ Ex :- The instruction for addition is defined as

ADD x

where x is the address of the operand.

Instruction. The Ex:-

→ This transfer-type instruction has two address fields to specify the

→ Each address field may contain a Memory word. An

ADD R_1, X

It has two address

→ Computers with stack organization have push & pop instructions which use a single address field i.e., PUSH X or POP X . The stack pointer is updated automatically.

Three - Address Instruction

→ Computers with three
each address field to store
or a memory operand.

→ The Program in assembly

$$X = (A+B) * C$$

ie, ADD R₁, A, B

ADD R₂, C, B

MUL X, R₁, R₂

R₁, R₂ are two Processor

MOV X, R₁

→ The MOV Instruction moves data from memory to register & register to memory & register to register. Source & destination are source & destination.

One - Address Instructions

→ One - address Instruction uses one register for all data. Here we will neglect the fact that the AC contains the data.

→ The Program to evaluate

Load A

ADD B

field to specify the
the stack.

→ The following Program

PUSH A

PUSH B

ADD

PUSH C

PUSH D

ADD

MUL

POP

X

TOS

top of stack

with all three Specific

→ The following is a P

load R_1, A

load R_2, B

load R_3, C

load R_4, D

Add R_1, R_1, R_2

Add R_3, R_3, R_2

MUL R_1, R_1, R_3

store x, R_1

$$R_1 \leftarrow M[A]$$

$$R_2 \leftarrow M[B]$$

$$R_3 \leftarrow M[C]$$

$$R_4 \leftarrow M[D]$$

$$R_1 \leftarrow R_1 + R_2$$

$$R_3 \leftarrow R_3 + R_2$$

$$R_1 \leftarrow R_1 * R_3$$

$$X \leftarrow R_1$$

To understand necessary to know
computes. [Instruction]

1. Fetch
2. Decode
3. Execute

Program Counter (PC) :

Instruction to be
each time an

Mode field :- It is

needed for the
may not be

Instruction format with

The operands
definition of the

Ex:- The instructions
→ Zero-address
computer are

(2) Immediate-Mode:-

→ The operand
→ An immediate
field rather

Ex:- Move
→ It is useful
constant

→ When this
operation the

(5) Auto increment of Address
→ When the address
refers to a table
necessary to increment
after every access

Effective Address:- It
address obtained
by the given
in a response

(6) Direct Address Mode
Effective address is
instruction

(5) Auto increment or

→ When the address refers to a table necessary to increment after every access

Effective Address :- It is the address obtained by the given instruction in a response

(6) Direct Address Mode
Effective address is given in instruction

(9) Indexed Addressing
the content
to the address
the Effective address

→ The Index
that contains

$$EA = \text{Index} \uparrow$$

Index

Relative
Numerical Example :-

$$PC = 200$$

$$RI = 400$$

$$XR = 100$$

$$AC$$

* In Direct mode the effective address is 500 and the operand is 3001.

* In Indirect mode the effective address is 800 and the operand is 3001 loaded into Ac.

* In Relative mode the effective address is 702 and the operand is 325. During the fetch stage and during the execute stage.

* In Index mode of addressing the effective address is 801 and the operand is 900 stored in Ac.

* The Auto increment & auto decrement modes. In the auto increment mode here R₁ is 401 & 399 lie

CISC [Complex Instructions]

→ It consists of

Major characteristics

1. A large number of instructions
2. Some instructions are simple and ~~more~~ are complex
3. A large variety of instructions
4. Variable-length instructions
5. Instructions are executed sequentially

another (computer) without

most common transfers

→ memory & process

→ b/w process Registers

→ Input & Output

Instruction in Data transfer

Name

Mnemonic

Load

LD

Store

ST

Move

MOV

Exchange

XCHG

Input
Output

IN

OUT

Push

PUSH

Pop

POP

another (computer) without

most common transfers

→ memory & process

→ b/w process Registers

→ Input & output

Instruction in Data transfer

Name

Mnemonic

Load

LD

Store

ST

Move

MOV

Exchange

XCHG

Input
output

IN
OUT

Push

PUSH

Pop

POP

Data manipulation instructions

These perform operations
computational capabilities
divided into 3 basic types

1. Arithmetic instructions
2. shift "
3. logical & bit manipulation

Arithmetic Instructions

The basic operation
addition, subtraction, multiplication
these Increment, Decrement
with carry, Negate.

clear

compliment

AND

OR

Exclusive-OR

clear carry

set carry

compliment carry

Enable interrupt

Disable interrupt

- The clear operation
by 0's

- The compliment instruct
inverting all the bits

One. The same is done

Therefore ORing is done with another operand which will be set to 1.

- The XOR is used to be an operand. This is be

$$x \oplus 1 = x' \quad \text{and} \quad x \oplus 0 = x$$

when x is XOR with

1 " " " "

Rotate left
Rotate right through
Rotate left through

The logical shift inserts

The end is the

leftmost bit for

rightmost bit for

The rotate operations process
at one end of the word
but are circulated back

2. Jump

3. Skip

4. Call

5. Return

6. Compare (by subtraction)

7. Test (by ANDing)

1. Branch:- It is a one instruction in assembly language with a symbolic name for address. This may be conditional. A branch instruction causes the program counter to be loaded with an address without any further instruction specifies a conditional branch if the program counter is

next instruction in sequence
an unconditional branch
pair of instructions can
is not met; while a
causes a branch if the

Call :- The call instruction
with subroutines. A call
of an operation code to
specifies the beginning

Return :- The return instruction
in conjunction with subroutines
instruction available in
a temporary location.

maintaining the result or
status bits of interest are
a Zero condition and a
generation of these status
conditional branch instructions

is a digital computer

The control function is a binary variable. When a control variable is in a state corresponding to a micro operation of a control variable, it specifies the state of the control variable. In a bus-organized system, the control variable specifies the micro operation paths in multiplexers, decoders, and other control logic.

The control unit is responsible for the sequence of micro operations. During the execution of a program, the control unit initiates the micro operations to be initiated; while the control unit maintains the control variables at any given time.

Control units that employ a writable control can be used for writing memory that is part of a ^uControl memory.

A computer that employs a control unit have two separate memory.

- 1) Main Memory.
- 2) Control Memory.

The Main Memory is where programs are stored.

The Control Memory holds the control instructions.

Control Memory can't be altered by the user.

Instruction read from the

The micro instruction
one or more micro operation
operations are executed,

next address. The next address
external input conditions
executed, the next address
generator circuit and the

address register to read

The next address generator
microprogram "sequencer"

sequencer ~~and~~ are incremented
by one.

address register.

The other two Control memory are connected to a clock.

The main advantage is the fact that once established, there should be no changes.

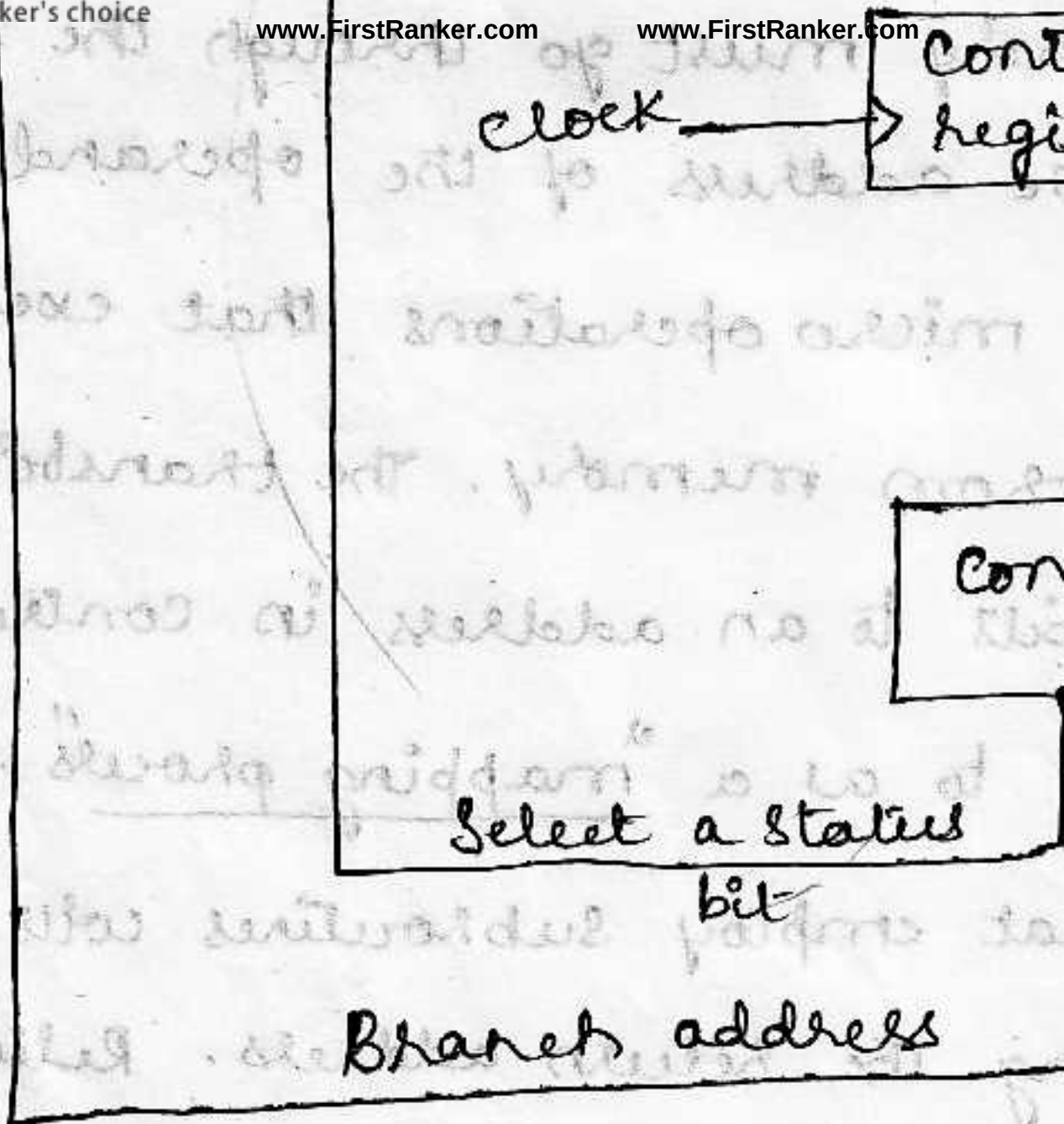
RISC architecture has less than a control memory with

The Control unit that determines the execution of the next step is to generate the instruction, fetched from the instruction code where the routine is ^{located}.

Micro programs are stored in an external register for storing addresses can't be stored.

When the execution of the micro program is over, the control must return to the start.

The address sequences in the control memory are:



The fig. shows a bl
and the associated η/w need
instruction address.

The diagram
which the CAP receives th

Conditions Information
actions initiated based
1 or 0. The Conditional
branch logic.

The branch logic
Variety of ways. The simple
condition & branch to the
address reg. is incremented
a multiplexer.

Eg: There are eight sta
Three bits in the micro
selection variables for the
Status bit is in the '1'

- routine call and restore

- routine return. The sub.

source for transferring

main ~~memory~~ routine. The

file that stores addresses

reg. in a last-in, first

Mapping Instruction :-

A special type of branch instruction is a branch to the
a micro program routine for a

for eg, a computer with

shown in fig.

it can use addresses lower than four micro-instructions would be available.

One can extend the mapping rule by using function. The bits of the mapping ROM. The control bits for the control address.

The mapping control instructions for control mapping.

The mapping control integrated circuit called Boolean expressions.

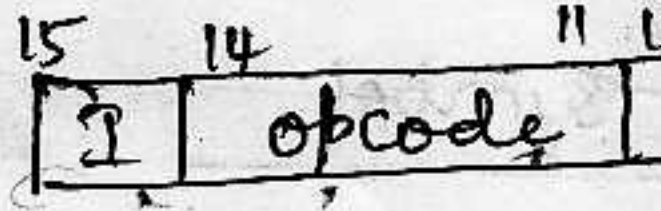
Control memory
128 x 20

Control unit

It consists of ^{two} memory units for instructions and data, and the micro program. Four registers in the processor unit & two write

The Computer

beg. It consists of three



1) 'I' bit is for indirect
for address.

The below lists four
reference instructions.

<u>Symbol</u>	<u>op</u>
ADD	00
BRANCH	0
STORE	0
EXCHANGE	00

Three bits

The each Three bits

PC microoperation

000	None
001	$Ac \leftarrow Ac + DR$
010	$Ac \leftarrow 0$
011	$Ac \leftarrow Ac + 1$
100	$Ac \leftarrow DR$
101	$AR \leftarrow DR (0-10)$
110	$AR \leftarrow PC$
111	$M[AR] \leftarrow DR$

010

 $AC \leftarrow \overline{AC}$

100

 $AC \leftarrow shr A$

101

 $PC \leftarrow PC + 1$

110

 $PC \leftarrow AR$

111

Reserved.

CD

Condition

00

Always $z = 1$

01

DR(15)

10

Ac (15)

11

 $Ac = 0$

May

142

8-43

29-142

29-142

Constitution

Symbol

Unconstitutional (primary)

U

Unlawful arrest

F

Supp. det. of the

S

Value in the

T

status bits when we write
form.

The BR (Branch) bit
in conjunction with the address
of the next instruction

and defining their own sy

Each symbolic micro

fields : label, micro opera

- 1) The label field may be symbolic address. label
- 2) The micro operations field symbols, separated by than one symbol then used when there is no translated by the assem
- 3) The CD field has one
- 4) The BR field contains
- 5) The AD field specifies

occupied by the routines b
last 64 words may be u

The microinstru
are:

$$AR \leftarrow PC$$

$$DR \leftarrow M[AR], PC$$

$$AR \leftarrow DR(0-10), C$$

The address part is t
transferred to one of 16 re
code part of the instruet

The fetch routine nee
are placed in control memoe

help is useful for writing
language format. The binary
memory.

Symbolic microprogram :-

The execution of the M
routine results in a branch
the four bits of the operation

The first address for the
0000100 (decimal 4) and 0

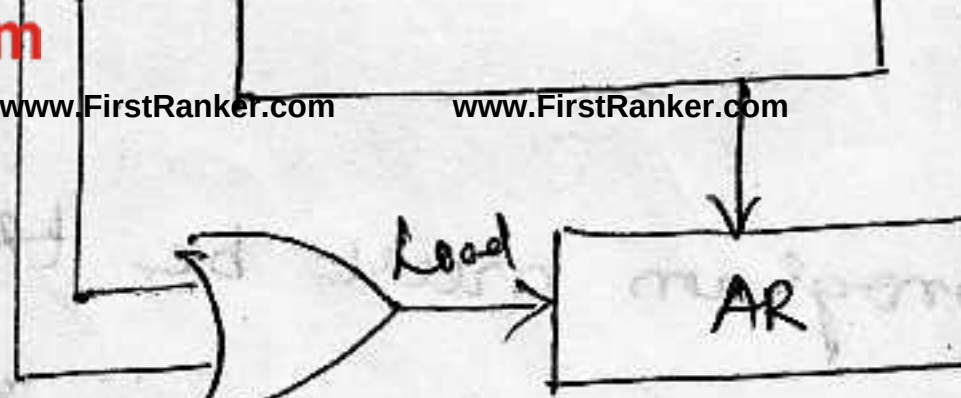
The first address for the other

12, 16, 20, - - - 60.

The indirect address mode
reference instructions. The su

The symbolic micro
binary either by means of
if the microprogram is si

The equivalent binary
listed in table.



decoders are connected to the unit.

Fig. shows the three that must be made from of the microinstruction present control memory are decoded eight outputs. Each of these the proper circuit to initial as specified.

For example, when $F1 =$ transition transfers the content when $F1 = 110$ (binary 6) the As shown in fig, outputs connected to the load input

Micro program sequencer :-

The basic components of the control memory and address. The address selection sequencer.

A microprogram sequencer functions to suit a particular. There are large ROM and circuit packages, so are general for the construction of microprogram. a wide range of acceptable must provide an internal

fig:- Microprogram sequence

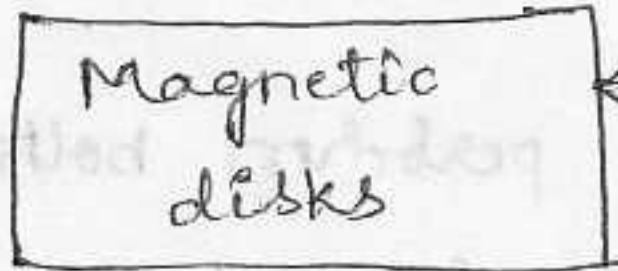
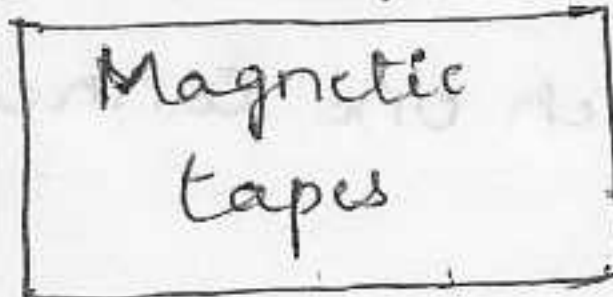
The control memory is included in the interaction between the sequencer and the control memory. There are two multiplexers. One multiplexer selects an address from the sequencer and routes it into a control address register. The other multiplexer tests the value in the control address register. The result of the test is used to determine the next microprogram to be executed.

from the BR (branch) field
The input logic in a particular
operations that are available

Typical sequences of
call and return from subroutines
pop the stack, and other address
inputs, the sequencer can perform
operation. The input logic
I₀, I₁ and T, and three outputs

variables S₀ and S₁
for CAR. Variable L enables
values of the two selection
multiplexer. For example S₁, S₀
selected and establishes
CAR.

Auxiliary
memory



The Memory hierarchy
employed in a computer system
auxiliary memory to a relational

The Semiconductor array including a number of rows & columns, a no. of cells in a row and a no. memory cells of a column.

The most common access memory (RAM). RAM write operations. The character must be provided with a C interrupted, then the data only as temporary storage

RAM technology

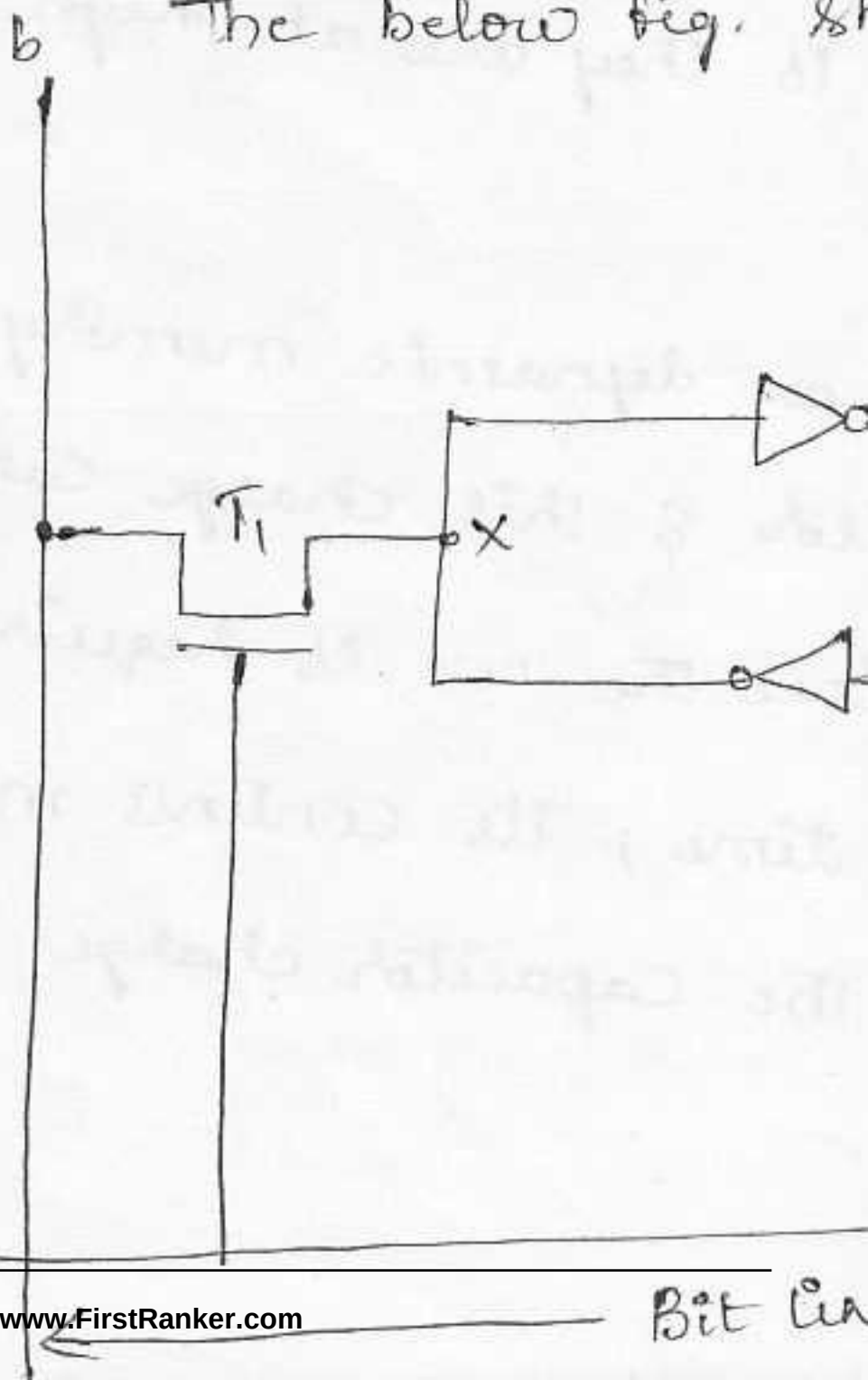
(i) Static RAM

(ii) Dynamic RAM.

RAM is more dense and less
are generally faster than

There is another type of memory
known as Read-only memory

The below fig. shows



Wordline.

A major advantage is consumption because current is being accessed and also

The disadvantages

Dynamic RAM's:

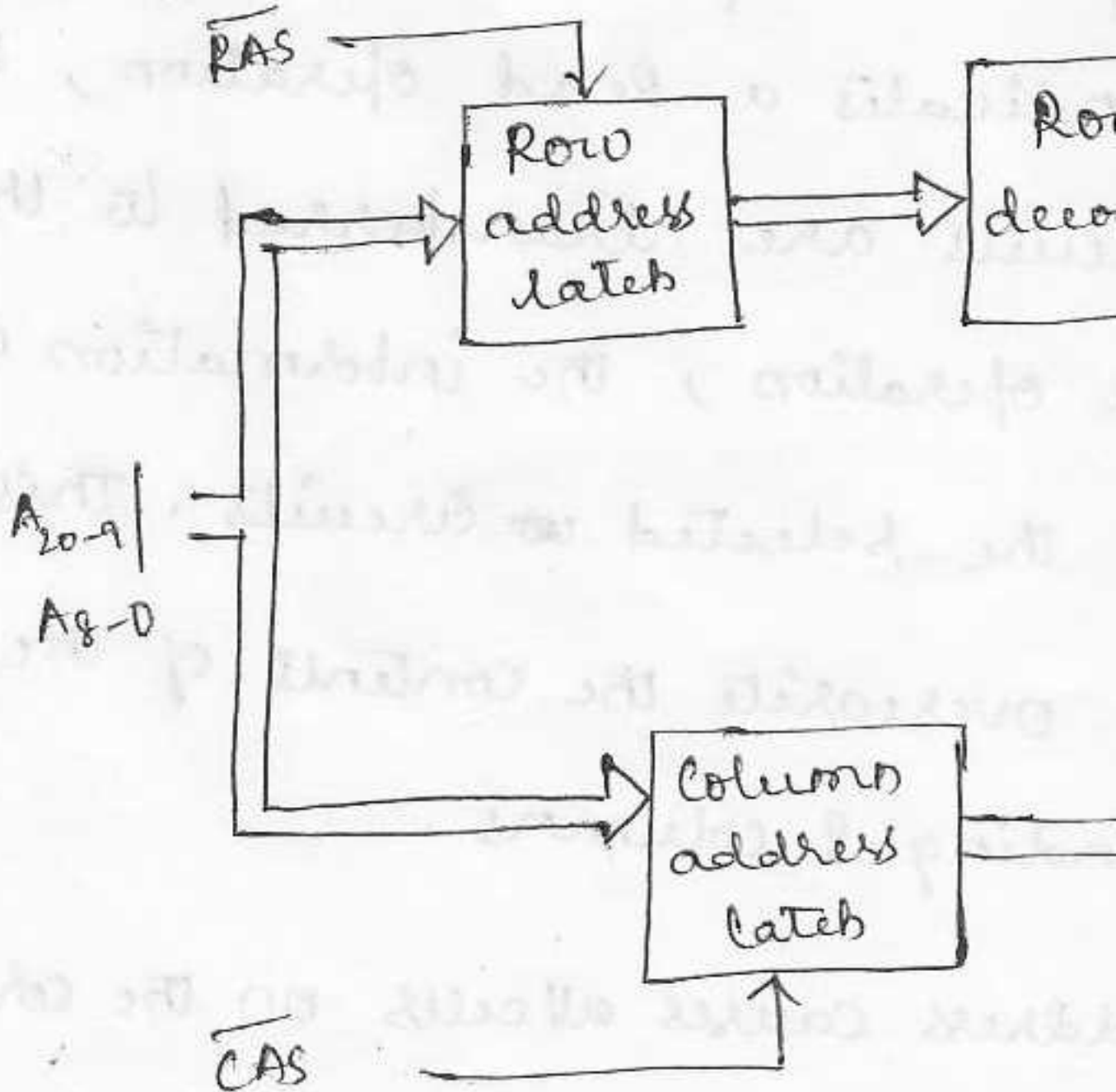
Information is stored in the form of a charge on a capacitor for only tens of milliseconds. Information for a much longer time is constantly refreshed by restoring its value.

discharge

Asynchronous DRAM's :-

The below fig. shows

as $2M \times 8$, as



applied to the address pins

latch under the control of

The information
and the appropriate group

to the $R/\bar{W}$ control signal

O/p values of the selected

data lines D_{7-0} . For a w

D_{7-0} lines is transferred

information is then used

selected cells in the corre

Applying a row

-ding how to be read an

3) A DRAM chip is organized of bits in parallel.

4) It provides flexibility

Fast page Mode :-

The contents of a row are sensed, but only 8 bytes

D_7-0 . This byte is selected

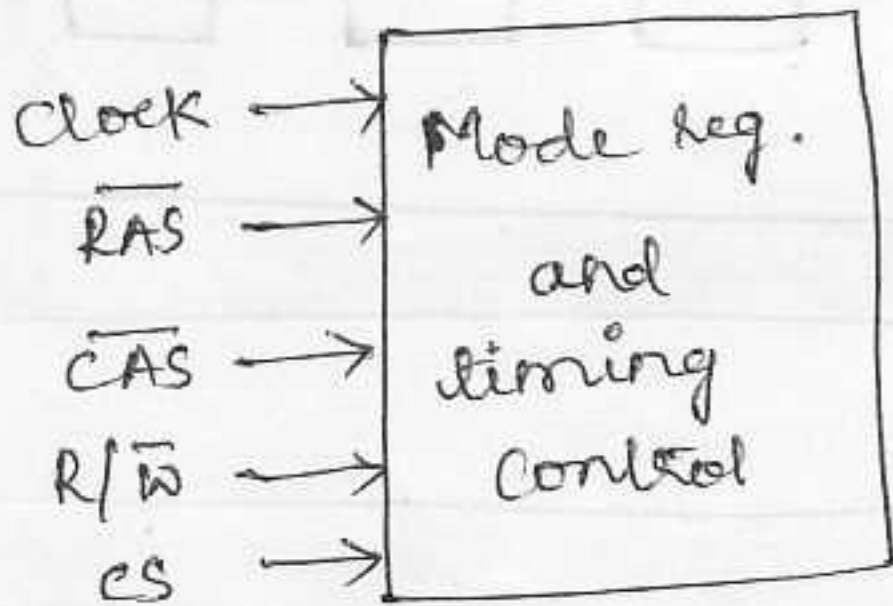
A_8-0 .

A simple modification can access the other bytes in the row. A latch selects the row. A latch

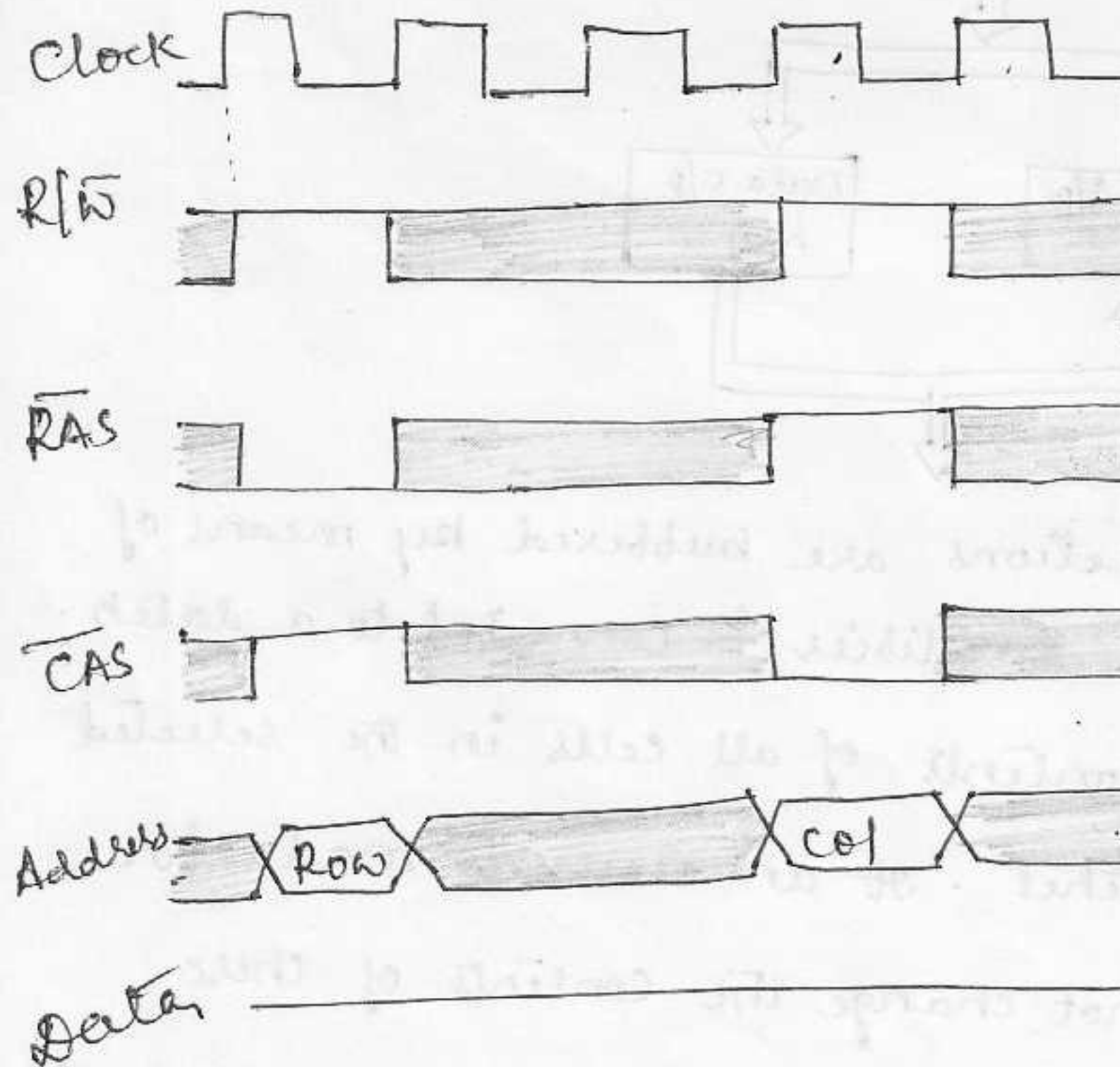
resolving the conflict between

and placed in the state for the
of the column which will

in can make it possible to
some row with one having to
to be added at the top of the page



The address & data control registers. The O/p of each set of registers. A read operation causes the data to be loaded into these registers for refreshing purposes but it will not refresh the data in the latches that correspond to the data being transferred into the data output.



First, the row address is
 $\overline{RAS}$ signal. The memory
 to activate the selected row

Memory cells are in an array, each cell is capable

The fig. is an array consisting of 16 words of 8

Each row of cells. Cells of a row are connected as the word line, which is a chip. The cells in each column are connected by two bit lines.

The sense/write enable lines of the chip. During

Sense, the read the information

Data i/o lines : b_7

Word line and transmit t
During a write operation,
information and store the cell

The data i/p &
circuit are connected to a
can be connected to the data

Two control lines
to the address & data lines.
operation & the CS (chip select)
multi chip memory system.

This is an type of ROM
erased and new data to
-programmable ROM is usually

EPROM's are capable of
long time, they can be use

The EPROM & ROM if
the connection to ground is
special transistor is used,

as either as a normal tra
that is always turned off
to behave as a permanently

The advantage
and reprogrammed. The di
from the circuit for reprog

Cache memories :-

The speed of the memory is a very important factor for good performance, the processor needs to access instructions and data quickly.

An efficient solution is to use cache memory, which essentially makes the memory access to be faster.

The effectiveness of a cache is a property of computer programs. Most of programs show that most of the instructions in a program are executed in a routine in which many instructions are repeated.

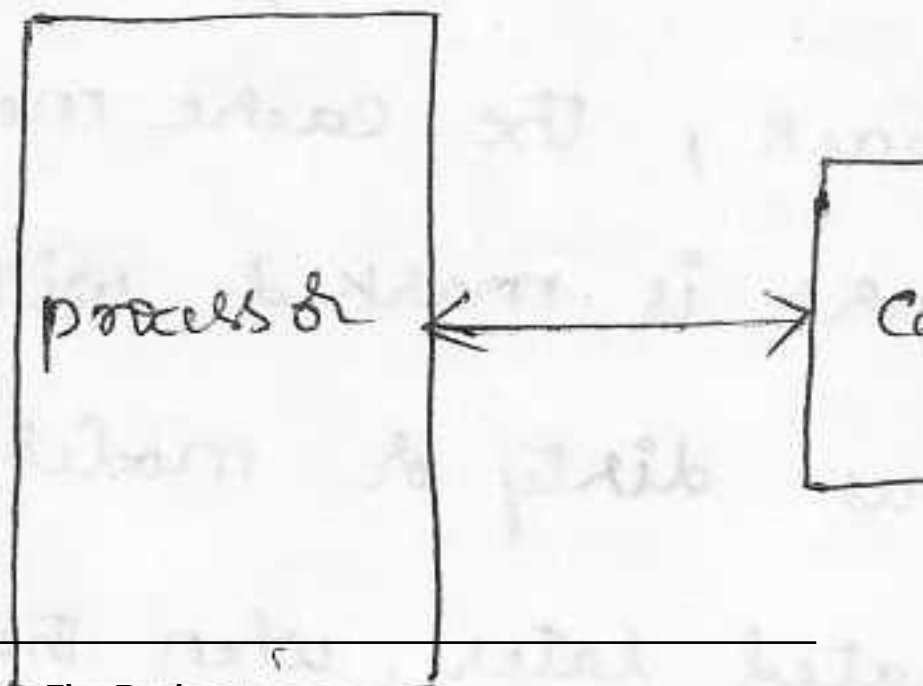
The instructions in

item is first needed, this where it is needed.

The spatial aspect one item from the main memory batch several items that are

The term 'block' address locations of some size block is 'cache line'.

The below fig.



The cache Controller
requested word currently ex
operation is performed on the
Read operation, the main m

The write operation ma

1) write-through protocol 2)

In write-through protocol
memory locations are updat

write-back or
In ~~dirty~~ or ~~modified~~ copy

updated and that updated

flag bit. That bit is know

main memory location is

In the case of write-through, the data is written to the cache and then the desired information is written to the new information.

Mapping Functions :-

Direct Mapping :-

The simplest way to store memory blocks is the direct mapping technique.

In this technique, block address is taken modulo 128 of the cache, to find the cache block.

One of the main memory blocks is mapped to a cache block. In a direct mapping cache, it is

are stored in cache block 1.



Main Memory address.

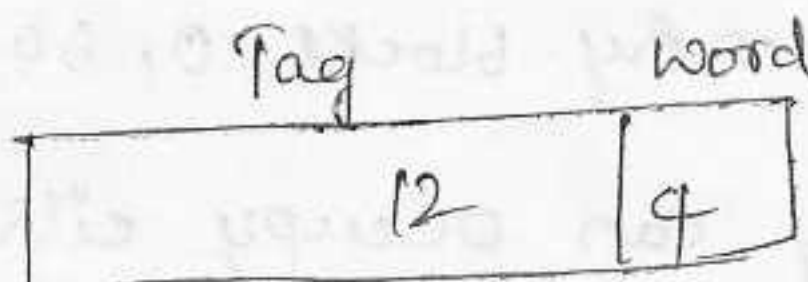
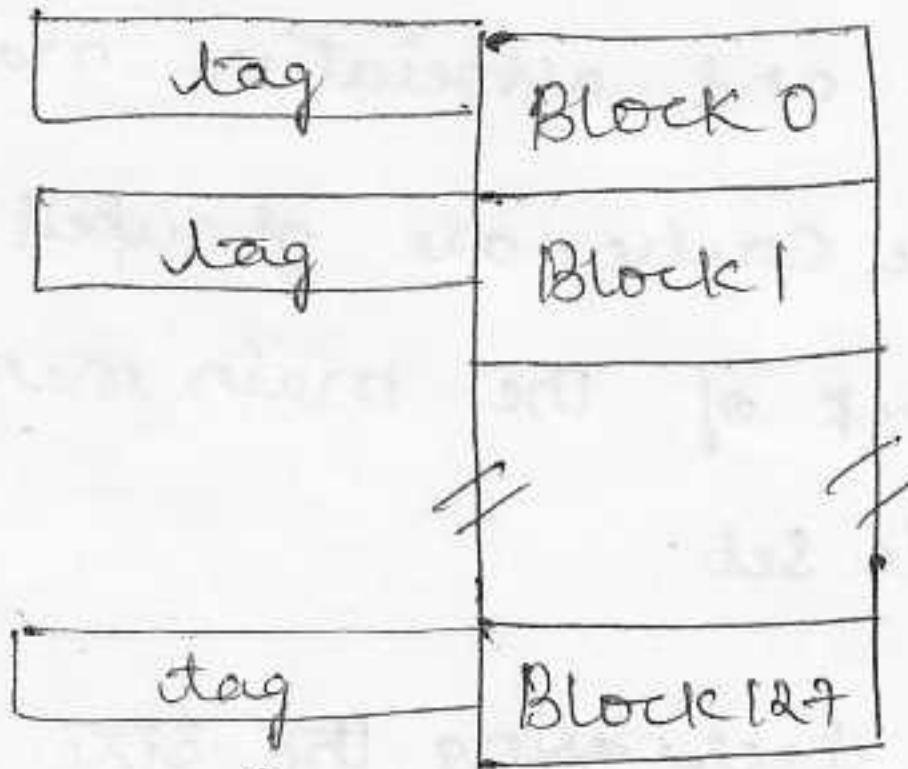
Since more than one m
Cache block position, Contention
When the Cache is not full.

for eg, instructions program
in block 129, after a branch.
of these blocks are transferred
Contention is resolved by allocating
currently resident block.

placement of a block in
memory address. The memory
fields as shown in fig.

not very flexible

Associative mapping :-



the cache

A search of this
search.

Set - Associative mapping :-

A combination of
techniques can be used. B
sets and the mapping all
to reside in any block of

The h/w cost is h
associative search.

An example is sh
two blocks per set. In t

4032 map into cache set
two block positions with i

Having 64 sets means
address determines which
desired block. The tag is
associatively compared to the
to check if the desired block

for the main
blocks per set can be access
eight blocks per set can be
and so on. The extreme corner
set bits & corresponds to the
tag bits. The other extreme
mapping method. A cache is
as a k -way set-associative

A difficulty arises
and the cache uses the wrong data.

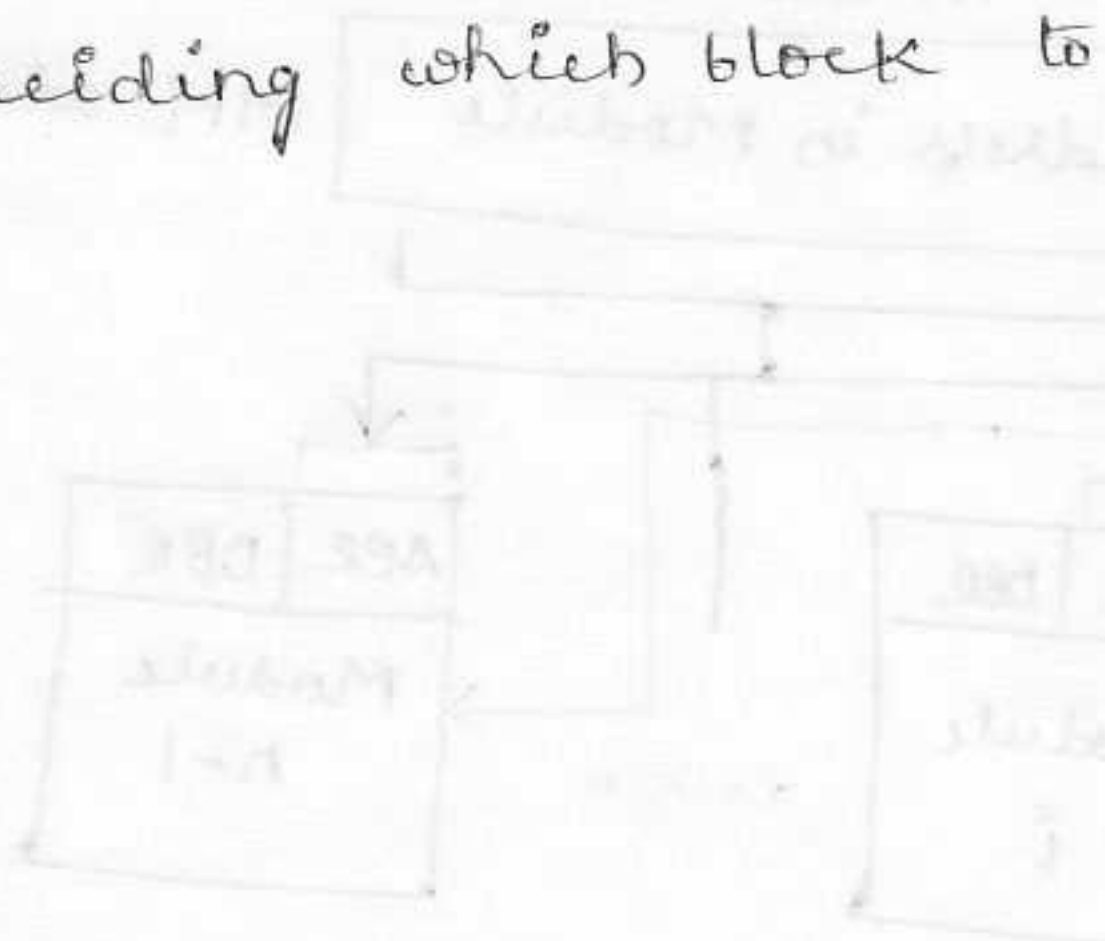
One solution to this problem is
forcing the dirty data to be written to
the DMA transfer takes place.

The need to ensure consistency between
(processor & DMA) uses the
cache-coherence problem.

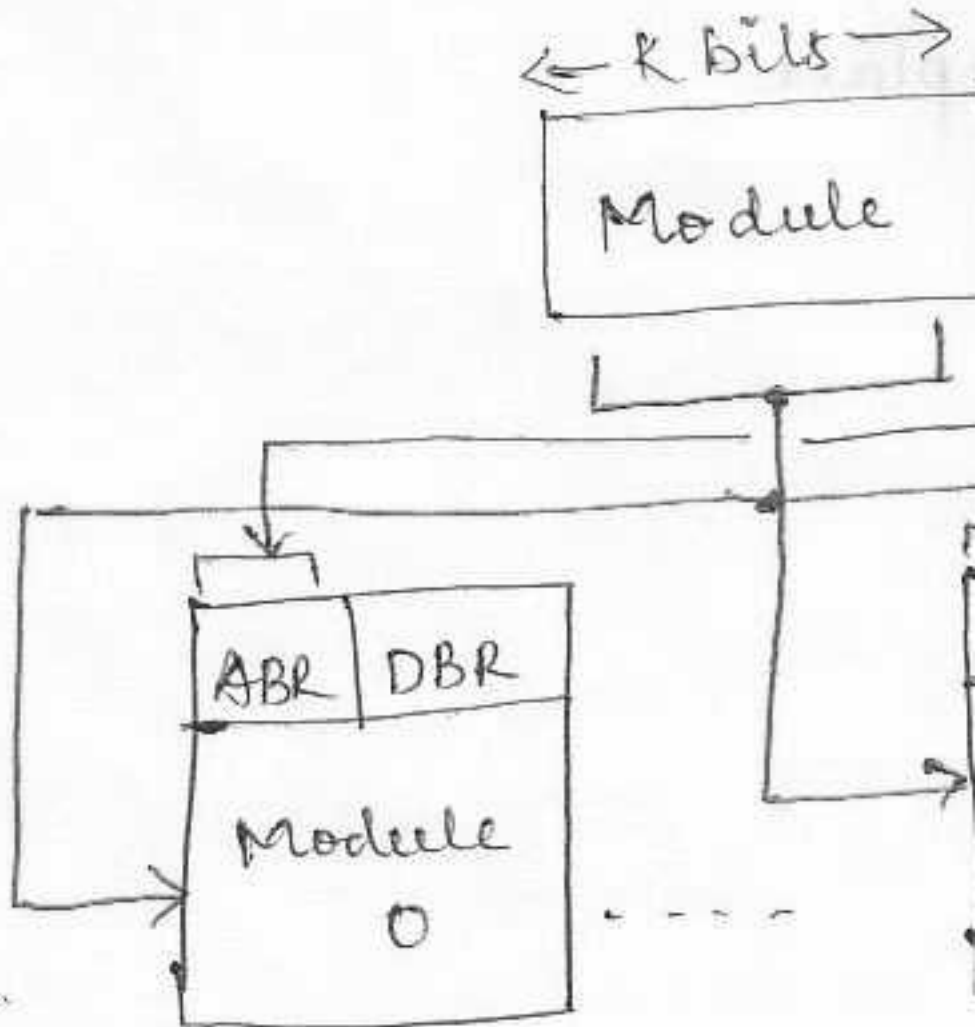
Replacement algorithms :-

When a block is to be replaced,
overwrite the one that has
least been referenced. This block is
the least recently used
block and the technique is called LRU.

Accesses are made to sequ
too large to fit into the
can be improved by introdu
in deciding which block to



Two methods of ad



a) Consecutive

same time, however, device information in other memory

The fig (b) is a module is called Memory interleaving address select a module, location with is that module located in successive modules

Any component for access to consecutive modules busy at any one

This results in & higher avg. utilization of To implement the interleaving otherwise, there will be g

Virtual memories :-

Techniques to

data blocks into the physical memory. The amount of physical memory required for execution is

The binary addresses of instructions or data are called virtual addresses. These addresses are translated into physical addresses.

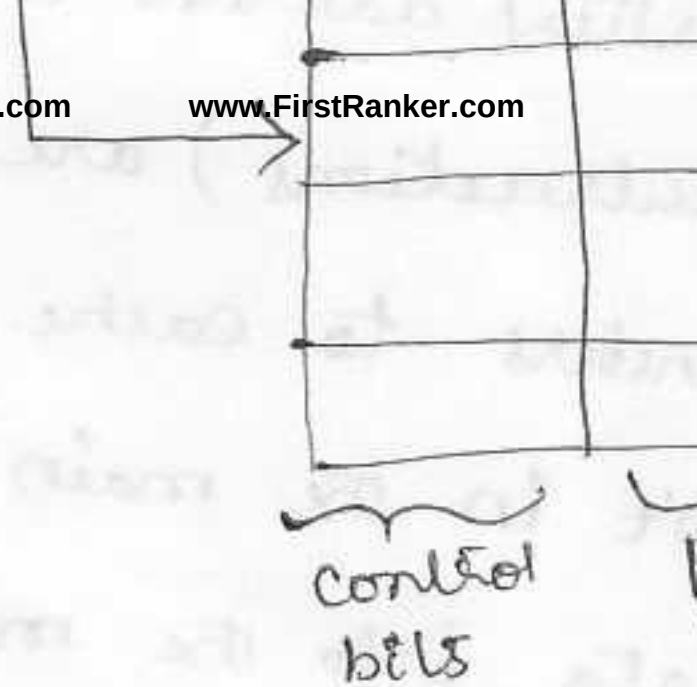
Combination of h/w & s/c

It is a virtual address space or data space that is currently in use. It contains the contents of the appropriate memory location accessed immediately. It is a virtual address space, its contents are mapped to a physical location in the memory.

When the desired data (or the data is fetched and it the data causes the O.S to bring the disk. Transfer of data performed using DMA.

Address Translation :-

A simple method into physical addresses is to are composed of fixed-length consists of a block of words in the main memory.



Each virtual address is for an instruction batch. It is interpreted as a virtual address followed by an offset (location of a particular instruction).

This information is used to find the memory that can hold one address.

The starting of the page

correspond to the most &

usually called the Trans

The operation
table in the main memory

The information that C
TLB must also include

The big. show
the associative mapping table

The contents
of page tables in the memory
of page tables, it must store
entries in the TLB.

One of the C
this perhaps. When an en

acquire the new information
response to access misses.

Address translation may

- 1) Given a virtual address,
page in the TLB.
- 2) If the page table entry
the physical address is
- 3) If there is a miss in the
obtained from the page d
the TLB is updated.

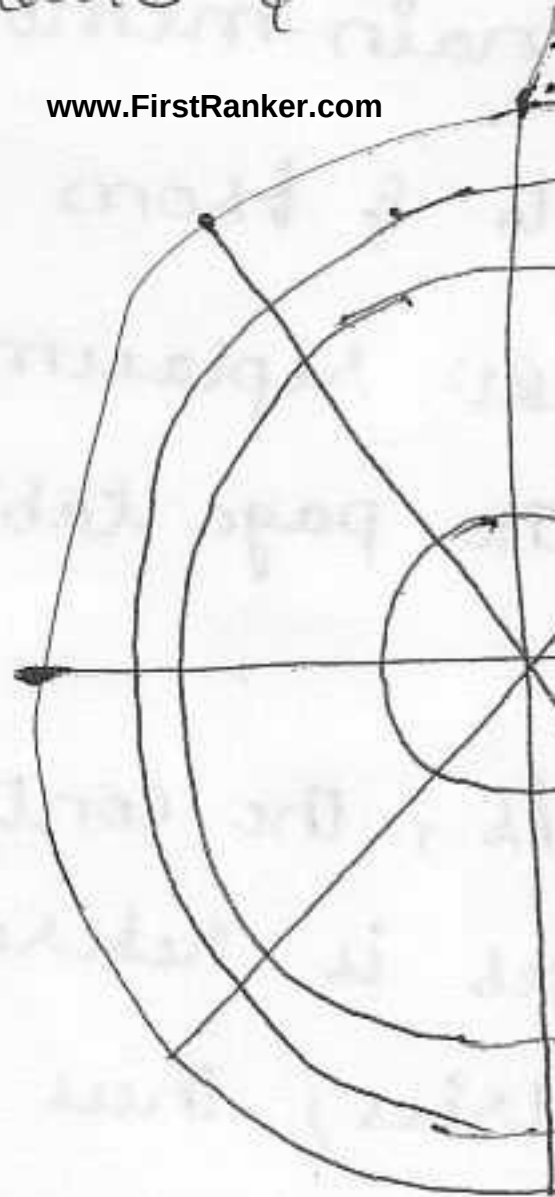
should have large portions in
the frequency of transfer

Here also, the
and the control bits in
usage.

One simple scheme
-ever the corresponding
this bit in all page-table
way of determining which

A modified page
before it is removed from

The address structure
some time to perform, and
to look up entries in the



Some units use a track address surface. The track address to move the head into the system. In other disk systems, separate each track in each surface. particular track electrical

Control mechanism for a magnetic tape

The tape itself is magnetic recording. Bits are recorded along several tracks simultaneously. Each track has a parity bit. Read/write heads move across the tape so that data can be read or written in characters.

Magnetic tape is used to move forward or is recorded in blocks. Reverse tape are inserted b/w.

The tape starts moving at a constant speed.

The data belonging
partitioned then, each partit
way each of these strips are
stripping.

Hence RAID is depe

- 1) Stripping of data in order
- 2) Data redundancy to iner

RAID has different

RAID Level 0 :-

In this level, stripes
introduced in each disks,
first disk, second strip is
of disks ends, the same pr
disk 1 as initiating disk.



identical disk. Hence, he

This can be

S_1

S_4

S_7

S_{10}

S_1

S_4

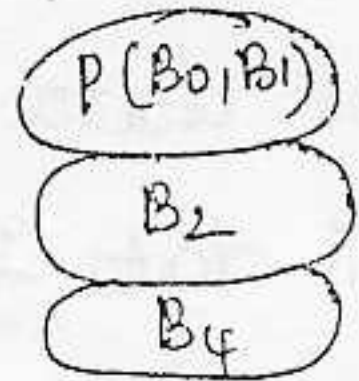
S_7

S_{10}

RAID level 3 reflects

well as stripping. Instead
a single large redundant
detecting & correcting code
parity bit is added to all
disk.





SRAM

- 1) Complex CKT
- 2) Costlier
- 3) Use latch & translational
- 4) Can be used as Cache
- 5) Temp. storage permanent
- 6) Current flow when it is
- 7) Has ~~latch~~ latch
- 8) Less power consumption
- 9) we store automatically in

* Here add & diff but not data

* If over write doesn't takes place

* Here data i/p & data o/p reg's are present.

* It can be used with clock above 100MHz

SRAM

- 10) Manipulated by selection of bits
- 11) Continuous power supply is req'd
- 12) fastest manipulation speed

These peripherals may be a parallel.

The common peripheral units and printers.

peripherals that peripheral system are magnetic disks &

Monitor And Keyboard :-

The most commonly They consist of a keyboard display unit as the O/p

There are different most popular use is Cathode

screen through the
In the block
stored in a local memory
is transferred to the com

Printer :-

printers provide a
computer output data or I

There are three dif

1) Daisy wheel printer 2) D

The daisy wheel p
characters placed along the
character, the wheel rotates
magnet then presses the let

Input - output Interface :-

I/O interface provides information between internal

The purpose of I/O interface is to resolve the differences that exist between the computer and each peripheral.

The major differences are

- 1) peripherals are electromechanical and their manner of operation is different from the operation of the CPU and memory devices. So, a conversion is required.
- 2) The data transfer of peripheral devices is slower than that of the CPU and memory.

Two main types of
that corresponds to the
2) input-output interface
output device.

I/O Bus and Interface Module

The communication
and several peripherals is

The I/O bus
lines and control lines. The
terminal are employed in

Each peripheral
an interface unit. Each

the transfer between perip
has its own controller tha
mechanical device.

The I/O bus
all peripheral interfaces.

To communicate with
device address on the address
to the I/O bus contains a
the address lines. When the
it activates the path betw
that it controls, remainin
interface.

It is used to
in the interbase and the p

Data o/p Command :-

It causes the in
data from the bus into o

Data i/p Command :-

The interbase
from the peripheral and p

The processor checks if da
a Status Command and

Then the interbase places
where they are accepted

can be implemented, where
in addition to CPU.

The memory
CPU and the IOP through
communicates with the in
a separate I/O bus with
control lines.

The purpose of
independent pathway for I
external devices and internal

The IOP is

associated with the instruction lines. At mean time, it write control line. This that are attached to the in the address lines is to not for a memory word.

When the CPU is operand from memory, on the address lines and or memory write control external components that word and not for an I

The Computer

Can use memory - type I/O

It allows the computer
for either I/O transfers

The advantage of memory-mapped I/O is that instructions used for memory access can be used as I/O registers.

Example of I/O interface

An example of I/O interface in the big.

CS	RSI	RSO
0	X	X
1	0	0
1	0	1
1	1	0
1	1	1

will only p/p data. But
data in both directions,
bidirectional lines.

A command
sending a word to the appx
system, the function code
because control is sent to
information is received from
transferred to and from
transfer of data, control
through the common data b
The differ

for eg, a status bit received a new data item bit indicates that, there data transfer.

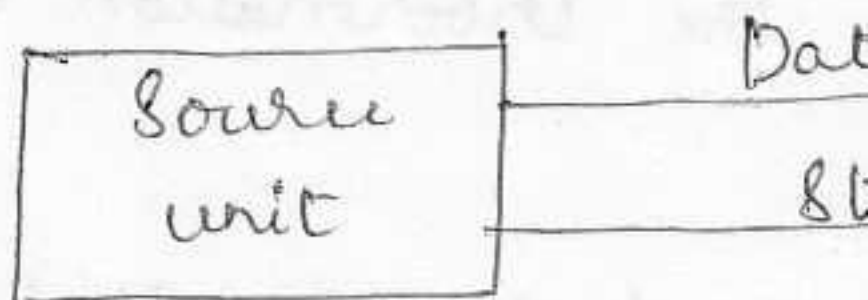
The interbase reg CPU through the bidirectional bus selects the interbase the two register select in

A circuit detect the address assigned circuit enables the chip is selected by the address

unit receiving the data item
Signal to acknowledge receipt
agreement between two indep

Strobe Control :-

The Strobe Control
employs a single control line
may be activated by either
(i) Source-initiated transfer :-
The below
transfer.



The data latches to a
activates the strobe pulse.

The information on
signal remain in the active
to allow the destination unit

The destination unit
strobe pulse to transfer
one of its internal registers

The source removes the
period after it disables it
source does not have to cha
bus.

The strobe signal is dis

to trigger a destination

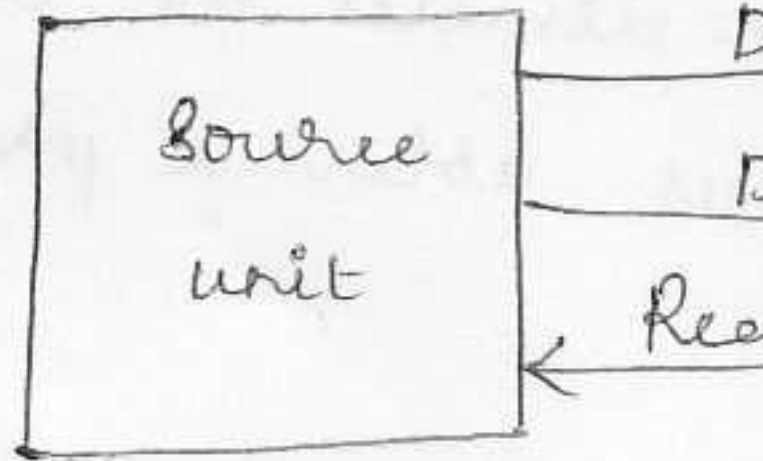
The destination unit
removes the data from the
interval.

Hand Shaking :-

In the strobe
destination unit to source
particular data item or

The same happens
whether the source unit has
bus.

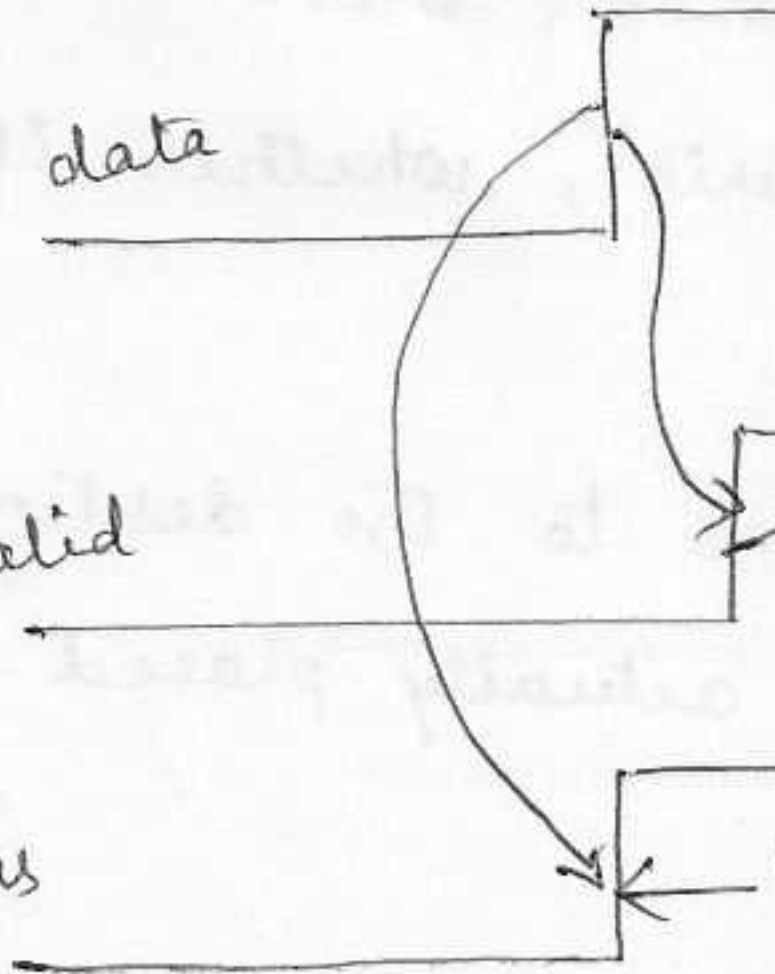
the data transfer



Ready for data

Data Valid

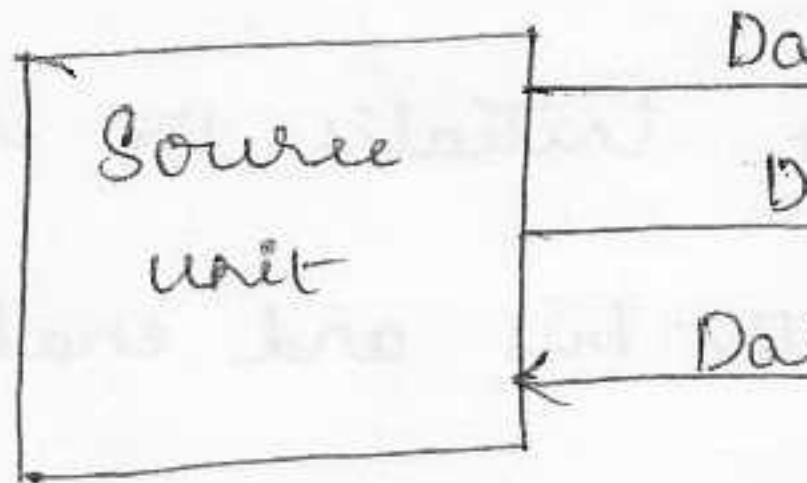
Data bus



data.

The source unit
until after it receives the
destination unit.

Source - initiated transfer



Enable data Valid

Disable data Valid
Invalidate data on bus

The source
data by placing the data on
'data Valid' signal.

The data acc

not be completed. Such a
of a timeout mechanism,
data transfer is not comp
The timeout is implemented
that starts counting time
handshaking control signals

If the receiver
respond with in a given time
an error has occurred.
interrupt the processor and
that takes appropriate error

In synchronous, the
at the rate of clock pulses.
Separate clock of the same

In asynchronous, binary
when it is available and
is no information to be transferred

In a asynchronous
special bits are inserted
Each character consists of

- 1) Start bit
- 2) character

The first bit called the start bit
it indicates the beginning of

The last bit called the stop bit

← start bit →

The 'baud rate' is defined as the number of bits of information transmitted per second.

Integrated circuit (UART) is designed to provide the interface between interactive terminals and a computer communication interface or a transmitter (UART).

The operation is to place a byte to the control register. The interface is a device that defines certain parameters such as how many bits are in each character, check parity and how many characters.

In status register, one bit is used to indicate if the transmitter is empty and another bit indicates if the receiver register is full.

Transmitter :-

The operation of transmitter

- The CPU reads the status

- The interbase is double
can be loaded as soon as

Receiver :-

- The receiver data input
line is idle.

- The receiver control m
a '0' signal to detect the
start bit has been detected,
are shifted into the shift

- After receiving the data
parity and stop bits.

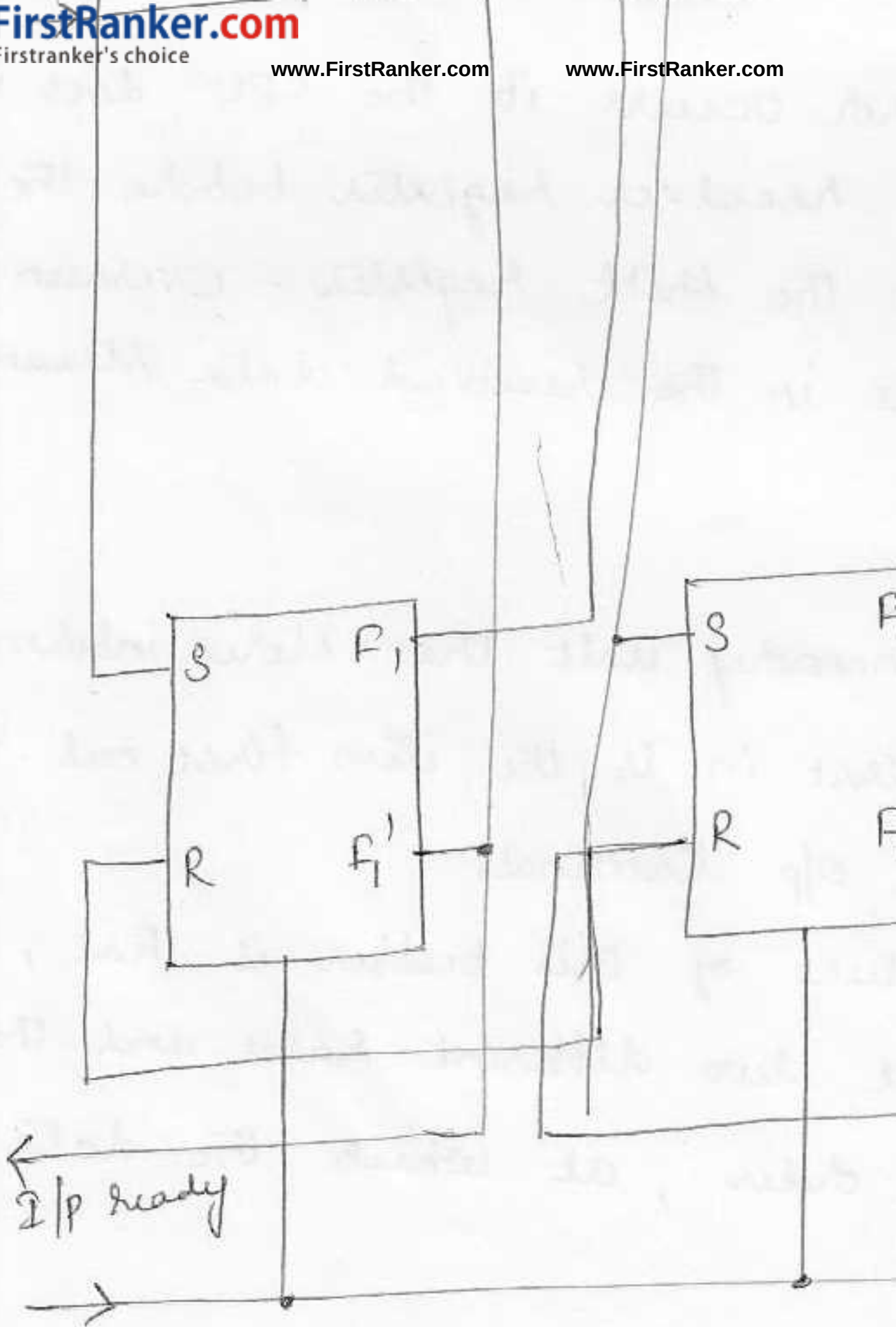
- The character with out the

An overrun
read the character from the
next one becomes available
results in a loss of character

First-In, First-Out Buffer

FIFO buffer is a
in a manner that the items
survive with separate i/p and o/p

The important feature
can i/p data and o/p data
data are always in the same
entered the buffer.



When ever the F_i bit is 0
and the F_{i+1} bit is reset
causing register $R(I+1)$ to

The clock transition set
causes the control flag to
together with the data.

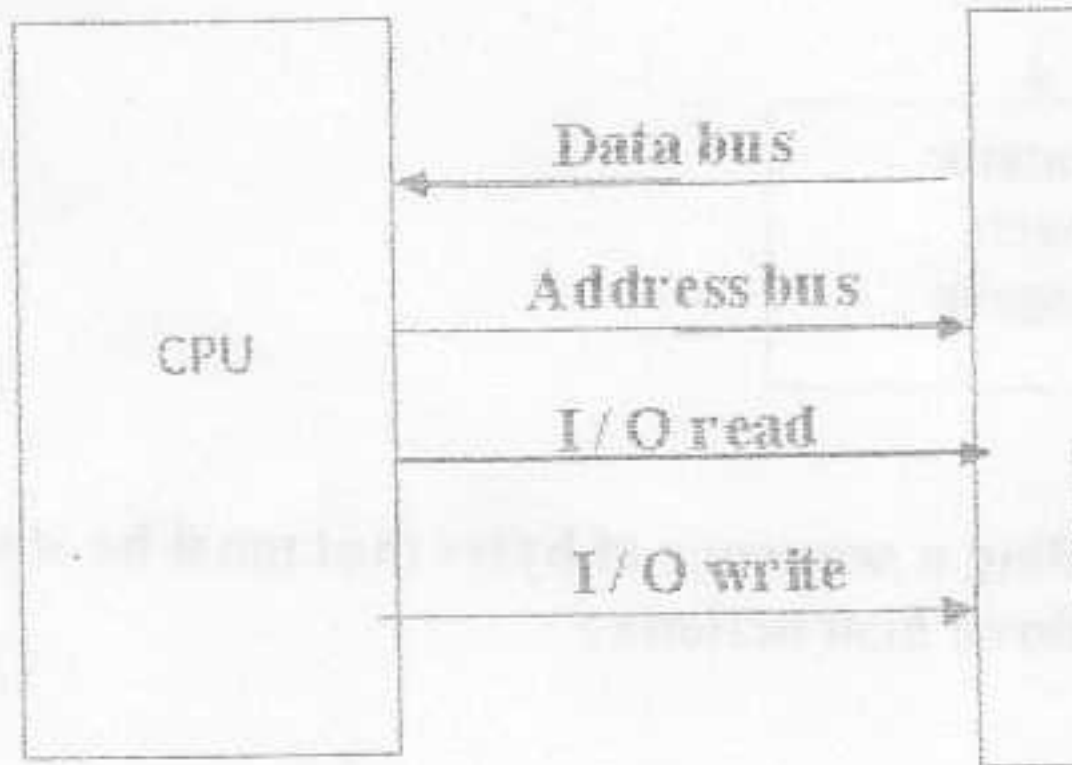
Data are inserted
input ready signal is enabled
control flip-flop F_1 is reset
empty.

Data are loaded the
clock is R_1 through the 'in

unit, which then enables

F_4 causing o/p ready to d
on the o/p are no longer

An example of data transfer
the CPU is shown in below fig:



The device transfers bytes of information available, the device places it in the I/O interface accepts the byte into its data register. The interface sets a bit in the status register to indicate the line, but it will not transfer another byte until the interface.

Now assume that the device is in memory . The transfer of byte requires

- 1) Read the Status register .
- 2) Check the status of the flag bit and
- 3) Read the data register.

Each byte is read into a CPU register and store instruction .A common I/O device transfers words from an I/O device and store

In vector interrupt, the source of information to the computer. This

Direct Memory Access (DMA) :

The transfer of data between peripheral devices is often limited by the speed of the CPU. For peripheral device manage the memory transfer. This transfer technique is called DMA and has no control of the memory buses manage the transfer directly between the

The CPU may be in idle state. One of the microprocessors is to disable the buses

The below fig. shows the control of DMA transfer.

DMA Controller:

The below figure shows the DMA data bus and control lines.

The registers in the DMA are enabling the DS (DMA select) and RS (rebidirectional). When the BG (bus grant) fr registers through the data bus to read fro

When BG=1 , the CPU has th the memory by specifying an address in th

The DMA communicates with acknowledge lines.

The DMA controller has thre

- 1) address register
- 2) word count register
- 3) control register.

The control register specifies the CPU as I/O initiator. Thus the under program control via databus.

The CPU initial through the databus:

- 1) The starting address of the memory data are to be stored (for write)
- 2) The word count, which is the
- 3) Control to specify the mode of
- 4) A control to start the DMA transfer

DMA transfer :

The position of the DMA control is shown in below fig.:

The CPU communicates with any peripheral device through its own address, which activates and initializes the DMA through the system bus.

Once the DMA receives the start signal, it starts the transfer between the peripheral device and the system bus.

When the peripheral device controller activates the BR line, the CPU deactivates the system bus.

X [The CPU initializes the DMA, and the DMA receives the start signal to begin the transfer between the peripheral device and the system bus.

or write operations and supply

The peripheral unit
through the data bus for direct
while the CPU is momentarily

For Each word that
its address register and data
If the word count does not
request line coming from the

for a high speed data
as the previous transfer is

initiated, and the process continues
transferred.

Channel has a request and
which are connected to separate

Each channel also has
word count register with in the
among the channels may be
with high priority are service
priority.

It is used for fast
between magnetic disks and

I/O processing:-

We know that,
set up by the CPU, with
its own instructions.

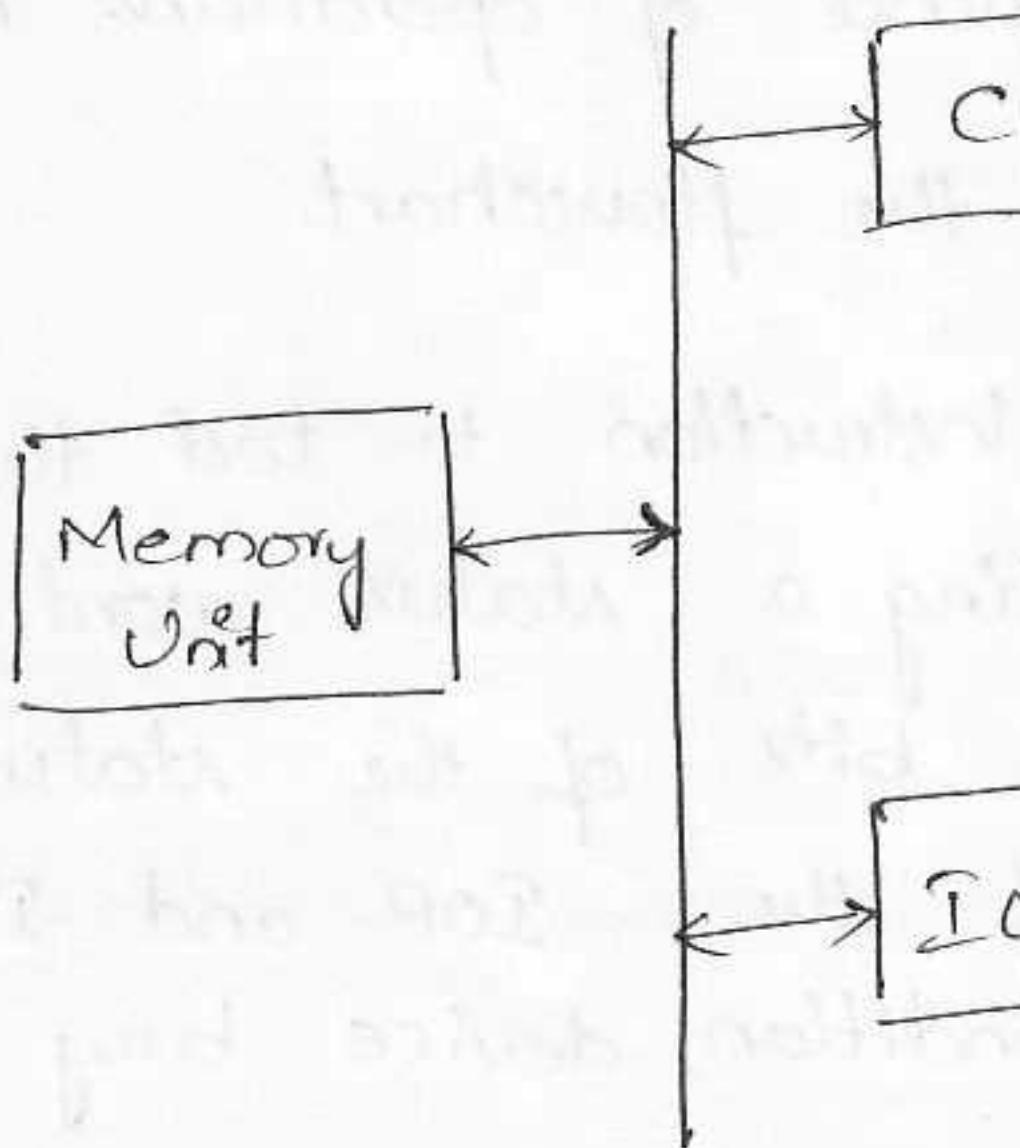


fig:- Block Diagram

direct access method.

CPU-IOP Communication:-

The communication take different forms. The carried out as shown

The CPU sends an
The IOP responds by in
for the CPU to check. Th
indicates the condition
such as IOP overload
another transfer etc.,

Check status word
for correct transfer

↓
continue

fig: CPU - I/O

The CPU refers
to decide the work to be
instruction to start I/O transfer
with this instruction refers
continue with another program
the I/O program.

The D/D processor is called a channel. A typical includes a number of channels attached to one (or) more

There are three types

i. Multiplexer

ii. Selector

iii. Block Multiplexer

The Multiplexer channels a number of slow and medium speed data capable of operating with multiple channels simultaneously.

Command
code

www.FirstRanker.com

Data
Address

www.FirstRanker.com

(c) Channel command

fig:- IBM 370 I/O re

The I/O instruction of code, channel address, and device address of channels of computer connected to several devices by device address. The operations of I/O instructions: start I/O, clear I/O, halt I/O, halt device. The CPU can check the Status Word to determine

channel.

The command field specifies one of the six

1. Read: Transfer data from

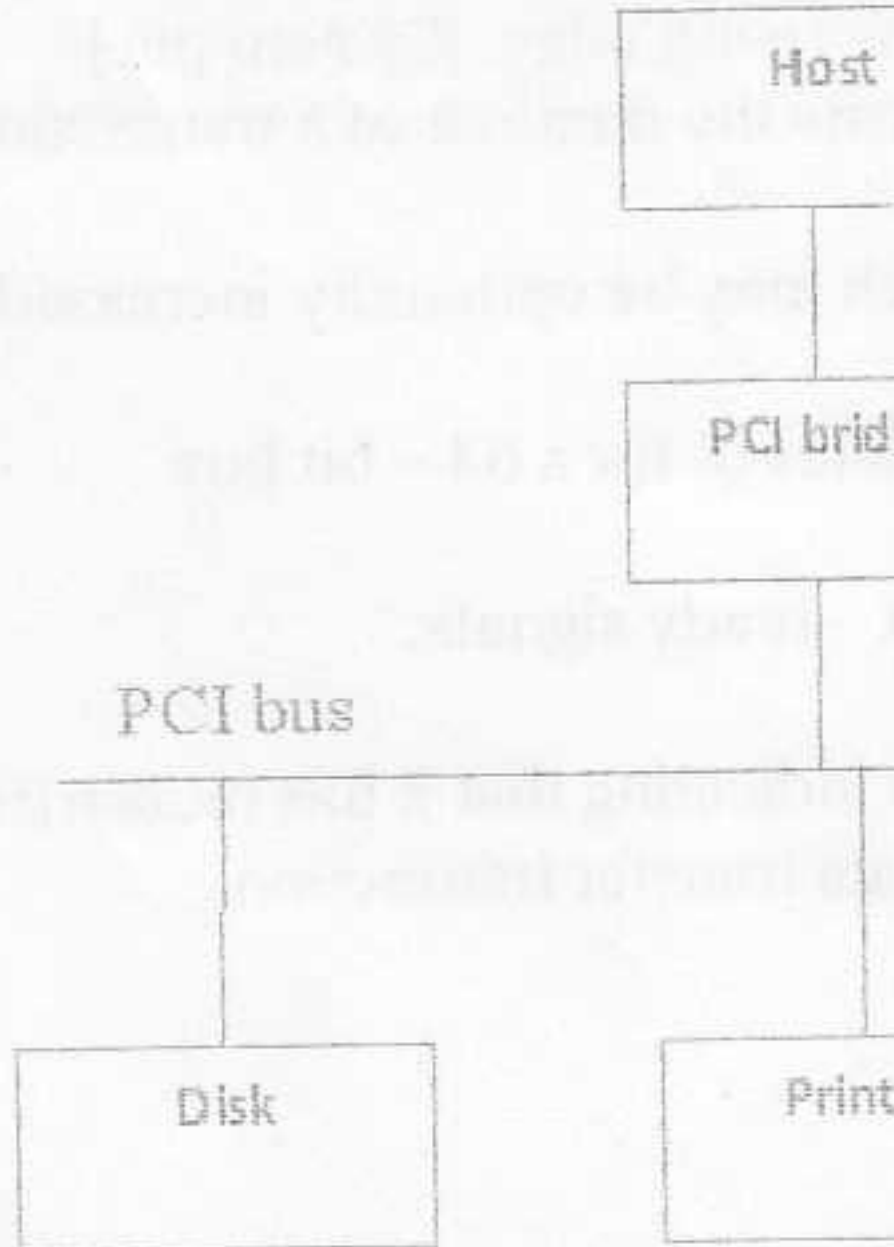
2. Write: Transfer data from

3. Read backwards: Read magnetic

4. Control: Used to initiate
of data.

5. Sense: Informs the channel
word to memory

6. Transfer in channel: - Used to
the data
address
executed



1. CLK -

A 33 MHz clock. P is sampled by all inputs

2. FRAME# -

Sent by the initiator to [Interface ctrl pins]

3. AD -

32 address / data lines [Address & Data pins]

4. C/BE# -

4 command / byte - enable [64 - bit bus extension]

5. IRDY#, TRDY# -

Initiator - ready and [Interface ctrl pins]

6. DEVSEL# -

A response from the controller Address and is ready for

(Device Select

[Interface ctrl pins]

7. IDSWL# -

Initialization Device Select

four parameters:

1. The baud rate of the transmission
2. The number of data bits encoding a character
3. The sense of the optional parity bit
4. The number of stop bits

Each transmitted character is packaged in a character frame, which is followed by the data bits, the optional parity bit, and the stop bits. A typical character frame encoding the letter m.

transmitted in about 1.146 ms.
Data bits are transmitted upside down and the order of transmission is from least significant to most significant. To interpret the data bits in a character frame, you need to invert the logic, 1 for negative voltage and 0 for positive voltage.

ASCII conversion table shows that this is the character 'A'.

An optional parity bit follows the data bits and also follows inverted logic, 1 for negative voltage and 0 for positive voltage. It is included as a simple means of error handling. The parity of the transmission is to be even or odd. If the parity is even, the transmitter sets the parity bit in such a way as to make the total number of 1s in the frame the parity bit. This transmission uses odd parity. Since the data is already an odd number, so the parity bit is set to 1.

The last part of a character frame consists of a stop bit, represented by a negative voltage. If no further data is to be transmitted, the negative (MARK) condition. The transmission is terminated by a start bit of positive (SPACE) voltage.

Hardware Overview

There are many different recommended standards for serial communication, following most common types.

Function	Signal	P
Data	TxD	
	RxD	
Handshake	RTS	
	CTS	
	DSR	
	DCD	
	STR	
Common	Com	
Other	RI	

The DB-9 connector is occasionally found on serial ports, yet has enough pins for the core set of signals specified in the RS-232 standard. Table 1 shows the signals specified in the standard. Table 2 shows the most RS-232 interfaces.

connected to a host computer. USB was designed using a single standardized interface socket allowing hot swapping; that is, by allowing rebooting the computer or turning off the device power to low-consumption devices, eliminating the need for allowing many devices to be used without reinstallation to be installed.

USB is intended to replace many varieties of computer peripherals such as mice, keyboards, digital cameras, printers, personal media players. In many of those devices, USB has become the standard for personal computers, but it has become common for video game consoles, and as a power cord between a wall plug for charging. As of 2008, there are about 6 billion total sold to date.^[1]

The design of USB is standardized by the USB standards body incorporating leading companies and industries. Notable members have included Apple Inc., Hewlett-Packard, Intel, Microsoft

USB device communication is based on *pipe* the host controller to a logical entity on the occasionally used to incorrectly refer to the pipes, 16 into the host controller and 16 out

Each endpoint can transfer data in one direction. A pipe is uni-directional. Endpoints are grouped with a single device function. An exception to this configuration and which is not associated with

When a USB device is first connected to a USB host, the enumeration process starts. The enumeration starts by sending a GET_DESCRIPTOR request to the USB device. The device is determined during the reset signal. The device information is read by the host, then the device is supported by the host, the device drivers are loaded and the device is set to a configuration. The enumeration process is repeated for all connected devices.

The system is commonly used for connection cameras, but is also popular in industrial systems. It is preferred over the more common distribution capabilities, and because it does important, FireWire makes full use of all SCSI transfer rates, a feature especially important that the sustained data transfer rates are high. Apple Mac OS X with more varied results on

However, the royalty which Apple Inc. and other users of FireWire (US\$0.25 per end-user system implement it (US\$1–\$2), both of which have from displacing USB in low-end mass-market major constraint.^[3]

not intended to be a primary bus for a computer. Serial ATA (SATA), and SCSI fulfill that role in that it is possible to install and remove devices it useful for external drives. Originally conceived for devices (CD-RW drives, DVD drives, etc.), portable USB hard drives, or empty enclosures to internal drives^[citation needed]. These external interfaces a drive of conventional technology to a USB port. Functionally, the drive appears to be competing standards that allow for external

Another use for USB Mass Storage devices is for clients without the need of installation on the host or clients^[6] etc.

IEEE 1394

The IEEE 1394 interface is a serial bus interface and isochronous real-time data transfer, frequently used for digital audio, digital video, automotive, and other applications known by the brand names of FireWire (Apple) and iLink (Sony Instruments). IEEE 1394 replaced parallel SCSI.

A multi processor system provides interaction b/w

the system cooperate in the

The benefits of this organization is an improve drives its high performance can proceed in parallel in or

- 1) Multiple independent jobs
- 2) A single job can be part

Multiprocessors are memory is organized.

- 1) Tightly Coupled Multiprocessor
- 2) Loosely coupled Multiprocessor

addressed to a specific processor.
available processor.

Loosely coupled
the interaction b/w tasks.
coupled systems can tolerate
b/w tasks

Interconnection Structures

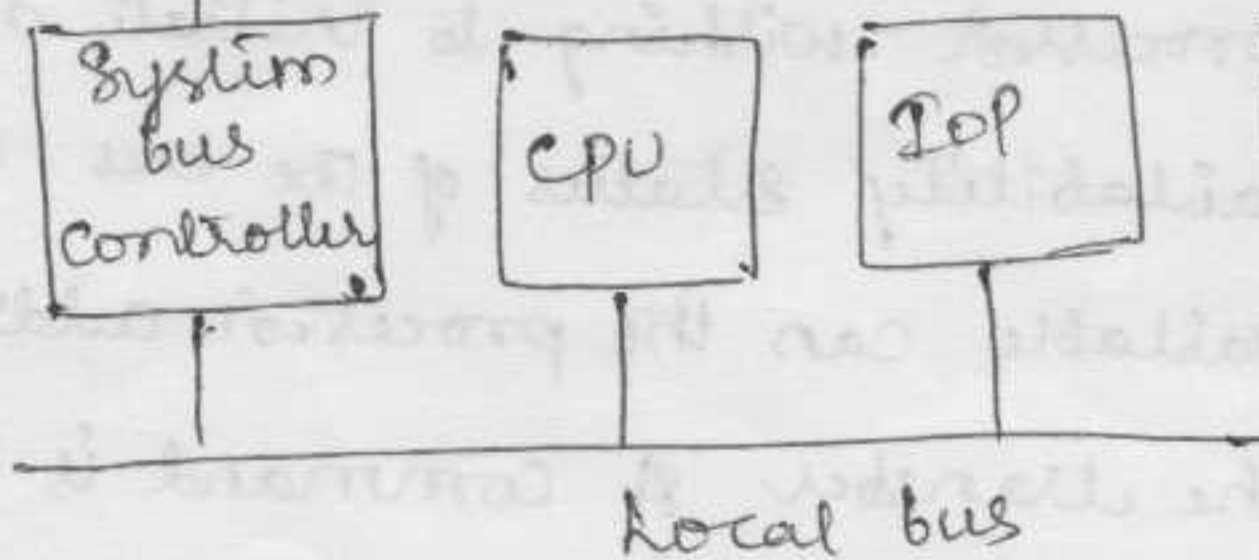
There are several
establishing an interconnect

- 1) Time - shared common bus
- 2) Multipoint Memory
- 3) cross bar switch

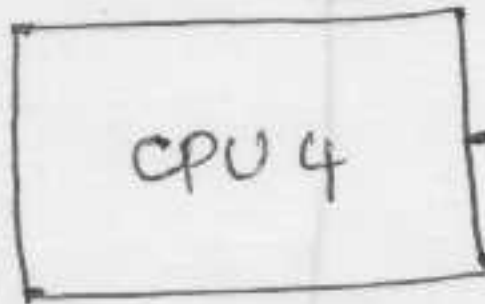
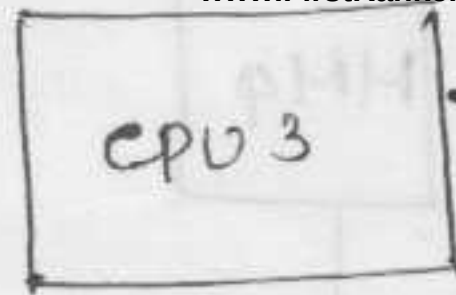
operations are conducted by bus at the time. Any other must first determine the only after the bus becomes destination unit to initiate to inform the destination. The receiving unit recognizes responds to the control signal the transfer is initiated.

Conflicts since one common. These conflicts must be resolved that establishes priorities.

A single common transfer at a time. i.e.



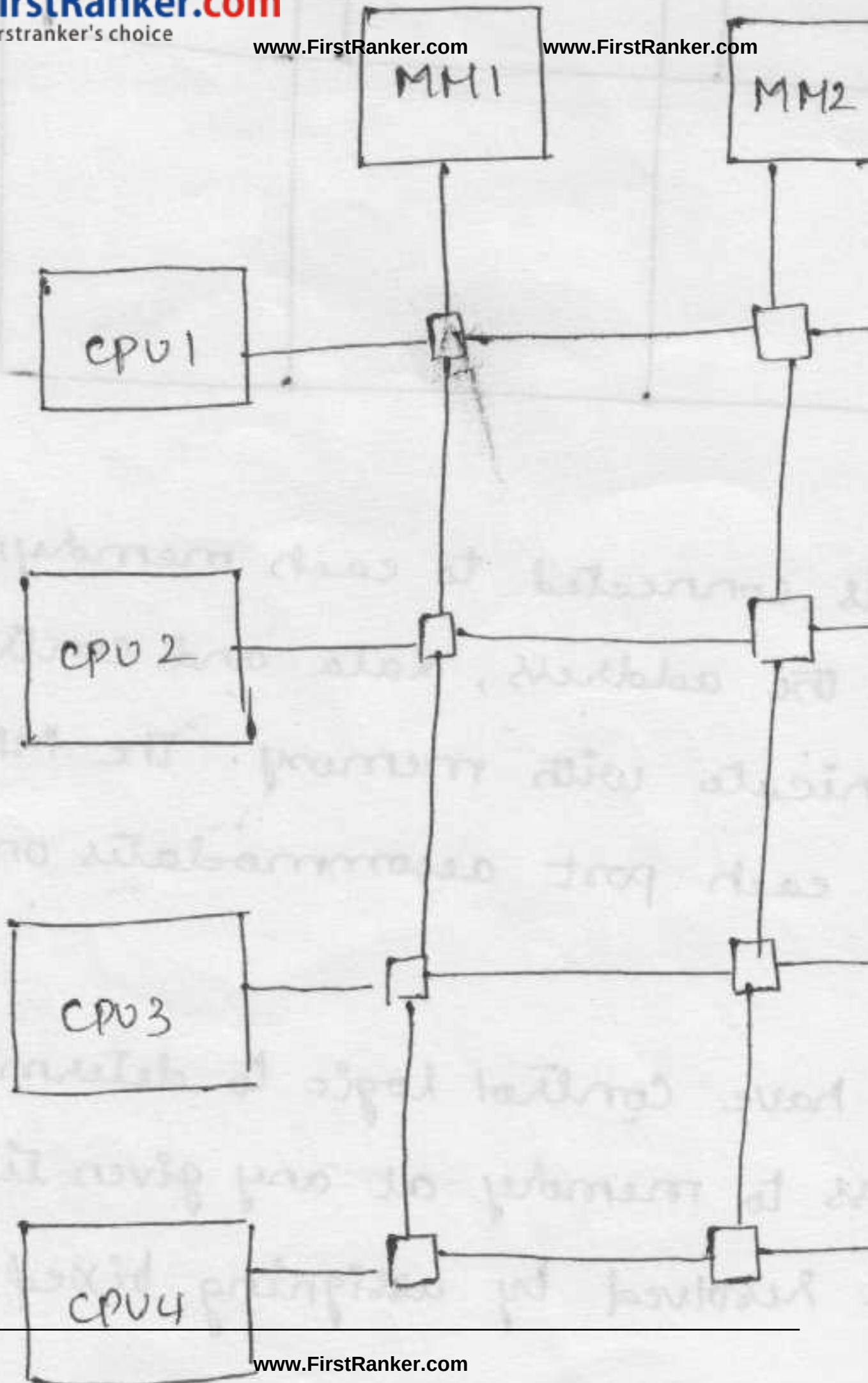
In this, there consists of local memory & to one or be connected to CPU, an IOP. A system bus controller line system bus. The memory is shared by all processors. designed as a cache memory way, the average access time to approach the cycle time

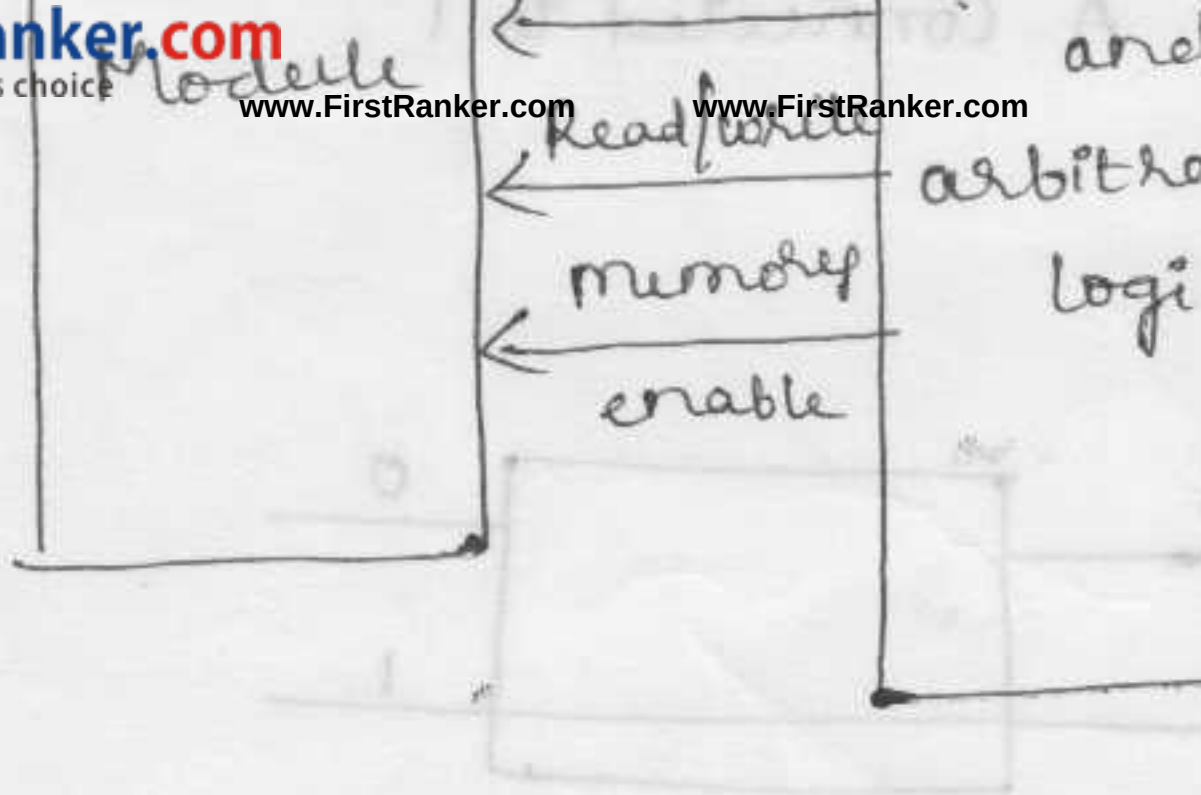


Each ~~A~~ processor bus

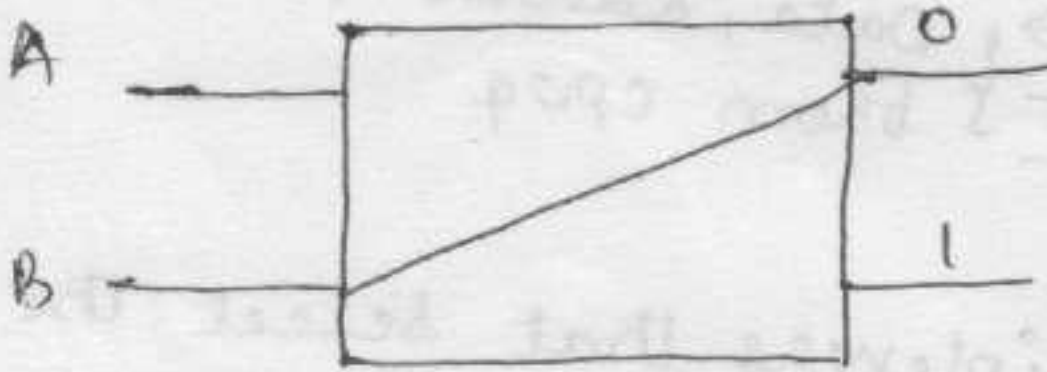
A processor bus consists of lines required to communicate. Said to have four ports to the buses.

The Module must which port will have a memory access conflicts.





The circuits connect data, address & control with the memory module. priority levels are to select one CPU when to the same memory. The binary code that is generated the arbitration logic.



B Connected to 0

The switch has the capability of the output's. If inputs A & B only one of them will be connected.

Using the 2x2 switch, now to control the communication.

The construction of binary tree

path from a source to a destination
bits of the destination number

The first bit of the
switch output in the first level.
Second level. The third bit of the

Many different topologies
multistage switching networks to
in a tightly-coupled multiprocessor
between the PE's in a loosely-coupled
the omega switching network where

there is exactly one path from
destination. Request patterns
for eg, any two sources can

destination number. As the
each level examines a different
setting. Level 1 inspects the
level 3 inspects the LSB. When
the 2×2 switch, it is routed
is '0' or to the lower output if

In a tightly
processor & the destination
through the network sets up the
transfer the address into the
direction, depending on whether

In a loosely
destination are PEs. After
processor transfer a message



Two-cube

A one-cube structure has processors interconnected by a square. A two-cube structure has $n=2$ & $2^2=4$. It consists of a cube. A three-cube structure has a cube. A n -cube structure has n processors residing in each node. Each node is connected to exactly one bit position.

A Memory bus
data, address & read/write
to transfer information
bus that connects major
such as CPU's, IOP's & ROM's

The processor
same time. Arbitration
multiple contention for the
logic would be part of the
local bus & the system bus

System Bus :-

A typical system
lines. These lines are divided
i.e., data, address & control

of approximately
synchronization signals are

In Asynchronous bus,
accompanied by hand shake
the data are transferred
destination.

The Control lines
information b/w units. The
validity of data & address
specify operations to be performed.

Some of the control
write, acknowledge of a
control signals such as bus
arbitration procedures.

the line is a
the busy line while the lower
of the bus completes its opera
to its inactive state, the hi
busy line & then required ~~ack~~ ^{data}

parallel Arbitration logic

This technique use
decoder can be shown in

Each bus arbit
request o/p line & a bus a
enables the request line wh
to the system bus. The pro
acknowledg i/p line is ena

The request lines from the priority encoder. The output of which represents the highest-priority bus.

The 2-bit code from which enables the proper address to the highest-priority unit.

The bus priority signals are used for daisy-chain.

The signals used in the procedure are bus request (BR) and bus clock (BCLK) is used to synchronize the bus.

interval. The priorities are
device an opportunity to a

In first-come, first
in the order received. To
controller establishes a queue
that the bus requests arrive
its turn to use the bus on

The rotating daisy
extension of the daisy-chain
is no central-bus controller
from the priority-out of
of the first device in a close

determine to there the value
of this procedure can be

An efficient processor
to alert the receiving processor
signal. This can be ~~also~~
processor interrupt by means
one processor which when
condition in a second processor

In addition to shared
may have other shared resources
unit connected to an IOP
provides a facility for sharing
is also established.

To prevent corruption
several processors there must

In a distributed O.S.,
among the available processes
assigned to only one processor
assigned to different processors
suitable for bloating operation

In a loosely
is distributed among the p
for passing information
by means of message passing

A message is the
objects used to communicate
possible paths available to se
The O.S. in each node contain
alternative paths that can be

basic mechanisms for primitive

for more complex mechanism

The h/w mechanism

i.e., binary semaphore.

Mutual Exclusion with a Semaphore

The multiprocessor

the orderly access to shared

is necessary to protect the

has been called 'mutual exclusion'

Mutual exclusion

enable one processor to ex

resource by other processes

Critical section is a program

A semaphore can be set instruction in conjunction with lock is a processor generation other processors whom using is active.

The Test and set instruction and activates the lock when instruction is being executed.

Assume that of a memory word whose

Let the mnemonic TSL do operation. The instruction

TSL SEI will be executed in two

second (to write)

is to reduce the average information may reside in a memory. The ability of a correctly, the multiple copies arises a 'Cache coherence' problem.

A memory returned on a load instruction the latest store instruction is proper solution to the Cache used in bus-oriented multi

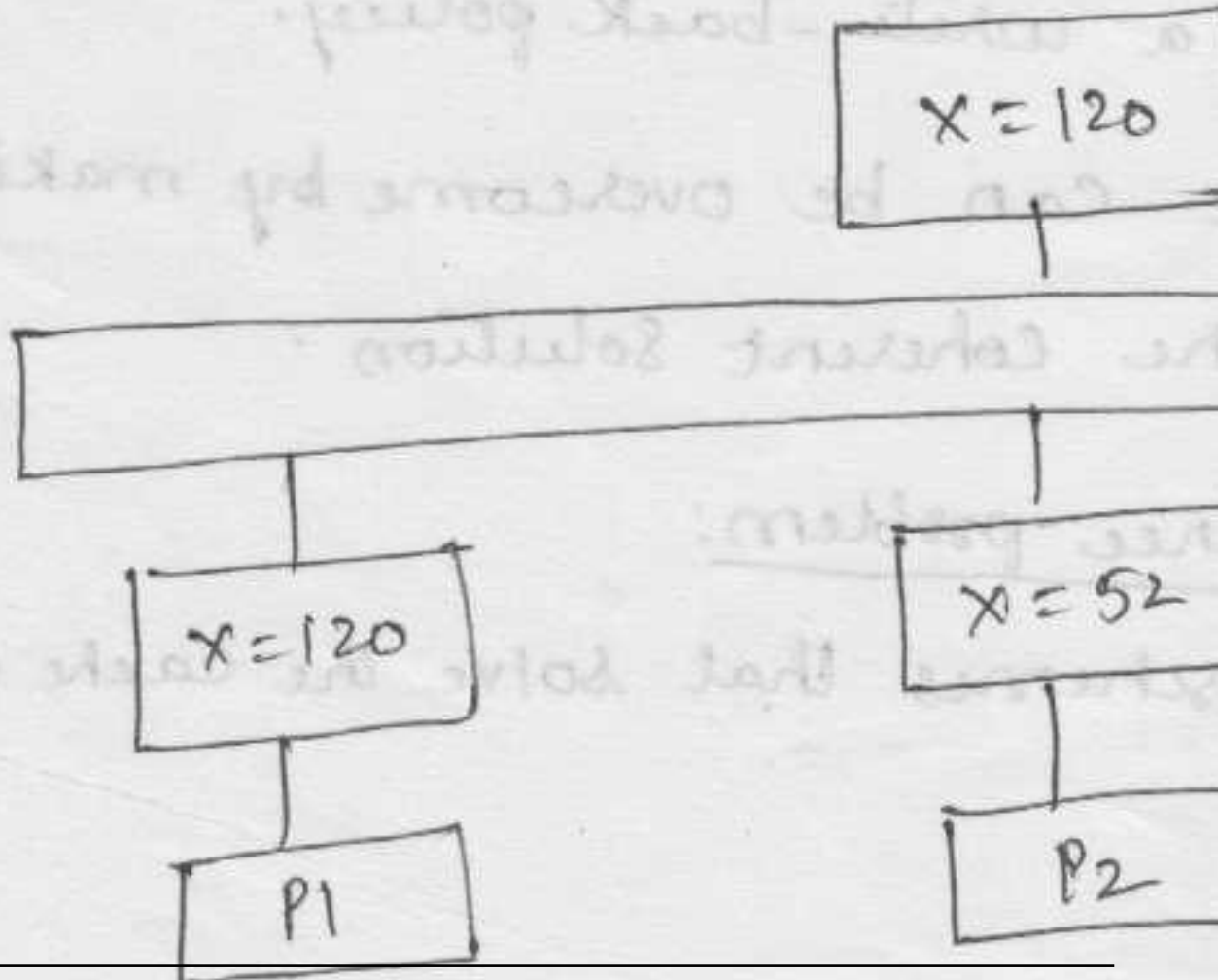
Conditions for Incoherence

Cache coherence exists

because of the need to share be safely placed without

Results in consistent copies in
to one of the processors
of X in the caches become in
will not return the latest value

Depending on the
the main memory may also be
This shown below:



In big (b) Main Memory is

The copies in other
inconsistent.

Another configure
problems is a DMA. During
that also reside in cache with
DMA o/p, memory locations
from the cache, while using

I/O based inconsistency
Top a participant in the C

Solutions to the cache coh

We have different
problem.

Block is identified as Ke
All caches can have copies
Cache can have a copy of

The cache co
means of a combination
only schemes.

H/w-only solutions are
& have the advantage of h/w

In h/w solution, the
to monitor all bus requests

Depending on the method
invalidate their own cache

The bus controller that monitors
'Snoopy Cache Controller'.

