Chapter

Introduction

Page:
Date: / /

1 Introduction and difference between Computer Architecture and Computer organization:

Computer architecture refers to those attributes of a Computer which are visible to programmers and have direct impact on a logical execution of a program.

Computer Architecture includes the instruction set, the number of bits used to represents various instructions and data types, input - output mechanism and technique for addressing memory.

Computer organization refers to those attributes of a computer which are related with operational unit and their inter connection that realize the architectural specification.

Computer organization include those attributes i.e hardware details transparent to the programmer such as control signals, interface between peripherals and Computers and memory technology.

Example: It is an architectural design that issues whether a Computer will have multiple instruction It is an organization that issues whether that instruction will be implemented by using a special multiply unit or by a mechanism that uses sucessive addition.

CamScanner

Page :
Date : / /

Evolution or generation of Computer system:-

Generation refers to the involvment of technique towards technology. Computer generation is drastically changing from historical period.

The following are the generation of Computer:

(i) First Generation (1946 – 1957 AD)

All the Computers which uses vaccum tubes as an electronic Components are refered to as first generation of Computer.

Some major characteristics of this generation are as follows:

- very big size, unreliable & slower.
- produce a large amount of heat.
- operating speed was in terms of millisecond.
- Machine level programming.
- High cost and not for Commercial use

Example: ENIAC, EDVAC, Mark-I, etc.

(ii) Second Generation (1958 – 1964 AD):-

The Computers of this generation used transistor as an electronic Component.

The major characteristics of second generation are as follows:

- Smaller in size than previous generation as 1 transistor replaces 1000 vacuum tubes
- Faster processing speed in terms of microsecond.
- little bit faster I/O devices were used.
- used assembly language.

Example: Mark-II, LEO, PDP, I, etc.

CamScanner

(iii) Third Generation (1965 – 1971) AD:-

In this generation the computer used IC (Integrated circuit) as main component. Some major characteristics of this generation are:

- Smaller in size, more reliable.
- Generates less heat and consumes less power.
- processing speed increased to nanosecond.
- Computers were portable.
- uses high level language.
- Magnetic disk were used as secondary memory.

Example: IBM 360 series, UNIAC-9000, etc.

(iv) Forth Generaton (1972 –

The computer of this generaton used microprocessor as main electronic ~~computer~~ component. The characteristics of this generation are as follows:

- High accurate and totally reliable.
- operating speed in terms of picosecond or MIPS.
- uses high level programming language.
- magnetic and optical storage.
- multiprogramming, multimedia, graphics and distributed computing is possible.
- Advancement in Software package.

Example: IBM-PC, pentium-PC, etc.

(v) Fifth Generation of Computer (not yet developed):-

The computers which will use AI and ULSI chip as main electronic circuit and 'BIO-chips' as main

Component are known as fifth generation Computers.

The features of Computers of this generation will be as:

- Ability to understand natural language.
- equivalent with multiprocessing & parallel processing
- PROLOG & LISP as programming language.
- Voice recognition & vision system will be provided.

[structure of IAS Computer]

Von-Neumann Architecture:-

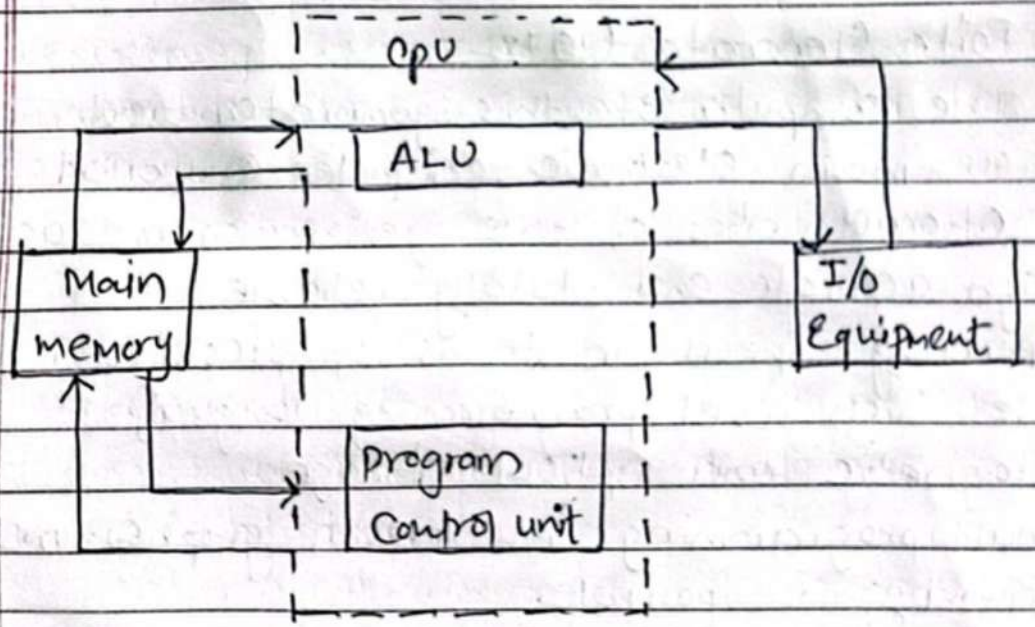


fig:- General structure of IAS Computer.

It is the 'stored program concept' (SPC) developed by Von-Neumann in 1940's. In development process of Computers so many new concepts are invented and used one of which is

Page:
Date: / /

stored program Concept. In this the instruction and data are stored in memory before executing by the processor.

In 1946 AD von-Neumann and his friends designed stored program Computer called IAS (Institute for Advance study).

(i) Main memory:-
It stores data and instruction before and after execution of the instruction.

(ii) ALU:-
It performs arithmetic & logical operations such as addition, subtraction, multiplication, AND, OR, NOT, etc.

(iii) program Control Unit:-
It interprets instruction in memory & cause them to be executed.

(iv) I/O equipment:-
It performs input-output operations.

Memory and Instruction format of von-Neumann machine:-

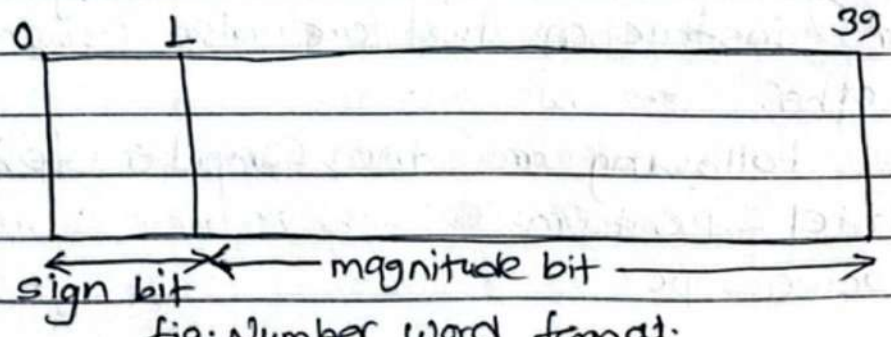


fig: Number word format.

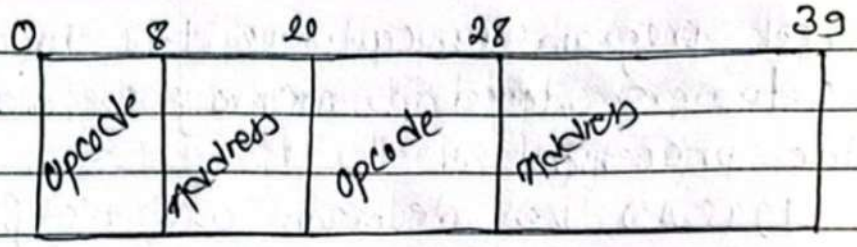


fig: Instruction word format.

IAS machine Consist of thousands memory location, 1 location is called word and it has 40 bits. Both data and instruction are stored in these locations.

The memory format are given ~~below~~ above. A number word Contain 1 bit sign bit and remaining 39 bit is the magnitude bit. The instruction word Contain two 8- bit for op code and two 12- bits for address.

The address must be started from 0 to 999.

Computer families:-

A Computer family is a Category with the ~~small~~ same design and microprocessor that are Compatible with each other.

In general a family is a group of product that have similarities, ~~Capa~~ Compatible and made by same manufacture Company. It has similar instruction that are also Compatible with each other.

Following are the Computer families:-

- Intel - pentium.
- power - pc

Page:
Date: / /

17 Intel - pentium (CISC):-

The intel - pentium represents the result of millions of design efforts on CISC architecture. It incorporates the sophisticated design principle on a computer.

Following are the excellent example of this family.

Family	Features
8080	- 8 bit machine with 8-bit data path to memory.
8086	- 16 bit machine - Support instruction that fetches few instruction before execution.
80286	- It is extension of 8086 with 16mb memory.
80386	- first intel processor to support multi-tasking, multiprogramming & multi-processing
80486	- Support Cache memory technology and instruction pipelining.
Pentium	- 32 bit - used superscaler technology.
Pentium-pro	- Branch prediction of data flow analysis

pentium - II	- process audio, video, graphics and multimedia by incorporating with intel technology.
pentium - III	- Support 3D graphics and multimedia software.
pentium - IV	- It also supported 3D graphics and multimedia.
Itanium	- 64 bit organization with 64-bit architecture.
Itanium 2	- Support hardware enhancement to increase speed.

2) power_pc:

It represents the RISC architecture concept. The power pc is used in millions of Apple Macintosh machine and in numerous embedded chip application.

The following excellent example of this family.

Member	Features
	- 32 bit machine - Its main purpose is to bring power pc architecture to market value

603	- 32 bit machine. - lower cost, more efficient and portable.
604	- 32 bit machine - Advance Superscalar technology for better performance.
620	- first member of power-pc family to implement 64-bit architecture 64-bit register and data path.
740/750	- Used Cache technology. - provide Significant performance improvemac
G4	- Increase internal speed of processor chip.
G5	- Increase processor speed by using additional improvement or hardware enhancement.

Future trends in Computer:-

Computer trend is drastically changing from historical period to present period and going on for future planning.

Present trend is not sufficient for scientific purpose, so scientists and programmer are just day to day research to obtain the following future trends in the computer:

- Bio Chips.
- Natural language processing.
- Artificial intelligence (AI).
- Voice recognition system.
- Vision recognition system.
- Cloud Computing.
- I/p Clouding
- Maximum utilization of server.
- Massive use of natural language processing etc.

Instruction, Instruction set & Instruction format

Commands given to a Computer system to perform specific task is called instruction.

Instruction set is a collection of all the instruction which can be executed by CPU. If we define all the instruction of computer we can ~~sg~~ say that we define instruction set. Instruction means machine instruction or Computer instruction.

Opcode	Reference to source operand	Reference to destination operand
← 4 bit →	← 6 bit →	← 6 bit →

Elements of instruction set:

(i) operation Code (opcode):

It specifies the operation to perform. The

operation is specified by binary code.
Example: ADD, SUB, MUL, DIV, etc.

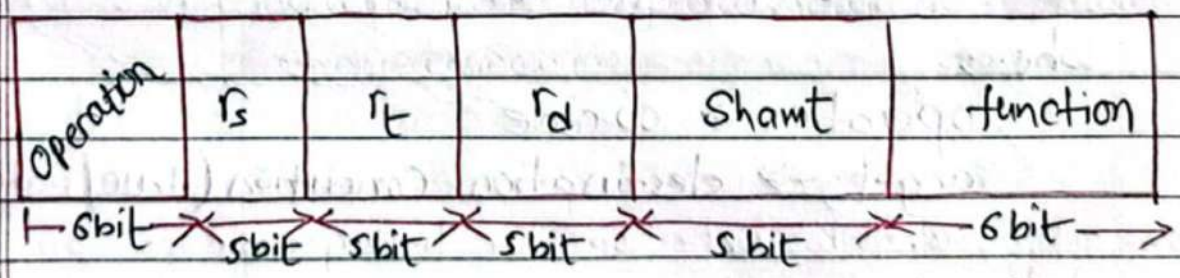
(ii) Source- operand Reference:-
It is a reference to operand on which data processing is performed.

(iii) Destination / Result operand Reference:-
A reference to operand which stores the result after data processing.

(iv) Next Instruction Reference:-
A reference to which Computers alert to fetc next instruction.

Instruction format in MIPS:-

1) R- type format (Register-type):-



where,
operation → opcode
r_s → first Source operand.
r_t → second Source operand.
r_d → result operand.
shamt → shift, function → ALU/ shift specifier

Page :
Date :

2> I - type format (Instruction - type) :-

Operation	r_s	r_t	Immediate
← 6 bit →	← 5 bit →	← 5 bit →	← 16 bit →

where,

operation → opcode
r_s → Source operand specifier.
r_t → result operand specifier.
Immediate → Immediate, branch or address displacements.

3> J- type format (Jump type)

Operation	Target
← 6 bit →	← 26 bit →

where,

operation → op code
Target → destination condition (True/False).

Imp
Addressing modes in MIPS :-

The address fields in a typical instruction format are relatively small. Using the values gives in the field are not able to reference a large number of location in main memory. To overcome this problem a variety of addressing technique can be used.

Page:
Date: / /

The process of specifying the operand is called addressing and multiple form of addressing is called addressing mode.

In general, addressing mode is a technique to address the operands in memory or register or instruction.

Suppose (x) is a content of memory location x

E_A = actual (effective) address which contain the reference operand.

A = Content of address field in instruction which refer. register.

Some common ~~register~~ addressing modes are as follows:

(i) immediate addressing mode:-

opcode	operand

instruction

Here E_A = operand = A

It is simplest addressing mode in which the operand is actually present in the instruction. This mode can be used to define and initialize value of variables.

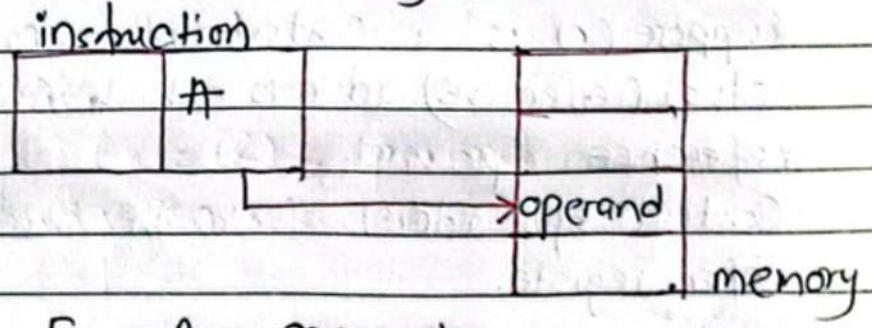
Advantages:

- It is simple and fast.
- No memory reference is required to obtain the address.

Page :
Date :

Disadvantages:
- The size of memory is restricted to the size of address field which is very small i.e limited operand magnitude.

(ii) Direct Addressing Mode:-



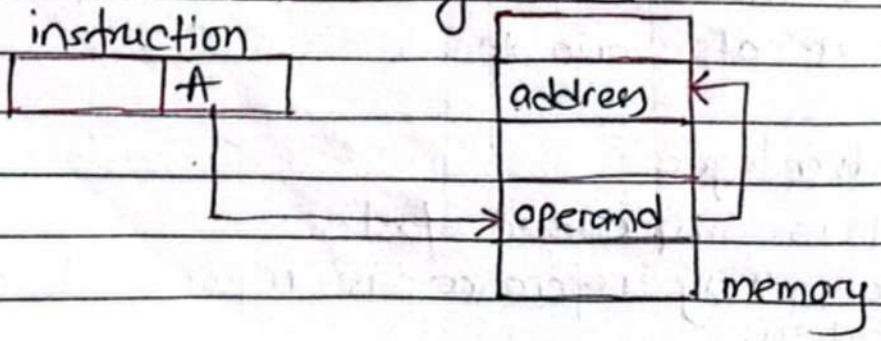
Here $E_A = A =$ operand.

In direct addressing mode the address field contains effective address of the operand. It is also very simple form of addressing.

Advantages:
- It require only one memory reference and simple.

Disadvantages:
- It provides only a limited address space.

(iii) Indirect Addressing mode:-



Page:
Date: / /

Here

$E_A = (A) =$ operand

In this addressing mode, the address field refers to the address of a word in memory which contains the full length address of the operand.

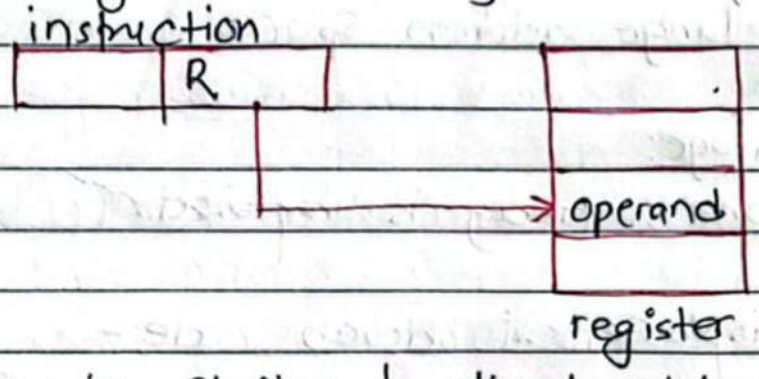
Advantages:

- It support large address space.

Disadvantages:

- Extra memory reference is required to fetch the operand.

(iv) Register Addressing mode:-



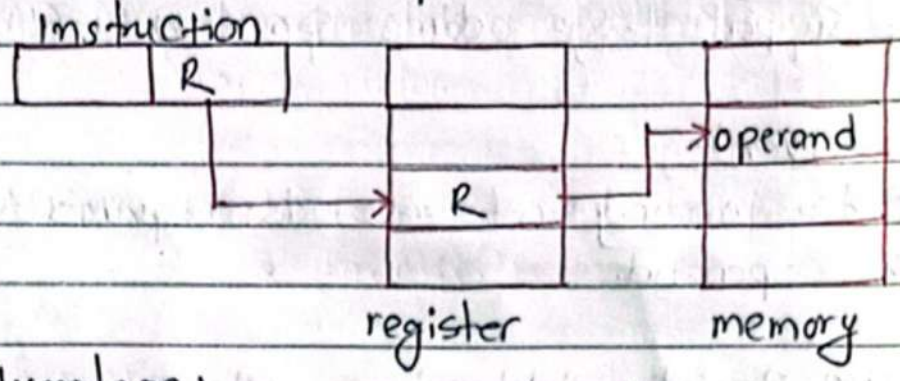
It is similar to direct addressing mode. The address field of the instruction refer to the register which has actual operand.

Advantages:-

- No memory reference is required. So it is very fast.

Disadvantages:
- It has limited address space.

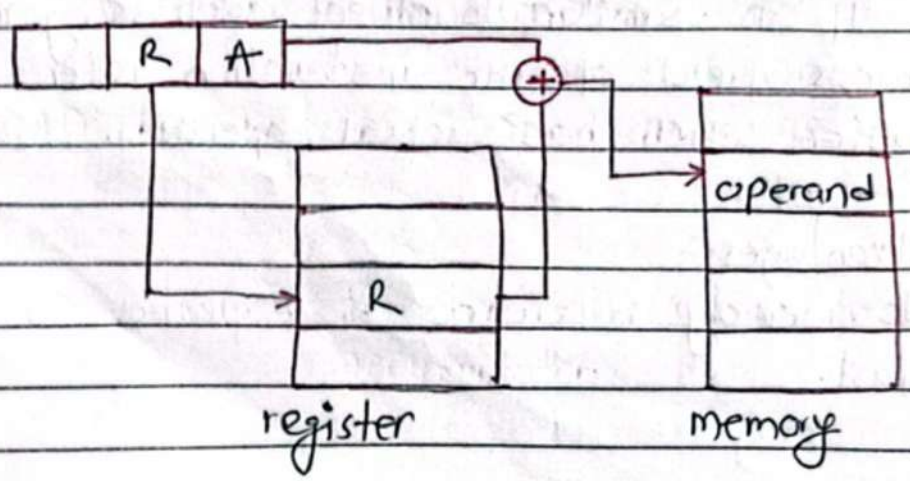
(v) Register Indirect addressing mode:-
In this addressing mode, the address fields of instruction refers to a register and this register has full length address of operand.
Here,
$E_A = (R) =$ operand



Advantage:
- It has large address space.

Disadvantage:
- Extra one memory is required.

(vi) Displacement address mode:-



Page:
Date: / /

Here, $E_A = (R) + A$ = operand

It is very powerful addressing mode that combines the capabilities of direct addressing and register indirect addressing modes.

It requires that the instruction has two address fields i.e the value contained in the address field is used directly called as 'A' and the other address field refers to a register whose address is indirect called as (R).

Advantages:
- It is more powerful addressing mode.
- It is flexible and very fast.
- It has large addressing space.

Disadvantages:
- It is more complex.

(vii) Stack addressing mode:-

The stack is linearly located memory in which we can reference only from the top of that linear memory.

It is also referred to as push down list or LIFO queue. It is a temporary addressing technique.

Page :
Date :

Imp # IAs Instruction Set:-

In IAs Computer instruction were developed as below:

(i) Data Transfer:-

It transfer data from one to another register as well as one memory to another. It is Categorized into following as mentioned below.

- LOAD MQ: transfer the Contents of register MQ to accumulator.

- LOAD MQ, M(X):- transfer Contents of memory location X to MQ.

- ~~LOAD MQ,~~
- STOR M(X):- transfer Contents of accumulator to memory location X.

- LOAD m(X): transfer value of m(X) to accumulator

- LOAD -m(X): transfer value of -m(X) to the accumulator.

- LOAD |m(X): transfer absolute value of m(X) to accumulator.

- LOAD -|m(X): transfer -|m(X)| to the accumulator.

CamScanner

Page:
Date: / /

(ii) Unconditional branch:-
It take next instruction from memory without Conditional branch. It is Categorized into following:

- JUMP M(X, 0 : 19): take next instruction from left half of m(x).

- JUMP M(X, 20 : 39): take next instruction from right half of m(x).

(iii) Conditional branch:-
It take next instruction from memory with certain conditional branch. It is categorized into following:

- JUMP +M(X, 0:19): If the no. in the accumulator is non-negative, it take next instruction from left half of m(x).

- JUMP +M(X, 20:39): If the number in accumulator is non-negative it take next instruction from right half of m(x).

(iv) Arithmetic:-
It is used to Calculate the arithmetic Calculation. It is Categorized into following:

- ADD m(x): Add value of m(x) to accumulator & Store result in accumulator.

CamScanner

Page :
Date : / /

- ADD |m(x)| : add value of |m(x)| to AC & put result in AC.

- SUB m(x) : Subtract m(x) from AC and put result in Ac

- SUB |m(x)| : Subtract |m(x)| from Ac and put result in AC

- MUL m(x): multiply m(x) by MQ & put MSB of result in AC and LSB in MQ.

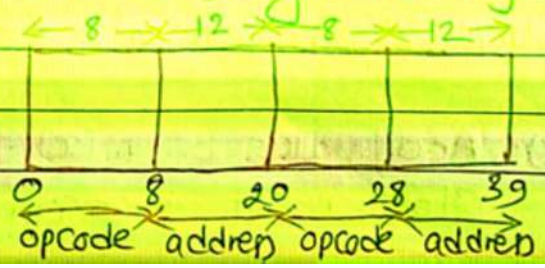
- Div m(x): Divide Ac by m(x) & put quotient in MQ & remainder in Ac

- RSH (Right shift): Divide Ac by 2 i.e shift right 1-bit position.

(V) Address modify:-
It replace the address field only in memory
It is Categorized as:

- STOR m(x, 8:19): Replace left address field at m(x) by 12 right most bit of Ac.

- STOR m(x, 28:39): Replace right address field at m(x) by 12 rightmost bit of Ac



Instruction format:-

An instruction format defines the layout of the bits of an instruction in terms of its Constituent fields.

It must include an opcode and implicitly or explicitly zero or more operands.

Each explicit operand is referenced using one of the addressing modes.

CamScanner

Chapter - 4

Register Transfer and Microoperation

Page :
Date :

Register Transfer (RT) and Register Transfer language (RTL):-

Register Transfer language (RTL):-

A digital system is a discrete system having interconnection of digital hardware modules that accomplish a specific information processing task.

Digital system vary in size and complexity from a few integrated circuit to a digital computer. The modules are constructed from such digital components as control logic, arithmetic elements, registers and decoders, etc.

The symbolic notation that is used to describe the micro ~~operating~~ operation transfer among registers is called register transfer (RT).

The term register transfer implies the availability of hardware logic circuit that can perform specific micro operation and transfer the result of operating to same or another register.

A programming language is used to execute process. It is called register transfer programming language.

Register Transfer (RT):-

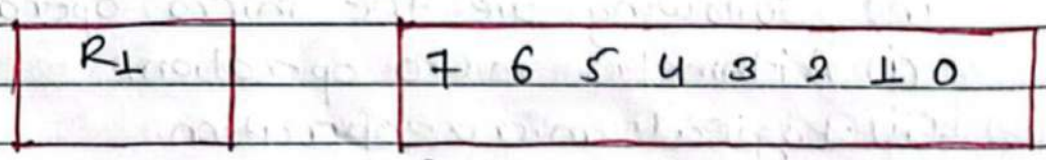
Computer registers are designated by capital letter to denote the function of the register.

The register that holds an address for

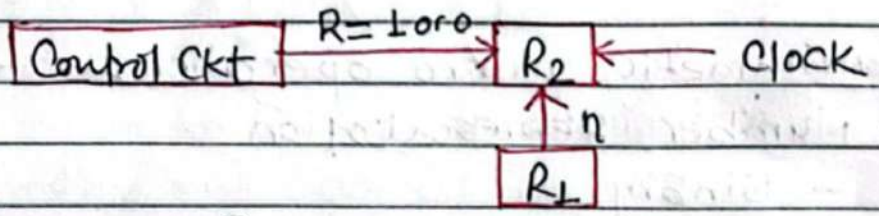
the memory unit is usually called memory address register (MAR) and other designated register program counter (PC).

The registers are numbered from 0 to (n-1) starting from 0 in the rightmost position and increasing the numbers towards the left.

The representation of register as $R_2 \leftarrow R_1$ it denotes transfer of contents of register R_1 to register R_2. It means that the replacement of content of R_2 by the contents of R_1.



fig(i) Register R_1 fig(ii) showing individual bits.



fig(iii) transfer content of R_1 to R_2 (Block Diagram of content transfer of R_1 to R_2).

Here p is a logical control function as well as Boolean function that gives true or false condition.

True denotes $p=1$ and false denotes $p=0$

Now if $p=1$ then control circuit are available from source register to destination register upto (n-1) with time (t+1)

Page :
Date : / /

Micro operations

A digital system is defined by digital module that are referred to as registers that contain binary data stored. The operations on the data and instructions executed on data stored in registers. These operations are known as micro operations.

The elementary operations performed on one or more registers.

The following are the micro operations:

(i) Arithmetic micro operation.
(ii) Logical micro operation.
(iii) Shift micro operation.
(iv) Data transfers micro operations.

(i) Arithmetic micro operations:-

a) Number Representation
- Binary.
- Decimal.

$(-18)_{10} = (10010010)_2$
$(+0)_{10} = (00000000)_2$
$(-0)_{10} = (10000000)_2$

Page:
Date: / /

c) 1's Complement:

$(+18)_{10} = (00010010)_2$

$(-18)_{10} = (11101101)_2$

$(+0)_{10} = (00000000)_2$

$(-0)_{10} = (11111111)_2$

d) 2's Complement

2's Complement is 1's Complement + 1

$(+18)_{10} = (00010010)_2$

$(-18)_{10} = 11101101 + 1 = (11101110)_2$

$(+0)_{10} = (00000000)_2$

$(-0)_{10} = (11111111 + 1) = (100000000)_2$

$= 00000000$

ignore Carry

short cut method for 2's Complement

$(+18)_{10} = (00010010)_2$

$(-18)_{10} = (11101110)_2$

$(+0)_{10} = (00000000)_2$

$(-0)_{10} = (00000000)_2$

Advantage of 2's Complement representation:

(i) All the positive number have zero in the

representation and 1's Complement

Example of arithmetic operation using Reg transfer language (RTL)

1) $R_2 \leftarrow R_1$
2) $R_3 \leftarrow R_2 + R_1$
3) $R_3 \leftarrow R_1 - R_2$
4) $R_3 \leftarrow R_1 * R_2$
5) $R_3 \leftarrow R_1 \div R_2$
6) $R_2 \leftarrow R_1 + 1$
7) $R_2 \leftarrow R_1 - 1$

(ii) Logical micro operation:-

a) OR operation

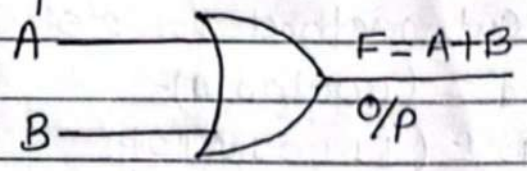


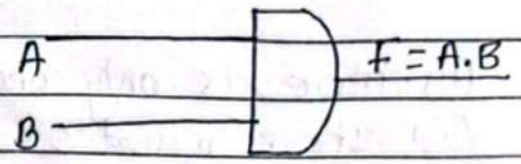
fig: Logical OR operation.

Input		output
A	B	F = A+B
0	0	0
0	1	1
1	0	1
1	1	1

where,
0 → off state
1 → on state

fig: Truth Table

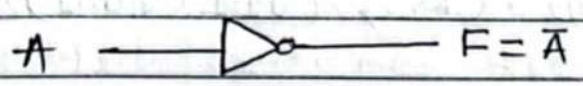
b) AND operation:



Input		output
A	B	$F = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

fig: Truth table for AND operation.

c) NOT operation:



Input (A)	output ($F = \bar{A}$)
0	1
1	0

fig: truth table for NOT operation.

(iii) shift microoperation:-

a) Left shift

$(110100 10)_2 \xrightarrow{LS} (10100100)_2$

b) Right shift

$(110100 10)_2 \xrightarrow{R.S} (0110100 1)_2$

c) Arithmetic left shift

$(110100 10)_2 \xrightarrow{A.L.S} (10100100)_2$

$(110100 11)_2 \xrightarrow{A.L.S} (10100111)_2$

d) Arithmetic Right shift

$(11010010)_2 \xrightarrow{ARS} (11101001)_2$

$(11010011)_2 \xrightarrow{ARS} (11101001)_2$

e) Circular left shift

$(11010010)_2 \xrightarrow{CLS} (10100101)_2$

f) Circular Right shift

$(11010010)_2 \xrightarrow{CRS} (01101001)_2$

(iv) Data transfer micro operations.

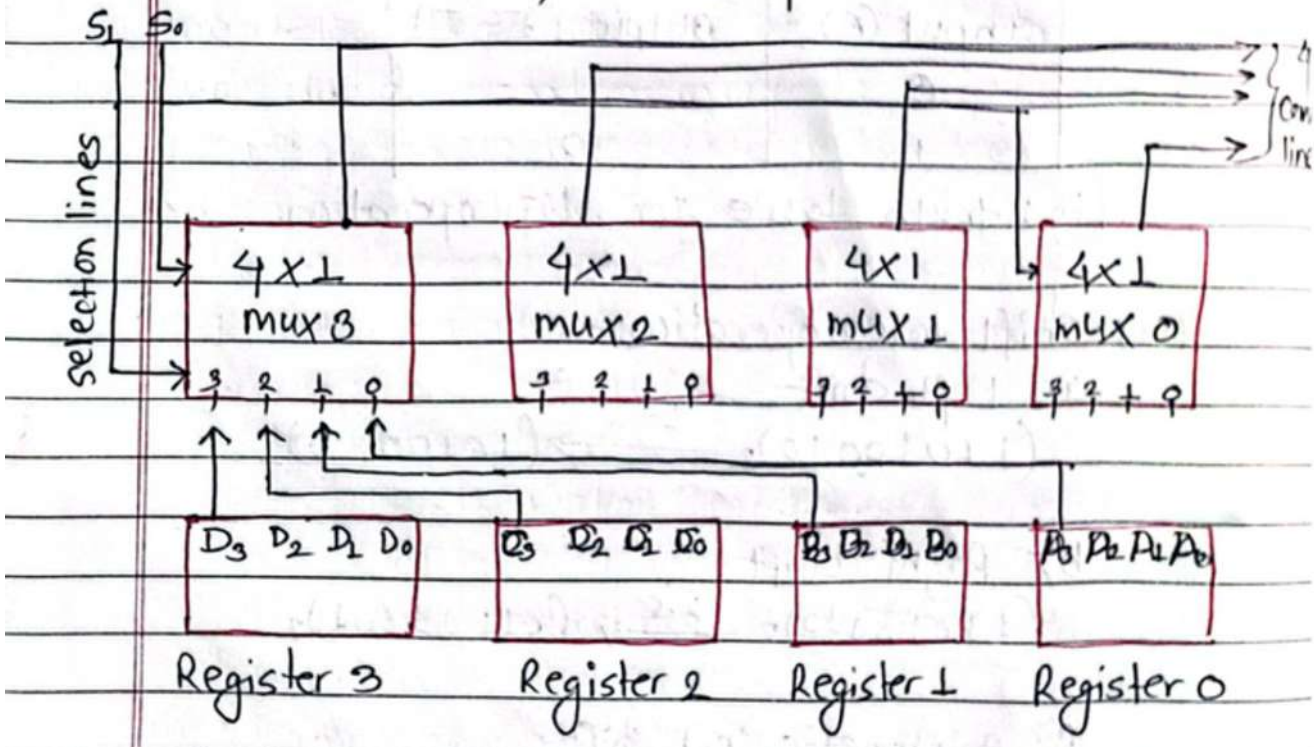


fig: Data transfer by using 4x1 mux with 4-register.

A typical digital Computer has many registers and data bus that must be provided to

transfer information from one register to another register.

A more efficient scheme for transferring information between registers in a multiple register configuration is known as common data bus system.

A bus structure consist of a set of a common line through which binary information is transferred one at a time.

Control signals determine which register is selected by the bus during each particular register transfer.

One way of constructing a common data bus system with multiplexer to transfer data from register to register.

The mux selects the source register whose binary information is placed on the bus. The construction of a data bus system of four register is shown in the above figure.

The register can be selected by using selection lines as:

S_0	S_1	Register
0	0	Register 0 (A)
0	1	Register 1 (B)
1	0	Register 2 (C)
1	1	Register 3 (D)

Page :
Date : /

Memory Transfer:-

The transfer of information or data from a memory to the outside environment or register and vice-versa is known as memory transfer.

The transfer of information from a memory word to the outside register is called READ operation.

The transfer of information to be stored into the memory from outside register is called WRITE operation.

A memory word will be symbolized by the letter 'M'. If we want to transfer the data from memory to register or register or to memory we can state the specific address.

(i) Read operation will be done as

Read : $DR \leftarrow M[AR]$

where,

DR = Data Register.

M = Memory word.

AR = Address Register.

(ii) WRITE operation will be done as:

Write :

$M[AR] \leftarrow R_1, R_2, R_3 \ldots R_n$

where,

$R_1, R_2, R_3 \ldots$ = no. of registers.

CamScanner

Introduction to HDL and VHDL:-

HDL is a Hardware description language that means it is a specialized Computer language used to program the structure, design and operation of electronic circuit and mostly called as digital logic circuit.

HDL enables an electronic circuit that allows for the automated analysis simulation and simulated testing of an integrated electronic circuit.

HDL specially defines the logical program hardware interface program with the parallel processing on the digital logic circuit.

HDL forms an integral part of electronic design automation system especially for complex circuit such as microprocessor.

VHDL is a VHSIC HDL that stands for very high speed integrated circuit hardware Description language used in electronic design automation to describe digital and mixed signal system such as logical gates and integrated circuit programming language.

VHDL can also be used as a general purpose parallel programming language.

Structure of HDL:-

(i) Architecture or Class libary.
(ii) Body.
(iii) Result / end.

Chapter III

Central processing Unit

Organization/structure of CPU [processor organ

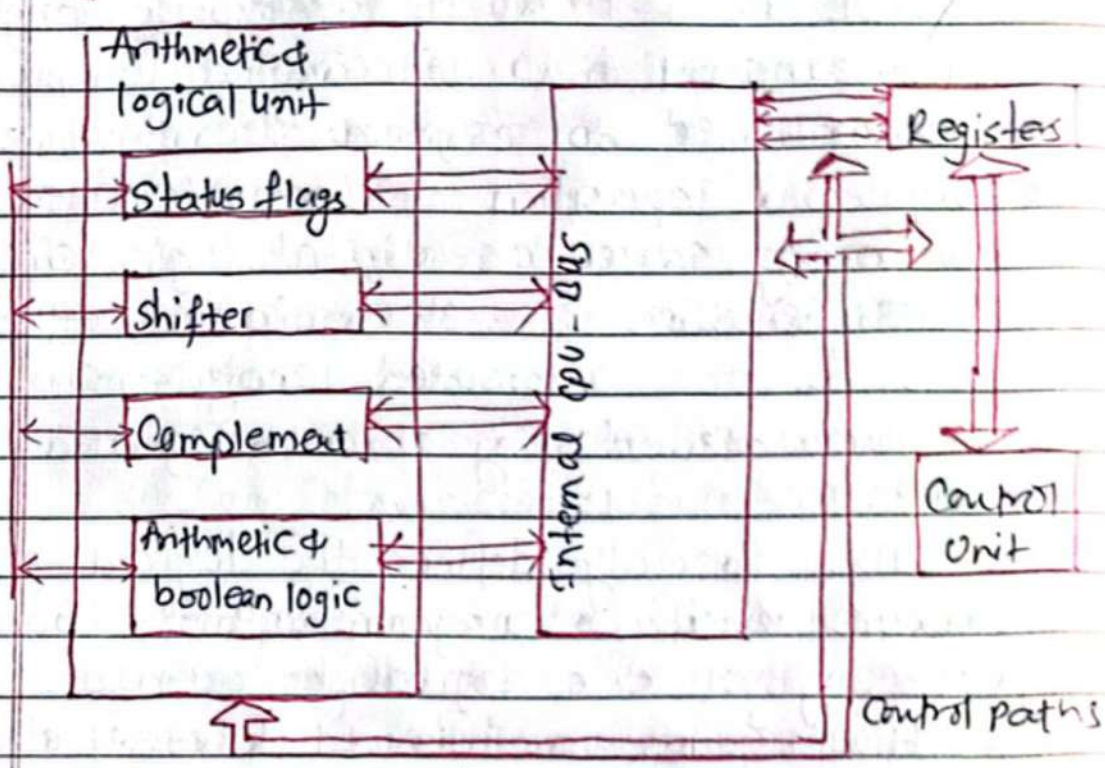


fig: internal structure of CPU.

To know the organization of processor we have to understand how the instruction is executed to perform the specific task.

Following are the general instruction executio steps:

(i) Fetch Instruction (FI):-

In this stage the CPU reads instruction from the register or memory.

(ii) Decode Instruction (DI):-

Control unit decodes the instruction to know what operation to do. It is also known as Interpret instruction (II)

Page:
Date: / /

(iii) Fetch Data (FD):-
Before executing an instruction the operand data is required for an instruction to be fetched.

(iv) process Data (PD):-
After fetching instruction and data, the interpret instruction are executed to give result.

(v) Write Result (RL):-
The result after executing of instruction may be written into register, memory and input/output devices. To Complete above operation the components which are used. These Components may be register, control unit, processing elements, control path, etc.

The data from one register to another register are flown from data path. The control unit Controls all the elements of processor from Control path as shown in above figure. The shifter performs shift operation Complement perform shift operation Complement performs 2's Complement operation Status flag Contains the single bit status of a result and arithmetic and boolean logic performs arithmetic and boolean operation.

Registers use CPU internal memory which stores the small unit of data.

CamScanner

Page :
Date :

Register organization OR structure of Register within CPU:-

Within the CPU there is a set of register that works as a internal memory of a CPU. The number of register set in different Computers may be different. All these registers can be categorized into follows as mentioned below:

- User/programmable visible Register.
- Status and Control Register

A) User/programmable visible Register:-

These register can be used by machine or assembly level language program to minimize main memory reference.

In general there are four types of user visible register.

They are:-

(i) General purpose Register:-

These are used for various functions desired by the processor. It may contain either an operand or calculation of operands and even to store some data.

(ii) Data Register:-

These are used only for storing intermediate result or data or calculation of intermediate operands. These data register are not used for calculation of address of any operand.

CamScanner

Page:
Date: / /

(iii) Address Register :-
It is also a general purpose register but in some cases few dedicated address registers are used as:

- Segment pointer:
It is used to point out segment of memory

- Index Register:
It is used for index addressing scheme

- Stack pointer:-
It is used for stack addressing to point out ~~and~~ top of stack for push and pop operation.

(iv) Condition Code Register:-
It is also known as flag which is set by Cpu as result of operation.

Example: An ~~art~~ arithmetic operation may produce an overflow, zero, negative, positive result, etc.

B) Status and Control Register:-
It is a very important register to control the various operations as well as overall register used.

These registers cannot be used for data manipulation. However these registers can also be used to execute following

Page :
Date :

instruction:
(i) PC (iii) MAR
(ii) IR (iv) MBR

(i) program Counter (PC):-
It ~~consist~~ counts the number of instructio executed.

(ii) Instruction Register (IR):-
It Controls the instruction within registes.

(iii) Memory Address Register (MAR):-
It is used to control the data and instructions either in memory or register by providing address location.

(iv) Memory Buffer Register (MBR):-
It is used for memory buffering.

Data path:-

Data path refers to the Component of the processor that performs arithmetic operation.
A functional unit used to operate or hold data within a processor is called data path element.
The data path element includes the instruction data, register file, ALU and adder.
Control refers to the Component of the processor that Commands the data path, memory and IO devices according to the

instruction of the program.

The data path can be categorized into following as mentioned below:

(i) Data path for R-type.
(ii) Data path for I-type.
(iii) Data path for J-type.

(i) Data path for R-type:-
(Data path for arithmetic & logical operations)

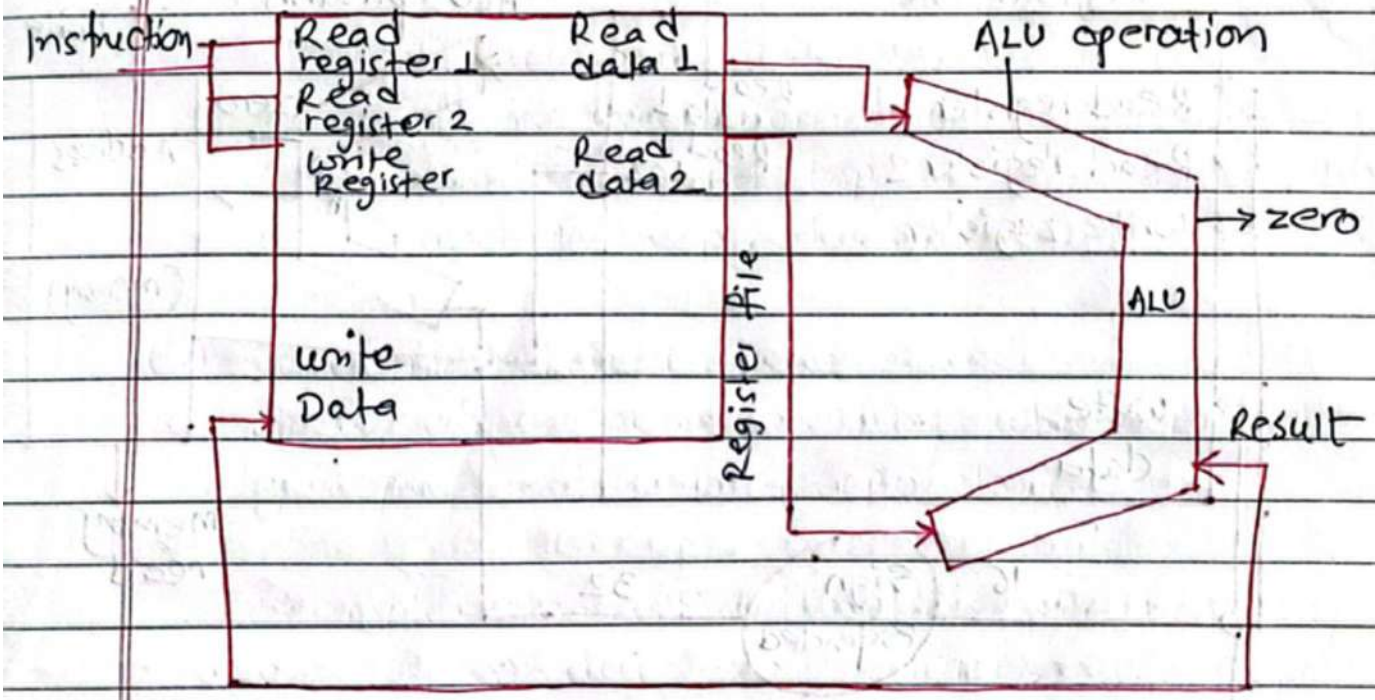


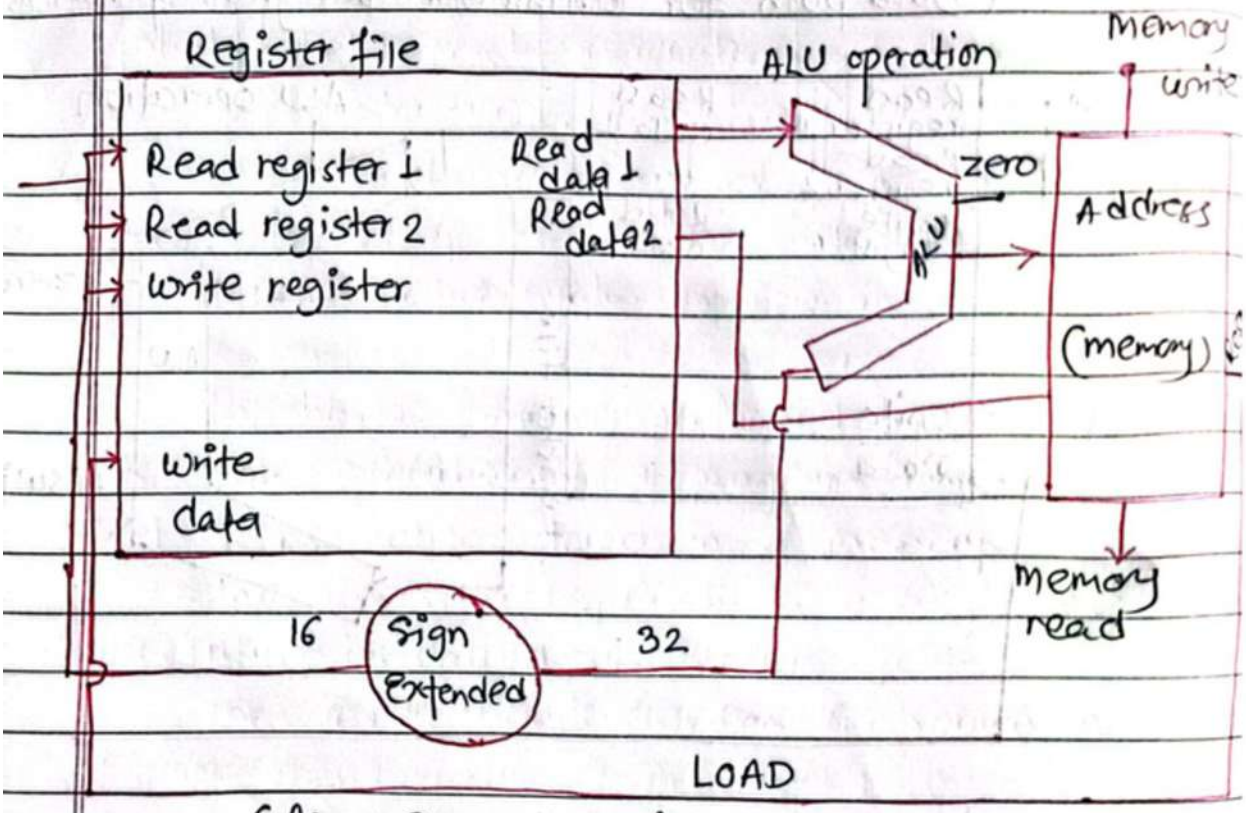
fig:- Data path for R-type.

The R-type instruction reads two registers, programs and ALU operations on the content of register and write the result.

These instructions are also known as arithmetic or logical operation instruction. The 32-bit register are stored in a structure

Called a register file because the R-type format instruction have three register operands and we need to read them and write into the register file for each instruction.

(ii) Data path for I-type :-
(Data path for memory instruction)



(fig: Data path for I-type)

The ~~load~~ LOAD and STORE instruction also need the same register file as in the R-type instruction format.

In addition we need a sign extended unit which takes 16 bit input i.e sign extended into a ~~bit~~ 32 bit result appearing

on the output. The memory unit takes input for the address and write data and has single output for the read result.

There are separate read and write controls but only one of those may be active on any given clock.

(iii) Data path for J-type:-
[Data path for Branch instruction]

The branch data path must do two operations as following
(i) Compute the branch target address.
(ii) Compare the register content.

To Compute the branch target address, the branch data path includes a sign extension unit and address adder.

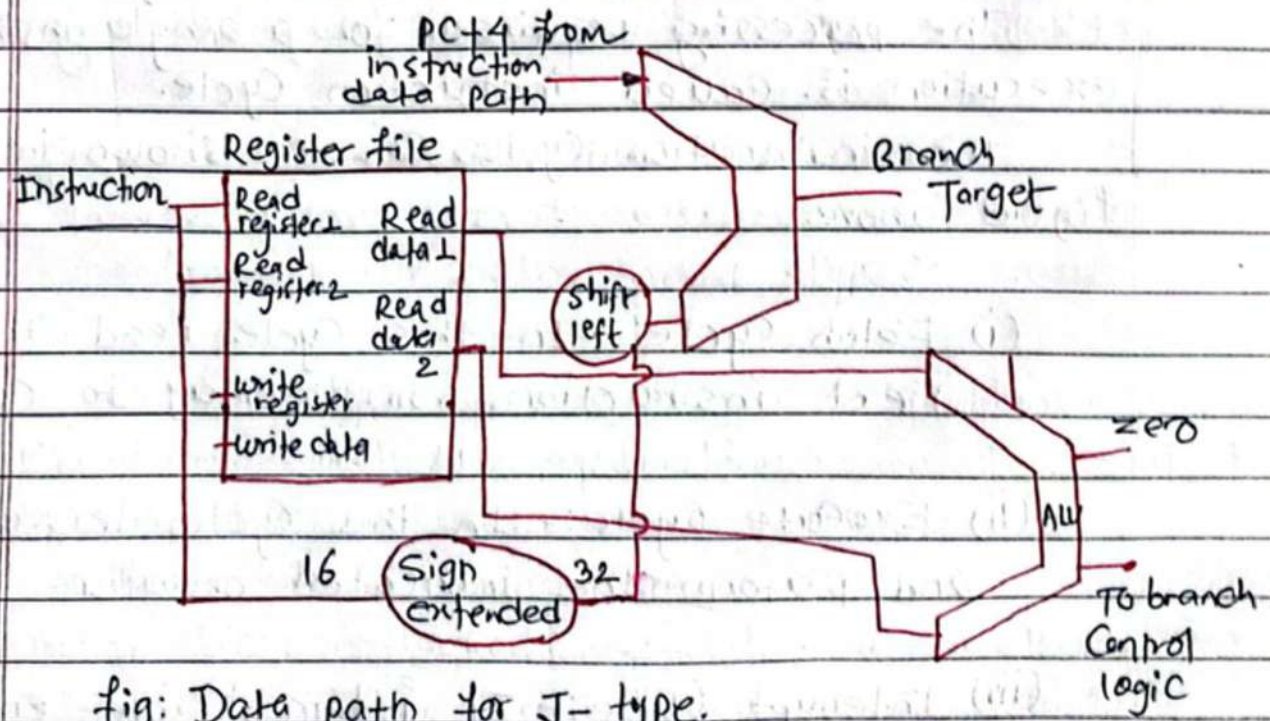


fig: Data path for J-type.

To perform the compare we need to use register file to supply two register operands. The jump instruction operates by replacing lower bit of program counter (pc) with 9 bit of the instruction shifted left by 2 bits.

Instruction Cycle:-

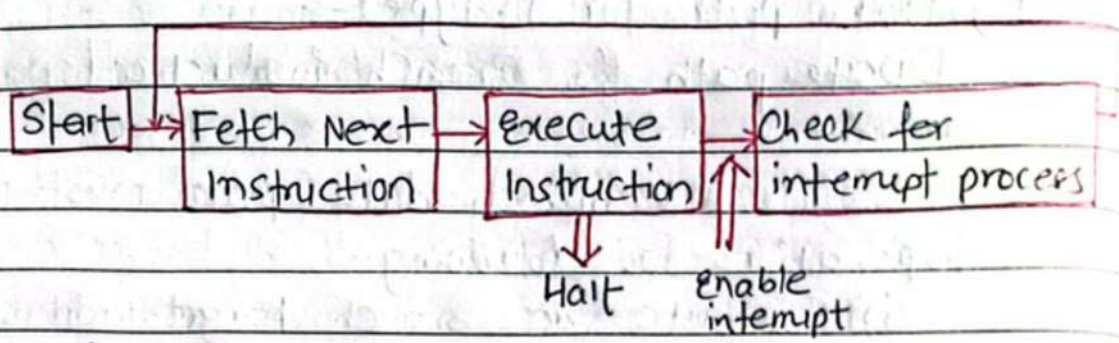


fig:- Instruction Cycle.

The basic function performed by the computer to each execution of a program which consist of set of instruction stored in a memory.

The processing required for a single instruction execution is called instruction cycle.

The instruction cycle can be shown in the figure above.

(i) Fetch Cycle:- In this cycle read the next instruction from memory to CPU.

(ii) Execute Cycle:- In this cycle decode opcode and perform the indicated operation.

(iii) Interrupt Cycle:- If interrupt are enable

or occured then set the current process and service the interrupt.

Execution of an instruction may involve one or more operand in memory each of which requires a memory.

If indirect addressing is used then additional memory access are required so, Fetch of indirect address must be considered as extra sub cycle in instruction execution as shown below:

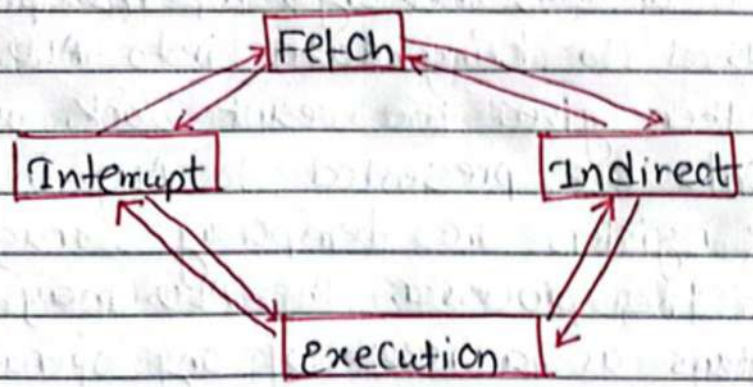


fig: Instruction Cycle using indirect addressing

State Diagram of Instruction Cycle:-

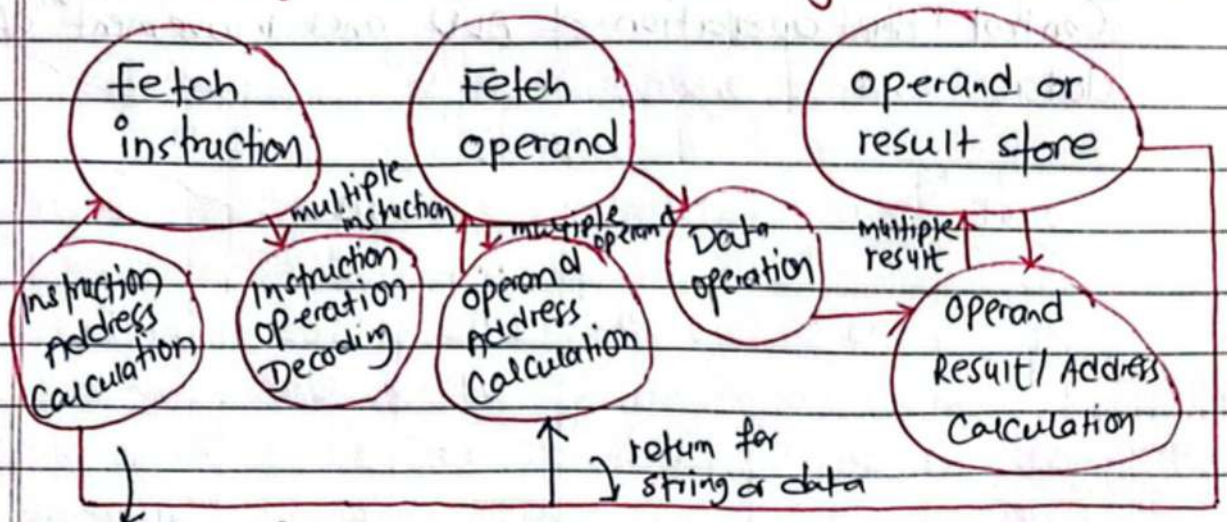


fig: Instruction cycle state Diagram

Arithmetic And Logical Unit:-

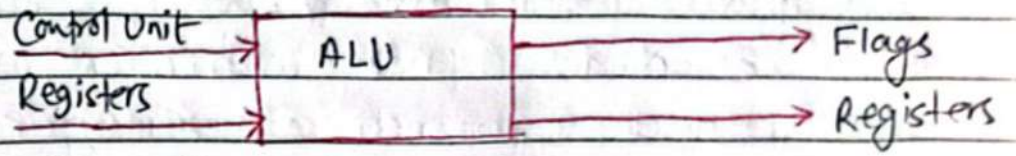


fig: ALU

The ~~alu~~ ALU is part of Computer that actually perform arithmetic and logical operation in stored data and instruction.

All of the other Components of Computer such as Control unit, memory, input/output, registers are used to bring data into ALU for processing and then gives the result back out.

Data are presented to ALU in a register. These registers are temporary storage location within the processor. The ALU may be also set off flags as a result of any operation.

Example: In addition, result may be Carry flag or not.

The Control unit provides control signals to Control the operation of ALU and movement of data.

Chapter IV

Computer Arithmetic

Multiplication And Division:

1) Multiplication:-

A) Unsigned multiplication:

Flowchart:

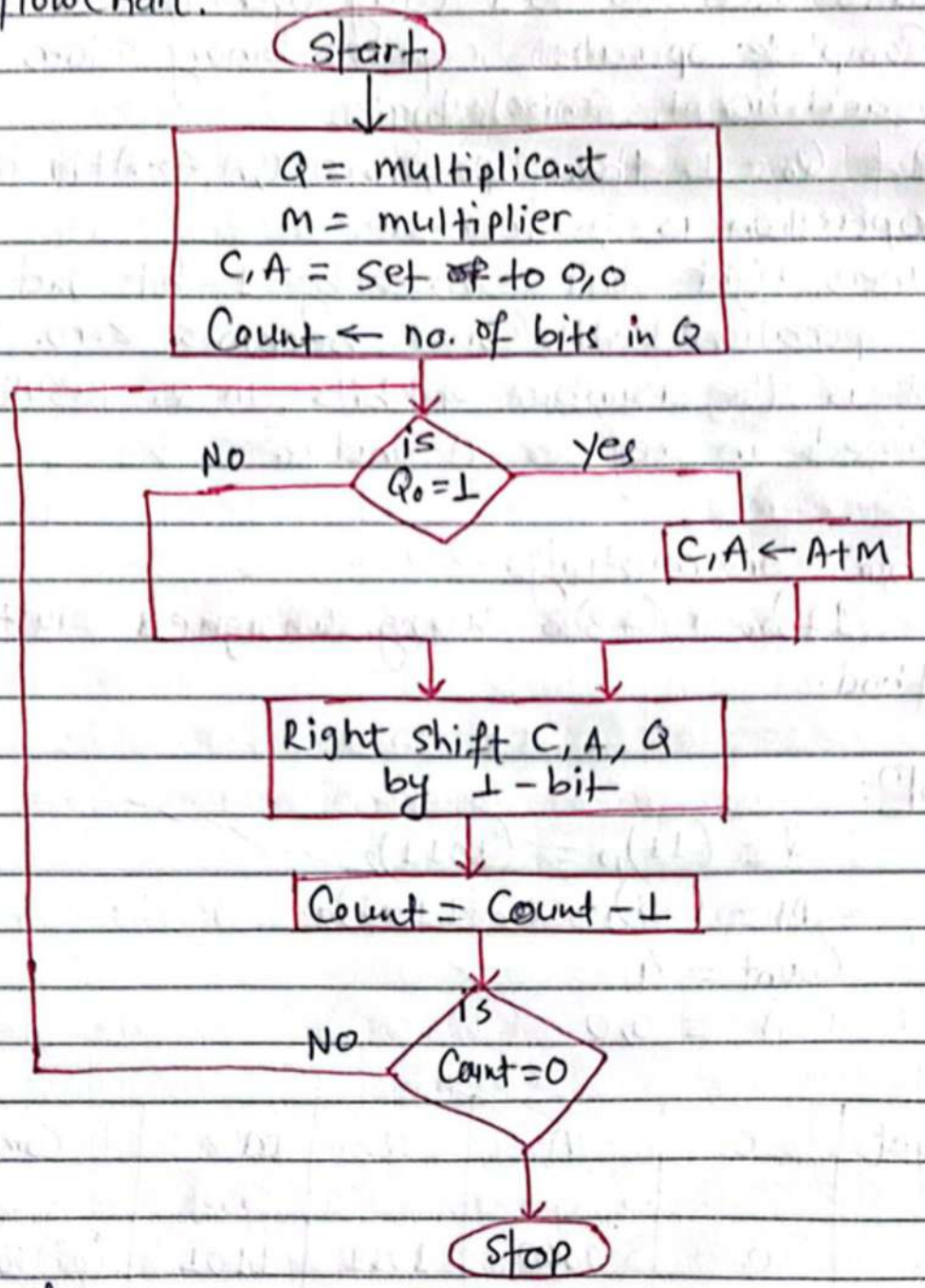


fig:- Unsigned multiplication.

The multiplicant and multiplier are loaded in two registers Q and M respectively.

A third register A which is initially set to zero (0) is also needed. A 1-bit Carry register C is also used to hold Carry over if exits.

Complete operation requires binary adder, shifter and additional control logic.

If $Q_0 = 1$ then peforms $C,A \leftarrow A+M$ otherwise no operation is performed.

Then right shift C,A, Q by 1-bit and repeat this operation until Count becomes zero. Here, Count is the number of bits in Q. Result is Contained in register A and Q.

Q. Perform or multiply:
$(11)_{10} * (13)_{10}$ using unsigned multiplication method.

→ Soln:

$Q = (11)_{10} = (1011)_2$
$M = (13)_{10} = (1101)_2$
Count = 4
C,A = 0,0

Count	C	A	Q	M	Comment
4	0	0000	1011	1101	initialize $C,A \leftarrow A+M$
4	0	1101	1011	1101	Right shift C,A,Q Count = Count − 1

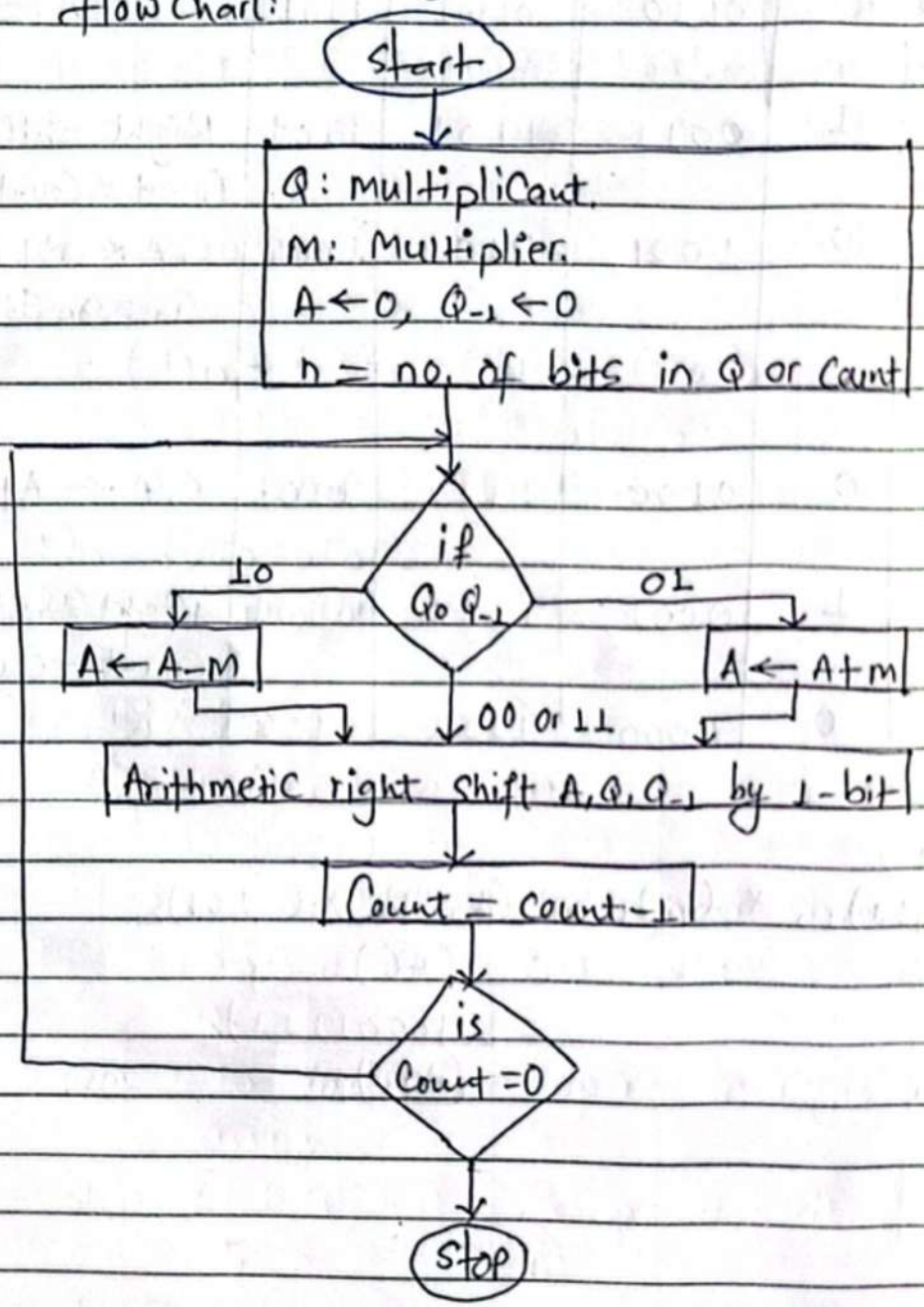
Count	C	A	Q	M	Comment
3	0	0110	01101	1101	C, A ← A+M
3	1	0011	1101	1101	Right shift C,A,Q Count = Count − 1
2	0	1001	1110	1101	C, A ← A+M $Q_0 = 0$, S_0
1	0	0100	1111	1101	C, A ← A+M
1	1	0001	1111	1101	Right shift C, A, Count = Count − 1
0	0	1000	1111	1101	–

Now,

$$(11)_{10} * (13)_{10} = (1011)_2 * (1101)_2$$
$$= (AQ)$$
$$= (10001111)_2$$
$$= (143)_{10}$$ ans:

B) Signed multiplication:-
[Booth's Algorithm]

Flow Chart:



In Case of arithmetic right shift sign bit is not Changed and shift that sign bit to other position.

In this algorithm all signed numbers are represented in 2's Complement from and positive numbers are represented only in binary form.

The register A which is used to store the part of result whose value is also used in the operation A 1-bit register Q_{-1} is also used and initialize to zero.

If $Q_0 Q_{-1} = 01$ i.e $Q=0, Q_{-1}=1$ then it performs $A \leftarrow A+M$. If $Q_0 Q_{-1} = 10$ then $A \leftarrow A-M$ otherwise no operation i.e 00 or 11.

Now, perform the arithmetic right shift A, Q, Q_{-1} by 1 bit and decrease the count by 1. This process is repeated until Count becomes zero (0). The result of operation is stored in A and Q i.e AQ.

Q. Perform $(7)_{10} * (-3)_{10}$ by using Booth's algorithm.

→ Soln:

$Q = (7)_{10} = (0111)_2$ 2's Complement of M

$M = (-3)_{10} = (1101)_2$ $-M = (3)_{10} = 0011$

$n = \text{Count} \leftarrow 4$

$Q_{-1} = 0$, $A \leftarrow 0000$

Count	A	Q	Q_{-1}	M	Comment
4	0000	0111	0	1101	initialize $A \leftarrow A-M$
4	0011	0111	0	1101	A.R.S A, Q, Q_{-1} Count = Count − 1
3	0001	1011	1	1101	RS A, Q, Q_{-1} Count = Count − 1
2	0000	1101	1	1101	A.R.S A, Q, Q_{-1} Count = Count − 1
1	0000	0110	1	1101	$A \leftarrow A+M$

Count	A	Q	Q_{-1}	M	Comment
1	1101	0110	1	1101	ARS A, Q, Q_{-1} Count = Count - 1
0	1110	1011	0	1101	

A = 1110
Q = 1011

Hence,

$$(7)_{10} * (-3)_{10} = (0111)_2 * (1101)_2$$
$$= (11101011)_2$$
$$= (-21)_{10}$$

$(21)_{10} = (00010101)_2$
$(-21)_{10} = (11101011)_2$

2) Division:-

A) Unsigned Division:-

Algorithm:

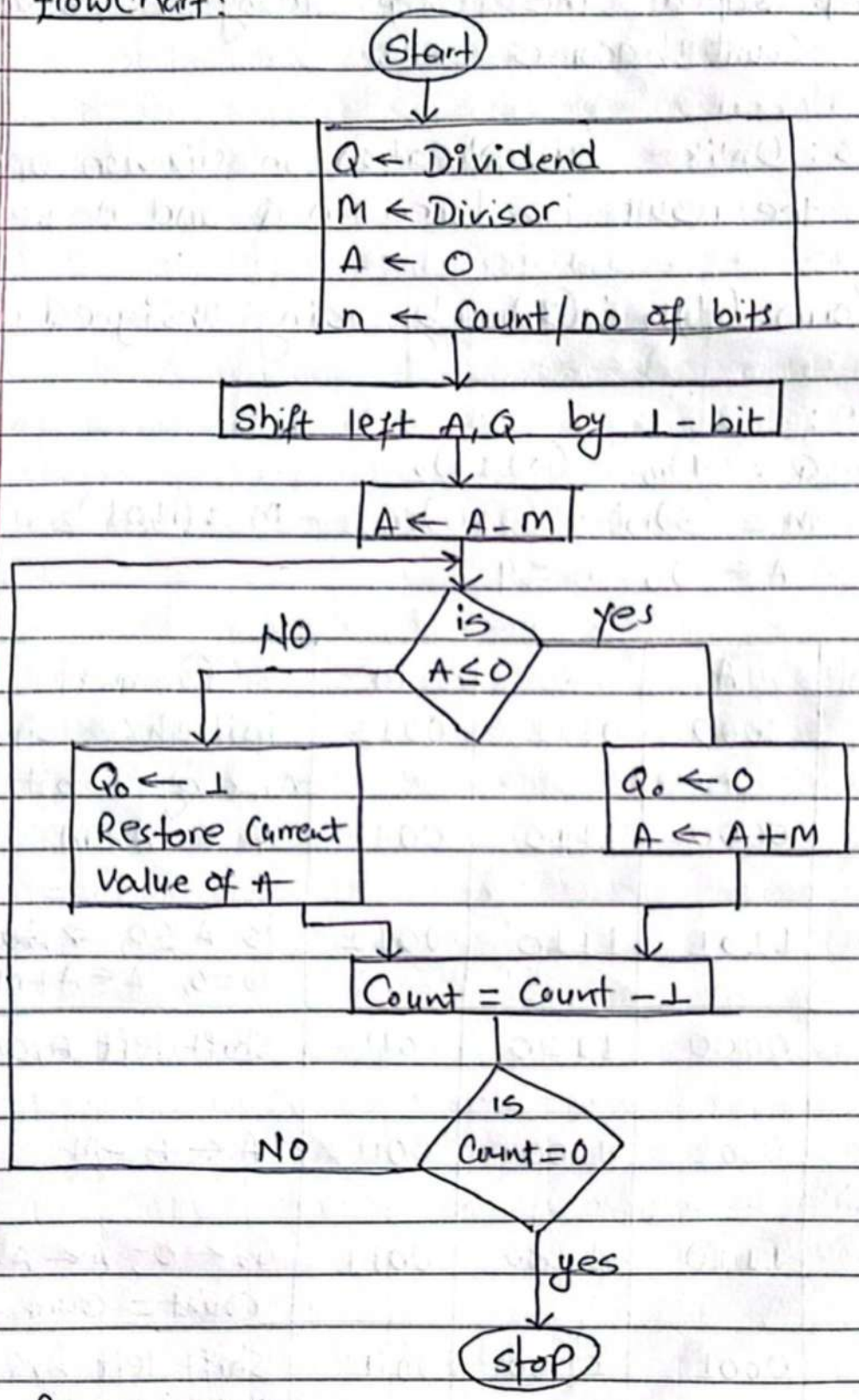
Step 1: To perform unsigned division the divisor is placed in M & dividend in Q.
To initialize operation register 'A' assumed as 0

Step 2: shift left A, Q by 1-bit position and then A ← A − M

Step 3: Check the value of A where if condition is satisfied then perform $Q_0 \leftarrow 1$ and restore current value of A.

Page:
Date: / /

flowchart:



Result = Q.A

otherwise perform $Q_0 \leftarrow 0$ and $A \leftarrow A + M$

Step 4: Repeat these steps many times until Count becomes zero.

Step 5: Unlike multiplication in division operation the result is stored in Q and A i.e Q A

Q. perform $(7)_{10} \div (3)_{10}$ by using unsigned division

→ Soln:

$Q = (7)_{10} = (0111)_2$

$M = (3)_{10} = (0011)_2$, $-M = (1101)_2$

$A \leftarrow 0$, $n = 4$

Count	A	Q	M	Comment
4	0000	0111	0011	initialize shift left A, Q by 1-bit
4	0000	1110	0011	$A \leftarrow A - M$
4	1101	1110	0011	is $A \leq 0 \Rightarrow$ ~~no~~ yes $Q_0 = 0$, $A \leftarrow A + M$, Count
3	0000	1110	0011	shift left A, Q by 1 bit
3	0001	1100	0011	$A \leftarrow A - M$
3	1110	1100	0011	$Q_0 \leftarrow 0$, $A \leftarrow A + M$ Count = Count − 1
2	0001	1100	0011	shift left A, Q by 1 bit
2	0011	1000	0011	$A \leftarrow A - M$

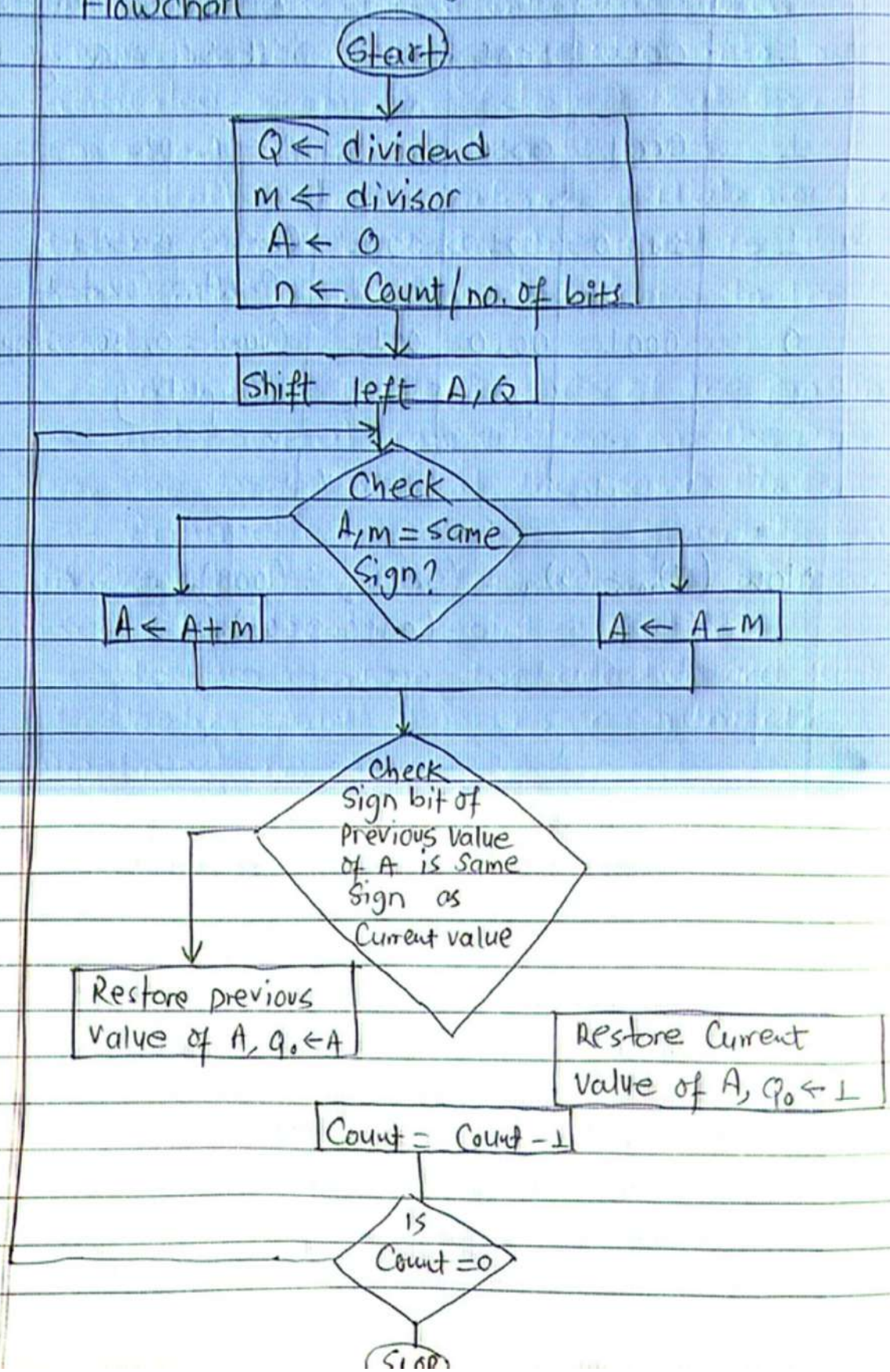
Count	A	Q	M	Comment
2	0000	1000	0011	$Q_0 \leftarrow 1$, restore content of A, Count = Count − 1
1	0000	1001	0011	shift left A,Q by 1 bit
1	0001	0010	0011	$A \leftarrow A - M$
1	1110	0010	0011	$Q_0 \leftarrow 0$, $A \leftarrow A + M$ Count = Count − 1
0	0001	0010	0011	Count = 0, so stop Computing

Now $(7)_{10} \div (3)_{10} = (0111)_2 \div (0011)_2 = Q.A$
$= (0010.0001)_2$

Page :
Date : / /

B> Signed Division :-
[Booth's Algorithm]

Flowchart



Notes for Numerical

Example	value of A	flowchart used
$(7)_{10} * (3)_{10}$	A = 0000	Unsigned multiplication
$(-7)_{10} * (3)_{10}$	A = 1111	Signed multiplication
$(-7)_{10} * (-3)_{10}$	A = 1111	Unsigned multiplication
$(7)_{10} \div (3)_{10}$	A = 0000	Unsigned division
$(7)_{10} \div (-3)_{10}$	A = 0000	Signed ~~de~~division
$(-7)_{10} \div (3)_{10}$	A = 1111	Signed division
$(-7)_{10} \div (-3)_{10}$	A = 1111	Unsigned division

1) In multiplication the result must be stored in (AQ) and in division the restored result must be stored in (Q.A)

2) If according to the question sign bit occurs then no change i.e 0 represent +ve and 1 represent -ve. otherwise must take 2's Complement of AQ in multiplication & Q in division.

Page :
Date :

Arithmetic pipelining :-

Pipelining is a technique of decomposing of sequential process into sub-process with each subprocess being executed into special dedicated segments (stages) that operates concurrently with all other segments.

Each stage performs partial processing of the complete task. The result is obtained from each segment is transfered to the final result.

The main characteristics of the pipelining is that several computations in the pipeline can be progressed at the same time.

Arithmetic pipelining is generally used in very high speed computers. It is mainly used for floating point operations.

Let us explain the addition and subtraction of floating point numbers in arithmetic pipelining as:

Suppose,

$$X = A * 2^a$$

$$Y = B * 2^b$$

Where A and B are mantissa

a and b are exponents.

The floating point addition and subtraction can be performed in following sequence.

(i) Compare exponents :- Subtract the

Page:
Date: / /

exponent and find the larger exponent for the resultant exponent.

(ii) Align mantissa :-
Shift the mantissa to right by the no. of digit position given by exponent difference.

(iii) Add or subtract mantissa

(iv) Normalize the result.

Example:
let $X = 0.9504 \times 10^3$ & $Y = 0.82 \times 10^2$

Soln → Difference of exponents $= 3 - 2 = 1$
larger 3

→ Align mantissa

$X = 0.9504 \times 10^3$
$Y = 0.082 \times 10^3$

make both of large degree i.e $\times 10^3$

→ Add or subtract

$X - Y = 0.8684 \times 10^3$
$X + Y = 1.0324 \times 10^3$

→ Normalize the result

$\boxed{X + Y = 0.10324 \times 10^4}$

$\boxed{X - Y = 0.8684 \times 10^4}$

CamScanner

Page :
Date :

BCD Adder (Binary Coded Decimal Adder)

A BCD Adder is a circuit that adds two BCD digits in parallel and produce zero sum in BCD.

A BCD adder must include the correction logic in its internal structure for decimal number greater than 9 we add 6(0110) to the binary sum to convert correct BCD number.

Derivation of BCD Adder:-

K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	Decimal
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

CamScanner

Page:
Date: / /

Here,

$$C = K + Z_8 Z_4 + Z_8 Z_2$$

The correction logic is derived from above table.

A correction is needed when the binary sum has output carry $K = 1$. The other six combination from 1010 to 1111 that need a correction in position Z_8.

To distinguish them from binary 1000 and 1001 i.e decimal 9 which also have 1 in position K.

we further specify by using following formula to calculate C.

$$C = K + Z_8 Z_4 + Z_8 Z_2$$

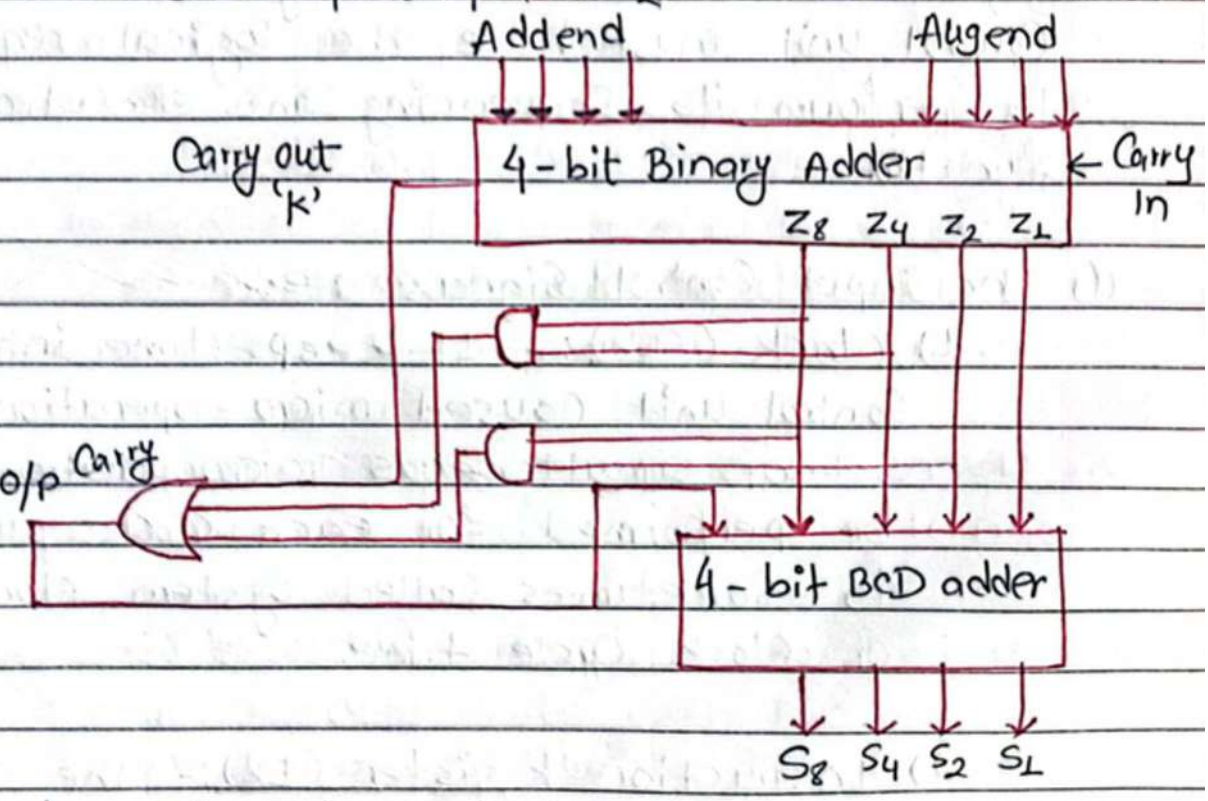


fig: Block diagram of BCD Adder.

Chapter V

Control Unit

Page :
Date : /

Structure of Control unit:-

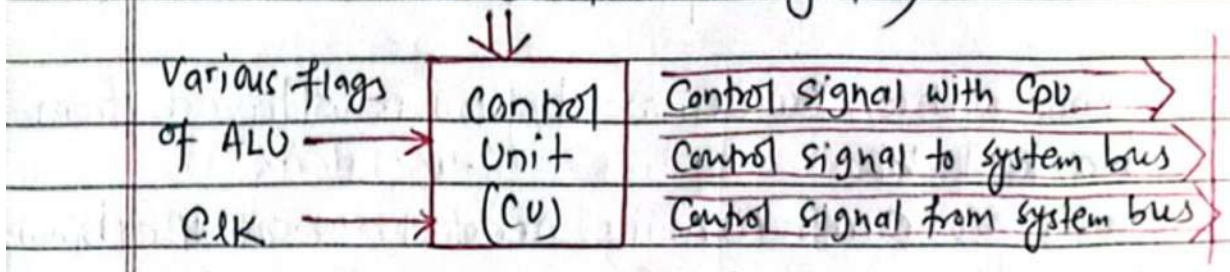


fig: Structure of Control Unit.

Control unit is a very important funct. of CPU that controls all the mechanism of Computer as well as it generate the next instruction into the control memory.

It has two signals i.e input control signal and output control signal. Internally Control unit must have the logical required to perform its sequencing and execution function.

(i) The input Control Signals are:-

1) Clock (CLK):- It keeps time into the Control unit caused micro-operation or a set of simultaneous microoperation to be performed for each clock pulse. This is sometimes called system clock time or clock cycle time.

2) Instruction Register (IR):- The opcode of the current instruction is used to determine which operation is to be

performed during the execution cycle.

3) Various flag of ALU :- These are needed by the control unit to determine status of the processor and output of previous ALU operation. It gives the different types of generation of flash.

4) Control signal from system bus :- It provides signals to the control unit such as interrupt signal and acknowledgement signal (ack) on the basis of input signal. The control signal activates certain output control signal which are responsible for execution of instruction that only depends on it.

(ii) The output control signals are :-

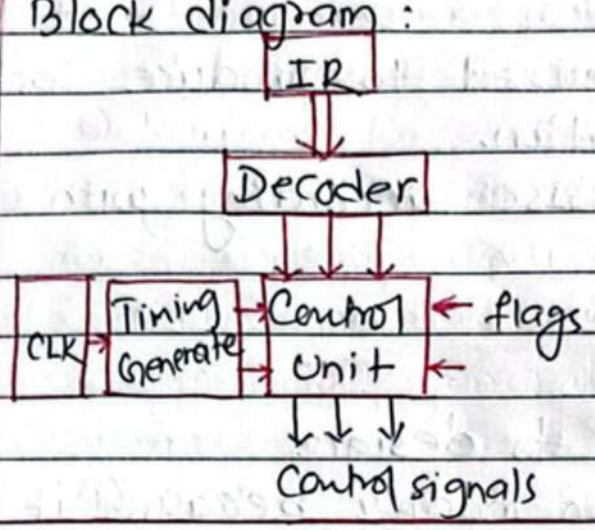
1) Control signal within or to CPU :-
It is of two types of micro operation either for data transfer or register transfer for performing ALU operation by using input or output register.

2) Control signal to system Bus :-
The main purpose of these signals are to transfer data from CPU register to memory or input output module.

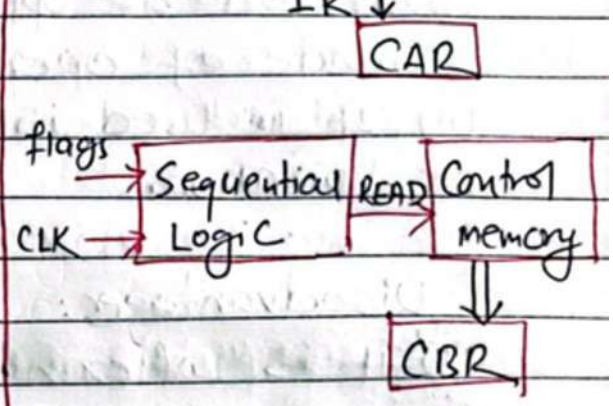
Difference between: Hardwired Control unit and Micro-programmed control unit.

Hardwired Control Unit	Micro-programmed Control unit
(i) It uses decoder logic gates, flags and other digital circuit.	(i) It uses sequence of micro-instruction in micro-programming language
(ii) It implies hardware Control unit.	(ii) It is a mid-way between hardware and software.
(iii) On the basis of input Signal, output signal is generated.	(iii) Micro-operation generates sequence of micro-instruction on the control line.
(iv) It is inflexible to modify	(iv) It is flexible to modify.
(v) It is difficult to design, test and implement	(v) It is easy to design test, less error and implementation.
(vi) It is faster mode of operation.	(vi) It is slower mode of operation.
(vii) It is used in RISC processor.	(vii) It is used in CISC Processor.

(viii) Block diagram :



(viii) Block diagram:-



Functioning of Hardwired Control Unit

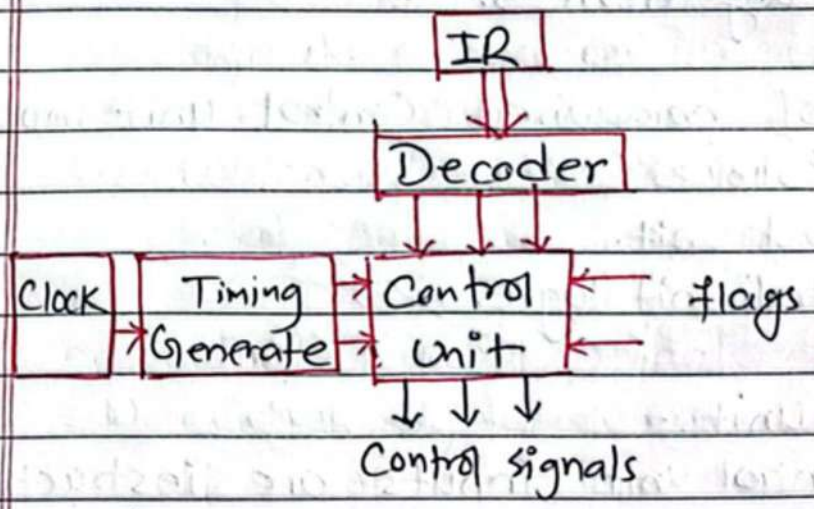


fig: Hardwired Control Unit

In hardwired organization Control logic is implemented with logic gates, flip-flop decoder and other digital circuit.

In hardwired control unit the output signal is generated only depends upon the input signal i.e clock signal and control bus signal.

Advantages:

(i) It can be optimized to produce fast mode of operation.

(ii) It is used in RISC according with cost variation.

Disadvantages:

(i) It is inflexible to design.

(ii) It is difficult to modify because it require complete change in circuit design.

(iii) It is quite difficult to test and implement as in many devices.

Issues of hardwired control unit implemented are as follows:

- Control unit.
- Control unit logic.

(i) Control unit:-

The control unit inputs are instruction register (IR), clock, flags and control bus signals. In this case of flags and control bus signal, each individual bit typically has some meaning.

Example : overflow.

The other two encode unit i.e IR and clock signal are directly useful to the control unit. The main issue is that it should be dependent upon the input signals.

Page:
Date: / /

(ii) Control unit logic:- It is also a main issue where internal logic of the control unit that produce output control signal as a function of its input signal.

Actually what must be done for each Control signal to a device as is a function of its input signal. Boolean expression defines the behaviour of Control unit must Control the state of different Cycle that issues a signal and then boolean logic gates gives the desired result.

The main issue is that it only defines the boolean expression or logical expression by using its input signal such as:

$PQ = 00$, Fetch Cycle.
$PQ = 01$, Instruction cycle.
$PQ = 10$, Execute cycle.
$PQ = 11$ Indirect Cycle.

Let the following Boolean expression is defined by 5 as:

$$C_5 = \bar{P}\bar{Q} \cdot T_2 + \bar{P} Q T_2$$

where,
C_5 = Control signal.
T_2 = Timing generate.

Page :
Date : / /

Imp # Functioning of microprogrammed Control Unit:
[Architecture of μ-programmed] CU.

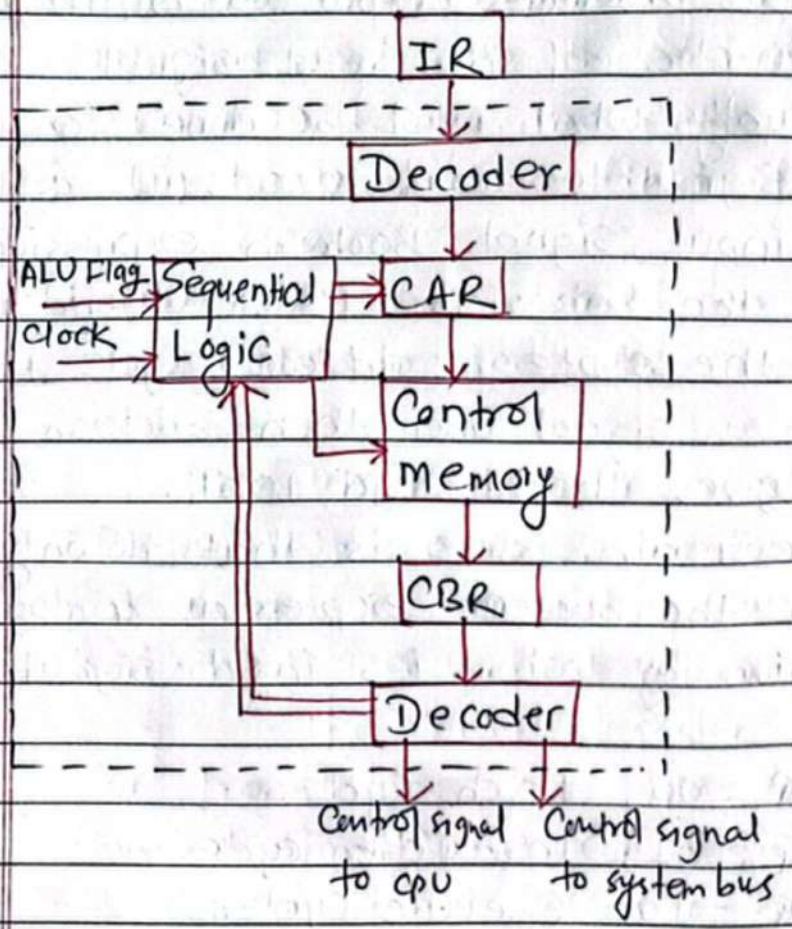


fig: Block diagram of μ-programmed Control unit.

CAR → Control Address Register.

The set of micro instruction is stored in the control memory. The control address register (CAR) stores the address of next instruction to be read. when a micro instruction is read from the control memory then it is transferred to the Control Buffer Register (CBR).

Page:
Date: / /

The CBR is connected to the control lines. Hence micro instruction is read from the control memory as executing that micro instruction.

The third element as shown in above figure is a sequential logic unit that leads CAR and issues READ commands. Control unit has the input such as ALU flags, clock and IR. The control unit functions as follows:

(i) The sequential logic unit issues a READ command to the control memory to execute the instruction.

(ii) The word or block whose address is specified in the CAR is read into the CBR.

(iii) The CBR generates control signal and next address information for the sequential logic unit.

(iv) The sequential logic unit loads new address into the CAR based on the next address information from the CBR and ALU flags.

All these happens during the single clock pulse.

CamScanner

At Conclusion of each micro-instruction the sequential logic unit loads new address into the CAR. Depending upon the value of ALU and CBR the three decision is met.

(i) ADD 1 to the CAR for next instruction
(ii) LOAD address field of CBR into the CAR.
(iii) LOAD CAR based on the operation Code in the IR.

The upper decoder translates the opcode of IR into the CAR. The lower decoder is not used for translating but only used for horizontal and vertical micro-instruction.

The lower decoder is also used for translating this Code into individual Control signal i.e control signal to CPU & Control signal to System bus.

Micro-Instruction and its types.

It is a symbolic notation that is used in micro-programming language which describes micro-operations.

Normally micro-instruction is also known as μ-program or sequence of instruction. The format of micro-instruction must include following fields:

(i) Micro-instruction address.
(ii) Jump Condition (unconditional branch, zero and overflow)

(iii) System bus Control signals.
(iv) Internal CPU Control signals.

micro-instruction format can be classified into 2 Categories:
- Horizontal μ-instruction format.
- Vertical μ-instruction format.

A> Horizontal micro-instruction format:-

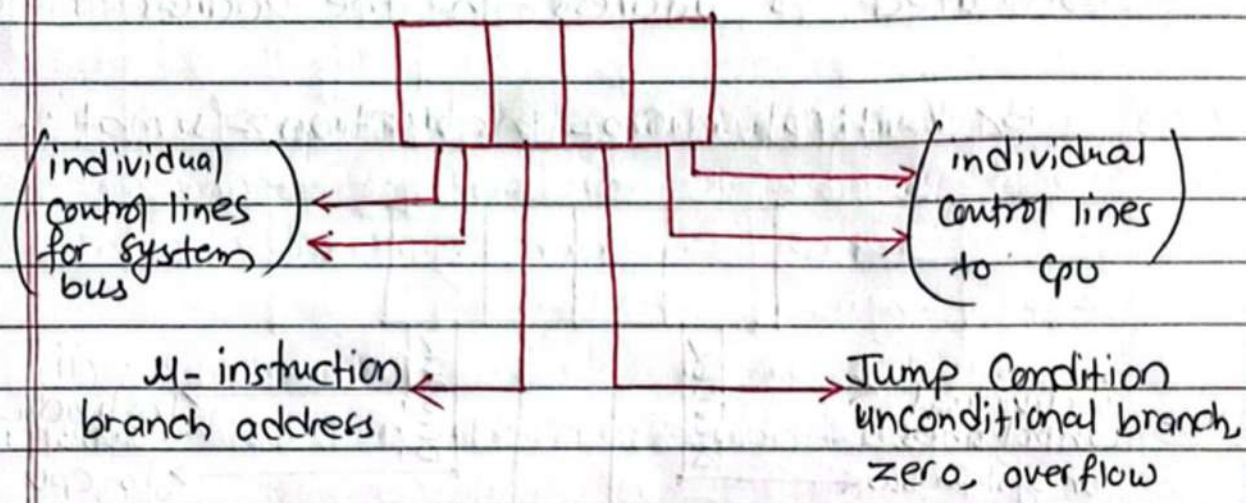


fig: Horizontal μ-instruction format.

In horizontal μ-instruction format, there is 1-bit for each internal processor control lines and another 1-bit for control lines or signals.

Therefore the length of μ-instruction is of 100 bits. The horizontal μ-instruction format is interpreted as follows:

(i) To execute micro-instruction turn ON all the control lines indicated by 1-bit and

turn OFF all the Control lines indicated by 0- bit.

(ii) If the Condition indicated by Condition bit is false then execute next micro-instruction in a sequence.

(iii) If the condition indicated by condition bit is true then next micro- instruction to be executed to address in the address field.

B> Vertical Micro- instruction format :-

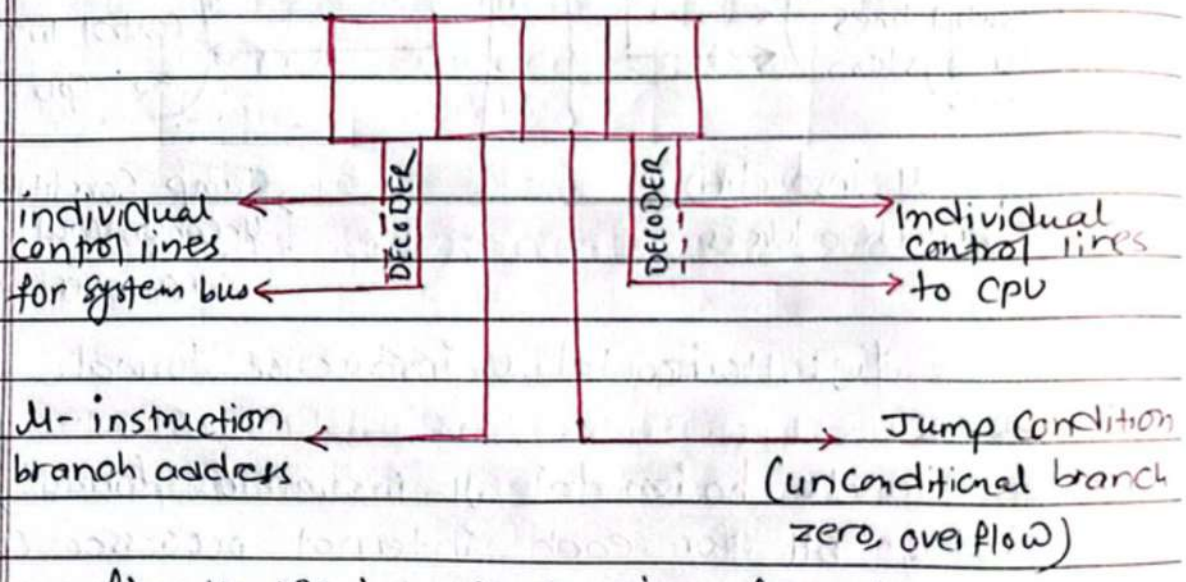


fig: Vertical μ- instruction format

In vertical μ- instruction format, the function code is used for each micro-instruction to be performed.

Function code is also known as decoder. Decoder is used to translate this opcode into individual control signals. The main

advantages of this format is that is has more compact (fewer bits) than horizontal micro instruction format.

It has 16-bit individual control lines.

Micro-Instruction Sequencing & Techniques:-

Micro-instruction sequencing means getting the next micro instruction from the control memory. Two consequences point are involved in μ-instruction sequencing:

(i) Size of μ-instruction.
(ii) Address generation time.

The first point is to try to minimize the size of control memory according to the size of micro instruction.

The second point is used to develop a design of micro-instruction that can be executed as fast as possible.

Micro-Instruction Sequencing Technique:-

Based on the current μ-instruction, condition flags and content of IR, a control memory address must be generated for the next micro-instruction.

The μ-instruction sequencing technique can be categorized into three types:-

- Two Address field.

- Single - Address field.
- Variable - Address field.

1) Two-Address field μ-instruction technique.

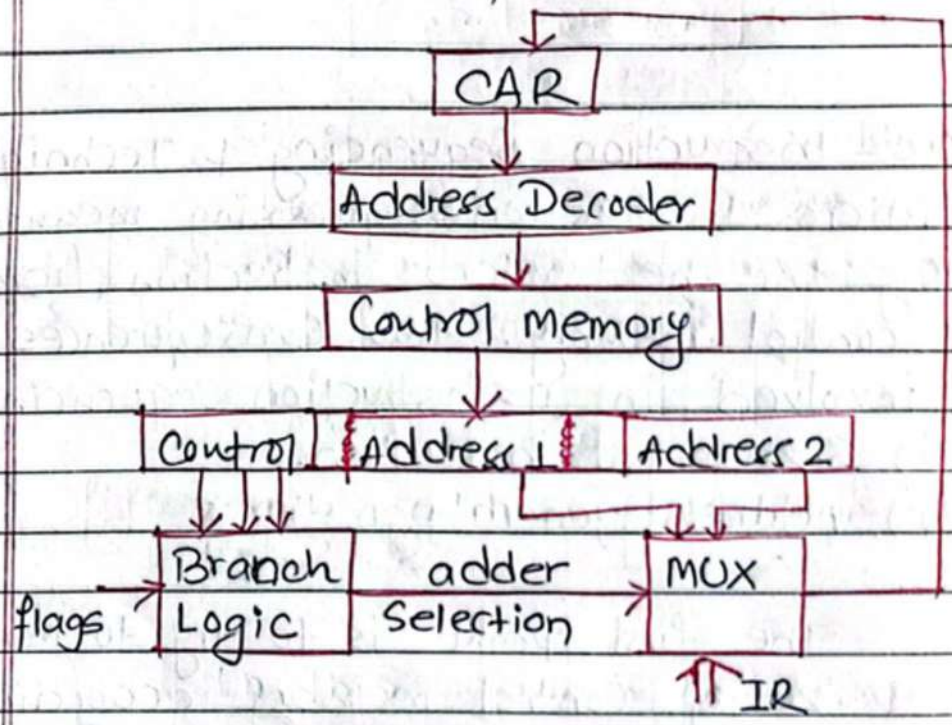


fig: two-address field μ-instruction technique.

The simplest approach two address field micro instruction technique is to fetch the next micro-instruction from Control memory.

~~Above Avobe~~ Above figure shows that how these information is to be used. A MUX is provided that serves as a destination for both serves as a destination for both address field with instruction register (IR).

Based on address selection input, the MUX translates the opcode and one of two address to the Control address register (CAR).

The CAR is subsequently decoded to produce next μ- instruction address. The address selection signal are provided by a branch logic and control signals from control unit.

Although, the two address μ- instruction sequencing technique is very simple because it has just two address field to control the logical address.

2) Single Address Field μ- instruction technique:

In this approach the address selection determines which option is selected. This techniques reduce the number of address field to one.

However, this is inefficient technique in the micro-instruction coding scheme.

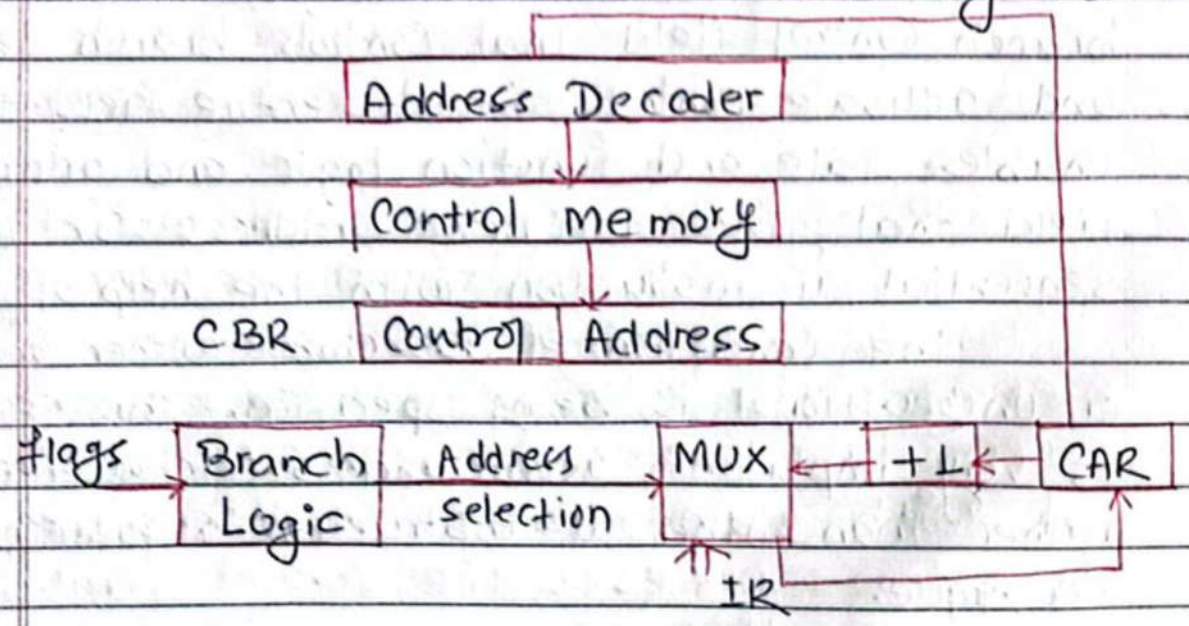


fig:- Single address field μ- instruction technique.

3) Variable Address field μ-instruction technique.

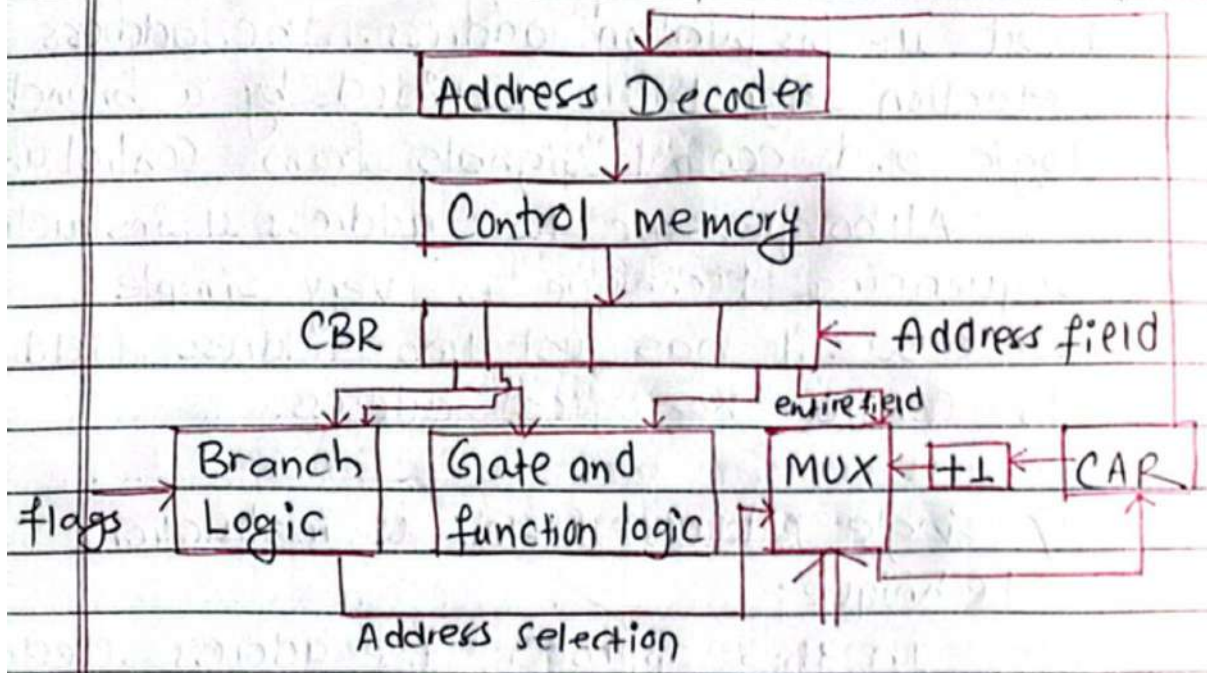


fig:- variable address field μ-instruction technique.

In this approach three entirely different micro-instruction field is designed such as branch control field that controls branch logic and activate control signals, entire field that enables gate and function logic and address field that gives the next address for next sequential μ-instruction with the help of IR.

The condition of branching either conditional or unconditional is being specified. This technique is very important, useful and sufficient technique rather than other μ-instruction sequencing technique.

Page:
Date: / /

Micro - Instruction execution :-

It is the process of getting the control signals from Control memory to perform the Specific operation. μ- instruction execution derived from the control memory based on IR, ALU flags, program Counter and clock.

The μ- instruction also derived from Sequencing logic that gives the signal to CAR and READ Command to Control memory with the help of next address control from CBR.

Application of Hardwired and micro-programmed Control Unit.

1. Realization of Computer.
2. It offers a systematic technique for control unit implementation is referred to as emulation.

   Emulation refers to use a micro-programming on machine to execute program for another machine.
3. It Supports operating systems.
4. Realization of special purpose devices.
5. It is used to Support ~~mainta~~ monitoring detection, isolation and repair of system error known as diagnostic of micro-programming

Chapter VI

Memory organization

Page:
Date: / /

Memory Hierarchy :-

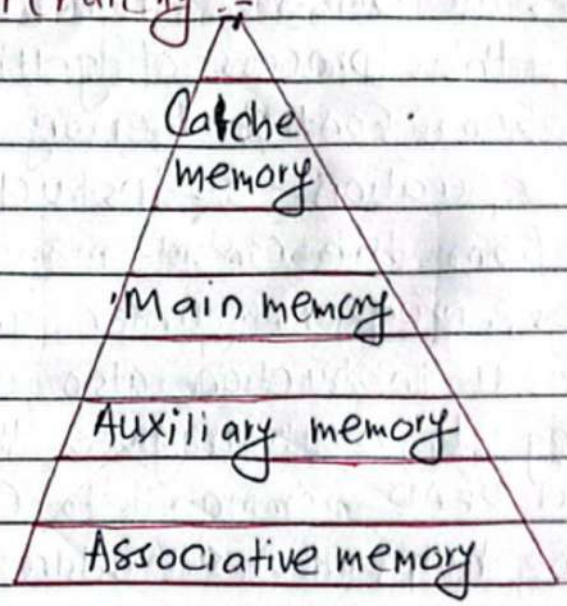


fig:- Memory hierarchy.

Cache Memory:-

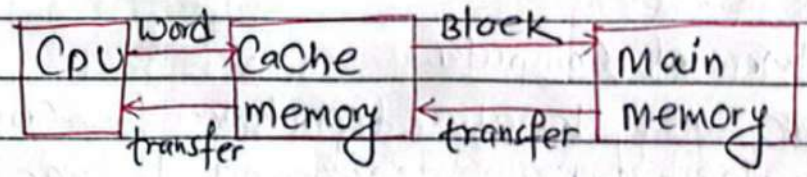


fig:- Block diagram of Cache memory.

Cache memory is a special type of memory which resides in between CPU and main memory. This memory is special memory and faster than main memory. CPU access data from Cache memory directly, but not with main memory. Therefore the speed of system increases the parallel operation.

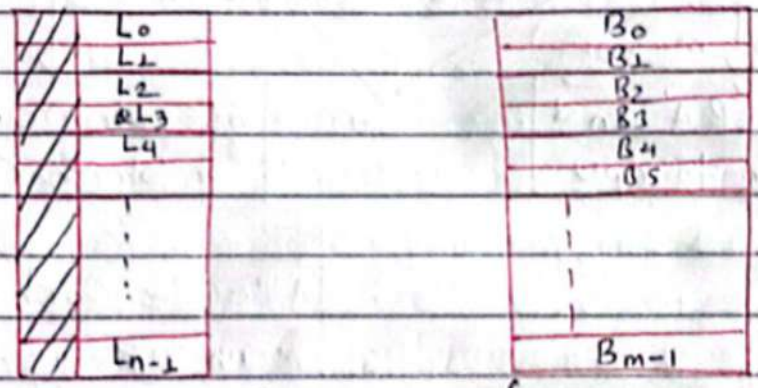
If block is found in Cache memory then this Condition is Called Cache hit and if not then Called as Cache miss.

Page :
Date : / /

Main memory is divided into multiple addressable Code and fixed number of Codes are Called blocks.

Cache memory is also divided into multiple lines known as Cache lines. A tag is associated with Cache line which gives information about which line is associated with which block.

Tag (Link)



(Cache Memory) (Main Memory)

Page:
Date: / /

Cache READ operation:-

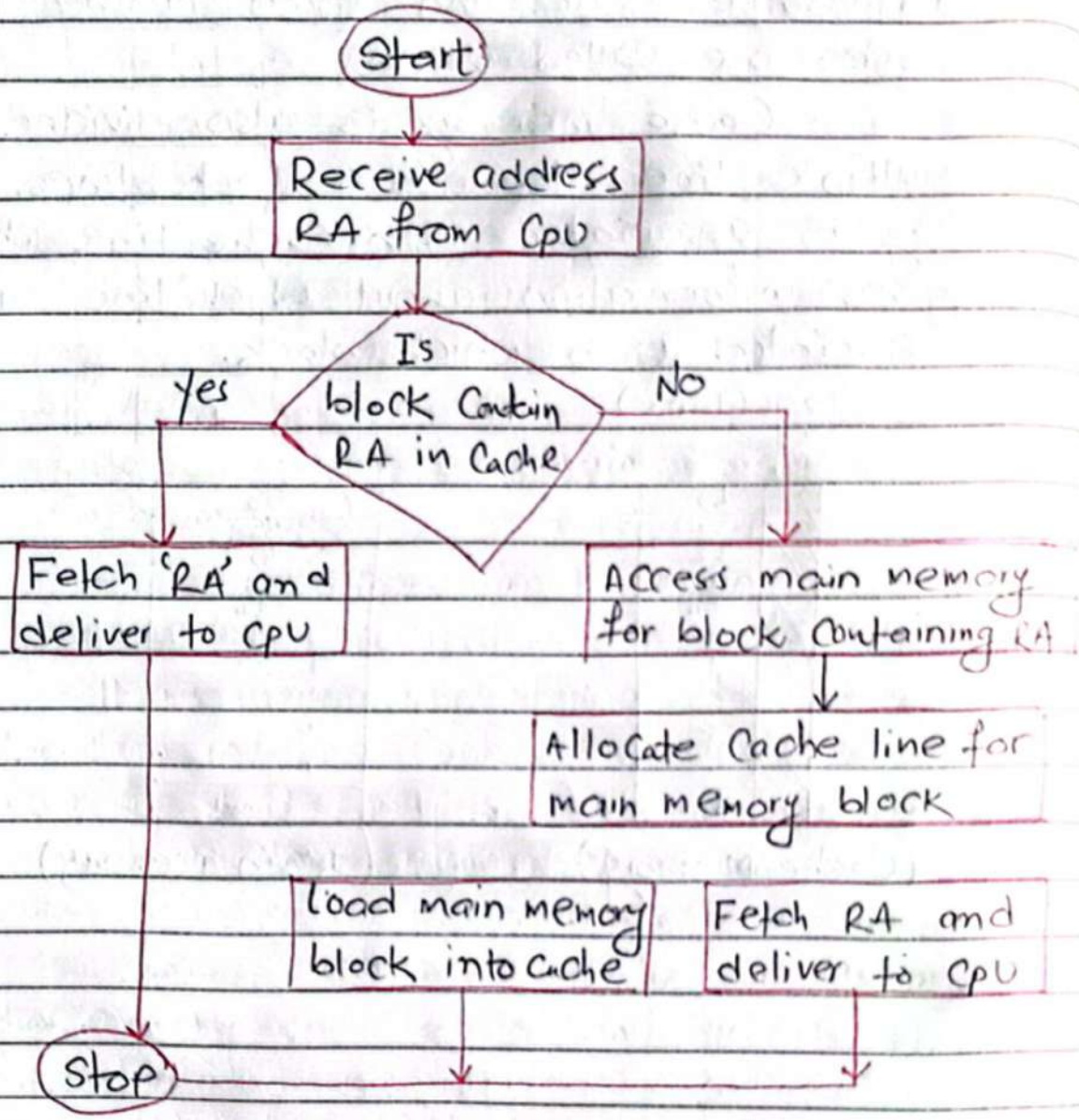


fig: flowchart for Cache READ operation.

CamScanner

Page :
Date : / /

Mapping Cache memory or Mapping function in Cache Memory :-

[Cache design issues in Cache Memory]

There are large number of main memory blocks which should reside in fewer Cache line. Mapping function gives which block belongs to which cache line.

There are three types of mapping functions. They are:-

(i) Direct Mapping.
(ii) Fully associative (associative) Mapping.
(iii) Set-associative Mapping.

Suppose there are 'n' number of Cache lines and 'm' number of memory blocks indicated by $L_0, L_1, L_2 \ldots L_{n-1}$ and $B_0, B_1, B_2 \ldots B_{m-1}$ respectively.

We know that the number of memory blocks is always greater than number of Cache lines. So to transfer the data or information from main memory to Cache memory, we must have to perform mapping between Cache and main memory.

1) Direct Mapping :-

Here fixed line number is mapped to fixed blocks. The mapping is done using module function.

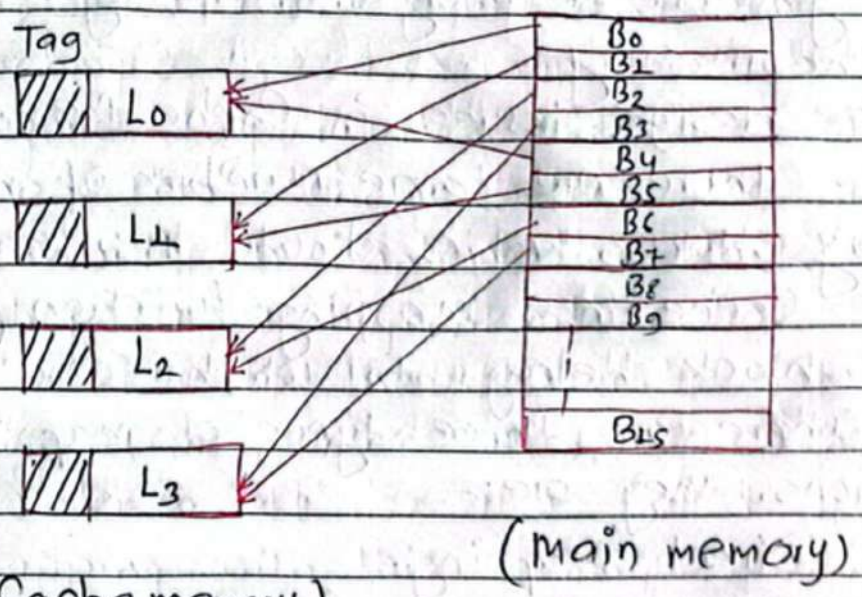


fig:- Direct mapping

It is very simplest way to assign a location in the Cache for each word in memory & to assign the Cache location based on the address of word in memory. This mapping is known as direct mapping.

Since each memory location is mapped to exactly one location in the ~~cat~~ Cache and is calculated as:

$B_j \rightarrow L_i$, if $i = j \bmod M$

where

i = Cache line number.

j = memory block number.

M = Total no. of cache lines.

Here $M = 4$

for $y = 0$, $0 \bmod 4 = 0 \rightarrow$ so $B_0 \rightarrow L_0$

for $y = 1$, $1 \bmod 4 = 1 \rightarrow$ so $B_1 \rightarrow L_1$

for $y = 2$ $2 \bmod 4 = 2 \rightarrow$ so $B_2 \rightarrow L_2$

for $y = 3$, $3 \text{ MOD } 4 = 3 \rightarrow$ so, $B_3 \rightarrow L_3$
for $y = 4$, $4 \text{ MOD } 4 = 0 \rightarrow$ so, $B_4 \rightarrow L_0$
for $y = 5$, $5 \text{ MOD } 4 = 1 \rightarrow$ so, $B_5 \rightarrow L_1$
for $y = 6$, $6 \text{ MOD } 4 = 2 \rightarrow$ so, $B_6 \rightarrow L_2$
for $y = 7$, $7 \text{ MOD } 4 = 3 \rightarrow$ so, $B_7 \rightarrow L_3$
⋮
so, on

2) Associative / fully associative mapping:-

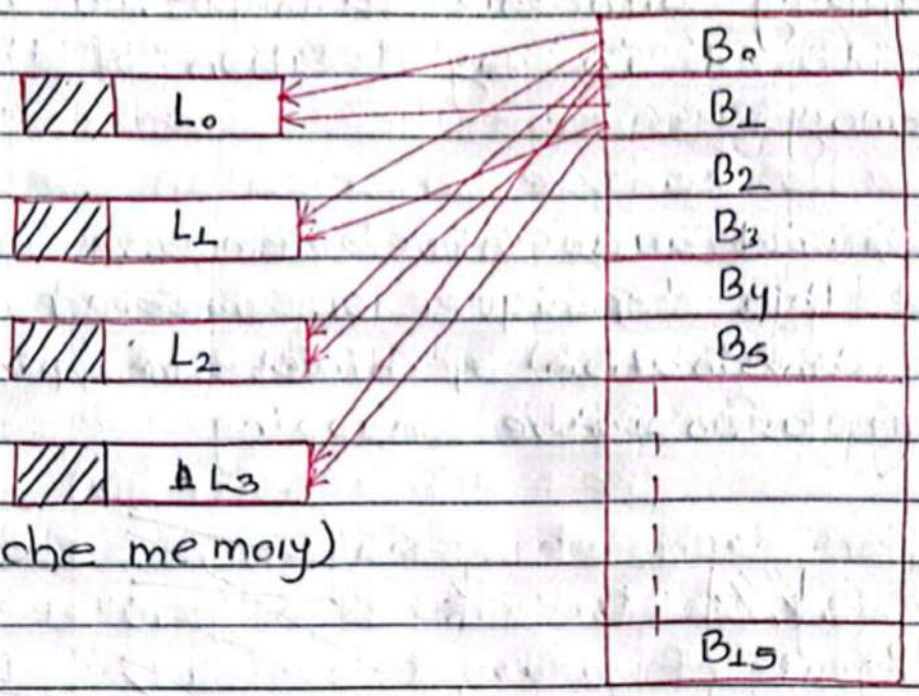


fig:- associative Mapping.

This mapping offers the most flexibility in mapping main memory block into Cache lines. Each block main memory can replaced to the any of available Cache memory.

Because of the flexibility cache tag

is most needed in each cache lines. This technique use associative search method wh. require tag to be compared with allocate lines. This scheme offers the greatest flexibility in implementing block replacement policy for a higher hit ratio.

Due to the flexibility some problems may occur during mapping such as traffic problems. It is also called fully associati mapping with this technique, a block can be placed in any location of the cache memory.

3) Set-associative memory:-

This technique is compared only the form of combination of direct mapping and fully associative mapping

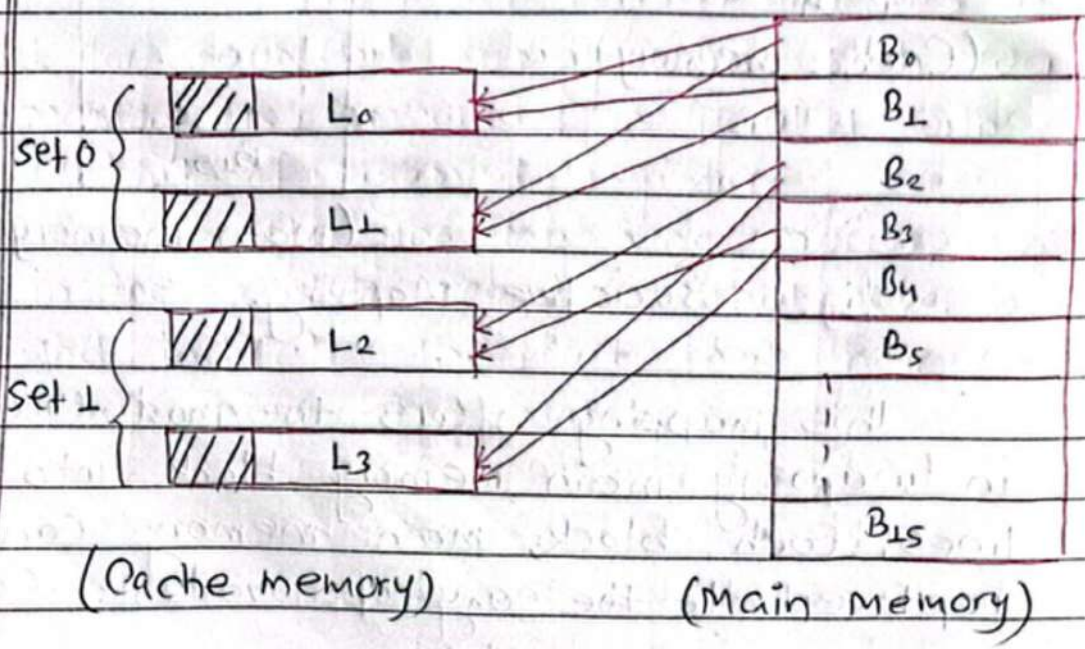


fig- Set-Associative Mapping.

Page :
Date : / /

In this mapping technique there are fixed number of location where each block can be placed. A set-associative mapping with 'n' locations for a block is called set-associative cache.

Each block in the memory maps to a unique set in the cache given by the tag and a block can be placed in any element of that set. Thus, a set-associative mapping technique combines direct and fully associative mapping in which directly mapped to a cache line and then all the block in the set are searched for a match.

Associative Memory (CAM) :-

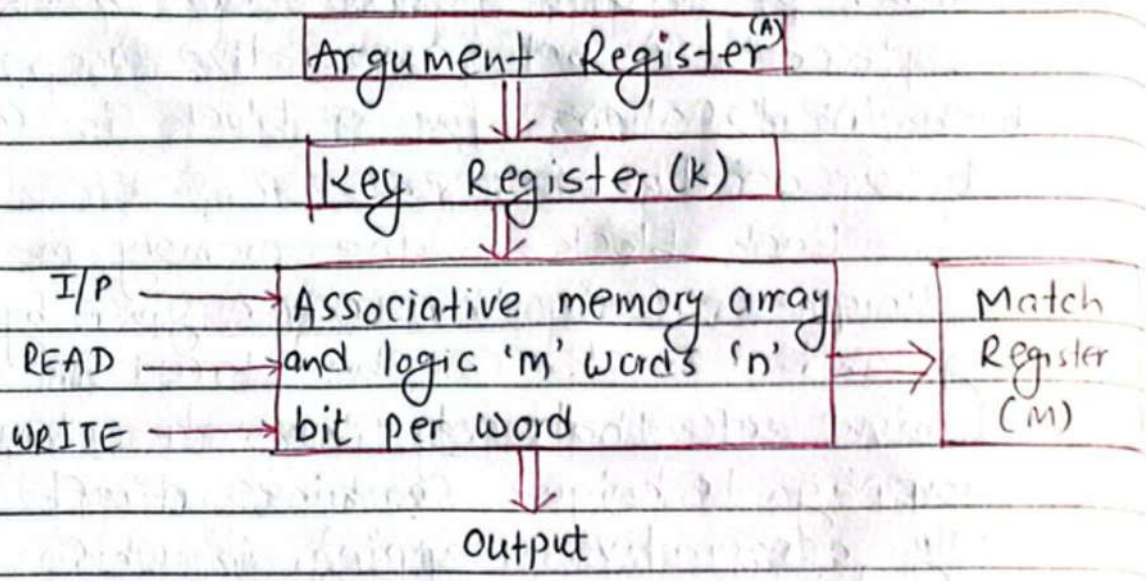
The time required to find an item stored in memory can be reduced if stored data can be identified for access by the content of data itself rather than by a address. A memory unit is accessed by the content called an associative memory or Content Addressable Memory (CAM).

It is used in application where the search time is very critical and must be very short. CAM can be described by the following components:

(i) Hardware organization.
(ii) Address matching logic.
(iii) READ/WRITE operation.

Page:
Date: / /

i) Hardware organization:-



Its hardware organization consist of a memory array and logic with 'm' words 'n' bits per word. The argument Register (R) and Key Register (K) each have n-bits for each bit of a word.

The match register (M) has m bits. 1 for each bit memory word. Each word in memory is compared in parallel with the content of argument register.

After the matching process, those bits in the match register that have been set and indicate the fact that their corresponding words have been matched.

Page :
Date : / /

ii) Address Matching Logic :-

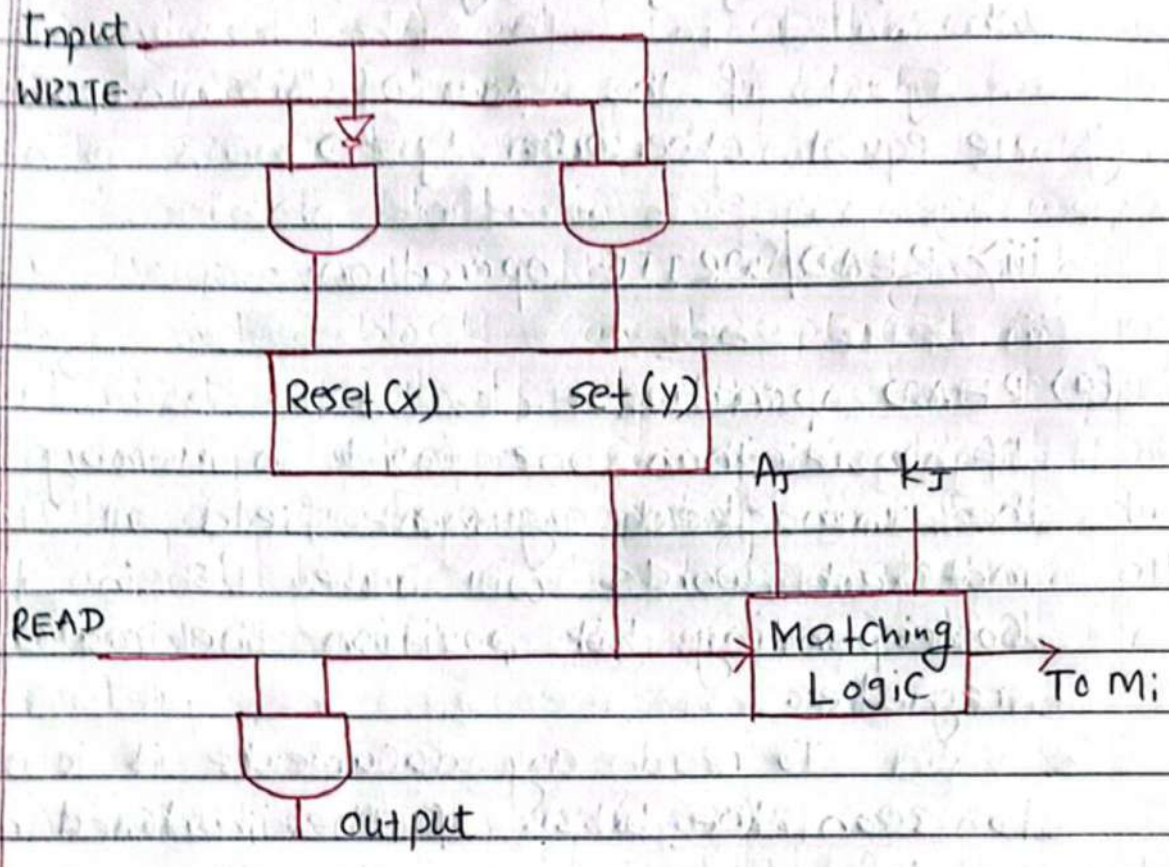


fig :- Logical Address Matching block.

The match logic for each code can be derived from the comparision algorithm for two binary numbers. For comparision we must compare the argument in 'A' with the bits stored in the cell (chunks) of the words.

words 'i' is equal to the argument in 'A' if $A_y = F_{iy}$ for $y = 1, 2, 3 \ldots n$. Two bits are equal if they are both 1 or both 0.

The equality of two bits can be expressed

logically by the Boolean function as:

$$x_y = A_y F_{iy} + A_i F_{iy}$$

where,

$x_y = 1$ if the pair of bits in position y are equal otherwise $x_y = 0$

iii> READ/WRITE operation:-

(a) READ operation:-

If more than one word in memory matches the unmasked argument field all the matched words will have 1's in the corresponding bit position the matched register.

To read any documents it is necessary to scan the bits of the matched register

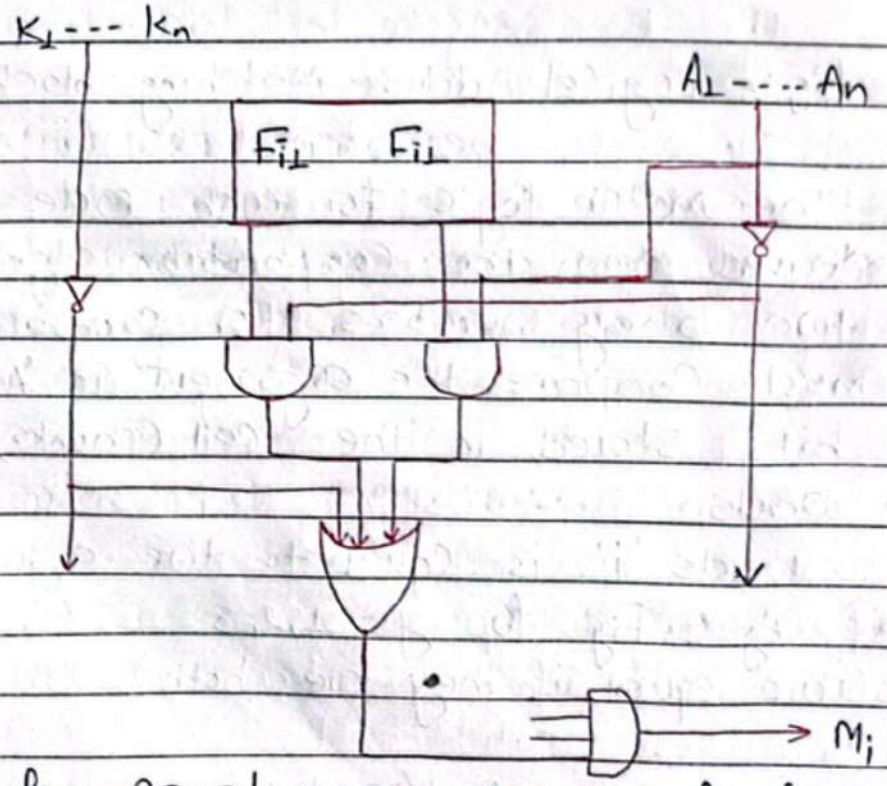


fig:- READ/WRITE operation for 'n' cell

Page :
Date : / /

The matched word are read in sequence by applying a READ signal to each word whose corresponding M_i bit is 1.

(b) WRITE operation:-
CAM must have a write capability for storing the information to be searched. It can take different forms depending upon the ~~opera~~ application.
If the entire memory is loaded with new information then search operation can search and then writing can be done by addressing each location in a sequence. It will make the device for waiting any data into the associative memory.

WRITE policy:-
It is the technique of updating main memory in which data can be updated either in main memory or in a cache memory. when a block is to be updated to main memory, it only update a cache memory block if input - output modules or any other devices wants to access block from main memory.
WRITE policy algorithm can be categorized as follows:-
- WRITE through.
- WRITE Back.
- WRITE once.

Page:
Date: / /

(i) WRITE through:-

In WRITE through policy, the blocks are updated in both Cache memory and main memory by Considering main memory block is always valid. The main problem in this policy is traffic problem.

(ii) WRITE Back:-

To avoid traffic problems of WRITE through policy another policy is used known as WRITE back policy.

According to this policy, the update Can be performed in main memory first and then Cache memory. The updated memory is valid if main memory block is also valid. To check the updating, check update bit is used that is always.

(iii) WRITE once:-

WRITE once policy is a unipolicy that only update the main memory block by checking either the main memory block is valid or not.

Replacement Algorithm:-

(i) FIFO
(ii) LIFO
(iii) LRU
(iv) OPR
(v) ~~RANDOM~~ Random.

CamScanner

Chapter VII

Input- output organization

Page :
Date : / /

External device Interface :-

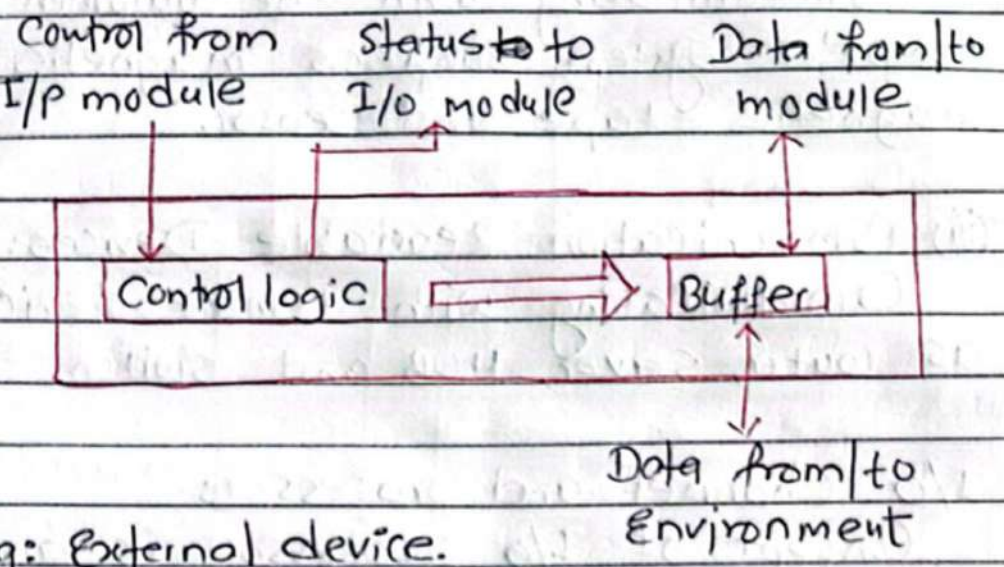


fig: External device.

It is a means of exchanging data between the external environment and Computer. It Can be used to Communicate between the Computer and outside world. To connect external devices and any sources for Computer, I/O module Can be used.

External devices Can be Categorized into 3 types:

(i) Human Readable Devices.
(ii) Machine Readable Devices.
(iii) Communication Readable Devices.

(i) Human Readable Device :-

Communicating with the equipment of Computer System such as printers and video Display terminals.

Page:
Date: / /

(ii) Machine Readable Devices:-
Communicating with the equipment of Computer System such as magnetic disk, magnetic tape and sensor.

(iii) Communication Readable Devices:-
Communicating with remote devices such as router, server, HUB and switch.

Imp # I/O Channel and processor:-
Concept of I/O processor is an extension of DMA (Direct Memory Access) Concept. The I/O channel and processor can execute Specialized input-output program existing in the memory with the intervention of CPU.

Therefore, CPU only needs to specify a sequence of I/O activities by the I/O channel and processor. It executes the necessary I/O instruction which are required for the task and sends the interrupt signal to the CPU. So in a system with I/O channel, the CPU doesnot execute I/O instruction.

The Concept of I/O Channel and processor refer to as DMA but it is more powerful than DMA Concept. It is Categorized into two types:
(i) Selector channel.
(ii) Multiplexing channel.

CamScanner

Page :
Date : / /

17 Selector channel :-

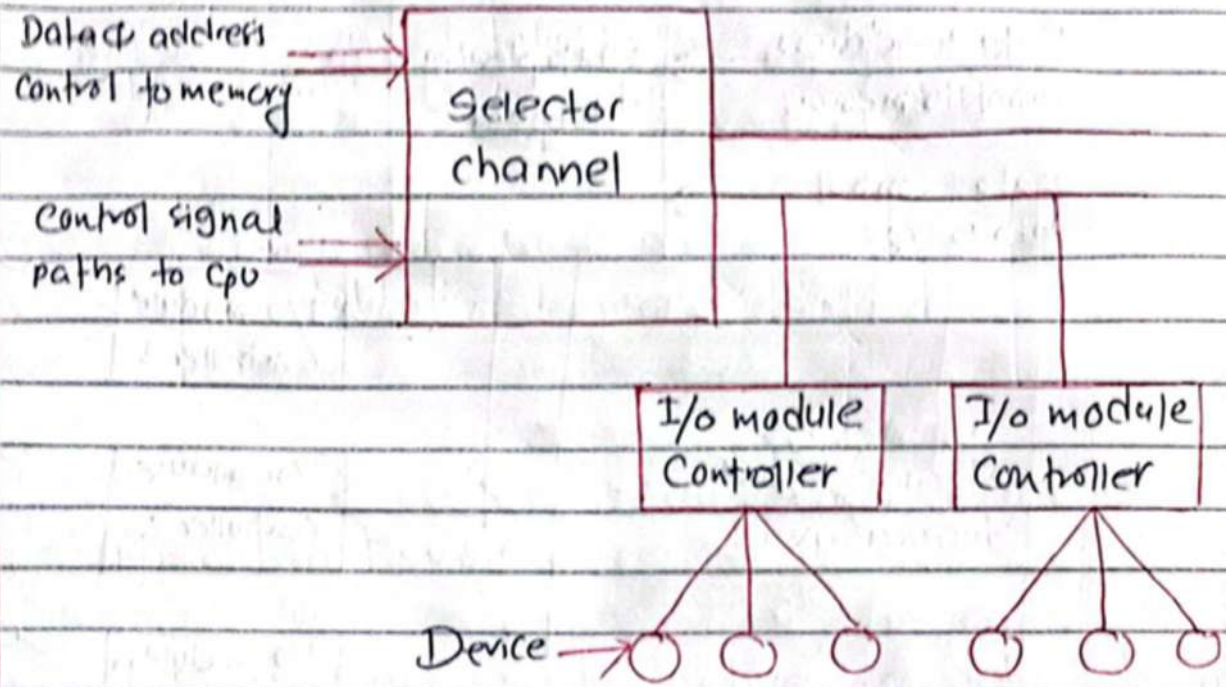


fig:- Selector channel

For high speed devices selector channel is used. The channel can transfer data from the high speed devices at a time. Therefore the input output channel select one device and initiates the data transfer at a one clock pulse.

Each I/O device is handled by input output (I/O) module controller. Therefore, the I/O channel serve as CPU that can transfer the data.

Page:
Date: / /

2) Multiplexing Channel:-

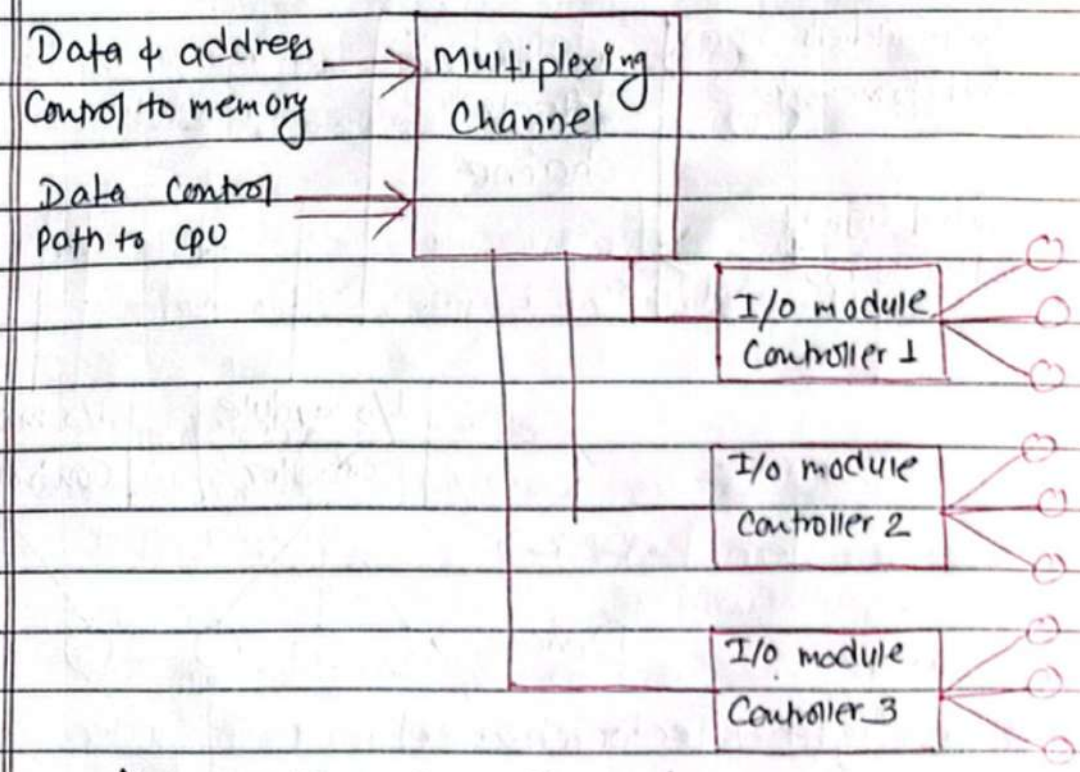


fig:- Multiplexing Channel

A multiplexing channel can handle I/O with multiple devices at the same time. For low speed devices a multiplexer channel transmit character as fast as possible to multiple devices.

Suppose, we have 3 I/O module Controller which needs to send individual bytes as:

$$\left\{ \begin{array}{l} A_1, A_2, A_3, A_4 \ldots \\ B_1, B_2, B_3, B_4 \ldots \\ C_1, C_2, C_3, C_4 \ldots \end{array} \right\}$$

A multiplexing channel may send above byte as: $\{A_1\ B_1\ C_1\ A_2\ B_2\ C_2\ A_3\ B_3\ C_3\ A_4\ B_4\ C_4\}$

Page :
Date : / /

Input output techniques:-

1> programmed I/o technique:-

The programmed I/o operations are Completely Controlled by the cpu. Cpu executes program that initiate, terminates and direct the I/o operation.

when the processor issues a Command to the I/o module, it must wait until the I/o operation is Complete.

If the processor is faster than I/o module then this is a wastage of processor time i.e limitation of programmed I/o technique.

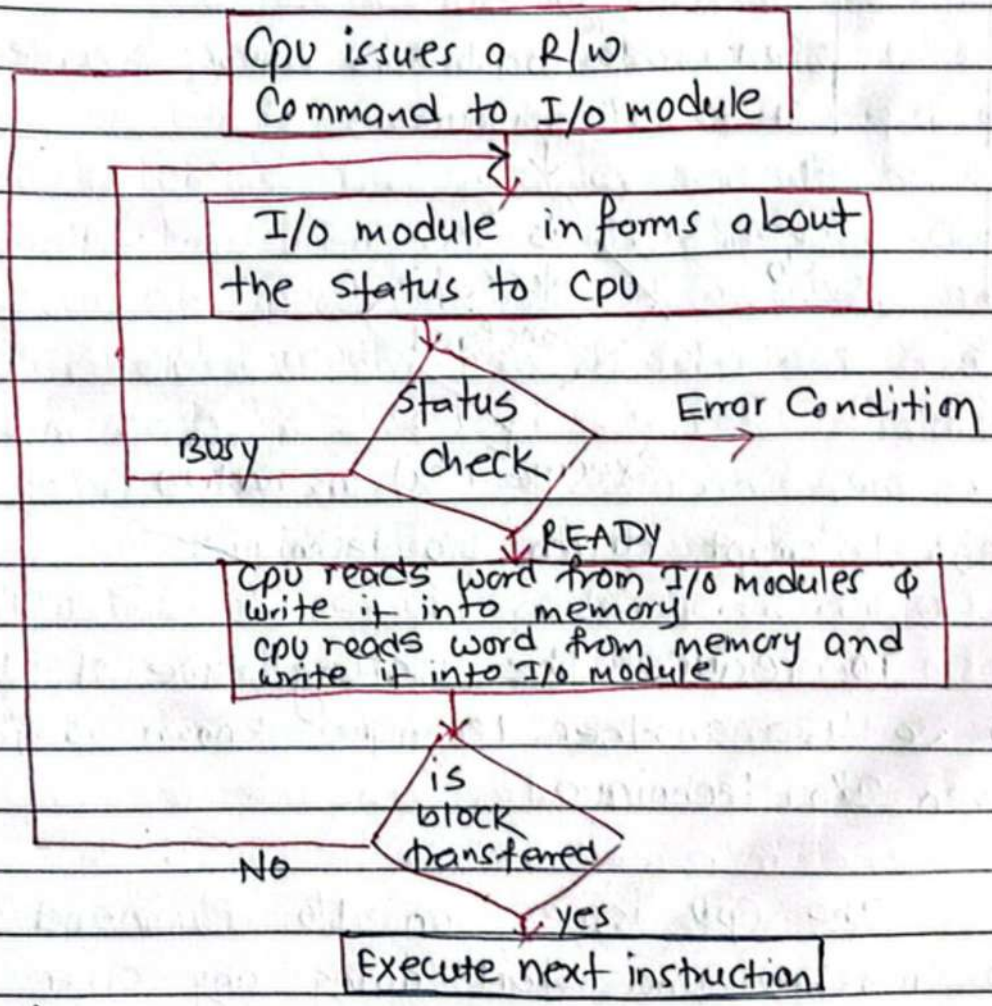


fig:- programmed I/o technique.

Page:
Date: / /

2) Interrupt driven I/O technique:-

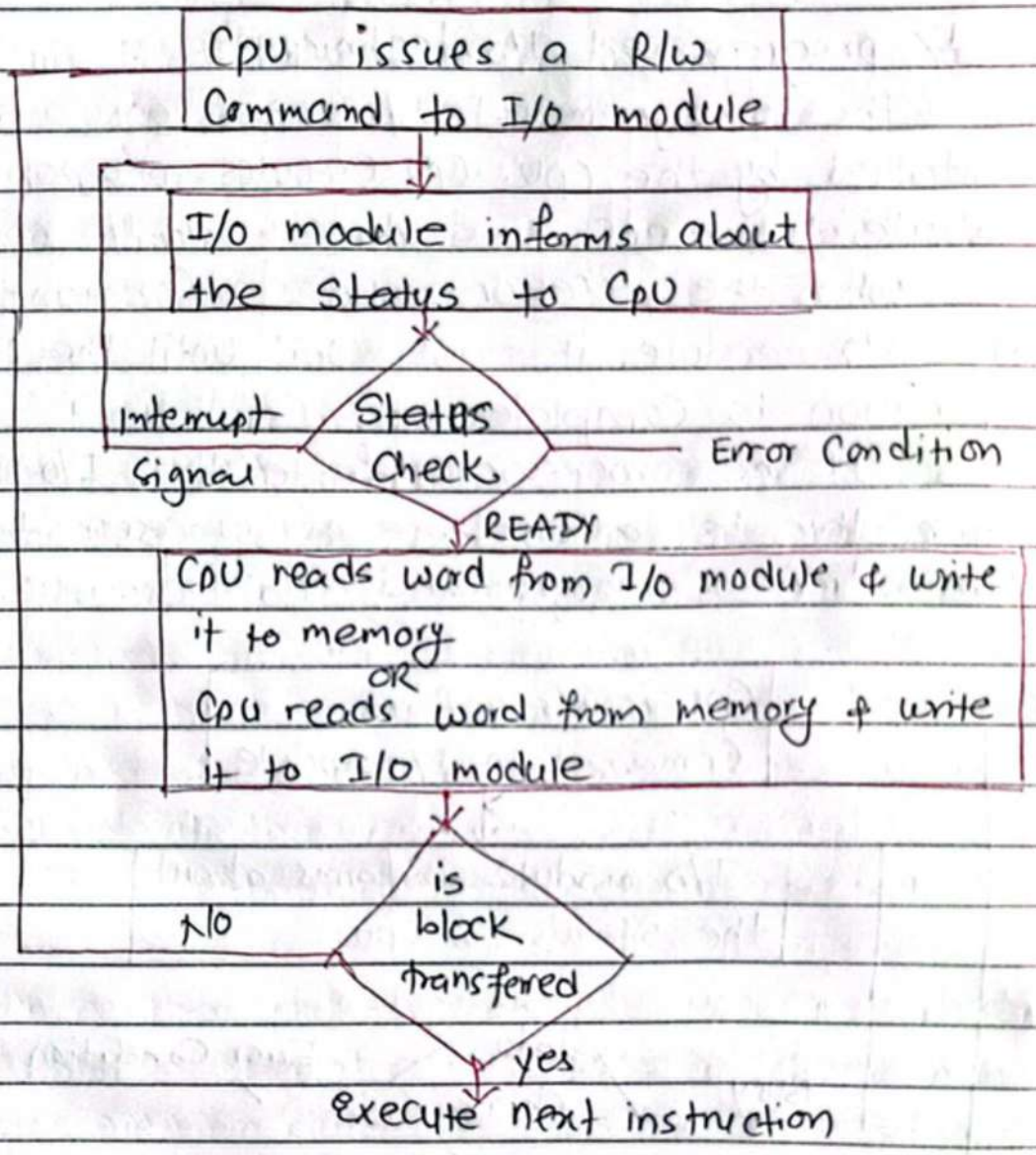


fig: Interrupt driven I/O technique.

To reduce the waiting time of the Cpu we use another technique known as interrupt driven I/O technique.

The Cpu issues an I/O Command to the I/O module and start doing any other work which may be execution of Seperate program

Without waiting Completion of I/O operation.

I/O module processed reading or writing of data from associated devices and when I/O operation is complete, the I/O module interrupt CPU by informing the CPU i.e I/O operation has finished then the CPU resume (restart) the execution of previous program.

The processor is all time busy for executing number of instruction for data transfer. It is the limitation of interrupt driven I/O technique.

3) Direct Memory Access (DMA) technique:-

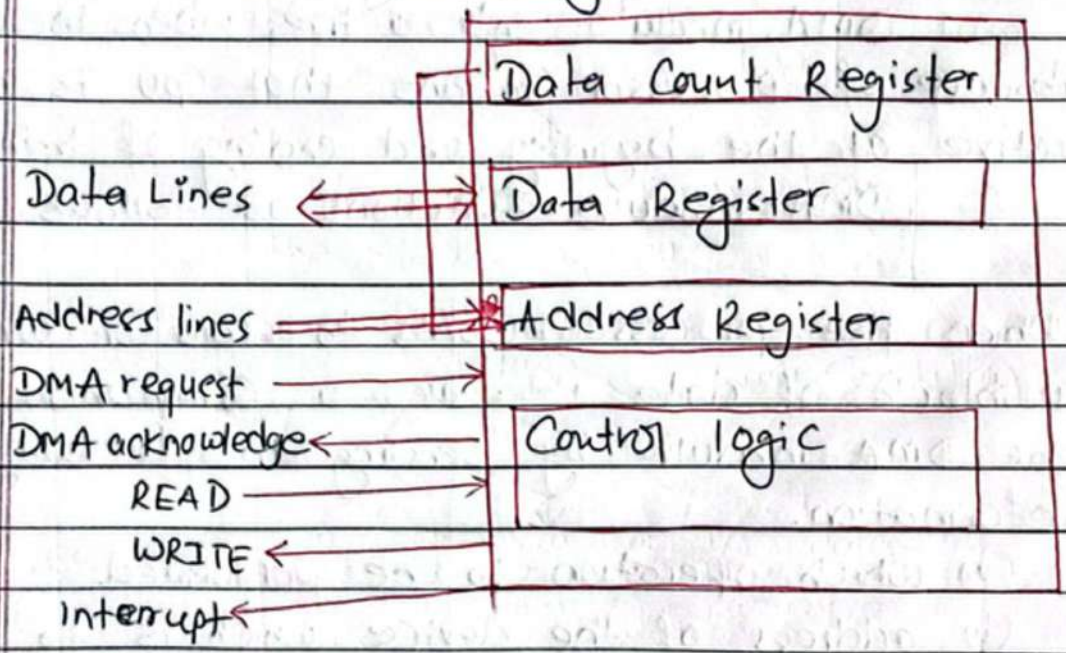


fig: block diagram of DMA or Data Cycle.

The drawback of programmed I/O and interrupt driven I/O technique are:

(i) wastage of CPU time in programmed I/O

(ii) The processor is all time busy for executing

number of instructions for data transfer in interrupt driven I/O technique.

(iii) the data transfer ~~in interrupt driven~~ rate only depends on the processor speed.

To eliminate above limitations DMA Concept has been used.

DMA function:-

DMA module as a processor must use the system bus when the processor doesnot need or it must force the processor to suspend operation temporarily.

DMA module only active when there is absence of processor means that CPU is only active at the begining and ending of transfer.

DMA module functions as follows:

(i) When the processor wants to read or write a block of data it issues a command to the DMA module by sending the following information.

(a) which operation to be performed?

(b) address of the device which is to be used.

(c) Starting location on the memory where the information to be read or write and stored by DMA module in address register.

Page :
Date : / /

(ii) The processor then Continues with other work. The DMA module transfer the entire block and data directly to or from memory without involvement of CPU.

(iii) Since DMA module share the system bus. therefore it need to have some way to take Control of the system bus. so that data Can be transfered by using the system bus supported by DMA module only when the processor does not need it.

CamScanner

Chapter VIII

Reduced Instruction set Computers (RISC)

Page:
Date: / /

RISC vs CISC

	RISC	CISC
(i)	It has a few number of instruction	(i) It has large number of instruction.
(ii)	It has few no. of addressing modes.	(ii) It has a variety of addressing modes.
(iii)	All operations are done within the register by CPU.	(iii) All operations can be manipulated in memory
(iv)	variable length instruction format is not suitable.	(iv) Variable length instruction format is suitable.
(v)	Single cycle execution	(v) Multiple cycle instruction execution.
(vi)	Hardwired Control unit rather than micro-programmed Control unit	(vi) μ-programmed Control unit rather than hardwired Control unit.
(vii)	Example: power-PC	(vii) Example: pentium-PC

CamScanner

Page :
Date : / /

RISC Instructions:-

1) LB (Load Byte).
2) LBU (Load Byte Unsigned).
3) LH (Load Half word).
4) LHU (Load Half word Unsigned).
5) LW (Load word).
6) LWU (Load word Unsigned).
7) LWL (Load word Left).
8) LWR (Load word Right).
9) SB (Store Byte).
10) SH (Store Half word).
11) SW (Store word).
12) SWL (Store word Left).
13) SWR (Store word Right).

Note:

parallel processing
- Pipelining processing technique
  - Arithmetic pipelining
  - RISC pipelining
  - Instruction pipelining
- Array processing
- vector processing

CamScanner

Page:
Date: / /

RISC pipelining:-

In RISC architecture most of the instruction are register to register and it has following instruction cycle phase:-

(i) 2 phase instruction cycle
I: Instruction Fetch.
E: Execution Instruction.

(ii) 3 phase Instruction cycle
(for load and store)
I: Instruction fetch.
E: Execution Instruction.
D: Register to memory or memory to register.

Example:

Suppose we have five instructions as:

I_1 = LOAD, $R_1 \leftarrow M$
I_2 = LOAD, $R_2 \leftarrow M$
I_3 = ADD R_3, R_2, R_1 i.e $R_3 = R_2 + R_1$
I_4 = STORE, $M \leftarrow R_3$
I_5 = Branch X

The sequential execution of above instructio will be as

	t_0	t_1	t_2
I_1 :	I	E	D

	$t=3$	$t=4$	$t=5$
I_2 :	I	E	D

Page :
Date : / /

	t=6	t=7
I_3:	I	E

	t=8	t=9	t=10
I_4:	I	E	D

	t=11	t=12
I_5:	I	E

fig: Sequential execution of above 5-instructions i.e Non-pipelining execution.

In RISC pipelining above five instruction will be as:

	t=0	t=1	t=2
I_1:	I	E	D

	t=1	t=2	t=3
I_2:	I	E	D

	t=2	t=3
I_3:	I	E

	t=3	t=4	t=5
I_4:	I	E	D

	t=5	t=6
I_5:	I	E

fig: RISC pipelining of above 5-instruction i.e parallel processing.

CamScanner

Page:
Date: / /

Instruction pipelining:

pipelining process occurs not only in data stream but also in instruction stream and instruction pipelining is achieved by dividing the instruction execution job into number of segments or stages and applying overall segments of different instructions.

The instructions can be decomposed into the following six segments.

- Fetch Instruction (FI)
- Decode Instruction (DI).
- Calculate operand Address (CO).
- Fetch operand (FO).
- Execute instruction (EI).
- write Result operand (WO).

The Sequential execution of instruction can be illustrated as

Instruction	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
Instruction 1	FI	DI	CO	FO	EI	WO							
Instruction 2							FI	DI	CO	FO	EI	WO	
Instruction 3													FI

fig: sequential execution of instruction (non-pipelining).

Page :
Date : / /

The execution of instruction in instruction pipelining can be illustrated as:

Instruction	t=0	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8
Instruction 1	FI	DI	CO	FO	EI	WO			
Instruction 2		FI	DI	CO	FO	EI	WO		
Instruction 3			FI	DI	CO	FO	EI	WO	

fig: Instruction pipelining or timing diagram of instruction pipelining.

Each instruction requires all six segments, equal clock pulse, no data dependency and no control unit. So that instruction pipelining deals with parallel processing or parallelism or overlapping of instructions.

Conflicts in instruction pipelining and solution [pipelining Hazards]

Due to some problems, a pipelining cannot work with its full performance then it is known as pipelining hazards. So many conflicts may arise in the instruction.

Pipelining can be called as parallel processing in which some problems may occur is called as pipelining hazards. There are

Page:
Date: / /

three types of pipelining hazards they are:-
- Structural Hazards.
- Data Hazards (Data dependency).
- Control Hazards (Branch difficulty).

1> Structural Hazards:-
It is problem due to the limitation of hardware.
For example: If two instruction want to access memory in the same clock cycle then it is not possible.

Solution:
This hazard can be minimized by using separate memory for data and instruction.

2> Data Hazards:-
It arise when an instruction depends on the result of previous instruction but this result is not available. Due to this the performance of pipelining system degrades.
This can be minimized by the following solution:
(i) delayed load.
(ii) operand forward.
(iii) Hardware interlock.

3> Control Hazards:-
One of the major problem in instruction

Page :
Date : / /

pipelining is the occurence of branch instruction. A branch instruction can be conditional or unconditional.

In conditional branch, the target instruction is selected if condition is satisfied & the next sequential instruction is selected if condition is not satisfied.

In unconditional branch always change the sequential flow of execution by loading program counter with the target address.

Solution:

(i) Branch prediction.
(ii) Delayed branch.
(iii) Loop buffer.
(iv) Multiple buffer.
(v) Multiple stream.
(vi) prefetch branch.

Numerical:

1. Assume that pipelining has $k=8$ segment and executes $n=120$ tasks in a sequence. Let the time taken to process a sub-operation in each segment is 40 ns. Calculate the speed-up ratio in the pipeline.

→ Soln:-

Time taken to complete all the operation in a pipeline $= (k+n-1)t_p$

Page:
Date: / /

Where,

k = no. of segment = 8

n = no. of tasks = 120

t_p = time required to complete each task in a pipeline = 40 ns

Now,

Time taken to complete all the operation in a pipeline = $(8+120-1) \times 40$ ns

= 5080 ns

* Time taken to complete all the operation in a non-pipeline = nt_n

Where,

t_n = time required to complete each task in a non-pipelining.

$t_n = kt_p$

$= 8 \times 40$ s

$= 320$ s

Now,

Time taken to complete all the operation in a non pipeline = $n.t_n$

$= 120 \times 320$ ns

Again

Speed-up ratio $= \dfrac{nt_n}{(k+n-1)t_p} = \dfrac{n.kt_p}{(k+n-1)t_p}$

$$S = \frac{n.k}{k+n-1}$$

$$= \frac{120 \times 8}{8+120-1} = 7.55$$

$\therefore$ Speed up ratio, $S = 7.55$

CS CamScanner

Page :
Date : / /

Overlapping Register window:-

[Register window / Large Register file]

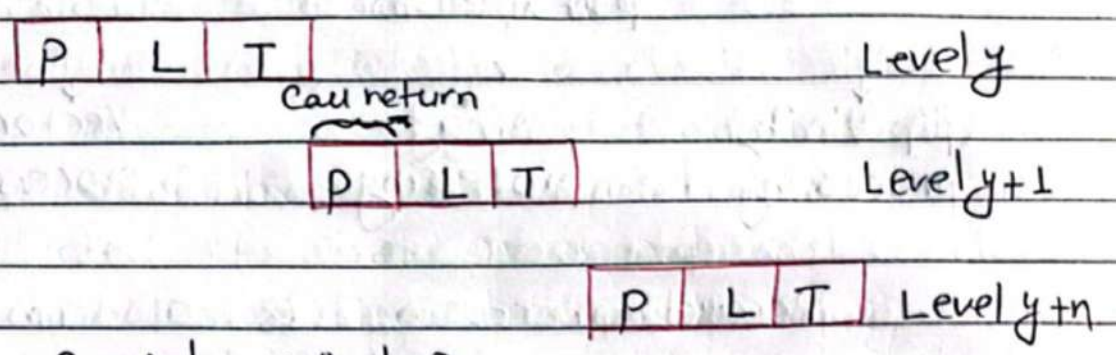


Fig: Register window.

p → parameter register.
L → local register.
T → temporary register.

In the face of it, the use of large set of register should decrease the need to access memory because the most of operand reference are local and maximum global operand or variable reserved in memory globally.

So some problems may occur. These types of problems can be removed by using register window that converts global variable into local that must be stored by using the call programs i.e call or return.

Overlapping register window in RISC processor minimize the use of memory.

Chapter IX

Introduction ~~Introduction~~ to parallel processing.

Page:
Date: / /

Parallelism in Uniprocessor System :-

parallelism

- pipelining processing
- Array processing
- Vector processing

Parallel processing is a term used to denote a large class of techniques that are used to provide simultaneously data processing task for the purpose of increasing computational speed of the computer system instead of processing on sequential data operation.

It is about to perform current data processing to achieve fast execution time. The system may have two or more processors. If system have one processor that executes instruction parallely called as parallelism in uniprocessor system.

But mostly parallel processing is occured in two or more processors. The main purpose of parallel processing is to speed up the computer processing capacity and to increase the troughput.

For parallel processing extra hardware may require which may increase the cost of the system. parallel processing is is divided into three types:

(i) pipelining processing.

Page :
Date : / /

(ii) Array processing.
(iii) vector processing.

(i) pipelining processing:-

It is the technique of decomposing sequential process into sub-process with each sub process being executed into special dedicated segments that operates concurrently with all other segments. Each segment paforms partial processing of the complete task. The result is obtained from each segments is transferred to the resultant. The final result is passed and obtained defined operation.

The main purpose of pipelining is several computation in the pipelining can be progressed by overlapping of multiple instruction

Example: Instruction pipelining.

(ii) Array processing:-

The main concept of array processing is to implement many processing element by a single CPU. All processing element have a single CPU that enhance ALU operations and local memory.

All the array processor are designed properly that means it can perform operation in a parallelism concept.

Array processing is mainly responsible for computer and floating point operation but following issues may arise during array processing.

CamScanner

Page:
Date: / /

- how the processing element structurized?
- how the processing element are interconnected
- how much the local memory have elements?

(iii) vector processing:-

To increase the performance of computer new concept and technology has been invented known as vector processing. To increase the speed of computer while performing complex calculation like vector, vector processing is used.

The complex calculations are used in super computers so it is also used in super scaler technology. In vector processing, vector values can be calculated directly.

characteristics of multiprocessor system:-

A multi-processor system is an inter-connection of two or more cpu with memory and input-output device. The term processor in a multiprocessing system means combination of cpu having memory and an I/o module. However, a system with a single cpu is usually not included in the defination of multi-processor system.

Following are the characteristics of multi-processor system:-

(i) It can improve performance by decomposing a program into parallel executable task.

(ii) processing is very fast rather than uniprocessor system.

CamScanner

Page :
Date : / /

(iii) Execution Speed is also fast.
(iv) Complex operations can be done easily.
(v) Not dependent to the data dependency system.
(vi) Data processing task can be done routinewise.

Flynn's classification.
[Enslow's Taxonomy]

Most common way of categorizing system with parallel processing capabilities is known as flynn's classification. It is categorized into four types:-

(i) SISD (Single Instruction Single Data stream):-
Single processor executes a single instruction to operate on data stored in a single memory.

(ii) SIMD (Single Instruction Multiple Data stream):-
Single processor executes single instruction to operate on data stored in multiple memory. It is used in vector and array processing.

(iii) MISD (Multiple Instruction Single Data stream):-
Multiple processor executes multiple instruction to operate data stored in a single memory. It is not yet developed and implement.

(iv) MIMD (Multiple Instruction Multiple Data stream):-
Multiple processor executes multiple instruction

to operate on data stored in a multiple memory.

V.V.Imp # Interconnection structure in Multiprocessor system :-

The Components that form a multiprocessor system are CPU, I/p devices and memory unit that may be partitioned into number of seperate modules.

The interconnection between the Components can have different physical configuration. There are several physical configuration that establish the ~~ins~~ interconnection of network. The configuration are as follows :-

(i) Time Shared Common Bus :-

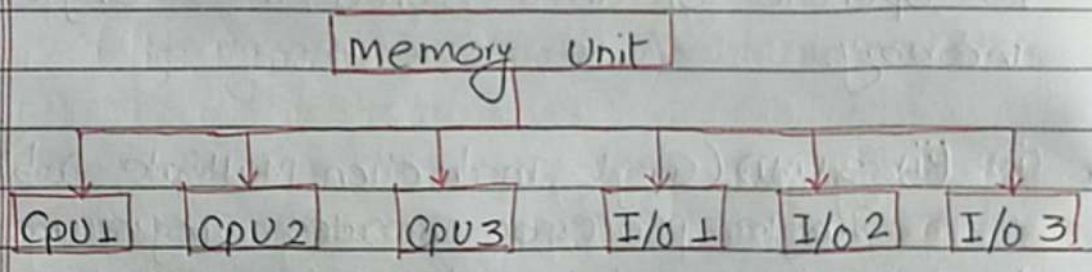


fig: Time-shared Common Bus Configuration).

In this structure the two or more CPU two or more I/O module and memory unit has been interconnected by using the Common bus system. In this architecture the dividing of timing or sharing mechanism take place.

Page :
Date : / /

(ii) Multiport memory :-

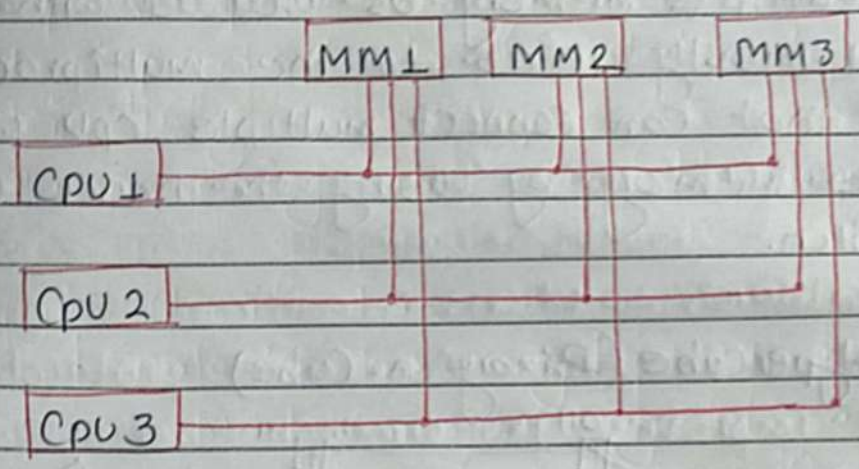


fig:- Multiport memory (MM) Configuration)

In this structure one CPU can connect to the multiport memory but not interconnected CPU to each other.

one CPU can interpret signals to the multiple memory by using the connection point called as code.

(iii) Cross - Bar switch :-

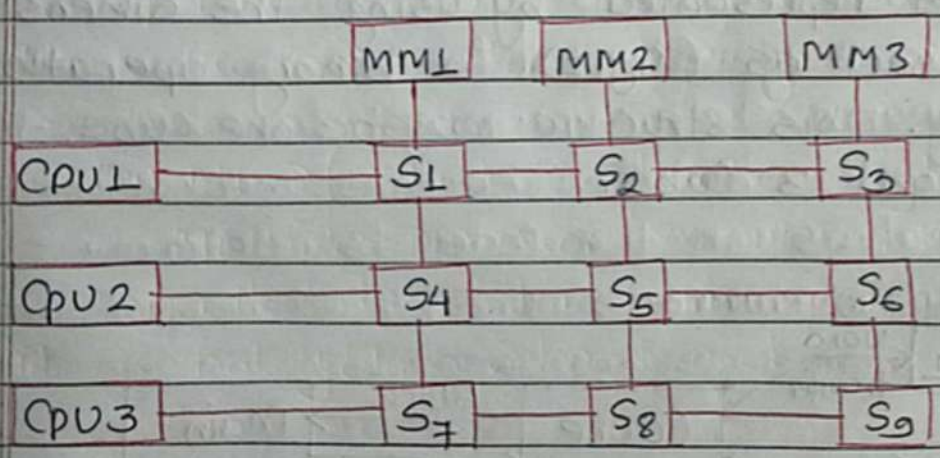
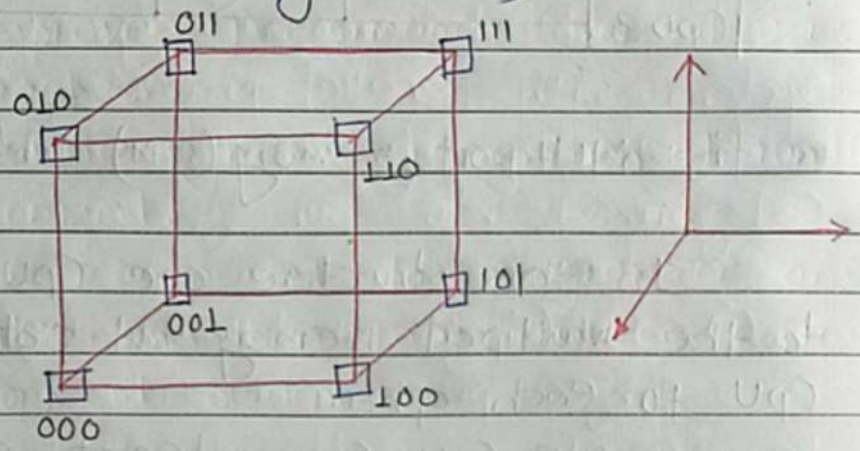


fig: Cross- bar switch Configuration.

In this structure single cpu can connect with multiple memory as well as with multiple cpu. It mostly refers to the multiprocessor system that can connect multiple cpu with multiple memory by using the nodes called as switch.

(iv) HyperCube (Binary n-Cube)



(fig: Hypercube structure)

This structure is very important in a multiple processor system in which connection can be represented by using the dimensional point such as (x,y,z) for binary operation directly. This structure may be one cube, two cubes and 3 cubes.

Cache Coherence:-

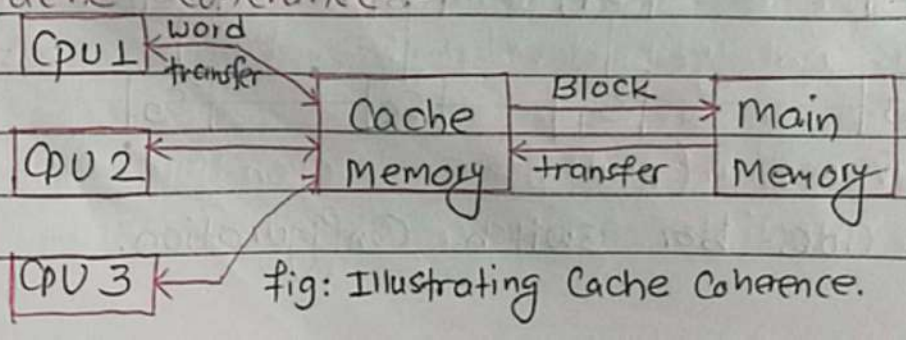


fig: Illustrating Cache Coherence.

Page :
Date : / /

In uniprocessor system, single processor have single cache memory mapping with the main memory.

The mapping techniques between cache memory and main memory has been occured easily rather than multi-processor system. In multiprocessor system, each processor have two or more level of cache. This organization is essential to achieve reasonable performance but due to the conflict in multi-processor system that creates some problems known as cache coherence.

In different cache, mutiple copies of same data exists. If the processor are allowed to update their own copies freely in consistent view of memory can result. The drawbacks to the bus organization in performance are :-

(i) To improve performance, each processor is equivalent with cache memory.

(ii) It reduces the number of bus access.

(iii) Each cache memory contains an image of a portion of main memory.
If a word is altered then it could be invalided in another cache memory level.

(iv) To prevent these problems the other processor must be used that update has taken place.

CamScanner

(v) This type of problem is known as Cache Coherence. It is addressed in hardware usually rather than software.

Solution for Cache Coherence:-

1) Hardware Solution:-
- It uses Cache Coherence protocol.
- It improves performance over software approaches.
- It provide dynamic recognition at runtime environment.
- It is transparent to programmer and compiles and then that reduces software development burden.

Hardware solution has two protocols i.e
(a) Snoopy protocol.
(b) Directory protocol.

2) Software Solution:-
- It is very simplest approach to prevent any shared data variables from being cached.
- Compiler based Coherence mechanism performs an analysis on the code to determine which data item becomes unsafe.

Page :
Date : / /

Introduction to Multithreaded Architecture :-

The approach which allows for a high degree of instruction level parallelism without increasing circuit complexity or decreasing power consumption for decomposing sub instructio into the number of stream is known as concept of multithreading.

Actually the instruction stream is divided into several smaller stream known as threads such that the thread can be executed in parallel. The variety of specific multithreading design realizes in both commercial system and experimental system is very fast.

The impartant measurement of performance for a processor is the rate in which it executes instruction. This can be expressed as: MIPS rate = $f * IPC$

where, f is the processor block frequency and IPC is the instruction per cycle.

Multithreading can be divided into two types.

(i) Implicit multithreading.
(ii) Explicit multithreading.

It will define the following components:-

→ process.
→ process switch.
→ Thread
→ Thread switch.

CamScanner

Page:
Date: / /

Approach of multithreading can be listed below:

(i) Superscalar.
(ii) Chip multi-processing.
(iii) Simultaneous multi-threading.
(iv) Block multithreading.
(v) Interleaved multi-threading.

CamScanner

Multicore Computer

Page :
Date : / /

Multicore Computer & multicore organization:-

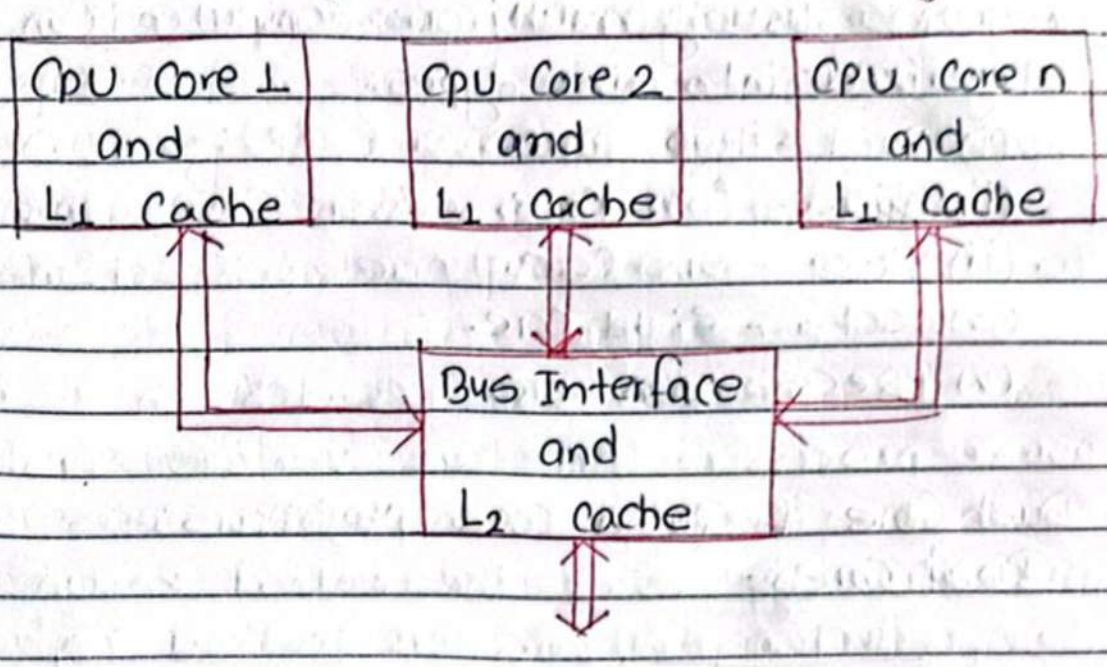


fig:- multicore organization

Each CPU Core is actually a seperate CPU which is the part of CPU that actually does the work. A multi-core processor is an integrated circuit (IC) to which two or more processor (chip or core) has been attached for enhancing performance, reduced power consumption and more efficient process of multiple task.

Multicore organization is a growing industry trend as single core processor that reduce the limitation of single core processing system.

A multicore computer is a single computer component with two or more independent cores which are the units that can be used to read and executes program and instructions.

Multicore can run multiple instruction at the

Same time that increases overall speed of programs usually multicore computer can be categorized into five types:

(i) Dual - two core.
(ii) Quad - four core.
(iii) Hexa - six core.
(iv) octa - eight core.
(v) Deca - Ten core.

Dual Core & Quad core processors:-

Dual Core:-

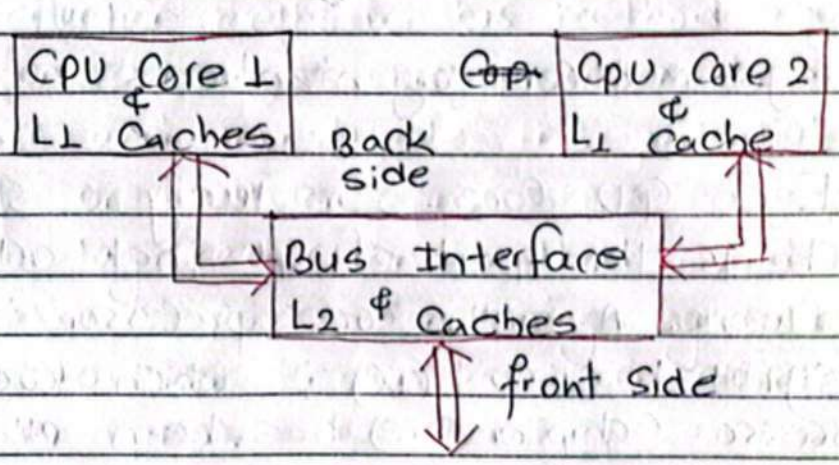


fig: Dual core processor

A dual core chip looks like a single CPU chip but it actually has two physical units called as dual core processor.

In this organization CPU has been participated to do multiple things at once. It has two seperate cores on the same die or chips each with its own cache memory. Actually, it has single micro-processor. It can often provide better performance rather than a

traditional single core processor. Today, dual core often have more capability for multitasking multiprocessing, multi-threading, audio and video, graphics and networking.

Quad core:-

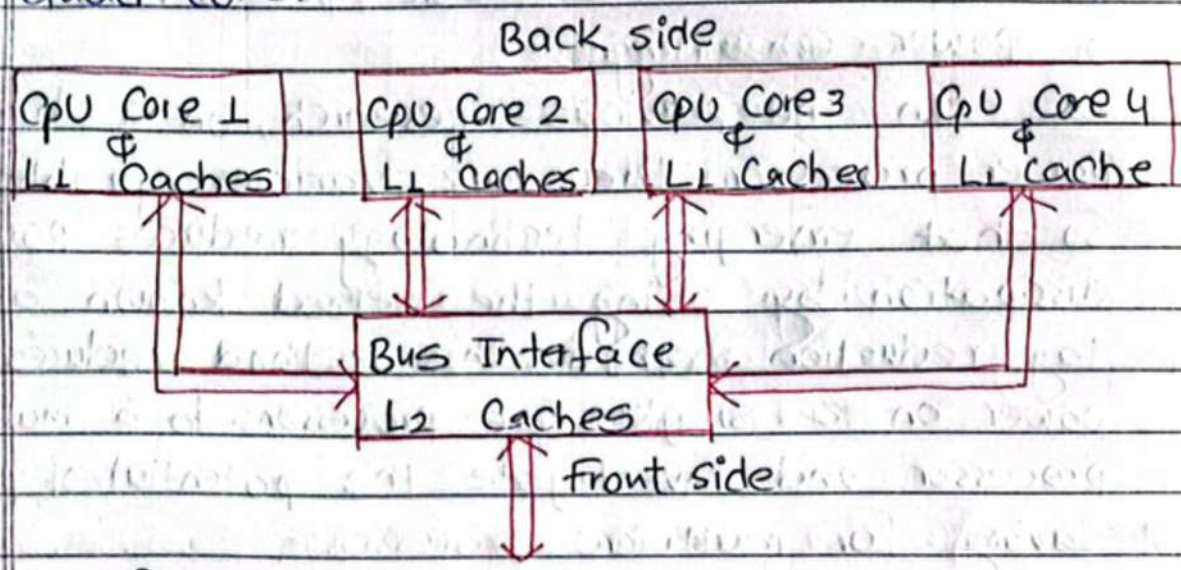


fig:- Quad Core processor

It is a chip with four independent units called quad cores. It reads and executes CPU instruction in parallel. The individual core in quad core processor can run multiple instructio at a same time. It has more benefit over dual core processor.

The benefits of quad core processor are as follows:

(i) faster multitasking.
(ii) HD quality gaming.
(iii) Low power consumption.
(iv) High performance for complex application.
(v) Faster execution

(vi) Faster operation.
(vii) Easy server configuration.

Power Efficient processor (PEP):-

power efficient processor saves the problem for caches on a multicore processor in case of power consuming.

In the previous research, on a high power processor there are various problems so that emerging technology reduces power dissipation by using the method known as tag reduction method. This method reduces the power on a single core processor to a multicore processor and investigate the potential of power saving for multicore processor.

Tag reduction method avoid the problems of power consuming by using certain extra processor known as power efficient processor (PEP).

Performance Issues in Multicore processor:-

Multicore computer has been invented with the principle of parallelism but it has some issues as listed below:-

(i) Hardware performance issues.
(ii) Software performance issues.

(i) Hardware Performance Issues:-

- Increase in cost.
- Increase in parallelism.

Page :
Date : / /

- power consumption.
- Alternative chips organization.
- Extra Hardware Component is required.

(ii) Software performance Issues:-
- Compatibility with software.
- Software on multicore processor must be specified.
- Unsuitable environment may affect on a software quality.
- platform dependent.

characteristics of memory:-

1) Location
- processor.
- Internal.
- External.

2) Capacity
- word size.
- No. of word.

3) Unit of transfer
- word.
- Block.

4) Access method.
- Sequential.

CamScanner

Page:
Date: / /

- Random.
- Direct.
- Associative.

5) physical tape
- Semiconductor
- magnetic
- flash.
- optical.

6) physical characteristics
- Volatile | Non-Volatile.
- Erasable | Non-Erasable.

SuperScalar Technology (SST):-

A Super-Scalar Technology is based on the Superscalar processors in which multiple independent instruction pipelines are used.

In Superscalar technology, multiple instructions are fetched at a time and then the processor attempts to find nearby instructions that are independent of one another and can therefore be executed in parallel. If the input to one instruction depends on the output of preceeding instruction, then the later instruction can't complete execution at the same time or before the former instruction.

The processor may eliminate such dependencies by the use of additional registers and renaming of register references in the original code.

CamScanner

Page :
Date : / /

Most of the Superscalar machines use traditional branch prediction methods to increase efficiency.

Differences between pentium PC and power-PC.

	pentium- pc	power- pc
(i)	It represents CISC architecture Concept.	(i) It represents RISC architecture Concept.
(ii)	It incorporates the Sophisticated design principle on Computers.	(ii) It is used in apple Machintosh machine & embedded Systems.
(iii)	It uses micro programmed control unit.	(iii) It uses hardwired control unit.
(iv)	The operation speed is Comparitively slower than power pc.	(iv) The speed of operation is faster than pentium pc.
(v)	All the operations are manipulated in memory.	(v) All the operations are done with the register of CPU.
(vi)	Less Security to data than power pc.	(vi) It provides exceptional security to data.
(vii)	Examples of pentium pc family are: Pentium III, Pentium IV, Itanium, etc.	(vii) Examples of power-pc family are:- 740/750, G4, G5, etc.

CamScanner

Structure of I/O module:-

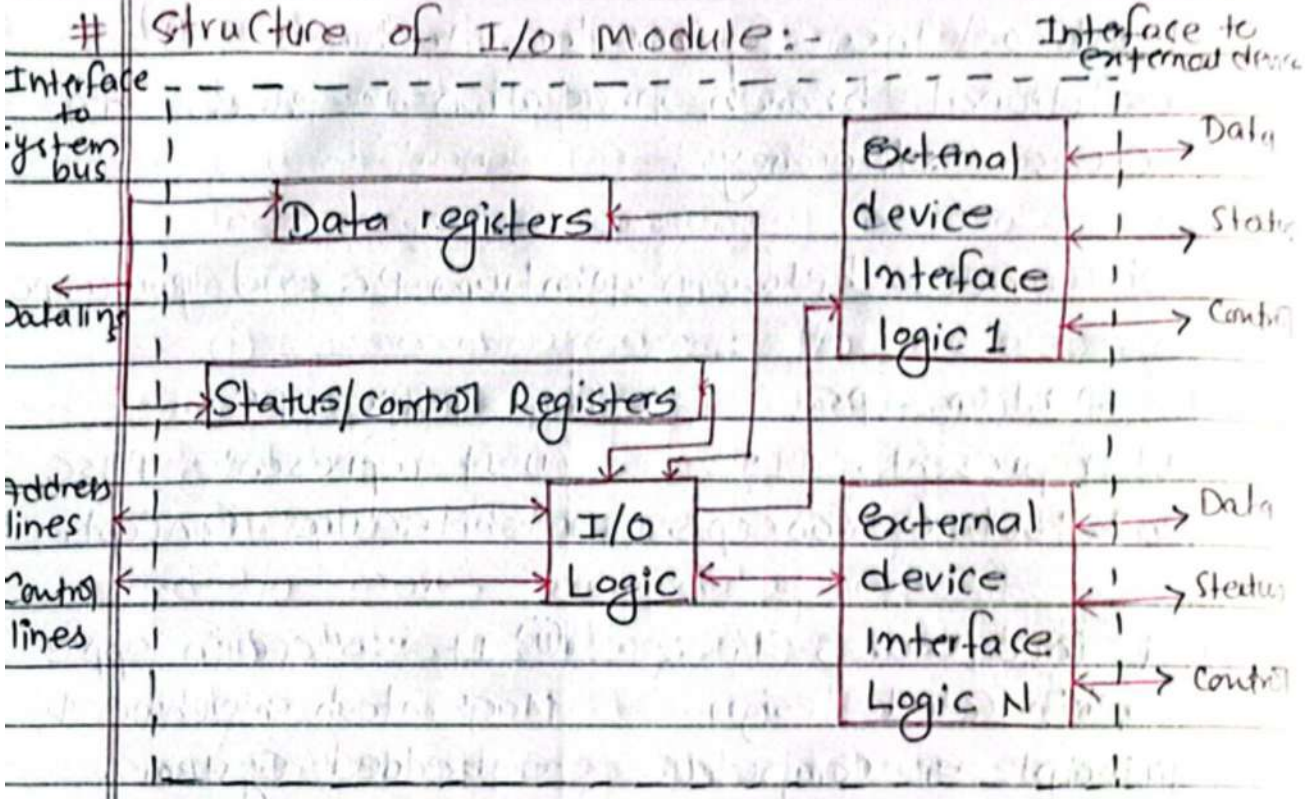


fig:- Block diagram of I/O module.

I/O module vary in Complexity and the number of external devices that they control. The above figure shows the block diagram of I/O module. The module connects to the rest of the Computer through a set of Signal lines.

Data transferred to and from the module are buffered in one or more data registers. There may be one or more status registers that provide Current Status information. A Status register may also function as Control register, to accept detailed control information from the processor.

The logic within the modules interacts

Page :
Date : / /

with the processor via a set of control lines. The processor uses control lines to issue commands to the I/O module. Each I/O module has a unique set of address to control more than one external device. The module must be able to recognize and generate addresses associated with the devices it controls.

Q.1 Why are external devices not connected directly with bus structure of computer system?

→ The external devices and bus structure are not directly connected because the internal resources such as main memory and the system bus must be shared among the number of activities including I/O data. Thus, I/O function include a timing and control requirement, to co-oidinate the flow of traffic between internal resources and external devices.

The basic steps involved during data transfer between external device to processor are :-

(i) processor interrogate I/O module to check status of attached device.
(ii) I/O module returns status of devices.
(iii) If device is operation and ready to transmit data, processor request the transmit of data.
(iv) I/O module obtains data from device.

CamScanner

(v) Data is transferred from I/O module to processor.

The processor - Communication involves the following:

(i) Command Decoding :- I/O modules accept Command from processor and send Signal on control bus.

(ii) Data :- Data are exchanged between processor and I/O moves over data bus.

(iii) Status reporting :- The status of I/O module is reported with status Signals Busy and READY. It may also include error Conditions.

(iv) Address Recognition :- I/O module must recognize one unique address for each peripheral it Controls.

Q.2. Are addressing modes and instruction format Co-related Justify.

→ An instruction format defines the layout of bits of an instruction. An instruction format must include an opcode and implicitly or explicitly, zero or more operands.

An explicit operand is referenced using one or more addressing modes. The format must, implicitly or explicitly indicate the addressing mode for each operand. For most instruction sets more than one instruction format is used. Depending upon the instruction format addressing mode is determined.

Page :
Date : / /

Instruction Cycle:

Fetch cycle:-

It occurs at the begining of each instruction cycle and cause the instruction to be fetched from memory. In this process 4 registers are involved.

(i) MAR : Specifies address in memory for read and write operation.

(ii) MBR : It contains value to be stored in memory or last value read from memory.

(iii) PC :- Holds next instruction's address to be fetched.

(iv) IR :- Holds last instruction fetched.

Execute Cycle:-

The execute cycle takes many forms; the forms depends on which of the various machine instructions is in IR.

The cycle involves transferring of data among registers, read or write from memory or I/O and the invocation of ALU. This cycle interpret the opcode and peform indicated operation.

Interupt Cycle:-

At the Completion of the execute cycle, a test is made to determine whether any interupts had occured. If so, the interupt

Cycle occurs. If the interrupts are enabled and and interrupt has occured then save the current process state and service the interrupt.

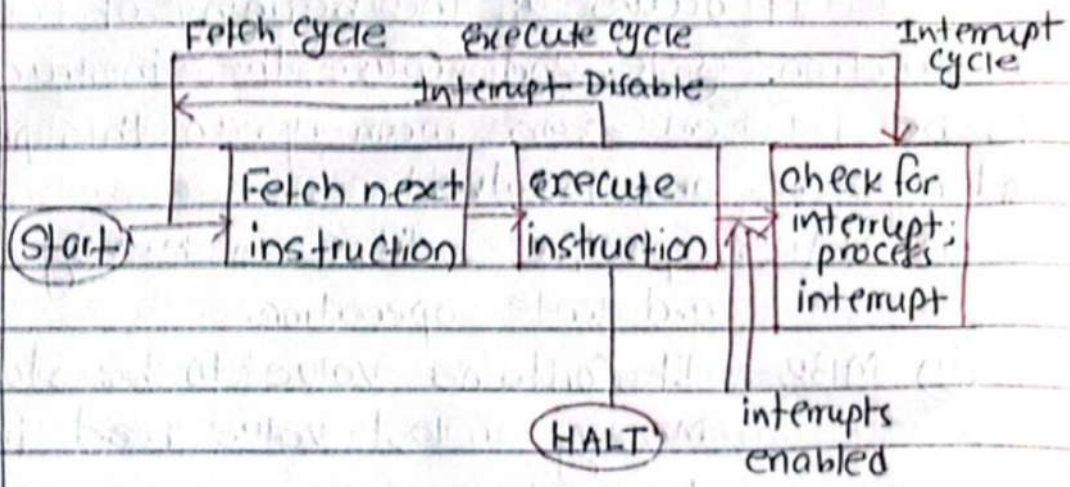
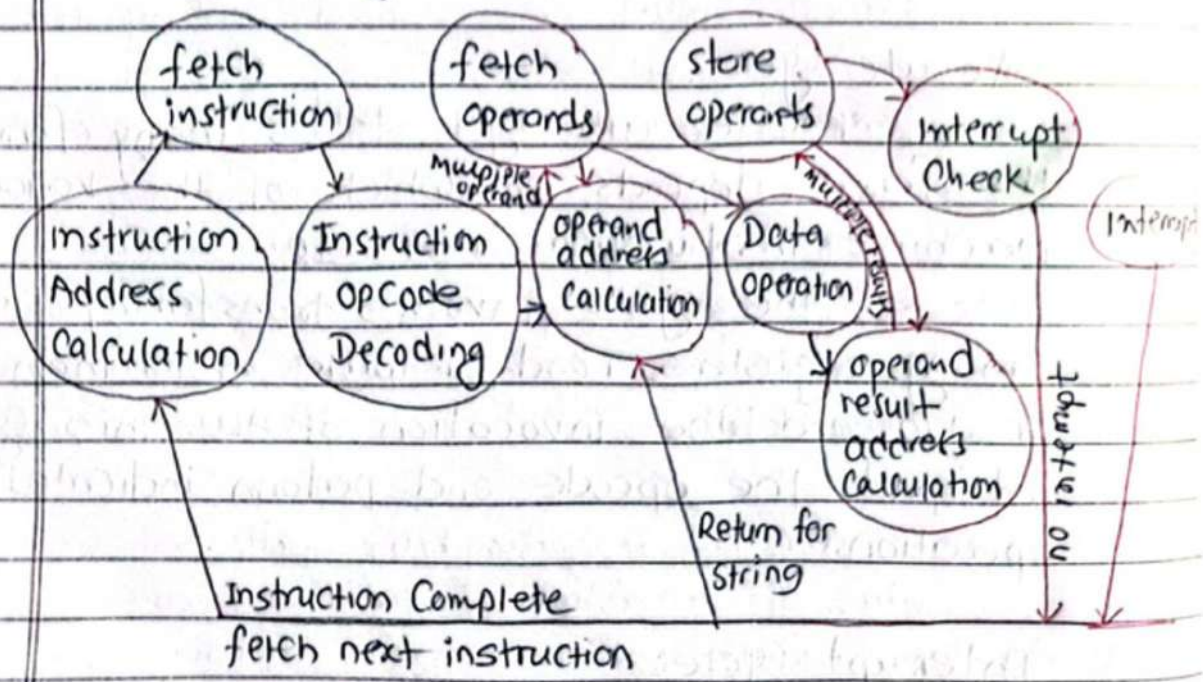


fig: Instruction Cycle.

Instruction cycle State Diagram with Interrupt



(i) Instruction Address Calculation
(ii) Instruction fetch.
(iii) Instruction opcode Decoding.

Page :
Date : / /

(iv) operand Address Calculation.
(v) operand fetch.
(vi) Data operation (Execute).
(vii) Result operand Address Calculation.
(viii) Store result operand.
(ix) check for interupts.
(x) No interupts (go to step (i) for
(xi) If interupts handle interupt & go to step(i).

Good Luck
Best Wishes

By:
Ram Poudel
Bhuwan Paneru

CamScanner