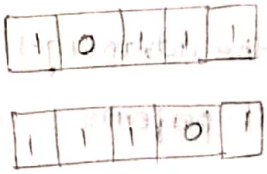
Computer Organization and Architecture Dt:-9/7/11

Computers- Computer operating machine particularly used technology education and research.

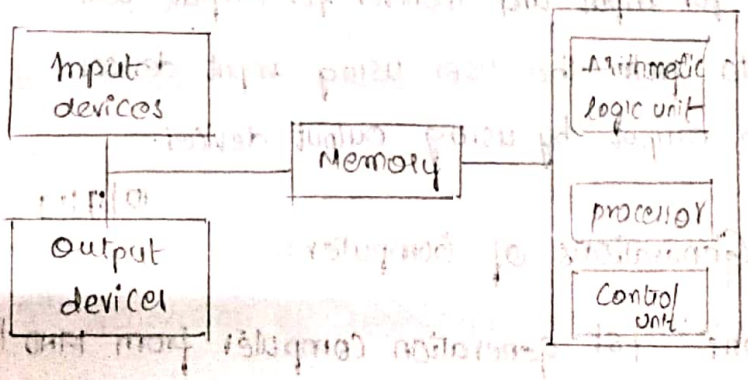
Memory:- It is a collection of registers. Registers are the collection of flip-flops and flipflops are the coulection of logic gates.

Memory — flipflops
Registers — D flip-flop



Computers- A Computer is a machine of common purpose can be programed to carryout a set of pre-determined arthematic and logic functions. It will take the set of instructions commonly called programs and execute them.

Basic structure of a Computer:-



Input devices:-

1) keyboard 2) mouse 3) Scanner
4) Web scanner 5) microphone 4) USB.

Output devices:-

1) Monitor 2) Speaker 3) printer
4) projector

→ Computer Organisation:- It is concerned with the way the hardware components operate and the way they are connected together to form a computer system.

→ computer Architecture:-

It is concerned with the structure and behaviour of the computer. It includes the information formats, instruction set and techniques for addressing memory.

① The central processing unit by taking the advantages of arithematical logic unit, control unit performs the computers computational and logical functions.

② Memory: memory place a crutial role in the computer to store the data and also ~~to~~ transfer data between registers.

③ I/p and o/p devices are man machine interfaces such as keyboard for input and monitor for output will takes the data from the user using input devices and gives the output by using output devices.

10/7/19

* Evolution and Generations of Computer:-

1. 1st Generation:- 1st Generation computer from 1940 to 1956 the 1st Generation of computers used vaccum tubes as a major technology, because of their larger size the computer taking up a lot of space in a room or entire room.

Ex:- ENIAC (Electrical Numerical Integrator and Calculator)

In this computer 20,000 vaccum tubes where used and it weighs 30 tones.

2) 2nd Generation from 1956 to 1963.

It is used transistors as a major technology and resulted a computer which occupies a reduced space and much costes.

Ex:- IBM 7070.

3) 3rd Generation from 1964 to 1971

In third generation IC (Integrated circuit) was introduced by Jack Kilby (UK), helped reducing the size of computers even more and as well as made them faster.

Ex:- IBM 230.

4) 4th Generation from 1972 to 2010.

The fourth generation of computers took the advantage of invention of microprocessor mostly ~~use~~ called cpu. Microprocessor along with ICs helped making possible for computer fit easily on a desk and also introduction of laptop

Eg: IBM 5100.

5) 5th Generation of computers are ~~begging~~ is beginning to use Artifical Intelligent (AI) which is an exiting technology that has many potential applications around the world.

Ex:- IBM Watson.

Instructions:- A computer instruction is a binary code that specifies a sequence of micro operations.

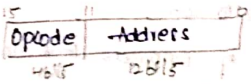
Program:- A program is a set of instructions. The user of a computer can control the program by means of the program.

Instruction program:- An instruction program is a group of bits that instruct the computer to perform specific operations.

It is divided into two parts

1. 1st part stores opcode
2. 2nd part stores address of data (operand)

The total instruction is 16 bits



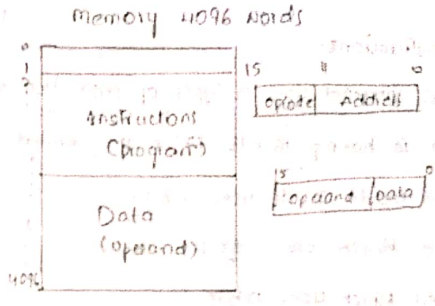
Instruction code.

Instruction code together with data are stored in memory the computer reads each instruction from memory and places it in control register. The control then interprets the binary code of the instructions then proceeds to Execute.

→ Stored program organization:-

→ The simplest way to organize the computer is to have a processer register and an instruction code.

→ Instructions are stored in one section of memory, data is stored in another section of memory.

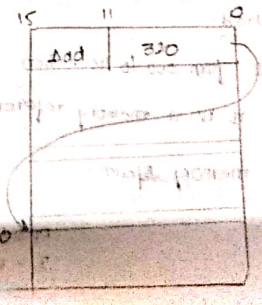


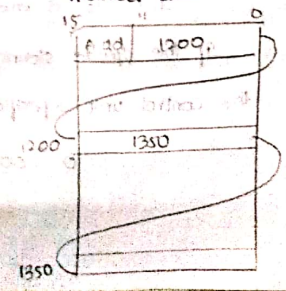
→ for a memory unit having 4096 words, we need 16 bits to specify the address of the location.

→ As we know instructions are stored in the memory. now the control unit reads each instruction and specifies the type of operation to be performed and also specifies the address of the data (operand) then the control unit executes the instructions.

→ Difference between direct and indirect address.

Direct address	Indirect address
When the second part of the instructions code specifies the address of the operand then it is called a direct address	When the second part of the instruction code specifies the address of the memory word in which address of operand is found then it is called indirect address.





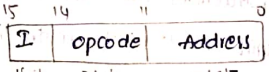
11/07/19

Computer Instructions-

The basic computer uses 3 types of instruction code formats. Every format is having 16 bits which are divided into 3 parts.

1) Address mode I it uses 1 bit.
2) Opcode which uses 3 bits.
3) Address which uses 12 bits.

Based on address mode bit I the control decides whether it is a direct address (or) Indirect address



When I=0 ⇒ Direct address
I=1 ⇒ Indirect address

Based on these address mode bits we are having 3 types instruction formats -

→ memory reference Instructions
→ Register reference Instruction
→ Input / output Instruction

1) Memory Reference Instructions-

It uses 12 bits to specify address & 1 bit to specify address mode.

If I=0 it is direct address
I=1 it is indirect address.

If the opcode storing 3 bits from 000 to 110. then the control unit specifies that it is a memory reference.

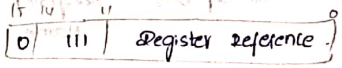
opcode - 000-110
I = 0 (or) 1

2) Register reference Instructions-

It also uses 16 bits to specify register operation i.e., 1bit for address mode 3 bits for opcode & 12 bits for register operation.

The control unit reads the opcode & if it is 111 then it will reads the address mode.

If address mode = 0 ⇒ Register reference Instruction

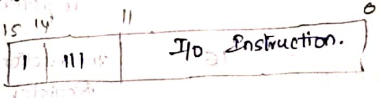


opcode - 111
I = 0

3) Input/output Instructions-

It also uses 16 bits. It uses 12 bits to specify I/O operation 3 bits for opcode & 1 bit for address mode.

If opcode is 111 & address mode is 1. then it is called I/O instruction.



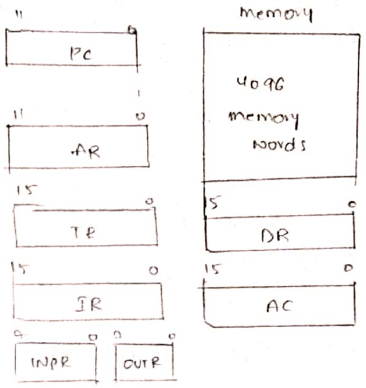
opcode = 111
I = 1

→ Computer Registers (or) Register
→ The control unit reads an instruction from a specific address in memory and executes it.
It then continues by reading the next instruction in sequence and executes it, and so on.
→ This type of instruction sequencing needs a counter to calculate the address of the next instruction after completion of the current instruction.
→ It is also necessary to provide a register in control unit for storing instruction code after it is read from memory

→ The computer needs the a processor register for manuplating the data and register for holding memory address these requirements given rise to a set of 8 registers inside the computer.

List of registers:-

	Register symbol	No. of bits	Register Name	function.
1)	DR	16	Data register	It holds memory operand.
2)	AR	12	Address Register	It holds the address of the operand.
3)	AC	16	Accumulator (processor register)	It process the data
4)	IR	16	Instruction Register	It holds instruction code.
5)	PC	12	program counter	It holds the address of the next instruction to be read.
6)	TR	16	Temporary register	It holds temporary data
7)	INPR	8	Input register	It holds the input character
8)	OUTR	8	Output register	It holds the output character



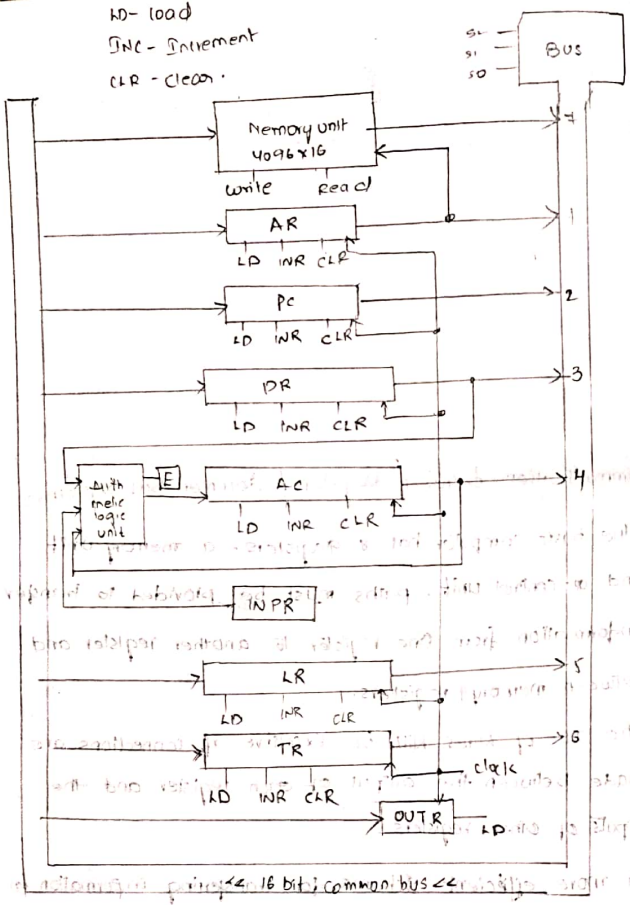
Communication between Registers / Common bus System:-

→ The basic computer has 8 registers, a memory unit and a control unit. paths must be provided to transfer information from one register to another register and between memory & registers.

-the no. of wires will be excessive if connections are made between the output of each register and the inputs of other registers.

→ A more effecient scheme for transferring information in a system with many registers is to use a common bus.

LD- load
INC - Increment
CLR - Clear.



→ The outputs of 7 registers and the memory are connected to common bus the specific output that is selected for the bus at any given time is determined by the binary value of the selection inputs S_0, S_1, S_2.

→ The number along each output shows the decimal equivalent of the required binary selection.

Ex:- the number along the output of direct register is 3, now the 16 bit output of DR placed on the bus. When $S_0, S_1, S_2 = 0, 1, 1$

→ The particular register whose load input (LD) is enabled receives the data from the bus.

→ the particular register whose input clear input (CLR) is enabled will erase the data.

→ When $S_0, S_1, S_2 = 1, 1, 1$ then the read input of memory is activated.

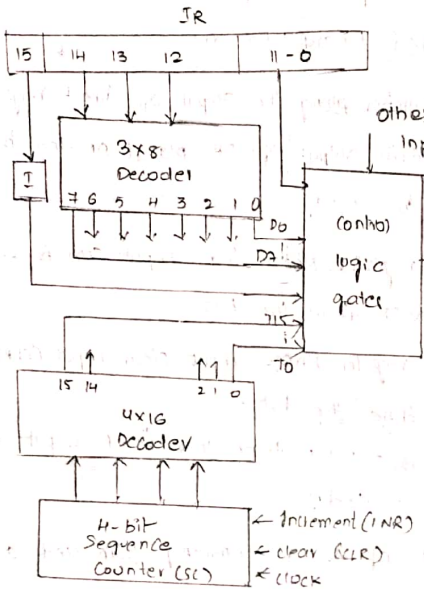
→ The 16 inputs of AC are comming from and adder at logic circuit.

→ This circuit has 3 set of inputs one input is coming from output of AC, another input is coming from output of data register. and 3rd input is coming from input register. In this way all registers and memory inside the compoles will communicate by using this common bus system.

Timing and control:-

→ It consists of two decoders a sequence counter and a number of logic gates. An instruction read from memory is placed in the register IR where it is divided in to three parts.
i) addres mode (I) ii) opcode iii) Address bits (0-11)

→ the control unit is decoding every part of the instruction and generates the required signals.



→ The 3 bits in opcode gives to 3x8 Decoder where 8 signals i.e., D0 to D7 will be generated and send to control logic gates.

→ 15 bit is transferred to a flipflop I from there it also goes towards the control logic gates.

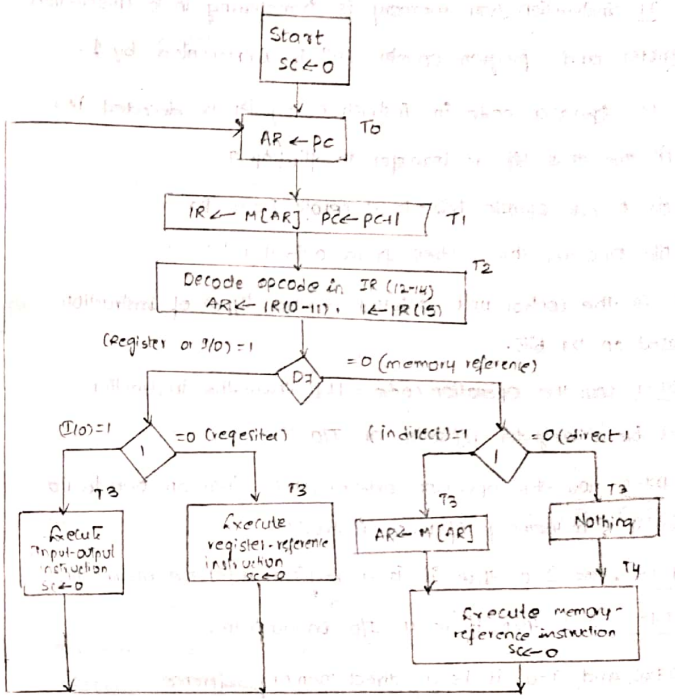
→ The address bits from 0 to 11 also transferred to control logic gates

→ the 4bit sequence counter can count from 0 to 15 and generates 16 timing signals from T0 to T15.

→ The sequence counter can be implimented or cleared synchronously for example SC is incremented to provide signals T0, T1, T2, T3 in sequence at time T_4 SC is cleared to zero then the Sequence counter will returns to T0.

Date: 04/11/19

Instruction Cycle:-



→ The above is the algorithm for instruction cycle through which every instruction must pass in order to complete the execution.

→ Initially the program counter is loaded with address of the first instruction the sequence counter is clear to zero providing a decoder signal for each clock pulse to count the number of instructions or to increment the program counter.

→ At T_0 address in program counter transfered to the Address register.

→ At T_1 instruction from memory is transferring in to instruction register and program counter will be incremented by 1.

→ At T_2 operation code in instruction register is decoded ie.,

(i) The 15th bit is transfer to flipflop I.

(ii) Decode opcode bits to generate D_0 to D_7.

(iii) Decodes the address from (0 to 11) bits.

→ At T_3 the control unit determines the type of instruction based on D_7 bits.

→ If $D_7 = 1$ now the operation code = 111 then the instruction must be a register reference or I/O.

→ If $D_7 = 0$ now the operation code must be between 000 to 110 then it is a memory reference instruction.

→ If $D_7 = 1$, $I = 0$ then it is a register reference instruction.

→ If $D_7 = 1$, $I = 1$ then it is a I/O instruction.

→ If $D_7 = 0$ and $I = 0$ it is a direct memory reference

→ If $D_7 = 1$ and $I = 1$ it is a indirect memory reference.

→ After executing the current instruction at T_4 then the control goes to T_0

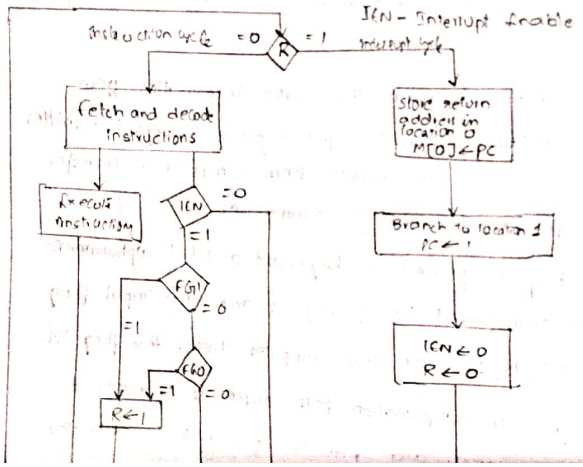
→ The output register works similarly but the direction of information is reversed. Initially the o/p flag is set to '1'. Now the computer checks the flag bit. If it is one the information is transferred from Ac to OUTR, and FGO is clear to '0'.

→ Now the output device accepts the information & prints the corresponding character. & When the operation is completed FGO is set to 1.

→ Interrupt cycle:-

When a process is executing inside the computer by the cpu & when user requests another process then this will create disturbance for the running process, this is called Interrupt.

R- Interrupt flipflop
IEN- Interrupt Enable



An alternative to the program controlled procedure (computer keeps checking the flag bit, and when it is set it initiates an information transfer. like this the computer is wasting time while checking the flag instead of use doing some other useful task)

It's to let the external device inform the computer when it is ready for the transfer. In the mean time the computer can be busy with some other tasks. These type of transfer uses the interrupt facility. It uses a flipflop IEN which can be cleared to 0 or set to 1.

→ From the above flow chart an interrupted flipflop (R) is included in the computer. When R=0 the computer goes through an instruction cycle. During the execution face it checks the IEN flipflop → If IEN=1 the control checks the flag bits if both flags are 0 it indicates no interrupt and control goes to the execution of next instruction.

→ If IEN=0 it indicate there is no interrupt it goes to next instruction.

→ If IEN=1, either flag is 1 then it sets flipflop R to 1. and goes to interrupt cycle.

31/8/19

→ Micro programmed control:-

→ The major building blocks of a computer are cpu, memory and I/O devices.

→ The digital hardware functional units inside the cpu are ALU, CU and Registers.

→ The complexity of digital computer based on the number of micro operations that a control unit is under taken.

→ there are two methods of implimenting control unit

1) Hardwired control
2) Micro programmed control.

Difference b/w Hardwired and Micro programmed control

Hard wired control	Micro programmed control
1) uses fixed number of instructions.	1) We can change the Size of instructions
2) fixed logic blocks Ex:- encoders, decoders	2) We can change the logic blocks.
3) High speed operation	3) Slow compared to hardwired control
4) Design is very Expensive Ex:- INTEL 8085,	4) Design is cheap/less when compared to hard-wired control Ex:- INTEL 8051
5) It belongs to Risc -Reduced Instruction set computer	5) It belongs to CISC Complex Instruction set

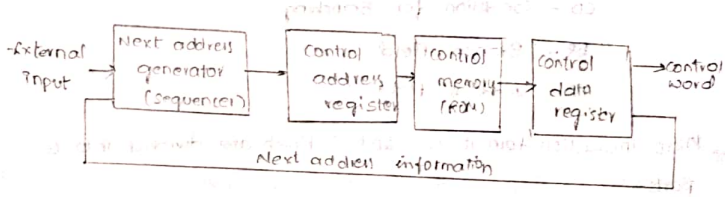
→ A control unit whose binary control variables are stored in memory is called a micro programmed control unit.

→ Each word in control memory contains a micro instruction.

→ The micro instruction specifies one or more micro operations for the system, the sequence of these micro instructions constitutes a micro program.

→ A Computer that employes a micro programmed control unit. We have two seperate memories. i) main memory
ii) control memory

→ Micro programmed control organization:-



→ The above is the general configuration of a micro-programmed control unit. In which the control memory is assumed to be a ROM, within which all control information is permanently stored. 1/9/19

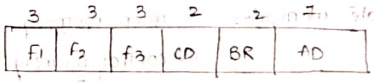
→ The control address register specifies the address of the micro instruction. control data register holds the micro instruction reads the memory.

→ The micro instruction contains a control word that specifies one or more micro operations for the data processing

→ Once these operations are executed the control must determined the next address. The next address is computed in the address generator circuit and then transfer to control address register.

→ The next address generator is sometimes called micro-program sequencer.

→ Micro instruction format:-



F1, F2, F3 — Micro operation fields

CD - Condition for Branching

BR - Branch field

AD - Address field.

→ Micro Instruction format uses 20 bits which are divided into 6 parts.

→ first three parts F1, F2, F3 are micro operation fields based on these control unit generate micro operations.

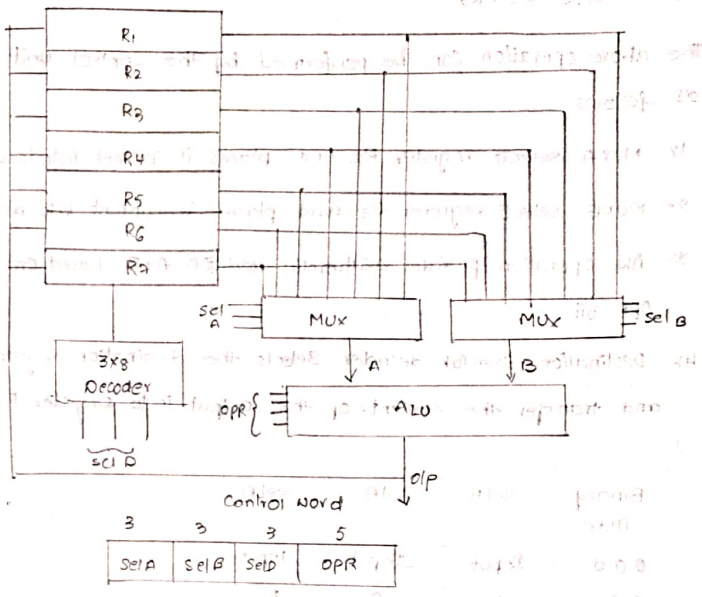
→ CD field selects the status field for conditioning.

→ BR field selects the branch to be used.

→ AD field is used to specify branch address.

2/08/19

→ General Register organisation:-



→ The above is the bus organization for 7 cpu register. The o/p of each register is connected to 2 multiplexers to form two buses A and B

→ The selection lines in each multiplexer select one of the A and B buses form the inputs to a common ALU.

→ The Operation selected in ALU determines the arithemic or logical operations that is to be performed.

→ The register that receives the information from output bus is selected by a decoder.

→ The decoder activates one of the register providing a transfer path between the data in the output bus and the inputs of

the selected register.

Ex:- $R_1 \leftarrow R_2 + R_3$

The above operation can be performed by the control unit as follows.

1) MUX A selects register R_2 and places it content into bus A.
2) MUX B selects register R_3 and places it content into bus B
3) ALU operation provides arithmetic addition A+B based on OPR bits.
4) Destination selector decoder selects the destination register and transfer the content of the output into Register R_1.

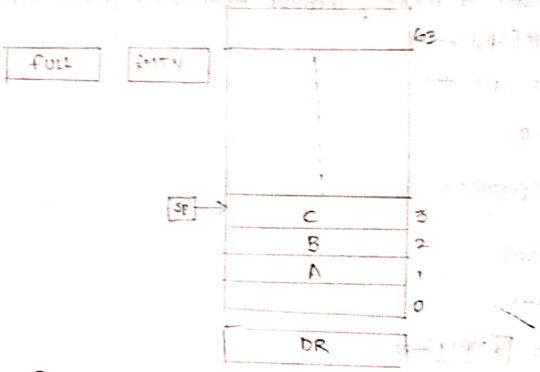
Binary code	SelA	SelB	SelD
000	Input	Input	Input
001	R_1	R_1	R_1
010	R_2	R_2	R_2
011	R_3	R_3	R_3
100	R_4	R_4	R_4
101	R_5	R_5	R_5
110	R_6	R_6	R_6
111	R_7	R_7	R_7

Table is decoding operation bits :-

OPR	operation
00000	Transfer A
00001	Increment A
00010	ADD A+B
00101	Subtract A-B
00110	Decrement A
01000	AND A&B
01010	OR A&B
01100	XOR A&B
01110	Complement A

Diagram

→ Stack Organisation:



SP - Stack pointer which holds the address of the memory location.

DR - Data register

FULL - It is also a register which own 1 bit and sets to 1 when stack is full.

EMTY - It is a 1 bit register which sets to 1 when stack is empty

→ Stack is a storage device that stores the information in such a manner that the item stored last is the first item retrived.

→ Two operations of a stack are :-

- → Insertion
- → Deletion

→ The operation of insertion is called push down or push

→ The operation of deletion is called pop up or pop

Steps for push operation :-

1) stack pointer will be incremented to 1.
   i.e., SP ← SP+1

2) Data register to memory location when stack pointer is M
   M[SP] ← DR

3) If SP = 63 full ← 1

4) EMTY ← 0

steps for pop operation :-

1) DR ← M[SP]

2) SP ← SP-1

3) if SP=0 EMPTY → 1

4) FULL ← 0

8/8/19

Instruction format :-

A computer usually have variety of instruction code formats. It is the function of control unit to interrupt is instruction code and provide memory control function to procell the instruction.

The bits of instruction are divided into roots are field.

The number of addres fields in the no. of addres fields of a computer depend on the organization are it register must computers fall in thrugh three types of cpu organisation

1) single accumulator organisation
2) General register organisation
3) stack organisation.

* Instruction format for single accumulator organisation:

→ The instruction format in this type of computer uses one addres field.

Ex:- The instruction that specifies an arithmetic addition is defined by an allembly language instruction as follows.

1) ADD X

   AC ← AC + M[X]

2) MUL Y

   AC ← AC * M[Y]

3) SUB Z

   AC ← AC - M[Z]

* Instruction format for general Register Organization:-

1) ADD R_1, R_2, R_3

$R_1 \leftarrow R_2 + R_3$

2) ADD R_1, R_2

$R_1 \leftarrow R_1 + R_2$

3) MOV R_1, R_2

$R_1 \leftarrow R_2$

4) ADD R_1, X

$R_1 \leftarrow R_1 + M[X]$

* Instruction format for stack organisation:-

Computers with stack organisation would have PUSH and POP instructions which require an address field.

1) PUSH X

Where X is the address of the data in data register DR.

2) POP Y

→ Three address instructions:-

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

ADD R_1, A, B	$R_1 \leftarrow M[A] + M[B]$
ADD R_2, C, D	$R_2 \leftarrow M[C] + M[D]$
MUL X, R_1, R_2	$X \leftarrow R_1 * R_2$

Data Transfer and Manipulation:-

→ Computers provide an extensive set of instructions to give the user the flexibility to carry out various computational tasks.

→ The instructions set of different computers differ from each other in the way the operands are determined from the address fields.

→ Most computer instructions classified into three categories.

1) Data transfer instructions
2) Data Manipulation instructions
3) Program control instructions.

1) Data Transfer Instructions:-

Data transfer instructions will transfer data from one location to another without changing the binary information content.

Name	Mnemonic
Move	MOV
Load	LD
store	ST
Exchange	XCH
Push	PUSH
Pop	POP
Input	IN
output	OUT

* Instruction format for general Register Organization:-

1) ADD R_1, R_2, R_3

$R_1 \leftarrow R_2 + R_3$

2) ADD R_1, R_2

$R_1 \leftarrow R_1 + R_2$

3) MOV R_1, R_2

$R_1 \leftarrow R_2$

4) ADD R_1, X

$R_1 \leftarrow R_1 + M[X]$

* Instruction format for stack organisation:-

Computers with stack organisation would have push and pop instructions which require an address field.

1) PUSH X

Where X is the address of the data in data register DR.

2) POP Y

→ Three address instructions:-

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

ADD	R_1, A, B	$R_1 \leftarrow M[A] + M[B]$
ADD	R_2, C, D	$R_2 \leftarrow M[C] + M[D]$
MUL	X, R_1, R_2	$X \leftarrow R_1 * R_2$

Data transfer and Manipulation:-

→ Computers provide an extensive set of instructions to give the user the flexibility to carry out various computational tasks.

→ The instructions set of different computers differ from each other in the way the operands are determined from the address fields.

→ Most computer instructions classified into three categories.

1) Data transfer instructions
2) Data Manipulation instructions
3) Program control instructions.

1) Data Transfer Instructions:-

Data transfer instructions will transfer data from one location to another without changing the binary information content.

Name	Mnemonic
Move	MOV
Load	LD
store	ST
Exchange	XCH
Push	PUSH
Pop	POP
Input	IN
output	OUT

→ The load instruction used to transfer data from memory to processor register (Accumulator).

→ store instruction used to transfer data from accumulator to Memory

→ Move instruction used to transfer (move) data from one register to another.

→ the Exchange instruction used to swap the data from two registers.

→ PUSH and POP transfer data from between processor register to memory stack.

→ IN|OUT - transfer data from processor register to I/O terminal.

21/8/18

2) Data Manipulation Instructions:-

Data manipulation Instruction perform operations on the data (operand) and provide computational capabilities to the computer.

for a typical computer these instructions classified into three categories

1) Arithmetic Instructions
2) Logical Instructions
3) Shift Instructions.

1) Arithmetic Instructions:-

Name	Mnemonic
Add	ADD
Subtract	SUB
Multiply	MUL
Divide	DIV
Increment	INC
Decrement	DEC
Add with carry	ADDC
subtract with Borrow	SUBB

2) Logical Instructions:-

Name	Mnemonic
And	AND
Or	OR
Exclusive OR	XOR
Exclusive NOR	XNOR
clear	CLR
clear carry	CLRC
Compliment	COM
Compliment Carry	COMC

3) Shift Instruction:-

Name	Mnemonic
Shift Right	SHR
Shift left	SHL
Rotate Right	ROR
Rotate Left	ROL
Rotate Right through carry	RORC
Rotate Left through carry	ROLC

3) program Control Instructions:-

→ After execution of transf. data transfer or data manipulation control returns to fetch cycle of next program based on the program counter status.

→ While the control executing instructions in a sequence, based on the user requirement the control will jump to the another instruction which is not in sequence.

→ In order to make this we need to change the value in the program counter.

→ finally in order to break the sequence of execution of instructions we need to execute a program controlled instruction.

Name	Mnemonic
Branch	BR
Jump	JMP
Skip	SKP
Call	CALL
Compare	CMP
Test	TST
Return	RET

imp

Characteristics / features of CISC:-
[Complex Instruction set computer]

1) uses a large number of instructions typically from 100 to 250.
2) uses large number of addressing modes.
3) variable length instruction formats.
4) Instructions manipulate operands in memory
5) Multiple cycle instruction execution.

Characteristics / features of RISC:-
[Reduced Instruction set computer]

1) Relatively few instructions.
2) Relatively few addressing mode
3) fixed length instruction format
4) single cycle instruction execution
5) operations done with in the registers.

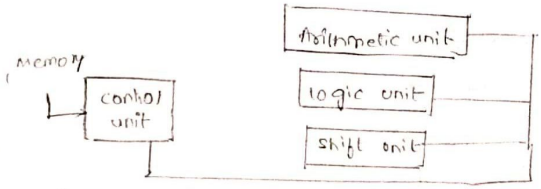
CISC	RISC
1) It uses multicycle execution.	1) It uses single cycle execution.
2) It uses large number of addressing modes	2) less number of addressing modes.
3) It uses variable length instruction format	3) It uses fixed length instruction format
4) It uses more number of bits to represent a instruction	4) It uses less number of bits to represent a instruction
5) operation done inside the memory	5) operation done inside the registers

→ Parallel processing:-

porallel processing is a term used to denote a large class of techniques that are used to provide simultaneous data processing to increase speed of a computer.

parallel processing is established by distributing the data among multiple functional unit.

Ex:- Computer having seperate arithmatic, logic unit, shift unit



there are 8 types of parallel processing techniques

i) pipeline processing

ii) vector processing

iii) array processing

i) pipe line processing:

It is a technique of decomposing a sequential process into sub operations, with each subprocess being executed in a special dedicated segment that operates simultaneously with all other segments.

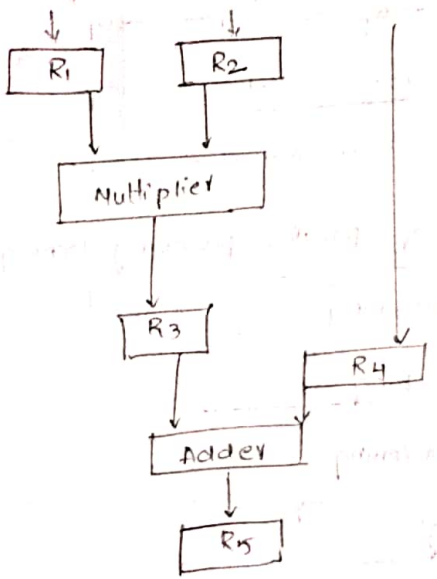
Ex:- $A_i * B_i + C_i$ for $i = 1, 2, 3, 4 \cdots$

$R_1 \leftarrow A_i$

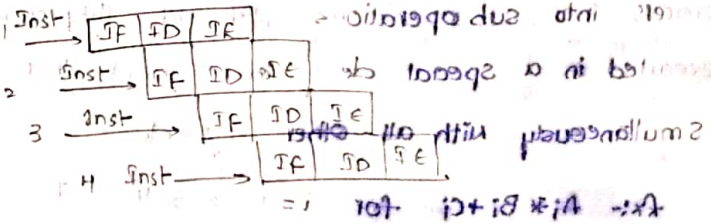
$R_2 \leftarrow B_i$, $R_4 \leftarrow C_i$

$R_3 \leftarrow A_i * B_i$

$R_5 \leftarrow R_3 + R_4$



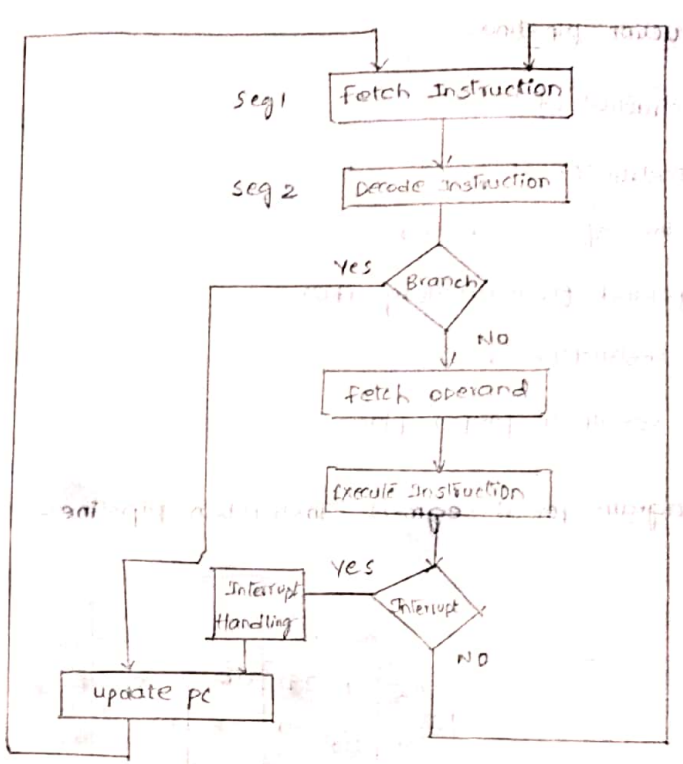
→ Instruction pipeline:-



* Space time diagram for 3 segment pipe-line:-

1	T1	T2	T3			
2		T1	T2	T3		
3			T1	T2	T3	
4				T1	T2	T3

An instruction pipeline reads consecutive instructions from memory while previous instructions are being executed in other segments. This causes the instruction fetch & execute phase to overlap & perform simultaneous operations.



Space diagram for four segment Instruction pipeline:-

Seg-1	T1	T2	T3	TH			
Seg-2		T1	T2	T3	TH		
Seg-3			T1	T2	T3	TH	
Seg-4				T1	T2	T3	TH

→ H segment instruction pipeline

The H segments of instruction pipeline are

a) fetch instruction (FI)

b) Decode instruction (DI)

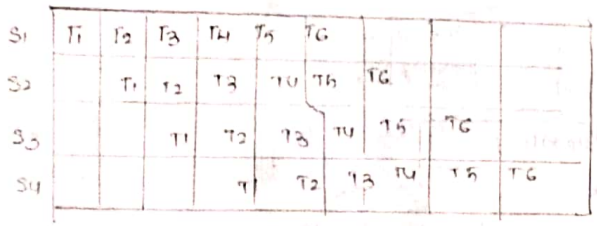
c) fetch operand (FO)

d) Execute the instruction (EX)

6-Segment Instruction pipeline:-

1. Fetch Instruction (FI)
2. Decode Instruction (DI)
3. Calculate the effective address
4. Fetch operand from memory (FO)
5. Execute Instruction (EI)
6. Store the result in proper place

Space time diagram for 6-segment Instruction pipeline:-



unit-III

Dt:- 5/09/19

Input | output organisation:-

→ I/O Interface:-

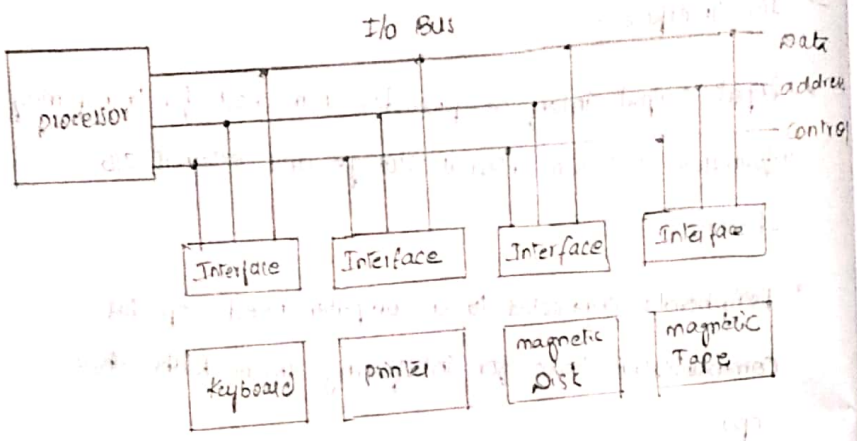
- Input output interface provides a method for transferring information between internal storage and external I/O devices.
- Peripherals connected to a computer need special communication links for interfacing them with the cpu.
- The data transfer rate of peripherals is usually slower than the transfer rate of cpu, so synchronisation mechanism is needed.
- The operating modes of peripherals are different from each other and each must be controlled so as not to disturb the operation of other peripherals.

→ Interface:-

- Computer systems include special hardware components between cpu and peripherals to supervise and synchronise all input and output transfers. This components are called interface units.
- The main function of I/O interface are data transfer, data receiving, data conversion,

data supervision and data synchronization.

I/o bus and interface Models:-



- The I/o bus consists of data lines, address lines and control lines. Each peripheral device has associated with an interface unit. Each interface decodes the address and control received from the I/o bus, interprets them and provides signals for the peripheral controller.
- It also synchronises the data flow and supervises the transfer between peripheral and processor.
- The I/o bus from the processor is attached to all peripheral interfaces. To communicate with a particular device the processor places a device address on the address line.
- When the interface detects its own address, it activates the path between the bus line and the devices.

I/o buses versus Memory bus:-

- In addition to communicating with I/o, the processor must communicate with memory unit.
- like the I/o bus, the memory bus contains data, address, read/write control lines.
- There are 3 ways that computer buses can be used to communicate with memory and I/o.

  i) use seperate buses, one for memory and other for I/o.

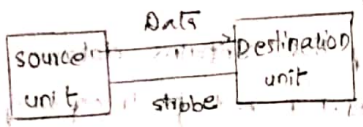
  ii) use one common bus for both memory and I/o but have seperate control lines.

  iii) used one common bus for memory and I/o with common control lines.

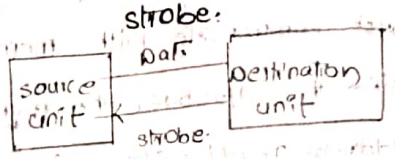
Asynchronous Data transfers-

The internal operations in a digital system are synchronized by means of clock pulses supplied by a common pulse generator. Two units such as cpu and I/o interface are designed independently of each other. Asynchronous data transfer between two independent units requires control signals to be transmitted btw the communicating units to indicate that time at which data is transmitted.

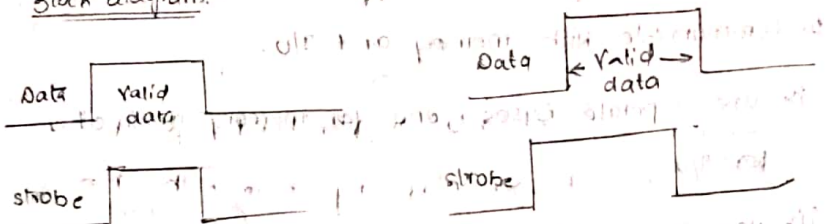
Source initiated strobe:



Destination Initiated strobe:



Block diagram:-



1) strobe is activated by source unit

2) Data bus carries data from source to destination unit

3) strobe is activated whenever data is ready with the source unit

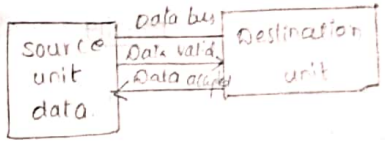
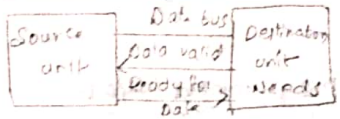
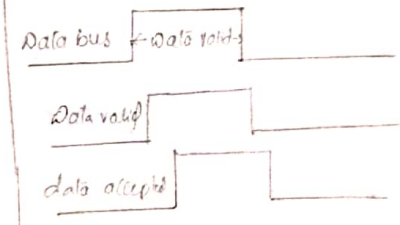
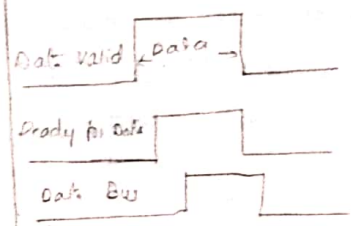
1) strobe is activated by destination unit

2) Data bus carries the data from source unit to destination unit

3) strobe is activated whenever destination unit needs the data.

The unit receiving the data or sending the data responds with another signal to acknowledge receipt or sent the data. This type of agreement between two independent units is reffered to as handshaking.

Handshaking:-

Source initiated transfer using handshaking	Destination initiated transfer using handshaking
	
	
Data valid is generated by source unit. Data accepted is generated by destination unit	Data accepted valid is generated by destinatron unit- Ready for data is generated by source unit.

The hand shaking schemes provides a high degree of flexibility and reliability because the successful completion of data transfer depends on active participation of both units.

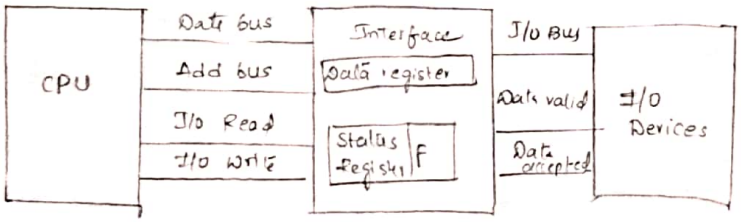
Modes of transfer :-

Binary information received from external device is stored in memory. Data transfer to and from peripherals may be handled in one of 3 modes.

1) programmed I/O
2) Interrupt Initiated I/O
3) Direct memory access

13/9/19

1) programmed I/O :-



f - flag Bit

1) Data transfer from an I/O device through an interface in to cpu is shown in above diagram. The device transfers bytes of data one at a time by placing on I/O bus and enables data valid line.
2) The interface accepts the data into data register and enables the data accepted line.
3) After this interface also sets flag bit inside the status register to 1. (flag = 1).
4) Now the cpu reads the status register and checks the flag bit, if it is 1 data transmitted from data register to cpu and sets the flag bit to 0.
5) The transfer of each byte requires three instructions
   i) reads the status register
   ii) check the status of the flag
   iii) Read the data register.

Disadvantage :-

1) programmed I/O method is useful in small low speed computers that are dedicated to monitor a device continuosly 2) It is inefficient for high speed computers because the cpu is wasting time while checking the flag instead of doing some other useful tasks.

An Alternative to this is Interrupt initiated I/O

2) Interrupt initiated I/O :-

An alternative to cpu constantly monitoring the flag is to let the interface inform the computer when it is ready to transfer data. This mode of transfer uses interrupt facility.

While the cpu is running a program it doesnot check the flag. However when the flag is set to 1 the computer is momentorily interrupted from proceding with the current programme and informed that

flag has been set to 1.

Now the cpu deviates from what it is doing to take care of I/O transfer and after completion it returns to previous program.

16/9/19

Priority Interrupt:-

In a typical applications the number of I/O devices are attached to the system which each device being able to originate an interrupt request. In these case the system must decide which devices to serve first.

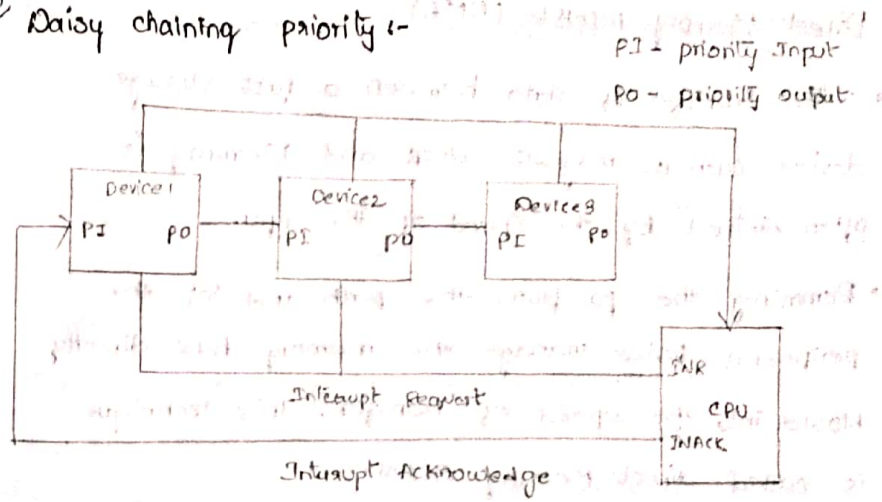
A priority interrupt is a system that establishes the a priority over the various sources to determine which condition to be serviced first when two or more requests arrived simultaneously.

Devices with high speed transfers such as magnetic disk are given high priority, slow devices such as keyboard given low priority.

When two devices interrupt the at same time the computer services the device with high priority first.

6M

Daisy chaining priority :-



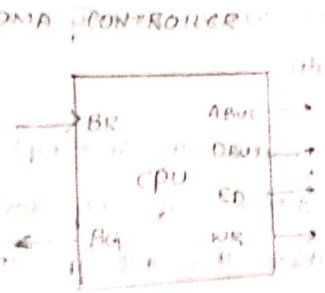
Daisy chaining method of establishing priority consists of serial connection of the devices that request an interrupt. The device with highest priority is placed in the first position, followed by lowest priority devices up to the device with the lowest priority, which is placed last in the chain.

From the above diagram, the interrupt request line is common to all devices. If any device has its interrupt signal enables the interrupt input in the cpu.

The cpu responds to an interrupt Request by enabling interrupt acknowledge line. The acknowledge signal passes through all devices.

Direct Memory Access (DMA)

- The transfer of data between a fast storage device such as magnetic disk and Memory is often limited by the speed of the cpu.
- Removing the cpu from the path and let the peripheral device manage the memory bus directly would into the speed of transfer. This technique is called direct Memory Access.
- During DMA transfer the cpu is idle and has no control of buses. Now the DMA controller takes over the buses to manage the transfer directly between IO devices and memory.



BR → Bus Request

BG → Bus Grant

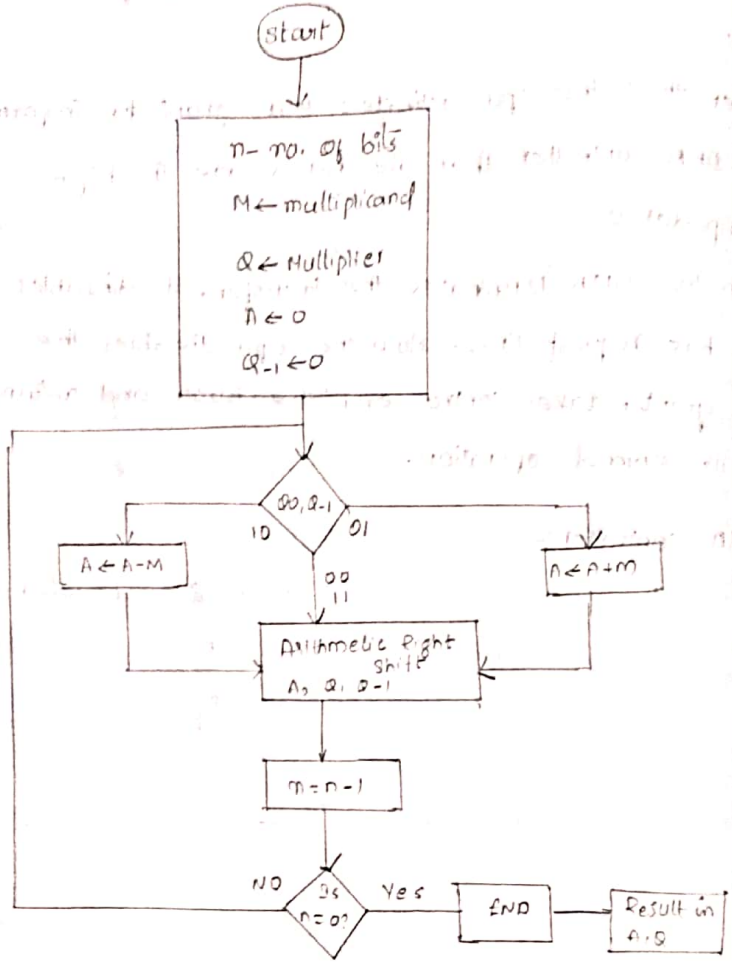
Bus request and Bus Grant are two control signals to facilitate the DMA transfer. The Bus request input is used by the DMA controller to request cpu for control of Buses. When this input is active, cpu places address bus, data bus, Read Write lines into high impedance state. that means the output is disconnected from the cpu.

After this the cpu activates Bus grant to inform the DMA controller that the buses are in high impedance.

When the DMA terminates the transfer, it disables the bus request line. Now the cpu disables the bus grant, takes control over the buses and returns to its normal operation.

DMA controller:-

Booth's Multiplication Algorithm



Eg:- 7 ← multiplicand → 0111
3 ← multiplier → 0011

n	A	Q	Q-1	Operation
4	0000	0011	0	Initial values
	1001	0011	0	A ← A-M
3	1100	1001	1	(Right shift)
3	1110	0100	1	(Right shift)
2	0101	0100	1	A ← A+M
1	0010	1010	0	(Right shift)
0	0001	0101	0	(right shift)

∴ n = n-1

We have to take result of A, Q

00010101 ⇒ 21

Eg:2 5×4 = 20

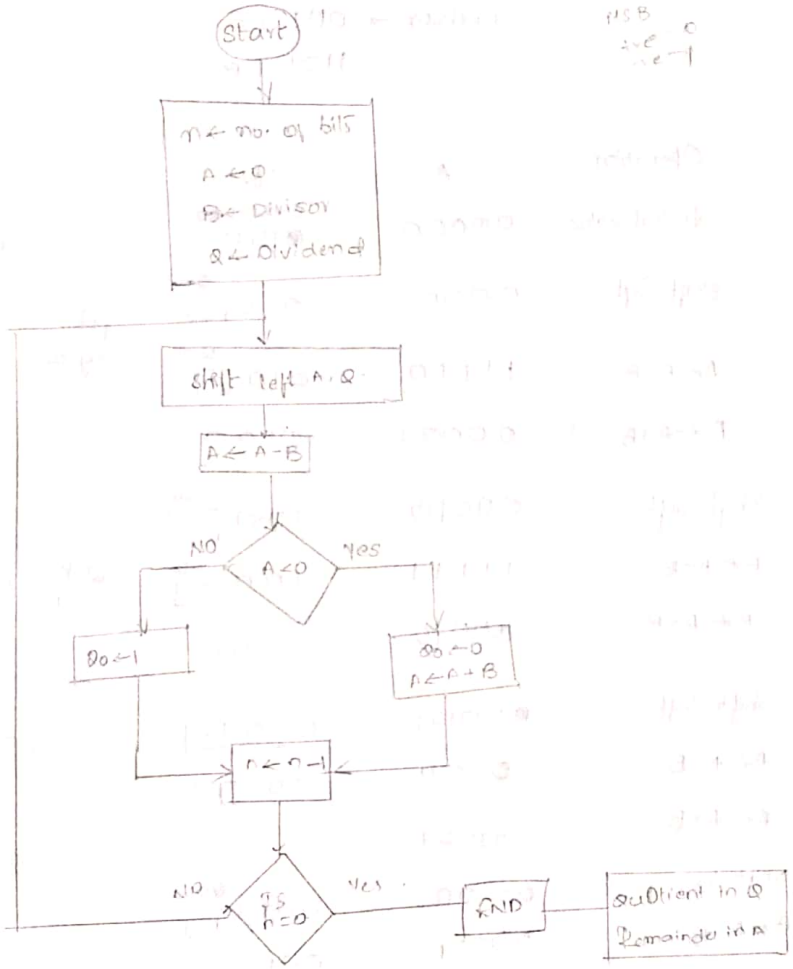
5 ← multiplicand → 0101
4 ← multiplier → 0100

n	A	Q	Q-1	Operation
4	0000	0100	0	Initial
3	0000	0010	0	Right shift
	0000	0001	0	A ← A-M
2	1011	0001	0	Right shift
	1101	1000	1	A ← A+M
1	0010	1000	1	(Right shift)
0	0001	0100	0	

∴ n = n-1

Answer = 00010100
= 16+4 = 20

Division using Restoring & Non Restoring Algorithm.



$10 \div 3$

10 - Dividend → 1010 (Q)

3 - Divisor → 0011 (B)

1101 (-B)

↓

5

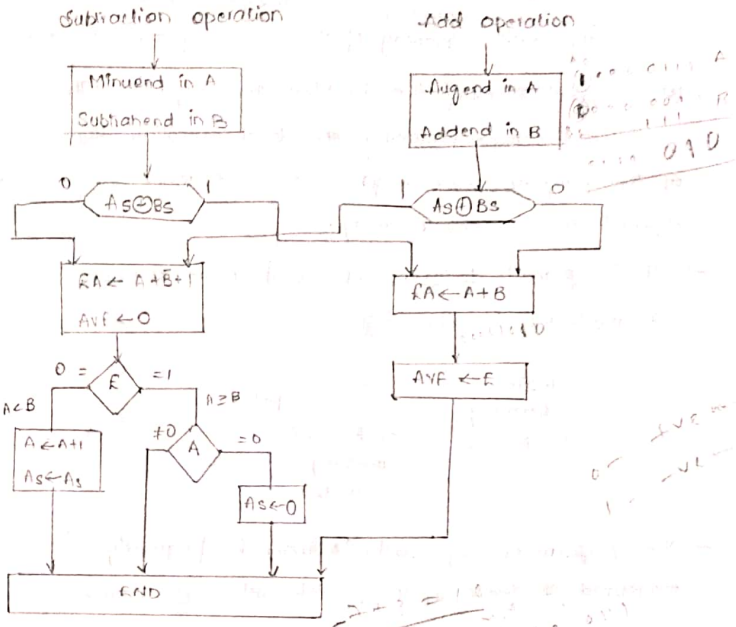
Operations	A	Q.
Initial value	00000	$Q_3 Q_2 Q_1 Q_0$ 1010
Shift left	00001	010 Q_0
A ← A-B	11110	010 Q_0 0
A ← A+B	00001	0100
Shift left	00010	100 Q_0 0
A ← A-B	11111	100 Q_0 0
A ← A+B	00010	1000
Shift left	00101	000 1
A ← A-B	00010	000 1
A ← A+B	00100	
Shift left	00100	001 Q_0 1
A ← A-B	00001	0011

1st cycle

2nd cycle

① Remainder ③ Quotient

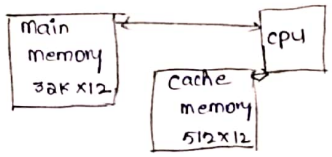
Flowchart for Addition and subtraction Algorithm



Cache Memory:-

Program and data position of the active memory are placed in a fast, small memory the average memory time can be reduced. Thus reducing the total execution time of the program. Such a fast small memory is referred as a cache memory.

→ It is placed between cpu and main memory as illustrated in the fig.



→ The performance of cache memory is frequently measured in terms of a quantity called Hit Ratio.

When the cpu refers to memory and finds the word in cache it is said to be hit.

→ If the word is not found in cache, it is in main memory and it counts as miss. The ratio of the number of hits divided by total cpu references to memory is called hit ratio.

→ The basic characteristic of cache memory is its fast access time.

→ Therefore very little or no time is wasted in order to search memory words in cache memory.

→ The transformation of data from main memory to cache memory is referred to as mapping process

There are three types of mapping processes or procedures in cache memory

1) Associative Mapping
2) Direct Mapping
3) Set Associative Mapping

Memory Management Hardware:-

A Memory management system is a collection of hardware and software procedures for managing various programs residing in the memory. The memory management software is a part of an overall operating system available in computers.

The basic components of a memory management unit are:-

1) A facility for dynamic storage that maps logical memory references into physical memory address.
2) A provision for sharing common programs stored in memory by different users.