* Books recommended:

"computer system Architecture"
— Morris Mano.

* Theor

computer Organization:

→ Mejor component - functional units.
→ Handware components.

Physical aspects:

→ circuit design
→ control singnal
→ Memory types.

computer design / design of computer.
↳ specifications — formulated designer dev
of handware.

computer Architecture
↳ the formate
the instruction set.

techniques for addresing memory.
It concerns with specification of processors, memories & structering them together into computer system.

* It deals with all logical aspects
  - → instruction formates
  - → Instuction sets
  - → Data types
  - → Addressing mode.

2014-12 1.(b)

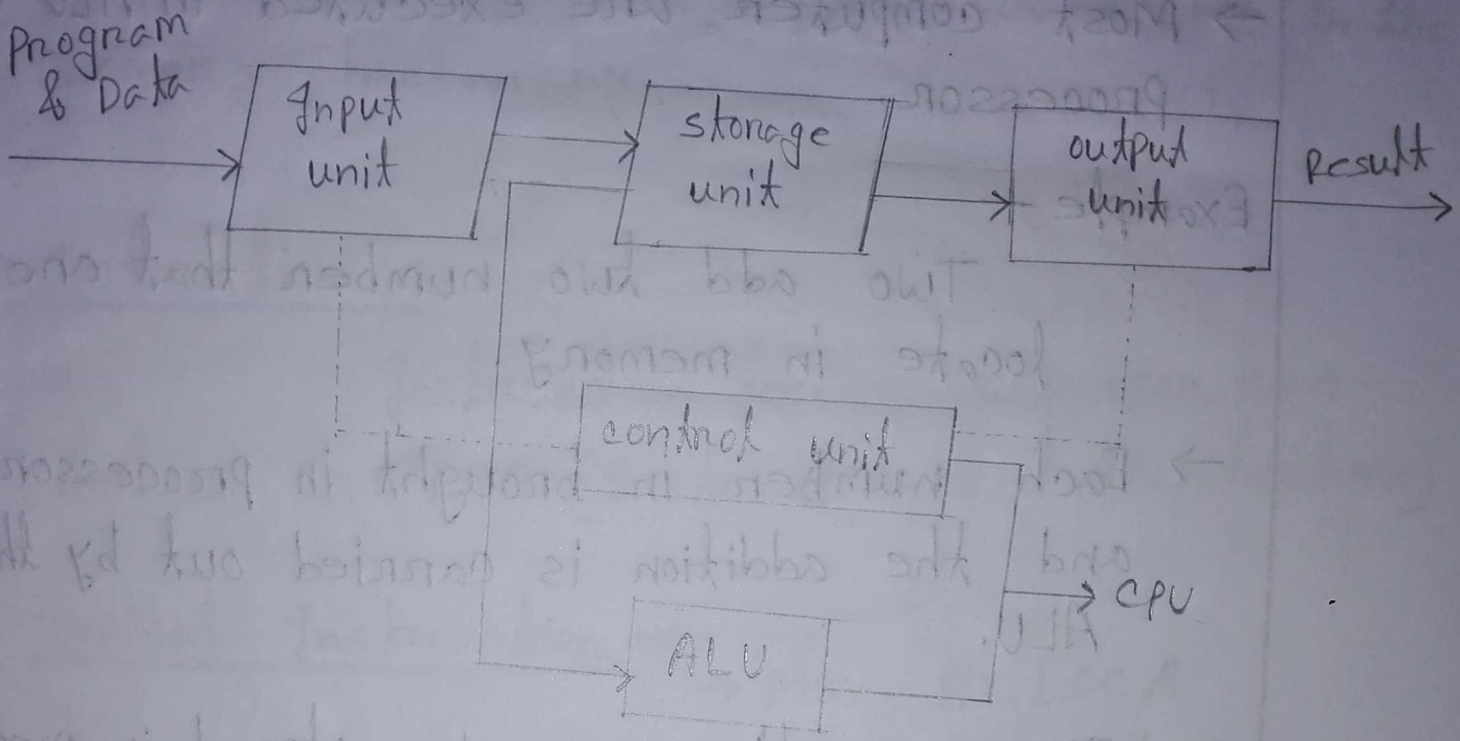
Q. Why we study computer Architecture?

Ans:-

Design Better Programs:

(i) System software such as compiler.
(ii) Operating system.
(iii) Device Drivers.
(iv) Optimise program behaviours.
(v) Evaluate computer system performance.
(vi) Understanding time, space, and trade-offs.

* Basic computer Operation :



* Five mejor Operation :

(i) It accepts data or instruction by way of input.

(ii) It store data.

(iii) It can process data as required by user.

(iv) It gives results in the form of output.

(v) It controls all operations include a computer.

* Basic Operation concepts of computer:

→ Most computer are executed in ALU of a processor.

Example –

Two add two number that are both locate in memory.

→ Each number in brought in processor. and the addition is carried out by the ALU.

→ The sum them may be stored in memory or retained in the processor for immediate use.

* Registers:

When operands are brought into processor they are store in high speed elements.

Register

→ A Register can store one piece of Data. (8 bit, 16 bit, 32 bit, & 64 bit registers).

Access time to registers are faster than access time to the faster cache unit in the memory hierarchy.

* Instruction :

Instruction for a processor are define in ISA (Instruction set Architecture)

Typical Instruction —

→ MOV BX, LOCA
→ Fetch the instruction
→ Fetch the content of memory location.
→ store the content of general purpose register BX.

AU 2014 1.(c)

* BUS structure :

A BUS is basically sub system. which transfers data between the components of computer.

→ It connect peripheral devices.

Some properties:

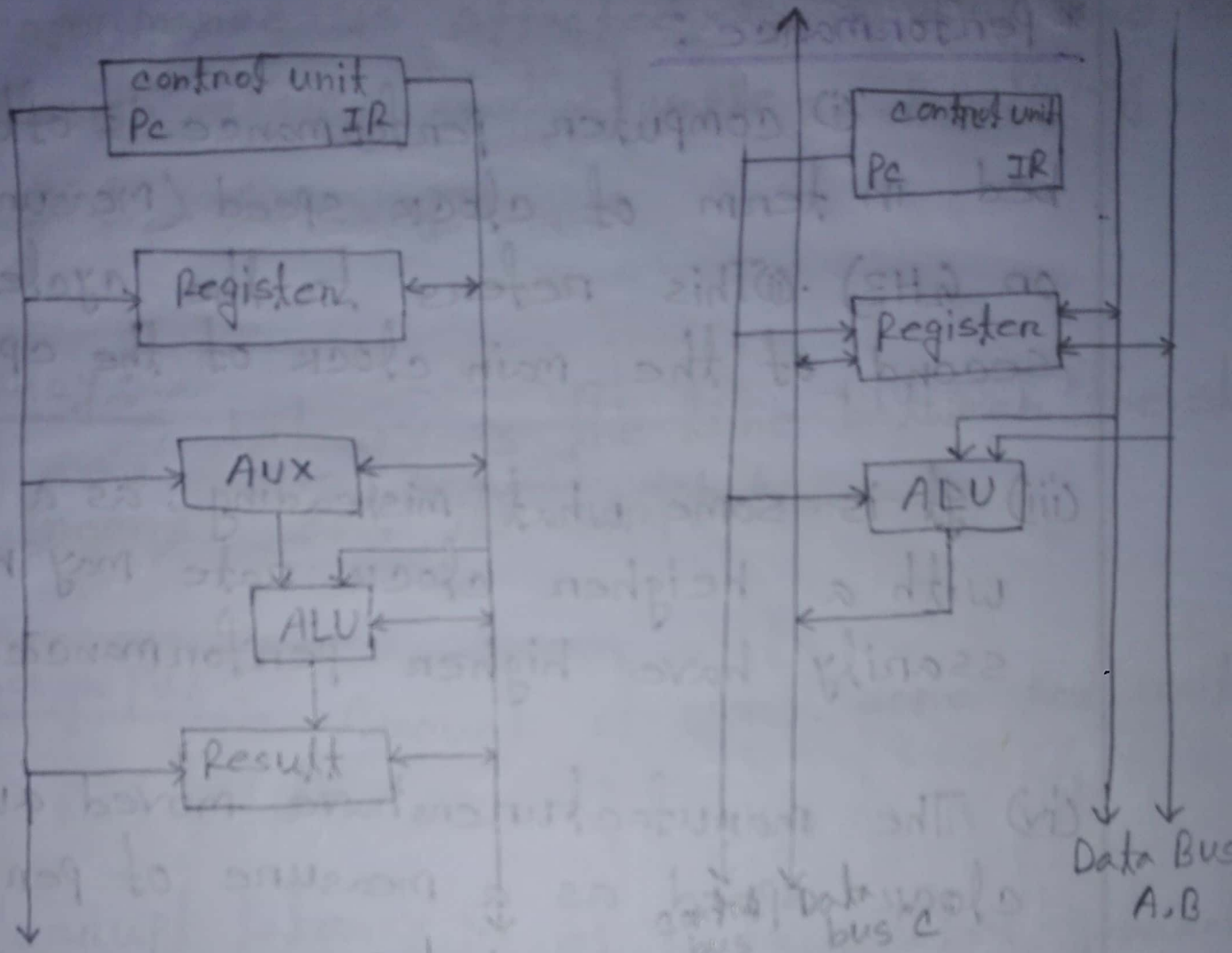
(i) If single bus structure all unit are connected in the same bus than connecting different buses as multiple bus structure.

(ii) Multiple Bus structure performance is better than a single Bus structure.

(iii) Single bus structure is cheap than multiple bus.

campaision between two bus structure b/w. 2012 1.(a)

Diagram:



(a) Single bus.

(b) Multiple Bus.

fig: Bus structure.

* Performance:

(i) computer performance is often described in term of clock speed (measured by MHz or GHz). (ii) This refers to the cycles per second of the main clock of the cpu.

(iii) It is some what misleading, as a machine with a heighen clock rate may not necessarily have higher performance.

(iv) The manufacturers have moved away from clock speed as a measure of performance

(v) Modern cpu can execute multiple instruction per clock cycle to speed up. Other factors influnce speed.

→ The mix of functional units, bus speeds

→ Available main and cache memory.

→ The type and other of instruction in the program being run.

(vi) Performance is affected by a very wide range of design choice. For example — pipelining, Parallel processing.

2014 1.(a) * **Latency:** Latency is the time between the start of process and its completion.

2014 1.(a) * **Through put:** Amount of work done per unit time.

* **Interrupt latency:** It is ~~guranted~~ guaranted maximum response time of the system to an electronic event.

* **Design trade off:**
→ Maximum performance.
→ ~~Maximum~~ cost.
→ Best performance.

* Execute time :

Elapsed time
cpu time.
Our focous on cpu time.

* Measure of performance :

→ clock time = time between ticks = secondly cycle.

→ clock rate - cycles per second.

Ⓔ A 200 MHz clock has cycle time.

Sln:

$$\frac{1}{200 \times 10^6} \times 10^9 = 5 \text{ ns}$$

$$\therefore 3\text{GHz cycle time} = \frac{1}{3 \times 10^9} \times 10^9$$

$$= 0.33 \text{ ns} \rightarrow$$

* $$\text{MIPS} = \frac{\text{clock/second}}{\text{Average clocks per instruction}}$$

$$= \frac{\text{Frequency in MHz}}{\text{CPI}}$$

8. Machine has clock cycle time of 10 ns an CPI of 2.0.

sln:

$$MIPS = \frac{10^6/10\times10^{-9}}{2}$$

$$= 50$$

8. Machine has clock cycle time of 20 ns an CPI of 1.2.

sln:

$$MIPS = \frac{10^{-6}/20\times10^{-9}}{1.2}$$

$$= 41.66$$

$$\therefore \frac{50}{41.66}$$

$$= 1.2$$

* Maximum performance measure by the numbe of instruction executed persec Maximum cost :

Measured by the size of circuit.
Best performance price.
Measured by the ration of MIPS to size. In power sensitive application MIPS / Watt is important too.

2019 1.(a) * Execution time :

Elapsed time/wall clock time :

count everything (i.e ; disk and memory access I/O etc)

a useful numbers, but often not good for comparison purpose.

* CPU time :

Doesn't include I/O on time spent running other program can be broken up into system time and user time.

* Our focus on cpu time: Time spent executing actual instructions of our program.

2014 1.(e) * cpu: cpu time is the amount of time for which cpu was used for processing instruction of a computer program.

* Machine Instruction:

(i) A program is a set of instruction that specify the operation, operand and the sequence.

(ii) An instruction is a binary code that specifies a sequence of micro operation.

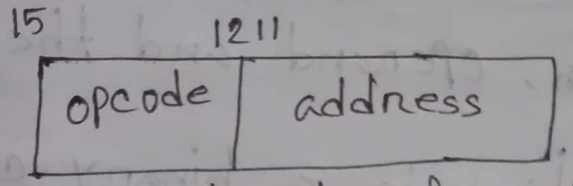
(iii) code and data stored in memory.

(iv) The computer reads each instruction from memory and places it in a control register.

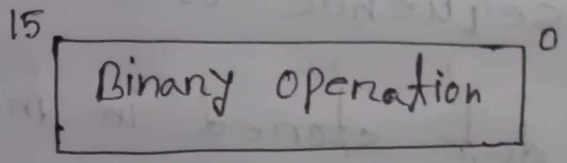
(v) The control then interprets the binary code and process to execute it.

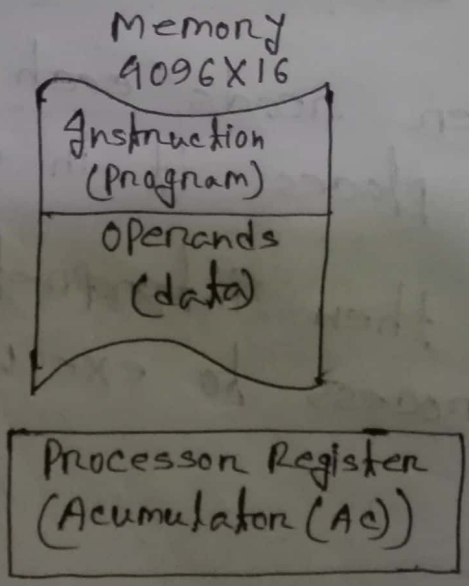
* **Instruction code:** A group of bits that instruction the computer to perform specific operation. It usually divided into part.

* **Operation code:** The most basic part of instruction code. A group of bits that define such operations as add, subtract, multiply, shift and complement.



Instruction format





* Stone Program Operation :

Instruction code for format with two parts opcode + address.

Opcode: specify 16 possible operations. (4 bit)

Address: specify the address of ~~operatio~~ operand. (12 bit)

* Memory :

12 bit = 4096 word

(Instruction + data are stored stone each instruction code (Program) and operand (data) in 16 bit memory.

List of registers in basic computer :

Register symbol	Number of bit	Register Name	Function
DR	16	Data Register	Hold memory operand
AR	12	Address "	" Address for memory
AC	16	Accumulator	Processor register.
IR	16	Instruction Regis-ter	Hold instruction code
PC	12	Program counter	Hold address of instruction.

TR	16	Temporary Register	Hold temporary data.
INPR	8	Input Register	Hold input character
OUTR	8	output Register	Hold output character.

counter

* Program control: hold the address of the next instruction to be read from memory after the current instruction is executed.

A branch instruction calls for transfer to a non-consecutive instruction in the program.

This address part often branch instruction is transfered to pc to pc to become the address of next instruction. To read instruction memory read cycle is initiated and pc is incremented by one.

* Timing and control:

- A master clock generator controls the timing for all register in the basic computer.
- The clock pulse are applied to all F/Fs and register in system.

The clock pulse do not change the state of a register unless the register is enabled by control signal.

- The control signals are generated in the control unit. It provides controls for MUX in the common bus, register etc.

* Hardware control:

- The control logic is implemented with gates F/Fs, decodes and other digital circuit.
- For any change, need to change hardware.
- First operation. / operation is fast.

* Microprogram control:

- The control information is stored in a programmed control memory.
- Any required change can be done by updating the microprogram in control memory.
- Operation is slow.

* Instruction cycle:

(i) Instruction fetch from memory: Read instruction form memory through the data bus.

(ii) Instruction Decode: The control unit determine the type of instruction that was just read from memory.

(iii) Fetch the Operand: Accumulate the operands to make operation. Read from memory in case of indirect addressing mode.

(iv) Instruction Excution: Perform the operation based on decode opcode

(v) Go to step (i) Next instruction [Pc + 1].

The address field depend on the arrangment of register.

- Most computes fall into one of three type:

(i) single Accumul organization : ADD X

(ii) General Register organization : ADD R1, R2, R3

(iii) stack operation : PUSH X.

* Instruction with different address field :

(i) Three address instruction :

ADD R1, A, B R1 ← M[A] + M[B]

MUL X, R1, R2 M[X] ← R1 * R2

- Each address fields specify either a processor register on a memory operand.
- short program.
- Require too many bit to specify 3 address.

(ii) Two address instruction :

MOV R1, A R1 ← M[A]

MUL R1, R2 R1 ← R1 * R2

MOV X, R1 M[X] ← R1

> The most common in commercial computer.
> Each address fields specify either a Processor register or a memory operand.

(iii) One address instruction :

LOAD A	AC ← M[A]
ADD B	AC ← AC + M[B]

All operation are done between the AC register or a memory operand.

(iv) Zero Address instruction.

PUSH A	TOS ← A
PUSH B	TOS ← B
ADD	TOS ← A+B

> stack organised computer does not use an address field for the instructions ADD and MUL.
> PUSH and POP instruction need an address field to specify operand.
> Zero Address : Absence of Address (ADD, MUL)

* Exam — 22-10-19
Introduction
Machine instruction.

* Addressing Mode:

- It specifies a rule for interpenting the address field operands.

* Implied Mode:

Operands are specified implicity in defined of the instruction.

Example — COM → Complement Accumulator.

→ Operand in AC is implied in the defination of the instruction.

→ PUSH — STACK PUSH

→ Operands is implied to be on top of the stack.

* Addressing Mode:

(i) Immediate Mode:

→ Operand field contain the actual operand.
→ It is useful for initializing register to a constant value.
Example – LD # NBR.

(ii) Register Mode:

Operands are in Register. Register is selected from a register field in the instruction

Example – LD R1 · AC ← R1

(iii) Direct Addressing Mode:

Effective address is equal to the addressing field of the instruction.

Example – LD ADR AC ← M [ADR]

(iv) Indirect Address:

Address field of the instruction gives the address where the effective address is stored in memory.

Example –

LD @ ADR $Ac \leftarrow M[M[ADR]]$

(v) Relative Addressing Mode:

PC is added to the address part of the instruction to obtain the effective address.

Example –

LD $ ADR $Ac \leftarrow M[PC+ADR]$

(vi) Index Addressing Mode:

→ XR (Index Register) is added to the address part of the instruction to obtain effective address.

Exam –

LD ADR (XR) $Ac \leftarrow M[ADR+XR]$

ii) Base Register address Mode:

The content of a base register is added to the address part of the instruction to obtain effective address.

→ base register hold a base address.

Exam — LD ADR (BR)

AC ← M [BR + ADR]

*I/O configuration:

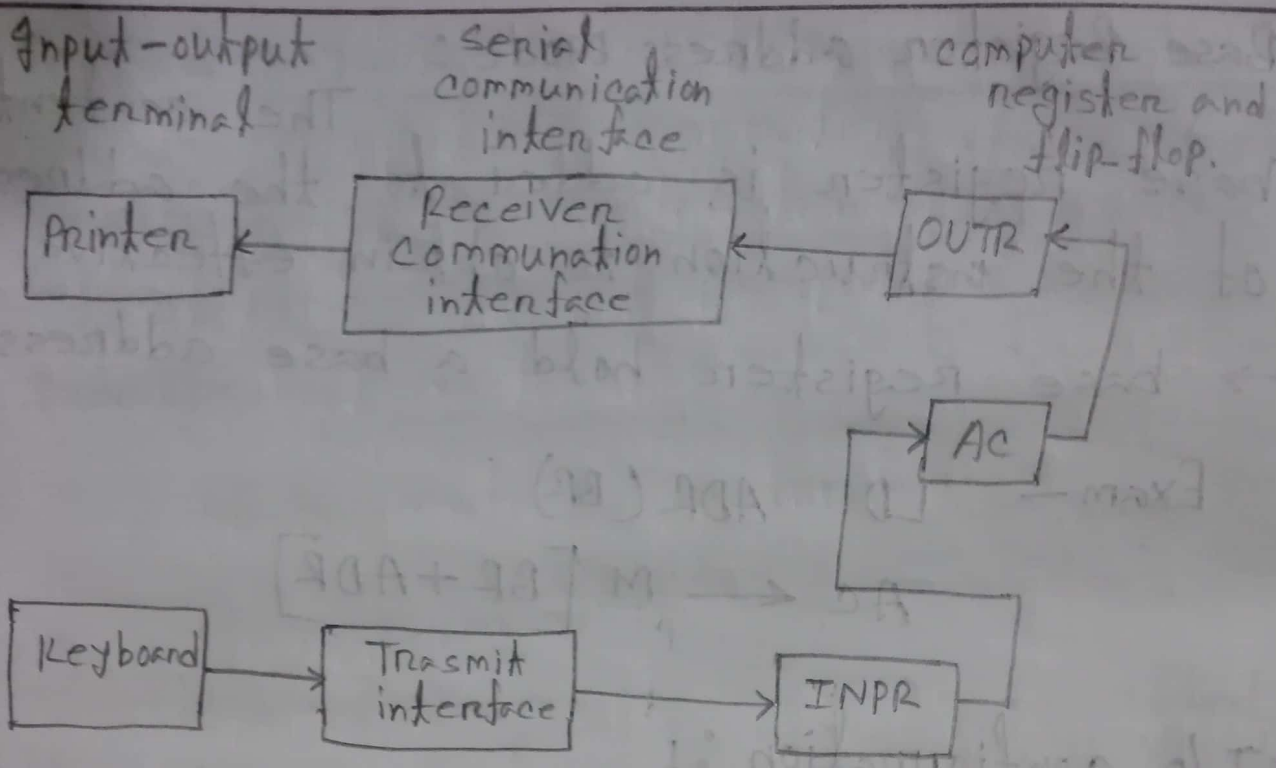
The configuration with I/O is performed through two register

(i) Input register (INPR)

(ii) output register (OUTPR)

The registers communication I/O serially and with the AC in parallel.

(P.T.O) →



* I/O instruction :

→ INP and out are the common instruction for I/O Processing. INP – AC(0-7) ← INPR

FGI ← 0 : Input character.

FGI → input flag

OUT → OUTR ← AC(0-7)

FGO ← 0, output character.

FGO → output Flag.

H.W

*Program control with status bits.

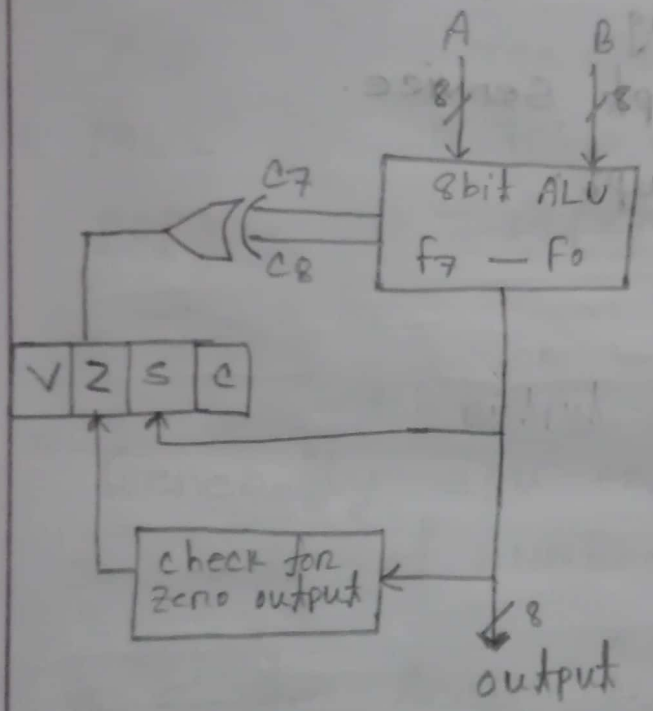
4 bit status

(1) bit c (carry)

(2) bit s (sign)

(3) bit z (zero)

(4) bit v (overflow).



*Data transfer and manipulation.

3 categories

(i) Data transfer

(ii) " Manipulation

(iii) Program control instruction.

* Data transfer instruction.
* Data manipulate instruction.
    (i) Anithmetic instruction.
    (ii) Logic and Bit manipulation
    (iii) shift instruction.
* program control instruction.
* Interupt Handling
* step of Interupt service.
* Type of interupt.

* Zero address:

PUSH A	TOS A
PUSH B	TOS B
ADD	TOS A+B
PUSH C	TOS C
PUSH D	TOS D
ADD	TOS C+D
MUL	TOS = (A+B) × (C+D)
POP X	M(X) = TOS

$X = (A \oplus B) \times (C+D)$

post Fixed → $X = AB+CD+X$

TOS → TOP of STACK..

* Generally cpu organization are three type on basis of number of address field.

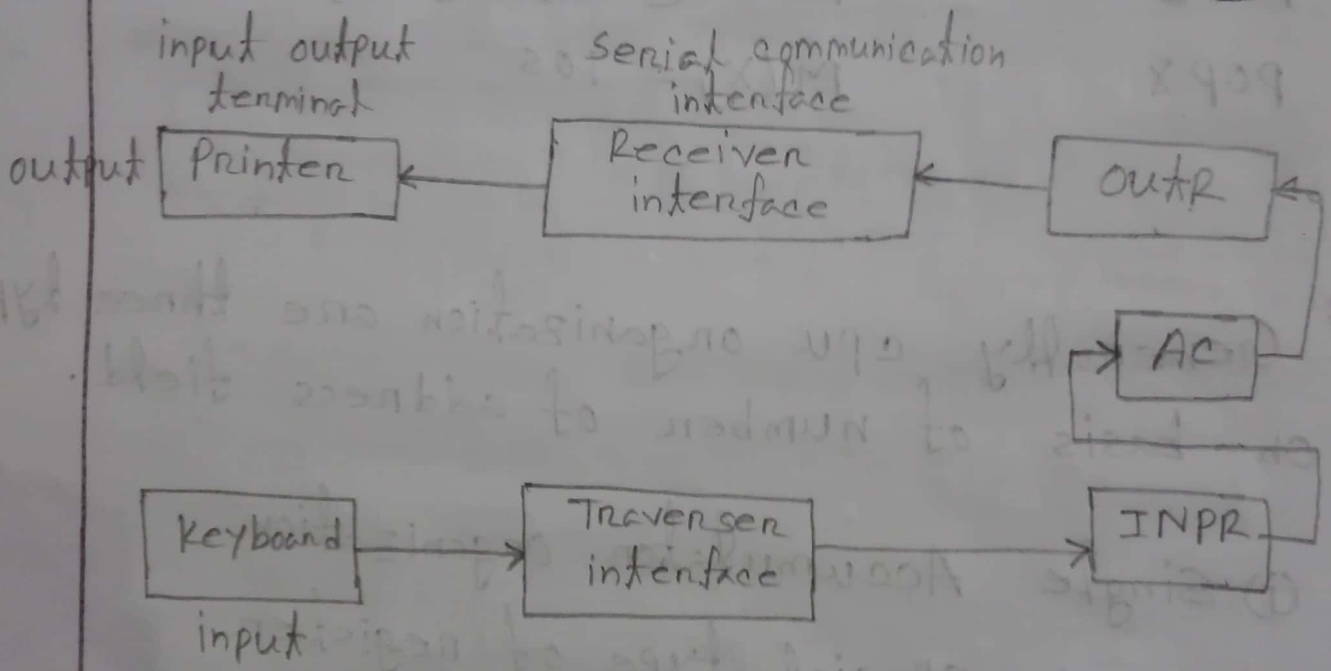
(i) Single Accumulator organization
→ special type of register.

(ii) Generally Register → use for computation purpose.

(iii) STACK organization : The work on basis operation due to which it ~~does~~ does not contain any address field.

* Input output configuration :

Manis Mano book : Fig-5.12 , Page-151



*Flag Register:

8-bit Register

5 bit used. Unused 3 bit

S	Z	-	Ac	-	P	-	Cy

Carry flag: When carry is generated it is set 1. No carry it is set 0.

Parity flag:

1	0	1	0	1	1	0	0

no of 1 is = 4 N is even - set 1.

no of

1 জোড় সংখ্যা হলে

sign flag: (S flag) → Negative - 1
positive - 0

Zero flag::

□ − □ = 0

2 − 2 = 0 Zero flag − 0

□ − □ = 1

3 − 2 = 1 Zero flag − 1

Ac = Auxilary canny flag.

A → 1101 1101

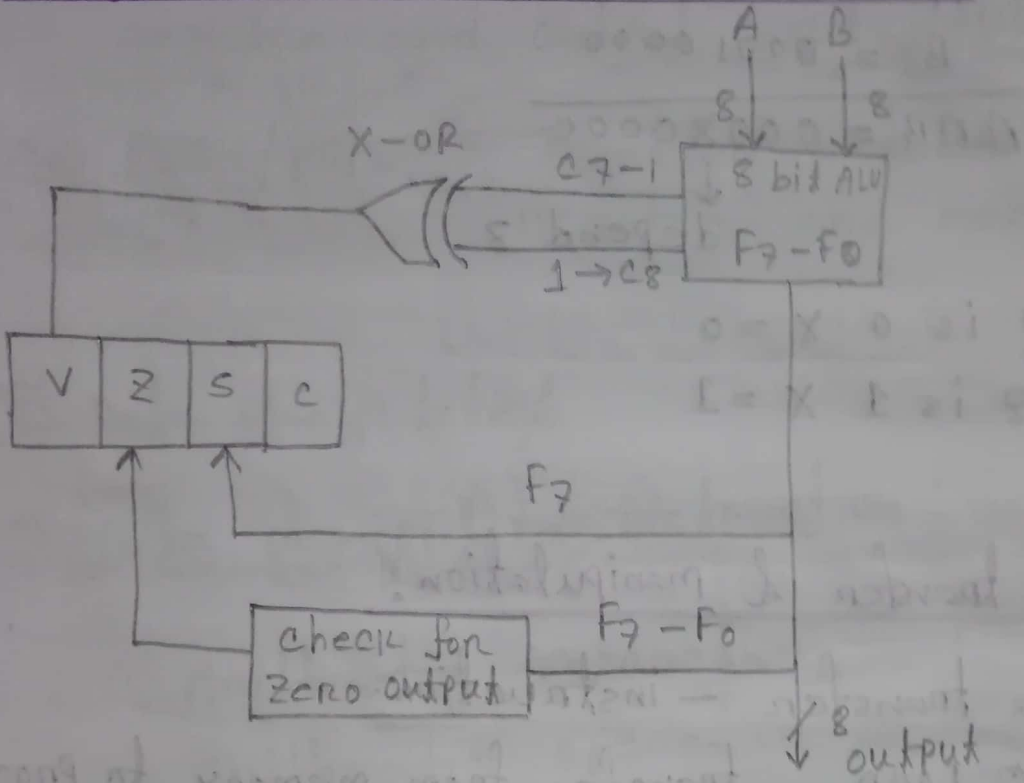
001 1101

(1)000 (1)0010

Ac = 1.

S	Z	-	Ac	-	P	-	CY
↓	↓		↓		↓		↓
0	1		(1)		0		1

* Program control with status bit:



Bit c: Is set 1, if carry 1 is generate.

S → Is set to 1, if highest orden bit F8 is 1.

If set to 'z' the output of ALU contain zero.

Bit V (overflow) it set 1. If X-OR of the last two carries (C_8 and C_7) equal 1, else 0.

unknown

Ex — $A = 101X\ 11\ 00$

$B = 0001\ 0000$

$A \cap B = 000X\ 0000$

↓

depend z

if z is 0 X = 0

if z is 1 X = 1

* Data transfer & manipulation:

① Data transfer — instruction.

(i) LOAD — transfer from memory to processor register usually Ac.

② (ii) STORE — tranfer from memory to processor register into memory.

(iii) MOVE — transfer from one register to another register.

(iv) Exchange — swap instruction from one register to another register.

between two register / a register and a memory word.

(v) Input Output → Transfer data among processors register and output input device.

(vi) PUSH / POP → Transfer between processor register and memory stack.

~~(2) Data manipulation~~

(2) Data manipulate instruction :

(i) Arithmatic instruction — Perform basic arithmatic operation on specific operand

Exam — INC, DEC, ADD, SUB, MUL, DIV.

(ii) Logical and Bit manipulate instruction :
Logical instruction perform binary operations on the string of Bits stone in register

Exam — CLR, OR, AND, COM. → (complement).

(iii) shift operation : shift operation in which the bits of words are moved to the left

(P.T.O) →

or ↓ right.

Ex — SHR, ROR, SHL → (Logical shift Left).

↳ (Logical shift Right). (Rotate Right).

* Program control instruction:

Is used to make the flow of program controlled to be ~~alert~~ altered. If specified condition for altering the content of the program counter breaking the sequence of instruction execution.

Example — BR (conditional Branch), call, JMP (Jump),

* Interupt handling:

Interupt is an event that changes the sequence in which the process execute instructions.

Transfer program control from a current running program to another service.

(P.T.O) →

control return to the original program after the service program is execute.

Type of interrupt: (i) External Interrupt
(ii) Internal "
(iii) software "

(i) External Interrupt: comes form I/o device, from a timing ~~clock~~ device from a circuit monitoring the power supply. and any other external source.

(ii) Internal Interrupt: caused by register over-flow attempt to devide by zero an invalied operation stack operation and protection violation.

(iii) software Interrupt: Initiated by executing an instruction by the programmer to initiate

(P.T.O) →

an interrupt procedure at any desire point in the program.

* STEPS of Interrupt service taken by CPU:

(i) An interrupt is initiated by internal or extern signal.

(ii) The address of Interrupt service Routine (ISR) is determined by handware.

(iii) An interrupt procedure all the information necessary to define the state of the cpu.

(iv) Exceute the ISR.

(v) Restone the original state

(P.T.O) ⟶

9-10-19

* computer Generation :-

(i) First generation (1940-1956) :→ used vacuum tube as a mejor piece of technology. Vacuum tube ~~was used~~ were widely used from 1940 to 1956. Vaccum tube were large component and result in first generation computer. being quite large in size taking up a lot of space in room. some of first generation computers took up an entire.

ENIAC : is a great example of first generation computer. It consisteds of nearly 20,000 vacuum as well a 10,000 capasition and 70,000 registons.

It ~~recure~~ weighted over 30 tons and took up lot of space, requining large room to house it. other example of first generation computer including EDSAC, IBM 701 and Manchester Mark 1.

2014
1.(b)

Limitation of first generation of computer:

(i) Operating speed was very slow.
(ii) power consumption was very high.
(iii) They required large space for installation.
(iv) The Program capability was very low.
(v) Quite langer, they generate lots of heat. require special housing.
(vi) The medium internal store.

(ii) Second Generation (1956-1963) → The second generation computer saw the use of transistons insted of vaccum tube. Transistor are widely used in computer from 1956. to 1963. Transistor were smaller in size, faster in speed. and cheaper to build. The first computer to use transistors was the TX-0. and was introduced in 1956.

other computer that used transistors include the IBM-7070, philco transac s-1000, and RCA 501.

2016 1.(b) Q. Which limitation of first generation solved in 2nd generation?

⇒

Transistor replaced the vaccum tubes of the first generation computer.

Transistor allows computer become smaller faster, cheaper, enagy efficient and, reliable.

2017 1.(a) **Limitation of second generation of computer:**

(i) They used transistor as their main componen

(ii) Maintainance requirement is high.

(iii) Very costly.

Third generation computer (1964-1971) → The

third generation of computer introduced the use of Ic in computer. The Ic in computer helped reduce in size of components even more compared to second generation computer. as well as make more faster. They used Ic is popularly known as chips. The capacities of main memory were greatly enlarged. They used operating system that allow mechines to run many different program simultaneously

Limitation of third generation of computer:

(iv) Forth generation computer (1972-2010) →

The fourth generation computer took advant of the invention of microprocessor most comm. only know as CPU. Microprocessor along with integrated circuits helped make it possible to fit computer easily on a desk. and for introduction of the laptop. They have high computing power and extremly large memories than eartier computer. They used large scale integrated circuit and very large scale integrated circuit.

Limitation of Fourth generation computer:

(v) Fifth generation computer (2010 - at present) →

The fifth generation of computer is begining of ~~Article~~ Artificial Inteligence (AI) an exciting technology that has many potential application around the world. Leaps have been mode in AI technology and computers. But these is still much room for improvement.

Limitation of Fifth generation computer: