* INTRODUCTION *

Computer: computer is an electronic device used for storing & processing data, carrying out sequence of operations. It is a device which is invented to simplify complicated tasks.

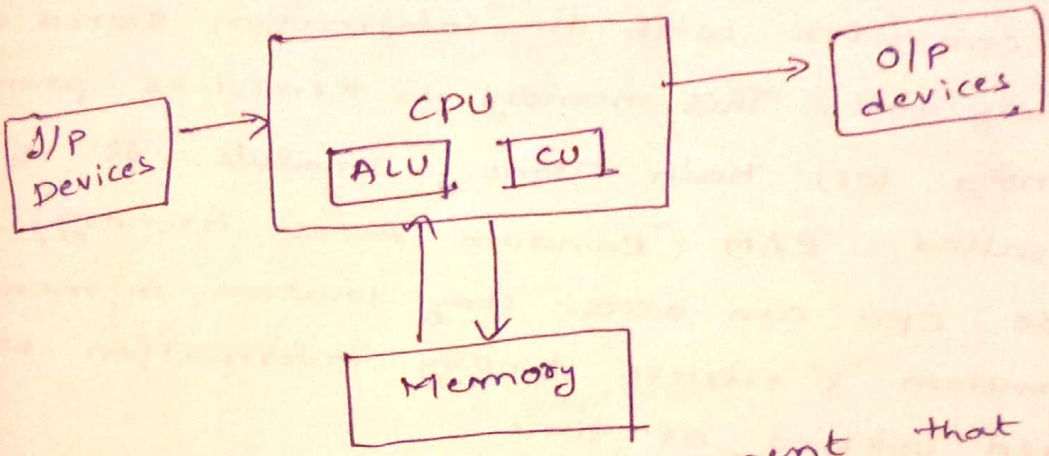
Computer Organization & Architecture:

Computer Architecture	Computer organization
* It refers to those attributes of a system visible to a programmer, that have direct impact on the logical execution of a program	* It refers to the operational units & their interconnections that realize the Architectural specifications
* Examples of Architectural attributes include instruction set, no. of bits used to represent various data types, I/O Mechanisms, techniques for addressing Memory.	* organizational attributes include those hardware details transparant to the programmar such as control signals, interfaces blw computer & peripherals & Memory technology used.
*It is an architectural issue whether a computer will have a multiply instruction	* It is an organizational issue whether that instruction will be implemented by a special multiply unit (or) by method of repeated additions.
* high level (Architecture)	* low level (Micro Architecture)
* what to do	* How to do (Implementation of Architecture)
* (looks at design of the house)	* (It examines lumber, bricks, nails & other building material)

Basic Computer Model & different units of computer

The Model of a computer can be described by 4 basic units in high level abstraction. These basic units are:

- central processing unit (CPU)
- Input unit
- output unit
- Memory unit



CPU: is the computer component that carries out instructions of a computer program by performing basic arithmetic, logical, control, I/o operations specified by the instructions

It is divided into two parts

a) ALU (Arithmetic Logic unit): Executes arithmetic & logical operations. CPU contains Set of registers for storing data needed by ALU.

b) CU (control unit): It contains set of registers & control circuit to generate control signals, to carry out stored program instructions

I/p unit: It helps in getting data from outside to the computer. program (or) data is read into main storage from I/p device (or) Secondary storage under control of CPU i/p instouction

Ex:- keyboard, Mouse, Hard disk, Floppy disk, CD-ROM etc.

O/p unit: It helps in providing the computer results to the user (or) storing in storage device permanently for future use. O/p data from main storage go to o/p device under the control of CPU o/p instructions.

Ex:- printer, Monitor, plotter, Hard disk, etc..

Memory unit: It is used to store data & program. CPU can work with the information stored in Memory unit. This memory is termed as primary memory (or) Main Memory module. It is also called RAM (Random Access Memory) because CPU can access any location in memory at random & retrieve binary information within a fixed interval of time.

Other types of Memory are:

ROM (Read only Memory): ROM, PROM, EPROM, EEPROM

Secondary Memory: Hard disk,

Computer's Functional view:

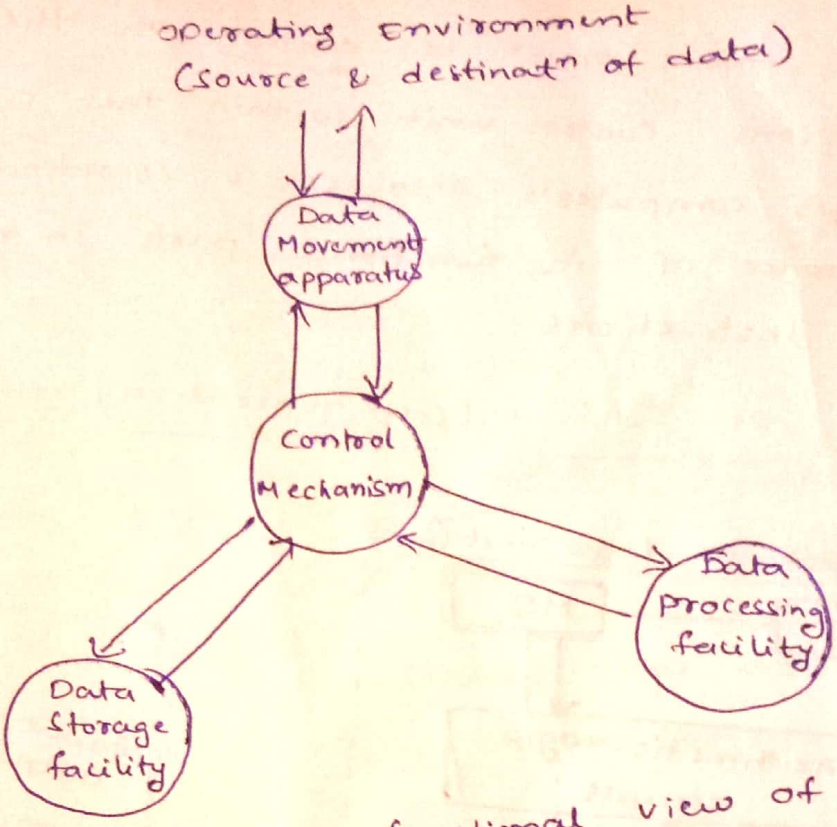


fig: A functional view of computer.

There are 4 basic functions a computer can perform: They are

1) Data Processing
2) Data Storage
3) Data Movement
4) Control

1) A computer must be able to process data. There must be a variety of instructions for processing, of data.
2) For processing of data, data must reside somewhere. There should be atleast a short-term data storage function. But Computer equally performs long-term data storage function as files of data can be stored on computer for subsequent retrieval & update
3) Computer must be able to move data blw itself & outside world. Computer's operating environment consists of devices that serve as either sources (or) destinations of data

Data can be moved to I/O devices (or) remote devices

4) Finally, there must be control of the above 3 functions. Control unit within the computer manages computer's resources & co-ordinates the performance of its functional parts in response to the instructions.

Structure of IAS (First generation) computer:

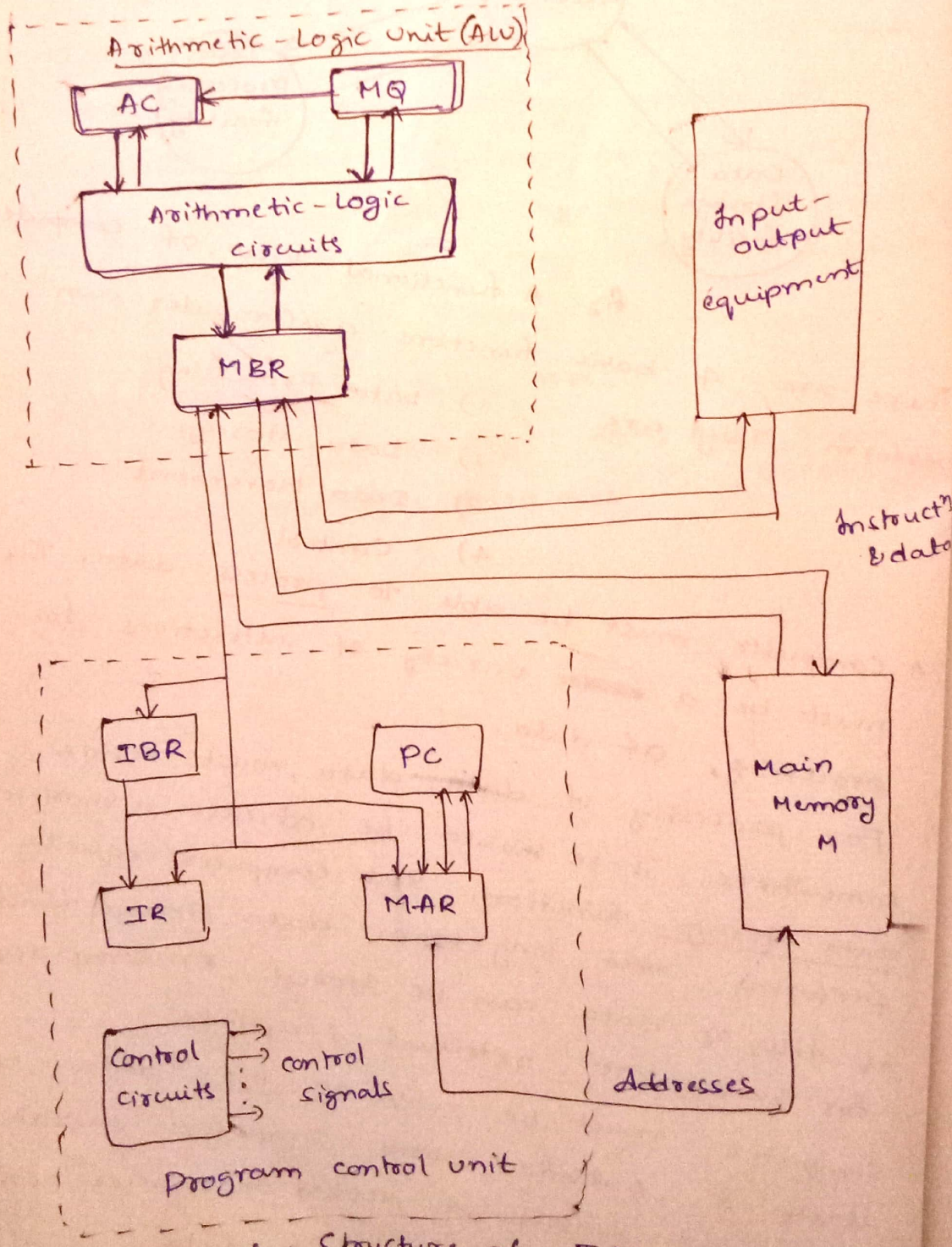


fig: Structure of IAS computer

Memory buffer Register (MBR) contains a word to be stored in memory (or) sent to the I/o unit, (or) is used to receive a word from memory (or) from I/o unit.

Memory address Register (MAR): specifies the address in memory of the word to be written from (or) read into MBR

Instruction Register (IR): contains 8-bit opcode instruction being executed.

Instruction Buffer Register (IBR): Employed to hold temporarily the right hand instruction from a word in memory.

program counter (PC): contains the address of the next instruction pair to be fetched from Memory.

Accumulator (AC) & Multiplier quotient (MQ): Employed to hold temporarily operands & results of ALU operations. Most significant bits are stored in AC & least significant in the MQ of Multiplicat^{n} result.

Instructions & Instructions Execution Cycle:

Instruction: A computer instruction is a binary code that specifies a sequence of Micro operations for the computer.

- Instruction codes & 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 instruction & proceeds to execute it by issuing a sequence of micro operations.
- Every computer has its own unique instruction set.
- An **instruction code** is a group of bits that instruct the computer to perform a specific operation.
- The **operation code** of an instruction is a group of bits that define such operations as add, subtract, multiply, shift & complement.

{ The no. of bits required for the operation code of an instruction depends on the total no. of operations available in the computer. Ex:- The operation code must contain at least 'n' bits for a given 2^n (or less) distinct operations. }

- Apart from **operation code**, an instruction code specifies registers (or) memory words where the **operands** are to be found, as well as register (or) memory word where the **result** is to be stored. Memory words are specified using their addresses. Registers are specified using a binary code of k bits that represent one of 2^k registers.

Instruction cycle:

The basic function performed by a computer is execution of a program which consists of a set of instructions stored in memory.

- The processing required for a single instruction is called an instruction cycle.
- Instruction processing consists of two steps:
  - The processor reads (fetches) instructions from memory one at a time &
  - Executes each instruction.
- Program execution is repeating the above steps for each instruction.

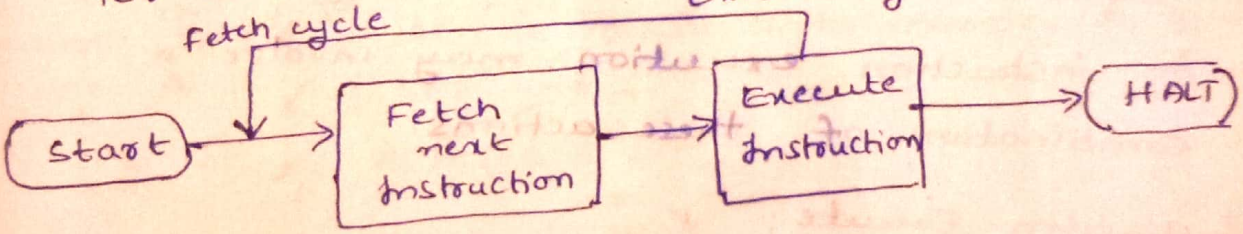


fig: Basic instruction cycle.

Instruction Fetch:

At first, the processor fetches an instruction from memory. A register called Program counter (PC) holds the address of next instruction to be fetched.

- Unless specified, processor increments PC after each instruction fetch {as the program instructions execution is sequential}
- The Fetched instruction is loaded into a register in the processor known as Instruction Register (IR)
- The processor interprets the instruction & performs the required action.

In general, these actions can be one of the following:

1. Processor - Memory: Data may be transferred from processor to memory or from memory to processor
2. Processor - I/O: Data may be transferred to (or) from a peripheral device by transferring b/w the processor & an I/O Module
3. Data processing: The processor may perform some arithmetic (or) logic operations on data
4. control: An instruction may specify that the sequence of execution be altered (Ex: Jump).

- An instruction execution may involve a combination of these actions.

Instruction Execute:

The Execution cycle for a particular instruction may involve more than one reference to memory (or) It may specify an I/O operation. (or) It may be a data processing instruction. Based on the interpretation of the instruction processor executes it.

The following diagram specifies detailed instruction cycle in the form of state diagram. The states can be described as follows:

- Instruction Address calculation: Determine the address of the next instruction to be executed. Usually this involves adding a fixed number to the address of the previous instruction.

Instruction Fetch: Read instruction from its memory location into the processor.

Instruction operation decoding: Analyze instruction to determine type of operation to be performed & operand(s) to be used.

Operand Address calculation: If the operation involves reference to an operand in memory or available via I/O, then determine address of the operand.

Operand Fetch: Fetch the operand from memory or read it in from I/O.

Data Operation: Perform the operation indicated in the instruction.

Operand Store: write the result into memory (or) to I/O

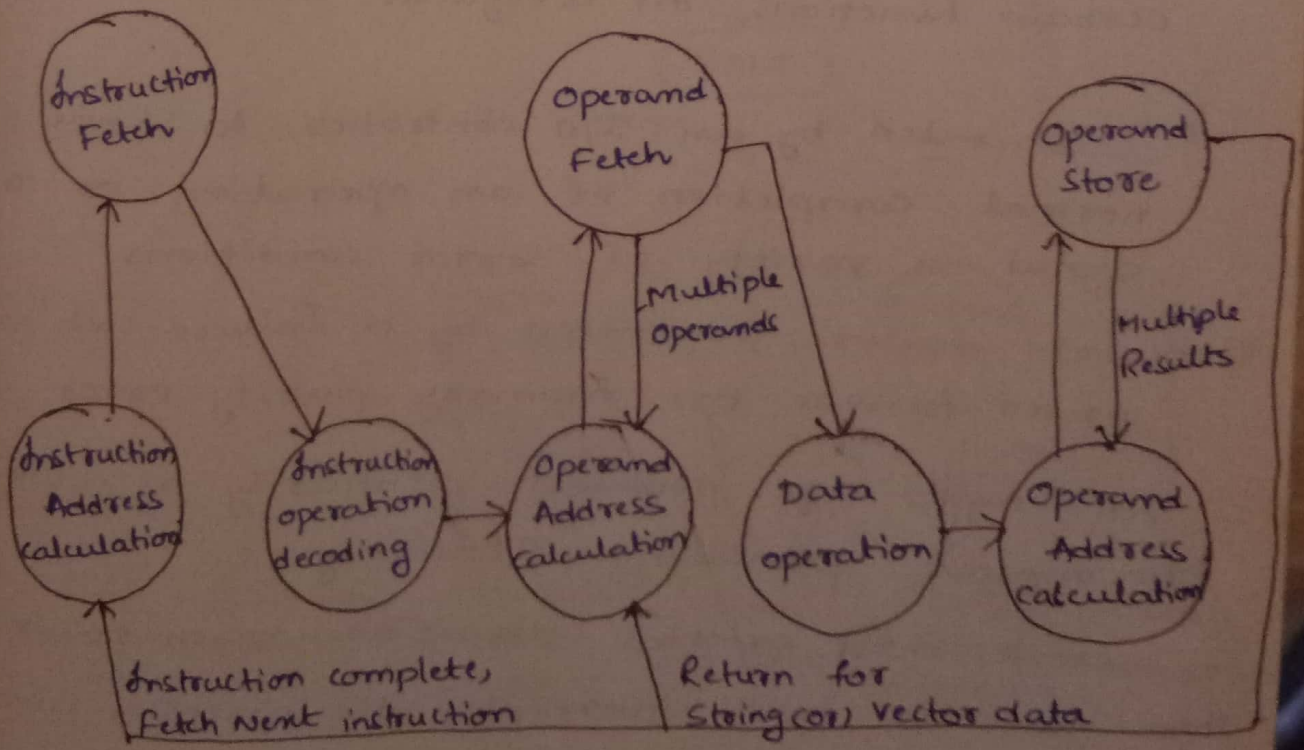


fig: Instruction cycle state diagram

Instruction cycle with Interrupts:

Interrupts: virtually all computers provide a mechanism by which other modules (I/O, Memory) may interrupt Normal processing of the processor. Generally, The interrupts can be divided into following 4 classes.

Program : Generated by some condition, that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, (or) reference outside a user's allowed memory space.

Timer : Generated by a timer within the processor. This allows the Operating System to perform certain functions on a regular basis.

I/O : Generated by an I/O controller, to signal normal completion of an operation or to signal a variety of error conditions.

Hardware failure: Generated by a failure such as power failure (or) memory parity error.

- Interrupts are provided primarily as a way to improve processing efficiency.
- For example, most external devices are much slower than processor. with interrupts, the processor can be engaged in executing other instructions while an I/O operation (which caused interrupt) is in progress
- For I/O operation, say an output operation, like printing some information by a printer. printer is much slower than CPU. CPU puts some information on output buffer. while printer is busy printing these

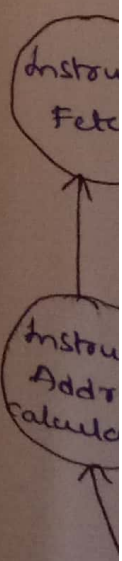


fig:

information from output buffer, CPU is lying idle. During this time CPU can perform some other task which does not involve the memory bus.

- To accomodate interrupts, an interrupt cycle is added to the instruction cycle.

The following will be the stages of instruction cycle.

Fetch: Read the next instruction from memory into processor

Execute: Interpret the opcode & perform indicated operation

Interrupt: If interrupts are enabled, & an interrupt has occured, save the current process state & serve interrupt.

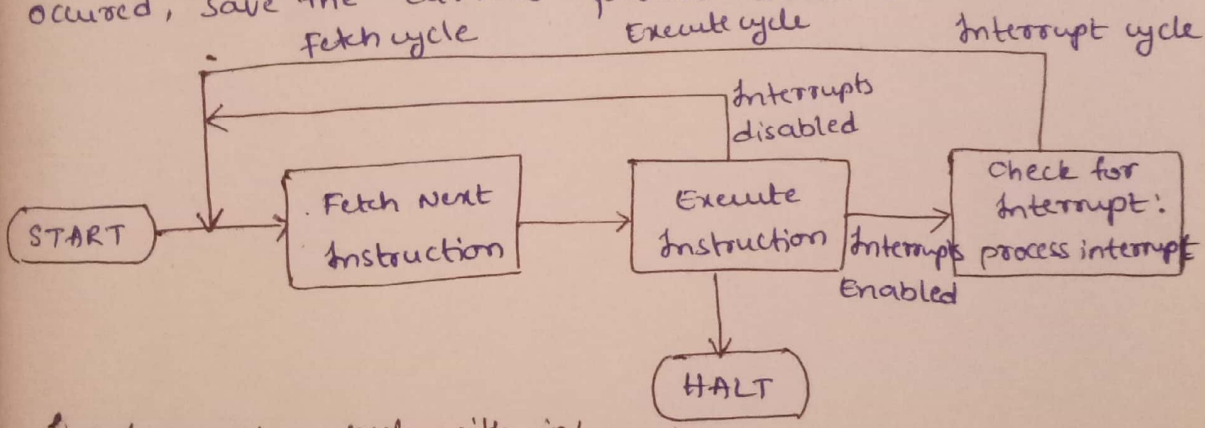


fig: Instruction cycle with interrupts

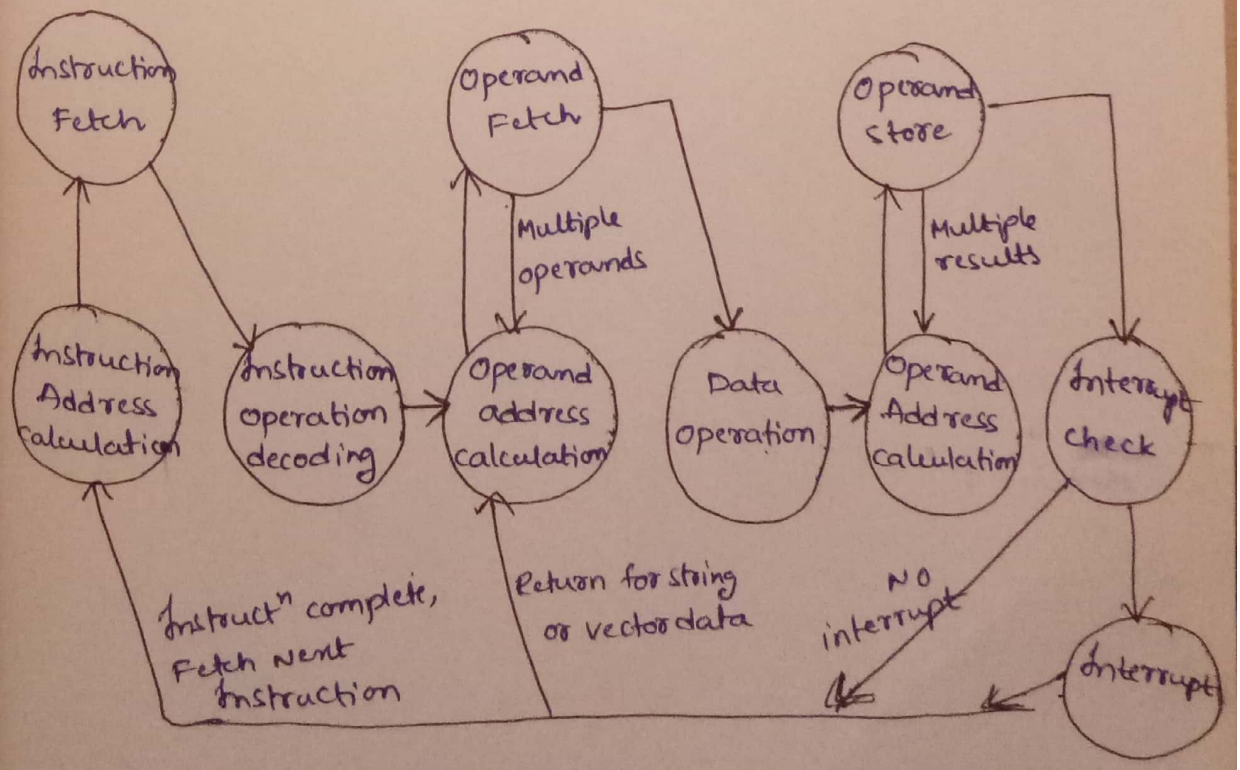


fig: Instruction cycle state diagram, with interrupts