PROCESSOR

There are several characteristics that differentiate processors:

- **Instruction set** – These are set of instructions that the processor can execute.
- **Bandwidth** - These are number of bits that can be executed or processed in a single instruction, usually bits per second.
- **Clock speed** – These determines how many instructions per second the processor can execute, usually in Ghz.

Processor registers

- **Memory address registers** – This contains the address in memory for the next read/write instructions.
- **Memory buffer register**- This contains the data to be written into memory.

Similarly, for I/O we have similar registers

- **Input/ Output Address Register (I/OAR)** – this specify the address of a particular device.
- **Input/ Output Buffer Register (I/OBR)** – this is used for exchange of data between the I/O module and the processor.

How instructions are executed by CPU (Machine cycle)

There are four main steps in executing instructions

1. FETCH – Here the control unit retrieves instructions from memory.
2. DECODE – The control unit interpret the meaning of instruction.
   Those two steps are called I-Time (Instruction Time).
3. EXECUTE – Arithmetic and Logic Unit performs the operations on data that is to be carried out.
4. STORE – The results are stored in memory or a secondary storage device.
   These two other steps are called E-Time (Execution Time).

Traps and Interrupts

These are the events that interrupt the normal sequence of instructions executed by CPU.

Trap - This is abnormal condition which is detected by CPU and is usually an indicative of error, e.g. a program is instructed to divide x by 0 or program told to get a location that doesn't exist.

Interrupt – This is a signal sent to CPU by an external device typically an I/O device requesting the CPU to interrupt its current activities and attend to the interrupting device need.

Normally, before the processor goes to the next instruction, it must check for interrupt before it fetches a subsequent one. The CPU responds to the traps and interrupts by saving the current value of the program counter and resetting it to a new address. This allows the CPU to return execution to the point where it stops before the interrupt or trap occurred.

There are several types of interrupts;

1. **Program Interrupt** – this is generated by some condition that occurs as a result of an instruction. For instance, arithmetic overflow (when assigned memory is inadequate),

reference outside a user allowed memory (user gets into other user's file and it's not shared file).

2. **Timer interrupts** – these are generated by the timer within the processor and allow the OS to perform certain functions on regular basis. E.g. when screen turns blank if the user has stopped working.
3. **I/O interrupt** – this is generated by I/O controller to signal normal completion of an operation or a variety of error condition.
4. **Hardware failure interrupts** – this is usually generated by such hardware as power unit or a memory parity error. Parity bit checks whether error occurred during data transmission.

Interrupt processing

The occurrence of interrupt triggers a number of events both in the processor and software. For example, when an I/O device completes an I/O operation, the following hardware events must occur.

i. The device issues an interrupt signal to the processor.
ii. The processor finishes execution of current instruction before responding to the interrupt.
iii. The processor tests for an interrupt and if there is one, it sends an acknowledgment signal to the device that issued the interrupt. This acknowledgement allows the device to remove its interrupt.
iv. The processor now prepares to transfer control to the interrupt routine. Before it does so, it saves the information needed to resume the current program at the point of interrupt.
v. The processor now loads the program counter with the entry location of the interrupt handling program that will respond to this interrupt.