TASK 1 : Two ways to modify the architecture of CPU, what is the best ?, and why ?

---

1. Von Neumann Architecture:

- Von Neumann Architecture is a digital computer architecture whose design is based on the concept of stored program computers where program data and instruction data are stored in the same memory.
- It is ancient computer architecture based on stored program computer concept.
- Same physical memory address is used for instructions and data.
- There is common bus for data and instruction transfer.
- Two clock cycles are required to execute single instruction.
- It is cheaper in cost.
- CPU can not access instructions and read/write at the same time.
- It is used in personal computers and small computers.

2. Harvard Architecture:

- Harvard Architecture is the digital computer architecture whose design is based on the concept where there are separate storage and separate buses (signal path) for instruction and data. It was basically developed to overcome the bottleneck of Von Neumann Architecture.
- It is modern computer architecture based on Harvard Mark I relay based model.
- Separate physical memory address is used for instructions and data.
- Separate buses are used for transferring data and instruction.
- An instruction is executed in a single cycle.
- It is costly than Von Neumann Architecture.
- CPU can access instructions and read/write at the same time.
- It is used in micro controllers and signal processing.

Task 2 : What is the difference between Stack and Heap ?

Memory System

- System of memory is separated into two areas, the stack and the heap.
- Value types, like local variables, are stored in the stack.
- Reference types, like class instances, are stored in the heap.
- When we use the new keyword, we allocate space on the heap.
- Before members can be accessed, we must create a reference on the stack that points to the allocation on the heap.

- When we overwrite a reference from the stack to point to a new allocation on the heap, we won't have access to the previous reference.
- At some point in the future, the garbage collector will come and clean up the unlinked reference.
- When we assign a local variable to a class member the value is copied.
- The local variable will not have any connection to the class member.

Stack

- Stack is a linear data structure
- Stack memory will never become fragmented
- Stack accesses local variables only
- Stack variables can't be resized
- Stack memory is allocated in a contiguous block
- Stack doesn't require to de-allocate variables
- Stack allocation and deallocation are done by compiler instructions

Heap

- Heap is a hierarchical data structure.
- Heap memory can become fragmented as blocks of memory are first allocated and then freed.
- Heap allows you to access variables globally.
- Heap variables can be resized.
- Heap memory is allocated in any random order.
- in Heap de-allocation is needed.
- Heap allocation and deallocation is done by the programmer.