Part - B

1. Define SMT. How it differs from CMP?

   SMT stands for Simultaneous Multithreading, which is a technique for improving the overall efficiency of super-scalar CPUs with hardware multithreading. SMT permits multiple independent threads of execution to better utilize the resources provided by modern processor architectures.
   The main difference between SMT and CMP is that SMT allows multiple threads to share the same execution resources within a single processor, while CMP uses multiple processors on a single chip to execute multiple threads in parallel. In other words, SMT is a technique for improving the efficiency of a single processor, while CMP is a technique for increasing the processing power of a system by using multiple processors.

2. Differentiate between Strong Scaling and Weak Scaling

   Strong Scaling:
   Strong Scaling refers to the speedup achieved on a multiprocessor without increasing the size of the problem. It aims to reduce the execution time of a fixed-size problem by using more processors.
   Weak Scaling:
   Weak Scaling refers to the speedup achieved on a multiprocessor while increasing the size of the problem proportionally to the increase in the number of processors.It aims to maintain a constant execution time per problem size as the number of processors is increased

3. Differentiate branch taken from branch not taken

   Branch taken:
   Branch taken is a type of branch where the branch condition is satisfied, and the program counter (PC) becomes the branch target. All unconditional jumps are taken branches.
   Branch not taken:
   Branch not taken, also known as an untaken branch, is a type of branch where the branch condition is false, and the program counter (PC) becomes the address of the instruction that sequentially follows the branch.

4. Differentiate UMA from NUMA

   Uniform Memory Access (UMA) :
   UMA stands for Uniform Memory Access, which is a type of multiprocessor in which the latency to any word in main memory is about the same, no matter which processor requests the access.
   Non-uniform Memory Access (NUMA) :
   NUMA stands for Non-Uniform Memory Access, which is a type of single address space multiprocessor in which some memory accesses are much faster than others, depending on which processor requests which word.

5. Give the importance of pipelining

   The importance of pipelining are
   1. Increased Instruction Throughput: Pipelining allows multiple instructions to be executed simultaneously, which increases the overall throughput of the processor.
   2. Reduced Instruction Latency: Pipelining reduces the time it takes to execute an instruction by breaking it down into smaller stages that can be executed in parallel.
   3. Efficient Use of Resources: Pipelining allows the processor to make more efficient use of its resources by overlapping the execution of multiple instructions.

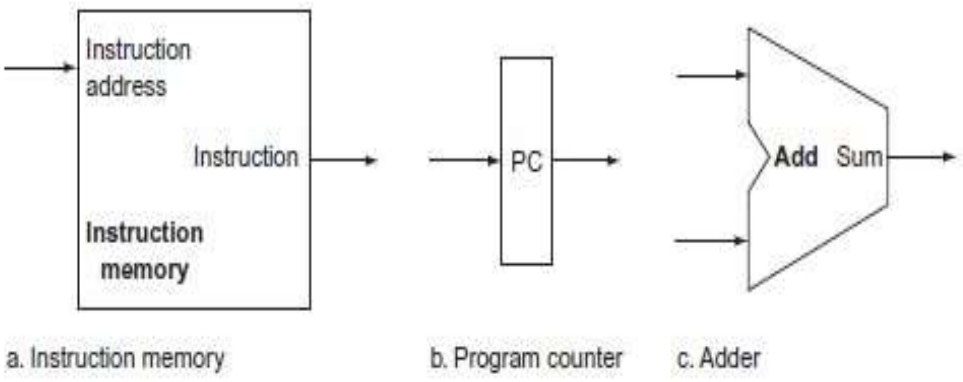
6. List out the cons of pipelining?
    - Data & Control Hazard
    - Branch penalties.
    - In order execution.
    - Increased Complexity
7. List out the seven control signals and define it

    1. RegDst: selects the destination register for the register write operation in the write-back stage of the pipeline.
    2. RegWrite: enables the register write operation in the write-back stage of the pipeline.
    3. ALUSrc: selects the second operand for the ALU operation in the execution stage of the pipeline.
    4. PCSrc: selects the next value of the program counter in the instruction fetch stage of the pipeline.
    5. MemRead: determines whether the instruction in the instruction memory is a load instruction or not.
    6. MemWrite: determines whether the instruction in the instruction memory is a store instruction or not.
    7. MemtoReg: determines whether the data to be written back to the register file comes from the ALU or from the data memory.
8. State the importance of GPU in parallelism

    GPU Parallel computing enables GPUs to break complex problems into thousands or millions of separate tasks and work them out all at once instead of one-by-one like a CPU needs to.

    The GPU parallel computing ability is what makes GPUs so valuable. It is also what makes them flexible and allows them to be used in a wide range of applications, including graphics and video rendering.
9. What are the two state elements needed to store and access an instruction?

    The instruction memory and the program counter are the two state elements that are needed to store and access instructions and an adder is needed to compute the next instruction address.



10. What is ILP?Write down its uses

    ILP stands for Instruction-Level Parallelism, which refers to the potential overlap among instructions that can be executed in parallel. The goal of ILP is to increase the performance of a processor by executing multiple instructions at the same time.

The uses of ILP include:

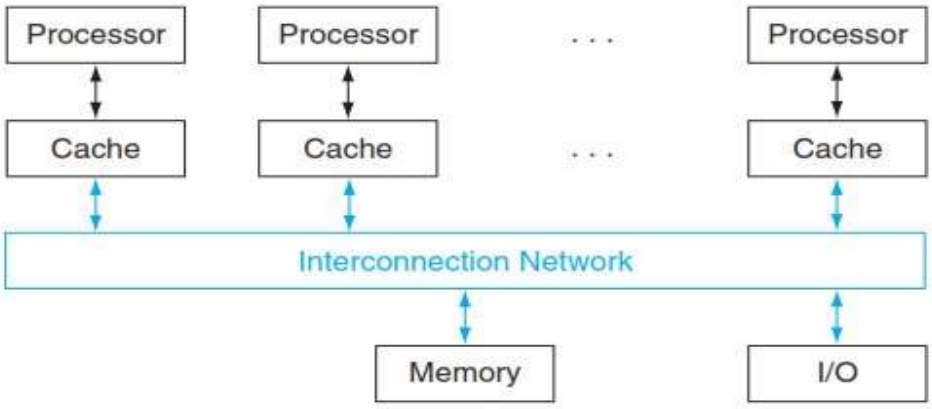
1. Increasing the performance of processors.
2. Reducing the impact of pipeline stalls.
3. Improving the efficiency of instruction execution.
4. Enhancing the performance of superscalar processors

11. What is meant by a data path element?

    A data path element is a unit used to operate on or hold data within a processor. In the MIPS implementation, the data path elements include the instruction and data memories, the register file, the ALU, and adders.

12. What is shared memory multiprocessor

    A shared memory multiprocessor (SMP) is one that offers the programmer a single physical address space across all processors. Processors communicate through shared variables in memory, with all processors capable of accessing any memory location via loads and stores.



13. What is stall in pipelining and give an example

    A stall in pipelining occurs when an instruction cannot proceed to the next stage of the pipeline because the required resources are not available. This can result in a delay in the execution of subsequent instructions, reducing the overall performance of the pipeline.

    An example of a stall in pipelining is a data hazard, where an instruction depends on the result of a previous instruction that has not yet completed. In this case, the pipeline must stall until the required data is available, which can result in a delay in the execution of subsequent instructions.

14. What is the need of multicore processors?

    A multicore processor is a single computing component with two or more “independent” processors called “cores”. The multiple cores can run multiple instructions at the same time, increasing overall speed for programs amenable to parallel computing. This allows for more efficient processing of tasks that can be divided into smaller sub-tasks that can be executed simultaneously. Multicore processors also allow for better utilization of system resources and can improve system performance for applications that require high levels of processing power.

15. Write down the five stages of instruction execution

    1. Instruction Fetch & PC Increment (IF)
    2. Instruction Decode and Register Read (ID)
    3. Execution or calculate address (EX)
    4. Memory access (MEM)

5. Write back into register (WB)

16. Write the instruction format for the jump instruction.

The destination address for a jump instruction is formed by concatenating the upper 4 bits of the current PC + 4 to the 26-bit address field in the jump instruction and adding 00 as the 2 low-order bits.

Field	000010	address
Bit positions	31:26	25:0

PART - C

1. What is pipelining? What is a hazard? Discuss the three types of hazards with suitable examples and also illustrate the techniques to overcome them.
2. Basic MIPS implementation with necessary multiplexer and control lines
3. Building Datapath
4. Flynn's classification
5. Fine grained multithreading, coarse grained multithreading and simultaneous multithreading
6. Compare Shared and Distributed Memory Multiprocessor
7. Describe about the symmetric and asymmetric cores with necessary layers