EX L13 Advanced Architectures

1. Indicate whether each of the following applies to CISC or RISC by placing either a C (for CISC) or an R (for RISC) in the blank.

__R__ 1. Simple instructions averaging 1 clock cycle to execute
__C__ 2. Single register set
__R__ 3. Complexity is in the compiler
__R__ 4. Highly pipelined

2. A RISC processor has 8 global registers and 10 register windows. Each window has 4 input registers, 8 local and 4 output. How many total registers are in this CPU? (HINT: Remember, due to the circular nature of the windows, the output registers of the last window are shared as the input registers of the first window.)

Ans

Global registers + Register windows(Local * Input|Output) = Total registers

8 + 10(8 + 4) = 128 total registers

3. A RISC processor has 152 total registers, with 12 designated as global registers. The 10 register windows each have 6 input registers and 6 output registers. How many local registers are in each register window set?

Ans

Total registers = Global registers + Register windows(Local + Input|Output)

((Total registers – Global registers)/Register windows) - Input|Output = Local

((152 – 12)/10) – 6 = 8 local

4. How are SIMD and MIMD similar? How are they different? Note, you are not to define the terms, but instead compare the models.

Ans

Similar

1.) Both are models of parallel computing that enable multiple tasks to be executed simultaneously.

2.) Both are used to improve the performance and efficiency of computing systems.

Different

1.) Approach to parallel processing: SIMD applies a single instruction to multiple data items in parallel, while MIMD allows multiple instructions to be executed on multiple data items in parallel.

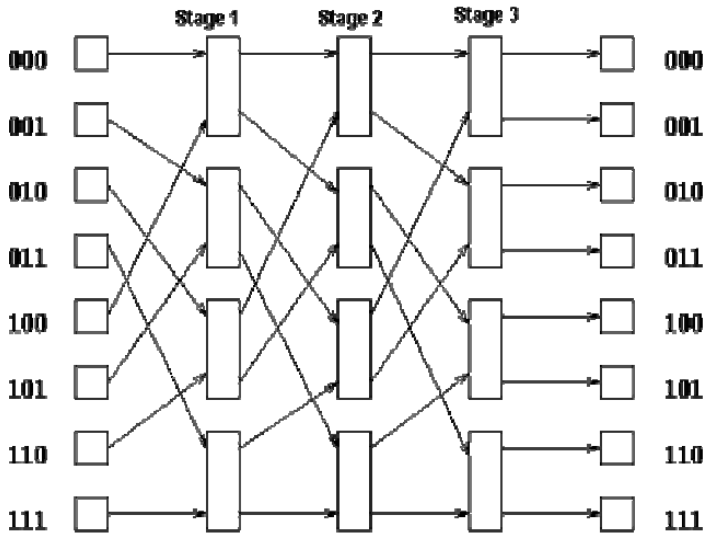
2.) Hardware requirements: SIMD typically requires specialized hardware, such as a GPU, to achieve parallel processing, while MIMD can be implemented using more general-purpose hardware, such as multi-core CPUs.

3.) Flexibility: MIMD is more flexible than SIMD because it allows for multiple instructions to be executed on different data items simultaneously, making it better suited for a wider range of applications.

4.) Performance: SIMD is best suited for applications that require a large amount of data to be processed in parallel, while MIMD is better suited for applications that require more complex processing, such as scientific simulations and weather modeling.

5.) Programming complexity: SIMD programming can be simpler than MIMD programming because the same instruction is applied to all data elements, but MIMD programming can be more complex due to the need to manage multiple instructions and data dependencies.

5. Given the following Omega network, which allows 8 CPUs (P0 through P7) to access 8 memory modules (M0 through M7):



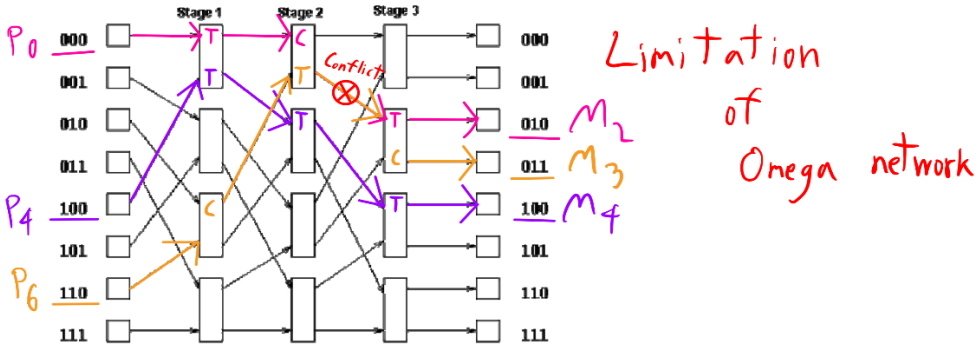
a. Show the following connections through the network:

i) P0 → M2

ii) P4 → M4

iii) P6 → M3

Ans



b. Can these connections occur simultaneously or do they conflict? Explain.

Ans i and iii conflict

c. List a processor-to-memory access that conflicts (is blocked) by the access P0 M2 that is not listed in part (a).

Ans iii

d. List a processor-to-memory access that is not blocked by the access P0 M2 and is not listed in part (a).

Ans ii

6. For what type of program-level parallelism (data or control) is SIMD best suited? For what type of program-level parallelism is MIMD best suited?

Ans

SIMD is best suited for data-level parallelism

MIMD is best suited for control-level parallelism.

7. Why are distributed systems desirable?

Ans

Distributed systems offer scalability, fault tolerance, flexibility, speed, and security advantages over centralized systems. They can handle large amounts of data, continue to function even if one or more machines fail, can be customized to meet different needs, perform computations faster, and are less vulnerable to attacks.

8. Flynn's taxonomy consists of four primary models of computation. Briefly describe each of the categories and give an example of a high-level problem for which each of these models might be used.

Ans

SISD: This type of architecture can process only one instruction and one data stream at a time. It is suitable for tasks that involve complex mathematical calculations.

MISD: This type of architecture is not commonly used as it is difficult to find tasks that can be efficiently solved using it. It involves processing multiple instruction streams on a single data stream.

SIMD: This type of architecture is used for parallel processing tasks such as graphics and video rendering. It can process multiple data streams at the same time using the same instruction.

MIMD: This type of architecture is commonly used in modern parallel computing systems and is considered the most efficient model for high-performance computing applications. MIMD allows multiple processors to execute different instructions on different data sets, which makes it suitable for tasks that require a high degree of parallelism and scalability.