Pipelined Microprocessor

A **pipelined microprocessor** is an advanced design that improves the efficiency and speed of instruction execution by overlapping the execution of multiple instructions. Instead of waiting for one instruction to complete before starting the next, pipelining allows the microprocessor to process several instructions simultaneously, breaking down the execution process into distinct stages. This technique is fundamental to modern high-performance CPUs.

Key Concepts:

- **Stages:** Pipelining divides instruction execution into several stages, typically including Fetch, Decode, Execute, Memory Access, and Write Back. While these are the typical stages, it depends on the particular processor's architecture.
- **Throughput:** The main advantage of pipelining is increased throughput—the number of instructions executed per unit of time.
- **Latency:** While individual instruction latency may not decrease, the overall processing time for a series of instructions is reduced.
- **Hazards:** Pipelining introduces challenges like data hazards (dependencies between instructions), control hazards (due to branches), and structural hazards (resource conflicts).

Sample Instruction Execution in a Pipelined Microprocessor

- Let's consider a simple 4-stage pipeline with the following stages: *Fetch (F), Decode (D), Execute (E),Write Back (W)*
- Below is an illustration of how 3 instructions would be executed in a pipelined microprocessor over five clock cycles:

Cycle	Instruction 1	Instruction 2	Instruction 3
1	Fetch (F)		
2	Decode (D)	Fetch (F)	
3	Execute (E)	Decode (D)	Fetch (F)
4	Write Back (W)	Execute (E)	Decode (D)
5		Write Back (W)	Execute (E)
6			Write Back (W)

Amdahl's Law and Performance Comparison

- Calculates the performance gain that can be obtained by improving some portion of a computer
- Performance improvement to be gained from using some faster mode of execution is limited by the fraction of the time the faster mode can be used.
- Amdahl's law defines the speedup that can be gained by using a particular feature.

$$\text{Speedup} = \frac{\text{Execution time for entire task without using the enhancement}}{\text{Execution time for entire task using the enhancement when possible}}$$

$$SystemSpeedUp = \frac{1}{\frac{f}{S} + (1 - f)}$$

Machine A is n times faster than machine B iff
perf(A)/perf(B) = time(B)/time(A) = n

Machine A is x% faster than machine B iff
perf(A)/perf(B) = time(B)/time(A) = 1 + x/100

E.g. time(A) = 10s, time(B) = 15s
- 15/10 = 1.5 => A is 1.5 times faster than B
- 15/10 = 1.5 => A is 50% faster than B

Questions

a. Consider a pipelined processor with four stages: fetch instruction (FI), decode instruction and calculate addresses (DA), fetch operand (FO), and execute (EX).
   i. Draw a diagram for a sequence of 10 instruction executions with the clock cycles. (Assume that there are no data dependencies).
   ii. How many clock cycles are need to execute all 10 instructions?
   iii. Citing the above questions describe the advantage of a pipelines processor.

b. Compare and contrast hardwired and micro programmed control unit architectures. In your response, address the following aspects seperately
   i. Definition and Basic Operation
   ii. Design Complexity and Flexibility
   iii. Performance and Speed
   iv. Application and Use Cases
   v. Advantages and Disadvantages (Summarize the main advantage and disadvantage of hardwired and micro programmed control units.)

c. Compare and Contrast Cisc and Risc Architecture.
d. Machine A: clock 1 ns, CPI 2.0, for program x, Machine B: clock 2 ns, CPI 1.2, for prog ram x. Which is faster and how much?
e. Keep clock(A) at 1 ns and clock(B) at 2 ns. For equal performance, if CPI(B)=1.2, what is CPI(A)?
f. Keep CPI(A)=2.0 and CPI(B)=1.2 For equal performance, if clock(B)=2 ns, what is clock(A)?
g. Your boss asks you to improve performance by:
   i. Improve the ALU used 95% of time by 10%
   ii. Improve memory pipeline used 5% of time by 10x

   Let f=fraction speed up and s = speedup on that fraction