Change Assignment 1 ,Question 2 into

Assignment 1

2. Explain the **Speedup** in case of the following instruction level pipeline architectures: linear pipeline ,superscalar pipeline and superpipeline architectures?

The following slides explain concepts of CPI and throughput of superscalar pipeline, superpipeline and VLIW processor architectures .

CPI-Superscalar

- https://en.wikipedia.org/wiki/Cycles_per_instruction

classic RISC pipeline, with the following 5 stages:

Instruction fetch cycle (IF).

Instruction decode/Register fetch cycle (ID).

Execution/Effective address cycle (EX).

Memory access (MEM).

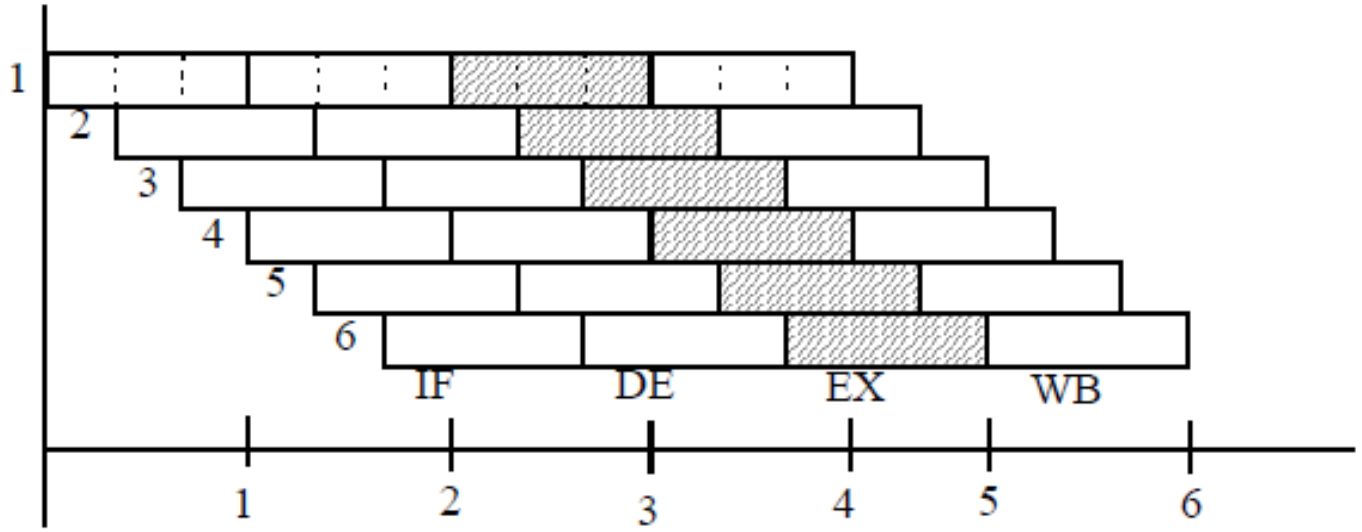
Write-back cycle (WB).

Each stage requires one clock cycle and an instruction passes through the stages sequentially. Without pipelining, a new instruction is fetched in stage 1 only after the previous instruction finishes at stage 5, therefore the number of clock cycles it takes to execute an instruction is 5 (CPI = 5 > 1). In this case, the processor is said to be *subscalar*. With pipelining, a new instruction is fetched every clock cycle by exploiting instruction-level parallelism, therefore, since one could theoretically have 5 instructions in the 5 pipeline stages at once (one instruction per stage), a different instruction would complete stage 5 in every clock cycle and on average the number of clock cycles it takes to execute an instruction is 1 (CPI = 1). In this case, the processor is said to be *scalar*.

CPI Superscalar contd..

- With a single-execution-unit processor, the best CPI attainable is 1. However, with **a multiple-execution-unit processor**, one may achieve even better CPI values (CPI < 1). In this case, the processor is said to be *superscalar*. To get better CPI values without pipelining, the number of execution units must be greater than the number of stages. For example, with 6 executions units, 6 new instructions are fetched in stage 1 only after the 6 previous instructions finish at stage 5, therefore on average the number of clock cycles it takes to execute an instruction is 5/6 (CPI = 5/6 < 1). To get better CPI values with pipelining, there must be at least 2 execution units. For example, with 2 executions units, 2 new instructions are fetched every clock cycle by exploiting instruction-level parallelism, therefore 2 different instructions would complete stage 5 in every clock cycle and on average the number of clock cycles it takes to execute an instruction is 1/2 (CPI = 1/2 < 1).
- If CPI = ½ then Instructions/clock = 2. If the frequency of the clock is known ie f clocks/sec
- then
  - Instructions /sec= Instructions/clock * f clocks/sec= 2 * f/sec = throughput

- Superpipelined: cycle time = 1/m of baseline
  - Issue parallelism = IP = 1 inst / minor cycle
  - Operation latency = OP = m minor cycles
  - Peak IPC = m instr / major cycle



Instruction Level Parallelism Machine

CPI and Throughtput Superpipelined machines

- Superpipelined machines can issue one instruction per minor cycle, minor cycle time is shorter than the time required for any operation(stage)-major cycle..

Cycle time shorter means frequency (f) higher of minor cycle.

Throughput = function(f) (See previous slide 3)

In previous slide

1 major cycle = 3 minor cycles

i.e 3 instuctions / 1 major cycle

3 * f(frequency of major clock cycle) = throughput

Therefore

CPI =1/3 major Clock cycle /Instruction

VLIW

- have much lower CPI.
- can issue 2 integer, 2 FP, 1 mem reference, and 1 branch instruction per cycle.
- can issues as many as 6 instructions at a time