**Pipelining Lecture Assignment**

**Dr. Abraham**

1. **Name** Pipe 5 Stages for MIPS - Instruction Fetch, Instruction Decode, Execute, Memory access and Write Back.

Define Throughput – how often an instruction exits the pipeline

define Machine cycle – time to move through one pipe stage

define Stall – a situation in which a pipe stage requires more than one machine cycle for completion.

How is Time per instruction on a pipelined computer calculated?

TPI = TPI unpipelined / number of pipe stages

1. If an unpipelined processor has a 1 ns clock cycle and that it uses:

4 cycles for ALU operation and branches

5 cycles for memory operations .

Their relative frequencies are: 40% ALU operations, 20% Branch, and 40% for Memory.

What is the average instruction execution time?

1 ns x [(40% + 20%) x 4 cycles ALU + 40% x 5 cycles memory] = 4.4 ns

1. If you were to pipeline this processor, one instruction should be completed each cycle (ignoring filling at start and emptying pipeline at the end). Pipelining adds and overhead of .2ns for various things such as latching. So effective time it takes for one instruction is 1.2 ns. What is the speed up gained by pipelining?

4.4 ns / 1.2 ns = 3.7 times

1. Name and explain pipeline hazards.

Structural Hazard – when because of a lack of resources the hardware cannot support all requests at the same time

Data hazard –When the data requires a previous instruction execution

Control hazard – When the program counter gets a different value because of branches so the next instruction fetched becomes unusable.

1. Explain the purpose of data forwarding.

When a register must be written before reading. So, we wait until the WB stage, even though the data was calculated by CPU. Also, data can be forwarded to a instruction directly without waiting.

1. Explain RAW, WAR and WAW data hazards.

Read after Write: If the write is incomplete, the next instruction using the same register will get old data. MIPS only has this hazard.

Write after Read: Since write should come first, this can cause inconsistent data.

Write after Write: If the second write gets there first because of a long instruction. Basically a situation where the second write writes before the first write.