Chapter 06: Instruction Pipelining and Parallel Processing

Lesson 05:

Structure hazards and Score boarding

Objective:

- Learn stall due to structural hazards and ways of reducing them

Stalls due to Structure Hazards

Stalls Due to Structural hazards

- Structure hazards occur when the processor's hardware is not capable of executing all instructions in the pipeline simultaneously

Example of Stall Due to Structural hazard

- Register file did not have enough ports to allow an instruction in the WB (write back) stage to write its result into the register file in the same cycle that another instruction in the RR (register read) stage read from the register file

Necessary to stall any instruction in the RR stage

- When there was also an instruction in the WB stage on that cycle
- Choosing to stall the instruction in the WB stage to allow the instruction in the RR stage to proceed would be a poor choice, as the stall in the WB stage would prevent instructions at the EX (execution) stage from advancing

More than one cycle/instruction Case

- Structural hazards within a single pipeline are relatively rare on modern processors
- Because their hardware and instruction sets have been designed to support pipelining

More than one cycle/instruction Case

- However, processors that execute more than one instruction in a cycle, often have restrictions on the types of instructions that the hardware can execute simultaneously

Example

- A process might be able to execute two instructions in each cycle, but only if one of the instructions was an integer operation and the other a floating-point computation

Overcoming Register Read Stage Structure hazards

Overcoming Register Read Stage Structure hazards

- Pipelined processors need to keep track of which registers will be written by instructions that are already in the pipeline
- Subsequent instructions determine whether their input registers available when they reach the RR stage

Register score boarding

- In score boarding, a bit, known as the *presence* bit, is added to each register in the register file
- The presence bit records whether the register is available for reading (full) or waiting for an instruction to write its output value (empty)

P-Bit for Score Boarding for presence of data

P	Register bits
P	Register bits
P	Register bits
P	Register bits

P = 1, when data available for read and 0 just after read so the register available for write back of result

Operation

- When an instruction enters the RR, the hardware checks to see if all of its input registers are full
- If so, the hardware reads the values of all the input registers

Operation

- Marks the output register of instruction empty, and allows the instruction to proceed to the execute stage on next cycle

Operation

- If not, the hardware holds the instruction in the register read stage until its input values become full, inserting bubbles into the execute stage on each cycle until this happens

Operation

- When instruction reaches the write back stage and writes its result into its destination register, that register marked full, allowing operations that read the register to proceed

Summary

We learnt

- Structure hazard
- Score boarding a register by Presence-bit for overcoming structure hazard
- P = 1, when data available for read and 0 just after read so the register available for write back of result

End of Lesson 05 on
Structure hazards and Score boarding