Toby Chappell

CPSC 250

November 25, 2019

Homework 4

Processor Datapath

1. Would a two-stage instruction pipeline cut the instruction cycle time in half, compared with the use of no pipeline? (5 pts)

It is unlikely due to the fact that the execution time will normally take longer than the fetch time, meaning that the fetch stage may be forced to wait until it can empty its buffer. Furthermore, if the program encounters a conditional branch instruction, the address of the next instruction to be fetched is unknown. Therefore, the fetch stage must wait to receive the next instruction address from the execute stage and then the execute stage may have to then wait for the next instruction to be fetched.

1. A microprocessor is clocked at a rate of 5 GHz. (4 pts)
   1. How long is a clock cycle?

Clock cycle = 1/frequency = 1/5,000,000,000 clock cycles/second =

2\*10^-10 seconds/clock cycle = 0.2 ns/clock cycle

* 1. What is the duration of a particular type of machine instruction consisting of three clock cycles?

2\*10^-10 seconds \* 3 clock cycles = 6\*10^-10 seconds = 0.6 ns

1. Assume a pipeline with four stages: fetch instruction (FI), decode instruction and calculate addresses (DA), fetch operand (FO), and execute (EX). Draw a diagram for a sequence of 7 instructions, in which the third instruction is a branch that is taken and in which there are no data dependencies. (7 pts)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instructions | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| I1 | FI | DA | FO | EX |  |  |  |  |  |  |
| I2 |  | FI | DA | FO | EX |  |  |  |  |  |
| I3 |  |  | FI | DA | FO | EX |  |  |  |  |
| I4 |  |  |  | FI | DA | FO |  |  |  |  |
| I5 |  |  |  |  | FI | DA |  |  |  |  |
| I6 |  |  |  |  |  | FI |  |  |  |  |
| I7 |  |  |  |  |  |  | FI | DA | FO | EX |

1. A pipelined processor has a clock rate of 2.5 GHz and executes a program with 1.5 million instructions. The pipeline has five stages, and instructions are issued at a rate of one per clock cycle. Ignore penalties due to branch instructions and out-of-sequence executions. (6 pts)
   1. What is the speedup of this processor for this program compared to a non-pipelined processor (ignore the cost for initial filling and emptying the pipeline)

Speedup = performance of pipelined/performance of nonpipelined = execution time of nonpipelined/execution time of pipelined = (instructions\*stages)/(stages+(instructions-1)). When instructions significantly more than stages, Speedup = instructions\*stages/instructions = 5

* 1. What is throughput (in MIPS) of the pipelined processor?

Throughput = instructions/(instructions\*stages) = 1/stages = 0.2

1. A microprocessor provides an instruction capable of moving a string of bytes from one area of memory to another. The fetching and initial decoding of the instruction takes 10 clock cycles. Thereafter, it takes 15 clock cycles to transfer each byte. The microprocessor is clocked at a rate of 10 GHz. (3 pts)
   1. Determine the length of the instruction cycle for the case of a string of 64 bytes.

(10+15\*64)/(10000000) = 9.7\*10^-8 ns = 97 ns

1. Consider the following sequence of instructions: (8 pts)

*or r1, r2, r3*

*or r2, r1, r4*

*or r1, r1, r2*

* 1. Indicate dependences and their type.

On r1 between the first and second or

On r1 between the first and third or

On r2 between the second and third or

* 1. Assume there is no forwarding in this pipelined processor. Indicate hazards and add *nop* instructions to eliminate them.

There are multiple data hazards, resolved by adding two nop instructions between each or instruction

*or r1, r2, r3*

*nop*

*nop*

*or r2, r1, r4*

*nop*

*nop*

*or r1, r1, r2*

1. Consider a 5-stage pipelined processor. Identify all of the data dependencies in the following code. Which dependencies are data hazards that can be resolved via forwarding? Which dependencies are data hazards that would cause a stall? (5 pts)

*add $3, $4, $2*

*sub $5, $3, $1*

*lw $6, 200($3)*

*add $7, $3, $6*

On $3 between first add and sub – resolved via forwarding

On $3 between first add and lw – creates stall due to load-use hazard

(last add dependent on first add, but takes place after the first add is executed regardless if forwarding used)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Instructions | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| add | FI | DA | EX | MEM | WB |  |  |  |  |
| sub |  | FI | DA | EX | MEM | WB |  |  |  |
| nop |  |  | - | - | - | - | - |  |  |
| lw |  |  |  | FI | DA | EX | MEM | WB |  |
| add |  |  |  |  | FI | DA | EX | MEM | WB |

1. Suppose you are designing a microprocessor that uses special instructions to access I/O devices (instead of mapping the devices to memory). What special instructions would you need to include? (3 pts)

You would need instructions to via memory mapped I/O to read and/or write into the following 3 types of registers: status registers (provide status information), configuration/control registers (configure and control the device), and data registers (read data from or send data to the device).

1. A hardware architect asks you to choose between a single 32-bit bus design that multiplexes both data and address information across the bus or two 16-bit buses, one used to send address information and one used to send data. Which design would you use choose and why? (4 pts)

A single 32-bit design is the better design because it could for one store more data compared to 2 16-bit buses. Additionally, it would require less hardware (multiplexing allows for the reduction of the number of buses which means reducing the number of connectors, crimps, and splices).

1. List the steps taken by the processor’s hardware when handling an interrupt. (5 pts)

The processor reads the cause and transfers to the relevant handler. It then determines action required. If restartable, it takes corrective action and uses EPC to return to program. Otherwise, terminates program and reports error using EPC.