ELEC 5200-001/6200-001 (Spring 2019)
Homework 4 - Solution
Assigned 03/18/19, due 03/25/19
(45 points possible)

Problem 1: Read the article (*Communications of the ACM*, volume 59, number 3, pp. 40-45, March 2016) in which David Patterson (an author of your textbook) interviews John Hennessey (the other author of textbook). Then write an opinion in 3-4 sentences on what you think is the most significant idea that Hennessey expresses about computer architecture. The article is available on the Canvas course website (attached to the homework assignment, look for the file p40-patterson.pdf). (4 points)

Answer: Any thoughtful answer will be acceptable.

Problem 2: Assume that a pipeline register and the program counter each have a delay of 100ps. Time taken by other major hardware units in a MIPS processor are:

Memory read or write	800ps
Register file read and write	450ps
ALU	1ns
Add unit	1ns

All other hardware, including the control and multiplexers, have negligible delays. Then,

a. What is the maximum clock frequency for a five-stage MIPS pipeline datapath? (2 points)

Answer:
For a pipeline, the clock cycle time should accommodate the longest hardware unit (ALU, 1ns) and a register (0.1ns). Thus, the maximum clock frequency = 1/(1.1ns) = 0.909 GHz or 909 MHz.

b. Neglecting any hazards and resulting stalls, what are the pipeline latency (in units of time) and the cycles per instruction (CPI) for a long instruction sequence? (2 points)

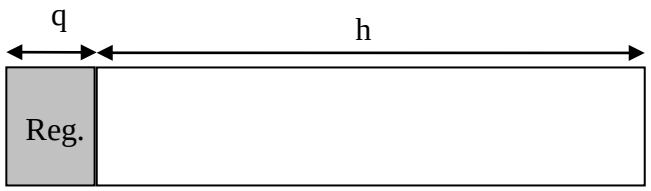
Answer:
Latency is 5 clock cycles or 5×1.1ns = 5.50ns. For a long instruction sequence, we can neglect the pipeline latency. Then, assuming no stalls in the pipeline, CPI = 1.

c. Compare this datapath with a single cycle MIPS datapath using similar hardware: (10 points)
Answer:
Information needed for a comparison is as follows:
Single cycle clock period = 0.100 + 0.800 + 0.450 + 1 + 0.800 + 0.450 = 3.6 ns

Datapath	Clock cycle time	Clock frequency	CPI	Time per instruction	Million instructions per second (mips)
Single-cycle	3.6 ns	278 MHz	1	3.6 ns	278
Pipeline	1.1 ns	909 MHz	1	1.1 ns	909

Note: Show your work!

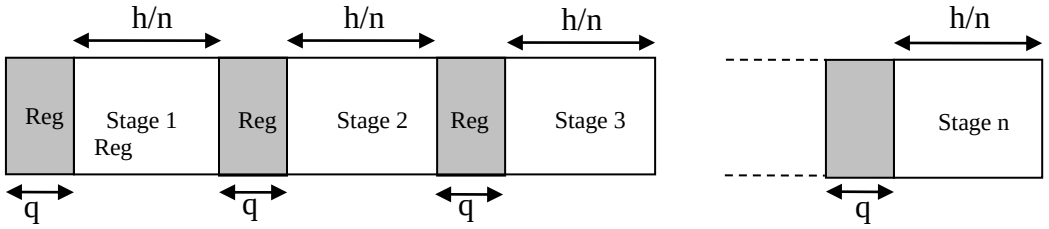
Problem 3: Consider a single-cycle datapath as a one-stage pipeline. It consists of combinational and asynchronous circuitry and a single clocked register, the program counter (PC). Its total cycle time consists of an interval q required by the register and an interval h used by the rest of the circuitry:



(a) Assume that the delay h can be partitioned into n equal delay hardware stages separated by clocked registers each having a delay q, the first stage register being the PC. Neglecting the latency and any hazard penalties, compute the average execution time of an instruction by this n-stage pipeline. (2 points)

Answer:

An n-stage pipeline is shown below:



The total time taken by the combinational and asynchronous circuitry is h. This is equally divided among n stages. Each stage also contains a pipeline register requiring time q. The cycle time of the n-stage pipeline is q + h/n, which is also the average execution time per instruction if we neglect the latency and hazards (ie, for CPI = 1).

(b) Show that the performance limit for the pipeline when we neglect the latency and hazards is determined by the register delay q. Find the upper bound on the clock frequency for this pipeline datapath. (2 points)

Answer:

The pipeline cycle time continues to reduce as the number of stages increases. In the hypothetical limit, $n = \infty$ and the cycle time or the execution time of an instruction becomes q, the register delay. Therefore, an upper bound on the clock frequency is 1/q Hz.

(c) For the n-stage pipeline, suppose the average hazard penalty is $\alpha(n-1)$ cycles per instruction, where $0 < \alpha < 1.0$ and $n \geq 1$. Derive an equation for the optimum number of pipeline stages. (3 points)

Answer:

Average execution time per instruction (with hazards) is given by,

$$t = (q + h/n)(1 + \alpha n - \alpha) = q + q\alpha n - q\alpha + h/n + \alpha h - \alpha h/n$$

To minimize t, we differentiate it with respect to n:

$$\partial t/\partial n = q\alpha - h/n^2 + \alpha h/n^2 = 0$$

or $$n = [h(1-\alpha)/(q\alpha)]^{1/2}$$

This gives the minimum execution time because $\partial^2 t/\partial n^2$ is positive.

(d) Assume a circuit design where $h = 4 \times q$ and an ISA where $\alpha \approx 0.1$. What is the optimum number of stages for this pipeline and what is the corresponding speed up over a single-cycle datapath? (3 points)

Answer:

For $h = 4q$ and $\alpha = 0.1$, optimum stages $n = [4(1 - 0.1)/(0.1)]^{1/2} \approx 6$

i.e. a 6 stage pipeline. For this six-stage pipeline,

Average execution time per instruction = $(q + 4q/6)(1 + 0.1 \times 6 - 0.1) = 2.5q$

For a single-cycle datapath,

Av. execution time per instruction = $h + q = 4q + q = 5q$

Pipeline to single-cycle performance ratio = $5q/(2.5q) = 2$

(e) For q = 90ps, tabulate clock cycle time, clock rate, average CPI (include hazard penalty, if any) and performance in million instructions per second (mips) for single-cycle datapath and the pipeline datapath with optimum number of stages found in (d). (8 points)

Answer: For q = 90ps

Datapath	Cycle time	Clock frequency	Av. CPI	mips
Single-cycle	450ps	2.22 GHz	1.0	2,222
Pipelined	150ps	6.67 GHz	1.5	4,444

Problem 4: The following MIPS instruction sequence is executed on a 5-cycle pipeline datapath, implemented with hazard detection and forwarding units.

```
lw      $t0, 0($t1)
sub     $s2, $t0, $s1
lw      $t4, 4($s2)
add     $s2, $t4, $s1
```

How many bubbles, if any, will be required this instruction sequence? (2 points) Can a compiler improve the performance? (1 point)

Answer: Instructions 1 and 2: Assuming that the first lw starts in cycle 1, it will read memory data for $t0 in cycle 4. The following sub needs that value at the beginning of cycle 4 but it is not available from lw until the end of cycle 4. Hence the sub must be delayed by one cycle by inserting a bubble in the pipeline to allow a correct forwarding.

Instructions 2 and 3: Even though the destination $s2 for sub is the same as the source for lw, the new value of $s2 will be available at the output of the ALU at the time it is needed by lw. Therefore, forwarding will eliminate the hazard.

Instructions 3 and 4: The add needs the value to be written to $t4 by the preceding lw and it will require one bubble to delay add by one cycle.

Thus, the execution of the given four-instruction sequence will require two bubbles.

The two cycle hazard penalty cannot be eliminated in this case.

Problem 5: A program consists of two nested loops, with a branch instruction at the end of each loop and no other branch instruction anywhere. The outer loop is executed 15 times and the inner loop 25 times. Determine the prediction accuracy percentage for the following three prediction strategies:

(a) always predict branch not taken, (2 points)

Answer:

The inner loop is executed 25×15 = 375 times, 360 times branch is taken and 15 times not taken. Similarly, outer loop is executed 15 times where 14 times the branch is taken and 1 time not taken.

A “Branch not taken” strategy yields 374 wrong predictions (14 in the outer loop and 360 in the inner loops) from a total of 390 branches. Accuracy = (16/390) * 100% = 4.1%

(b) always predict branch taken, (2 points)

Answer:

A “Branch taken” strategy yields 16 wrong predictions (1 in outer loop and 15 in the inner loops) from a total of 390 total branches. Accuracy = (374/390) * 100% = 95.9%

(c) use a 1-bit history buffer initialized to the “taken” state for each branch instruction when executed for the first time. (2 points)

Answer:

The 1-bit buffer is initialized to “taken” for each branch instruction. We have 1 wrong prediction for the outer loop and 24 for the inner loops, giving a total of 25 wrong predictions from a total of 390 branches. Accuracy = (365/390) * 100% = 93.6%