MS108: Computer System 1

Spring 2013

Homework #1

Due: Two Weeks from Assignment

TA: Haopeng Liu

Course Email: ms108_2013@163.com

Collaboration Policy

These homework sets will be extremely valuable as tools for learning the material and for doing well on the midterm and final. You are required to obey the following rules:
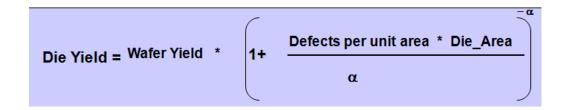(a) Each student should write out their solution independently and in their own words.
(b) Same applies to programming assignments – you should do your own coding.
Above all, make sure that you understand the solution to these homework problems. They really are assigned to help you understand the material and be prepared for the types of problems on the midterm and final!

Q1. Cost and Price

You are trying to figure out whether to construct a new fabrication facility for your IBM Power5 chips. It costs $1.5 billion to build a new fabrication facility. The benefit of the new fabrication is that you predict that you will be able to sell 3 times as many chips at 2 times the price of the old chips. The new chip will have an area of 186 mm2, with a defect rate of .7 defects per cm2. Assume the wafer has a diameter of 300 mm. In both the old and new fabrications, it costs $1000 to fabricate a wafer, and the packaging and testing cost is $20 per chip (after final testing). You were selling the old chips for 40% more than their cost. Assume α = 4, Wafer Yield = 1. The old fabrication process' parameters are shown in the following table.

Chip	Die size (mm^2)	Estimated defect rate (per cm^2)	Manufacturing feature size (nm)	Transistors (millions)
IBM Power5	389	.30	130	276

Note: In this question, you will use equations based on empirical study and approximations. For simplicity, when you need to round to integer numbers, use the mathematical round function (e.g., round 3.24 to 3.2 or round 3.51 to 4) without considering whether the floor or ceiling function would be more appropriate. Remember not all calculation results need to be rounded to integer numbers. Use the formula below for complexity factor:

$$\text{Die Yield} = \text{Wafer Yield} * \left(1 + \frac{\text{Defects per unit area} * \text{Die\_Area}}{\alpha}\right)^{-\alpha}$$

a) With the old fabrication, how many dies can we get from a wafer before we test individual dies? (Round the result to an integer number)

b) With the old fabrication, what is the die yield?

c) What is the cost of the old Power5 chip?

d) With the new fabrication, how many dies can we get from a wafer before we test individual dies? (Round the result to an integer number)

e) With the new fabrication, what is the die yield?

f) What is the cost of the new Power5 chip?

g) What was the selling price of each old Power5 chip?

h) What is the selling price of each new Power5 chip? What is the difference between the selling price and the cost of the new chip?

i) Suppose 50% of the difference between the selling price and the chip cost is your profit. If you sold 500,000 old Power5 chips per month, how long would it take for the accumulated profit to recoup the costs of the new fabrication facility?

Q2. Performance Metrics
An application running on a 1GHz pipelined processor has 55% load-store, 30% arithmetic, and 15% branch instructions. The individual CPIs of these instructions are 5, 4 and 4, respectively.

a) Determine the overall CPI of this program execution on the given processor.

A new embedded version of the processor is being modified to operate at 600 MHz. In this new version, the individual CPIs of load-store and arithmetic instructions are remaining unchanged. However, the individual CPI of branch instructions is getting stretched to 6 clock cycles. A new compiler is also developed for the new processor which eliminates 25% of load-store and 5% of arithmetic instructions for the given application (e.g., if there were 1 million load-store instructions before, now the program compiled by the new compiler would have only 750000).

b) Determine the overall CPI of this program execution on the new processor together with the new compiler technology.

c) Determine the factor by which the application will run faster or slower on the new processor with the new compiler technology.

Q3. Amdahl's Law

Suppose there is a program which takes 200 seconds to execute. Of this time, 30% is used for multiplication, 60% for memory access instructions and 10% for other tasks. Suppose your goal is to enhance the performance of a processor by 2 times and there are two ways of doing so: either make multiply instructions run faster than before, or memory access instructions run faster than before, but not both.

a) How much shall we improve the memory access instructions in order to achieve the performance enhancement? Is it possible?

b) How much shall we improve the multiplication in order to achieve the performance enhancement? Is it possible?

c) Suppose we now improve the memory access by 5 times (Time(old execution) / Time(new execution) = 5). Unfortunately, the design makes multiplication slow down by 20% (i.e., Time(old execution) / Time(new execution) = 5/6). What will the overall speedup be?

Q4. Pipelining

Consider the code segment below. Assume that full bypassing/forwarding has been implemented. Assume that the initial value of register R23 is much bigger than the initial value of register R20. Assume that all memory references take a single cycle. Assume that both load-use hazards and branch delay slots are hidden using delay slots. You may-not reorder instructions to fill such slots, but if a subsequent instruction is independent and is properly positioned, you may assume that it fills the slot. Otherwise, fill slots with additional no-ops as needed.

```
LOOP:      lw     R10, X(R20)
           lw     R11, Y(R20)
           subu   R10, R10, R11
           sw     Z(R20), R10
           addiu  R20, R20, 4
           subu   R5, R23, R20
           bnez   R5, LOOP
           nop                      ; 1 delay slot
```

a) Draw a pipeline diagram of 2 iterations of its execution on a standard 5-stage MIPS pipeline (for clarity, use graph paper or a computer). Assume that the branch is resolved

using an ID control point. In the box below, write the total number of cycles required to complete 2 iterations of the loop.

Cycles =	

b) On the second grid page that follows, draw a pipeline diagram of 2 iterations of the loop on the pipeline below. Note that the loop has a single branch delay slot nop included – you may need to add more. You cannot assume anything about the program's register usage before or after this code segment. Fill in the boxes below.

Pipeline Branch Delay =	
Pipeline Load Delay =	
Cycles =	

Pipeline:

IF1	IF2	ID	RF	EX1	EX2	M1	M2	WB				
	IF1	IF2	ID	RF	EX1	EX2	M1	M2	WB			
		IF1	IF2	ID	RF	EX1	EX2	M1	M2	WB		
			IF1	IF2	ID	RF	EX1	EX2	M1	M2	WB	
				IF1	IF2	ID	RF	EX1	EX2	M1	M2	WB
					IF1	IF2	ID	RF	EX1	EX2	M1	M2	WB

IF1: Begin Instruction Fetch
IF2: Complete Instruction Fetch
ID: Instruction Decode
RF: Register Fetch
EX1: ALU operation execution begins. Branch target calculation finishes. Memory address calculation. Branch condition resolution calculation begins.
EX2: Branch condition resolution finishes. Finish ALU ops. (But branch and memory address calculations finish in a single cycle).
M1: First part of memory access, TLB access.
M2: Second part of memory access, Data sent to memory for stores OR returned from memory for loads.
WB: Write back results to register file

Q5. Pipeline Hazards

Identify all dependences by type in the code segment below; list the two instructions involved; identify which instruction is dependent; and, if there is one, name the storage location involved.

```
1   LD      R1, 45(R2)
2   DADD    R7, R1, R5
3   DSUB    R8, R1, R6
4   OR      R1, R5, R1
5   BNEZ    R7, branch_target
6   DADD    R10, R10, R5
7   XOR     R2, R3, R4
```

	Dependence type	Independent instruction	Dependent instruction	Storage location
1 (example)	RAW	1(LD)	2(DADD)	R1
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				

Q6. Scoreboarding

For the code sequence shown below, draw a pipeline diagram of how instructions would issue in a machine using scoreboarding as discussed in class. Use the execution mix and scoreboard structure as given in the class example. Assume that the FP Add unit has 4 EX phases, the FP Multiply unit has 7 EX phases, and divide has 24 EX phases. FP Adds, Subtracts, and Multiplies are fully-pipelined, while divide operations are NOT pipelined.

```
LD      F6, 12(R2)
LD      F2, 16(R3)
ADDD F0, F2, F4
DIVD  F10, F0, F6
SUBD F8, F6, F2
ADDI  R2, R2, 8
ADDI  R3, R3, 16
ADDD F6, F8, F2
```

Q7. SimpleScalar Assignment

Go to http://www.simplescalar.com look at the "Documentation" under the Support category. Browse through the documents available there – in particular, the user's guide (v2 is fine) and hacker's guide (v2) will be useful.

For this assignment we will use four simple benchmarks, "anagram", "cc1" (gcc), "compress", and "go". The benchmarks are in the "Instructor benchmarks" under the "Benchmarks" category. Download the benchmarks and de-compress them into your local folder. The README in this folder describes how to run the benchmarks. Some of these benchmarks may take a bit of time to run, depending on which simulator is used. You may want to write a shell/perl script to automate running the benchmarks.

Use sim-profile to collect instruction profiles, instruction-class profiles, branch profiles, and address mode profiles for the four benchmarks for the Alpha-ISA binaries. Summarize the results in your report -- do not just print out the dump from the simulator – mention some interesting characteristics (top 10 instructions, popular addressing modes, etc) that these benchmarks utilize.