CprE 381 Homework 6

[Note: This homework's purpose is to increase your comfort at calculating and analyzing the performance of processors. By the end, you should be able to accurately estimate how a change to software or hardware will likely impact the overall execution time of an application.]

1. Performance Analysis

   You are in charge of selecting processors for a new embedded chore-reminder system you and your roommates are building. You are considering two processors, A and B (the only ones you and your roommates can afford) that have the following CPIs:

Instruction Type	Cycles per Instruction	
	Processor A	Processor B
Arithmetic, Logical, Shifts, Stores	1	4
Jumps	1	2
Conditional Branch	1	2
Loads	1	5

   The following applications are the primary ones that you will need to run on your system:

```
# Application 1:
# This is an implementation of a thresholding algorithm.
# $5 contains size $4 contains &in (in is word-sized array) $2 contains
count
  blez  $5,exit
  sll   $5,$5,2
  addu  $5,$4,$5
  addu  $2,$0,$0
loop:
  addiu $4,$4,4
  lw    $3,0($4)
  slt   $3,$6,$3
  addu  $2,$2,$3
  bne   $4,$5,loop
exit:

# Application 2:
# This is an implementation of tiny encryption algorithm (tea8).
# The value inputs are in $6 and $10-$14 as half words.
  addiu    $7,$0,0x0a00
  addiu    $9,$0,0x5a00
loop:
  andi     $8,$6,0xffff
  sll      $5,$8,4
  addu     $2,$8,$7
  sra      $3,$6,5
  addu     $5,$13,$5
  xor      $5,$5,$2
  addu     $3,$12,$3
  xor      $3,$5,$3
```

```
addu    $2,$3,$14
andi    $2,$2,0xffff
sll     $14,$2,16
sll     $5,$2,4
sra     $14,$14,16
addu    $2,$2,$7
addu    $5,$11,$5
sra     $3,$14,5
xor     $5,$5,$2
addu    $3,$10,$3
xor     $3,$5,$3
addiu   $7,$7,2560
addu    $3,$8,$3
sll     $6,$3,16
andi    $7,$7,0xffff
sra     $6,$6,16
bne     $7,$9,loop
```

a. Consider the two applications above. Calculate the average CPI for each application on each processor (two applications cross three processors means you should have 6 different CPI values). Assume size is always 20 for application 1.

b. Which processor has better performance? *[Careful answering this part…it is a tricksy professor question.]* Provide the quantitative evidence of "better performance" and include a description of how you evaluated the two applications together. What would the relative frequencies have to be between the three processors in order for their performance to be identical (i.e., calculate the "breakeven frequency")?

2. Amdahl's Law

Your company builds hardware for doing machine learning inference for image classification. 'Inference' is predicting the class of an image based on the model you learned using sophisticated machine learning algorithms. While everyone is busy trying to develop the next Neuromorphic or Quantum computing chip, your supervisor assigns you the tasks of optimizing the execution of the time of the only application that actually generates the $$$ for your company (i.e., image classification of cats). For image inference: **Convolution** is the most fundamental operations which consists of convolving or "sliding" a filter (sometimes called the learned kernel 2D array) of size k*k across the input image of size n*n which generates an output of size (n-k+1)*(n-k+1). If you want more information about this sort of convolution, you can view the following video: https://www.youtube.com/watch?v=XuD4C8vJzEQ&list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF&index=2.

The attached C and MIPS code implement a vertical edge detector that is an important portion of the cat image classification algorithm. As you can see, if there is a hardware floating point multiplier in the system the program will use * that will

compile to mul.s MIPS instructions, otherwise it will use library call that performs software floating point multiplication. Your specific task is to determine whether or not you should add a hardware multiplier to your MIPS processor.

a. Based on your understanding of the MIPS and the C code *estimate* the total number of clocks cycles the application will require to execute on your single-cycle processor. Also estimate the fraction of these cycles that the software multiplication function requires. *[Note: this is a somewhat open-ended problem and you can make any reasonable assumption. Please state all the calculations and assumptions you make. Note that you can actually run the MIPS code on MARS with and without the hardware multiplier to get instruction counts.]*

b. Based on the above question identify which part of the application will benefit from the hardware multiplier and why? What is maximum speed up possible? *[Again, you will have to make assumptions regarding the multiplier unit.]*

c. What would the maximum speedup be if your hardware multiplier slowed the clock frequency by 30%? *[You can still solve this by using Amdahl's law—you just have to appropriately adjust the equation from b.]*