NAME: ______________________
Student ID#: __________________

OPEN BOOK, OPEN NOTES. NO COMPUTERS, OR SOLVING PROBLEMS DIRECTLY USING CALCULATORS.

Your signature is your promise that you have not cheated and will not cheat on this exam, nor will you help others to cheat on this exam.

Signature: _____________________________________

Each Question; 10 points

Question 1 _____________
Question 2 _____________
Question 3 _____________
Question 4 _____________
Question 5 _____________
Question 6 _____________
Question 7 _____________
Question 8 _____________
Question 9 _____________
Question 10 _____________

TOTAL ______________
1. Assume that you have a pipelined implementation of MIPS with a perfect cache. Draw the pipelining diagram for the code along the left side. I have started the diagram for you with the first statement. **Indicate all data dependencies by circling the registers in the code.** Indicate all necessary forwarding from the output of one functional unit to the input of another functional unit. Indicate all necessary stalls with a nop. **Keep everything that happens in the same clock cycle in a straight column.** You do not have to shade the active units.

   add $6, $1, $2

   subi $6, $6, 1

   lw $2, 100($6)

   beq $2, $6, NEXT

   sub $6, $7, $2

   sw $6, $6($2)

2. Assume you have a loop with a branch statement at memory address 4000. Every time the program encounters this loop, the branch is taken 999 times and then not taken once which ends the loop. Assume this execution of the program has encountered this loop before and that no other branch statement indexes to the same entry in the branch prediction array.

   a) If a 1-bit prediction scheme is used, how many times will it be correct for the branch at memory address 4000 out of the 1000 times it is executed in the course of the loop?

   b) If a 2-bit prediction scheme is used, how many times will it be correct for the branch at memory address 4000 out of the 1000 times it is executed in the course of the loop?
3. The CPI of a machine with a perfect cache is 3.14 for a certain benchmark.
Now, assume an instruction cache miss rate of 2% and a data cache miss rate of 6%.
Assume the miss penalty (for either cache) is 270 cycles.
The frequency of data access instructions in this benchmark is 48%.
How much faster is the machine with the perfect cache than the one with the imperfect cache? Otherwise the machines are identical and running the same code. Express your answer either as a fraction or as a decimal rounded to the hundredths place. (X.XX)
Your answer should be > 1.

4. Assume an 80-bit virtual address and a 64-bit physical address. The page size is 8 KB.
How many entries are there in the page table? Express your answer in powers of 2.
Show your work. (KB = 2^10 bytes)

5. What is the order in which the system accesses the physically indexed/tagged L1 cache, the page table, the physically indexed/tagged L2 cache and the TLB assuming misses? (Assume all accesses are done sequentially, not in parallel.)

1. ________________________________

2. ________________________________

3. ________________________________

4. ________________________________
6. See the diagram below. The TLB is fully associative and contains the 4 bits: valid (V), protection (P), dirty (D), and reference (R). The page size is 8 KB. The virtual address is 37 bits long. The physical address is 32 bits long.
7. **Assume we have a perfect cache.** Assume the MIPS instruction set we have been studying all term. (Multicycle: loads = 5 cycles, stores and ALU instr = 4 cycles, branches and jumps = 3 cycles.)
Assume the following functional unit times: (ps = 10\(^{-12}\) sec)
800 ps for memory access
130 ps for ALU operation
100 ps for register file read or write

Assume a certain benchmark has the following instruction frequencies:
48\% loads
12\% stores
6\% branches
4\% jumps
30\% ALU instructions

For the pipelined design: Loads take 1 clock cycle when there is no load-use data dependency and 2 cycles when there is. Branches take 1 clock cycle when correctly predicted and 2 cycles when not. Jumps always cause a stall and therefore always take 2 cycles.

20\% of the load instructions have a load-use data dependency.
1/4 of the branch instructions are incorrectly predicted.

a) What is the minimum clock period for a single cycle implementation (in ps)? __________

b) What is the minimum clock period for a multi-cycle implementation (in ps)? __________

c) What is the minimum clock period for a pipelined implementation (in ps)? __________

d) What is the CPI for a single cycle implementation? __________

e) What is the CPI for a multi-cycle implementation? __________

f) What is the CPI for a pipelined implementation? __________

g) How long (in ps) does it take for an instruction to come out of the single cycle implementation? Round to the hundredths place. __________

h) How long (in ps) does it take (on average) for an instruction to come out of the multi-cycle implementation? Round to the hundredths place. __________

i) How long (in ps) does it take (on average) for an instruction to come out of the pipelined implementation? Round to the hundredths place. __________
8. Assume we have a 16 KB cache direct mapped cache. (KB = $2^{10}$ bytes)
The block size is 8 words.
Given a 32-bit physical address, divide up all the bits and indicate what they are used for to find and access the requested word in the cache. Do NOT draw the cache entries, the mux, the AND gate, ...
Just indicate exactly how many bits and which bits of the address are used for each purpose (the tag, the index, ...). **Label your groups of bits with their purpose.**

<table>
<thead>
<tr>
<th>31</th>
<th>0</th>
</tr>
</thead>
</table>

9. Assume we have a 16 KB cache 8-way set associative cache. (KB = $2^{10}$ bytes)
The block size is 2 word.
Given a 32-bit physical address, divide up all the bits and indicate what they are used for to find and access the requested word in the cache. Do NOT draw the cache entries, the mux, the AND gate, ...
Just indicate exactly how many bits and which bits of the address are used for each purpose. **Label your groups of bits with their purpose.**

<table>
<thead>
<tr>
<th>31</th>
<th>0</th>
</tr>
</thead>
</table>
10.

A) Write a loop in MIPS that adds up all the ints stored in an array of size 100. I have started it for you. Use my registers ($t0, $s1, $s2) as declared in the comments. Use only MIPS core instructions. Feel free to use registers $t1 - $t7.

```mips
# $t0 holds the address of the first int in the array
addi $s1, $t0, 400  #first word not in the array
addi $s2, $zero, 0  #sum initialized to zero

LP:
```

B) Unroll your loop (just the code that starts at LP) so that it executes 50 times and has no data hazards. Write your code to the right of your code from part A).

C) Given the registers you see used in the starter code of A), what register(s) must be saved to the stack frame in order to follow MIPS calling conventions?

__________________