Computer architecture 2, sample exam

(this is a closed book exam, however you can freely use a two-page MIPS reference sheet)

1. (Patterson-Hennessy; Exercises 2.10.4. - 2.10.6)

We consider the following two instructions of the MIPS processor:

```
add $t0, $t0, zero
lw $t1, 4($s3)
```

a) For the instructions above, show their hexadecimal representation.
b) What type (I-type, R-type) instruction do the instructions above represent?
c) What is the hexadecimal representation of the opcode, rs and rt fields in these instructions?
d) For R-type instructions, what is the hexadecimal representation of the rd and funct fields?
e) For I-type instructions, what is the hexadecimal representation of the immediate field?

2. (Patterson-Hennessy; Exercises 4.12.1. - 4.12.3.)

Assume that individual stages of the MIPS datapath have the following latencies:

IF	ID	EX	MEM	WB
300 ps	400ps	350ps	500ps	100ps

a) What is the clock cycle time in a pipelined and nonpipelined processor?
b) What is the total latency of a lw instruction in a pipelined and nonpipelined processor?
c) If we can split one stage of the pipelined datapath into two new stages, each with half the latency of the original stage, which stage would you split and what is the new clock cycle time of the processor?

3. (Patterson-Hennessy; Exercises 4.13.1. - 4.13.3)

Assume that the MIPS processor with five-stage pipeline executes the following sequence of instructions:

```
lw $1, 40($6)
add $6, $2, $2
sw $6, 50($1)
```

a) Indicate dependencies and their types.
b) Assume there is no forwarding in this pipelined processor. Indicate hazards and add nop instructions to eliminate them.
c) Assume there is full forwarding. Indicate hazards and add nop instructions to eliminate them.

4. (Patterson-Hennessy; Exercises 5.4.1. - 5.4.6)

For a direct-mapped cache design with 32-bit address, bits 31-12 of the address are used as the cache line tag and bits 11-5 as the cache index.

a) What are the cache size and the line size (in bytes)?

b) What is the ratio between total bits required for such a cache implementation over the data storage bits?

Starting from power on, the following byte-addressed cache references are recorded: `0x0, 0x4, 0x10, 0x84, 0xe8, 0xa0, 0x400, 0x1e, 0x8c, 0xc1c, 0xb4, 0x884`.

c) How many blocks are replaced and what is the hit ratio?

d) List the final state of the cache, with each valid entry represented as a tuple (index, tag).

5. (Patterson-Hennessy; Exercises 5.10.1. - 5.10.3)

A computer program produces the following stream of virtual addresses: `4095, 31272, 15789, 15000, 7193, 4096, 8912`. Assume 4KB pages, a four entry fully associative TLB and true LRU replacement.

Let a TLB entry be described with a triplet containing the validity flag, the virtual address tag and the physical page number. Then the initial TLB state is given by the following four triplets : (1, 11, 12), (1,7,4), (1,3,6), (0,4,9). Let a page table entry be described with a tuple containing the presence flag and the physical page. Then the page table is given by the following tuples: (1,5), (0,0), (0,0), (1,6), (1,9), (1,11), (0,0), (1,4), (0,0), (0,0), (1,3), (1,12).

a) Given the address stream and the initial states of TLB and page table, show the final state of the system. For each reference, state whether it is a TLB hit, a page table hit, or a page fault.

b) Repeat the exercise a) but this time use 16KB pages instead of 4KB pages. Discuss advantages and disadvantages of larger page sizes.

c) Show the final contents of the TLB if it is i) two-way set-associative and ii) direct-mapped. Discuss the importance of the TLB performance. How would virtual memory accesses be handled if there were no TLB?

6. (Patterson-Hennessy; Exercises 7.12.1. - 7.12.3)

Consider the following three CPU organizations:

1. CPU SS: a superscalar processor that offers out-of-order issue capabilities on two functional units (FUs). Only a single thread can run on each core at a time.
2. CPU MT: A fine grained multithreaded processor with two FUs that allows instructions from two threads to be run concurrently, though only instructions from a single thread can be issued on any cycle.
3. CPU SMT: An SMT processor with two FUs that allows issuing instructions from two threads concurrently.

Assume we have two threads (A and B) to run on these CPUs that include the following instructions. Instruction A1 takes 2 cycles to execute. Instruction A2 depends on the result of A1. Instruction A3 has a FU conflict with A1. Instruction A4 depends on the result of A2. Instruction B1 has no dependencies. Instruction B2 has an FU conflict with B1. Instruction B3 has no dependencies. Instruction B4 depends on the result of B2. Assume all instructions take a single cycle to execute unless noted otherwise or they encounter a hazard.

Draw the execution diagram for these two threads on the three different processors. How many total cycles does it take and how many issue slots are wasted due to hazards in each of the three cases?