PhD Qualifier Exam, Spring 2018

Computer Architecture and Systems

1. (7 points) A virtual memory system uses 15-bit virtual addresses. The physical memory consists of 8 Kbytes. The page size is 4 Kbytes. The Translation Lookaside Buffer (TLB) has 3 entries. Both the TLB and the page table are replaced using the LRU (least recently used) policy.

(a) (2 points) Indicate using a diagram how the Memory management Unit (MMU) would split a 15-bit virtual address to get the Virtual Page Number (VPN).

Number of bits for Page Offset = $\log_2$ 4K = 12 bits

(b) (3 points) Consider the following sequence of memory address accesses: 0x6ffc, 0x7ffc, 0x6000, 0x4000, 0x3000, 0x2000, 0x7ffc, 0x2008, 0x74fc, 0x64fc. For each of these accesses, indicate if it would be a TLB hit or miss. Also indicate if it would be a page fault or not.

Virtual address	VPN	TLB Hit/Miss	Page Fault?	TLB Entry	PFN
0x6ffc	0x6	Miss	Yes	0	0
0x7ffc	0x7	Miss	Yes	1	1
0x6000	0x6	Hit	No	0	0
0x4000	0x4	Miss	Yes	2	1
0x3000	0x3	Miss	Yes	1	0
0x2000	0x2	Miss	Yes	0	1
0x7ffc	0x7	Miss	Yes	2	0
0x2008	0x2	Hit	No	0	1
0x74fc	0x7	Hit	No	2	0
0x64fc	0x6	Miss	Yes	1	1

(c) (2 points) Draw the final state of the TLB and the page table.

One possible final state of the TLB and PT are given below:

TLB

V	VPN	PFN
0	0x2	0x1
1	0x6	0x1
1	0x7	0x0

PT

V	PFN
0	
0	
0	0x1
0	0x0
0	0x1
0	
1	0x1
1	0x0

2. (7 points)
(a) (3 points) With the help of an example code snippet, show how control hazards can affect the performance of a pipelined datapath.

Consider the following MIPS code snippet, in which the branch instruction ends up being taken. The right side of the figure shows the pipeline timing diagram, assuming the standard MIPS 5-stage pipeline. The two instructions that sequentially follow the branch instruction are automatically fetched into the pipeline before the outcome of the branch is known. If these two instructions are allowed to proceed, a control hazard is manifested. Correct functioning of the pipeline involves detecting this control hazard, and taking appropriate corrective staeps, which in this example involves squashing those two instructions. Because of the squashing, performance is affected.

```
         beq $1, $2, label1      F  D  E  M  W
         add $4, $5, $6             F  D  -  -
         sub $5, $7, $8                F  -  -
label1:  or  $4, $5, $2                   F  D
```

(b) (4 points) Discuss two different techniques for mitigating the impact of control hazards in a pipelined datapath. Mention one disadvantage of each technique.

1. Delayed brach: The semantics of a branch instruction is modified to indicate that the branch takes effect after executing the subsequent n instructions, where n is the number of delay slot instructions. One disadvantage of this technique is that the compiler may not always find useful instructions to place in the delay slots, and may end up filling them with nops, which affect the execution time.

2. Branch prediction: The hardware maintains the recent history of each (static) branch instruction and makes a prediction about the outcome of each dynamic branch instruction based on previous history. One disadvantage of this scheme is the increased power consumption.

3. (6 points) Short answer questions.
(a) (3 points) Consider the following C program snippet:

```
foo(int *p)
{
    int *ptr;
    static int s = 100;

    ptr = p + 100;
    s += *ptr;
}
```

When a compiler translates this program to an assembly language program, in which section of the memory—`.text`, `.data`, `heap`, or `stack`—will it allocate each of the two variable, `ptr` and `s`? Give reasons for your answer.

`ptr` *will be allocated in the stack frame, as it is a local variable of function* `foo()`*, as each instance of* `foo()` *requires a new instance of* `ptr`*.*

`s` *will be allocated in the* `.data` *section, as all instances of* `foo()` *must access the same* `s` *and the value of* `s` *needs to be maintained even after control returns from each instance of* `foo()`*.*

(b) (3 points) Consider a computer system that implements multitasking. When the system temporarily stops a process, its state—register values and memory values—need to be preserved so that it can be run again as if there was no interruption. The register values are preserved by saving them in the process' PCB (Process Control Block) or kernel stack. Explain how the memory values are preserved in the multitasking system.

Unlike the physical register file, the physical memory is (space) partitioned among the active processes. A virtual memory system is implemented to map logical (virtual) addresses to physical memory addresses. When a process is context switched, the portion of physical memory allocated to it will continue to maintain its values; this makes it possible to do fast context switching.