# 0907432 Computer Design (Summer 2014)

## Midterm Exam

الاسم: الرقم التسلسلي: رقم الشعبة: 1

==========

**Instructions**: Time **50** minutes. Open book and notes exam. No electronics. Please answer all problems in the space provided and limit your answer to the space provided. **No questions are allowed**.

*<Good Luck>*

**Q1.** Assume that you have the MIPS with static dual issue processor studied in the class. Remember that this processor can issue packets each packet comprising of one ALU/branch instruction and one memory instruction. Using the table below, prepare your best code schedule for the following instruction sequence. Also specify how many cycles are needed to fetch and complete executing these instructions.

*[10 marks]*

```
add  r5,r2,r1
lw   r3,4(r5)
lw   r2,0(r2)
or   r3,r5,r3
sw   r3,0(r5)
lw   r11,0(r10)
addi r10,r10,4
sw   r11,0(r10)
```

| Packet | ALU/branch | Load/store |
|---|---|---|
| 1 | `add r5,r2,r1` | `lw r2,0(r2)` |
| 2 | `nop` | `lw r3,4(r5)` |
| 3 | `addi r10,r10,4` | `lw r11,0(r10)` |
| 4 | `or r3,r5,r3` | `sw r11,0(r10)` |
| 5 | `nop` | `sw r3,0(r5)` |
| 6 | | |
| 7 | | |

Number of cycles = **5 + 4 = 9 cycles**

**Q2.** Assume that the following code sequence is executed by a triple-issue speculative pipelined processor. This processor uses reservation stations, common data buses, and reorder buffer. The fetch stage takes one cycle and the issue stage takes one cycle. The integer latency is 1 cycle and the load latency is 2 cycles (1 cycle for address calculation and 1 cycle for data memory access). The processor has two address calculation units, two memory access units, two integer ALU units, and one branch unit. Assume that the `addi` instruction generates an overflow exception. Using the multi-cycle pipeline diagram below, specify the execution of these instructions in this processor pipeline.

*[10 marks]*

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| `add r5,r2,r1` | **F** | **I** | **E** | **W** | **C** | | | | | | | | | | |
| `lw r3,4(r5)` | **F** | **I** | | | **A** | **M** | **W** | **C** | | | | | | | |
| `lw r2,0(r2)` | **F** | **I** | **A** | **M** | **W** | | | **C** | | | | | | | |
| `or r3,r5,r3` | | **F** | **I** | | | | | **E** | **W** | **C** | | | | | |
| `sw r3,0(r5)` | | **F** | **I** | | **A** | | | | | **C** | | | | | |
| `lw r11,0(r10)` | | **F** | **I** | **A** | **M** | **W** | | | | **C** | | | | | |
| `addi r10,r10,4` | | | **F** | **I** | **E** | **W** | | | | | **C** | | | | |
| `sw r11,0(r10)` | | | **F** | **I** | | | **A** | | | | **N** | | | | |
| **Exception Routine** | | | | | | | | | | | | **F** | **...** | | |

**Q3.** Assume that you have a MIPS processor with direct-mapped primary cache that has 128 blocks and its block size is 32 bytes.

**A)** How many bits is the cache index?

*[2 marks]*

**<index> = $\lg_2$ 128 = 7 bits**

**B)** How many bits is the cache tag?

*[2 marks]*

**<block offset> = $\lg_2$ 32 = 5 bits**
**<tag> = 32 – 7 – 5 = 20 bits**

**C)** What is the total number of bits in this cache assuming that it is a write-back cache?

*[2 marks]*

**Total size = number of blocks × (block size + tag size + valid bit + dirty bit)**
**= 128 × (32×8 + 20 + 1 + 1)**
**= 35,584 bits**

**D)** Complete the following table that shows the byte addresses of six accesses to this cache in hexadecimal. Assume that the cache was empty before these accesses. For every byte address, specify whether the access is a hit or miss and specify the miss type (compulsory, capacity, conflict) for the miss accesses.

*[4 marks]*

| Byte Address | Hit or miss | Miss Type |
|---|---|---|
| 00000000 | **Miss** | **Compulsory** |
| 00000010 | **Hit** | |
| 00001004 | **Miss** | **Compulsory** |
| 00001008 | **Hit** | |
| 00000000 | **Miss** | **Conflict** |
| 0000001C | **Hit** | |