University of Maryland Baltimore County
Department of Computer Science and Electrical Engineering

CMSC 611-101, Advanced Computer Architecture

Assignment #2 Due: Tuesday 3/4/13 in class

Question 1: *(34 Points)*

For following code, assume that the loop index R10 is a multiple of 8 (to ensure word alignment):

```
Loop: LW   R2,  0(R10)
      SUB  R4,  R2,  R3
      SW   R4,  0(R10)
      LW   R5,  4(R10)
      SUB  R6,  R5,  R3
      SW   R6,  4(R10)
      ADDI R10, R10, 8
      BNE  R10, R30, Loop
```

A) Schedule this code for fast execution on the standard MIPS pipeline (assume that it supports ADDI instruction).
B) Assume initially R10 is 0 and R30 is 40. How does the scheduled code compare against the original unscheduled code?

Question 2: *(24 Points)*

Consider the execution of the following code on a 5-stages pipeline processor with forwarding logic:

```
Loop:     LW   R2,  100(R6)
          LW   R3,  200(R7)
          ADD  R4,  R2,  R3
          ADD  R6,  R3,  R5
          SUB  R8,  R4,  R6
          LW   R7,  300(R8)
          BEQ  R7,  R8,  Loop
```

You asked to change this program in order to *maximize* performance.
Hint: you need to trace the calculation.

Question 3: *(42 Points)*

For these problems, we will explore a pipeline for a register-memory architecture. The architecture has two instruction formats: a register-register format and a register-memory format. There is a single-memory addressing mode (offset + base register). There is a set of ALU operations with format

ALU_{op} R_{dest}, R_{src1}, R_{src2}

or

ALU_{op} R_{dest}, R_{src1}, MEM

where the ALU_{op} is one of the following: Add, Subtract, And, Or, Load (R_{src1} ignored), Store. R_{src} or R_{dest} are registers. MEM is a base register and offset pair.

Branches use a full compare of two registers and are PC-relative. Assume that this machine is pipelined so that a new instruction is started every clock cycle. The following pipeline structure—similar to that used in the VAX 8700 micro-pipeline [Clark 1987]—is

IF	RF	ALU$_1$	MEM	ALU$_2$	WB					
	IF	RF	ALU$_1$	MEM	ALU$_2$	WB				
		IF	RF	ALU$_1$	MEM	ALU$_2$	WB			
			IF	RF	ALU$_1$	MEM	ALU$_2$	WB		
				IF	RF	ALU$_1$	MEM	ALU$_2$	WB	
					IF	RF	ALU$_1$	MEM	ALU$_2$	WB

The first ALU stage is used for effective address calculation for memory references and branches. The second ALU cycle is used for operations and branch comparison. RF is a cycle for both decoding and register-fetching. Assume that when a register read and a register write of the same register occur in the same clock the write data is forwarded.

A) Find the number of adders needed, counting any adder or incrementer; show a combination of instructions and pipeline stages that justify this answer. You need only give one combination that maximizes the adder count.

B) Find the number of register read and write ports and memory read and write ports required. Show that your answer is correct by showing a combination of instructions and pipeline stage indicating the instruction and the number of read ports and write ports required for that instruction.

C) Determine any data forwarding for any ALUs that will be needed. Assume that there are separate ALUs for the ALU$_1$ and ALU$_2$ pipe stages. Put in all forwarding among ALUs needed to avoid or reduce stalls. Be careful to consider forwarding across an intervening instruction, for example,

ADD $R_{1,}$ ...
any instruction
ADD ..., R_1, ...