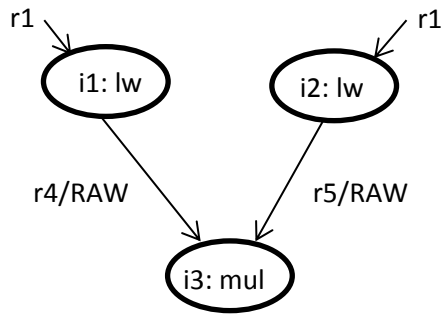
CPSC 3300 – Spring 2015
Homework 6
Due at class time on Monday, April 6

1. For the MIPS instruction sequence below, complete the data dependency diagram. (Destination register is listed first except for sw instruction; sw writes into memory rather than a register.)

```
i1: lw   r4, 0( r1 )  // reg[4] <- memory[ reg[1] + 0 ]
i2: lw   r5, 4( r1 )  // reg[5] <- memory[ reg[1] + 4 ]
i3: mul  r6, r4, r5   // reg[6] <- reg[4] * reg[5]
i4: sub  r8, r6, r7   // reg[8] <- reg[6] - reg[7]
i5: sw   r8, 8( r1 )  // memory[ reg[1] + 8 ] <- reg[8]
i6: add  r1, r1, r2   // reg[1] <- reg[1] + reg[2]
```



2. Draw the dependency diagram for the following MIPS code

```
i1: add  r3, r4, r6    // reg[3] <- reg[4] + reg[6]
i2: sub  r5, r3, r2    // reg[5] <- reg[3] - reg[2]
i3: lw   r7, 0( r5 )   // reg[7] <- memory[ reg[5] + 0 ]
i4: add  r5, r5, r8    // reg[5] <- reg[5] + reg[8]
i5: sw   r7, 0( r5 )   // memory[ reg[5] + 0 ] <- reg[7]
```

3. For the following MIPS instruction sequence, complete the pipeline cycle diagram for the standard 5-stage pipeline without forwarding. Assume register file writes occur in the first half cycle and reads in the second half cycle.

```
i1: lw   r1, 0( r5 )   // reg[1] <- memory[ reg[5] + 0 ]
i2: lw   r2, 4( r1 )   // reg[2] <- memory[ reg[1] + 4 ]
i3: addi r2, r2, 1     // reg[2] <- reg[2] + 1
```

```
i1:lw    IF    ID    EX    MEM    WB
i2:lw          IF
i3:addi
```

4. For the following MIPS instruction sequence, complete the pipeline cycle diagram for the standard 5-stage pipeline with forwarding. Assume register file writes occur in the first half cycle and reads in the second half cycle.

```
i1: lw   r1, 0( r5 )  // reg[1] <- memory[ reg[5] + 0 ]
i2: lw   r2, 4( r1 )  // reg[2] <- memory[ reg[1] + 4 ]
i3: addi  r2, r2, 1   // reg[2] <- reg[2] + 1
```

```
i1:lw    IF      ID      EX      MEM    WB
i2:lw            IF
i3:addi
```

5. Consider a one-bit history for branch prediction. It records the state of the last branch as taken (T) or untaken (U) and predicts the next branch will be the same. Assume the bit is initialized to U. Determine the prediction accuracy on the following branch trace; include all trace entries in your calculation.

(a) T T T T U T T T T U T T T T U T T T T U

(b) T U T U T U T U T U T U T U T U T U T U

6. An HP processor used a three-bit branch history shift register in each branch history table entry and a "majority vote" of the BHSR bits to predict whether the next branch is taken (T) or untaken (U). (E.g., TUT => predict taken.) Assume the BHSR is initialized to UUU. Determine the prediction accuracy on the following branch trace; include all trace entries in your calculation.

(a) T T T T U T T T T U T T T T U T T T T U

(b) T U T U T U T U T U T U T U T U T U T U