The University of Alabama in Huntsville
ECE Department
Homework Assignments
CPE 431/531 01/91/92
Fall 2015

Homework #1 – Due September 3, 2015

1.5(10), 1.7(15), 1.9.1(5), 1.9.3(10), 1.11.1(5), 1.11.9(10), 1.11.11(10), 1.12.3(10), 1.12.4(10), 1.13.2(5), 1.14.1(10)

Homework #2 – Due September 11, 2015

2.2(5), 2.4(15), 2.5(15), 2.7(5), 2.13.1(5), 2.13.2(5), 2.13.3(5), 2.15(5), 2.24(5), 2.26(15), 2.40(5), 3.3(5), 3.6(5), 3.8(5)

Homework #3 – Due September 22, 2015

3.21(10), 3.22(10) use 0xD780 0000, 3.23(10) using 279.4375, 3.24(10) using 279.4375, 3.28(20), 4.3.1(10), 4.3.2(10), 4.7.4(10), 4.7.5(10)

Homework #4 – Due October 2, 2015

4.8.1(5), 4.8.3(10), 4.9.2(10), 4.9.4(10), 4.9.5(10), 4.9.6(10), 4.13.1(5), 4.13.4(20), 4.15.1(10), 4.15.2(10)

Homework #5 – Due October 23, 2015

Use the code below for 4.18, 4.18.2(10), 4.18.3(10), 4.18.4(10), 4.18.5(10), 4.19 (given below) (30), 5.1.1(5), 5.1.2(5), 5.1.3(5), 5.1.4(10), 5.1.6(5)

```
        add   R5, $zero, $zero
Again:  beq   R5, R6, End
        sll   $t0, R5, 2
        add   R10, R1, $t0
        lw    R11, 0(R10)
        lw    R10, 4(R10)
        sub   R10, R11, R10
        add   R11, R2, $t0
        sw    R10, 0(R11)
        addi  R5, R5, 2
        beq   R0, R0, Again
End:
```

4.19 In this exercise, we consider the execution of a loop in a statically scheduled superscalar processor. To simplify the exercise, assume that any combination of instruction types can execute in the same cycle, e.g., in a 3-issue superscalar, the three instructions can be three ALU operations, three branches, three load/store instruction, or any combination of these instructions. Note that this only removes a resource constraint, but data and control dependences must be still be handled correctly. Problems in this exercise refer to the following loop:

```
Loop: lw    $1, 40($6)
add   $5, $5, $1
sw    $1, 20($5)
```

```
addi   $6, $6, 4
addi   $5, $5, -4
beq    $5, $0, Loop
```

Unroll this loop so that four iterations of it are done at once and schedule it for a 2-issue static superscalar processor. Assume that the loop always executes a number of iterations that is a multiple of 4. You can use registers **$10** through **$20** when changing the code to eliminate dependences.
Reschedule it for a 2-issue processor like the one in the book which can issue one memory and one non-memory instruction in each cycle.

Homework #6 – Due November 3, 2015

5.2.2(10), 5.2.3(20), 5.2.6(15), 5.6.3(5), 5.6.5(5), 5.6.6(10), 5.7.1(10), 5.7.3(15), 5.7.5(10)