Data Provided: None

The University Of Sheffield.

DEPARTMENT OF ELECTRONIC AND ELECTRICAL ENGINEERING

Spring Semester 2015-16 (2.0 hours)

EEE336 Digital Design

Answer **THREE** questions. **No marks will be awarded for solutions to a fourth question.** Solutions will be considered in the order that they are presented in the answer book. Trial answers will be ignored if they are clearly crossed out. **The numbers given after each section of a question indicate the relative weighting of that section.**

1. a. Briefly explain the following terms when applied to the addressing modes of a microprocessor instruction.

i) Implied Mode **(2)**

ii) Immediate Mode **(2)**

iii) Register Indirect **(2)**

iv) Based Memory Addressing **(2)**

b. Calculate $1101_2 \div 100_2$ (decimal 13 divided by 4) by non-restoring division. The data values must all be held in byte wide storage locations. Show each step of the calculation in binary. You must start the process by left shifting the divisor by three places (multiplication by 2^3) in order to demonstrate a process whereby any four bit positive integer could be divided by any three bit positive integer. **(7)**

c. Explain with the aid of a simple diagram what is meant by the term '*synchronous pipeline'* in relation to a digital system. **(2)**

A multi-stage synchronous pipeline is to be modelled in Verilog. Would you use blocking or non-blocking assignments? Explain your reasoning. **(3)**

2. **a.** Sketch a possible hardware implementation of a serial shift-and-add multiplier (using a parallel adder) for multiplying two n-bit numbers. Carefully describe the initial circuit conditions. **(6)**

For the more general case of multiplying an m-bit multiplicand by an n-bit multiplier, how long will it take to generate the product? **(2)**

Outline the operation of a serial shift-and-add multiplier with reference to the multiplication of two unsigned integers, 110_2 by 101_2 using a table to show the states of the circuit at each stage. **(4)**

b. Using the example of 110_2 multiplied by 011_2 (where 110_2 is the multiplicand and 011_2 is the multiplier) describe how basic shift and add multiplication can be extended to handle multiplication of a signed multiplicand by an unsigned multiplier. **(4)**

c. Rather than forming partial products by successively left-shifting the multiplicand, an alternative implementation for a shift and add multiplier is to *right-shift* the accumulated partial product after each stage. Again with reference to 110_2 by 011_2, the steps in right-shift multiplication are:

110	
101 ×	
000	Initial value of partial product total
110	First partial product
110	Running total of partial product
0110	Right shift
000	Second partial product
0110	Running total of partial product
00110	Right shift
110	Third partial product
11110	Running total of partial product
011110	Right shift = Final answer

Notice that at each stage, after forming the running total of partial products, the result is right-shifted by one place to correctly align the running total to receive the next partial product.

What are the main advantages of this right-shift multiplication over the left-shift version in part (a) of this question? **(4)**

3. a. Considering the architecture of a Reduced Instruction Set Computer (RISC)

i) Why do RISC machines contain a large number of registers? **(2)**

ii) What is the principal advantage gained by reducing the number of instructions available in a RISC machine? **(2)**

iii) Why is the instruction set in a RISC machine limited to simple instructions? **(2)**

b. Explain the difference between the following terms in the Verilog Hardware Description Language (HDL):

i) The types **wire** and **reg** **(2)**

ii) An **initial** procedure and an **always** procedure **(2)**

iii) **inertial** and **transport** delay **(3)**

c. Draw a truth table for the circuit described by the following Verilog code.

```
module digcomponent (output reg [3:0] y, input [1:0] a);
integer i;
always @ (*)
  for (i = 0; i <= 3; i = i + 1)
    if  (a == i)
      y[i] = 1;
    else
      y[i] = 0;
endmodule
```

(3)

Describe the function that is implemented by this code. Hence, rewrite the code to expand the same functionality for a component with a three bit input and an eight bit output. **(4)**

4. **a.** Briefly describe the two principal elements that make up the central processing unit (CPU) of a microprocessor. How are they inter-connected? What is the role of any feedback connection between these two elements? **(4)**

b. In a central processing unit (CPU), the sequence of control signals can be generated either using hardwired logic or a look-up table (microcoding). What is the advantage of the microcoding approach? **(2)**

c. **i)** Sketch a simple microcoding arrangement. **(4)**

ii) Briefly describing the operation of each functional block. **(6)**

d. What is the disadvantage of generating micro-instructions using the simple microcoding arrangement in **(c)** above? Describe how *vertical microcoding* can be used to address this problem. What is the disadvantage of using *vertical microcoding*? **(4)**

NJP/ JROD