Data Provided: None

The University Of Sheffield.

DEPARTMENT OF ELECTRONIC AND ELECTRICAL ENGINEERING

Spring Semester 2013-14 (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. i) Describe the format of a floating point number. **(4)**

ii) Outline the process whereby two binary floating point numbers are added together. **(4)**

b. The Booth algorithm can be used to multiply two integers by encoding the multiplier as follows:

00 : middle of a run of 0s, do nothing

10 : beginning of a run of 1s, subtract multiplicand

11 : middle of a run of 1s, do nothing

01 : end of a run of 1s, add multiplicand

Show how the Booth algorithm can be used to multiply the positive binary integers 011101 (multiplicand) and 01110 (multiplier). **(8)**

c. Compare the efficiency of Booth encoded multiplication with a traditional shift and add technique. How could the efficiency of Booth encoding be improved upon? **(4)**

2. **a.** Describe the basic interrupt mechanism and explain why this is a better way of handling external events than continually polling some external status flag. **(6)**

b. What do you understand by *interrupt latency*? Give an example of an application where interrupt latency may be a critical factor. **(3)**

c. Why do some people regard interrupts as inherently unsafe? **(3)**

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

i) Why do RISC machines contain a large number of registers? **(3)**

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

iii) What is the fundamental limit on the internal clock speed of a central processing unit (CPU)? **(1)**

iv) Explain how a RISC architecture may improve this limit. **(2)**

3. **a.** The *Harvard architecture* is frequently used in digital signal processing. Explain why this has advantages over the conventional *von Neumann* architecture for digital signal processing. In particular, what is the *von Neumann bottleneck*? **(6)**

b. Calculate $1110_2 \div 101_2$ (decimal 14 divided by 5) 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. **(6)**

c. A digital system is required which can produce a sine wave with a frequency of 50KHz. Draw a block diagram to show how this can be achieved using a Read Only Memory (ROM) based look-up table. The ROM is 1K x 8 and contains 1024 data samples for one complete cycle of a sine wave. Explain how your circuit gives the desired frequency.

Could the size of the ROM be reduced? If so, explain how this could be achieved. **(8)**

4. **a.** 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) A blocking assignment (=) and a nonblocking assignment (<=). **(2)**

b. Draw the circuit that will result from synthesis of the following Verilog code:

```
module pipeline (output reg [7:0] C_out, input [7:0] A_in, input clk);
  reg [7:0] A_out, B_out;
  always@( posedge clk )
  begin
    A_out = A_in;
    B_out = A_out;
    C_out = B_out;
  end
endmodule
```

How would you alter this code in order to produce three pipelined registers after synthesis? What is the advantage of using a pipeline in digital design? **(4)**

c. Develop an Algorithmic State Machine and Datapath (ASMD) chart for a shift-and-add multiplier. It should examine the least-significant bit of the multiplier (MR), add the multiplicand (MD) to the partial product (PP) if the bit is a 1, and then shift the partial product and the multiplier one bit position to the right. Draw the ASMD chart and explain what each of your states represents and the operations required in the datapath. **(10)**

NJP/ JR