One and a half hours

Section A is COMPULSORY

An additional answersheet is provided for Question 4.
Please remember to complete the additional answersheet with your University ID number and attach it to your answerbook(s).

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Fundamentals of Computer Engineering

Date: Wednesday 22nd January 2014
Time: 14:00 - 15:30

Answer ALL Questions in Section A and ONE Question from Section B

This is a CLOSED book examination

The use of electronic calculators is permitted provided they are not programmable and do not store text

[PTO]
SECTION A

COMPULSORY

Please answer both questions

1. Answer any 5 parts of this question. Each part is worth 2 marks.

a) For the logical function

\[ Q = \overline{A} \overline{B} + A.B \]

show, using DeMorgan’s theorem, how it can be implemented using 3 NAND gates; sketch a schematic of the design (the true and complemented input signals are provided for you).

b) For the Verilog stimulus code shown below sketch waveforms to show how the signal values vary in time relative to each other.

```verilog
initial
begin
  clock = 0;
  #20
  clock = 1;
  #20
  clock = 0;
  #20
  clock = 1;
  #20
  clock = 0;
  #20
  clock = 1;
  #20
  $stop;
end

initial
begin
  data_out = 0;
  #30
  data_out = 1;
  #30
  data_out = 0;
  #30
  data_out = 1;
end
```

(Question 1 continues on the following page)
(Question 1 continues from the previous page)

c) What is the basic component of a register? What is the purpose of the CE input to a flip-flop when present?

d) Figure 1 illustrates a state transition diagram of a finite state machine. However, there are some errors with the design. Identify all the errors.

![State Transition Diagram](image)

Figure 1

e) CAD tools are essential for the digital designer to manage complex designs. Give two examples of CAD tools used in digital hardware design and briefly state what each tool is used for.

f) What are the equivalent 6-bit 2’s complement binary numbers for the following signed decimal values?

   i.  +23  
   ii. -1   
   iii. -32  
   iv. +36

(Question 1 continues on the following page)
(Question 1 continues from the previous page)

h) Sketch an equivalent schematic representation of the logical function described by Verilog module given below

```verilog
module acircuit (input a, b, c, output p, q);
    wire avariable;
    assign avariable = a | b;
    assign p = avariable & ~c;
    assign q = ~p;
endmodule
```

(Answer continues from the previous page)
2. Answer any 5 parts of this question. Each part is worth 2 marks.

a) Briefly discuss the advantages and disadvantages of the Harvard architecture as compared to the von Neumann architecture.

b) Sort the following types of memories in terms of (i) read speed and (ii) storage density and annotate whether each memory type is volatile or not:

- SRAM,
- magnetic discs,
- DRAM,
- Flash memory.

c) Please answer True or False to the following statements:

i. Tristate signals should normally be used as control signals.
ii. An asynchronous system requires a clock signal.
iii. A clock signal of 250MHz has a period of 4\(\mu\)s.
iv. Communication by handshaking requires at least two control signals.

(\(\frac{1}{2}\) mark will be awarded for a correct answer, and \(\frac{1}{2}\) mark will be lost for an incorrect answer)

d) Assuming that a user program stored in the main memory is divided into 10 parts. The dynamic utilization (normalized to 1) of each part of this program for some data set is indicated by the numbers in the main memory space illustrated in Table 1. A cache that can accommodate 20% of the program is also included in our architecture (not shown here). Which parts of the program would you choose to place in the cache and why? Which properties of the program code have you used for your assignment? (The hex memory addresses provide the beginning address for each segment of the program).

<table>
<thead>
<tr>
<th>Main Memory</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>hx1xxx</td>
<td>0.04</td>
</tr>
<tr>
<td>hx2xxx</td>
<td>0.15</td>
</tr>
<tr>
<td>hx3xxx</td>
<td>0.08</td>
</tr>
<tr>
<td>hx4xxx</td>
<td>0.13</td>
</tr>
<tr>
<td>hx5xxx</td>
<td>0.02</td>
</tr>
<tr>
<td>hx6xxx</td>
<td>0.18</td>
</tr>
<tr>
<td>hx7xxx</td>
<td>0.19</td>
</tr>
<tr>
<td>hx8xxx</td>
<td>0.08</td>
</tr>
<tr>
<td>hx9xxx</td>
<td>0.05</td>
</tr>
<tr>
<td>hxAxxx</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Table 1

(Question 2 continues on the following page)
(Question 2 continues from the previous page)

e) Briefly discuss what TDM is and mention an example application where it is typically used.

f) A von Neumann machine with a word width of 16-bits has a main memory which, in turn, has 16-bit address bus. The smallest addressable quantity is one word. From the 16-bits of the memory address, three are used to select the proper memory chip and the rest are employed to determine a word within each chip. How many chips constitute the memory? What is the size in bits of each memory chip?

g) Assume a daisy chain configuration for prioritizing interrupts among peripherals is illustrated in figure 2. The CPU generates IAck 500 ps after a (active) clock edge. The wire delay is 1 ns (shown in the figure) and the logic delay through each peripheral is 500 ps. How many peripherals can be accommodated if the lowest priority peripheral must receive the IAck signal within the 7 ns clock period?

![Figure 2](image)

h) Describe the operation of the PC register in the MU0 processor. How general is the use of this register to other microprocessors?
3. The Verilog code in figure 3 describes a partially completed module.

```verilog
module mystery (input clock, start, reset,
                 output reg [4:0] sequence,
                 output reg flash);

reg [3:0] next_state;
reg [3:0] current_state;

always @ (*) // AA
    case (current_state)
        3'b000: if(start) next_state = 3'b001;
                else next_state = 3'b000;
        3'b001: next_state = 3'b010;
        3'b010: next_state = 3'b011;
        3'b011: next_state = 3'b100;
        3'b100: next_state = 3'b101;
        3'b101: next_state = 3'b110;
        3'b110: next_state = 3'b000;
        default: next_state = 3'b000; // BB
    endcase

always @ (posedge clock) // CC
    if(reset == 1)
        current_state <= 3'b000;
    else
        current_state <= next_state;

// assignment of output signals here
// ********
// ********
// ********
endmodule
```

**Figure 3**

a) Describe the syntactic structure of a Verilog module. How would you represent a module in a circuit diagram?  
(4 marks)

b) What is the purpose of the statement at line AA?  
(1 mark)

c) Why should we include a default case as shown in line BB?  
(1 mark)
d) When will the `always` block starting at line CC be executed? (1 mark)

e) Sketch and label a state transition diagram for the machine described in the Verilog module. (5 marks)

f) The module shown in figure 3 is used to describe the operation of a pedestrian crossing, where there is a set of traffic lights with the usual arrangement of red, amber and green lights, and pedestrian lights consisting of a red man and a green man. The required light sequence with respect to each state of the Finite State Machine (FSM) is illustrated in Figure 4.

![State Transition Diagram](image)

Figure 4

(back to state 000)
The output sequence, which represents the state of the lights, is configured as shown in table 2, where the appropriate bit is set to ‘1’ to turn on the light, or ‘0’ to turn off the light.

<table>
<thead>
<tr>
<th>bit number of output bus sequence</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>red traffic light</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>amber traffic light</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>green traffic light</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pedestrian red man</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pedestrian green man</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2

When the output flash is taken high any lights that are lit will flash at a rate determined by an external circuit.

Complete the design of the module by producing the Verilog code to configure the output signals sequence and flash.

(8 marks)
4. a) Assuming that communication between a peripheral and the master CPU is realized with the strobing method, explain why the address and data signals should not change at least as long as the strobe signal is asserted. Provide a timing diagram that illustrates the timing of this method.  

(3 marks)

b) A combinational circuit is flanked by two registers, as illustrated in figure 5. By using the delays of the gates as given in figure 6 annotate on the additional answer sheet the critical path of the circuit. What is the delay of this path? If the registers have a set up time of 1ns, what is the maximum clock frequency that this circuit can operate? (Do not attempt to optimize the circuit first.)

(5 marks)

Please write your University ID clearly on the additional answer sheet and attach it securely to your answer booklet.

(Question 4 continues on the following page)
c) Within an instruction set, an instruction compares two operands and sets a flag bit in one of the registers of the CPU if the numbers are equal. Describe two possible ways to implement this instruction. Provide the logic circuit (i.e., schematic) for both ways. The schematic of the circuit for a single bit of the two operands suffices (i.e., a bit slice).

   (4 marks)

d) If USB is used to transmit the following data stream:

   \[ 0100111111000000010 \]

   (The underlined bit is the 1st transmitted bit of the stream.)

   i) Provide the proper encoding without bit stuffing by drawing the appropriate NRZI waveform for the entire data stream.

   (1 mark)

   ii) How would the data stream change when bit stuffing is considered? Also depict the NRZI waveform with bit stuffing. What is the reason for using bit stuffing? (Hint: USB adds a transition to ensure that there is at least one transition every six bit times.)

   (2 marks)

e) The block shown in Figure 7 corresponds to a priority encoder for hardware interrupts with three outputs O1, O2, and INT. The outputs O1 and O2 encode the interrupt source and INT is asserted only if an interrupt is to be served. The four inputs (active high) to the encoder correspond to 4 different interrupt sources. The interrupt priority among these sources is defined as follows: an interrupt will be served only if any interrupt of higher priority has not been asserted. The inputs in descending order of priority are: D3, D2, D1, D0.

   i) Provide the truth table for the priority encoder.

   (3 marks)

   ii) Based on the truth table, provide the Boolean expression and the related schematic only for the output INT.

   (2 marks)
(Question 4 continues from the previous page)

![Diagram of a Priority encoder](image)

**Figure 7**

END OF EXAMINATION