2012 Paper E2.1/ISE2.2: Digital Electronics II

Answer ALL questions.

There are THREE questions on the paper.

Question ONE counts for 40% of the marks, other questions 30%

(Not to be removed from the Examination Room)

Information for Candidates:

The following notation is used in this paper:

1. Unless explicitly indicated otherwise, digital circuits are drawn with their inputs on the left and their outputs on the right.
2. Within a circuit, signals with the same name are connected together even if no connection is shown explicitly.
3. The notation X2:0 denotes the three-bit number X2, X1 and X0. The least significant bit of a binary number is always designated bit 0.
4. Signed binary numbers use 2's complement notation.

1. (a) *Figure 1.1* shows a circuit to evaluate the parity P of a serial data stream D_{in}. Write a Verilog description of this circuit.

[4]

The basic logic element (LE) of a Cyclone II FPGA consists of a 4-input lookup table (LUT) and a register. The maximum delay of the LUT is 1.5 ns; the setup time and the clock-to-output delay of the register are both 0.75 ns. Estimate the maximum clock frequency below which the parity circuit would operate correctly.

[4]

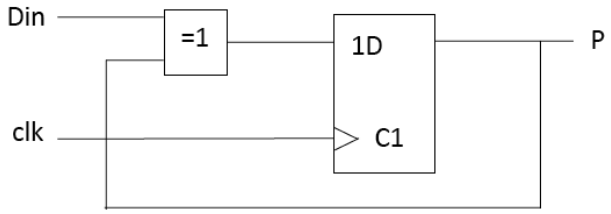


Figure 1.1

(b) *Figure 1.2* shows a circuit that performs an arithmetic operation. Given that $X7:0 = AF_{16}$, evaluate Y7:0. Hence or otherwise, derive an expression relating X and Y. State any assumptions used.

[8]

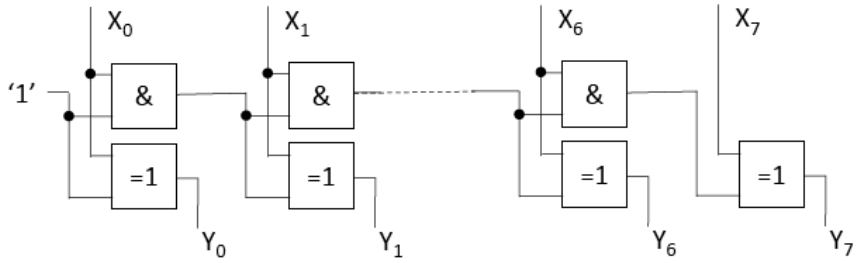


Figure 1.2

(c) *Figure 1.3* shows an R-2R ladder network used in an 8-bit Digital-to-Analogue Converter (DAC). Show that, under certain assumption, the current I_k at the k^{th} branch is given by: $I_k = 2^{k-8} I_T$. State the assumption.

[8]

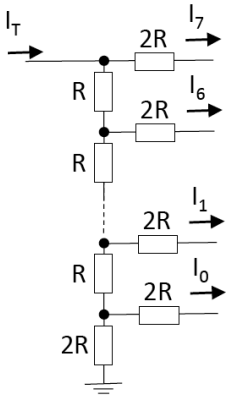


Figure 1.3

(d) In the circuit of *Figure 1.4*, the propagation delay of the flip-flops is t_p = 4 ns, and the setup and hold time are t_s = 1.5 ns and t_h = 1 ns respectively. The inverter and the logic module LUT has a propagation delay in the range of 1 ns < t_d < 3 ns. The clock signal *CLK* is symmetrical with period T.

(i) Derive the inequalities for the setup and hold times applied to the rightmost flip-flop.

[6]

(ii) Hence or otherwise, find the maximum clock frequency for the circuit.

[2]

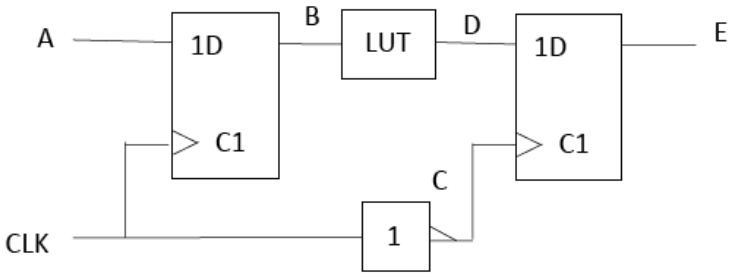


Figure 1.4

(e) *Figure 1.5* shows the circuit of a synchronous state machine with one input X and four states defined by the 2-bit binary number S0:1.

(i) Construct the state table for the circuit so that when X = 0, the state S0:1 follows the sequence 0, 1, 2, 3, 0, …., and when X = 1, it follows the sequence 0, 1, 3, 2, 0, ….

[4]

(ii) Draw a state diagram for the circuit.

[4]

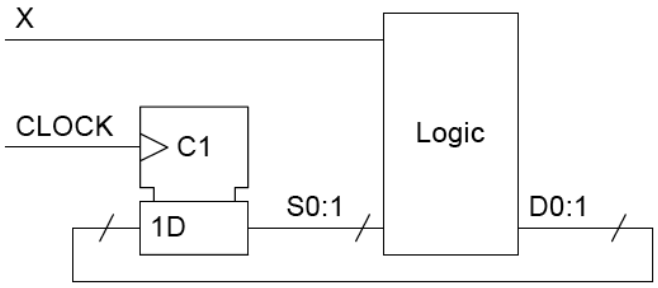


Figure 1.5

2. *Figure 2.1* shows a finite state machine (FSM) with 1-bit input IN and 2-bit output OUT1:0, implemented using an 8 x 4-bit ROM and a 4-bit register. The input IN to the FSM drives the upper bit of the ROM address A2. The output of the ROM D3:0 is connected to the data input of the register. The lower 2 bits of the ROM data D1:0 provide the next state value, which are fedback to the lower 2 bits of the ROM address A1:0 through the state register S1:0. The upper 2 bits of the ROM data D3:2 provide the FSM output value. The reset input RST clears the register.

Figure 2.2 shows the state diagram of the FSM.

(a) Determine the contents of the ROM in order to implement this FSM.

[8]

(b) A Logic Element (LE) in Altera's Cyclone II FPGA device consists of a 4-input lookup table and a register. Design the SSM using LEs in the Cyclone II device using *one-hot state encoding*. Estimate how many LEs are required to implement the FSM.

[12]

(c) Write a synthesizable Verilog module to describe the FSM in (b).

[10]

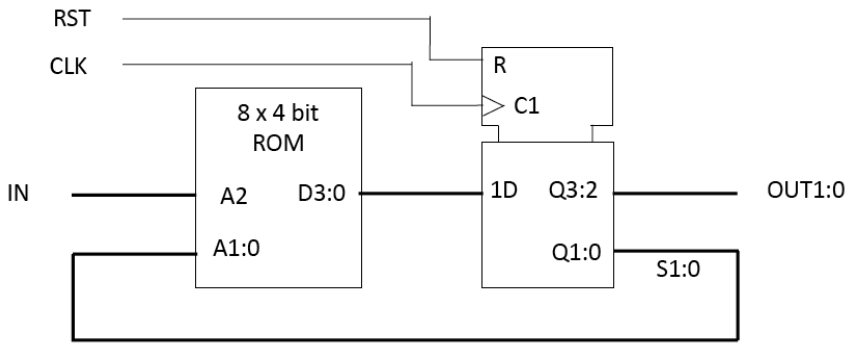


Figure 2.1

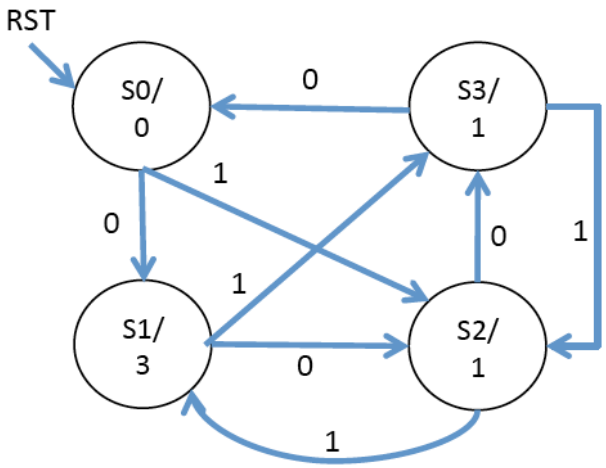


Figure 2.2

3. For this question, assume that the following timing specification applies:

Propagation delay of all combinatorial gates	t_p	5.5 ns
Setup time for D flipflop	t_{su}	2 ns
Hold time for D flipflop	t_h	0 ns
Clock to output delay for D flipflop	t_{cq}	7.5 ns

(a) *Figure 3.1* shows a simple clocked circuit. Draw the waveforms for the signals *P*, *Q*, *X* and *Y* over 5 clock cycles indicating the time at all signal transitions.

[12]

(b) What function does this circuit perform? What is the maximum clock frequency beyond which the circuit fails?

[6]

(c) *Figure 3.2* shows a clock generation circuit. Assume that CIN is a symmetric clock signal, draw the timing waveform for the output signals A and B. What is the relationship between B and CIN? What is the maximum frequency of CIN for this circuit to work correctly?

[12]

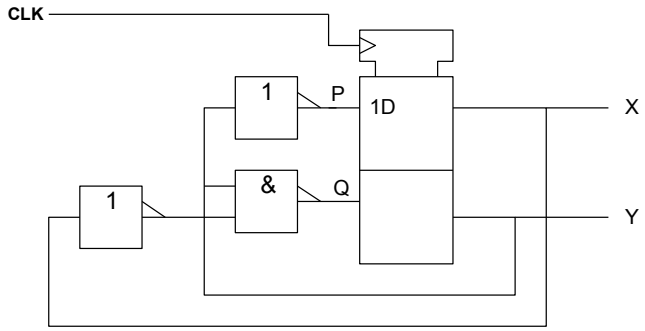


Figure 3.1

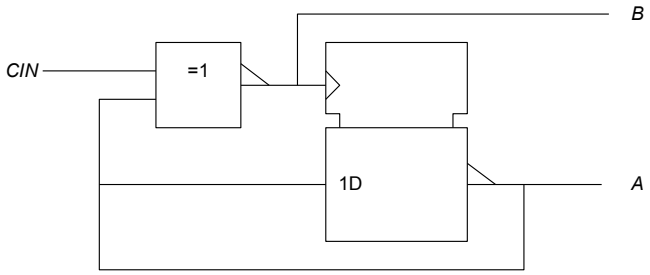


Figure 3.2

2012 Paper E2.1/ISE2.2: Digital Electronics II- Solutions

(with comments)

1. (a) This question tests student's basic knowledge on Verilog coding and FPGA timing.

One possible solution: [4]

```
module  Q1 (P, Din, clk);
    input       Din, clk;
    output      P;
    reg         P;
    always @ (posedge clk)
    begin
          P <= P ^ Din
    end
endmodule
```

Worst-case delay is through the LUT (implementing the XOR) + setup time + register delay. Therefore clock period T >= 3 ns. Fmax = 333.3 MHz. [4]

(Most students who had learnt some Verilog during the course did this part well. There is of course no unique answer to the Verilog code. Some students made a meal out of this with a complicated multiple if-else type of construct. The part on timing is more variable. Some considered the path from input-gate-setup etc. as the path that determines the max frequency, which is of course wrong. The critical path is the feedback path round the FF and the LUT. This is fundamental to any sequential circuit timing.)

(b) This tests student's grasp of basic computation circuit. Working through the circuit shows that if X7:0 = AF_{16} then Y7:0 = $B0_{16}$.

[3]

The circuit basically adds one to the input by detecting where lower bits stop being consecutive '1's, and invert it and all the lower bits! Very clever increment-by-1 circuit. Therefore Y = X + 1. Assumption is that X is unsigned integer.

[5]

(Majority of students managed to get something on this question. Some made the mistake of working out the Boolean equations for each output (there are 8!), and then worked out the Y value for X=AF. This would take ages. I would actually work it out simply from the diagram, propagating X7:0 through the circuit to get the answer. It would then be very easy to see that it is actually a +1 circuit. Of course if you don't get the first part, the chance is that you can't get the second part either.)

(c) This question is basically bookwork. Solution is from the notes.

The basic assumption is that the output node must be at ground potential, as can be achieved with a virtual earth node. [2]

With this assumption, moving from the lowest node up. At every node, the current is split into equal half. The equivalent resistance seen below the node is 2R//2R = R. This is in series with the vertical resister R before reach the next upper node. Hence I_T is split by ½ each time a node is reached. Hence the relation: $I_k = 2^{k-8} I_T$. [6]

(This DAC question is actually straight from notes. However, majority of students lost 2 marks because they did not specify that the output nodes must be at ground potential (not connected to earth) usually through a virtual earth. Otherwise, it is quite easy to show the specified equation is true.)

(d) This question tests student's ability to work out setup and hold timing constraints.

Setup inequality:

$t_p + t_{d\_max} + t_s < t_{d\_min} + ½\ T$

⇨ $4 + 3 + 1.5 < 1 + ½\ T$

⇨ T > 15ns [3]

Hold inequality:

$t_{d\_max} + t_h < ½\ T + t_{d\_min} + t_p$

⇨ $3 + 1 < ½\ T + 1 + 4$

⇨ T > -2 ns [3]

Hence T > 15ns, fmax < 66.7MHz. [2]

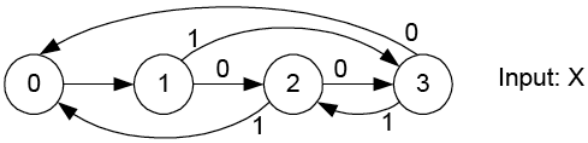
(The most common mistakes are that the value td appears in the setup and hold constraint equations, but not specifying whether it is t_{d_max} or t_{d_min}. Another common mistake is using T instead of ½T. The majority of students answered correctly.)

(e) This tests student's understanding of basic state machines.

(i) From the specification, we get the following state transition table: [4]

D1,D0	X	
S1,S0	0	1
00	01	01
01	10	11
11	00	10
10	11	00

(ii) From the state transition table, we can derive the transition diagram: [4]



(This is a straight forward question where most students gain maximum marks.)

(Comments on Q1: The level of difficulties of this "mastery" question, which tests students' basic understanding of the subject, is appropriate. The average marks for the year is ???, which is close to the target of 30 marks out of a possible total of 40.)

2. (a) This tests student's ability to analysis a FSM designed using ROM. [8]

ROM Address A2:0 (IN S1:0)	ROM Data D3:0
0 00	11 01
0 01	01 10
0 10	01 11
0 11	00 00
1 00	01 10
1 01	01 11
1 10	11 01
1 11	01 10

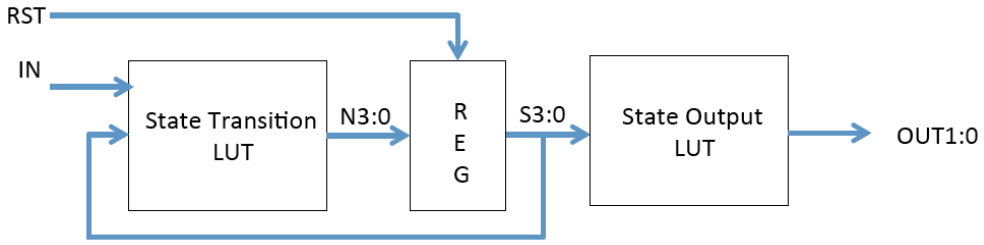
(I was surprised by how many students found this question difficult to answer. My intention was to combine in the same question something about FSM, ROM and FPGAs. I think this made the question unfamiliar and appears harder than it really is. It was not help by undiscovered typo-errors. A3 in the question was meant to be A2, otherwise A2 becomes a 'don't care' input, which makes the question harder.

There is nothing strange about using ROM to implement the combinatorial part of a FSM. However, most students went to great length in drawing various transition tables from the state diagram, before working out what the ROM contents should be. In fact the table able could be filled directly by inspecting Fig 2.2. It could have been an easy 8 marks, but unfortunately many did not do so, and probably spent too much time on this part.

A small number of students, who obviously were in a habit of memorising methods, did not even answer the question. They blindly design a state machine in the form of Boolean equations – a waste of time for hardly any marks!

There is one 'tricky' part to this question. Many students did not notice that there are registers for the output values. Therefore D3:2 should show NEXT OUTPUTS, not CURRENT STATE OUTPUTS. Anyone with this mistake only lost 1 mark.)

(b) This tests student's ability to design with a FPGA logic element and map a FSM to it. [12]



Note that one-hot state encoding makes the design much easier. The states are now encoded as 0001, 0010, 0100, and 1000. The Boolean equations for N3:0 and OUT1:0 could be almost written down directly from the state diagram!

N0 = S3*!IN + !S0*!S1*!S2*!S3 (2 LEs)
(the 2nd term is not necessary, ensure all zero state will go back to state 0)
N1 = S0*!IN + S2*IN (1 LE)
N2 = S1*!IN + S0*IN + S3*IN (1 LE)
N3 = S2*!IN + S1*IN (1 LE)
OUT0 = S1 + S2 (1 LE)
OUT1 = S3

Assuming that N0 has the 2nd term, each LE can implement an arbitrary 4-input function (including the register). Therefore total LEs = 6. (OUT1 does not need any logic).

(Most students could not do this part. I think I must take the responsibility of the ambiguity in the question. The word "implement" is too vague, and it would not be clear what is to be expected from the question. One-hot encoding is something I mentioned a lot during lectures. Still some students encode state 0 with 000, which is a mistake. Here are some of the mistakes by students:

1. Ignore the specification of one-hot encoding and continue to use binary encoding.
2. Use LE to implement ROM in a) and use 1-hot encoding at same time, making the answer long and clumsy.
3. Lots of details of content of LE, which is totally not required (but partly my fault in not notice other possible interpretation of the word "implement").

All in all, this is really hard question and the average marks is lower than my target of 15 marks.)

(c) This test student's ability to specify a FSM using Verilog. [10]

```
module  Q2c_FSM (OUT, IN, CLK, RST);
    input       IN, CLK, RST;
    output      [1:0] OUT;
//  define states
    parameter  NSTATE = 4;
    parameter   S_0 = 4'b0001;
    parameter   S_1 = 4'b0010;
    parameter   S_2 = 4'b0100;
    parameter   S_3 = 4'b1000;

    reg  [NSTATE-1:0]     state;
    wire [1:0]            OUT;
    wire                  IN, RST, CLK

    // specify state machine
    always @ (posedge CLK)
     if (RST == 1'b1) begin
          state <= S_0;
          OUT < = 2'b00;  end
    else
          case (state)
          S_0: begin
                  OUT <= 0;
                  If (IN == 1'b0)
                     state <= S_1;
                  else  state <= S_2;
               end
          S_1: begin
                  OUT <= 3;
                  If (IN == 1'b0)
                     state <= S_2;
                  else  state <= S_3;
               end
          S_2: begin
                  OUT <= 1;
                  If (IN == 1'b0)
                     state <= S_3;
                  else  state <= S_1;
```

```
            end
        S_3: begin
               OUT <= 2;
               If (IN == 1'b0)
                   state <= S_0;
               else  state <= S_2;
            end
        else                        // defensive coding
            state <= S_0;
            OUT <= 0;
          end
endmodule
```

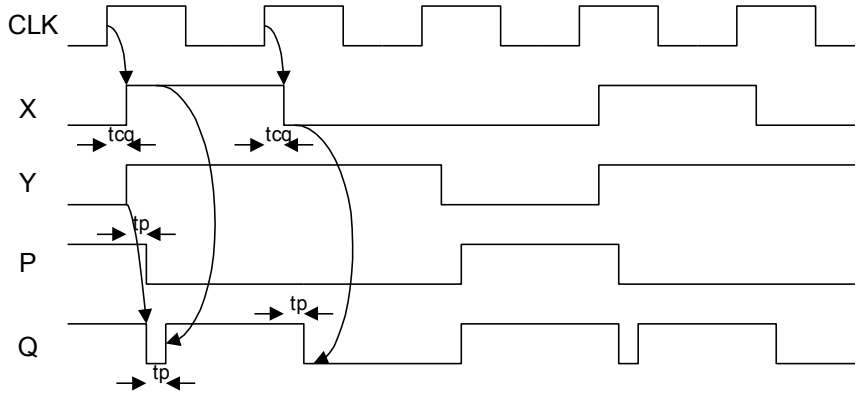
(Most students who attempted this part did very well. This type of questions is easy to earn marks because there are not too many ways to specify a FSM in Verilog. However, I did not expect a possible misunderstanding from some students – by saying that they should specify their design in b), they assume that they should have a one-to-one mapping of the design in b) to Verilog code. This created a number of problems:

1. They would then use structural description with detail of the design in b), instead of the much better behavioural description as shown here.
2. If a student could not do b), he/she thought c) also cannot be attempted. In fact, the only important part of b) is the one-hot encoding specification. Using generic FSM description in Verilog, detail answer in b) is irrelevant as shown above.

I therefore inadvertently made this question much harder than intended.)

3. This tests student's understanding of basic digital circuits, timing diagram and timing constraints.

a)



[12]

(Around half the class got this one right, although most did not get the glitches of Q, but I did not deduct marks as long as X and Y are correct.

This question is intended to test student's ability to draw timing diagrams – something that's very important to the understanding of digital circuits. Some students wrongly derive Boolean equations, state transition diagram and table, before even attempting the timing diagram. I deliberately not specify the initial value of X and Y (i.e. initial state of circuit), because it would not have mattered – after 4 cycles, one would get back to any original state anyway, so the waveform would look the same, but time-shifted. Asking for time of transition made this question harder and also time-consuming. With hindsight, I should not have required this and therefore I did not deduct marks even if the exact timing was not indicated in the diagrams.)

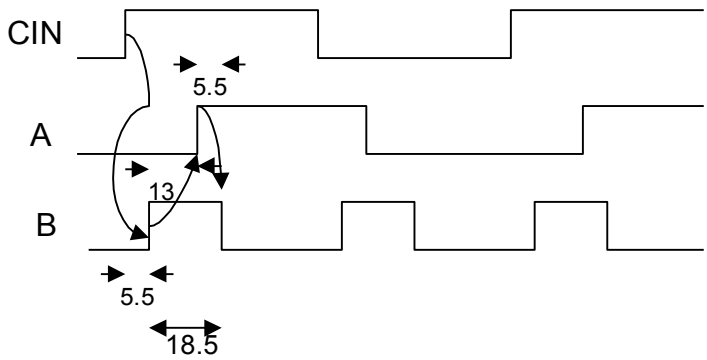
(b) This is a 1/3 and 2/3 divider circuit.

Tperiod ≥ Tcq + Tsu + 2* Tp ≥ 20.5 ns. Therefore Fmax <= 48.78MHz

[6]

(Part b) could be answered without getting part a) correct. Indeed a significant number of students got nearly full marks in b) while getting a) wrong!)

(c)



This is a clock frequency doubler. Maximum period of CIN is

2*(2*tp + tcq) = 2*18.5 = 37ns. Fmax = 27MHz.

[12]

(Many students did not get the clock doubling function at all because they did not analyse the timing diagram correctly. Some show the waveform for B as glitches (2 per cycle) which is OK. Overall, most students did well in this question, although it is a bit time-consuming if one is not used to drawing timing diagrams.)

(Comments on the whole paper:

I think this paper is a bit long and have too much timing relating components. As a 1.5 hour paper, unless students is really good and did not stumble on any question, one would probably run out of time while attempting the third question. However, early indications are that the mark profile and the average marks are both consistent with previous years.)