ELEC50001 – Circuits and Systems

2020 - 2021

Answer ALL questions.

There are THREE questions on the paper.

Question ONE counts for 50% of the marks, other questions 25% each

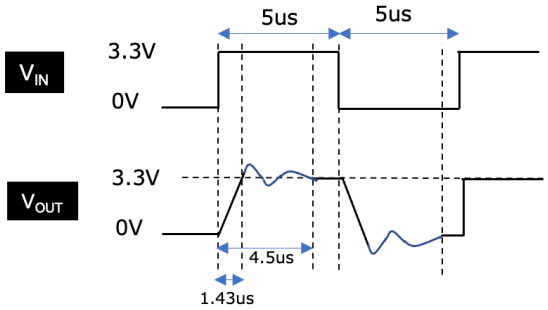
Time allowed: 2 hours

SOLUTIONS

1. (a) This question tests student's ability to read datasheet of an op-amp, and to demonstrate their understanding of the implications of the various parameter and thus what to expect from the performance of the op-amp.

(i) The relevant parameter are those related to step response:

Step Response						
Slew Rate	SR	—	2.3	—	V/µs	G = +1 V/V
Settling Time (0.01%)	t_{settle}	—	4.5	—	µs	G = +1 V/V, 3.8V step



The most important part is the slew rate limitation. The typical settling time specification is for 3.8V step in datasheet and is less important for this case.

[4]

(ii)

Frequency Response					
Gain Bandwidth Product	GBWP	—	2.8	—	MHz

The gain-bandwidth product of this op-amp is 2.8MHz. Therefore the gain for a sinewave at 2.8MHz is -3dB. The DC remains the same at 1.5V, but the peak-to-peak amplitude is reduced to $1/\sqrt{2}$ or 0.71V. Hence the signal at Vout is a sinewave at 2.8MHz between 1.15V and 1.85V.

[3]

(iii)

Input Current and Impedance					
Input Bias Current	I_B	—	1	—	pA
Industrial Temperature	I_B	—	20	60	pA
Extended Temperature	I_B	—	450	5000	pA
Input Offset Current	I_{OS}	—	±1	—	pA
Common Mode Input Impedance	Z_{CM}	—	$10^{13}\|\|6$	—	Ω||pF
Differential Input Impedance	Z_{DIFF}	—	$10^{13}\|\|3$	—	Ω||pF

Input impedance is so high that we can use, say 100k resistors for R3 and R4 to establish a 2.5V dc offset at +ve input of the op-amp. The lower break frequency is:

$f_{lower} = \frac{1}{2\pi(R3||R4)C1} \ll 10Hz$ (say, = 1Hz)

Therefore C1 = 3.2uF.

Assuming impedance of C2 << R1, then the gain of the amplifier is 1+R2/R1 = 10. Therefore R2/R1 = 9. If R2 = 180k, then R1 = 20k.

Again assuming the corner frequency for R1 and C2 is 1Hz as before, C2 = 5*3.2uF = 6uF.

[5]

(b) This question tests student's knowledge about address decoding (normally applied to memory circuits, but here is a slight twist using IP addresses instead). At a simpler level, it also tests student's ability to convert between decimal and hexadecimal code.

(i) This is like decoding 32-bit memory address where only the top 14-bits need to be decoded. The bottom 18-bits are don't cares. This is because

192 is 8'hC0 and 168 is 8'hA8 and 171 is 8'hAC. Therefore the address range for decoding is:

1100 0000 1010 1000 0000 0000 0000 0000 to

1100 0000 1010 1011 1111 1111 1111 1111

Therefore the 14-bit to decode is: **11 00 00 00 10 10 10** **or** 14'h302A.

Hence

$$Y = IP[31] \,\&\, IP[30] \,\&\, \sim IP[29] \,\&\, \sim IP[28] \,\&\, \sim IP[27] \,\&\, \sim IP[26] \,\&\, \sim IP[25] \,\&\, \sim IP[24] \,\&\, IP[23] \,\&\, \sim IP[22] \,\&\, IP[21] \,\&\, \sim IP[20] \,\&\, IP[19] \,\&\, \sim IP[18]$$

[4]

(ii) Two possible implementations:

Straight forward, lots of typing version:

```
1  module ip_detect(ip, match);
2     input [31:0]   ip;   // 32-bit IP address
3     output         match;   // high if address is between 192.168.x.x and 192.171.x.x
4
5     assign  match = ip[31]&ip[30]&~ip[29]&~ip[28] & ~ip[27]&~ip[26]&~ip[25]&~ip[224]
6                     & ip[23]&~ip[22]&ip[21]&~ip[20] & ip[18]&~ip[18];
7  endmodule
```

Smart version:

```
1  module ip_detect(ip, match);
2     input [31:0]   ip;   // 32-bit IP address
3     output         match;   // high if address is between 192.168.x.x and 192.171.x.x
4
5     assign  match = ip[31:18] ==14'h302A;
6  endmodule
```

[4]

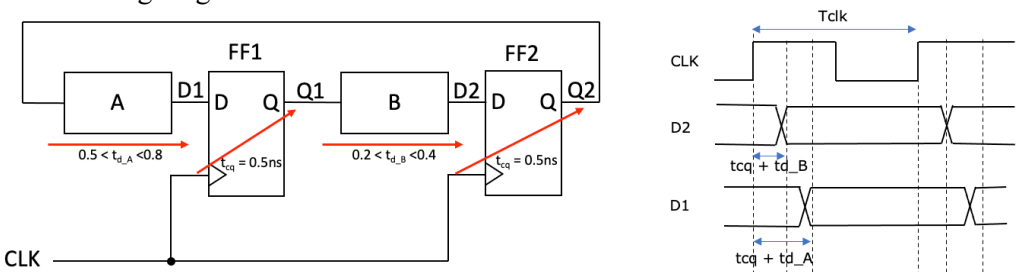
(c) This question tests student's ability to write a simple Verilog HDL code.. The answer is not unique and here is a possible solution.

[8]

```
1   module updn (tick, up, value);
2
3      input          tick;       // clock signalinput
4      input          up;         // high to count up
5      output [6:0]   value;
6
7      reg [6:0]      value;
8
9      initial value = 7'b0;
10
11     always @ (posedge tick)
12        if (up == 1'b1) begin
13           if (value!=7'd100)
14              value <= value + 1'b1;
15           end
16        else begin
17           if (value!=7'd0)
18              value <= value - 1'b1;
19           end
20  endmodule
```

(d) This question tests student's understanding of timing constraints in digital circuit.

(i) The timing diagram is:



Ignore any hold time violation, the critical path is from Q2 to D1. This gives:

$$T_{CLK} \geq t_{cq(\max)} + t_{d\_A(\max)} + t_s$$

Therefore,

$$T_{CLK} \geq 1.8ns, \quad f_{CLK} \leq 555.5MHz$$

[4]

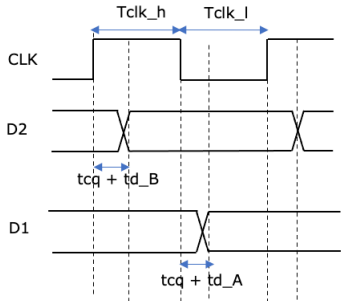
(ii) Hold-time violation occurs when signal changes too soon after the active clock edge. In this case, we need to use the minimum timing constraints.

For FF2, $t_{cq(\min)} + t_{d\_B(\min)} \geq t_h$. But 0.5 + 0.2 = 0.7ns$\leq$ t_h which is 0.9ns. Therefore there is hold-time violation.

For FF1, $t_{cq(\min)} + t_{d\_A(\min)} = 0.5 + 0.4 \geq t_h$, which is 0.9ns. So no hold-time violation.

[4]

(iii) Here is the new timing diagram:



Setup-time requirement is same as before excepted that clock is now divided into high and low period.

For FF1:

$$T_{CLK\_l} \geq t_{cq(\max)} + t_{d_{A(\max)}} + t_s \geq 1.8ns$$

For FF2:

$$T_{CLK\_h} \geq t_{cq(\max)} + t_{d_{B(\max)}} + t_s \geq 1.4ns$$

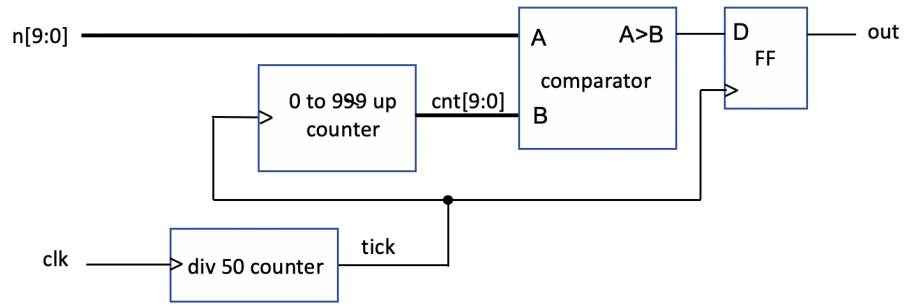
Since the clock is symmetrical, clock period is 2* max(T_{CLK_l}, T_{CLK_h}), or $f_{CLK} \leq 277.8MHz$.

As can be seen in the timing diagram, hold-time is never violated.

[4]

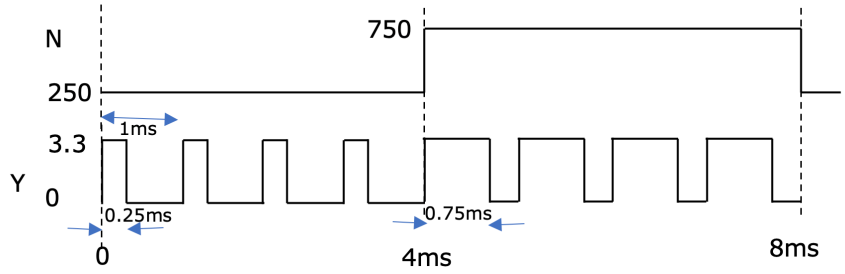
(e) This question test student's able to reverse engineer the code to hardware, and the idea of PWM.

(i)



[5]

(ii)



[5]

2. This question tests student's understanding of how FSM is implement on FPGA hardware, the FPGA internal circuits, Mealey FSM, one-hot encoding and how to specify a FSM in Verilog HDL.

(a)

S1, S0	NS1,NS2 / Q P=0	P=1
0 0	0 1/0	0 1/0
0 1	1 0/1	1 1/0
1 0	1 1/1	0 0/0
1 1	0 0/0	1 0/1

[5]

(b) A3 is always 0:

P S1 S0 / A2 A1 A0	S1'	S0'
	Y1	Y0
0 0 0	0	1
0 0 1	1	0
0 1 0	1	1
0 1 1	0	0
1 0 0	0	1
1 0 1	1	1
1 1 0	0	0
1 1 1	1	0

[7]

(c) LE2 is LUT only with D-FF bypassed.. The LUT content (A3 is again 0) is:

P S1 S0 / A2 A1 A0	Q
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	0
1 0 0	0
1 0 1	0
1 1 0	0
1 1 1	1

[5]

(d)

```
1   module FSM (clk, p, q);
2      input    clk, p;
3      output   reg   q;
4
5      parameter S_0 = 4'b0001;
6      parameter S_1 = 4'b0010;
7      parameter S_2 = 4'b0100;
8      parameter S_3 = 4'b1000;
9
10     parameter NSTATE = 4;
11     reg [NSTATE - 1:0] state;
12     initial state = S_0;
13
14  //--------state machine transition-----------//
15     always @ (posedge clk)
16        case (state)
17           S_0:  state <= S_1;
18           S_1:  if (p == 1'b1) state <= S_3;
19                 else  state <= S_2;
20           S_2:  if (p == 1'b1) state <= S_0;
21                 else  state <= S_3;
22           S_3:  if (p == 1'b1) state <= S_2;
23                 else  state <= S_0;
24           default: state <= S_0;
25        endcase
26
27  // -------- state machine outtput -----//
28     always @ (*)
29        case (state)
30           S_0:  assign q = 1'b0;
31           S_1:  assign q = ~p;
32           S_2:  assign q = -p;
33           S_0:  assign q = p;
34           default: assign q = 1'b0;
35        endcase
36  endmodule
```

[5]

3. This question tests student's understanding of R-2R ladder network, and then apply the same principle to a completely new circuit (that is actually not that useful but interesting.

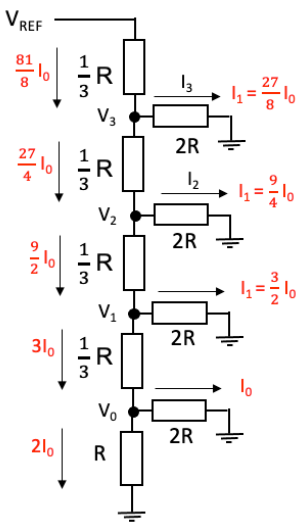
(a) Bookwork.

Every node "sees" a resistance of R, made up of 2 branch resistors 2R, splitting the current equally. Therefore

$I_3 = V_{REF}/4R$ $I_2 = V_{REF}/8R$ $I_1 = V_{REF}/16R$ $I_0 = V_{REF}/32R$

$V_3 = V_{REF}/2$ $V_2 = V_{REF}/4$ $V_1 = V_{REF}/8$ $V_0 = V_{REF}/16$

[5]



(b) This part is quite tricky. With this network, all branch current split into 1/3 on the right brand and 2/3 on the downward branch. Further, every node now "sees" a resistance of R||2R = 2/3 R. Starting from I_0 and work up the ladder. The right branch resistor is connect to either GND or virtue earth. Therefore we get:

Hence: $V_{REF}/R = (81/8)I_0 \Rightarrow I_0 = (8/81)\, V_{REF}/R$.

$I_1 = (4/27)\, V_{REF}/R$, $I_2 = (2/9)\, V_{REF}/R$, $I_3 = (1/3)\, V_{REF}/R$.

$V_0 = (16/71)V_{REF}$, $V_1 = (8/27)V_{REF}$, $V_2 = (4/9)V_{REF}$, $V_3 = (2/3)V_{REF}$

[10]

(c) The right terminal of all 2R resistors are connected either to GND or virtual GND of the inverting op-amp, which sum the current that flow into the negative input of the op-amp. The SPDT switch steers the current, hence minimizing the current swings and eliminate the problem of changing current quickly. The op-amp acts as a current summer that add the branch current where the digital control signal is 1.

[5]

(d) $V_{OUT} = -2R(I_3 + I_1) = -2VREF(1/3 + 4/27) = -(26/27)VREF = = -4.81V$.

[5]

