

ELEC50001 – Circuits and Systems

2021 - 2022

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 parameters are:

Parameter	Conditions	Min	Typ	Max	Unit
DYNAMIC PERFORMANCE					
-3 dB Bandwidth, $V_o \leq 0.2V$ p-p	$G = +1$	12	15		MHz
Full Power Response	$V_o = 2V$ p-p		3.2		MHz
Slew Rate	$G = -1, V_o = 2V$ Step	13	20		V/ μ s
Settling Time	$G = -1, V_o = 2V$ Step		250		ns
			to 0.01%	300	

Assumption: we use typical parameter.

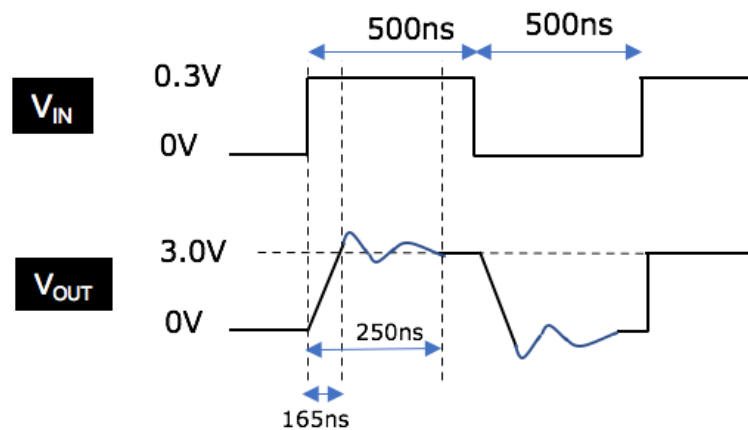
This is a non-inverting x10 amplifier. Therefore, ignoring gain-bandwidth product limitations, the output should be a sinewave with dc offset of 2V and amplitude of 1V. Since the supply is 3.3V, and AD823 has rail-to-rail capability, the output can swing between 1V to 3V.

This is however only part of the answer. The gain bandwidth product for small output swing (0.2Vp-p) is 15MHz. This is much higher than the signal frequency.

However, with $R_L = 2k$, the GBP drops to 3.2MHz. The gain is now -3dB lower than 20dB. Therefore, the amplitude is drop to 0.707V. Hence the correct answer would be between 1.3V and 2.7V.

[4]

- (ii) The slew rate is 20V/us, therefore it takes 165ns to rise to 3.0V. However, the settling time is longer at 250ns.



[3]

(iii)

This is actually a recommended circuit from the datasheet (which is not provided during the in-person examination).

- I. The gain of the amplifier is 1.5 at the audio frequency range.
- II. The DC operating voltage at A and B is 1.1V, and at C is $1.1 \times 1.5V = 1.65V$.
- III. The input resistance seen from the $1\mu F$ capacitor is $R//2R = 2/3 R$. The value of R must be chosen such that $2/3 \times RC \ll 20Hz$. For example, $R = 48k$.
- IV. C_{out} again should be chosen such that $C_{out} \times 32 \ll 20Hz$. Therefore C_{out} could be 500uF.

[5]

FEEDBACK COMMENTS:

There was a mistake in reproduction of this examination paper, which resulted in the wrong datasheet was initially distributed with the examination paper. It was corrected after 15 minutes and students were given extra time at the end of the examination to compensate.

In true, this mistake (entirely my (Peter Cheung's) fault) caused disruption to student's flow and resulted in some students probably doing worse than they would if this did not happen. I therefore tried to marked more leniently on Q1a).

- (i) Many students tried to derive the gain of the circuit from first principle – this resulted in too much time being spent for a question that is worth only 4 minutes. In previous years, I gave a revision lecture before the examination started when I taught students examination techniques. It did not happen this year (not sure why) and this definitely disadvantaged students.

Most students managed this part of Q1a). Also most students did not consider the impact of GBP on this part of the question. They lose 1 mark out of 4.

- (ii) Significant number of students only consider the gain of the circuit and NOT slew rate. Given that a similar question appears last year, I am surprised that that number of student did not get this part of Q1a)
- (iii) With hindsight, this part of Q1a) is probably too long – takes more than 5 minutes.

General comments on Q1a)

- Given that this question is similar to Q1a) of last year's paper, I thought it would be rather easy for students. It turned out NOT to be. I think the reason is partly because these students are doing in-person examination for the first time. With hindsight, I think this question is harder and longer than desired.

(b) This question tests student's basic understanding of memory sizes and address decoding.

(i)

Signal	Address range
RAM_CS	0000 – 7FFF
ROM_CS	A000 - BFFF
IO1_CS	FE00 – FE03
IO1_CS	FF00 – FF03

[4]

(ii)

$$\text{RAM_CS} = \text{A15}$$

$$\text{ROM_CS} = \sim(\text{A15} * \sim\text{A14} * \text{A13})$$

$$\text{IO1_CS} = \sim(\text{A15} * \text{A14} * \text{A13} * \text{A12} * \text{A11} * \text{A10} * \text{A9} * \sim\text{A8} * \sim\text{A7} * \sim\text{A6} * \sim\text{A5} * \sim\text{A4} * \sim\text{A3} * \sim\text{A2})$$

$$\text{IO2_CS} = \sim(\text{A15} * \text{A14} * \text{A13} * \text{A12} * \text{A11} * \text{A10} * \text{A9} * \text{A8} * \sim\text{A7} * \sim\text{A6} * \sim\text{A5} * \sim\text{A4} * \sim\text{A3} * \sim\text{A2})$$

[4]

(iii) Simple Verilog implementation using continuous assignment.

[2]

FEEDBACK COMMENTS:

A large proportion of students did not attempt this. Many of those who attempted this question got it wrong. Firstly, this year's students appear to have problem handling HEX numbers. For example 32k byte of RAM correspond to hex 8000 is something that they found difficult. Given that this is something both in tutorial problems and past paper, I am surprised!

Also many students got (iii) wrong because they forgot what is "continuous assignment", and the Verilog code has clock signal etc.

- (c) This question examines student's understanding of digit circuit timing including setup and hold times, and the idea that inserting pipeline registers between logic potentially increases the system clock frequency.

(i)
$$\max(t_{c-q} + t_{lut} \times 2 + t_{setup}) \leq \min(t_{clk} + t_{buffer})$$
 [4]

(ii) $1 + 2 \times 5 + 2 \leq t_{clk} + 2$, therefore $t_{clk} \geq 11$ ns. $F_{max} = 91$ MHz [2]

- (iii) Clock frequency can be increased by inserting a flip-flop between LUT_1 and LUT_2. Assuming that the inserted FF is also clocked by the same clock signal as FF2, and have the same timing constraints as FF1 and FF2, then the setup time constraint becomes:

$$1 + 5 + 2 \leq t_{clk} + 2, \text{ and } t_{clk} \geq 6 \text{ ns, } F_{max} \text{ is now } = 167 \text{ MHz.}$$
 [4]

FEEDBACK COMMENTS:

Those who understood timing constraints found (i) and (ii) easy and got full marks for both. Those who only memorised formula got it wrong. Most common is not to account for t_{buffer} .

Solution of (iii) is missed by most. Apart from the solution above, another acceptable answer is to increase the clock buffer delay.

(d) This question tests student's understanding of basic DAC ideas.

(i) Resolution for 10-bit DAC is $3.3\text{V}/1024 = 3.22\text{mV}$ per LSB.

[2]

(ii) The reference voltage source "sees" a resistance of R (basic concept of the R - $2R$ network is that every node downstream appears as a resistor R).

$$I_{\text{in}} = 16 I_o = V_{\text{ref}}/R, \text{ therefore } I_o = 3.3\text{V}/(16 \times R) = 0.206/R \text{ A.}$$

With switch set at 9, the current flowing into the virtual earth terminal is $9 \cdot I_o$.

$$\text{Therefore } V_{\text{out}} = -9 \cdot 0.206 = -1.854\text{V.}$$

[6]

FEEDBACK COMMENTS:

Many students got it perfect. However, some students don't have an clue on (i) and some made (ii) much more difficult than desired.

- (e) (i) This question tests student's knowledge in designing a combinatorial circuit in Verilog, and show some knowledge in the FPGA internal architecture

```
module decoder (X, Y);
  input  [3:0] X;
  output [3:0] Y;

  reg  [3:0] Y;

  always @ (X)
  case (X)
    //***** input <5, pass to output unchanged *****
    4'b0000: Y <= 4'b0000;
    4'b0001: Y <= 4'b0001;
    4'b0010: Y <= 4'b0010;
    4'b0011: Y <= 4'b0011;
    4'b0100: Y <= 4'b0100;

    //***** input >=5 and < 12, output = input + 3 *****
    4'b0101: Y <= 4'b1000;
    4'b0110: Y <= 4'b1001;
    4'b0111: Y <= 4'b1010;
    4'b1000: Y <= 4'b1011;
    4'b1001: Y <= 4'b1100;
    4'b1010: Y <= 4'b1101;
    4'b1011: Y <= 4'b1110;

    //***** input >= 12, output = input - 2 *****
    4'b1100: Y <= 4'b1010;
    4'b1101: Y <= 4'b1011;
    4'b1110: Y <= 4'b1100;
    4'b1111: Y <= 4'b1101;
  endcase
endmodule
```

[8]

- (ii) Each LUT can take 4 inputs, and therefore can implement one output. There are four outputs, therefore we need 4 LUTs or 4 LEs.

[2]

FEEDBACK COMMENTS:

- (i) Many students forgotten how to specify combinatorial circuits in Verilog, particularly given that this is a close book exam. Many even include a clock signal. Given that we had the 7 segment decoder in Lab, I am disappointed that so few students map that across to this problem.
- (ii) Most students could not answer this – it shows that they have either forgotten the FPGA internal architecture or did not pay attention to what is inside the LE.

2. This question tests student's understanding application of op-amp from first principle.

(a) -ve input is virtual earth, therefore

$$I_{IN} = \frac{V_{IN}}{R}$$

Current most flow through capacitor C due to high input impedance of op-amp. Since $I_C = C \frac{dV_C}{dt}$,

$$I_{IN} = \frac{V_{IN}}{R} = -C \frac{dV_{OUT}}{dt}$$

Hence

$$\frac{dV_{OUT}}{dt} = -\frac{1}{RC} V_{IN} \text{ or } V_{OUT} = -\frac{1}{RC} \int V_{IN} dt$$

Take Laplace transform of both sides:

$$s V_{OUT}(s) = -\frac{1}{RC} V_{IN}(s)$$

Hence, transfer function

$$K(s) = -\frac{1}{RC} \times \frac{1}{s}$$

[5]

(b) The integrator now has three input branches. Using principle of superposition, we can consider these separately.

For X: $V_1(s) = -\frac{1}{10RC} \frac{1}{s} X(s)$

For Y: $V_1(s) = -\frac{1}{10RC} \frac{1}{s} Y(s)$

For lower branch: $V_1(s) = -\frac{1}{10RC} \frac{1}{s} \frac{V_1(s)}{2}$

Therefore, total $V_1(s)$ is sum of these three terms:

$$V_1(s) = -\frac{1}{10RC} \left(\frac{1}{s} X(s) + \frac{1}{s} Y(s) + \frac{1}{s} \frac{V_1(s)}{2} \right) \text{ Eq. 1}$$

Hence, $V_1(s) = -\frac{1}{\frac{1}{2} + 10RCs} (X(s) + Y(s))$ Eq. 2

[10]

(c) For the given component values, $R=100k$, $C=0.1\mu F$, therefore $RC = 0.01$.

$$X(s) = V_{OUT}(s) = \frac{1}{RC} \frac{1}{s} V_1(s) = 100 \frac{1}{s} V_1(s) \quad \text{Eq 3}$$

$$Y(s) = V_{IN}(s)$$

Substitute Eq 2 into Eq. 3 to eliminate $V_1(s)$ gives

$$\begin{aligned} V_{OUT}(s) &= 100 \frac{1}{s} V_1(s) = -100 \frac{1}{s} \left\{ \frac{1}{\frac{1}{2} + 0.1s} (V_{OUT}(s) + V_{IN}(s)) \right\} \\ &= - \left\{ \frac{1000}{5s + s^2} (V_{OUT}(s) + V_{IN}(s)) \right\} \end{aligned}$$

Therefore:

$$\begin{aligned} V_{OUT}(s) \left(1 + \frac{1000}{5s + s^2} \right) &= - \frac{1000}{5s + s^2} V_{IN}(s) \\ H(s) = \frac{V_{OUT}(s)}{V_{IN}(s)} &= - \frac{1000}{s^2 + 5s + 1000} \end{aligned}$$

[10]

FEEDBACK COMMENTS:

The majority of students did this well and many even got full marks. The problem is that those who apply KCL “blindly” took lots of time and pages to arrive at the final answer. As shown above, if one uses the principle of superposition, the solutions sort of fell out easily!

3. This question tests student's understanding of the serial peripheral interface, which is covered both through the lectures and the laboratory experiment.
- (a) This is relatively straightforward:

```
// --- Submodule: Generate internal clock at 1 MHz ----
reg      clk_1MHz; // 1Mhz clock derived from 50MHz
reg [4:0] ctr;    // internal counter
parameter TIME_CONSTANT = 5'd24; // change this for dif
initial begin
    clk_1MHz = 0; // don't need to reset - don't care
    ctr = 5'b0;  // ... Initialise when FPGA is config
end

always @ (posedge sysclk) //
    if (ctr==0) begin
        ctr <= TIME_CONSTANT;
        clk_1MHz <= ~clk_1MHz; // toggle the output clock for
    end
    else
        ctr <= ctr - 1'b1;
// ---- end internal clock generator -----
```

[5]

(b)

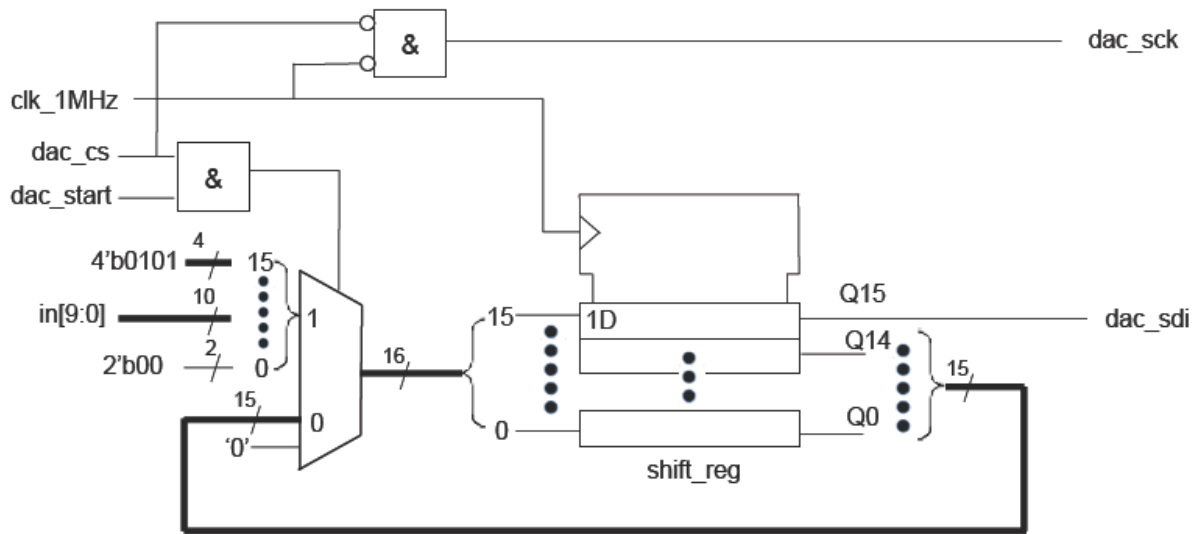
```
// ---- Detect posedge of load with a small state machine
// .... FF set on posedge of load
// .... reset when dac_cs goes high at the end of DAC output cycle
reg [1:0] sr_state;
parameter IDLE = 2'b00, WAIT_CSB_FALL = 2'b01, WAIT_CSB_HIGH = 2'b10;
reg      dac_start; // set if a DAC write is detected

initial begin
    sr_state = IDLE;
    dac_start = 1'b0; // set while sending data to DAC
end

always @ (posedge sysclk)
    case (sr_state)
        IDLE: if (load==1'b0) sr_state <= IDLE;
              else begin
                  sr_state <= WAIT_CSB_FALL;
                  dac_start <= 1'b1;
              end
        WAIT_CSB_FALL: if (dac_cs==1'b1) sr_state <= WAIT_CSB_FALL;
                      else sr_state <= WAIT_CSB_HIGH;
        WAIT_CSB_HIGH: if (dac_cs==1'b0) sr_state <= WAIT_CSB_HIGH;
                      else begin
                          sr_state <= IDLE;
                          dac_start <= 1'b0;
                      end
        default: sr_state <= IDLE;
    endcase
//----- End circuit to detect start and end of conversion
```

[10]

- (c) This part of the question tests student’s understanding of how Verilog HDL is mapped to digital circuits.



[10]

FEEDBACK COMMENTS:

- (a) Many students answer this part without symmetrical 1MHz clock, but a reproduction of the “tick” function in Labs.
 (b) Most students got this correctly – showing that they understood how to specify FSM in Verilog
 (c) Many ran out of time to do this. Those who knew Verilog well was able to sketch the circuit easily.

GENERAL COMMENTS ON THIS PAPER

I think I unwittingly made this paper too hard. It is similar level of difficulties as last year’s paper, except that last year, it was an open-book, remote-timed paper. This year was an in-person paper with a hiccup at the start of the examination.

Q1 average 26.4 out of 50

Q2 average 18.7 out of 25

Q3 average 10.9 out of 25

This is particularly true with Q1 – average only 26 out of 50 marks, when my target was 36 out of 50 marks. Q2 is what I expect with nearly 19 out of 25. Q3 average is lower, I think mostly due to running of time.

The overall average of Exam is 56% and Courework is 65%, which results in around 60% overall average. This is lower than last year, but not by too much.