IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2025

CIRCUITS AND SYSTEMS

Tuesday, 13 May 2025, 2:30 PM

Time allowed: 2 hours

SOLUTIONS

There are THREE questions on the paper.

Answer ALL questions.

1. (a) This question tests student's ability to use an op-amp and demonstrate their understand of its limitations.

(i) (Bookwork, but from first principle). C2 blocks dc from Vs. Therefore, the gain of the dc offset of Vs is 0.

R3 and R4 forms a voltage divider and must be equal to ac signal of Vs.

$$Vout * \frac{R3}{R4 + R3} = Vs$$

Hence the AC gain is (1+R4/R3) = 2.33.

Also, student may derive the equation relating Vout to Vs:

$$Vout = \frac{R4 + R3}{R4} * \left(Vs + 3.3 * \frac{R2}{R1 + R2}\right) = 2.33 * Vs + 2.56$$

[4]

(ii) V+ dc voltage is determined by R1 and R2 and it is 1.1V. The gain of the amplifier is 2.33 for both dc and ac at V+. Therefore, Vout an ac signal with pk-pk voltage of 2.33 x 1V = 2.33V, with a dc offset of 2.33 x 1.1V = 2.56V. Therefore Vout(max) = 3.73V and Vout(min) = 1.4V. Unfortunately, the power supply is at 3.3V.

Therefore, Vout is clipped for any voltage above 3.3V.

[3]

(iii) R1 and R2 gives a total resistance of R1 || R2 = 6.67k, with C2, the corner frequency is 24Hz.

Therefore, 1Hz sine wave does not pass through C2 unimpeded. Since the GBP is 1MHz, it also cannot provide a gain of 2.33 at this signal frequency. Therefore, the gain of the amplifier is lower than the predicted 2.33.

[3]

COMMENTS

(i) Most students got this part right. Some did not realised that the DC gain is 0, but report the DC value as the answer.

(ii) Some students failed to show the saturation at 3.3V. They would have lost 1 mark.

(iii) This is the harder part. Must consider both the lower corner frequency and the effect of the GBP. Many fail to realised that the 1HZ signal is not amplified.

(b) This question tests student's understanding pseudo random binary sequence generator circuit and how to implement such component in SystermVerilog..

(i) The signature for zero input is:

Cycle	Q[5:1]
1	00001
2	00010
3	00101
4	01010
5	10101
6	01011

[3]

(ii) When merging input sequence, the signature is:

cycle	in	Q[5:1]	Q5^Q2^in
1	0	00001	0
2	1	00010	0
3	0	00100	0
4	0	01000	0
5	0	10000	1
6	0	00001	0

[3]

(iii)

```
module signature (
    input  logic        clk,    // clock
    input  logic        init,   // reset ro initial state
    input  logic        in,     // input signal
    output logic [5:1]  Q       // output signature
);

    // shift register
   always_ff @(posedge clk)
    if (init)    Q <= 5'b00001;
    else         Q <= {Q[4:1], Q[5]^Q[2]^in};

endmodule
```

[4]

COMMENTS

Most students found this easy and many got full marks.

(c) This question examines student's understanding of propagation delay in digital circuit and how to calculate the maximum clock frequency in a sequential circuit..

(i) P0 to C2A = delay$_{\text{P0->CO}}$ + delay$_{\text{CI->CO}}$ + delay$_{\text{CI->CO}}$ = 2 + 2 + 2 = 6ns
C-1 to C2A = delay$_{\text{CI->CO}}$ + delay$_{\text{CI->CO}}$ + delay$_{\text{CI->CO}}$ = 2 + 2 + 2 = 6ns

Two paths from P0 to C2B: (accept either answer)

1) Via adder chain – delay from P0 to C2A + delay$_{\text{mux-data}}$ = 6 + 2 = 8ns
2) Via gates = delay$_{\text{xor}}$ + delay$_{\text{and}}$ + delay$_{\text{mux-sel}}$ = 2 + 1 + 3 = 6ns

Two paths from C$_{-1}$ to C2B: (accept either answer)

1) Directly from C-1 $\rightarrow$ C2B = delay$_{\text{mux-data}}$= 2ns
2) Via 3 adders + MUX = 8 ns

The following explanation is not expected, but it is included here for educational purpose in future years.

Note that although there is an apparent path $C_{-1} \rightarrow C0 \rightarrow C1 \rightarrow C2A \rightarrow C2B = 8.$

This in fact never actually arises - it is never true that a change in C-1 will cause this chain of cause and effect because it would require all the full-adders to be propagating the carry (to enable $C_{-1} \rightarrow C0 \rightarrow C1 \rightarrow C2A$ *).*

Also, SEL=0 (to enable the final step C2A → C2B). The entire point of the circuit (known as carry-skip adder) is that we prevent this from happening. Normally when analysing a circuit for the worst-case propagation delay, we assume that any path that is present in the circuit will be enabled for some set of input conditions; the carry-skip circuit is one of the rare cases when this assumption is untrue.

[6]

(ii) The worst-case path of the adder is $P0 \rightarrow C2B$ = 8ns.

P0 is the output of the pipeline register from the stage before. Therefore, it arrives 1.5ns after the clock edge. The setup time is 2ns. Therefore, the clock period Tc must be:

$$Tc \geq 8 + 2 + 1.5 \text{ ns.}$$

Hence maximum clock frequency is 1000/(11.5) MHz = 87MHz. Hold time does affect the operation of the circuit.

[4]

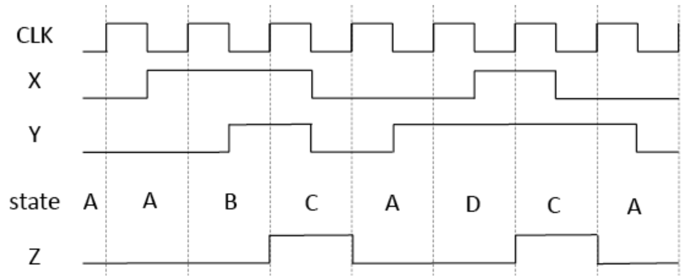
COMMENTS

(i) There is some ambiguity in the question. The critical path depends on the values of the numbers to be summed. Therefore, those who got most of the delay got full marks (2ns answer not required).

(ii) Full marks for those who include the 2ns setup and 1.5ns register delay, no matter what the adder delay value was used.

(d) This tests students' ability to analyse a state diagram and from this specify its implementation in SystemVerilog.

(i)



[4]

(ii) A sample SystemVerilog design:

```
module FSM (
    input   logic X, Y, CLK,
    output  logic Z
);

// define states
    typedef enum{SA, SB, SC, SD} my_states;
    my_state current_state, next_state;

// state transition
    always_ff @(posedge CLK)
        current_state <= next_state;

// next state logic
    always_comb begin
        next_state = current_state;
        case (current_state)
            SA: if      ({X,Y}==1'b10) next_state = SB;
                else if ({X,Y}==1'b01) next_state = SD;
                else if ({X,Y}==1'b11) next_state = SC;
            SB: if      ({X,Y}==1'b11) next_state = SC;
                else if ({X,Y}==1'b00) next_state = SA;
            SC: next_state = SA;
            SD: if      ({X,Y}==1'b00) next_state = SA;
                else if ({X,Y}==1'b11) next_state = SC;
            default:    next_state = SA;
        endcase
    end

// FSM outputs
    always_comb
        case (current_state)
            SA: Z = 1'b0;
            SB: Z = 1'b0;
            SC: Z = 1'b1;
            SD: Z = 1'b0;
            default: Z = 1'b0;
        endcase
endmodule
```

[6]

COMMENTS

(a) Nearly all students got full marks for this.

(b) Many students have z output delayed by 1 cycle and lost 2 marks. Those who paid attention in labs and actually wrote the FSM code for the challenges got perfect answers.

.

(e) This questions tests student's understand of flash ADC, voltage range and simple encoder to translate the thermometer code to binary code.

(i) Assumptions: all resistor values R are identical, the comparators have zero offset error, and when both inputs are the same, the comparator output is '1'.

ABCD	Vin
0000	Vin < 0
0001	0 ≤ Vin < 1
0011	1 ≤ Vin < 2
0111	2 ≤ Vin < 3
1111	Vin ≥ 3

[5]

(ii)

```
module encoder (
    input logic A, B, C, D,      // input signals
    output logic [2:0]  X;       // output signals
)
    always_comb
        case {A,B,C,D}
            4'b0000: X = 3'd0;
            4'b0001: X = 3'd1;
            4'b0011: X = 3'd2;
            4'b0111: X = 3'd3;
            4'b1111: X = 3'd4;
        default
            X = 3'd0;    // should never happen
        endcase
endmodule
```

[5]

COMMENTS

(a) The assumptions are important because the answer depends on whether output is '0' or '1' when both inputs of the comparator are the same. The vast majority of students had the range with > and <, therefore not specifying what happens when the input equals to the thresholds.

(b) There are many different answers to this. As long as they work, full marks was awarded.

2. 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. It further tests student's ability to derive SPICE model parameters from the op amp specification.

(a) The relevant parameters are:

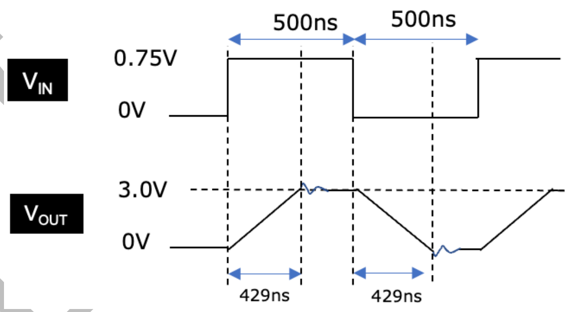
Parameters	Sym	Min	Typ	Max	Units	Conditions
AC Response						
Gain Bandwidth Product	GBWP	—	10.0	—	MHz	
Phase Margin at Unity-Gain	PM	—	65	—	°	G = +1 V/V
Slew Rate	SR	—	7	—	V/μs	

Assumption: we use typical parameter.
This is a non-inverting x4 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 MCP6291 has rail-to-rail capability, the output can swing between 1V to 3V without clipping.

However, the gain bandwidth product is only 10MHz. With a gain of 4, the break frequency of the x4 amplifier is expected to be at 2.5MHz. The gain at this frequency is 3dB (i.e. 0.707) lower than the expected gain of 4.
Therefore, the amplitude is drop to 0.7V. Hence the correct answer would be between 1.3V and 2.7V.

[6]

(b) The slew rate is 7V/us, therefore it takes 429ns to slew 3.0V. Hence the output waveform is as shown here:



[4]

(c) For maximum output voltage range, VOUT should be 3.3V when E_θ is 1mW/cm^2. From the photodiode characteristics, the reverse current is at about 70μA. Therefore, R should be 3.3V/70μA = 47kΩ.

[3]

(d) (i) From datasheet $Z_{DIFF} = 10^{13}\Omega \parallel 3pF$, R1 = 1TΩ, C1 = 3pF, V1 = V_{OS} = 3mV.

[2]

(ii) GBP = 10M. Open-loop gain A_{OL}=110dB or 400,000.
Dominant pole $f_p = \frac{GBP}{A_{OL}} = \frac{10^7}{400000} = 25Hz = 1/2\pi R_2 C_2$.
Assume $C_2 = 1nF$, R2 = 6.37MΩ.

$$gm \times R_2 = A_{OL}, \quad \text{hence} \quad gm = \frac{400{,}000}{6.37} \times 10^{-6} = 0.0628.$$

[6]

(iii) Slew rate = max(dV_{C2}/dt) = Imax_$_{G1}$/C2. Imax_$_{G1}$ = 7V/us * 1nF = 7mA.
Therefore, the SPICE statement for G1 is:

G1 0 int_gain value={limit(0.0638*V(offset, in-), 7m, -7m)}

[3]

(iv) D1 and D2 are to limit the output voltage to that of V+ and V-.

[1]

COMMENTS

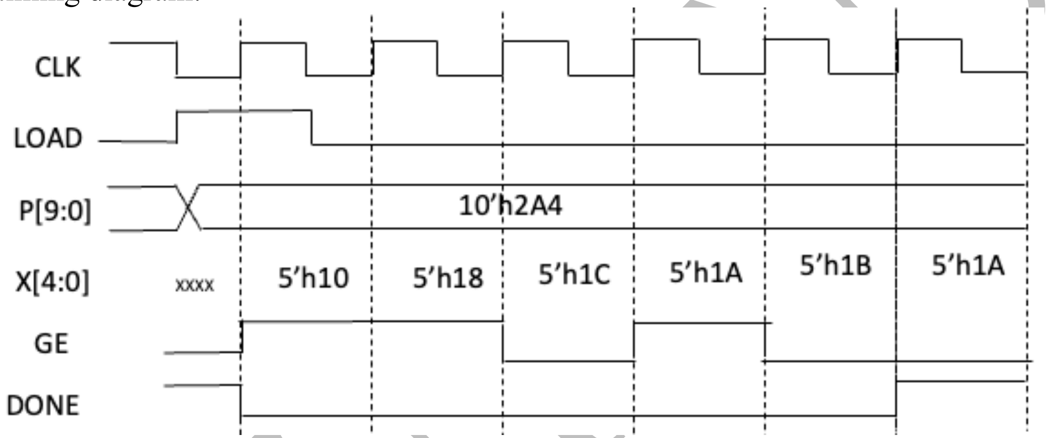
(a) Many found this easy. Those who did not consider the effect of GBP = 10MHz, gain = 4, and signal frequency is 2.5MHz at all lost 1 mark. But those who considered this but assuming the gain is sufficient (in spite of it being -3dB) did not lose mark.
(b) Easy questions were most students got right.
(c) This is ambiguous because some student assumed max Vout as some voltage other than 3.3V because I did not explicitly say so in the question. Further, some interpret the log scale in the diode characteristics wrongly, assuming the current to 40uA, or 60uA, instead of the 70uA used in the answer. So I gave full mark for all answers that got the method right, but the values different from the model answer.
(d) This proved to be the hard question. Most students did not get (ii) and (iii).

3. This question tests student's understanding the principle of successive approximation algorithm and how to apply it. It also tests student's ability to design a reasonably complicated FSM in SystemVerilog.

(a) Although not required by the question, it could be easier to draw up the following table:

X[4:0]	X^2	GE	RESULT
5'h10	10'h100	TRUE	X[4] = 1
5'h18	10'h240	TRUE	X[3] = 1
5'h1C	10'h310	FALSE	X[2] = 0
5'h1A	10'h2A4	TRUE	X[1] = 1
5'h1B	10'2C9	FALSE	X[0] = 0

The timing diagram:



[10]

(b) The principle of SAR is really bookwork. However, the application here is rather special. Instead of the DAC in a SAR based ADC, we use the multiplier circuit to perform the squaring function. The rest of the SAR algorithm is exactly the same as in the successive approximation DAC.

The assumptions are: 1) LOAD is asserted just before CLK going high, 2) P[9:0] is stable throughout the computation.

[5]

(c) The design of the FSM in SystemVerilog will depend on students.

The best solution would be to have idle state, following by 5 states, each for setting each bit of X from MSB to LSB. One can also use X[4:0] and DONE as the state variable.

The poor solution, although works, is to create 33 states, one for IDLE, waiting for LOAD to activate, followed by all possible state for 5 bits output.

[10]

COMMENTS

Most students found this question difficult. The problem being that many students over think the problem. If one understands the SAR algorithm, one can derive the timing diagram without knowing any implementation hardware. However, the complication comes from having to use binary, or better, hexadecimal. Those who convert numbers to decimal wasted lots of time fiddling with the conversion.

Notwithstanding, several students got full marks for this question with more-or-less working FSM for part (c).

OVERALL COMMENTS ON THIS PAPER

This paper turns out to be harder than expected. The target for Q1 average was 35/50. The class average was lower at 32.5/50 (65%), which is reasonable.

The problem comes with Q2 and Q3, where the class averages were 13.7/25 (45%) and 10.1/25 (40%) respectively. My target would have been 50% for both.

Overall, the class average was 56.7% which is lower than my target of 62%. Fortunately the coursework marks lifted this average to a respectable 60.6%.

13 out of 66 students got A grades for this module and no one failed overall.