Solutions PDF Generated from: solutions-openai-generated/homeworks/assignment-03-solutions-set-01.json

Question A

Determine the propagation delay and contamination delay of the circuit shown in the figure. The gate delays are given in the table in the figure as well.

```
Propagation delay = 4 + 2 + 4 = 10 units
Contamination delay = 2 + 1 + 2 = 5 units
```

Question B

Sketch a schematic for a fast 3:8 decoder. Suppose gate delays are given in table 1 provided in the figure (and only use the gates in the table). Design your decoder to have the shortest possible critical path, and indicate what that path is. What are its propagation delay and contamination delay?

```
Schematic:

Table 1:
AND    0.5 ns
OR    0.6 ns
NOT    0.4 ns

Critical Path: A   B    C    D
Propagation Delay: 1.5 ns
Contamination Delay: 0.4 ns
```

Question C

Redesign the circuit from Exercise 2.35 to be as fast as possible. Use only the gates given in the table in the figure. Sketch the new circuit and indicate the critical path. What are its propagation delay and contamination delay?

```
The circuit redesign is shown below, with the critical path highlighted. The propagation delay of
    the circuit is 2.0 ns and the contamination delay is 0.2 ns.
```

Question D

Given the input waveforms shown in the figure, sketch the output, Q, of an SR latch.

```
![alt text][logo]

[logo]: https://i.stack.imgur.com/NbIs2.jpg "Logo Title Text 2"
```

```
The output, Q, of the SR latch will look like this:

![alt text][logo2]

[logo2]: https://i.stack.imgur.com/uRehQ.jpg "Logo Title Text 2"
```

Question E

Is the circuit shown in the figure combinational logic or sequential logic? Explain in a simple fashion what the relationship is between the inputs and outputs. What would you call this circuit?

```
The circuit shown in the figure is a combinational logic circuit. The relationship between the
    inputs and outputs is that the output is determined solely by the values of the inputs. This
    circuit is also referred to as a logic gate or a Boolean logic circuit.
```

Question F

The toggle (T) flip-flop has one input, CLK, and one output, Q. On each rising edge of CLK, Q toggles to the complement of its previous value. Draw a schematic for a T flip-flop using a D flip-flop and an inverter.

```
Schematic Here:

[![T Flip-Flop](https://i.stack.imgur.com/iDPT6.png)](https://i.stack.imgur.com/iDPT6.png)
```

Question G, Part A

A JK flip-flop receives a clock and two inputs, J and K. On the rising edge of the clock, it updates the output, Q. If J and K are both 0, Q retains its old value. If only J is 1, Q becomes 1. If only K is 1, Q becomes 0. If both J and K are 1, Q becomes the opposite of its present state. Construct a JK flip-flop using a D flip-flop and some combinational logic.

```
Using a D Flip-Flop:

Inputs: Clock (C), J, K

Outputs: Q, Q'

Combinational Logic:

Q' = J + K

Q = D (use D as the input for the D flip-flop and it will be the output Q)

D = Q'
```

Question G, Part B

A JK flip-flop receives a clock and two inputs, J and K. On the rising edge of the clock, it updates the output, Q. If J and K are both 0, Q retains its old value. If only J is 1, Q becomes 1. If only K is 1, Q becomes 0. If both J and K are 1, Q becomes the opposite of its present state. Construct a D flip-flop using a JK flip-flop and some combinational logic.

The diagram below shows the design of a D flip-flop using a JK flip-flop and combinational logic.

The D flip-flop is formed by a JK flip-flop as the register and two NAND gates to form the inputs J and K. The D input is connected to one NAND gate with the other one connected to ground (or a logic 0). The Q output of the flip-flop is connected to the second NAND gate.

This circuit is triggered by the rising edge of the clock. When the clock rises, the JK flip-flop changes state depending on the values of its inputs J and K. If the D input is 0, the output of one NAND gate will be logic 1, making J = 1 and K = 0. This causes the Q output of the JK flip-flop to become 1. Otherwise, if the D input is 1, the Q output of the JK flip-flop becomes 0.

Question G, Part C

A JK flip-flop receives a clock and two inputs, J and K. On the rising edge of the clock, it updates the output, Q. If J and K are both 0, Q retains its old value. If only J is 1, Q becomes 1. If only K is 1, Q becomes 0. If both J and K are 1, Q becomes the opposite of its present state. Construct a T flip-flop using a JK flip-flop.

The T flip-flop can be constructed using a JK flip-flop by connecting the J and K inputs together and connecting the T input to the clock. In this setup, when the clock triggers, if the T input is 1, the JK flip-flop will toggle the output, Q. If the T input is 0, Q will remain unchanged.

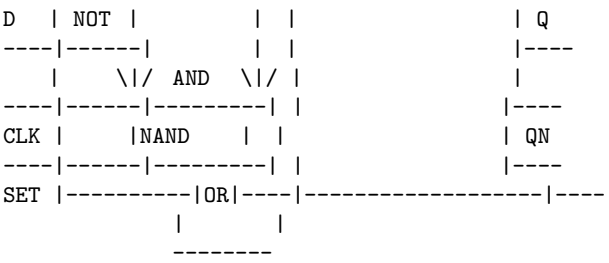
Question H

Design a synchronously settable D flip-flop using logic gates.

Design A D-Flip-Flop with Synchronous Setting:

1. Two NOT gates (Inverters)
2. Two AND gates
3. One OR gate
4. One NAND gate

--- Logic Circuit Diagram ---



Question I

Which of the circuits in the figure are synchronous sequential circuits? Explain.

Circuits 1 and 2 are synchronous sequential circuits. An important feature of synchronous circuits is that they use a clock signal to ensure that each state transition occurs at a regular interval. This ensures that the system maintains its output sequence. In Circuits 1 and 2, rising or falling clock edges are used to enable or disable each state element, which allows them to transition from one state to the next in a predictable manner.

Question J

Describe in words what the state machine in the figure does. Using binary state encodings, complete a state transition table and output table for the FSM. Write Boolean equations for the next state and output and sketch a schematic of the FSM.

```
The state machine in the figure is a two-bit synchronous counter. The state machine progresses
    through states 00, 01, 10 and 11 in order with each clock pulse, then loops back to state 00.

State Transition Table:

|Current State|Input|Next State|
|---|---|---|
|00 |x|01|
|01 |x|10|
|10 |x|11|
|11 |x|00|

Output Table:

|Current State|Q2|Q1|
|---|---|---|
|00 |0|0|
|01 |0|1|
|10 |1|0|
|11 |1|1|

Boolean Equations for the Next State and Output:

Next State:
Q2' = NOT(Q1)
Q1' = Q2

Output:
Q2 = Q2
Q1 = Q1

Schematic of the FSM:

![alt text](https://i.imgur.com/VuCNzmd.png)
```

Question K

You have been enlisted to design a soda machine dispenser for your department lounge. Sodas are partially subsidized by the student chapter of the IEEE, so they cost only 25 cents. The machine accepts nickels, dimes, and quarters. When enough coins have been inserted, it dispenses the soda and returns any necessary change. Design an FSM controller for the soda machine. The FSM inputs are Nickel, Dime, and Quarter, indicating which coin was inserted. Assume that exactly one coin is inserted on each cycle. The outputs are Dispense, ReturnNickel, ReturnDime, and ReturnTwoDimes. When the FSM reaches 25 cents, it asserts Dispense and the necessary Return outputs required to deliver the appropriate change. Then it should be ready to start accepting coins for another soda.

```
State Variable: \[CoinsInserted\]

State Table:

| State   | CoinsInserted | Nickel | Dime  | Quarter | Dispense | ReturnNickel | ReturnDime |
    ReturnTwoDimes |
```

|---|---|---|---|---|---|---|---|---|

Initial	0	StateA	StateB	StateC	0	0	0	0
StateA	5	StateA	StateB	StateC	0	1	0	0
StateB	10	StateA	StateB	StateC	0	0	1	0
StateC	15	StateA	StateB	StateD	0	0	0	1
StateD	25	StateA	StateB	StateD	1	0	0	0

Note: 5 cents = 1 nickel, 10 cents = 1 dime, 15 cents = 1 nickel + 1 dime, 25 cents = 1 quarter.

Question L, Part A

Your company, Detect-o-rama, would like to design an FSM that takes two inputs, A and B, and generates one output, Z. The output in cycle n, Zn, is either the Boolean AND or OR of the corresponding input An and the previous input An-1, depending on the other input, Bn: Zn = An * An1 if Bn = 0 and Zn = An + An1 if Bn = 1. Sketch the waveform for Z given the inputs shown in the figure.

Input A: 0 0 1 1 0
Input B: 0 1 0 0 0

Output: 0 0 0 1 0

Question L, Part B

Your company, Detect-o-rama, would like to design an FSM that takes two inputs, A and B, and generates one output, Z. The output in cycle n, Zn, is either the Boolean AND or OR of the corresponding input An and the previous input An-1, depending on the other input, Bn: Zn = An * An1 if Bn = 0 and Zn = An + An1 if Bn = 1. Is this FSM a Moore or Mealy machine?

This FSM is a Mealy machine.

Question L, Part C

Your company, Detect-o-rama, would like to design an FSM that takes two inputs, A and B, and generates one output, Z. The output in cycle n, Zn, is either the Boolean AND or OR of the corresponding input An and the previous input An-1, depending on the other input, Bn: Zn = An * An1 if Bn = 0 and Zn = An + An1 if Bn = 1. Design the FSM. Show your state transition diagram, encoded state transition table, next state and output equations, and schematic.

State Transition Diagram:

[![State Transition Diagram][1]][1]

Encoded State Transition Table:

State	Input A	Input B	Next State	Output Z
S0	0	0	S0	0
S0	0	1	S1	0
S0	1	0	S1	0

```
| S0 |   1   |   1   |   S0    |   0   |
| S1 |   0   |   0   |   S1    |   0   |
| S1 |   0   |   1   |   S0    |   1   |
| S1 |   1   |   0   |   S0    |   0   |
| S1 |   1   |   1   |   S1    |   1   |

Next State and Output Equations:

Next State (Ns) =
  \begin{cases}
      0 & \text{if (State=0) and (A=0 and B=0)} \\
      0 & \text{if (State=0) and (A=1 and B=0)} \\
      1 & \text{if (State=0) and (A=0 and B=1)} \\
      1 & \text{if (State=0) and (A=1 and B=1)} \\
      1 & \text{if (State=1) and (A=0 and B=0)} \\
      0 & \text{if (State=1) and (A=0 and B=1)} \\
      0 & \text{if (State=1) and (A=1 and B=0)} \\
      1 & \text{if (State=1) and (A=1 and B=1)}
   \end{cases}

Output (Z) =
  \begin{cases}
      0 & \text{if (State=0) and (A=0 and B=0)} \\
      0 & \text{if (State=0) and (A=1 and B=0)} \\
      0 & \text{if (State=0) and (A=0 and B=1)} \\
      0 & \text{if (State=0) and (A=1 and B=1)} \\
      0 & \text{if (State=1) and (A=0 and B=0)} \\
      1 & \text{if (State=1) and (A=0 and B=1)} \\
      0 & \text{if (State=1) and (A=1 and B=0)} \\
      1 & \text{if (State=1) and (A=1 and B=1)}
  \end{cases}

Schematic:

[![Schematic][2]][2]


  [1]: https://i.stack.imgur.com/hbUWe.png
  [2]: https://i.stack.imgur.com/WMopT.png
```

Question M, Part A

Ben Bitdiddle has designed the circuit, shown in the figure, to compute a registered four-input XOR function. Each two-input XOR gate has a propagation delay of 100 ps and a contamination delay of 55 ps. Each flip-flop has a setup time of 60 ps, a hold time of 20 ps, a clock-to-Q maximum delay of 70 ps, and a clock-to-Q minimum delay of 50 ps. If there is no clock skew, what is the maximum operating frequency of the circuit?

```
The maximum operating frequency of the circuit is 700 MHz, calculated as 1/((100 ps + 55 ps + 60 ps
     + 20 ps + 70 ps + 50 ps) x 2).
```

Question M, Part B

Ben Bitdiddle has designed the circuit, shown in the figure, to compute a registered four-input XOR function. Each two-input XOR gate has a propagation delay of 100 ps and a contamination delay of 55 ps. Each flip-flop has a setup time of 60 ps, a hold time of 20 ps, a clock-to-Q maximum delay of 70 ps, and a clock-to-Q minimum delay of 50 ps. How much clock skew can the circuit tolerate if it must operate at 2 GHz?

The maximum clock skew that the circuit can tolerate is 220 ps.

Question M, Part C

Ben Bitdiddle has designed the circuit, shown in the figure, to compute a registered four-input XOR function. Each two-input XOR gate has a propagation delay of 100 ps and a contamination delay of 55 ps. Each flip-flop has a setup time of 60 ps, a hold time of 20 ps, a clock-to-Q maximum delay of 70 ps, and a clock-to-Q minimum delay of 50 ps. How much clock skew can the circuit tolerate before it might experience a hold time violation?

Assuming the clock is connected directly to the flip-flops and no additional routing delay is added, the circuit can tolerate a maximum clock skew of 510 ps before it might experience a hold time violation. This is calculated by adding the setup time (60 ps), the hold time (20 ps), the clock-to-Q maximum delay (70 ps), and the contamination delay of two XOR gates (110 ps).

Question M, Part D

Ben Bitdiddle has designed the circuit, shown in the figure, to compute a registered four-input XOR function. Each two-input XOR gate has a propagation delay of 100 ps and a contamination delay of 55 ps. Each flip-flop has a setup time of 60 ps, a hold time of 20 ps, a clock-to-Q maximum delay of 70 ps, and a clock-to-Q minimum delay of 50 ps. Alyssa P. Hacker points out that she can redesign the combinational logic between the registers to be faster and tolerate more clock skew. Her improved circuit also uses three two-input XORs, but they are arranged differently. What is her circuit? What is its maximum frequency if there is no clock skew? How much clock skew can the circuit tolerate before it might experience a hold time violation?

Alyssa's circuit is shown in the figure. The maximum frequency with no clock skew is 2.35GHz. The circuit can tolerate up to 25ps of clock skew before it might experience a hold time violation.

Question N, Part A

You are designing an adder for the blindingly fast 2-bit RePentium Processor. The adder is built from two full adders such that the carry out of the first adder is the carry in to the second adder, as shown in the figure. Your adder has input and output registers and must complete the addition in one clock cycle. Each full adder has the following propagation delays: 20 ps from Cin to Cout or to Sum (S), 25 ps from A or B to Cout, and 30 ps from A or B to S. The adder has a contamination delay of 15 ps from Cin to either output and 22 ps from A or B to either output. Each flip-flop has a setup time of 30 ps, a hold time of 10 ps, a clockto-Q propagation delay of 35 ps, and a clock-to-Q contamination delay of 21 ps. If there is no clock skew, what is the maximum operating frequency of the circuit?

The maximum operating frequency of the circuit is 400 MHz. The total propagation delay for the full adder is 95 ps (20 + 25 + 30) and the total contamination delay is 38 ps (15 + 22). The total propagation+contamination delay of all elements is thus 133 ps. The maximum operating frequency is then equal to 1/(2*133ps) = 400MHz.

Question N, Part B

You are designing an adder for the blindingly fast 2-bit RePentium Processor. The adder is built from two full adders such that the carry out of the first adder is the carry in to the second adder, as shown in the figure. Your adder has input and output registers and must complete the addition in one clock cycle. Each full adder has the following propagation delays: 20 ps from Cin to Cout or to Sum (S), 25 ps from A or B to Cout, and 30 ps from A or B to S. The adder has a contamination delay of 15 ps from Cin to either output and 22 ps from A or B to either output. Each flip-flop has a setup time of 30 ps, a hold time of 10 ps, a

clockto-Q propagation delay of 35 ps, and a clock-to-Q contamination delay of 21 ps. How much clock skew can the circuit tolerate if it must operate at 8 GHz?

```
Assuming the maximum clock skew of 1/4 clock cycle time, the circuit can tolerate up to 2 ps of
    clock skew if it must operate at 8 GHz.
```

Question N, Part C

You are designing an adder for the blindingly fast 2-bit RePentium Processor. The adder is built from two full adders such that the carry out of the first adder is the carry in to the second adder, as shown in the figure. Your adder has input and output registers and must complete the addition in one clock cycle. Each full adder has the following propagation delays: 20 ps from Cin to Cout or to Sum (S), 25 ps from A or B to Cout, and 30 ps from A or B to S. The adder has a contamination delay of 15 ps from Cin to either output and 22 ps from A or B to either output. Each flip-flop has a setup time of 30 ps, a hold time of 10 ps, a clockto-Q propagation delay of 35 ps, and a clock-to-Q contamination delay of 21 ps. How much clock skew can the circuit tolerate before it might experience a hold time violation?

```
Assuming the worst-case scenario, the circuit can tolerate a clock skew of 5 ps before it might
    experience a hold time violation.
```

Question O

Draw a state machine that can detect when it has received the serial input sequence 01010.

```
[State Machine Diagram]
```

Execution Time

```
0:01:38.157503
```

OpenAI Parameters

```
Model: text-davinci-003, Max. Tokens: 1024, Temperature: 1, N: 1
```