Q1. Clock skew

1. Given the circuit in figure 1, each 2-input or gate has a propagation delay of 60 ps and a contamination delay of 40 ps. Each flip-flop has a setup time of 30 ps, a hold time of 20 ps, a clock-to-Q propagation delay of 80 ps, and a clock-to-Q contamination delay of 40 ps.

a. Find the critical path of the circuit
b. If there is no clock skew, What is the maximum frequency of the circuit?
c. Is there any hold time violation in the circuit? If there is not, how much clock skew can the circuit tolerate? If there is, how would you fix the circuit?
Q2. HLSM design
As a fruit-lover and an engineering student, your constant craving for fresh orange juice has prompted you to purchase an exquisite blender and an infinite supply of oranges. However, as the final week approaching, peeling the oranges and tossing them into the blender becomes a tedious task that you wish you can automate. Luckily, your foresightedness has allowed you to invent an orange-peeling-and-tossing machine beforehand. Now all you need is to design a circuit between the fancy machine and the blender to automate the rest of the process:

- Your circuit will have an orange counter which increments by 1 every time the fancy machine sends a signal that indicates that it has peeled and tossed an orange into the blender.
- When the counter reaches 8, a stop signal will be sent out which will stop the fancy machine from peeling and tossing more oranges into the blender.
- The stop signal does not turn off until the orange counter receives a reset signal when you pour the juice out of the blender.
- You may assume that you will not reset the counter until a cup of juice is produced, and you will not reset the counter when the fancy machine is tossing an orange (the signal from the fancy machine is high)

(a) Draw the HLSM to represent the system. Make sure to define all the inputs, outputs, and variables.
(b) Draw the datapath for this system
(c) Connect control and datapath
(d) Draw the FSM for the control
Q3. HLSM from C Code
(a) Draw an HLSM that models the following C code.
- Since we are using C, recall that both inputs and outputs can be passed by reference.
- Assume all input/output have enough allocated space and pointers are valid such that
  the code does not access invalid values or segfault.
- In HLSM drawing, clearly mark both the inputs that trigger transitions and outputs for
  states where applicable. If states do not produce outputs, do not force an output. Your
determination of which values are inputs vs outputs is an important first step here.
(b) Draw the corresponding datapath and control unit for HLSM

```c
void max_str(uint * count, bool * done) {
    uint i = 0;
    *done = 0;

    while(!init) {
        continue;
    }

    while(i < *count) {
        i++;
    }
    *done = 1;
}
```
Q4. System Implementation

Control Unit for the above system:

For the system given above, answer the following questions:

a. Show the excitation table for the control FSM
b. Design control using D-FFs and gates
   c. Assuming all gates have 50 ps propagation delay, what is the critical path of the circuit and how much time does it take?
   d. Calculate the maximum clock frequency of the circuit.