FoS Exam Sample Questions: Computer Engineering (A)

Question 1: This question has 4 parts (a)-(d)

(a) Consider the following sequential building block.
   (i) What type of sequential block it is (Synchronous or Asynchronous)?
   (ii) Assume that $S = R = 1$. What happens to the outputs ($Q$ and $\overline{Q}$) when CLK transitions from a 1 to 0? Briefly explain your answer.

   ![Synchronous Block Diagram](image1)

(b) Consider the following inverter chain.
   (i) Explain how the circuit shown below works. If the propagation delay of each inverter is between 4ns and 10ns, determine the output frequency range of the oscillation of the circuit.
   (ii) What happens to the frequency of the oscillation if we change the number of inverters to four.

   ![Inverter Chain](image2)

(c) Consider the following circuit. Determine the frequency of the circuit in which with a presence of the SET event with the pulse width of 15ps, the circuit continues to work correctly 90% of the time (Imagine the gates have a delay with an average of 20ps with a standard deviation of 4).

   ![SET Circuit](image3)

(d) Consider the following FSM. Try to implement it by using memory blocks 16*4 (16 words of 4bit length). The labels in the diagram are (input/output)
Question 2: This question has 6 parts (a)-(f).

Assume a 5-staged pipelined MIPS implementation and consider the following instruction sequence:

```
addi $s0, $s0, 20
lw  $s1, 0($s0)
add $s2, $s0, $s1
add $s3, $s3, $s0
add $s3, $s1, $s4
```

(a) Find all the data dependencies and their types.
(b) Assuming the sequence of instructions is executed correctly, that the initial values of $s0, $s1, $s2, $s3, and $s4 are 1000, 200, 50, 7, and 9 respectively and that memory locations 1000 and 1020 contains the values 300 and 150 respectively, what will be the final values of $s0, $s1, $s2, and $s3?
(c) List the hazards assuming there is neither forwarding nor hazard detection units. What will be the final values of $s0, $s1, $s2, and $s3 in this case?
(d) Add nop instructions to eliminate the hazards in the previous case.
(e) Assume there is full forwarding. Indicate hazards and add nop instructions to eliminate them.
(f) Assuming a clock period of 100ps, what is the total time to execute this instruction sequence correctly without forwarding? What is the total time in case of full forwarding? What is the speed-up achieved by adding full forwarding? Assume that the clock period is increased by 10% in case of full forwarding.
Question 3: This question has 3 parts (a)-(c).

(a) By using labeled diagram explain the Harvard and von-Neumann processors. Indicate where the data arrays and the source code are stored.

(b) Will MIPS be categorized as Harvard or von-Neumann processor? Explain.

(c) Consider a pipelined MIPS architecture. Explain the possibility of executing an IIR filter on MIPS architecture.

Note: IIR filter equation is like

$$y(n) = \sum_{k=0}^{N} a(k)x(n - k) + \sum_{j=0}^{p} b(j)y(n - j)$$

Where $a(k)$ and $b(j)$ are filter coefficient and $x(n)$ is the input signal and $y(n)$ is the output.
Question 4: This question has two parts (a)-(b)

The following circuit is a D-latch, with clock signal represented by C. It is alleged that this circuit is not worth fabrication because it produces glitches.

(a) Identify the type(s) and condition(s) for hazard(s).
(b) Mask the hazard(s).
1. **(C++ and Pipeline processors)** Explain the following questions:
   (a) Explain how C++ works from source code to executable file.

   (b) What is the difference between compiler, Assembler and linker?

   (c) Discuss and calculate the **execution time difference** of the following assembly code running on a 2.5 GHz processor designed with and without a 5 stages pipeline (IF, ID, EX, M and W).

   ```
   // ADD TWO INTEGER ARRAYS
   LW R4, # 400
   L1:
   LW R1, 0 (R4) ; --Load first operand
   LW R2, 400 (R4) ; -- Load second operand
   ADDI R3, R1, R2 ; --Add operands
   SW R3, 0 (R4) ; --Store result
   SUB R4, R4, #4 ; --Calculate address of next element
   BNEZ R4, L1 ; --Loop if (R4) != 0
   ```

   (d) What kind of hazards exist in the given code? Provide an efficient technique to solve these hazards.

2. **SPI:** Assume you want to send data between a microcontroller and small peripheral device for example a shift register. One problem is that the shift registers and microcontroller use separate clock sources. So, there is no guarantee that both sides can synchronize data transferred and it can lead to data lost.

   (a) According to the above scenario, propose an **asynchronous** solution to communicate between peripheral device and microcontroller.

   (b) Try to provide a **synchronous** solution and determine which ports do we need to have a synchronous connection.

   (c) If you want to develop system from one-way communication to two-way communication, meaning that sending data back in the opposite direction how you will do that?
3. **(VHDL and FPGA)** Push Button always got the mechanical property of bouncing state at micro sec. The below figure shows the output results, when the Push Button is pressed. When you pull down the push button from high to low state. It bounces back to high and low few times before it settles at proper output.

(a) In order to avoid such bouncing state, propose a logic circuit. What this logic circuit is called? Draw the circuit and time diagram and show that your circuit can solve bouncing problem.

(b) Complete the following VHDL code for implementing the proposed logic circuits.

```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;

ENTITY Entity IS
PORT(
    clk : IN STD_LOGIC;
    button : IN STD_LOGIC;
    result : OUT STD_LOGIC);
END Entity;
ARCHITECTURE logic OF Entity IS
Begin

END logic;
```

(c) Consider implementing the circuit on a FPGA. Create a sketch of such implementation using the following CLB. Each CLB has two LUT with two D-flip flops as shown in below. Determine how many CLB do you need to implement your circuit on FPGA.
4. **(Combinational and Sequential Circuits, FSM)** The figure below shows a schematic of Serial Adder. The serial adder performs a binary addition bit by bit. It includes input shift register (SREG1 and SREG2), one FSM Adder (Full-Adder with D-Flipflop) and one output shift register where contains the summation of initial values after four clocks. Each input register has a shift enable signal (SEN), load signal (LD), clock and reset signals. When SEN=1 and the clock is pulsed the least significant bit (LSB) of register is entered the connection bus between SREGs and Adder FSM, and the registers are shifted right one position.

![Adder FSM Schematic](image)

(a) Design the Adder FSM circuit for above schematic. It includes full adder and a flip flop to store carry to be used at the next clock cycle. Draw the **Moore state diagram** and state table for the serial Adder FSM.

(b) Design the controller (based on Mealy FSM) so that after receiving a start signal, N, load input data into SREG1 and SREG2 and then it will output SEN=1. Then, after four clocks the valid signal should be one that means that the data SREG3 has a valid summation data. **Note:** during summation process the adder is not able to load new value to both input shift registers.

(c) Implement the controller circuits for controlling the **valid** signal by using D-flip flop.
(d) The designers decided to remove SREG3 to use resource sharing techniques for reducing hardware usage. We consider SREG1 as accumulator, where after four shifts its content is replaced with the sum of initial number. They also need to recover the value of SREG2 after four clocks. Determine necessary changes in the given serial adder to meet these requirements.
1- Answer the followings:
   a. Find a minimal Boolean equation for the function with the following table:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

   b. Design a combinational circuit for the function?

   c. Explain if the circuit has a hazard and modify it such that the hazard is fixed.

2- Design an edge detector circuit. The output should go HIGH for one cycle after the input makes a 0 → 1 transition.
   a. Give the state transition diagram.
   b. Specify its table.
   c. Design the FSM.

3- Perform the following analyses.
   a. Perform time analysis for the following circuit if flip-flops have a clock-to-Q contamination delay of 30 ps and a propagation delay of 80 ps. They have a setup time of 50 ps and a hold time of 60 ps. Each logic gate has a propagation delay of 40 ps and a contamination delay of 25 ps.
b. Give an expression for the following logical function.

4. A pipeline system has five stages: IF, ID, EX, M and W, and each stage has one clock cycle. All memory references are hit in cache. Assuming the following program should be processed:

```
// ADD TWO INTEGER ARRAYS
LW    R4   # 400
L1:   LW    R1,  0 (R4) ; Load first operand
      LW    R2, 400 (R4) ; Load second operand
      ADDI   R3, R1, R2 ; Add operands
      SW     R3,  0 (R4) ; Store result
      SUB    R4, R4, #4  ; Calculate address of next element
      BNEZ   R4, L1      ; Loop if (R4) != 0
```

a. Compute how many clock cycles is required to execute this code segment on the regular (nonpipelined) architecture.
b. Compute how many clock cycles is required to execute this code segment on the simple pipeline without forwarding or bypassing when result of the branch instruction (new PC content) is available after W stage.