EE214 Digital Circuits Laboratory

Wadhwani Electronics Laboratory
Electrical Engineering IIT Bombay

Total duration: 2 hr 40 min

Max marks: 40 (scaled to 30) | END-SEMESTER EXAM - Part B | November 8, 2023

1. **Objective**
   Design a 6-bit **Fibonacci sequence generator circuit**. Only structural modelling is allowed. **Note: FSM based design will not be awarded any marks**.

2. **Theory**
   The Fibonacci sequence is a sequence in which each number is the sum of the two preceding numbers. The sequence starts from 1. The first few values of the sequence are: 1, 1, 2, 3, 5,...

3. **Design** **[10 marks]**

   - Design the Fibonacci sequence generator circuit which has two inputs: **Clk** and **Reset** (active high), and a 6-bit binary output **Seq_Out** representing Fibonnaci series.
   - Circuit implementation will require a minimum of two 6-bit registers, and one 6-bit adder. Design a Parallel-in Parallel-out register of 6-bits using D-flip-flops (which you have designed in practice problem). It should have both set and reset functionality. Make sure to keep the **Enable** input of the flip-flops **HIGH** all the time.
   - Since 50 MHz clock is too fast to observe the sequence on board using LEDs, make a clock divider which provides a 1 Hz clock output named **"Clk_Out"**. Use **"Clk_Out"** as the clock input to the Fibonacci sequence generator.
   - Refer the below pseudo code to understand the working of the circuit:

```
let clock, reset be the two inputs to the circuit
 while(1) //Generator circuit will keep working endlessly
{
 if reset = 1
    { intialize reg_a = 1, reg_b = 0 }
 else
    {
    c = reg_a + reg_b //This operation does not depend on clock
    when positive edge of clock arrives
        {
            reg_b = reg_a
            reg_a = c
        }
    }
}
```

   - Fill in the below blank spaces based on the above pseudo code. **[2 marks]**

No. of clock cycles	reg_a	reg_b	c
0	1	0	1
1	1	1	2
2	-	-	-
3	-	-	-
4	-	-	-
5	-	-	-
6	-	-	-

   - Figure out which register to take the output from, to obtain the correct sequence. **[2 marks]**
   - Draw block-level design and label each wire properly. Briefly explain the internal blocks used in designing the sequence. **[6 marks]**

4. **Implementation** **[20 marks]**

- **PART-A** **[6 marks]**
  - Write VHDL description of 6-bit PIPO (Parallel-in, Parallel-out) register as mentioned in **"Design"** section above.
  - Use entity declaration as follows:

```
entity Register6bit is
        port(Clk, Reset : in  std_logic;
             data_in    : in  std_logic_vector(5 downto 0);
             data_out   : out std_logic_vector(5 downto 0));
 end entity;
```

  - Verify the functionality of this block with the testbench given in "Testbench_for_Register".

- **PART-B** **[8 marks]**
  - Write VHDL description for Fibonacci sequence generator circuit.
  - Use entity declaration as follows:

```
entity FibGenerator is
        port(Clk, Reset : std_logic;
             Seq_out    : std_logic_vector(5 downto 0));
 end entity;
```

  - Perform simulation using the provided testbench named "Testbench_for_Random_Sequence". Below is the expected output waveform.
  - Use the Clock_Divider module given to you. **Modify the Count value to 25000 while performing simulations**.

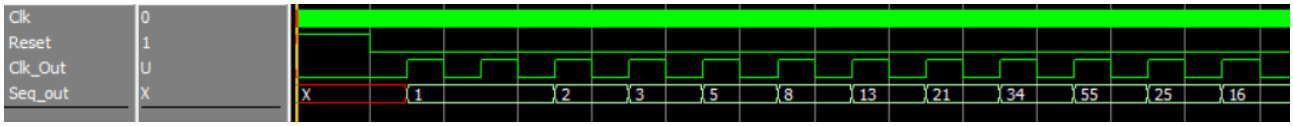

Figure 1: Input-Output Waveforms

  - Referring to the above figure, after 55 the next Fibonacci series number should be 89. But the output shows 25. Briefly explain the reason behind this apparent error.
  - **NOTE: You may need to modify the set and reset connections for some flipflops being used inside register to incorporate it in this design**

- **PART-C** **[6 marks]**

  In this part, we will generate the Fibonacci sequence which will repeat itself. Modify the circuit such that it will reset the registers as soon as the wrong sequence is detected.
  - Write VHDL description of the required circuit by modifying PART-A circuit.
  - Perform simulation using the provided testbench named "Testbench_for_Fibonacci_Sequence". Below is the expected output waveform.
  - Use the Clock_Divider module given to you. **Modify the Count value to 25000 while performing simulations**.

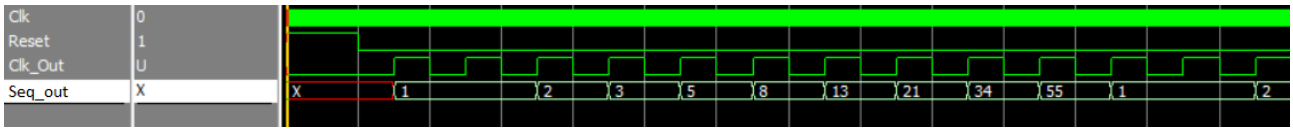

Figure 2: Input-Output Waveforms

5. **On-Board implementation** **[10 marks]**

- Simulate using the testbench and then perform Pin-Planning step. Pin plan Reset to **SW1**, **Clk** to the on-board 50 MHz clock, and 6-bit output from LED1 to LED6 (LSB of output connected to LED1 and MSB to LED6).
- **NOTE: Change the count value inside Clock_Divider for 1Hz**
- Pin mapping of on-board peripherals:

Switch	FPGA Pin no.	LED	FPGA Pin no.
SW 8	47	LED 8	60
SW 7	46	LED 7	59
SW 6	45	LED 6	58
SW 5	44	LED 5	57
SW 4	43	LED 4	56
SW 3	41	LED 3	54
SW 2	39	LED 2	52
SW 1	38	LED 1	50

Figure 3: Pin Planning for Switches and LEDs

Clock Source Frequency	FPGA Pin no.
1 Hz CLK	55
50 MHz CLK	26
Ext CLK	27
10 MHz CLK	29

Figure 4: Pin Mapping for Clock Sources