Lab 3
Sequential Logic for Synthesis. FPGA Design Flow.

Part 1

Task 1

Develop a VHDL description of a Debouncer specified below.

The following diagram shows the interface of the Debouncer.

The following diagram shows the inside of the Debouncer unit.
Task 2

Develop a VHDL description of a Rising Edge Detector specified below.

The following diagram shows the Rising Edge Detector circuit.

```
data_i
            ↘
            ↗
clk_i
```

Task 3

Develop a VHDL description of a Counter specified below.

The following diagram shows the interface of the Counter.
The following diagram shows the inside of the Counter unit.

![Diagram of the Counter unit](image)

**Task 4**

Develop a VHDL description of a Clock Divider circuit specified below. The output frequency should be 1 kHz. This clock will be used to drive the seven segment display unit.

The following diagram shows the interface of the Clock Divider.

![Diagram of the Clock Divider](image)
The following diagram shows the inside of the Clock Divider unit.

![Clock Divider Diagram]

**Task 5 (40%)**

Describe in VHDL, verify, synthesize, implement, and test experimentally a circuit shown in the diagram below:

![7-Seg Display Unit Diagram]

Make sure that your debouncer operates correctly.
Task 6

Develop a VHDL description of Pseudo-Random Number Generator (PRNG) specified below. The following diagram shows the interface of the PRNG.

The following diagram shows the inside of the PRNG unit.
The inputs of XOR gates are specified in the diagram below:

The shift registers holding the internal state of the circuit should be initialized using one of the described below methods.

**Option 1 (required):**
Initialization to ALL ONES, using the signal SET common to all shift registers (connected to rst_i).

**Option 2 (required):**
Initialization to ALL ONES by shifting '1' to all shift registers for 6 clock cycles after reset.

**Option 3: (bonus 20%):**
Initialization to arbitrary sequence of 16 bits, by shifting in internal state serially, using the special input SIN (not shown in the interface above), one bit per clock cycle. The circuit should also have an input MODE, which allows switching between initialization mode and random number generation mode.

Please implement at least Options 1 and 2. Implement Option 3 for extra credit.

Compare all implemented versions of the circuit in terms of:
- resource utilization (in CLB slices)
- minimum clock period (in ns)
- number of clock cycles required for initialization (assuming that initialization is performed at full speed).
**Task 7 (60%)**

Combine the previous tasks to make the top unit described below. You may use the code developed in above tasks, the code that has been provided by the instructors, and any other circuitry necessary.

Verify, synthesize, implement, and test experimentally the entire circuit.

**Part 2**

**Task 8 (bonus)**

Develop a VHDL description of a Debouncer specified below. It should be written using a finite state machine with an enumerated type.

The following diagram shows the interface of the Debouncer.
The following diagram shows the state machine description of the debouncing circuit.

Task 9 (bonus)

Develop a VHDL structural description of a Digital Clock Manager (DCM). This also includes the clock buffers (IBUFG and BUFG), as well as any necessary surrounding logic.

The following diagram shows the clock managing circuit.

The reset should be connected to all of the resets for the rest of the chip and the clk0 should be used instead of the 50MHz clock. The IBUFG and BUFG circuits are shown below.
Buffer the 1 kHz clock using a BUFG. The BUFG component is shown above and explained in the lab lecture slides.

**Task 10 (bonus 30%)**

Combine the previous tasks (and tasks from Part 1) to make the top unit described below. You may use the code developed in above tasks, the code that has been provided by the instructors, and any other circuitry necessary.

Verify, synthesize, implement, and test experimentally the entire circuit.

**For Tasks 5, 7 (Options 1-2 required, Option 3 bonus), and 10 (bonus) perform the following steps:**

1. Synthesize your code using Xilinx XST.
2. Prepare a correct UCF (User Constraints File) file and use it during Implementation phase.
3. Implement your circuit using Xilinx ISE.
4. Check thoroughly all implementation reports. Pay attention to timing, resource usage, and pin allocations.
5. Perform functional, post-synthesis, and timing simulations of your circuit.

6. Perform static timing analysis.

7. Check very carefully your pin allocations listed in the report files, and only if these pin allocations are correct, download your bitstream to the FPGA board.

8. Test the operation of your circuit experimentally using the Digilent Basys II Board.

Deliverables (for Tasks 5, 7 (Options 1-2 required, Option 3 bonus) and 10):

1. All source files used for synthesis and implementation of your circuit.

2. Testbench.


4. All synthesis and implementation report files.

5. Simulation waveforms from the functional, post-synthesis, and timing simulations, proving the correct operation of your circuit.

6. Report file from the static timing analysis.

7. Your own report containing the following major components:
   - Resource utilization.
   - Minimum clock period and maximum clock frequency after synthesis and after implementation.
   - Latency (number of clock cycles) required for initialization.
   - Status of your codes
   - Description of experiments you have performed.

Task 11 (required; tested during demonstration; may affect up to 50% of your score)

Be prepared to demonstrate the operation of all your codes using
   - functional simulation
   - post-synthesis simulation
   - timing simulation
   - experimental testing using the Basys II board.

Please be also ready to demonstrate the entire FPGA design flows: based on Aldec Active-HDL and based on Xilinx ISE.
### Important Dates

<table>
<thead>
<tr>
<th>Event</th>
<th>Monday Section</th>
<th>Tuesday Section</th>
<th>Wednesday Section</th>
<th>Thursday Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hands-on Session and Introduction to the Experiment</td>
<td>02/14/2011</td>
<td>02/15/2011</td>
<td>02/16/2011</td>
<td>02/17/2011</td>
</tr>
<tr>
<td>Demonstration and Deliverables Due</td>
<td>02/28/2011</td>
<td>03/01/2011</td>
<td>03/02/2011</td>
<td>03/03/2011</td>
</tr>
</tbody>
</table>