Lab 7: Latches and Flip-flops

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION,
YOUTH AND SPORTS

The Study of Modern and Developing Engineering BUT
CZ.02.2.69/0.0/0.0/18_056/0013325

Learning objectives

In this laboratory exercise, you will study the differences between a statically controlled latch and flip-flops that are synchronized with a clock signal. In VHDL, combinational and synchronous processes will be used and the difference between asynchronous and synchronous reset will be illustrated.

Preparation tasks (done before the lab at home)

Write characteristic equations and complete truth tables for D, JK, T flip-flops where `q(n)` represents main output value before the clock edge and `q(n+1)` represents value after the clock edge.

$q_{n+1}^{D} =$
$q_{n+1}^{JK} =$
$q_{n+1}^{T} =$

clk	d	q(n)	q(n+1)	Comments
	0	0		
	0	1		
	1			
	1			

clk	j	k	q(n)	q(n+1)	Comments
	0	0	0	0	No change
	0	0	1	1	No change
	0				
	0				
	1				
	1				

clk	j	k	q(n)	q(n+1)	Comments
	1				
	1				

clk	t	q(n)	q(n+1)	Comments
	0	0		
	0	1		
	1			
	1			

Part 1: Synchronize repositories and create a new folder

Run Git Bash (Windows) of Terminal (Linux), navigate to your working directory, and update local repository. Create a new working folder `Labs/07-ffs` for this exercise.

Part 2: VHDL code for D latch

A latch is a level triggered element. Perform the following steps to model the D latch circuit.

1. Create a new Vivado RTL project `flip_flops` in your `Labs/07-ffs` working folder.
2. Create a VHDL source file `d_latch` for D latch circuit.
3. Choose default board: `Nexys A7-50T`.
4. Define entity with `en`, `arst`, `d`, `q`, `q_bar`, use example from lecture *5 Sequential logic circuit* and complete the latch with asynchronous reset.
5. Create a simulation source `tb_d_latch`, set input conditions and run the simulation. Verify the reset and enable functionality.

Part 3: VHDL code for flip-flops

As specified by the teacher, create at least two entities for flip-flop D (with asynchronous reset, with synchronization reset), flip-flop JK (with synchronization reset), or T flip-flop (with synchronization reset). Try to simulate them together in a single testbench with a maximum duration of 200 ns.

Entity	Inputs	Outputs	Description
`d_ff_arst`	`clk`, `arst`, `d`	`q`, `q_bar`	D type flip-flop with an async reset
`d_ff_rst`	`clk`, `rst`, `d`	`q`, `q_bar`	D type flip-flop with a sync reset
`jk_ff_rst`	`clk`, `rst`, `j`, `k`	`q`, `q_bar`	JK type flip-flop with a sync reset
`t_ff_rst`	`clk`, `rst`, `t`	`q`, `q_bar`	T type flip-flop with a sync reset

Synchronize repositories

Use git commands to add, commit, and push all local changes to your remote repository. Check the repository at GitHub web page for changes.

Experiments on your own: Shift register

Use D type flip-flops with synchronous reset and perform the following steps to implement the 4-bit shift register on the Nexys A7 board.

1. Create a new design source `top` in your project.
2. Use **Define Module** dialog and define I/O ports of entity `top` as follows.

Port name	Direction	Type	Description
`BTNU`	in	`std_logic`	Clock emulator
`BTNC`	in	`std_logic`	Synchronous reset
`SW`	in	`std_logic_vector(1 - 1 downto 0)`	Shift register serial input
`LED`	out	`std_logic_vector(4 - 1 downto 0)`	Shift register parallel outputs

3. Use direct instantiation and define an architecture of the top level.

```
----------------------------------------------------------------------------
-- Architecture body for top level
----------------------------------------------------------------------------
architecture Behavioral of top is

    -- Internal signals between flip-flops
    -- WRITE YOUR CODE HERE

begin

    ----------------------------------------------------------------------------
    -- Four instances (copies) of D type FF entity
    d_ff_0 : entity work.d_ff_rst
        port map(
            clk   => BTNU,
            rst   => BTNC,
            -- WRITE YOUR CODE HERE
        );

    d_ff_1 : entity work.d_ff_rst
        port map(
            clk   => BTNU,
            rst   => BTNC,
            -- WRITE YOUR CODE HERE
        );

    -- WRITE YOUR CODE HERE

end architecture Behavioral;
```

4. Create a testbench file `tb_top` and simulate it or create a new constraints XDC file: `nexys-a7-50t` and uncomment used pins according to the entity.

5. Compile the project and download the generated bitstream `flip_flops/flip_flops.runs/impl_1/top.bit` into the FPGA chip.
6. Test the functionality of the shift register by pressing the push buttons and observing LEDs.
7. Use **IMPLEMENTATION > Open Implemented Design > Schematic** to see the generated structure.

Lab assignment

1. Preparation tasks (done before the lab at home). Submit:
    - Characteristic equations and completed tables for D, JK, T flip-flops.
2. D latch. Submit:
    - VHDL code listing of the process `p_d_latch` with asynchronous reset and syntax highlighting,
    - Listing of VHDL reset and stimulus processes from the testbench `tb_d_latch.vhd` file with syntax highlighting and asserts,
    - Screenshot with simulated time waveforms; always display all inputs and outputs. The full functionality of the entity must be verified.
3. Flip-flops. Submit:
    - Listing of VHDL code of architectures of selected flip-flops with syntax highlighting,
    - Listing of VHDL clock, reset and stimulus processes from the testbench file(s) with syntax highlighting and asserts,
    - Screenshot with simulated time waveforms with a maximum duration of 200 ns; always display all inputs and outputs. The full functionality of the entities must be verified.
4. Shift register. Submit:
    - Image of the shift register schematic. The image can be drawn on a computer or by hand. Name all inputs, outputs, components and internal signals.

Prepare all parts of the assignment on a computer (not by hand), insert them in your README file `Digital-electronics-1/Labs/07-ffs/README.md`, export the formated output (not the listing in markdown language) from HTML to PDF, use BUT e-learning web page and submit a single PDF file. The deadline for submitting the task is the day before the next laboratory exercise.