Exercise 1: Traffic light control

G. Kemnitz*, TU Clausthal, Institute of Computer Science

May 22, 2012

Abstract

A state graph for a given text description has to be designed and from it a VHDL description for simulation and synthesis. The designed circuit should be simulated, synthesized, programmed in the FPGA and tested.

1 Target function and the design of the state graph

Describe the following function by a state graph:

> In the initial state pedestrian light is red and cars have green. If a pedestrian presses the button, the light for cars turns one clock later to yellow and after two clocks to red. Two clock periods after the cars have red, the pedestrian light becomes five clock periods green and then red again. The cars still keep red for two clock periods. Then the light for cars turns for one clock to red-yellow and for at least five clock periods to green. If a pedestrian presses the button afterward, the cycle repeats.

Before drawing the state graph it is recommended to count the different states and input values.

The functional description up to now is incomplete. It doesn't describe e.g. what happens, if the control is not in the initial state and a pedestrian presses the button. Add the following functionality to the state graph:

> If a pedestrian presses the button having red and the traffic light control has still not finished its cycle, the control should store the button event and automatically restarts the cycle after reaching the initial state. If a pedestrian presses the button having still green the button event should be ignored.

In case of a malfunction e.g. if one of the red light bulbs is broken, a safety critical system should switch to a safe state, in which no harm happens. Add to the graph:

> In case of a detected malfunction the pedestrian light should switch off immediately (neither red nor green) and the care light should blink yellow. This state can only be left by a re-initialization.

The clock period should be 2s. The pedestrian button should be BTN0. A malfunction should be signalized by pressing BTN1. Re-initialization should be done via BTN3. Don't forget debouncing! The clock for the circuit has to be derived from the 50 MHz input clock. The entity of the traffic light control should be:

*Tel. 05323/727116

```
entity traffic_light_control is
 port(
   clk : in STD_LOGIC;
   reset : in STD_LOGIC;                        -- initialization
   signal_in : in STD_LOGIC;                    -- pedestrian button
   fehler_in : in STD_LOGIC;                    -- button to simulate malfunctions
   pkw : out STD_LOGIC_VECTOR(2 downto 0);      -- car signals (green, yellow and  red)
   fuss : out STD_LOGIC_VECTOR(1 downto 0);     -- pedestrian signals (green, red)
   signal_out : out STD_LOGIC;                  -- LED »pedestrian button pressed«
   zustand : out STD_LOGIC_VECTOR(3 downto 0);-- LEDs to display the state
   fehler_out : out STD_LOGIC                   -- LED to display an error state
 );
end entity;
```

2 Complete circuit and test

Figure 1 shows the whole circuit to be designed. From figure also the entries for the ucf-file should be taken. The daughter board with the traffic light LEDs is supposed to be plugged on connector B1 of the FPGA board.

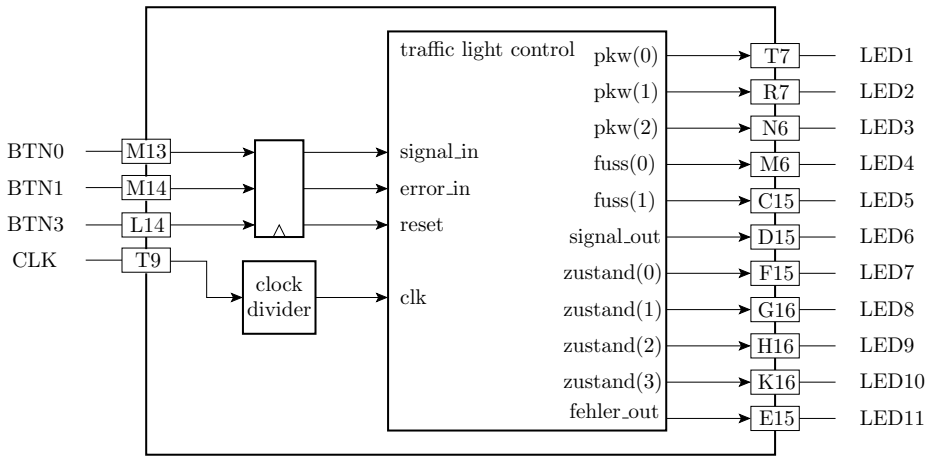

Figure 1: Complete circuit

1. Write a VHDL-description for the traffic light controller as described in the last section.
2. Write a test bench and simulate it. Clock period is 2 s and an example wave form is shown in figure 2.
3. Program the circuit in the FPGA and test it.

3 Check list for the compliance test

1. State graph (on a sheet of paper)
2. Presentable simulation model (see figure 2)
3. Running programmed circuit.

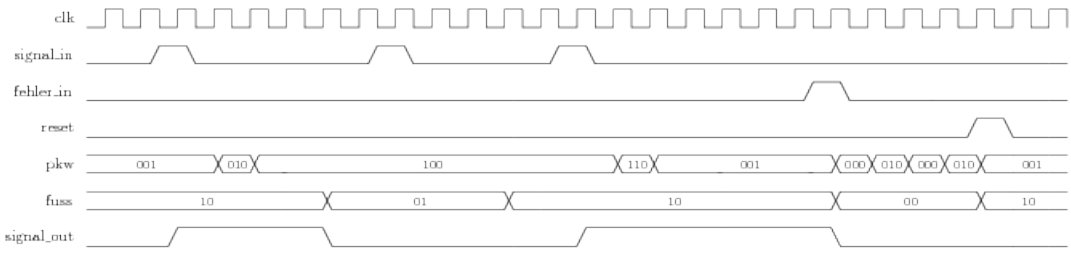

Figure 2: Simulation input