Author Jasim Abbasi		Task Uppgift_ 3a_extra		
Email engr.jasim@gmail.com	Approved		Version	File VHDL 3 extra

TIES AB

VHDL Assignment 3 extra

Embedded system that counts presses from a pressure contact

Jasim Abbasi
2015-01-01

CONTENTS

1 REQUIREMENTS SPECIFICATION

1.1 Functional requirements

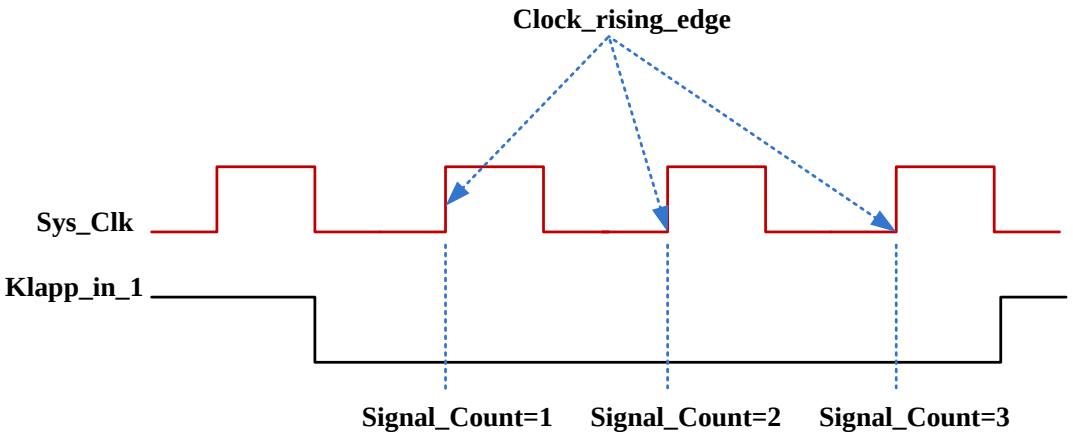
Construct an embedded system that counts presses from a pressure contact. A pressure switch, seven segment and three LEDs should be connected to the FPGA. The LEDs should be binary count up the number of key presses. The task is to verify some values with the simulator and validate the code on the DE2 board.

2 TEST PROTOCOL

Table 1: Test Protocol

Case	Description	OK	Verifying Modelsim	Validation DE2-115
Test 1	Push button is not pressed	The LEDs should not count and 7 segment display 0	OK	OK
Test 2	Push button is pressed	The LEDs are listed with a (binary) 7-segment display no	OK	OK
Test 3	Push button is released	The LEDs and 7 segment should not count	OK	OK
Test 4	Push button is not pressed	The LEDs should not count and 7 segment display 0	OK	OK
Test 5	Reset_n active	The LEDs turn off and 7-segment display 0	OK	OK

3 VERIFICATION



3.1 Result from simulation

HEX0 is used to display the no of key presses on the DE-115 board.

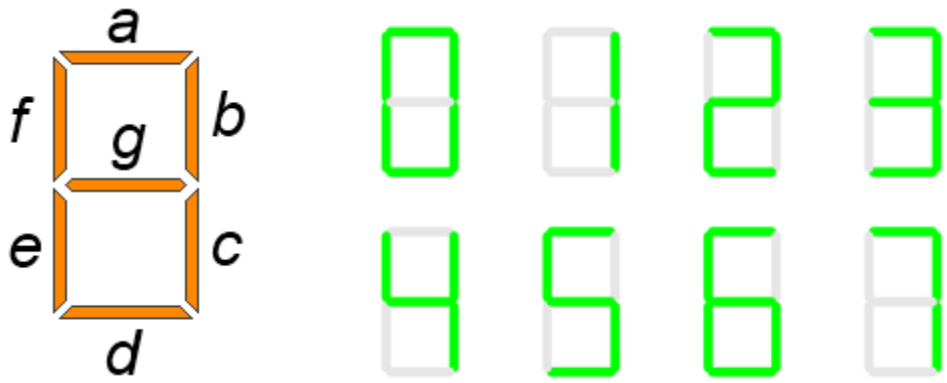


Figure 13: Seven segment display

The values are shown in hexadecimal format in simulation. For example to display the zero on the seven segment 01000000 binary or 40 in hexadecimal form will display in the simulation. The table 2 shows the digits and its corresponding value in binary and hexadecimal form.

Table 2: Seven segment display values for simulation

g	f	e	d	C	B	a	Display	Hexadecimal
1	0	0	0	0	0	0	0	40
1	1	1	1	0	0	1	1	79
0	1	0	0	1	0	0	2	24
0	1	1	0	0	0	0	3	30
0	0	1	1	0	0	1	4	19
0	0	1	0	0	1	0	5	12
0	0	0	0	0	1	0	6	02
1	1	1	1	0	0	0	7	78
0	0	0	0	0	0	0	8	00
0	0	1	1	0	0	0	9	18
0	0	0	1	0	0	0	A	08
0	0	0	0	0	1	1	B	03
1	0	0	0	1	1	0	C	46
0	1	0	0	0	0	1	D	21
0	0	0	0	1	1	0	E	06
0	0	0	1	1	1	0	F	0E
1	1	1	1	1	1	1	nothing	7F

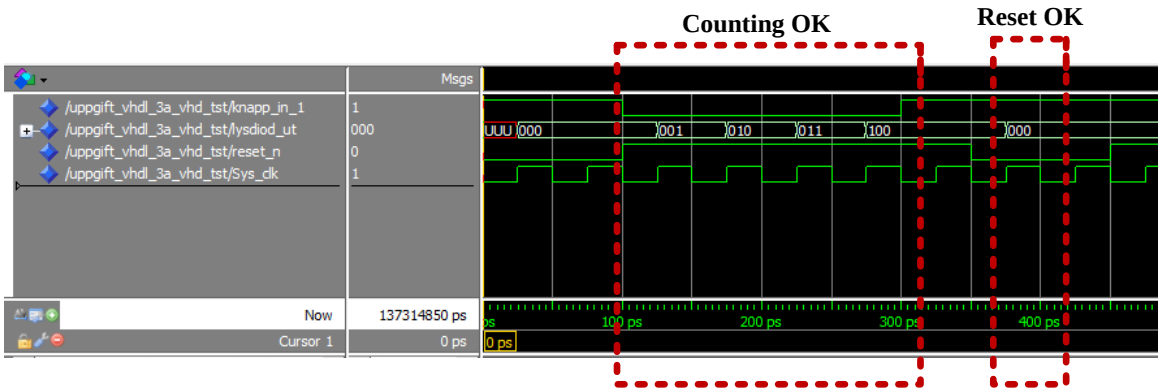


Figure 1: Result without seven segment display

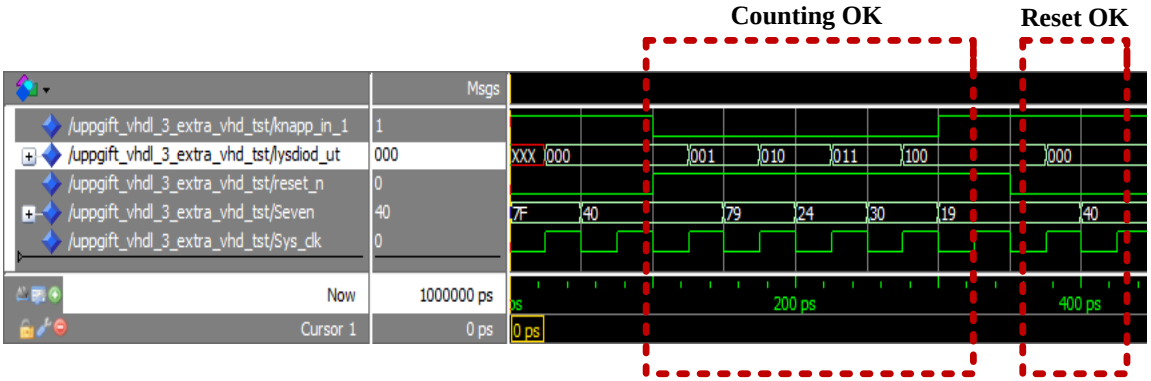


Figure 2: Result with seven segment display

APPENDIX

VHDL Code

```
- Company: TEIS AB

-- Engineer: Jasim Abbasi

-- Create Date: 2014 November 21

-- Design Name: uppgift_vhdl_3b

-- Target Devices: ALTERA Cyclone IV EP4CE115F29C7

-- Tool versions: Quartus v11 and ModelSim

-- I/O Pin Description

-- Sys_clk: PIN_Y2

-- Reset_n: PIN_Y23

-- lysdiod_ut[2..0]:PIN_E24,PIN_E25,PIN_E22

-- knapp_in_1 : PIN_M23

library ieee;

use ieee.std_logic_1164.all;

use ieee.numeric_std.all;

use ieee.std_logic_unsigned.ALL;

entity uppgift_vhdl_3_extra is

port

( Sys_clk, reset_n, knapp_in_1 : in std_logic;

  Seven: out std_logic_vector(6 downto 0);

lysdiod_ut : out std_logic_vector(2 downto 0)

);

End entity uppgift_vhdl_3_extra;

architecture rtl of uppgift_vhdl_3_extra is

-- internal signal declarations
```

```
signal signal_raknarvarde : std_logic_vector(3 downto 0);
signal old_knapp_in_1 : std_logic;
begin
process (sys_clk, reset_n)
begin
if (rising_edge(sys_clk)) then
if reset_n = '0' then -- Reset the counter to 0
    signal_raknarvarde <= "0000";
   elsif knapp_in_1 = '0' then
      signal_raknarvarde <= signal_raknarvarde + '1';
else
       signal_raknarvarde <= signal_raknarvarde;
end if;
end if;
case signal_raknarvarde is
      when "0000" => Seven <= "1000000";----If input is 0000 then display 0
      when "0001" => Seven <= "1111001";----If input is 0001 then display 1
      when "0010" => Seven <= "0100100";----If input is 0010 then display 2
      when "0011" => Seven <= "0110000";----If input is 0010 then display 3
                  when "0100" => Seven <= "0011001";----If input is 0100 then display 4
      when "0101" => Seven <= "0010010";----If input is 0101 then display 5
      when "0110" => Seven <= "0000010";----If input is 0110 then display 6
      when "0111" => Seven <= "1111000";----If input is 0111 then display 7
                  when "1000" => Seven <= "0000000";----If input is 1000 then display 8
      when "1001" => Seven <= "0011000";----If input is 1001 then display 9
      when "1010" => Seven <= "0001000";----If input is 1010 then display A
```

```
      when "1011" => Seven <= "0000011";----If input is 1011 then display B
                  when "1100" => Seven <= "1000110";----If input is 1100 then display C
      when "1101" => Seven <= "0100001";----If input is 1101 then display D
      when "1110" => Seven <= "0000110";----If input is 1110 then display E
      when "1111" => Seven <= "0001110";----If input is 1111 then display F
      when others => Seven <= "1111111";----All other combination display nothing


    end case;
end process;
-- utanför processen
lysdiod_ut <= signal_raknarvarde(2 downto 0);
end rtl;
```

Test Bench

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


ENTITY uppgift_vhdl_3_extra_vhd_tst IS
END uppgift_vhdl_3_extra_vhd_tst;
ARCHITECTURE uppgift_vhdl_3_extra_arch OF uppgift_vhdl_3_extra_vhd_tst IS
-- constants
-- signals
SIGNAL knapp_in_1 : STD_LOGIC;
SIGNAL lysdiod_ut : STD_LOGIC_VECTOR(2 DOWNTO 0);
SIGNAL reset_n : STD_LOGIC;
SIGNAL Seven : STD_LOGIC_VECTOR(6 DOWNTO 0);
SIGNAL Sys_clk : STD_LOGIC;
```

```
COMPONENT uppgift_vhdl_3_extra
	PORT (
	knapp_in_1 : IN STD_LOGIC;
	lysdiod_ut : OUT STD_LOGIC_VECTOR(2 DOWNTO 0);
	reset_n : IN STD_LOGIC;
	Seven : OUT STD_LOGIC_VECTOR(6 DOWNTO 0);
	Sys_clk : IN STD_LOGIC
	);
END COMPONENT;
BEGIN
	i1 : uppgift_vhdl_3_extra
	PORT MAP (
-- list connections between master ports and signals
	knapp_in_1 => knapp_in_1,
	lysdiod_ut => lysdiod_ut,
	reset_n => reset_n,
	Seven => Seven,
	Sys_clk => Sys_clk
	);
	clock: process
begin
-- Clock period is set to 20 ns => 50MHz
sys_clk <= '0';
WAIT FOR 25ps;
sys_clk <= '1';
WAIT FOR 25ps;
```

```
end process clock;
reset: process
begin
-- Reset at start up
reset_n <= '0';
WAIT FOR 100ps;
reset_n <= '1';
WAIT FOR 250ps;
end process reset;
init : PROCESS
-- variable declarations
BEGIN
        -- code that executes only once
--- Test case 1
knapp_in_1 <= '1'; -- Key not pressed
WAIT FOR 100ps;
--- Test case 2
knapp_in_1 <= '0'; -- Key pressed
WAIT FOR 200ps;
-- Test case 3
knapp_in_1 <= '1'; -- Key not pressed
WAIT FOR 100ps;
WAIT;
END PROCESS init;
always : PROCESS
-- optional sensitivity list
```

```
-- (        )
-- variable declarations
BEGIN
        -- code executes for every event on sensitivity list
WAIT;
END PROCESS always;
END uppgift_vhdl_3_extra_arch;
```

DO file

```
onerror {resume}
quietly WaveActivateNextPane {} 0
add wave -noupdate /uppgift_vhdl_3_extra_vhd_tst/knapp_in_1
add wave -noupdate -radix binary /uppgift_vhdl_3_extra_vhd_tst/lysdiod_ut
add wave -noupdate /uppgift_vhdl_3_extra_vhd_tst/reset_n
add wave -noupdate -radix hexadecimal /uppgift_vhdl_3_extra_vhd_tst/Seven
add wave -noupdate /uppgift_vhdl_3_extra_vhd_tst/Sys_clk
TreeUpdate [SetDefaultTree]
WaveRestoreCursors {{Cursor 1} {0 ps} 0}
quietly wave cursor active 0
configure wave -namecolwidth 265
configure wave -valuecolwidth 100
configure wave -justifyvalue left
configure wave -signalnamewidth 0
configure wave -snapdistance 10
configure wave -datasetprefix 0
configure wave -rowmargin 4
configure wave -childrowmargin 2
```

```
configure wave -gridoffset 0
configure wave -gridperiod 1
configure wave -griddelta 40
configure wave -timeline 0
configure wave -timelineunits ps
update
WaveRestoreZoom {0 ps} {893 ps}
```