

T.C. KOCAELİ ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ



SEVEN SEGMENT DISPLAY WITH VHDL

TÜMLEŞİK DEVRELERİNE GİRİŞ

FİNAL RAPORU

JORDI HAKO NINTCHEU

170207095

Bölümü: Elektronik ve Haberleşme Mühendisliği

Danışman: Prof. Dr. Ali TANGEL

KOCAELİ, 2021

CONTENT

INTRODUCTION	3
OBJECTIVES	3
GENERAL OVERVIEW	3
FULL ADDER TRUTH TABLE	4
VHDL CODE	5
TESTBENCH CODE	6
RTL ANALYSIS	8
SIMULATION	8

INTRODUCTION

Due to the increase demand in FPGA's as compared to ASIC for circuit prototyping and testing, I opted to implement a project in order to learn how to design circuits using the vivado suite and the VHDL language.

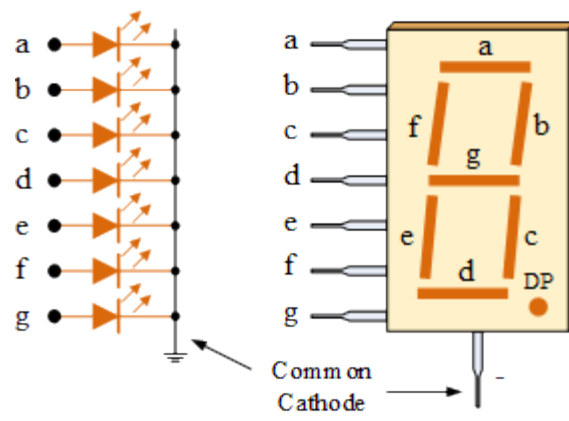
OBJECTIVES

Creating a seven segment display using the VHDL language in Vivado and simulating it using a test bench code.

GENERAL OVERVIEW

The 7-segment display, also written as “seven segment display”, consists of seven LEDs (hence its name) arranged in a rectangular fashion as shown. Each of the seven LEDs is called a segment because when illuminated the segment forms part of a numerical digit (both Decimal and Hex) to be displayed. An additional 8th LED is sometimes used within the same package thus allowing the indication of a decimal point, (DP) when two or more 7-segment displays are connected together to display numbers greater than ten.

Each one of the seven LEDs in the display is given a positional segment with one of its connection pins being brought straight out of the rectangular plastic package. These individually LED pins are labelled from a through to g representing each individual LED. The other LED pins are connected together and wired to form a common pin.



COMMON CATHODE SEVEN SEGMENT DISPLAY TRUTH TABLE

Digital Digits	Individual Segments Illuminated On the Seven Segment Display						
	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	1	1	0	0

VHDL CODE

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity bcd_7segment is
Port ( BCDin : in STD_LOGIC_VECTOR(3 downto 0);
Seven_Segment : out STD_LOGIC_VECTOR(6 downto 0));
end bcd_7segment;

architecture Behavioral of bcd_7segment is

begin

process(BCDin)
begin

case BCDin is

when "0000" =>
```

```

Seven_Segment <= "0000001"; ---0
when "0001" =>
Seven_Segment <= "1001111"; ---1
when "0010" =>
Seven_Segment <= "0010010"; ---2
when "0011" =>
Seven_Segment <= "0000110"; ---3
when "0100" =>
Seven_Segment <= "1001100"; ---4
when "0101" =>
Seven_Segment <= "0100100"; ---5
when "0110" =>
Seven_Segment <= "0100000"; ---6
when "0111" =>
Seven_Segment <= "0001111"; ---7
when "1000" =>
Seven_Segment <= "0000000"; ---8
when "1001" =>
Seven_Segment <= "0000100"; ---9
when others =>
Seven_Segment <= "1111111"; ---null
end case;

end process;

end Behavioral;

```

TESTBENCH CODE

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY tb_bcd_7seg IS
END tb_bcd_7seg;

ARCHITECTURE behavior OF tb_bcd_7seg IS

-- Component Declaration for the Unit Under Test (UUT)

```

```

COMPONENT bcd_7segment
PORT(
BCDin : IN std_logic_vector(3 downto 0);
Seven_Segment : OUT std_logic_vector(6 downto 0)
);
END COMPONENT;

--Inputs
signal BCDin : std_logic_vector(3 downto 0) := (others => '0');

--Outputs
signal Seven_Segment : std_logic_vector(6 downto 0);

BEGIN

-- Instantiate the Unit Under Test (UUT)
uut: bcd_7segment PORT MAP (
BCDin => BCDin,
Seven_Segment => Seven_Segment
);

-- Stimulus process
stim_proc: process
begin

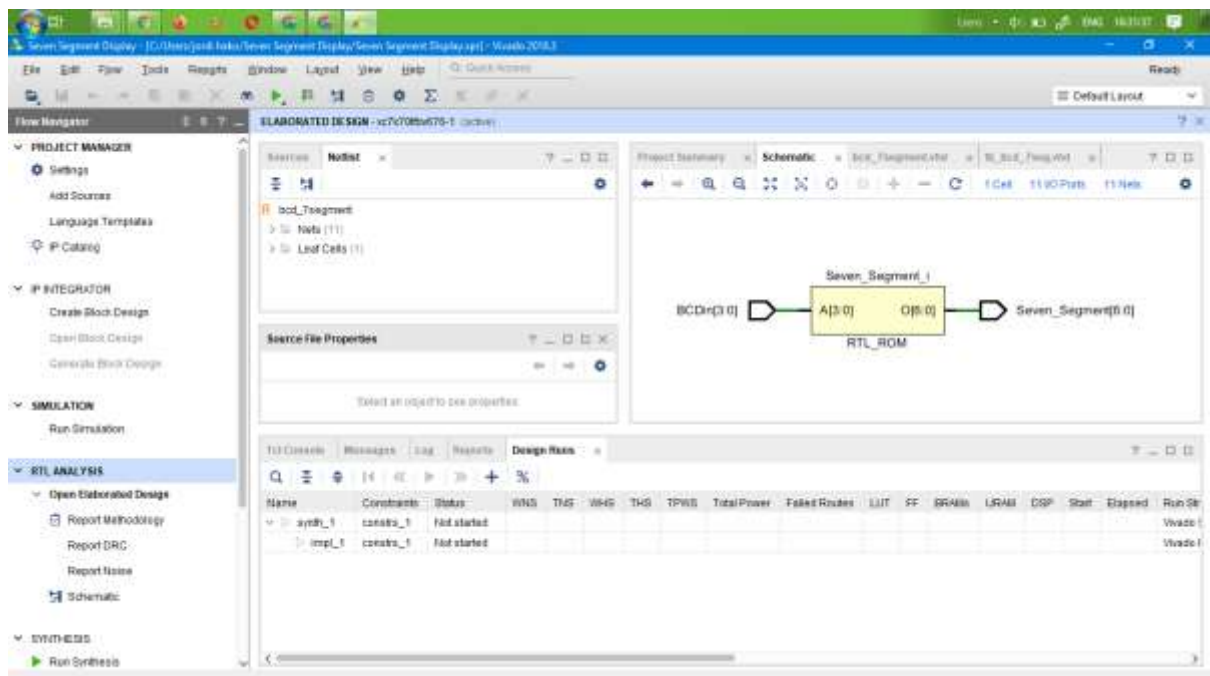
BCDin <= "0000";
wait for 100 ns;
BCDin <= "0001";
wait for 100 ns;
BCDin <= "0010";
wait for 100 ns;
BCDin <= "0011";
wait for 100 ns;
BCDin <= "0100";
wait for 100 ns;
BCDin <= "0101";
wait for 100 ns;
BCDin <= "0110";
wait for 100 ns;
BCDin <= "0111";
wait for 100 ns;
BCDin <= "1000";
wait for 100 ns;

```

```
BCDin <= "1001";
wait for 100 ns;
end process;
```

```
END;
```

RTL ANALYSIS



SIMULATION

