EE3921 Digital System Design

Section 031, Fall 2020

Professor: Kerry R. Widder, Ph.D.

Electrical Engineering and Computer Science Department

Milwaukee School of Engineering

Laboratory 3: “ROM counter logic design"

Jorge J. Jurado-Garcia

Date: 09/17/2020

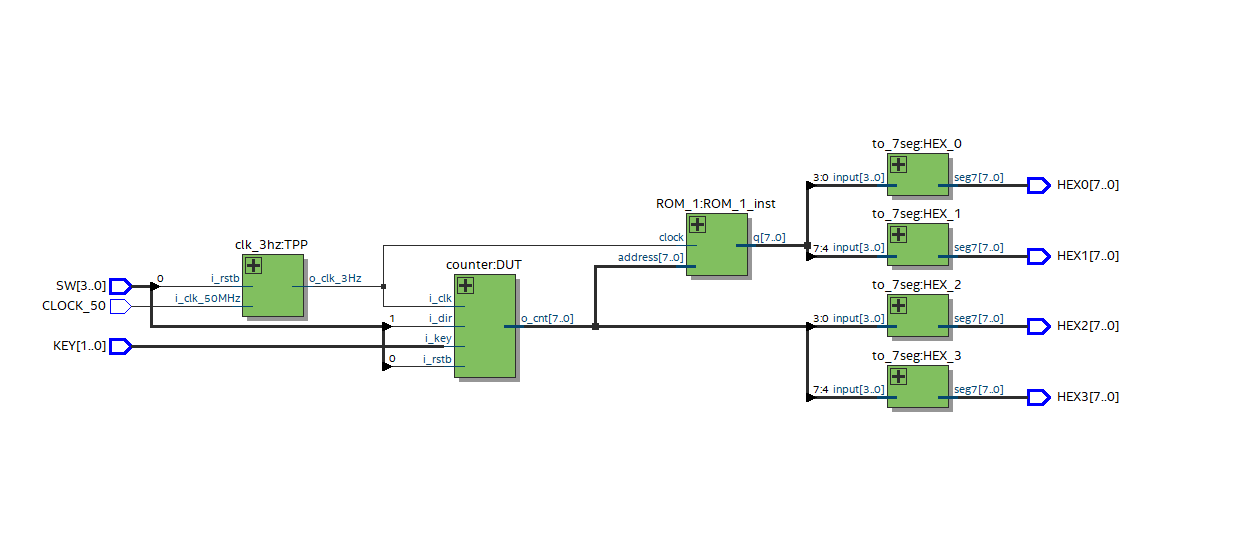
Level Attempted: BASIC

Objective: The objective of this lab was to create a ROM component using Mega Wizard (IP Catalog). This will be a one-port ROM with a data bus width of 8-bits and a depth of 256. Using the DE10-Lite create a counter that foes up to 0xFF once per second and display the memory value on a 7-segment display.

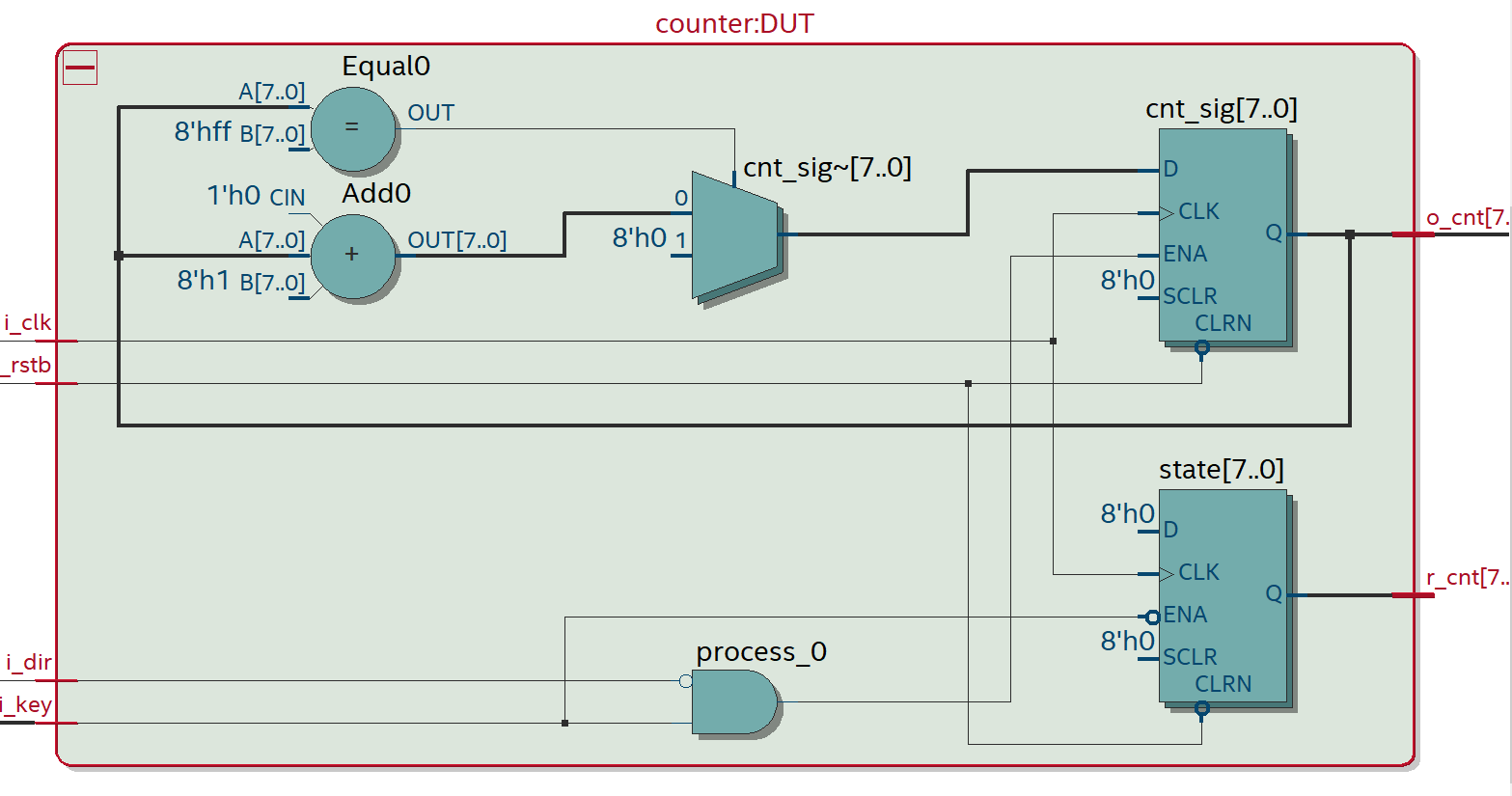
Description: In this lab I created my ROM using the Mega Wizard (IP catalog). With data bus width of 8-bits and a depth of 256. Using the romtest1.mif supplied by Dr.Widder for the contents of the ROM. I then also created a counter that would increment count at every clock edge if i\_DIR is low and would use one of the pushbutton switches (KEY) to allow pausing of the count sequence. All of the information will be displayed on a 7-segment display. With Hex3 and Hex2 displaying the current count value and Hex1 and Hex0 displaying the ROM Data.

Conclusion: Overall, this lab was a success, the toughest part I had with this lab would be creating the ROM component using the Mega Wizard I had trouble creating the proper one to make sure it would work properly. The easiest part was creating the counter since I just reused on of my previous code and changed some my sequential logic in order to fit the criteria for this one. Overall, this lab was a success.

RTL Viewer



Counter\_RTL\_Viewer



VHDL CODE

Counter.vhd

----------------------------------------------

--counter.vhdl

--

--created 9/21/21

--Jorge Jurado-Garcia

--rev 0

----------------------------------------------

--

--Inputs: rstb, clk, dir

--outputs: cnt

--

----------------------------------------------

--

--counts up when dir = 0

--counts down when dir = 1

--

----------------------------------------------

**library** ieee**;**

**use** ieee**.**std\_logic\_1164**.all;**

**use** ieee**.**numeric\_std**.all;**

**use** ieee**.**std\_logic\_unsigned**.all;**

-----------------------------------------------------------------------------

---ENtity

-------------------------------------------------------------------------------

--created a 8 bit generic counter with clk, rstb, and direction as inputs with a 4 bit output

**entity** counter **is**

**generic(**

n**:** natural **:=** 8 --this parameters controls the parameter of are counter

**);**

**port** **(**

i\_clk**:** **in** std\_logic**;**

i\_rstb**:** **in** std\_logic**;**

i\_dir**:** **in** std\_logic**;**

i\_key**:** **in** std\_logic**;**

o\_cnt**:** **out** std\_logic\_vector**(**n**-**1 **downto** 0**);**

r\_cnt**:** **out** std\_logic\_vector**(**n**-**1 **downto** 0**)**

**);**

**end** **entity;**

-------------------------------------------------------------------------

--architecture

-------------------------------------------------------------------------

**architecture** behavioral **of** counter **is**

--

--internal signals, signals are unsigned in order to start from 0 to 2^(n-1)

--

**signal** cnt\_sig**:** unsigned**(**n**-**1 **downto** 0**);**

**signal** rcnt\_sig**:** unsigned**(**n**-**1 **downto** 0**);**

**signal** state**:** unsigned**(**n**-**1 **downto** 0**);**

**begin**

**process(**i\_clk**,** i\_rstb**,** i\_key**)** --sequential processing

**begin**

--

--reset

--

--when rstb is 0 signal is 0

**if(**i\_rstb **=** '0'**)** **then**

cnt\_sig **<=** **(others** **=>** '0'**);** --make sure all signals are zero when rstb is zzero

rcnt\_sig **<=** **(others** **=>** '0'**);**

state **<=** **(others** **=>** '0'**);**

--rising clk edge

**elsif(rising\_edge(**i\_clk**))** **then**

**if(**i\_key **=** '0' **)** **then** --button logic

cnt\_sig **<=** cnt\_sig**;**

state **<=** rcnt\_sig**;** --when state rcnt\_sig

**end** **if;** --end of if statement for i-rstb

**if(** **(**i\_dir **=** '0'**)** **and** **(**i\_key **=** '1'**)** **)then**

cnt\_sig **<=** cnt\_sig **+** 1**;** --adding one

**if(**cnt\_sig **=** "11111111"**)** **then** --0x 11001

cnt\_sig **<=** "00000000"**;**

**end** **if;**--end if of wrapping up

**end** **if;** --end of if statement for i-rstb

**end** **if;**

**end** **process;**

--

--Output logic

--

o\_cnt **<=** std\_logic\_vector**(**cnt\_sig**);**

r\_cnt **<=** std\_logic\_vector**(**state**);**

**end** behavioral**;** --end of counter architerial

TIMER\_N\_SEC.

**library** ieee**;**

**use** ieee**.**std\_logic\_1164**.all;**

**use** ieee**.**numeric\_std**.all;**

**entity** clk\_3hz **is**

**port(**

i\_clk\_50MHz**:** **in** std\_logic**;**

i\_rstb**:** **in** std\_logic**;**

o\_clk\_3Hz**:** **out** std\_logic

**);**

**end** **entity;**

**architecture** behavioral **of** clk\_3hz **is**

--

--constants and parameters

--

**constant** CLOCKS\_PER\_HALF\_PERIOD**:** signed**(**24 **downto** 0**)** **:=** **to\_signed((((**50\_000\_000**/**2**)/**1**)-**2**),**25**);**

--internal signals

**signal** cnt**:** signed**(**24 **downto** 0**);**

**signal** clk\_sig**:** std\_logic**;**

**begin**

**process(**i\_clk\_50MHz**,** i\_rstb**)**

**begin**

**if(**i\_rstb **=** '0'**)** **then**

cnt **<=** CLOCKS\_PER\_HALF\_PERIOD**;**

clk\_sig **<=** '0'**;**

**elsif** **(rising\_edge(**i\_clk\_50MHz**))** **then**

cnt **<=** cnt **-** 1**;**

**if(**cnt**<** 0**)** **then**

cnt **<=** CLOCKS\_PER\_HALF\_PERIOD**;**

clk\_sig **<=** **not** clk\_sig**;**

**end** **if;**

**end** **if;**

**end** **process;**

o\_clk\_3Hz **<=** clk\_sig**;**

**end** behavioral**;**

TO\_7Seg

----------------------------------------------

--

-- to\_7seg.vhdl

--

--created 9/11/21

--jorge jurado-garcia

--

--rev 0

------------------------------

-- takes a for bit bit value and then translates into a seven segment display

--------------------------------------------

--Inputs: input

--Outputs: seg7

------------------------------------------------

**library** ieee**;**

**use** ieee**.**std\_logic\_1164**.all;**

**use** ieee**.**numeric\_std**.all;**

**entity** to\_7seg **is**

**Port** **(** input **:** **in** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

seg7 **:** **out** STD\_LOGIC\_VECTOR **(**7 **downto** 0**)**

**);**

**end** to\_7seg**;**

**architecture** Behavioral **of** to\_7seg **is**

**begin**

--'input' corresponds to MSB of seg7 and 'g' corresponds to LSB of seg7.

-- when a bit is low the led on the seg7 will be on else '0'

**process** **(**input**)**

**BEGIN**

**case** input **is**

**when** "0000"**=>** seg7 **<=**"01000000"**;** -- '0'

**when** "0001"**=>** seg7 **<=**"01111001"**;** -- '1'

**when** "0010"**=>** seg7 **<=**"00100100"**;** -- '2'

**when** "0011"**=>** seg7 **<=**"00110000"**;** -- '3'

**when** "0100"**=>** seg7 **<=**"00011001"**;** -- '4'

**when** "0101"**=>** seg7 **<=**"00010010"**;** -- '5'

**when** "0110"**=>** seg7 **<=**"00000010"**;** -- '6'

**when** "0111"**=>** seg7 **<=**"01111000"**;** -- '7'

**when** "1000"**=>** seg7 **<=**"00000000"**;** -- '8'

**when** "1001"**=>** seg7 **<=**"00011000"**;** -- '9'

**when** "1010"**=>** seg7 **<=**"00001000"**;** -- 'A'

**when** "1011"**=>** seg7 **<=**"00000000"**;** -- 'b'

**when** "1100"**=>** seg7 **<=**"01000110"**;** -- 'C'

**when** "1101"**=>** seg7 **<=**"00000000"**;** -- 'D'

**when** "1110"**=>** seg7 **<=**"00000110"**;** -- 'E'

**when** "1111"**=>** seg7 **<=**"00001110"**;** -- 'F'

**when** **others** **=>** **NULL;**

**end** **case;**

**end** **process;**

**end** Behavioral**;**

VGA\_COUNTER\_DE102

--------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------

--LAB 3 ROM COUNTER

--DATE: 9/21/2021

--AUTHOR: JORGE JURADO-GARCIA

--REV 1

--

----------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------

**library** ieee**;**

**use** ieee**.**std\_logic\_1164**.all;**

**use** ieee**.**numeric\_std**.all;**

**entity** VGA\_COUNTER\_DE102 **is**

**port(**

CLOCK\_50**:** **in** std\_logic**;**

SW **:** **in** std\_logic\_vector**(**3 **downto** 0**);**

HEX0**,**HEX1**,**HEX2**,**HEX3**:** **out** std\_logic\_vector**(**7 **downto** 0**);**

KEY**:** **in** std\_logic\_vector**(**1 **downto** 0**)**

**);**

**end** **entity** VGA\_COUNTER\_DE102**;**

**Architecture** behavioral **of** VGA\_COUNTER\_DE102 **is**

---signals that will be used

**signal** CLK\_1HZ**:** std\_logic**;**

**signal** null\_value**:** std\_logic\_vector**(**7 **downto** 0**);**

**signal** ADDRESS\_SIG**:** std\_logic\_vector**(**7 **downto** 0**);**

**signal** q\_sig**:** std\_logic\_vector**(**7 **downto** 0**);**

---------------------------------

--component declarations

---------------------------------

**component** counter **is**

**generic(**

n**:** natural **:=** 8 --this parameters controls the parameter of are counter

**);**

**port** **(**

i\_clk**:** **in** std\_logic**;**

i\_rstb**:** **in** std\_logic**;**

i\_dir**:** **in** std\_logic**;**

i\_key**:** **in** std\_logic**;**

o\_cnt**:** **out** std\_logic\_vector**(**n**-**1 **downto** 0**);**

r\_cnt**:** **out** std\_logic\_vector**(**n**-**1 **downto** 0**)**

**);**

**end** **component;**

**component** ROM\_1

**PORT**

**(**

address **:** **IN** STD\_LOGIC\_VECTOR **(**7 **DOWNTO** 0**);**

clock **:** **IN** STD\_LOGIC **:=** '1'**;**

q **:** **OUT** STD\_LOGIC\_VECTOR **(**7 **DOWNTO** 0**)**

**);**

**end** **component;**

**component** clk\_3hz **is**

**port(**

i\_clk\_50MHz**:** **in** std\_logic**;**

i\_rstb**:** **in** std\_logic**;**

o\_clk\_3Hz**:** **out** std\_logic

**);**

**end** **component;**

**component** to\_7seg **is**

**Port** **(** input **:** **in** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

seg7 **:** **out** STD\_LOGIC\_VECTOR **(**7 **downto** 0**)**

**);**

**end** **component;**

--------------------------------------------

**begin**

-----------------------------------

--Device under test (DUT)

-----------------------------------

TPP**:** clk\_3hz

**port** **map(**

i\_clk\_50MHz **=>** CLOCK\_50**,**

i\_rstb **=>** SW**(**0**),**

o\_clk\_3Hz **=>** CLK\_1HZ

**);**

DUT**:** counter

**port** **map(**

i\_clk **=>** CLK\_1HZ**,**

i\_rstb **=>** SW**(**0**),**

i\_dir **=>** SW**(**1**),**

i\_key **=>** KEY**(**1**),**

o\_cnt**(**7 **downto** 0**)** **=>** ADDRESS\_SIG**,**

r\_cnt **=>** null\_value

**);**

ROM\_1\_inst **:** ROM\_1

**PORT** **MAP** **(**

address**(**7 **downto** 0**)** **=>** ADDRESS\_SIG**(**7 **downto** 0**),**

clock **=>** CLK\_1HZ**,**

q **=>** q\_sig

**);**

HEX\_0**:** to\_7seg

**PORT** **MAP(**

input **=>** q\_sig**(**3 **downto** 0**),**

seg7 **=>** HEX0

**);**

HEX\_1**:** to\_7seg

**PORT** **MAP(**

input **=>** q\_sig**(**7 **downto** 4**),**

seg7 **=>** HEX1

**);**

HEX\_2**:** to\_7seg

**PORT** **MAP(**

input **=>** ADDRESS\_SIG**(**3 **downto** 0**),**

seg7 **=>** HEX2

**);**

HEX\_3**:** to\_7seg

**PORT** **MAP(**

input **=>** ADDRESS\_SIG**(**7 **downto** 4**),**

seg7 **=>** HEX3

**);**

**end** **architecture;**

Counter\_n\_bits\_tb

----------------------------------------------------------

--

--Counter\_n\_bit\_tb.vhdl

--

--created: 9/10/21

--by Jorge Jurado-Garcia

--rev: 0;

--

--testbench for up down counter

-- of counter\_n bit

--

---------------------------------------------------------

**library** ieee**;**

**use** ieee**.**std\_logic\_1164**.all;**

**entity** Counter\_n\_bit\_tb **is**

**generic(**

N**:** natural **:=** 8

**);**

**end** **entity;**

**architecture** testbench **of** Counter\_n\_bit\_tb **is**

**signal** CLK**:** std\_logic**;**

**signal** RSTB**:** std\_logic**;**

**signal** DIR**:** std\_logic**;**

**signal** KEY**:** std\_logic**;**

**signal** CNT**:** std\_logic\_vector**((**N **-** 1**)** **downto** 0**);**

**signal** RCNT**:** std\_logic\_vector**((**N**-**1**)** **downto** 0**);**

**constant** PER**:** time **:=** 15 ns**;**

------------------------------------

--Component Protype

------------------------------------

**component** counter **is**

**generic(**

n**:** natural **:=** 8

**);**

**port** **(**

i\_clk**:** **in** std\_logic**;**

i\_rstb**:** **in** std\_logic**;**

i\_dir**:** **in** std\_logic**;**

i\_key**:** **in** std\_logic**;**

o\_cnt**:** **out** std\_logic\_vector**(**n**-**1 **downto** 0**);**

r\_cnt**:** **out** std\_logic\_vector**(**n**-**1 **downto** 0**)**

**);**

**end** **component;**

----------------------------------------------

**begin**

------------------------------------

--DUT

------------------------------------

DUT**:** counter

**generic** **map(**

n **=>** N

**)**

**port** **map(**

i\_clk **=>** CLK**,**

i\_rstb **=>** RSTB**,**

i\_dir **=>** DIR**,**

i\_key **=>** KEY**,**

o\_cnt **=>**CNT**,**

r\_cnt **=>** RCNT

**);**

----------------------------------------

--Test Processes

----------------------------------------

--clock process

clock**:** **process**

**begin**

CLK **<=** '0'**;**

**wait** **for** PER**/**2**;**

infinite**:** **loop**

CLK **<=** **not** CLK**;** **wait** **for** PER**/**2**;**

**end** **loop;**

**end** **process;**

--Reset Process

reset**:** **process**

**begin**

RSTB **<=** '0'**;** **wait** **for** PER**;**

RSTB **<=** '1'**;** **wait;**

**end** **process** reset**;**

--run process

run**:** **process**

**begin**

-- intitalize inputs

DIR **<=** '0'**;**

--wait for reset

**wait** **for** 2**\***PER**;**

--run code

**wait** **for** **(**2**\*\***N**)\***PER**;**

DIR **<=** '1'**;**

**wait** **for** **(**2**\*\***N**)\***PER**;**

**end** **process** run**;**

--run process

runn**:** **process**

**begin**

-- intitalize inputs

KEY **<=** '0'**;**

--wait for reset

**wait** **for** 3**\***PER**;**

infinite**:** **loop**

KEY **<=** **not** KEY**;** **wait** **for** 3**\***PER**;**

**end** **loop;**

**end** **process** runn**;**

-----------------------------

--End test processes

--------------------------------

**end** **architecture;**