Lab Notebook

FPGA Capstone Project

Academic Integrity

By signing below you acknowledge the CU Honor Code, "On my honor, as a University of Colorado Boulder student I have neither given nor received unauthorized assistance" applies to this assignment.

Signed: _Glenn Frey Olamit_____

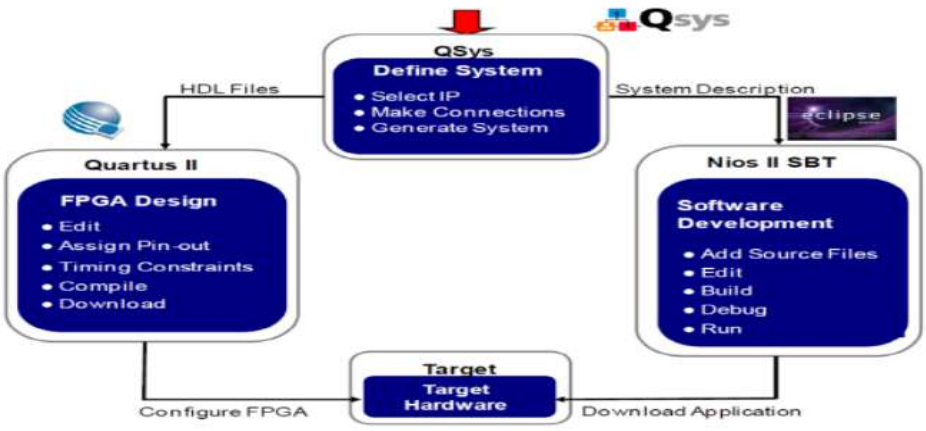
Module 4

NIOS II Software Design and System Test

Author: Glenn Frey Olamit
Date: March 28, 2022
Procedure/Description of Test:

The goal of this module is to develop the software for a System on a Chip (SoC), specifically the NIOS II Soft Processor built in Module 3.

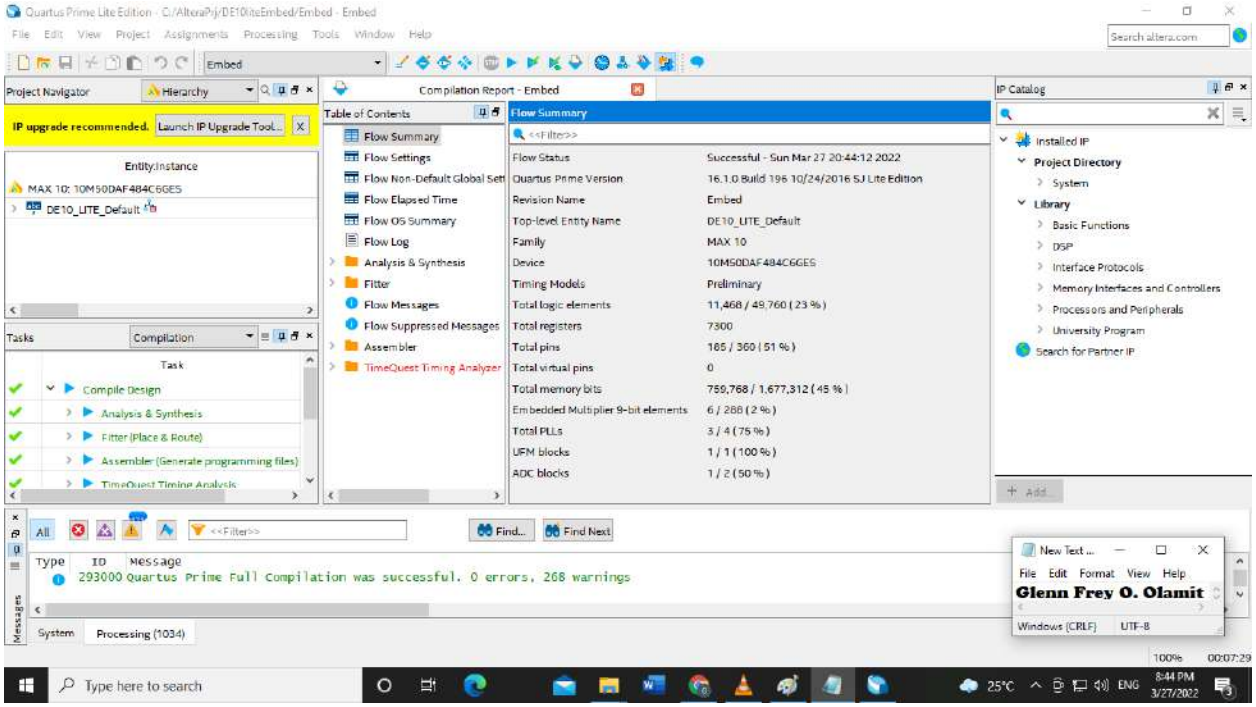
In this module, we will follow the flow for a system design (see figure below) but we will focus this time on the Software Development using the NIOS II Software Build Tools as Software Development Tool.

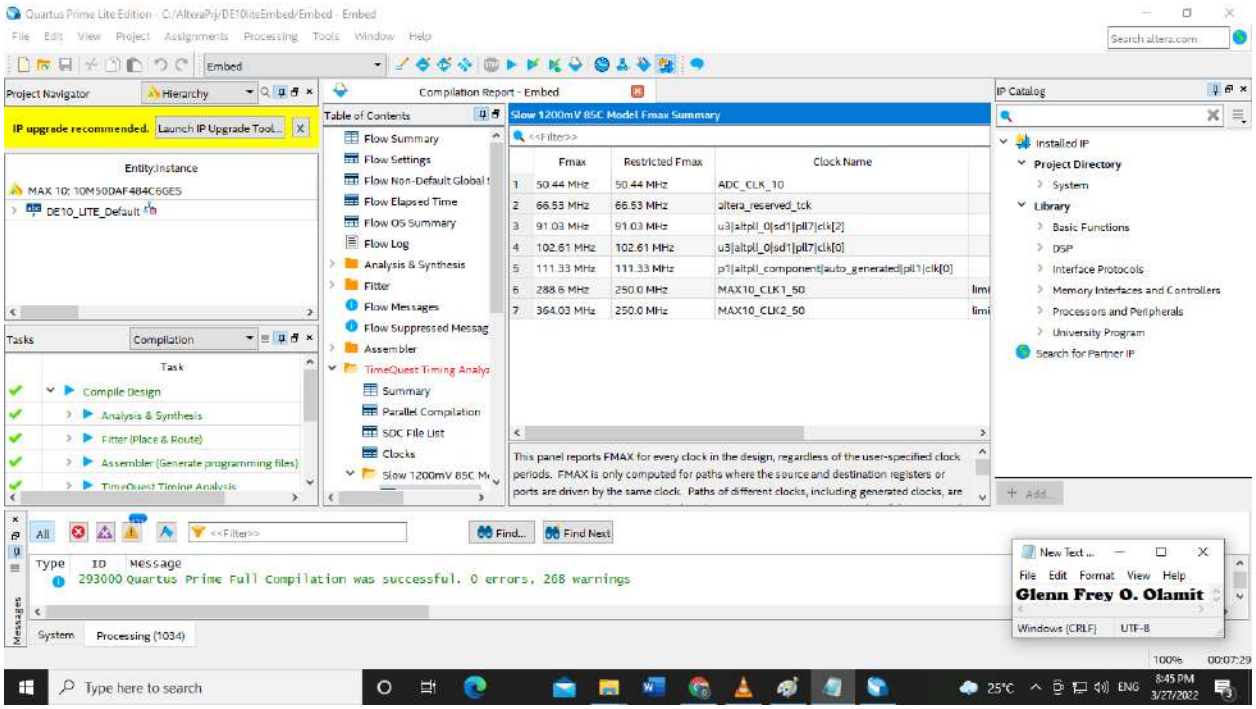


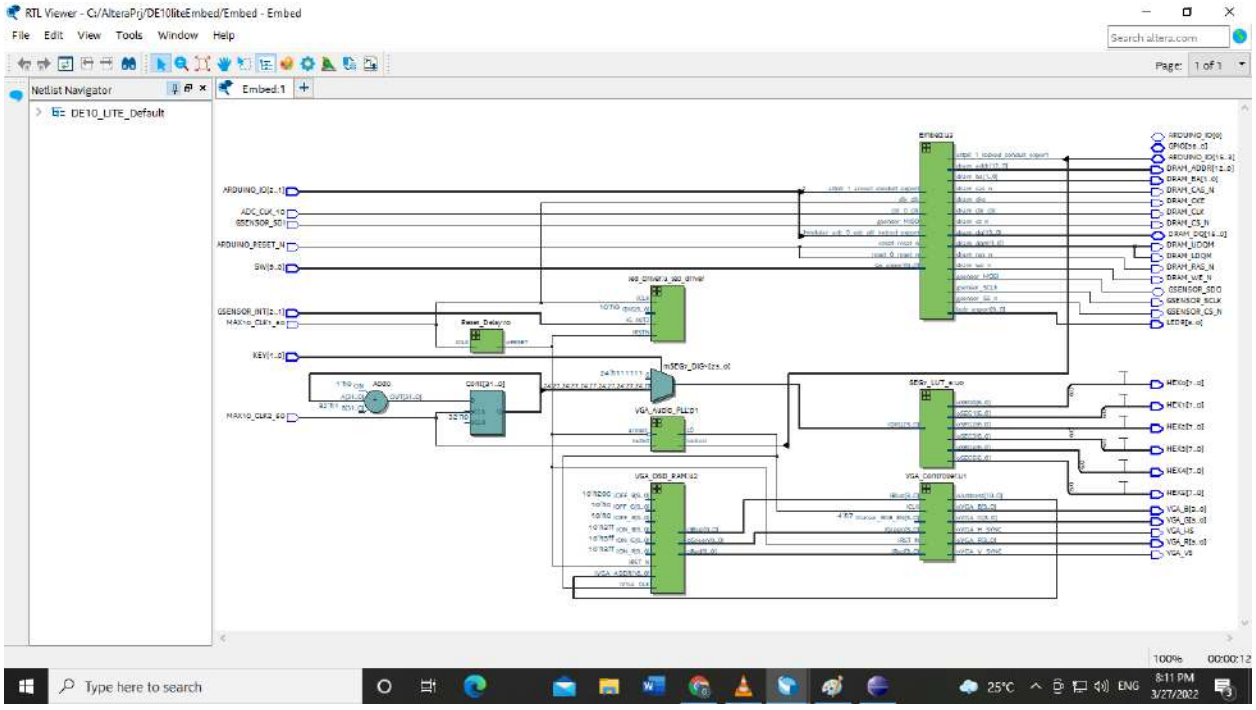
Observations:

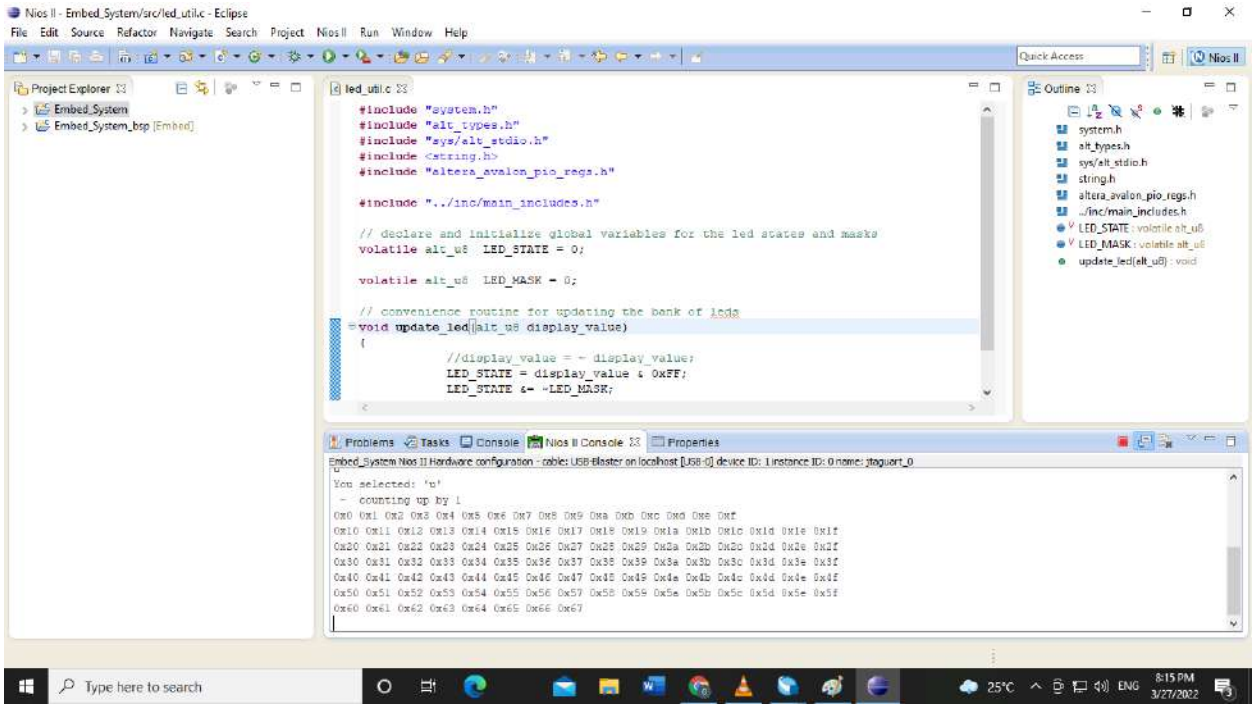
Data:

Images/Drawings:

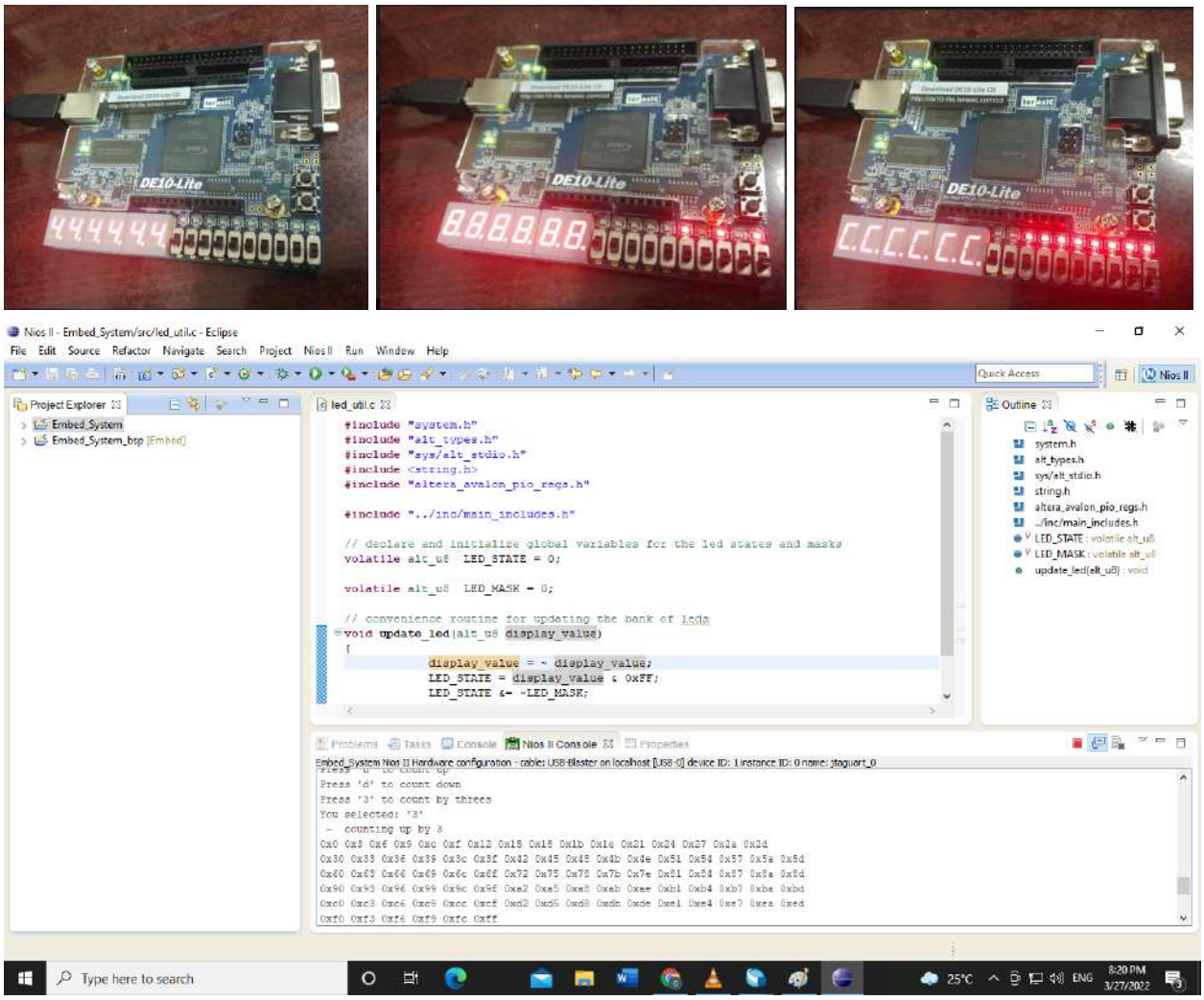





RTL Viewer - C:/AlteraPrj/DE10liteEmbed/Embed - Embed
File Edit View Tools Window Help
Netlist Navigator
DE10_LITE_Default
Embed.1
Type here to search
25°C
ENG
8:11 PM
3/27/2022


Nios II - Embed_System/src/led_util.c - Eclipse
File Edit Source Refactor Navigate Search Project Nios II Run Window Help
Project Explorer
Embed_System
Embed_System_bsp [Embed]
led_util.c
#include "system.h"
#include "alt_types.h"
#include "sys/alt_stdio.h"
#include <string.h>
#include "altera_avalon_pio_regs.h"
#include "../inc/main_includes.h"
// declare and initialize global variables for the led states and masks
volatile alt_u8 LED_STATE = 0;
volatile alt_u8 LED_MASK = 0;
// convenience routine for updating the bank of leds
void update_led(alt_u8 display_value)
{
//display_value = ~ display_value;
LED_STATE = display_value & 0xFF;
LED_STATE &= ~LED_MASK;
Outline
system.h
alt_types.h
sys/alt_stdio.h
string.h
altera_avalon_pio_regs.h
../inc/main_includes.h
LED_STATE : volatile alt_u8
LED_MASK : volatile alt_u8
update_led(alt_u8) : void
Problems Tasks Console Nios II Console Properties
Embed_System Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0 name: jtaguart_0
You selected: 'n'
- counting up by 1
0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xa 0xb 0xc 0xd 0xe 0xf
0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17 0x18 0x19 0x1a 0x1b 0x1c 0x1d 0x1e 0x1f
0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27 0x28 0x29 0x2a 0x2b 0x2c 0x2d 0x2e 0x2f
0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x39 0x3a 0x3b 0x3c 0x3d 0x3e 0x3f
0x40 0x41 0x42 0x43 0x44 0x45 0x46 0x47 0x48 0x49 0x4a 0x4b 0x4c 0x4d 0x4e 0x4f
0x50 0x51 0x52 0x53 0x54 0x55 0x56 0x57 0x58 0x59 0x5a 0x5b 0x5c 0x5d 0x5e 0x5f
0x60 0x61 0x62 0x63 0x64 0x65 0x66 0x67
Type here to search
25°C
ENG
8:15 PM
3/27/2022

After ‘u’ is selected the LEDs start counting up (as shown below image from left to right) in accordance with the displayed values in Nios II console(as shown above image).



After adding the line “display_value = ~display_value;” and ‘3’ is selected the LEDs start counting up in multiples of 3 in binary format (as shown below image from left to right) in accordance with the displayed values in Nios II console(as shown above image).

Results:

Fmax	ADC_CLK_10: 50.44 MHz altera_reserved_tck: 66.53 MHz p1|altpll_component|auto_generated|pll1|clk[0]: 91.03 MHz u3|altpll_0|sd1|pll7|clk[0]: 102.61 MHz u3|altpll_0|sd1|pll7|clk[2]: 111.33 MHz MAX10_CLK2_50: 364.03 MHz MAX10_CLK1_50: 288.06 MHz
Logic Utilization	23%
Total registers	7300

Questions:

1. Is the control of the 10 LEDs implemented in hardware or in software? The control of the LEDs is implemented in software. We did invert the LEDs behavior by changing the software code. The hardware as it is can not control the LEDs as we commented out the LEDR assignment statement in Module 3.
2. Is the control of the 7-segment LEDs done by hardware or by software? The control of the 7-segment LEDs is done by hardware as there is an assignment statement outputting a counter to the 7-segment displays in the hardware design.
3. How much memory is required to run your Nios II program? Can you fit it into the onchip RAM if you redesign the onchip RAM block? to run the NIOS II program 24KB of memory are required. It is possible to fit it into onchip RAM if it is redesigned to be 32KB instead of 16KB for example.
4. Your Nios II processor is running at what clock speed? How much faster can it run in your MAX10 design? The NIOS II processor is running at 80MHz. The Fmax is 102.61 Mhz. Perhaps this can be improved by adding some timing constraints.

Conclusions:

A software application that runs on a NIOS II soft processor was created. We were able to control the Hardware behavior by Software. The NIOS II SBP for Eclipse is a powerful Software Development Environment for the NIOS II Soft Processor.

The Board Support Package can be configured by using the BSP Editor. The BSP Editor controls what drivers will be built into the Board Support Package

Graded 98/100.

Lessons Learned (What did you learn?):

I learned to set up and run the NIOS II SBP for Eclipse. I learned how to create a Software Application for a NIOS II Soft Processor. I learned how to use the NIOS II SBP for Eclipse as well as the BSP Editor. I learned how to run a Software Application into a NIOS II Soft Processor.