

PROJECT #1 GUIDE

UCB ECEN 5863 SPRING 2023 PROJECT #1: ALTERA MAX10

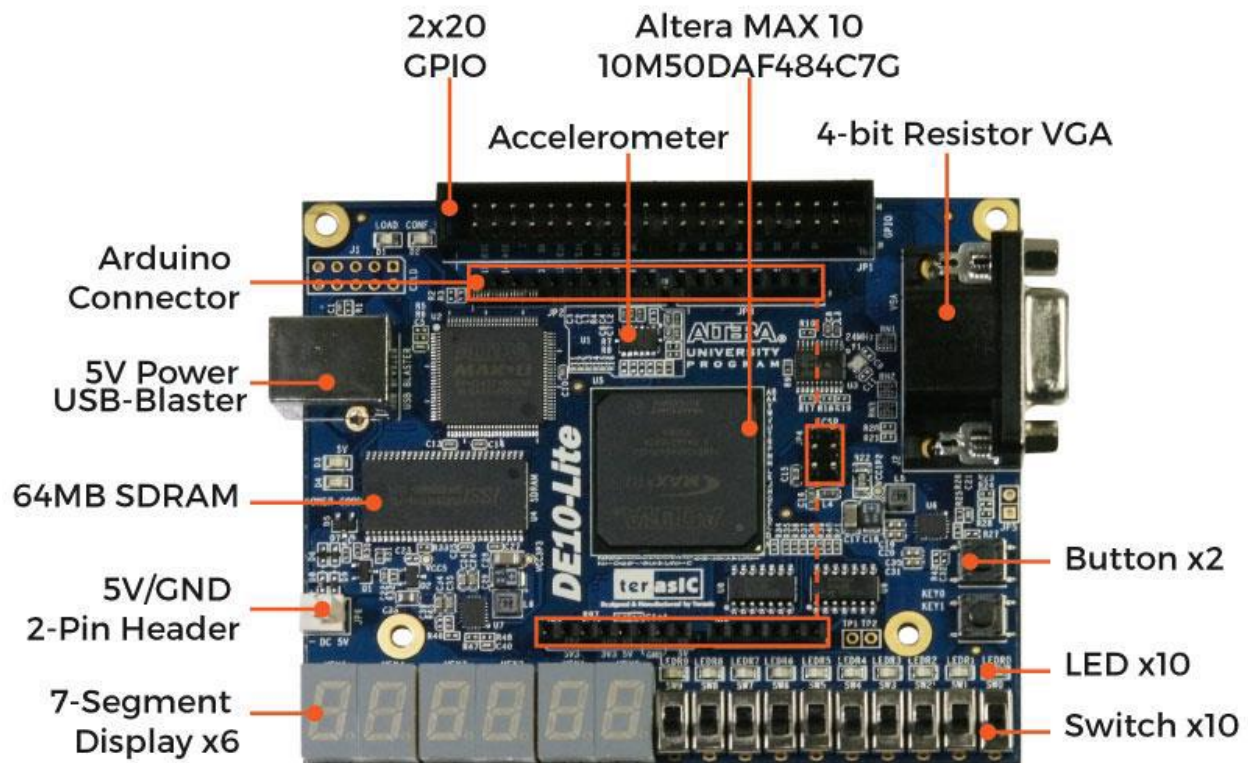
TABLE OF CONTENTS

Introduction.....	2
Learning Objectives	2
I. Procedure	4
Setup MAX10 FPGA	4
Module I: Develop a Mixed-Signal system.....	4
DEMO I: (TA will Review the Demos).....	4
Module II: DISPLAY HELLO WORLD AND TOGGLE LEDS Using NIOS II and Qsys	5
DEMO II:.....	5
Module III: Create a System on a chip with Programmable Hardware and a Soft PProcessor	6
DEMO III.....	6
II. Deliverables	7
Technical Report	7
Recorded Observations, Test Data, and Images	7
FPGA Project directory zipped for each of 3 Modules.....	7
Software Files for the final module in a zip file	7
III. Evaluation.....	8

INTRODUCTION

The DE10-Lite is a FPGA evaluation kit that is designed to get you started with using an FPGA. The DE10-Lite adopts Altera's non-volatile MAX[®] 10 FPGA built on 55-nm flash process. MAX 10 FPGAs enhance non-volatile integration by delivering advanced processing capabilities in a low-cost, instant-on, small form factor programmable logic device. The devices also include full-featured FPGA capabilities such as digital signal processing, analog functionality, Nios II embedded processor support, and memory controllers.

The DE10-Lite includes a variety of peripherals connected to the FPGA device, such as 8MB SDRAM, accelerometer, digital-to-analog converter (DAC), temperature sensor, thermal resistor, photo resistor, LEDs, pushbuttons and several different options for expansion connectivity.



LEARNING OBJECTIVES

For this project, the objective is for students to:

- Become familiar with the FGPA development flow, particularly in the case of a SoC with software development flow included.
- Appreciate the capability of the MAX10 to create whole systems on a chip.
- Learn how to build systems using the Qsys system design tool.
- Learn to create hardware using schematic capture input.
- Learn how to integrate software with hardware in the same device.

- Understand the rationale for each phase of the hardware development flow, including timing constraints, simulation, and programming.
- Design and build a several hardware examples using the MAX10.
- Consider hardware and software tradeoff possibilities available in the SoC architecture.

Do not be afraid to ask questions as you work on this project. It involves many processes and actions that may seem complicated and confusing at first, but will become clearer as you work through the modules.

The modules will get progressively more difficult and take more time. Try to finish Modules 1 and 2 in the first week, so that you will have the second week to complete Module 3 and write the report.

Note: Use the DE10 Lite User's Manual to understand the user-defined switches, push buttons, LED, etc. terms. The manuals in this project have Switches defined as SW0-9 (required in M1 and M2), and Push Buttons as KEY0 and KEY1 (required in M2).

I. PROCEDURE

SETUP MAX10 FPGA

1. If you have not already done this, download and install Quartus PrimeFPGA development software from Altera.
2. Download and install the Tools in the Tools directory for Project 1 on Canvas

MODULE I: DEVELOP A MIXED-SIGNAL SYSTEM

1. From Canvas, download the pdf document Prj1M1PWM_Guide.pdf.
2. Follow the instructions with these exceptions:
 - a. If your results look different than the lab guide, do not be alarmed, as there are some differences between installations is not unexpected. Try to determine the most reasonable course of action if this happens.
 - b. Do not be concerned about trying to determine what .vhd or .v files do. We will learn this later in the course.
3. Do not be surprised when the initial compile has errors. This will point you to the additional work you need to do.
4. Record the fmax.
5. Estimate the % utilization of the FPGA logic.
6. Record your observations of the board behavior once the FPGA is programmed. Does it behave as you expected?
7. From Canvas, download the pdf document Prj1M1ADC_Guide.pdf.
8. Follow the instructions:
9. Record the fmax.
10. Estimate the % utilization of the FPGA logic.
11. Record your observations of the board behavior once the FPGA is programmed. Does it behave as you expected? Is this a good voltmeter? What could you change in either the board hardware or FPGA logic to make it perform better?

DEMO I: (TA WILL REVIEW THE DEMOS)

1. Demo LEDs intensity variation using the switch inputs.
2. Demo 7-segment output variation using the switch inputs.



MODULE II: DISPLAY HELLO WORLD AND TOGGLE LEDS USING NIOS II AND QSYS

1. From Canvas, download the pdf document Hello_World_Lab_Manual_DE10_Lite_DevKit.pdf.
2. Follow the instructions from pg 6- pg 37 and Appendix A (if you need to). While doing this Module, follow these instructions:
 - a. Do both the hardware and software sections of the manual.
 - b. Don't use the .sof files (which is an option in the guide) as you need to build the complete hardware using Qsys and submit all Quartus files to CANVAS as part of the submission.
 - c. Don't be worried about Verilog coding, the code segments are provided in the guide. Be careful with port mapping in nios_setup_v2 to avoid any cut and paste errors.
 - d. While doing the software part (NIOS II) make the workspace be NIOS II in your project directory for easier submissions.
 - e. **Hint:** You may get errors in the C program and header file paths. You may need to do some additional work to debug them. (Don't use absolute paths in header files, it can't be replicated by us while grading.)
3. Record the Fmax.
4. Estimate the % utilization of the FPGA logic.
5. Drill down into the second layer of in the RTL Viewer. You can look at your connections made using Qsys (or Platform Designer as it's now known). Submit a screen shot of this.
6. Record your observations of the board behavior once FPGA is programmed. Does it behave as you expected?
7. Answer the following
 - a. Which component is the Instruction master of the NIOS II connected to? What is the need for this and why it is connected only to one component (slave)?
 - b. How are the LED and the switch connected using S/W?
 - c. How are the LED and the switch connected using H/W?

DEMO II:

1. Demo the original toggling of LEDs using Switches (SWn) and "Hello World" on the NIOS II Console.
2. Implement the code so that your name appears only once while pushing the button 1 and your partners name while pushing another button 2 (the ones used for toggling the LED) on the NIOS II Console. Comment on whether or not you should debounce the push buttons (KEY0 and KEY1 are push buttons, configure these in the Qsys designer and regenerate).





MODULE III: CREATE A SYSTEM ON A CHIP WITH PROGRAMMABLE HARDWARE AND A SOFT PROCESSOR

1. ALLOW PLENTY OF TIME FOR THIS MODULE.
2. From Canvas, download the pdf document DE10_Embedded_System_LabR2.pdf.
3. Follow the instructions with these exceptions:
 - a. If your results look different than the lab guide, do not be alarmed. Try to determine the most reasonable course of action if this happens.
 - b. Do not be concerned about trying to determine what .vhd or .v files do. We will learn this later in the course.
4. While doing the module keep in mind following :
 - a. Do not be surprised when the initial and even subsequent compiles have errors. This will point you to the additional work you need to do.
 - b. While following the guide you (Quartus v15.0 and above) may get the following issue: "The Serial Flash Controller not supported". You may need to edit "Supported Devices" of file serial_flash_controller_hw.tcl.
 - c. You may not be able to rename the clocks. It doesn't cause any issues, and you can move ahead without renaming but be careful with connections, they should be according to the guide.
 - d. In a new menu options, "i", the LED Display could be inverted. According to DE10_Embedded_System_LabR2.pdf this may not appear on the options. Can it be implemented from scratch?
5. Ponder the Nios II for a moment. Why do you think the architecture is as it is? What advantages come from building a processor out of programmable logic?
6. Open the software source files, including all .c and .h files. What do you observe about the coding style and use of comments?
7. When you modified the software, how did the LED display change?
8. Record the fmax. What is the highest speed clock used in the design?
9. Estimate the % utilization of the FPGA logic.
10. Record your observations of the board behavior once the FPGA is programmed. Does it behave as you expected?
11. Answer the following
 - a. Why is the instruction master of NIOS II connected to multiple components? Explain the usage of each of them.
 - b. What's the function for exporting external connection for different components?
 - c. What is requirement for multiple clocks in this design?

DEMO III

1. Demo all options of the menu displayed.
2. Implement a down counter by ten and an up counter by five by adding more options on the list of menu.

II. DELIVERABLES

All deliverables should be completed on the schedule listed:

Module 1: 2023/02/11

Module 2: 2023/02/16

Module 3: 2023/09/24

Report: 2023/02/25 11:59 PM.

Please submit them to Canvas.

(Look out for Extra credit opportunities given within the module files and submit them with the respective modules)

TECHNICAL REPORT

Each partner pair must provide a technical report that explains the project objectives and test results. At a minimum, it should include:

1. Executive Summary
2. Objectives
3. Procedure
4. Module Test Results
5. Lessons Learned
6. Conclusions
7. Appendix: References

The report should be no longer than 5 pages in length in a Word document (excluding table of contents and References). Be sure the report addresses the questions posed in the module procedures.

RECORDED OBSERVATIONS, TEST DATA, AND IMAGES

Include observations recorded in a lab notebook, test data taken, and any digital pictures of the proceedings.

FPGA PROJECT DIRECTORY ZIPPED FOR EACH OF 3 MODULES

Starting with the top level of each module, zip up the entire directory including subfolders. This will allow us to replicate your work and help provide feedback on any errors you encounter. Label them appropriately so we can find the information for each module. With submission include a ReadMe.txt to describe and list the locations of deliverables.

SOFTWARE FILES FOR THE FINAL MODULE IN A ZIP FILE

Zip up all source files used to run the Nios II software.

III. EVALUATION

Any award to be made pursuant to this project will be based upon deliverables. The following elements will be the primary considerations in evaluating all submitted projects and in the selection of a partner team for Top Prizes:

20% Technical Report

20% TA Demos

10% Deliverables for Project Module 1

20% Deliverables for Project Module 2

30% Deliverables for Project Module 3