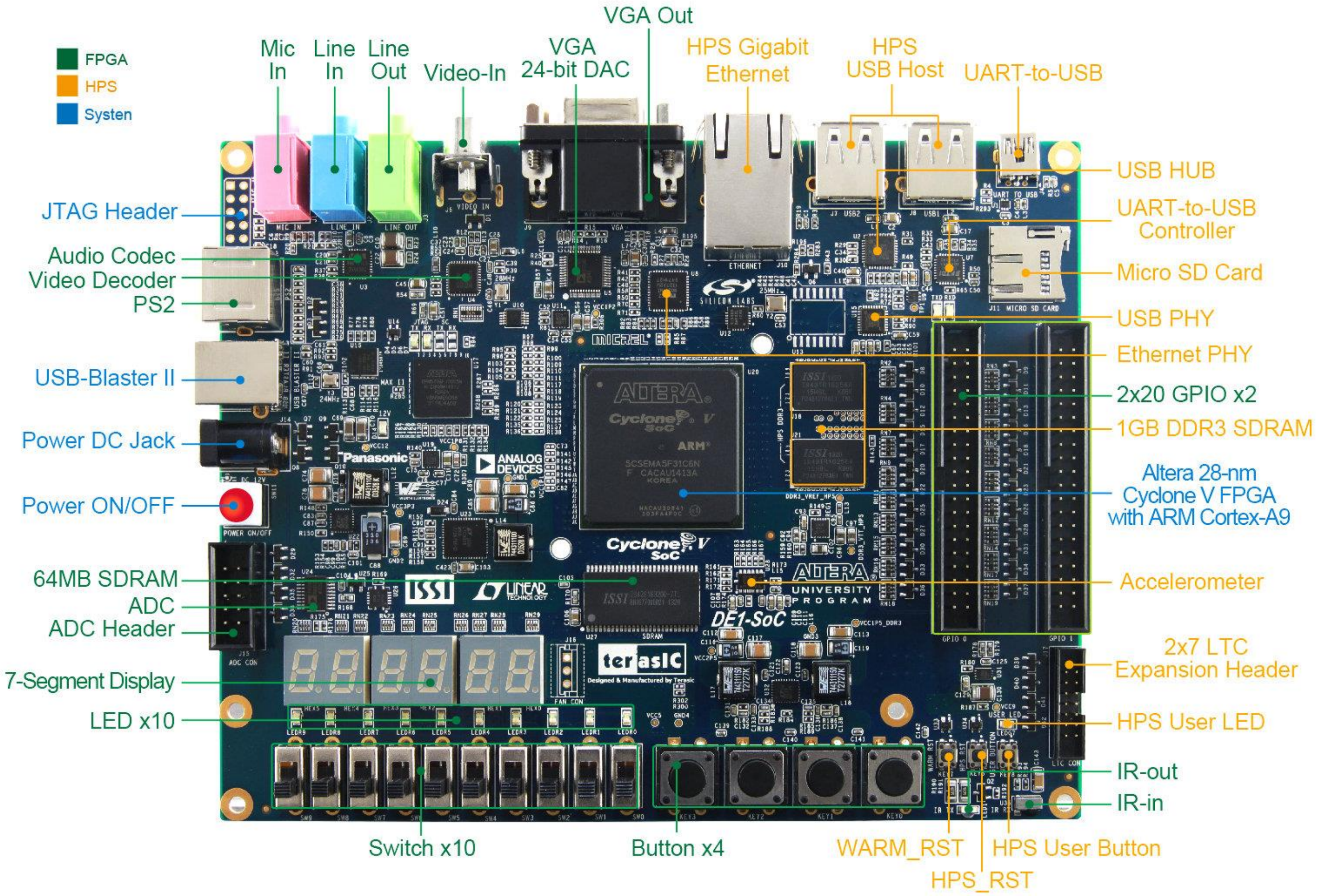
ECE 554: Minilab 0

Getting started on DE1-SoC Board with Quartus

DE1-SoC Board Layout



Minilab 0 Description

- Objectives:
  - Design simple digital logic in Verilog and simulate it in QuestaSim
  - Use Quartus to program the DE1-SoC board
  - Use IPs from the Quartus IP Catalog
  - Understand resource utilization report from Quartus synthesis tool
- The lab is structured into two parts:
  - **Part 1**: Design your custom logic in Verilog/SystemVerilog
  - **Part 2**: Use IP blocks from Quartus IP Catalog for the same logic
- This lab will use the following peripheral of the FPGA board:
  - 7-Segment Display
  - LEDs
  - Buttons (More specifically Key 0 will be used as active low reset)

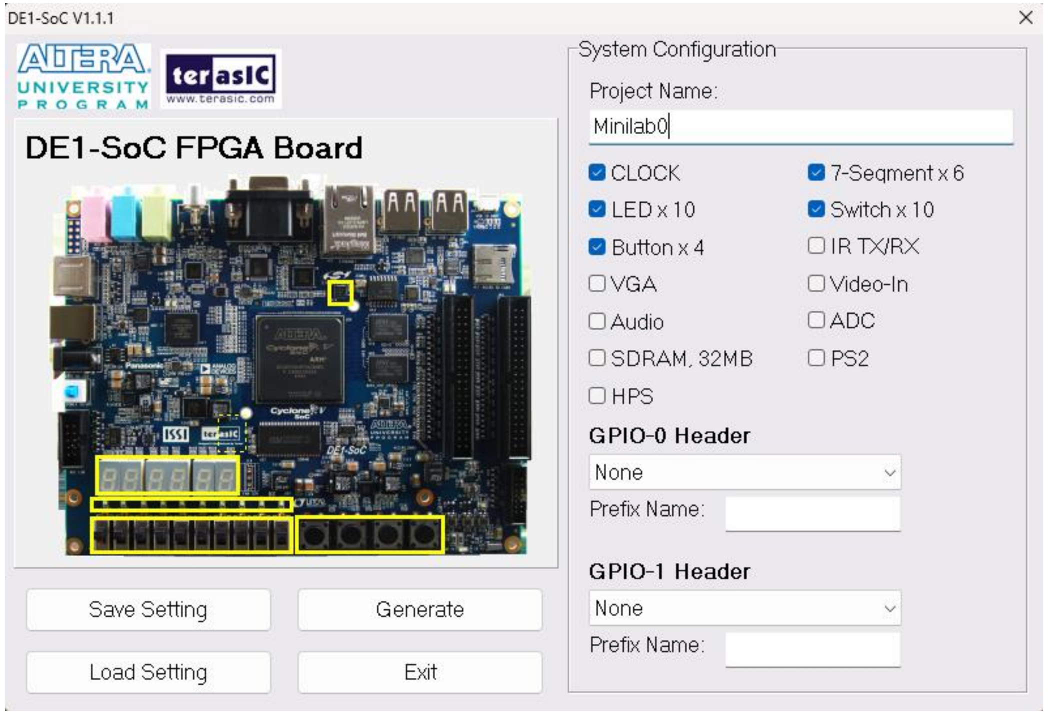
Create a project

- DE1-SoC evaluation board provides a software that can help us generate all the project files. Using DE1-SoC “System Builder”, we simply generate the project files with all the settings
- Setting include setting the correct device and pin information to use in Quartus software
- To run System Builder, download Altera DE1-SoC SystemCD from the course website under Minilab0
- Open

DE1-SoC_v.5.1.1_HWrevF_SystemCD\Tools\SystemBuilder\DE1SoC_SystemBuilder.exe

Create a project

- Set “Project Name” to “Minilab0”
- Select
  - CLOCK
  - Buttonx4
  - LEDx10
  - 7-Segmentx6
- Set GPIO-0 Header to “None”
- Click “Generate”, save as “Minilab0.qpf”
- The generated files will be placed under the “CodeGenerated/DE1-SoC” folder under SystemBuilder



System Configuration

Project Name: Minilab0

☑ CLOCK
☑ LED x 10
☑ Button x 4
□ VGA
□ Audio
□ SDRAM, 32MB
□ HPS
☑ 7-Segment x 6
☑ Switch x 10
□ IR TX/RX
□ Video-In
□ ADC
□ PS2

GPIO-0 Header: None

Prefix Name:

GPIO-1 Header: None

Prefix Name:

File Description

- Minilab0.qpf: Quartus project file
- Minilab0.qsf: Quartus setting file, including device, pin, assignments, etc.
- Minilab0.sdc: Synopsys design constraint file used for compiling the design
- Minilab0.v: Top level Verilog HDL template file

The use of system builder is very handy as it ensures the pin mappings in the .qsf file are correct.

In most cases you may wish to edit the generated .qsf file to use more “standard” signal names.

Add sources to the project

- Normally you would copy the generated files to your work area and edit both the .qsf and the top -level .v file to define your desired signal names.
- However, for this lab, you will use the provided files (from the zip file) which have been edited to have more meaningful signal names for some ports.
- Download Minilab0.zip from the website
- In the lab zipfile, we have provided the top-level .v, a couple of .sv files and other Quartus project-related files for you. Copy all the files to the **Minilab0** folder under **CodeGenerated/Minilab0** folder created by the system builder software. Make sure to also copy the .qsf file from the .zip and overwrite the generated .qsf file.

Open the project

- From **Minilab0** folder, click **Minilab0.qpf** to open the project in Quartus
- Click “project”-> “add/remove file in project”
- Add all the source files into your project
- The .qsf file has the top-level pin assignments for the peripherals used in this lab (Check tables 3-5 to 3-9 in *DE1-SoC_v.5.1.1_HWrevF_SystemCD/UserManual/ DE1-SoC_User_manual.pdf*). If you need to modify any future project by adding more peripherals, you can add signals to your top-level Verilog file for that peripheral and directly assign the pins in the .qsf file according to tables in the above-mentioned pdf.

What to Code

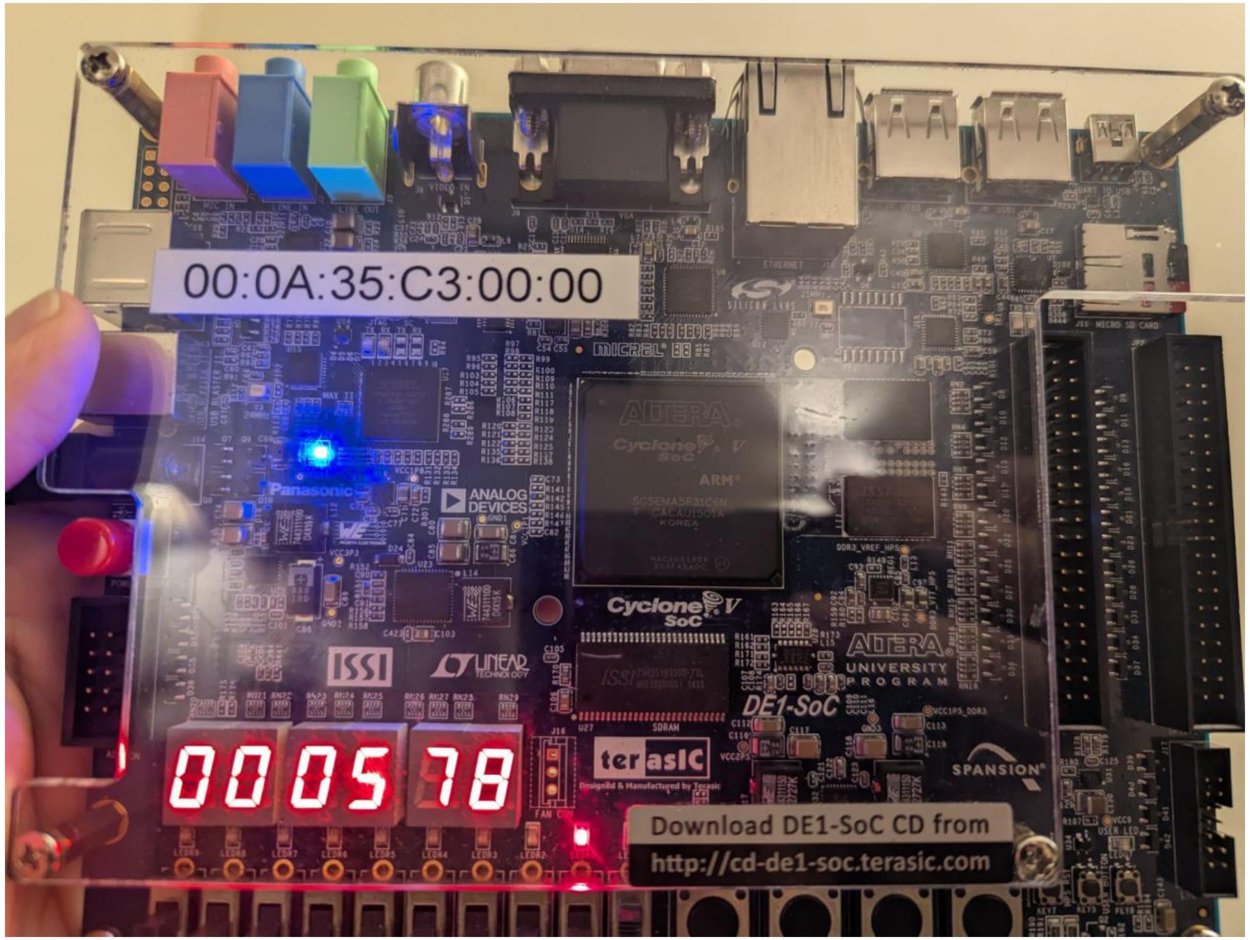
- For this lab, you will be writing the code for the fifo and multiply-accumulate (MAC) units in Verilog/SystemVerilog. It will help you jog your memory if you have forgotten how to code in Verilog/SystemVerilog.
- Fill up the provided *fifo.sv* and *mac.sv* files with your code
- The purpose of the logic that you will be helping in completing is to perform a dot product of two arrays. The result of the dot product is stored in an output register that is displayed in **hex** format on the **7-segment display** when the switch **SW0** is turned **ON**. **LED1** should also turn **ON** indicating that the logic is in **DONE** state.
- Write a testbench and simulate your code in QuestaSim

Compile code and program the FPGA

- After completing the coding, compile the project in Quartus, and observe the progress of the compilation steps from the bottom-left. Full compilation may take some time to go through place&route stage.
- You will then find the generated "Minilab0.sof" file under your project (Minilab0) folder. This file contains the bitstream to download into the FPGA to program it.
- From the DE1-SoC files, locate the following file

DE1-SoC_v.5.1.1_HWrevF_SystemCD\UserManual\My_First_Fpga.pdf and carefully follow chapter 4.2 to download the sof file into the FPGA.

Sample Demo


00:0A:35:C3:00:00
000578
Download DE1-SoC CD from
http://cd-de1-soc.terasic.com

Use Quartus IP Catalog

- Quartus provides a catalog of different IPs that can be configured and used instead of custom Verilog code. The purpose of these IPs to provide users generic building blocks to reduce coding effort for designing systems.
- To access the IP catalog, go to **Tools -> IP Catalog** and search for an IP.
- For this lab, you will be replacing the fifo and MAC units that you have written in Verilog with a fifo, multiplier and an adder IP (although there exists a Multiply-Accumulate IP, unfortunately Quartus Prime Lite Edition does not provide it).
- You will be using the **FIFO**, **LPM_MULT** and **LPM_ADD_SUB** IPs. Search for them in the catalog. (Feel free to experiment with other IPs)
- You will have to change the top-level code to support the IPs

Use Quartus IP Catalog

- A separate window should pop up for each IP. Name the IP and click **OK.** A **MegaWizard Plug-In Manager** window will pop-ip. Configure the IP according to the requirements of the design for this lab. Consult the documentation for each IP to do so. (Select separate clocks for reads and writes to the fifo during configuration but use the same clock when you make the signal connections)
- The documentation can be found by clicking on the “**Documentation**” button in the IP configuration **pop-up window**. **Reading documentation is a necessary skill for any Engineering job and will be useful if you want to succeed in this course.**
- Configure the IP by clicking “**Next**” and selecting the options for that IP. Then click “**Finish**” and “**Yes**” to add the IP files to your current project.

Use Quartus IP Catalog

- Go to “**Project Navigator**” window and select “**Files**” from the drop-down menu. You should see a **.qip** file under **Files**.
- Open the **.v** file under the **.qip** file. This file has the interface to your IP which you can use to instantiate the IP block in your code.
- Compile and simulate your code in QuestaSim.
- The following paths are to the IP libraries that you will be using for this lab: **C:/intelFPGA_lite/23.1std/questa_fse/intel/verilog/220model** and **C:/intelFPGA_lite/23.1std/questa_fse/intel/verilog/altera_mf.** (For those who are interested, all these paths are specified in the **modelsim.ini** file.)
- To run the simulation, type this command in QuestaSim:
  - **vsim -L C:/intelFPGA_lite/23.1std/questa_fse/intel/verilog/altera_mf -L C:/intelFPGA_lite/23.1std/questa_fse/intel/verilog/220model -vopt work.testbench_tb -voptargs=+acc**
- Program the board following the steps discussed in the previous slides.

Submission

- Submit the .v and .sv files to the dropbox for Minilab0 on Canvas (1 submission per team)
- Click on "**View Report**" from the drop-down menu of "**Analysis and Synthesis**" from the Tasks window. A window with the title "**Compilation Report**" should appear inside the Quartus window. Right click on the "**Resource Usage Summary**" inside the "**Analysis and Synthesis**" folder inside that window and export the file.
- This should be done after synthesis for both the halves of the lab. Save the file as **Minilab0_Resource_Usage_Summary1.rpt** for the first half (without IP) and as **Minilab0_Resource_Usage_Summary2.rpt** for the second half (with IP). Submit these files too. (1 submission per team)
- Create a **zipped file** with all the submission files and name it as **Minilab0_<member1 netid>_<member2 netid>.zip.**

Demo and Report

- You must demo your working designs (with and without IPs) to the instructor (George) or the TA (Abhishek) by the end of lab period on 01/28 to avoid any penalty. Demos should be done in teams. See Grading for more details.
- Create a repository with the name "**ECE554SP25_Minilab0**" on **Github**. Upload all .v and .sv files to your individual Github accounts. If you don't have a Github account yet, please create one and become familiar with Git commands.
  - Github: Getting Started
  - Git Cheat Sheet
- The report should be submitted individually and should have your **Github repository info** and details about how you created that the repository and added the files. It should also have your **simulation log** and a **snapshot of the waveform** from QuestaSim. In addition, discuss briefly about the differences in resource utilization between the two submitted resource usage reports.

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON