Introduction / Ramp-Up Workshop

Lab 01

Table of contents

01

Introduction to Course.

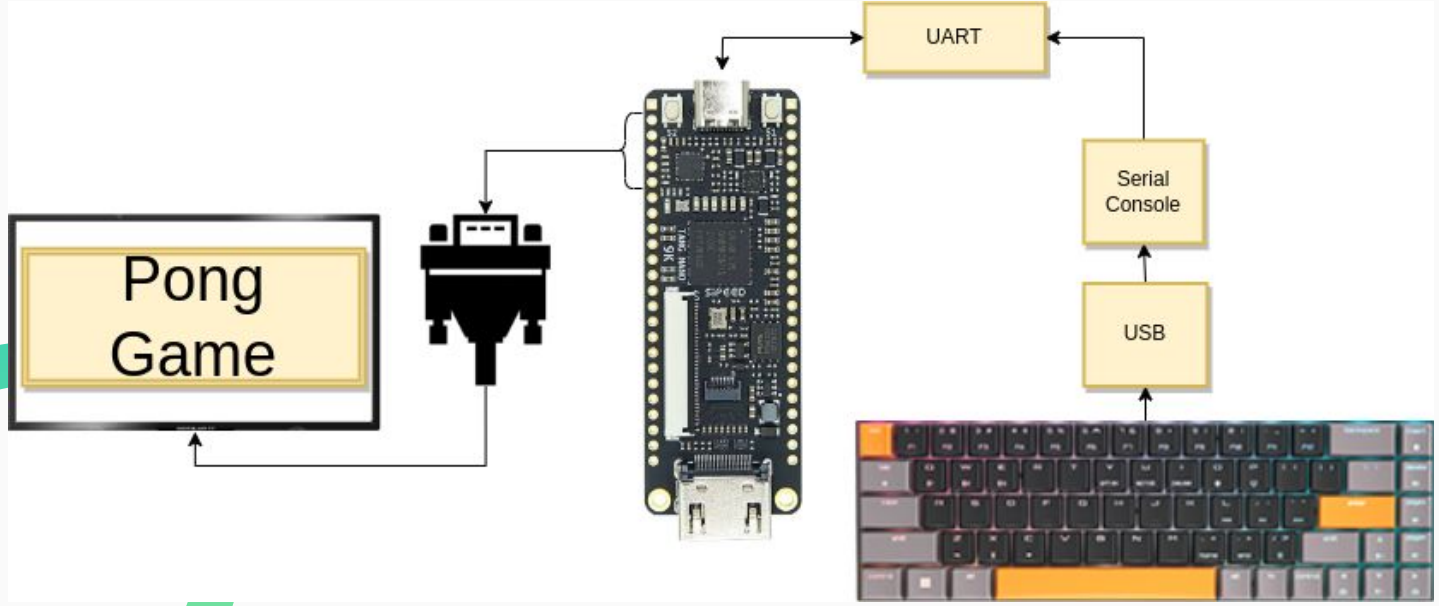
Big Picture Overview Of the Course

Introduction

Welcome to our beginner-friendly course on hardware design and FPGA programming! This program is tailored for those eager to learn the essentials of creating hardware designs and effectively porting them onto FPGA (Field Programmable Gate Array) platforms. Central to this learning experience is the indispensable role of Lushy Lab, providing a user-friendly VS Code extension that facilitates seamless hardware design and FPGA testing.

Throughout this course, participants will embark on a comprehensive learning journey. Starting with the setup of the Lushy Lab extension, learners will progress through the fundamentals of Digital Logic Design (DLD), gaining a solid foundation for the subsequent exploration of Verilog, a hardware descriptive language crucial for expressing complex hardware designs. Moving beyond theory, students will engage in practical hands-on experiences, crafting their own games from scratch, including popular titles like Pong. The curriculum also encompasses interfacing with peripherals like VGA (Video Graphic Array) for screen display and keyboards for user control.

System Context Of Project



02
Getting Setup For Lushay Lab

Getting Setup Of Lushy Lab Extension On VS Code.

Setup Guide

Step 01: OSS-CAD-Suite

- OSS-CAD-Suite is a project managed by the yosys
- enabling users to visually set up and execute the operating system toolchain.

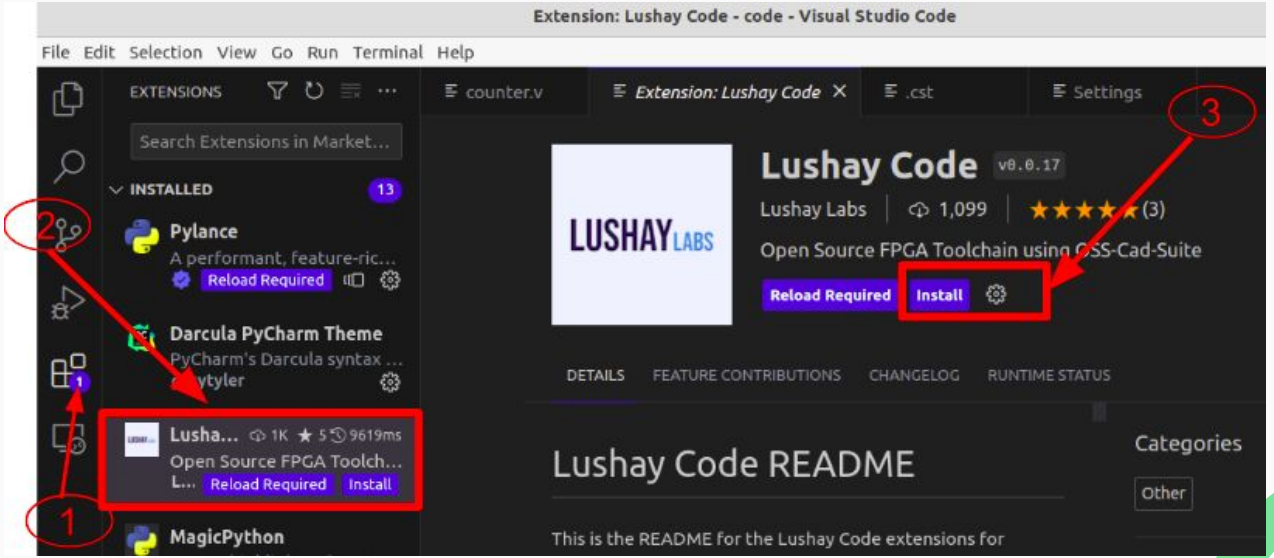
OSS-CAD-Suite Installation Link:
https://github.com/YosysHQ/oss-cad-suite-build/releases/download/2023-04-06/oss-cad-suite-linux-x64-20230406.tgz

- You can use the above link for OSS-CAD-Suite installation.

Setup Guide

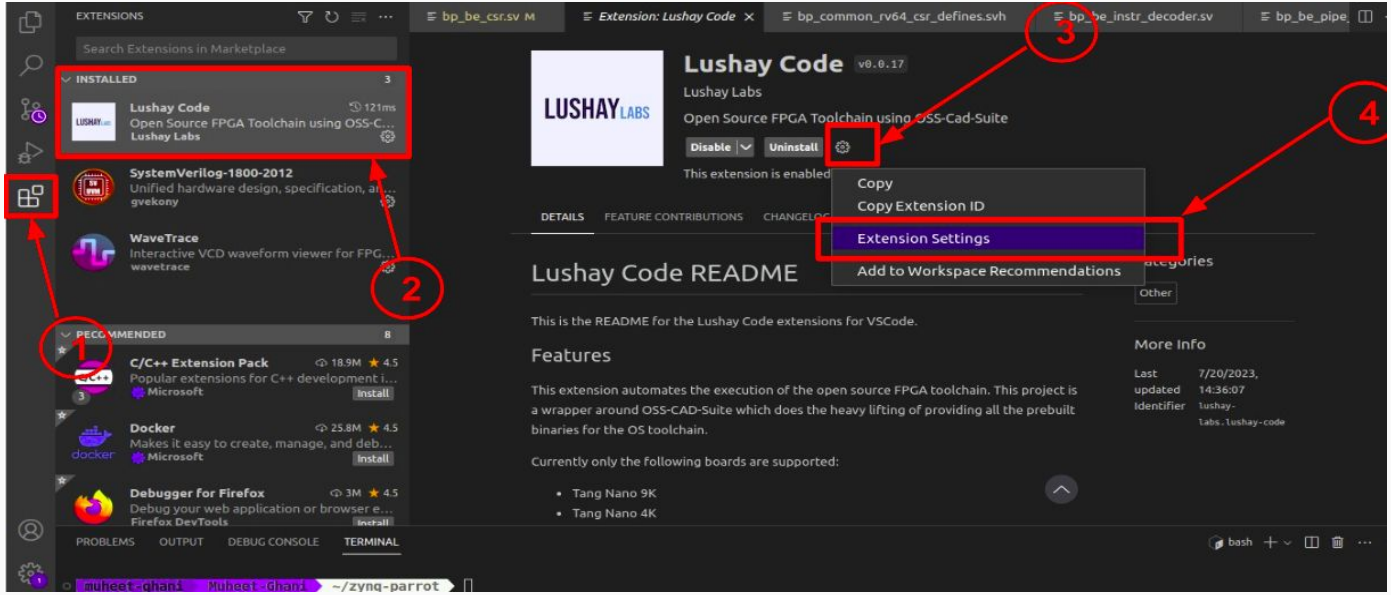
Step 02: Lushay Lab Extension Installation In VS Code

- Second step is to open up the VSCode and go to the Extensions tab and install the Lushay code.

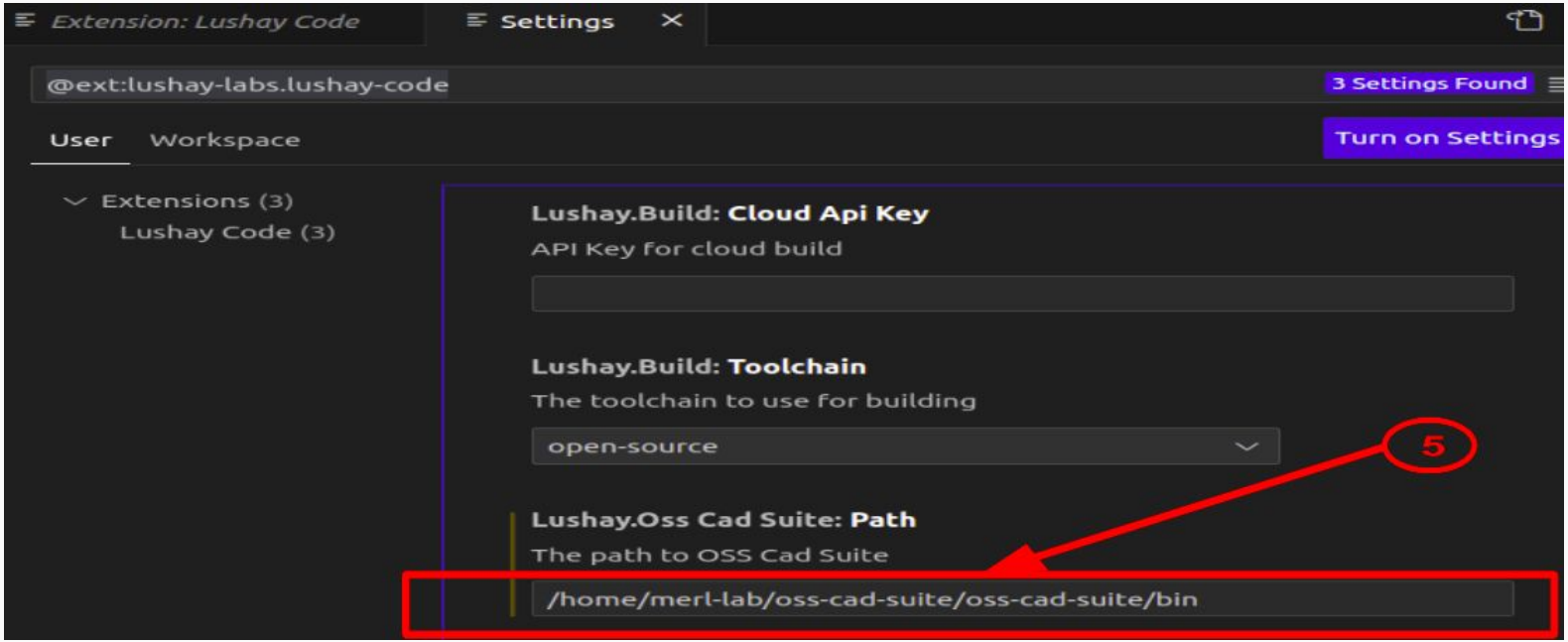


Setup Guide

Choose the recently extracted OSS-CAD-Suite folder specifically, the folder named ”oss-cad-suite” containing the ”bin” subfolder.



Setup Guide


Extension: Lushay Code
Settings
@ext:lushay-labs.lushay-code
3 Settings Found
User
Workspace
Turn on Settings
Extensions (3)
Lushay Code (3)
Lushay.Build: Cloud Api Key
API Key for cloud build
Lushay.Build: Toolchain
The toolchain to use for building
open-source
5
Lushay.Oss Cad Suite: Path
The path to OSS Cad Suite
/home/merl-lab/oss-cad-suite/oss-cad-suite/bin

03

FPGA Porting.

Implement Logic with VS code Lushy lab Extension
and Ported on FPGA

FPGA Porting

Step 01: Write Verilog Code

Make the folder i.e Counter and create verilog file i.e counter.v. and copy the below just for the reference.

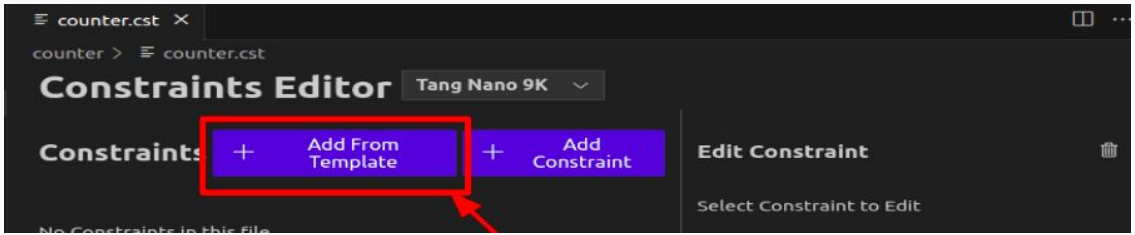
```
module top
   (
       input clk,
       output [5:0] led
   );
   localparam WAIT_TIME = 13500000;
   reg [5:0] ledCounter = 0;
   reg [23:0] clockCounter = 0;

   always @(posedge clk) begin
       clockCounter <= clockCounter + 1;
       if (clockCounter == WAIT_TIME) begin
           clockCounter <= 0;
           ledCounter <= 6'b111111;
       End
   end
   assign led = ~ledCounter;
endmodule
```

FPGA Porting.

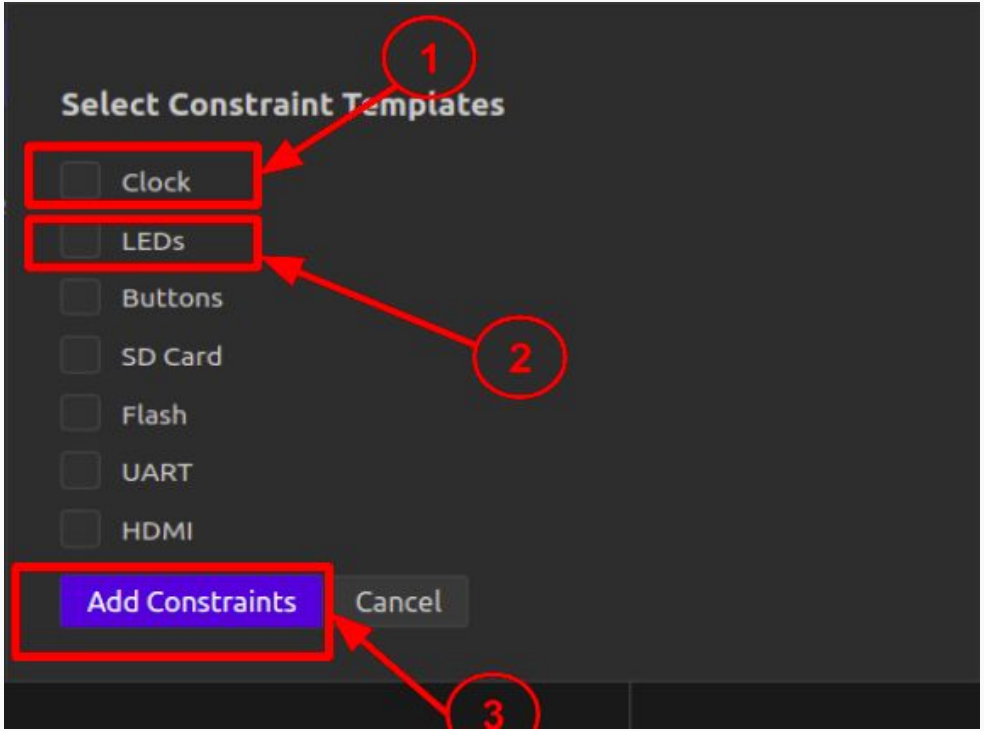
Step 02: Create .cst file

When .cst is created on VS Code. this type of interface will be open for constraints. Now click on “Add From Template”.



FPGA Porting

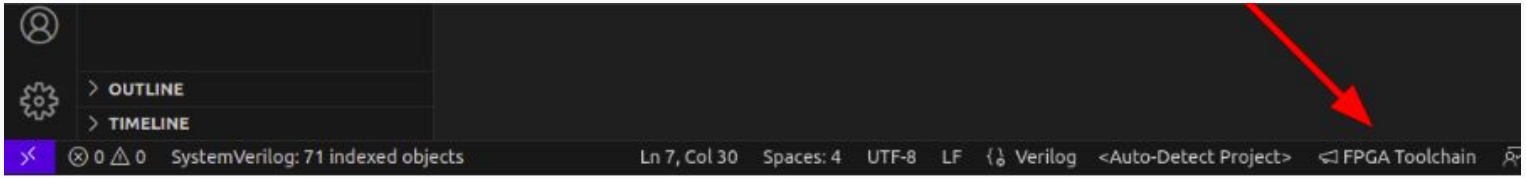
Next you have to select clk, and led and then click “add constraint”.



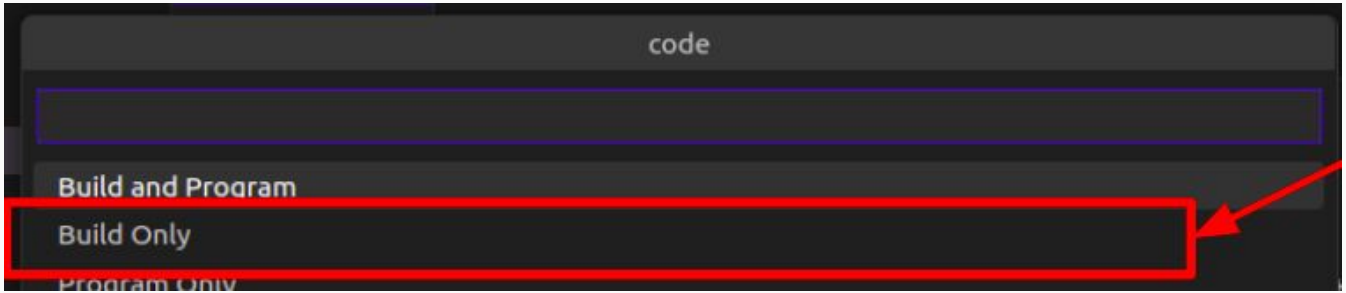
FPGA Porting.

Step 03: Synthesis And PNR

Now click on the bottom right hand side called “fpga toolchain”



Now click on “build only” which starts synthesis and pnr



FPGA Porting.

Step 04: Program FPGA

Now After completing the “build only” step successfully bitstream have generated with .fs extension file now it’s time to program onto the FPAG. Now again click on the bottom right side “FPGA toolchain” button and hit the “program only” option just below the “build only” option.

If it show’s the error message of the “FTDI port not found” then run this command on terminal.

```
curl -sSL
https://raw.githubusercontent.com/lushaylabs/openfpgaloader-ubuntufix/main/setup.sh | sh
```

04

Exercise.

Exercise Task.

Exercise.

Now Add button on the counter logic and make the counter which depend on the button.

Testimonial

Author:

Abdul Muheet Ghani, Research Assistant at MERL-UITU.

Under The Supervision Of:

Dr.Ali Ahmed (Team Lead MERL).
Sajjad Ahmed (Research Associate)

Sponsored By:

Edmund from Symbiotic EDA. for sponsoring FPGA.

Thanks:

Lushay Lab (They've provided crucial resources and guidance throughout the project), Also thanks to Slidesgo.

Future Work

This is version 1.0 of our course. We will continue this training and very soon release further versions. For any questions or to stay connected with us.

Github: Abdul Muheet Ghani.
Gmail: Dr.Ali Ahmed. Sajjad Ahmed. Abdul Muheet Ghani.