Accessing the GPIO core using Vivado:

Our idea was to look for a method which enables us to access the GPIO core(LEDs and Switches) which can help us in interfacing some sensors with our system.
For this purpose, we tried using a tutorial from Xilinx whose aim was to access the RGB LEDs using Vivado.

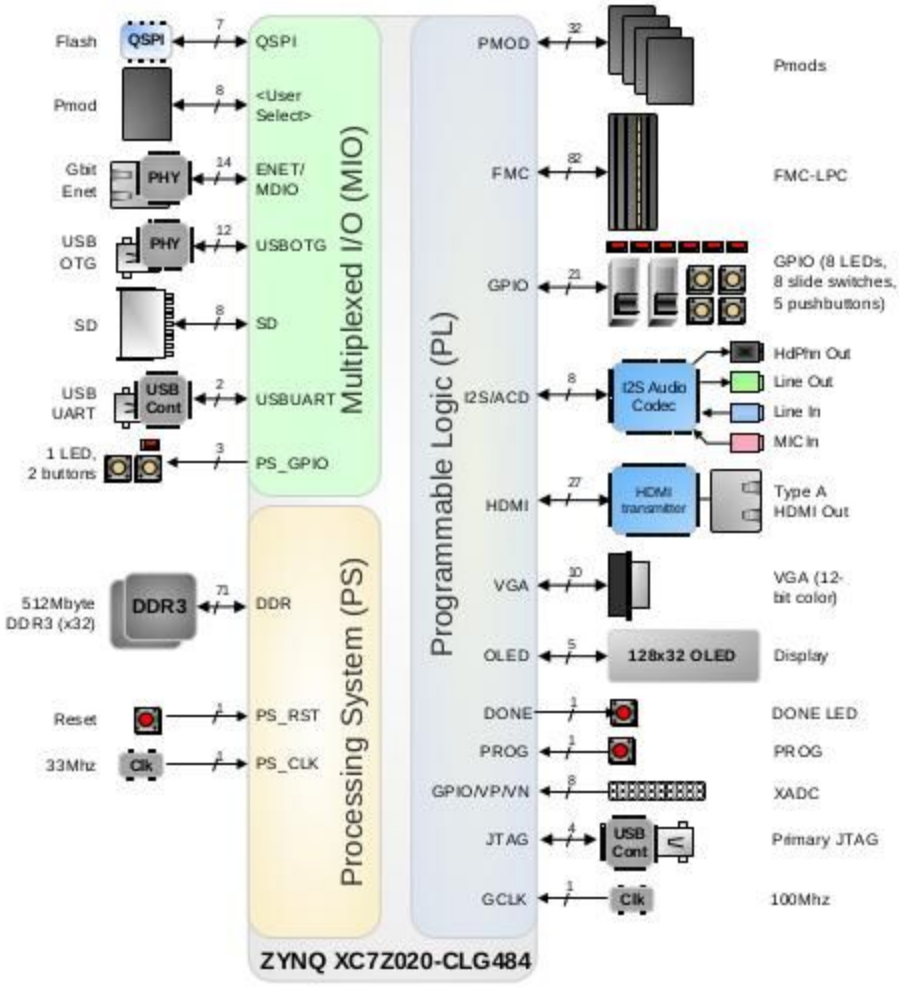

Figure 1 – ZedBoard Block Diagram

Summary of the steps followed:
I. Vivado

- Open Vivado
- Create a new block design
- Add the Zynq core IP and automate it
- Add the DigiLEDs custom IP to the project's IP repository
- Add the DigiLEDs IP to the design and configure it.

- Validate and save block design
- Create HDL system wrapper
- Run design Synthesis and Implementation
- Generate Bit File
- Export Hardware Design including the generated bit stream file to SDK tool
- Launch SDK

Now the Hardware design is exported to the SDK tool. The Vivado to SDK hand-off is done internally through Vivado. We will use SDK to create a Software application that will use the customized board interface data and FPGA hardware configuration by importing the hardware design information from Vivado.

II. SDK

- Create new application project and select Empty Application template
- Import main.c
- Program FPGA

Problems faced:

When we tried to program with the bitstream generated, then the processor got reseted in the device and our terminal access was blocked and we were unable to run the demo.

Possible remedies:

We will have to generate boot.bin and device tree from scratch along with the bitsteam.

Arch Linux on Zedboard.

Booting Arch Linux was done before trying the Android part to increase our comfort and knowledge about the system.

Tasks Done:

1. Booted ARM Arch Linux on Zedboard successfully
2. Edited the sdboot parameter with correct addresses to boot it successfully after initial boot failed
3. Link to the Zedboard Arch Linux Project.