SYNOPSIS

Project Team Members:
Mahima Agrawal
Isha Apurva
Nainshree Raj
Karthik.S

ABSTRACT
The mini-project work is aimed at the implementation of **Memory-Mapped FIFO** which is useful to have a structured interface between the HPS which stands for hard processor system and the hardware on the FPG with the available hardware equipment DE1-SoC.The on-chip FIFO memory core buffers data and provides flow control in a Qsys system. **HPS to FPGA FIFO with feedback via SRAM scratchpad** is a step toward full FIFO serial communication between HPS and FPG.The project provides an opportunity to introduce us to real hardware implementation, building electronic devices and acquiring knowledge of the tools and skills needed to achieve it.

PROBLEM STATEMENT
To implement HPS and FPGA interfacing via memory mapping using bridges and FIFO.

OBJECTIVE
The objective is summarized as **:**Data is written sequentially into the FIFO and read sequentially such that the first data written is the first data read out and so on with the remaining sequential data.

TOOLS USED (SOFTWARE /HARDWARE)

- **Linux,**
- **PuTTy software**
- **Quartus Prime** (Use Qsys (recently rebranded as Platform Designer) to instantiate and connect HPS components)
- **DE1-SoC (**a robust hardware design platform built around the Altera System-on-Chip (SoC) FPGA, which combines the latest dual-core Cortex-A9 embedded cores with industry-leading programmable logic for ultimate design flexibility)
- **SDMMC card** preloaded with default GSRD image

PROPOSED METHOLODOGY& BLOCK DIAGRAM

• HPS logic and the FPGA fabric are connected through a series of Bridges.
• HPS supports communication with the FPGA/peripherals through the L3 interconnect, which is connected to the HPS' (DDR3) SDRAM Controller.
• Therefore, it is essential that SDRAM pins are configured correctly so that the HPS may read/write data to/from the SDRAM controller and establish communication between the L3 interconnect and FPGA.
• Once all hardware has been correctly prototyped, communication between the HPS and FPGA is programmed through a memory mapped C application.
• Memory mapping allows the CPU to view and access the FPGA's address space (containing our components) so that we may read/write information as necessary, controlling the hardware through software.
• The C application you will develop uses APIs to send write (or receive read) data to (and from) specified memory addresses.
• Once your C application is complete, a binary is generated by compiling your software on a host computer.
• The binary must be placed on the microSD card.
• This can be done also by first placing the binary on a USB drive, which will be inserted and mounted to the HPS/FPGA system. You must then copy the binary from the USB to your HPS home directory to execute the application.
• Upon execution, the HPS will communicate with the FPGA prototype based on the APIs and functionality you have coded in your C application.
• You may access and interact with the HPS/ (Yocto) Linux OS from your host computer using a serial connection (Putty or minicom or the DS-5 terminal).

BLOCK DIAGRAMS