Class Report 3: FPro System Development

Samuel Hussey

1 Introduction

In this project, the FPro system development tutorial is demonstrated in which the hardware and software for system timer, GPIO, and UART modules are derived and implemented. For each module the memory-mapped IO core is created and interfaced with the FPro bus, drivers are then implemented to access register locations and utilize the cores to create a simple LED pattern application. This application demonstrates the ability of the system timer to halt program execution for a set interval, receive input from the switch array, blink LEDs, and transmit information via UART. Source files can be found here and a video demonstration here.

2 Implementation

Implementation began with adding the MicroBlaze MCS soft-core processor IP to the project. A number of HDL files were then added to instantiate the CPU, bridge to the FPro bus, and IO subsystem. The MMIO subsystem contains the controller for interfacing with the IO bus as well as instantiations of the IO cores and their respective memory slots. The cores added include the GPI, GPO, timer, and UART MMIO cores.

Construction of each core entailed a similar procedure: a custom digital circuit is designed, the IO register map for the slot interface is determined, a wrapping circuit that complies with the slot interface specification is created, and the software driver to utilize the hardware is developed in the Vitis IDE. Each core is allocated 32 32-bit registers and a base address offset from the bridge base at 0xC0000000. These memory addresses are the locations that are directly read from and written to from the device drivers.

3 Results

Once the hardware paltform is imported from Vivado and the application project created in Vitis, the project can be built. After specifying the bit and memory map information files that inform the IDE how to properly route instructions between the CPU and custom hardware, the device is programmed. The resulting LED patterns exercised each of the IO cores by reading or writing to their corresponding registers through a bare-metal interface.