Implementation Of Signal Processing application On FPGA

Abstract

The development in technology has evolved today, to the extent that, multiple processing applications have been designed.The processing applications such as video processing, audio processing, RF signal processng, image processing etc. include processes that compute transforms of the signals, being processed.One of the transforms that is used recursively is FFT(Fast Fourier Transform).Computation of FFT samples turns out to be expensive in terms of speed, performance and throughput.A software FFT unit is most commonly used for this, and has few advantages and disadvantages.The software N-point FFTs are reprogrammable and N can be any integer, which is a power of 2.But slow computation is a disadvantage.This deteriorates the performance of the system by increasing latency.Whereas, FFT processor, is an ASIC, and can also compute FFT samples at higher speed.Although, it improves the system performance, the ASIC is not reprogrammable, or reconfigurable.Thus,an intermediate method, that provides both,reconfigurability and high throughput, is hardware-software codesign on FPGA.This improves the overall system performance.The Zynq 7000 Soc, is a combination of FPGA and embedded processor.This SoC tightly couples the programmable logic with a dual core Cortex ARM processor.It provides low latency,high throughput and cache coherent communication between programmable logic and the ARM processor. Zedboard is one such platform,and is used in the implementation of the project.The challenges faced during the implementation include the design of software driver and integration of the FFT core in the application.

Conceptual Design

The design consists an HDL core, which is generally implemented in VHDL or Verilog.As shown in figure below, the HDL IP core is connected to the system design.The easiest way to do this is the use of the AXI peripheral bus.The design and the use of this bus is explained in detail in the next section of this chapter.The system design is then connected to the linux kernel device driver.This driver is a simple C code, which is run on the ARM processor.Since it is run on the processor itself, and consists of some functions, which map to the HDL IP core, it acts as the important part of the design.

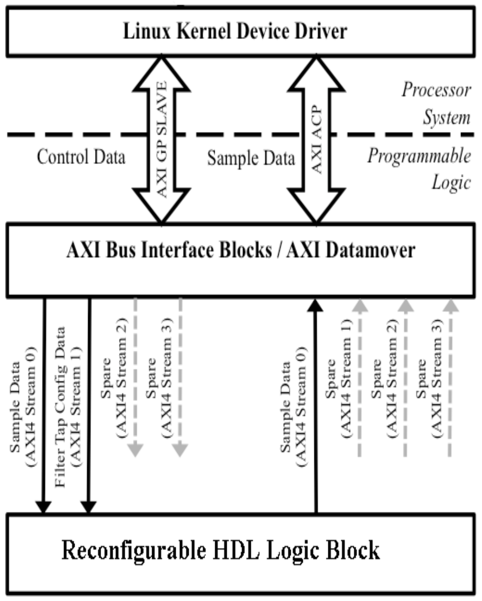
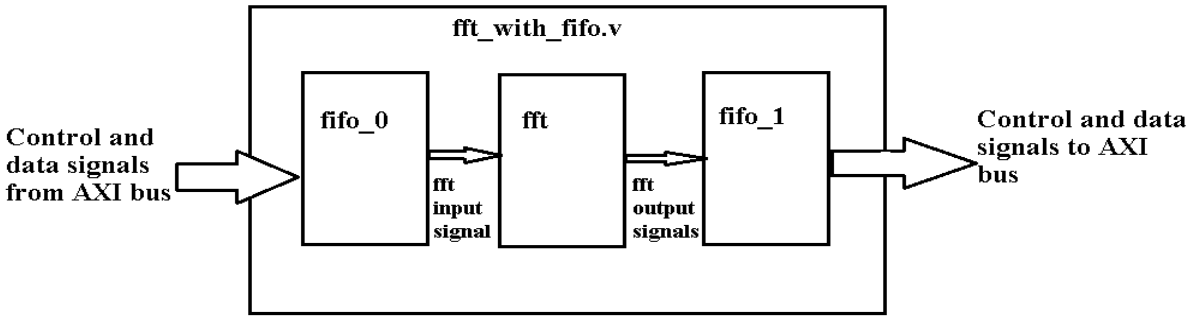

Linux Kernel Device Driver
Processor System
Programmable Logic
Control Data
AXI GP SLAVE
Sample Data
AXI ACP
AXI Bus Interface Blocks / AXI Datamover
Sample Data (AXI4 Stream 0)
Filter Tap Config Data (AXI4 Stream 1)
Spare (AXI4 Stream 2)
Spare (AXI4 Stream 3)
Sample Data (AXI4 Stream 0)
Spare (AXI4 Stream 1)
Spare (AXI4 Stream 2)
Spare (AXI4 Stream 3)
Reconfigurable HDL Logic Block

Figure : Conceptual Design

As shown in the figure above, the complete system consists of two parts:PS(processor system) and PL(programmable logic).The processor system consists of the ARM processor.It runs various processes.The linux kernel device driver is one amongst them.The programmable logic consists of the reconfigurable logic and the interface,as shown in the figure.The reconfigurable logic communicates with the AXI bus interface, through AXI4 stream bus.Also, the the AXI bus interface communicates back to the HDL logic through the same AXI4 stream bus.The PS or PL both,generate control and data signals.Both kind of these signals are transported separately between PS and PL,as shown in the figure.The AXI GP slave bus carries the control signals, and the AXI ACP bus carries the data signals.Communication over these two buses is important for the device driver to communicate with the HDL logic block.This system forms a FPGA accelerated system, and thus helps to improve the speed and performance of the PS.

The below figure shows the reconfigurable HDL logic block.


fft_with_fifo.v
fifo_0
fft
fifo_1
Control and data signals from AXI bus
fft input signal
fft output signals
Control and data signals to AXI bus

General Conclusion

The implementation of a signal processing application on FPGA has been successfully completed.The signal processing application, chosen for this purpose was FFT module, which has been successfully connected to two FIFO modules.A module formed from this coneection has been successfully mapped to the AXI periphral bus, and slave IP has been generated,and connected in the system design.A bitstream and a BOOT.BIN file for this design has been successfully generated.A device driver has also been designed to obtain the desired output from the FPGA.Thus,a FPGA accelerated FFT filter module has been implemented.From the study of three methods of programming Zedboard, it has been concluded that programming the board through a bootable SD card was the best of all.Building the BOOT.BIN file required 1s.368ms of time.Amongst three different types of AXI peripheral buses, the AXI4-Lite bus was suitable for the implemented design.It is concluded that the design has 35179 fully routed nets, with 0 failed nets.Also, all the user specified timing constraints are met.The total on-chip power was found to 3.032W with junction temperature of 59.9◦C.The total dynamic power of the design is 2.820W and the static power is 0.202W.

Future Work

Implementation of signal processing application, only generates a FPGA accelerated FFT filter module.But, this design can be improvised so that the hardware design of the FFT module, can be used to accelerate the working of the software application, running on ARM processor.For example, GNU Radio, is a software that runs on the ARM processor, and is used to design real-time radio systems.A radio system, that consists of FFT filter, can be designed in this software.By implementing FPGA accelerated FFT filter module, the speed and performance of the software radio system can be improved.The implemented work can also be used in other SDR based projects.