

Interface Techniques for Microprocessors

Embedded Within FPGAs

Joshua Noseworthy and Dr. Miriam Leeser
Department of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115
{jnosewor, mel}@ece.neu.edu

1 Introduction

As system on-chip architectures continue to receive more and more attention from the embedded systems community, FPGA manufacturers such as Xilinx are responding with a new generation of FPGA architectures that contain a variety of embedded resources. One of several recent additions to Xilinx's Virtex family architecture is the Embedded PowerPC405 Processor Core. The motivation for the introduction of the PPC405 core comes from the idea that most FPGAs contained within an embedded system require some level of interaction with an external processor. Moving this processor onto the chip allows the FPGA and the processor to communicate without the bottlenecks associated with communicating with off-chip devices. This solution raises questions on how to efficiently interface an FPGA with a processor that shares the same fabric. In this investigation, we will consider two existing interfaces that allow the PowerPC405 Processor Core to exchange information with the FPGA's surrounding fabric: the On-Chip Memory Interface and the CoreConnect Interface. The results of the investigation will be a quantitative analysis comparing the two interfaces. This analysis will be performed by examining several different implementations of a Software Defined Radio application.

2 System On-Chip

System on-chip (SoC) is a design methodology that suggests the integration of various system level components onto a single piece of silicon. System on-chip technologies aim at providing low resource consumption, low cost, and high reliability.

Compared to conventional ASICs, FPGAs have significantly lower development costs and offer comparable performance. Furthermore, the high degree of reconfigurability that is associated with FPGAs can make them attractive solutions in applications such as Software Defined Radio where the hardware platform is required to adapt to its current environmental conditions. This flexibility, coupled with the large savings in development costs, make FPGAs a popular implementation fabric for SoC designers.

FPGA manufacturers have responded to this interest in reconfigurable SoC technologies by integrating several specialized hardware cores into their FPGA architectures. One of the more recent additions that has been made by Xilinx is the integration of the PowerPC Processor core into the Virtex family architecture.

3 The Virtex-II Pro

The Virtex-II Pro FPGA was the first FPGA introduced by Xilinx that contained the embedded PowerPC405 Processor core. The Virtex-II Pro FPGA combines a sea of reconfigurable logic with an embedded processor. The device targets applications that make use of both FPGAs and general purpose processors. Implementations that make use of both of these devices could use this architecture to take advantage of the benefits that are associated with system on-chip solutions. The success of this type of architecture will require the development of interfaces that will allow efficient communication between the processor and surrounding logic blocks. The focus of this research is in the different types of interfaces and how to use them efficiently. Currently, the primary methods for moving data in and out of the processor are the processor's CoreConnect and On-Chip Memory (OCM) interfaces. Each has its advantages and disadvantages.

3.1 The CoreConnect Interface

Xilinx supports the interfacing of the processor to the surrounding fabric through its distribution of the PowerPC405 CoreConnect Architecture. The CoreConnect Architecture, seen in Figure 1, consists of three hierarchical busses: the Processor Local Bus Control (PLB), the On-Chip Peripheral Bus (OPB), and the Device Control Register Bus (DCR). These three busses provide an interconnect topology that is capable of moving data between the PowerPC405 and various devices [2]. A potential problem with the CoreConnect Architecture is that it is engineered to be a general solution to the problem of interfacing the PowerPC405 processor with the surrounding reconfigurable logic. For instance, the CoreConnect interface to the processor is a shared interface, meaning that any one of several devices could be communicating

through the interface at any given time. This type of interfacing targets general purpose applications where a processor is time multiplexing its resources between several devices. However, the servicing of multiple devices by a single processing element is characteristic of processor centric systems, of which an FPGA is not. Alternatively, a logic centric model would suggest that the processor be used for application specific processing. This type of processing does not require the ability for the processor to communicate with several devices over a shared interface. Despite this lack of need for a shared interface, there is no mechanism that allows the processor's CoreConnect interface to be configured as a dedicated interface. As a result, there will always be overhead associated with an instance of the CoreConnect interface so that it can maintain support for shared interfaces.

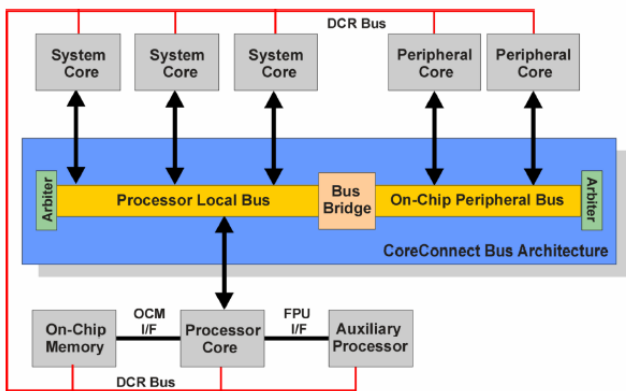


Figure 1: Diagram of CoreConnect Architecture [4]

3.2 The On-Chip Memory Interface

The processor's OCM interface is a dedicated interface that provides connectivity between the processor and the FPGA's BlockRAM[2]. Because the OCM interface is a dedicated interface, the additional logic needed by CoreConnect to resolve bus contention is not required. These factors can make the OCM interface perform substantially better than the CoreConnect interface. The problem with the OCM interface is that it has a very limited address space. Furthermore, as the amount of

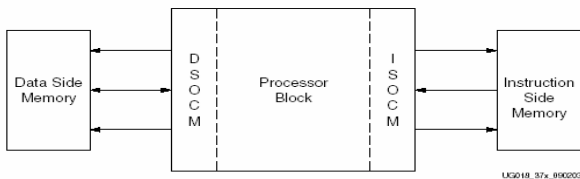


Figure 2: Diagram of On-Chip Memory Interfaces [2]

BlockRAM connected to the interface increases, the interfaces' performance drops. This performance drop is caused by an increase in the amount of routing resources that are required to connect the processor to the BlockRAM. In addition, the resources that are consumed by the increase in routing resources could make the placement of additional IP blocks more difficult.

4 Experiment and Results

To provide a means to explore the characteristics of these interfaces further, we have developed an application that uses a Virtex-II Pro FPGA to perform a subset of FM3TR Waveform processing. This is representative of the type of waveform processing that would occur within a Software Defined Radio (SDR) system. SDR applications suit the exploration of these types of FPGA architectures nicely due to the fact that the radio often requires processing that maps well to either general purposes processors or FPGAs. In many cases an SDR system will be implemented using a heterogeneous architecture consisting of both general purpose processors and FPGAs. The Virtex-II Pro provides a means for integrating the general purpose processor and the FPGAs within an SDR system onto a single chip. Assuming that the reconfigurable logic and the processor contained within the Virtex-II Pro are both capable of performing their respective processing tasks, the challenge becomes the interfacing of the processor and the reconfigurable logic. The development of this interface is the focus of this investigation.

Our application implements the modulation and digital up conversion that is required in FM3TR waveform processing. The modulation is performed using the Virtex-II Pro's embedded PowerPC processor. The modulated data is then converted from a complex digital baseband signal to real passband signals using the Digital Up Converter implemented in the FPGA's reconfigurable logic. The investigation requires that several implementations of this application exist. Each implementation uses either a different interfacing mechanism or a different memory organization. Implementations that use different interface mechanisms allow us to see how the different interfaces compare to each other with respect to their maximum bandwidth, resource utilization, and design complexity. Implementations that use different memory organizations allow us to examine factors that may influence the performance of a specific interface under a given set of conditions. For instance, how the performance of an interface is affected when both the application's data and the processor's instructions are communicated through that interface. We will present quantitative results of these studies at the workshop in September.

6 References

- [1] Gordon Brebner, Eric Keller and Phil James-Roxy. Software Decelerators. *12th Annual IEEE Symposium on FCCM*, pages 3-12, 2004.
- [2] Kraig Lund. *PLB vs. OCM Comparison Using the Packet Processor Software*. Xilinx, v1.0 edition, July 2002. XAPP644 <http://www.xilinx.com>.
- [3] Xilinx. *PowerPC405 Block Reference Guide*. 2.0 edition, August 2004.
- [4] IBM Microelectronics. *CoreConnect Bus Architecture Product Brief*. IBM, 1999. <http://www.chips.ibm.com>.