

SOCWIRE: A SPACEWIRE INSPIRED FAULT TOLERANT NETWORK-ON-CHIP FOR RECONFIGURABLE SYSTEM-ON-CHIP DESIGNS IN SPACE APPLICATIONS

Session: Networks and Protocols

Long Paper

B. Osterloh, H. Michalik, B. Fiethe

IDA TU Braunschweig, Hans-Sommer-Str.66, D-38106 Braunschweig, Germany

E-mail: b.osterloh@tu-bs.de, michalik@ida.ing.tu-bs.de, fiethe@ida.ing.tu-bs.de

ABSTRACT

Configurable System-on-Chip (SoC) solutions based on state-of-the arte FPGA have successfully demonstrated flexibility and reliability for scientific space applications like the Venus Express mission. Future space missions demand high-performance on board processing because of the discrepancy of extreme high data volume and low downlink channel capacity. Furthermore, in-flight reconfigurability and dynamic reconfigurable modules enhances the system with maintenance potential and at run-time adaptive functionality. To achieve these advanced design goals a flexible Network-on-Chip (NoC) is proposed which supports in-flight reconfigurability and dynamic reconfigurable modules. The Configurable System-on-Chip solution is introduced and the advantages are outlined. Additionally we present our newly developed NoC approach, System-on-Wire (SoCWire) and outline its performance and applicability for dynamic reconfigurable systems.

1 INTRODUCTION

Configurable System-on-Chip (SoC) solutions based on FPGAs have already been successfully demonstrated in Data Processing Units (DPUs) for scientific space applications like the Venus Express mission. This approach provides the capability of both flexibility and reliability for system design. For future space missions the demand for high performance on-board processing has drastically increased. High resolution detectors with high data rate and data volume need intensive data processing (e.g. image compression) on-board because of low downlink channel capacity. A significant advantage would also be in-flight reconfigurability and dynamic reconfigurable modules for maintenance purposes and run-time adaptive functionality. To achieve such an enhanced Dynamic Reconfigurable System-on-Chip (DRSoC) approach a flexible communication architecture is needed, which provides high data transfer rates, is suitable for dynamic reconfigurable modules as well and guarantees the qualification of the system even after a module update.

In this paper we will focus on our newly developed Network-on-Chip (NoC) architecture approach System-on-Chip Wire (SoCWire). First we will introduce the configurable System-on-chip approach for the Venus Express Monitoring Camera (VMC), successfully demonstrated in space. We will then outline the essential for a

NoC and introduce our SoCWire based architecture and outline its performance, which has been measured in a demonstration implementation

2 CONFIGURABLE SOC APPROACH

Available radiation tolerant SRAM-based FPGAs (e.g. Xilinx XQR Virtex-I, Virtex-II and the Virtex-4QV series) enable the integration of special functions (e.g. data compression) and processor system completely in a single or few high density FPGAs. The major advantages of such a system are flexibility, (re)programmability and module re-use, which can be easily adapted to specific requirements. SRAM based FPGAs are sensitive to Single Event Upset (SEU) but with dedicated SEU mitigation techniques (configuration memory scrubbing, Triple Modular Redundancy TMR) the SEU rate can be reduced to negligible or at least tolerable value. The Venus Express Monitoring Camera (VMC) is one example of such a flexible SoC approach. It is based on a “LEON-2” processor, which is provided as highly configurable VHDL model, achieving a computer power of 20 Million Instructions Per Second (MIPS) in our system. Additionally the FPGA includes all peripheral logic and interfaces to different sensors and communication units (e.g. memory controller, spacecraft RTU interface logic, 1355 SpaceWire interface controller)[1]. VMC science operation has been started in the mid May 2006. Since then VMC was switched on for an accumulated time of 6800 h, taking more than 103,000 images and is running well. So far only four non-correctable SEUs in the Xilinx FPGAs have been observed, which is in the expected range and shows the suitability of this approach for space applications.

3 REQUIREMENTS FOR FUTURE SPACE MISSIONS

VMC demonstrated a successful and space suitable configurable SoC approach. For future space mission advanced data processing needs to be done on-board. With high resolution detectors of >2Mpixel the data rate and data volume increases drastically but the average data rate to spacecraft remains at 20...60 kbps. Therefore classical ground processing steps like scientific parameter extraction and subsequent data

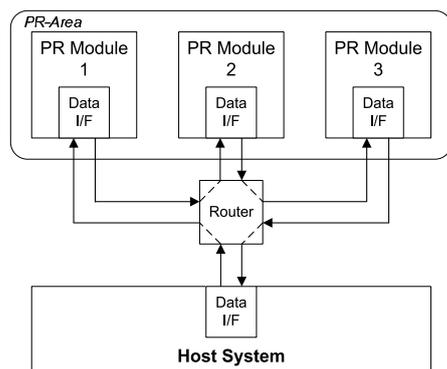


Fig.1:DPU architecture macro-pipeline system

evaluation need to be performed on-board. In-flight update of processing modules enhances the system with maintenance potential and performance improvement; an image compression core could be replaced by a sophisticated core to calculate scientific parameters directly on-board. Dynamic partial reconfiguration enhances the system with runtime adaptive functionality, which is an improvement in terms of resource utilization and power. Our framework for in-flight reconfigurability is based on the VMC approach with the additional features of dynamic partial reconfiguration. The FPGA is subdivided into static and Partial Reconfigurable Areas (PR-Areas) which can be updated during operation. The static area remains unchanged during the whole mission and comprises all critical interfaces (e.g. processor, communication interfaces to S/C). This offers the advantage that only the updated module has to be qualified in a delta-qualification step, not the whole system [2]. The architecture model we use for our instrument DPU designs is usually a macro-pipeline system with pre- and post-

evaluation need to be performed on-board. In-flight update of processing modules enhances the system with maintenance potential and performance improvement; an image compression core could be replaced by a sophisticated core to calculate scientific parameters directly on-board. Dynamic partial reconfiguration enhances the system with runtime adaptive functionality, which is an improvement in terms of resource utilization and power. Our framework for in-flight reconfigurability is based on the VMC approach with the additional features of dynamic partial reconfiguration. The FPGA is subdivided into static and Partial Reconfigurable Areas (PR-Areas) which can be updated during operation. The static area remains unchanged during the whole mission and comprises all critical interfaces (e.g. processor, communication interfaces to S/C). This offers the advantage that only the updated module has to be qualified in a delta-qualification step, not the whole system [2]. The architecture model we use for our instrument DPU designs is usually a macro-pipeline system with pre- and post-

processing steps in a dedicated structure as depicted in Figure 1. This architecture covers the typical processing chain requirements in classical instruments up to complete payload data handling.

4 PARTIAL RECONFIGURATION IN VIRTEX-4 FPGA

The Xilinx Virtex-4 Pro-V family is available as space suitable radiation tolerant FPGA. In contrast to earlier Virtex families the internal hardware architecture has changed. The FPGA is now divided into clock regions each comprising 16 CLBs, which equals one frame the smallest addressable segment of the configuration memory space. The advantage of this hardware architecture is that logic left, right, top and bottom of a Partial Reconfigurable Module (PRM) can be used for the static area [3]. Xilinx provides new un-directional Bus-Macros in the Virtex-4 family which can connect modules horizontal and vertically. These bus-macros are suitable for hand shaking techniques and bus standards like AMBA or Wishbone. As mentioned before, classically the instrument DPU architecture is a macro-pipeline system with high data rate point-to-point communication. To realize this architecture in a bus structure, multi master and bus arbitration are needed. Furthermore the partial reconfiguration process does not have an explicit activation. New frames become active as they are written. If frame bits change, those bits could glitch when the frame write is processed. Additionally some selections (e.g. the input multiplexers on CLBs) have their control bits split over multiple frames and thus do not change atomically. Therefore a fault tolerant bus structure with hot-plug ability is necessary to guarantee data integrity. With this limitation a bus structure based system would encounter the following disadvantages: (i) an SEU in the PRM bus interface logic could stop the system, (ii) failure tolerant bus structure (high efforts) is needed to guarantee data integrity and (iii) during the reconfiguration process a PRM could block the bus and stop the system. With the issues mentioned before we consider instead a networked architecture with a Network-on-Chip (NoC) approach providing: (i) reconfigurable point-to-point communication (ii) support of adaptive macro-pipeline and (iii) hot-plug ability.

5 SYSTEM-ON-CHIP WIRE (SOCWIRE)

Our approach for the NoC communication architecture, which we have named SoCWire, is based on the ESA SpaceWire interface standard [4]. SpaceWire is a well established standard, providing a layered protocol (physical, signal, character, exchange, packet, network) and proven interface for space applications. It is an asynchronous communication, serial link, bi-directional (full duplex) interface including flow control, error detection and recovery in hardware, hot-plug ability and automatic reconnection after a link disconnection. These SpaceWire features are ideal for our NoC approach.

5.1 SPACEWIRE

SpaceWire uses Data Strobe (DS) encoding and the performance of the interface depends on skew, jitter and the implemented technology. Data rates up to 400 Mb/s can be achieved. The SpaceWire character level protocol is based on the IEEE Standard 1355-1995 with additional Time-Code distribution. The character level protocol includes data character, control character and control codes. A data character (10bit length) is formed by 1 parity bit, 1 data-control flag and 8 data bits and includes

data to be transmitted. The data-control flag indicates, if the current character is a data (0) or control character (1). Control characters (4-bit length) are used for flow control: A flow control token (FCT), end of packet markers (EOP or EEP) and an escape character (ESC) is used to form higher level control codes (8-14bit length) e.g. NULL (ESC+FCT) and Time-Code (ESC + Data character).

5.2 SoCWIRE CODEC

As mentioned before, SpaceWire is a serial link interface and the performance of the interface depends on skew, jitter and the implemented technology. For our NoC approach we are in a complete on-chip environment with up to 6 reconfigurable modules, which can be operated by one switch. The maximum character length in the SpaceWire standard without time code, which is not needed in our NoC, is 10bit (data character). Therefore we have modified the SpaceWire interface to a 10bit parallel data interface. The advantage of this parallel data transfer interface is that we can achieve significantly higher data rates as compared to the SpaceWire standard.

<i>DWord Width</i>	f_{Core} (MHz)	<i>DRate [Mb/s]</i>	
		<i>Unidirect.</i>	<i>Bi-direct.</i>
8	100	800	700
32	100	3200	2800

Table 1. SoCWire CODEC data rates for given core clock frequency

Additionally, we have implemented a scalable data word width (8-128bit) to support medium to very high data rates shown in Table 1. For bi-directional data transfer every eight data character a FCT need to be included in the parallel data transfer and therefore the maximum data rate can be calculated by: $DRate[Mb/s] = f_{core}[MHz] * DWord\ Width[Bit] * 7/8$. For the unidirectional data transfer the FCT can be processed in parallel and the maximum data rate is therefore: $DRate[Mb/s] = f_{core}[MHz] * DWord\ Width[Bit]$. On the other hand we keep in our implementation the advantageous features of the SpaceWire standard including flow control and hot-plug ability. Also the error detection is still fully supported making it suitable even for an SEU sensitive environment. Furthermore link initialization time could be reduced from 19,2 μ s to 400 ns @ 100 MHz and link error recovery time from 20,05 μ s to 1,13 μ s @ 100 MHz.

5.3 SoCWIRE SWITCH

The switch enables the transfer of packets arriving at one link interface to another link interface on the switch, and then sending out from this link. The SoCWire Switch and its protocol are again based on the SpaceWire standard and is a fully scalable design, supporting data word width from 8-128bit and 2 to 32 ports. It is a totally symmetrical input and output interface with direct port addressing, including header deletion and wormhole routing. The SoCWire Switch basically consists of a number of SoCWire CODECs according to the number of ports and additional fully pipelined control machines and a cross-bar. The maximum data rate is therefore equivalent to the SoCWire CODEC. With additional SoCWire Switches and path routing the network could be extended to support even complex systems. Time Code distribution has not been implemented because it is used to synchronise a Spacewire network to a global time base. Since we are in a complete on-chip environment, each node is synchronised with the system clock and therefore Time Code distribution is not needed.

6 TEST AND RESULTS

We have implemented four SoCWire CODECs, one in the Host system, three in the PRMs and one SoCWire Switch in a dynamic reconfigurable macro-pipeline system, see Figure 1. The Host system and SoCWire Switch were placed in the static area and the PRMs in the partial reconfigurable area. All SoCWire CODECs were

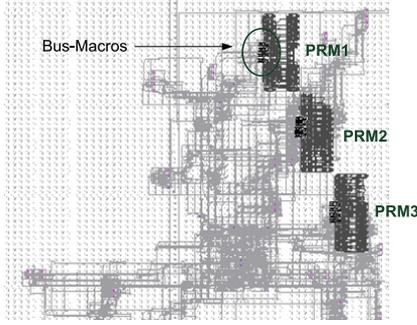


Fig. 2 SoCWire Macro-Pipeline System

configured with an 8 bit data word width. The implementation of the system with reconfigurable areas could be easily implemented with the standard unidirectional Xilinx Bus-Macros. Figure 2 shows a cut out of the placed and routed SoCWire macro-pipeline system: the PRMs (PRM1, PRM2 and PRM3) and Bus-Macros in a Virtex-4 LX 60. The static area is distributed over the FPGA. The PRMs were configured as packet forwarding modules. We have tested different configuration of packet forwarding e.g. between modules, through the whole macro-pipeline system, under the condition

of parallel communication between nodes. The system runs at 100MHz and the maximum data rates of the simulation could be validated to be 800 Mbps. We dynamically reconfigured one PRM in the system. During the reconfiguration process the communication between the PRM and SoCWire Switch was interrupted, the other PRMs connections were still established. After the reconfiguration process was completed the communication between the two nodes was built up automatically without any further external action (e.g. reset of node or switch). This makes the system ideal for dynamic reconfigurable systems. The Partial Reconfiguration Time (PRT) can be calculated by:

$$PRT[s] = \frac{PRM[Bytes]}{CCLK[Hz] \times SelectMap_{DWord Width}[Bytes]}$$

The size of one PRM was 37912 Bytes (64 Bytes command + 37848 Bytes data) and therefore the PRT 758 μ s (SelectMap, 8Bit data word width at 50Mhz). For this test system the area for one PRM was set to utilize 0.6 % of the logic resources.

7 CONCLUSION

Configurable System-on-Chip Data Processing Units based on state-of-the art FPGAs are a proven solution for space applications. For future space applications the demands for high performance on-board processing increases enormously. Additionally, in-flight reconfigurability and dynamic reconfiguration is a further enhancement to support update of hardware functions on-board even on demand. To meet these requirements an enhanced architecture with a NoC approach is needed. In this paper we presented our NoC approach SoCWire. SoCWire meets all requirements for a high speed dynamic reconfigurable architecture. High data rates are achieved with significantly small implementation efforts. Hardware error detection, hot-plug ability and support of adaptive macro-pipeline are provided. The high flexibility of the reference design allows the designer to adapt the SoCWire system quickly to any possible basis architecture. Additional SoCWire Switches can be implemented to extend the network. With SoCWire and PR-Areas system qualification can be guaranteed (with PRM self-testing features).

8 REFERENCES

- [1] B. Fiethe, H. Michalik, C. Dierker, B. Osterloh, G. Zhou, “Reconfigurable System-on-Chip Data Processing Units for Miniaturized Space Imaging Instruments” Proceedings of the conference on Design, automation and test in Europe (DATE), pp. 977-982, ACM, 2007, ISBN 978-3-9810801-2-4
- [2] B. Osterloh, H. Michalik, B. Fiethe, F. Bubenhausen, “Enhancements of reconfigurable System-on-Chip Data Processing Units for Space Application”, AHS’07. pp. 258-262, Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007), Edinburgh, August 2007
- [3] Xilinx, Virtex-II Pro and Virtex-II Pro X Platform FPGAs: Functional Description, www.xilinx.com, September 2005
- [4] ECSS, *Space Engineering: SpaceWire–Links, nodes, routers, and networks*, ESA-ESTEC, Noordwijk Netherlands, January 2003, ECSS-E-50-12A