PERFORMANCE OF THE STANDARD FAIR EQUIPMENT CONTROLLER PROTOTYPE

Stefan Rauch, Ralph C. Bär, Wolfgang Panschow, Matthias Thieme, GSI, Darmstadt, Germany

Abstract

For the control system of the new FAIR accelerator facility a standard equipment controller, the Scalable Control Unit (SCU), is presently under development. First prototypes have already been tested in real applications [1]. The controller combines an x86 COM Express™ Board and an Altera Arria™ II FPGA. Over a parallel bus interface called the SCU bus, up to 12 slave boards can be controlled. Communication between CPU and FPGA is done by a PCIe link. We discuss the real time behaviour between the Linux OS and the FPGA Hardware. For the test, a Front-End Software Architecture (FESA) class, running under Linux, communicates with the PCIe bridge in the FPGA. Although we are using PCIe only for single 32 bit wide accesses to the FPGA address space, the performance still seems sufficient. The tests showed an average response time to IRQs of 50 μs with a 1.6 GHz Intel Atom CPU. This includes the context change to the FESA user space application and the reply back to the FPGA. Further topics are the bandwidth of the PCIe link for single/burst transfers and the performance of the SCU bus communication.

DESCRIPTION OF THE SCU

- SCUB connections with LVDS and PCIe
  - strict Master/Slave bus
  - 16Bit data bus, 16Bit address bus
  - in addition serial LVDS and PCIe channels
- COM Express™ module Type II with Intel Atom
- 32Mbyte Parallel Flash
- RJ45 Gigabit Ethernet from COM Express™ module
- two SFP connectors
  - Timing (White Rabbit over GbE)
  - AUX LAN (GbE)
- Arria™ II FPGA
- White Rabbit Receiver (FPGA) [2]
- 128Mbyte DDR3 memory
- IO Extension for existing timing system
  - DEV-BUS: GSI fieldbus
  - Event-In: Timing Events
  - both based on MIL-STD-1553
- USB 2.0
- 2x EIA-232

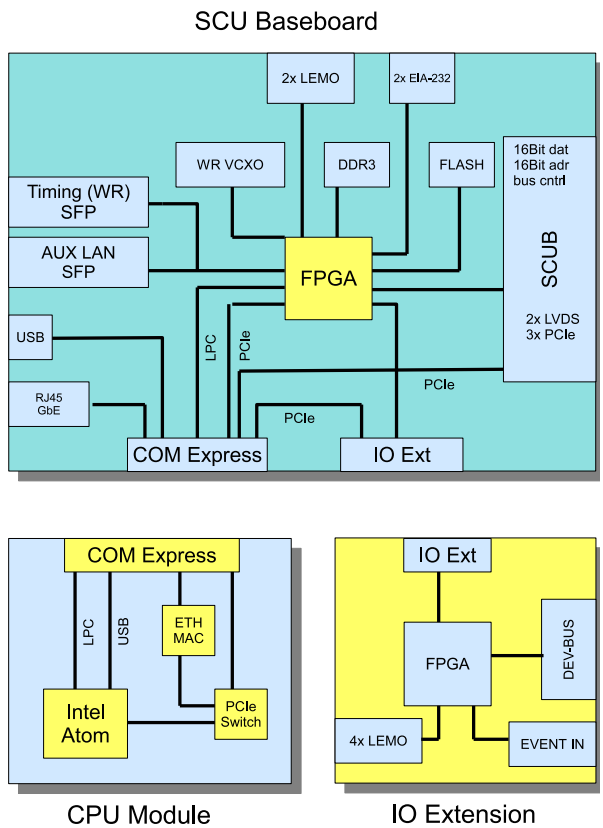

Figure 1: Layout of the SCU Carrier Board.

STATUS REPORT OF THE PROJECT

The prototype should have been ready in July 2011, but to complications during the design phase it came to a delay. The layout for the prototype baseboard is now finished and ready for production (see Figures 1, 2, 3). The first boards are expected to arrive in November 2011.

Copyright © 2011 by the respective authors — cc Creative Commons Attribution 3.0 (CC BY 3.0)

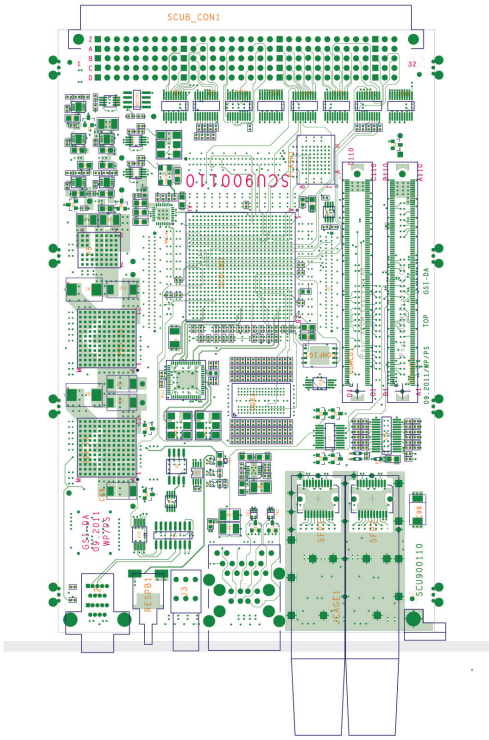

Figure 2: Layout of the SCU Carrier Board.

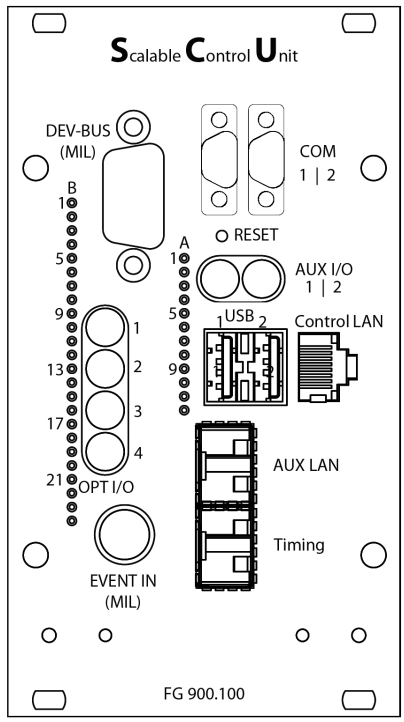

Figure 3: Front Panel of the SCU Cassette.

PROBLEMS DURING DESIGN PHASE

Early during the design phase came the decision for the combination of an FPGA and an COM Express™ module. The connection of the Module requires an FPGA with PCIe capability. The decision fell for the Altera Arria™ II FPGA with high speed serial transceivers. The Arria™ FPGA comes with a hard IP cell for PCIe so no license fees are needed. This FPGA and the design tools were quite new, when the testing of base components of the system like PCIe or DDR3 communication began. We had the possibility to test these technologies in another project and realized some problems with the design tools. In FPGA prototyping you typically implement the design first in the FPGA due to constraints in the pinning of the PCB. In this special case, the design tool allowed the hard IP core to be connected to the transceiver block 1 of the Arria™. The problem is, there is no hardware connection in the FPGA and the PCBs did not run PCIe with the hard macro cell. One possibility to solve this problem was the use of a soft core implementation. This would result in high license fees and unnecessary loss of logic cells in the FPGA. Another temporary solution was the rewiring on the PCB. After patching the PCIe to transceiver block 0 the connection worked perfectly. Altera fixed the issue in the next version of their design tool, but for the final solution a redesign of the PCB was needed.

Another problem worth mentioning was the DDR3 controller. With the first version of the design tool we could not get an reliable system talking to an DDR3 memory. Only after applying several patches the DDR3 accesses were working. But on the PCB we used for testing PCIe, the DDR3 gave us some more trouble. The chip we used is an 16Bit wide DDR3, but only 8Bit wide accesses were successful. This issue could be solved with the newest design software.

LESSONS LEARNED

If we had to do the project again, we would plan more time for testing the new (at least for us) technologies. We were aware of the fact, that their would be problems with a new and complex FPGA like the Arria™ II and this was not the first FPGA project we did. But when dealing with FPGAs like the Arria™ II, there are a lot of constraints in placing the external connections. This is especially true for connecting the pins of the memory controllers. The same applies for the high speed serial transceivers and the clocking of these. You have to be totally sure, how you are going to use use the transceivers, because most changes cannot be done after finishing the PCB. This was a new experience for us as we have only dealt with low scale FPGAs like Altera Cyclon™ or Xilinx Spartan™ before. The vendors are aware of the problem and try to help with wizards for pin placing and constraining. During layout phase this tools can be used again to find optimal routes on the PCB.

Copyright © 2011 by the respective authors — cc Creative Commons Attribution 3.0 (CC BY 3.0)

IN DETAIL: IMPLEMENTING A LPC UART

- Problem: SuperIO Chip does not fit in Layout
- Solution: extending LPC bus to FPGA, implementing UART functionality in logic
- Low Pin Count Bus, four address/data lines, serial clk, frame signal

The COM Express™ module of the SCU needed an external Winbond SuperIO chip for the UART functionality. This is mainly needed for debugging with a serial console during the boot phase. The BIOS of the module supports up to two UARTs. During the layout phase of the board it came clear, that there is no space left on the base board for the SuperIO chip. This and the fact, that only the UART functionality of the chip is used, lead us to moving the logic into the FPGA. The communication between SuperIO and COM Express™ module uses the LPC bus [3]. This bus is a serialized version of the ISA bus and uses four address/data lines and a serial clock. It runs at 33Mhz and offers bus modes as e.g. IO read/write, MEM read/write. The bus lines are connected to the FPGA instead of the SuperIO. In the FPGA we use logic for decoding the LPC bus accesses to UART transfers. The logic modules are based mainly on existing projects from OpenCore.org: a Whishbone LPC Peripheral bridge [4] and a 16750 compatible UART [5]. If you want to emulate the behaviour of the Winbond SuperIO chip, you have to understand, how the extended addressing of the SuperIO works. The chip is connected to the IO port space of the COM Express™ module. It supports up to 11 logical devices (e.g. UARTs, printer port, MIDI). To access the configuration registers of these devices you have to enter the extended function mode. This is done with two writes of 0x87 to the base address of the SuperIO. Then you select the device number and the configuration register. After reading or updating the register content you send the value 0xAA to the base address. This mechanism has to be implemented in the FPGA to allow the BIOS to configure the logical devices, like setting a base address of the UART. The 8 UART registers of the 16750 compatible core are then mapped to the configured base address. The device can now be accessed by the BIOS code during boot up and by an operating system driver. It is possible to implement further devices in the FPGA like another UART or printer port with this scheme.

REFERENCES

[1] M. Thieme, S. Rauch, M. Zweig, W. Panschow, “Single Board Computer for Equipment Control”, ICALEPCS2009, Kobe, October 2009, TUP060, http://www.JACoW.org.

[2] P. Moreira, J. Serrano, T. Włostowski, P. Loschmidt, G. Gaderer, “White Rabbit: Sub-Nanosecond Timing Distribution over Ethernet”,
IEEE Precision Clock Synchronization for Measurement, Control and Communication 2009, pp1-5, Brescia, October 2009

[3] Low Pin Count Interface Specification
http://www.intel.com/design/chipsets/industry/lpc.htm, last visited on 5.10.2011.

[4] Wishbone LPC Peripheral Bridge
http://opencores.org/project,wb_lpc, last visited on 5.10.2011.

[5] 16750 compatible UART
http://opencores.org/project,uart16750, last visited on 5.10.2011.

Copyright © 2011 by the respective authors — cc Creative Commons Attribution 3.0 (CC BY 3.0)