THE USE OF FPGAS AS A PLATFORM FOR DISTRIBUTED CONTROL SYSTEMS*

C.Timossi[1], M. J. Chin[1]
[1]*Lawrence Berkeley National Laboratory, Berkeley, CA, USA*

ABSTRACT

With the advances in Field Programmable Gate Array (FPGA) capacity and design tools, it has become feasible to consolidate into one small package both application specific control functions and a general purpose embedded processor (complete with various network interfaces). Such a package is now capable of performing the same role in a control system as the typical bus-based system such as VME. The Advanced Light Source completed an upgrade to the electronics for the Stanford Linear Accelerator PEP2 bunch-by-bunch Transverse Feedback (TFB) system [1]. The new TFB board uses a Xilinx VirtexII-Pro XC2VP7 FPGA to receive 12-bit ADC data, perform a configurable digital filtering, and finally output a 12-bit value to a DAC; all of this runs at 238MHz. Parameter and mode setting of the FPGA feedback system are controlled through a sockets interface (or alternatively through EPICS [2]) running on the embedded PowerPC hardcore in the same VP7. The TFB board is derived from the Xilinx ML300 demonstration board; the 128Mbyte DDR interface, 10/100 Ethernet, RS-232, System ACE/200MByte micro hardrive FileSystem, and non-volatile RAM interfaces are identical to the ML300. We describe using the Xilinx Embedded Device Kit to build the FPGA hardware, followed by the generation of the VxWorks Board Support Package for the hardware design. We then discuss in detail building the VxWorks kernel, and loading EPICS on the new TFB board. Finally we discussed a new design proposed for use in the ALS control system.

EMBEDDED SYSTEM DEVELOPMENT WITH THE XC2VP7

Xilinx Tools Overview

The primary development tool suite is the Embedded Development Kit (EDK). The designer uses the EDK “hardware” tools to generate parameter files that specify which peripherals (Ethernet, RS-232, SDRAM, custom core, etc) are to be used and physical pinouts on the FPGA to which these peripherals are connected. The tools then synthesize net lists, and then call the Integrated Software Environment (ISE) to “route” the FPGA’s configurable logic cells. The EDK “software” tools take other parameter files and generate libraries to support each peripheral in the design. The designer specifies what the target Operating System (OS) will be, and the EDK will automatically generate a set of Board Support Package (BSP) files for that particular combination of peripherals. These steps complete the FPGA design.

Once the FPGA design is complete, if the target is either “standalone” (no OS) or the xilKernel OS from Xilinx, then the designer may remain within the EDK environment to generate, download, and debug programs using a Xilinx-customized GNU environment. Conversely, if using a commercial OS such as the Wind River Systems real time OS VxWorks, once a successful kernel is built and then merged with the FPGA configuration bit file to generate a loadable image (in the Xilinx System-ACE format), then the designer can proceed with software development in Tornado.

XC2VP7 Embedded Architecture Overview

The XC2VP7 contains a number of “hard” resources such as the PPC405 core, Digital Clock Managers (DCMs), 792Kbits of Block RAM (BRAM), 18x18 multiplier blocks, plus 11,088 logic cells. The configuration of these logic cells are what make up the “soft” resources. In an embedded system, logic cells provide busses for the PPC405 core, and the peripherals that connect to these busses.

* Work supported by the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Xilinx uses the IBM CoreConnect standard for bus definitions. The main busses in a PPC405 project are usually the 64-bit Processor Local Bus (PLB) and the 32-bit On-Chip Processor Bus (OPB). The PLB usually connects to high-bandwidth peripherals like DDR-SDRAM, while the OPB, which is more economical in terms of FPGA resources, is usually connected to UARTs, General Purpose I/O (GPIO), System-ACE and Ethernet. PPC405 access to these peripherals is through memory-mapped locations.

The EDK provides many no-cost Xilinx-created peripherals for these busses that can essentially be "dropped-in". Some of the more complicated peripherals such as the Ethernet controller are not free; if the designer does not have a purchased license for such a core then a fully-functional but timed usage core gets placed by ISE.

TFB PROJECT

Objectives

The original digital electronics for the Stanford Linear Accelerator PEP2 bunch-by-bunch Transverse Feedback (TFB) system were designed in 1996. In anticipation of a full ring fill, it was designed to run at 476MHz. It used 8-bit ADCs and DACs, discrete ECLinPS logic, and essentially provided a digital delay that was adjustable via thumbwheel switches; there was no computer interfacing needed. The correction kick amplitude was generated externally by the analog subtraction of the delayed value from the DAC from the current bunch position.

SLAC approached LBL in summer of 2004 to design an upgrade. Because of ring limitations, PEP2's maximum fill is every other bunch; this means that current FGPAs can easily keep up with the actual bunch rate of 238MHz. The new system would use 12-bit digitizers, with a total Effective Number Of Bits of 9.8 bits. A two-tap filter with adjustable delays and coefficients would be provided. For beam diagnostics, digitized ADC data would be continuously stored in a real-time 32MegaSample FIFO until halted by either a hardware trigger or Control System command. Interfaces to the SLAC Control System would include RS-232 and Ethernet.

FPGA Selection

The Xilinx VirtexII-Pro FPGA was chosen for several reasons. Test designs of the filter core were routed using the Xilinx ISE, and indicated that the XC2VP7-6 could comfortably keep up with 238MHz data. The XC2VP7's I/O standards, such as Low-Voltage-Differential Signalling (LVDS), on-chip 50ohm termination, and Double Data Rate (DDR) meshed well with the high-speed external digitizers. Although many vendors offer "soft-core" processors) for their FPGAs, the Xilinx VirtexII-Pro and Virtex4-FX series are unique in having a "hard-core" core embedded in the chip die. The PowerPC core (PPC405) on the XC2VP7 can run commercial operating systems such as VxWorks and Montavista Linux. Finally, Xilinx offered the ML300 demonstration board, which included a XC2VP7, Ethernet, RS-232, 128Mbytes of DDR-SDRAM, and System-ACE (a Xilinx configuration standard allowing usage of Compact Flash (CF-1 or CF-2) cards for FPGA configuration and as a file-system for the embedded system). Included on the System-ACE CF were example embedded system builds, including VxWorks, Linux, and web servers. Xilinx also made available to LBL the CAD design files for both the schematic (Viewdraw) and the layout (PADS Power-PCB).

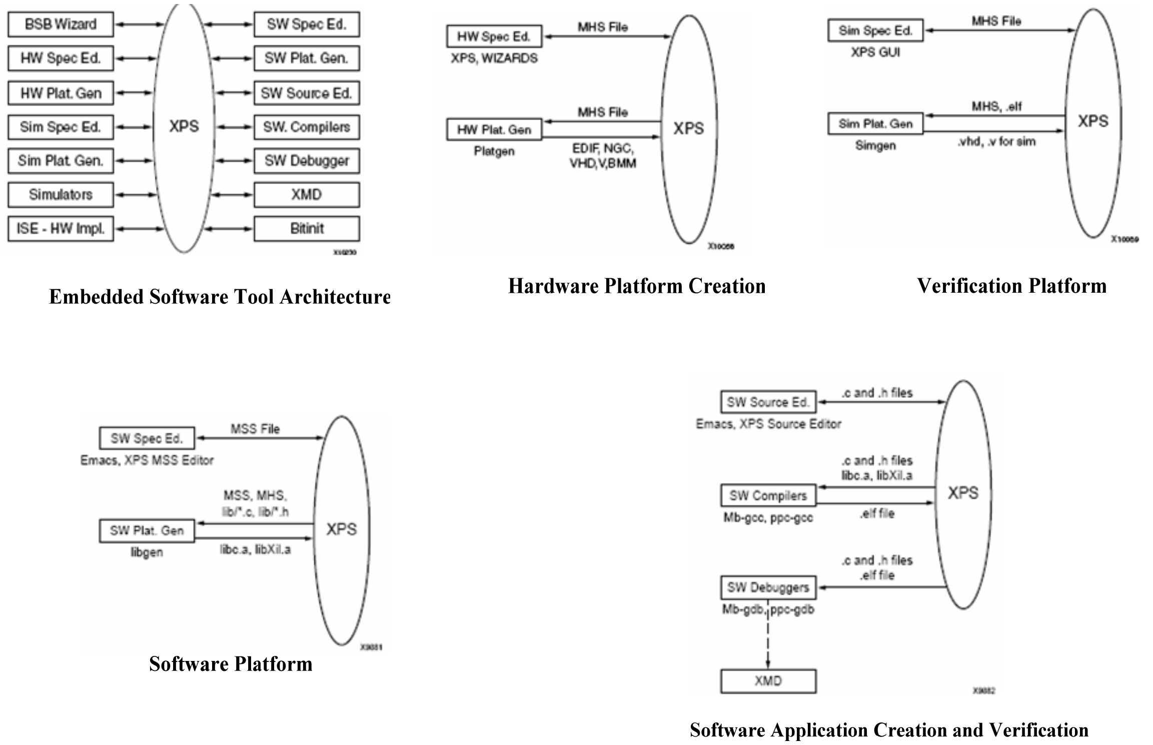
Design Process

Generation of the parameter files can be done using the GUI table-entry tools of Xilinx Platform Studio (XPS) section of the EDK, or alternatively via a simple text editor. The files themselves are only a few pages long, and fairly clear to interpret. However, it is easy to make a mistake in entering them, which in the worst case generates in a system that is functional but not correct. For the TFB board, this process was greatly simplified because it is such a close derivative of the ML300. The EDK provides the Base System Builder Tool wizard that has a library of demonstration boards including the ML300 that allows the user to generate an error-free set of parameter files for generic configurations of the demonstration boards.

By removing un-needed sub-systems (PCI interface, Rocket IO, LCD driver) from the ML300 design, sufficient board space and XC2VP7 I/O pins were made available to add the new digitizer and clocking components for the TFB board. While PCB design proceeded, the EDK was used to experiment with designing simple PLB and OPB-based interfaces on the ML300; this would enable eventual connection to the TFB filter core. Also, testing of burst-writing to SDRAM, RS-232 configuration as standard I/O (stdio), and Ethernet-based communication, and System-ACE configuration could proceed on the ML300 well before the TFB board was ready because these sub-systems were identical on both boards.

A key element to the design was the generation of a PLB interface to the ADC data stream. This interface was needed to continuously store ADC data to the SDRAM, while still giving the PPC405 access to the data after a fault trigger stopped the capture in order to send this data through the Ethernet back to the Control System. Generation of custom peripherals is made using the "IP Wizard", and is based upon the designer connecting the custom logic requirements to simplified versions of the CoreConnect busses, namely the PLB-IPIF and OPB-IPIF interfaces. Direct design for CoreConnect is apparently very demanding.

ModelSim SE 6.0 was used for system simulation, primarily to gain understanding of the SDRAM interfacing. For initial development of the TFB PLB master, the IBM Bus Functional Models (BFM) were used. The BFM allows direct stimulation of the OPB and PLB busses, allowing for much faster simulation cycles. After successful BFM modelling, the entire FPGA embedded system was simulated. This included the PPC405, external SDRAM, run-time software loaded into BRAM, and simulated ADC inputs.



Embedded Software Tool Architecture

Hardware Platform Creation

Verification Platform

Software Platform

Software Application Creation and Verification

Figure 1: XPS Tool Flows from "Embedded System Tools Reference Manual v4.2"

PowerPC Software

The embedded processor core and Ethernet soft core were included in the design to run the software needed for network access from the PEP2 control system. Since the control system uses the EPICS system, one obvious choice for integration was to run IOC core on the PPC. The ML300 kit included high level tools, instructions and examples for creating BSPs for several real-time OSs but Wind River's VxWorks seemed the best selection for hosting EPICS. It was a relatively simple process of using Wind River's Tornado development environment to develop a VxWorks kernel for the PPC. Since the OS would not fit on the approximately 100Kbytes of on-chip BRAM a VxWorks boot kernel was targeted for the off-chip SDRAM. Building the EPICS software for the PPC405 architecture was also relatively simple though it was some effort setting up the GNU compilers and upgrading Tornado before the EPICS binary would execute. The demo board includes an assortment of non-volatile storage that could be used for booting and loading software. Of particular interest is the compact flash card which can be formatted with a DOS-like file system and is already used to store the and load the System-ACE file containing the gate array logic. We decided not to pursue this path for booting VxWorks since it wasn't an option provided 'out of the box'. Instead, we configured the board to boot in the more traditional way from a VxWorks boot host.

Design Decisions

The SDRAM was intended to be used as the 32MSample FIFO. Since the SDRAM data interface is 32-bits wide, and DDR-clocked at 119MHz, 12-bit ADC data at 238MHz would seem to less than half of the bus bandwidth assuming the SDRAM could be continuously burst written. It seemed likely that FIFO writes could share the SDRAM with PPC405 program space accesses.

One way to implement this dual-access to the SDRAM would have been to reverse-engineer the no-cost EDK-provided PLB-SDRAM peripheral, insert the ADC data stream, and do the arbitration in this new peripheral. This potentially would have the highest performance; it would have also been the most work since DDR-SDRAM interfaces in general, and the ML300's in particular are complex

Another option was continue to use the PLB-SDRAM to allow PPC405 access, and then with moderate effort to create a PLB master interface for the ADC data stream. The ADC PLB master would use the burst capability of the PLB to stream ADC data to the SDRAM, and the PLB arbitration logic would resolve interleaving access for the PPC405. This option was chosen, principally because it seemed the easiest.

Unfortunately, simulations showed that the Xilinx PLB implementation has a burst limit of 16 x 64-bit packets. In addition, there is no pipelining for data setup on the PLB-IPIF. The bottom line was that the effective continuous burst write to the SDRAM was under 50% of the theoretical max, resulting in little margin for the original specification even without including PPC405 access.

SLAC relaxed the FIFO requirement so that only 8 of the 12 bits of the ADC data stream had to be stored. In addition it was decided, for architectural reasons, that it was not desirable to run the TFB board as an IOC but instead to run a simple sockets based server application on the PPC that could communicate parameters and settings to an existing IOC. Both these decisions meant that the PPC405 could run "standalone" software which would fit in the on-chip BRAMs and thus remove any time-sharing for access to the SDRAM bus.

Unfortunately, this still wasn't quite enough tinkering. The generic BSB-based design for the ML300 has the PPC405 connected first to PLB elements, and then through a PLB-OPB bridge to the OPB peripherals. An initial TFB design derivation showed occasional FIFO data corruption; this was traced to the OPB peripherals (RS-232, Ethernet) randomly interfering with the ADC PLB master bursting. This was resolved by adding an OPB-PLB bridge, and moving the SDRAM and ADC Master and their PLB segment to the "other side" of the OPB bus.

The "standalone" software uses both the RS-232 and the Ethernet ports for TFB configuration and status. A primitive command interpreter was written in "C", where the RS-232 mapped as stdio is used primarily for network configuration and status. Most of the functionality of the TFB board is controlled through a sockets based server adapted from the Memec web-server example.

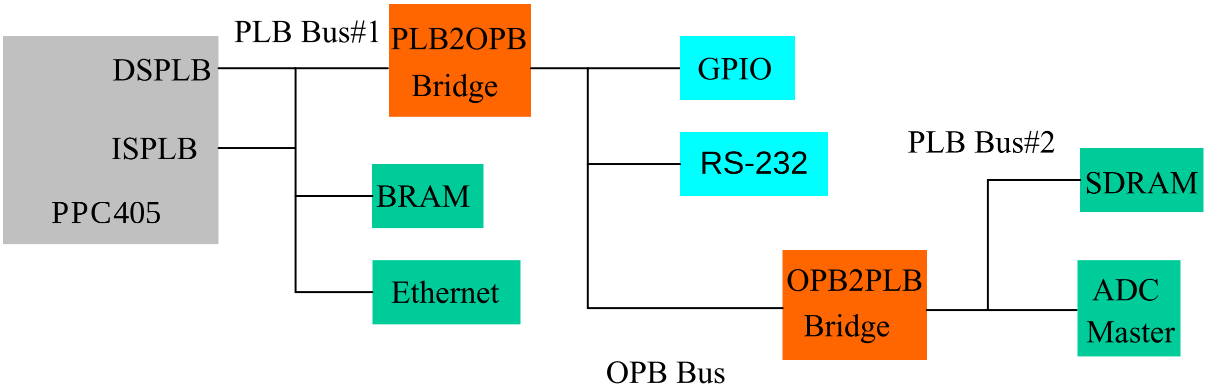


Figure 2: Final TFB FPGA Architecture

USE IN ALS COMPUTER CONTROL SYSTEM

Intelligent Local Controller (ILC)

Migration of the original ALS control system [3] to an EPICS based one is an ongoing process at the ALS. At this time most of the storage ring magnet power supply controls and some of the instrumentation has been moved to IOCs. However, the injection system-the gun, linac, and booster-are still controlled by over 200 of the original ILCs. These ILCs are highly distributed the in order to locate them in close proximity to the equipment being controlled to limit long runs of signal cabling. They were designed and built at ALS to be a general purpose controller but with the specific requirement that they have enough I/O and processing capability for controlling one Beam Position Monitor. Various designs have been investigated for replacing the ILCs as part of the general migration to EPICS but until now there has been no particular urgency.

Booster Top-Off Project

A project to fill the storage ring continuously at full energy called ‘Booster Top-Off’ is currently underway at the ALS. As part of this project the decision has been made to build a new control system for the booster. Rather than adding bus-based IOCs for this purpose, we are proposing an FPGA ILC replacement that would fit in the existing ILC enclosure. This design has several key points. First, a design that can be deployed in the existing ILC chassis allows us to preserve the cable plant; recabling is both expensive and error prone. Second the application--power supply timed ramping--is best accomplished with the aid of hardware. Finally, the new controllers would be capable of running EPICS which fits our long term plans.

Current Activities

We are evaluating a later generation of the Xilinx FPGA, the Virtex4VFX12, and associated ML403 demo board. The M403 is smaller and with some different peripherals than the ML300 but the PPC405 binaries, such as the EPICS IOC core, executed without a rebuild. The Virtex4 runs at a higher clock rate and lower temperature. The lower power requirement is an important feature since the ILC that it would replace is packaged in an almost sealed enclosure.

CONCLUSION

FPGAs with embedded processor cores, such as the Xilinx Virtex 4VFX12 with an embedded PPC, and configurable peripherals, like Ethernet, provide a flexible platform for building and deploying accelerator instrumentation. These parts offer the designer the ability to package instrument-specific logic and a general purpose processor capable of hosting an OS (such as VxWorks) together on a single IC. We have presented experience with one such design, the TFB, as well a proposal for another.

ACKNOWLEDGEMENTS

The authors thank Jonah Weber from LBNL who did the VHDL coding, ModelSim simulations, PPC405 interpreter C programming, and LabVIEW programming to test and demonstrate the sockets interface. We also thank Larry Doolittle of LBNL who suggested much of the FPGA architecture, and performed preliminary test filter routings that led to the selection of the XC2VP7.

REFERENCES

[1] J. Weber, PEP-II Transverse Feedback Electronics Upgrade, PAC2005, Knoxville, TN, USA.
[2] L. R. Dalesio, et al., ICALEPCS '93, Berlin, Germany, 1993.
[3] S. Magyary et al, NIM A 293, p36, 1990. S. Magyary, IEEE PAC'93, 93CH3279-7, p1811, 1993.