

EMPLOYING RTEMS AND FPGAS FOR BEAMLINe APPLICATIONS AT THE APS*

D.M. Kline, S.K. Ross[#], ANL, Argonne, IL 60439, U.S.A.

Abstract

At the Advanced Photon Source (APS), the power and flexibility of an Altera Cyclone-II FPGA combined with the Arcturus uC5282 embedded microprocessor running RTEMS, provides a low cost solution for implementing beamline applications.

In this paper, we discuss the approach of coupling an Altera FPGA and the Arcturus uC5282 to implement a time-resolved 32-channel scaler, development using the Altera Quartus-II design environment and the RTEMS tools, as well as an ASYN based EPICS device driver and its integration to the standard scaler record support. Furthermore, we discuss how this approach has been applied to other control system applications, such as for photon counting and flexible CCD shutter timing control.

By employing this approach, a variety of applications can be quickly developed on one hardware platform which realizes real-time performance within the FPGA and provide a cost effective EPICS IOC for exporting data to scientists and users.

INTRODUCTION

There exist many approaches for developing beamline and detector applications at various levels of complexity. The “Generic Digital” approach we employ abstracts the application behaviours and compartmentalizes them, serving as our “design pattern.”

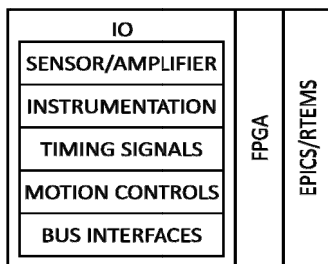


Figure 1: “Generic Digital” conceptual model.

The Input-Output (IO) component represents the specific hardware designed for the particular needs of the application. This may include inputs and outputs at different signal levels, such as NIM, TTL, LVDS, or ECL. Bus adapters, such as PC/104, can be developed to take advantage of IO modules developed in-house or commercial, such as motion controllers and input sensors. Typically, the IO component connects to an in-house developed or purchased transition board, which then connects to existing infrastructure.

* Use of the Advanced Photon Source at Argonne National Laboratory was supported by the U. S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357.

[#]skross@aps.anl.gov

The FPGA component includes in-house developed base boards as well as commercially available boards, such as Altera’s Cyclone or Stratix development kits. It handles the real-time aspects of the application and serves as a mediator between the IO and EPICS components.

The EPICS [1] component uses the Arcturus uC5282 [3] microcontroller as the IOC and runs the RTEMS [4] real-time operating system. It connects to the FPGA using carrier boards developed at the APS. ASYN [6] drivers are written to interface with the IO hardware. Furthermore, EPICS databases and MEDM screens have been included to implement the application behaviour and user interface.

HARDWARE

Generally, both in-house developed and commercially available hardware is employed within the model. The FPGA component uses two generations of hardware that was developed in-house. More recent versions use development kits offered by Altera.

Generation I

The generation I (GEN-I) base board consists of Altera’s FLEX10K FPGA. This implementation is targeted for less complicated applications that don’t require many logic elements and requires frequencies lower than 50MHz, such as “divide-by” circuits, combinational logic consisting of a few logic gates, or simple scalars.

Communication between the IOC and FPGA is through an in-house developed Serial Peripheral Interface (SPI). Typically, this is implemented on a Linux based system, such as an EPICS brick (EBRICK) [6]. The EBRICK is a Poseidon Single Board Computer offered by the Diamond Systems Corporation [5]. The FPGA is housed in a 1U rack mountable chassis and uses an external wall mount or desktop power supply.

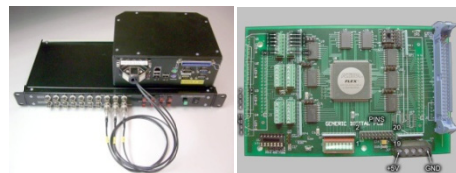


Figure 2: GEN-I/w EBRICK (left) and base board (right).

Generation II

The generation II (GEN-II) base board uses an Altera Cyclone-II FPGA [2] in a 3U VME footprint. It has IO for the uC5282, 24bits TTL IO, and 40bits LVTTTL expansion IO for additional daughter boards. This implementation is targeted at higher-end applications that require more logic elements and frequencies greater than

50MHz, but less than 100MHz. Some applications include a 32-channel scaler, CCD shutter timing control, bunch intensity adjustment, and bunch photon scaler.

The uC5282 is mounted in a carrier board that was developed at the APS. EPICS is implemented on the uC5282 running the RTEMS operating system. These were chosen because of licensing, hardware and software costs, footprint, usage at the APS and in the EPICS community, known working designs, in-house expertise and support, and development environment. The FPGA base board, uC5282 carrier, and IO daughter board is housed in a 2U rack mountable chassis and use a wall mount or desktop power supply.



Figure 3: GEN-II chassis (left) and base board (right).

Altera Development Kits

Recently, development kits from Altera, which have advantages and disadvantages, are being employed. One advantage is that the development effort has already been incurred, leaving us to focus more on the application and IO component. Some disadvantages are product lifetime and the footprint, particularly when there are space constraints.

This platform is targeted for high-end applications, such as detectors, that require more logic elements and IO, frequencies greater than 100MHz, and on-board memory. In some cases, the uC5282 doesn't have enough processing power to handle the interrupt frequency, and cannot transfer the required amount of data within a short time. The FPGA handles the real-time aspects as well as storage of data to local memory. The uC5282 uploads the memory in an interleaved manner, to populate EPICS waveform records.



Figure 4: Altera Stratix-IV development kit.

SOFTWARE

Software development is split between Windows and Linux-based environments. Linux has the tools to develop EPICS, synApps [6], IOC, and the RTEMS cross-compiler. The FPGA design is developed using the Windows version of Altera's Quartus-II [7]. EPICS and synApps modules are taken from a central repository shared by APS users as well as developers from various groups. This proves to be synergistic amongst developers for support and consistency.

The IOC application resides on a central server as well, but is independent from other applications. It uses symbols derived from a script provided by synApps to

reference modules and EPICS base. Our applications use the synApps EBRICK module as a departure point. "Out of the box" it supports RTEMS and the GEN-II hardware platform, including access to the uC5282 and FPGA. An ASYN driver has been written to access the common features, such as the 24bits IO. ASYN provides standard support to a variety of EPICS records. The developer doesn't need to write EPICS device support, only the driver need be written. In addition, EPICS databases and MEDM screens are available. Any specific functionality is coupled with the IOC, EPICS base and synApps modules, providing the framework and the underlying functionality required by the application.

Altera's Quartus-II design software is used to develop the FPGA logic. Interaction between the uC5282 and the FGPA is by means of the "Avalon Bus" [8]. The bus bridge, developed at the APS, mediate data transactions and interrupt processing between the uC5282 and FPGA. It gives the application visibility of the functionality provided by the FPGA and notifies it when events occur.

APPLICATIONS

The "Generic Digital" model has been applied to many applications, of varying complexity, for a number of years. Most applications use the GEN-II hardware platform which has become the "one hardware platform." The model gives flexibility allowing hardware components to be interchanged based on application complexity. It did not make sense to use GEN-II hardware when GEN-I hardware would be sufficient. Some of the GEN-II applications are discussed below.

One of these applications is a 32-channel scaler. Although commercial ones are available, there were several reasons for developing one in-house, such as lower hardware cost, ability to deploy them in beamlines without VME, and mobility between beamlines.

The primary requirement of the scaler was to behave identically to existing systems, and make use of current EPICS support and infrastructure. The GEN-II hardware was used and an IO board was developed to level shift the LVTTTL to TTL. A breakout board with Lemo connectors and LED indicators was developed that mounts to the front panel and connects to the IO board with a flat ribbon cable.

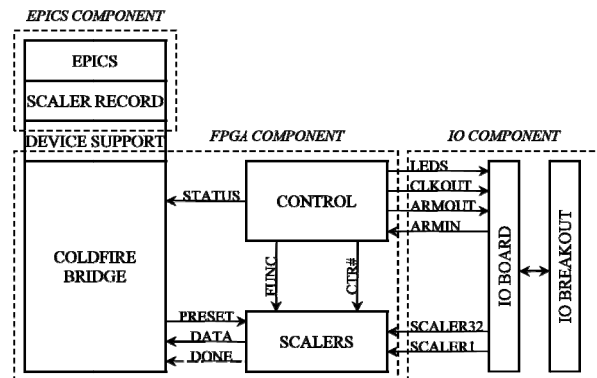


Figure 5: Scaler block diagram.

The front panel consists of Lemo connectors, a power ON/OFF switch and indicator. There is input for each scaler, and an ARMIN and GATEIN for daisy chaining with external equipment. There is a 10MHz CLKOUT output for external synchronization, and an ARMOUT output that is controllable through EPICS. Typically, for time resolved applications, the ARMOUT is connected to the ARMIN and the CLKOUT to the first scaler.

The back panel has a power input connector. Typically, an external wall mount or desktop power supply is used for electrical safety concerns. There are LEDs to indicate activity from EPICS, ARMED, ARMIN, and GATEIN.

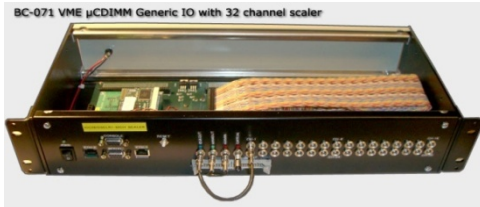


Figure 6: Scaler unit.

Most of the EPICS components were already written. EPICS databases, MEDM screens, and scaler record support were taken from the STD synApps module [6]. An ASYN driver was written to provide device support and define interfaces that allow PVs access to the scalers outside of the record context. Record support provides a framework which calls methods from the ASYN driver in a programmed sequence depending on its counting mode. The driver responds by sending commands and read/write data through the bridge to control the scalers.

The driver defines methods for initializing hardware and scalers, reporting its status, processing scaler values, and handling interrupts from the FPGA. During initialization, the driver registers its methods with the record using the Device Support Entry Table (DSET) structure [9] hooking them into the framework. An interrupt handler is registered with RTEMS to process status and read out scalers, and post them to EPICS.

The FPGA component defines Programmed IO (PIO) registers which hook into the bridge. Registers are accessed by the ASYN driver as memory locations. Commands and data received by the driver indicate what function to perform, such as arm, preset, or read a scaler. All scalers are up counters. Preset values are converted to negative prior to loading into the scaler. When a scaler reaches zero, it generates an interrupt to the driver and dispatches it to a process which posts the status and data to EPICS.

Similar to the scaler, other applications have been developed which use the GEN-II hardware platform. A flexible timing module was developed for CCD shutter controls. It allows the user to delay and stretch timing signals from field instrumentation with 20ns resolution. For this application, no new hardware needed to be developed. A “bunch scaler” was developed to count the number of photons per accelerator bunch from an Avalanche Photo Diode (APD) for a Laser-based pump

probe experiment. A new IO component board was required because the field equipment had intermixed signal levels, such as TTL, ECL, and NIM. Software development for both these applications only required an ASYN driver and MEDM screens for the application specific behaviour and user interfaces.

CONCLUSION

The “Generic Digital” approach provides a design pattern that can be employed to develop and rapidly deploy many beamline and detector applications. The model allows flexibility and the ability to adapt to applications of varying configurations and complexities. Existing hardware components can be easily interchanged and new ones developed. Coupling the uC5282 with an FPGA is a hardware configuration that has been proven to be reliable at the APS and other laboratories throughout the community. Using EPICS, RTEMS, and synApps, reduces overall project cost and allows one to focus more on the application development, thus minimizing the hardware and software development time. Both future beamline and detector applications and those currently under development, along with the APS users, will benefit by the approach discussed in this paper.

REFERENCES

- [1] EPICS; <http://www.aps.anl.gov/epics>.
- [2] Altera Cyclone-II Field Programmable Gate Array; <http://www.altera.com/products/devices/cyclone2>.
- [3] Arcturus Networks, uC5282 microprocessor module; <http://www.arcturusnetworks.com/products/uc5282/>.
- [4] RTEMS; <http://www.rtems.org>.
- [5] Diamond Systems; <http://www.diamondsystems.com>.
- [6] synApps; <http://www.aps.anl.gov/bcda/synApps>.
- [7] Altera Quartus-II Development Software Suite; <http://www.altera.com/literature/lit-qts.jsp>.
- [8] Avalon bus memory mapped interface specification; <http://www.altera.com/literature/lit-sop.jsp>.
- [9] M.R. Kraimer, et al., “EPICS Application Developers Guide,” February 2010, p. 170 (2010); <http://www.aps.anl.gov/epics>.