SSC11-IX-7

Rapid Development of Electronic Systems for Space Applications

M. Brett McMickell, Thom Kreider, Lindsay Harding, Ron Strong
Honeywell Engineering and Technology,
19019 N. 59th Ave. Glendale, AZ, 85308;
(602) 822-4620;
Brett.M.McMickell@honeywell.com

Kosta Ilic and Brad Mouring
National Instruments,
11500 N. Mopac Expy, Austin, TX, 78759;
(512) 683-6610
kosta.ilic@ni.com

ABSTRACT

This paper explores a rapid development approach for creating new electronic systems for digital data processing, control, and interfacing with other components of satellite systems. In order to deliver increasingly complex satellite designs on reduced schedules and budgets, it is critical to rapidly integrate existing knowledge, designs, and components in novel ways. The traditional approach of creating every component of a new system from scratch can divert time and energy from solving higher-order problems.

Field-programmable gate array (FPGA) technology has matured to the point where entire small satellite digital processing could reside in a single active component. In addition, graphical design tools for FPGA development have reached a point where complex, high speed designs can be efficiently mapped to FPGAs, and an array of available off-the-shelf board-level solutions make it possible to rapidly prototype and test new system-level designs. In response to the growing demand of increasingly complex electronic space systems, Honeywell taken advantage of FPGA technology and graphical design tools to develop an alternative approach for digital data processing, control and interfacing with other satellite system components.

This paper shows how the Honeywell designed Responsive Digital Electronics (RDE) board and the graphical programming language LabVIEW were used to rapidly develop and implement a new Thrust Control System (TCS). The development cycle from desktop simulation (TRL-2) to hardware implementation, including designing and building custom drive electronics (TRL-5), took weeks instead of months [6]. The RDE is based on the recently announced Xilinx Virtex-5QV radiation-hardened FPGA, and supports multiple missions. It provides a migration path from commercial electronics to space-rated hardware without rewriting IP.

INTRODUCTION

There is an international need for more responsive spacecraft to support commercial, science, and defense missions ([1], [2], [3], and [4]). As new technologies and system architectural approaches are used to meet this demand, requests for “quick-turn” demonstrations is also rapidly growing. The traditional approach of creating every component of a new system from scratch no longer meets customers’ expectations. To meet the demand, there has been a shift away from large, heavy, expensive, complex space systems to ones which are smaller, lighter, less expensive and rapidly developed. This has created an opening for designers to use and reuse more technologies which are of an ‘off-the-shelf’ type, enabling rapid development at a lower cost.

The Air Force Research Laboratory predicts that 50% of the functionality in future satellite systems will be instantiated in Field Programmable Gate Arrays (FPGAs) [5]. To this end, AFRL has sponsored development of the Xilinx Viretex-5QV FPGA, a radiation-hardened version of the high-performance, commercial Virtex-5 FPGA. Development of the Virtex-5QV provides the foundation necessary to realize reduced schedule and cost of space electronics because it offers designers a two-step migration path from low-cost development to high-performance flight operations.

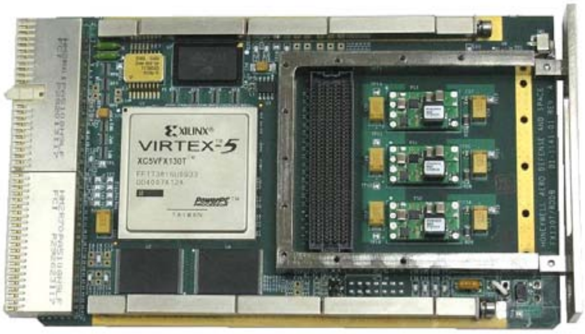

Figure 1 : Honeywell Responsive Digital Electronics (RDE) FPGA board

Honeywell also has an interest in creating technologies and products to facilitate quickly developing and deploying space applications. One example is our Responsive Digital Electronics (RDE) FPGA controller board, designed to take advantage of commercial development environments and extending them into the space marketplace. Figure 1 shows a commercial-grade version of the RDE with a Xilinx Virtex-5 FPGA, memory, power supplies, and user I/O interfaces via CompactPCI and VITA-57 FMC connectors. A flight version of the RDE, using the compatible Xilinx Virtex-5QV radiation-hardened FPGA, also exists with supporting components suitable for operation in the radiation environment of space.

The RDE hardware family is built around this capability to use the lower-cost, commercial/industrial FPGA for early development, and re-target applications to a radiation-hardened FPGA for flight without significant rework. By using a common board design and compatible interfaces to memory and I/O, the RDE extends the concept of the two-step migration path to the board-level. This provides systems designers a credible path from early Technical Readiness Level (TRL) developments towards higher TRL projects without the need to incur large expenses and risky hardware development [6].

This paper illustrates how Honeywell used the RDE hardware platform, with National Instruments LabVIEW FPGA programming tools, to quickly build a Thrust Vector Control demonstration, spanning TRL-2 to TRL-5 in seven months.

RESPONSIVE DIGITAL ELECTRONICS

The Honeywell Responsive Digital Electronics (RDE) platform accelerates development of FPGA-based electronics for space-flight and rugged commercial applications. The RDE centerpieces a large Field Programmable Gate Array (FPGA) and can host IP written with the National Instruments LabVIEW programming language.

This system is a revolutionary approach to solving design challenges for space applications. It blends intuitive graphical programming and flexible, high-performance space-grade hardware to help engineers and scientists more efficiently design, prototype, and deploy embedded systems in demanding environments.

A core concept to the RDE hardware is the minimalistic approach taken to the board itself. The RDE provides a platform to host the FPGA, including memories, power supplies, oscillator, thermal management, mechanical support, and off-board I/O interfaces. All of these features are standards-compliant to typical space and industrially proven guidelines. By preserving compliance to standards and reducing the complement of signal conditioning electronics, the RDE can serve as a highly reusable system design component, limiting the non-recurring engineering normally associated with design elements of this complexity.

General Specifications

Two versions of the Responsive Digital Electronics have been developed: a rugged, commercial-grade device, as shown in Figure 1, and a functionally identical flight-grade variant. Both versions share a standard 3U CompactPCI form factor with conduction cooling. These features include thermal ground-planes and wedge-locks for use in sealed or evacuated applications. Common design specifications include high shock and vibration tolerance across the family.

Both RDE versions include local power supplies, oscillators, buffers, Configuration and User Flash memories, fast external user SRAM memory, a JTAG port, and several user I/O access mechanisms. Each form is optionally available with conformal coatings which meet low out-gassing requirements, space-rugged cPCI connectors, and selective depopulation of components.

FPGA Mezzanine Connector

One of two significant I/O resources is a high-density, vacuum-rated FPGA Mezzanine Connector (FMC). This connector provides 160 single-ended or up to 80 50-ohm differential pairs of reconfigurable digital I/O channels. In addition, 4 Xilinx GTX signal pairs are exposed to the user for multi-gigabit serial communication, which may include PCI-Express or high-speed image conversion interfaces [7]. The FMC site is the user's primary path to steer signals through to the FPGA. This site is compatible with the growing ecosystem of Xilinx accessories and is documented as the VITA-57 standard [8, 9]. An option exists to replace the FMC connector with a custom connector rated for operation through the complete RDE mechanical vibration specification.

PCI Interface

The second I/O resource is the CompactPCI interface is shown as Figure 1. This connector exposes a 64-bit cPCI signaling interface which is directly tied to the FPGA user I/O pins. By avoiding the use of a cPCI bridge and FPGA control device, the user is free to include PCI logic in their applications or use this interface to steer another 100 digital I/O signals to their application. Similar to the FMC connector, the standard cPCI connectors may selectively be replaced with reverse gender contacts for higher reliability throughout the RDE mechanical performance specification.

User and Configuration FLASH

The RDE has two 512Mbit NOR Flash components to store non-volatile configuration and application data. One Flash is designated as the Configuration Flash which stores the configuration pulled by the FPGA at power-up. A single FPGA configuration image is no larger than 64Mbit; therefore the configuration Flash can hold up to 8 independent FPGA configuration images. At power-up, any one configuration image block may be selected in-service through user accessible image selector signals. The second Flash part is designated as the User Flash. This component can be used to hold any type of user data and is entirely under control of the FPGA application after power-up. Both devices can be read-from or written-to at run-time.

Soft-core Processor MicroBlaze

With the availability of large, radiation-tolerant FPGA’s, it becomes cost effective to use soft-core central processor units (CPU’s) in place of discrete, dedicated CPU chips. A soft-core CPU is an implementation of an entire processor completely within the user configurable FPGA logic resources, often tied to other functions unique to the users design. This tightly couples the CPU to the application logic within the FPGA and presents many options to configure the CPU beyond the off-the-shelf component design experience.

The driving motivations to move to an IP-defined CPU in place of separate components are related to component and technology obsolescence. By owning the CPU in the form of IP to be run on an FPGA, designers can ensure long term availability of the CPU, even if the providing supplier of the IP goes out of business, or moves away from supporting the IP for other reasons [10]. Another significant advantage of running a soft-core CPU is that the FPGA designer is free to tailor their IP to incorporate new technology as the state-of-the-art improves. Particularly valuable, is that the designer has freedom to choose the time of the technology insertion as the existing implementation is not under pressure to accommodate a new design due to component availability or vendor support. Ultimately, the FPGA designer has control over processor obsolescence, including the ability to create an obsolescence risk-mitigation plan at program inception.

With the RDE, Honeywell has a working implementation of the Xilinx soft-core processor, MicroBlaze [11] tied to the LabVIEW FPGA programming environment. MicroBlaze runs software stored within RDE Flash and SRAM components, allowing the designer to tailor the memory model and memory protection within LabVIEW, while also memory mapping the users LabVIEW FPGA IP into the CPU address space. This tight coupling to the CPU allows the FPGA designer to control the entire design, clock by clock from within the LabVIEW environment.

LABVIEW GRAPHICAL PROGRAMMING

LabVIEW is a programming and design environment created to reduce development time and cost compared to traditional text-based instrumentation and automation programming languages. LabVIEW delivers this through a combination of a complete graphical language, integrated development environment, multiple compilers, and a strong user community and National Instruments support infrastructure [12].

LabVIEW is a 'blocks and wires' graphical programming language, well suited to expressing the parallel nature of applications in a natural and intuitive representation. It has common programming constructs, such as data types (numbers, strings, arrays, enumerations, custom data-types, etc.), structures for handling conditional and repetitive execution (for loops, while loops, case structures, etc.), functions (data operators, I/O access, comparisons, etc.) and sub-function/library support. A snippet of LabVIEW code is shown in Figure 2.

In addition to targeting FPGA’s, code written in LabVIEW can also be compiled to execute on desktop computers, on processors such as PowerPC with real-time operating systems, and even on microprocessor architectures such as ARM. The variety of computing platforms supported by LabVIEW makes it possible to reuse developer expertise over multiple computing platforms, and carry designs within a common language from initial desktop modeling to flight on the RDE.

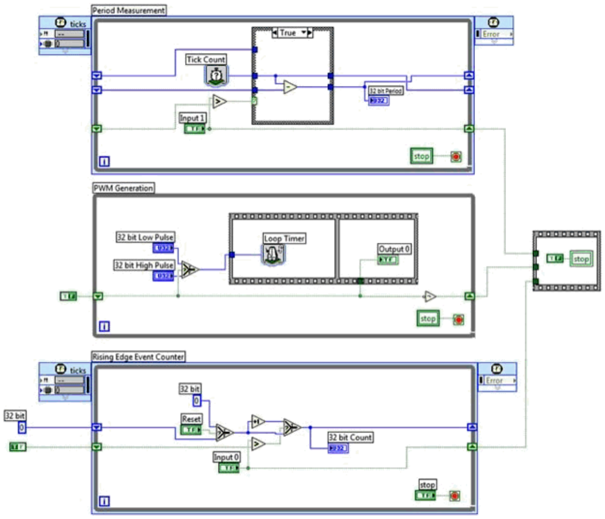

Figure 2: LabVIEW code expressing within three while-loops.

LabVIEW FPGA Support on Honeywell RDE

Honeywell has worked with National Instruments to bring full LabVIEW FPGA support to the RDE. National Instruments has run extensive testing of LabVIEW running on the RDE target hardware, similar to how National Instruments tests LabVIEW against their own FPGA hardware products.

THRUST CONTROL SYSTEM

A prototypical, real-world application was required to prove the developmental advantages of the RDE design philosophy. A Honeywell Thrust Control System was chosen as a final demonstration challenge for the new RDE hardware and LabVIEW programming tools. During the project's life-cycle, the initial RDE hardware was created, the port of LabVIEW FPGA to the RDE was accomplished, and the TCS application was written in LabVIEW against a reference model of a flight actuation system. A custom motor driver and signal conditioning chassis was also created to show the viability of using a common FPGA hardware platform with application specific interfacing electronics. The demonstration was a real-time, full authority test of a heritage actuator running exclusively from a Honeywell application written with LabVIEW FPGA on the RDE hardware target.

Honeywell took the TCS demonstration from a Technical Readiness Level (TRL) of 2 to 5 in only seven months. This project went from conception, through desktop and Hardware-in-the-Loop simulation, to hardware implementation and demonstration in a laboratory environment, and ended ready to go into simulated environmental testing. This effort included the development of the RDE itself, porting LabVIEW, and testing the platform before integration with the TCS hardware.

Requirements

The goal of the project was to develop and demonstrate a prototype thrust control system using actuators from an existing Honeywell product. The system was to provide control of roll, pitch and yaw actuators as shown in Figure 3. Given a set of currently available actuators and sensors, the team was to provide and demonstrate the controller IP and electronics and compare their performance against the original systems reference model.

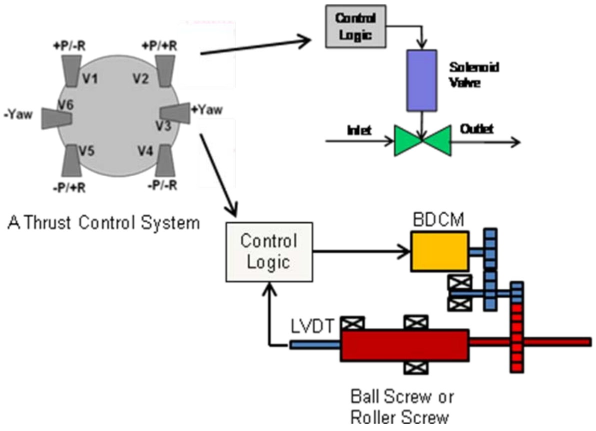

Figure 3: A Thrust Vector Control System

System Design

The TCS system consists of a desktop PC, an RDE with the commercial Virtex-5 FPGA, application specific conditioning electronics, and IP to drive four solenoid driven valves and two brushless DC motors. User commands to the FPGA running the actuator controller are issued to the system from a desktop PC, designated as the higher-order controller in the demonstration application. The PC is responsible for issuing commands and parsing telemetry, as well as programming the FPGA and providing a graphical view of the top-level FPGA interface from within LabVIEW for monitoring and instrumentation of tests.

The functional blocks representing the FPGA and the Flash memory, shown in Figure 4, are located on-board the RDE. Command and telemetry to the FPGA are carried out to the PC via a dedicated port not shown in the figure. The remaining functions are located on a custom, application-specific Printed Circuit Board (PCB), the Motor Driver. The sensor feedback path includes an Analog-to-Digital Converter (ADC), which digitizes conditioned signals for the FPGA to sample, controlled entirely by the FPGA. The ADC and signal conditioning circuits are located on the Motor Driver PCB, which only exposes digital signals to the RDE via an FMC connector.

A pair of motor and solenoid driver circuits buffer digital signals from the FPGA to the high-power coils of the motors and solenoids and provide buffered current feedback signals to the ADC.

The FPGA executes the system controller logic implemented with LabVIEW FPGA. The controller is responsible for driving the various peripherals, including the ADC, analog multiplexer, solenoid and motor drivers, and communication interfaces. The controller is also tasked to run the closed-loop motion control algorithms.

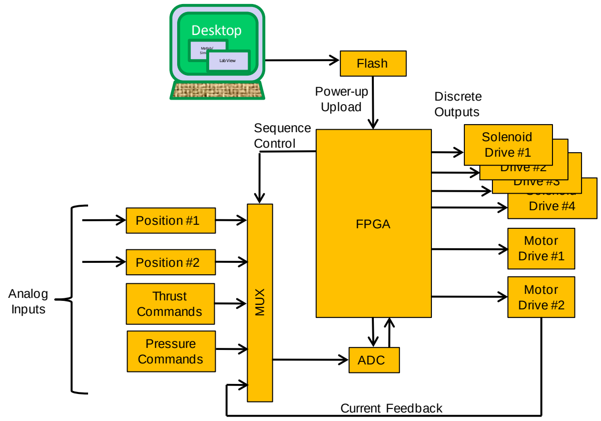

Figure 4: TCS Demonstration System Functional Block Diagram

Rapid Development Execution

Due to the unique rapid development nature of this effort, the hardware, FPGA IP, and software tools where built on an aggressively parallelized schedule. The motor controller FPGA IP was started and tested on commercial, off-the-shelf, National Instruments hardware-in-the-loop platforms. This happened concurrently with the design and build of the RDE and the supporting I/O peripheral drivers on the Motor Driver PCB. Once each element was completed, it was integrated with other available elements until the final system was eventually integrated and tested.

Early work on the motor controller FPGA IP consisted of building LabVIEW FPGA code using the reference model of the system provided with the flight actuators. The primary effort was to properly capture the required behavior from the model, and to express the behavior using the limited mathematical and logical constructs available on an FPGA target. Since a given LabVIEW program can execute on a number of different hardware platforms, the initial control algorithm was developed along with a plant model (motor simulation) to verify the controller operation using LabVIEW for Windows.

Once sections of code were developed on the desktop, they were tested for timing on an National Instruments R-Series Multifunction RIO with a Virtex-5 LX110 FPGA. This provided verification of the application code running on a similar FPGA in real-time before the RDE card and Motor Driver PCB were available. Interfaces to peripherals such as the motor and solenoid drivers were verified with an oscilloscope. Once the application firmware was all integrated, controller operation was verified by closing a control loop with the controller running in the FPGA and the plant model running on a real-time LabVIEW PXI controller.

Up to this point, all of the firmware development was accomplished in parallel to the development of the electronics. When the RDE and custom I/O hardware became available, integration of the TCS application on the RDE FPGA proceeded in a method similar to the initial development. Sections of code were loaded and tested in the FPGA in simulation and with real hardware in iterations. Initially, feedback sensor and motor driver interfaces were each loaded individually and tested with the hardware in controlled, unit-test fashion. The developers were able to directly communicate with these functions using LabVIEW on the desktop and the JTAG interface on the RDE card. Once these functions were operating correctly, integration proceeded with successive iterations adding current and position control loops until the application was complete. A photograph of the demonstration unit is shown in Figure 5.

With all the hardware components and motor controller logic verified, and the RDE application finally integrated, the system was able to drive an actuator with performance very similar to the existing product controller. Verification against the original model was possible by comparing the internal states of the FPGA and telemetry against simulation runs with the same parameters on the reference model as well as comparisons to the telemetry from the original product.

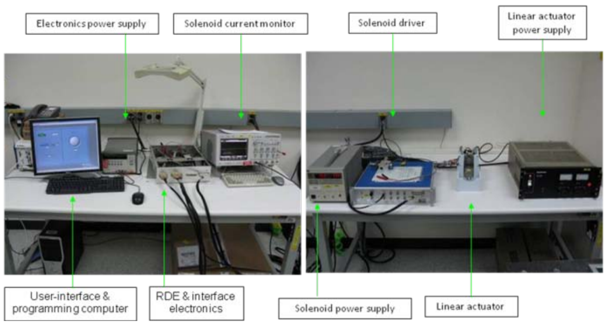

Figure 5: Thrust control system demonstration

CONCLUSION

Combining the radiation-hardened Xilinx Virtex 5QV FPGA with National Instruments LabVIEW FPGA development tools, the Honeywell Responsive Digital Electronics platform provides a viable solution to rapid development of complex digital computing and control applications for space. This paper shows how the RDE platform can be used as a core component across projects through the extensive use of standard interfaces, encouraging reuse and commonality in hardware, tools, and design expertise at a higher level of abstraction and productivity than traditional approaches.

By using the outlined tools, hardware, and design methods, Honeywell has demonstrated that building rapid-turn FPGA applications can be significantly accelerated. Leveraging a commercially successful FPGA programming tool and supporting product line of prototyping hardware from National Instruments brings significant cost savings to the development of complex applications. Designers now have a single platform to develop new concepts and systems from the lab to space, at a lower cost and on a compressed schedule.

REFERENCES

1. Defense Science Board (DSB) and Air Force Scientific Advisory Board (AFSAB), *Report of the Defense Science Board/ Air Force Scientific Advisory Board Joint Task Force on Acquisition of National Security Space Programs*, May 2003.

2. Joint Committee on Technology for Space Science and Applications of the Aeronautics and Space Engineering Board and the Space Studies Board, National Research Council, *Reducing the Costs of Space Science Research Missions: Proceedings of a Workshop Joint Committee on Technology for Space Science and Applications of the Aeronautics and Space Engineering Board and the Space Studies Board Commission on Engineering and Technical Systems Commission on Physical Sciences, Mathematics, and Applications*, National Academy Press, 1997.

3. D. Hastings and H. McManus, Space System Architecture: Final Report of SSPARC: the Space Systems, Policy, and Architecture Research Consortium (Thrust I and II), Massachusetts Institute of Technology; 2009.

4. Department of Defense, Plan for Operationally Responsive Space: A Report to Congressional Defense Committees; 2007.

5. Alexander, David; Hunt, Ken; Owens, Marc; Lyke, Jim; “Affordable Rad-Hard – An Impossible Dream?”, *Proceedings of the AIAA/USU Small Satellite Conference*; SSC08-XI-5; 2008.

6. Department of Defense, *Technology Readiness Assessment(TRA) Deskbook*, July 2009.

7. FPGA Mezzanine Card (FMC) Standard, http://www.xilinx.com/products/boards_kits/fmc.htm, xilinx; Referenced June 20, 2011.

8. VITA Open Standards, http://www.vita.com/fmc.html; Referenced June 20, 2011.

9. Raj Seelam, I/O Design Flexibility with the FPGA Mezzanine Card (FMC), Xilinx; August 19, 2009, http://www.xilinx.com/support/documentation/white_papers/wp315.pdf.

10. Xilinx, Inc.; *Obsolescence be History?*, WP169 (v1.0) October 25, 2002.

11. *MicroBlaze Processor Reference Guide UG081(v12.0)*; Xilinx; Referenced June 20, 2011.

12. National Instruments, *What is LabVIEW*, http://www.ni.com/labview; Referenced June 20, 2011.