Final Report Certification
for
CRADA Number 0665

ENTERED FM 5-30-07

Between

UT-Battelle, LLC

and

RIS Corp., Knoxville, TN
(Participant)

Instructions:

Mark the appropriate statement in 1a or 1b below with an "IX." Refer to the articles in the CRADA terms and conditions governing the identification and marking of Protected CRADA Information (PCI).

If no PCI is identified, the report will be distributed without restriction. If PCI is identified, the report distribution will be limited in accordance with the CRADA terms and conditions governing release of data. In all cases items 2 and 3 must be true. That is, the report cannot contain Proprietary Information and a disclosure must be filed prior to release of the report.

This certification may either be made by using this form or may be made on company letterhead if the Participant desires. A faxed copy of this completed form is acceptable.

The following certification is made for the subject final report:

1. (a) The final report contains information that qualifies as "Protected CRADA Information" (PCI). The PCI legend is printed on the report cover, and the PCI is clearly identified.

OR

[X] (b) The final report does not contain "Protected CRADA Information." The "Approved for Public Release" legend is printed on the report cover.

2. The final report does not contain Proprietary Information.

3. By the signature below, the Participant has no objection to the public distribution of the final report due to patentable information.

For the Participant:

Richard A. Todd
(Name)

PRESIDENT, RIS CORP.
(Title)

MAY 25, 2007
(Date)

CRADA 0665 – Final Report

Graphical Environment Tools for Application to Gamma-Ray Energy Tracking Arrays

Prepared by
David C. Radford
ORNL Physics Division

M. Blair, S. Pauly and R. Todd
RIS Corporation

Abstract

In this CRADA, Oak Ridge National Laboratory (ORNL) assisted RIS Corporation of Knoxville, TN, in the development of graphical environment tools for the development and programming of high speed real-time algorithms to be implemented in a Field-Programmable Gate Array (FPGA). The primary application was intended to be digital signal processing for gamma-ray spectroscopy, in particular for Gamma-Ray Energy Tracking Arrays such as the GRETINA project. Key components of this work included assembling an evaluation platform to verify designs on actual hardware, and creating various types of Simulink functional blocks for peak-shaping and constant-fraction discrimination.

Statement of Objectives

This DOE-STTR effort sought to develop a set of user-friendly graphical tools for developing and programming high speed real-time digital signal-processing algorithms for FPGAs. It also sought to expand and refine the number and types of digital nuclear signal-processing algorithms available to the nuclear physics community and provide these functions and algorithms within a set of reusable libraries. This effort should allow experimenters to easily adapt the functionality of signal digitizers for nuclear physics experiments, without the need to learn the complex VHDL programming language and techniques. It made use of graphical FPGA development tools based on the Simulink™ graphical development environment from The Mathworks and the System Generator™ FPGA development add-on tool from Xilinx.

Benefits to DOE Office's Mission

Highly segmented, position-sensitive germanium detector systems, such as Gamma-Ray Energy Tracking array GRETINA, are being developed for nuclear

physics research where traditional electronic signal processing with mixed analog and digital function blocks would be either impossible, or enormously complex and costly. Future systems will be constructed using pipelined processing of high-speed digitized signals as is done in the telecommunications industry. Techniques which provide rapid algorithm and system development for future systems are desirable, especially if they do not require expert knowledge of HDL programming languages. This effort resulted in graphical environment toolsets and modular libraries of nuclear physics functions to enhance digital pulse processing capabilities for next-generation detector arrays at DOE nuclear physics research sites, and allowing the nuclear physics community to generate their own customizable algorithms.

The benefits of this graphical approach to FPGA programming include:

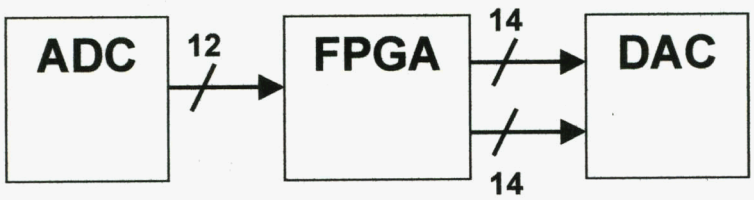
- Interactive approach provides immediate performance feedback
  - Simulation functionality provides scope view of signals
  - A variety of Matlab and Simulink functions can be used to analyze results or provide stimulus to the algorithm
- Extensive knowledge of VHDL is not required
  - System Generator will produce VHDL for the targeted Xilinx FPGA
  - Makes extensive use of Xilinx cores including Virtex hardware multipliers
  - Makes use of synchronous design with one system clock
- Subsystems and custom libraries permit design re-use
  - Drag and drop of Xilinx library functions
  - Drag and drop of User library functions
  - Drag and drop of Simulink functions for creation of stimulus (e.g. a model of a detector, including noise)

Technical Discussion of Work Performed

Software licenses for Matlab™ and Simulink™ from The Mathworks were purchased, and these tools were used in conjunction with System Generator from Xilinx. Together, these tools used a block-diagram approach for algorithm and system design, and directly generate VHDL code which is optimized for Xilinx FPGAs.

An evaluation platform was assembled to verify designs on actual hardware, using evaluation boards for an ADC, a Xilinx FPGA, and a DAC to allow digitized signals to be manipulated with the resulting waveforms viewable directly on an oscilloscope (See figure 1).

Simulink blocks along with Xilinx In/Out blocks were used to provide input signals to the functions and to specify the FPGA pinout. Another System Generator block was used to generate the corresponding VHDL code for specific Xilinx FPGA targets. It also allows setting the global clock sample period.



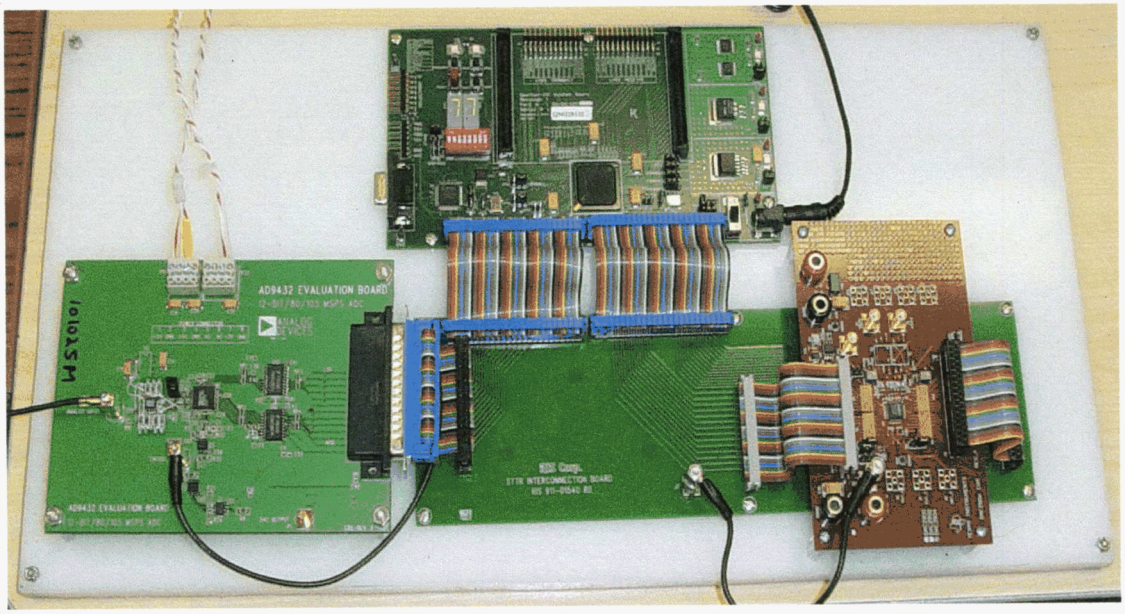


Figure 1: The evaluation platform constructed to verify implementation of programmed algorithms and proper programming of a target FPGA. It combines an ADC (Analog Devices AD9432, 12 bits, 105 MSPS), an FPGA (Xilinx Spartan IIE, 300K gates, 100 MHz clock) and a DAC (dual Texas Instruments DAC2904, 14 bits,125 MSPS).

A number of functional blocks implementing specific operations were developed, and these were then combined to create blocks for specific signal-processing functions. The functions developed included:

- Signal shaping functions, such as
  - Trapezoidal shaper, with adjustable rise time and dwell time
  - Quasi-Gaussian shaper, with adjustable number of poles and rise time
  - Parabolic quasi-cusp shaper
  - True exponential cusp shaper, for comparison purposes
- Exponential cancellation ("pole-zero" correction) with adjustable time constant
- A baseline restorer to correct for fluctuating DC offsets
- A constant-fraction discriminator (CFD) for amplitude-independent timing; includes adjustable CFD delay and fraction

- A pile-up rejection function, for discarding events that are distorted by being too closely spaced
- Histogramming memory, for creating and storing energy spectra (histograms), with variable resolution and dynamic range

Using the Simulink and System generator tools, it was then relatively simple to

- Draw a block diagram for the entire design, by dragging and dropping the functional blocks and connecting them to specify the flow of data
- Simulate the design and observe the response of the algorithms to measured stimuli, using the simulation and "scope" capabilities
- Modify the diagram as needed to achieve the desired responses
- Generate VHDL code for the desired FPGA platform, and
- Compile the VHDL code to create a bit file, and download this into the FPGA

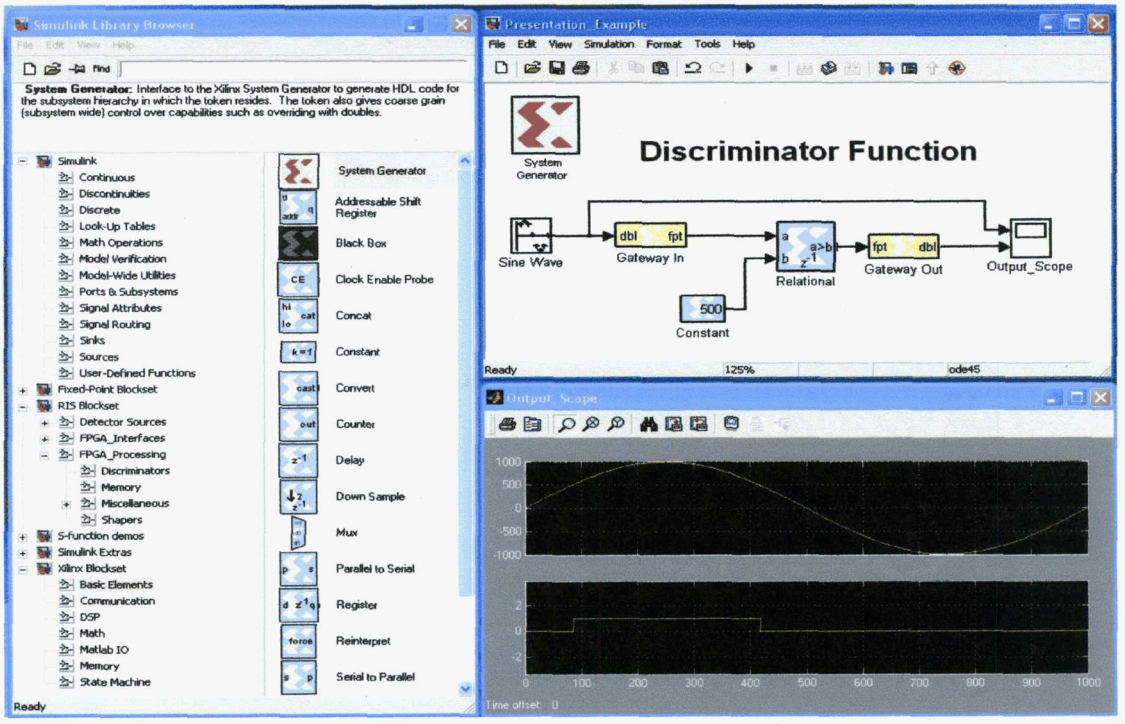


Figure 2: Example of graphical environment showing a simple function and signal displays.

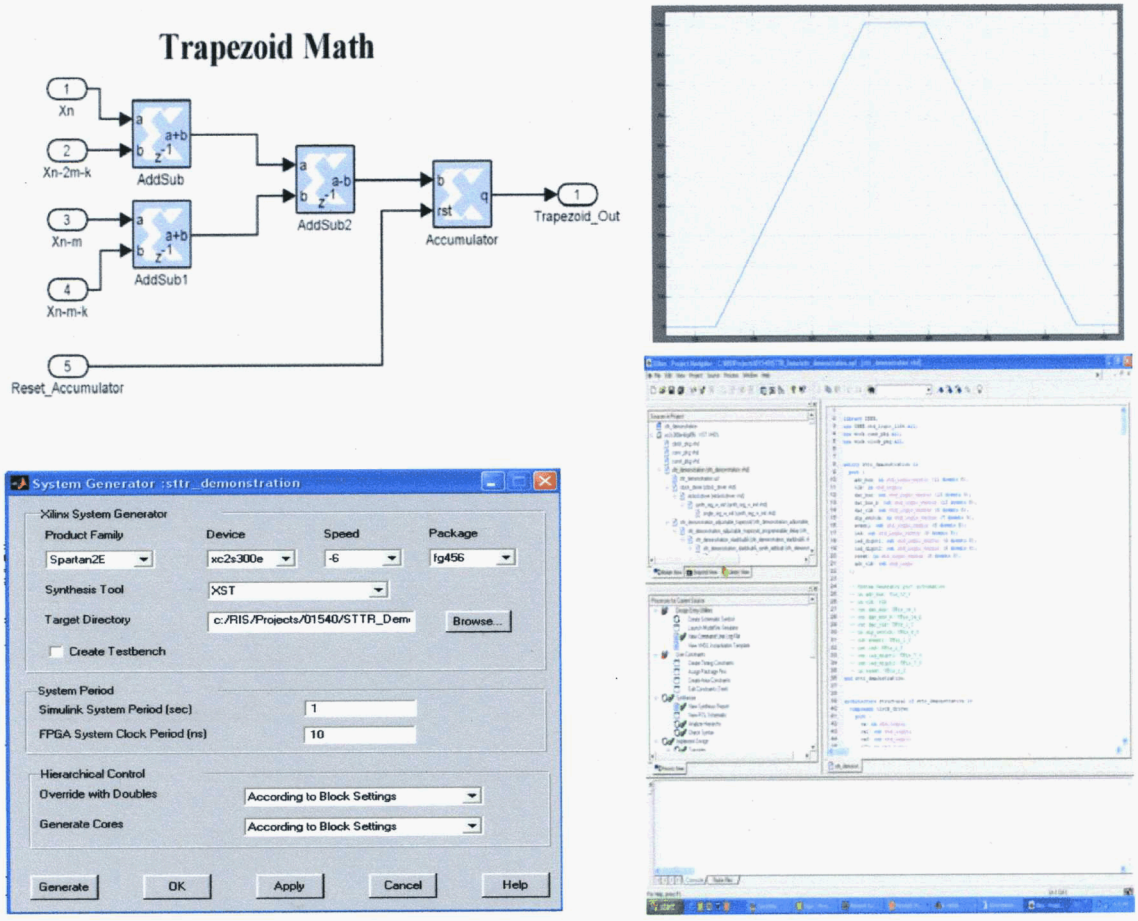


Figure 3: Example of graphical environment showing an implementation of a trapezoidal shaper.

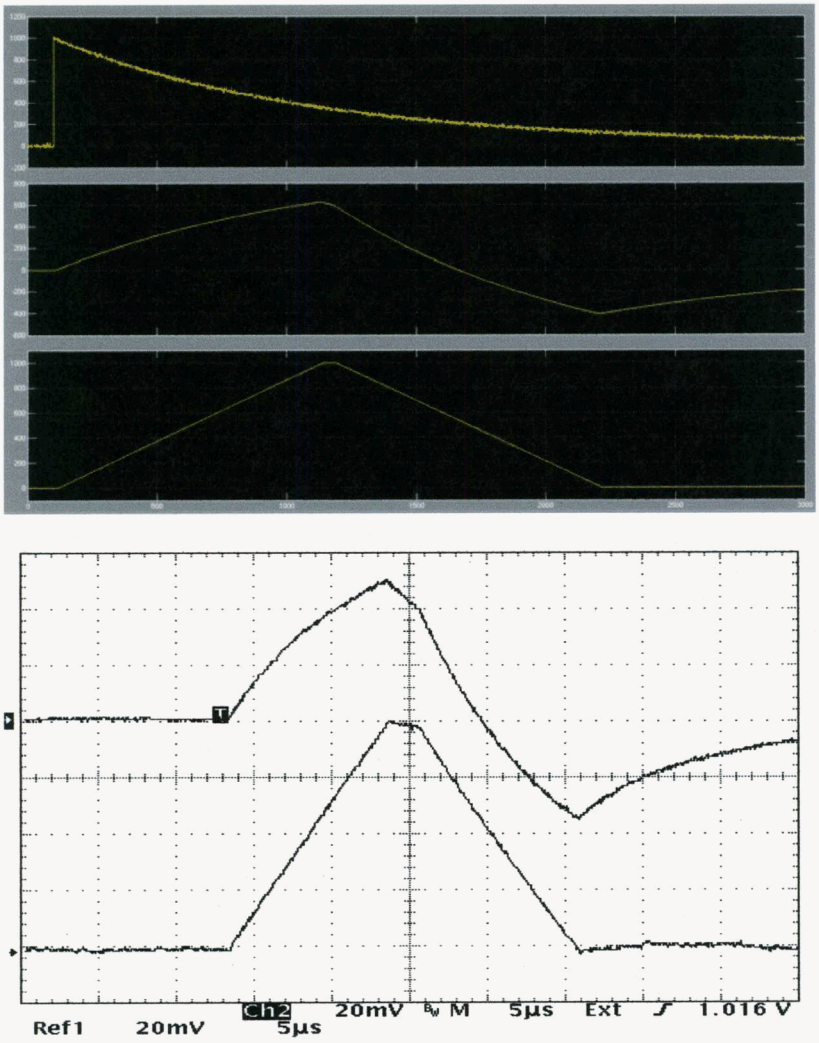


Figure 4: Trapezoidal shaper outputs, both simulated and measured from the evaluation platform, and both with and without the pole-zero correction function.

Commercialization Possibilities

We anticipate four areas of commercialization for this technology.

1. Direct application of the algorithms to DOE-sponsored research experiments will immediately add value to researchers by bringing the FPGA-simulation and programming tools into a graphical environment, saving countless hours in VHDL code generation. RIS Corp. will provide contract services or specific Intellectual Property (IP) to implement these algorithms on an individual basis. Some of the functions developed as part of this effort have already been provided (as VHDL code) to the GRETINA electronics group at LBNL, for implementation in the GRETINA digitizer board FPGA.

2. RIS Corp. has been already contacted by two commercial companies that produce digitizer hardware. Discussions have centered around RIS Corp. providing custom firmware for these companies to make the use of their products much easier for their customers.

3. RIS has identified applications for this technology that are outside of the physics research community. In particular, they have teamed with an office of the U.S. Army Corps of Engineers in Huntsville in proposing an improved signal processing algorithm platform development system for the detection and classification of unexploded Ordnance (UXO).

4. RIS has applied this graphical technique to the implementation of other prototype instruments, such as random amplitude and frequency pulsers, spectrum generators, and pulse-shape discrimination circuitry

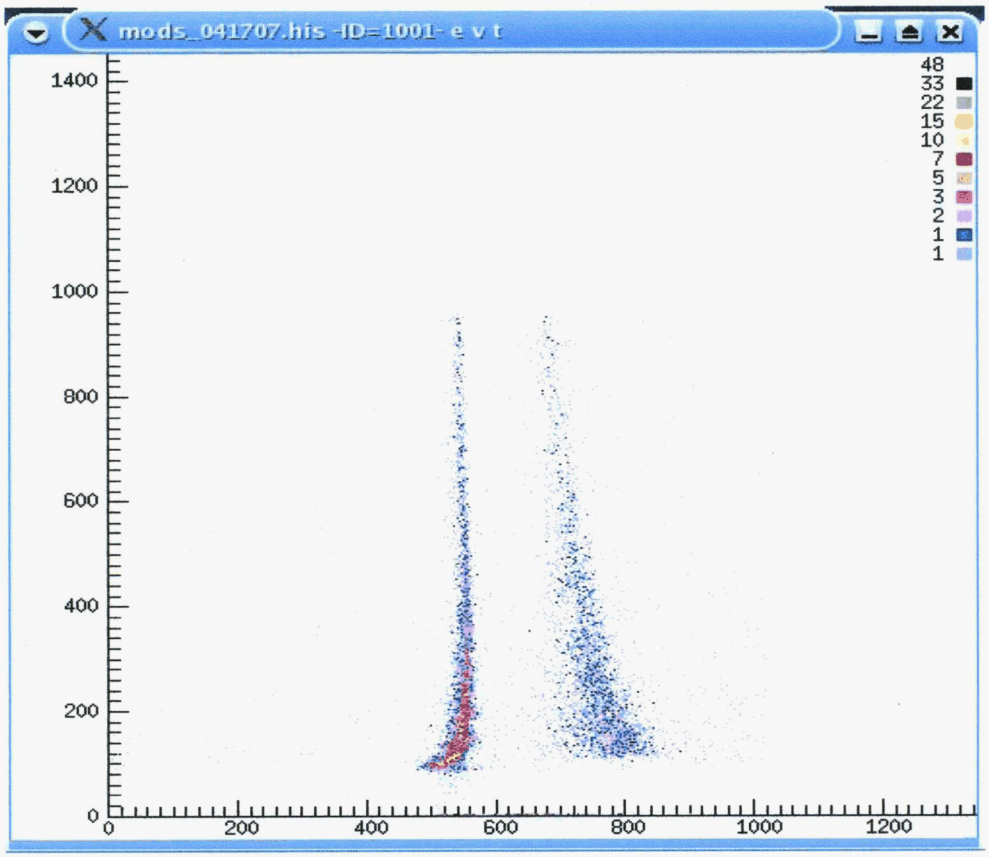


Figure 5: Real-time neutron-gamma discrimination plot using graphical environment toolsets showing amplitude vs. timing figure-of-merit with gammas (left) and neutrons (right) detected in large liquid scintillators.

facilitates generation of customized algorithms by the nuclear physics community.

The benefits of this graphical approach to FPGA programming include:

1. The interactive approach provides immediate performance feedback,
2. Extensive knowledge of VHDL is not required,
3. Subsystems and custom libraries permit design re-use, and
4. Drag and drop of library functions and user-created functional blocks allows high-level programming of DSP algorithms by non-specialists.

However, the design and programming of FPGA circuits still requires familiarity with the tools and concepts while the commercial toolsets are in a state of constant revision. Efficient utilization of the graphical techniques requires more than casual usage of the toolsets, and in the optimum would be implemented in a team approach where the graphical illustration clarifies the signal processing algorithm and the black box or wrapper file is designed with detailed knowledge of the hardware or board-level circuitry.

While intended primarily for application to detector systems such as GRETINA, the technology can be readily commercialized for numerous other applications, due to the extensive use of FPGAs in modern systems.

RIS Corp. has continued to develop products based on the concepts initiated in this research. Signal processing algorithms have been applied to segmented germanium detectors, position-sensitive silicon strip detectors, neutron-gamma discrimination in liquid scintillators, and with other scintillating crystals such as NaI(Tl) in a 1024-channel dual MCA implementation.

Conclusions

In this CRADA, Oak Ridge National Laboratory (ORNL) assisted RIS Corporation in the development of graphical environment tools for the development and programming of high speed real-time FPGA algorithms for gamma-ray spectroscopy. Various types of functional blocks for digital signal processing