

TARANIS 0^+ Detector Hardware Prototype System Overview

Nader Moussa (*nwmoussa@stanford.edu*)

Abstract—this report summarizes the hardware prototype development for the TARANIS 0^+ detector. This system detects 0^+ whistler-mode radio signals, which are interesting because they may exhibit correlation with optical observations of Transient Luminous Events (TLEs). By detecting 0^+ events in real-time, this system will enable a more complete statistical analysis of the 0^+ incidence rate, providing insight into the environmental factors that affect this incidence rate and the association of 0^+ radio signals with optical TLE events. This prototype implements the detector in hardware on an FPGA, enabling low-power application-specific parallel processing for the complex signal processing detection. The system operates in real time, and has been tested by injecting pre-recorded data from the earlier DEMETER satellite, and the prototype functions even in the presence of many various noise environments and interfering signals. Effort has been made to provide some level of parameter configurability to adjust sensitivity to noise and 0^+ signal profile variations.

Index Terms— FPGA prototype, 0^+ , signal processing, TARANIS,

I. INTRODUCTION

Earlier work involved the design of a method for automated detection of 0^+ whistler incidence in a simulated data environment. More recently, this algorithm has been implemented on a Field-Programmable Gate Array (FPGA) prototyping platform, a hardware system suitable for real-time, low-power, in-situ satellite observations with automatic detection of these signals of interest.

The current hardware is implemented via an Altera DE2 Development Board^[3], which is a low-cost, commonly available utility and test platform for the Cyclone II family of FPGAs^[4]. In the eventual satellite deployment, a similar FPGA may be used; if a radiation-tolerant FPGA is preferred, some effort will be required to port the current design onto that platform, but most of the system is suitable for platform portability.

The algorithm was originally designed for the FPGA platform, incorporating parallelization, hardware-optimized Fast Fourier Transform (FFT), and low memory footprint. The current prototype emulates the data input/output system of the deployable system, capitalizing on peripherals available on the DE2 Development Board.

II. BACKGROUND

For a more complete overview of the scientific background, please see my earlier report, *Detection, Characterization, and Categorization of 0^+ Events for TARANIS*^[1], from 1 April 2008.

A. Scientific Overview

Sferics are radioatmospheric impulsive noise signals caused by lightning strikes. In most cases, the rapid discharge of electricity produces a wide-band electromagnetic emission. The very low frequency component of this emission, in the 30Hz – 30 kHz band, can propagate in the Earth-ionosphere waveguide. In some cases, the signal penetrates vertically through the ionosphere into the magnetosphere. This mode of propagation sends the signal through a non-ideal medium, with a constantly varying ionization level (N_e) and consequently a varying level of dispersion.

The term ‘sferic’ or ‘radio atmospheric’ specifically refers to a lightning-generated radio signal which remains trapped below the ionosphere. Most signals do not pass through the ionosphere’s lower boundary, located at approximately 80 kilometers above the ground. However, in some cases, the wave is able to penetrate through the ionized layers. These events are known as whistler-mode waves, characterized by a highly dispersed time-frequency signature because the varying charge-density creates a frequency-dependent speed of propagation. This dispersion occurs at audio-frequency (base-band or “Very Low Frequency”) and is audible; hence the name “whistler” is applied to any dispersed signal. Specifically, the 0^+ whistler is the nomenclature for the signal during its first upward-propagating path. It is these 0^+ signals which are of scientific interest due to their potential association with rare optical emissions known as Transient Luminous Events^[2].

The 0^+ detector will be connected to the signal from the TARANIS Electric Field antenna. This signal will be pre-conditioned in the analog domain, and then digitized at 1 MHz x 2-channels. This data is finally passed to the 0^+ digital circuits for processing and detection. The 0^+ detector outputs a low data rate output indicating detection, current-state, and some environmental parameter estimates useful for 0^+ analyses. The external system interconnect is determined by the TARANIS MEXIC design and is outside the scope of the 0^+ detector hardware prototype.

B. Algorithm Overview

After systematic evaluation of many alternative options, it was determined that the most effective algorithm for 0^+ detection is an “Ultra-Low Frequency Pattern Matcher.” This algorithm has a very high detection rate, a very low false-trigger rate, and a strong resilience to noise and environmental interference. It also has inherent configurability to seek a variety of 0^+ profiles; this may leverage an uplink command to load new 0^+ detection profiles into the system in flight.

The basic operation of the algorithm is a spectral pattern match. The signal frequency spectrum is calculated at each time-step. The spectrum is scanned for signal power above a pre-determined threshold in each band of interest, determined by the desired 0^+ signature to match. If a sequence of bands trigger above the threshold, in correct order, then a 0^+ is recorded for that time.

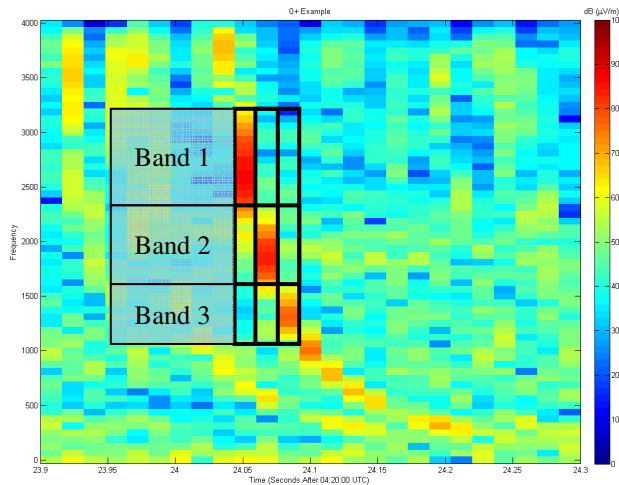


Fig. 1 - Visualization of the spectral band matrix in MATLAB simulation. The output levels of each boxed region are summed, and then the entire grid is matrix-multiplied against a pattern matrix of constant values.

The method to determine when a signal is above the detection threshold is a simple matrix multiplication. The spectrum is averaged per band of interest to yield a 5-point spectral summary at each time slice. Then, the previous 5 time-slices are stored, creating a 5x5 matrix (25 elements). This is multiplied against a “pattern matching filter” which isolates the 0^+ signature. If the result is above the threshold, a 0^+ is triggered and the 5x5 matrix, which summarizes the time-frequency power profile of the incident 0^+ , can be saved for downlink and post-processing. In addition, the system clock tags the time of arrival, corrected for internal processing delay. The time of arrival is defined as the time of incidence of the first band above the detection threshold (in current implementation, this is centered at 3500 Hz).

Most of the *computation* is performed in specialized digital hardware (such as the FFT unit, and the matrix multiplication system) on the FPGA. *Data control and flow* is managed by a general purpose Central Processing Unit (CPU), which is best described as a soft-core FPGA microcontroller. The terminology “central” processing unit is standard nomenclature, but this particular application is highly parallel,

with special-purpose hardware to perform the “central” calculations. As such, the “CPU” might best be described as a peripheral control processor (PCP); this non-standard nomenclature is used in some source-code and low-level documentation.

Earlier work implemented an OpenRISC1000 (OR1K) processor, because it is an open-source, flexible, and portable code-base. However, it is also more complex, larger, and more difficult to configure and compiler software. Interaction with prototype debugging peripherals became an issue with the OR1K, so a standard Altera NIOS II processor was substituted for rapid development. In the deployed satellite system, either of these processors is suitable; the specific FPGA used in flight will dictate what processors are compatible. Since the data control software is simple and written in the C language, it will not be difficult to port this layer to the new CPU.

III. HARDWARE SUBSYSTEMS

The FPGA development environment is designed as a prototype for the eventual space-borne device. The eventual system must be radiation tolerant, and is specifically aimed to be a single FPGA in the “system-on-chip” paradigm. All of the 0^+ detection mechanisms are implemented on the FPGA device (without specifically requiring external memory, peripheral devices, etc). Peripheral hardware, suitable for the test and design environment, is available on the prototyping board, and has been used extensively. This streamlines the ease-of-use for data input and output, with considerable emphasis on human-readable and visual displays of the processing effort. These subsystems are designed to be easily detached from the 0^+ detector, since they will be unnecessary in the deployed satellite version.

The basic 0^+ detector consists of the following systems:

1. a data acquisition block,
2. digital signal conditioning and filters,
3. analog and digital clock timing control,
4. an FFT unit,
5. a matrix calculator,
6. automatic gain adjustment system,
7. simple CPU for programmable data flow, and
8. a data output and formatting block.

These systems are intended for inclusion on the deployed satellite hardware, with minimal modifications.

In addition, the following hardware peripherals are implemented for prototype and debugging purposes:

- UART (serial-port) communication
- Analog audio input for experimental data injection
- RAM block for extra data and profile storage
- Video controller for rapid display of real-time data
- RAM block for video subsystem
- FPGA system interconnect fabric to unify command and data-flow to all digital components
- Error-injection system to simulate radiation-induced logic errors

A. 0^+ Detector Blocks

1) Data Acquisition

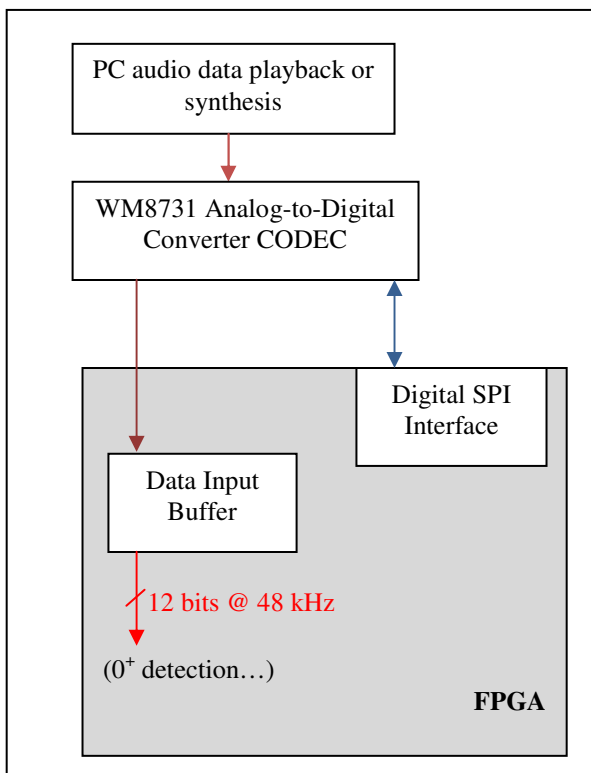
When deployed, TARANIS will provide input data to the 0^+ detector. The specified format will be a digital data bus which connects the output of an Analog-to-Digital Converter to the 0^+ hardware unit. The current specifications^[6] call for 12-bit parallel data in at 2 MHz (time-sharing alternately between two 1MHz channels). The ADC will be an Analog Devices AD9235 chip, capable of oversampling even at 1MHz; since the 0^+ detector operates in the VLF regime, significant downsampling will be performed on the FPGA.

All analog preconditioning, including the antenna design, pre-amplifier, and ADC subsystem, will be handled by the TARANIS, and is outside the scope of the 0^+ detector in both the deployed form and the current prototype.

For prototype purposes, I used the DE2 prototype's onboard analog-to-digital converter. This device, a Wolfson WM8731 internet audio CODEC chip, is profoundly more complicated than the intended final device. Briefly summarized, the device has a Serial Peripheral Interface (SPI) for control and a digital audio interface for input and output data.

The DE2 board takes an input analog signal through a 3.5mm audio connector cable. This allows signal synthesis or data-playback from a standard personal computer (PC). The analog signal is routed into the WM8731, which performs a high-frequency sampling and digital downsampling to standard audio frequencies. For the purposes of the 0^+ detector, the WM8731 takes 12-bit stereo samples at 48000 Hz.

The FPGA emulates the TARANIS interface with an onboard logic block that seamlessly configures the WM8731 and buffers its data into a 12-bit parallel format.



The WM8731 specifies a 90dB dynamic range for analog input. This parameter has been verified as part of the integrated system; the FPGA receives data for processing with less than 1 bit of noise in the quiescent case. The spur-free dynamic range (SFDR) has been measured at better than 70dB but this is largely a digital artifact of the on-board FFT windowing.

Because the input data is not at the ideal rate for processing and detecting 0^+ signals, the digital system must perform anti-aliasing and downsampling to 8000 Hz sample rate. In the final deployed solution, the input data will arrive at 1 MHz; this will change the parameters on the input filters.

2) Digital Signal Conditioning and Filters

The desired signal rate is 8 kHz for best 0^+ signal detection. Since the input data is provided at higher rate in both the prototype and in the specifications for the final TARANIS system, the FPGA must down-sample to 8 kHz.

The unique FPGA platform allows such a straightforward operation as down-sampling to be performed entirely in hardware, parallelizing the task and reducing the computational burden on other parts of the system. Using a standard digital two-tap Infinite Impulse Response (IIR) low-pass filter to anti-alias the signal allows down-sampling by a factor of approximately 10 with negligible signal corruption. Cascading these blocks allows arbitrary down-sampling (e.g. from 1 MHz to 100 kHz to 8 kHz). On the prototype, it is possible to directly down-sample from 48 kHz to 8 kHz, but to preserve the spirit of the deployed system and to test the pipelining delay, the down-sampler was implemented as a two-stage cascade to 24 kHz and then to 8 kHz.

Using MATLAB algorithms designed during the simulation stage gives optimal coefficients for the IIR filter, and these are directly coded into the firmware as registers. In a later version, these may be writeable via control-software to enable configuration of the anti-aliasing.

3) Clock Control

To accurately process the data and to preserve scientific utility, the TARANIS 0^+ detector must precisely manage its clocks. The prototype FPGA is driven by a 50MHz crystal oscillator, which has been measured to drift by less than 1 part per million ($50.000000 \text{ MHz} \pm 5\text{Hz}$); this corresponds to a measured oscillator $Q = 15.2 \times 10^6$. It is hoped that the final satellite system can provide as good or better crystal oscillator input. The current satellite specifications^[6] call for a 40MHz system; this should not affect performance, but will change some digital clock counter parameters.

Immediately on entry to the FPGA device, the 50MHz signal is passed into a hardware-implemented, digitally controlled analog phase-locked-loop (PLL). This provides an extremely well-behaved clean clock signal across the digital die. In the current prototype, it also generates a 100MHz phase-controlled clock for the operation off off-chip DRAM. This signal will not be needed for the spacecraft version.

The 50MHz clock unit drives most of the on-board digital logic, including the Central Processing Unit (CPU). There are also significant sections of digital logic which are clocked at the sample-rate, 8 kHz. The Cyclone II does not have analog hardware PLLs suitable for generating such low-frequency signals, so it is necessary to create these with custom digital logic. I developed a system of counters which generate digital pulses at integer multiples lower than the clock rate. These counters are assigned as clock units, and drive signal processing logic across the board. Because they operate at very low frequencies, there is negligible clock skew over the 2mm die, and no specialized hardware is needed to handle them.

It is important to note that the analog PLL control is platform-specific. The final FPGA used for the flight-hardware may be compatible with the Altera digital PLL interface; but if not, this analog front-end clocking system could need some re-engineering. This is heavily dependent on the desired flight-unit hardware.

4) FFT Hardware Unit

The Ultra-Low-Frequency pattern matching algorithm requires spectral data analysis. Once again, the FPGA's configurable digital logic enables parallel processing of the signal FFT. I selected a 128 point transform because simulation results indicate that is sufficient resolution for effective detection of 0^+ units. It also provides a small enough data-buffer to operate entirely on the FPGA without need for a large external memory for signal buffering.

The FFT operates in-place on floating-point data in, and generates floating-point data out. This is the preferred solution, since integer-data requires a 1.5-bit-shift for each FFT folding stage (i.e. 11 bits for 128-point FFT); 12-bit sample data would thus require a 32-bit word to ensure no loss of precision, wasting much space and complicating the calculation. Instead, a 32-bit IEEE754 floating point representation is used. A convenient side-effect of floating-point representation (which stores numerical exponent in a hardware segment) is that logarithm is a simple and calculation-free operation which can be performed in less than one clock-cycle.

The CPU must pre-process signal input by casting it to a 32-bit floating point representation. This requires one CPU cycle per sample, which is negligible:

<i>Sample Duration</i>	<i>125.00 μs</i>
<i>CPU Casting Operation</i>	<i>0.02 μs</i>

The FFT then takes 171 CPU clock-cycles to operate. This occupies 3.42 μ s, still less than a single sample at 8 kHz. Evidently, most of the CPU and FFT unit time is spent stalling while waiting for input data, meaning that there is potential to under-clock the system. This could potentially result in a reduction of power consumption, but that has not been experimentally tested or verified. The present recommendation is to maintain 40MHz digital clock rate.

5) Matrix Calculator

This subsystem is responsible for the actual detection of 0^+ signatures. Although heavily numerical, the implementation is fairly straightforward. For further implementation details, the source code is heavily documented (in-line) for this process; the previous report [1] also details the algorithm operation without the additional complexity of hardware-description.

The 0^+ detection algorithm operates on a past-history of 5 prior time-slices. The FFT spectral output is first converted into decibel format via a hardware-implemented logarithm computation. These decibel values are normalized to the range 0 to 65 dB, with a noise-floor estimate at 0 dB augmented by the Automatic Gain Compensation circuitry discussed in section 6). The spectral data are segmented into pattern-defined spectral bands. All the data in a single band is summed, integer-rounded, averaged, integer-rounded again, and compared against a threshold to set a flag-bit. (Equivalently, at the hardware level, the value is either 0 or POWER_LEVEL, in order to preserve information about the actual signal strength for later data reporting; specialized digital logic can treat any non-zero value as a "1" bit).

The result is a 5x5 binary-matrix (single-bit elements) indicating the time-frequency profile of the 0^+ signature. These are multiplied against a normalized trace matrix, with a Gaussian blur in the x-axis.

Most of this subsystem is implemented directly in VERILOG Hardware Description Language, with some control operations performed by the CPU and coded in C. Earlier versions used OR1K assembly-code to handle data flow; this was discarded as it was platform-specific.

6) Automatic Gain Compensation

Simulation studies indicated that the constantly fluctuating background environment could disrupt the detector sensitivity. After extensive research, an Automatic Gain Compensation solution was developed and tested on simulated data with encouraging result. The automatic gain compensation unit (AGC) measures and parameterizes the background environment, in parallel with the 0^+ detection system, adjusting the sensitivity setting to "float" above the background noise.

The AGC creates a two-parameter description of the noise environment as two 32-bit words. The first parameter indicates the measured noise power level and the second parameter characterizes the noise spectral distribution on a Gaussian scale by estimating the expected standard-deviation of the time-domain signal. These two parameters are computed by specialized IIR and multiplier units to minimize hardware cost and parallelize their results.

A configurable parameter, which may be modifiable by the uplink command system, is the time-rate of refresh for these parameters. Simulation indicates that a 1 to 5 second refresh rate gives best resilience to interfering noise without generating false detections.

The outputs of the AGC subsystem are fed into the Matrix Calculator. These are used as thresholds for the detector triggering logic. Simulation has provided valuable insight into these parameters; after normalizing for physical units, the

hardware threshold levels exactly correspond to the MATLAB simulations and yield identical detection behavior.

7) System Control (CPU)

As mentioned in the Algorithm Overview section (II.B), the control processor is used for managing data flow, verifying timing constraints, and converting data formatting between various hardware-specific bit orders.

The current implementation uses an Altera NIOS II CPU due to its lightweight design, rapid readiness, and integrated development environment. The OpenRISC1000 CPU was used for some work including radiation-tolerance hardware simulations, because its open-source hardware description allowed injection of transient or permanent bit-flip errors in an integrated test bed. Such low-level access to individual logic-gates is not enabled on the Altera NIOS II processor for licensing reasons; but at a system-level, the NIOS II processor out-competes the OR1K with regard to available documentation, software and hardware deployment tools, and general ease-of-use.

8) Data Output Block

The TARANIS specification requires a low data downlink, because the satellite has limited bandwidth and must share this among all the instrumentation. The specification requires a UART (“Universal Asynchronous Receiver/Transmitter”), which is a standard serial-communication method for system interconnection.

The downlink data is guaranteed to be less than 10 kilobits per second, because even if a 0^+ is detected at every single time slice, the total data transmitted is very reduced.

This information is available for downlink:

- 0^+ detection time (system-clock estimate, or GPS time provided by the external TARANIS system)
- 25-element “mini-spectrogram” summarizing the dispersion profile and power
- Total power (integral of 0^+ spectral power)
- Average background environment parameters (noise power, dB, and standard-deviation noise-spread)
- Automatic Gain Compensator current settings

This information is packetized into the standardized format for TARANIS downlink by the control CPU on the FPGA. Then, the data is stored into the buffer for transmission to the main MEXIC control; eventually, the data is relayed back to the ground station (this usually occurs once per half-orbit).

IV. PRELIMINARY BENCHMARK RESULTS

A. 0^+ Detection Rate

In brief summary, the hardware prototype exactly matches the MATLAB 0^+ detection algorithm, as tested on a large data-set from DEMETER.

The data set consists of approximately 2 hours of data (selected 2-minute burst-data from four full-orbits and some half-orbits, as well as longer-duration data sets):

- Orbit 057671
- Orbit 057680
- Orbit 057681
- Orbit 057720
- Orbit 057721

The data set also includes the scientifically interesting “Langmuir Lab Observations”^[8], consisting of burst-data from 2005-07-28 and 2005-08-03.

Data is pre-processed in MATLAB to generate an audio-file, in the WAV-file format:

- “Microsoft Windows WAV file” compatible
- PCM format
- $F_{\text{SAMPLE}} = 48000$ Hz
- Channels: 1 or 2 (only left-channel is used for detection)
- Bit Rate: 16-bit samples

Details of operating the conversion script and connecting the prototype can be found in the TARANIS 0^+ detector hardware manual.

The hardware detects signals in real-time, which means that data must be played at actual speed; this means that testing can take significant amounts of time. The data is sent by downlink over the UART back to a controlling workstation; the hardware-detection reports are compared to an equivalent MATLAB algorithm which operated on an identical data set.

At present, there is a 100% correlation between the MATLAB algorithm and the hardware implementation, indicating correct implementation of the algorithm and minimal noise interference due to the digital circuitry.

Further statistical details are being compiled at this time including numerical comparisons with the manually-compiled 0^+ database.

B. Radiation-Induced Error Simulations

Benchmark analysis of the software control demonstrated that the algorithm and hardware implementation is not resilient to radiation-induced single-bit-flips, despite best-efforts to address that issue. Effort to add error-correction circuitry did not solve the general-case problem of bit-flip errors. This indicates that the flight system should be implemented on a radiation-hardened platform.

The error injection test framework consisted of modifications to the OR1K CPU hardware description at the gate level. Error injection sites, with the capability to control a temporary or permanent bit-flip at a single logic-control gate inside the CPU, were added to the overall system. Each injection site is uniquely addressable, allowing a control-terminal to command a simulated “alpha-particle-strike”-like event, flipping the result of a single logic element inside the

CPU. This bit cascades into the entire data-flow at the gate-level, enabling a test of overall system resilience.

A radiation-test trial consisted of running a software test, such as the matrix calculation and data flow control for 0^+ detection. This calculation can be repeated 100,000 times, while a controlling terminal injects bit-flip errors at random times and locations to each of the pre-designed error injection sites.

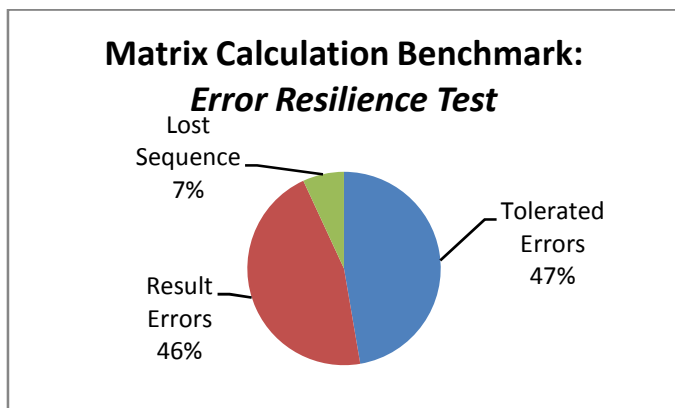


Fig. 2 – Results of the error-injection test on the 0^+ detector. Tolerated errors mean the CPU was unaffected by the injected bit-flip. Result errors mean that the CPU performed normally but the computational result was incorrect for the input data. Loss of sequence indicates that the CPU behaved in a non-deterministic way.

Many variations on this test were performed. Even the addition of parity-checking, arithmetic residue and recovery for multiplication operations, and other industry-standard techniques, were not sufficient to protect the control CPU. The CPU could tolerate nearly half of the injected errors, but an almost equal amount caused either a minor or major error in the computational result (affecting 0^+ detection rate). Most worrisome is the 7% of errors which cause the CPU to crash, requiring a reboot. Fortunately, a watchdog-timer system can be implemented to automatically reboot the CPU in this case.

It should be emphasized that these errors occurred during *intentional injection of radiation-errors* for testing hardware resilience. During normal operation, the system has entirely deterministic behavior and computational results accurately match the input data (up to the limits of the analog signal noise floor at 80 dB below maximum).

V. CONCLUSIONS

A hardware prototype is currently available to detect 0^+ whistler signals. This prototype emulates the flight-deployable system in many ways, but before it is flight-worthy, it will require more information on the data input and output format from the TARANIS MEXIC system, as well as a conclusive decision on the exact FPGA to be deployed in flight.

VI. REFERENCES

- [1] Moussa, Nader. "Detection, Characterization, and Categorization of 0^+ Events for TARANIS." Stanford University EE391 Report, April 2008.
- [2] M.P. Johnson and U.S. Inan, "Sferic Clusters Associated with Early/Fast VLF Events." *Geophysical Research Letters*, vol. 27, no. 9, pgs. 1391-1394. May 1, 2000.
- [3] Altera DE2 Development and Education Board. Altera Corporation, available at: <http://www.altera.com/education/univ/materials/boards/unv-de2-board.html>
- [4] Altera Cyclone II FPGA Product Family. Altera Corporation, <http://www.altera.com/products/devices/cyclone2/cy2-index.jsp>
- [5] Wolfson Microelectronics WM8731 Data Sheet. Retrieved 15 Sept 2008. <http://www.wolfsonmicro.com/products/WM8731/>
- [6] Stanford FPGA MFA Interfaces Proposal, February 2008. P. Zamora, CNES-Orléans.
- [7] M. Platino, U.S. Inan, T.F. Bell, M. Parrot, and E.J. Kennedy. "DEMETER observations of ELF waves injected with the HAARP HF transmitter." *Geophysical Research Letters*, vol. 33. August 17, 2006.
- [8] R. A. Marshall and U.S. Inan, "Possible direct cloud-to-ionosphere current evidenced by sprite-initiated secondary TLEs." *Geophysical Research Letters*, vol. 34. February, 2007.
- [9] Robert Helliwell. *Whistlers and Related Ionospheric Phenomena*. Dover Books, 1965.
- [10] R.L. Smith and J.J. "Angerami. Magnetospheric Properties Deduced from OGO 1 Observations of Ducted and Nonducted Whistlers." *Journal of Geophysical Research*, vol. 73, no 1, pgs. 1-20. January 1, 1968.
- [11] Richard L. Burden and J. Douglas Faires. *Numerical Analysis, 8th Edition*. 2005. Excerpts available at North Carolina State University MA427 Numerical Analysis course web site: <http://www4.ncsu.edu/~kksivara/ma427/ma427outline.html>
- [12] CNES TARANIS web site: <http://smc.cnes.fr/TARANIS/>

VII. APPENDIX

A. List of Figures

Fig. 1 – Visualization of the spectral band matrix in MATLAB simulation. The output levels of each boxed region are summed, and then the entire grid is matrix-multiplied against a pattern matrix of constant values.**Error! Bookmark not defined.**

Fig. 18 – Example output of the Low Frequency Spectral Pattern Matching 0^+ system. Note that long-dispersion whistlers and other events which do not fit the correct profile are neglected.**Error! Bookmark not defined.**