"A PROPOSAL FOR A SPACE FLIGHT DEMONSTRATION OF A DYNAMICALLY RECONFIGURABLE PROGRAMMABLE MODULE WHICH USES FIRMWARE TO REALISE AN ASTRIUM PATENTED COSMIC RANDOM NUMBER GENERATOR FOR GENERATING SECURE CRYPTOGRAPHIC KEYS"

Adam Taylor, Peter Bennie, and Fredric Guyon

EADS Astrium, APP, Stevenage, Hertfordshire, SG1 2AS, UK

T: 00 44 1438 77 3000

Iain Cameron, and James Glanfield

EADS Astrium, APP, Portsmouth, PO3 5PU, UK

T: 00 44 23 9270 5705

Omar Emam

EADS Astrium, ENS, Stevenage, Hertfordshire, SG1 2AS, UK

T: 00 44 1438 77 4277

E-mail: adamp.taylor@astrium.eads.net , peter.bennie@astrium.eads.net, fredric.guyon@astrium.eads.net, iain.cameron@astrium.eads.net, james.glanfield@astrium.eads.net, omar.emam@astrium.eads.net

INTRODUCTION

This paper describes a proposal for a space flight demonstration of a low power, compact Dynamically Reconfigurable Programmable Board (DRPB) based upon a minor evolution of the Astrium Janus payload for UKube 1. The Janus payload is one of a number of the payloads selected to be part of the first national UK-Cube satellite (UKube) [1] to be sponsored by the UK Space Agency. In the UKube configuration the demonstrator performs two experiments the first uses firmware to realise an Astrium patented cosmic random number generator for generating secure cryptographic keys while the second monitors the large high performance SRAM based FPGA for SEU and SEFI events allowing correlation with predicted upset rates. This experiment is called the Janus experiment after the two-faced roman god of beginnings and transitions, transitioning from clear text to encrypted and marking the beginning of flying advanced FPGA's on suitable missions.

Discussion of current Janus Capabilites

The UKube mission allows engineers to gain valuable experience designing and demonstrating new technology concepts at low cost and within a short period of time. Typically a cube satellite payload programs take six to twelve months from specification to state where it is ready for flight, compared to a two years for a traditional medium to high cost satellite. In this case, the Janus payload is a new concept intended to demonstrate the use of volatile configuration memory re-programmable FPGAs in a space application. This technology demonstrator will enable taking the next evolutionary step of implementing, in flight, full /partial re-

configurability in a follow up demonstrator.

Like any semi-conductor device both the user logic / memory and configuration memory for these FPGAs are susceptible to SEUs which may in certain circumstances result in a change to the devices programmed functionality. While this is rare it is necessary to develop mitigation schemes and a number of studies have been carried out in order to develop techniques which deal with such upsets and ensure the reliable operation of the system. One such study is the ESA funded DRPM studies, of which one [2] is being lead by Astrium-UK working together with DPI-Germany but such work is outside the scope of this paper.
The payload will provide valuable information on the effect of SEUs on such systems by counting SEU events in the Flip Flop, RAMS, Digital Clock Managers etc. to determine how often we see these and if they correlate with performance during on ground SEU testing. The more important experiment is the one to demonstrate the feasibility of the Astrium patented cosmic random number generator for generating secure cryptographic keys [3].

Existing Payload Architecture

As the proposed DRPB is based so closely upon the Janus architecture a detailed explanation of the existing architecture follows; being targeted at a Cubesat application which is such a small satellite, brings with it its own challenges, namely in the power budget, physical constraints and device selection.

- Power Budget – Sunlight average 400 mW
- Physical constraints – Limited to 310g for the payload PC104 sized, with a height restriction of 35 mm
- Device selection – Flight-grade components are expensive and need to be selected such that they will allow the power budget to be achieved.

Ukube1 payloads have no redundancy and this satellite has a specified in-orbit mission life of only 12 months. The architecture (figure one) developed for the Janus payload was based around two FPGA's: a lower-performance but radiation-hard one-time programmable RTAX2000 and a Xilinx XQR4VSX55. The roles of the FPGA's were split, with the RTAX performing the random-number experiment (in conjunction with a large SRAM) and monitoring the Virtex-4 device for SEU or SEFI events. It is the RTAX which will communicate with the Ukube1 mission interface computer controlling the satellite over the I2C bus. The XQR4VSX55, for its part, is the subject of the second experiment—the flying of a high-performance FPGA in space.
The power architecture of the UKube1 provides three regulated rails to each payload at 3.3V, 5V and 12V, with a maximum acceptable current of 600 mA to be drawn from any one of these supplies. The sunlight average must remain at 400 mW. However, it is permissible to draw more power but operate for a shorter period of time. Therefore, achieving the power budget is one of the major challenges of this mission.

Of course, on future missions who are not power limited the power dissipation of the payload of less than

4 Watts worse case is not prohibitive for the functionality offered.

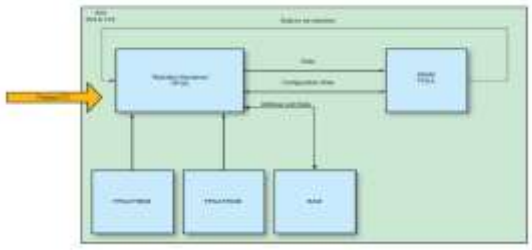

Figure 1: The Ukube Astrium Payload Architecture Concept

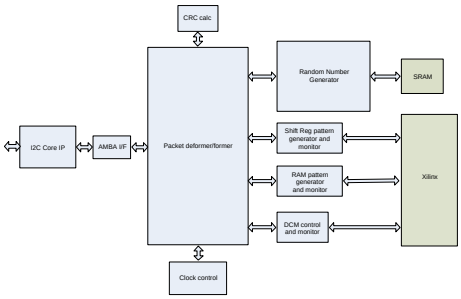

Figure 2: The Internal Architecture of the RTAX FPGA

FPGA selection for the Janus project was driven by factors of cost, as this was a project with a limited budget. We would use an FPGA that we held within stock as long as it was suitable for the first experiment, and could control and monitor the second, high-performance FPGA. For these reasons, an RTAX2000 was selected. Selection of the second FPGA was a little more complicated, being limited by the power the device required and what could feasibly be supplied by the UKube1 power architecture, the physical size of the FPGA and, just as important, what Xilinx had available within our time scales. The final device selected was the Virtex-4 XQR4VSX55.

The power architecture of the design was complicated by not only the power and current limits, but also by the number of voltage rails required. The first experiment required 1.5V and 3.3V, while the second experiment required four voltages: 1.2, 2.5, 3.3 and 1.8V. To ensure maximal performance from the experiment it was decided, due to the power required, to endow the payload with the ability to operate the random-number experiment either on its own, with the second experiment powered down, or in tandem with the second FPGA. This would give the mission controllers the maximum flexibility in terms of scheduling of the payloads and when they were operating. To ensure the power requirements could be achieved, high-efficiency switching converters were used to minimize losses in the voltage regulation for the FPGAs.

What challenges faced by the SRAM FPGA

As the proposed demonstrator is to provide reconfiguration in orbit using a SRAM based FPGA this proposal must also address how the functionality of the FPGA will be ensured in the presence of SEUs. It is hoped data from the Janus experiment can be used to correlate the occurrence of these events with the predicted model to provide a more accurate estimate for the DRPB.

When considering soft errors which can affect the user logic / memory and the configuration memory the engineer must consider the following.

Single Event Transients – are caused when a neutron effects signal line or combinatorial function it creates a temporary glitch on the line, if this occurs on a line near where a clock edge samples that signal there is the potential for it to be clocked into a register hence creating a SEU.

Single Event Upsets – This is what people generally think about when they consider single event effects, a SEU flips the state of a register or memory. This could occur within the

Block or Distributed RAM, User Flip Flops or FPGA configuration memory. These events can be corrected using detection and error correction schemes if necessary or if on data paths e.g. filters they will clear on the next sample.

Single Event Functional Interrupt – This is similar to a SEU in that a bit is flipped however normal operation of the device can only be recovered by power cycling the device or a complete reconfiguration of the devices. When considering the Virtex 4 device used on this application there are four potential causes [4]

1. Power On Reset – A global reset of all internal storage.
2. Select MAP interface – Loss of read / write ability through the Select Map interface.
3. Frame Address Register – FAR increments uncontrollably.
4. Global Signal – Corruption of the Global Set / Reset, Global Write Enable etc.

Of the SEFI listed above 50% will result in the user application being impacted (1 & 4) while the others will prevent the ability to detect further SEFI, without impact on the user application. However, to prevent the accumulation of errors a reconfiguration is required.

Multi Bit Upset – This is when an SEE effects more than one register or memory element, many mitigation techniques can detect and correct MBU's.

Mitigating the selected FPGA

FPGAs for space flight will often have a specified SEFI / SEU rate. For the Xilinx V4QVSX55 the rate for SEFI is determined as 1.5e-6 Upsets per device per day when working in a GEO orbit. This equates to one SEFI every 666666 days or 1825 years between events [5]. While this number is substantial this accounts for SEFI only and not SEUs affecting the configuration or user spaces. As the Virtex 4 is radiation tolerant as opposed to radiation hard both of these spaces will be subject to SEU and the user is responsible for putting in place mitigation strategies.

Mitigation of the user design can be implemented using techniques such as Triple Modular Redundancy using tools like XTMR, BL-TMR or Synplify premier. The design engineer could implement this by hand within the RTL. Block RAMS can use error detecting and correction codes and scrubbing to ensure the contents of the RAM are not corrupted. Of course, implementing these features within an FPGA will result in a reduced logic footprint available and this also has the potential to impact the maximum clock frequency achievable. If availability is key, and the FPGA cannot be taken off line due to the time taken to correct for a SEFI, then triplicated FPGAs and a radiation hardened voter must be used. Alternatively if the mission can allow the FPGA to be reconfigured periodically then analysis may show that no mitigation is required.

Mitigation of the configuration memory requires a more in depth approach and in this case requires a supervising FPGA. When using later family devices this monitoring device may or may not be required. Detection of SEFIs can be achieved by implementing with ease monitoring the status of the FPGA done pin, monitoring the FPGA busy pin, monitoring the status and control

registers and reading and writing a predefined value into the FAR register. All of the above can be easily implemented within the Control FPGA provided.

Of course not all of the FPGA configuration memory is used within a user design and therefore, there is the potential for an SEU to occur within the configuration memory and for it to have no impact upon the user design. However, the accumulation of events within a design could lead to failure and as such errors must be corrected. For this reason a configuration scrubbing algorithm is required.

Dynamic Reconfiguration

The Janus payload is an ideal platform for demonstrating further in orbit dynamic reconfiguration. Using the two flash devices each of which contains the initial FPGA programme the control FPGA can then store an updated FPGA programme within the SRAM memory which could contain an algorithm update or an entire new function. As the memory is actually 64 Mb and the configuration data stream for the FPGA is only 22,708,160 bits it is possible for the control FPGA to have two complete different programmes for the FPGA or numerous partial reconfiguration programmes for the FPGA stored which could be selected under the control of the platform and the FPGA totally / partially reconfigured. The Janus payload includes two PROMS both of which contain duplicates of the same programme in case of failure of one. This architecture easily lends itself to one which adds scrubbing and reconfiguration in flight if required. The choice of full device or partial reconfiguration is then up to the end user.

What Modifications are required?

To enable the upgrade of the Janus experiment to function as the DRPB a minor schematic upgrade would be required to change the configuration mode from the Serial Slave to Select Map which will allow the SEFI and SEU scrubbing algorithm to be implemented. Redesign of the Virtex 4 IO to support the required mission interfaces would also be required. Changes may also be required to the current space craft bus interface which for the Janus experiment is I2C. These are however minor updates to the schematics. It is expected that most upgrade effort would be focused on the RTL design to support the new scrubbing algorithm and to code the initial function of the virtex FPGA.

System Level Analysis

SRAM FPGAs have been subjected to numerous research efforts looking in to the effect of SEUs on configuration memories and user logic. This platform offers several opportunities for correlation of both vendor developed tools such as XTMR, the Isolated Design Flow process and potentially essential bits technology (This would require support from Xilinx to enable this feature on the Virtex 4 QV). The performance of these tools could be compared against third party independent tools such as STAR which determines the number of sensitive bits. Therefore agreement between STAR and Essential bits tools would be very positive. A comparison of VPlace against the XTMR and Isolated design flow to see the effects on sensitive configuration bits would also be beneficial. These comparisons would use the same base design; tools like FT-Unshades 2 could be used to determine the performance impacts of vendor and third party tools.

The Random Number Generator:

The Janus experiment along with collecting performance statistics of the Virtex 4 FPGA also implements a true random number generator using SEU based upon a Astrium held patent [xx]. While the proposed DRPB would prevent the utilisation of the existing SRAM memory for the true random number generator function as this would be used to store the FPGA updated configuration data. The virtex FPGA does contain 5.7 Mb of RAM therefore one potential configuration of the FPGA could be to implement the same true random number generation algorithm however being transplanted from the RTAX FPGA and the SRAM into the virtex FPGA. This approach would require the block RAMS used for this experiment were not protected by an error detection and correction code.

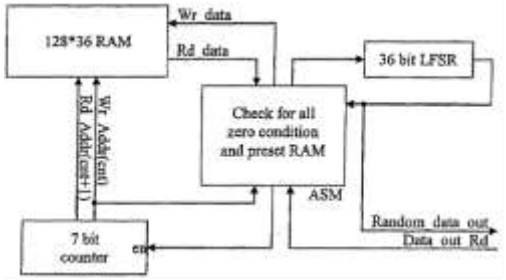

Figure 3 : The Astrium Random Number Generator Architecture Concept

Conclusion

In conclusion the Astrium Janus payload delivered for UKube 1 provides all of the hooks required with a minor design update to be expanded to utilise a dynamic reconfiguration in flight. The control FPGA is more than capable of supervising the Virtex FPGA and monitoring for SEFI and SEU. Both the Flash memory provided and the SRAM can be used to store the FPGA application (flash initial programme) and the SRAM the uploaded dynamic programme. This dynamic update could replace the entire device function or use partial reconfiguration to fine tune the algorithm. Crucially the platform developed is low cost, low power and mass. We currently have sufficient spares from the Janus payload to construct another module, which could include dynamic reconfiguration and we are interested in opportunities to demonstrate this in orbit.

References

[1]http://www.bis.gov.uk/ukspaceagency/missions/ukube-pilot-programme

[2]http://www.esa.int/TEC/OBDP/SEMM6K5OJCG_0.html

[3] Patent 20090316898 Random Number Generation

[4] Xilinx application note XAPP1088

[5] VIRTEX-4VQ STATIC SEU CHARACTERIZATION SUMMARY http://trs-new.jpl.nasa.gov/dspace/bitstream/2014/40768/1/08-16.pdf