Reliability Prediction of a Fault Tolerant COTS Parallel Computing Payload

Sharon Lim Siok Lin¹  Mok Yin Liong²
1) Satellite Engineering Centre, Nanyang Technological University, Singapore
2) DSO National Laboratories, Singapore
Fault Tolerant, High Performance Parallel Processing Unit (PPU) onboard XSat

A 120 kg Polar and Sun-synchronous LEO micro-satellite developed by CREST

Collaboration between Nanyang Technological University of Singapore and DSO National Laboratories, Singapore

Remote-sensing mission
Launch date: 2009
Piggy-back launch with PSLV at Sriharikota (SHAR)

Primary Mission Payload:
IRIS Multispectral Camera (10m resolution)

Secondary Mission Payloads:
Parallel Processing Unit (PPU)
DLR GPS Module
PPU Overview

Secondary Mission Payload:
Parallel Processing Unit (PPU)

Objective:
To demonstrate a low-cost, fault tolerant and high performance payload for a micro-satellite such as XSat.

Research Rationale:
To research on enabling technology for low cost small satellites to have a high performance and reliable parallel processing capability onboard.

To enable a new class of applications formally not performed on microsatellites.
Design Philosophy:

- Based on the usage of COTS components to take advantage of their
  - Lower cost
  - Ease in parts procurement and ITAR restrictions
  - Higher computation power (e.g. COTS processors are in general a few magnitudes higher in computation power than their space-grade counterparts).
  - Reduced power consumption, mass and volume than their space-grade counterparts.

- Uses parallelism to increase computation power as well as for fault tolerance through hardware redundancy (mixed redundancy).

- Ensures reliability through the incorporation of hardware redundancy and software/hardware fault-masking techniques
Project Goals

- A computing payload designed strictly within constraints of mass, volume, power and costs of XSat Project, which are typical of a micro-satellite remote sensing mission.

- Performance/power ratio of at about 200 MIPS/watt, scalable up to 40 watts.

- A demonstrator of a fault tolerance methodology that can achieve a hardware reliability of at least 0.9 for 1GIPS of computation power.

- Provision of a familiar application environment for users (e.g. Beowulf Linux Cluster in space)

- A software reconfigurable payload that function as a test bed for new applications and various software fault tolerant techniques.

- A platform to enable new class of applications for low cost small satellites (e.g. continuous surveillance or disaster monitoring).
1. Hardware Architecture

**4 FPGA Network Elements** (AX1000-fg484)
- Four Actel antifuse FPGAs (AX1000-fg484)
- Each FPGA has a 25 wire interface with two neighbouring FPGAs in a square matrix configuration

**Connected to each FPGA**
- Five Intel StrongArm SA1110.
- Three Serial Flash Chips (Atmel AT45DB642D-CNU)

**Connected to each Processor Node (SA1110)**
- Two 32 MB SDRAM (Samsung K4S561632H)
- Two power-on transistor switches for its 1.8V core voltage and 3.3V IO voltage.

Linux Beowulf Distributed Parallel cluster of COTS processors
2. Communication Network Architecture

The FPGA provides a simple one-way ring-type bus henceforth referred to as the Variable Time-slotted Global Backplane (VTGB). This Packet Network spans across four FPGA clusters and has a one-way ring network topology.

**VTGB Short Message Format**
- Destination (8-bit)
- Source (8-bit)
- Check (16-bit)
- Command (7-bit)
- Data (24-bit)

**VTGB Long Message Format**
- Destination (8-bit)
- Source (8-bit)
- Check (16-bit)
- Data (24-bit)

Supports physical/virtual addressing schemes
Supports variable-length packets
Supports one-to-one or broadcast communication mode
3. Software Platform

- COTS Open-source software suite (blob boot-loader, open-source Linux Kernel Version 2.6.4)

- Current design does not limit migration to use of other software suite, like VxWorks operating system …etc.

- A set of triple-redundant serial flash in each cluster stores all the SA1110 bootloader code, Kernel and ramdisk images that are used to program the processor.

- The codes in the serial flash can be dynamically uploaded in orbit.
Component Selection

Parts Selection
- Preference of Components with Space Heritage.
- Preference for components with Radiation Data
- Select components which provide savings in power, space, cost and mass.
- Wide usage of COTS BGA components
- Usage of processors with high MIPS/watt and MIPS/cost

<table>
<thead>
<tr>
<th>Role</th>
<th>Part</th>
<th>Heritage</th>
<th>Availability of Radiation Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>PN</td>
<td>StrongArm SA1110</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>NE</td>
<td>Actel AX1000S</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Volatile Memory</td>
<td>Samsung K4S561632H SDRAM</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Non Volatile Memory</td>
<td>Atmel Serial Flash</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>DC-DC converter</td>
<td>Interpoint MTR283R3SF</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
Component Selection

- PN selection: Why SA1110
  - High Performance Per Watt and High Performance/Cost
  - Predecessor SA1100 used onboard SNAP1 nanosatellite

- Network Element selection: Why Actel AX1000
  - More resilient towards single-event-upsets (SEUs) as its internal architecture will not be affected by SEUs (cells are programmed by blowing physical fuses).
  - However there will be SEUs in the registers and especially in the internal SRAM (which has very low SEU threshold)

<table>
<thead>
<tr>
<th>Radiation Specs</th>
<th>RTAX1000S</th>
<th>AX1000S</th>
<th>X-Sat Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEU LET(_{\text{th}}) (MeVcm(^2)/mg) (\xi)</td>
<td>63</td>
<td>1.4 (for SRAM)</td>
<td>37 (satellite industry benchmark)</td>
</tr>
<tr>
<td>SEU Rate (errors/flip flop/ day)</td>
<td>(6.7 \times 10^{-9})</td>
<td>(8.9 \times 10^{-9})</td>
<td></td>
</tr>
<tr>
<td>SEL LET(_{\text{th}}) (MeVcm(^2)/mg) (\xi)</td>
<td>117</td>
<td>120</td>
<td>40</td>
</tr>
<tr>
<td>Single-Event Transient (SET)</td>
<td>No anomalies up to 150MHz</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>Total Ionizing Dose</td>
<td>300 krad (Si)</td>
<td>200 krad (Si)</td>
<td>10 krad (Si)</td>
</tr>
</tbody>
</table>
Samsung K4S561632H SDRAM

- Based on “IEEE 2006 Radiation Workshop” published results on radiation testing for its predecessor, K4S561632E-TL75 chip.

- Samsung K4S561632H SDRAM 256Mb is estimated to have a MTBF of 27 days and to be latch-up immune.

- Based on (255,223)RS EDAC and assuming a probability of an uncorrectable upset per day < $10^{-6}$, the memory scrubbing interval is computed to be greater than the operational period of the payload.

- Hence scrubbing is not required for PPU operation for this level of reliability.
PPU Fault Tolerant Methodology

PPU fault tolerance methodology lies in adopting five main strategies

1. Having no single point of failure through hardware redundancy for all modules (either 1 for 1 redundancy or N for K redundancy where N > K).

- 20 Redundant processing nodes
- 4 Redundant network elements
- Dual Redundant Command Bus Link
- Dual redundant LVDS Data Bus Link
- Dual redundant power supply module
- 4 sets of triple-voting serial flash chips
- 20 distributed memory banks

2. Software Adaptability for the deployment of spare processors for replication of code execution or adopting a N-voting computing paradigm.
   - PPU is totally software reconfigurable and hence adaptive to different mission needs on reliability goals
   - Spare processors can be used to replace faulty processors during faults.
   - Spare processors can also be deployed to run the same program with the primary processor in a triple-voting for higher order voting fashion for more important mission tasks
3. Autonomous Fault detection and Processor Recovery Routine

- The FPGA network element contains the logic to monitor processor health and to control their on/off or reset states
- Each network element maintains a watch-dog timer for all the processors in its cluster.
- Processors have to write to FPGA health register at regular intervals to indicate that it is “alive”
- Each FPGA network element has the ability to reset or power-cycle processors that are deemed faulty.
4. Fault Masking Techniques
   • Software EDAC schemes (Hamming/Reed-Solomon etc)
   • Software multiple reboot tries
   • Triple-voting read of the serial flash
   • Dynamic VTGB network reconfiguration capability

Various FPGA VTGB network configuration modes
5. Virtual Resource Management
   • Mapping of virtual requested resource to healthy resources dynamically in runtime
   • External interfaces communicates to entities using virtual addresses
   • Parallel applications uses different virtual addressing schemes to adapt to different parallel processing structures, e.g. logical positions in a mesh logical processing arrays. (see figure below).
   • A specific virtual Address can identify a node with a specific functional role in a Master/Slave programming structure such as the concept of a “System Node”.

A physical-to-logical processor mapping in a faulty n=2 network
Reliability prediction of PPU over a period of 3 years mission operation under Xsat space environment will be assessed in quantitative terms:

**Part 1:**
Derive the failure rates of the various components in the payload using on the methods below (Lamda Predict Software from Reliasoft is used for the failure rates computation)

- Manufacturers’ High Temperature Operating Life (HTOL) test data

**Part 2:**
Compute the overall system reliability of payload by way of constructing a Reliability Block Diagram.

- Reliability modelling breaks a system into its functional blocks and model them as series, parallel or complex configurations, depending on their interdependencies.
- Each functional block has a reliability figure calculated by summing up the failure rates of the components that makes up the block.
Reliability Block Diagram

RBD drawn to reflect all possible scenario paths in the PPU operation.

There are two nodes in the PPU RBD

Upon reaching the first node, it means that there are at least one FPGA, one bank of serial flash, one CAN and one LVDS link operational.

Each of the 20 Paths to the second Node, corresponds to a successful scenario of a StrongARM processor being able to boot up healthily, and with access to at least one CAN or LVDS links.
By plotting the probability of $N$ successful paths to the node, the probability of PPU operating with $N$ healthy StrongArm processors can be obtained.

### Table 1: Probability of $N$ number of SA1110s in Operation in PPU (duty cycle of 10%)

<table>
<thead>
<tr>
<th>$N$ Number of Healthy Operating SA1110s</th>
<th>Failure Rate</th>
<th>Duty Cycle</th>
<th>Effective Failure Rate</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Active</td>
<td>Dormant</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9517</td>
<td>0.0952</td>
<td>0.1808</td>
<td>0.9953</td>
</tr>
<tr>
<td>2</td>
<td>0.9750</td>
<td>0.0975</td>
<td>0.1853</td>
<td>0.9951</td>
</tr>
<tr>
<td>3</td>
<td>0.9790</td>
<td>0.0979</td>
<td>0.1860</td>
<td>0.9951</td>
</tr>
<tr>
<td>4</td>
<td>0.9907</td>
<td>0.0991</td>
<td>0.1882</td>
<td>0.9951</td>
</tr>
<tr>
<td>5</td>
<td>1.0260</td>
<td>0.1026</td>
<td>0.1949</td>
<td>0.9949</td>
</tr>
<tr>
<td>6</td>
<td>1.1980</td>
<td>0.1198</td>
<td>0.2276</td>
<td>0.9940</td>
</tr>
<tr>
<td>7</td>
<td>1.4030</td>
<td>0.1403</td>
<td>0.2666</td>
<td>0.9930</td>
</tr>
<tr>
<td>8</td>
<td>2.3990</td>
<td>0.2399</td>
<td>0.4558</td>
<td>0.9881</td>
</tr>
<tr>
<td>9</td>
<td>4.6487</td>
<td>0.4649</td>
<td>0.8833</td>
<td>0.9771</td>
</tr>
<tr>
<td>10</td>
<td>8.0418</td>
<td>0.8042</td>
<td>1.5279</td>
<td>0.9606</td>
</tr>
<tr>
<td>11</td>
<td>10.139</td>
<td>1.0139</td>
<td>1.9264</td>
<td>0.9506</td>
</tr>
<tr>
<td>12</td>
<td>10.283</td>
<td>1.0283</td>
<td>1.9538</td>
<td>0.9500</td>
</tr>
<tr>
<td>13</td>
<td>10.674</td>
<td>1.0674</td>
<td>2.0281</td>
<td>0.9481</td>
</tr>
<tr>
<td>14</td>
<td>11.330</td>
<td>1.1330</td>
<td>2.1527</td>
<td>0.9450</td>
</tr>
<tr>
<td>15</td>
<td>12.979</td>
<td>1.2979</td>
<td>2.4660</td>
<td>0.9372</td>
</tr>
<tr>
<td>16</td>
<td>16.994</td>
<td>1.6994</td>
<td>3.2289</td>
<td>0.9186</td>
</tr>
<tr>
<td>17</td>
<td>25.540</td>
<td>2.5540</td>
<td>4.8526</td>
<td>0.8803</td>
</tr>
<tr>
<td>18</td>
<td>42.198</td>
<td>4.2198</td>
<td>8.0176</td>
<td>0.8100</td>
</tr>
<tr>
<td>19</td>
<td>72.494</td>
<td>7.2494</td>
<td>13.7739</td>
<td>0.6963</td>
</tr>
<tr>
<td>20</td>
<td>128.80</td>
<td>12.8800</td>
<td>24.4720</td>
<td>0.5256</td>
</tr>
</tbody>
</table>

### Table 2: Probability of $N$ number of SA1110s in Operation in PPU (duty cycle of 100%)

<table>
<thead>
<tr>
<th>$N$ Number of Healthy SA1110s</th>
<th>Failure Rate</th>
<th>Duty Cycle</th>
<th>Effective Failure Rate</th>
<th>Reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Active</td>
<td>Dormant</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.9517</td>
<td>0.0952</td>
<td>100</td>
<td>0.9517</td>
</tr>
<tr>
<td>2</td>
<td>0.9750</td>
<td>0.0975</td>
<td>100</td>
<td>0.9750</td>
</tr>
<tr>
<td>3</td>
<td>0.9790</td>
<td>0.0979</td>
<td>100</td>
<td>0.9790</td>
</tr>
<tr>
<td>4</td>
<td>0.9907</td>
<td>0.0991</td>
<td>100</td>
<td>0.9907</td>
</tr>
<tr>
<td>5</td>
<td>1.0260</td>
<td>0.1026</td>
<td>100</td>
<td>1.0260</td>
</tr>
<tr>
<td>6</td>
<td>1.1980</td>
<td>0.1198</td>
<td>100</td>
<td>1.1980</td>
</tr>
<tr>
<td>7</td>
<td>1.4030</td>
<td>0.1403</td>
<td>100</td>
<td>1.4030</td>
</tr>
<tr>
<td>8</td>
<td>2.3990</td>
<td>0.2399</td>
<td>100</td>
<td>2.3990</td>
</tr>
<tr>
<td>9</td>
<td>4.6487</td>
<td>0.4649</td>
<td>100</td>
<td>4.6487</td>
</tr>
<tr>
<td>10</td>
<td>8.0418</td>
<td>0.8042</td>
<td>100</td>
<td>8.0418</td>
</tr>
<tr>
<td>11</td>
<td>10.139</td>
<td>1.0139</td>
<td>100</td>
<td>10.139</td>
</tr>
<tr>
<td>12</td>
<td>10.283</td>
<td>1.0283</td>
<td>100</td>
<td>10.283</td>
</tr>
<tr>
<td>13</td>
<td>10.674</td>
<td>1.0674</td>
<td>100</td>
<td>10.674</td>
</tr>
<tr>
<td>14</td>
<td>11.330</td>
<td>1.1330</td>
<td>100</td>
<td>11.330</td>
</tr>
<tr>
<td>15</td>
<td>12.979</td>
<td>1.2979</td>
<td>100</td>
<td>12.979</td>
</tr>
<tr>
<td>16</td>
<td>16.994</td>
<td>1.6994</td>
<td>100</td>
<td>16.994</td>
</tr>
<tr>
<td>17</td>
<td>25.540</td>
<td>2.5540</td>
<td>100</td>
<td>25.540</td>
</tr>
<tr>
<td>18</td>
<td>42.198</td>
<td>4.2198</td>
<td>100</td>
<td>42.198</td>
</tr>
<tr>
<td>19</td>
<td>72.494</td>
<td>7.2494</td>
<td>100</td>
<td>72.494</td>
</tr>
<tr>
<td>20</td>
<td>128.80</td>
<td>12.8800</td>
<td>100</td>
<td>128.80</td>
</tr>
</tbody>
</table>
Based on a duty cycle of 10% as would be the case for PPU Mission Operations, reliability figures are as shown.

- Survival Probability = 0.9953
- Nominal Performance Probability = 0.9951
- Peak Performance Probability = 0.9185

However based on a duty cycle of 100%, there is a drastic drop in reliability figure for the Peak Performance State

- Survival Probability = 0.9753
  (This is comparable to the probability of Xsat Onboard Data Handler, a dual redundant ERC32 OBC and its I/O interfaces modules of 0.976)
- Nominal Performance Probability = 0.9743
- Peak Performance Probability = 0.6398

- Several design improvements can be made to increase reliability figure.
Conclusion

- PPU is a COTS Parallel computing payload that has an electronic hardware reliability figure > 0.9 for 3700 Dhrystone 2.1 MIPS of computation power for Peak Performance State (10% duty cycle).

- It is a software reconfigurable payload, capable of uploading new applications on the fly, and for testing the effectiveness of different software fault tolerant schemes.

- It adds value to the XSat mission by providing it with the means to run computation-intensive image processing algorithms to maximise useful image data throughput from the satellite.