FPGA Verification Accelerator (FVAX)

MAPLD 2008

Authors: Jane Oh and Gary Burke
Affiliation: Jet Propulsion Laboratory, California Institute of Technology
Address: 4800 Oak Grove Drive, Pasadena, CA 91109
Email: Gary.R.Burke@jpl.nasa.gov
        Jane.oh@jpl.nasa.gov
Presentation Outline

• Key Challenge
• Objectives and Goals
• Current verification techniques
• FVAX Approach to verification
• Expected Benefits
• Conclusion
Key Challenges

• The increased complexity of the devices required by today’s and future missions
  – the 32k gate devices used on MER
  – the 2 million gate devices used on MSL
• Verification of complex hardware/software systems for space missions is very time consuming
  – a typical FPGA (e.g. the 32k gate devices used on MER) can take 3 months to verify
• FPGAs are becoming a more critical component of space systems
Objectives

• Faster: Reduce the amount of time required to verify critical FPGAs
• Effectiveness: Improved testing to reduce test escapes
• Improved validation: Increased confidence that the FPGA is correct
Task Goals

• Develop a new technology (FVAX, FPGA-based Verification Accelerator System) to improve verification and validation of FPGAs that
  – Works with any FPGA
  – Has more capability than existing tools
  – Provides a standard FPGA BTE for any Board
  – Speeds up verification/validation process by giving
    • High visibility of FPGA internal signals and nodes
    • Easy user interface
    • Comparison of actual with model
    • Method to step to sequence causing the problem
Typical FPGA Verification

- **Simulation**
  - Extensive HDL test benches
  - Model external world

- **Breadboard**
  - FPGA COTS or Custom Board
  - Re-programmable FPGA preferred
  - Use Bench test Equipment (BTE)

- **System test**
  - Engineering Model (EM) in system
  - Run system with software, external hardware

- **Assembly and Test**
  - Assemble final board and test in system.
Current State of Verification - simulation

• Simulation will get most of the problems
  – Time consuming to cover every case
  – Manual effort to build test plan

• Subtle errors remain due to
  – Errors in test bench
    • Test Bench designer same as FPGA designer
  – External world not modeled correctly
    • Asynchronous effects difficult to model
  – Unexpected interaction with other components
    • Incorrect or ambiguous Interface Description (ICD)
Current State of Verification

- board test

• Board test will get most of remaining problems but difficult to find source of problems

• Lack of probe points inside FPGA
  – Need to bring out internal nodes onto unused pins

• Lack of probe points on board
  – Difficult to probe small parts

• Hard to set up error conditions
  – Error may appear only occasionally
Existing FPGA debug

• Existing methods for probing an FPGA during test exist and are effective

• Chipscope is very useful as a way to probe Xilinx FPGAs
  – Uses embedded code compiled with user code
  – Uses JTAG port

• Silicon Explorer
  – Uses FPGA structure to probe any node
  – Uses JTAG/Probe pins
New Debug Tool

• Works with any FPGA
• Has more capability than existing tools
  – See chart
• Provides a standard FPGA test port and BTE for any Board
• Speeds up debug process by giving
  – High visibility of FPGA nodes
  – Easy user interface
  – Comparison of actual with model
  – Method to step to sequence causing the problem
Using FVAX Technology

- Designer integrates FVAX technology (supplied as an IP) into their designs
- Designer simulates and verifies the FPGA (e.g., ModelSim)
- Designer tests and validates the breadboard design using FVAX technology (i.e., FVAX board and FVAX support software)
FVAX System Architecture

Verification Target (FPGA to be verified - provided by customer)

Verification Target IPs (provided by this task, included on Verification Target)

Verification support software (simulation/analysis software to support verification)

Verification FPGA (custom-made FPGA used to facilitate/accelerate verification of target)
## Expected Benefits

<table>
<thead>
<tr>
<th></th>
<th>Silicon explorer</th>
<th>Chipscope</th>
<th>Our Test FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General purpose</strong></td>
<td>No: Actel FPGA only</td>
<td>No: Xilinx FPGA only</td>
<td>Any FPGA/ASIC</td>
</tr>
<tr>
<td><strong>Monitor internal nodes</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>At speed monitoring</strong></td>
<td>Yes: but problem with signal integrity</td>
<td>Yes: but limited sample size</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Number of signals monitored</strong></td>
<td>2-4 only (depends on device type)</td>
<td>Limited by internal memory Typical 32 out of 32</td>
<td>limited by bus bandwidth; typically 32 out of 1024</td>
</tr>
<tr>
<td><strong>Logic analyzer display</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Comparison against model</strong></td>
<td>No</td>
<td>No</td>
<td>Yes: by comparing against the model, problems can be found before they have a major effect on the I/Os</td>
</tr>
<tr>
<td><strong>Static Stimulus</strong></td>
<td>No</td>
<td>Yes</td>
<td>Yes: large number of static stimulus possible</td>
</tr>
<tr>
<td><strong>Dynamic stimulus</strong></td>
<td>No</td>
<td>No</td>
<td>Yes: full pattern generator included. This allows easy setup of conditions leading to problem.</td>
</tr>
<tr>
<td><strong>Internal FPGA resources needed</strong></td>
<td>No</td>
<td>Yes: large amount of on-chip storage needed to store results</td>
<td>Yes: but no onchip storage needed, and on-chip logic is a very small overhead.</td>
</tr>
</tbody>
</table>
Conclusion

• Is Verification Acceleration Possible?
  – Increasing the visibility of the internal nodes of the FPGA results in much faster debug time
  – Forcing internal signals directly allows a problem condition to be setup very quickly

• Is this all?
  – No, this is part of a comprehensive effort to improve the JPL FPGA design and V&V process.