L-3 Fuzing & Ordnance Systems
59th Annual Fuze Conference
May 5, 2016
Quality Driven Complex Logic Development Process

May 3rd - 5th, 2016

Open Session

Nick Adams
Objective and Exit Criteria

• Objective
  – Allow Groups an opportunity to understand the basic technical details of an FPGA, the overall development process and the specific design details that are undertaken by the L-3 FOS Complex Logic design team to provide a quality Complex Logic Device to our customer.

• Exit Criteria
  – Demonstrate an understanding of the FPGA complex logic component
  – Demonstrate an understanding of the development process
  – Demonstrate an understanding of the implementation methodology
Design Methodology

Areas involved in the development of a complex logic design

- Project planning and tracking
- Tool selection in the EDA (Engineering Design Automation) space
- Source control, version control and configuration management mechanisms
- Requirement traceability
- Design strategy and criteria
- Verification methods and coverage metrics
- Validation goals and a release and life cycle management plan

“Do it Once ... Do it Right”
Background Information

What is an FPGA?

An FPGA is **HARDWARE**.

- FPGA is a **Field Programmable Gate Array**
  - It **IS** an Integrated Circuit
  - It **IS NOT** Software
  - It **IS** Programmed
  - It **IS NOT** a program

FPGAs contain programmable components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together".

Logic blocks can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR.

In FPGAs, the logic blocks also include memory elements or registers.
Hardware Descriptive Language (HDL)

In integrated circuit design, Register Transfer Level (RTL) description is a way of describing the operation of a synchronous digital circuit. Behavior is defined in terms of:

- Flow of signals (transfer of data) between hardware registers.
- Logical operations performed on those signals.

RTL abstraction is used in HDL’s or Hardware Description Languages like Verilog to create high-level representations of a circuit, from which lower-level representations and ultimately actual wiring is derived.

- FPGA’s are programmed with HDL

```verilog
classic Example
assign D = ~Q;
always @ (posedge clk)
begin
  Q <= D;
end
```
FPGA Design Methodology

• FPGA design process utilizes a number of supporting procedures and guidelines to facilitate in the development of safe and reliable quality driven complex logic devices.

• These include company as well as project specific procedures and guidelines:
  • FPGA Development Procedure
  • FPGA Design Plan
  • Development Checklist
  • L-3 FOS RTL Coding Guidelines
  • FPGA Directory Structure and Signal Naming Guidelines
  • Internal Design Review Procedure
  • Design Control Procedure
FPGA Design Process

1. Review Customer Requirements
2. Derive FPGA Design Requirements
3. Generate FPGA Design Plan
4. Create FPGA Design Specification
5. Execute FPGA Design
6. Create FPGA Validation Plan and Report
7. Execute FPGA Validate Plan
8. Create FPGA Verification Plan and Report
9. Execute FPGA Verification Plan
10. Release FPGA Design and Documentation
<table>
<thead>
<tr>
<th>Action</th>
<th>Entrance Document</th>
<th>Exit Document</th>
<th>Responsible Engineer(s)</th>
<th>Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review Customer Requirements</td>
<td>Drafted Customer Requirements</td>
<td>Baselined / Released Customer Requirements</td>
<td>Lead Electrical Engineer</td>
<td>SRR</td>
</tr>
<tr>
<td></td>
<td>(SOW, ICD, EICD)</td>
<td>(SOW, ICD, EICD)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Derive FPGA Design Requirements</td>
<td>Baselined Customer Requirements</td>
<td>FPGA Design Requirements</td>
<td>Lead Electrical Engineer</td>
<td>IDR</td>
</tr>
<tr>
<td>Generate FPGA Design Plan (optional)</td>
<td>Baselined / Released Customer Requirements</td>
<td>FPGA Design Plan</td>
<td>Engineering Manager</td>
<td></td>
</tr>
<tr>
<td>Create FPGA Design Description</td>
<td>FPGA Design Requirements</td>
<td>FPGA Design Description</td>
<td>FPGA Design Engineer</td>
<td>IDR</td>
</tr>
<tr>
<td>Create FPGA Verification Plan and Report</td>
<td>FPGA Design Requirements</td>
<td>FPGA Verification Plan and Report</td>
<td>FPGA Verification Engineer</td>
<td>IDR</td>
</tr>
<tr>
<td>Create FPGA Validation Plan and Report</td>
<td>FPGA Design Requirements</td>
<td>FPGA Validation Plan and Report</td>
<td>Lead Electrical Engineer</td>
<td>IDR</td>
</tr>
<tr>
<td></td>
<td>FPGA Verification Plan and Report</td>
<td>FPGA Validation Plan and Report</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release FPGA Documentation Package</td>
<td>FPGA Validation Plan and Report</td>
<td>FPGA Design File</td>
<td>Lead Electrical</td>
<td>IDR</td>
</tr>
</tbody>
</table>

Use or disclosure of data contained on this sheet is subject to the restrictions on the title page of this document.
FPGA Project Tracking

<table>
<thead>
<tr>
<th>Project: &lt;#####&gt;</th>
<th>Charge Number: &lt;#####&gt;</th>
<th>Project Type: Choose item</th>
</tr>
</thead>
</table>

**Complex Logic Development Review Progress Tracking – Sign Off when completed**

<table>
<thead>
<tr>
<th>Drawing Title</th>
<th>Number and Revision</th>
<th>Author</th>
<th>Review Sign-off</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA Requirements</td>
<td>&lt;#####&gt;</td>
<td></td>
<td></td>
<td>Completed</td>
</tr>
<tr>
<td>FPGA Design Description</td>
<td>&lt;#####&gt;</td>
<td></td>
<td></td>
<td>Choose item.</td>
</tr>
<tr>
<td>FPGA Verification Plan/Report</td>
<td>&lt;#####&gt;</td>
<td></td>
<td></td>
<td>Choose item.</td>
</tr>
<tr>
<td>FPGA Programmed Part</td>
<td>&lt;#####&gt;</td>
<td></td>
<td></td>
<td>Choose item.</td>
</tr>
<tr>
<td>FPGA Label</td>
<td>&lt;#####&gt;</td>
<td></td>
<td></td>
<td>Choose item.</td>
</tr>
<tr>
<td>FPGA Programming File</td>
<td>&lt;#####&gt;</td>
<td></td>
<td></td>
<td>Choose item.</td>
</tr>
<tr>
<td>FPGA Design Files</td>
<td>&lt;#####&gt;</td>
<td></td>
<td></td>
<td>Choose item.</td>
</tr>
<tr>
<td>FPGA Validation Plan</td>
<td>&lt;#####&gt;</td>
<td></td>
<td></td>
<td>Choose item.</td>
</tr>
<tr>
<td>FPGA Validation Report</td>
<td>&lt;#####&gt;</td>
<td></td>
<td></td>
<td>Choose item.</td>
</tr>
</tbody>
</table>

**FPGA Requirements Review**

<table>
<thead>
<tr>
<th>Checklist Item</th>
<th>Response</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is the Customer Specification referenced?</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Is this a single or multiple FPGA Design?</td>
<td>Choose item</td>
<td></td>
</tr>
<tr>
<td>Has a maximum FPGA resource utilization been specified?</td>
<td>Choose item</td>
<td></td>
</tr>
<tr>
<td>Is the target FPGA package/device/technology specified?</td>
<td>Choose item</td>
<td></td>
</tr>
<tr>
<td>Is the target FPGA Grade (Commercial/Industrial/Military) or temperature range specified?</td>
<td>Choose item</td>
<td></td>
</tr>
<tr>
<td>Is the design input method specified (schematic or HDL)?</td>
<td>Choose item</td>
<td></td>
</tr>
</tbody>
</table>
FPGA Design Requirements

• Project begins with **Customer Requirements**
  – Externally supplied document for consumption by design team
  – Reviewed by design team; First release point once accepted
  – FPGA design requirements come directly from customer requirements

• FPGA **Design Requirements**
  – Gated by Customer Requirements
  – High level requirements document incorporating [REQ_TAG_ID]
  – “What” does the FPGA need to accomplish in the system
  – “What” are the constraints
  – “What” are the inputs and **how do** they behave
  – “What” are the outputs and **how should** they behave
  – Reviewed by design team; First release point once accepted
  – Risks understood and assessed for all TBD’s
FPGA Design Plan

- Project management resource
- Defines people, methods and strategies to be implemented
  - Scope and Schedule
  - Team members and responsibilities
  - Entrance and exit criteria
  - Meeting and review procedures
  - Status and progress reporting practices
  - Issue / Bug reporting and tracking
  - Methods, tools and conventions utilized
FPGA Design Description

- Gated by FPGA Design Requirements
- Details “How” the design requirements will be met in hardware
- Describes device architecture
- Defines structural blocks functions and interfaces
- Reviewed by design team; First release point once accepted
FPGA Design Strategy

Use or disclosure of data contained on this sheet is subject to the restrictions on the title page of this document.
FPGA Design Execution

• Design execution and verification typically run in parallel
• Design Execution includes:
  – Logic partitioning
  – Design Entry
  – Design Synthesis
  – Block level verification
  – Design implementation
  – Device programming
• Design ensures hardware implements defined requirements
• Verification ensures hardware meets derived requirements
• Verification and Validation
  – Verification is 100% functional coverage done in EDA environment
  – Validation is a subset of Verification done in hardware
    • Not all verification points can be realized in hardware
Verification Plan

• Design verification analyzes a design for proper performance and function as defined in the Architecture Specification.
  – Performed repeatedly during design development.
  – Ensures that the building blocks exhibit required behaviors and then that the entire design exhibits the required behaviors.
  – Done in an EDA environment not in hardware.
    (when we hit hardware we have entered Validation).
FPGA Verification:
Simulation Environment

- Universal Verification Methodology (UVM) version 1.2
- Questa 10.4a simulator
- Separate simulation environments for each FPGA, and a combined simulation environment with all three FPGAs connected together
- Full code coverage and functional coverage collection with links to FPGA requirements document
FPGA Verification

- FPGA Verification Test Plan and Report document contains inspection and simulation test cases identify the requirements that are being verified, the stimulus that is being performed, and the expected behavior of the FPGAs.
- Simulation test environment is also documented in the verification test plan and report.
- Verilog designs are linted as part of the verification.
- Functional Covergroups are used to ensure all simulation test case stimulus and conditions are simulated.
- Verilog Code Coverage metrics are captured to ensure the design is tested.
- Synthesis and Layout configuration and log files are inspected to verify the FPGAs meet timing, have the correct pin-out, device selection, etc.
Verilog Linting with Aldec’s ALINT

- ALINT is a design rule checking software that is used to identify coding style, functional, and structural problems that are difficult to debug in simulators and in hardware.
- FPGA designs are “linted” early in the development as part of the design verification.
- Target Goal: 100% Design Quality (Info, Warning, Critical Warning and Error Free).

<table>
<thead>
<tr>
<th>Rule ID</th>
<th>Severity Level</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALDEC_VLOG.1003</td>
<td>Warning</td>
<td>Avoid unconnected ports</td>
</tr>
<tr>
<td>ALDEC_VLOG.1001</td>
<td>Warning</td>
<td>Constant conditional expressions should be avoided</td>
</tr>
<tr>
<td>ALDEC_VLOG.1004</td>
<td>Warning</td>
<td>Decimal constants should be used for repeat statement expressions</td>
</tr>
<tr>
<td>ALDEC_VLOG.1005</td>
<td>Warning</td>
<td>Do not assign input signals</td>
</tr>
<tr>
<td>ALDEC_VLOG.1007</td>
<td>Warning</td>
<td>At least one loop iteration is required</td>
</tr>
<tr>
<td>STARC_VLOG.1.1.1</td>
<td>Warning</td>
<td>File names should be as follows: &quot;./&quot;.</td>
</tr>
<tr>
<td>STARC_VLOG.1.1.1.1</td>
<td>Warning</td>
<td>Do not use the same instance name or cell name as the ASIC library being used</td>
</tr>
<tr>
<td>STARC_VLOG.1.1.1.2</td>
<td>Warning</td>
<td>Only alphanumeric characters and the underscore '_' should be used, and the first character should be a letter of the alphabet</td>
</tr>
</tbody>
</table>

Design Quality: 87%.

This report was generated based on the following settings:
- Base classification: Rules
- Include unused/disabled rules: No

Violated Rules:

Not Violated Rules:

Use or disclosure of data contained on this sheet is subject to the restrictions on the title page of this document.
FPGA Code Coverage

<table>
<thead>
<tr>
<th>Coverage Report Totals BY FILES: Number of Files 41</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled Coverage</td>
</tr>
<tr>
<td>--------------------</td>
</tr>
<tr>
<td>Stmts</td>
</tr>
<tr>
<td>Branches</td>
</tr>
<tr>
<td>Conditions</td>
</tr>
<tr>
<td>UDP Condition Rows</td>
</tr>
<tr>
<td>FEC Condition Terms</td>
</tr>
<tr>
<td>Expressions</td>
</tr>
<tr>
<td>UDP Expression Rows</td>
</tr>
<tr>
<td>FEC Expression Terms</td>
</tr>
<tr>
<td>FSMs</td>
</tr>
<tr>
<td>States</td>
</tr>
<tr>
<td>Transitions</td>
</tr>
</tbody>
</table>

Total coverage (Code Coverage Only, filtered view): 85.8%

Use or disclosure of data contained on this sheet is subject to the restrictions on the title page of this document.
Issue Tracking and Revision Control

SharePoint Issue Opened

A task was assigned to you by Desch, Noah @ SSG - PE - FOS on 4/6/2016.
Due by: 4/8/2016

Task: Resolution Task - 521 - Validity check of accel response "s" bits not working

Description: Please take a look at the bug report below:

After coming up with a resolution, please fill out additional info.

Please complete the task form located here.

Click below to see the bug report:

Issue is assigned a unique ID
Subversion Revision Control

Issue fixed, design updated and committed into subversion repository for verification

Issue ID’s and Design Revision are linked together in Subversion
SharePoint Issue Tracking

Issue resolved, committed in subversion project repository and reassigned to be verified and closed

Use or disclosure of data contained on this sheet is subject to the restrictions on the title page of this document.

Design Revision is captured in Issue Resolution Comment
Requirement Traceability

ReqTracer from Mentor Graphics

Requirements are tagged

RTL is tagged

Verification tests are tagged

Tags in Requirements, Verification and Design are all linked together
FPGA Package Includes:

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA Programmed Part</td>
</tr>
<tr>
<td>FPGA Label</td>
</tr>
<tr>
<td>FPGA Fuse File</td>
</tr>
<tr>
<td>FPGA Electronic Design &amp; Verification File</td>
</tr>
<tr>
<td>FPGA Design Requirements</td>
</tr>
<tr>
<td>FPGA Design Description</td>
</tr>
<tr>
<td>FPGA Verification Test Plan &amp; Report</td>
</tr>
<tr>
<td>FPGA Validation Plan &amp; Report</td>
</tr>
</tbody>
</table>
Conclusion

The end result is a complex logic design process that approaches or is equivalent to a CMMI (Capability Maturity Model Integration) Maturity level 4, Quantitatively Managed, for product development.

CMMI is a process appraisal program and service required by many DoD and U.S. Government contracts applying to firmware and hardware development for complex logic devices.
L-3 Fuzing & Ordnance Systems
Business Development Department
3975 McMann Road
Cincinnati, Ohio 45245
513-943-2000