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 **F**ield **P**rogrammable **G**ate **A**rray
  - 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.

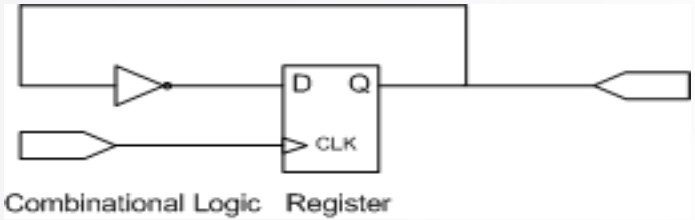
Behavior is defined in terms of:

- 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

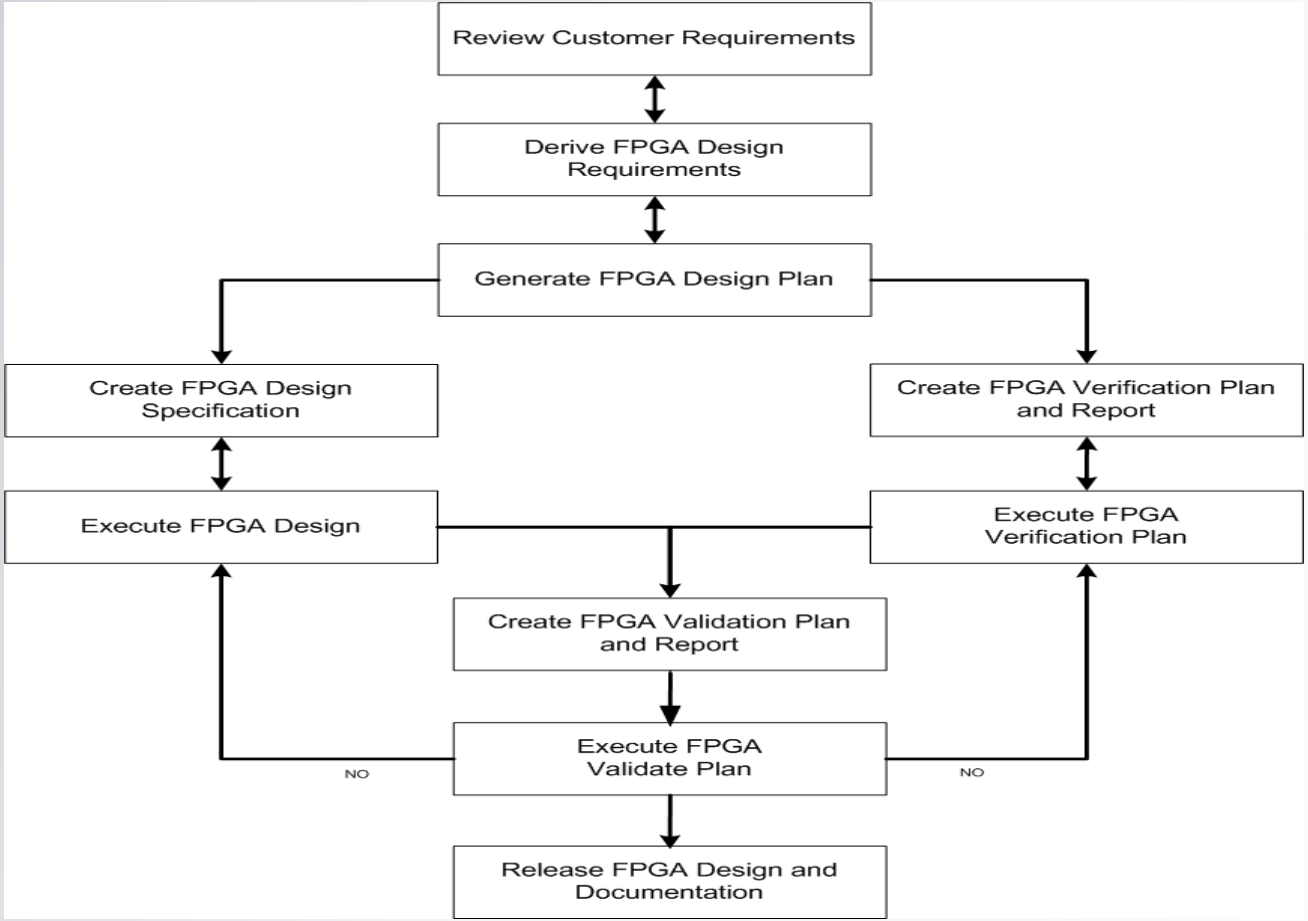
```
assign D = ~Q;
always @ (posedge clk)
    begin
      Q <= D;
    end
```



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



FPGA Development Reviews

Action	Entrance Document	Exit Document	Responsible Engineer(s)	Review
Review Customer Requirements	Drafted Customer Requirements (SOW, ICD, EICD)	Baselined / Released Customer Requirements (SOW, ICD, EICD)	Lead Electrical Engineer	SRR
Derive FPGA Design Requirements	Baselined Customer Requirements	FPGA Design Requirements	Lead Electrical Engineer	IDR
Generate FPGA Design Plan (optional)	Baselined / Released Customer Requirements	FPGA Design Plan	Engineering Manager	FPGA Team Review
Create FPGA Design Description	FPGA Design Requirements	FPGA Design Description	FPGA Design Engineer	IDR
Create FPGA Verification Plan and Report	FPGA Design Requirements	FPGA Verification Plan and Report	FPGA Verification Engineer	IDR
Create FPGA Validation Plan and Report	FPGA Design Requirements FPGA Verification Plan and Report	FPGA Validation Plan and Report	Lead Electrical Engineer	IDR
Release FPGA Documentation Package	FPGA Validation Plan and Report	FPGA Design File	Lead Electrical	IDR

FPGA Project Tracking

Project: <####>	Charge Number: <####>	Project Type: Choose item		
Complex Logic Development Review Progress Tracking – Sign Off when completed				
Drawing Title:	**Number and Revision:**	**Author:**	**Review Sign-off:**	**Status**
FPGA Requirements	**<####>**			Completed
FPGA Design Description	**<####>**			Choose item.
FPGA Verification Plan/Report	**<####>**			Choose item.
FPGA Programmed Part	**<####>**			Choose item.
FPGA Label	**<####>**			Choose item.
FPGA Programming File	**<####>**			Choose item.
FPGA Design Files	**<####>**			Choose item.
FPGA Validation Plan	**<####>**			Choose item.
FPGA Validation Report	**<####>**			Choose item.

FPGA Requirements Review	**Date:**	Click here to enter a date.
Checklist Item	**Response**	**Comments**
Is the Customer Specification referenced?	YES	
Is this a single or multiple FPGA Design?	Choose item	
Has a maximum FPGA resource utilization been specified?	Choose item	
Is the target FPGA package/device/technology specified?	Choose item	
Is the target FPGA Grade (Commercial/Industrial/Military) or temperature range specified?	Choose item	
Is the design input method specified (schematic or HDL)?	Choose item	

FPGA Design Requirements

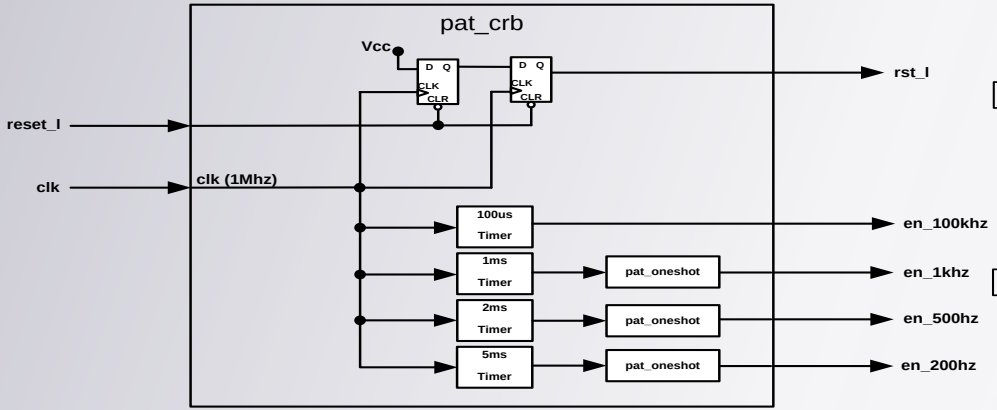
- Project begins with **Customer Requirements**
  - Externally supplied document for consumption by design team
  - Reviewed by design team; First release point once accepted
  - FPGA design requirements come directly from customer requirements
- FPGA **Design Requirements**
  - Gated by Customer Requirements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

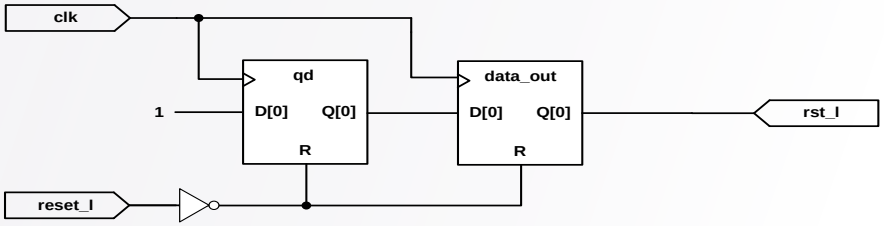
FPGA Design Plan

- 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

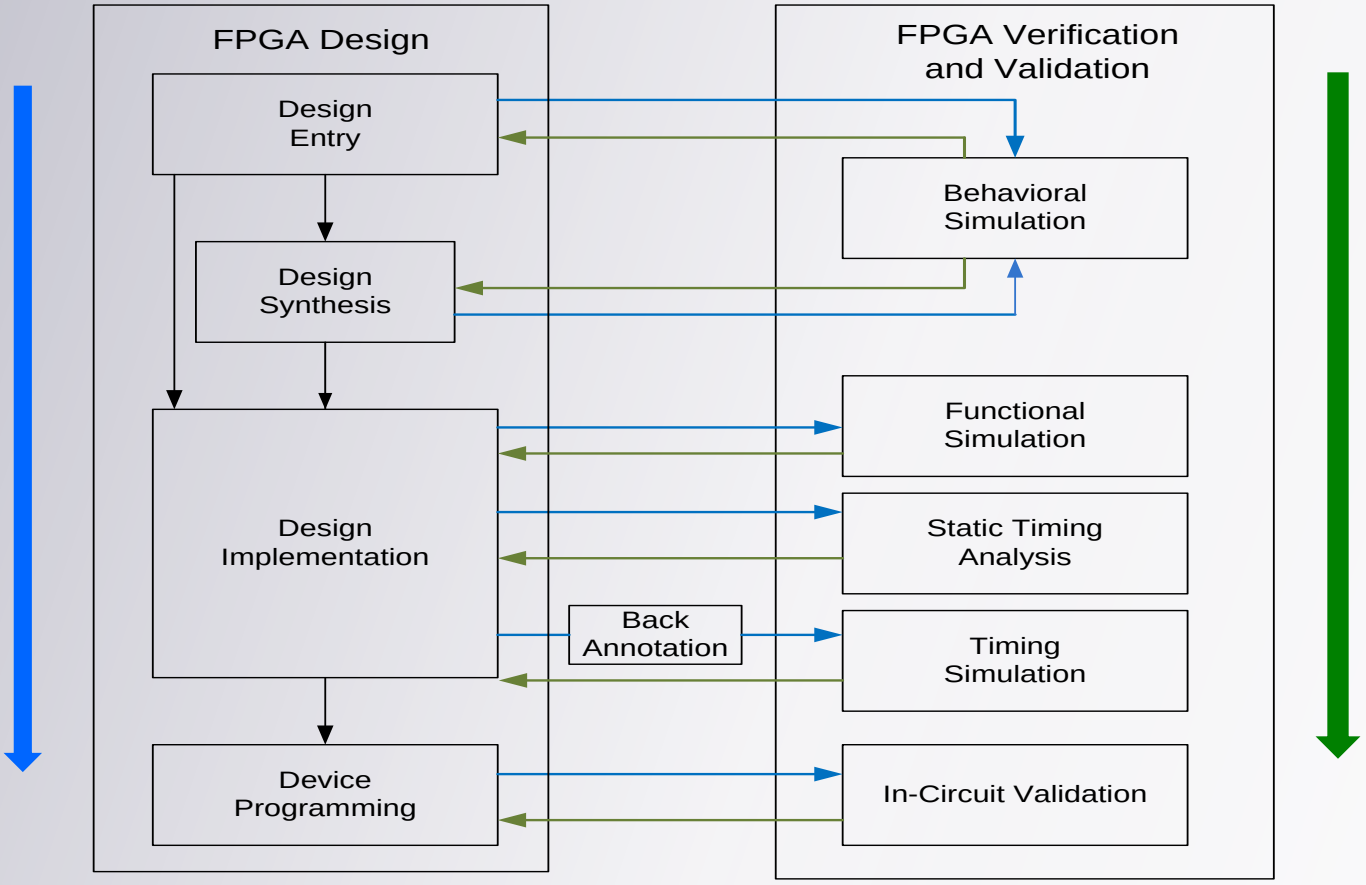
FPGA Design Description

- 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



Independence Maintained Between Design & Verification

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

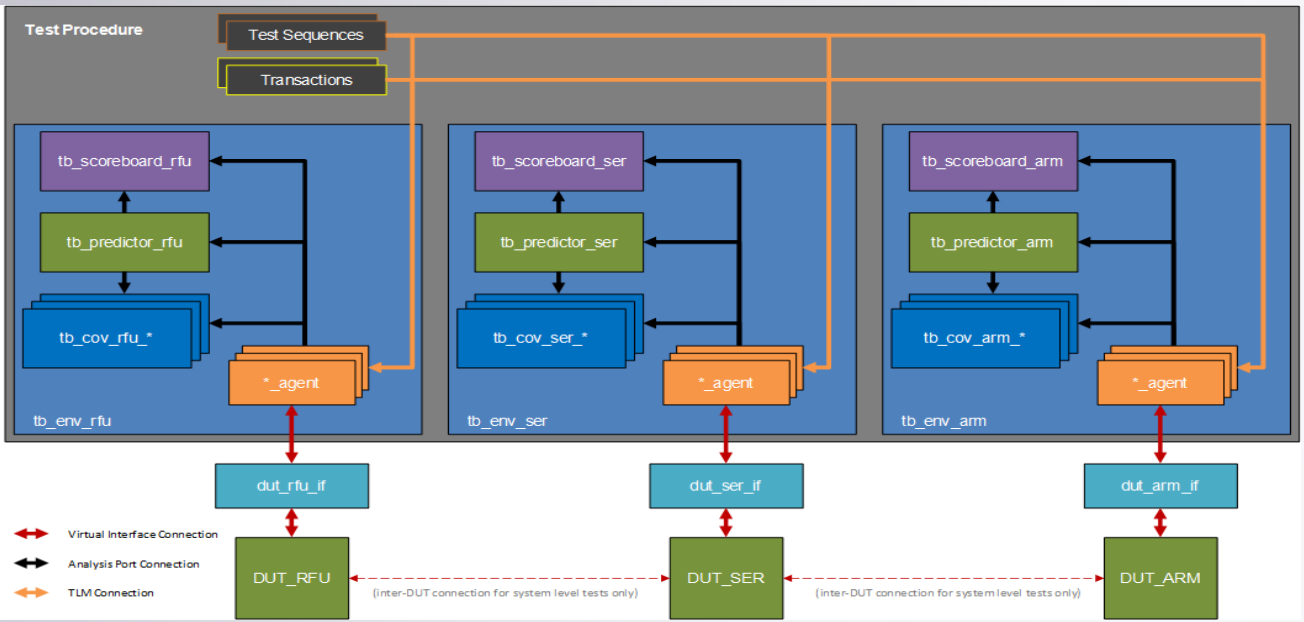
- Design ensures hardware implements defined requirements
- Verification ensures hardware meets derived requirements
- Verification and Validation
  - Verification is 100% functional coverage done in EDA environment
  - Validation is a subset of Verification done in hardware
    - Not all verification 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).

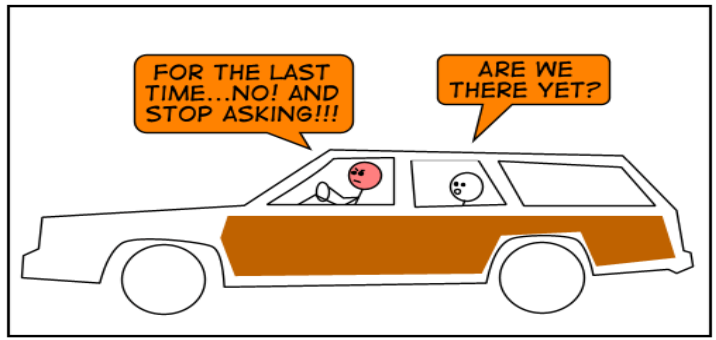
FPGA Verification: Simulation Environment



- 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).

Design Quality: 87%.

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

Violated Rules:

Rule ID	Severity Level	Title
ALDEC_VLOG.1003	Warning	Avoid unconnected ports

...

Not Violated Rules:

Rule ID	Severity Level	Title
ALDEC_VLOG.1001	Warning	Constant conditional expressions should be avoided
ALDEC_VLOG.1004	Warning	Decimal constants should be used for repeat statement expressions
ALDEC_VLOG.1005	Warning	Do not assign input signals
ALDEC_VLOG.1007	Warning	At least one loop iteration is required

...

STARC_VLOG.1.1.1.1	Warning	File names should be as follows: ".v".
STARC_VLOG.1.1.1.10	Warning	Do not use the same instance name or cell name as the ASIC library being used
STARC_VLOG.1.1.1.2	Warning	Only alphanumeric characters and the underscore '_' should be used, and the first character should be a letter of the alphabet

...

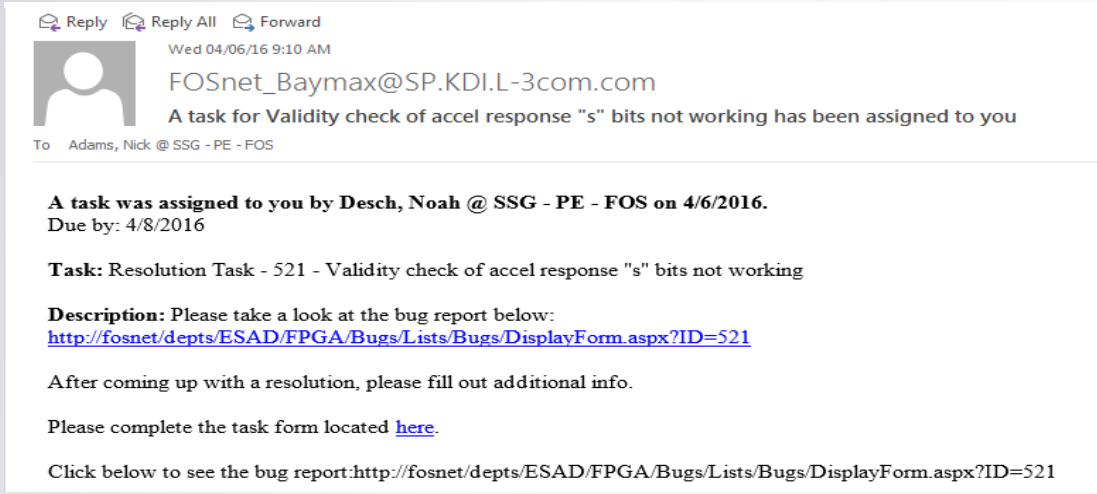
FPGA Code Coverage

```
Coverage Report Totals BY FILES: Number of Files 41

    Enabled Coverage              Active      Hits      Misses     Weight % Covered
    ----------------              ------      ----      ------     ------ ---------
    Stmts                           1726      1641          85          1      95.0
    Branches                        1540      1453          87          1      94.3
    Conditions                                                    1      84.1
        UDP Condition Rows             0         0           0          1     100.0
        FEC Condition Terms          500       341         159          1      68.2
    Expressions                                                   1      88.8
        UDP Expression Rows            0         0           0          1     100.0
        FEC Expression Terms         301       234          67          1      77.7
    FSMs                                                          1      94.2
        States                       116       113           3          1      97.4
        Transitions                  158       144          14          1      91.1

Total coverage (Code Coverage Only, filtered view): 85.9%
```

Instance	Total coverage	Stmt %	Stmt graph	Branch %	Branch graph	State %	State graph	Transition %	Transition graph	FEC Condition %	FEC Condition graph	FEC Expression %	FEC Expression graph
top													
clkgen_arm													
clkgen_ser													
dut_arm	94.2%	98.9%		99%		100%		100%		82.2%		90.7%	
acb	93.9%	97.4%		97.7%		100%		100%		80%		94.4%	
crb	100.0%	100%		100%						100%			
dib	90.0%	98.1%		98.3%						66.7%		97%	
plb	100.0%	100%		100%						100%		100%	
scb	95.2%	99.6%		99.6%		100%		100%		96.3%		80.5%	
tob	97.7%	100%		100%		100%		100%		90.9%			
dut_ser	82.2%	93.7%		92.4%		96.2%		85.7%		59.6%		74.2%	
acb	91.1%	97.9%		98.7%		100%		100%		87%		71.9%	
crb	100.0%	100%		100%						100%			
dib	100.0%	100%		100%								100%	
gcb	89.8%	98.7%		98.7%		100%		100%		78.6%		73%	
mcb	87.0%	96.9%		96.7%		100%		100%		64%		77.5%	
scb	62.0%	82.2%		76.5%		50%		30%		45.2%		66.2%	
tob	73.0%	85.7%		83.3%						50%			

Issue Tracking and Revision Control

SharePoint Issue Opened



Reply Reply All Forward

Wed 04/06/16 9:10 AM

FOSnet_Baymax@SP.KDI.L-3com.com

A task for Validity check of accel response "s" bits not working has been assigned to you

To Adams, Nick @ SSG - PE - FOS

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:
http://fosnet/depts/ESAD/FPGA/Bugs/Lists/Bugs/DisplayForm.aspx?ID=521

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

Please complete the task form located here.

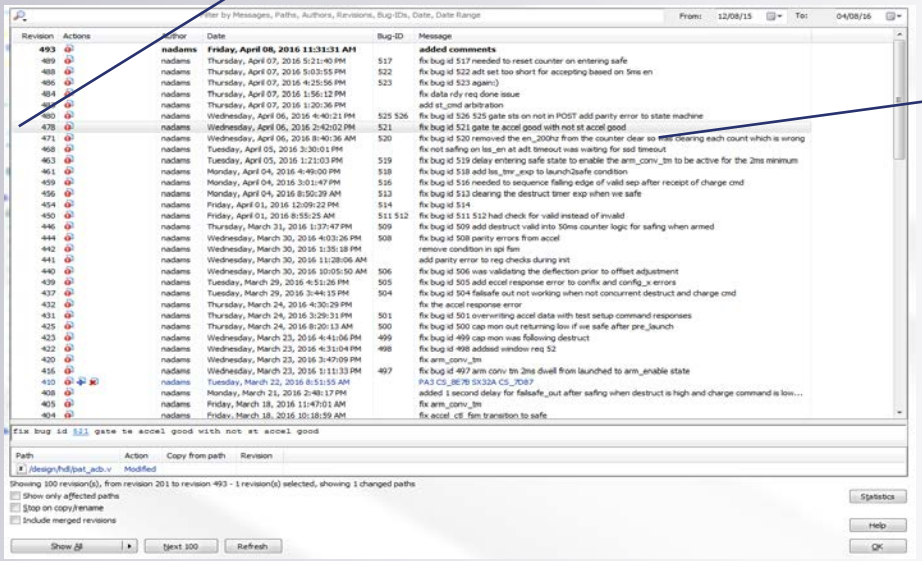
Click below to see the bug report:http://fosnet/depts/ESAD/FPGA/Bugs/Lists/Bugs/DisplayForm.aspx?ID=521

Issue is assigned a unique ID

Subversion Revision Control

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

478	nadams	Wednesday, April 06, 2016 2:42:02 PM	521	fix bug id 521 gate te accel good with not st accel good

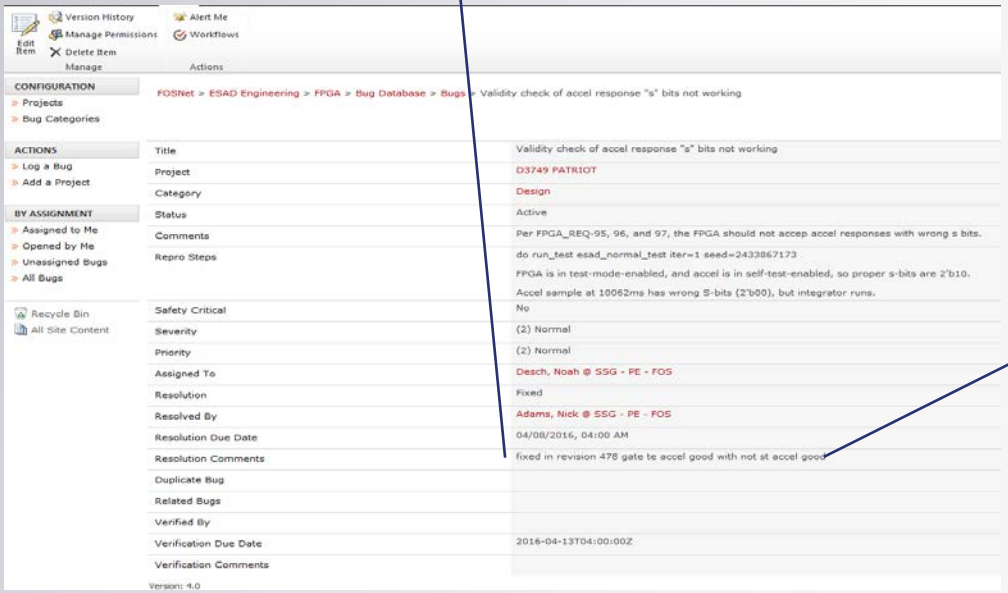


Issue ID's
and
Design Revision
are linked together in Subversion

SharePoint Issue Tracking

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

fixed in revision 478 gate te accel good with not st accel good

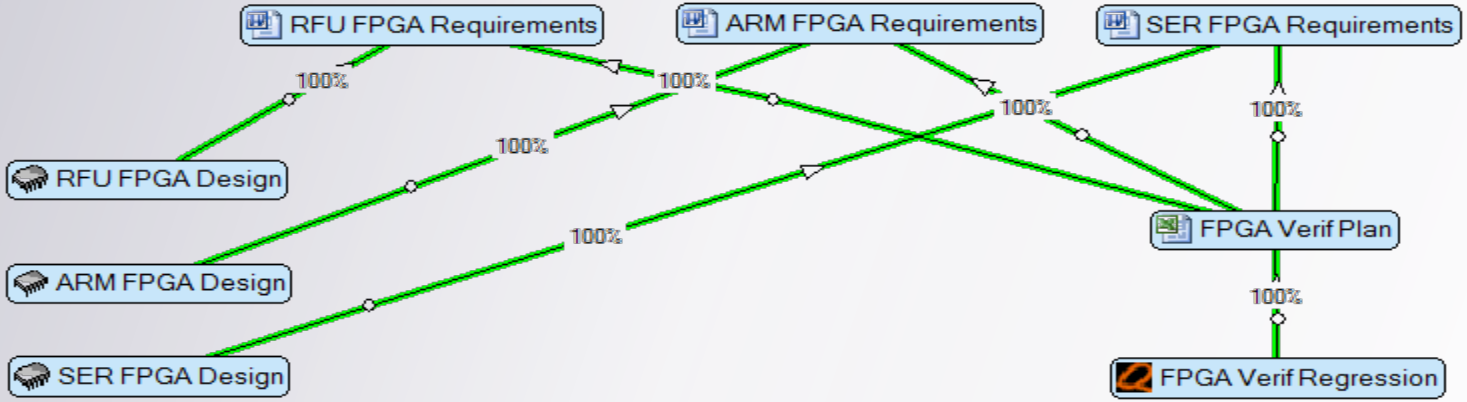


Design Revision is captured in Issue Resolution Comment

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 Documentation

FPGA Package Includes:

Description
FPGA Programmed Part
FPGA Label
FPGA Fuse File
FPGA Electronic Design & Verification File
FPGA Design Requirements
FPGA Design Description
FPGA Verification Test Plan & Report
FPGA Validation Plan & Report

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.

Contact Information

L-3 Fuzing & Ordnance Systems

Business Development Department

3975 McMann Road

Cincinnati, Ohio 45245

513-943-2000