Center-Wide Procedures and Guidelines

DIRECTIVE NO.	500-PG-8700.2.8A	**APPROVED BY Signature:**	***Original Signed by***
EFFECTIVE DATE:	June 3, 2013	**NAME:**	Felicia Jones
EXPIRATION DATE:	January 16, 2021	**TITLE:**	Director, AETD

COMPLIANCE IS MANDATORY

Responsible Office: 500/Applied Engineering and Technology Directorate

Title: Field Programmable Gate Array (FPGA) Development Methodology

PREFACE

P.1 PURPOSE

This document provides a set of Field Programmable Gate Array (FPGA) development procedures and guidelines that ensure that adequate and consistent rules are applied to the development of FPGA designs intended for spaceflight applications. This document also establishes a procedural methodology for the development of FPGA designs that meet design requirements, have performance that can be demonstrated during ground testing, have positive design margins, and will perform with robustness under space environments. It is not concerned with detailed design guidelines, but rather with the development process itself. A separate document referenced in P.4, 500-PG-8700.2.7, covers specific technical recommendations for the design and evaluation of FPGAs and is intended primarily for use by FPGA designers. The two documents are intended to complement each other.

P.2 APPLICABILITY

This procedure applies to the development of Goddard Space Flight Center (GSFC) flight products that include the use of FPGAs. This includes those developments conducted in-house, as well as those performed by second/third-party developers providing products in support of GSFC projects to the extent specified in their contracts.

The procedures and guidelines in this document represent best-practices derived from the development of critical spaceflight electronics as would be found in spacecraft bus avionics. Adherence to the guidelines in this document can be tailored based on the criticality of the mission or complexity of the application. Such tailoring should be made as a result of justification and approval from project leads and/or contractual agreements.

P.3 AUTHORITY

GPR 8700.2, Design Development
500-PG-8700.2.2 Electronics Design and Development Guidelines

P.4 REFERENCES

500-PG-8700.2.7, Design of Space Flight Field Programmable Gate Arrays

300-PG-8730.0.1, Assurance Activities for Digital Electronics for Spacecraft, Instruments, and Launch Vehicles

P.5 CANCELLATION

500-PG-8700.2.8-

P.6 SAFETY

NONE

P.7 TRAINING

NONE

P.8 RECORDS

GPR 8700.2, Design Development, and 500-PG-8700.2.2, Electrical Design Guidelines call out Work Order Authorizations (WOA's) and Verification Test Results as the only official records that shall be retained. Those documents were mainly developed to address the needs of traditional electrical designs involving Printed Circuit Boards (PCB) and their related enclosures, and do not address the unique requirements of FPGA design. This document further expands that list in order to provide a clear definition of the documentation that should be generated for each FPGA design.

Record Title	Record Custodian	Retention
FPGA Design Requirements Document	Project Configuration Management (CM) Officer	* NRRS 8/101. Permanent. Cut off records at close of program/project or in 3-year blocks for long-term programs/projects. Transfer to Federal Records Center (FRC) after cutoff. Transfer to National Archives and Records Administration (NARA) 7 years Or * NRRS 8/103 Temporary. Destroy between 5 and 30 years after program/project termination.
FPGA Design Specification Document	Project CM Officer	* NRRS 8/101 or 8/103

Record Title	Record Custodian	Retention
FPGA Design Architecture Document	Project CM Officer	* NRRS 8/101 or 8/103
FPGA Schematic Drawings/ HDL Code Listing/ Other Design Tool Products/ All Electronic Design Files	Project CM Officer	* NRRS 8/101 or 8/103
FPGA/Board Verification Plan	Project CM Officer	* NRRS 8/101 or 8/103
FPGA Analysis and Simulation Results	Project CM Officer	* NRRS 8/101 or 8/103
FPGA Peer Review Documentation, Requests for Action (RFA's) Generated and Disposition	Project CM Officer	* NRRS 8/101 or 8/103
FPGA Test Results / Includes Oscilloscope Pictures to Document Proper Signal Integrity and Timing	Project CM Officer	* NRRS 8/101 or 8/103

** NRRS – NASA Records Retention Schedule (NPR 1441.1) or as required by the Project Configuration Management System*

P.9 METRICS

NONE

PROCEDURES

In this document, a requirement is identified by "shall," a good practice by "should," permission by "may" or "can," expectation by "will," and descriptive material by "is."

Introduction

The Product Design Lead (PDL), or appropriate design lead authority, shall implement a process that meets the intent of this document throughout the design and development of each FPGA. This responsibility applies to second/third-party developers providing deliverables to the PDL or design authority.

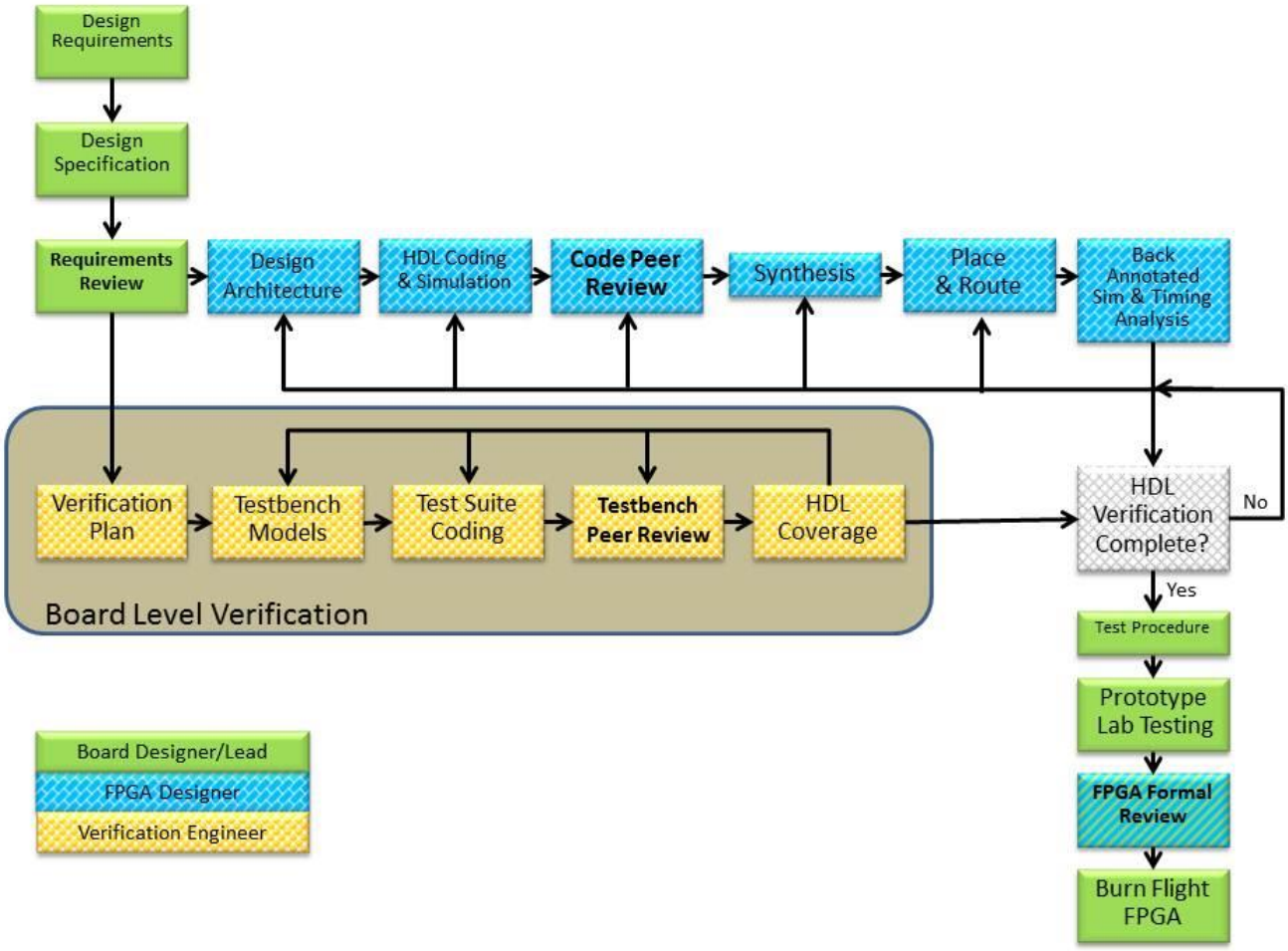
The FPGA Design Process should include documenting a development plan that all developers should follow. This plan should document the systematic approach used to manage, design, develop, test

(verification, validation, and qualification), document, and review all FPGAs delivered. The level of detail can be tailored to the complexity or state of each FPGA design.

The FPGA Development Plan can address:

1. Required documentation (requirements, specification)
2. FPGA development flow diagram
3. Test and flight FPGA part types selection, justification and usage
4. Use of 3rd party or government developed FPGA designs (design-reuse) and/or intellectual property code (includes core selection, management and verification)
5. FPGA design capture process
6. FPGA verification plan, including:
    a. Simulation
    b. Any other analysis methods
    c. Non-flight hardware test configuration(s)
    d. Flight test configuration(s)
7. FPGA computer-aided engineering (CAE) tools including version number
8. FPGA utilization/margin guidelines at specified stages of development for I/O, combinatorial logic, sequential logic, RAMs, cores, etc. as well as internal and interface timing
9. Design Review process
10. Quality Assurance involvement/monitoring/oversight/audit of FPGA development processes
11. Revision control, configuration management and archive processes used at specified stages of development

An example FPGA development flow is shown in the figure below. Appendix D shows an example Flight FPGA Project Timeline that shows how an FPGA design process would line up with typical flight project milestones.



FPGA Design Process Flowchart

1. FPGA Design Requirements Document – The FPGA design process should include a requirements document that compiles all of the derived requirements for the FPGA design. This may be a separate stand-alone document or an embedded section of the board-level requirements document. References to higher level documents can be used. This document will provide traceability for the implementation. The FPGA requirements typically include:

a. Functions to be implemented (i.e. hardware commands, Fault Detection and Correction (FDC), time tags)
b. Performance (speed, critical timing, throughput)
c. Interface description (signal levels, timing, software, data formats)
d. Environmental constraints (thermal, radiation level at part, mission duration)
e. Testability requirements (JTAG, board scan, software, observable internal points)
f. Unique requirement identification number

2. FPGA Design Specification - The FPGA design process should include a specification document detailing the FPGA design functions (what the FPGA does). This document should be used by the FPGA designer as a designer's guide and by others such as software developers. It is analogous to a commercial part's datasheet document. This document should cover topics such as:

a. An introduction and/or scope
b. An FPGA top-level functional block diagram
c. Reset tree and individual reset descriptions
d. Clock tree and individual clock descriptions
e. Data packet descriptions including headers
f. Specific features for mitigating radiation effects (scrubbing, triple module redundancy, etc.)
g. Constraints on board-level implementation (critical pins for board routing, proximity to other devices, input slew-rate limitations, etc.)
h. A description of all interfaces
i. Plan for observability of functions while in a chamber
j. I/O signal names and their description/function
k. I/O pin assignments
l. If there is a software interface to the FPGA, a Software User's Guide (Depending on the complexity of this interface, a separate document may be required for this purpose)
m. Timing diagrams/information
n. Memory maps
o. Register assignments/descriptions
p. JTAG/Debug port description
q. References to the data sheet and relevant application notes used to implement the design
r. List of acronyms/abbreviations.

3. FPGA Design Architecture Documentation - The FPGA design process should include documenting details of the FPGA design implementation (how the FPGA does what it does). Documenting these details should be done in parallel with the creation of each functional block in the FPGA design. The level of details should be such that another engineer could recreate the part and simulations. FPGA design architecture details may include:

a. A list of files/blocks needed to create the FPGA
b. The same FPGA block diagram shown in the FPGA specification broken down in more detail showing lower-level blocks
c. Lower-level blocks broken down into sub blocks if they exist in detail
d. Finite State Machine (FSM) bubble diagrams
e. Flowcharts with their descriptions
f. Detailed timing diagrams
g. Synthesis guidelines
h. Place and Route guidelines

4. FPGA Part Selection – The part to be used for the flight implementation should be selected early in the FPGA requirements definition phase so that requirements can be defined that are consistent with the FPGA's capabilities. Part selection usually requires approval from a parts control board. The part selection process should include consideration of the following factors:

a. Functional and performance capabilities such as
    1) Combinatorial logic
    2) Sequential logic
    3) I/O quantity
    4) I/O types
    5) RAM and/or other embedded cores
    6) Achievable clock frequencies
b. Package style (Quad Flat Pack (QFP), Pin Grid Array (PGA), Column Grid Array (CGA), etc)
c. Package size (pin and gate count) – based on estimate of utilization:
    1) Find a similar design and determine gate count for target technology
    2) Overestimate if a guess is necessary
    3) Estimate internal RAM needed (due to possible TMRing needs)
d. Quantity needed
e. Power consumption
f. Reliability / Flight qualification status / Heritage
g. Radiation performance (Total Dose and Single Event Effects)Secondary voltages required
h. Programming technology (RAM based, Flash based, anti-fuse)
i. Flight parts
    1) Cost
    2) Availability
j. Non-flight test parts
    1) Similarity
    2) Cost
    3) Availability

5. Design Guidelines – The FPGA Designer (FPGAD) should gather all documents that establish design guidelines and requirements. This should include:

a. Official Documents – Refer to section P4, most notably 500-PG-8700.2.7.
b. Project Documents – Specific guidelines set by the individual projects to address specific issues such as any package capacitor decoupling layout, part programming procedures, CM procedures, etc.
c. Branch/Division/Project Standards such as:
    1) FPGA Style Guide/Coding Standards Usage – The FPGAD should use a style guide that allows for ease of testability, readability, and simulation. This can be accomplished by the following:

a) Using a coding convention that allows the function of signals to be recognized by their name. Project-defined coding standards are preferred, but self-made conventions are acceptable if allowed by the project. Document the standards, either in the header of the code itself, or in an existing document.
b) Using comments to describe all code.

d. Manufacturer's Application Notes and Frequently Asked Questions – See manufacturer's website
e. Useful websites:
   1) NASA Engineering Network (NEN)- https://nen.nasa.gov/web/pld
   2) NASA Engineering and Safety Center's (NESC) - http://www.nasa.gov/offices/nesc/home/index.html

6. Verification Plan - The FPGA design process should document the verification plan used to ensure proper functional operation of the hardware, showing how each requirement and feature of the board/FPGA is verified. Automation should be used as much as possible. For this document, only FPGA specifics are listed. This document should describe the following in detail:

a. Requirements verification matrix
   1) Show traceability for each unique requirement ID
b. Simulation
   1) List of test names and full description (tests should do positive and negative testing)
   2) Diagram showing test simulation configuration
   3) Specify whether code/functional coverage reporting is used
c. Non-flight testing (if applicable, not all projects may have)
   1) Flight Software use (typically tests only normal modes, positive testing)
   2) Special test code (plan early for in-situ debugging using special software)
   3) List of test names and full description (may be the same as used in simulation), positive and negative testing
   4) Diagram showing test configuration with Ground System Equipment (GSE)
   5) Requirements verification matrix
   6) Features verification matrix
   7) Temperature testing at box level using thermal chamber
d. Flight testing
   1) List of test names and full description (should be the same as used on ETU), positive and negative testing
   2) Flight Software use
   3) Diagram showing test configuration with GSE

7. Test Procedure – (aka: Acceptance Test Procedure (ATP)) A document should be written delineating the test steps required to verify full compliance of the FPGA implementation to the requirements document. It should be written clearly, so it can be handed to an independent test conductor. This procedure should be used on all hardware regardless of board stage (breadboard, ETU, Flight) as it

also serves as a check for proper manufacturing. For this document, only FPGA specifics are listed. The FPGA specifics of the test procedure should include the following topics:

a. Verification
    1) FPGA Identification
        a) FPGA name
        b) FPGA revision number
        c) FPGA type (socket or flat pack)
    2) Component verification
        a) Placement
        b) Orientation
b. Aliveness tests – verify power, clocks, and reset to/from FPGA
c. Power supply ripple measurements while simultaneously testing FPGA functions
d. Functional and Timing Tests
    1) Detailed instructions on how to test each function in the FPGA.
    2) How to test the mitigation or error correction techniques.
    3) Links the tests to each item in the specification (which follows requirements).
    4) Positive and Negative tests. Make sure it works how it is intended, and reacts safely to unintended inputs.
e. ATP closure
    1) Archiving the as-run ATP and log files

8. Implementation of File Version Control - This should be used for all documents identified in section P.8 and all FPGA design data files included in the design project directory. This is pre-CM version control and should begin with the first draft of a document through submitting the document to Project CM. RCS/CVS (tortoise CVS/others)/SVN are example Version Control applications.

a. For ease of tracking changes
b. For managing working files in a team environment
c. For the ability to revert to previous versions if needed
d. For backup purposes (on another machine)

Follow Project guidelines for entry into CM (i.e. CM at beginning of ETU build).

9. Implementation of FPGA Bug Tracking – This should be used to log and track defects/errors found during independent verification and hardware testing. Bug tracking provides the ability to keep any bugs from being overlooked. All bugs found should be addressed before any FPGA final review and flight burn. Bug status reports should be reported at the Project Critical Design Review (CDR) and used as a metric for design progress and maturity. Software such as JIRA and Mantis may be used to accomplish efficient bug tracking. Please note that the bug tracking is for pre-burn flight designs. Once the FPGA has been approved for flight use and flight parts have been programmed, the use of the project's PR/PFR system should be used to keep track of flight implemented issues.

10. Application Code Development - VHDL is the hardware design language generally preferred at Goddard, though other HDL's, such as Verilog or SystemC, may be acceptable. At the Project/Branch level, an agreement should be reached as to the language to be used for consistency across all the designs.

a. Follow Guidelines, checklists, Style Guides, and Coding Standards.
b. Document your code properly. Inline documentation will help later, and will help the next engineer if a personnel change is made in the middle of the Project.
c. Write the purpose of each procedure or function in the header.
d. Use comments to explain code that is not intuitively obvious to the reader.
e. For any Intellectual Property (IP) core to be used in the design, coordinate with the Office of Chief Counsel/Patent Counsel for review and advice regarding IP license and usage terms and related matters, which could impact current and future use of said IP core.

11. Test Code Development – Independent verification is strongly recommended. The following guidelines should be observed:

a. Follow the test sequence identified in the test plan.
b. Use Self-Checking/documenting test-benches.
c. Analyze code/functional coverage of simulation and test vectors.
d. Automate tests using scripts for repeatability and unattended runs.

12. Simulation of Functional Code – All functional code should be simulated.

a. The Board Lead (BL) should review tests and verify they do what the test plan calls out.
b. Run tests
c. Review waveforms and verify correctness.
d. Capture I/O to other chips/systems
    1) Verify results at this point
    2) Share with interfacing design engineers.
e. Disposition all warnings and errors reported by simulator.
    1) Understand why they are there
    2) Document any decision to ignore them

13. Synthesize the design

a. Set timing constraints in synthesis using constraint files.
    1) Define clocks
    2) Instantiate buffers
    3) Set critical paths if pushing part speed in any particular path
b. Review output files and logs for synthesis.
    1) Modify RTL code as needed to mitigate warnings

2) Justify all warnings not eliminated during formal FPGA design review

14. Vendor-Specific Place & Route Tool

a. Set timing constraints. Document and archive constraints files for reproducibility and review.
b. Define false paths / multi-clock paths, as appropriate.
c. Set part attributes.
    1) Package
    2) Temp range (MIL range suggested to ensure sufficient timing margin)
    3) Voltages (Core, I/O)
    4) Radiation level
d. Define and fix pin locations/attributes.
e. Run Place and Route.
f. Export Min-Typ-Max timing files for back-annotated simulations and datasheet timing for worst-case conditions analysis
    1) Min/Max delays will be contained within this file, ranging from the best case to the worst case.

15. Post-Route Verification

a. Review EACH entry of all logs from vendor tools for errors, warnings, and notes.
b. Modify RTL code as needed to eliminate errors, warnings, and notes
c. Review timing report to verify that the longest routes make sense.
d. Perform Timing Analysis
    1) Use the vendor's Static Timing Analysis (STA) Tool
    2) Include delays to/from pads on board
    3) Consider clock source and delays
    4) Include loading on outputs
    5) Get min/max data for any device interfacing with FPGA
    6) Enter all constraints into the STA tool
        a) All unconstrained pins should be listed and justified
    7) Set clock constraints at a percentage higher than actual (should be dictated by the project) to ensure required timing margin
e. Perform Back-Annotated Simulations
    1) Re-run simulations that were run on RTL using acquired SDF files
    2) Run at least these two conditions:
        a) Best-case simulation: Max Voltage, Min Temp, Zero Radiation, Highest Speed.
        b) Worst-case simulation: Min Voltage, Max Temp, with Radiation.
    3) Read every warning and error the tools generate.
    4) Justify all warnings or errors being ignored
    5) Verify that timing and functionality are both met.

16. Peer Reviews - The goal for the peer review is for the FPGAD to demonstrate to the review panel that the design meets all its requirements, has been designed per guidelines, and all analyses and simulations have been performed to verify it will work in the intended application, over the temperature range and for the life of the mission.

16.1 The PDL shall be responsible for ensuring that each FPGA design under his/her purview is peer reviewed. The review panel should include at least:

a. One FPGA designer from outside the project, who will serve as the chairperson for the review team, preferably with experience using the same part type.
b. One FPGA designer from the project, preferably one who designs a chip interfacing with the one being reviewed.
c. Other reviewers as needed, as described below.

16.2 Due to the nature of FPGA devices, the review process should involve other engineers to verify relevant aspects of the design. This group typically includes:

a. All owners of requirements that are flowed down to review the FPGA requirements.
b. The board-level designer and box lead to review all interfaces.
c. Software engineers to review functional interfaces and test requirements.

16.3 The peer review of an FPGA design should be conducted in stages. The first stage should be held early in the project timeline (i.e. pre – Preliminary Design Review (PDR)). The recommended stages are as follows:

a. Initial Review – to verify requirements and interfaces are understood and to assess the design implementation process
b. Mid-term Review – to verify coding styles and formats meet desired coding guidelines of the project
c. Final Review – to verify requirements and interfaces are understood and have been implemented and tested

Additional or delta reviews are up to the discretion of the Project or as a result of the outcomes of the main reviews.

16.4 The following list provides a guideline for the topics that should be addressed as part of the peer review process:

a. Requirements Review
b. Design Overview – Include context drawings, schematics
c. Interface Descriptions. Discuss timing/ functionality of external interfaces
d. Code Structure – include block diagrams to show functional block connections
e. Code Walkthrough – Discuss topics such as:
    1) Reset handling
    2) How illegal states in each FSM are handled

3) Use of global vs. routed clock signals
4) Clock domain crossings

f. Use of Intellectual Property (IP) cores (if applicable) discussion:
1) Verify that designer has reviewed all IP core data sheet specifications, application notes, revision notes, test benches, addendums or change notices for core modifications, updates, license agreement, and any vendor documentation describing the extent of vendor verification of their IP cores
2) Discuss the need for any recovery modes or strategies in the event of a failure or hang-up
3) If the design is being re-used because of heritage, verify that the IP has been demonstrated to meet all requirements in the new intended application
4) Verify that the verification (simulation and hardware test) strategy is robust enough to maximize coverage of functional or operational scenarios and configuration (e.g. test as you fly, fly as you test)
5) Verify project requirements on radiation have been implemented in the IP core design – ie that the IP core can be synthesized to incorporate any necessary radiation-mitigation strategies such as Triple Modular Redundancy (TMR) or, for the case of a hard core, comes implemented with appropriate radiation-mitigation strategies

g. Implementation discussion:
1) Pinouts
2) I/O Selection
3) External clocks (draw clock tree for each oscillator)
4) Clocking(rates, routing resources, distribution)
5) Reset (source, location, duration)
6) Combinatorial and sequential modules utilization percentages

h. Test Plan discussion:
1) Walk through test procedure document and test sequence flowchart
2) Review verification matrix
3) Discuss hardware test configurations

i. Present results:
1) Simulation results
a) If code/functional coverage was on during simulations, then any missed code, either by directive or a miss in simulation, should be documented and explained
2) Synthesis results
3) Place & Route results
4) Timing Analysis. Show how margins are met
5) Interface Analysis (drive strengths, I/O levels, power supply levels, sampling of input signals, no bus left floating)
6) Board Implementation (power supply decoupling, signal integrity analysis, routing)

j. Hand off the entire design directory package to the peer review team preferably 2 weeks before the scheduled review:
1) Code
2) Test Code

3) Documents – board-level review charts
4) List of design tools and version numbers
5) Constraint files
6) Vendor tool output files
7) Manufacturers datasheets
8) Completed design checklist (i.e. 500-PG-8700.2.7, Appendix G or equivalent)
9) Anything else needed to understand and test the design

16.5 The final peer review should be conducted with each reviewer present as well as all design team members, PDL and QA. At this review, the review chairperson will communicate the following:

a. The results of the previously held reviews
b. Any RFA's generated during the reviews
c. Proposed plan for the resolution of open RFA's

16.6 The peer review process is considered complete once all open issues are closed. The chairperson will provide the PDL with:

a. A formal report indicating that the design has been reviewed and includes a summary of the findings
b. A signed copy of the FPGA Review Checklist Form provided in Appendix C or equivalent

16.7 If for any reason the FPGA is changed after the review process is closed, the review team should be notified and allowed to review said changes.

17. Presentation of Peer Review Results at formal Project Reviews

17.1 While each individual FPGA design will not be covered at project-level formal reviews, the PDL will present the results of the peer review process to Project so that questions can be answered regarding:

a. Demonstration of margins and how they are calculated
b. Results of Peer Reviews / issue resolution
c. Any outstanding RFA's

Appendix A – Definitions

A.1 **Product Design Lead (PDL)** – A Product Development Lead is the individual responsible for developing and delivering a mission subsystem that meets technical requirements given cost and schedule restraints (constraints). The PDL is responsible for managing the design activity, managing the technical and organizational interfaces identified during the design planning, and where required, forming and leading the Product Design Team (PDT).

A.2 **Board Lead (BL)** – A Board Lead is a PDT member and is responsible for the board design, implementing all board requirements and identified technical interfaces. The BL is also responsible for leading the Board Design Team (BDT) if applicable.

A.3 **FPGA Designer (FPGAD)** – An FPGA Designer is a BDT member responsible for the FPGA design and implementing all FPGA requirements.

A.4 **Verification Engineer (VE)** – A Verification Engineer is a BDT member responsible for requirements and functional verification of the FPGA and board working very closely with the BL and FPGAD. Ideally, the VE is not involved in the FPGA design at all, thus qualifying as an independent verifier of the FPGA.

A.5 **Record Custodian** – An individual who is responsible for collecting, indexing, accessing, filing, storing, maintaining, and dispositioning a record or collection of records. See GPR 1440.8, Records Management.

Appendix B – Acronyms

Abbreviation/ Acronym	DEFINITION
ATP	Acceptance Test Procedure
CDR	Critical Design Review
CGA	Column Grid Array
CM	Configuration Management
ETU	Engineering Test Unit
FDC	Fault Detection & Correction
FlatSat	Flat Satellite
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
FSW	Flight Software
GSE	Ground System Equipment
GSFC	Goddard Space Flight Center
HDL	Hardware Description Language
I/O	Input/Output
IP	Intellectual Property
JTAG	Joint Test Action Group
JWST	James Webb Space Telescope
MIL	Military
NEN	NASA Engineering Network
NESC	NASA Engineering and Safety Center's
PCB	Printed Circuit Board
PDR	Preliminary Design review
PGA	Pin Grid Array
PR/PFR	Problem Reporting/Problem Failure Reporting
QFP	Quad Flat Pack
RAM	Random Access Memory
RFA	Request For Action
RTL	Resistor Transistor Logic
SDF	Standard Delay Format
SpW	SpaceWire
STA	Static Timing Analysis
TMR	Triple Modular Redundancy
VHDL	VHSIC (Very High Speed Integrated Circuit) Hardware Description Language
VLSI	Very Large Scale Integration
VN	Version Number
WOA	Work Order Authorization

Appendix C - FPGA Design Peer Review Checklists

- ☐ Requirements are met
- ☐ Simulations successfully completed for best and worst case conditions
  - ☐ Log files submitted
- ☐ Functional Block Diagrams
  - ☐ Timing diagrams
- ☐ State Machines
  - ☐ Compared architecture document and synthesizer outputs
- ☐ I/O Interfaces
  - ☐ Block diagrams
  - ☐ Timing diagrams
- ☐ Clocks
  - ☐ Frequency/phase requirements
  - ☐ Assignments
  - ☐ Tree
  - ☐ Timing
  - ☐ Clock domains crossings are handled properly
- ☐ Resets
  - ☐ Block diagram
  - ☐ Table or tree
  - ☐ Synchronization methods
- ☐ Design Tool Reports and Analysis
  - ☐ Synthesis
  - ☐ Pin
  - ☐ Utilization
  - ☐ Place & Route
  - ☐ Code coverage report
  - ☐ Clock domain crossings
  - ☐ Timing
  - ☐ Worst case analysis
- ☐ Key documents have been CM (Configuration Management) released prior to the FPGA review
- ☐ Design Checklist for Designers completed and reviewed (Appendix G of 500-PG-8700.2.7 or equivalent)
- ☐ All review issues/concerns have been documented

FPGA Review Chair _____

Print Signature Date

APPENDIX D - Flight FPGA Project Timeline

	Mission Level Reviews	SRR	SDR	PDR	CDR	TRR	FRR	Launch
Development Steps	1	Derive box level requirements						
	2	Derive board level requirements						
	3	**Derive FPGA requirements**						
	4	Digital Electronics Assurance Plan Review						
	5	**Create FPGA Requirements Verification Matrix**						
	6	**Create FPGA Test Plan**						
	7	**FPGA Requirements Review**						
	8	**Create FPGA Specification Document**						
	9	**Create HDL Verification Plan**						
	10	**FPGA PDR**						
	11	Box PDR						
	12	**HDL Code Development**						
	13	**HDL Testbench Development**						
	14	**HDL Code Peer Review(s)**						
	15	**Create Board Test Procedures**						
	16	**FPGA ETU Test**						
	17	**FPGA CDR**						
	18	Box CDR						
	19	**FPGA Pre-Flight Burn Review**						
	20	**Flight FPGA Test**						
	21	Box Environmental Test						
	22	System Environmental Test						

CHANGE HISTORY LOG

Revision	Effective Date	Description of Changes
Baseline	11/13/06	Initial Release
A	6/3/13	Updated P1, P2, P3, P5, P8. Removed P10. Updated Procedure section to reflect what we now do in more detail. Updated and added appendixes.
	07/11/13	This document was administratively changed to include a comment from Legal (section 10.e).
	08/07/2013	This document has been administratively changed at the template block from Procedures and Guidelines to Center-Wide Procedures and Guidelines.
	05/14/2018	Administratively extended for 1 year.
	05/29/2019	Administratively extended for 6 months.
	11/20/19	Administratively extended for 3 months.
	01/21/20	Administratively extended for 1 year.