Procedures and Guidelines

DIRECTIVE NO.	500-PG-8700.2.8	**APPROVED BY Signature:**	***Original Signed by***
EFFECTIVE DATE:	November 13, 2006	**NAME:**	Madeline Butler
EXPIRATION DATE:	November 13, 2011	**TITLE:**	Deputy Chief Engineer

COMPLIANCE IS MANDATORY

Responsible Office: 500 / Applied Engineering and Technology Directorate

Title: Field Programmable Gate Array (FPGA) Development Methodology

PREFACE

P.1 PURPOSE

As FPGA technology advances, there is a growing trend for incorporating into these devices an increasing amount of the circuitry traditionally implemented as discrete digital components with degrees of complexity ranging from small to Very Large Scale Integration (VLSI). In some cases, the equivalent of an extremely large number of Printed Circuit Boards (PCB) based on such technology can be condensed into a single FPGA device. There is a need for FPGA development guidelines that ensure that adequate and consistent rules are applied to all FPGA designs, equivalent to those followed for traditional PCB implementations.

This document establishes a methodology for the development of FPGA designs and is intended for use by engineering leads responsible for designs that include the use of FPGA devices. 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 FPGA designs and is intended primarily for use by FPGA designers. The two documents are intended to compliment each other.

P.2 APPLICABILITY

This procedure applies to the development of all Applied Engineering and Technology Directorate flight products that include the use of FPGAs. This includes those developments conducted in-house, as well as those performed by second-party developers providing products to the AETD in support of GSFC Projects.

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

P.5 CANCELLATION

NONE

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 WOA's and Verification Test Results as the only official records that must be retained. Those documents were mainly developed to address the needs of traditional electrical designs involving PCB's and their related enclosures, and did 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 must be generated for each FPGA design. For that reason, some of the documents listed here are defined as Design Inputs (requirements documents) and Design Outputs (drawings, schematics, and specifications) in other documents. It was felt that, since this document specifically addresses the development of FPGA's, it was important to make it as comprehensive as possible by including a complete list.

Record Title	Record Custodian	Retention
FPGA Design Requirements Document	Product Design Lead (PDL)	* See note below
FPGA Design Specification	Product Design Lead (PDL)	* See note below
FPGA Schematic Drawings/ VHDL Code Listing/ Other Design Tool Products/ All Electronic Design Files	Product Design Lead (PDL)	* See note below

FPGA Analysis and Simulation Results	Product Design Lead (PDL)	* See note below
FPGA Peer Design Review Documentation, Requests for Action (RFA's) Generated and Disposition	Product Design Lead (PDL)	* See note below
FPGA Test Results / Includes Oscilloscope Pictures to Document Proper Signal Integrity and Timing	Product Design Lead (PDL)	* See note below

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

P.9 METRICS

NONE

P.10 DEFINITIONS

a. <u>Product Design Lead (PDL)</u> – For the purposes of this document, PDL is synonymous with lead engineer. The lead engineer 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).

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."

It will be the responsibility of the Product Design Lead (PDL) to ensure adhesion to this procedure during the design and development of each FPGA. It will also be his/her responsibility to ensure that these guidelines are implemented by second-party developers providing products to the AETD.

This document is intended to assist the PDL in successfully developing spaceworthy FPGA designs. It delineates a methodology to maximize the chances of developing components that meet design requirements, have performance that can be demonstrated during ground testing and will perform with robustness under space environments. Another goal of this document is to prevent most of the common pitfalls encountered in today's FPGA designs. The document is written in the rough chronological order an FPGA designer should follow to develop parts for a space flight program.

Appendix A consists of a check list to be used as part of the review process for a new FPGA design. It aims at providing consistency so the same measuring stick can be applied to multiple designs without the worry of items "falling through the crack" and being missed. This check list shall be used as a check for completeness.

1. FPGA Design Requirements Document – The designer shall generate a written document that compiles all the required characteristics for the FPGA design. References to higher level documents can be used. This document will provide traceability for the implementation, and establish the guidelines for the test and verification steps. The FPGA requirements are derived from the board-level requirements, and will typically include:

   - Functions to be implemented
   - Performance (speed, critical timing, throughput)
   - Interface description (signal levels, timing, software, data formats)
   - Environmental constraints (thermal, radiation level at part, mission duration)
   - Testability requirements (JTAG, board scan, software, observable internal points).

2. Select FPGA Part – The part to be used for the flight implementation shall be selected as early in the development cycle as feasible to allow for the long procurement cycles normally associated with flight FPGA devices. The selection must take into account the availability of equivalent commercial grade devices that may be desirable for cost reasons in the development of breadboards and test beds. The part selection process will include consideration of the following factors:

   - Package style
   - Reliability / Flight qualification status / Heritage
   - Radiation specs (Total Dose and Single Event Effects)
   - Estimate of utilization:
     - Use prior experience
     - Find similar design and get gate count for target technology
     - Overestimate if a guess is necessary
     - Quantity needed
   - Speed rating

3. Design Guidelines – The designer should gather all documents that establish design guidelines and requirements. These will normally include:

   - Official Documents – Refer to section P4, most notably 500-PG-8700.2.7.
   - Project Documents – Specific guidelines set by the individual projects to address specific issues such as the ACTEL RT54SX32 208-Pin Package capacitor decoupling layout, part programming procedures, Configuration Management Procedures, etc.
   - Branch/Division/Project Standards – Ask branch if they exist based on lessons learned from other projects

- Common Knowledge do's and don'ts. – Do a web search of the Office of Logic Design website at www.klabs.org
- Manufacturer's Application Notes and Frequently Asked Questions – See manufacturer's website

4. FPGA Design Specification - The designer shall generate a document detailing the FPGA design information. The specification document contains enough detailed guidance to allow a junior engineer to re-create the part and simulations. This document contains:

    - A block diagram of the FPGA architecture
    - Specific requirements for scrubbing, triple module redundancy, etc.
    - Specific requirements for the ability to inject errors in order to test mitigation or error correction techniques
    - Constraints on board-level implementation (critical pins for board routing, proximity to other devices, input slew-rate limitations, etc.)
    - A description of all interfaces including pin-out assignment
    - A functional description of the device
    - 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.
    - Timing information
    - Place and Route Guidelines
    - A list of all files needed to create the FPGA
    - A copy of the data sheet and relevant application notes used to implement the design

5. Test Procedure - A document shall be written delineating the test steps required to verify full compliance of the FPGA implementation to the requirements. It must be written clearly, so it can be handed to an independent tester. The test procedure will include the following topics:

    - Functional and Timing Tests
        - Detailed instructions on how to test each function in the FPGA.
        - How to test the mitigation or error correction techniques.
        - Link the tests to each item in the specification (which follows requirements).
        - Positive and Negative tests. Make sure it works how it is intended, and reacts safely to unintended inputs.
        - Number the tests in the document. These numbers are referred to in the testbench code.

    - Test Plan
        - Simulation environment testing
        - Breadboard testing
            - Flight Software. Typically tests only normal modes, positive testing
            - Special Test Code. Plan early for in-situ debugging using special software
        - ETU testing
            - Temperature testing

        - Verification Suite and Flight Software
    - Flight Unit testing
        - Plan for observability of functions while in a chamber

6. Use Style Guide/Coding Standards

- Follow some conventions to allow you to recognize the function of signals by their name. Project defined coding standards are preferred, but self-made conventions are acceptable if allowed by the project. (as an example, refer to JWST Coding Style Guidelines).
- Document the standards, either in the header of the code itself, or refer to an existing document.
- Use modular design to ease testability, readability, and simulation.

7. Implement Version Control - Shall be used for all documents identified in section P.8

- For ease of tracking changes.
- Can back up after 'oops' changes.
- For backup purposes (on another machine).
- RCS / CVS (tortoiseCVS/others).
- Follow project guidelines for entry into CM (i.e. CM at beginning of ETU build).

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

- Follow Guidelines, checklists, Style Guides, and Coding Standards.
- Document your code properly. Inline documentation will help you later, and will help the next engineer if a personnel change is made in the middle of the Project.
- Write the purpose of each procedure or function.
- If you are using any tricks to achieve the design, use inline comments to explain why and how.
- Consider the electrical implications of your code. An FPGA design is a hardware implementation, not software. Some of the points below require action at the board level, outside the part. Communicate issues with the board designer:
    - Reset Practices
        - Typically asynchronously applied and synchronously removed
    - Timing Practices
        - Synchronous design
        - Asynchronous inputs
        - Signals which cross different clock boundaries
    - Logic Practices
    - Error Handling

- Design for return to safe state if the unexpected occurs in inputs
- Consider Error-handling in every circuit … "what happens if…" and design the circuit to get to a safe state and continue.

- Power Related
  - Proper power supply decoupling.
  - Power supply sequencing.
  - Distribute simultaneously switching output pins around periphery to avoid overloading supplies and causing ground-bounce.
- Interfacing
  - Verify correct I/O levels are being used. Choose best I/O drivers.
  - Use the slowest edge rates possible given the design constraints.
  - Handle power-up/power-down where I/O may not be valid, to prevent an invalid state.
  - Don't allow bus contention.
  - Don't allow tri-state buses to float in the center region.
  - Input slew rate specification must be met.
  - Perform signal integrity analysis of all the interfaces to determine the need for external impedance matching termination.
  - De-bounce and de-glitch interfaces from mechanical devices. Use minimum bandwidth necessary to observe the signal.
- Testability
  - Plan your design with testing in mind and incorporate the resources needed to facilitate it. Consider observability as you implement your design. Think about how you will debug the circuit while the part is on the (BB/ETU/Flight) board.
  - Reserve test pins as test-only pins. Buffer the signals provided to the test pins from the internal circuitry.

9. <u>Develop Test Code</u> – Ideally an independent FPGA tester is preferred, but is not mandatory. The following guidelines should be observed:

- Follow the test sequence identified in the test procedure. Refer to the assigned test number for each test.
- Use Self-Checking/documenting test-benches.
- Analyze code coverage of simulation and test vectors.
- Automate tests using scripts for repeatability and unattended runs.

10. <u>Simulate Functional code using test-benches</u>

- Review tests.
- Review waveforms for sanity check.
- Capture I/O to other chips/systems
  - Share with interfacing design engineers.

  - Take the time to discuss results at this point, it can save lots of hassles later.
- Chase down all warnings and errors reported by simulator.
  - Understand why they are there.
  - Document any decision to ignore them.

11. <u>Synthesize the design</u>

- Use equivalent to flight part from the beginning.
- Set timing constraints in synthesis using constraint files.
  - Use loading for each pin by reviewing schematics and specs for each interfacing part.
  - Set critical paths if pushing part speed in any particular path.
  - Begins familiarity with critical paths.
  - Using constraint files assists with self-documenting design.
- Review output files and logs for synthesis.
  - Understand all warnings and if you decide to ignore any, document the reason why.
- Search through the netlist for issues.
  - For example, search for flip-flops with both asynchronous preset and clear. These should not be used, and point to interpretation issues in the code. Search for latches; may be unintended result of coding style.

12. <u>Transfer netlist to vendor-specific place & route tool</u>

- Set timing constraints. Document and archive constraints files for reproducibility and review.
- Double-check false paths / multi-clock paths.
- Set proper flight part
  - Package
  - Temp range (MIL range suggested to ensure sufficient timing margin)
  - Voltages (Core, I/O)
  - Radiation level
- Fix pin locations
- Run Place and Route.
- Export Min-Typ-Max Standard Delay Format (SDF) files for simulation
  - Min/Max delays will be contained within this file, ranging from the best case to the worst case.

13. <u>Post-Route Verification</u>

- Review all logs from vendor tools for errors, warnings, and notes.
- Review timing report to verify that the longest routes make sense.
- Timing Analysis
  - Use the vendor’s Static Timing Analysis (STA) Tool
  - Include delays to/from pads on board
  - Consider clock source and delays
  - Include loading on outputs
  - Get min/max data for any device interfacing with FPGA

  - Enter all constraints into the STA tool
  - Set clock constraints at 20% higher than actual to ensure required timing margin
- Back-Annotated Simulations
  - Re-run simulations that were run on RTL
  - Run at least these two conditions:
    - Best Case beginning of life (BOL) sim: (Max Voltage, Min Temp), Zero Radiation, Highest Speed.
    - Worst Case BOL sim: (Min Voltage, Max Temp), Zero Radiation.
  - Read every warning and error the tools generate. If you decide to ignore a warning, document the reason.
  - Verify that timing and functionality are both met.

14. Hold Peer Review - The peer review is the most important review of the design process. The goal for the peer review is for the FPGA design engineer to demonstrate to the review panel that the design meets all its requirements, has been designed properly, 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.

The PDL will be responsible for insuring that every single FPGA design under his purview is peer reviewed. He will plan each review in coordination with the Electrical Engineering Division (EED) person responsible for this function, or the EED Chief Engineer if applicable. Together they will select a peer review panel. The panel should include at least:

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

Due to the nature of FPGA devices, the review process will involve other engineers to verify relevant aspects of the design. This group will normally include:

- All owners of requirements that are flowed down must review the FPGA requirements.
- The board-level designer and box lead must review all interfaces.
- Software engineers must review the functional interfaces and test requirements.
- PWB designers must review requirements relevant to layout.
- Thermal engineers should be advised as to expected power dissipation.

Peer Review Format – The peer review of an FPGA design will normally be conducted in several stages. The following list provides a guideline for the topics that should be addresses as part of the peer review process, as well as a recommendation for how the process can be implemented:

Initial Meeting

- Requirements Review
- Design Overview – Include context drawings or schematics
- Interface Descriptions. Discuss timing/ functionality of external interfaces
- Code Structure – include block diagrams
- Code Walkthrough – Discuss:
  - Reset handling
  - How illegal states in each FSM are handled
  - Use of global vs. routed clock signals
  - Clock boundary signal resynchronization
- Implementation discussion:
  - Pinouts
  - I/O Selection
  - External clocks (draw clock tree for each oscillator)
  - Clocking(rates, routing resources, distribution)
  - Reset (source, location, duration)
  - Combinatorial and sequential modules utilization percentages
- Test Plan – Walk through test procedure document and test sequence flowchart.
- Present results:
  - Simulation results
  - Timing Analysis. Show how margins are met (20% margin)
  - Interface Analysis (drive strengths, I/O levels, power supply levels, sampling of input signals, no bus left floating)
  - Board Implementation (power supply decoupling, signal integrity analysis, routing)
- Hand off CD with design package to the peer review team:
  - Code
  - Test Code
  - Documents – board-level review charts
  - List of design tools and version numbers
  - Constraint files
  - Vendor tool output files
  - Manufacturers datasheets
  - Anything else needed to understand and test the design

Independent Analysis

Individual reviewers independently review design aspects assigned to them by the chairman of the peer review team. This step will accomplish:

- Review of the schematics/code
- Review board implementation, including results of signal integrity analysis
- Verify critical interfaces and implementation details
- As needed, run simulations of critical sections of the design

- Develop questions and comments (informal RFA's) and communicate them to the other review team members for their consideration. The communication at this point can be via email or alternate agreed upon method.
- Each reviewer submits to the chairman his assessment of the review using the FPGA Review Checklist Form provided in Appendix A
- The chairperson ensures that all reviewers are satisfied that the flight implementation will work

Final Peer Review Meeting

At this meeting, held between the design team and the peer review panel members, the review chairperson will communicate the following:

- A summary of the issues that arose during review process and their resolutions
- The results of the peer review
- Any Requests for Action (RFA's) generated during the review
- Proposed plan for the resolution of open RFA's

End of Peer Review

Once all open issues are resolved, the chairperson will provide the PDL with:

- A memorandum indicating that the design has been successfully reviewed and is acceptable for flight
- A signed copy of the FPGA Review Checklist Form provided in Appendix A

15. Presentation of Peer Review Results at formal Project Reviews

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 so that questions can be answered regarding:

- Demonstrate margins and how they are calculated
- Results of Peer Review / issue resolution
- Any outstanding RFA's
- Peer Review Checklist certifying successful completion

APPENDIX A

FPGA Design Peer Review Checklist

Reviewer:
Date:

Project:
Subsystem:
Board:
FPGA Designation:

- ☐ Requirements are met
- ☐ Simulations successfully completed for best and worst case conditions:
  - ☐ Best Case (Lowest Temperature, Highest Operating Voltage, Zero Radiation, Best Process)
  - ☐ Worst Case (Highest Temperature, Lowest Operating Voltage, Maximum Radiation, Slowest Process)
- ☐ Simulation adequately tests design (tests all sections of code and circuitry)
- ☐ Timing analysis completed successfully
- ☐ Resets handled properly
- ☐ Clocking handled properly
- ☐ All clock-domain crossings are handled properly
- ☐ Asynchronous inputs are filtered for meta-stability issues
- ☐ I/Os properly selected (SSO, levels, slew rates, etc.)
- ☐ Relevant manufacturer recommendations and app notes followed
- ☐ Asynchronous circuits clearly identified and analyzed for robust operation
- ☐ Board-level issues addressed (decoupling, routing, signal integrity, etc.)
- ☐ Margins have been demonstrated and are acceptable (timing, utilization)
- ☐ Reviewer issues satisfied

CHANGE HISTORY LOG

Revision	Effective Date	Description of Changes
Baseline	11/13/06	Initial Release