

# FPGA Design Process Checklist

Martin Fraeman

Pete Eisenreich

JHU/APL

Laurel, MD

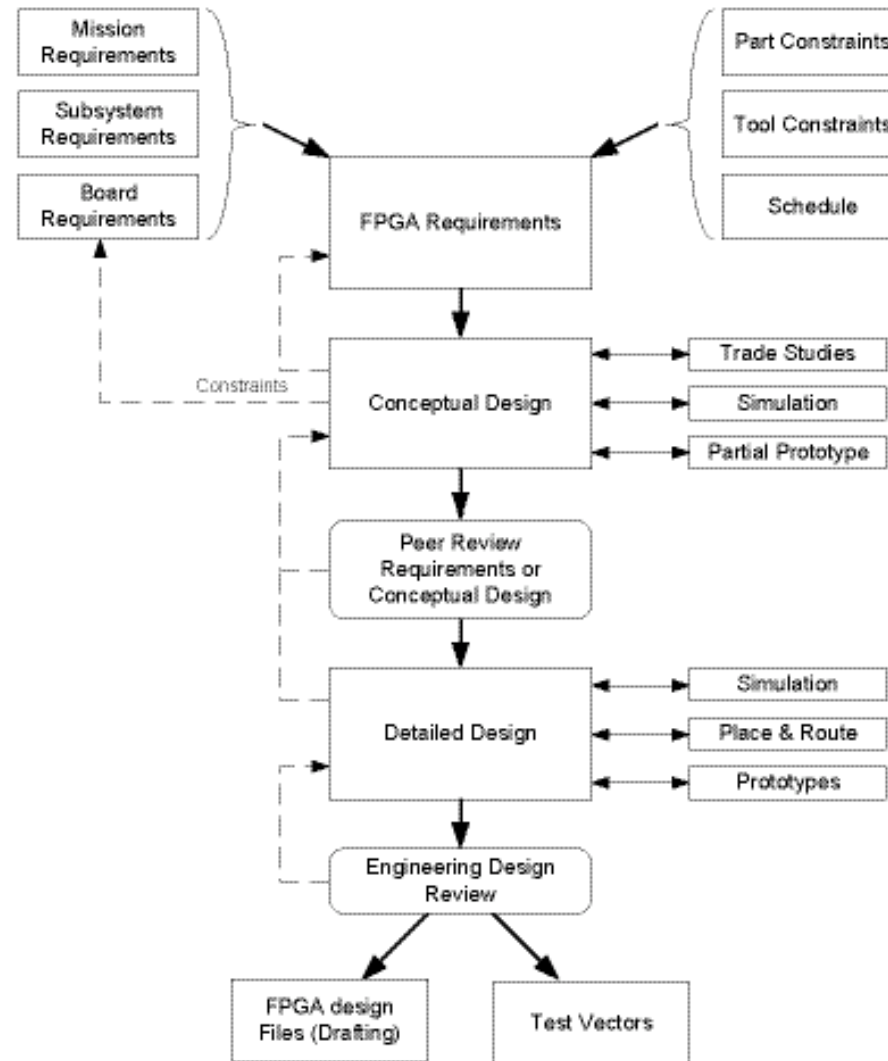
# Checklist Motivation

- Develop a process to consistently design FPGAs for space applications
- Useful to wide range of staff
  - Guide/reminder for experienced FPGA developers
  - Transition aid for designers new to FPGA
  - Training aid for inexperienced staff
- Evolved from experience developing several spacecraft
- First upfront applications now in progress
- May also be applied to ASIC design

# Checklist Goals

- Minimum Requirements for Hi-Rel apps
  - Minimally burdensome to small or quick programs
  - Program may require more
  - Team may require more
  - You have the freedom to do more if necessary
- Technology Independent
  - Design rules for Actel 1020 different than 54SX-S or Virtex II
  - Peer organizations Design Processes from mid 90's, ISO9000
    - Based on Actel 1020, not updated in 8 years
    - Contain requirements that do not universally apply
  - Either schematic or HDL
  - Use technology dependent supplements when needed

# Development Flow



# Requirement Requirements

- Requirements must be defined, vetted, and documented
  - Memo to file, requirements review, or conceptual design review presentation material
  - Intent is to write it down without adding additional development burden
- Keep up to date and distribute
- Firm “chip spec” is rare; FPGA spec developed concurrently so requirements are often a moving target

# Requirement Contents

- Functions
  - What not how
  - Identify critical functions and effect of failure
- Interface definitions
  - Signal levels
  - Timing
  - Circuit def'ns
  - Software
  - Data formats
  - Impact of failures
- Environment constraints
  - Radiation
    - Mitigation strategies for SEU, total dose
  - Thermal
  - Mission duration
- Testability provisions
  - Scan based test port
  - Board JTAG chain
- Constraints imposed by target technology/tools
  - Target part
    - SSO impact on pinout
    - Timing methodology
  - CAD tool flow

# Requirements Review

- Discuss FPGA req'ts with all concerned
  - Board, component, subsystem, system, and software engineers
  - Others with interfaces to the FPGA
- Informal review, not a power point contest; Paper and sketches are OK
- May be delayed until conceptual design completed or as part of board requirements review
- TBDs and Tentative Requirements are OK. State what's understood and what still needs to be figured out
- Understand why of req'ts to distinguish between firm, goal, desire, assumption
- Reduce to minimum
- Avoid over-performance, unneeded features
- Keep up to date as design works out details

# Conceptual Design

- Consider alternate design approaches to satisfy the requirements
- Outputs
  - Detailed block diagram
  - Functional block descriptions
- May be appropriate to develop requirements and conceptual design together



# Conceptual Design Factors FPGA Related

- Evaluate and select target technology
  - Capacity, speed
  - Environmental
  - Board design impact
  - Thoroughly understand app notes, alerts, warnings, especially for space applications
- How implementations fit target FPGA architecture
- IP availability, quality, suitability, prior designs, other reuse
- Functions needing special attention
  - SEU mitigation
  - SSO impact on pinout
  - High speed serial interfaces
- Choose CAE tools
  - Capability
  - Availability
  - Productivity
  - Teams expertise in tools and need for training

# Conceptual Design Factors

## Project Related

- KISS
  - Easier to get right
  - Easier to know its right
  - Easier to know its reliable
- Breadboard things
  - To clarify vendor docs
  - Check performance
  - Verify requirement is feasible
- Simulation/test approach to verify req'ts met
- Can satisfying requirements still meet schedule?
  - Then one or the other *MUST* change!
  - Plan for that now or watch it happen anyway

# Detailed Design

- Terminated after resolution of action items raised at Engineering Design Review (EDR)
- EDR must be completed before programming flight parts
- Frequently breadboard/engineering model will have been built and debugged
- EDR is late in development so significant problems exposed are tough to fix
- Requirement review should have caught those sorts of problems
- Consider informal reviews during breadboard development

# Detailed Design Issues For EDR

- Global net distribution
- Edge sensitive races caused by routing
- Decode noise, SEU impact on async inputs
- Target tech SEU rates
- SEU mitigation
  - Target tech SEU rates
  - Critical circuits (config reg, state reg) SEU rates and impact
  - Effect on combinational circuits like async reset
  - FSM invalid state response
- Other radiation effects
- I/O impact on internal/board routing
- I/O and SSO bounce
- Programmed I/O modes (threshold, level, slew, termination, ...)
- I/O state impact on I/F during power on/off
- Worst case simulation (process, temp, radiation)
- External failure impact through I/O on FPGA

# Detailed Design

## Good Design Practices

- Modular design to ease testability and simulation
- Use spare pins to increase observability during debug and test
- Build test board to verify over temp, volt
- Add test points to flight board to help debug and test
- Make sure board test verifies FPGA functions
- Verify margins on critical signals
- Avoid using probe/diagnostic pins for application signals
- Do a static timing analysis in both worst and best case corners
- Code coverage analysis of simulation and test vectors

# Engineering Design Review Goals

- Present updated requirements
  - Functional, program, and organization process
  - Use verification matrix to show how met
- Must be readable by the uninitiated
  - Must include clear block diagrams
  - HDL must show as synthesized logic for critical/interesting modules
- Review board
  - Chair independent (not working on same board)
  - At least one member familiar with board/box
- Publish: presentation, minutes, action items, responses

# Engineering Design Review

## Design Walkthrough

- Detailed Design EDR issues
- As synthesized gate level schematics for critical circuits with source code
- How FSM illegal states are handled
- Decode glitches/SEUs don't effect asynchronous or edge sensitive circuits
- Appropriate use of global clock signals
- Asynchronous set/reset inputs are glitch free
- Power on reset and how sync'd with clocks
- Resynchronization across clock boundaries
- SEU susceptibility or immunity where important
- Other circuits of interest, particularly for SEU review
- Worst/Best case simulations/timing analysis results concentrating on external interfaces

# Formal Documentation

- Organizational tracking number
  - Drawing number at APL
- Store/Archive all design files under drawing number
- “all design files” includes
  - Hierarchy of files sufficient to reconstruct design
  - Tool independent format such as text or pdf
  - List of design tools and version #
  - Tool dependent key files (fuse or memory map)
  - Any other information deemed important by the designer



# Other Documentation

- Design review material (especially EDR)
- Encouraged to also prepare
  - User guide
  - Documentation for module with potential for reuse

## Options to Consider

- Centralized flight parts programming
- Independently verify that flight version matches archived version
- Post-programming electrical test
  - Controversial
  - consensus of users heading towards “no”
  - Some vendors strongly against

## Future Revisions to FPGA Process

- Same review team follows the design thru several reviews?
- Suggestions?