



Center-Wide Procedures and Guidelines (PG)

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

APPROVED BY Signature: Original Signed By
NAME: Felicia Jones
TITLE: Director, AETD

COMPLIANCE IS MANDATORY

Responsible Office: 500/Applied Engineering and Technology Directorate

Title: Design of Space Flight Field Programmable Gate Arrays

TABLE OF CONTENTS

1	SPECIAL PINS	7
1.1	CONFIGURATION PINS	7
1.2	UNUSED INPUTS	7
1.3	TEST INTERFACE	8
1.4	DEBUG INTERFACE	8
2	INPUT/OUTPUT (I/O)	8
2.1	SIMULTANEOUS SWITCHING OUTPUTS (SSOs).....	8
2.2	SIGNAL TERMINATION.....	10
2.3	TRI-STATE BUS CONSIDERATIONS	10
2.4	INPUT TRANSITION TIMES	11
2.5	SHORTING OUTPUTS TOGETHER	11
2.6	MIXED I/O STANDARDS.....	12
2.7	POWER SWITCHING AND COLD SPARING	13
3	CLOCKS	15
3.1	CLOCK BUFFERS.....	15
3.2	CHIP-TO-CHIP TIMING STRATEGY	15
3.3	DELAY LOCK LOOPS, DLLS, AND PHASE LOCK LOOPS, PLLS	15
3.4	CLOCK TREE DIAGRAM	16
3.5	CROSSING CLOCK DOMAINS.....	17
3.6	FLIP-FLOP REPLICATION	17
3.7	OPPOSITE EDGE CLOCKING	17
3.8	METASTABILITY	17
3.9	LATCHES	17
4	FINITE STATE MACHINES, FSM	18
4.1	CRITICAL STATE MACHINES.....	18
4.2	STATE ENCODING	18
4.3	HDL SYNTHESIZED MACHINES	19
4.4	ERROR DETECTION AND CORRECTION, EDAC	20
5	RESET	21
5.1	RESET LOGIC CIRCUIT (CONSIDER TRANSIENT BEHAVIOR)	21
5.2	RESETS	22
5.3	RESET TREE.....	22
5.4	COMPONENT STARTUP TIME	23
5.5	MISSION CRITICAL SIGNALS.....	23
6	HAZARD ANALYSIS	24

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

6.1	STATIC HAZARD	24
6.2	DYNAMIC HAZARD.....	24
7	POWER.....	25
7.1	SUPPLY SEQUENCING	25
7.2	SIGNALS INTO UNPOWERED CMOS I/O'S.....	25
7.3	STARTUP VOLTAGE RISE TIME.....	26
7.4	BYPASSING AND DISTRIBUTION.....	26
8	INTERFACING TO NON-VOLATILE MEMORIES (EEPROM, FLASH, ETC)	27
8.1	PROTECTION DURING POWER-UP/DOWN TRANSITIONS	27
8.2	ANALYSIS OF DAMAGE DURING WRITE CYCLES	27
8.3	CYCLE COUNT.....	27
8.4	TRANSIENTS AND NOISE.....	28
8.5	RELIABILITY	28
8.6	REFRESHING AND RELOADING	29
8.7	RECOMMENDATIONS AND TIPS	29
9	TIMING ANALYSIS.....	31
9.1	CLOCKS	31
9.2	FLIP FLOPS	31
9.3	ENVIRONMENTAL EFFECTS.....	32
9.4	SPEED GRADE.....	32
9.5	ASYNCHRONOUS CIRCUITS.....	33
9.6	TIMING MARGIN	34
10	MISCELLANEOUS DESIGN GUIDELINES AND CRITERIA.....	35
10.1	NOISE IMMUNITY AND QUIET DESIGNS	35
10.2	DEFENSIVE DESIGN AND DESIGNING FOR OFF-NOMINAL EVENTS	35
10.3	DESIGNING FOR TESTABILITY	36
11	RECONFIGURABLE FPGA TECHNOLOGY	37
11.1	CONFIGURATION MEMORY.....	37
11.2	REDUNDANCY	37
11.3	EMBEDDED FUNCTIONS	38
11.4	IN-FLIGHT RECONFIGURATION	38
	APPENDIX A – DEFINITIONS	40
	APPENDIX B – ACRONYMS	41
	APPENDIX C – SPECIAL PINS	43
C.1	ACTEL RTAX.....	43
C.2	ACTEL SX	43
C.3	XILINX 5VQV	44
	APPENDIX D – SYSTEM ON CHIP (SOC) FPGA DESIGN PRACTICES.....	45
D.1	DESIGN FLOW.....	45
D.2	CODE REUSE	46
D.3	VERSION CONTROL	46
D.4	IP CORES	46

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

APPENDIX E – REVIEW OF FIELD PROGRAMMABLE GATE ARRAYS 48

E.1 INTRODUCTION 48

E.2 USE THE CORRECT FPGA DATA SHEETS 48

E.3 COLLECT THE NECESSARY REVIEW FILES 48

E.4 PERFORMING THE REVIEW..... 49

E.5 REVIEW THE POR AND RESET CIRCUITRY AND POWER-UP CONDITIONS 54

E.6 REVIEW THE PLAN FOR FUNCTIONAL VERIFICATION..... 55

E.7 REFERENCES, NOTES, AND RELATED DOCUMENTS 55

APPENDIX F – REFERENCES 57

F.1 SPECIAL PINS 57

F.2 INPUT/OUTPUT (I/O)..... 58

F.3 CLOCKS 60

F.4 FINITE STATE MACHINES 62

F.5 RESETS 62

F.6 HAZARD ANALYSIS 63

F.7 POWER 63

F.8 INTERFACING TO NON-VOLATILE MEMORIES (EEPROM, FLASH, ETC) 63

F.9 TIMING ANALYSIS 64

F.10 MISCELLANEOUS DESIGN GUIDELINES AND CRITERIA 64

F.11 RECONFIGURABLE FPGA TECHNOLOGY 65

APPENDIX G – FPGA DESIGN CYCLE CHECKLIST FOR DESIGNERS..... 65

CHANGE HISTORY LOG 98

PREFACE

P.1 PURPOSE

The purpose of this document is to discuss guidelines and criteria that form a basis for the design and evaluation of Field Programmable Gate Arrays (FPGAs) for high-reliability space-flight applications.

P.2 APPLICABILITY

This document is applicable to the development of all Goddard Space Flight Center FPGAs intended for high-reliability space-flight applications, as per GPR 8700.2, Design Development.

P.3 AUTHORITY

GPR 8700.2, Design Development

P.4 REFERENCES

500-PG-8700.2.8, Field Programmable Gate Array (FPGA) Development Methodology

P.5 CANCELLATION

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

P.6 SAFETY

NONE

P.7 TRAINING

NONE

P.8 RECORDS

Record Title	Record Custodian	Retention
Design Verification Test and Analysis Reports and/or Summaries	Product Design Lead (PDL)	* <u>NRRS 8/103</u> Engineering test and evaluation data. Temporary. Destroy between 5 and 30 years after program/project termination.

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

Page 5 of 102

**NRRS- NASA Records Retention Schedules (NPR 1441.1)*

P.9 MEASUREMENT/VERIFICATION

NONE

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

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

As space vehicle missions have become increasingly complex, the use of onboard digital computers and high-density programmable logic has become more prevalent. The functions that the avionics are assigned to perform are also expanding in number and magnitude. As a result, specifying and designing digital avionics for space vehicles has increased in complexity.

The flight performance of spaceborne digital avionics has generally, but not always, been successful. A number of recurring problems have been experienced during the design, development, and testing of these machines. Failure to develop and adhere to proven practices and processes has resulted in costly redesigns that have caused significant schedule delays or, if fixes are not implemented, caused the project to accept a higher level of risk. Most difficulties have resulted from:

- a. Lack of established or proven design/analysis practices
- b. Incomplete knowledge of the newer technologies and tools coupled with their impact on the design and analysis
- c. Inadequate reviews

This document concentrates on items that have caused problems in space flight digital hardware, particularly FPGAs. This information does not provide guidance on how to design or code a particular circuit or how to perform analyses, but instead outlines recommendations that should be considered for a successful and robust design.

This document shall be used as a design resource to supplement other available formal manufacturer’s resources including, but not limited to, datasheets, application notes, and errata for the selected FPGA device. Understanding and complying with the appropriate manufacturer’s information is vital to a successful FPGA design. Although this document cites specific FPGA examples, it is the responsibility of the designer to confirm these recommendations in this document with the latest manufacturer’s information before any implementation.

Although this document is not in a checklist format, a sample checklist is provided in Appendix G - FPGA Design Cycle Checklist for Designers. This checklist mirrors the recommendations found in this document. It is recommended that the checklist found in the appendix be used as a design aid throughout the design process. A filled in checklist can also be specified as a deliverable for design reviews.

In addition to what is textually included in this document, each of the following sections includes one or more links to additional, web-based material in the form of case studies, application notes, papers, and other material and references.

1 SPECIAL PINS

This section contains general information on special pins for all FPGAs. However, specific information on commonly used FPGAs can be found in Appendix F.1 – Special Pins

1.1 CONFIGURATION PINS: TERMINATE CONFIGURATION PINS PROPERLY

Rationale: A common problem identified during design reviews is the improper termination of special pins. For every device, carefully review data sheets and design schematics to confirm that each special pin is properly terminated. Termination of many of these special pins cannot be verified by test.

Ensure that each configuration pin is carefully checked against the latest data sheet. Some pins have very high internal pull-up resistors which can be compromised by high-speed signals on the board level. Also, some configuration pins can naturally just happen to float to the desired state with nominal operation observed. Beware of special pins such as programming pins that are required to be terminated appropriately for flight. A rule of thumb is to design defensively and ensure that intended signal levels are solid.

Different devices will have different pins and there is no overarching, general rule, other than to check each pin.

1.2 UNUSED INPUTS: DO NOT LEAVE UNUSED INPUTS FLOATING

Rationale: In general, all devices should have properly terminated inputs. For normal Complimentary Metal-Oxide Semiconductor, CMOS, devices, this is a requirement. Certain programmable devices such as FPGAs will often take care of unused pins via software, exploiting the programmable nature of the microcircuit. However, the "fine print" for each pin should be read carefully. For example, in Actel SX and SX-S, clock inputs such as HCLK or the global routed clocks do not have an output stage -- they are special purpose -- and thus have to be terminated by the user. Failure to do so can result in large unintended currents that could cause device damage.

Depending on the device, pins labeled as "N/C" may be reserved by the manufacturer for internal purposes and terminating them on the board may result in problems. Conversely, not terminating N/C's in certain cases can be bad. Check each pin carefully according to the specification and clarify with the manufacturer if necessary.

1.3 TEST INTERFACE: FOLLOW MANUFACTURER'S RECOMMENDATIONS

Rationale: Many devices have custom test interfaces that will have to be handled on a case-by-case basis. Since they hook up to test equipment, care should be taken in following the manufacturer's instructions. For example, Actel SX-S device test pins should be series terminated.

1.4 DEBUG INTERFACE: DISABLE FOR FLIGHT CONFIGURATION

Rationale: If FPGA input/output, I/O, are used to implement a debug interface for development, make sure that the inputs are safely jumpered or driven and that outputs are not toggling in the final flight configuration, causing unnecessary Electro-Magnetic Interference, EMI, and noise.

2 INPUT/OUTPUT (I/O)

This section looks at the various aspects of device I/O that a designer should consider.

2.1 SIMULTANEOUS SWITCHING OUTPUTS (SSOs): ADHERE TO THE VENDOR'S RECOMMENDATIONS FOR HANDLING SSO

Rationale: There may be limits to the number of output pins that can switch at one time. Sometimes these limits are specified by the manufacturer in a data sheet, described in an application note, and/or left to the discretion of the designer. With devices that switch faster and with large pin counts and lower Alternating Current, AC, and Direct Current, DC, noise margins, Ground/ V_{DD} bounce can be a serious issue which can dynamically affect input switching thresholds, decreasing system noise margins. It is also important to note that for many devices, t_{PD} can be negatively affected by the number of SSOs.

Care and planning is also important for pin assignments. Pin assignments that "look pretty" with all the data bits on a bus lined up in a row have been notorious for causing both ground bounce problems on the printed circuit card and routing problems inside FPGAs. Power integrity tools should be considered as they provide a means to accurately predict the effects of SSOs on a given design. Note the considerations below for simultaneous switching outputs and noise immunity and quiet designs.

Consider the following guidelines to minimize "bounce" issues.

- a. Use the lowest possible I/O slew rate and drive strength the design timing will support.
- b. Don't group SSOs together; break them up. Refer to the device datasheet for recommendations on allowable SSO signaling per ground pin.
- c. Control number of SSOs through sequencing. Example: Do address or data bus bits all need to switch at the same time?
- d. For some families, programming "unused" outputs will improve internal grounding or supply for output stages if terminated to the rail on the printed circuit board.
- e. When FPGA output drive is not sufficient, particularly for large memory arrays or long lines, use external buffers, being careful to adhere to proper PWB design techniques.
- f. Use sockets with caution. Do not use sockets for Flight applications.
- g. Choose input thresholds wisely.
 1. Transistor-Transistor Logic, TTL, $V_{IL} = 0.8V$ - very sensitive. Try to avoid this setting, as it is sensitive to both ground bounce and ringing.
 2. Try to choose input voltage threshold options, such as programmable 5V CMOS, that mitigates the effects of ground bounce
- h. Keep clocks physically away from pins that can cause ground bounce (i.e., high frequency switching pins, pins with high rise time, and address/data busses).
- i. Assign clocks to pins that are close to ground pins.
- j. Driving test data through the Joint Test Action Group, JTAG, test interface, especially over multiple parts can induce data pattern sensitivities, particularly with large data busses. For example, switching patterns from FFFFFFFF to 00000000. Though this may be an artificial failure or an artifact of the test, this can damage or potentially overstress hardware through a loss of control.
- k. Test cabling, particularly for vibration, thermal/vacuum, and EMI tests will present different conditions for normal bench testing or systems application. Design for the worst- case over the entire project flow.

- l. If applicable, consider the use of lower voltage I/O standards. FPGAs often have lower voltage I/O standards available. Lower voltage I/O have lower transient currents which can reduce SSO.

2.2 SIGNAL TERMINATION: ENSURE THAT OUTPUT SIGNALS ARE TERMINATED PROPERLY

Rationale: Start by using termination resistor values equal to the trace impedance minus the output impedance of the driver ($R_{term} = Z_{trace} - Z_{driver}$) then perform signal integrity analysis to optimize the termination resistor values.

- a. Address edge sensitive signals, such as Clock output signals, with special care to ensure that there is a smooth transition through the threshold. For loaded clocks, perhaps traveling over long runs, reflections may often result in non-monotonic transitions causing false or double clocking. Note that this may happen on the "inactive" edge. Similarly, overshoot and ringing can also cause false clocking, particularly on the transition to ground. Unterminated nets could result in ringing which is a source of EMI even when it doesn't contribute to logic failures.
- b. Most manufacturers have tight limits on how far outside the rail a signal may travel, sometimes coupled with maximum time outside of the recommended limits. Ensure good signal quality as damage to I/O's may happen.
- c. Do plan on termination resistors in advance to support signal integrity analysis efforts. The signal integrity analysis may show that they can be eliminated. However, if they are required, adding them later could require additional time in layout, debugging, rework, and/or costly printed wiring board (PWB) respin.
- d. Review schematics for proper terminations on interfaces such as the Peripheral Component Interconnect, PCI, interface.

2.3 TRI-STATE BUS CONSIDERATIONS: AVOID CONTENTION AND FLOATING

Rationale: Bus contention wastes power, needlessly generates noise, and stresses components.

- a. Avoid contention when actively driving tri-state busses. Have a guaranteed off-time between drivers on the bus in the worst-case. A clock cycle between tri-stating one driver and enabling another may be sufficient but a thorough timing analysis is necessary. Be sure to consider timing parameters, which need to be added together. For example, the tri-state time of an external SRAM's OE (Output Enable) that is

controlled by an FPGA's state machine would be the sum of the Tco ("clock-to-out" delay) out of the FPGA + the travel time on the board + the SRAM's tri-state time.

- b. Do not allow the bus to float for a long time or have slow transition times, as this will increase power and noise and may negatively affect reliability.
- c. Consider parking the bus when not in use (drive to 1's or 0's) instead of using pull-up/ down resistors. Some FPGA's have a "keeper" I/O standard which does this.
- d. When parking a bus and still using pull-up/down resistors ensure that the bus is fully driven to the parked state before it is tri-stated to avoid ringing.
- e. For portability, infer a tri-state buffer in Register Transfer Level, RTL, code instead of instantiating a device-specific tri-state buffer.

2.4 INPUT TRANSITION TIMES: *EXAMINE INPUT SLEW RATE*

Rationale: Some high-speed devices have very stringent restrictions on input transition times, often being surprisingly tight. Failure to meet the requirements may result in oscillations (Figure2-1), multiple clocking, or damage.

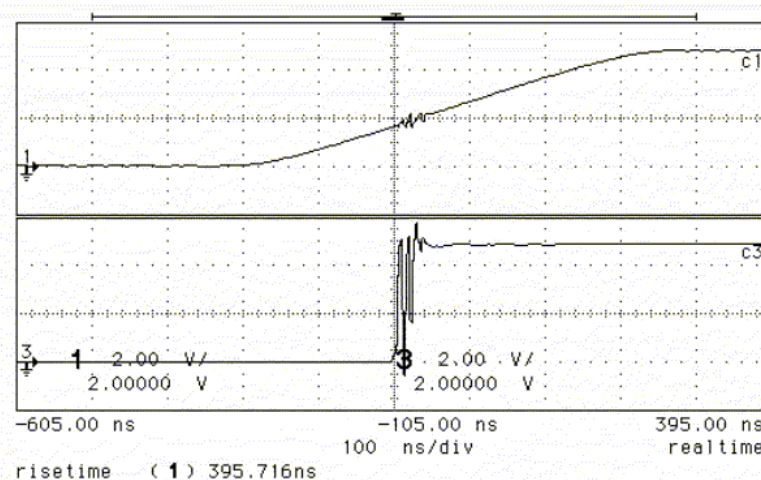


Figure 2-1 – Glitches Due To Input Slew Rate Violations

Simple pull-up or pull-down resistors, with transition times in the hundreds of nanoseconds, may be too slow. Take appropriate precautions if older digital logic families are used which may have outputs that are not compatible (e.g. too slow) with high-speed devices

2.5 SHORTING OUTPUTS TOGETHER: *AVOID SHORTING OUTPUTS TOGETHER*

Rationale: This is sometimes done to increase drive on the board. This should be avoided since it may damage components if the switching speeds are not matched and it can be

difficult or impractical to test this redundant topology. If this needs to be done, consider using an external buffer or splitting the loads between 2 or more nets, each driven by a single output.

2.6 MIXED I/O STANDARDS: *EXAMINE VOLTAGE THRESHOLDS, DC COMPATIBILITY, AND NOISE MARGINS*

Rationale: When mixing devices from multiple families, even from the same manufacturer, Exercise extreme care to ensure that the devices are reliably operated and that there is sufficient noise margin. This may be problematic when substituting parts for either upgrading circuit performance or dealing with obsolescence issues.

For inputs, many CMOS technology devices advertise "TTL compatible" inputs. However, these inputs may in fact differ rather significantly from their TTL counterparts. The first major difference for many but not all devices is the impedance presented to the interface when power is removed from the device. For example, when radiation-hardened CMOS latches were substituted for soft 54LS373's in the Galileo attitude control computer's memory units, block redundancy circuits failed since the engineers didn't take into account the sneak path through the radiation-hardened inputs electrostatic discharge (ESD) protection diodes when power was removed. Another related difference is the maximum voltage that can be applied. Some bipolar devices are useful for reliable level shifting from higher voltages to lower ones; CMOS replacement devices will forward bias the protection diodes resulting in unintended current flows and possible damage or circuit failure. Lastly, many CMOS inputs have logic thresholds, which are not truly TTL compatible. That is, the TTL V_{IH} specification is often not met, with $V_{IH(max)}$ values of 2.2V, 2.4V, and sometimes 2.5V being specified whereas true TTL devices have a threshold defined by two diode drops, typically in the range of 1.2V to 1.4V. TTL outputs are only guaranteed to drive to $V_{OH} = 2.4V$ so there may be little or even negative noise margins present in these situations. The switching point difference can also lead to circuit failure, depending on the signal integrity. Often TTL outputs, when switching, have a "bump" in the waveform, particularly with heavy and/or long loads. While this "bump" is often at a high enough voltage so that TTL devices operate correctly, the often higher V_{IH} of CMOS devices may result in multiple clocking. Pull-up resistors can restore adequate DC noise margins in these situations if given enough time to settle, which may be quite a while for this passive circuit. Note, however, that TTL to CMOS clock interfaces designed in this fashion will often fail logically since the CMOS input may see multiple transitions resulting in double clocking.

CMOS output stages can also be tricky and subtle device characteristics can cause errors. Check all specifications carefully! For example, many CMOS devices when driving loads are

specified at only very low current levels for high or logic '1' signals. However, TTL inputs take substantial currents and do not present the high impedance seen by CMOS FET inputs and the output may be dragged down. For output loads that are a mix of CMOS and TTL inputs, split the loads to guarantee the high voltage needed for the CMOS inputs, typically 70% of V_{DD} , and the high current needed for TTL inputs, with the lower V_{IH} of 2.0V. Another factor to consider is the structure of the output stage in the CMOS device. For example, some devices will not swing all the way to the high rail and are voltage limited. This may result in some totem-pole current if the p-channel FET in the next input stage is not cut off. Some devices, even with a 5V I/O supply like the RT54SX series, will only drive outputs to the core voltage of 3.3V, making this CMOS output incompatible with 5V CMOS inputs on the same board! This was fixed in the 2.5V core RT54SXS series, with full 5V voltage swings when supplied with a 5V I/O bias.

Components today can typically have many supply voltages, including 1.5V, 1.8V, 2.5, 3.3V, and 5.0V. There are also an abundance of I/O standards with the newest devices being very programmable so their characteristics are not obvious or even known from a viewing a circuit schematic. Thus, carefully verify I/O compatibility, particularly when substituting "new and improved" devices or alternate devices.

2.7 POWER SWITCHING AND COLD SPARING: *EXAMINE ALL CONFIGURATIONS*

Rationale: A system designed with blocks that are independently powered should be analyzed in all different power switching configurations. Many CMOS devices present a low impedance when powered down through either the intrinsic or ESD protection diodes; others, with cold sparing inputs, may have high input impedance that is suitable for operation. For programmable devices, selecting 3.3V PCI compatibility, as one example, can result in a "cold sparing" device no longer being high impedance since a clamping diode will be enabled. While many bipolar devices are compatible with cold sparing architectures, some devices have a sneak path (Figure 2-2) to V_{CC} through the output. Be sure to consider test setup not just the flight configuration. For example, does a piece of test equipment need to be powered up and down co-incident with the flight unit?

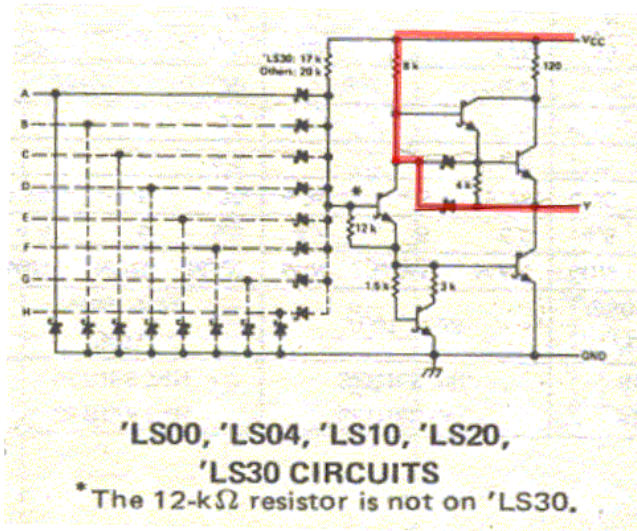


Figure 2-2 – Sneak Path in Some LSTTL From Output to Vcc

3 CLOCKS

Clocking, finite state machine design, and timing analysis are all intimately interrelated. This section will discuss some design criteria for clocks.

3.1 CLOCK BUFFERS: *USE LOW SKEW CLOCK BUFFERS ON CLOCK AND RESET NETS WHERE POSSIBLE*

Rationale: Low skew clock buffers simplify timing analysis, allow higher clock frequencies, and are less susceptible to SET events. In general, when designers use routed clock resources, the chip may more or less "work" with perhaps some unexplained glitches or a poor "programming yield" that is susceptible to specific routing. So, when sequentially adjacent flip-flops are clocked on a common edge, ensure that low-skew clock resources are used. It is acceptable to design with routed clocks and this can often result in a reduction of power or an effective increase in the number of clocks available. However, ensure that careful skew-tolerant design techniques and analyses are employed. Also, routing clock signals over long distances inside the FPGA makes it vulnerable to crosstalk from nearby aggressors. This can result in unpredictable behavior. Ensure that FPGA clock pin is close to clock buffer input.

3.2 CHIP-TO-CHIP TIMING STRATEGY: *PERFORM BOARD LEVEL TIMING ANALYSIS*

Rationale: Many analysis tools are good at analyzing logic within a single chip. However, many are ineffective at analyzing system or chip-to-chip timing. It is tempting to simply use a low-skew clock on a board to hook up various digital devices. However, that is not always guaranteed to work so employ proper timing analysis to address setup and hold time. This is often overlooked or done improperly. While the worst-case behavior of the clock-to-out of the source chip is easily analyzed using "minimum" or "best case" timing parameters, analyze the hold time of the sink chip assuming a slow path for the clock and a fast path for the data, for the same calculation. Automated tools often do all min or all max but are not capable of doing a mixed analysis; often requiring a human to perform this task. A good goal for the sink chip is to have a hold time of 0 ns or less (negative hold) but many devices, particularly some models of FPGA, do not satisfy this condition. So, alternate techniques for passing signals should be used, such as opposite edge clocking, treating signals as asynchronous, etc. The criteria for passing is that all worst-case setup and hold times are always satisfied or that sufficient metastable state protection is included.

3.3 DELAY LOCK LOOPS, DLLS, AND PHASE LOCK LOOPS, PLLS: *CONSIDER THE AMOUNT OF ANALYSIS REQUIRED BEFORE USING THE DLLS AND PLLS*

Rationale: DLLs and PLLs can have many useful functions in digital systems. However, understand and address all design considerations. First check that the worst-case frequencies (both slowest and fastest) are compatible with the circuits; often the acceptable ranges are very limited. Additionally, there are often signal quality conditions that need to be satisfied. Next, when these circuits clock finite state machines or other sequential logic, note the time to lock and stabilize for these circuits and ensure that the device and system powers up safely. Another item to check is the worst-case performance when the DLL or PLL is hit by an SEU. This can result in a change of programming of the DLL or PLL, which is sometimes a little subtle, or a change in mode. Ensure safe operation of the system occurs during these off-nominal conditions. Furthermore, an SEU or SET can cause the DLL or PLL to unlock or glitch and consequently make the entire circuit that is within the clock tree become chaotic (unstable) or inoperable, necessitating a reset.

Use of internal DLL/PLL FPGA circuitry should not even be considered without careful analysis of the program’s radiation requirements and radiation test data on the FPGA’s DLL/PLL circuitry.

3.4 CLOCK TREE DIAGRAM: *DRAW A BOARD LEVEL CLOCK TREE DIAGRAM*

Rationale: A diagram should be drawn showing the clock trees for the circuit. These diagrams should include PLLs, DLLs, clock buffers, clock dividers, and all chips that use the clock. See Figure 3-1 below as an example.

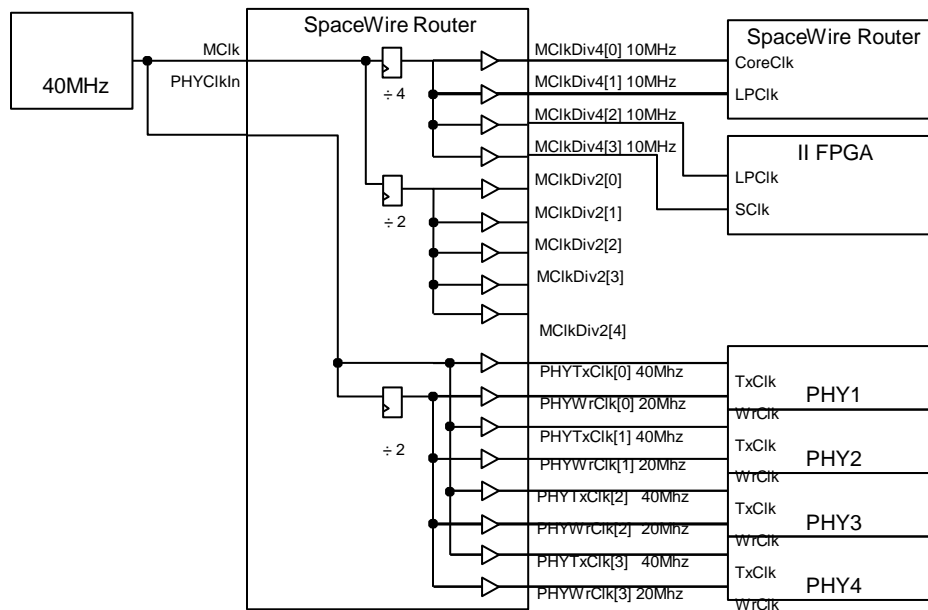


Figure 3-1 – Sample Clock Tree Diagram

3.5 CROSSING CLOCK DOMAINS: *PERFORM CLOCK DOMAIN CROSSING ANALYSIS*

Rationale: Based on analysis of the clock trees, identify all blocks and signals crossing clock domains and determine the need for metastable state resolution. Additionally, ensure that the latency involved in signal synchronization is tolerable to the system. Note that synthesis and place and route tools can help with this analysis.

3.6 FLIP-FLOP REPLICATION: *AVOID UNINTENDED FLIP-FLOP REPLICATION*

Rationale: Flip flop replication should be avoided in re-synchronizers because one part of the re-synchronizer will regularly disagree with another due to meta-stability and routing delay differences. This is likely to cause unpredictable behavior.

3.7 OPPOSITE EDGE CLOCKING: *CONSIDER DUTY CYCLE IN TIMING ANALYSIS FOR DESIGNS THAT USE BOTH CLOCK EDGES*

Rationale: For designs passing data from one edge of a clock to the other, ensure that the worst-case duty cycle for each phase is properly computed. Often designers will assume a 50% duty cycle which is not the case. Sources of duty cycle distortion include oscillator characteristics where duty cycle variation can be as much as +/- 10% and uneven delays through logic gates and buffers. If opposite edge clocking is not required consider avoiding this technique as it complicates the timing analysis for the design.

3.8 METASTABILITY: *FILTERING TECHNIQUE SHOULD CONSIDER THE DURATION OF THE COMPONENT METASTABILITY*

Rationale: Ensure that proper synchronizers are used for each asynchronous signal. Often designers will simply use two series D flip-flops. While an often used and acceptable topology, for very high-speed circuits for the technology in question, the failure rate of this synchronizer can be non-negligible. It may be necessary to add a third series D flip-flop. Also note the conditions for which the flip-flop's metastable parameters are taken, with large differences possible in resolution time when moving from nominal temperature and voltage to the extremes. Ensure that there is margin in these circuits as they are impractical to test and verify. Also note that for ASICs, different flip-flop macros may have significantly different metastable parameters. This can also be a consideration in FPGAs.

3.9 LATCHES : *AVOID THE USE OF LATCHES*

Rationale: Use of latches complicates the timing analysis of a design. Furthermore, place and route tools do not analyze timing paths with latches well. It is not uncommon that a latch can be replaced in a design by a flip flop. Therefore, the recommended approach is to replace latches with flip flops. <http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

4 FINITE STATE MACHINES, FSM

What is the best style for a finite state machine? Should the human or the machine perform state assignment? How do we design safe finite state machines? There is no best answer for all situations and there is no magical style to be checked. It does, however, have to follow the basic principles of good logic design. It is noted that many engineers now use Hardware Description Languages, HDLs, to design the state machine and never see the logic. Extreme care is a requirement for critical applications. Finally there are very few designs with a single independent state machine. Most designs have several, if not many, interconnected state machines. Any correction algorithm would need to take into account all of the interconnected state machines and be thoroughly analyzed and tested in order to verify the proper operation. The correction method may even reside at a higher level such as at the subsystem card, or box level.

4.1 CRITICAL STATE MACHINES: *ANALYZE ALL POSSIBLE STATE TRANSITIONS AND IMPLEMENT A DEFAULT STATE*

Rationale: For critical state machines, the analysis should cover all possible logic states and demonstrate that the machine behaves in a deterministic and desired fashion. This includes consideration of off-nominal events. One credible failure mode example is an SEU. State machine analysis should include all physical states (all possible state vector values). It is a credible failure mode to be in any of these states as a result of a disturbance on the power bus, an ESD event, etc. Any high reliability machine is required to be robust under all credible failure modes. Additionally, verify that the FSM always starts in a legal state and then transitions through the desired sequences. One method is to use a power-on reset (POR) indicator. This should be checked to ensure that it is synchronous with the clock. One may not need any reset for a finite state machine if it can be shown to always go into a desired state. This can be done in the trivial case of a divide by n master counter, for example, where a reset is not needed and a fault on the reset line can halt the machine. Another technique is to gate the inputs with the POR signal and design an FSM such that it is guaranteed to go into a hold state. One consideration with the reset function is design-for-test and design-for-simulation, which sometimes results in additional reset connections.

4.2 STATE ENCODING: USE STATE VECTOR ENCODING SCHEME THAT MEETS REQUIREMENTS

Rationale: The choice of state vector encoding is one that should factor in radiation effects, timing

constraints, and criticality of operation. Often, designers allow the synthesizer to choose the encoding scheme which is optimal for timing constraints; however the designer should review the synthesizer's choice factoring in radiation effects and criticality of operation. Here are some factors to consider.

- a. In FPGA where the flip-flops are inherently triplicated, upsets are more common from SETs than SEUs, thus, combinatorial logic poses a greater vulnerability. One-hot encoding uses more SEUs, but less combinatorial logic to encode the states and becomes a robust high-speed option.
- b. With one-hot encoding, all single bit errors are detectable, however, when one-hot encoded state machines experience an upset, it is likely that two state bits will become 'hot' and activate two parts of the design that aren't normally activated simultaneously. The designer should consider if this situation could cause any damage.
- c. With binary coded state machines, detecting illegal states and transitions requires the use of additional logic which increases susceptibility to radiation effects.
- d. It is tempting to think that if binary encoding is used and all 2^n are defined that the FSM cannot lock up, however, this may not be the case if the FSM 'hand-shakes' with external logic. In this situation, an SEU could disrupt the normal sequence of operation and cause grid-lock.
- e. In the big picture, the difference in upset rates between state machine types is insignificant as they all show very low upset rates. Hence, a basic rule of thumb is to use one-hot encoding for speed and binary encoding for circuits with a large number of states.

4.3 HDL SYNTHESIZED MACHINES: *ANALYZE THE SYNTHESIS REPORTS AND SYNTHESIZER OUTPUTS*

Rationale: Obviously, all of the criteria for schematic-based machines apply. However, there are special considerations for designing with HDL, as the Computer-Aided Engineering, CAE, writer might generate circuits that are not desirable for high-reliable circuits. Hence, for critical circuits, examine the output reports from the synthesizer very carefully. Common things to check for include: recognized state machines; lockup states; outputs of Gray encoded machines that can glitch: unintended flip-flop replication; not implementing the desired and specified style (sometimes the synthesizers just think they know better than the human and will substitute one type of state machine for another). Additionally, some logic synthesizers will generate "safe" state machines. Use of this feature is not recommended because it typically increases the use of combinatorial logic which increases the SET susceptibility. If this feature is used, examine the generated design carefully. For

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

instance, it has been seen that sometimes the logic will explode with excessive gates. Other times there are resets generated on the opposite edge of the clock resulting in tight timing for the removal of clears which are not visible to the designer. Note that when using enumerated states in HDLs, not all physical states will be covered (only enumerated states are covered). Hence, the "others" clause will only refer to states in the enumerated type and not the physical realization. The HDL doesn't know if it is a one-hot or binary or gray coded implementation and what flip-flops have been replicated. This is not detectable at the black box simulation level nor by Boolean equations for logical equivalence.

4.4 ERROR DETECTION AND CORRECTION, EDAC: *ADDRESS FSM LOCKUP AT A HIGHER LEVEL AND ANALYZE FSM OUTPUTS*

Rationale: It is often tempting to design robust state machines by simply appending a Hamming code and correction circuits. Hazard events are not synchronized to the system clock and the logic network is not guaranteed to be glitch free; do not rely on the ability of this type of structure to provide robust operation. In the general case, analyze the combinational circuits which implement the next-state logic and their inputs to the flip-flops making up the state register. In particular, for any of these schemes, look at whether or not the circuit implementations are static hazard free (see Section 6) and, if not, can an erroneous transition to a state (or set of states) be made.

Heavy ion testing has proven that upsets in the EDAC logic will upset a state machine. Thus, EDAC protection for FSMs is not recommended.

In general, each FSM should be analyzed to make sure that the system can detect a locked up FSM and return it to a known state in a timely manner. This can be automated into the design or as simple as power cycling the box if acceptable to the system design. For mission critical applications, the FSM outputs should have external protection requiring FSW to 'arm' the FSM outputs.

5 RESET

The term “reset” in this document refers to signals that drive asynchronous reset or preset inputs of clocked logic, typically flip-flops.

Synchronous resets can typically be analyzed with just static timing analysis as far as the FPGA is concerned but require board, box and/or system analysis to determine if indeterminate FPGA outputs are acceptable until the synchronous reset is clocked through.

5.1 RESET LOGIC CIRCUIT (CONSIDER TRANSIENT BEHAVIOR): *PROVIDE SUFFICIENT NOISE MARGIN, ADEQUATE SLEW RATE, AND GLITCH FILTERING*

Rationale: Transient effects analysis are a major focus when analyzing reset circuitry performance. For the application of power, the output of the POR or reset circuit should ideally be a solid logic level and be glitch-free. This requires the POR circuitry to be designed using logic elements which operate correctly at the low ramping up voltages seen during power up. This insures that the POR signal is active at earliest time possible in the power-up-down sequence of events. Verify Inrush currents to timing capacitors do not exceed the maximum for that capacitor type. Verify rise times to logic gates, if used as a comparator, do not exceed the gate input's specifications; often gates with hysteresis inputs are used. Note that even with that type of input, output glitches may occur and several stages of logic gates may be required. The most robust solutions often utilize a comparator. Another transient factor to consider is the rise time of the flight power supply, both best and worst cases. These will often differ substantially from laboratory supplies and may be non-monotonic or have substantial overshoot and ringing. Note that flight power supplies are often slew-rate limited to minimize conducted emissions on the power bus. The time constant of the supply may exceed that of the POR circuit! For discharge, ensure that there is a low impedance path for timing capacitor discharge and that the inputs of logic gates are protected. Most CMOS inputs, but not all, have ESD diodes from the input to the supply rail. Discharging a large capacitance through that input may damage it. Also, consider the requirements and response of the circuit to momentary disruptions on the power bus. While many circuits may recover or be recoverable from a power-on reset, this is not true for all circuits. One such example is non-volatile, erasable memories, which need to be carefully protected.

Many diagrams of reset circuits show "asynchronous application, synchronous removal" of the reset circuit. However, note that for many devices, in particular many programmable devices, the inputs can be blocked or ignored during the power-on transient. This may be

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

because of the need for charge pumps to start or configurations to be loaded and then released. For devices with synchronized inputs, clock oscillators are required to start, perhaps taking many tens of milliseconds, before the reset can be applied. This is a board-level consideration that is necessary to prevent the reset, which may look just fine on the schematic or in the HDL code, from being ignored by the real circuits. See Figure 5-1 below.

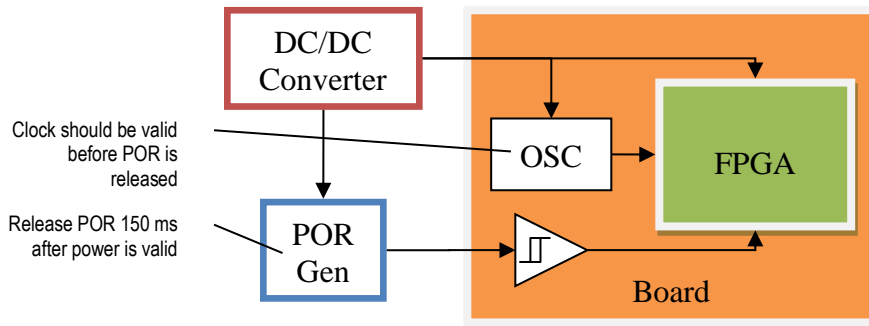


Figure 5-1 – Recommended Power On Reset Implementation

Steady state or DC effects are also important. Check the leakage currents of timing capacitors and logic gates, as the amount of leakage current times the resistance of the timing resistor may result in a voltage drop that eliminates all noise margins.

5.2 RESETS: *IMPLEMENT SYNCHRONOUSLY DE-ASSERTED RESET USING A GLOBAL BUFFER, IF AVAILABLE*

Rationale: For asynchronous presets and clears, there are two basic parameters that need to be met. First, removing the preset or clear from a device asynchronously to the clock may result in meta-stable states in the sequential circuit. This parameter is frequently called the removal time and is denoted as t_{REM} . Unfortunately, many data sheets do not specify the removal time. Use a synchronously de-asserted reset to ensure that the removal time requirement is met.

Second, if a global buffer is not available, the reset signal should be buffered to meet overall fan-out constraints. If this buffering takes too many resources, then the reset signal's fan-out constraint may be relaxed as long as timing is still met. Also, some flip flops may not need to be reset if their initial values don't matter.

5.3 RESET TREE: *GENERATE RESET TREE DIAGRAM*

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

Rationale: Drawing a tree of all of the reset sources, buffers and domains is often helpful in ensuring that the reset logic is well defined. Often there are multiple forms of reset from system resets, software resets, watchdog timers, etc., and having a good tree diagram shows the relationships between them. Ensure that proper synchronization is made when required. Additionally, if the reset needs to be activated fast, for instance to protect non-volatile memories from false writes, or other circuits from initiating one-time events such as firing pyrotechnics, the tree will help ensure that the logic and delays are well understood.

5.4 COMPONENT STARTUP TIME: *ENSURE THAT THE GUARANTEED RESET TIME IS SUFFICIENTLY LONG FOR ALL CIRCUITS IN THE SYSTEM*

Rationale: Many FPGAs require time to "start," where a charge pump builds up voltage and charges internal capacitances, waits for a delay, and then releases its outputs. Premature release of the POR signal may result in an indeterminate state. Other FPGAs may require a sequence of resets for proper loading and release, with many circuits having internal power-on reset circuits. Therefore, analyze best and worst case timing behavior for all resets. Additionally, some standard components on digital logic boards such as crystal clock oscillators can have a substantial startup time, often many tens of milliseconds. Complicating this further, components such as FPGAs and crystal clock oscillators may have startup times that are a function of the rise time of the power supply. Even worse, this behavior is often poorly specified or not specified at all. Robust start times are critical.

5.5 MISSION CRITICAL SIGNALS: *PROTECT MISSION CRITICAL FPGA OUTPUTS DURING POR*

Rationale: Note that many logic elements do not follow their truth tables as the power supply ramps up. Thus, design the POR signal to act as a gate (via external circuitry) to block false signals during the power supply rise time transient and then to release after all circuits are stable. On the other side, when the power comes down, the POR circuit may need to be asserted early, ensuring that critical circuits are safe before the logic elements lose control as the voltage drops. Devices that often need protection are pyrotechnic initiators, Electrically-Erasable Programmable Read- Only Memories, EEPROMs, flash memories, etc. Note that some devices such as microcontrollers have internal flash memories, so evaluate all components and system interfaces for necessary protection by the POR signals.

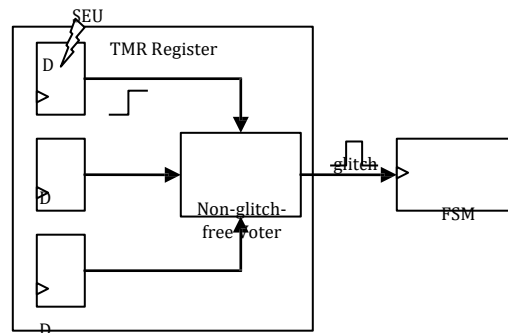
6 HAZARD ANALYSIS

6.1 STATIC HAZARD: *USE SYNCHRONOUS DESIGN TECHNIQUES AND PERFORM STATIC TIMING ANALYSIS*

Rationale: A static hazard exists when a change to a single variable to a combinational network causes a transient or unintended momentary change in other variables to occur (e.g., 1→0→1). Normally this is not a problem in synchronous design as long as there is sufficient time for the signals to settle. When the output of the combinatorial network is being used by a circuit sensitive to static hazards, they should be filtered out.

6.2 DYNAMIC HAZARD: *USE SYNCHRONOUS DESIGN TECHNIQUES AND ANALYZE EDGE SENSITIVE INPUTS IN YOUR DESIGN*

Rationale: Dynamic hazards, exists if there is a transition of the form (1→0→1→0). That is, it did not switch cleanly. Any circuit free of static hazards will be free of dynamic hazards. This topic is not covered in many logic classes and with the use of HDLs and functional simulation many designers are not familiar with these concepts. For 100% synchronous designs with a single clock and a common edge there are normally no concerns. Yet during reviews hazards are often present, unknown to the designer. One example of this is the use of Triple-Module Redundant, TMR, circuits to generate a clock signal to a finite state machine. The change in one input to the voter, which is used to mitigate the effects of SEUs, can result in a double clock from the "glitch" coming out of the voter, unless the voter is hazard free. Often a component will appear to be hazard free, but carefully analyze the implementation in the logic family that you are using. For example, are multiplexers, the foundation of some FPGA families, glitch free? There is no guarantee that they will be and hence cannot be considered safe clock generators without a lot of care. Another example is when a voted output is brought off-chip and used as a clock input for an external device. Logic synthesizers have been observed to generate hazards in the circuits they generate, unknown to the engineer running the tool.



DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

Page 25 of 102

Figure 6-1 – TMR Voter Hazard

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

7 POWER

7.1 SUPPLY SEQUENCING: *FOLLOW DEVICE FAMILY DATASHEET TO ENSURE PROPER POWER SEQUENCING PROFILE*

Rationale: Many of the newer technology devices require two or more power supplies. Often these are divided into supplies to power the core of a logic device and a second supply to operate the Input/Output cells. Additional supplies may be needed for PLLs and DLL's, special I/O standards, or various bias supplies such as external charge pumps. It is obvious that the supplies should meet all of the DC standards as well as ripple characteristics, particularly for circuits such as PLLs. What is often not obvious is that the sequence that power is supplied to a single device can, in certain cases, affect circuit behavior and performance as well as reliability. For certain devices, such as SX-S series devices, if the I/O supply is brought up before the logic core, then a large inrush current may be present; this would not be the case if the order of the supplies was reversed. For certain devices, incorrect power sequencing can result in overstress or damage. This is the case for multiple vendors. Often the requirements for sequencing are in either application notes or the "fine print." When parts that require sequencing are present, they should be flagged and the design should be done very carefully, incorporating circuit protection, as required. Note that power sequencing requirements may differ between flight and prototype devices.

7.2 SIGNALS INTO UNPOWERED CMOS I/O'S: *ANALYZE DESIGN FOR SNEAK PATHS BETWEEN I/O THAT INTERFACES POWERED AND UNPOWERED DEVICES*

Rationale: The power supply sequencing between interfacing IC's, either on the same or separate boards, should be carefully considered. Many IC's, particularly CMOS ones, present a low impedance to the system when powered off. Most of these IC's require that the power supply be brought up prior to the application of signals on either the inputs or the outputs (many FPGA outputs also have inputs active in the general purpose I/O modules). Some programmable Integrated Circuits are not analyzable by inspection, The specific design details are often needed to do a proper worst-case analysis. For instance, some I/O modules provide for cold sparing; that is, they present a high impedance to the system when powered off. That same I/O, configured differently, may have clamp diodes switched in while powered off for PCI compatibility.

7.3 STARTUP VOLTAGE RISE TIME: PERFORM VOLTAGE SUPPLY RISE TIME MEASUREMENT ON ACTUAL DESIGN AND VERIFY THAT THE RESULTS MEET FPGA REQUIREMENTS

Rationale: Startup current transients are common in many devices. The size of the current can be a function of time between power cycles, temperature, ramp rate of the supply, radiation exposure history, power supply sequencing, etc. These currents can be rather large for certain devices, often as high as several amps. It is critical that the power supply system does not limit current in these cases to steady state levels with margin as insufficient current during the startup sequence can result either a failure to properly initialize, power device shutdown or recycling in an infinite loop, or a system lockup, the deadly embrace. Similarly, some parts have hard restrictions on minimum and maximum power supply rise times; failure to meet these levels may result in circuit failure.

7.4 BYPASSING AND DISTRIBUTION: *FOLLOW DEVICE DATASHEETS AND/OR DEVICE APPLICATION NOTES FOR PROPER DECOUPLING*

Rationale: Logic devices can be rather large, consisting of billions of gates. Synchronous design techniques, high operating frequencies, and large I/O counts can result in a challenge to the power distribution and conditioning system. Most of the manufacturers supply details in application notes. These rules should be followed unless a power integrity analysis and testing of the system for worst-case conditions proves otherwise. Worst-case test patterns can be exploited to ensure high-fidelity power and then replaced with the flight application. JTAG interfaces may also be used and care should be given that the JTAG test patterns do not violate design limits, such as SSOs.

8 INTERFACING TO NON-VOLATILE MEMORIES (EEPROM, FLASH, ETC)

8.1 PROTECTION DURING POWER-UP/DOWN TRANSITIONS: *SHOULD HAVE POWER DOWN WARNING AND ENOUGH BULK CAPACITANCE TO COMPLETE A WRITE ACCESS*

Rationale: This has been noted as a common problem for erasable non-volatile memories. The analysis and test should carefully examine all of the signals for proper and safe operation during power-up, power-down, and brown out transients. Note that the actual power supply and its bounded characteristics should be used, not laboratory supplies which most likely will have substantially different characteristics. Some devices have a reset pin to help protect against inadvertent writes. The design, analysis, and test/evaluation of this circuit under all conditions is critical for maintaining the integrity of the non-volatile memories contents. Consider circuit operation if the power is shut down during a write cycle, either planned or unexpected and the design should ensure the proper completion of write cycles to ensure that the contents of the non-volatile memory is protected. The write cycle often includes not only the time for the bus operation to complete, but for the time for writing internal to the part, which can take on the order of 10 ms. Another related consideration is the unexpected application of a system reset signal. Shutdown states should be entered to help ensure that write cycles are fully completed and properly shut down, with the critical signals put in a safe mode.

8.2 ANALYSIS OF DAMAGE DURING WRITE CYCLES: *IMPLEMENT MECHANISM TO DETECT CORRUPTED WRITES*

Rationale: The technology of the non-volatile memory should be carefully considered if the memory is to be written in flight. Some of these devices, such as EEPROMs, use high voltage to write the cell. If struck by a heavy ion with high voltage applied, the failure mode should be analyzed and dealt with appropriately. Thus, writing in flight should be considered a high risk operation.

8.3 CYCLE COUNT: *DESIGN INTERFACE TO MAXIMIZE USEFUL LIFE OF MEMORY*

Rationale: Many non-volatile erasable memories have limited number of access cycles. Treat each device on a case-by-case basis with system lifetime and radiation factored in. There are some subtle specifications that will be noted here, as examples. The 128k x 8 Hitachi die, for example, has a lifetime write specification limit of 10^4 cycles in byte mode with 10^5 cycles in page mode. The write mechanism for this device utilizes an 8-byte subpage as the smallest unit that can be written. Hence,

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

writing the same memory space one byte at a time is more stressful than page writes since entire subpages are first fetched and then re-written.

8.4 TRANSIENTS AND NOISE: *TREAT CONTROL SIGNALS TO NON-VOLATILE MEMORIES AS CRITICAL BY MINIMIZING SSO*

Rationale: It is critical that the signals interfacing with non-volatile memories be clean and system noise kept to a minimum and always meet all specifications. In this case, signals includes not only logic signals but power and ground connections; robust bypassing should be used. Noise glitches on EEPROMs, for example, can cause false write cycles to be generated, resulting in inadvertent altering of the device's contents. See 2.1 above.

8.5 RELIABILITY: *DESIGN INTERFACE TO IMPLEMENT REQUIRED EDAC*

Rationale: The required reliability of the non-volatile, erasable memory device is highly dependent on its application. If the device operates as part of a large memory array, then some bit failures and even page failures can be tolerated either by error correction techniques or by error detection and mapping the failed segment out of service.

Applications such as boot Read-Only Memory, ROM, for a central processing unit or memory contents for an FPGA, require perfect system performance. For single bit failures a Hamming code may suffice, although that may be awkward for serial Programmable Read-Only Memory, PROMs. Note that some failure modes of non-volatile memory devices may result in a bit oscillating or not providing a valid logic level; in this case, an EDAC device may or may not correct the single bit error, depending on the logic design of the EDAC device being used and whether or not it is static hazard free. In any event, ensure that the devices employed, combined with the architecture of the particular system, are not susceptible to lockup states from any credible failures. Credible failures include any single bit error and an inadvertent corruption of a non-permanent memory's contents.

Other forms of redundancy may be required such as TMR with switchable spares. Some options include the ability to switch in alternate devices, the use of permanent memory such as PROM, or the use of storage buffers to replace erasable non-volatile memory functions, using operational overhead to manage the risk. For example, if a configuration memory device for an FPGA fails, a storage buffer and Central Processing Unit, CPU, may configure the FPGA using a different loading mode, assuming that, of course, the FPGA isn't needed to run the computer. In general, for critical applications, permanent memories such as PROM are to be used to ensure that the spacecraft or other system cannot be permanently lost. This can take the form of boot and safe-hold code for a processor or a basic operating configuration for an FPGA.

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

8.6 REFRESHING AND RELOADING: *FACTOR THE DEVICE'S SPECIFIED DATA*

RETENTION AGAINST MISSION LIFE

Rationale: Another consideration is the guaranteed storage time of the device vs. mission length. There is no hard and fast rule so analyze each device on a case by case basis. Ten years is a frequent specification for the retention of memory contents, however, system lifetimes of several decades is not uncommon. Refreshing can be risky and the usefulness of it should be verified with the manufacturer's assistance, to ensure a guarantee of storage integrity, particularly in the radiation environment. Obviously, when the device is refreshed, it may be susceptible to the contents, such as a computer crash, brown out, or the unexpected removal of power due to a bus fault or a spacecraft entering a safe mode. Also, each write cycles takes away from the operational lifetime of the component.

8.7 RECOMMENDATIONS AND TIPS

- a. Many designers use a simple RC timing circuit for the generation of a POR or "Power On Reset" signal. Looking closely at the acronym, it has the word "on" in it and the "O" does not stand for "Off." Use of such a circuit will often protect memories for power up but assertion of the protection circuit will lag either during a brown out or when power is removed.
- b. POR circuits are often best generated in the power supply module.
- c. Ensure that critical memory controls behave properly during power transient conditions. They are often incorrectly implemented by an FPGA that is not guaranteed to be under control during the power-on, power-off, and periods when power is disrupted. FPGA and configuration memory device internal power-on reset circuits may be active along with initialization sequences, charge pumps have to supply sufficient charge and voltage to turn on high-voltage isolation Field Effect Transistors, FETs, etc.
- d. Erasable memory device protection is typically an analog function so take caution if digital functions are used in protection circuits. Along with timing, many memory devices require non-standard voltage levels and currents for protection.
- e. Device Protection: Consider the likelihood of a software fault is 100%. Many erasable devices implement "software write protection" to prevent against inadvertent writes to the memory. Joint Electron Device Engineering Council, JEDEC, has published a standard on this type of protection. Do not keep the "keys" to unlock the memory on-board unless absolutely necessary.
- f. Subsystem Protection: System level write protection limits should be implemented in

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

hardware, to protect against software faults. Some systems implement this in software which is risky; see previous item. Use external hardware discrete command as an additional barrier to prevent inadvertent writes.

- g. Analyze and test devices for lockup states. These can occur in many memory types from illegal loads into command registers, poor signal integrity, poor power quality, or an SEU. Some device lockup states require power cycling to clear. Lockup states in memory devices are often not considered either in memory controller designs (soft repairs) or system designs (power cycle required for clearing of faults).
- h. Critical switching between memory images for booting implemented as a software function cannot be guaranteed to function under all credible faults resulting in system lockup. Use a hardware signal to implement recovery from faults to prevent system lockups.
- i. Boot and Safe-Hold Code: High-reliability, radiation-hardened, ROM-based memories should normally be employed for boot and safe-hold functions. For applications such as instruments, Direct Memory Access, DMA, functions, properly implemented, can load memories with boot code. In this case, the instrument should be safed by hardware logic. It is recommended that boot-up copy functions should not require any operational software and hardware should clamp the processor into reset.
- j. Verify Margins of All Protection Signals: DC voltage margin; AC voltage margins (e.g., cross talk); Timing (protection signals for power up, power down, and during glitches). The power down rate of voltage buses is often ignored or idealized.
- k. Multiple copies of the same code in the same technology is risky, if the fundamental technology is not reliable. Storing redundant copies of code in separate blocks of one device can be subject to common mode failures.
- l. Treating bit, block, and device failures in software can be done in many instances, such as recorders. However, for critical boot code, as an example, treating failures should not be a function relegated to software.
- m. Consider using a Cyclic Redundancy Check, CRC, or checksum in the non-volatile memory, which is updated during writes to help with detecting corrupted writes.

9 TIMING ANALYSIS

Timing analysis of digital systems can be summarized quite simply: ensure that every parameter on the data sheet is met for all elements of the design. In practice it can be a significant effort and care should be taken to ensure that the calculations are performed correctly. A circuit properly designed and analyzed will work properly for all combinations of components over the entire specified operating environment.

9.1 CLOCKS: *ANALYZE MINIMUM PULSE WIDTH AND JITTER*

Rationale: The basis of all timing analysis is the clock and the flip-flop. The clock, for both high and low phases, has to meet minimum pulse width requirements. Certain circuits, such as PLLs, may have other requirements such as maximum jitter. As the clock speeds increase, jitter becomes an increasingly important parameter. For clocks that are close to the device's specifications, note how the high and low time are measured and the characteristics of the clock, as the threshold voltage may differ between the specification of the clock and the input device. Also, the transition time of the clock signal, effected by loading and the environmental factors, can degrade the available pulse width. Failure to maintain a proper pulse width can result in the flip-flop going "metastable."

When "passing" data from one clock edge to the other, ensure that the worst-case duty cycle is used for the calculation. A frequent source of error is the analyst assuming that every clock will have a 50% duty cycle.

9.2 FLIP FLOPS: *ENSURE THAT SETUP AND HOLD TIMES ARE MET FOR EACH FLIP FLOP EXCEPT RESYNCHRONIZERS*

Rationale: Verify that all flip-flop parameters are always met. The only exception to this is when synchronizers are used to synchronize asynchronous signals, the topic of another section of these guidelines.

For data (or J, K, T, EN, synchronous clear, etc.) inputs, show that all setup and hold times are met for the earliest/latest arrival times for the clock. One of the leading causes of digital logic malfunction is hold time violations. Check the specification for the device carefully, for FPGAs, to see if the manufacturer will guarantee that hold times will always be met when using the global clocks. This is not always the case.

When passing data from one clock domain to another, ensure that there is either known phase relationships which will guarantee meeting setup and hold times or that the data signals are properly resynchronized.

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

9.3 ENVIRONMENTAL EFFECTS: *ENSURE THAT CAE TOOLS ARE CONFIGURED PROPERLY TO ANALYZE ENVIRONMENTAL EFFECTS*

Rationale: For robust circuits, designs have to be tolerant of various environmental effects. These include:

- a. Temperature
- b. Voltage
- c. Aging
- d. Radiation

In general, analysts will do an extreme value analysis based on the widest possible corners of each environmental factor, simultaneously. This will result in a system with very wide margins and tolerance of unforeseen, off-nominal conditions. However, this process will also in many cases needlessly limit performance, increase resource consumption, or force more complex architectures and analysis. For example, for two flip-flops located on the same die just a few microns apart, one flip-flop will not be at -55 °C while its neighbor is at +125 °C. Assuming 100% tracking is not valid either for this parameter; for others, no tracking can be assumed. Often the designer/analyst will be limited by the data and/or models available and will not be able to determine how much tracking will occur. In this case, the least amount of tracking will have to be assumed, a conservative approach.

The temperatures and voltages used will be a function of each particular mission and the location of the electronics. Ensure that worst-case values are used plus margin, as specified in the project's reliability plan, and not the more optimistic expected values. There have been many missions where the actual values were outside the bounds of the expected values.

Component performance and characteristics change with age and radiation exposure. However, one cannot assume that all propagation delays, as an example, will track and that the relative delays will remain unchanged. For example, studies of life test data for certain FPGAs showed that not only will the delays not track, but that they may not even have the same sign, with devices sampled from a single manufacturing lot. Hence, one cannot demonstrate hold time margin by test. In general, most programs will specify $\pm 10\%$ for propagation delay change over the mission lifetime. Be sure to meet your programs timing requirements.

9.4 SPEED GRADE: *USE THE CORRECT SPEED GRADE PART FOR THE BEST CASE ANALYSIS*

Rationale: The speed grade setting can be misleading for timing analyses. For the worst or slowest case, the speed grade, as stamped on the part, is the correct setting to use. For

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

the best or fastest case, using the speed grade on the case can potentially give you an incorrect answer. For example, some FPGAs are binned by measuring their speed and ensuring that it is more than some slower threshold value. This may be a one-sided relation and parts that would have passed a faster speed grading might be binned and stamped with the lower one. So, to be conservative, the designer may elect to run an additional analysis using the fastest speed grade the tool supplies.

9.5 ASYNCHRONOUS CIRCUITS: *ANALYZE RACE CONDITIONS OVER ENVIRONMENTAL EFFECTS*

Rationale: Asynchronous design techniques are difficult to analyze, error-prone, and are thus discouraged in FPGA designs unless required. If asynchronous circuits are implemented, then the designer should analyze race conditions over environmental effects. Typical timing analysis programs will allow one to select a setting for process, typically best, typical and worst. To effectively use these settings, note that this is not a predictor of circuit speed but a bound for circuit speed. Many engineers and analysts assume that this will predict speed or prove that two circuits cannot lose a race. This is not the case. For example, no two transistors will be processed identically, although often it will be fairly close. There are lot to lot variations, wafer to wafer variations within a lot, die to die variations on a wafer, and transistor to transistor variation on a die. Hence, treat these values as bounds and not as actual values. There will be a certain degree of tracking. How much you can use in an analysis depends on the data available and algorithms available in the CAE tools.

For anti-fuse based FPGAs, the amount of "tracking" that can be assumed in an analysis will be less than is often found in other device types. While the transistors on a die will track to a certain degree, as they are fabricated together, the distribution of programmed anti-fuse resistance will resemble a random variable which depends on voltage, temperature, device dosage, process variations, etc.

Taken together, this means that if you wish to guarantee that signal A always arrives before signal B by T nanoseconds, running a dynamic simulation with all values set to the worst-case will give an incorrect answer as there is no guarantee that all paths will be the worst. In reality, they will not. That is why min-max or extreme value analysis is required for accurate timing analysis.

Furthermore, it should be noted that asynchronous circuits are more susceptible to transients than synchronous circuits. Such transients can lock up an asynchronous design and are generally not able to be mitigated. Designers should stay away from asynchronous designs whenever possible.

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

9.6 TIMING MARGIN: *VERIFY THAT TIMING REQUIREMENTS ARE MET WITH ADEQUATE MARGIN*

Rationale: Typically, designers will use 10% as the goal for timing margins. Many designers will interpret this to mean that the FPGA should be designed to run with a clock that is 10% faster than required. While this technique does ensure setup time margin, it does not address hold time or I/O margins. See diagram below. To ensure adequate hold time margin, analyze the minimum delay timing report to verify that each flip flop has *enough* slack/margin on top of the required hold time. The mission requirements are expected to define what *enough* margin means for hold time and setup time. If there are no timing margin requirements specified, use 20% on the initial place and route and 10% for the final.

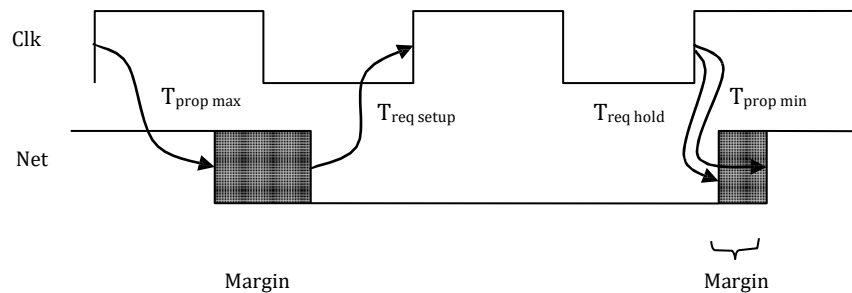


Figure 9-1 – Setup and Hold Time Margins

10 MISCELLANEOUS DESIGN GUIDELINES AND CRITERIA

10.1 NOISE IMMUNITY AND QUIET DESIGNS: *CONSIDER THE FOLLOWING ACTIONS, IF APPLICABLE TO YOUR DESIGN, TO ENSURE ADEQUATE AND ROBUST NOISE IMMUNITY*

- a. Choose differential signals, particularly for connections between cards. Newer logic devices are directly supporting differential standards. Additionally, high-speed, lower power differential devices support standards such as Low Voltage Differential Signal, LVDS, are now qualified.
- b. Serializer-Deserializer, SERDES, components/cores can cut down the number of lines, reducing noise, and hence, increase the noise immunity of the system.
- c. Use hysteresis buffers to clean up noisy inputs.
- d. Inputs that are "TTL compatible" often have specifications and real thresholds that are not TTL compatible, particularly for V_{IH} . Use conversion buffers as needed.
- e. Outputs, particularly from some CMOS families, may not be able to drive TTL loads to a valid logic '1' with sufficient noise immunity. Calculate worst-case currents and voltage output vs. worst case input thresholds. Use conversion buffers as needed.
- f. Make sure your FPGA I/O standards are compatible with external interfaces.
- g. Adequate bypass capacitance for several decades of noise frequency on the VIO pins will greatly reduce ground bounce and noise problems.

10.2 DEFENSIVE DESIGN AND DESIGNING FOR OFF-NOMINAL EVENTS: CONSIDER CREDIBLE BUT UNPLANNED EVENTS

Often many of these situations can be economically handled with a bit of thought. Here are a few sample issues to consider.

- a. **Perform limit and validity checking.** The system should respond in a reasonable fashion to unreasonable inputs. For data passed from one source to another, simple bounds checks can detect and cause appropriate action for many off-nominal conditions, such as a disconnected source, perhaps resulting in all F's being returned on a data bus. For floating point numbers, is the input in a valid format? A minimum criteria is that any credible input should not damage hardware and prevent recovery. Assume that the probability of software failure is 100%.

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

- b. **Provide fail-safe interfaces.** Analyze the performance and safety of the circuits if a wire breaks in a connector, for each wire. For power, use multiple wires such that if any one wire breaks the remaining set can carry the load (and be sure to test this redundancy). For signals, consider on-board terminations that will pull floating signals into a safe and operational state. This can also provide protection if the board or subsystem is powered with a connector not hooked up, perhaps by test error. Avoid putting signals such as power and ground on adjacent pins, as a short can take out the system.
- c. **Lockup states:** If interfacing to a device that could potentially lock up, refresh command words often.
- d. **Protecting FPGA Pins:** Avoid having FPGA outputs directly driving cables or massive capacitive loads.
- e. **Grounding FPGA Lid:** Follow manufacturer's recommendation for grounding FPGA lid. The lid may need grounding to ensure that there will be no buildup of charges and thus prevent ESD events.

10.3 DESIGNING FOR TESTABILITY: CONSIDER ADDING SIGNALS TO FACILITATE DESIGN DEVELOPMENT AND DEBUGGING

Debugging designs in the lab comes with many constraints. One of them is visibility of signals. There are many steps that a designer can take to address these constraints.

- a. **JTAG Interface:** FPGAs often come with JTAG interfaces that can be used to probe internal signals. Such interfaces can come in very handy, but often come with frequency limitations and limit the number of signals that can be viewed simultaneously. Be sure to accommodate the FPGA signals associated with the JTAG interface.
- b. **Debug Mux:** A multitude of internal signals can be brought to a multiplexor whose output connects to FPGA outputs. The multiplexor select signals can be driven by FPGA inputs or by other suitable means. This approach can be used when an FPGA's JTAG interface has limitations that pose a problem.
- c. **Aliveness Output:** A quick and easy method of checking if an FPGA is 'functional' is to generate an aliveness output signal that pulses periodically when certain critical events occur. The utility of such a signal depends greatly on how the designer chooses to generate the output.

11 RECONFIGURABLE FPGA TECHNOLOGY

11.1 CONFIGURATION MEMORY: *ENSURE INTEGRITY OF DESIGN CONFIGURATION*

Rationale: Reconfigurable FPGA's contain internal static RAM, which is used to configure the logic blocks within the FPGA to perform the required logic functions. The internal configuration memory is not radiation hardened and can be upset by radiation events. When the internal configuration memory is altered through a radiation event, the designers intended logic can be permanently changed unless the FPGA configuration memory is reloaded or scrubbed. The rate of configuration memory upsets is dependent on the FPGA technology and the radiation environment. The simplest form of protection used on FPGA configuration memory is "blind scrubbing". This consists of using a separate radiation hardened device, which contains the golden configuration and is used to continuously overwrite the configuration memory of the FPGA. Another technique is to read out the configuration memory in blocks, comparing each block to a "gold" copy and overwrite back any blocks that have been detected as corrupted by radiation. A variation on the read back technique is to compute a checksum on each block that is read back and compare the checksums to make sure the configuration blocks have not been corrupted. There are even techniques where the reconfigurable FPGA can self scrub its own configuration memory. This paragraph is not an exhaustive examination of this topic. Any engineer designing with reconfigurable FPGAs should be aware of the potential upset rate of the configuration memory and select a memory scrubbing technique by weighing the pros and cons for the particular design.

11.2 REDUNDANCY: *USE A VOTING SCHEME, AS NEEDED, TO MEET MISSION REQUIREMENTS*

Rationale: Reconfigurable FPGA's are not internally redundant by design. Some Actel FPGA's are triple redundant at the internal gate level, which is transparent to the designer, so no additional design effort is required to achieve radiation hardness. Reconfigurable FPGA's are not triple redundant at the internal gate level. Any circuitry placed in a reconfigurable FPGA is by default single string and prone to radiation upsets and transients. This lack of redundancy is typically handled by two different methods. The first method is to use three separate reconfigurable devices (each device being single string) and then vote their outputs in a radiation hardened device. This method can be effective, but is costly in terms of board space and weight and has significant complications. The second method used is to Triple Module Redundant (TMR) the design in a single reconfigurable part. The TMRing of a design can be done manually by the designer or by using design tools. The designer needs to be extremely careful to make sure that the tools do not try and reduce/prune the TMR circuits that are

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

redundant for radiation protection. With this technique, a single radiation hit in a critical circuit can still disrupt the entire FPGA and therefore upset all three TMR designs within it. This type of critical radiation hit is called a Single-event Functional Interrupt (SEFI). SEFI radiation hits happen much less frequently than other radiation upsets, as the cross area of the FPGA with the critical circuits (clock, reset) is very small. Sometimes no effort is required by the designer to add additional redundancy, even when using soft reconfigurable FPGA logic. Depending on the mission radiation environment and the frequency of upsets, it may be acceptable to a mission to have the system reboot once a day. The designer using reconfigurable FPGA's needs to review mission requirements to determine what level of redundancy is required.

11.3 EMBEDDED FUNCTIONS: *ANALYZE RADIATION HARNESS OF EMBEDDED FUNCTIONS*

Rationale: FPGA devices can contain embedded functional blocks, such as PowerPC processors, Ethernet MAC's, and Digital Signal Processing (DSP). These blocks are typically more prone to radiation upsets than the more common general purpose logic cells. The designer should not be lured into believing that the radiation numbers listed on the top page of the datasheet cover every design element that is embedded in the device. The designer should specifically verify the radiation hardness of each and every specialty embedded resource.

11.4 IN-FLIGHT RECONFIGURATION: *THINK IT THROUGH*

Rationale: In-flight reconfiguration of an FPGA should be approached cautiously. Any change in the FPGA configuration could potentially cause the device to stop functioning in-flight. For example a reconfiguration could cause the device to enter a dead state from which there is no way to recover. In-flight reconfiguration should be considered very carefully if the FPGA is performing system critical functions such attitude control or communication. In-flight reconfiguration may be better suited for scientific instruments. When planning a system to be reconfigurable, one should design in a fail safe mode, where the device can either be reloaded from a known good hard-coded boot area (PROM), or has a back door interface which allows reprogramming, even potentially from the ground, in the event of a reconfiguration error. The systems engineering team and project management should have complete buy-in to the advantages and pitfalls in any in-flight reconfiguration schemes.

If in-flight reconfiguration is going to be used then the project should be required to have ground based flat-sat set-up of their flight system and thoroughly test all new code patches and configuration updates before attempting them on orbit. Configuration bit files can be quite large for reprogrammable FPGA's so the designer should plan for file compression

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

functions that allow new code and configuration files to be uplinked from the ground in a compressed state.

If one is considering partial reconfiguration then remember that partial configuration files can be an arbitrary size, so the programming device should be able to handle configuration files of different sizes. If the FPGA device you want to perform partial reconfiguration on is normally programmed with a PROM in flight you should include partial reconfiguration regions in the configuration file programmed on the PROM. Then the user can uplink new partial reconfiguration files without having to worry about loading a new static configuration file that has partial regions first.

The use of the Internal Configuration Access Port (ICAP) in Xilinx FPGAs can be used for internal configuration scrubbing (self-scrubbing). There are two ICAP controllers in a Xilinx FPGA. The user selects the primitive they want to use by setting the "ICAP_SELECT" bit in the SelectMAP CTL0 register. Internal scrubbing requires triplicated circuitry using XTMR (Xilinx Triple Modular Redundancy) or GTMR (Global Triple Modular Redundancy) to be effective (note that the ICAP cannot be triplicated). Using the ICAP adds complexity if you want to use SelectMAP for partial reconfiguration. The user still has the option of doing internal or external partial reconfiguration. In both cases the scrubbing routine is required to be paused until completion of the partial reconfiguration. For internal, the partial bit stream would be read through general purpose I/O to the configuration control circuit, and fed to the ICAP. For external, the internal configuration circuit would first change the state of the "PERSIST" bit in the SelectMAP CTL0 register to re-enable external SelectMAP. The external controller would perform the partial reconfiguration and then change the state of the "PERSIST" bit back so that internal scrubbing can be continued.

APPENDIX A – DEFINITIONS

Term	Definition
SSO	When multiple output drivers switch simultaneously, they induce a voltage drop in the chip/package power distribution. The simultaneous switching momentarily raises the ground voltage within the device relative to the system ground. This apparent shift in the ground potential to a non-zero value is known as ground bounce.
Crosstalk	Any phenomenon by which a signal transmitted on one circuit or channel of a transmission system creates an undesired effect in another circuit or channel. Crosstalk is usually caused by undesired capacitive, inductive, or conductive coupling from one circuit, part of a circuit, or channel, to another.
Meta-stability	The ability of a digital electronic system to persist for an unbounded time in an unstable equilibrium or meta-stable state. In meta-stable states, the circuit may be unable to settle into a stable '0' or '1' logic level within the time required for proper circuit operation. As a result, the circuit can act in unpredictable ways, and may lead to a system failure.
Hysteresis	A phenomenon wherein two (or more) physical quantities bear a relationship which depends on prior history. More specifically, the response Y takes on different values for an increasing input X than for a decreasing X.
ESD Event	An event which causes a transfer of electrostatic charge between bodies at different electrostatic potentials caused by direct contact or induced by an electrostatic field.
Hamming Distance	The Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. Put another way, it measures the minimum number of substitutions required to change one string into the other, or the number of errors that transformed one string into the other.
Brown Out	A lowering of AC power voltage for some period of time. Brownouts can be very harmful to electronic equipment if sustained for long periods.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

APPENDIX B – ACRONYMS

Acronym	Description
AC	Alternating Current
ASIC	Application Specific Integrated Circuit
CAE	Computer Aided Engineering
CDR	Critical Design Review
CM	Configuration Management
CMOS	Complementary Metal Oxide Semiconductor
COTS	Commercial Off The Shelf
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DC	Direct Current
DFF	D-Flip Flop
DLL	Delay Lock Loop
DMA	Direct Memory Access
DRAM	Dynamic Random Access Memory
EDAC	Error Detection and Correction
EEPROM	Electrically Erasable Programmable Read Only Memory
EMI	Electromagnetic Interference
EPROM	Erasable Programmable Read Only Memory
ESD	Electro-Static Discharge
FET	Field Effect Transistor
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
FSW	Flight Software
GIDEP	Government Industry Data Exchange Program
GTMR	Global Triple Modular Redundancy
HDL	Hardware Description Language
IC	Integrated Circuit
ICAP	Internal Configuration Access Port
I/O	Input/Output
JEDEC	Joint Electron Device Engineering Council
JTAG	Joint Test Action Group
IEEE	Institute of Electrical and Electronics Engineers
LVDS	Low Voltage Differential Signal
MAPLD	Military and Aerospace Programmable Logic Devices
MCM	Multi – Chip Module
N/C	No Connect
OE	Output Enable
PCB	Printed Circuit Board
PCI	Peripheral Component Interface
PDR	Preliminary Design Review
PLL	Phase Lock Loop
POR	Power On Reset
P&R	Place and Route
PROM	Programmable Read Only Memory

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

Page 43 of 102

REAG	Radiation Effects and Analysis Group
ROM	Read Only Memory
SDRAM	Synchronous Dynamic Random Access Memory
SEL	Single Event Latch – Up
SERDES	Serializer-Deserializer
SEU	Single Event Upset
SRAM	Static Random Access Memory
SSO	Simultaneously Switching Output
STA	Static Timing Analysis
Tco	Clock to Out Delay Time
Tpd	Propagation Delay Time
TMR	Triple Modular Redundancy
TTL	Transistor-Transistor Logic
VHDL	Very High Speed Integrated Circuit (VHSIC) Hardware Description Language
Vih	Voltage threshold for input high
Vil	Voltage threshold for input low
WCA	Worst Case Analysis
XTMR	Xilinx Triple Modular Redundancy

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

APPENDIX C – SPECIAL PINS

This section discusses special pins on Actel and Xilinx FPGA devices and their proper configuration for flight designs. Since these manufactures have released new device types/families, the latest datasheet should be referenced to ensure that all special pins are identified and properly configured.

C.1 ACTEL RTAX

JTAG Interface:

Many modern digital microcircuits have this interface. One optional pin, which is highly desired for high-reliability designs, is the TRST*. If present, hard ground this pin since the IEEE 1149.1 specification requires a pull-up resistor inside of the part. Use of a pull-down resistor, such as what some designers use for the MODE pin, can result in the TRST* pin's input voltage being at or above the logic threshold. If the TRST* pin is not present, then the TCLK should be a free-running independent system clock with TMS held to a logic '1'. Do not use the system clock as the TCLK input, because during a malfunction, the chip's operational clock input may turn into an output and clamp the clock.

Vpump

This pin should be hard grounded.

PLL

Flight parts do not implement the PLL pins, but commercial parts do. If using a socket and commercial parts for development, PLL pins should be terminated properly. Half are no connect, NC, the other half connect to 1.5V. See datasheet for specific PLL pin numbers.

Unused CLK/HCLK

Unused CLK or HCLK pins should be connected to ground. Use a zero ohm resistor instead of a hard for flexibility.

C.2 ACTEL SX

JTAG Interface:

See section C.1 – JTAG Interface

MODE Pin:

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO.	<u>500-PG-8700.2.7B</u>
EFFECTIVE DATE:	<u>August 13, 2012</u>
EXPIRATION DATE:	<u>March 30, 2021</u>

Page 45 of 102

This pin, present on early generation of Actel devices, is required to be grounded for flight. It is recommended that this pin be grounded with a 10 kohm resistor and a hard jumper to ground in parallel, with the default setting the hard ground installed.

C.3 XILINX 5VQV

There are differences between the flight and commercial Xilinx devices. The differences require that particular pins are to be grounded. The designer should consult the latest recommendations found in the Xilinx users guide when using either the commercial or flight part in a design.

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

APPENDIX D – SYSTEM ON CHIP (SOC) FPGA DESIGN PRACTICES

When designing with very large FPGAs, the internal architecture of the FPGA design takes on elements of SOC design. This section addresses some of issues of SOC designs to consider.

D.1 DESIGN FLOW

When the SOC design contains a CPU, the design flow should allow for parallel development of both hardware and software that is more tightly coupled. Consider the following diagram.

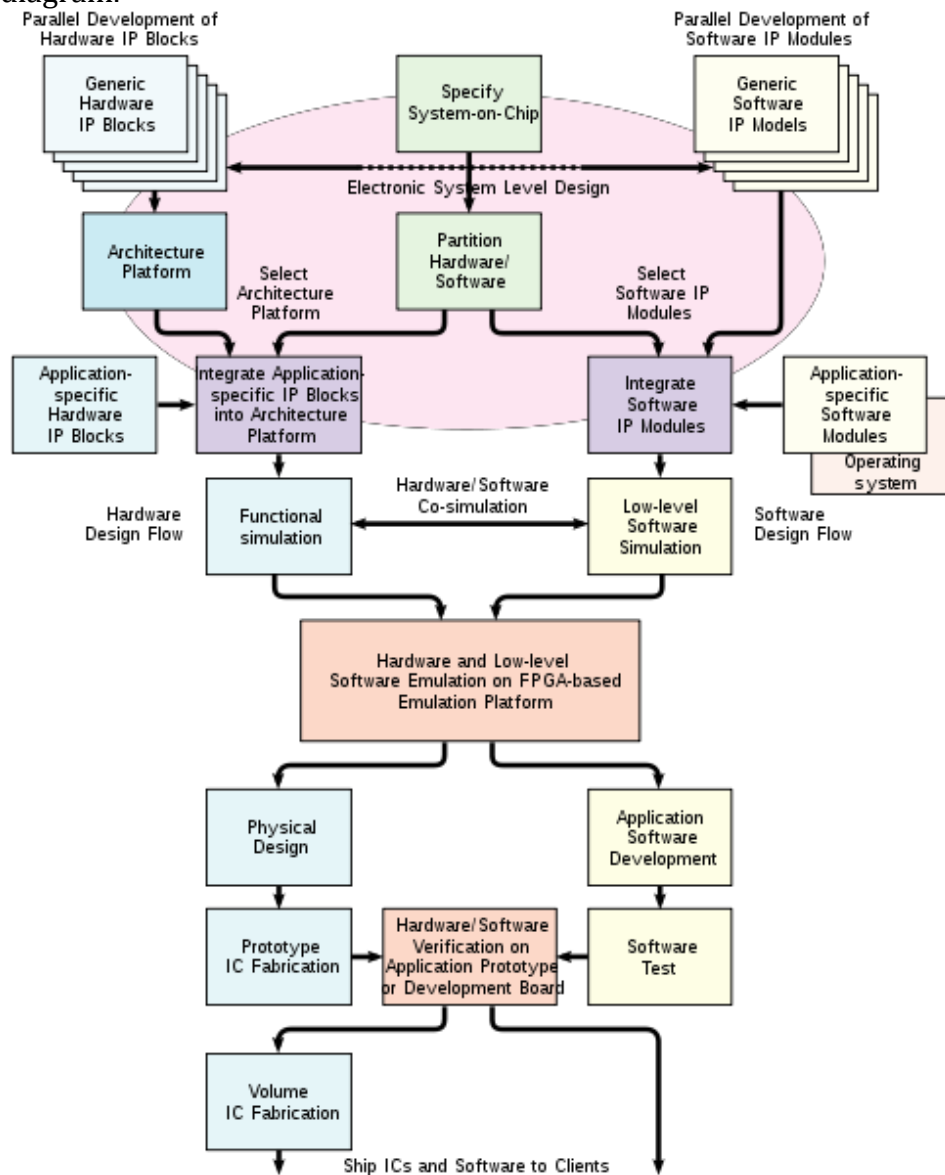


Figure 0-1 – SOC Design Flow Diagram

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

D.2 CODE REUSE

Due to the nature of larger designs, SOC design should focus heavily on code reuse. Software and hardware should be written in a way that promotes reuse by using high-level languages for software and writing behavioral HDL that is not target specific. Libraries should be developed for hardware and software functions, complete with good documentation on usage and implementation.

D.3 VERSION CONTROL

Using a version control tool becomes critical to help make the development of the design more manageable. Many version control tools are available with various feature sets. With large design involving multiple designers, use of a version control tool is strongly recommended to keep track of the various development stages of each part of the design.

D.4 IP CORES

IP cores can be purchased or custom developed. They can come to a designer in different forms ranging from Soft cores, which are synthesizable and are supplied with technology-independent HDL files, to Hard cores, which are supplied as technology-dependent modules that incorporate physical layout information (e.g. no supplied HDL files).

When designing custom IP cores, consider using a standard interconnect bus such as Wishbone to integrate the IP cores together. This minimizes the integration effort and reduces the number of integration bugs. IP cores should be designed to optimize timing between interfaces. IP core inputs and outputs should be registered immediately with combinatorial logic to allow for the best possible timing.

Each IP core should report its release number through the use of a register or fixed signal. This allows software to check and ensure its compatibility with the hardware.

Apply same best practices regarding synthesis, place and route, and timing analysis. This includes thorough review of any warnings from corresponding design tools. Review all data sheet specifications, application notes, revision notes, addendums, test benches, change notices for core modifications, and any documentation that describes the extent of the amount of vendor verification of their IP cores. Ensure that the IP core can be synthesized to incorporate, or comes supplied with, any necessary radiation-mitigation strategies such as TMR to meet radiation requirements. The information from these sources could help determine recovery modes or strategies in the event of a failure or hang-up.

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

Page 48 of 102

For designs being reused, verify that the IP's functions and specifications meet the requirements of the new intended application. Implement a verification strategy, utilizing both simulation and test, that maximizes coverage of functional or operational scenarios (e.g. test as you fly, fly as you test).

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

APPENDIX E – REVIEW OF FIELD PROGRAMMABLE GATE ARRAYS

E.1 INTRODUCTION

The review of a digital electronic circuit is simply no more and no less than proving that the design will reliably meet all requirements and specifications. That of course is the job of the designer/analyst and the reviewer's function is redundant. This section gives some insight into the process by explaining the steps to be taken in reviewing an FPGA-type digital design.

E.2 USE THE CORRECT FPGA DATA SHEETS

Device datasheets can be updated at any time, and there may be subtle differences between a manufacturer's part types, so be sure you have the correct datasheet for the part being reviewed. Check the manufacturer's web site for the latest datasheet, as with web-based specification distribution, updates can come at any time and often without notice. The governing military datasheets are also available on-line. Utilize similar care for all other devices in the system being reviewed.

E.3 COLLECT THE NECESSARY REVIEW FILES

Most FPGA designs are done in an HDL (hardware description language), such as VHDL or Verilog, and that is the assumption here. The use of standard tools from a well-known manufacturer and, preferably, from the FPGA vendor, is encouraged. For this discussion, we will assume that the Synplify synthesis tool is being used; the principles are similar for other manufacturers.

The files required for review include those that describe the system, the FPGA, and the electronics surrounding the FPGA. For each review of an FPGA, provide the FPGA review name and where it occurs within this FPGA's development flow. For a final FPGA review:

- a. A system description: Preliminary Design Review/Critical Design Review, PDR/CDR package, system specification, etc.
- b. A set of board schematics.
- c. Signal integrity analysis results.
- d. The FPGA HDL files along with other files that guide the synthesis process.
- e. Existing test benches and code coverage reports.
- f. Results of synthesis and timing analysis runs
- g. I/O Compatibility analysis.
- h. The synthesis log file (typically the .srr file for Synplify)
- i. The FPGA database file after place and route -- this is the design. (.adb for Actel FPGA's)

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

Appropriate subsets of these files should be provided for reviews prior to the final FPGA review. Before starting the review, familiarize yourself with the system operation and requirements and look over the board schematics to get a feel for the design. Make note of the FPGAs place in the overall system and its criticality.

- j. Is the correct operation of the device safety critical?
- k. Does the device control any safety critical functions such as pyrotechnic initiation circuits, thrusters, or high-voltage power supplies in test and flight?
- l. Does the device issue spacecraft critical or mission critical commands (one-time or irreversible functions), set latching relays, deployment, maneuvers, or otherwise perform configuration functions?
- m. How many different power sources feed the board, and how are they sequenced? Consider both power-up and power-down sequences.
- n. How is the circuitry reset?
- o. Is it critical that the FPGAs functions be performed correctly the first time tried, or is there opportunity for retries?
- p. Does the FPGA receive asynchronous data or commands or perform processing on asynchronous events?
- q. How many clock sources are there, and what are their frequencies, duty cycles and phase relationships? A clock tree should be provided by the designer or it can be generated as part of the review process.
- r. The printed circuit board artwork should be readily available, as needed, to support signal and power integrity analysis.

E.4 PERFORMING THE REVIEW

There are several levels of detail to which a review can be performed, and ideally every design receives the most detailed review, in which correct FPGA usage and the overall electronic and logic design are considered and proven correct. This isn't always possible because of time and budget limitations, but the steps in reviewing a design are the same regardless of the ultimate review level, the difference being how many steps in the review are accomplished. Often, a "spot check" or "scan" of a design is all that may be performed, because of the aforementioned limitations.

One critical thing to remember is that the HDL is not the design, but simply the designer's description of the desired logic. Running RTL pre-synthesis simulations and test benches is insufficient proof of a design's correctness. Running gate-level netlist simulations with back-annotated timing offers a better assurance of design correctness as it accounts for the synthesizers output and is a closer representation of reality.

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

The design is the output of the back end place and route function and the hardware is the physical chip, after configuration.

The fidelity of the actual design to the intended design depends on the quality of the synthesizer, which is unknowable, and the ability of the designer to

- a. Write synthesizable HDL
- b. Understand the synthesis process and tool employed
- c. Control the synthesis process
- d. Verify that the synthesis process produced what was intended (i.e., FSM encoding, width and depth of RAMs, ROMs, FIFOs, etc)
- e. Correctly guide the back-end place and route tools. These tools may also alter the intended design through logic replication, combining, elimination of logic functions, setting I/O module parameters such as I/O thresholds, output slew rates, the presence or absence of clamping diodes, cold-spares functionality, etc. While not as abstract and complex as logic synthesizers, failure to understand the processes in the back-end design process has been seen to cause design errors.

One of the limitations of FPGA design is that the static timing analysis tool is primarily designed to analyze fully synchronous logic that uses only one clock edge. Dynamic logic simulators are insufficient for proving design correctness. Asynchronous design techniques are extremely difficult to analyze with the available tools, are error-prone, and are thus discouraged where these techniques are not required. Therefore, an important part of the review process is ferreting out design techniques that are error-prone and should not be used in an FPGA.

E.4.1 REVIEWING THE BOARD SCHEMATICS

The FPGA application cannot be properly reviewed without knowing its electrical environment. The following list, which is not exhaustive, shows several classes of issues to examine.

- a. Of primary importance are that the special pins, e.g., TRST*, are treated properly. Review the FPGA specification for the requirements of unused clock pins and other special pins such as device configuration or programming pins and verify they have been properly terminated on the board.
- b. Look for unusual loads (e.g., high capacitance or non-logic loads).
- c. Look for unusual sources (e.g., questionable logic levels, excessive transition times, mixing of logic families, devices powered by different supplies, etc.).

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

- d. Note circuitry that may be powered up or down independently of the FPGA and the cold- sparing capability of each device.
- e. Determine the number of simultaneously switching outputs and their distribution around the FPGAs I/O ring.
- f. Determine the length of Printed Circuit Board, PCB, traces and how the signals are terminated, ensuring that overshoot and undershoot specifications are met. In particular, carefully examine all signals that leave the PCB.
- g. Ensure that the manufacturer's recommendation for bypass capacitors and power/ground planes are being followed. Past reviews have found boards with inadequate capacitance and routing, including one case where zero bypass capacitors were used and another where placement of the capacitors led to poor performance.

E.4.2 READING THE SYNTHESIZER OUTPUT LOG FILE

The .srr file, the output log file written by Synplify as it reads though and processes the HDL files, can tell the reviewer quite a bit about the design.

- a. The first part of the .srr file shows two passes made through the VHDL. In the first, Synplify finds the VHDL modules and state machines, and in the second the state machines are revisited and reset logic is created for any which the designer gave the "safe" attribute to deal with illegal states.
 - 1) Note the state machine names found in the first pass, then note in the second pass any for which reset logic is not created. All state machines should have their illegal states handled, because illegal states may cause inappropriate behavior. Determine the requirements for each state machine and ensure they are handled in the logic, by periodic local resets, or by a POR or other reset command. It is often found that designers concentrate on the correct functioning of the circuits and not on the effects of "glitches" or recovering from them. Glitches may result from, for example, power transients, radiation, or ESD.
 - 2) Having the reset logic created, however, does not mean the FPGA will perform the correct functions if illegal states are entered. One subtlety of Synplify 's synthesizer- generated reset logic is that under some conditions a half-edge flip-flop (e.g., a falling edge flip-flop in a rising edge design) is used to generate the reset. The designer generally doesn't recognize this because Synplify doesn't point it out, and its timing isn't analyzed. This timing analysis relies on the duty cycle of the state machine's clock, which may vary considerably, and not the

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

period, which is generally quite accurate, as crystal controlled clock oscillators are the norm.

- b. Replicated flip-flops in synchronizers of asynchronous signals are not permitted. In general, replicated flip-flops could cause inappropriate operation if transient events cause logically equivalent flip-flops to take on different values, and should be discouraged. If replicated flip-flops are employed in the design, thoroughly analyze and document the acceptability of each instance. This task is both labor intensive and, hence, error-prone so it should be performed after each synthesizer run.
- c. Compare the flip-flops used with the FPGA manufacturer's macro library to see if any of the following types are used:
 - 1) Flip-flops without sets or clears, indicating circuitry that will not be reset on POR or reset command;
 - 2) Flip-flops with both sets and clears, indicating possible asynchronous design techniques (the absence of set/clear flip-flops does not indicate the absence of asynchronous design techniques);
 - 3) Latches, for which the timing has to be checked by hand;
 - 4) Opposite edge flip-flops (e.g., falling edge flip-flops in a predominantly rising edge design) that could place constraints on clock symmetry and be more difficult to analyze with the timing verifier. Some opposite edge flip-flops could result from the use of the "safe" attribute, noted above, and the designer is often unaware of their presence.

While the above are not necessarily design errors, they indicate items that should be checked.

Some HDL coding errors can result in unexpected latches or set/clear flip-flops.

- d. The logic type list discussed above will also note which of the clock resources were used, and give statements such as "clock found" or "clock inferred."
 - 1) If local clocks were used (i.e., clocks that do not use the global clock resources), they will show up here, e.g., when there are 4 clocks in an FPGA with 3 clock drivers. Local clocks potentially have much higher skew than is acceptable and their use should not be allowed for clocking sequentially adjacent flip-flops that are triggered on the same edge.
 - 2) If the routed clocks (CLKA/B) are used in RT54SX-S, RTSX-SU, or A54SX-A devices, verify circuitry incorporates appropriate skew tolerant design techniques.
 - 3) A table will show which clock edges have logic between them, e.g., from the rising

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

edge to falling edge of HCLK, or between edges of different clocks. Carefully scrutinize logic crossing clock domains, and the symmetry requirements of clocks of which both edges are used.

The remainder of the .srr file contains timing analysis information that is calculated before place and route, and is thus of dubious value. The correct timing analysis will be shown by Static Timing Analysis, STA, when the .adb file is accessed. STA will not analyze half-edge clock flip-flops or any asynchronous techniques on its own. Such instances will have to be analyzed by hand, in conjunction with STA.

When reading the .srr file, carefully note any warnings given. Designs can synthesize even when warnings are given. Disposition each warning after each synthesis run.

E.4.3 BACK-END TOOLS: USING THE FPGA'S VENDOR SPECIFIC PLACE AND ROUTE FILE

The vendor specific design file contains the details of the design, including the actual netlist, timing analysis, pin information, etc. For the purposes of this discussion, we will refer to Actel's Database file, .adb file. The Actel Designer place and route tool includes a netlist viewer to allow the reviewer to see the actual FPGA design (rather than HDL) in a schematic representation, although it is an awkward view (as most schematic generators produce).

- a. Check the temperature, voltage, and radiation settings for which the timing analysis was done. These should be the full military ranges for temperature and voltage, and whatever the program radiation requirement is. Justify use of reduced temperature or voltage range. Note that the tools assume that temperature is the device junction temperature and not the case temperature or the temperature of the board's thermal control surfaces. If a slower speed grade FPGA will be used, check minimum timing using the fastest speed grade, in case the FPGA vendor delivers a faster FPGA (from their current inventory). See section 10.4. Check maximum timing against purchased speed grade.
- b. Run STA to see how much timing margin there is. Even when the full military ranges are used, as above, there should be some margin for aging, inaccuracy in calculation, etc. A \pm 10% margin for propagation delay is appropriate.
- c. Run Pin Edit to determine what I/O options were chosen. Verify that the choices were appropriate by comparing them with the inputs and outputs seen on the schematics.

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

- d. Open the Netlist Viewer and view the schematic to resolve the issues found in section 0, above, especially to understand the unusual flip-flop usages found in section 0 (c)
- e. In the Netlist viewer, scan through the schematic looking for flip-flops with gated sets or clears, and to assure that all the settable and resettable flip-flops connect to a valid reset and are not involved in asynchronous design techniques. Starting at the reset or POR input, the reset lines can be highlighted and followed through all the pages.
- f. In the Netlist viewer, ensure that all mission-critical and safety-critical circuits are implemented correctly. HDL synthesizers have been seen to implement poor circuits such as static hazards in clock generation circuitry.

E.5 REVIEW THE POR AND RESET CIRCUITRY AND POWER-UP CONDITIONS

The ideal power-on-reset (POR) is asserted as soon as the power supplies are turned on and remains asserted until the voltages reach valid operating levels. Some factors may require that the POR be asserted beyond that point in the power-up cycle:

- a. If there is an oscillator on the board, it should remain asserted until the oscillator has begun proper operation, which could be as long as tens of milliseconds after its power supply has reached a valid level.
- b. If there are flip-flops that require the oscillator to be running in order to be reset, ensure the reset is kept asserted until these flip-flops are reset.

Beyond this, the criticality of the FPGA and its potential to cause damage to the spacecraft or an instrument, as discussed in section 2 above, may require careful scrutiny of its reset and power up/down conditions. During some portion of the power up time, the FPGA is not a circuit but simply a collection of unconnected gates, and transients may appear on its outputs.

External circuitry capable of causing damage or undesirable operation during power up and down, or during brown-outs, should be carefully reviewed to verify safe operation during these periods. For example, circuits constituting an arm and fire mechanism should not have both the arm signal and the fire signal originate in FPGAs that are powering up or down simultaneously. For outputs used as clocks to other logic and/or externally back to this FPGA, explain how these meet board-level interface requirements when the FPGA enters, is in, and exits each type of reset. For outputs used as reset to other logic and/or externally back to this FPGA (a frequent cause of problems), explain how these meet board-level interface requirements when the FPGA enters, is in, and exits each type of reset; include any delay or stretching of these outputs. External components should also

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT <http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

be reviewed to determine whether special reset requirements exist. Notable in this class are EEPROMs, which require protection during power-on, power-off, and other transient conditions such as brown-outs.

E.6 REVIEW THE PLAN FOR FUNCTIONAL VERIFICATION

It is very difficult, if not impossible, to review the functionality of the FPGA design during the review. Thus, it is very important to review the verification plan, which should explicitly state how the FPGA design's functionality is verified through simulation and testing. The plan should describe the test bench and include a list or table/matrix showing each test, its description, and its traceability to requirements or specifications. During the review, emphasis should be placed on determining the sufficiency of the verification plan and ensuring that proper verification practices are being employed. Recommendations for simulations include:

- a. Tests should be automated and self-checking, reporting a pass/fail flag upon completion. Do NOT rely on visual waveform analysis.
- b. Simulation models of external interfaces should employ ASSERT statements to validate proper signal timing and adherence to protocols.
- c. Exercise boundary conditions such as FIFO full/empty scenarios as well as buffer overflow/underflow conditions.
- d. Exercise asynchronous interfaces properly by using asynchronous clocks and allowing enough simulation time to elapse to verify the entire range of skew between interfaces on different clock domains.

Use code coverage to analyze the sufficiency of the test bench and find holes where functionality is not being tested.

E.7 REFERENCES, NOTES, AND RELATED DOCUMENTS

1. [Suggestions for VHDL Design Presentation](#)
2. "A Designer's Checklist," 2004 MAPLD International Conference. [design_checklist.pdf](#)
3. [OLD News #13: Minimum Delays and Clock Skew in SX-A and SX-S FPGAs](#)
4. [Index of DSCC Mil Specs & Drawings](#)
5. "PCB Layout Issues," Design Seminar on Actel SX-A and RTSX-S Programmed Antifuses, Tuesday, April 13, 2004. [g_pcb_layout_issues.ppt](#)
6. "Case Study: Simultaneous Switching Outputs," Design Seminar on Actel SX-A and RTSX-S Programmed Antifuses, Tuesday, April 13, 2004
7. "Drive Strength." Design Seminar on Actel SX-A and RTSX-S Programmed Antifuses, Tuesday, April 13, 2004
8. "Sequential Circuit Design for Spaceborne and Critical Electronics," Rod Barto, 2000 MAPLD

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

Page 57 of 102

International Conference.

9. "Is It Safe?" from "Programmable Logic Applications Notes, EEE Links," August 1999.
[EEE_Links_Aug99.PDF](#)
10. "[When Should You and When Should You Not Use VHDL?](#)" 2004 MAPLD International Conference
11. "[Some Characteristics of Crystal Clock Oscillators During the Turn-On Transient](#)"
12. [Appendix F of the WIRE Mishap Investigation Board Report, June 8, 1999.](#)
13. "[SX-S Output Transients](#)"
14. "[Act 3 Output Transients](#)"

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

APPENDIX F – REFERENCES

Reference documents can be found on the Internet with web search using the blue text shown in the reference.

F.1 SPECIAL PINS

- a. It is critical to ensure that all pins are properly terminated. Some will affect the functionality of the chip and these may or may not be caught in test. Some unterminated pins have been shown to have parametric and perhaps long-term reliability effects ([Figure F-1](#)).

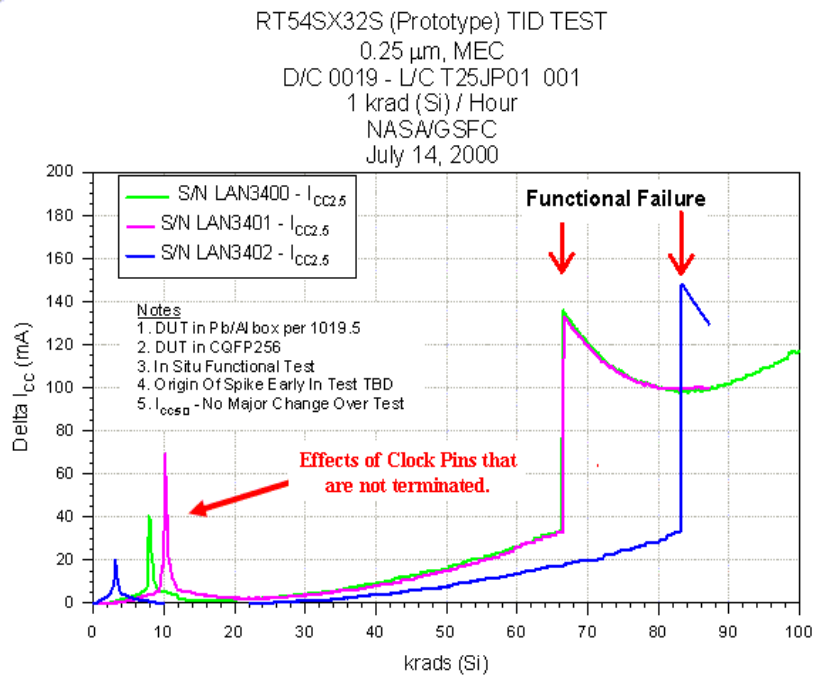


Figure F-1 – TID Effects of Improper Clock Termination

- b. ["Special Pins" from "Advanced Design: Designing for Reliability"](#), presented at the 2001 MAPLD International Conference, Laurel, MD, September 2001.
- c. ["TRST* and the IEEE JTAG 1149.1 Interface,"](#) OLD News #7, January 2003.
- d. ["Signal Terminations for the Silicon Explorer"](#)

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

- e. "[Use of SX Series Devices and IEEE 1149.1 JTAG Circuitry.](#)" This white paper reviews basic 1149.1 principles, radiation results on SX Series devices, and finishes with mitigation techniques and design considerations.
- f. "[GROUND THE MODE PIN NOW!!!!!!!!!!!!!!!!!!!!!!!](#)," Termination of MODE Pins in Actel Field Programmable Gate Arrays.

F.2 INPUT/OUTPUT (I/O)

- a. "[SX-A/RT54SX-S SSO Preliminary Results](#)," October 2, 2002, Actel Corp. [sso-10-1-02_actel.pdf](#)
- b. "[Input Transition Times for SX-S FPGAs](#)," OLD News #3, June 24, 2002.
- c. "Input Transition Times," Section 6 of Programmable Logic Application Notes: November, 2000. [EEE_Links_Nov00.pdf](#).
- d. "[Supply-Voltage Migration, 5V to 3.3V.](#)" Covers background on processing technologies with implications for supply voltages, distributing multiple supply voltages on a PCB, interfacing between devices operated at different supply voltages, supply voltage sequencing considerations, and migrating designs. [xapp080.pdf](#).
- e. "Input Stages," presented at the 2001 MAPLD International Conference, Laurel, MD. [C_Input_Stages.ppt](#).
- f. Slow transition times on the clock input of an RH1020 (Figure 2-1) shows oscillation, although the rise time is less than the specified 500 ns. For this series of tests, the conditions were room temperature and $V_{CC} = 5.0$ V. Oscillations detected consistently at $t_R = 360$ ns and sporadic output pulses at $t_R = 300$ ns. Note that the transition time performance of the input stages were not symmetric with oscillations detected consistently at $t_F = 1.5$ μ s and sporadic output pulses observed at $t_F = 1.0$ μ s.
- g. With the input held at the threshold level, representing the case of a floating input, an RH1020 input stage breaks into full oscillation (Figure F-2.2), as seen on the output of the device. For some input stages, the oscillation is not easily seen on the input pin but will propagate within the device.

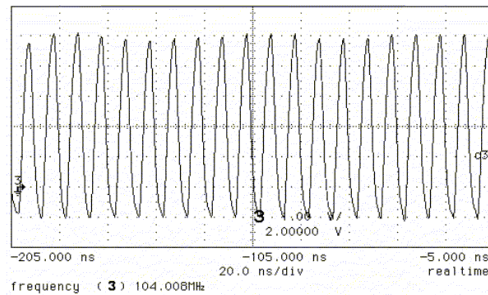
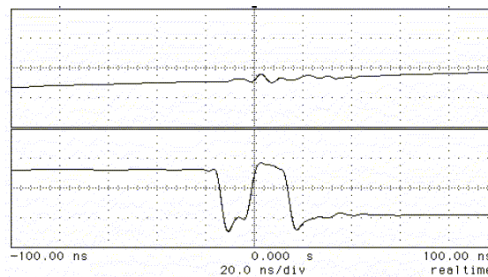


Figure F-2.1 – Metastability Due To Floating Input

- h. This example shows the slow rising input resulting in multiple clocking (Figure 2-1) of an RT54SX16 input. A zoomed in view, Figure F-2.2.



RT54SX16 output (bottom trace) with a slow rising input (top trace) which clocks a divide by two counter resulting in a "glitch." The clock input was provided by an HP8110A pulse generator.

Figure F-2.2 – Multiple Clocking Due To Slow Rising Input (Zoomed In)

- i. "Signals Into Unpowered CMOS" provides additional discussion.
- j. "A radiation-hardened cold sparing input/output buffer manufactured on a commercial process line," Benedetto, J.M. Jordan, A., Radiation Effects Data Workshop, 1999, Location: Norfolk, VA. pp. 87-91. *Abstract:* The radiation hardness of a cold sparing buffer manufactured on a commercial process line is demonstrated. The buffer is shown to be resistant to total dose ionizing radiation and immune ($>128 \text{ MeV-cm}^2/\text{mg}$) to effects from heavy ions such as single event upset (SEU) and single event latch-up (SEL)
- k. The specifications for inputs should be carefully read as not all device or MCM inputs are truly TTL compatible. See "TTL Compatible" Inputs in CMOS Devices.

- l. The note "[Signal Integrity: IBM Luna C DRAM](#)" gives examples of the requirements for signal integrity, noise levels, and included not only logic signals but the power line. Note that for this device, non-monotonic switching on the control lines may result in unpredictable results.
- m. "Designing For Signal and Power Integrity in FPGA Systems," Mark Alexander, 2002 MAPLD International Conference, Laurel, MD, September 2002. [b5_alexander_p.pdf](#)
- n. "[Drive Strength of Actel FPGAs](#)," *Introduction*: Many modern CMOS digital microcircuits have very strong drivers; the device characteristics have changed over the years. Another change is the widespread use of HDL synthesis for logic generation and simulators for logic simulation. These simulators do not replace the need to perform proper electrical engineering of spaceborne digital electronics, in particular signal and power integrity.
- o. "[IBIS Models and Simulation](#)," presented at "Design Seminar on Actel SX-A and RTSX-S Programmed Antifuses," Tuesday, April 13, 2004, NASA Goddard Space Flight Center. Review of IBIS and tools along with flight design samples used as case studies. [i_ibis_models_si_rev_a.ppt](#)
- p. "[The Effects of Slew Rate on SX-S Series FPGAs](#)," July 18, 2004.
- q. "Simultaneously Switching Noise and Signal Integrity," June 2006, Actel Corp. [SSN_AN.pdf](#).

F.3 CLOCKS

- a. "[Clock Skew](#)" from "[Logic Design: Clocking, Timing Analysis, and State Machine Design](#)," presented at the 2002 MAPLD International Conference, Laurel, MD, September 2002.
- b. "[Clock Timing and Skew: Real Devices](#)" from "[Logic Design: Clocking, Timing Analysis, and State Machine Design](#)," presented at the 2002 MAPLD International Conference, Laurel, MD, September 2002.
- c. [Skew-Tolerant Circuit Design](#), David Harris, Harvey Mudd College © 2001 by Academic Press ISBN 1-55860-636-X.
- d. Start times of oscillators may be a function of power supply rise time and may not start up clean. Example with a 50 ms power supply rise time (Figure F-3.1). For the same oscillator, this is a summary of performance over a range of rise times (Figure F-3.2).

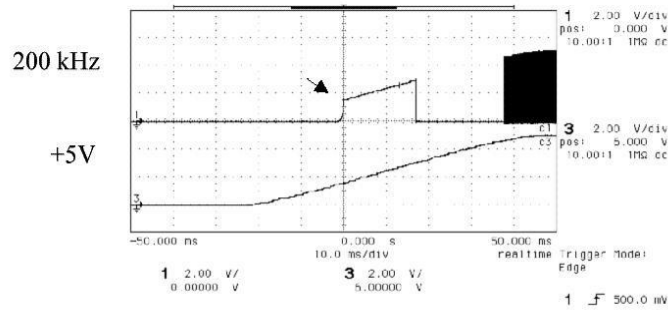


Figure 4-2 Start time characteristics of a flight spare oscillator with a power supply rise time of 50 msec. The horizontal scale is 10 msec per division. The start time from the application of power is approximately 58 msec.

Figure F-3.1 Oscillator Start Time

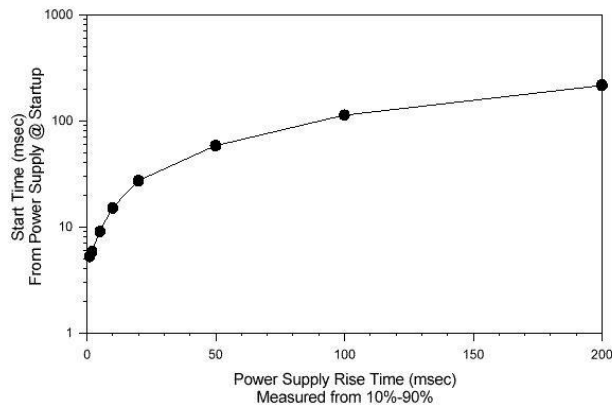


Figure 4-4 Summary of start time characteristics of a flight spare oscillator at 10°C. A logarithmic scale is used for the Y-Axis to facilitate reading of actual values for relatively small rise times.

Figure F-3.2 Oscillator Start Time vs Power Supply Rise Time

- e. ["Some Characteristics of Crystal Clock Oscillators During the Turn-On Transient."](#)
 This application note discusses and shows what the output of an oscillator may be during the turn-on transient. Examples shows include runt pulses of various sizes and polarities.
- f. ["Startup Transient,"](#) from Advanced Design: Designing for Reliability, 2001 MAPLD International Conference, Laurel, MD, September 10, 2001. D_StartupTransient.ppt
- g. [Timing Analysis of Asynchronous Signals](#)

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

- h. Discussion of [Metastable States. 33_metastablestates.ppt](#)
- i. "[Clock Skew and Short Paths Timing](#)," Actel Corporation, March 2004.

F.4 FINITE STATE MACHINES

- a. SETs first observed by NASA GSFC Radiation Effects and Analysis Group (REAG), Code 561. Heavy ion testing performed by REAG, Code 561.
- b. For visibility into the operation of the system, debug, and test, bring FSM state flip-flops to spare I/Os and test points.
- c. "[Sequential Circuit Design for Spaceborne and Critical Electronics](#)," R. Barto, presented at the 2000 MAPLD International Conference.
- d. "[Logic Design: Clocking, Timing Analysis, Finite State Machines, and Verification](#)," Presented at the 2002 MAPLD International Conference, Laurel, MD, September 9, 2002.
- e. [Logic Design: Flip-Flop Replication](#). This application note gives an introduction to the topic and examples. Cases examined are VHDL synthesis, netlist translation, and backend place and routing.
- f. "[XC4000XL/Spartan PAR - Router duplicates registers for use as output-to-output route- thrus](#)," Xilinx answers database #3813.
- g. "[Asynchronous & Synchronous Reset Design Techniques - Part Deux](#)"
- h. "[Startup Transient](#)," from Advanced Design: Designing for Reliability, 2001 MAPLD International Conference, Laurel, MD, September 10, 2001. D_StartupTransient.ppt
- i. [Logic Design: Analysis of POR Circuit Topologies](#)
- j. Discussion of [Metastable States. 33_metastablestates.ppt](#)
- k. [Timing Analysis of Asynchronous Signals](#)

F.5 RESETS

- a. Start times of oscillators may be a function of power supply rise time and may not start up clean. Example with a 50 ms power supply rise time (Figure 3 1). For the same oscillator, this is a summary of performance over a range of rise times (Figure 3 2).
- b. "[Some Characteristics of Crystal Clock Oscillators During the Turn-On Transient](#)." This application note discusses and shows what the output of an oscillator may be during the turn-on transient. Examples shows include runt pulses of various sizes and polarities.
- c. "[Asynchronous & Synchronous Reset Design Techniques - Part Deux](#)"
- d. [Timing Analysis of Asynchronous Signals](#)
- e. [Logic Design: Analysis of POR Circuit Topologies](#)
- f. "[Board Level Considerations](#)" Actel Application Note [AC276](#)

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

F.6 HAZARD ANALYSIS

- a. "Hazards," from Advanced Design: Designing for Reliability, 2001 MAPLD International Conference, Laurel, MD, September 10, 2001. E_Hazards.ppt.
- b. [Analysis and Design of Digital Circuits and Computer Systems](#), Paul M. Chirlian, Stevens Institute of Technology, ©1976. pp. 261-264.

F.7 POWER

- a. [Designers Must Take Care When Powering High-Speed CMOS](#), Robert M. Hanrahan, ED Online ID #5415, Electronic Design, August 4, 2003.
- b. ["RT54SX32S High ICCI Inrush Current,"](#) OLD News #10, May 16, 2003.
- c. ["Analysis of Printed Circuit Board Artwork: Bypassing,"](#) Rod Barto, Office of Logic Design, March 2004.
- d. ["PCB Layout Issues,"](#) presented at "Design Seminar on Actel SX-A and RTSX-S Programmed Antifuses," Tuesday, April 13, 2004, NASA Goddard Space Flight Center. Discusses layout issues for bypass capacitors, vias, and power and ground planes, in the context of "before and after" of a flight printed circuit board.
- g. [pcb_layout_issues.ppt](#)
- e. ["Board-Level Considerations for Power – Up and Power – Down of RTAX – S/SL FPGAs,"](#) Actel Corporation Application Note AC344, May 2010.

F.8 INTERFACING TO NON-VOLATILE MEMORIES (EEPROM, FLASH, ETC)

- a. ["Summary of Recent EEPROM Failures,"](#) OLD News #12, July 3, 2003.
- b. ["Maxwell EEPROM Bit and Page Failure Investigation Report,"](#) Y. Chen, June 3, 2003.
- c. ["EEPROM Bit and Page Failure Investigation,"](#) Yuan Chen, Rich Kemski, Duc Nguyen, Frank Stott, Ken Erickson, Leif Scheick, Richard Bennett, and Tien Nguyen, 2003 MAPLD International Conference, Washington, D.C., September 9-11, 2003.
- d. [Reliability Report: HN58C1001 Series CMOS 1M EEPROM](#)
- e. [EEPROM Evaluation and Reliability Analysis](#), Aerospace Report No. TOR-2000(3000)-01. June 28, 2000.
- f. ["Usage of EEPROM in Digital Designs,"](#) Saab Ericsson Space, D-G-NOT-00385-SE, 2004 g. ["Design of Memory Systems for Spaceborne Computers,"](#) 2004 MAPLD International Conference, Washington D.C., September 8-10, 2004.
- h. ["An Application Engineer's View,"](#) 2004 MAPLD International Conference, Washington D.C., September 8-10, 2004.
- i. ["Observations in Characterizing a Commercial MNOS EEPROM for Space,"](#) 2004 MAPLD International Conference, Washington D.C., September 8-10, 2004.

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

- j. "[Maintaining Data Integrity in EEPROMs](#)," 2004 MAPLD International Conference, Washington D.C., September 8-10, 2004.

F.9 TIMING ANALYSIS

- a. [Digital Timing Analysis Tools and Techniques](#)
- b. [Root-Sum-Square \(RSS\) Calculations of Digital Timing Delays](#)
- c. [NSCAT Digital Subsystem Design Documentation and Analyses](#)
- d. [Galileo AACSE: Worst Case Analyses \(WCA\) Description and Criteria](#)
- e. "[Propagation Delay and Aging](#)," *OLD News* #4, August 3, 2002.
- f. "[Minimum Delays and Clock Skew in SX-A and SX-S FPGAs](#)," *OLD News* #13, July 15, 2003.
- g. "[Logic Design: Clocking, Timing Analysis, Finite State Machines, and Verification](#)," Presented at the 2002 MAPLD International Conference, Laurel, MD, September 9, 2002.
- h. [Timing Analysis of Asynchronous Signals](#)
- i. Discussion of [Metastable States](#). [33_metastablestates.ppt](#)
- j. Signal integrity of the clocks is important; not only for ensuring that the propagation delays are calculated correctly, but that the devices function properly. Often the clock inputs have more stringent requirements than typical signals, with fast transition times specified as well as lower values for V_{IL} and higher values for V_{IH} .
- k. "[RT54SX72S: Propagation Delay vs. Life](#)," June 6, 2004.

F.10 MISCELLANEOUS DESIGN GUIDELINES AND CRITERIA

- a. [OLD News #11 Interface Components and ESD](#), May 28, 2003. ESD and proper device handling practices are nothing new and normally would not warrant an OLD News posting. Indeed, ESD practice and component tolerance have improved so much over the years that ESD damage hasn't been a major source of problems for quite a while, for regular digital integrated circuits and interface components. However, there have been some recent surprises. ...
- b. The specifications for inputs should be carefully read as not all device or Multi-Chip Module, MCM, inputs are truly TTL compatible. See "[TTL Compatible" Inputs in CMOS Devices](#)."

- c. "Case Study: Simultaneous Switching Outputs," presented at "Design Seminar on Actel SX-A and RTSX-S Programmed Antifuses," Tuesday, April 13, 2004, NASA Goddard Space Flight Center. Presents 4 cases of "staggering" I/O switching, trading off lower di/dt for increased data transfer time and analyzes software performance and the effect of module placement.

F.11 RECONFIGURABLE FPGA TECHNOLOGY

- a. "Blind Scrubbing" technique invented by NASA GSFC REAG, Code 561.

APPENDIX G – FPGA DESIGN CYCLE CHECKLIST FOR DESIGNERS


This section contains a sample checklist for FPGA designers to complete before a review. It is suggested that the designer reads through this checklist before starting the FPGA design, and fills it out during the course of designing the FPGA.

The recommendation of this document is for each NASA project to take the checklist provided here as a starting point, add project-specific items to **Table 26** and manage the resulting checklist document within the project.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

The person starting the checklist for this review should edit the heading and fill in Table 1 except for the end date. The lead reviewer should enter the end date.

Table 1 - FPGA Review Context in FPGA Development

Start Date ¹	
End Date ¹	
FPGA review name and where it occurs within this FPGA's development flow	
This FPGA's development flow, as a picture, embedded diagram, reference to FPGA development plan, etc.	 Example Flight FPGA Dev Flow.ppt
1. FPGA reviews are likely worked intermittently between the start and end dates	

The lead reviewer should record the status and recommendation quantities for this FPGA review in Table 2.

Table 2 - FPGA Review Status

Qty	Color	Status
	White	Not reviewed or not completely reviewed Due to low risk, information not readily available and/or time limitations
	Purple	Deferred Until an appropriate review later in the FPGA's development (not applicable for the final FPGA review)
	Yellow	Open Concerns remain based on information provided or assumptions, inspections and/or analyses performed
	Green	Acceptable No concerns or issues found or remain
	Red	Unacceptable Implementation is faulty or too risky
	Turquoise	Recommendations Changes that aren't necessary but would be "better" and should be considered if the FPGA is otherwise revised or re-

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

		used
--	--	------

The lead reviewer should record the key personnel for this FPGA review in Table 3.

Table 3 - Key FPGA Review Personnel (edit/add/delete rows and titles as necessary)

Role	Name	Affiliation	Email	Phone(s)
Responsible FPGA Engineer				
Responsible Board Engineer				
Verification Engineer				
Lead Reviewer				
Reviewer				
Reviewer				

This FPGA checklist:

³⁵₁₇ Is based on

- NASA GSFC 500-PG-8700.2.7 Procedures and Guidelines for Design of Space Flight Field Programmable Gate Arrays
 - Note: Each section of this document contains a subsection named “Recommendations and Tips”. The information in these subsections is not necessarily covered in the check-list, however, it is recommended that designers and reviewers read through them.
- NASA GSFC 500-PG-8700.2.8 Field Programmable Gate Array (FPGA) Development Methodology
- Lessons learned from several GSFC and other programs within and outside of NASA

³⁵₁₇ Is to be used as part of a thorough design analysis, not as or in lieu of one

³⁵₁₇ Looks for questionable/problematic digital circuits in space applications

³⁵₁₇ Does not specifically assess whether the design meets its requirements

- Requirements compliance should be assessed via other reviews

³⁵₁₇ Provides specific questions for some types of FPGAs

FPGA designs should not necessarily “comply” with this checklist’s items. Some checklist items reflect design preferences (such as synchronous design methods) but deviations may be acceptable if sufficient justification is provided.

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> **TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.**

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

Page 68 of 102

FPGA designer(s), verifier(s), tester(s) and/or reviewer(s) should record comments, actions, responses and/or recommendations for each checklist item using track-changes, to log the author and date. Comments, actions and responses may be brief text, lists, tables, figures, etc, and/or references to specific sections of other documents, including references to other comments, actions, responses or recommendations in this checklist (as some items may yield similar answers).

FPGA designer(s), verifier(s), tester(s) and/or reviewer(s) should highlight in turquoise any recommendations. Recommendations should include rationales.

For better readability, table row formatting is set to not allow rows to break across pages. For Comments / Actions / Responses / Recommendations that span more than one page, re-format the table row to allow it to break across pages (Word: table properties – row tab – check “Allow row to break across pages” – click OK).

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> **TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.**

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

Table 4 - FPGA Information

4 FPGA INFORMATION		Comments / Actions / Responses / Recommendations
PG Ref: E.3		
4.1	List the FPGA design's name.	
4.2	List the FPGA manufacturer and part information (family, speed grade, package, etc.) for each phase of this FPGA's development (DU, ETU, flight, etc.).	
4.3	List reason(s) for selecting this FPGA.	
4.4	List the FPGA design's database file name(s).	
4.5	List the FPGA design's unique identifier(s) (ex: fuse checksum, silicon signature).	
4.6	List the board name(s) using this FPGA design and quantity per board applicable to this review.	
4.7	List the subsystem/box name(s) using this FPGA applicable to this review.	
4.8	List the program(s) using this FPGA applicable to this review	
4.9	List the FPGA design's requirements document(s).	
4.10	List the FPGA design entry guidelines/constraints (ex: VHDL coding styles/standards)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

4 FPGA INFORMATION		Comments / Actions / Responses / Recommendations
PG Ref: E.3		
4.11	List the FPGA design's specification/architecture document(s) that defines the FPGA functions, capabilities implementation, etc.	
4.12	List the FPGA design's verification document(s) (ex: requirements compliance matrices, test plan(s), test procedure(s)).	
4.13	List presentations, spreadsheets and/or other documents pertinent to this FPGA design (ex: concept, PDRs, CDRs).	
4.14	List other documentation pertinent to this FPGA design (ex: datasheets, app notes, GIDEP alerts).	
4.15	List the flight FPGA programming procedure	
4.16	Summarize the revision control and configuration management of this FPGA	

Table 5 - Criticality

5 CRITICALITY	PG Ref.	Comments / Actions / Responses / Recommendations
5.1	2.7 (5.5, E.3j)	Explain the FPGA's role in any human life critical functions and any corresponding mitigations within and/or outside of the FPGA.

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

5 CRITICALITY	PG Ref.	Comments / Actions / Responses / Recommendations
5.2 Explain the FPGA's role in any one-time or irreversible functions, such as some types of deployments and manoeuvres, and any corresponding mitigations within and/or outside of the FPGA (ex: N/A – all FPGA functions can be retried and the FPGA can be reset or power cycled, drivers only powered when needed via ground commands so any errant FPGA pulses when the drivers are off are ignored).	2.7 (5.5, E.3j)	
5.3 Explain the FPGA's role in any autonomous functions such as attitude control and fault management and any corresponding mitigations within and/or outside of the FPGA (ex: 2 of 3 instances of this FPGA must agree to switch spacecraft sides, FPGA can initiate a Comm power cycle via external analog circuitry but can't permanently turn off the Comm).	2.7 (5.5, E.3j)	
5.4 Explain the FPGA's role in any safety critical functions such as pyrotechnic initiation circuits, thrusters, or high-voltage power supplies in test and flight and any corresponding mitigations within and/or outside of the FPGA (ex: arm and fire command sequences, pyros disabled while umbilical attached).	2.7 (5.5, E.3i, E.5)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

5	CRITICALITY	PG Ref.	Comments / Actions / Responses / Recommendations
5.5	Explain any radiation upset mitigations for the FPGA that are implemented outside of the FPGA (ex: none, the FPGA is power-cycled prior to each contact whether or not a problem was detected)	2.7 (10.2, 11.2)	

Table 6 - Design Entry

6	DESIGN ENTRY	PG Ref.	Comments / Actions / Responses / Recommendations
6.1	List the sources of the design files used (ex: in-house developed VHDL and Actel's CorePCIF V2.0)	2.8	
6.2	List design capture method(s) and tool(s) used, such as VHDL via text editor, schematic via DxDesigner, etc. Specify the versions of any EDA tools used. Listing the tool version(s) can be useful if a bug is later discovered in the tool(s) to see if this design is affected.	2.8	
6.3	Explain each synchronous process with a sensitivity list that includes signals other than 1 clock and 1 asynchronous reset or preset.		

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

Table 7 - Simulation

7	SIMULATION	PG Ref.	Comments / Actions / Responses / Recommendations
7.1	Summarize if and when simulation tools will be used (ex: The VHDL code will be simulated but, since the design is synchronous, static timing analysis will be used instead of back-annotated gate-level simulation).	2.8	
7.2	List name and version of each simulator used, whether these are the latest versions and, if not, why. Listing the tool version(s) can be useful if a bug is later discovered in the tool(s) to see if this design is affected.	2.8	
7.3	Explain why each internal FPGA signal that is forced/initialized in simulation but won't be in operation is OK (ex: Clk24 is generated by dividing Clk48 by 2 using a FF without reset or preset so that Clk24 oscillates before POR negates)		

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

7.4	<p>For behavioral level simulations:</p> <p>³⁵₁₇ Show code coverage</p> <p>³⁵₁₇ Disposition code not covered (ex: “when others” synthesized out, all Reed Solomon error correction permutations are impractical to simulate but many will be verified by test, FIFO full flag is unused but remains in the code as the FIFO core is re-used as-is but this design never fills the FIFO more than a few words passed half full)</p> <p>³⁵₁₇ Explain any use of coverage on/off directives (ex: code coverage turned off around “when others”)</p>	2.8	
7.5	<p>For post place and route unit delay simulations:</p> <p>³⁵₁₇ Show toggle coverage</p> <p>³⁵₁₇ Disposition logic not covered</p>	2.8	
7.6	<p>For post place and route back-annotated delay simulations:</p> <p>³⁵₁₇ List environmental corner cases run (ex: 2 corners – max MIL temp & min MIL volt, min MIL temp & max MIL volt)</p> <p>³⁵₁₇ Show toggle coverage</p> <p>³⁵₁₇ Disposition logic not covered</p>	2.7 (E.4)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

Table 8 - Synthesis

8	SYNTHESIS	PG Ref.	Comments / Actions / Responses / Recommendations
8.1	List name and version of each synthesis tool used whether these are the latest versions and, if not, why. Listing the tool version(s) can be useful if a bug is later discovered in the tool(s) to see if this design is affected.	2.8	
8.2	Summarize any synthesis warnings by type and explain why these are acceptable (ex: unused bits in registers retained for code readability). If 3 rd party or in-house IP cores are utilized, consider separating any warnings from those from any warnings unique to this design.	2.7 (E.4.2), 2.8	
8.3	Are there any sensitivity list warnings when the design is synthesized? If yes, then explain why (ex: sensitivity warnings exist in code which is being reused but has been previously validated).	2.7 (E.4)	
8.4	Verify expected/acceptable synthesis of the number and sizes (widths and depths) of RAMs, ROMs, FIFOs, etc.	2.7 (E.4)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

8	SYNTHESIS	PG Ref.	Comments / Actions / Responses / Recommendations
8.5	Verify that any user-specified redundant logic has not been optimized out by the synthesizer (ex: Triple voted RAM blocks are not logically optimized down to one RAM block as the expected number of RAM blocks remains).	2.7 (11.2)	
8.6	Verify any removed logic is intended and explain why it remains in the source code (ex: this design does not use all outputs from this code but the code is from a common library used in multiple designs).	2.7 (E.4)	
8.7	Explain why each replicated FF is OK (ex: replicated FF is not used for asynchronous signal synchronization or at the source or destination of a clock domain crossing)	2.7 (Error! eferenc e source not found.)	

Table 9 - Place & Route

9	PLACE & ROUTE	PG Ref.	Comments / Actions / Responses / Recommendations
9.1	List the name and version of the place and route tool used, whether this is the latest version and, if not, why. Listing the tool version(s) can be useful if a bug is later discovered in the tool(s) to see if this design is affected.	2.8	
9.2	Disposition all compiler warnings	2.7 (E.4)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

9	PLACE & ROUTE	PG Ref.	Comments / Actions / Responses / Recommendations
9.3	List flip-flop (FF) utilization		
9.4	List combinatorial logic utilization		
9.5	List utilizations of low-skew routing resources (ex: 2 of 4 HCLKs, 4 of 4 RCLKs)		
9.6	List utilizations of other logic/circuit elements (ex: 54 of 64 RAM blocks, 0 of 8 PLLs)		
9.7	List the total user I/O utilization and how many, if any, of these are spares or user test pins (ex: 100%, 198 of 198. 18%, 36, of which are spare outputs driving low).		
9.8	Explain how the FPGA pin assignments are verified with the board's pinout for the FPGA (ex: manually by 1 person, manually by 2 people, fully automated)		
9.9	List the operational and specified (constraint) junction (die) temperature ranges for all instances of this FPGA, including cold-start (ex: Operate between 0 C - 75 C with cold-start at -10 C, MIL range specified)	2.7 (9.3)	Operational: Specified:
9.10	List the operational and specified (constraint) voltage range(s) for all instances of this FPGA. (ex: Operational: 3.3V ± 5%, Specified: MIL (± 10%))	2.7 (9.3)	Operational: Specified:

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

9	PLACE & ROUTE	PG Ref.	Comments / Actions / Responses / Recommendations
9.11	List the operational and specified (constraint) radiation dosages for all instances of this FPGA. (ex: Operational: 35 krads, Specified: 100k rads)	2.7 (9.3)	Operational: Specified:
9.12	For the static timing reports, list all combinations of operating conditions (temperature, voltage, TID, delay analysis) analyzed. If a slower speed grade FPGA will be used, check minimum timing using the fastest speed grade, in case the FPGA vendor delivers a faster FPGA (from their current inventory). Check maximum timing with purchased speed grade.	2.7 (9.3, 9.4, App. E)	
9.13	Briefly explain why each suggested constraint in the constraints coverage report is not needed (ex: input is static, input is asynchronous, output is not used synchronously to its clock)		
9.14	Verify that any user-specified redundant logic has not been optimized out by the Place and Route tool (ex: Triple voted RAM blocks are not logically optimized down to one RAM block as the expected number of RAM blocks remains).	2.7 (11.2)	
9.15	Show that FPGA clock buffer input nets are as short as possible using floor planner tool (ex; FPGA clock input pin is on same side of FPGA as clock buffer)	2.7 (3.1)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

9	PLACE & ROUTE	PG Ref.	Comments / Actions / Responses / Recommendations
9.16	Explain each clock net that is not point to point from the clock source to the clock buffer (ex: CLK routes from INBUF to HCLKINT and OUTBUF to a testpoint but these are all co-located to minimize crosstalk and signal integrity concerns)	2.7 (3.1)	
9.17	For asynchronous circuits, show analysis of race conditions over range of environmental effects.	2.7 (9.5)	

Table 10 - Clocks


10	CLOCK(S)* PG Ref: 0, 9.1, 9.2, 9.6	Worst Case Operating		Worst Case Post-Route		Timing Utilization % (max path delay / min edge separation x 100%)				Comments / Actions / Responses / Recommendations
		Frequency	Duty Cycle with Jitter	Frequency	Duty Cycle with Jitter	Rising to Rising	Rising to Falling	Falling to Rising	Falling to Falling	
	example_clk1	18.2 MHz	60/40	27 MHz	50/50	67%	80%	75%	n/a	
	example_clkA and example_clkB	33 MHz	70/30	36.3 MHz	50/50	91%	n/a	n/a	n/a	
10.1										
10.2										
10.3										
10.4										
10.5										
10.6										
10.7										

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

Table 11 - Clocking

11	CLOCKING	PG Ref.	Comments / Actions / Responses / Recommendations
11.1	Provide or provide reference to a clock diagram showing all clock inputs, derived clocks, clock buffers and high-level blocks of all user logic, memories, IP cores, etc.	2.7 (3.4)	
11.2	List the number of clock domain crossings (CDCs) and if there are any, fill in the embedded spreadsheet. Hover over spreadsheet cells with notched corners for help and examples.	2.7 (3.5, 3.8,9.1)	Number of CDCs:  CDC Table
11.3	For each CDC or group of CDCs, explain timing margins (ex: source pulse is stretched for 3 source clock periods to ensure capture by the destination clock which runs at ½ the frequency of the source clock and the minimum back to back source pulse timing is 20 source clocks)	2.7 (3.8)	
11.4	Explain how any clocks which are NOT on low skew / global clock networks are skew safe	2.7 (3.1)	
11.5	Explain how any clocks which are NOT on low skew / global clock networks and more susceptible to SET and SEU effects are acceptable	2.7 (3.1)	
11.6	Explain any clock domains that use both negative and positive edges	2.7 (3.6)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

11 CLOCKING		PG Ref.	Comments / Actions / Responses / Recommendations
11.7	Explain any clock domains that use only negative edges	2.7 (E.4.2c)	
11.8	Explain use of any gated, including by resets or presets, clocks or clocks containing dynamic hazards	2.7 (6.1,6.2)	
11.9	Explain any clock to clock frequency and/or phase requirements / assumptions (ex: 10 MHz made from free-running divide by 4 from 40 MHz, 10 to 40 MHz CDCs are treated as synchronous as 10 MHz lags the 40 MHz but 40 to 10 MHz CDCs are treated as asynchronous)	2.7 (9.1)	
11.10	Explain any clock and asynchronous reset/preset requirements and assumptions (ex: clock may be running upon reset assertion, clock must be running prior, during and after reset negation)	2.7 (5.4)	
11.11	Explain each clock domain that does NOT have its own synchronously negated reset(s)	2.7 (5.2)	
11.12	Explain how any Actel SX CLKA/B (aka RCLK) domains used for clocking are skew-safe (ex: RCLK buffer only drives CLK inputs of "standard" FFs (not CC-FFs) and no C-cells therefore clocking is skew-safe per Actel)		

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

11 CLOCKING	PG Ref.	Comments / Actions / Responses / Recommendations
11.13		Explain how any Actel SX72 QCLK domains used for clocking which span quadrants are skew-safe (ex: N/A – each QCLK domain is driven by a QCLKINT, not QCLKBUF, which forces each domain's FFs to be in a single quadrant)
11.14	2.7 (3.3)	Explain how the implementation of any PLLs or DLLs used by the FPGA meet board requirements. Show analysis of SEU effects, frequency drift and jitter over temperature and supply voltage.

Table 12 - Asynchronous Reset and Preset Driver(s)

12 ASYNCHRONOUS RESET AND PRESET DRIVER(S)* PG Ref: 5, 5.2	Purpose and scope within and outside of the FPGA	Assertion and/or Negation Method	Timing margin relative to oscillator, PLL, clock tree, etc. start-up times, min operational vs. min needed pulse widths and/or clock edge(s)	Verified by test, sim, other analysis, etc.	Comments / Actions / Responses / Recommendations
Example_POR	Power on reset assertion to all FFs except free-running clock divider FFs	Asynchronous assertion	Asserted during power-up and for ~20 msec after 3.3V is good. Oscillator start-up time is 10 msec.	Test and simulation	
Example_POR_release	Power on reset negation to all FFs except free-running clock divider FFs	Synchronous negation via 2 rising edge FFs	At least 23 ns slack from negation to next clock edge	Test and simulation	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

12 ASYNCHRONOUS RESET AND PRESET DRIVER(S)* PG Ref: 5, 5.2		Purpose and scope within and outside of the FPGA	Assertion and/or Negation Method	Timing margin relative to oscillator, PLL, clock tree, etc. start-up times, min operational vs. min needed pulse widths and/or clock edge(s)	Verified by test, sim, other analysis, etc.	Comments / Actions / Responses / Recommendations
	Example_Discrete_Reset	Only resets 1553 interface	Synchronous assertion if >= 1 msec for 8 clocks		Simulation	
	Example_CPU_Wdog	Only resets CPU interface (and CPU)	1 clock wide synchronous pulse if watchdog not petted			
12.1						
12.2						
12.3						
12.4						
12.5						
12.6						

Table 13 - Asynchronous Resets and Presets

13 ASYNCHRONOUS RESETS AND PRESETS		PG Ref.	Comments / Actions / Responses / Recommendations
13.1	Provide or provide reference to a reset diagram showing all reset input pins, internal resets, reset synchronizers with clock names, reset buffers and high-level blocks of all user logic, memories, IP cores, etc.	2.7 (5.2)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

13 ASYNCHRONOUS RESETS AND PRESETS		PG Ref.	Comments / Actions / Responses / Recommendations
13.2	Explain why each FF that is NOT asynchronously reset or preset is OK (ex: 128 FFs are not reset as these FFs are data bits of a FIFO so the initial values don't matter, initial values of data path FFs don't matter)	2.7 (5.2)	
13.3	Explain all FFs that have both asynchronous reset and presets (ex: Each RT address bit is reset or preset, but not both, depending on its input pin setting)		
13.4	Explain why each FF that is self-clearing or self-setting (feedback from FF's output to FF's asynchronously reset and/or preset) is OK (ex: No self-clearing FFs, internal watchdog resets all FFs including the watchdog FFs but resetting the watchdog FFs is OK as this negates the reset)		
13.5	For each clock domain, explain why any path delays which exceed one clock period for synchronously generated signals that drive resets or presets are OK, as these may release some FFs one clock later than the rest of the FFs in the domain		

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

Table 14 - Storage Elements

14 STORAGE ELEMENTS		PG Ref.	Comments / Actions / Responses / Recommendations
14.1	Explain any additional upset effect, detection, correction and/or recovery method(s) for any TMR FFs (ex: none as the frequency of upsets to these FFs is low enough to not require additional mitigations)	2.7 (E.4)	
14.2	Explain handling of detection, correction and/or recovery method(s) for upsets of any non-TMR FFs (ex: n/a or upsets to these FFs result in an acceptable level of data corruption)	2.7 (4.4)	
14.3	Explain upset effect, detection, correction and/or recovery method(s) for any CC-module FFs (ex: n/a).	2.7 (4.1, 10.2)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

14	STORAGE ELEMENTS	PG Ref.	Comments / Actions / Responses / Recommendations
14.4	Explain use of any latches (as opposed to FFs)	2.7 (E.4.2)	
14.5	Explain handling of detection, correction and/or recovery method(s) for upsets of any latches	2.7 (E.4.2)	
14.6	Explain upset effect, detection, correction and/or recovery method(s) for any RAMs	2.7 (8.6,8.7)	
14.7	Are status registers designed as clear on read and/or are FIFOs accessed directly from an entity that performs pre-fetching (such as PCI) that could erase/corrupt data? If yes, then explain mitigation(s).	2.7 (10.2)	
14.8	For multi-bit settings (software loadable registers, input busses, packet fields, etc.), are the zero values clearly defined/documented? (ex: Does a value of 0 for the watchdog timer means it is disabled or it's enabled with its smallest interval)		
14.9	Explain all means provided to verify the correct version of the FPGA is installed (ex: software readable version register manually incremented by the FPGA designer per revision, JTAG I/O connected to header on PWA to read checksum).		

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

14 STORAGE ELEMENTS		PG Ref.	Comments / Actions / Responses / Recommendations
14.10	Explain how timing is met for any cascaded Actel AX/RTAX block RAMs per http://www.actel.com/documents/CN1103_RTAX_signal_coupling.pdf (ex: N/A, this FPGA uses no cascaded block RAMs, block RAMs cascades were made via SmartGen and timing scaled per the app note)		

Table 15 - Watchdogs

15 WATCHDOGS		PG Ref.	Comments / Actions / Responses / Recommendations
15.1	How is the watchdog enabled at power-up (ex: Watchdog is disabled at power-up. Watchdog is automatically enabled after power-up with 16 second timeout)?		
15.2	How is the watchdog serviced (aka kicked, petted)?		
15.3	List the watchdog interval(s) and if more than one, indicate the default interval, explain how an interval is selected and when the interval can be changed (ex: 1 to 16 (default) seconds in 1 second steps. SW can only change the watchdog interval by writing the watchdog register when the watchdog is disabled)		
15.4	If the watchdog can be disabled, explain how and why (ex: SW cannot disable the watchdog once it's enabled)		

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

15 WATCHDOGS	PG Ref.	Comments / Actions / Responses / Recommendations
15.5		What happens if the watchdog activates (aka expires, times-out, asserts, triggers)? (ex: The CPU and FPGA are reset)?
15.6		How is the watchdog (re-)enabled after it activates (aka expires, times-out, asserts, triggers)? If the watchdog supports more than one interval, state which interval is used (ex: The watchdog disables after activating (one-shot)).

Table 16 - State Machine(s)

16 STATE MACHINE(S)* PG Ref: 0	Qty	Encoding (ex: compact, one-hot)	Number of defined states	Undefined states after synthesis (Y/N)	Error detection and/or correction method (ex: none, TMR FFs, TMR FFs + Hamming)	Error notification (ex: none, status bit asserted, interrupt asserted)	System consequence(s) and response(s) to an upset PG 2-7 (4.4)	Comments / Actions / Responses / Recommendations
Example_arbiter	1	Compact	8	N	TMR FFs + Hamming	Interrupt asserted	No or 1 data word corruption	
Example_Blk_RAM_RW	3	One-hot	7	Y	TMR FFs	None	Data corruption until ground intervention	
16.1								
16.2								
16.3								
16.4								
16.5								
16.6								

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT

<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

* Disposition all state machines used in this FPGA.

Table 17 - Interrupts to Software

17	INTERRUPTS TO SOFTWARE	PG Ref.	Comments / Actions / Responses / Recommendations
17.1	Show/explain the interrupt tree, if applicable (ex: the FPGA's interrupt cause register shows the 1553 interrupts, among others, but software must clear these by accessing the SuMMIT chip)		
17.2	Explain interrupt generation, masking and clearing		
17.3	Explain how the simultaneous occurrence of a second interrupt coincident with the clearing of the first interrupt is properly handled or not possible		

Table 18 – Interfacing to Non-Volatile Memories


18	INTERFACING TO NON-VOLATILE MEMORIES	PG Ref.	Comments / Actions / Responses / Recommendations
18.1	Explain how non-volatile memory is protected against write-corruption during power-down or reset.	2.7 (8.1)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

18 INTERFACING TO NON-VOLATILE MEMORIES		PG Ref.	Comments / Actions / Responses / Recommendations
18.2	Explain how corrupted content (by radiation, degradation, or interrupted writes, etc) in non-volatile memory is detected (parity, checksum, RS, etc) and corrected (RS, spare image, etc) to meet board requirements	2.7 (8.2, 8.5)	
18.3	Explain how design maximizes useful life of non-volatile memory (e.g., performing page-writes)	2.7 (8.3)	
18.4	Show how FPGA design along with non-volatile memory's data retention capabilities meet mission life requirements	2.7 (8.6)	

Table 19 - User I/O

19 USER I/Os		PG Ref.	Comments / Actions / Responses / Recommendations
19.1	Complete the first (leftmost) embedded worksheet "All User Pins" then either complete the other worksheets or provide the requested timing information if it exists in another format. Hover over spreadsheet cells with notched corners for help and examples.		 User IO Types and Timing

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

Table 20 - Input Pins

20	INPUT PINS	PG Ref.	Comments / Actions / Responses / Recommendations
20.1	For all inputs, including inputs of bidirectionals, explain steady-state and switching compatibility with external logic (technology(ies), edge rates, overshoot/undershoot, pull-ups/downs, etc.)	2.7 (2.4,2.6)	
20.2	Explain how driving device(s) are precluded from degrading or damaging unpowered FPGA inputs (ex: All FPGA inputs are driven by devices powered from the same 3.3V supply as powers the FPGA I/O, the LVDS receivers are powered before the FPGA but are held in tristate until the FPGA powers up)	2.7 (2.7)	
20.3	Explain how floating too long is precluded when the driving device(s) are tristated and/or powered off during operation as well as reset(s) (ex: N/A - input driver is powered by same 3.3V supply, pull-downs preclude floating)	2.7 (2.3, 10.2)	
20.4	For FPGA inputs that are directly connected (not buffered) to the outer assembly/chassis, explain precautions taken during I&T to prevent degradation or damage to the input (ex: N/A – input is on net internal to the board/box, 1 kohm series resistor)	2.7 (2.7, 10.2)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

20	INPUT PINS	PG Ref.	Comments / Actions / Responses / Recommendations
20.5	Explain any pulse width checking/rejection for each input (ex: HW_DEC_CMD_RST discrete from Comm may glitch when switching but will assert for 10 usec so FPGA uses shift register to reject pulses <1 usec)		

Table 21 - Output Pins

21	OUTPUT PINS	PG Ref.	Comments / Actions / Responses / Recommendations
21.1	For all outputs, including outputs of bidirectionals, explain steady-state and switching compatibility with external logic (technology, drive strength, slew rate, load pF, overshoot/undershoot, pull-ups/downs, etc.).	2.7 (2.2, 2.5, 2.6)	
21.2	List or reference the FPGA manufacturer's Simultaneous Switching Outputs/Noise (SSO/SSN) rules/recommendations and explain SSO/SSN mitigation(s) used or why none are needed.	2.7 (2.1, 8.4)	
21.3	Explain signal integrity mitigations and verification or why not needed (ex: series termination on all outputs).	2.7 (2.2)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

21	OUTPUT PINS	PG Ref.	Comments / Actions / Responses / Recommendations
21.4	Explain how floating too long is precluded when the FPGA output is powered off or tristated operationally or due to reset (ex: pull-down resistors on board, N/A - all devices on this net are powered by the same 3.3V supply, FPGA performs bus parking with 1 clock of float time at the beginning and end of SRAM read cycles).	2.7 (5.5)	
21.5	For outputs used as clock and data (source-synchronous to other logic and/or externally back to this FPGA), explain how setup and hold times are ensured at the receiving logic.	2.7 (3.2)	
21.6	For outputs used as clocks (without data, to other logic and/or externally back to this FPGA), explain how these meet board-level interface requirements when the FPGA enters, is in, and exits each type of reset.	2.7 (E.5)	
21.7	For outputs used as resets (to other logic and/or externally back to this FPGA), explain how these meet board-level interface requirements when the FPGA enters, is in, and exits each type of reset.	2.7 (E.5)	
21.8	For outputs used as resets (to other logic or externally back to this FPGA), explain any delay or stretching of these reset outputs.	2.7 (E.5)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

21 OUTPUT PINS		PG Ref.	Comments / Actions / Responses / Recommendations
21.9	Explain how each FPGA output is not damaged or degraded when driving un-powered devices during operation, power up and power down (ex: N/A - FPGA outputs drive devices powered by the same 3.3V that powers the outputs)	2.7 (7.1, 7.2)	
21.10	Explain how each FPGA output does not damage or degrade un-powered devices during operation, power up and power down (ex: the FPGA outputs to the SDRAMs drive low until the SDRAM is powered via FPGA input of the SDRAM power)	2.7 (7.1, 7.2)	
21.11	For outputs that directly connect (are not buffered) to the outer assembly/chassis, explain precautions taken during I&T to prevent degradation or damage to the input (ex: N/A – output is on net internal to the board/box, 1 kohm series resistor)	2.7 (11.2)	

Table 22 - Bidirectional/Tristate Pins

22 BIDIRECTIONAL/TRISTATE PINS		PG Ref.	Comments / Actions / Responses / Recommendations
22.1	Explain how floating too long is precluded during resets (ex: pull-down resistors on board)	2.7 (2.3)	
22.2	Explain how floating too long is precluded during idle operation (ex: FPGA drives its outputs to park on the bus)	2.7 (2.3)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

22 BIDIRECTIONAL/TRISTATE PINS		PG Ref.	Comments / Actions / Responses / Recommendations
22.3	Explain how floating too long is precluded during active operation (ex: FPGA performs bus parking with 1 clock of float time at the beginning and end of each read cycle))	2.7 (2.3)	
22.4	Explain how contention is precluded during resets (ex: FPGA tristates its outputs during resets)	2.7 (2.3)	
22.5	Explain how contention is precluded during idle operation (ex: FPGA parks on the data bus and holds the external SRAM in tristate)	2.7 (2.3)	
22.6	Explain how contention is precluded during active operation (ex: One 40 MHz clock of float time between bus drivers exceeds worst case tristated time of 19 ns among all devices on the bus)	2.7 (2.3)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

Table 23 - Spare/Unused I/O Pins

23	SPARE, UNUSED AND USER-DEFINED TEST I/O PINS	PG Ref.	Comments / Actions / Responses / Recommendations
23.1	Explain FPGA and board configuration/handling of spare, unused and/or user-defined test I/O pins relative to manufacturer, NASA and/or other rules and recommendations (ex: 15 I/Os are defined as outputs always driving low and connect to testpoint pads, 25 unused I/Os are undefined in the design but 10 of these pins are connected to series resistor pads in case additional signals are needed)	2.7 (1.2, 1.3)	

Table 24 - Special/Test Pins

24	JTAG AND FPGA SPECIAL/TEST PINS	PG Ref.	Comments / Actions / Responses / Recommendations
24.1	Explain test and flight handling of JTAG TRST pin relative to manufacturer, NASA and/or other rules and recommendations	2.7 (App. C)	
24.2	Explain test and flight handling of JTAG TCK, TMS, TDI, and TDO pins relative to manufacturer, NASA and/or other rules and recommendations	2.7 (App. C)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

24 JTAG AND FPGA SPECIAL/TEST PINS		PG Ref.	Comments / Actions / Responses / Recommendations
24.3	Explain test and flight handling of other dedicated test pins (ex: Actel PRA, PRB, PRC & PRD) relative to manufacturer, NASA and/or other rules and recommendations	2.7 (App. C)	
24.4	Explain test and flight handling of unused clock inputs. For Actel AX FPGAs which use single ended (positive) clock pins, are the corresponding negative clock pins (CLKEN, CLKFN, CLKGN, CLKHN, HCLKAN, HCLKBN, CLKCN, HCLKDN) grounded or configured as static inputs/outputs? (Coupling has been observed from the negative clock pins configured as outputs to the corresponding positive clock input).	2.7 (App. C)	
24.5	Explain handling of any pin differences between the test/COTS and flight parts (ex BGA to CQFP adapter disables COTS inputs not present in the flight part, Actel AX PLL pins not present in RTAX but connected on the flight board to allow use of COTS part during test)	2.7 (App. C)	
24.6	If user implemented test modes (not manufacturer implemented like JTAG) are provided, explain how they are disabled or avoided in flight (ex: active high test mode input pin which speeds-up the msec counter for simulation is grounded for flight)	2.7 (1.4)	

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

24.7	For Actel AX devices, Explain test and flight handling of Vpump	2.7 (C.1)	
24.8	Explain the handling of any extra-ESD sensitive pins	2.7 (10.2)	

Table 25 - Power, Ground and Thermal

25	POWER, GROUND AND THERMAL	PG Ref.	Comments / Actions / Responses / Recommendations
25.1	List power modes (ex: record, playback and idle) for this FPGA and for each mode, the estimated power per supply (ex: 1.5V and 3.3V) and maximum die temperature	2.7 (7.4)	
25.2	For FPGAs with power sequencing requirements, explain how these are met (ex: core voltage is linearly regulated from the I/O voltage and therefore follows and will not exceed the I/O voltage during power up. On-board capacitance ensures the I/O voltage exceeds the core voltage during power down. Power supply rise time is sufficient)	2.7 (7.1, 7.3, 7.4)	
25.3	Describe how power is estimated and list manufacturer datasheet, app notes, spreadsheets, etc. used (ex: toggle rate method: percentage or simulation vectors, clock frequency(ies) specified)	2.7 (7.4)	
25.4	Describe how heat is transferred from the FPGA (ex: conducted through PWB to aluminum heat sink)		

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

25	POWER, GROUND AND THERMAL	PG Ref.	Comments / Actions / Responses / Recommendations
25.5	Describe how FPGA power integrity is addressed (ex: FPGA datasheet, FPGA application note(s), power integrity analysis tool(s), power integrity measurements). List name and version of each document and tool used, whether these are the latest versions and, if not, why. Listing the tool version(s) can be useful if a bug is later discovered in the tool(s) to see if this design is affected.	2.7 (7.4)	
25.6	Describe FPGA decoupling (ex: 3 decades of capacitors placed around the FPGA, near each power pin with wide breakout traces and, where possible, multiple vias per application note XYZ and power integrity simulation)	2.7 (7.4)	

Table 26 - PWB and Assembly Constraints

26	PWB AND ASSEMBLY CONSTRAINTS	PG Ref.	Comments / Actions / Responses / Recommendations
26.1	Describe PWB and assembly constraints driven by this FPGA (ex: Fine pitch CGA required via-in-pad and/or microvias, CGA required N layer PWB, CGA coplanarity required stiffeners)		

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

Table 27 - Others

27 OTHERS		PG Ref.	Comments / Actions / Responses / Recommendations
27.2	Explain any defensive design against credible but unplanned events (invalid register or signal values, floating external inputs, interfaces to devices that could lock up, ESD on external inputs, etc)	2.7 (10.2)	
27.3	Show how integrity of configuration memory for reprogrammable FPGAs is maintained during all operational and test conditions.	2.7 (11.1)	
27.4	Describe SEE mitigation techniques and use of TMR schemes to meet requirements (TMR at register level, embedded block level, component level, etc)	2.7 (11.2)	
27.5	List radiation hardness levels of embedded FPGA functions and show how they are being used to meet requirements	2.7 (11.3)	
27.6	Show how FPGA will be reconfigured in-flight such that normal operation is not interrupted and operation of the spacecraft is not jeopardized.	2.7 (11.4)	
27.7			
27.8			

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.

DIRECTIVE NO. 500-PG-8700.2.7B
EFFECTIVE DATE: August 13, 2012
EXPIRATION DATE: March 30, 2021

CHANGE HISTORY LOG

Revision	Effective Date	Description of Changes
Baseline	08/12/2005	Initial Release
A	08/12/2010	Removed hyperlinks, clarified wording, added content in the form of text and diagrams, removed board-level content, added checklist.
B	8/13/2012	Updated and removed “must” statements. Other minor corrections. Edits and additions made to Appendix D
	08/07/2013	This document has been administratively changed at the template block from Procedures and Guidelines to Center-Wide Procedures and Guidelines.
	07/28/2017	Administratively extended for 1 year.
	07/10/18	Administratively extended for 1 year.
	08/05/19	Administratively extended for 1 year.
	08/17/20	Administratively extended for 6 months.

CHECK THE GSFC DIRECTIVES MANAGEMENT SYSTEM AT
<http://gdms.gsfc.nasa.gov> TO VERIFY THAT THIS IS THE CORRECT VERSION PRIOR TO USE.