JOTP-051
10 February 2012

DEPARTMENT OF DEFENSE

JOINT ORDNANCE TEST PROCEDURE (JOTP)

TECHNICAL MANUAL FOR THE USE OF LOGIC DEVICES IN SAFETY FEATURES

DoD Fuze Engineering Standardization Working Group (FESWG)

FOREWORD

I. Beneficial comments (recommendation, additions, and deletions) and any pertinent data which may be of use in improving this document shall be addressed to:

Chairman
DoD Fuze Engineering Standardization Working Group
U.S. Army Armament Research, Development and Engineer Center
ATTN: Army Fuze Management Office (RDAR-EIF)
Picatinny Arsenal, NJ 07806-5000

II. DoD JOTP-051, Technical Manual for the use of Logic Devices in Safety Features, dated 10 February 2012, supersedes DoD FESWG document 2007-1, Technical Manual for the use of Logic Devices in Safety Features, dated 08 March 2011. No changes to the requirements occurred.

1. SCOPE

There are many safety issues and requirements involved with the use of logic devices. Some are addressed by MIL-STD-1316, MIL-STD-1911, MIL-STD-1901 and STANAG-4187, STANAG-4497, STANAG-4368 and may be reiterated here. This document is intended to clarify these requirements as applied to Safety Features (SFs) implemented with logic devices. These include but are not limited to: programmable logic devices (PLDs), complex programmable logic devices (CPLDs), field programmable gate arrays (FPGAs), application specific integrated circuits (ASICs), microcontrollers, discrete logic, etc.

While some logic devices may be viewed as better suited for safety applications, it is important to note:

a. All logic devices can be implemented in an unsafe manner.
b. There are potential safety issues associated with the use of any type of logic device in safety critical applications.
c. Individual technologies may require measures not specifically addressed here.

Appendix A provides further clarification of the requirements in Section 2. Appendix B provides definitions.

2. REQUIREMENTS

2.1. To minimize the subversion of SF(s) due to unintentional and/or unrecognized modes of operation, including failure modes, each SF implemented with logic shall use the least complex logic device that can practically perform the required functionality.

2.2. While fixed-in-structure devices are acceptable and preferred, to avoid degradation of a safety feature, any logic device used in the implementation of that feature:

2.2.a. Shall not be re-programmable.

2.2.b. Shall not be alterable by credible environments.

2.2.c. Shall not have the SF logic configuration reside on volatile memory.

2.2.d. Should be rated to meet or exceed the lifecycle environments of the system. Shall have engineering rationale provided and associated risk(s) for logic devices not rated to meet or exceed the lifecycle of the system.

2.3. In order to minimize the potential for common cause failures, where all SFs are implemented with logic devices, at least two SFs shall be implemented with dissimilar logic devices. The degree of dissimilarity shall be sufficient to ensure that any credible common cause susceptibility will not result in unsafe operation of all logic devices. Where practical, at least one SF shall be implemented with discrete component(s).

2.4. In order to ensure the device operates as intended, SF logic shall be implemented in accordance with the device manufacturer's latest specifications and notes. In addition, all programming functionality, testing functionality, unused pins, and any other normally non-operational functionality of the SF logic shall be appropriately disabled and terminated, according to the device manufacturer's latest specifications and notes. Non-compliance shall be subject to review and approval by the cognizant safety authority.

2.5. During and after exposure to power transfers, transitions, and/or transients, logic devices shall not operate in a manner that result in the degradation of SF(s).

2.6. SF timing functions, excluding arming delay, implemented within logic shall not be susceptible to single point or common cause failures resulting in unintended arming.

2.7. Single point or common cause failures of the arming delay shall be reduced to a minimum during the arming cycle. The time window associated with these failures shall be reduced to a minimum and shall exist only at or near expiration of the intended arming delay.

2.8. The logic implementation shall replicate the documented design.

2.9. Where all SFs are implemented with logic devices, the SF logic shall be physically and functionally partitioned from each other. This minimizes the potential for inadvertent subversion such as sneak paths or Single Event Upsets.

2.10. In order to minimize the potential for unknown failure modes, all logic and/or functionality available within a device shall be disclosed, documented, and assessed in safety analyses and evaluations.

2.11. SF documentation shall include the complete logic flow with all inputs and outputs defined, along with timing and interdependence of events.

2.12. Manufacturing documentation and processes shall ensure that logic devices within an approved design are produced with an identical configuration.

2.13. Development tools shall be documented and controlled via configuration management procedures.

2.14. Power for SF logic should be partitioned from other power such as communication or platform power.

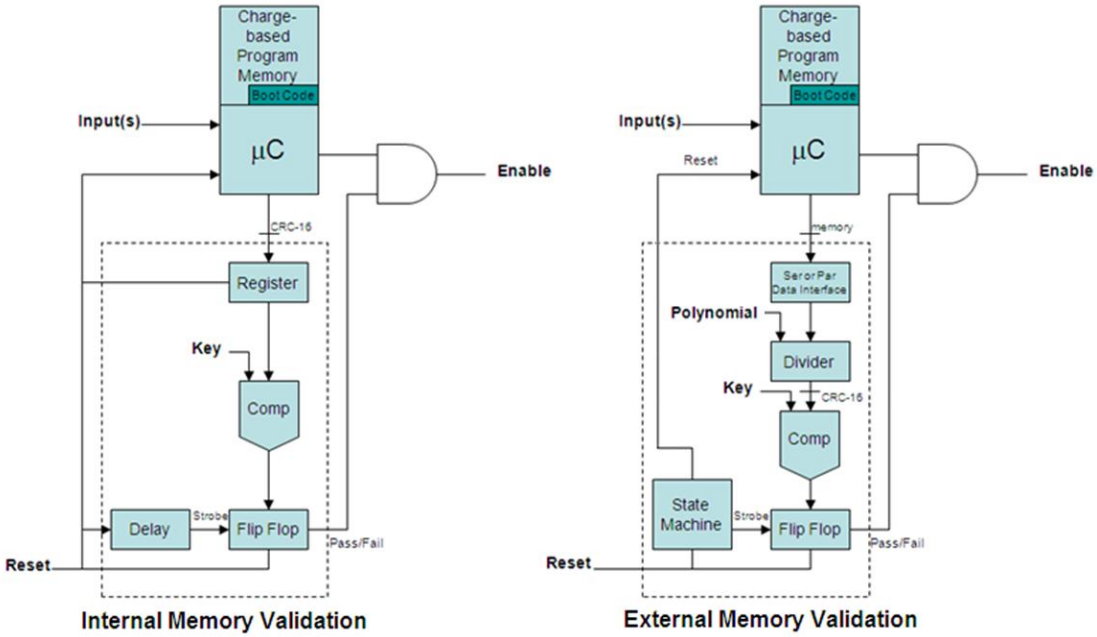
2.15. Power for SF logic should be applied as late in the launch sequence or operational deployment as practical.

Appendix A

Guidelines for the Application of the DoD FESWG requirements for the use of Logic Devices in Safety Features

A.1. The preferred design approach is to use the simplest logic device that can perform the required functionality. For example, a simple 'AND' function does not need to be implemented using an FPGA. A complex device will require more analysis, documentation, testing and more scrutiny by the safety authority.

A.2. Since, any changes to the SF hardware can adversely compromise the safety of the design, it is critical to establish and maintain a stable configuration throughout the lifecycle of the SF. A programmable logic device, including associated memory devices, may be considered as satisfying the requirements of paragraph 2.2 if, once programmed or configured during manufacturing, the configuration of its internal logic cannot be changed. Additionally, for devices relying on charged-based memory to implement a SF, a method of validating the integrity of the memory shall be performed prior to executing the safety function. The memory must be validated with the rigor equivalent to, or better than, that of a 16 bit Cyclic Redundant Check (CRC16). This computed result shall be externally compared against a known value that is stored externally. The external device(s) shall (1) be fixed-in-structure, (2) be dedicated circuitry, (3) not contain and be exclusive from any other functions, and, (4) when feasible, not be implemented as part of any other SF. Consult with the appropriate Service Safety Authority for guidance. A notional example of an internal and external memory validation is provided below:



A.3. It is recognized that all logic devices can be susceptible to unpredictable operating states in the presence of certain environmental stresses/conditions as well as non-optimal and/or undesired design or manufacturing implementations. For this reason, at least two SF should be implemented with dissimilar logic devices resulting in SFs that have dissimilar failure modes. Dissimilar logic refers to distinct methods and/or materials used to develop a particular device that result in devices with minimal, but known and assessed, common cause failures. Some examples would be a Full Custom ASIC, discrete components, Metal-to-Metal Antifuse FPGA, Oxide/Nitride/Oxide Antifuse FPGA, microcontroller, etc.

A.4. A compliance matrix is an acceptable format for recording that the design and programming procedure are in accordance with all the device manufacturer's specifications and notes. This includes programming, power up, power down, timing, operational, etc. Appropriately disabled means that the failure of the disabling method will not result in an unsafe failure of the device. If a conflict between the manufacturer's specifications (including notes) and other requirements (safety or otherwise) exists, then the justification for the deviation from the manufacturers specifications shall be reviewed and approved by the cognizant safety authority. If a design deviates from the manufacturer's specifications then it shall be shown that this deviation does not negatively impact safety.

A.5. Power transfer refers to switching from one power source to another. Transition of power refers to the expected power-up and power-down voltage/current profiles (including rise/fall times) that the logic device power/ground inputs could be subjected to during its lifecycle. Power transients include any noise, voltage/current spikes or surges, brown outs, etc., to which the logic device power/ground inputs could be subjected during its lifecycle. One approach to mitigate the negative effects of power transfers, transitions, and/or transients is for each SF to utilize multiple independent resets logically OR'd together. Multiple SFs, given adequate isolation, could employ these same resets. Prior to intended arming, SF logic should initialize in a safe state and reset to a safe state.

A.6. Any safety critical clocks should have a method of verification. The preferred method would be independent clocks or verification of the safety critical clock(s) with a known timed event. Multiple SFs could employ the same time basis for verification. The intent of this requirement is to ensure arming events/environments are correctly validated and invalid environments are not recognized as valid due to clock skew or failure.

A.7. Some methods to mitigate the potential for single point or common cause failures of the arming delay include: (1) The use of independent timers is preferred. (2) The shortest arm delay set in hardware should be set to the maximum practical value. (3) Any transmission and validation of arm times must be as robust as practical (checksum, parity, CRC, etc.).

A.8. This requirement is to ensure the intended design (Logic Device schematics, software code, etc.) is actually what is in hardware/software. For example: (a) in VHDL, if the design has a binary state machine, the hardware does not have a one-hot state machine, which is functionally equivalent but physically different, (b) a synthesizer's optimizer should not adversely affect the approved design (the preferred approach is for the designer to disable any optimizers), and (c) a Logic Device vendor should not optimize/change/make additions to the approved design.

A.9. The preferred partitioning method is to use distinct components with separate electrical paths. Electronic circuits controlling independent SFs should be physically partitioned into functionally dissimilar elements, neither of which can, during normal operation or upon failure, independently arm the system. Additionally, functional and physical partitioning of the SF logic and non-safety logic is encouraged.

A.10. Undocumented functions or logic within a SF can compromise the safety of the design and is unacceptable. All the logic and functionality within a logic device used in the SF should be documented. This includes any programming functionality, testing functionality, JTAG, SCAN, MODE, etc.

A.11. The documentation is intended as a record to assist in design verification.

A.12. The documentation is intended as a record to assure that logic devices within an approved design are reproduced consistently throughout production. The manufacturer should maintain configuration control on the manufacturing tool suite to include version numbers and manufacturer's datasheets for the computers, operating systems, compilers, synthesizers, analysis tools, testing tools, and other tools used to manufacture, analyze, test, document, and maintain the logic device application. The documentation should include all the files generated during this process. Subsequent optimization of an approved design is unacceptable.

A.13. The documentation is intended as a record to assure the logic device configuration matches the intended design. These tools can be, but are not limited to: support software/compilers, storage devices, and any other intermediate entity that poses the potential to alter the final intended "as embedded" design. The configuration management practices should be commensurate with the criticality of the end product produced.

A.14. The relevant safety board(s) will determine the appropriate partitioning of power.

A.15. The relevant safety board(s) will determine the appropriate timing for the application of power to the SF logic.

Appendix B

Definitions

a. Charged-based memory. A type of memory that relies on the storage of electrical charge to establish a binary state.

b. Common Cause Failures. Multiple component failures that result from or are caused by a single failure or an adverse environment.

c. Fixed-in-structure Devices. The contents are physically fixed by the structure of the device. Examples: core rope devices (wire wound through or around a core), fusible link devices, masked devices, and anti-fuse based devices.

d. Memory Check. A technique or process that develops a numeric value based on performing a mathematical/Boolean function on the content of memory.

e. Memory Check Validation. The process of comparing a Memory Check value against the known value to determine whether or not the content of memory has changed.

f. Safety Critical. Characterization of a condition, event, function, operation, process, or item of a system whose proper recognition, control, performance, or tolerance is essential to the operation of a SF.

g. Safety Feature. An element or combination of elements designed to prevent unintended arming and/or functioning. All the components from the environmental sensing, environment verification and safety interlock are included in the safety feature.

h. Single Event Upset. Situation, event, bit change, or other state change, not necessarily a hardware failure, which can propagate through the system causing undesirable results.

i. Single Point Failure. Situation, event, or single component failure that on its own may lead to an unacceptable state or event.

j. Sneak Path. A path or logic flow that can initiate an undesired function or inhibit a desired function. Sneak paths may be caused by unexpected paths, order of events, wrong indications or wrong interpretations of observations. The path may consist of hardware, software, or operator actions, or any combination of these.

k. Volatile Memory. The contents are not retained after the cycling of power. This class is subdivided into two subclasses: static, which will retain state indefinitely; and dynamic, where the memory must be read and subsequently refreshed. Examples: SRAM, DRAM, and SDRAM.