



Software in Programmable Components

*UL COPYRIGHTED MATERIAL –
NOT AUTHORIZED FOR FURTHER REPRODUCTION OR
DISTRIBUTION WITHOUT PERMISSION FROM UL*



Documents
Document Sharing Hub

UL Standard for Safety for Software in Programmable Components, UL 1998

Third Edition, Dated December 18, 2013

Summary of Topics

This new edition of ANSI/UL 1998 includes the following:

Clarification of Requirements for Negative Condition Branch Failure Mode

Revision of the Requirements to Address the Effects of Power Supply Voltage Variations

The requirements are substantially in accordance with Proposal(s) on this subject dated January 20, 2012 and February 15, 2013.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form by any means, electronic, mechanical photocopying, recording, or otherwise without prior permission of UL.

UL provides this Standard "as is" without warranty of any kind, either expressed or implied, including but not limited to, the implied warranties of merchantability or fitness for any purpose.

In no event will UL be liable for any special, incidental, consequential, indirect or similar damages, including loss of profits, lost savings, loss of data, or any other damages arising out of the use of or the inability to use this Standard, even if UL or an authorized UL representative has been advised of the possibility of such damage. In no event shall UL's liability for any damage ever exceed the price paid for this Standard, regardless of the form of the claim.

Users of the electronic versions of UL's Standards for Safety agree to defend, indemnify, and hold UL harmless from and against any loss, expense, liability, damage, claim, or judgment (including reasonable attorney's fees) resulting from any error or deviation introduced while purchaser is storing an electronic Standard on the purchaser's computer system.

The requirements in this Standard are now in effect, except for those paragraphs, sections, tables, figures, and/or other elements of the Standard having future effective dates as indicated in the note following the affected item. The prior text for requirements that have been revised and that have a future effective date are located after the Standard, and are preceded by a "SUPERSEDED REQUIREMENTS" notice.

**UL COPYRIGHTED MATERIAL –
NOT AUTHORIZED FOR FURTHER REPRODUCTION OR
DISTRIBUTION WITHOUT PERMISSION FROM UL**



Documents
Document Sharing Hub

No Text on This Page

**UL COPYRIGHTED MATERIAL –
NOT AUTHORIZED FOR FURTHER REPRODUCTION OR
DISTRIBUTION WITHOUT PERMISSION FROM UL**



Documents

Document Sharing Hub

DECEMBER 18, 2013



ANSI/UL 1998-2013

1

UL 1998

Standard for Software in Programmable Components

First Edition – January, 1994

Second Edition – May, 1998

Third Edition

December 18, 2013

This ANSI/UL Standard for Safety consists of the Third Edition.

The most recent designation of ANSI/UL 1998 as an American National Standard (ANSI) occurred on December 16, 2013. ANSI approval for a standard does not include the Cover Page, Transmittal Pages, Title Page, or effective date information.

Comments or proposals for revisions on any part of the Standard may be submitted to UL at any time. Proposals should be submitted via a Proposal Request in UL's On-Line Collaborative Standards Development System (CSDS) at <http://csds.ul.com>.

UL's Standards for Safety are copyrighted by UL. Neither a printed nor electronic copy of a Standard should be altered in any way. All of UL's Standards and all copyrights, ownerships, and rights regarding those Standards shall remain the sole and exclusive property of UL.

COPYRIGHT © 2013 UNDERWRITERS LABORATORIES INC.

**UL COPYRIGHTED MATERIAL –
NOT AUTHORIZED FOR FURTHER REPRODUCTION OR
DISTRIBUTION WITHOUT PERMISSION FROM UL**



Documents
Document Sharing Hub

No Text on This Page

**UL COPYRIGHTED MATERIAL –
NOT AUTHORIZED FOR FURTHER REPRODUCTION OR
DISTRIBUTION WITHOUT PERMISSION FROM UL**



Documents

Document Sharing Hub

CONTENTS

PREFACE	4
1 Scope	5
2 Definitions of Terms Used	6
3 Risk Analysis	10
4 Process Definition	11
5 Qualification of Design, Implementation, and Verification Tools	11
6 Software Design	12
7 Critical and Supervisory Sections of Software	12
8 Measures To Address Microelectronic Hardware Failure Modes	13
9 Product Interface	14
10 User Interfaces	14
11 Software Analysis and Testing	15
11.1 Software analysis	15
11.2 Software testing	15
11.3 Failure mode and stress testing	16
12 Documentation	17
12.1 User documentation	17
12.2 Software plan	17
12.3 Risk analysis approach and results	17
12.4 Configuration management plan	17
12.5 Programmable system architecture	18
12.6 Programmable component and software requirements specification	18
12.7 Software design documentation	18
12.8 Analysis and test documentation	19
13 Off-the-Shelf (OTS) Software	19
14 Software Changes and Document Control	20
15 Identification	20

APPENDIX A – EXAMPLES OF MEASURES TO ADDRESS MICROELECTRONIC HARDWARE FAILURE MODES

A1 Scope	A1
A2 Examples of Acceptable Measures for Microelectronic Hardware Failure Modes	A1
A3 Software Classes	A8
A4 Description of Fault Models	A9
A5 Description of System Structures	A9
A6 Example of the Application of Table A2.1	A10
A7 Descriptions of Acceptable Measures for Providing the Required Fault/Error Coverage Specified in Table A2.1	A11
A7.1 Descriptions of fault/error control techniques	A11
A7.2 Description of memory tests	A13
A7.3 Word protection	A14

**UL COPYRIGHTED MATERIAL –
NOT AUTHORIZED FOR FURTHER REPRODUCTION OR
DISTRIBUTION WITHOUT PERMISSION FROM UL**



Documents
Document Sharing Hub

PREFACE

The requirements in UL 1998 address non-networked embedded software residing in programmable components which are application-specific. Embedded software is software that resides in a programmable component and that performs some of the requirements of the programmable component. Non-networked embedded software is embedded software that executes on a single microprocessor/microcontroller or on redundant microprocessors/microcontrollers residing in the same physical enclosure. Application-specific means that the software is limited to a designated application which permits effective evaluation of the hazards and risks associated with the software. Programmable components are any microelectronic hardware that can be programmed in the design center, the factory, or in the field.

The requirements in UL 1998 are applicable when used in conjunction with an application-specific standard that contains requirements for safety-related functions implemented using software. UL 1998 does not apply to software in programmable components used in general purpose applications when the risks for the end-application cannot be identified. Safety-related functions are control, protection, and monitoring functions which are intended to reduce the risk of fire, electric shock, or injury to persons. When UL 1998 is applied to a specific product, it is intended that the requirements address product safety risks associated with the specific purpose (as components only) use of software in the programmable component. A product is an instrument, apparatus, implement, or machine intended for personal, household, industrial, laboratory, office, or transportation use.

The requirements in UL 1998 are not intended to be used as the sole basis for reviewing programmable components. UL 1998 is intended to be used in conjunction with other safety standards that address the programmable component hardware. Requirements in UL 1998 may be amended or superseded by requirements in a product safety standard, a directive, regulation, or a purchasing specification.

Due to the diversity of software functions and the application-specific nature of testing programmable components, UL 1998 indicates neither testing protocols nor tools. Instead, UL 1998 contains requirements that define test objectives and criteria for the general case. This permits the user to choose from many testing protocols and tools as long as the test objectives and criteria are met. Users of UL 1998 are encouraged to specify and make available to the public the test protocols and tools used when applying the requirements of UL 1998.

The UL 1998 Standard covers handling of changes to the software in the programmable component after release. The recognition of maintenance processes for the handling and qualification of software and programmable component modifications that occur after release will be considered. This consideration will include consideration of all requirements stipulated by Authorities Having Jurisdiction.

The use of field performance data to demonstrate compliance with the software analysis and test, off-the-shelf software, and tool validation requirements of UL 1998 will be considered for a period determined for each product type.

1 Scope

1.1 These requirements apply to non-networked embedded microprocessor software whose failure is capable of resulting in a risk of fire, electric shock, or injury to persons.

1.2 This is a reference standard in which the requirements are to be applied when specifically referenced by other standards or product safety requirements.

1.3 These requirements address the risks unique to product hardware controlled by software in programmable components.

1.4 These requirements are intended to supplement applicable product or component standards and requirements, and are not intended to serve as the sole basis for investigating the risk of fire, electric shock, or injury to persons.

1.5 These requirements are intended to address risks that occur in the software or in the process used to develop and maintain the software, such as the following:

- a) Requirements conversion faults that cause differences between the specification for the programmable component and the software design;
- b) Design faults such as incorrect software algorithms or interfaces;
- c) Coding faults, including syntax, incorrect signs, endless loops, and other coding faults;
- d) Timing faults that cause program execution to occur prematurely or late;
- e) Microelectronic memory faults, such as memory failure, not enough memory, or memory overlap;
- f) Induced faults caused by microelectronic hardware failure;
- g) Latent, user, input/output, range, and other faults that are only detectable when a given state occurs; and
- h) Failure of the programmable component to perform any function at all.

1.6 Product standard requirements may amend or supersede the requirements in this standard, as appropriate.

2 Definitions of Terms Used

2.1 For the purpose of this standard, the following definitions apply.

2.2 APPLICATION-SPECIFIC INTEGRATED CIRCUIT (ASIC) – An electronic device comprised of many transistors and other semiconductor components which integrate standard cells and arrays from a library into one piece of silicon intended for a particular use.

2.3 BUILT-IN TEST – A design method that allows a product to test itself by adding logic for test signal generation and analysis of test results.

2.4 CENTRAL PROCESSING UNIT (CPU) – The unit of a computing and controlling system that includes the circuits controlling the interpretation of instructions and their execution.

2.5 CRITICAL SECTION – A segment of the software that is intended to perform the functions that address or control risks.

2.6 DATA – A representation of facts, concepts, or instructions in a manner suitable for storage, communication, interpretation, or processing.

2.7 DESIGN – The process of defining the software architecture, components, modules, interfaces, test approach, and data for a software system to satisfy specified requirements.

2.8 ELECTRONICALLY ERASABLE PROGRAMMABLE READ ONLY MEMORY (EEPROM) – A reprogrammable read-only memory in which cells may be erased electrically and in which each cell is capable of being reprogrammed electrically.

2.9 EMBEDDED SOFTWARE – Software that is physically part of a product and whose primary purpose is to maintain some property or relationship between other components of the product in order to achieve the overall system objective.

2.10 ERASABLE PROGRAMMABLE READ ONLY MEMORY (EPROM) – A type of programmable memory device which can only be read and not altered under normal use. The memory is capable of being erased by ultraviolet light and reprogrammed.

2.11 ERROR – A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition.

2.12 FAIL-OPERATIONAL PROCEDURE – A procedure executed in the event that a failure has occurred which continues product operation but provides degraded performance or reduced functional capabilities.

2.13 FAIL-SAFE PROCEDURE – A procedure executed to maintain the Risks Addressed (RA) state of a product while transitioning into a non-operational mode.

2.14 FAILURE – The inability of a product or component to perform its specified function.

2.15 FAILURE MODE – The physical or functional manifestation of a failure.

2.16 FAILURE MODE TEST – A suite of tests that have been specifically developed based upon the failure modes that exist in a programmable component or product.

2.17 **FAULT** – A deficiency in a product or component which is capable of, under some operational conditions, contributing to a failure.

2.18 **FAULT-TOLERANT** – The capability of software to provide continued correct execution in the presence of a defined set of microelectronic hardware and software faults.

2.19 **FLASH MEMORY** – A type of non-volatile memory which is capable of being erased electrically and reprogrammed, but only in blocks, as opposed to one byte increments.

2.20 **HAZARD** – A potential source of physical injury to persons.

2.21 **INSTRUCTION** – A statement that specifies an operation to be performed and that is capable of identifying data involved in the operation.

2.22 **INTEGRITY** – The degree to which a system or component prevents unauthorized access to, or modification of, computer programs or data.

2.23 **MICROCONTROLLER** – A microcomputer chip capable of executing instructions.

2.24 **MICROELECTRONICS** – Monolithic, hybrid, or module circuits, where the internal connections are not accessible, which satisfy one or more of the following criteria:

- a) More than 1000 gates are used in digital mode;
- b) More than 24 functionally different external electrical connections are available for use; or
- c) The functions can be reprogrammed.

The circuits are capable of functioning in the analogue mode, digital mode, or a combination of the two modes. Examples of microelectronics include: ASICs, ROMs, RAMs, PROMs, EPROMs, PALs, and PLDs.

2.25 **NEGATIVE CONDITION BRANCH** – A code construct implementing an alternate path of program control flow intended to provide a deterministic path from the normal operating path to the Risks Addressed (RA) state whenever the normal operating path cannot be taken.

2.26 **NON-DETERMINISM** – The state in which output is not predictable in advance because data dependencies effect task execution order.

2.27 **NON-NETWORKED EMBEDDED SOFTWARE** – Embedded software that executes on a single microprocessor/microcontroller or on redundant microprocessors/microcontrollers residing in the same physical enclosure.

2.28 **NON-TERMINATING STATE** – Any state of the software characterized by the execution of instructions that do not permit exiting from that state.

2.29 **NON-VOLATILE MEMORY** – A storage device not alterable by the interruption of the power to the memory, for example; ROM, FLASH, PROM, EPROM, and EEPROM.

2.30 OFF-THE-SHELF (OTS) SOFTWARE – Software which is made available on the market and intended for broad distribution, generally without the need for additional tailoring, including, but not limited to, operating system software, runtime libraries, real-time executives, kernel primitives, shareable re-entrant library-type routines, and the like. This includes software that has been developed by another developer.

2.31 OPERATIONAL TESTING – Evaluations on a product or component using an input profile representative of its operational environment.

2.32 PARAMETER SETTINGS – A finite collection of values assigned to variables to select, enable or disable known pre-existing function(s) or features of the software.

2.33 PARTITIONING – The act of segregating the functions of a system into verifiably distinct and protected collections of functions.

2.34 POST-RELEASE TESTING – All testing for changes implemented after the final production software is released.

2.35 PROCEDURE – A course of action taken to perform a task.

2.36 PROCESS – A sequence of steps performed to produce a given result.

2.37 PRODUCT – An instrument, apparatus, implement, or machine intended for personal, household, industrial, laboratory, office, or transportation use.

2.38 PRODUCT HARDWARE – Any hardware that is part of a product and that provides electrical, mechanical, and (or) electromechanical functions.

2.39 PROGRAMMABLE COMPONENT – Any microelectronic hardware that can be programmed in the design center, the factory, or in the field. Here the term "programmable" is taken to be "any manner in which one can alter the software wherein the behavior of the component can be altered."

2.40 PROGRAMMABLE COMPONENT CONFIGURATION – The configuration of the microelectronic hardware and software that must be present and functioning for the software to operate and control the equipment as intended. This configuration includes, but is not limited to, the operating system or executive software, communication software, microcontroller, network, input/output hardware, any generic software libraries, database management and user interface software.

2.41 PROGRAMMABLE READ-ONLY MEMORY (PROM) – A memory chip whose contents are capable of being programmed by a user or manufacturer for a specific purpose.

2.42 RANDOM ACCESS MEMORY (RAM) – Data storage where the time required for data access is independent of the location of the data most recently obtained or placed in storage.

2.43 READ ONLY MEMORY (ROM) – Data storage that stores data not alterable by computer instructions.

2.44 RISK – The potential for fire, electric shock, or injury to persons associated with the intended use of the product as specified by the product safety requirements.

2.45 RISKS ADDRESSED (RA) STATE – A state that is characterized by all reasonably foreseeable risks associated with the intended use of the product being addressed such that there is no longer a likelihood of the risk.