


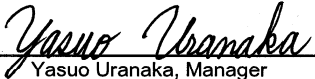
# **MELTAC Platform Basic Software Program Manual**


**Non Proprietary Version**


**January 2011**

**© 2011 MITSUBISHI ELECTRIC CORPORATION  
All Rights Reserved**

Prepared:  1/28/2011  
Yukiko Hirano, Engineer  
Control & Protection Systems Section Date

Reviewed:  1/28/2011  
Yasuo Uranaka, Manager  
Control & Protection Systems Section Date

Approved:  1/29/2011  
Hozumi Kadohara, Section Manager  
Control & Protection Systems Section Date

Approved:  1/31/2011  
Masanori Sugita, QA Manager  
Energy Systems Center Date

## Signature History

	Rev.0	Rev.1			
Prepared	Yukiko Hirano	Yukiko Hirano			
Reviewed	Yasuo Uranaka	Manabu Taniguchi			
Approved	Hozumi Kadohara	Hozumi Kadohara			
	Masanori Sugita	Masanori Sugita			

**Revision History**

Revision	Date	Page (section)	Description
0	Apr 2010	All	Original issue
1	2010 September	i (Abstract)	“MELTAC Basic Software” is changed to “MELTAC Platform Basic Software”.
		i (Abstract)	Document number is corrected. (“MUAP-0717” -> “MUAP-07017”)
		vi	Acronyms are added.
		1 (1.1)	Description of the purpose of the SPM is added
		1-2 (1.2)	Description of the scope of the SPM is added
		4-5 (2.2)	Description of the Software Life Cycle phases is added.
		5 (2.3)	Description of Software Life Cycle Process Design Outputs is added and modified.
		6 (3.0)	Figure 3.0-1 is modified.
		7 (3.1.1)	Description for applying MELTAC Platform to a specific plant is added.
		7 (3.1.1.1)	Description of MELTAC Software is added.
		13 (3.1.4.2.1)	Descriptions of (1) to (3) are modified.
		13 (3.1.4.2.1)	Descriptions of (6) to (9) are added.
		14 (3.1.4.2.1)	Description of (4) is modified.

Revision	Date	Page (section)	Description
1	2010 September	14-15 (3.1.4.4)  15 (3.1.4.6)  18 (3.2.3)  18 (3.2.4)  21 (3.2.5.1.1)	Description of personnel (1) and (2) are modified.  “is performed in accordance with” is changed to “plan complies with”.  Description of (4) for reusing pre-existing software is added.  Description of (3) for resolving risks is added.  Description for the feasibility examination is added.

Revision	Date	Page (section)	Description
1	2010 September	<p>35 (3.2.5.4)</p> <p>36 (3.3.1)</p> <p>36 (3.3.2)</p> <p>36 (3.3.3)</p> <p>37 (3.3.4.2)</p> <p>38 (3.3.4.3)</p> <p>40 (3.3.4.4)</p> <p>40 (3.3.4.5)</p> <p>41 (3.4.1)</p> <p>41 (3.4.1)</p>	<p>“is performed in accordance with” is changed to “plan complies with”.</p> <p>“written procedures” is changed to “MELCO internal procedures”.</p> <p>Description of the responsibilities of the QA Section is modified.</p> <p>“as prescribed in” is changed to “in accordance with”.</p> <p>Wording is corrected.</p> <p>Description of record keeping is modified.</p> <p>Description of tools for QA work is added.</p> <p>“is performed in accordance with” is changed to “plan complies with”.</p> <p>“SIP” is changed to “SIntP”.</p> <p>Descriptions of Software Integration and Software Installation are added.</p>

Revision	Date	Page (section)	Description
1	2010 September	<p>42 (3.4.3.3)</p> <p>42 (3.4.3.4)</p> <p>42 (3.4.4)</p> <p>45 (3.5.3.2)</p> <p>46 (3.5.4)</p> <p>47 (3.6.1)</p> <p>47 (3.6.1)</p> <p>47 (3.6.1)</p> <p>48 (3.6.3.1)</p> <p>49 (3.6.3.4)</p> <p>50 (3.6.4)</p>	<p>Description of Integration test result is added.</p> <p>Section 3.4.3.4 Method / Tools is added.</p> <p>“is performed in accordance with” is changed to “plan complies with”.</p> <p>Section 3.5.3.2 Method / Tools is added.</p> <p>“is performed in accordance with” is changed to “plan complies with”.</p> <p>“SMP” is changed to “SMaintP”.</p> <p>Description of maintenance manual is added.</p> <p>Description of retirement of software is added.</p> <p>Description of the compliance for RG1.152 is added.</p> <p>Section 3.6.3.4 Method / Tools is added.</p> <p>“is performed in accordance with” is changed to “plan complies with”.</p>

Revision	Date	Page (section)	Description
1	2010 September	<p>51 (3.7.1)</p> <p>51 (3.7.1)</p> <p>51 (3.7.1, 3.7.2)</p> <p>51 (3.7.2)</p> <p>51-52 (3.7.3)</p> <p>52-53 (3.7.4, 3.7.4.1)</p> <p>52 (3.7.4.2)</p> <p>53 (3.7.4.2)</p> <p>53 (3.7.5)</p> <p>54 (3.8.3)</p> <p>54 (3.8.4)</p>	<p>“STP is changed to “STrngP”.</p> <p>Description of training for development engineers is added.</p> <p>“Engineering Tool” is changed to “MELTAC Engineering Tool”.</p> <p>Description of training for development engineers is added.</p> <p>Section 3.7.3 Measurement is added.</p> <p>Descriptions of training manual, lecture classes and hands-on training are added.</p> <p>Section 3.7.4.1 Training for Customers is added.</p> <p>Section 3.7.4.2 Training for Development Engineers is added.</p> <p>“is performed in accordance with” is changed to “plan complies with”.</p> <p>“written procedures” is changed to “MELCO internal procedures”.</p> <p>Section 3.8.4 Security is added.</p>



Revision	Date	Page (section)	Description
1	2010 September		

Revision	Date	Page (section)	Description
1	2010 September		

Revision	Date	Page (section)	Description
1	2010 September		

Revision	Date	Page (section)	Description
1	2010 September	79 (3.11.1)	Description of the configuration control attributes (including security attributes) is added.
		80 (3.11.3.2)	Explanation about the procedure of configuration control is added.
		80 (3.11.3.2.1)	Explanation about the responsibility of each section is added.
		81 (3.11.3.2.2)	Spelling of section title is modified.

Revision	Date	Page (section)	Description
1	2010 September	<p>83 (3.11.4)</p> <p>84 (3.12.3)</p> <p>84 (3.12.5)</p> <p>84 (3.12.6)</p> <p>85-86 (4.0)</p> <p>87-92 (APPENDIX A)</p> <p>93-288 (APPENDIX B)</p> <p>289-300 (APPENDIX C)</p>	<p>Expression is modified. (no change about the content)</p> <p>Description of the compliance to IEEE Std 829-1983 is added.</p> <p>Section 3.12.5 (Record keeping) is added.</p> <p>Expression is modified. (no change about the content)</p> <p>Item 5 and 25 are added.</p> <p>APPENDIX A is added.</p> <p>APPENDIX B, B-1 and B-2 are added.</p> <p>APPENDIX C is added.</p>
2	2011 January	<p>All</p> <p>All pages of Section 2</p> <p>All pages of Section 3</p> <p>All pages of Section 4</p> <p>All pages of Appendix A</p> <p>All pages of Appendix B</p>	<p>Following items are revised to reflect feedback received from the NRC at the public meeting held on December 10, 2010 and RAI 07.01-24. Most of the contents of revision 1 were revised; therefore the revision bar is omitted. The description of implementation stage is revised.</p> <p>Change process control flow is added.</p> <p>Completely revised based on the comments and RAI 07.01-24 from the NRC.</p> <p>Reference is revised to add the lacked regulation and MELCO internal procedure</p> <p>Definitions are revised to follow the definitions in accordance with IEEE Std 610.12-1990.</p> <p>Detail compliance matrix of BTP 7-14 is deleted, and compliance matrixes for all IEEE which are endorsed by RG are added.</p>

## **Abstract**

The MELTAC Platform Basic Software is the low-level software that operates the MELTAC controllers. This software consists of initialization code, device drivers, communication layers, function blocks, diagnostics, etc. for the MELTAC platform. As safety system software, this software is designed to be as simplistic and deterministic as possible. The MELTAC Platform Basic Software consists of code residing in conventional computer memory devices and functions embedded in programmable logic devices, such as Field Programmable Gate Arrays.

The MELTAC Platform Basic Software was developed under a Japanese Nuclear QA program. Subsequently, MELCO initiated a QA program (QAP) that fully complies with 10 CFR 50 Appendix B (Reference 2). The software developed prior to the 10 CFR 50 Appendix B-based QAP was re-evaluated to demonstrate that it is equivalent to software generated under a 10 CFR 50 Appendix B program. The MELTAC Platform Basic Software re-evaluation is documented in the MELTAC Re-evaluation Program Report (MRP Report) (JEXU-1022-6301).

This Software Program Manual (SPM) provides the generic plans that are followed under MELCO's 10 CFR 50 Appendix B- based QAP for all activities related to the MELTAC Platform Basic Software Life Cycle conducted after the MELTAC Re-evaluation Program (MRP). This includes maintenance of the pre-MRP MELTAC Platform Basic Software, development of new software, and software retirement. Some of these generic plans require additional project specific information or activities, as defined within this document. Project specific content is documented in a Project Plan for each project. For the MELTAC Platform Basic Software, a project may be as small as a modification to an existing software module or as large as development of a completely new CPU Module and/or data communication network.

This document does not address the process followed for developing and maintaining application software for specific applications of the MELTAC controller. The Application Software Life Cycle is addressed in project specific documents, such as the "US-APWR Software Program Manual" (Application SPM) (MUAP-07017). The Application SPM also addresses integration of the Application Software with the Basic Software on the target hardware, validation of the fully integrated system, and post delivery installation and operations.

## Table of Contents

1.0 INTRODUCTION.....	1
1.1 Purpose .....	1
1.2 Scope .....	1
2.0 PROGRAM OVERVIEW .....	3
2.1 Planning Stage.....	3
2.2 Implementation Stage .....	4
2.3 Software Life Cycle Process Design Outputs .....	5
3.0 SOFTWARE DEVELOPMENT PROGRAM .....	6
3.1 Software Management Plan .....	9
3.1.1 Purpose, Applicability and Scope .....	9
3.1.2 Organization / Responsibilities .....	12
3.1.3 Oversight.....	18
3.1.4 Execution / Methods.....	19
3.2 Software Development Plan .....	27
3.2.1 Purpose / Applicability .....	27
3.2.2 Organization / Responsibilities .....	27
3.2.3 Oversight.....	29
3.2.4 Risks .....	29
3.2.5 Execution / Methods.....	29
3.3 Software Quality Assurance Plan .....	53
3.3.1 Purpose and Scope .....	53
3.3.2 Organization / Responsibilities .....	53
3.3.3 Security .....	54
3.3.4 Execution / Methods .....	54
3.4 Software Integration Plan .....	63
3.4.1 Purpose .....	63
3.4.2 Organization / Responsibilities .....	63
3.4.3 Execution / Methods.....	63
3.4.4 Standards .....	65
3.5 Software Installation Plan .....	66
3.5.1 Purpose .....	66
3.5.2 Organization / Responsibilities .....	66
3.5.3 Execution / Methods.....	66
3.5.4 Standards .....	70
3.6 Software Maintenance Plan.....	71
3.6.1 Purpose .....	71
3.6.2 Organization / Responsibilities .....	71
3.6.3 Execution / Methods.....	72
3.6.4 Standards .....	74
3.7 Software Training Plan.....	75
3.7.1 Purpose .....	75
3.7.2 Organization / Responsibilities .....	75
3.7.3 Measurement .....	76
3.7.4 Procedure .....	76
3.7.5 Execution / Methods.....	76
3.7.6 Standards .....	78

---

3.8 Software Operations Plan .....	79
3.8.1 Purpose .....	79
3.8.2 Organization / Responsibilities .....	79
3.8.3 Execution / Methods .....	79
3.8.4 Security .....	80
3.8.5 Standards .....	80
3.9 Software Safety Plan .....	81
3.9.1 Purpose .....	81
3.9.2 Organization / Responsibilities .....	82
3.9.3 Risks .....	84
3.9.4 Execution / Methods .....	84
3.9.5 Software Safety Analyses Activities .....	85
3.9.6 Standards .....	102
3.10 Software Verification and Validation Plan .....	103
3.10.1 Purpose .....	103
3.10.2 Organization / Responsibilities .....	103
3.10.3 Risks .....	106
3.10.4 Execution / Methods .....	106
3.10.5 V&V Reporting Documentation Requirements .....	130
3.10.6 V&V Administrative Requirements .....	132
3.10.7 Methods / Tools .....	133
3.10.8 Standards .....	133
3.11 Software Configuration Management Plan .....	135
3.11.1 Purpose .....	135
3.11.2 Management .....	135
3.11.3 Execution / Methods .....	138
3.11.4 Standards .....	146
3.12 Software Test Plan .....	147
3.12.1 Purpose .....	147
3.12.2 Organization / Responsibilities .....	147
3.12.3 Execution / Methods .....	147
3.12.4 Measurement .....	158
3.12.5 Methods / Tools .....	158
3.12.6 Record Keeping .....	158
3.12.7 Standards .....	158
4.0 REFERENCES .....	159
APPENDIX A DEFINITION .....	162
APPENDIX B CONFORMANCE TO BTP 7-14 AND IEEE STANDARDS .....	170
APPENDIX C RG 1.152 REV 2 COMPLIANCE .....	237



---

## List of Tables

Table 2.1-1 Software Development Plans .....	3
Table 3.2-1 Roles & Qualification Requirements of Design Section Personal (1/2) .....	28
Table 3.2-1 Roles & Qualification Requirements of Design Section Personal (2/2) .....	29
Table 3.2-2 Inputs and outputs in each phase of Development (1/3) .....	34
Table 3.2-2 Inputs and outputs in each phase of Development (2/3) .....	35
Table 3.2-2 Inputs and outputs in each phase of Development (3/3) .....	36
Table 3.2-3 Minimum Information Required in the Platform Specification (1/2).....	38
Table 3.2-3 Minimum Information Required in the Platform Specification (2/2).....	39
Table 3.2-4 Minimum Information Required in the Software Specification .....	42
Table 3.2-5 Minimum Information Required in the Program Specification.....	43
Table 3.2-6 Minimum Information Required in the FPGA Specification.....	45
Table 3.2-7 Requirements for Writing Source Code .....	47
Table 3.3-1 Document Types.....	61
Table 3.9-1 Potential Hazards .....	89
Table 3.9-2 Acceptance Criteria for Function .....	90
Table 3.9-3 Acceptance Criteria for Phase .....	91
Table 3.10-1 V&V Independence .....	105
Table 3.12-1 Alignment with IEEE Std 1012-1998 Testing Activities.....	148
Table 3.12-2 Alignment with IEEE Std 829-1983 Software Test Document.....	155

---

## List of Figures

Figure 3.0-1 Overview of Software Life Cycle Plan .....	6
Figure 3.0-2 Overview of Change Process .....	7
Figure 3.1-1 Organization Chart .....	13
Figure 3.2-1 MELTAC Platform Basic Software Life Cycle .....	33
Figure 3.2-2 Relationship between General Software Specifications and Program Specifications.....	41
Figure 3.2-3 Relationship between General Hardware Specifications and FPGA Specifications.....	45
Figure 3.5-1 Basic Software Installation Process .....	69
Figure 3.10-1 Organizational Structure and Responsibilities / Authorities.....	106
Figure 3.10-2 Overview of MELTAC Platform Basic Software V&V Activities, Tasks and Outputs .....	108
Figure 3.12-1 FPGA Unit V&V test workflow .....	151

## List of Acronyms

ANSI	American National Standards Institute
ASME	American Society of Mechanical Engineers
BTP	Branch Technical Position
CAD	Computer Aided Design system
CEAS	MELCO Corporate Electronic Archive System
CFR	Code of Federal Regulations
DFT	Design For Testability
EEPROM	Electrically Erasable Programmable Read Only Memory
ESC	Energy Systems Center in Mitsubishi Electric Corporation
FPGA	Field Programmable Gate Array
F-ROM	Flash Electrically Erasable Programmable Read Only Memory
HSI	Human System Interface
ID	Identification
IEEE	Institute of Electrical and Electronics Engineers
IO	Input/Output
LED	Light Emitting Diode
MELCO	Mitsubishi Electric Corporation
MELTAC	Mitsubishi Electric Total Advanced Controller
MRP	MELTAC Re-evaluation Program
NaN	Not a Number
NRC	U.S. Nuclear Regulatory Commission
QA	Quality Assurance
QAP	Quality Assurance Program
RG	Regulatory Guide
RTM	Requirements Traceability Matrix
SCMP	Software Configuration Management Plan
SDP	Software Development Plan
SInstP	Software Installation Plan
SIntP	Software Integration Plan
SMP	Software Management Plan
SMaintP	Software Maintenance Plan
SOP	Software Operation Plan
SPM	Software Program Manual
SQAP	Software Quality Assurance Plan
SSP	Software Safety Plan
STP	Software Test Plan
STrngP	Software Training Plan
SVVP	Software Verification and Validation Plan
UV-ROM	Ultra-Violet Erasable Programmable Read Only Memory
V&V	Verification and Validation
VDU	Visual Display Unit

## 1.0 INTRODUCTION

### 1.1 Purpose

The purpose of this Software Program Manual (SPM) is to document the key elements of the MELTAC Platform Basic Software (“the Basic Software”) Life Cycle process, as defined in MELCO’s internal Software Life Cycle procedures. Therefore, many of the life cycle descriptions in this document are written in present tense, as a reflection of the current procedures. However, all descriptions are applicable to the entire 10 CFR 50 Appendix B-based Software Life Cycle (i.e. present and future activities). Therefore, the present tense descriptions shall be considered mandatory (“shall”) requirements for all future Basic Software Life Cycle activities. Revisions to MELCO’s internal Basic Software Life Cycle procedures are permitted, and those revisions may change the implementation details of a specific life cycle process. However, the key elements of the Basic Software Life Cycle, as documented within this SPM, shall always be maintained.

This Software Program Manual describes the program measures incorporated to:

- Ensure that the MELTAC Platform Basic Software attains a level of quality commensurate with its importance to safety functions
- Ensure that the MELTAC Platform Basic Software performs the required safety functions correctly
- Conform to established technical and documentation requirements, conventions, rules, and industry standards

The MELTAC Platform Basic Software is common software that does not depend on application. The Basic Software controls the input-output operation with the modules that are incorporated in MELTAC (such as the Control Network Interface Module or Bus Master Module or etc), execution of application and the self-diagnosis. The Basic Software is designed for high reliability and determinacy required for Safety System.

The MRP Report (JEXU-1022-6301) assesses the legacy MELTAC Basic Software, which was developed prior to MELCO’s 10 CFR 50 Appendix B QAP and prior to this SPM. The MRP demonstrates that the legacy MELTAC product is equivalent to a product developed under a 10 CFR 50 Appendix B program, and is therefore suitable to be maintained under this SPM. This SPM applies to maintenance of the current MELTAC Basic Software and new development within the MELTAC product family.

This SPM describes the plans under which the MELTAC Platform Basic Software is developed and maintained throughout its entire life cycle from cradle (applicable to new development) to grave. The program follows the guidance of BTP 7-14 (Reference 1) and as such is designed to produce and maintain a high quality product that is suitable for use in U.S. nuclear facility safety systems. This program also follows the guidance in RG 1.152 for a secure development environment.

### 1.2 Scope

This SPM applies only to MELTAC Platform Basic Software. Applications are described in the Application SPM (MUAP-07017). The MELTAC Platform Basic Software is considered Class1E safety-related software. This SPM covers safety-related software only; non-safety-related software is out of scope.

BTP 7-14 contains three major document divisions: Software Life Cycle Process Planning, Software Life Cycle Process Implementation, and Software Life Cycle Process Design Outputs.

- **Software Life Cycle Process Planning:** In Software Life Cycle Process Planning, plans are developed that describe, at a high-level, all of the necessary ingredients for the software development effort to succeed. The plans span the range from product inception to end-of-life. Software Life Cycle Planning represents the 'upfront' work of the software development process. RG 1.152 refers to the planning phase of the software life cycle as the Concept Phase.

The plans ensure that the development effort proceeds along an acceptable path as defined by standard, state-of-the-art software development processes and procedures. The plans address all testing, verification, validation, and quality assurance requirements of the development process.

This SPM documents the results of the generic planning stage, which is generically applicable to all projects. Where project specific planning details are needed, the requirements to document those details are contained within the generic plans contained within this SPM. Those project specific details are documented within specific Project Plans.

- **Software Life Cycle Process Implementation:** Software Life Cycle Process Implementation represents the effort required to actually produce the software product. Software Life Cycle Process Implementation begins with requirements gathering and proceeds through the software end-of-life in the operations and maintenance phases.

Software Life Cycle Process Implementation progresses through a series of general 'phases'. The requirements of each phase are defined by one or more plans of the planning stage. The work progresses through the use of specific working procedures.

- **Software Life Cycle Process Design Outputs:** The Software Life Cycle Process Design Outputs concerns itself with all of the elements produced by the process, including the final software codes and associated documents. Outputs are produced throughout the entire life cycle of the software development process.

## 2.0 PROGRAM OVERVIEW

The software development program is characterized by Software Life Cycle Process Planning, Software Life Cycle Process Implementation, and Software Life Cycle Process Design Outputs.

Whereas BTP 7-14 introduces the term ‘activity group’ to describe the activities within the life cycle development, this SPM uses the term ‘phase’ to describe these activities in order to maintain consistency with other regulatory guides, standards, and product documentation.

### 2.1 Planning Stage

Table 2.1-1 lists the software plans included in this MELTAC Platform Basic Software Program Manual and the section number for the plan.

**Table 2.1-1 Software Development Plans**

Plan name	Section number
Software Management Plan (SMP)	3.1
Software Development Plan (SDP)	3.2
Software Quality Assurance Plan (SQAP)	3.3
Software Integration Plan (SIntP)	3.4
Software Installation Plan (SInstP)	3.5
Software Maintenance Plan (SMaintP)	3.6
Software Training Plan (STrngP)	3.7
Software Operations Plan (SOP)	3.8
Software Safety Plan (SSP)	3.9
Software Verification and Validation Plan (SVVP)	3.10
Software Configuration Management Plan (SCMP)	3.11
Software Test Plan (STP)	3.12

## 2.2 Implementation Stage

The implementation stage is characterized by the following phases in keeping with the Software Life Cycle of the MELTAC Platform Basic Software.

(1) Requirements Phase

Identify the software requirements that should be incorporated into the MELTAC Platform including functional requirements, characteristic features requirements, reliability requirements, and requirements from safety-related standards. And then, document the identified requirements that will be inputs to the design phase.

(2) Design Phase

Incorporate all software requirements into the software design. Describe the design outputs in appropriate documents (Software Specification, Program Specification, FPGA Specification).

(3) Implementation Phase

Create source codes in accordance with the software design documents. Perform the software unit test to assure that the source code is faithfully and correctly implemented to the software design.

(4) Test Phase

After completing unit tests, the compiled software modules are integrated with the MELTAC Platform hardware.

An integration test is performed by the V&V team as part of the validation of the MELTAC Platform Basic Software.

Completion of the integration test marks the end of the code development portion of the MELTAC Platform Basic Software.

(5) Installation Phase

Install the developed MELTAC Platform Basic Software to plant specific systems.

Each project is application-specific, but Basic Software installation process is designated in this SPM. Application Software installation process is described in the Application SPM.

(6) Operations and Maintenance Phase

Manage the following activities for the MELTAC Platform during site-specific operation.

- Nonconformance procedure and corrective action  
Change process of the MELTAC Platform associated with corrective actions is controlled with the process described in the SDP (Section 3.2).
- Training
- Software retirement

Outputs are produced as appropriate in each phase including phase-by phase progress report.

### 2.3 Software Life Cycle Process Design Outputs

Software Life Cycle Process Design Outputs include the following typical output products:

- (1) Requirements Specifications
  - Corresponding document: Platform Specification
- (2) Software Design Specifications
  - Corresponding document: Software Specification, Program Specification, FPGA Specification
- (3) Source Code Listings
- (4) Unit Test Documents
  - Corresponding document: Unit V&V Test Specification / Unit V&V Test Report
- (5) Integration Test Documents
  - Corresponding document: Integration V&V Test Specification / Integration V&V Test Report
- (6) Installation Configuration Tables
  - Corresponding document: Software Installation Drawing
- (7) Maintenance Manuals
- (8) Training Manuals

Documents such as Verification Reports, Software Safety Analysis, RTM (Requirements Traceability Matrix) and Configuration Management Reports may be generated on a recurring basis during each phase.



### 3.0 SOFTWARE DEVELOPMENT PROGRAM

Figure 3.0-1 illustrates the planning phase of the Basic Software life cycle and the relationship of the plans to each phase. The software development proceeds through the phases starting from the concept phase and proceeds to the following phases.

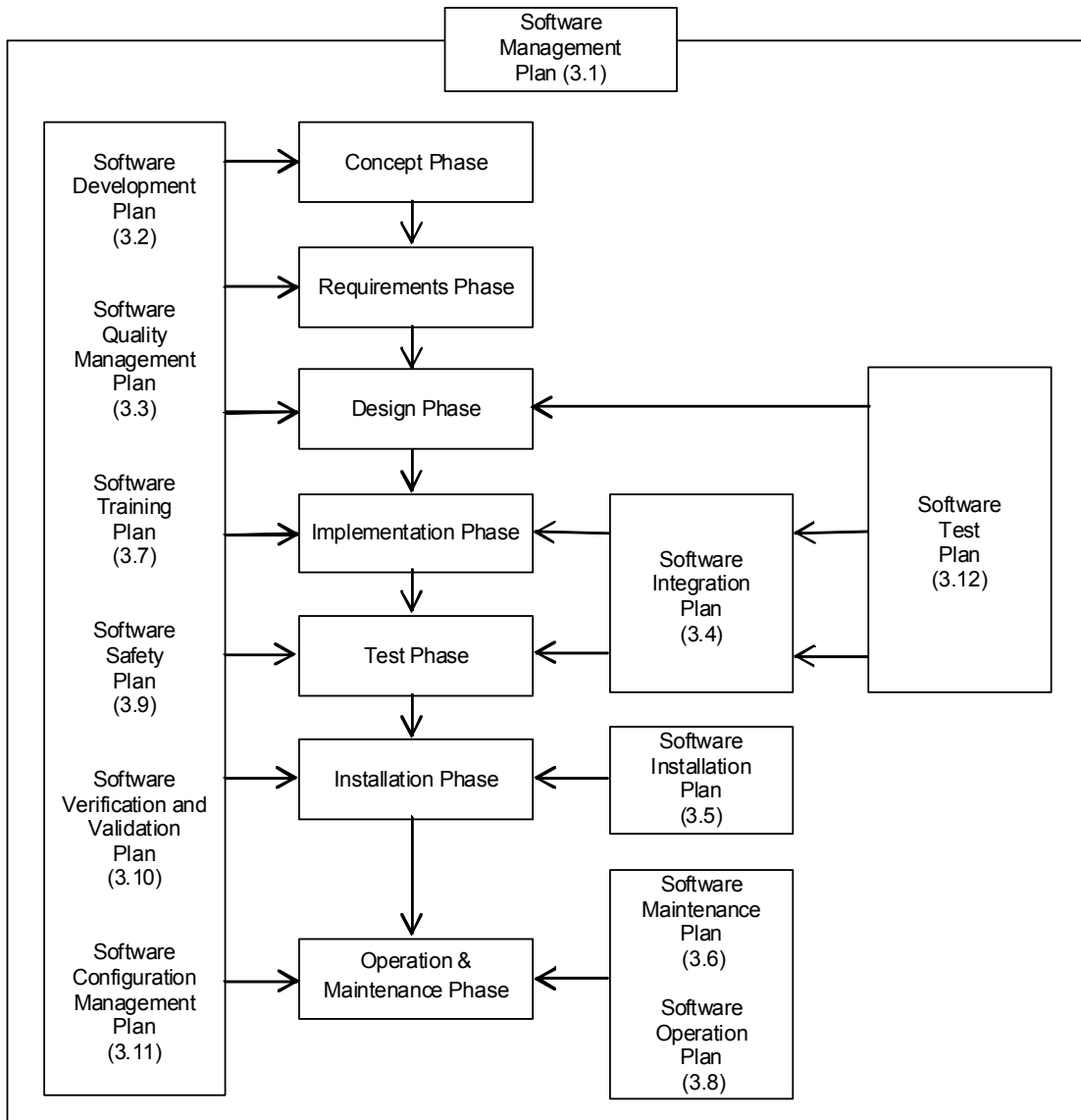
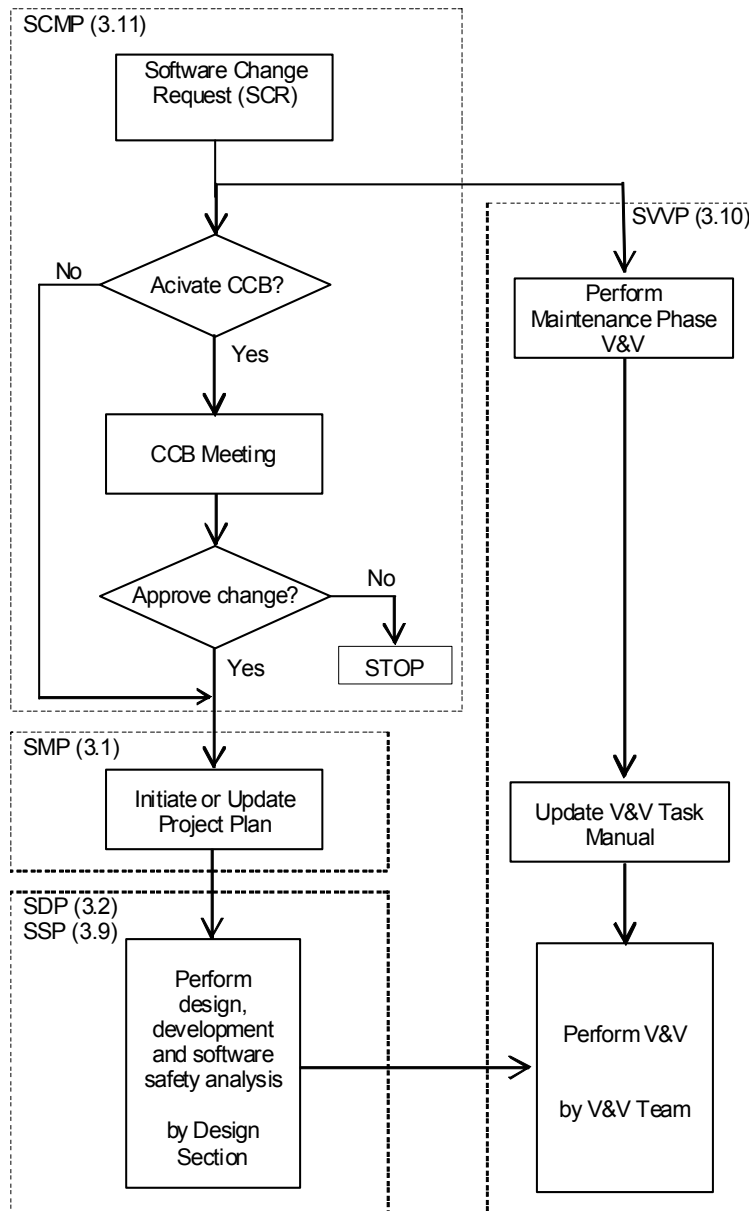


Figure 3.0-1 Overview of Software Life Cycle Plan



**Figure 3.0-2 Overview of Change Process**

The MELTAC Platform is a pre-established set of hardware and software items specified via the Platform Specification and no major changes are expected (e.g., development of a new platform). However, minor changes may be necessary from time to time to maintain or enhance platform features.

In this SPM, the term “change” includes one or more of the following meanings:

- (1) design change (functional change)
- (2) maintenance change (restore or maintain required functions)

(3) document revision

Figure 3.0-2 shows the overview of change process.

Software changes are initiated by the Software Change Request (SCR) process described in the SCMP (Section 3.11 of this SPM).

After software change is initiated, the Design Section Manager shall determine whether the CCB is activated or not as described in the SCMP (Section 3.11). If the CCB is activated, the CCB shall evaluate the adequacy of the change and the impact on the existing functions of the MELTAC platform, and determine whether the change is acceptable or not as described in the SCMP (Section 3.11).

If the change is acceptable, the Design Section Manager shall initiate a Project Plan as described in the SMP (Section 3.1) and perform development and software safety analysis as described in the SDP and SSP (Section 3.2 and Section 3.9).

The V&V Team shall prepare a V&V Task Manual for each change project as described in the SVVP (Section 3.10).

### **3.1 Software Management Plan**

#### **3.1.1 Purpose, Applicability and Scope**

##### **3.1.1.1 Purpose**

It is commonly recognized that software development projects require a number of organized groups working in various degrees of independence. These groups must be assembled of personnel with appropriate qualifications and skill levels. The groups must be organized in a manner that allows for effective communication between the groups and effective oversight of the groups. The groups must be appropriately funded and staffed to meet the overall schedule goals of the development project. In order to fulfill their charter, certain groups such as Verification and Validation Teams or Quality Assurance Teams must be able to maintain a high degree of independence from the other development teams.

The purpose of this Software Management Plan (SMP) is to describe the following:

- organizational structure, roles and responsibilities of the personnel involved in the activities described in this Software Program Manual (SPM)
- development/maintenance project execution methods
- development/maintenance project oversight methods

##### **3.1.1.2 Applicability**

The MELTAC Platform is qualified and applied to the Protection and Safety Monitoring System, which includes the Reactor Protection System, Engineered Safety Feature Actuation System, Safety Logic System and Safety Grade Human System Interface (HSI) System for the US-APWR.

The software that runs on an application of the MELTAC Platform consists of the Basic Software and the Application Software.

- The Basic Software is included in the MELTAC Platform, and it encompasses the following configuration items:
  - (1) software stored in nonvolatile memory on the platform's main CPU Modules,
  - (2) firmware stored in nonvolatile memory on peripheral modules, and
  - (3) code embedded in FPGAs on the CPU and peripheral modules.
- The Application Software is the software that implements application-specific functions specified for plant systems, such as logic symbol interconnections, setpoints and constants. Application Software is stored in F-ROM on the CPU Module.

This SMP applies to project management, engineering management, development/maintenance, V&V and QA activities associated with the Basic Software.

[

]

### 3.1.1.3 Scope

The scope of a project that can affect the Basic Software includes activities in the development, installation, training, operations & maintenance, and retirement life cycle phases. A Project Plan is prepared before starting any project activities. A Project Plan may encompass multiple software changes and activities across multiple life cycle phases.

#### (1) Development

Since the Basic Software has been previously developed and dedicated as described in the MRP Report (JEXU-1022-6301), the scope of development within this Software Program Manual (and therefore this SMP) starts with a demand for a change to the previously developed Basic Software. Changes can be additions or deletions of functions, maintenance related changes, or changes to documentation. The Development Phase includes software integration and testing activities.

#### (2) Installation

Installation refers to the process of installing the developed Basic Software in the target hardware for a specific MELTAC Platform module. Installation is part of the manufacturing/production process. While the Installation of the Basic Software is a generic activity applicable to the generic modules of the MELTAC Platform, the array of MELTAC modules that are applied in a specific plant system project are application dependent.

#### (3) Training

Training is an activity to provide the knowledge for the customer to perform equipment startup/shutdown, routine periodic maintenance and testing, and trouble shooting of the MELTAC Platform, including the Basic Software.

Training consists of lectures and hands-on-training, and training materials are developed by the Design Section. The Design Section Manager is responsible for assigning trainers with thorough knowledge of the MELTAC Platform hardware and software or providing a trainer training program.

#### (4) Operations and Maintenance

Operations refer to executing the plant-specific operations and maintenance activities covered by the STrngP (Section 3.7). This phase of the life cycle is the responsibility of the customer. A significant component of operations is the security of the system, to minimize the potential for unauthorized changes and to detect them should they occur.

Maintenance refers to activities to report and correct any nonconformance which may be discovered in the Basic Software after product delivery, or changes requested by customers.

Any non-conformance in the Basic Software is corrected by MELCO via this SPM.

### (5) Retirement

Retirement refers to activities that assure obsolete Basic Software is (a) not installed in MELTAC modules during the production process and (b) not shipped to any customers.

### (6) Security

While the operations and maintenance phase puts special emphasis on physical and cyber security, security is an important component of each life cycle phase to ensure no unauthorized software is introduced into the platform. The physical and cyber security controls for each life cycle phase are described in the applicable program plan in this SPM.

## 3.1.2 Organization / Responsibilities

The Basic Software is developed through a combined effort of several technical sections within MELCO.

The ultimate responsibility for the development of the Basic Software lies with the "Design Section," which is responsible for assuring that it is structured, staffed and qualified to meet the rigorous and technical demands for developing and maintaining the Basic Software.

Development of the Basic Software is performed by the Design Section in conjunction with the "QA Section", "V&V Team" and "Manufacturing Department".

The QA Section is responsible for assuring all activities conducted by MELCO throughout the MELTAC product life cycle (including activities of the Design Section, V&V Team and Manufacturing Department) follow the required regulations, standards, this SPM, and internal MELCO policies and procedures. QA audits are conducted independently from any activities or assessments of the Design Section, V&V Team or Manufacturing Department. MELCO maintains a 10 CFR 50 Appendix B Quality Assurance Plan (QAP) that is implemented via NQA-1.

The V&V Team executes independent verification and validation activities in accordance with the Software V&V Plan (SVVP) described in the SVVP (Section 3.10). The V&V Team is responsible for assuring the quality and correctness of the Basic Software, including the portions of the software that are critical to safety functions.

The Manufacturing Department manufactures the MELTAC Platform hardware modules and performs installation of the Basic Software to each hardware module during the production process

The Organization Chart relating to MELTAC platform development and maintenance is described in Figure 3.1-1.

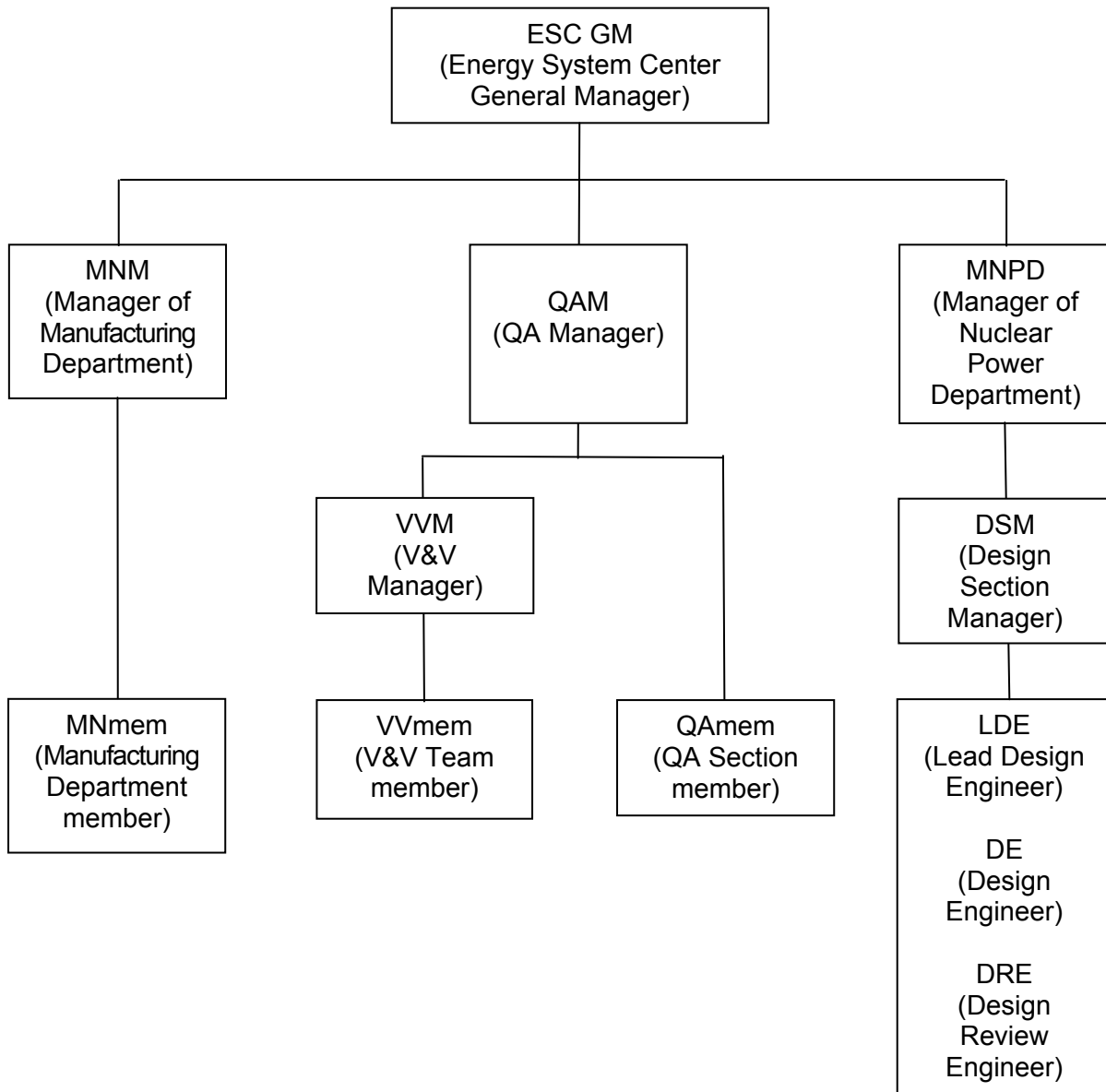


Figure 3.1-1 Organization Chart



The QA Section and V&V Team are independent from the Design Section and Manufacturing Department regarding management, budget and schedule. V&V Team independence is described in the SVVP (Section 3.10). Responsibilities of personnel in each section are as follows:

[

]

[

]

[

]

[

]

[

]

### **3.1.3 Oversight**

#### **(1) Development Progress**

The Design Section Manager or the person who is authorized by the Design Section Manager holds regular meetings to confirm progress according to the Project Plan.

Completion progress and project issues are captured in meeting minutes. The project Master Schedule is updated as a report of progress confirmation.

#### **(2) Development Quality**

The Design Section Manager periodically assesses the Risk Matrix, the Problem List, V&V Anomaly Reports and deviations from Project Plans to determine if there are any recurring issues in the life cycle process. Corrective actions are taken to prevent recurrence.

#### **(3) Security**

For the Basic Software, the following security management activities are performed:

#### a. Development

Source code and the development environment are installed in an area where only authorized development personnel can enter. (i.e. an area that requires a security card for entry and exit.)

The computer used for development is made inaccessible to unauthorized personnel. Only registered personnel with a prepared account can log in and access the computer.

Firewalls prevent access from outside of the office to the computers used for development

In order to reject unintended code, the Independent V&V Team verifies that design specifications and source code are matched.

#### b. Installation

Master data is stored in a data repository (locked by a security card) where access is restricted.

The installed data is verified to be the same as the master data, through cross-checking of both data.

#### c. Operation

The MELTAC Platform is installed in locked cabinets.

The Basic Software cannot be modified onsite.  
(Re-writing measures are not provided.)

### **3.1.4 Execution / Methods**

#### **3.1.4.1 Measurement**

By checking the following items, Project Plan progress, software quality and the effectiveness of this SMP and the SPM are monitored.

- (1) Progress rate of each document listed in the document list.
- (2) Number of items on the Problem List (open and closed)
- (3) Number and severity of V&V Anomaly Reports (open and closed)

#### **3.1.4.2 Procedures**

##### **3.1.4.2.1 Development/Maintenance Process**

---

All requirements regarding to development/Maintenance activity of the MELTAC Basic Software are described in the MELTAC Technical Report (MUAP-07005).

Prior to initiating development or maintenance activities, the Design Section Manager shall establish a Project Plan in accordance with, which shall include the following:

- Regulations, codes, and customer requirements
- Personnel assignments
- Project Master Schedule including milestones and task relationships and priorities
- Design control measures (procedures, prerequisites, and design verification method(s))
- Potential Risks.

The Design Section Manager, Lead Design Engineer, and project members shall execute the project as specified by the Project Plan.

The Project Plan shall state that the Independent V&V Team is established for the V&V activities and shall include the basic V&V schedule as provided by the V&V Manager.

The V&V Manager shall prepare an individual V&V Task Manual for the particular project. The Task Manual contains information that cannot be provided in the SPM, including a detailed schedule of the V&V activities and identification of V&V Team members assigned to the project. The V&V Manager and V&V Team members shall conduct V&V activities and tasks in accordance with the SVVP (Section 3.10) and the project-specific V&V Task Manual.

Major activities for the project management are as follows.

- (1) Manage progress of project through periodically held meetings. Project members report progress and any delays against the Master Schedule to the Design Section Manager. A Problem List is developed and maintained. Progress of items on the Problem List is reported. When a new problem occurs, it is added to the Problem List.
- (2) The Design Section Manager reports overall progress, open problems and V&V Anomaly Reports to the Manager of Nuclear Power Department at regular intervals, and receives review feedback.
- (3) The Design Section Manager revises the Project Plan when the items described in the Project Plan are changed, except when any changes are inconsistent with or in violation of this SPM and the implementing procedures. If any Deviations from this SPM are considered necessary by the Design Section Manager, a Deviation Request shall be submitted to the V&V Manager for review and approval in accordance with the SVVP (Section 3.10). Project Plan revisions are reviewed in the same manner as when it was first reviewed, and they are approved by the Design Section Manager.
- (4) The Project Plan is kept under configuration control using a Configuration Management Sheet as described in Section 3.11.3. The Configuration Management

Sheet is maintained by the Lead Design Engineer on the project and is approved by the Design Section Manager.

- (5) Support software (compiler, test tool, etc) and associated documentation are managed by a support software control list. A Design Engineer prepares the support software control list including
  - Software module name
  - Purpose for use
  - Scope
  - Revision
  - Person or Section in charge
  - Operating environmentAll support software in this list is kept under configuration control (Section 3.11.3).
- (6) The basis for not performing any “as applicable” activities specified in this SPM shall be documented in the Project Plan.

#### **3.1.4.2.2 Resolution of Identified Issues**

- (1) For all design documents, Design Review Engineers shall perform design reviews. The Design Review Engineer notes the review results in a comment sheet, and sends it to the Design Engineer who prepared the document. The Design Engineer examines the comments and incorporates them into the design document or provides justification for not incorporating a comment.
- (2) For all software design documents, the V&V Team implements Independent V&V. If there are any problems, the V&V Team provides feedback via V&V Anomaly Reports to the Design Section. The Design Section then either incorporates them into the design document via revision, or provides justification for not incorporating a comment.
- (3) When the V&V Team detects a nonconformance during a software test, a V&V Anomaly Report is sent to the Design Section. The Design Engineer carefully examines the V&V Anomaly Report, and corrects the software via the controlled change process described in this SPM, or provides justification for not making a software change. For example, if a V&V Anomaly Report detects a nonconforming condition where the software does not fulfill a specified requirement, it could be that the requirement is incorrect, whereby the Design Section would revise the requirements specification and not change the software.
- (4) If a post-release software nonconformance is detected by an internal report, or a customer report, a Nonconformance Report shall be initiated. The Design Section investigates the cause, the extent of condition, and identifies the necessary corrective actions to correct the nonconformance and to prevent recurrence. Corrective actions may include changing documents, changing the software, or both. Modifications are made in accordance with design change procedures. Additional details are described in the SDP (Section 3.2) and Section 3.11.3.2.5.

If nonconformance is associated with software that has been delivered to a customer, and the nonconformance has a potentially significant and adverse effect on nuclear



safety, a 10 CFR 21 Notice of Defect Report is issued to the customer(s) and the NRC.

#### **3.1.4.3 Budget**

- (1) The Design Section Manager determines the budget necessary for a project except QA and V&V activities, and specifies it in the Project Plan.
- (2) The budget for the QA Section and V&V Team are independent of the Design Section's budget. Budget of the QA Section shall be established by estimating cost necessary for all the annual QA activities at the beginning of the fiscal year. This budget is independent of those for each development project. Budget of the V&V Team shall be set by the V&V Manager upon formation of the team. The V&V Manager is authorized to acquire as much budget as necessary.

#### **3.1.4.4 Project Management Tools**

Projects shall be managed with the following tools, as a minimum:

[

]

#### **3.1.4.5 Tools for Development**

This section describes tools used for development of the MELTAC Platform.

[

]  
[

]

[

]

#### **3.1.4.6 Personnel**

- (1) The Project Plan shall specify the skills, roles and numbers of personnel required to execute the project. If personnel are added during a project, the Project Plan is revised to reflect the increased staffing.
- (2) For personnel working on design and V&V activities, the responsible Managers shall verify that each has the required skills for each activity and ensure that personnel are qualified before participating in the project. This will ensure that software safety and V&V personnel have sufficient skills and experience to effectively implement the SSP as described in Section 3.9 and the SVVP as described in Section 3.10.

#### **3.1.4.7 Standards**

This SMP complies with the following guidance and standards. Methods of compliance against the IEEE standards that are endorsed by Regulatory Guides are tabulated in Appendix B;

- (1) Clause 3 of IEEE Std 1074-1995 (Reference 23) which is endorsed by RG 1.173 (Reference 11)
- (2) Section 3.1.1 of NUREG/CR-6101 (Reference 3)

## **3.2 Software Development Plan**

### **3.2.1 Purpose / Applicability**

The purpose of the Software Development Plan (SDP) is to describe the activities required for the Basic Software Requirements, Design, Implementation and Test life cycle phases.

The MELTAC Platform, including the Basic Software, is a set of equipment and software items described in the MELTAC Technical Report (MUAP-07005). These items are pre-developed, qualified and dedicated as described in the MRP Report (JEXU-1022-6301). Therefore, going forward, this SDP applies to changes to the Basic Software configuration items identified in the SCMP (Section 3.11), including software documents, the processor software on the CPU Modules, firmware, and FPGAs on the peripheral Modules.

### **3.2.2 Organization / Responsibilities**

The Basic Software change activities described in this SDP are performed by the Design Section. The Design Section is comprised of a diverse group of engineers and experts with training or domain knowledge in computer control and software systems. The Design Section is responsible for changes to the Basic Software configuration items as described in this SDP.

Design Section personnel fill the roles of Lead Design Engineer (LDE), Design Engineer (DE), and Design Review Engineer (DRE) as described in the SMP (Section 3.1). The responsibilities assigned to these roles are described in this SDP.

The Design Section Manager (DSM) is responsible for assigning qualified and competent personnel to fill these roles as described in the SMP. The DSM fills the role of Approver of design outputs.

An overview of the roles and qualification requirements for activities assigned to the Design Section is shown in the following table:

The Design Section is also responsible for software safety analysis activities in each phase of the MELTAC Basic Software's software life cycle as described in the SSP (Section 3.9).

**Table 3.2-1 Roles & Qualification Requirements of Design Section Personnel (1/2)**

Role	Responsibilities	Qualification Requirements
Design Engineer	<p>Develop design documents.</p> <p>Confirm that design inputs are translated into design documents.</p>	<p>Have expertise and sufficient knowledge in the applicable design activities.</p> <p>Be certified by the Design Section Manager as being competent to perform the assigned design activities.</p>
Lead Design Engineer	<p>Review outcomes of document check (Design Management Chart and document checklist) performed by design personnel, and a design document to verify that the developed design document complies with the 10 CFR 50 Appendix B-based QAP and relevant corporate procedures.</p> <p>Confirm that design personnel are qualified.</p>	<p>Have good understanding of the software development procedures and have competence in determining if the design document complies with said procedures.</p> <p>Be certified by the responsible Design Section Manager as being competent to perform a check task.</p>
Design Review Engineer	<p>Perform an independent design review.</p>	<p>Have expertise in technical contents of the design document to be reviewed, and competence in assessing the technical adequacy of the design.</p> <p>Be individuals other than those who performed the original design but may be from same section.</p> <p>Be Certified by the responsible Design Section Manager as being competent to perform above task.</p>

**Table 3.2-1 Roles & Qualification Requirements of Design Section Personnel (2/2)**

Role	Responsibilities	Qualification Requirements
Approver	Confirm that activities performed by Design Section personnel, as described in this SDP, were performed by trained and qualified personnel, in accordance with implementing procedures.	Be the Design Section Manager or designee.  The Design Section Manager shall be appointed by the Manager of Nuclear Power Department, and the designee shall be appointed by the permanent Design Section Manager

### 3.2.3 Oversight

- (1) For how to monitor the progress of development, it is specified in the SMP (Section 3.1).
- (2) Errors that are detected during development phases (such as Review Comment and V&V Reports) shall be controlled as described in the SCMP (Section 3.11).
- (3) Pre-existing software developed under this or an equivalent process may be reused unchanged as part of developing new software.

### 3.2.4 Risks

Tracking, control and mitigation of identified project risks or problems shall be performed as described in the SMP (Section 3.1).

### 3.2.5 Execution / Methods

#### 3.2.5.1 Measurement

Following items shall be metrics of the Basic Software development.

- Number of comments identified through design reviews
- Number and severity level of V&V Anomaly Reports
- Open or recurring items on the Problem List (as described in the SMP) related to software development activities

#### 3.2.5.2 Procedures

The MELTAC Platform Basic Software Life Cycle model is based on the traditional “waterfall” model of software development. In this model, the output of each phase is used as input to the next phase. The structure of the waterfall model provides



distinguishable boundaries between the phases which are used as design review milestones, verification and validation activity points and auditing or review points for the quality assurance process.

The MELTAC Platform Basic Software Life Cycle is defined as follows. Note that certain V&V test activities are described here to show their relationship to design activities. A full description of all required V&V activities in terms of V&V Inputs, V&V Tasks, and V&V Outputs, for each life cycle phase, is provided in the SVVP (Section 3.10).

#### (1) Concept Phase

The MELTAC Platform is a pre-established set of hardware and software items specified via the Platform Specification. No major changes are expected (e.g., development of a new platform). However, minor changes may be necessary from time to time to maintain or enhance platform features.

Change Control Sheet may be initiated internally, by MELCO, or as a mechanism for initiating changes in the Operations and Maintenance phase as described in the SVVP (Section 3.10). The Change Control Sheet shall provide the necessary input information for evaluating potential changes to the Platform Specification in the Requirements Phase.

#### (2) Requirement Phase

The software requirements are specified in the Platform Specification, which is the highest-level document for the MELTAC platform. The Platform Specification defines the boundary between the functions of hardware and software. The minimum information required in the Platform Specification is described in Table 3.2-4 of this SDP.

#### (3) Design Phase

The Design Phase consists of the following activities:

- Software Design Activity

The software requirements specified in the Platform Specification are translated into a Software Specification. The Software Specification shall define the functional specifications for the entire Basic Software. The minimum information required in the Software Specification is described in Table 3.2-5 of this SDP.

- Program Design / FPGA Design Activity

The Software Specification is translated into detailed processor software specifications in one or more Program Specifications on a functional basis. The relationship between the Software Specification and multiple Program Specifications is illustrated in Figure 3.2-2. Each Program Specification shall define the structural specifications for each functional component/unit and the processing details for each component/unit. The minimum information required in a Program Specification is described in Table 3.2-6 of this SDP.

The FPGA Specification identified in the Hardware Specification is translated into the detailed FPGA Specification in the FPGA Specification. FPGA functions and structural specifications shall be described in the FPGA Specification. The minimum information required in the FPGA Specification is described in Table 3.2-7 of this SDP.

#### (4) Implementation Phase

The Implementation Phase consists of the following activities:

- Coding Activity

The Design Section shall generate the processor software source code and the FPGA source code based on the Program Specification(s) and the FPGA Specification.

- Unit V&V Test Activity

The V&V Team shall execute Unit V&V Testing on the processor software source codes and FPGA source codes as described in the SVVP and the STP (Sections 3.10 and 3.12, respectively).

After the V&V Team has executed the Unit V&V Test, and any V&V Anomaly Reports are dispositioned and closed, the Executable Module shall be generated by the Design Section.

#### (5) Test Phase

The V&V Team shall perform Integration V&V Tests as described in the SVVP and the STP (Sections 3.10 and 3.12, respectively).

#### (6) Installation Phase

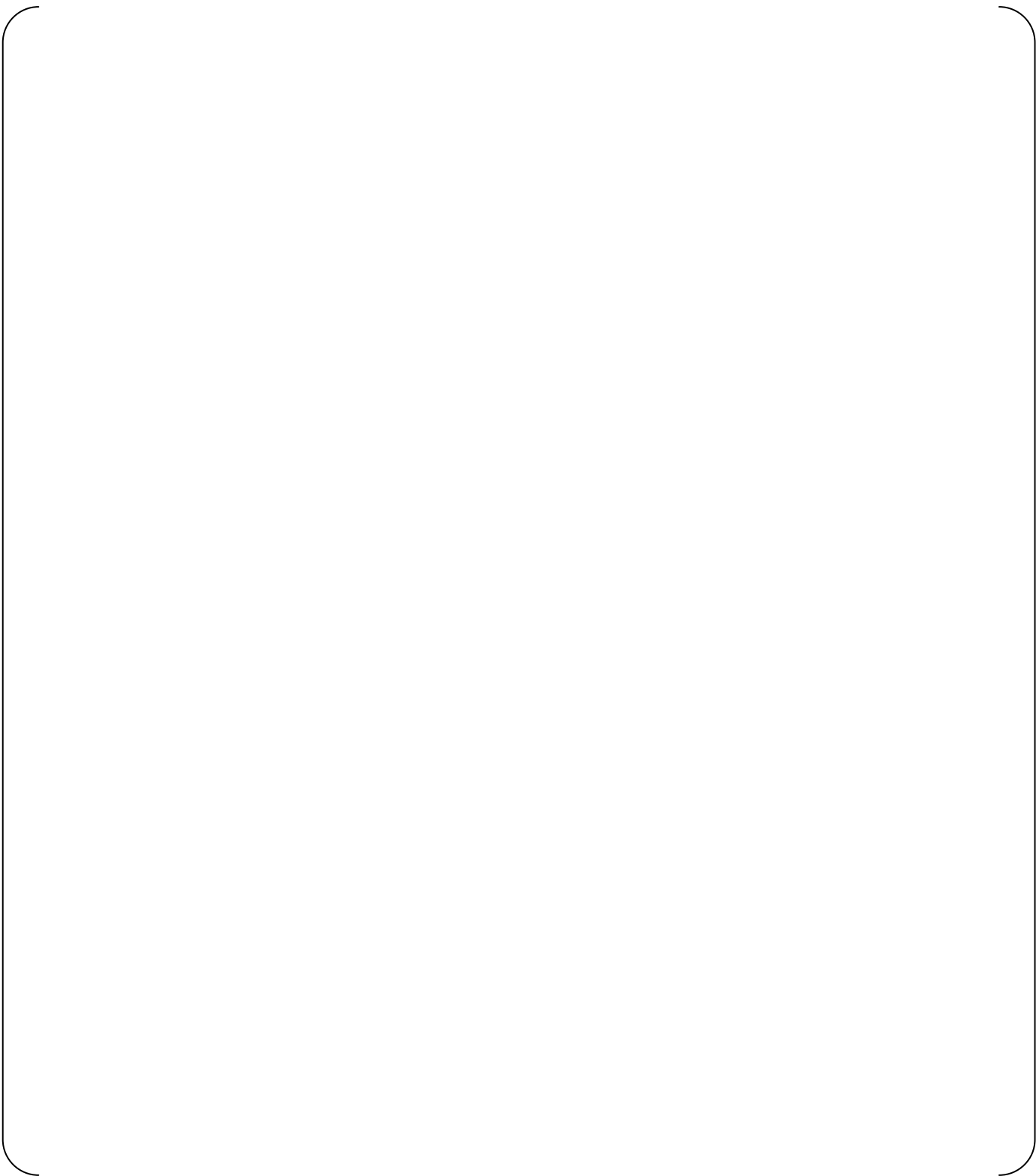
The Installation Phase in the context of this SPM is part of the hardware manufacturing process conducted in the production area (i.e. MELTAC factory environment). Installation of the Basic Software into target hardware modules is described in the SInstP (Section 3.5). There are no V&V activities associated with installation of the Basic Software during manufacturing of specific MELTAC hardware modules.

The Basic Software Life Cycle model also allows for localized iteration in a particular activity group when anomalies are discovered during the V&V process, or through audits by the QA section. Figure 3.2-1 is a diagram for general overview of this modified waterfall process.

In Figure 3.2-1, the left column of boxes shows the phase of software development, the center column shows the activities of the Design Section, and the right column shows the activities associated with the V&V Team. Solid arrows indicate how documents or source codes provide input into activities associated with each development phase, and dashed arrows indicate how feedback from design reviews and V&V activities affect subsequent design activities. Dashed-dotted arrows indicate the flow of higher-level information to inform V&V activities.

The development process progresses in close coordination with the V&V Team. Software is tested and reviewed within the Design Section before it is released to the V&V Team; however, the tests conducted by the Design Section are not credited in the V&V program. Each phase of the development process is driven by design inputs and generates design outputs. The list of inputs and outputs is tracked by control documents.

The software shall be officially released when all the V&V activities (including Software safety analysis V&V) are completed.

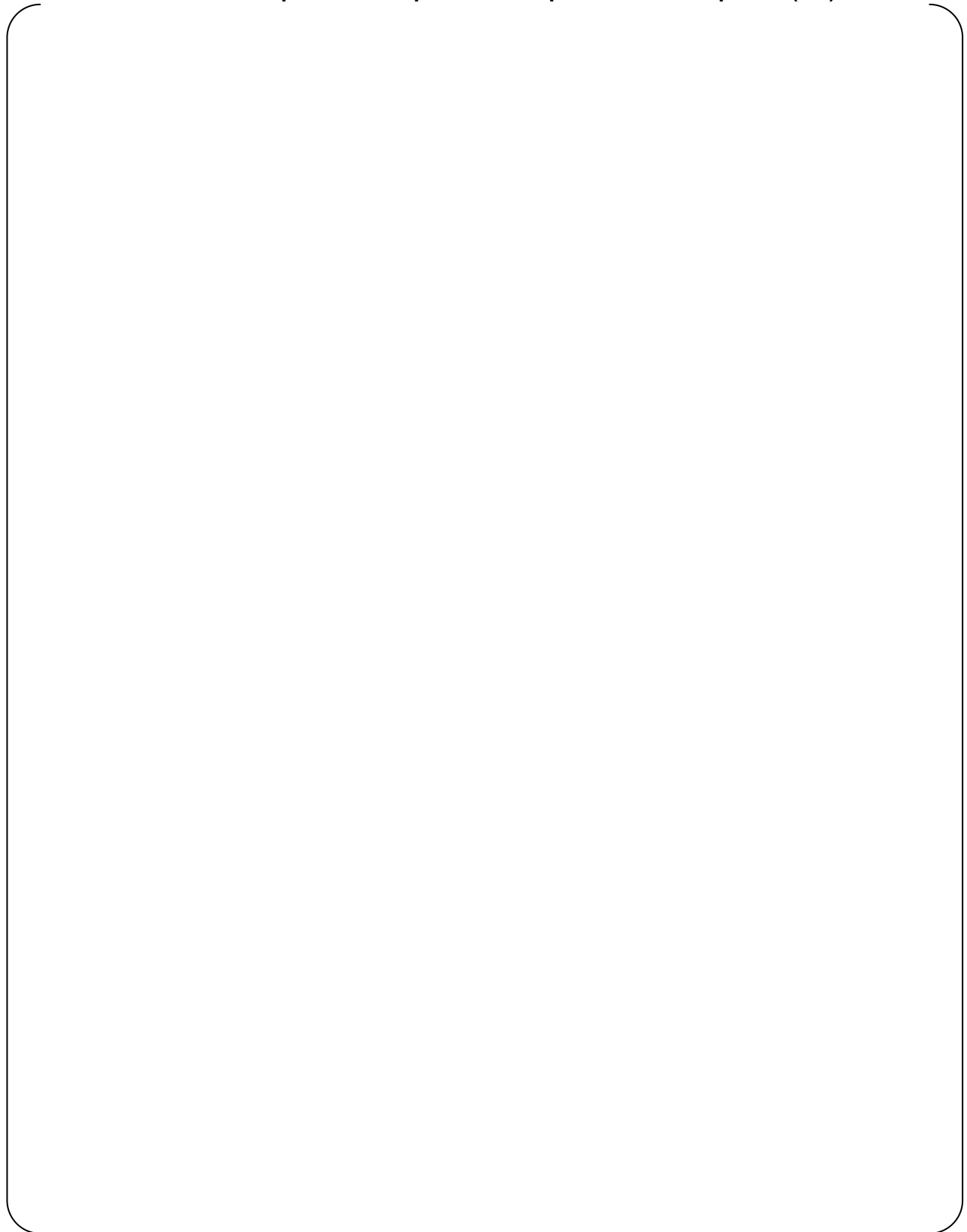


**Figure 3.2-1 MELTAC Platform Basic Software Life Cycle**

Inputs and outputs in each phase of development are as follows.

**Table 3.2-2 Inputs and outputs in each phase of Development (1/3)**

**Table 3.2-2 Inputs and outputs in each phase of Development (2/3)**



**Table 3.2-2 Inputs and outputs in each phase of Development (3/3)**



### 3.2.5.2.1 Requirements Phase

The MELTAC Platform Basic Software shall not use any procured software but only those developed by MELCO. The feasibility including technical issues, budget, and personnel shall be examined prior to the initiation of the development project, and the results shall be reflected to the Platform Specification.

All Software design requirements shall be identified and documented.

The above requirements are specified in the Platform Specification, which is the highest-level document for the MELTAC platform. The Platform Specification identifies the requirements for the Platform and defines the boundary between the functions of hardware and software.

The preparer of the Platform Specification is a member of the Design Section and the Platform Specification is approved by the Design Section Manager.


A design review shall be performed for the Platform Specification. The design review should be performed by the Design Review Engineers who are other than the Design Engineer who prepared that document.

Moreover, members from the V&V Team independently perform V&V of the Platform Specification to ensure that the requirements of industry standards (e.g., IEEE Std 7-4.3.2-2003 (Reference 13), IEEE Std 1012-1998 (Reference 20)) are included in the Platform Specification.

The V&V Team develops the Requirement Traceability Matrix (RTM). The RTM is used to ensure that the requirements from higher specifications, including critical safety functions, are completely included into the specifications and source code of each subsequent phase. Detailed procedures of V&V are specified in the SVVP (Section 3.10). Table 3.2-3 lists the general requirements of the Platform Specification.



**Table 3.2-3 Minimum Information Required in the Platform Specification (1/2)**



**Table 3.2-3 Minimum Information Required in the Platform Specification (2/2)**

Security Related and Proprietary Information - Withheld Under 10CFR2.390

{ [

Security Related and Proprietary Information - Withheld Under 10CFR2.390

] }

### **3.2.5.2.2 Design Phase**

The high-level system requirements are decomposed into multiple levels. The Platform Specification is decomposed into a hardware specification and a software specification.

The design documents shall include the following information as appropriate, in order for individuals other than those who performed the original design to perform an evaluation.

- Use of format, drawing method, symbols and abbreviations.
- Drawing title and number (drawing and sheet numbers for those developed by CAD)
- Identification of design documents by name of plant, systems, equipment and components and document number, revision mark.
- Identification of document status such as approved, reviewed or revised.

Recognizing that the NRC is considering the programmable portion of FPGA as software, the design process distinguishes between traditional 'Processor Software' and 'FPGA Software'.

(1) Processor software

The software design documentation for processor software consists of the Software Specification and the Program Specification. The Software Specification defines functional specifications of the entire software. Each Program Specification defines structure specifications of each functional unit and the processing details of each unit. Figure 3.2-2 shows the relationship between general Software Specifications and Program Specifications. Items to be included in the Software Specification are listed in Table 3.2-4. Items to be included in the Program Specification are listed in Table 3.2-5.

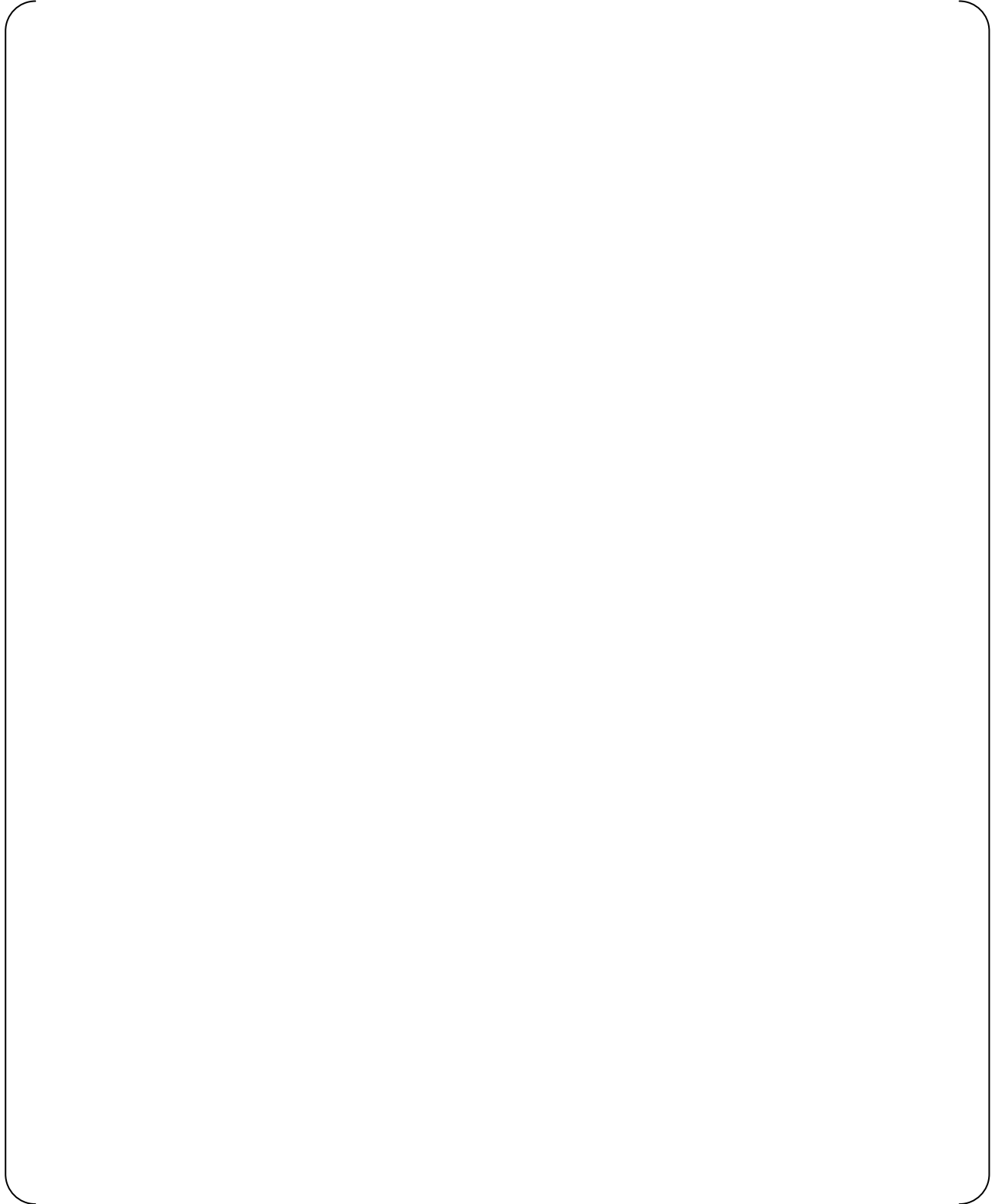


**Figure 3.2-2 Relationship between General Software Specifications  
and Program Specifications**

**Table 3.2-4 Minimum Information Required in the Software Specification**

Security Related and Proprietary Information - Withheld Under 10CFR2.390

**Table 3.2-5 Minimum Information Required in the Program Specification**



A design review shall be performed for the Software Specification and Program Specification. The design review should be performed by members other than the designer who prepared that document.

Moreover, the Independent V&V Team performs V&V of the Software Specification and Program Specification. The V&V Team verifies traceability between Platform Specification and Software Specification, and between Software Specification and Program Specification.

The V&V Team verifies that the requirements of IEEE Std 1012-1998 for Software Design are reflected in the Software Specification and Program Specification.

The V&V Team updates the RTM (Requirement Traceability Matrix). Detailed procedure of V&V is specified in the SVVP (Section 3.10).

{ [

Security Related and Proprietary Information - Withheld Under 10CFR2.390

] }

#### (2) FPGA software

The functional specification (requirements) shall be identified in the Hardware Specification of the hardware module (printed wiring board) with the applicable FPGA software installed. FPGA functions and structure specifications shall be described in the FPGA Specification.

Figure 3.2-3 shows the relationship between general Hardware Specifications and FPGA Specifications. FPGA functions defined in the Hardware Specification shall be translated

into the detailed design specification in the FPGA Specification.  
Table 3.2-6 lists items to be included in the FPGA Specification.



**Figure 3.2-3 Relationship between General Hardware Specifications  
and FPGA Specifications**

**Table 3.2-6 Minimum Information Required in the FPGA Specification**





A design review shall be performed for the FPGA Specification. The design review should be performed by members other than the designer who prepared that document.

Moreover, the Independent V&V Team performs V&V of the FPGA Specification. The V&V Team verifies traceability between the FPGA Specification and the higher level document.

The FPGA is implemented on board, and works as a hardware circuit. So the Hardware Specification defines functional requirements of FPGAs. Therefore, the higher level document of the FPGA Specification is the Hardware Specification.

The V&V Team extracts descriptions related to FPGA from the Hardware Specification, and checks traceability of the FPGA Specification.

The V&V Team updates the RTM (Requirement Traceability Matrix). Detailed procedure of V&V is specified in the SVVP (Section 3.10).

{ [

Security Related and Proprietary Information - Withheld Under 10CFR2.390

] }

### **3.2.5.2.3 Implementation Phase**

#### **3.2.5.2.3.1 Coding Activity**

This is the phase where a program (source code) is written using the Software Specifications and Program Specifications for Processor Software or using the FPGA Specifications for FPGA Software.

Source code shall be written in accordance with the coding rules established by the design or programming section.

The generated source code shall be compiled. If any compiling errors are identified, the source code shall be modified to remove all those errors.

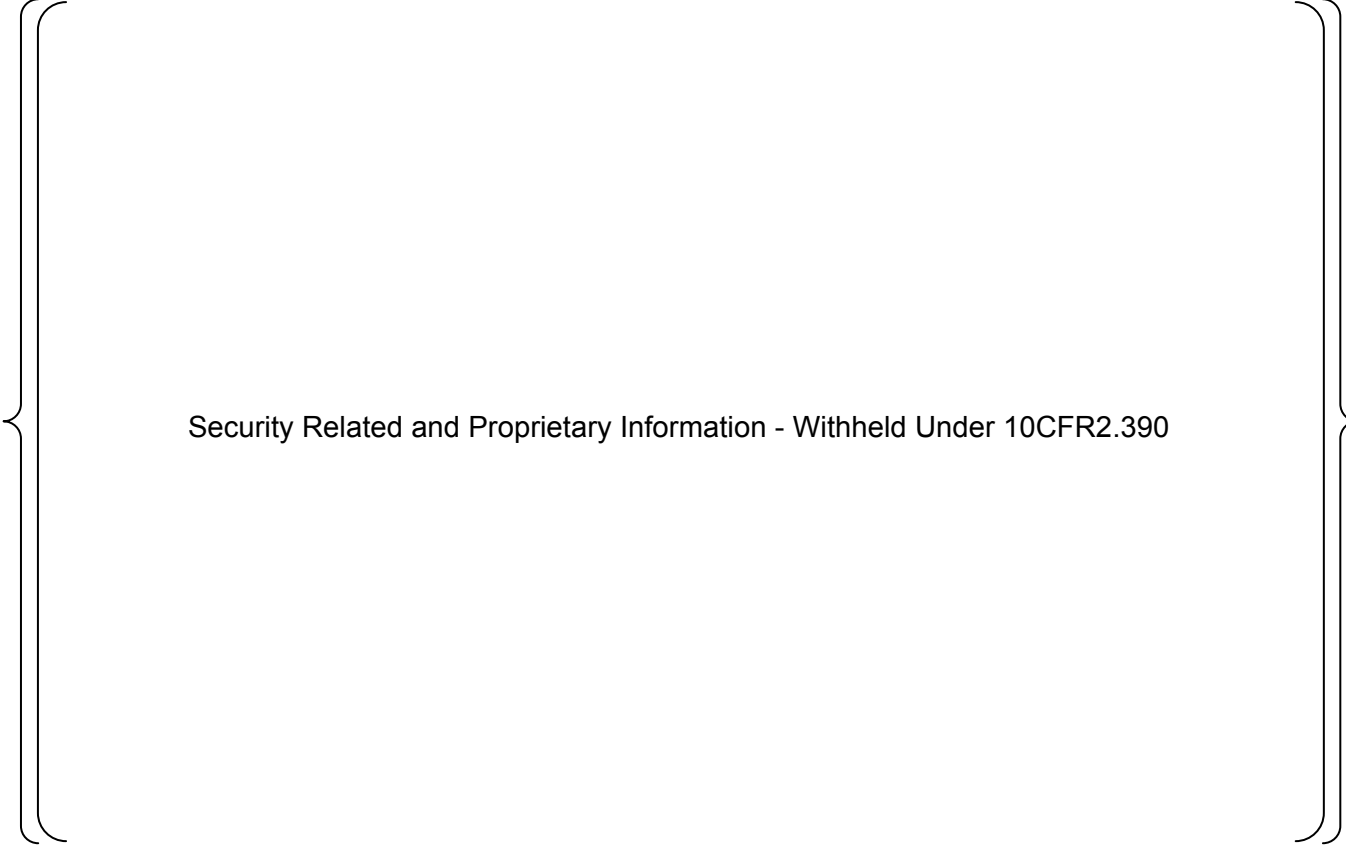
Software operating documentation shall be prepared prior to release of the software to the sections other than the Design Section.

Security control shall be applied for the access to the environment for writing source code, programming and generating modules (specifically computers in which particular

development environment is installed) to ensure that the access is restricted to only authorized personnel.

Table 3.2-7 lists the requirements for writing source code.

**Table 3.2-7 Requirements for Writing Source Code**



Security Related and Proprietary Information - Withheld Under 10CFR2.390

[

]

[

]  
[

]  
{[

Security Related and Proprietary Information - Withheld Under 10CFR2.390

Security Related and Proprietary Information - Withheld Under 10CFR2.390

1}

#### **3.2.5.2.3.2 Unit Test Activity**

The V&V Team shall prepare the Unit V&V Test Specification to validate the relationship between component/unit inputs and outputs, and demonstrate the following:

- The components/units under test meet the functional specifications described in the Program Specification, and
- All possible paths for every test case are executed as programmed, in terms of program control structure.

Unit test is described in the SVVP and the STP (Sections 3.10 and 3.12, respectively).

After performing unit tests, the Executable Module is generated by using compiler and linker. The Executable Module shall be installed into the hardware as preparation of the integration test.

#### **3.2.5.2.4 Test Phase**

The V&V Team shall prepare the Integration V&V Test Specification to validate that the functions and performance of the integrated system meet the specifications in the Platform Specification (including cyber security features) and the Software Specification by running software on the MELTAC Platform hardware.

Integration test is described in the SVVP and the STP (Sections 3.10 and 3.12, respectively).

After the Design Section activities described in the SDP, the V&V Team activities described in the SVVP (Section 3.10), and any project-specific reviews and audits of the configuration management activities are completed, the Design Section shall generate, review and approve the following documents as described in the SCMP (Section 3.11) and release the Basic Software.

- (1) Basic Software Installation Drawing
- (2) Software Release Note

#### **3.2.5.3 Schedule**

The procedure for Project Plan is specified in Section 3.1 of this SPM. The schedule of development is defined in the Project Plan. Schedule is produced to satisfy the following requirements.

- (1) Clarify important work items, milestones and hold points.
- (2) Project milestones include review and audit.
- (3) Secure enough time according to the work volume for each process.

#### **3.2.5.4 Methods / Tools**

[

]

### **3.2.5.5 Standards**

This SDP complies with the following guidance and standards. Methods of compliance against the IEEE standards that are endorsed by Regulatory Guides are tabulated in Appendix B.

- (1) Clause 5.3 and 5.9 of IEEE Std 603-1991 (Reference 14) which are endorsed by RG 1.153 (Reference 5)
- (2) Clause 5 of IEEE Std 7-4.3.2-2003 which is endorsed by RG1.152 (Reference 4)
- (3) IEEE Std 1074-1995 which is endorsed by RG 1.173
- (4) IEEE Std 830-1993 (Reference 18) endorsed by RG 1.172 (Reference 10)
- (5) NUREG/CR-6101

### 3.3 Software Quality Assurance Plan

#### 3.3.1 Purpose and Scope

The purpose of the Software Quality Assurance Plan (SQAP) is to describe the quality assurance requirements and methods used to assure high quality of the Basic Software throughout the Basic Software life cycle process.

The requirements of this SQAP as well as this entire SPM shall be implemented by procedures controlled in accordance with the 10 CFR 50 Appendix B-based MELCO Quality Assurance Program (QAP) [

]. All responsible groups that are assigned activities the described in this SPM shall follow these implementing procedures.

The quality of the following Basic Software life cycle process documents outputs shall be assured through the methods and processes described in this SQAP:

- Project Plan as described in the SMP (Section 3.1)
- Design documentation (Platform Specification, Software Specification, Program Specification, and FPGA Specification)
- Source code (for Processor Software and for FPGA)
- Test Descriptions, Test Specifications, and Test Reports (as described in the SVVP and the STP)

#### 3.3.2 Organization / Responsibilities

The organization of the groups responsible for Basic Software quality is described in Section 3.1.2.

The Quality Assurance Section and the V&V Team shall be independent from Design Section and Manufacturing Department members. V&V Team independence is described in detail in the SVVP (Section 3.10).

The level of the QA Manager within the overall MELCO organization is equivalent or higher than the levels of the Design Section and V&V Managers.

The Design Section Manager is responsible for ensuring that all Basic Software design activities are performed as described in accordance with the SDP (Section 3.2).

The Design Section shall generate and maintain the design outputs throughout the Basic Software life cycle as described in the SDP (Section 3.2) and shall also assure their correctness through reviews by Design Review Engineers.

The Design Section is responsible for performing the software safety analysis activities described in the SSP (Section 3.9).



The V&V Team Manager is responsible for ensuring that all the V&V activities are executed independently by the V&V Team, including software safety analysis V&V activities, as described in the SVVP (Section 3.10).

The QA Manager is responsible for assuring that the planned software development and V&V activities are appropriately conducted by these sections in accordance with this SPM and implementing procedures controlled under [

] The Quality Assurance Section has the following responsibilities:

- (1) Confirm the Basic Software development process is performed in accordance with this SPM and implementing procedures controlled under the MELCO QAP including the Safety Analysis is performed in accordance with the SSP (Section 3.9).
- (2) Confirm that V&V activities are performed by the V&V Team in accordance with the SVVP and implementing procedures controlled under the MELCO QAP.
- (3) Perform QA audits and document the results as described in this SQAP. The Quality Assurance Section personnel shall report audit results to the QA Manager.
- (4) Maintain and control the MELCO internal procedures in accordance with [ ]].

### 3.3.3 Security

Through audits performed by the Quality Assurance Section, it is verified that security of Basic Software development assets is managed in accordance with this SPM.

### 3.3.4 Execution / Methods

#### 3.3.4.1 Measurement

The number and age of open QA Audit findings is an important indicator for observing the quality of the Basic Software and the extent to which Basic Software life cycle activities are performed in accordance with this SPM. QA Audit findings shall be tracked to closure by the Quality Assurance Section.

If errors in the Basic Software life cycle process outputs not already discovered and documented by either the Design Section or the V&V Team (including their internal reviewer) are found via QA Audits, this indicates a potential weakness in the effectiveness of the overall organization and would merit further investigation.

Other Basic Software process metrics include the following:

- Number of comments identified through design reviews (as described in Section 3.2)

- Number and severity level of V&V Anomaly Reports (as described in Section 3.10)
- Number of issued Nonconformance Reports (as described in this SQAP)
- Number of Corrective Action Reports (as described in this SQAP)

### 3.3.4.2 Procedures

#### 3.3.4.2.1 Reviews

Various reviews shall be performed throughout the Basic Software life cycle using the methods described in IEEE Std 730-2002 (Reference 15) and IEEE Std 1028-1997 (Reference 21). The purpose of these reviews is to ensure that the inputs of each life cycle phase are appropriately translated into the required outputs, and that the required outputs are complete, correct and of high quality. Review activities and the organizations responsible for them are described below:

##### (1) Software Requirements Review

The software requirements review shall be performed upon completion of a change to the Platform Specification for the MELTAC Platform. The Design Section shall identify and documents all the requirements necessary for the Basic Software design (software design requirements) in the Platform Specification. After a change to the Platform Specification is prepared, the Design Section Manager shall assign a qualified Design Review Engineer to perform a design review using the technical review methods described in IEEE Std 1028-1997. The Platform Specification is reviewed to ensure that it continues to meet the requirements described in the Concept Phase documents. If any changes to the Platform Specification fail to meet one or more Concept Phase requirements, it shall be corrected prior to Design Section Manager approval.

When the Design Section Manager confirms that the results of the design review are appropriately incorporated into the Platform Specification, the Design Section Manager approves the Platform Specification.

The V&V Team verifies the Platform Specification in accordance with Section 3.10.4.4.1.

##### (2) Architecture Design Review

The architecture design review and the software requirements review described above are performed at the same time. The Platform Specification for the MELTAC Platform encompasses all the requirements necessary for the Basic Software detailed design.

The Design Section Manager confirms whether the software requirements are appropriately incorporated into the software basic design during the review process specified in paragraph (1).

### (3) Detailed Design Review

A detailed design review shall be performed upon completion of the Software Specification, the Program Specification and the FPGA Specification in the Design Phase. These Specifications translate the requirements of the Platform Specification into these individual specifications that describe software functions and structures. After changes to these specifications are prepared, the Design Section Manager assigns a qualified Design Review Engineer to perform a design review using the technical review methods described in IEEE Std 1028-1997. If any changes to the Software Specification, the Program Specification or the FPGA Specification fail to meet one or more Platform Specification requirements, it shall be corrected prior to Design Section Manager approval.

Detailed design reviews shall also be performed upon completion of software coding activities to confirm they implement the design characteristics described in the Software Specification, the Program Specification and the FPGA Specification.

When the Design Section Manager confirms that the results of the design review are appropriately incorporated into the affected documents or source codes, the Design Section Manager approves them.

The V&V Team shall verify these Specifications in accordance with Section 3.10.4.5.

Detailed design reviews are equivalent to the "Inspections" review method described in IEEE Std. 1028-1997.

### (4) V&V Task Manual Review

A V&V Team, independent of the Design Section, shall be established for Basic Software V&V activities as described in the SVVP (Section 3.10). An individual V&V Task Manual for each project or activity is prepared to document specific V&V resources, tasks and procedures. In the V&V Task Manual review, the V&V Team Manager and the Design Section Manager confirm the appropriateness of the V&V tasks described in the V&V Task Manual, using the management review methods described in IEEE Std 1028-1997. When the V&V Team Manager confirms that the review results are appropriately incorporated into the V&V Task Manual, the V&V Team Manager approves the V&V Task Manual.

### (5) Managerial Review

Managerial reviews are carried out by the Design Section Manager on a periodic basis. Specific managerial review activities are specified in the Project Plan. The purpose of this review is to periodically assess the Basic Software development process using the management review methods described in IEEE Std 1028-1997.

#### (6) Post-Implementation Review

The post-implementation review shall be conducted at the project closing meeting after release of the MELTAC Platform Basic Software, using the management review method described in IEEE Std 1028-1997. When the Design Section Manager confirms that all of the required Basic Software development activities were performed appropriately, with no problems left unsolved, the Design Section Manager approves the Project Report.

#### (7) Walk-Throughs

Walk-throughs are not conducted for the Basic Software development activities because the source codes are verified by Detailed Design Review activities as described in Step (3), above, and V&V activities as described in the SVVP (Section 3.10)

### **3.3.4.2.2 Software Audits**

The purpose of software audits is to provide an independent evaluation of conformance of Basic Software configuration items and processes to the requirements of this SPM. Software audit results shall be performed by the QA Section, and recorded in QA Audit Reports. The QA Manager is responsible for ensuring these audits are planned, scheduled and performed by the QA Section, and is responsible for final review and approval of QA Audit Reports.

#### (1) Functional Audit

The functional audit is performed by the QA Section prior to release of any changes to the MELTAC Platform Basic Software to verify that the affected software items conform to the software requirements, using the audit methods described in IEEE Std 1028-1997. The QA Section confirms that the RTM adequately traces the results of the Basic Software life cycle process to the Platform Specification. The QA Manager also confirms that V&V Anomaly Reports are adequately closed.

#### (2) Physical Audit

The physical audit is held prior to the release of any changes to the MELTAC Platform Basic Software to verify that the software and its design documentation are internally consistent and that their configurations are appropriately controlled, using the audit methods described in IEEE Std 1028-1997. The QA Section confirms the version of all

software products (documents and software) agrees with the version described in the Configuration Management Sheet as described in the SCMP (Section 3.11).

### (3) In-Process Audit

The QA Manager appoints a Software Audit Team, by assigning QA Section members who have knowledge and experience related to software development, before a project is completed. If sufficiently qualified QA Section resources are unavailable, the QA Manager may supplement the Software Audit Team with independent software QA specialists who have the necessary qualification. The Software Audit Team shall perform an In-Process Audit of activities from each Basic Software life cycle phase by confirming that the Design Section and the V&V Team are performing the activities and tasks described in this SPM and implementing procedures controlled under the MELCO QAP.

The In-Process Audit steps are as follows:

- a. Appoint Software Audit Team Leader (by QA Manager)
- b. Select Software Audit Team members (by Software Audit Leader)
- c. Create In-Process Audit Plan (by Software Audit Team)
- d. Perform In-Process Audit (by Software Audit Team)
- e. Prepare QA Audit Report (by Software Audit Team)
- f. Create Corrective Action Plan (by Design Section or V&V Team)
- g. Execute Corrective Actions (by Design Section or V&V Team)
- h. Verify Corrective Action Results (by Software Audit Leader)
- i. Review and approve QA Audit Report (by QA Manager)

#### **3.3.4.2.3 Software Tests**

The V&V Team shall conduct Unit Tests and Integration Tests to validate the MELTAC Platform Basic Software as described in the SVVP and STP (Sections 3.10 and 3.12), respectively.

#### **3.3.4.2.4 Problem Reporting and Corrective Action**

In each phase of the software life cycle, the Design Section and/or V&V Team shall identify, record, and implement corrective actions for problems that meet the criteria for a Nonconformance Report as described in [ ].

If the V&V Team detects an anomaly, a V&V Anomaly Report shall be written as described in the SVVP; if the severity level of the V&V Anomaly Report is high, the

V&V Team shall initiate a Nonconformance Report and send it to the Design Section for evaluation and corrective actions.

Nonconformance Reports shall be promptly initiated by any section that detects a nonconforming condition, and sent to the responsible organization (Design Section or V&V Team). The Nonconformance Report shall include the NCR number, date of nonconformance, title, description of nonconformance, responsible section and proposed due date for resolution.

Nonconforming conditions shall be evaluated for their safety significance in terms of safety function or performance, including the results of software safety analysis activities described in the SSP (Section 3.9). If the identified nonconformance has an adverse effect on a critical safety function, or is a significant condition adverse to quality, the requirements of [

] shall be followed.

The QA Section shall perform QA audits of the software life cycle process activities performed by the Design Section and the V&V Team by the In-Process Audit method described in Section 3.3.4.2.2. The results of these QA software audits are documented in QA Audit Reports. If there are any findings, the section that receives the finding shall determine, implement and document corrective actions as described in [ ]. The results of the corrective actions shall be described in the Corrective Action Report. The QA Section oversees the corrective action process to ensure Corrective Action Reports resolve the identified findings in a timely manner.

#### **3.3.4.2.5 Media Control**

Media control measures shall be implemented as described in Section 3.11.3.2.11 of the SCMP (Section 3.11) to control the MELTAC Platform Basic Software to assure that any unauthorized code is not contained in the software in the software development phase and that the correct Basic Software is released upon completion of the development activities.

#### **3.3.4.2.6 Supplier Control**

All Basic Software items are generated and maintained by MELCO. No commercial software is included in the Basic Software. Therefore there are no supplier controls described in this SPM.

#### **3.3.4.2.7 Training**

Each Section Manager shall determine the skills and abilities required for implementing the activities and tasks assigned to their section, and shall provide the necessary training to assigned personnel.

Each Section Manager shall review the Technical Map for each of their personnel as described in MELCO ESC N-0200, "Quality Assurance Activity Management Procedure" and related training records at the beginning of each fiscal year to design an annual training plan for assigned personnel. Identified training activities shall be recorded in the Technical Map for each person.

#### **3.3.4.2.8 Risk Management**

Risk management of the Basic Software shall be performed using the Risk Matrix and Problem List tools described in the SMP (Section 3.1).

The Design Section Manager shall periodically assess the risks identified in the Risk Matrix, using the Managerial Review method described in Section 3.3.4.2.1, to determine the likelihood that any identified risks will emerge into actual problems and their priority for mitigation. If the Design Section Manager determines that any identified risks are likely to become an actual problem, then they shall be added to the Problem List to assure additional oversight and mitigation. Nonconformance Reports shall be initiated, as described in Section 3.3.4.2.4, for any risks added to the Problem List that are likely to impact critical safety functions described in the SSP (Section 3.9).

#### **3.3.4.3 Record Keeping**

Documents associated with Basic Software life cycle activities described in this SPM shall be stored as QA Records in accordance with [ ].

Each QA Record shall be assigned a number depending on the document type, shall be uniquely identified, and shall be searchable via the Numbering List.

QA Records shall be kept in locked cabinets in an appropriate area where only an authorized administrator can enter.

The types of document to be treated as a QA Records are listed in Table 3.3-1:

**Table 3.3-1 Document Types****3.3.4.4 Methods / Tools**

This section describes the following methods and tools used in support of the activities described in this SQAP:

[



]

**3.3.4.5 Standards**

This SQAP complies with the following guidance and standards. Methods of compliance against the IEEE standards that are endorsed by Regulatory Guides are tabulated in Appendix B.

- Clause 5.3.1 of IEEE 7-4.3.2-2003 which is endorsed by RG 1.152 Rev. 2
- Clause 3.3 of IEEE 1074-1995 which is endorsed by RG 1.173
- Sections 3.1.2 and 4.1.2 of NUREG/CR-6101

### **3.4 Software Integration Plan**

#### **3.4.1 Purpose**

This Software Integration Plan (SIntP) describes three types of activities:

- (1) Integration of individual software units into complete executable modules for:
  - the software stored in the nonvolatile memory of the platform's main CPU Module
  - the firmware stored in peripheral modules
  - FPGA stored in the CPU Module and peripheral modules
- (2) Integrate the executable modules in Step (1), above, into the target MELTAC hardware modules.
- (3) Perform Integration Test of the MELTAC Platform hardware and software.

Software Integration is distinguished from Software Installation which is described in the SInstP (Section 3.5).

- Software Integration as described in this SIntP is an activity that only applies to changes to the MELTAC Basic Software.
- Software Installation as described in the SInstP is a recurring activity that applies to each target MELTAC hardware module produced for application projects.

#### **3.4.2 Organization / Responsibilities**

After the V&V Team completes the unit test of all individual software units in the Implementation Phase as described in the SVVP (Section 3.10), the Design Section integrates the software units into complete executable modules as described in the SDP (Section 3.2) and installs them on the target hardware in order to have an integrated system for use in the Test Phase environment.

The V&V Team is responsible for confirming the configuration of all units that have been integrated as described in the SCMP (Section 3.11).

The V&V Team is also responsible for preparing and executing the Integration V&V Test Specification as described in the SVVP and the STP.

If any anomalies occur during execution of the Integration V&V Test Specification, the V&V Team shall issue a V&V Anomaly Report as described in the SVVP.

#### **3.4.3 Execution / Methods**

##### **3.4.3.1 Security**

Throughout the Integration Phase, security management shall be performed, as described in Section 3.1.3 and Appendix C. These security controls ensure that unauthorized changes cannot be introduced during integration activities.

#### **3.4.3.2 Measurement**

Any anomalies detected by the V&V Team during execution of the Integration V&V Test Specification shall be recorded in a V&V Anomaly Report as described in the SVVP. The number and severity level of V&V Anomalies shall be measured as described in the SVVP.

The Design Section Manager and V&V Team Manager shall periodically assess the number and severity level of V&V Anomaly Reports that are generated during the Implementation Phase.

An excessive number of V&V Anomaly Reports arising from execution of the Integration V&V Test Specification is an adverse trend indicating poor quality in prior phases (e.g. design and/or unit testing), and shall result in generation of a Nonconformance Report as described in the SQAP (Section 3.3) for further corrective action.

#### **3.4.3.3 Procedures**

[

]

The Integration V&V Test Specification shall be executed as described in Section 3.10.4.7, "Integration Test Phase" of the SVVP.

The integration test validates the integrated design outputs. The results are recorded in the Integration V&V Test Report as described in the SVVP and the STP (Sections 3.10 and 3.12, respectively).

#### **3.4.3.4 Methods / Tools**

[

]

#### **3.4.4 Standards**

This SIntP complies with the following guidance and standards. Methods of compliance against the IEEE standards that are endorsed by Regulatory Guides are tabulated in Appendix B.

(1) Section 5.3.7 and 5.3.8 of IEEE Std 1074-1995 which are endorsed by RG 1.173

(2) Section 3.1.7 of NUREG/CR-6101

## **3.5 Software Installation Plan**

### **3.5.1 Purpose**

This Software Installation Plan (SInstP) describes the methods used to install the Basic Software in the MELTAC Platform. This SInstP ensures the following:

- (1) Verify that the correct software is being installed.
- (2) Verify that the software has been installed correctly.

This SInstP is applied in the Installation Phase. Procedures that implement the requirements of this SInstP shall be used by the Manufacturing Department at the MELCO factory for installing the Basic Software in nonvolatile memory, firmware or FPGA devices during the production process.

Section 3.5.3.1.3 of this plan describes how the Basic Software is upgraded by MELCO. Operational confirmation, performed after software upgrade, occurs as part of the installation test and uses the actual equipment configuration.

### **3.5.2 Organization / Responsibilities**

The Manufacturing Department shall perform the installation of the Basic Software in the actual hardware modules manufactured during the production process in accordance with implementing procedures.

The Design Section is responsible for producing, maintaining and identifying the master installation software as described in Section 3.5.3.1.1, the SDP, and the SCMP (Sections 3.2 and 3.12, respectively).

### **3.5.3 Execution / Methods**

#### **3.5.3.1 Measurement**

Following items shall be metrics of the Basic Software Installation.

- Number of anomaly report identified through installation.

#### **3.5.3.2 Procedure**

The procedures that implement the requirements of this SInstP shall be maintained by the Design Section and the Manufacturing Department.

If any anomalies occur during software installation activities described in this SInstP, the Manufacturing Department shall report the anomalies to the Design Section using a Nonconformance Report as described in the SQAP (Section 3.3).

### **3.5.3.2.1 Registration**

[

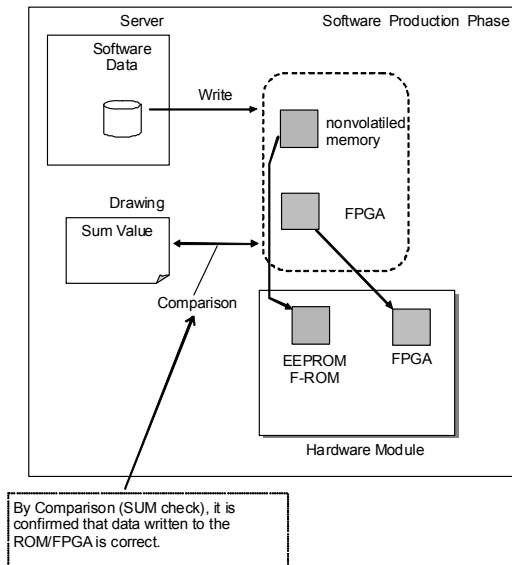
]

### **3.5.3.2.2 Installation**

[

1

An overview of the general installation process is shown in the following diagram:



**Figure 3.5-1 Basic Software Installation Process**

### 3.5.3.2.3 Installation of Software Changes in the Maintenance Phase

[



]

**3.5.3.3 Methods / Tools**

[

]

**3.5.4 Standards**

This SInstP complies with the following guidance and standards. Methods of compliance against the IEEE standards that are endorsed by Regulatory Guides are tabulated in Appendix B.

(1) Clause 6.1 of IEEE Std 1074-1995 which is endorsed by RG 1.173 (Reference11)

(2) Section 3.1.8 of NUREG/CR-6101

### **3.6 Software Maintenance Plan**

#### **3.6.1 Purpose**

The Software Maintenance Plan (SMaintP) describes the change activities of the current Basic Software.

There are two types of change factors which change the released software as follows:

- (1) reported problem
- (2) requested change

The same change process is applied for both change factors and the overview of the change process between the plans of this SPM is described in Section 3.0.

The change control process is described in the SCMP (Section 3.11).

The Software Maintenance Plan shall also control the MELTAC maintenance manual that includes plant specific documents which identify all hardware modules (including version numbers, locations within each chassis, and all software modules including version numbers). These documents contain explanations of the equipment's hardware and software including information necessary to perform all maintenance activities.

Retirement of the software shall be controlled in accordance with the Software Maintenance Plan.

The retirement of the software shall be informed formally to the affected sections. Shipment of the equipment with the retired software shall be prevented whether in a regular or improper manner after this phase by:

- (1) deleting it from production drawings, or
- (2) deleting Master Data for installation of the production (regular usage).

The retired software shall be archived and the report shall be recoded.

#### **3.6.2 Organization / Responsibilities**

The Software Maintenance Plan shall cover errors identified after product delivery. And this plan shall be applied to both cases where the error is discovered by an end-user or by MELCO personnel.

The Maintenance Section has the responsibility to interface with the customer when an error is identified at customer site. When the error is notified from the customer, the Maintenance Section shall inform the Design Section about the error occurrence with the related information.

The Maintenance Section also has the responsibility to record the problem and corrective action, and to report to the customer.

The Design Section has the responsibility to evaluate the problem. The Design Section investigates the cause of the error, and if a Basic Software or hardware modification is required, such modification will be executed by the Design Section according the design change procedure, as a corrective action.

The Design Section also has the responsibility to develop and maintain the MELTAC maintenance manuals.

### **3.6.3 Execution / Methods**

#### **3.6.3.1 Security**

Throughout the maintenance phase, security management shall be performed, in accordance with Section 3.1.3, which complies with RG 1.152. These security controls ensure that unauthorized changes cannot be introduced during maintenance activities.

#### **3.6.3.2 Measurement**

The maintenance section shall collect and record:

- (1) phenomena, causes, solutions, and all information pertaining to reported problem incidents from all customers.
- (2) in-depth information on all associated field equipment failures and errors.

An excessive error rate indicates poor quality in prior phases (e.g. design, unit testing, and/or validation). Therefore a reassessment of prior phase activities will be required.

#### **3.6.3.3 Procedure**

[

1

[

]

### **3.6.3.4 Methods / Tools**

The required development environment for the maintenance of the Basic Software is address in the SDP. Refer to Section 3.2.5.3 for details.

### **3.6.4 Standards**

This SMaintP complies with the following guidance and standards. Methods of compliance against the following IEEE standards that are endorsed by Regulatory Guides are tabulated in Appendix B.

- (1) Clause 5.4.2.3 of IEEE Std 7-4.3.2-2003 which is endorsed by RG1.152.
- (2) Clause 6.3 of IEEE Std 1074-1995 which is endorsed by RG 1.173.
- (3) Section 3.1.9 "Software Maintenance Plan" of NUREG/CR-6101.

## **3.7 Software Training Plan (STrngP)**

### **3.7.1 Purpose**

The purpose of this Software Training Plan (STrngP) for the Basic Software is to provide a description of the training that is required for equipment startup/shutdown, routine periodic maintenance and testing, and troubleshooting of the MELTAC Platform. This STrngP is intended for the customer's technicians and engineers.

This STrngP also describes provisions for the training that is required for the MELTAC Platform Basic Software Design Section engineers and V&V Team engineers.

It is not the intent of this STrngP to describe training activities for end-users, such as utility operators, to operate any specific PSMS application. Application level training is described in Section 3.7 of the Application Software Program Manual (MUAP-07017). Though these end-users may utilize elements of the MELTAC Platform Basic Software Training, such as use of the MELTAC Engineering Tool, it is important that training on the MELTAC Engineering Tool is provided in conjunction with training of the application software. This allows the use of the MELTAC Engineering Tool as a diagnostic or troubleshooting aid in the context of an operating protection system application. Training on physical and cyber security and configuration control shall be included in such training.

### **3.7.2 Organization / Responsibilities**

#### **3.7.2.1 Customer Training**

Training for the Basic Software can be provided by any cognizant members of the Basic Software design, V&V, or maintenance sections. This training class is highly-technical in nature, so it is taught by MELCO technicians or engineers to customer technicians or engineers. MELCO Trainers shall receive the training described in this STrngP and shall have sufficient knowledge on equipment startup/shutdown, routine periodic maintenance and testing, and trouble shooting of the MELTAC Platform.

Training for the Basic Software is typically provided at the MELCO site. This arrangement allows for flexibility in supplementing the training material with additional information that may be requested by customers attending the training. Training can also be held at other locations as long as it is equipped with the required training documentation, and platform equipment and MELTAC Engineering Tools for hands-on training.

#### **3.7.2.2 MELCO Training**

The Design Section and V&V Team engineers involved in MELTAC Platform Basic Software life cycle activities shall be trained at the MELCO site. The Design Section has the responsibility to conduct the training.

The Training Section Manager ensures the internal implementing procedures for training activities comply with this STRngP.

### 3.7.3 Measurement

To examine the effectiveness of training, tests shall be given to the trainees to check their level of understanding. The section responsible for training shall prepare the test. The test shall contain questions specific to the identified training objectives.

When testing is related to customer training, the section responsible for training administers and grades the test. The results shall be reported back to the appropriate section of the customer's organization.

MELCO trainee test results shall be reported to the immediate manager of the trainee.

The training process shall be periodically assessed by the Manager of the section responsible for training to determine its effectiveness, by reviewing the following information:

- V&V Anomaly Reports
- Nonconformance Reports related to Basic Software
- QA Audit Reports
- Trainee feedback information
- US-APWR operating experience

The assessment results shall be used to determine if training objectives or training materials need to be updated. Training material shall be updated when a change to the Basic Software is released.

### 3.7.4 Procedure

Training records shall be controlled in accordance with [ ]

### 3.7.5 Execution / Methods

The MELTAC Platform Training Manual shall be used in the training. The MELTAC Platform Training Manual and materials shall be periodically assessed as described in Section 3.7.3.

In the training, hardware and software similar to those of the customer shall be used. Customers and MELCO engineers shall be trained as follows.

### **3.7.5.1 Training for Customers**

[



]

**3.7.5.2 Training for MELCO Engineers**

[

]

**3.7.6 Standards**

This STRngP complies with the following guidance and standards. Methods of compliance against the IEEE standards that are endorsed by Regulatory Guides are tabulated in Appendix B.

- (1) Section 7.4 of IEEE Std 1074-1995 which is endorsed by RG 1.173
- (2) Section 3.1.10 "Software Training Plan" of NUREG/CR-6101.

### **3.8 Software Operations Plan (SOP)**

#### **3.8.1 Purpose**

This Software Operations Plan (SOP) governs the activities covered by the Software Training Plan, as described in Section 3.7.

In particular, this SOP describes measures to ensure the security and configuration control of the system, and in particular, the means used to ensure that there are no unauthorized changes to hardware, software, and system parameters. It describes how end users will be able to detect actual or attempted security breaches of an installed PSMS application, and it describes methods for responding to security problems or configuration control errors.

#### **3.8.2 Organization / Responsibilities**

The SOP is typically implemented by customer personnel under the customers QAP. However, the responsibility for system maintenance may be contracted to others, including MELCO.

#### **3.8.3 Execution / Methods**

The activities described in this SOP shall be implemented by procedures controlled under a 10 CFR 50 Appendix B-based QAP.

##### **3.8.3.1 Measurement**

The maintenance section shall collect and record:

- (1) phenomena, causes, solutions, and all information pertaining to reported problem incidents from all customers.
- (2) in-depth information on all associated field equipment failures and errors.

An excessive error rate indicates poor quality in prior phases (e.g. design, unit testing, and/or validation). Therefore a reassessment of prior phase activities will be required.

##### **3.8.3.2 Procedure**

Operation and maintenance manual shall be provided and submitted to customers. This manual will be developed and maintained as described in the SMaintP (Section 3.6).

The process of handling nonconforming items are the same as those described in the SMaintP (Section 3.6).

**3.8.4 Security**

{ [

Security Related and Proprietary Information - Withheld Under 10CFR2.390

] }

**3.8.5 Standards**

This SOP complies with the following guidance and standards. Methods of compliance against the IEEE standards that are endorsed by Regulatory Guides are tabulated in Appendix B.

(1) Clause 6.2 of IEEE Std 1074-1995 which is endorsed by RG 1.173.

### 3.9 Software Safety Plan (SSP)

This Software Safety Plan (SSP) describes methods for mitigating potential software hazards for the Basic Software.

This SSP complies with the guidance and standards listed in Section 3.9.6.

#### 3.9.1 Purpose

This Software Safety Plan (SSP) describes and provides a framework for ensuring:

- (1) Critical safety functions are identified and correctly implemented
- (2) Potential software or hardware hazards that may prevent those safety functions are postulated and correctly mitigated
- (3) Potential security vulnerabilities are mitigated before they can affect a critical safety function

Since at the platform level, the specific applications are unknown, the critical safety functions for the Basic Software are identified as the generic specifications for input processing, program execution and output processing as described in the MELTAC Technical Report (MUAP-07005).

Hazard mitigation at the platform level means that the software or hardware hazards that can adversely affect these functions are detected and alarmed, and the affected processing function(s) fails to a predictable state, as defined by the MELTAC Technical Report (MUAP-07005).

The Software Safety Plan for the Application Software, described in Section 3.9 of the "US-APWR Software Program Manual" (MUAP-07017), complements this SSP by providing a framework for the same activities described above, but at the application level. Critical safety functions are defined for each application, such as reactor trip or actuation of containment isolation.

The application level SSP describes and provides a framework for ensuring:

- (1) Critical safety applications are correctly implemented
- (2) Potential software hazards that may prevent those safety functions are postulated and correctly mitigated.

At the application level, correct mitigation considers overall system configuration, such as multiple controller groups, train level redundancy and inter-division and intra-division communications.

The software safety analysis activities described in this SSP are conducted for each phase of the MELTAC Basic Software's software life cycle as described in this SPM.

The Design Section is responsible for software safety analysis activities in the Concept, Requirements, Design, and Implementation Phases.

The V&V Team is responsible for verification of Design Section's software safety analysis outputs from these phases, as described in the SVVP (Section 3.10) and software safety analysis activities in the Test Phase as described in this SSP and the SVVP.

Independent V&V confirms that all safety significant requirements from the previous life cycle phase have been correctly analyzed, and are correctly reflected in the outputs of the current life cycle phase, with the appropriate level of detail to fulfill the life cycle phase requirements, as defined by this SPM.

The software configuration management activities, software quality assurance activities, and software V&V activities that support software safety are described in the SQAP (Section 3.3), SVVP (Section 3.10), and SCMP (Section 3.11), respectively.

The following activities described within IEEE Std 1228-1994 (Reference 24), Clause 4.3 and NUREG/CR-6101 Section 3.1.5 are not applicable to the Basic Software for the following reasons:

- Purchased Software  
All Basic Software items are generated and maintained by MELCO. Therefore, there is no purchased software within the Basic Software.
- Subcontract Management  
All Basic Software items are generated and maintained by MELCO. Therefore, there is no subcontract management involved.
- Process Certification  
There is no certification given to the Basic Software.

### **3.9.2 Organization / Responsibilities**

[

]

[

]

### **3.9.3 Risks**

Identified risks or problems that arise from the software safety analysis activities of each phase shall be tracked, controlled and mitigated using the Risk Matrix and Problem List tools described in the SMP (Section 3.1).

### **3.9.4 Execution / Methods**

#### **3.9.4.1 Measurement**

V&V Anomaly Reports shall be prepared as described in the SVVP (Section 3.10) for any identified software safety concerns. The number and severity level of V&V Anomaly Reports shall be measured for each Basic Software life cycle phase.

The Software Safety Analysis section of the V&V Summary Report for each Basic Software life cycle phase shall describe any V&V Anomaly Reports that identify any software safety concerns. The V&V Manager shall periodically assess V&V Anomaly Reports to determine if there are any adverse trends that can affect software safety, and initiate a Nonconformance Report for identifying and implementing corrective actions.

The Design Section Manager shall periodically assess Nonconformance Reports to determine if there are any adverse trends that can affect software safety, and initiate corrective actions as necessary.

Ultimately all safety concerns shall be resolved prior to release of the Basic Software for deployment in any safety application.

### **3.9.4.2 Procedures**

The Design Section shall document the critical safety functions analysis, as described in Section 3.9.5 and the SDP (Section 3.2).

The V&V Team shall perform software safety analysis V&V activities as described in Section 3.9.5 and the SVVP (Section 3.10). The software safety analysis V&V activities shall ensure proper documentation and traceability of all critical safety functions. If the V&V Team detects any software safety concerns, they shall initiate a V&V Anomaly Report and assign it to the responsible section for corrective action, as described in the SVVP (Section 3.10).

### **3.9.5 Software Safety Analysis Activities**

Section 3.9.5.1 describes the multiple software safety analysis activities, which were conducted to generate the documentation submitted to the NRC, to obtain approval of the MELTAC digital platform.

Section 3.9.5.2 describes the software safety analysis activities that will be conducted, going forward, for any proposed changes that are approved by way of the Change Control process described in the SCMP (Section 3.11).

Sections 3.9.5.2.1 through 3.9.5.2.5 describe the software safety analyses for each phase of the MELTAC Basic Software life cycle. For each phase, the software safety analyses ensure the original MELTAC licensing basis is maintained, and that no new safety hazards are introduced.

When combined together, these software safety analysis activities conform to the guidance of NUREG/CR-6101 for software safety analysis activities and to the requirements of IEEE Std 1228-1994 for a software hazards analysis process.

#### **3.9.5.1 Concept Phase Software Safety Analyses**

[



1

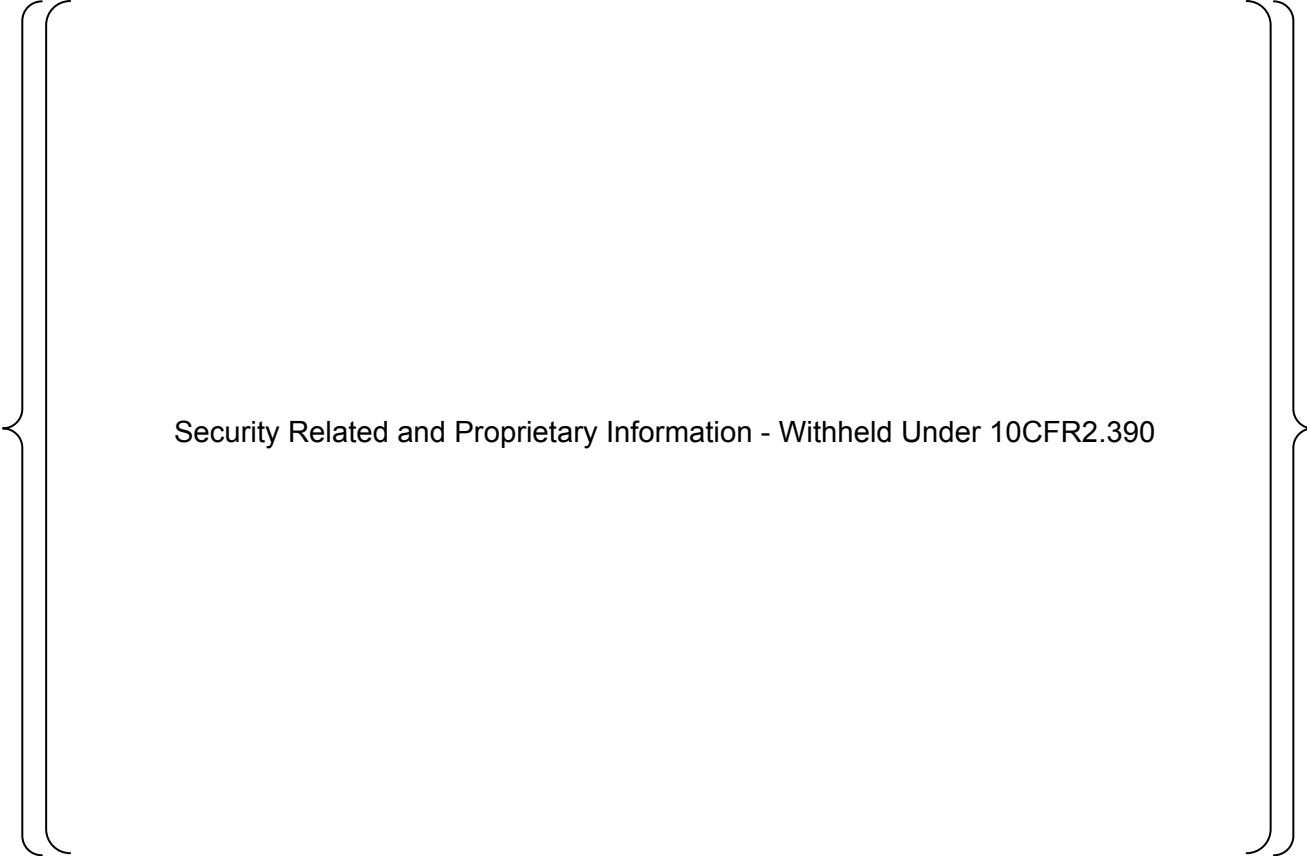
### **3.9.5.1.1 Hazard Analysis**

[

]  
[

]

**Table 3.9-1 Potential Hazards**



Security Related and Proprietary Information - Withheld Under 10CFR2.390

(2) Acceptance Criteria

The potential hazards noted in Table 3.9-1 were checked against the following criteria, as documented in the MELTAC Software Safety Report (JEXU-1015-1009).

**Table 3.9-2 Acceptance Criteria for Function**



Security Related and Proprietary Information - Withheld Under 10CFR2.390

**Table 3.9-3 Acceptance Criteria for Phase**

It is recognized that all hazards may not be mitigated by a single channel controller at the platform level. This recognition is the basis for using parallel-channel or redundant-controller architectures at the system application level. If compliance is addressed at the application level, rather than the MELTAC platform level, it was stated as such in the MELTAC Software Safety Report (JEXU-1015-1009).

**3.9.5.1.2 Response Time Analysis**

[

]

### **3.9.5.1.3 Criticality Analysis**

[

]

**3.9.5.1.4 Diversity and Defense-in-Depth Analysis**

[



]

**3.9.5.1.5 Failure Modes and Effects Analysis**

[

]

**3.9.5.1.6 Reliability Analysis**

[

]

**3.9.5.1.7 Security Analysis**

[

]

**3.9.5.2 Software Safety Analysis Activities for Changes**

[

]

**3.9.5.2.1 Requirements Phase Software Safety Analyses**

[

]

**3.9.5.2.2 Design Phase Software Safety Analyses**

[

]

[

]

### **3.9.5.2.3 Implementation Phase Software Safety Analyses**

[

]

### **3.9.5.2.4 Test Phase Software Safety Analyses**

[

]

**3.9.5.2.5 Post Release**

[

]

**3.9.5.7.4 Retirement and Notification**

The SMaintP (Section 3.6) describes retirement and notification activities. There is no specific safety activity required for MELTAC platform retirement.



### **3.9.6 Standards**

This SSP complies with the following guidance and standards. Methods of compliance against the IEEE standards that are endorsed by Regulatory Guides are tabulated in Appendix B.

- C.3 of RG 1.173
- IEEE Std 1228-1994
- NUREG/CR-6101 (Reference 3)

### **3.10 Software Verification and Validation Plan**

#### **3.10.1 Purpose**

Verification is the process of determining whether the requirements for a system or component are complete and correct, and whether the products of each life cycle phase fulfill or implement the requirements and criteria imposed by the previous phase.

Validation is the process of determining whether the final system or component complies with the specified requirements.

The overall V&V process includes analysis, evaluation, review, inspection, assessment, and testing.

This Software Verification and Validation Plan (SVVP) describes the V&V activities associated with the life cycle phases of the MELTAC Platform Basic Software. The life cycle phases are defined and described in Section 2.0.

All Basic Software supplied with the MELTAC Platform is developed and maintained by MELCO. There are no acquired software products or services used in the Basic Software; therefore the life cycle does not include an Acquisition Phase. The Supply Phase is associated with specific plant projects, and is therefore described in plant-specific Project Plans as described in the SMP (Section 3.1).

This SVVP serves as the output from the Planning Phase described in IEEE Std 1012-1998 (Reference 20).

#### **3.10.2 Organization / Responsibilities**

The V&V activities described in this SVVP are performed by a V&V Team organization made up of individuals that are independent of the Design Section that is made up of design/development/maintenance personnel. The V&V Manager and V&V Team Members shall be technically, organizationally and financially independent of the Design Section as defined in Table 3.10-1.

[

1

**Table 3.10-1 V&V Independence**

Standards Requirements (RG 1.168, IEEE Std 1012-1998)	Interpretation	Company Policy
Technical independence	V&V Team Members have technical competence in software engineering and are individuals not involved in the design of the MELTAC platform. (Note 1)	[ ]
Organizational independence	The V&V Team is not supervised or directed by Design Section members.	[ ]
Financial independence	The V&V Team is not subject to the same budget constraints as the Design Section.	[ ]

(Note 1) "Individuals who are not involved in the design" is defined as those individuals who have never signed in the signature spaces for Drawer, Designer, Reviewer or Approver in the documents produced by the Design Section for the applicable equipment.

The organization of the V&V Team is illustrated by the following diagram:



**Figure 3.10-1 Organizational Structure and Responsibilities / Authorities**

### **3.10.3 Risks**

Identified risks or problems that arise from the V&V activities of each phase shall be tracked, controlled and mitigated using the Risk Matrix and Problem List tools described in the SMP (Section 3.1).

### **3.10.4 Execution / Methods**

The V&V activity shall be conducted in accordance with [ ].

#### **3.10.4.1 Measurement**

[

]

**3.10.4.1.1 Software Integrity Level Scheme**

The Software Integrity Level (SIL) applied to the MELTAC Platform Basic Software is Level 4 as defined in IEEE Std 1012-1998 (per RG 1.168).

**3.10.4.2 Management of V&V**

[

]



**Figure 3.10-2 Overview of MELTAC Platform Basic Software V&V Activities, Tasks and Outputs**

**3.10.4.3 Concept Phase V&V**

[

]

**3.10.4.4 Requirements Phase V&V Activity**

[

]

**3.10.4.4.1 Requirements Phase V&V Tasks**

[



]

**3.10.4.4.2 Requirements Phase V&V Reporting**

(1) Requirements Phase V&V Anomaly Reporting

Anomalies identified by the V&V Team in the Requirements Phase shall be documented, assessed and dispositioned by way of V&V Anomaly Reports as described in Section 3.10.5 of this SVVP.

(2) Requirements Phase V&V Summary Report

After completion of the Requirements Phase V&V activities and disposition of all Requirements Phase V&V Anomaly Reports, a Requirements Phase V&V Summary Report shall be prepared and issued to the V&V Manager for review and approval. The Requirements Phase V&V Summary Report shall be formatted, prepared, checked and approved as described in Section 3.10.5 of this SVVP.

**3.10.4.5 Design Phase V&V Activity**

[

]

#### **3.10.4.5.1 Software Specification V&V Tasks**

[



[

]

#### **3.10.4.5.2 Program Specification V&V Tasks**

[



[

]

#### **3.10.4.5.3 FPGA Specification V&V Tasks**

[





[

]

#### **3.10.4.5.4 Design Phase V&V Reporting**

##### (1) Design Phase V&V Anomaly Reporting

Anomalies identified by the V&V Team in the Requirements Phase shall be documented, assessed and dispositioned by way of V&V Anomaly Reports as described in Section 3.10.5 of this SVVP.

##### (2) Design Phase Summary V&V Report

After completion of the Design Phase V&V activities and disposition of all Design Phase V&V Anomaly Reports, a Design Phase V&V Summary Report shall be prepared and issued to the V&V Manager for review and approval. The Design Phase V&V Summary Report shall be formatted, prepared, checked and approved as described in Section 3.10.5 of this SVVP.

#### **3.10.4.6 Implementation Phase V&V Activity**

The processor software source code and the FPGA software source code, generated by the Design Section during the Implementation Phase, shall be verified and validated as described below.

##### **3.10.4.6.1 Processor Software Source Code V&V Tasks**

[



[

1

### **3.10.4.6.2 FPGA Software Source Code V&V Tasks**

[



[  
.



]

### **3.10.4.6.3 Implementation Phase V&V Reporting**

#### (1) Anomaly Reporting

Anomalies identified by the V&V Team in the conduct of Implementation Phase V&V activities shall be documented, assessed and dispositioned by way of V&V Anomaly Reports as described in Section 3.10.5 of this SVVP.

#### (2) Implementation Phase Summary V&V Report

After completion of the Implementation Phase V&V activities and disposition of all Implementation Phase V&V Anomaly Reports, an Implementation Phase V&V Summary Report shall be prepared and issued to the V&V Manager for review and approval. The V&V Summary report shall be formatted, prepared, checked and approved as described Section 3.10.5 of this SVVP.

### **3.10.4.7 Test Phase V&V Activity**

This section describes the Test Phase V&V activity in the context of IEEE Std 1012-1998. The methods used to produce test specifications, test designs, test cases, test reports and the tools that are used in various test activities are Overall test activities are described in the STP (Section 3.12).

#### **3.10.4.7.1 Test Phase V&V Tasks**

[

1

[

]

#### **3.10.4.7.2 Prepare Final V&V Report**

The V&V Team shall issue a Final V&V Report after completion of the V&V activities as described in Section 3.10.5.2.

#### **3.10.4.8 Installation Phase V&V Activity**

The installation phase in the context of this SPM is part of the hardware manufacturing process conducted in the production area (i.e. MELTAC factory environment). Installation of the Basic Software into target hardware modules is described in the SInstP (Section 3.5). There are no V&V activities associated with installation of the Basic Software during manufacturing of specific MELTAC hardware modules (this is a manufacturing inspection activity performed under the MELCO QAP).

The released version of the Basic Software produced under this SPM is confirmed by the Application V&V Team during the Installation Phase V&V Activity as described in Section 3.10 of the Application Software Program Manual (MUAP-07017).

#### **3.10.4.9 Operations Phase V&V Activity**

This phase covers systems in the actual plant and is not applicable to the development process of the Basic Software in the MELTAC Platform.

#### **3.10.4.10 Maintenance Phase V&V Activity**

[

]

### 3.10.5 V&V Reporting Documentation Requirements

All V&V activities and task results shall be documented, including the personnel and procedures used to conduct the activities. V&V reports shall be prepared at the conclusion of each V&V task or activity as described within this SVVP.

The V&V reports to be prepared shall consist of the following:

- Task-specific documents and reports as called out in this SVVP
- V&V Phase Summary Reports
- V&V Anomaly Reports
- Final V&V Report
- Regression Analysis Report (for proposed changes)

#### (1) V&V Anomaly Reports

A V&V anomaly is anything observed in the documentation or operation of the software that deviates from expectations based on this SVVP, V&V reference documents (i.e., the documents to which V&V Inputs are compared), or previous technical experience and/or calculations.

The V&V Manager shall provide V&V Anomaly Reports to the Design Section Manager or other responsible manager for evaluation, resolution, and disposition. The V&V Manager shall review the final disposition of each V&V Anomaly Report and determine if it is complete, correct, and appropriate for the identified V&V anomaly.

Any detected V&V Anomalies shall be documented by way of one or more V&V Anomaly Reports in each phase of the life cycle. V&V Anomaly Reports shall contain the following information, as a minimum:

- a. Anomaly Report number & date
- b. Project name, number and applicable phase
- c. The date the anomaly was detected
- d. The name of the V&V Team Member that detected the anomaly

- e. The V&V Activity and Task that were underway when the anomaly was detected
- f. The V&V Input document that was being verified or validated
- g. The V&V reference document that was being used for the V&V task (i.e., upstream document)
- h. A detailed description of the anomaly
- i. The severity level of the V&V anomaly (see Section 3.10.4.1 of this SVVP)
- j. The date the V&V Anomaly Report was sent to the responsible manager for resolution
- k. The final disposition of the anomaly, including documents and/or software that were affected, and the V&V activities and tasks that were performed
- l. The date when the V&V Manager accepts the final disposition

## (2) V&V Phase Summary Reports

V&V phase summary reports shall contain, as a minimum:

- a. Summary report number & date
- b. Project name, number and applicable phase
- c. List of input documents reviewed (title, number and revision number)
- d. Phase-specific deviations from SPM as noted in the Project Plan
- e. Description of specific V&V activities, tasks, analyses and results
- f. List of V&V Phase output documents
- g. Summary of reported anomalies and their dispositions
- h. Summary of identified risks and problems
- i. Summary of lessons learned

### 3.10.5.1 Requirements Traceability Matrix

[

]

### **3.10.5.2 Final V&V Report**

Upon completion of the V&V activities for a given design/development/maintenance project, the V&V Manager shall prepare and issue a Final V&V Report that describes the V&V phase-specific activities, tasks, results, disposition of V&V Anomaly Reports, disposition of any significant Corrective Action Reports, and lessons learned.

The Final V&V Report shall also provide an assessment of the overall software project and recommendations, if needed, for updating this SVVP, the SPM, or implementing procedures.

The Final V&V Report shall include the following, as a minimum:

- Summary of V&V activities at all phases
- Summary of V&V task results
- Summary of V&V Anomalies and resolutions, including the number and severity of anomalies
- Assessment of overall software project
- Lessons learned/best practices
- Recommendations

### **3.10.6 V&V Administrative Requirements**

#### **3.10.6.1 Anomaly Resolution and Reporting**

[

]

#### **3.10.6.2 Task Iteration Policy**

[

]

**3.10.6.3 Deviation Policy**

[

]

**3.10.6.4 Control Procedures**

V&V Activity records shall be controlled as described in the SCMP (Section 3.11).

**3.10.6.5 V&V Test Documentation Requirements**

V&V test documentation requirements are described in the STP (Section 3.12).

**3.10.7 Methods / Tools****3.10.7.1 V&V Procedures**

The procedures that implement the requirements of this SVVP shall include a check sheet, including check results against acceptance criteria. Check sheets and results shall be documented in the associated V&V Output document or V&V Phase Summary Report. V&V results include these check sheets and results.

V&V procedures shall also list the interfacing procedures for record retention and corrective action reporting.

**3.10.8 Standards**

This SVVP complies with the following guidance and standards. Methods of compliance against the IEEE standards that are endorsed by Regulatory Guides are tabulated in Appendix B.



- (1) Clause 5.3 of IEEE Std 7-4.3.2-2003 which is endorsed by RG 1.152
- (2) IEEE Std 1012-1998 which is endorsed by RG 1.168 (Reference 6)
- (3) Section 3.1.4 of NUREG/CR6101-1993

### **3.11 Software Configuration Management Plan (SCMP)**

#### **3.11.1 Purpose**

This Software Configuration Management Plan (SCMP) describes the methods for identifying Basic Software configuration items, controlling the implementation and changes to Basic Software, and recording and reporting the status of changes. This SCMP assures that configuration items contain the necessary configuration control attributes, including security attributes, for software configuration items, including software documents.

{

Security Related Information - Withheld Under 10CFR2.390

}

This SCMP complies with the guidance and standards identified in Section 3.11.4.

The configuration management program for application software is described in Section 3.11 of the Application SPM (MUAP-07017). When application-related documentation, hardware or software is delivered to an end-user, it shall be maintained from that point forward under the user's configuration management program. However, documentation of the configuration of all documents, software and hardware related to the MELTAC Basic Software shall retained by MELCO as QA Records as described in the SQAP (Section 3.3).

#### **3.11.2 Management**

##### **3.11.2.1 Organization / Responsibilities**

[

]

**3.11.2.2 Schedule/Resource**

Configuration management activities described in this SCMP shall be included in the project Master Schedule. The Master Schedule is described in the SMP (Section 3.1).

The Design Section Manager shall identify and document the specific configuration management activities and necessary personnel for each life cycle phase in the Project Plan. As a minimum, the following information shall be included in the Project Plan:

- (1) Design Section personnel who are involved in the configuration management activities
- (2) Members of the Configuration Control Board (See Section 3.11.2.3)
- (3) Schedule for submitting outputs to the V&V Team in each phase
- (4) Schedule for Design Section reviews of the identified configuration management activities.
- (5) Schedule for releasing the Basic Software

**3.11.2.3 Configuration Control Board (CCB)**

The CCB activities described in this SCMP apply only to the Basic Software.

The Design Section Manager is responsible for calling a CCB meeting if a Software Change Request (SCR) meets any of the following conditions:

- (1) When a proposed change can affect Basic Software functions described in Concept Phase, Requirements Phase or Design Phase output documents
- (2) When adding new functions to the Basic Software
- (3) When a proposed change will affect Basic Software releases due to changes in its hardware environment (e.g., hardware changes due to discontinued products or upgraded products)
- (4) When proposed changes to the Basic Software is due to changes in external interfaces
- (5) When a change is proposed during the Test Phase

[

]

### **3.11.3 Execution / Methods**

#### **3.11.3.1 Measurement**

[

]

#### **3.11.3.2 Procedures**

The Design Section shall perform the following configuration management tasks, using the tools identified below:

[

]

**3.11.3.2.1 Configuration Items**

The following configuration items shall be managed and controlled as described within this SCMP:

[

]

**3.11.3.2.2 Purchased Software**

[

]

**3.11.3.2.3 Software Library**

[

]

#### **3.11.3.2.4 Configuration Item Naming and Labeling**

[

]

#### **3.11.3.2.5 Configuration Item Baselines**

[



]

**3.11.3.2.6 Status Accounting**

[

]

**3.11.3.2.7 Reviews and Audits**

[

]  
**3.11.3.2.8 Change Control**  
[

]  
**3.11.3.2.8.1 SCR Initiation**  
[

]  
**3.11.3.2.8.2 SCR Evaluation**  
[

]

**3.11.3.2.8.3 SCR Approval or Disapproval**

[

]

**3.11.3.2.8.4 SCR Implementation**

[

]

### **3.11.3.2.9 Software Release Process**

[

]

### **3.11.3.2.10 Problem Reporting**

[

]

**3.11.3.2.11 Record Keeping**

[

]

**3.11.4 Standards**

This SCMP complies with the following guidance and standards. Methods of compliance against the IEEE standards that are endorsed by Regulatory Guides are tabulated in Appendix B.

- IEEE Std 1074-1995, which is endorsed by RG 1.173
- IEEE 828-1990 (Reference 16), which is endorsed by RG 1.169 (Reference 7)
- IEEE 7-4.3.2-2003, which is endorsed by RG 1.169
- NUREG/CR-6101

## **3.12 Software Test Plan**

### **3.12.1 Purpose**

The purpose of the Software Test Plan (STP) is to describe the testing activities that are performed on the Basic Software. The SVVP (Section 3.10) and the STTrngP (Section 3.7) complement each other; the SVVP describes the required test activities and test documents to be performed by the V&V Team, and the STP provides additional details on specific test methods and tools for each test activity and the minimum required content for test documents.

Installation checks are described in section 3.5.3 and the regression analysis are described in section 3.6.3 for error corrections.

Plant system application testing is outside the scope of this SPM.

### **3.12.2 Organization / Responsibilities**

All testing of the Basic Software is performed by the V&V Team as described in section 3.10.2.

### **3.12.3 Execution / Methods**

Testing of the Basic Software is performed as part of the V&V test activities (Unit V&V Test and Integration V&V Test) as described in section 3.10.4.

The specific test description document conforms to the guideline of IEEE Std 829-1983 (Reference 17).

#### **3.12.3.1 Alignment with IEEE Std 1012-1998 Testing Activities**

IEEE Std 1012-1998 (Reference 20) addresses the following four testing activities:

(1) Component V&V Testing

This test activity corresponds to the following two types of Unit V&V testing, as required by the SVVP:

- Processor Software Unit V&V Test
- FPGA Software Unit V&V Test

Details of the Unit V&V Test are provided in Section 3.12.3.2.

(2) Integration V&V Testing

This test activity corresponds to the Integration V&V Test as required by the SVVP. The Integration V&V Test is conducted to verify that functions and performance satisfy the requirements in the Platform Specification (including cyber security features) and the Software Specification by running the Basic Software on the MELTAC Platform hardware.

Details of the Integration V&V Test are provided in Section 3.12.3.3.

### (3) System Testing

This test activity is included in the Integration V&V Test described above.

### (4) Acceptance Testing

The MELTAC platform is not by itself subject to Acceptance Testing in the context of customer acceptance as described in BTP 7-14 or IEEE Std 1012-1998. Acceptance testing of a solution developed for a plant-specific project is an activity that occurs in the course of application development.

Alignment with IEEE Std 1012-1998 Testing Activities is shown in Table 3.12-1.

**Table 3.12-1 Alignment with IEEE Std 1012-1998 Testing Activities**

IEEE Std 1012-1998 Testing Activity	Testing Activity for Basic Software
Component Testing	Unit V&V Test (Section 3.12.3.2 "Unit V&V Test")
Integration Testing	Integration V&V Test (Section 3.12.3.3 "Integration V&V Test")
System Testing	
Acceptance Testing	N/A

### 3.12.3.2 Unit V&V Test Activities

#### 3.12.3.2.1 Processor Software Unit V&V Test Activity

[

]

**3.12.3.2.2 FPGA Software Unit V&V Test**

[



]



**Figure 3.12-1 FPGA Unit V&V test workflow**

### **3.12.3.3 Integration V&V Test Activities**

Integration V&V tests shall be performed as described in the SVVP (Section 3.10) and the STP (Section 3.12) to validate that the functions and performance of the integrated platform meet the requirements in the Platform Specification (including cyber security features) and the Software Specification.

The V&V Team shall prepare an Integration V&V Test Specification, execute the Integration V&V Test, and prepare an Integration V&V Test Report as described in the SVVP and this STP. The Integration V&V Test Specification and Integration V&V Test Report shall conform to the guidance of IEEE Std 829-1983.

**3.12.3.3.1 Integration V&V Test Activity Workflow**

{ [

Security Related and Proprietary Information - Withheld Under 10CFR2.390

] }

**3.12.3.3.2 Integration V&V Test Characteristics**

{ [

Security Related and Proprietary Information - Withheld Under 10CFR2.390

Security Related and Proprietary Information - Withheld Under 10CFR2.390

1}

{ [

Security Related and Proprietary Information - Withheld Under 10CFR2.390

] }

#### **3.12.3.3.3 Integrated System Security**

{ [

Security Related and Proprietary Information - Withheld Under 10CFR2.390

] }

#### **3.12.3.4 Test Documents**

Documentation of Unit V&V tests and Integration V&V tests shall be in accordance with IEEE Std 829-1983 and 1008-1987.

**Table 3.12-2 Alignment with IEEE Std 829-1983 Software Test Document**

IEEE Std 829-1983 Software Test Document	MELTAC Platform Basic Software Test Document
Test Plan	V&V Test Description
Test Design Specification	V&V Test Specification
Test Case Specification	
Test Procedure Specification	
Test Summary Report	V&V Test Report
Test Item Transmittal Report	
Test Log	
Test Incident Report	V&V Anomaly Report

## (1) V&amp;V Test Descriptions

V&V Test Descriptions for processor and FPGA software tests shall contain the following items as a minimum:

- Test description name, number and revision level
- Project name, number and applicable phase
- Applicable procedures
- Test overview
- Configuration items to be tested
- Item pass/fail criteria
- Test deliverables
- Environmental needs (space, tools and equipment)
- Organizational structure
- Schedule

## (2) V&amp;V Test Specifications

V&V Test specifications that establish the test designs, test cases and test for processor and FPGA software tests, and shall contain the following items as a minimum:

- Test specification name, number and revision level
- Project name, number and applicable phase
- Applicable procedures

- Computer program, module or unit name and version to be tested
- Test design description
- Test case description
- Specific test procedure steps
- Required input values
- Expected output values
- Acceptance criteria
- Test environment (test equipment and software tools)

### (3) V&V Test Reports

V&V Test reports shall document test results and evaluations, and shall contain the following items as a minimum:

- Test Report name, number, and revision level
- Project name, number and applicable phase
- Applicable procedure
- Computer program, module or unit name and version tested
- Computer hardware used
- Test equipment serial number and calibrations
- Software tools used, including version number
- Persons performing the test or recording the test data
- Test results and acceptability (against acceptance criteria)
- Person evaluating test results
- Test data used and/or produced
- List of identified V&V Anomaly Reports
- Date of test

### (4) V&V Anomaly Reports

Any anomalies detected by the V&V Team during the Test Phase shall be documented by way of a V&V Anomaly Report. The requirements for V&V Anomaly Reports are provided in Section 3.10.5.

Each V&V Test document shall be reviewed by a V&V Team member who did not participate in its preparation.

The V&V Team checks that the software version to be tested is consistent with the one indicated in the software configuration management documentation (as described in Section 3.11.3.2.9). The V&V Team shall identify the intended version and indicate the results of the version check in the related V&V Test Report.

### **3.12.3.5 Test Reporting**

The V&V Team shall conduct the V&V testing in accordance with the V&V Test Specification, and produce the following test reports:

- V&V Test Report
- V&V Anomaly Report

The configuration items under test shall pass the test when the results anticipated in the V&V Test Specification meet the acceptance criteria and are obtained without any unexpected developments in the course of the test.

When any acceptance criterion is not met, or the test cannot continue as expected, a V&V Anomaly Report shall be issued. If the Design Section submits the response that indicates the result in question is acceptable in terms of the specification, the V&V Team shall review and determine adequacy of the response. If the V&V Team determines the disposition is adequate, and the V&V Team Manager approves the V&V Anomaly Report, the as-tested configuration items shall be considered acceptable.

### **3.12.3.6 Test Tools**

[



]

### **3.12.4 Measurement**

Measurement is described in Section 3.10.4.1.

### **3.12.5 Methods / Tools**

The methods used to carry out the testing of the Basic Software are described in Section 3.10.7.

### **3.12.6 Record Keeping**

The test documents, data, and programs described in this STP shall be kept under configuration management as described in Section 3.11.3.2.11. All revisions of the test documents shall be retained as QA Records in accordance with [ ]].

Test data and programs used shall be subject to configuration management in accordance with the SCMP (Section 3.11) and shall be stored in CEAS.

### **3.12.7 Standards**

This STP complies with the following guidance and standards. Methods of compliance against the IEEE standards that are endorsed by Regulatory Guides are tabulated in Appendix B.

- (1) IEEE Std 829-1983 which is endorsed by RG 1.170 (Reference 8)
- (2) IEEE Std 1008-1987 which is endorsed by RG 1.171 (Reference 9)

#### 4.0 REFERENCES

1. NUREG-0800, BTP 7-14 Revision 5 "Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control System", March 2007
2. 10 CFR Part 50, Appendix B, "Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants"
3. NUREG/CR-6101, "Software Reliability and Safety in Nuclear Reactor Protection Systems" 1993. BTP 7.14-68 Revision 5 – March 2007
4. Regulatory Guide 1.152, Revision 2, "Criteria for Use of Computers in Safety Systems of Nuclear Power Plants" Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 2006 (ML053070150)
5. Regulatory Guide 1.153, Revision 1, " Criteria for Safety Systems " Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 1996
6. Regulatory Guide 1.168, Revision 1, "Verification, Validation, Reviews and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants", Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 2004 (ML040410189)
7. Regulatory Guide 1.169, "Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants", Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 1997 (ML003740102)
8. Regulatory Guide 1.170, "Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants", Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 1997 (ML003740105)
9. Regulatory Guide 1.171, "Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants", Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 1997 (ML003740108)
10. Regulatory Guide 1.172, "Software Requirements Specifications for Digital Computer Software Used in Safety Systems of Nuclear Power Plants", Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 1997 (ML003740094)
11. Regulatory Guide 1.173, "Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants", Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission, 1997 (ML003740101)
12. ANSI/ASME NQA-1-1983, "Quality Assurance Program Requirements for Nuclear Facilities"

13. IEEE Std 7-4.3.2-2003, " IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations"
14. IEEE Std 603-1991, " IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations Sponsor"
15. IEEE Std 730-2002, "IEEE Standard for Software Quality Assurance Plans"
16. IEEE Std 828-1990, "IEEE Standard for Software Configuration Management Plans"
17. IEEE Std 829-1983, "IEEE Standard for Software Test Documentation"
18. IEEE Std 830-1993, "IEEE Recommended Practice for Software Requirements Specifications"
19. IEEE Std 1008-1987, "IEEE Standard for Software Unit Testing"
20. IEEE Std 1012-1998, "IEEE Standard for Software Verification and Validation Plans"
21. IEEE Std 1028-1997, "IEEE Standard for Software Reviews and Audits"
22. IEEE Std 1042-1987, "IEEE Guide for Software Configuration Management"
23. IEEE Std 1074-1995, "IEEE Standard for Developing Software Life Cycle Processes"
24. IEEE Std 1228-1994, "IEEE Standard for Software Safety Plans"
25. Technical Report MUAP-07005 R6, "Safety System Digital Platform –MELTAC–"
26. Technical Report MUAP-07004 R5, "Safety I&C System Description and Design Process"
27. Technical Report MUAP-07017 R2, "US-APWR Software Program Manual"
28. JEXU-1022-6301, "MELTAC Re-evaluation Program Report"
29. Topical Report MUAP-07006 R2, "Defense-in-Depth and Diversity"
30. Technical Report MUAP-07014 R2, "US-APWR Defense-in-Depth and Diversity Coping Analysis"
31. JEXU-1015-1009, "MELTAC Platform Basic Software Safety Report"
32. ESC Procedure N-G000, "Quality Assurance Manual for U.S. Nuclear facility Applications"
33. ESC Procedure N-0100, "Organization Procedure (NQA-1)"

34. ESC Procedure N-0200, "Quality Assurance Activity Management Procedure (NQA-1)"
35. ESC Procedure N-0300, "Design Control Procedure (NQA-1)"
36. ESC Procedure N-0352, "N-0352, "Safety System Software V&V Procedure (NQA-1)"
37. ESC Procedure N-1500, "Nonconforming Items Control Procedure (NQA-1)"
38. ESC Procedure N-1600, "Corrective Action Procedure (NQA-1)"
39. ESC Procedure N-1700, "Quality Assurance Record Control Procedure (NQA-1)"
40. ESC Procedure N-3000, "Procedure for Reporting of Defects and Noncompliance (10 CFR 21) (NQA-1)"

---

## APPENDIX A Definition

### Basic Software

The MELTAC Platform Basic Software is low-level software that operates the MELTAC controllers. This software consists of initialization code, device drivers, communication layers, function blocks, diagnostics, etc. for the MELTAC platform.

### Coding

In software engineering, the process of expressing a computer program in a programming language.

(2) (IEEE Std 1002-1987 [9]) The transforming of logic and data from design specifications (design descriptions) into a programming language.

### Compiler

A computer program that translates programs expressed in a high order language into their machine language equivalents.

### Concept Phase

The initial phase of a software development project, in which the user needs are described and evaluated through documentation.

### Configuration Control

An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification.

### Consistency

The degree of uniformity, standardization, and freedom from contradiction among the documents or parts of a system or component.

### Control Network

The Control Network is a MELTAC dedicated ring topology network and communicates plant process data and control signal data with a deterministic periodic cycle.

### Corporate Electric Archive System(CEAS)

CEAS is safe data storage system in MELCO.

### Data

A representation of facts, concepts, or instructions in a manner suitable for communication, interpretation, or processing by humans or by automatic means.

### Data Link

The Data Link communication is used to transmit process signals between controllers of different safety divisions. This communication is unidirectional.

### Delivery

Release of a system or component to its customer or intended user.

### Deviation

A departure from a specified requirement.

**Documentation**

- (1) A collection of documents on a given subject.
- (2) Any written or pictorial information describing, defining, specifying, reporting, or certifying activities, requirements, procedures, or results.

**Documentation Tree**

A list of documents related to a particular project. Documents are systematically identified in this Documentation Tree. It is prepared to identify input and output documents of the product design.

**Drawing**

A document that shows information needed for implementation of hardware modules, units and cabinets. Drawings include circuit diagrams, parts information, assembling sequences and wire connecting diagrams.

**Emulator**

A device, computer program, or system that accepts the same inputs and produces the same outputs as a given system.

**Error**

- (1) The difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. For example, a difference of 30 meters between a computed result and the correct result.
- (2) An incorrect step, process, or data definition. For example, an incorrect instruction in a computer program.
- (3) An incorrect result. For example, a computed result of 12 when the correct result is 10.
- (4) A human action that produces an incorrect result.

**Executable Module**

The module is generated by using compiler and linker. The Executable Module is installed into the hardware as preparation of the integration test.

**Fail-Safe Mode**

The reactor trip function takes fail-safe actions in response to failures.

**Factory Acceptance Testing**

Testing conducted to verify the correct implementation of the design and compliance with program requirements for one software element (e.g., unit, module) or a collection of software elements.

**Firmware**

Computer programs and data loaded in a class of memory that cannot be dynamically modified by the computer during processing.

**FPGA**

Abbreviation of field programmable gate array. FPGA has many internal logical blocks consisting of logic gates and arithmetic circuits. Internal logical blocks are located on the matrix. Required circuit configuration can be realized by connecting these internal logical blocks.

**Function**

- (1) A defined objective or characteristic action of a system or component. For example, a system may have inventory control as its primary function.
- (2) A software module that performs a specific action, is invoked by the appearance of its name in an expression, may receive input values, and returns a single value.

**Functional Testing**

- (1) Testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions.
- (2) Testing conducted to evaluate the compliance of a system or component with specified functional requirements.

**F-ROM**

Abbreviation of flash read only memory. F-ROM is also called flash memory. One of the nonvolatile semiconductor memories in which data do not disappear even after a device is turned off.

**Hardware**

Physical equipment used to process, store, or transmit computer programs or data.

**Identifier**

The name, address, label, or distinguishing index of an object in a computer program.

**Input**

Pertaining to a device, process, or channel involved in receiving data from an external source.

**Integration Testing**

An orderly progression of testing of incremental pieces of the software program in which software elements, hardware elements, or both are combined and tested until the entire system has been integrated to show compliance with the program design, and capabilities and requirements of the system.

**Interdivision data**

Data that are transferred between trains (divisions) via data link communication.

**Linker**

A computer program that creates a single load module from two or more independently translated object modules or load modules by resolving cross-references among the modules and, possibly, by relocating elements.

**Maintenance**

- (1) The process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment.
- (2) The process of retaining a hardware system or component can be retained in, or restored to, a state in which it can perform its required functions.

**Maintenance Network**

The Maintenance Network is used to communicate between the controllers and the MELTAC Engineering Tools to download new application software to the controllers, or to read/write inside memory of the controller.

**MELCO**

Mitsubishi Electric Corporation

**MELCO Corporate Electronic Archive System (CEAS)**

A system that stores software modules, documents and all other items under Configuration Management. Each item in the CEAS have a unique number for control, research and retrieval.

**MELTAC controller**

An exclusive device for MELTAC

**MELTAC Engineering Tool**

Software operating on a computer that generates application programs and installs them on the MELTAC platform. This tool also offers maintenance functions including indication of MELTAC failures and status.

**MELTAC Platform Basic Software**

The MELTAC Platform Basic Software is low-level software that operates the MELTAC controllers. This software consists of initialization code, device drivers, communication layers, function blocks, diagnostics, etc. for the MELTAC platform. This may be called Basic Software.

**MELTAC Re-evaluation Program (MRP)**

Activities to reevaluate the MELTACs that were developed under the old QAP before the current MELCO QAP is established on the basis of 10 CFR 50 Appendix B, in order to ensure that they satisfy the quality commensurate with those required by the current QAP (based on 10 CFR 50 Appendix B).

**MELTAC Technical Report**

Safety System Digital Platform -MELTAC-

**Metric**

A quantitative measure of the degree to which a system, component, or process possesses a given attribute.

**Module**

- (1) A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to, or output from, an assembler, compiler, linkage editor, or executive routine.
- (2) A logically separable part of a program.

**Nonvolatile memory**

Computer memory that can retain the stored information even when not powered.

**Operation and Maintenance Phase**

The period of the time in the Software Life Cycle during which a software product is employed in its operational environment, monitored for satisfactory performance, and modified as necessary to correct problems or to respond to changing requirements.

**Output**

Pertaining to a device, process, or channel involved in transmitting data to an external destination.

**POL**

Abbreviation of problem oriented language. A control language used in the MELCO's instrumentation controllers for nuclear power plants.



**Procedure**

- (1) A course of action to be taken to perform a given task.
- (2) A written description of a course of action as in (1); for example, a documented test procedure.
- (3) A portion of a computer program that is named and that performs a specific action.

**Project Plan**

A document that identifies information necessary to plan a system design and development project, including specification, organization, procedures, process, facilities, equipment and materials to be used, and inspection methods.

**Quality Assessment Index**

An index determined by evaluating how each function requirement is satisfied.

**Quality Assurance**

- (1) A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.
- (2) A set of activities designed to evaluate the process by which products are developed or manufactured.

**Regression Analysis**

An analysis that evaluates the influence of the modification of the design or implementation activity, and determines the verification and validation items to be performed.

**Requirement**

- (1) A condition or capability needed by a user to solve a problem or achieve an objective.
- (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.

**Requirements Traceability Matrix (RTM)**

A document that is used to ensure that the software specifications required by the Platform Specification are correctly incorporated to the Software Specification, Program Specification, source code, and Test Specification.

**ROM writing tool**

A tool to write data of execution module on UV-ROM, FPGA and flash ROM.

**Safety VDU**

Equipment used to monitor and control the safety system.

**Self-diagnosis**

The integrity of digital I&C components is continuously checked by their self-diagnosis features. These self-diagnostic features result in early detection of failures.

**Simulation**

A model that behaves or operates like a given system when provided a set of controlled inputs.

**Software**

Computer programs, procedures, and in some cases, associated documentation and data pertaining to the operation of a computer system.

**Software Design Description**

A representation of software created to facilitate analysis, planning, implementation, and decision making. The software design description is used as a medium for communicating software design information, and may be thought of as a blueprint model of the system.

**Software Error**

An error in the software

**Software Hazard**

A software error that could result in failure of functions or unintended operation including abnormal events, conditions and malicious modifications.

**Software item**

Source code, object code, job control code, control data, or a collection of these items.

**Software Library**

A controlled collection of software and related documentation designed to aid in software development, use, or maintenance. Types include master library, production library, software development library, software repository, and system library.

**Software Life Cycle**

The period of time that begins when a software product is conceived and ends when the software is no longer available for use.

**Software linker tool**

A program that combines pieces of the machine language program into an executable program.

**Software Risk**

A measure that combines both the likelihood that a software hazard will cause some problem and the severity of that problem.

**Source Code**

Computer instructions and data definitions expressed in a form suitable for input to an assembler, compiler, or other translator.

**Specification**

A document that specifies, in a complete, precise, verifiable manner, the requirement, design, behavior, or other characteristics of a system or component, and often, the procedure for determining whether these provisions have been satisfied.

**Standards**

Mandatory requirements employed and enforced to prescribe a disciplined uniform approach to software development, that is, mandatory conventions and practices are in fact standards.

**Static Analysis**

A method of computer software analysis where the execution file is not actually executed, but is analyzed.

**Structural Testing**

Testing that takes into account the internal mechanism of a system or component. Types include branch testing, path testing, statement testing.

**System Requirements Specification**

A document that specifies functional and non-functional requirements of the system.

**System Software**

Software designed to facilitate the operation and maintenance of a computer system and its associated programs; for example, operating systems, assemblers, utilities.

**Test Case**

A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

**Test Design**

Documentation specifying the details of the test approach for a software feature or combination of software features and identifying the associated tests.

**Test Phase**

The period of time in the Software Life Cycle during which the components of a software product are evaluated and integrated, and the software product is evaluated to determine whether or not requirements have been satisfied.

**Test Plan**

A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning.

**Test Report**

A document that describes the conduct and results of the testing carried out for a system or component.

**Test Unit**

A set of one or more computer program modules together with associated control data, (for example, tables), usage procedures, and operating procedures that satisfy the following conditions:

- (1) All modules are from a single computer program.
- (2) At least one of the new or changed modules in the set has not completed the unit test.

The set of modules together with its associated data and procedures are the sole object of a testing process.

**Transfer**

To send data from one place and receive it at another.

**Unit**

- (1) A separately testable element specified in the design of a computer software component.
- (2) A logically separable part of a computer program.
- (3) A software component that is not subdivided into other components.

**Unit Testing**

Testing of individual hardware or software units or groups of related units.

**UV-ROM**

Abbreviation of ultraviolet erasable programmable read only memory that is a PROM, where data can be erased by ultraviolet light.

**Validation**

The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. Contrast with: verification.

**Variable**

A quantity or data item whose value can change; for example, the variable Current\_time.

**Verification**

The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. Contrast with: validation.

**Verification and Validation**

The process of determining whether the requirements for a system or component are complete and correct, the products of each development phase fulfill the requirements or conditions imposed by the previous phase, and the final system or component complies with specified requirements.

**Version**

An initial release or re-release of a computer software configuration item that is associated with a complete compilation or recompilation of the computer software configuration item.

**Verilog-HDL**

A hardware description language used in the FPGA design.

**V&V Task Manual**

A V&V Task Manual provides (1) a description of the V&V activities and tasks, (2) identification of specific V&V Team Members, (3) identification of specific internal procedures and (4) a detailed schedule.

## APPENDIX B Conformance to BTP 7-14 and IEEE standards

This SPM provides the software program plans which conform to the guidance of Branch Technical Position (BTP) 7-14. The BTP is derived from review process described in SRP.

This Appendix B summarizes the compliance matrices for BTP 7-14 and endorsed IEEE standards. In the column "MELTAC Platform Basic Software SPM section" of each table, the applicable section of this SPM is described. If a certain requirement does not apply based on MELCO evaluation, it is shown as "N/A to this SPM".

The contents of this Appendix B are shown in the following table.

<u>Table</u>	<u>Regulation</u>	<u>IEEE/BTP</u>	<u>Page</u>
B-1	NUREG/0800	BTP 7-14	171
B-2	RG 1.152 Rev.2	IEEE Std 7-4.3.2-2003	176
B-3	RG 1.153 Rev.1	IEEE Std 603-1991	180
B-4	RG 1.169	IEEE Std 828-1990	184
B-5	RG 1.170	IEEE Std 829-1983	190
B-6	RG 1.172	IEEE Std 830-1993	194
B-7	RG 1.171	IEEE Std 1008-1987	199
B-8	RG 1.168 Rev.1	IEEE Std 1012-1998	201
B-9	RG 1.168 Rev.1	IEEE Std 1028-1997	208
B-10	RG 1.169	IEEE Std 1042-1987	216
B-11	RG 1.173	IEEE Std 1074-1995	222

**Appendix B-1 Compliance Matrix for NUREG-0800 BTP-7-14 Revision 5**



**Appendix B-1 Compliance Matrix for NUREG-0800 BTP-7-14 Revision 5**



**Appendix B-1 Compliance Matrix for NUREG-0800 BTP-7-14 Revision 5**





**Appendix B-1 Compliance Matrix for NUREG-0800 BTP-7-14 Revision 5**



**Appendix B-1 Compliance Matrix for NUREG-0800 BTP-7-14 Revision 5**



**Appendix B-2 Compliance Matrix for IEEE Std 7-4.3.2-2003 endorsed by RG 1.152 Rev.2**



**Appendix B-2 Compliance Matrix for IEEE Std 7-4.3.2-2003 endorsed by RG 1.152 Rev.2**




**Appendix B-2 Compliance Matrix for IEEE Std 7-4.3.2-2003 endorsed by RG 1.152 Rev.2**

--	--

**Appendix B-2 Compliance Matrix for IEEE Std 7-4.3.2-2003 endorsed by RG 1.152 Rev.2**

--	--

**Appendix B-3 Compliance Matrix for IEEE Std 603-1991 endorsed by RG 1.153 Rev.1**




**Appendix B-3 Compliance Matrix for IEEE Std 603-1991 endorsed by RG 1.153 Rev.1**

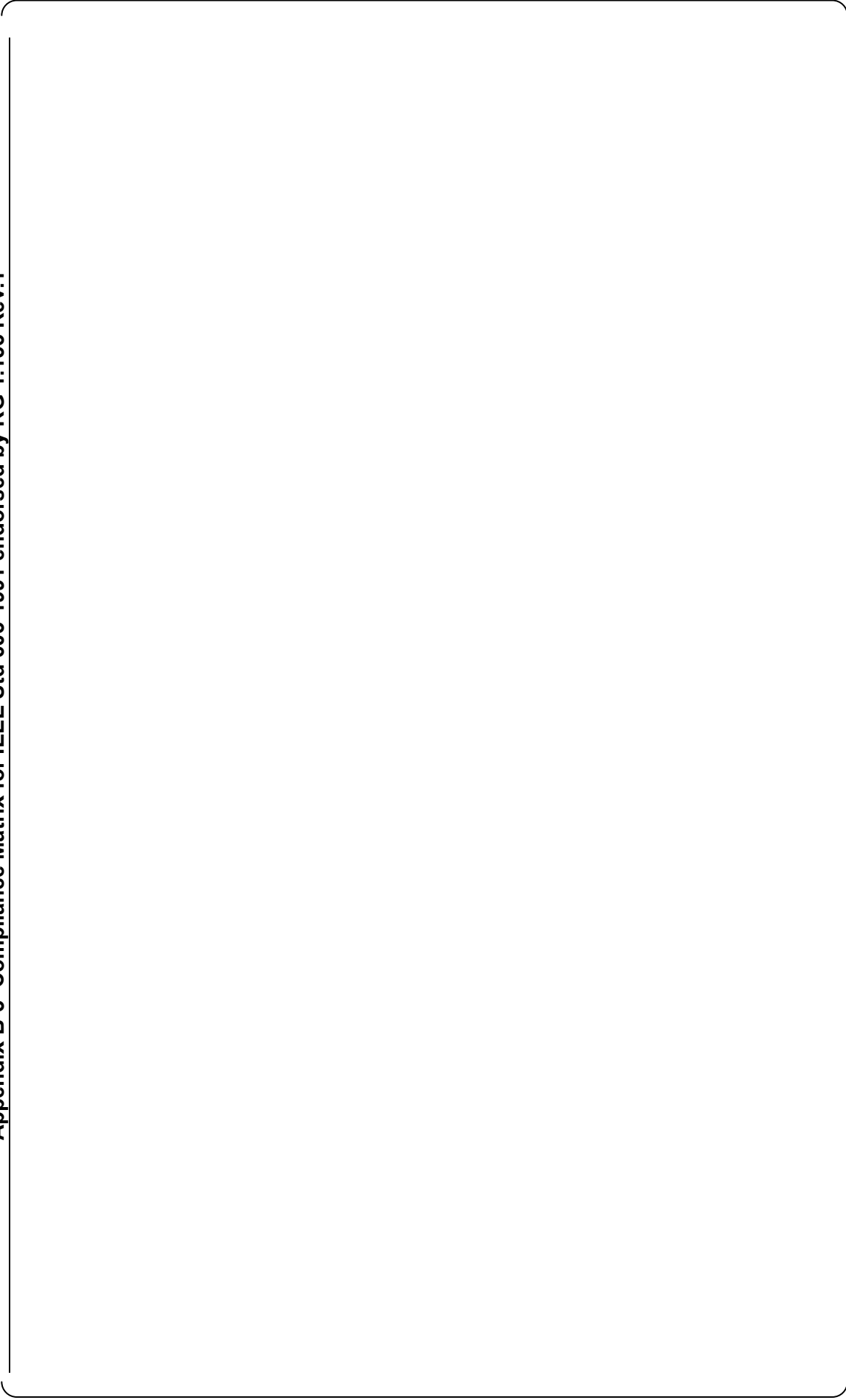
--	--



**Appendix B-3 Compliance Matrix for IEEE Std 603-1991 endorsed by RG 1.153 Rev.1**




**Appendix B-3 Compliance Matrix for IEEE Std 603-1991 endorsed by RG 1.153 Rev.1**



**Appendix B-4 Compliance Matrix for IEEE Std 828-1990 endorsed by RG 1.169**




**Appendix B-4 Compliance Matrix for IEEE Std 828-1990 endorsed by RG 1.169**



**Appendix B-4 Compliance Matrix for IEEE Std 828-1990 endorsed by RG 1.169**



**Appendix B-4 Compliance Matrix for IEEE Std 828-1990 endorsed by RG 1.169**



**Appendix B-4 Compliance Matrix for IEEE Std 828-1990 endorsed by RG 1.169**




**Appendix B-4 Compliance Matrix for IEEE Std 828-1990 endorsed by RG 1.169**






**Appendix B-5 Compliance Matrix for IEEE Std 829-1983 endorsed by RG 1.170**




**Appendix B-5 Compliance Matrix for IEEE Std 829-1983 endorsed by RG 1.170**



**Appendix B-5 Compliance Matrix for IEEE Std 829-1983 endorsed by RG 1.170**



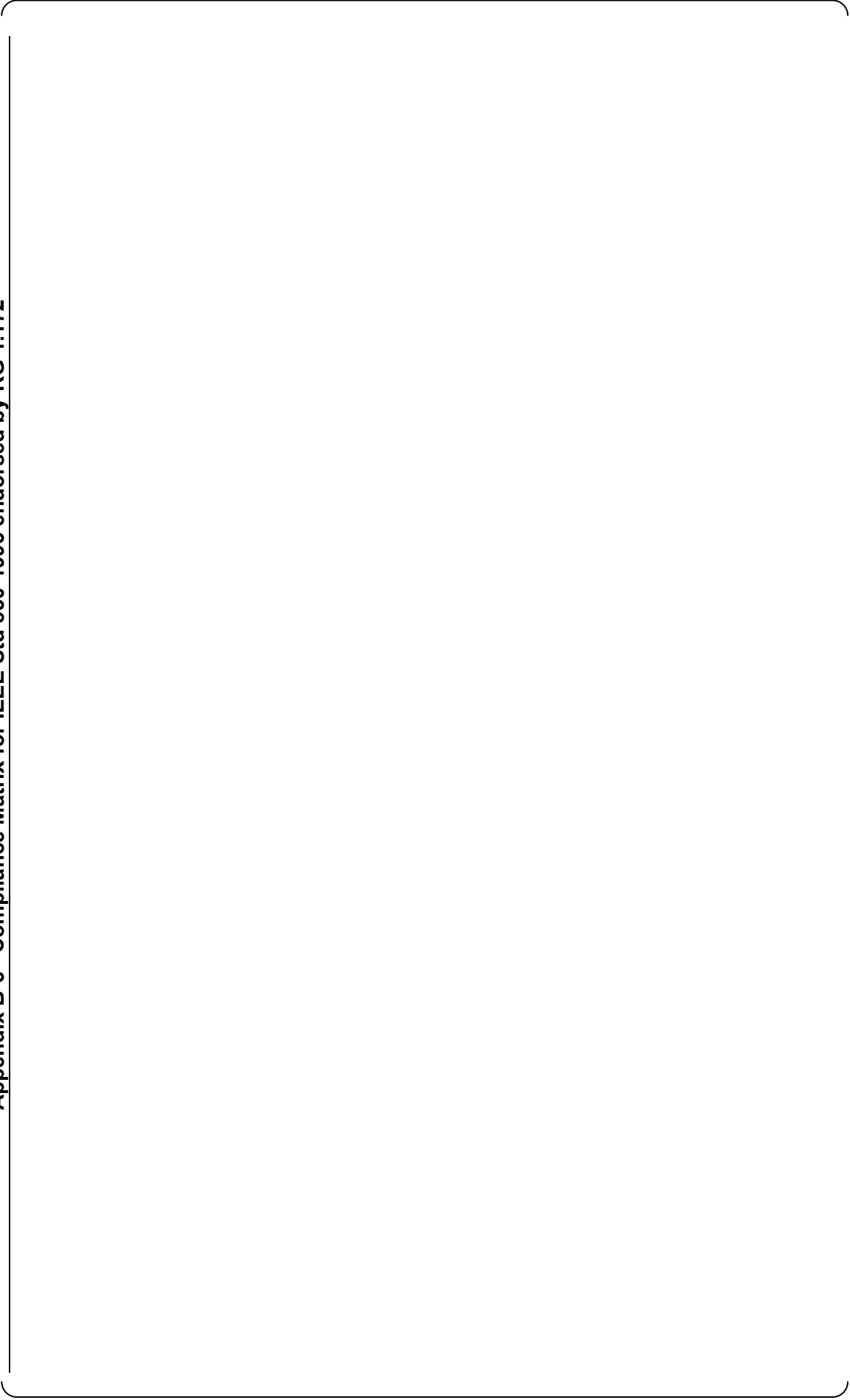
**Appendix B-5 Compliance Matrix for IEEE Std 829-1983 endorsed by RG 1.170**



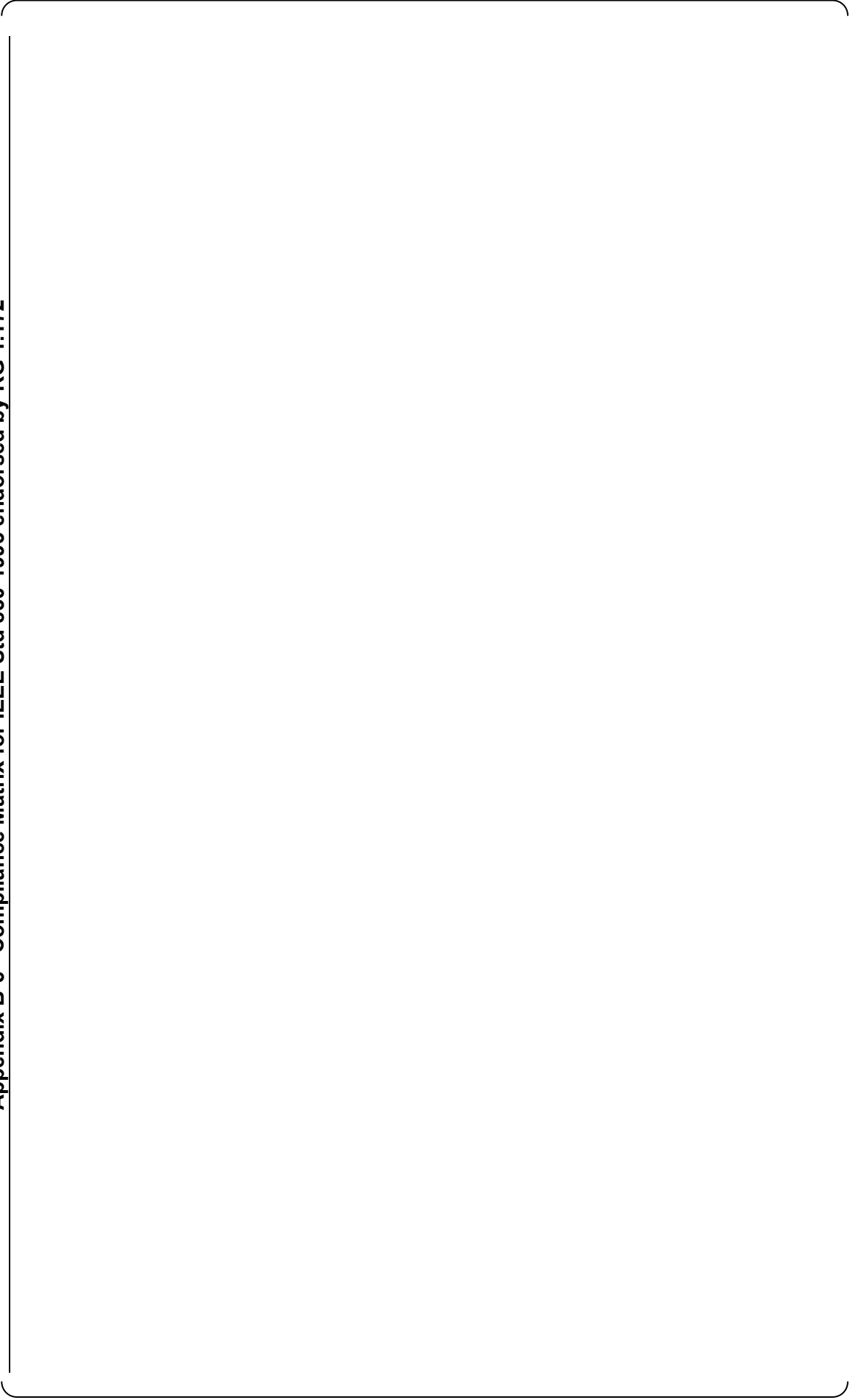
**Appendix B-6 Compliance Matrix for IEEE Std 830-1993 endorsed by RG 1.172**



**Appendix B-6 Compliance Matrix for IEEE Std 830-1993 endorsed by RG 1.172**



**Appendix B-6 Compliance Matrix for IEEE Std 830-1993 endorsed by RG 1.172**

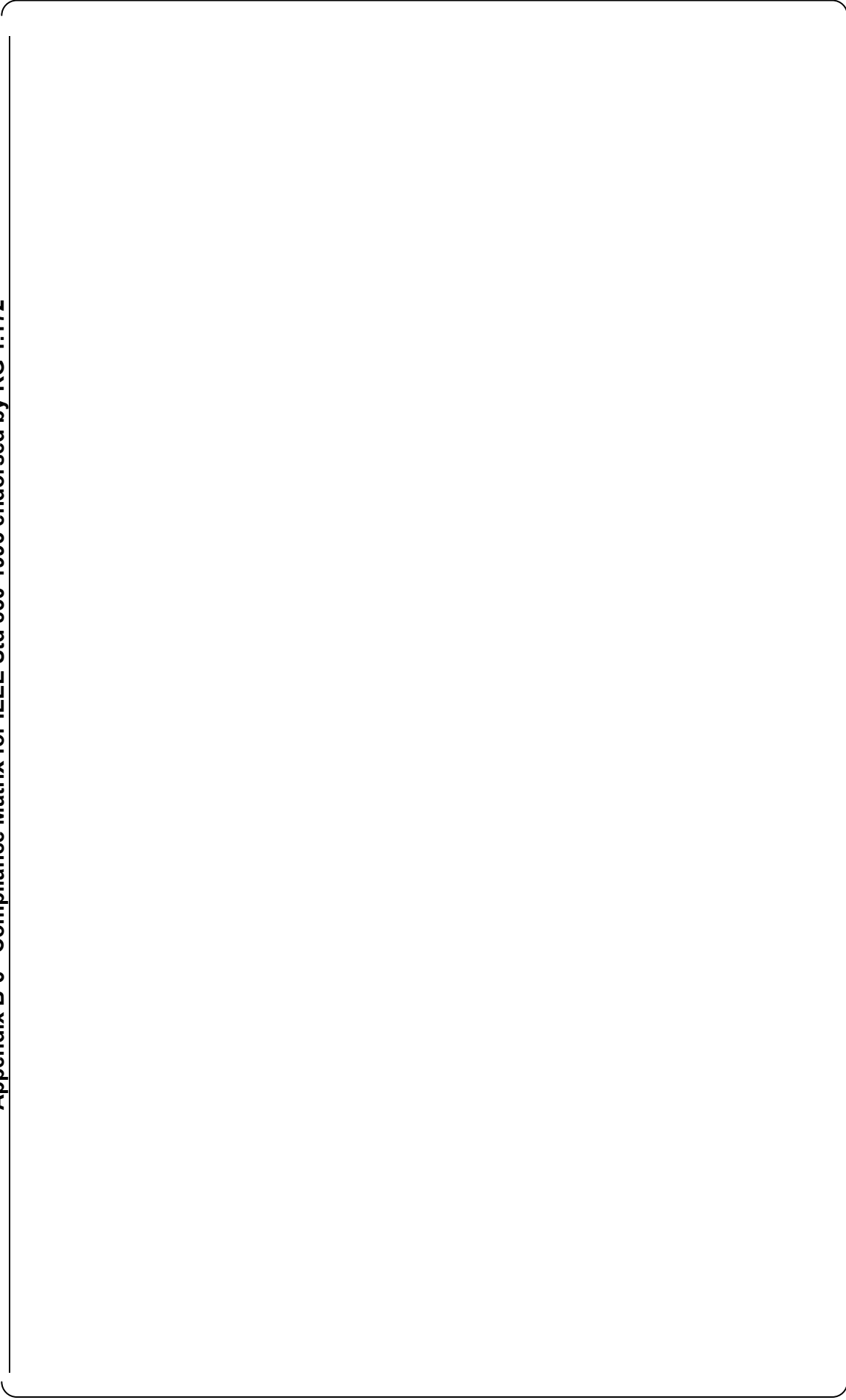


**Appendix B-6 Compliance Matrix for IEEE Std 830-1993 endorsed by RG 1.172**

--



**Appendix B-6 Compliance Matrix for IEEE Std 830-1993 endorsed by RG 1.172**



**Appendix B-7 Compliance Matrix for IEEE Std 1008-1987 endorsed by RG 1.171**

--


**Appendix B-7 Compliance Matrix for IEEE Std 1008-1987 endorsed by RG 1.171**



**Appendix B-8 Compliance Matrix for IEEE Std 1012-1998 endorsed by RG 1.168 Rev.1**




**Appendix B-8 Compliance Matrix for IEEE Std 1012-1998 endorsed by RG 1.168 Rev.1**



**Appendix B-8 Compliance Matrix for IEEE Std 1012-1998 endorsed by RG 1.168 Rev.1**



**Appendix B-8 Compliance Matrix for IEEE Std 1012-1998 endorsed by RG 1.168 Rev.1**




**Appendix B-8 Compliance Matrix for IEEE Std 1012-1998 endorsed by RG 1.168 Rev.1**






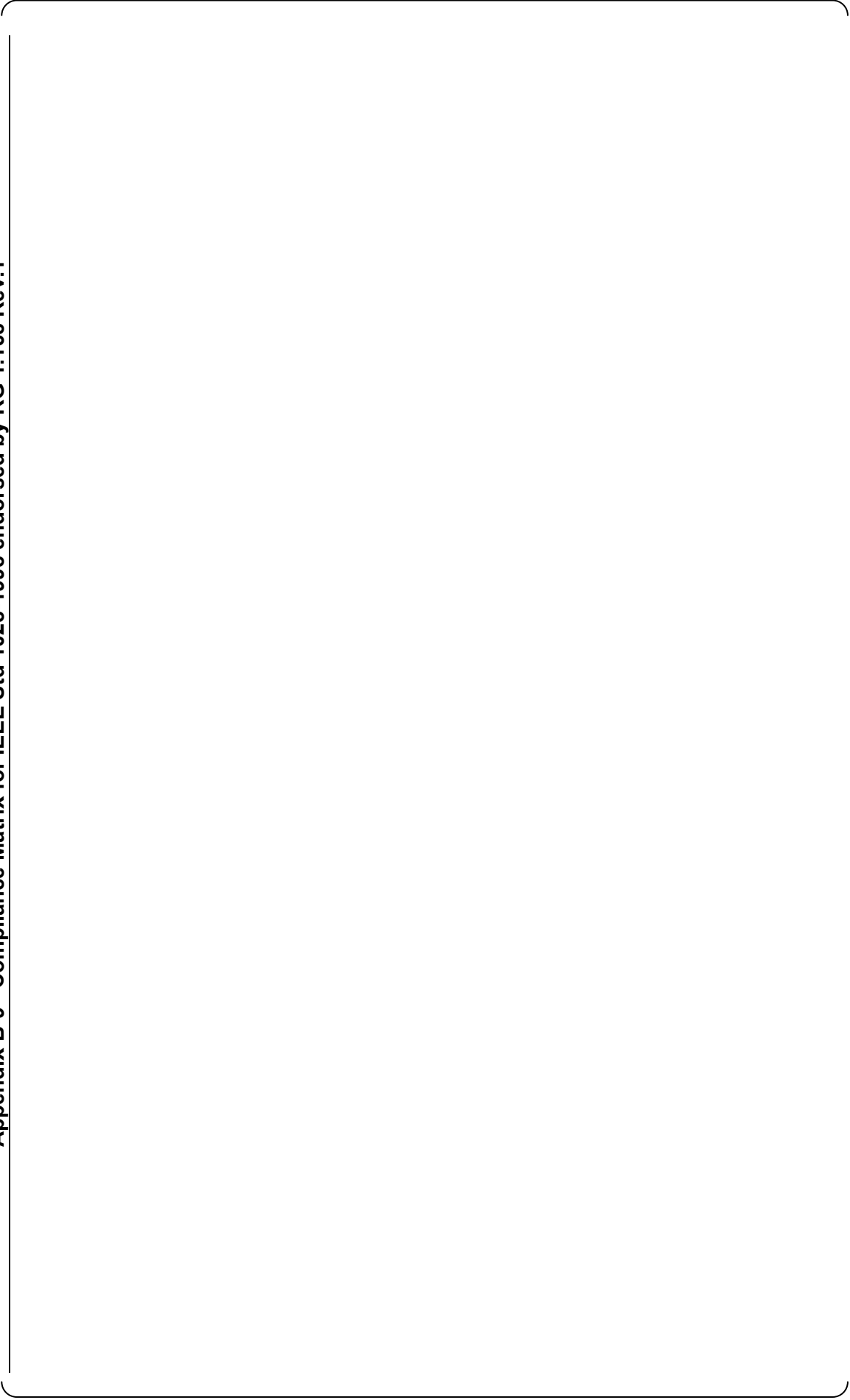
**Appendix B-8 Compliance Matrix for IEEE Std 1012-1998 endorsed by RG 1.168 Rev.1**



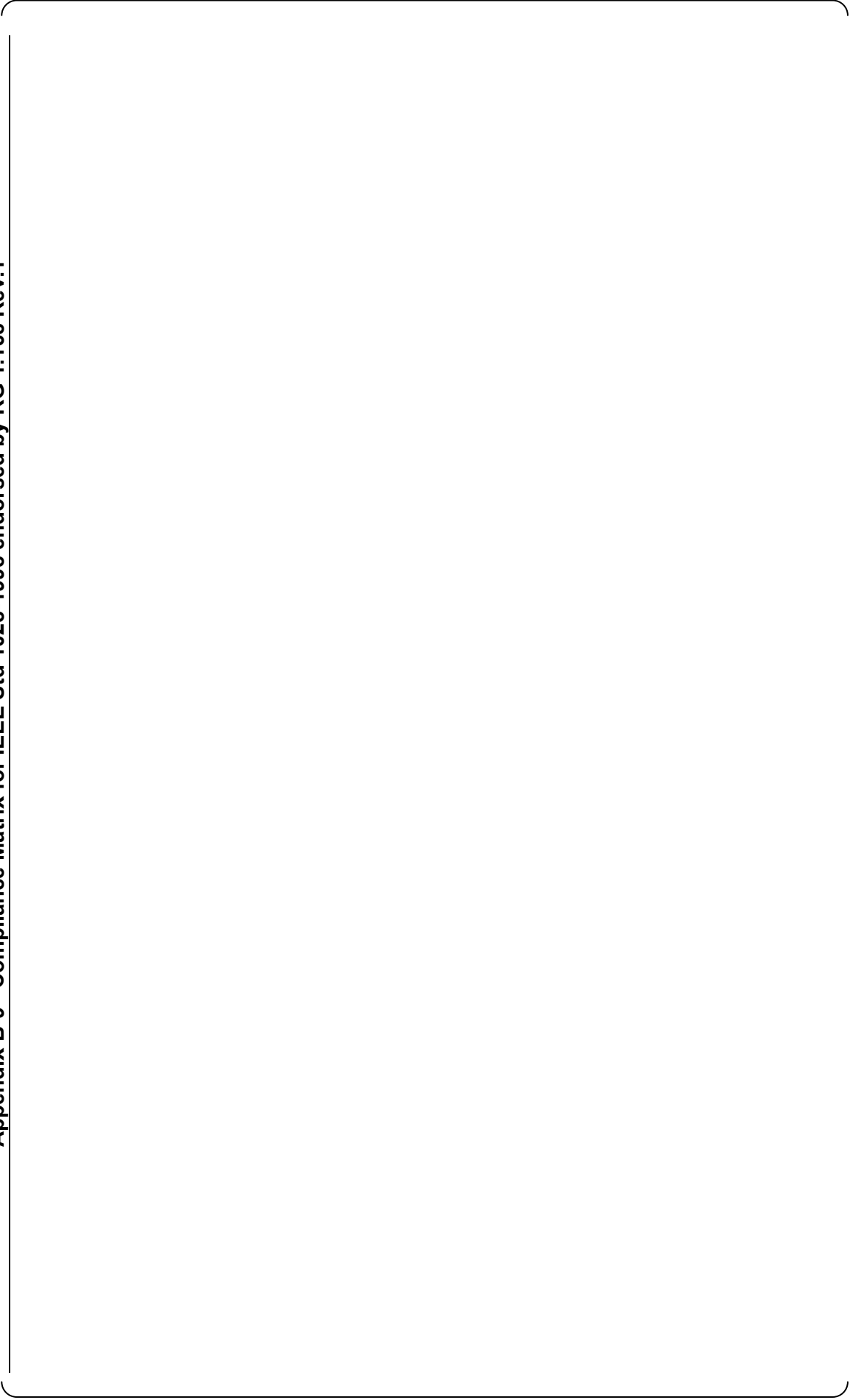
**Appendix B-8 Compliance Matrix for IEEE Std 1012-1998 endorsed by RG 1.168 Rev.1**



**Appendix B-9 Compliance Matrix for IEEE Std 1028-1998 endorsed by RG 1.168 Rev.1**



**Appendix B-9 Compliance Matrix for IEEE Std 1028-1998 endorsed by RG 1.168 Rev.1**



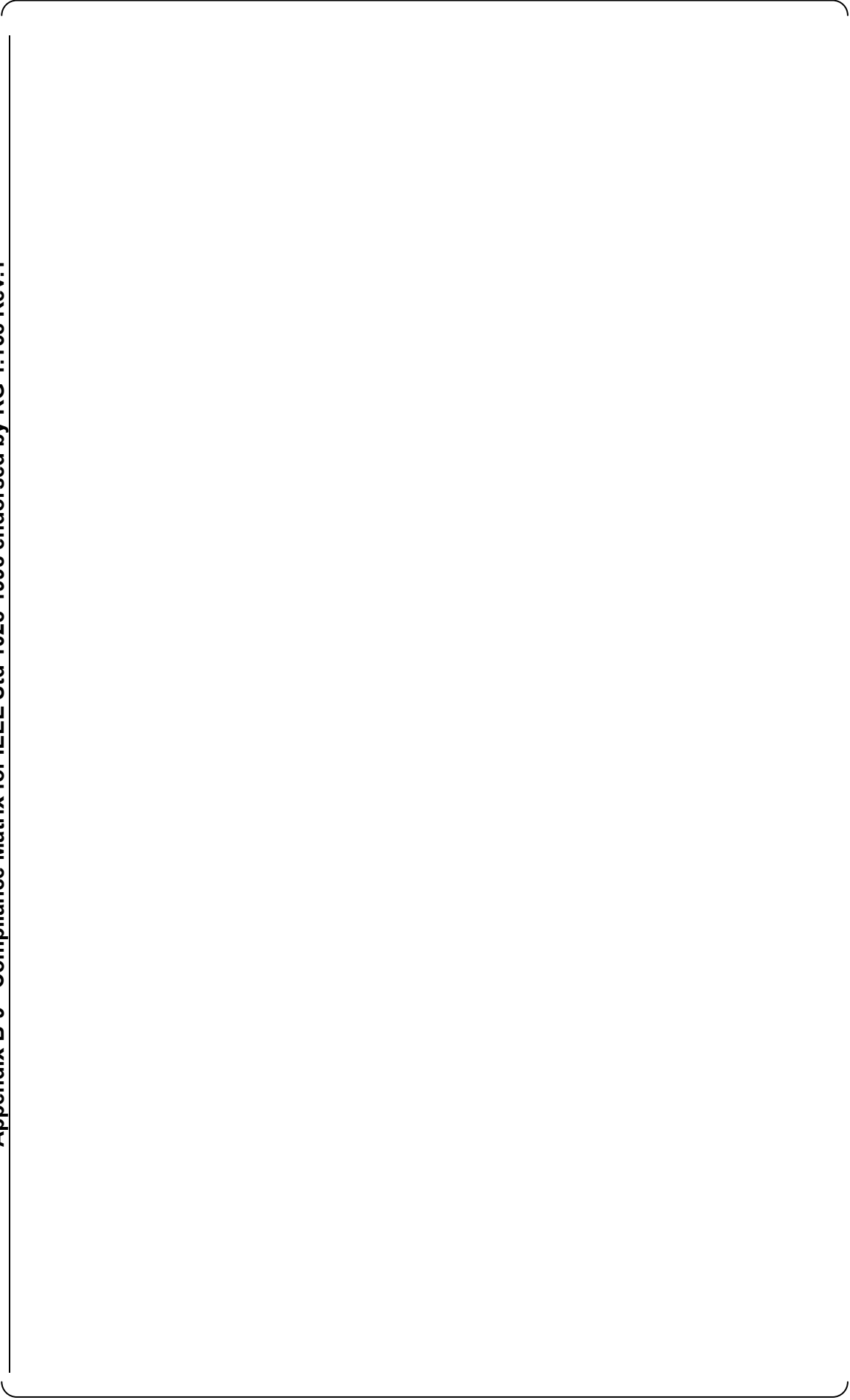
**Appendix B-9 Compliance Matrix for IEEE Std 1028-1998 endorsed by RG 1.168 Rev.1**



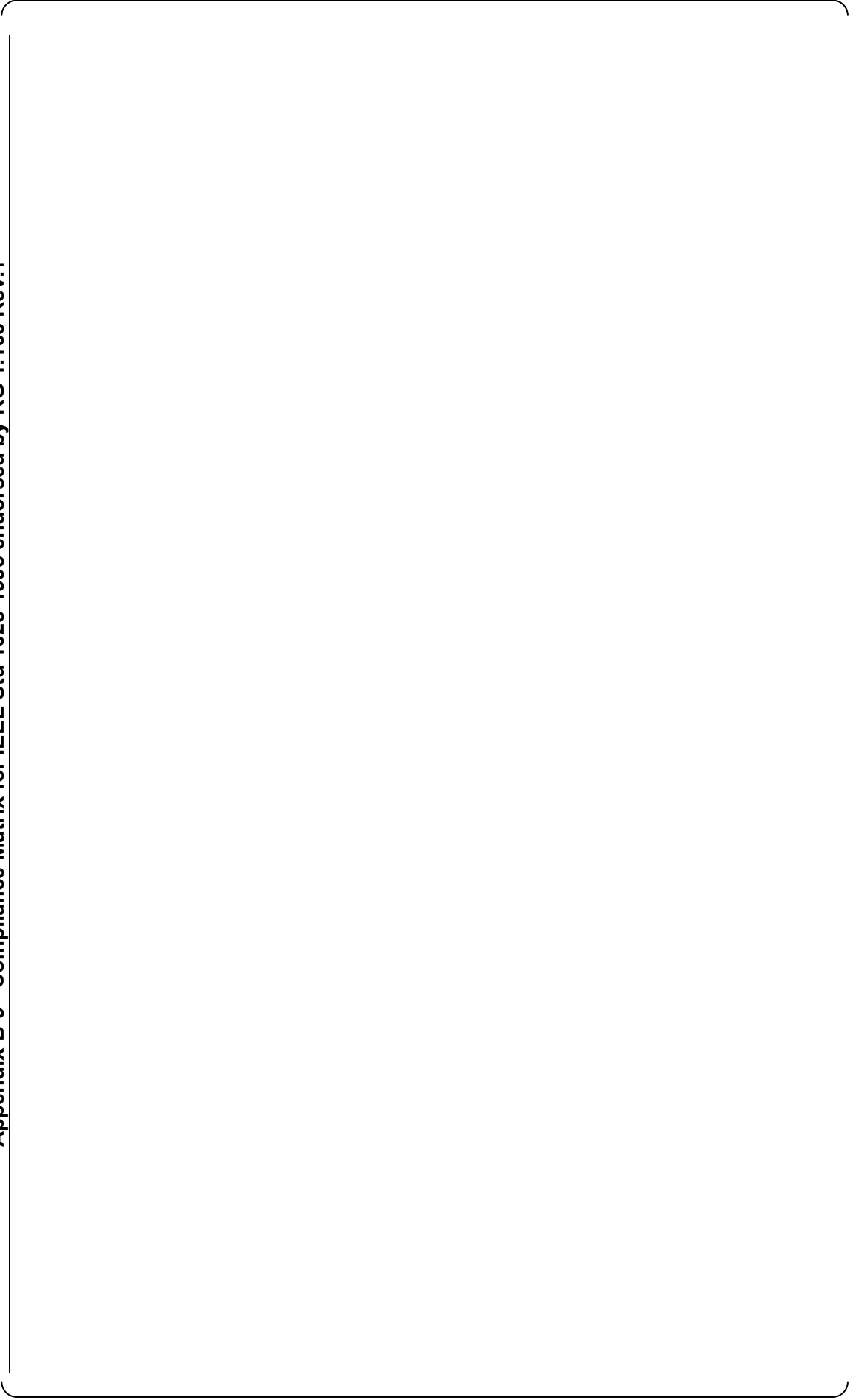
**Appendix B-9 Compliance Matrix for IEEE Std 1028-1998 endorsed by RG 1.168 Rev.1**



**Appendix B-9 Compliance Matrix for IEEE Std 1028-1998 endorsed by RG 1.168 Rev.1**



**Appendix B-9 Compliance Matrix for IEEE Std 1028-1998 endorsed by RG 1.168 Rev.1**





**Appendix B-9 Compliance Matrix for IEEE Std 1028-1998 endorsed by RG 1.168 Rev.1**



**Appendix B-9 Compliance Matrix for IEEE Std 1028-1998 endorsed by RG 1.168 Rev.1**



**Appendix B-10 Compliance Matrix for IEEE Std 1042-1987 endorsed by RG 1.169**



**Appendix B-10 Compliance Matrix for IEEE Std 1042-1987 endorsed by RG 1.169**



**Appendix B-10 Compliance Matrix for IEEE Std 1042-1987 endorsed by RG 1.169**



**Appendix B-10 Compliance Matrix for IEEE Std 1042-1987 endorsed by RG 1.169**



**Appendix B-10 Compliance Matrix for IEEE Std 1042-1987 endorsed by RG 1.169**



**Appendix B-10 Compliance Matrix for IEEE Std 1042-1987 endorsed by RG 1.169**





**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**

--

**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**

--

**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**



**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**

--

**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**

--

**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**

--

**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**

--

**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**

--



**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**

--

**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**

--

**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**

--

**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**

--

**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**

--

**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**

--

**Appendix B-11 Compliance Matrix for IEEE Std 1074-1995 endorsed by RG 1.173**

--

---

## **APPENDIX C RG 1.152 Rev 2 Compliance**

The MELTAC Re-evaluation Program (MRP) report, JEXU-1022-6301, assesses the legacy MELTAC Basic Software, which was developed prior to MELCO's 10CFR50 Appendix B QAP and prior to this SPM, against the guidance in RG 1.152. The MRP confirms (1) that the MELTAC security features are adequate to protect the safety functions of the MELTAC platform, (2) that those features are reflected in actual MELTAC documentation and (3) that those features have been developed with adequate quality assurance. The compliance assessment in this section confirms the processes planned for future MELTAC development (ie. the processes defined in this SPM) will not compromise those legacy MELTAC security features. It also confirms that the processes defined for future MELTAC development will adequately encompass security for all new MELTAC products.

Compliance to RG 1.152 is demonstrated in two sections below. Section C.1 describes compliance for the processes used to develop the platform in each phase of the product life cycle. Section C.2 assesses the effectiveness of these defensive strategies in mitigating potential threats that could lead to unauthorized changes in life cycle output products.

At NRC direction, this compliance description does not include the Operations and Maintenance life cycle phases, since these phases are expected to be excluded from RG 1.152 in the next revision.

### **C.1 MELTAC Platform Security**

MELCO cyber security control internal procedure governs security activities for the MELTAC Platform Basic Software. MELCO personnel involved in platform software activities are trained and qualified to this procedure.

The MELTAC platform design considered security issues from the earliest stages. The key management and design features related to current software and data security are described in the MELTAC Topical Report Sections 6.1.6 (Cyber Security Management), 6.1.6.1 (Software and FPGA Development/Storage Security Measures), 6.1.6.2 (Security Measures In Each Phase of Development Process), 6.1.6.3 (Cyber Security Measures During System Operation), MELCO performed a MELTAC Re-evaluation Program (MRP, JEXU-1022-6301) that included assessments of the MELTAC platform and legacy development activities against the security requirements of RG 1.152 (Positions C2.1 through C2.5), BTP 7-14, and EPRI TR-107330 (Section 4.9.2).

The MRP confirmed that MELTAC, which was developed under a Japanese Quality Assurance Program (QAP), is equivalent to a product developed under a 10CFR50 Appendix B QAP. MELTAC will now be maintained in accordance with this SPM, under MELCO's newly implemented 10CFR50 Appendix B QAP.

The following subsections describe the methods used by MELCO to fulfill the requirements of RG 1.152 Positions C 2.1 through C2.5:



**C.1.1 Concept and Requirements Phases (Positions C2.1 and C2.2)****RG 1.152 Position C 2.1 "Concept Phase" Requirements:**

In the concepts phase, the licensee and developer should identify safety system security capabilities that should be implemented.

The licensee and developer should perform security assessment to identify potential security vulnerabilities in the relevant phases of the system life cycle. The results of the analysis should be used to establish security requirements for the system (hardware and software).

Remote access to the safety system should not be implemented. Computer-based safety systems may transfer data to other systems through one-way communication pathways.

**Analysis:**

Security Related and Proprietary Information - Withheld Under 10CFR2.390

**RG 1.152 Position C2.2 “Requirements Phase” Requirements**

The licensees and developers should define the security functional performance requirements and system configuration; interfaces external to the systems; and the requirements for qualification, human factor engineering, data definitions, documentation for the software and hardware, installation and acceptance, operation and execution, and maintenance.

The security requirements should be part of the overall system requirements. Therefore, the V&V process of the overall system should ensure the correctness, completeness, accuracy, testability, and consistency of the system security requirements.

Requirements specifying the use of pre-developed software and systems (e.g., reuse software and commercial off-the-shelf systems) should address the vulnerability of the safety system (e.g., by using pre-developed software functions that have been tested and are supported by operating experience).

The development process should ensure the system does not contain undocumented code (e.g., back door coding), malicious code (e.g., intrusions, viruses, worms, Trojan horses, or bomb codes), and other unwanted and undocumented functions or applications.

Analysis:

Security Related and Proprietary Information - Withheld Under 10CFR2.390

**C.1.2 Design Phase (Position C2.3)****RG 1.152 Position C2.3 "Design Phase" Requirements**

The safety system security requirements identified in the system requirements specification should be translated into specific design configuration items in the system design description. The safety system security design configuration items should address control over (1) physical and logical access to the system functions, (2) use of safety system services, and (3) data communication with other systems. Design configuration items incorporating pre-developed software into safety system should address security vulnerabilities of the safety system.

Physical and logical access control should be based on the results of cyber-security qualitative risk analysis. Cyber-security risk is the combination of the consequence to the nuclear power plant and the susceptibility of a digital system to internal and external cyber-attack. The results of the analysis may require more complex access control, such as of knowledge (e.g., password), property (e.g., key, smart-card) or personal features (e.g., fingerprints), rather than just a password.

The development should delineate the standards and procedures that will confirm with the applicable security policies to ensure the system design products (hardware and software) do not contain undocumented code (e.g., back door coding), malicious code (e.g., intrusions, viruses, worms, Trojan horses, or bomb codes), and other unwanted and undocumented functions or applications.

Analysis:

Security Related and Proprietary Information - Withheld Under 10CFR2.390

**C.1.3 Implementation Phase (Position C2.4)****RG 1.152 Position C2.4 “Implementation Phase” Requirements(1/3)**

The developer should ensure that the security design configuration item transformations from the system design specification are correct, accurate, and complete.

The developer should implement security procedures and standards to minimize and mitigate tampering with the development system.

The developer’s standards and procedures should include testing with scanning as appropriate, to address undocumented code or malicious functions that might (1) allow unauthorized access or use of the system or (2) cause systems to behave beyond the system requirements.

The developer should account for hidden functions and vulnerable features embedded in the code, and their purpose and impact on the safety system. If possible, these functions should be disabled, removed, or (as a minimum) addressed (e.g., as part of the failure modes and affects analysis of the application code) to prevent any unauthorized access.

Scanning is dependent on the platform and code being used, and may not be available for the specified code and compiler. This may be a difficult task with little assurance that the results will be comprehensive and successful in uncovering hidden problems given the size and complexity of most modern computer systems.

Pure application code scanning may be partially successful, but many operating systems, machine code, and callable library function aspects of the system may not be able to be successfully scanned and are just as likely to be where avenues for exploitation exist.

COTS (commercial off-the-shelf) systems are likely to be proprietary and generally unavailable for review. It is likely that there is no reliable method to determine security vulnerabilities for Operating system (for example, Microsoft and other operating system suppliers do not provide access to the source code for operating systems and callable code libraries).

In such cases, unless such systems are modified by the application developer, the security effort should be limited to ensuring that the features within the system do not compromise security requirements of the system, and the security functions should not be compromised by other system functions.

RG 1.152 Position C2.4 "Implementation Phase" Requirements(2/3)
---

Analysis:
-----------

Security Related and Proprietary Information - Withheld Under 10CFR2.390

**RG 1.152 Position C2.4 "Implementation Phase" Requirements(3/3)**

Security Related and Proprietary Information - Withheld Under 10CFR2.390

**C.1.4 Test Phase (Position C2.5)****RG 1.152 Position C2.5 "Test Phase" Requirements(1/3)**

The security requirements and configuration items are part of validation of the overall system requirements and design configuration items. Therefore, security design configuration items are just one element of the overall system validation. Each system security features should be validated to verify that the implemented system does not increase the risk of security vulnerabilities and does not reduce the reliability of safety functions.

The developer should configure and enable the design security features correctly. The developer should also test the system hardware architecture, external communication devices, and configurations for unauthorized pathways and system integrity. Attention should be focused on built-in OEM features.

Analysis:

Security Related and Proprietary Information - Withheld Under 10CFR2.390

RG 1.152 Position C2.5 "Test Phase" Requirements(2/3)

Security Related and Proprietary Information - Withheld Under 10CFR2.390

RG 1.152 Position C2.5 "Test Phase" Requirements(3/3)

Security Related and Proprietary Information - Withheld Under 10CFR2.390



**C.2 Assessment of Defensive Strategies to Prevent Unauthorized Changes**

The possible user or developer activities with regard to the development phases described in this SPM are listed below:

{ [

Security Related and Proprietary Information - Withheld Under 10CFR2.390

] }

**C.2.1 Defense Against Unauthorized Changes in Requirements Specifications**

{ [

Security Related and Proprietary Information - Withheld Under 10CFR2.390

] }

**C.2.2 Defense Against Unauthorized Changes in Design Descriptions**

{ [

Security Related and Proprietary Information - Withheld Under 10CFR2.390

] }

**C.2.3 Defense Against Unauthorized Changes in Source Code or the Development Environment**

{ [

Security Related and Proprietary Information - Withheld Under 10CFR2.390

] }

**C.2.4 Defense Against Changes Masked by Misrepresentation in Test Reports**

{ [

Security Related and Proprietary Information - Withheld Under 10CFR2.390

] }

**C.2.5 Defense Against Unauthorized Changes to Approved Output Products**

{ [

Security Related and Proprietary Information - Withheld Under 10CFR2.390

] }

Table C.2-1 - Security Measures of the Software Development/Storage Environment

Object	Role	Security Measures
--------	------	-------------------

Security Related and Proprietary Information - Withheld Under 10CFR2.390

{[

Security Related and Proprietary Information - Withheld Under 10CFR2.390

]}

**C.3 Conclusion**

The scope of the MELTAC software and data security responsibilities include the Concept Phase through the Test Phase, as described in Regulatory Guide 1.152. The combination of the information in the MELTAC Topical Report (MUAP-07005) on system design features, and the specific information provided in this document fully address the software and data security issues associated with the MELTAC platform.