SOUTHERN CALIFORNIA RAPID TRANSIT DISTRICT

METRO RAIL PROJECT

HARDWARE AND SOFTWARE REQUIREMENTS FOR MICROPROCESSORS

May 1986

Prepared by

Booz·Allen & Hamilton Inc.
Transportation Consulting Division
523 West Sixth Street, Suite 502
Los Angeles, California 90014

TABLE OF CONTENTS

28857333

I N D E X O F E X H I B I T S

1.0 BACKGROUND

1.0 BACKGROUND

1.1 INTRODUCTION

The Southern California Rapid Transit District (SCRTD) has formed a task force to develop hardware and software specifications for the Metro Rail System. The specifications will be based upon lessons learned from previous microcomputer applications in rail transit equipment, from accepted procurement methods for complex digital hardware, from accepted state-of-the-art methods in software technology, and from software industry standards adapted to suit rail transit requirements.

The American Public Transit Association (APTA) has formed a Microprocessor Liaison Board to address the issue of microprocessor applications in rail transit. Unfortunately, the board's findings will not be sufficiently advanced for consideration in drafting the Metro Rail procurement specifications which are in the final stages of preparation. However, the SCRTD task force's recommended software specifications will be distributed to APTA's Microprocessor Liaison Board members for review and comment.

This document consists of a brief summary of rail transit software issues, a suggested methodology for managing the procurement of microprocessor software for Metro Rail subsystems, and recommended software specification language for inclusion in equipment procurement specifications.

1.2 RECENT TRANSIT HISTORY, PROBLEMS, AND TRENDS

Economics and emerging industrial practice are driving the application of computer-based products to rail transit, just as they are driving applications in all technical and equipment markets. For rail transit, the application of digital techniques involves unique requirements primarily related to safety and the system environment. Full realization in the transit industry of the benefits that computer-based products could yield must follow resolution of several problems. Key issues are:

- Some computer-based equipment designed for transit has not achieved acceptable levels of functional performance, reliability, and maintainability. Train control, vehicle

subsystems, and communications equipment must meet very high requirements for safety and reliability.

- The equipment must be maintainable by rail transit maintenance staffs with no degradation in safety.

- There is a lack of agreement in the transit industry on standards, or draft standards, for computer-based product development, acceptance, certification, documentation, and interface. Software standards from other industries may provide guidance but cannot entirely fill transit needs.

Transit suppliers are unilaterally deciding how and where microprocessors are applied to rail transit products. Major developments are occurring in train control, propulsion control, and brake control. Communications equipment suppliers are proposing to use microprocessors in transit system central control systems and SCADA (supervisory control and data acquisition) systems but have had mixed success with the performance and reliability of the installed systems. The microprocessor developments are not generally coupled with prudent or consistent control of software documentation, adequate software quality assurance, adequate software testing, or suitable software maintenance criteria. While transit suppliers are confident of their ability to develop microcomputer-based products, transit operators have expressed serious concerns about the risks of development, especially when conducted without the benefit of suitable methodologies for reliability and safety assessment.

1.3 SPECIAL CONSIDERATIONS OF TRANSIT SAFETY-CRITICAL EQUIPMENT

The introduction of any new technology into a safety-critical area will be impeded by the lack of experience with the new devices and by the absence of appropriate application and quality standards. Microprocessor technology in rail transit applications faces difficulties of this type related to:

- Lack of agreement on design configurations for safety-related equipment

- Lack of agreement on guidelines or standards for developing and testing software

- The inability to precisely define failure modes for microprocessor components

- The translation of functional, performance, and safety requirements into software programs that are reliable and safe

- Problems in defining and providing adequate preventive and corrective maintenance techniques.

In train control, vital relays and interlocking circuits are accepted because their failure modes are well defined and because there is an established methodology for translating these failure modes into "safe" states. Digital devices, and microprocessors in particular, are more complex and their failure modes are difficult to bound. Further, there are established standards and review procedures for the design of relay interlockings, whereas digital processors are based on software to which these standards are not directly applicable. Rail transit suppliers are taking substantially different approaches in configuring the equipment to prevent failures from resulting in an unsafe state. Some suppliers advocate a single processor running checked software, while others consider that at least two processors are required for adequate safety. Maintenance requirements for relay logic are substantially different from those of solid-state logic. The mechanical components of a relay are subject to wear. Solid-state logic is significantly more reliable than the equivalent relay logic. In order to counteract wear and other possible deterioration in vital relays, a well-defined set of maintenance and recalibration guidelines has evolved. There is general agreement that preventive maintenance is not required for solid-state logic. Furthermore, preventive maintenance can introduce more failures in solid-state logic than would randomly occur if the equipment were left undisturbed. Nevertheless, retesting or recalibration of solid-state logic may be essential to prevent degradation of equipment safety since not all failures can be made self-annunciating.

While the reliability of solid-state logic is higher, the accurate diagnosis of faulty relay logic is more straightforward than that of solid-state logic, particularly where complex combinations of digital logic, or microprocessors, are used. There is concern that diagnosis of faulty digital circuits may be beyond the capabilities and resources of most rail transit maintenance departments despite hiring and training efforts.

Recent deliveries of microprocessor-based equipment to rail transit operators is revealing that extensive problems exist in providing adequate maintenance of the equipment.

1.4 FAULT-TOLERANT DESIGN

The traditional approach in rail transit control systems has emphasized "fail-safe" attributes, and these still represent the minimum requirements. However, digital techniques also make it practical to implement "fail-operational" (fault-tolerant) configurations. These techniques not only make it possible to operate after a first fault but also permit repairs to be performed on a nonemergency basis. Self-diagnostics by the microprocessor and provision of powerful automatic test equipment can reduce the maintenance problems and manpower requirements.

Intermittent failures, or lock-up of the processors, however, pose a particular problem. It has been stated that over 90 percent of faults in contemporary digital circuits manifest themselves only temporarily or intermittently.* Such faults are not likely to be detected by periodic diagnostic checks, which are usually performed by connecting the equipment to special test equipment. The memory capabilities of data loggers and processors can be used to record crucial historical data prior to an equipment failure. While this provides some enhancement to troubleshooting, a fault-tolerant design approach may be required to maintain operational capability of controls and data transfer. Fortunately, fault-tolerance techniques can be extended to provide continuous monitoring of failure occurrence and frequently also point to the malfunctioning component. However, realization of this capability depends heavily on the hardware structures and software techniques selected by the suppliers.

1.5 QUALITY OBJECTIVES FOR MICROPROCESSOR HARDWARE AND SOFTWARE IN METRO RAIL EQUIPMENT

Microprocessor controls in Metro Rail equipment shall utilize embedded software. The software shall be located in permanent memory on printed circuit boards and shall execute automatically; loading from disk or tape shall not be required. Embedded software is strongly coupled to the associated hardware of the particular subsystem and has important, potentially complex functional interfaces with other components of the subsystem and often with associated subsystems.

---

* D. C. Bossen and M. Y. Hsiao, "ED/FI: A Technique For Improving Computer System RAS," Digest of Fault-Tolerant Computer Symposium, pp. 2-6, June 1981.

The quality objectives of the hardware specification include:

- Designing and manufacturing for the rail transit environment that take maintenance as well as shock and vibration into consideration
- Modular grouping of circuit functions to minimize interconnections
- Fully disclosing internal and external interfaces, including waveforms, rise and fall times, impedances, tolerances, etc.
- Providing built-in diagnostics with displays and communication interfaces
- Ensuring availability of solid-state devices and other components over the expected life of the equipment
- Packaging for durability and ease of maintenance
- Adequately protecting logic circuitry from electromagnetic interference.

In addition, the general quality attributes of electronic design and packaging should be specified, such as PC board keying, component identification, protection from contaminants, and cooling.

The quality objectives of the software specifications include:

- Developing structured top-down designs
- Providing management and technical staff with visibility of the software development process through a comprehensive evaluation of each phase of the software development
- Detecting and correcting design deficiencies and software errors at the earliest practicable stage of the software development cycle
- Enabling the SCRTD to assess and control software changes

- Enabling the SCRTD to effectively maintain the microprocessor hardware and software after acceptance, including the option of full in-house repair capability.

During the design review process, the technical monitoring objectives include:

- Ensuring provision of software design configurations that cause the equipment to revert to a predetermined state if the software fails to execute correctly

- Eliminating software errors that can cause reliability or safety problems

- Evaluating the software for adequacy of modular structure, timing, and testability, and for general performance commonly associated with quality code

- Evaluating software test procedures for normal and abnormal inputs or conditions.

In addition, the quality of software documentation provided by the suppliers must be sufficient to permit independent safety review and analysis.

2.0 GENERAL REQUIREMENTS FOR METRO RAIL HARDWARE AND SOFTWARE SPECIFICATIONS

2.0 GENERAL REQUIREMENTS FOR METRO RAIL HARDWARE AND SOFTWARE SPECIFICATIONS

Chapter 2.0 outlines general requirements for microprocessor specifications with emphasis on the software. The material is intended as a foundation for the development of microprocessor specification language for the passenger vehicle, automatic train control equipment, communication equipment, and fare collection equipment. The requirements also apply to the processors of minicomputers and mainframe computers.

2.1 BASIS FOR DEVELOPING THE SOFTWARE CRITERIA

There are several publications on software specification standards, some of which are listed in chapter 7.0 of this report, "Applicable Documents." None, however, has direct applicability to rail transit microprocessor applications. The standards of the Institute of Electrical and Electronic Engineers (IEEE) have been selected as the basis for developing the Metro Rail specifications. The IEEE Standards, listed in chapter 7.0, were judged the most applicable of the documents identified in a limited literature search. The IEEE Standard Glossary of Software Engineering Terminology (IEEE Std-729-1983) was used to develop definitions for the SCRTD procurement specifications.

IEEE Std-729 also describes an example software life cycle, or software development cycle, consisting of eight distinct but overlapping phases. They are:

- Concept Exploration. Concept exploration is not defined in IEEE Std-729.

- Requirements. 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 document. The set of all requirements forms the basis for subsequent development of the system or system component.

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

- Implementation. A realization of an abstraction in more concrete terms; in particular, in terms of hardware, software, or both.

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

- Installation and Checkout. The period of time in the software life cycle during which a software product is integrated into its operational environment and tested in this environment to ensure that it performs as required.

- Operation and Maintenance. The period of 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.

- Retirement. The period of time in the software life cycle during which support for a software product is terminated.

The eight phases of IEEE Std-729 were reviewed for inclusion in, and reinforcement of, the existing Metro Rail procurement specifications.

The Metro Rail procurement process specifies three sequential design reviews: the Conceptual Design Review (CDR), the Preliminary Design Review (PDR), and the Final Design Review (FDR). During these design reviews the hardware and software designs must be fully disclosed, reviewed, and approved. Exhibit 2-1, Summary of Activities, shows the basic relationship between the IEEE STD-729 example software life cycle, the Metro Rail design reviews, and the anticipated software development activities of subsystem suppliers.

2.2 HARDWARE AND SOFTWARE DEVELOPMENT LIFE CYCLE

The following nine sections describe the development cycle that is representative of Metro Rail needs. Each step in the development process will be an escalating progression in the level of detail with a corresponding escalation in resources expended by Metro Rail staff and

EXHIBIT 2-1
Summary of Activities

IEEE STD-729 PHASE	METRO RAIL PROJECT ACTIVITIES	TYPICAL SUBSYSTEM SUPPLIER ACTIVITIES
Not Applicable	Preliminary and final design.	Industry review of procurement specification.
Concept Exploration	Review of technical proposals.	Proposal to use microprocessors. Failure effects classification, reliability and safety.
Requirements	Conceptual design review.	Functional requirements analysis. Hardware requirements specification. Software requirements specification. Software verification and validation plan.
Design	Preliminary Design Review, Final Design Review, approval of baseline.	Preliminary design: • Hardware block diagrams • Software design description. Final Design: • Pseudo code • Flow charts • Circuit boards and mechanical assemblies.
Implementation	Begin configuration monitoring of baseline.	Coding Syntax debugging.
Test	Engineering test review and monitoring.	Incremental testing: • Module testing • Linking • Module interface testing • Hardware/software integration.
Installation and Checkout	Test review and witnessing.	Further incremental testing: • Factory testing • Post-installation testing • Qualification testing • Acceptance testing.
Operation and Maintenance	Revenue service operations. Warranty management. Post warranty maintenance.	Warranty support.
Retirement	Decision to upgrade or replace subsystem.	Guaranteed provision of replacement parts for life of subsystem equipment.

the suppliers. In order to manage the process, incremental reviews must be conducted and approval given in order to minimize incorrect interpretation and realization of the Metro Rail requirements.

2.2.1 Conceptual Design

The goal of the conceptual design phase is to identify all needs, analyze the constraints, and map out a solution that fulfills the needs.

In this phase, the suppliers will further explain their intended use of microprocessors to realize logic control functions. Safety-critical functions and safety design requirements shall be identified. Requirements for fault tolerance or operational redundancy shall be determined. Reliability goals will be established. Economic and technical feasibility evaluations of existing products against Metro Rail requirements must be completed. In a two-step procurement process,the requirements of the conceptual design shall be provided in the technical proposals, as required by the bid document.

At the conclusion of this step, the Conceptual Design Review (CDR) will take place. During the CDR any problems of practicability, limited transit application experience, and unresolved safety issues must be identified. The suppliers should provide a master schedule for the software development tasks.

2.2.2 Requirements Analysis

Requirements analysis is the process of gaining a detailed understanding of the procurement specification, evaluating the functional requirements, and developing hardware and software requirements specifications. The requirements specifications will summarize the proposed approach by:

- Delineating the microprocessor tasks
- Partitioning the tasks to be performed by hardware and software
- Identifying the nature of inputs and outputs of the microprocessor
- Summarizing the logic calculations to be performed by the software

- Identifying the maintenance concept, special test equipment, and training requirements.

Thus, during the requirements analysis, the initial partitioning of hardware and software will occur. The preliminary test plan will be developed during this phase.

The output of the requirements analysis will be a hardware requirements specification and a software requirements specification that detail how each requirement of the procurement specification is to be fulfilled. The software verification and validation plan will be completed at this time.

The SCRTD should review the requirements specifications and the preliminary test plan.

2.2.3 Preliminary Design

During the preliminary design phase, the microprocessor type and the programming language shall be identified and the test plan finalized.

The preliminary design of the hardware identifies the hardware as a set of assemblies. The hardware requirements specifications are produced in this step and will include such information as:

- General equipment arrangement and packaging and weight constraints
- Description of system operation
- Function block diagrams
- Interface definitions
- Input, output, and transfer function criteria
- Allocation of hardware resources and estimation of spare capacity requirements
- Structure of hardware functional elements showing logic to be performed in hardware, such as frequency-to-digital conversion, digital-to-analog conversion, and other signal conditioning

- Safety criteria
- Performance criteria.

The preliminary design of software identifies the software as a set of modules, or programs, and their interrelationships in the software architecture. Each module will consist of a limited set of inputs, a limited set of outputs, and a processing function. The Software Module Requirements Specifications, or equivalent documents, and system-level software flow charts are produced in this step. The Software Module Requirements Specifications will include such information as:

- General operating procedures
- A concise narrative, in English, of each module
- Data accuracy and consistency requirements
- Logic flow diagrams
- Data flow diagrams
- Memory budget size
- Interface definition
- Algorithm identification
- Timing requirements and constraints
- Safety criteria
- Performance criteria.

The PDR will be complete after review and approval of the Software Module Requirements Specifications, hardware block diagrams, and system-level software flow charts.

2.2.4 Detailed Design

The detailed software design phase expands the detail of the Software Module Requirements Specifications. The design documentation must be in sufficient

detail for a programmer to code and produce the software system. It should contain:

- Flow charts or structure charts that give an overview of processor software
- Completed algorithms expressed in program design language or pseudo code
- Input data definitions
- Output data definitions
- Interrupt structure definition
- Program parameters
- Diagnostic routines for processor self-test and subsystem self-test
- Error handling routines
- Data dictionary.

The output of the detailed design phase is a complete and configured set of program design language or pseudo code, flow charts or structure charts, and a data dictionary, which together with related support documentation form the software baseline. A corresponding set of documentation, which includes printed circuit board schematics, wiring diagrams, and mechanical assembly drawings, will form the hardware baseline.

At the conclusion of the detailed design phase, the final design review takes place. The basic purpose of the FDR is to determine that the design completely satisfies the hardware and software requirements specifications and the SCRTD Metro Rail procurement specification. If the design is satisfactory, approval will be given for the hardware and software baseline.

2.2.5 Software Implementation

After the software detailed design is completed and approved in the final design review, the supplier will proceed with the implementation phase. During implementation the source code is developed. The coding will consist of either translating the program design language or pseudo code into a higher level language such as PASCAL, into a higher level but hardware-specific language such as PL/M-86, or into a

hardware-specific assembly language such as ASM86. In this example, PL/M-86 and ASM86 are specific languages for the Intel 8086 microprocessor; similar languages exist for other microprocessors. The code will be written for each software module and checked for correct syntax by an automated software development system.

The output of the implementation phase will be a complete set of programs free of syntax errors that have been, or can be, translated automatically into machine-executable form (machine code).

During the implementation phase, Metro Rail should, through a series of reviews, monitor progress of the coding and syntax checking. Because the design tasks are extremely detailed, in-depth review and monitoring of this phase will probably be limited to pre-selected portions of the software and to configuration control. At this stage, desk checking and structured walk-throughs shall be conducted to evaluate the output of the implementation phase against the requirements established in the previous phases.

2.2.6 Test

An adequate test phase is crucial to the generation of reliable (error-free) software. Independent verification tests will be required for safety-critical software.

During the test phase and installation and checkout phase, software development tools will be used, particularly the in-circuit emulator of the software development system. It is essential that each independent path of the software be fully tested for hidden software errors. Test cases must be identified and designed to measure the quality of the code, particularly under abnormal circumstances such as those caused by hardware failures in the microprocessor or in the subsystem equipment.

The test phase will consist of module testing and linking, where each module is executed in the target microprocessor hardware or in the software development system. The module testing will be an incremental checkout using dummy, or simulated, input data until all modules are running as specified in the target hardware or in the software development system. When a development system is used, the effects of variations in timing must be accounted for.

Checking all combinations of independent paths is usually prohibitively time-consuming and costly, except for the simplest programs. A structured modular design, however, may be adequately tested by checking each independent path through the software and, if necessary, a manageable number of combinations of independent paths. During testing, the adequacy of checking must be demonstrated.

The output of the test phase will be an integrated set of software modules documented by up-to-date and configured source code and program design language or pseudo code.

During the test phase, Metro Rail will monitor progress and review results. When problems are identified during the test phase, the software documentation will need revision. Metro Rail review and approval will be required for all revisions that occur after the final design review since the baseline design may be altered. Metro Rail will also periodically review and verify testing on randomly chosen modules in order to judge the quality of the software design, documentation, and testing.

2.2.7 Installation and Checkout

The installation and checkout phase is also crucial to generating reliable (error-free) software. The installation and checkout phase begins with incremental integration of the software into the target hardware and does not end until the subsystem is in pre-revenue service and accepted by the SCRTD. As with each software development phase, the test phase and installation and checkout phase will overlap. A typical sequence of events in the installation and checkout phase will consist of:

- Incremental module testing until the entire software is executing correctly in the target hardware, if this was not accomplished during the test phase.

- Subsystem testing using a simulator to represent the external environment of the microprocessor hardware, software, and input-output devices. For example, the external environment of propulsion logic would consist of motors, line switches, tachometers, etc., which could be simulated by special test equipment or by hybrid or digital computers.

- Subsystem testing of equipment response to internal and external hardware failures, unusual operating conditions, and abnormal inputs. For example, test cases would be run to evaluate the response to injected faults. Fault detection, equipment reaction, fault recovery, and diagnostic adequacy are evaluated.

- Subsystem testing over the full range of environmental conditions, including the extremes of vibration, temperature, and humidity, using a simulator that duplicates the input and output conditions seen during normal revenue service.

- Subsystem testing using the actual external devices that constitute the subsystem. At this time, the functions and performance of the entire subsystem are evaluated in detail.

The end result of the installation and checkout phase is fully tested software running in its final configuration in the subsystem hardware.

Following subsystem testing, post-installation testing and system testing will be conducted. For example, the propulsion system will be tested in the passenger vehicle. System integration testing will follow by operating the passenger vehicle on the Metro Rail System.

At the end of each test step, the software documentation must consist of configured documents revised from each previous step. It is crucial that the software requirement specification, pseudo code or program design language, flow charts, and source code are kept under rigorous configuration control. The configured software and related documentation must be accompanied by comprehensive test reports.

2.2.8 Operation and Maintenance

Systems that require operator intervention have problems associated with software handling. Because embedded microprocessors initialize automatically upon power-up or reset, the operation is self-sufficient until a hardware failure occurs, or until

a software error is detected or results in a functional failure. Therefore, the operational concerns are somewhat minimized.

Maintenance of microprocessors includes two main categories:

- Correction of hardware failures that occur randomly and those that occur because of design deficiencies or stresses that exceed the design limits
- Identification and correction of software errors that were not discovered during the test, installation, and checkout phases.

A third category of maintenance is modification of the software to change functional and performance characteristics of the subsystem, within the performance limitations of the equipment.

All three categories require full software documentation, detailed technical knowledge of the software, and adequate software development tools and simulators. Technical support will be provided by suppliers during the warranty period. After the warranty expires, the SCRTD may elect to do maintenance in-house or by outside contract. Regardless of the maintenance approach taken, Metro Rail should have available the tools and documentation necessary for complete software maintenance and evaluation.

For corrective maintenance, a three-stage diagnostic and repair process shall be provided. In escalating levels of complexity and skill required, the three stages are:

- Observation of external indications on processor or logic units; diagnosis and replacement carried out by a general service technician.
- Connection of a communications interface to off-load internal data for analysis by an electronic technician. The decision to reset, replace the entire unit, or replace the subassembly will depend on the diagnosis.
- Connection of special test equipment and data communication equipment to analyze subsystem equipment in detail,

perhaps requiring a maintenance engineer for evaluation of complex failures.

In all cases of equipment removal, the unit will be returned to electronic repair for restoration and a full retest using bench test equipment.

2.2.9 Retirement

It is unlikely that the microprocessor software of Metro Rail subsystems will need to be retired during the expected life of the equipment. Retirement is usually forced only when major changes occur in equipment performance requirements, when the hardware is obsolete, or when supplier support is no longer available or becomes prohibitively expensive. Even if the microprocessors are replaced before the subsystem equipment wears out, much of the software product may be reusable. Specifically, documentation down to the level of the program design language or pseudo code may be fully or partially reused when the microprocessor hardware is replaced as long as performance requirements have not changed.

A major issue for Metro Rail is to preclude early forced retirement of software by ensuring the availability of processor hardware components and other replacement parts for the useful life of the related subsystem equipment.

2.3 CONFIGURATION MANAGEMENT

Configuration Management is the tool used to control the production and maintenance of all documents, products, and other material associated with the software development cycle.

2.4 ENVIRONMENTAL CONSIDERATIONS

Microprocessors are sensitive to temperature and humidity, shock and vibration, dirt and other contamination, and electromagnetic interference. With few exceptions, however, the design and packaging requirements are similar to those of other complex digital circuits.

3.0 RECOMMENDED CHANGES TO THE PASSENGER VEHICLE SPECIFICATION

3.0 RECOMMENDED CHANGES TO THE PASSENGER VEHICLE SPECIFICATION

The following inserts are recommended for inclusion in the Passenger Vehicle Specification. The inserts are based on the March 1985 Prefinal Submittal of the Procurement Specifications Book, Contract No. A650.

3.1 PROPOSAL REQUIREMENTS, SECTION 3.0 GENERAL REQUIREMENTS, PAGE PR-11

Add the following text at the end of 3.0.E:

"Describe the techniques proposed to prevent electromagnetic interference of low-power control circuits, including analog circuits, digital logic, and microprocessors. Provide results of electromagnetic compatibility testing on other rail transit systems and describe how microprocessor logic and other circuits will react and recover in the event of momentary malfunctions caused by EMI."

3.2 PROPOSAL REQUIREMENTS, SECTION 3.0 GENERAL REQUIREMENTS, PAGE PR-11

Add the following text as Section 3.0.F:

"For electronic equipment, the discussion shall include the proposed diagnostic process for control logic, emphasizing self-diagnostics, fault indication and logging, troubleshooting on the passenger vehicle, correction of malfunctions, and repair of the electronic units."

Note that the corresponding text should be deleted from section 14 of the proposal requirements (14.0.E).

3.3 PROPOSAL REQUIREMENTS, SECTION 20.0 REQUIREMENTS FOR MANAGEMENT APPROACH TO THE PROJECT, PAGE PR-17

Add the following text, as three separate items, between item 20.0.H and 20.0.I (reassign item identification letters accordingly):

"A description of the Proposer's use of microprocessors for implementation of control functions, especially in major subsystems such as propulsion and service braking."

"A description of the Proposer's microprocessor software quality assurance program to meet the requirements of the Technical Provisions. Provide samples of software development procedures, software test and maintenance procedures, validation and verification procedures, and software configuration management plans that are being used on similar programs."

"A discussion of the proposed maintenance approach to microprocessor hardware and software including software documentation standards, fault-tolerance features, fault-isolation provisions, management of software changes, software maintenance, software development, and automated design tools."

3.4 SPECIAL PROVISIONS, SECTION 5.2 SPARE PARTS, PAGE SP-4

Add the following text:

"Availability of electronic components, including microprocessor components and integrated circuits, shall be supported by the Contractor and subsystem suppliers for a minimum period of 15 years. The Contractor shall establish a method for notifying the District if subsystem suppliers and their vendors plan to cease production of components. All integrated circuits shall be available from more than one independent source at the time of the Final Design Review. Integrated circuits shall not require special selection except for reliability screening."

3.5 TECHNICAL PROVISIONS, SECTION 2.1.2 DEFINITIONS AND ABBREVIATIONS, PAGES TP-2-1 THROUGH TP-2-11

Add the following definitions to Section 2.1.2.A:

"Algorithm. A finite set of well-defined rules that gives a sequence of operations for performing a specific task."

"Assemble. To translate a program expressed in an assembly language into a machine language and perhaps to link subroutines. Assembling is usually accomplished by substituting machine language operation codes for assembly language operation codes."

"Automated Design Tool. A software tool that aids in the synthesis, analysis, modeling, or documentation of a software design."

"Baseline. A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures."

"Code. To represent data or a computer program in a symbolic form that can be accepted by a processor."

"Compile. To translate a higher order language program into its relocatable or absolute machine code equivalent."

"Configuration Management. The process of identifying and defining the configuration items in a system, controlling the release and change of these items throughout the system life cycle, recording and reporting the status of configuration items and change requests, and verifying the completeness and correctness of configuration items."

"Data Dictionary. A collection of names of all data items used in a software system, together with relevant properties of those items; for example, length of data item, representation, etc."

"Embedded Software. Software for an embedded computer system. In transit applications, embedded software is usually restricted to the firmware of read-only memory."

"Emulation. The imitation of all or part of one computer system by another, primarily by hardware, so that the imitating computer system accepts the same data, executes the same programs, and achieves the same results as the imitated system."

"Fault Tolerance. The built-in capability of a system to provide continued correct execution in the presence of a limited number of hardware or software faults."

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

"Flow Chart. A graphic representation of the definition, analysis, or solution of a problem in which symbols are used to represent operations, data, flow, and equipment."

"Machine Code. Instructions and data that are directly executable by a computer."

"Module. A software program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading."

"Program Design Language. A language with special constructs and, sometimes, verification protocols used to develop, analyze, and document a design."

"Pseudo Code. A combination of programming language and natural language used for computer program design."

"Requirements Analysis. The process of studying user needs to arrive at a definition of system or software requirements."

"Source Code. A computer program that must be compiled, assembled, or interpreted before being executed by a computer."

"Validation. The process of evaluating software at the end of the software development process to ensure compliance with software requirements."

"Verification. The process of determining whether or not the products of a given phase of the software development cycle fulfill the requirements established during the previous phase."

3.6 TECHNICAL PROVISIONS, SECTION 3.2.1 GENERAL, PAGE TP-3-2

Add the following text as separate items 3.2.1, after 1.B:

"Where microprocessors are used, provisions shall be included for self-test upon power turn-on and periodically during operation. Persistent failure of the self-test shall be annunciated as the appropriate subsystem failure in the operator's cab. Proposed techniques for minimization of false indications shall be submitted to the SCRTD for approval. An indication of self-test pass/fail status shall be provided on the outside of the logic enclosure."

"Design of propulsion and brake logic shall be fault-tolerant such that single-point failures do not necessarily result in equipment shutdown."

3.7 TECHNICAL PROVISIONS, SECTION 3.2.2 MAINTAINABILITY REQUIREMENTS, PAGE TP-3-3

Add the following text as separate items in 3.2.2, after 2.K:

"Design of microprocessor-based logic and other complex digital logic shall include static indicators and diagnostic displays for troubleshooting and fault correction without use of additional test equipment."

"If microprocessor-based logic is used, the diagnostics shall include provisions to store a suitable record of conditions that precede a propulsion or braking failure. The diagnostic data shall be contained in nonvolatile memory and provisions made to off-load the data into a portable memory medium, such as a floppy disk, at the passenger vehicle maintenance facility."

"Diagnostics for microprocessor-based logic shall provide details of failures on a display located on the logic enclosure. The display shall provide the identification of the component where practicable and error messages if the software ceases to operate correctly due to software error or hardware failure."

3.8 TECHNICAL PROVISIONS, SECTION 3.17.1 ELECTRICAL INTERFERENCE, PAGE TP-3-26

Add the following text to the end of Section 3.17.1.B:

"Inputs and outputs shall be protected from electromagnetic interference by frequency-selective filters, transient suppression devices, or both where necessary. Outputs shall be protected against short-circuit conditions. Inputs and outputs shall be electrically isolated to prevent ground currents, and an isolated switching mode power supply shall be provided."

3.9 TECHNICAL PROVISIONS, SECTION 6.1.1 CITED REFERENCES, PAGE TP-6-7

Add the following reference:

"MIL STD-883, Test Methods and Procedures for Microelectronics."

3.10 TECHNICAL PROVISIONS, SECTION 6.4.6 MISCELLANEOUS ELECTRONIC COMPONENTS, PAGE TP-6-31

Add the following text as separate items in 6.4.6.E, after E.3:

"Microprocessor components and other integrated circuits shall be screened to MIL STD-883 or an approved equivalent. Plastic packages shall not be used unless the equivalent is not available in a ceramic package. Use of plastic packages will require prior approval for each component application."

"Integrated circuit packages with more than 24 pins shall be mounted in sockets and not soldered directly to the printed circuit boards. The sockets shall retain the integrated circuits under worst-case resonance conditions when subjected to the shock and vibration specified in Article 3.16.1.A. Sockets with a circular internal cross section shall be the preferred choice."

"Integrated circuits containing microprocessor firmware shall be mounted in sockets to permit removal for reprogramming. The integrated circuits shall be positively identified with the configuration revision of embedded software. The microprocessor software shall include provisions to inhibit equipment operation unless a correctly configured set of integrated circuits containing the software are inserted into the correct sockets on the printed circuit boards. Provisions shall be provided for expansion of memory."

3.11 TECHNICAL PROVISIONS, SECTION 19.1.1 CITED REFERENCES, PAGE TP-19-1

Add the following reference:

"IEEE 730 Software Quality Assurance Plans."

3.12 TECHNICAL PROVISIONS, SECTION 19.1.3 SYSTEMS ASSURANCE APPROACH, PAGE TP-19-2

Add the following text as Section 19.1.3.F:

"Qualitative evaluation of the effects of microprocessor hardware failures and software errors on the safety of the system. The qualitative analysis of microprocessor software safety will be heavily dependent on demonstration by the suppliers of a top-down stepwise refinement of the design; thorough configuration control of the software documentation; and a comprehensive software verification and

validation process including testing of the software by rigorous test cases. Test cases shall include verification of correct response to normal input conditions and acceptable response to abnormal input conditions."

3.13 TECHNICAL PROVISIONS, SECTION 19.3.3 RELIABILITY ANALYSES (CDRL), PAGE TP-19-8

Add the following text as Section 19.3.3.D:

"Predict the reliability of microprocessor software from previous experience with identical or similar software in rail transit revenue service, or by a recognized reliability evaluation methodology."

3.14 TECHNICAL PROVISIONS, SECTION 19.4 MAINTAINABILITY, PAGE TP-19-10

Add the following as Section 19.4.4.

"Maintenance of Electronic Equipment

"A. The maintainability program shall detail the approach to electronic equipment maintenance.

"B. For corrective maintenance, a three-stage diagnostic and repair process shall be provided. In escalating levels of complexity and skill required, the three stages are:

1. Observation of external indications on processor or logic units; diagnosis and replacement carried out by a general service technician.

2. Connection of a communications interface to off-load internal data for analysis by an electronic technician. The decision to reset, replace the entire unit, or replace the subassembly will depend on the diagnosis.

3. Connection of special test equipment and data communication equipment to analyze subsystem equipment in detail, perhaps requiring a maintenance engineer for evaluation of complex failures.

"C. In all cases of equipment removal, the unit will be returned to electronic repair for restoration and a full retest using bench test equipment."

3.15 TECHNICAL PROVISIONS, SECTION 19.5.2 QUALITY ASSURANCE PROGRAM (CDRL), PAGE TP-19-11

Add the following text as Section 19.5.2.C:

"For microprocessor-based equipment, prepare a software quality assurance plan based on IEEE STD-730 or a comparable standard approved for use during the conceptual design review. A software configuration management plan shall be provided as a separate document."

3.16 TECHNICAL PROVISIONS, SECTION 20.6.2 GENERAL REQUIREMENTS, PAGE TP-20-4

Change the text to read:

"... encompassing system hardware, microprocessor software, and interfaces between subsystems."

3.17 TECHNICAL PROVISIONS, SECTION 20.6 CONFIGURATION MANAGEMENT PROGRAM, PAGE TP-20-5

Add the following text as a new section after 20.6.7 (renumber subsequent sections):

"Microprocessor Software Baseline

"For the purposes of change control, the microprocessor software baseline shall be established in the Final Design Review. Changes after the Final Design Review shall be documented in the form of Engineering Change Proposals and submitted for approval."

3.18 TECHNICAL PROVISIONS, SECTION 20.6.9.C CONCEPTUAL DESIGN REVIEW (CDR), PAGE TP-20-6

Add the following text as separate items in 20.6.9.C.1, after 1.f:

"Include design concepts and a requirements analysis for use of microprocessor-based logic; a separate analysis shall be provided for each subsystem in which microprocessors are proposed for use. The analysis shall include the partitioning of functions into hardware and software."

"Include master schedule for software development tasks."

3.19 TECHNICAL PROVISIONS, SECTION 20.6.9.D PRELIMINARY DESIGN REVIEW (PDR), PAGE TP-20-8

Add the following text as separate items in 20.6.9.D.2, after 2.j:

"Electronic assembly definition as follows:

- General equipment arrangement and packaging and weight constraints
- Description of system operation
- Function block diagrams
- Preliminary interface definitions
- Input, output, and transfer function criteria
- Allocation of hardware resources and estimation of spare capacity requirements
- Structure of hardware functional elements showing logic to be performed in hardware, such as frequency-to-digital conversion, digital-to-analog conversion, and other signal conditioning
- Safety criteria
- Performance criteria."

"Microprocessor documentation as follows:

- Description of the software architecture and the software modules that, when combined, form the complete software architecture
- Specifications for each of the software modules that fully define the module requirements, including:
  - General operating procedures
  - Description of each module
  - Inputs and outputs

- Logic flow diagrams
- Data flow diagrams
- Timing requirements and constraints
- Safety criteria
- Performance criteria
- Memory budget size.

• Description of the microprocessor hardware components and second sources of component supply

• Description of the programming language to be used

• The Test Plan for the microprocessor hardware and software, including details of the verification and validation process to be used with emphasis on the design, installation, test, and final checkout phases (CDRL)."

3.20 TECHNICAL PROVISIONS, SECTION 20.6.9.E FINAL DESIGN REVIEW (FDR), PAGE TP-20-8

Add the following text as separate items in 20.6.9.E, after E.3:

"Electronic assembly hardware documentation submitted for the FDR shall consist of:

A. Functional block diagrams

B. Printed circuit board schematics and interconnection diagrams

C. Final definitions of internal and external signal interfaces, including waveforms, rise and fall times, impedances, and permitted tolerances for the signals

D. Description of operations and fault isolation procedures."

"Microprocessor software documentation submitted for the FDR shall consist of:

A. Flow charts or structure charts that give an overview of the processor software

B. Completed algorithms, with complete annotations, expressed in program design language or pseudo code

C. Input data definitions

D. Output data definitions

E. Interrupt structure definition

F. Program parameters

G. An analysis of timing constraints and memory size that demonstrates a minimum operating margin of 30 percent and ability to expand memory up to 30 percent for future enhancements

H. Diagnostic routines for processor self-test and subsystem self-test

I. Error handling routines

J. Data dictionary."

"The hardware and software documentation submitted for the FDR shall require approval before proceeding with coding. The approved FDR hardware and software documentation shall form the configuration baseline."

3.21 TECHNICAL PROVISIONS, SECTION 21.2.3 PROCEDURES (CDRL), PAGE TP-21-3

Insert the following text, preferably following 21.2.3.C, and reletter accordingly:

"Microprocessor software test procedures and test cases designed to uncover software errors."

3.22 TECHNICAL PROVISIONS, SECTION 21.6 SUPPLEMENTAL TESTS, PAGE TP-21-16

Add the following text as Section 21.6.7 and renumber other sections accordingly:

"Tests of Electronic Assemblies

"A. In addition to normal quality control procedures, acceptance testing shall be performed on electronic hardware at each subsupplier's facility.

"B. Tests shall verify that equipment is in conformance with these specifications and demonstrate that each piece of equipment is operating properly prior to being shipped to the Contractor's facility.

"C. All assemblies that contain electronic components shall successfully pass a minimum 72-hour temperature-cycled burn-in test without failure.

1. Testing shall be performed with the electronic assemblies powered and connected to a simulator that shall duplicate the input and output conditions seen during normal revenue service. For processors, the testing shall be performed with a configuration of software that has been approved by the District.

2. The electronic assemblies shall be placed in an environment chamber and cycled continuously between 25 degrees F and 140 degrees F. Each cycle shall be 8 hours in duration, consisting of 3 hours and 40 minutes at 140 degrees F, followed by a 20-minute rate of change to 25 degrees F, followed by 3 hours and 40 minutes at 25 degrees F and a 20-minute return to 140 degrees F.

3. All outputs of the electronic assemblies shall be monitored continuously for anomalous operation.

"D. The Contractor and Subsuppliers shall submit test plans and test results to demonstrate that electronic assemblies are adequately designed and packaged for the shock and vibration conditions specified in Article 3.16."

3.23 TECHNICAL PROVISIONS, SECTION 22.4 MICROPROCESSOR-BASED PRODUCTS, PAGE TP-22-5

Delete the existing text and replace with the following section numbered 22.3.3 (renumber the following sections):

"Software Documentation, Technical Support, and Maintenance

"The manuals for microprocessor-based equipment shall

provide a complete set of the configured documentation developed in preliminary and final design and modified thereafter.

"In addition to the design documentation that includes pseudo code or program design language, system-level structure charts or flow charts, data flow diagrams, and the data dictionary, the manuals shall contain:

- A complete listing of fully configured and annotated source code.
- A full description of interrupt sequences and other protocols.
- A complete listing of machine code indicating location in the micro-processor memory hardware.
- Memory maps and input-output maps.
- Full documentation of automated design tools, including in-circuit emulators, required to modify, compile, assemble, test, and evaluate the software. All development software used in the microprocessor software development shall be furnished and details of development hardware provided. The equipment required to reprogram the microprocessor memory shall be provided.

"Guidelines shall be provided for modification of the software, including limits and impacts of parameter modification, and for reconstruction of software self-checks such as checksums.

"Assistance in setting up software support systems shall be provided by the Contractor and the subsystem suppliers. In addition, recommendations shall be provided on minimum Contractor and subsystem supplier participation in software maintenance to preserve safety and reliability levels when software is modified."

3.24 TECHNICAL PROVISIONS, SECTION 22.6.1 GENERAL, PAGE TP-22-10

Add the following text as Section 22.6.1.C:

"The propulsion and braking special test equipment shall permit analysis of the historical record prior

to the failure and shall provide comprehensive diagnosis of intermittent failures."

3.25 TECHNICAL PROVISIONS, SECTION 22.6.2 SHOP-LEVEL EQUIPMENT, PAGE TP-22-10

Add the following text as Section 22.6.2.E:

"Shop-level equipment shall isolate failures of microprocessor equipment to the circuit board level and to a component, or group of components, if practicable. A software development system with in-circuit emulation capability shall be provided if the bench or semiportable DTE cannot isolate failures to the same level as an in-circuit emulator."

3.26 TECHNICAL PROVISIONS, SECTION 23 CONTRACT DATA REQUIREMENTS, TABLE 23-1, PAGE TP-23-7

Add the following items after item 2101:

"Microprocessor Software Verification and Validation Plan
Reference paragraph 20.6.9
Plan/10 copies
120 days/1 time/Approval required"

"Microprocessor Software Quality Assurance Plan
Reference paragraph 19.5.2
Plan/10 copies
120 days/1 time/Approval required"

"Microprocessor Software Configuration Management Plan
Reference paragraph 19.5.2
Plan/10 copies
120 days/1 time/Approval required"

4.0 RECOMMENDED CHANGES TO THE AUTOMATIC TRAIN CONTROL SPECIFICATION

4.0 RECOMMENDED CHANGES TO THE AUTOMATIC TRAIN CONTROL SPECIFICATION

The Automatic Train Control (ATC) Specification includes requirements for safety-critical equipment. Traditional train control equipment uses vital relays and vital logic to achieve the required level of safety. New designs using minicomputers and microcomputers are being introduced by system suppliers in safety-critical areas. The hardware and software design of computer-based safety equipment requires special attention in SCRTD's procurement process.

The additional requirements detailed in this chapter are intended for inclusion in the ATC procurement documents. The objectives of the additional requirements are to provide adequate monitoring and control of the ATC equipment procurement, to permit independent review of safety, and to allow the SCRTD to maintain the equipment. The ATC equipment procurement for MOS-1 will not have safety-critical equipment that is computer-based. Solid-state interlocking, however, may be used in subsequent system segments. The requirements listed below are intended for use in solid-state vital interlockings, solid-state nonvital interlockings, and computer-based ATC equipment in general. The changes are based on the September 1985 Final Submittal of the Automatic Train Control Specification.

4.1 PROPOSAL REQUIREMENTS, SECTION 3.0 MANAGEMENT APPROACH TO THE PROJECT, PAGE PR-12

Add the following text to the end of the last paragraph:

"Include a description of the proposed methods for software verification and validation."

4.2 SPECIAL PROVISIONS, SECTION 6.2 SPARE PARTS, PAGE SP-4

Add the following text to Section 6.2:

"Availability of electronic components, including microprocessor components and integrated circuits, shall be supported by the Contractor for a minimum period of 15 years. The Contractor shall establish a method for notifying the District if their vendors

plan to cease production of components. All integrated circuits shall be available from more than one independent source at the time of the Final Design Review. Integrated circuits shall not require special selection except for reliability screening."

4.3 GENERAL PROVISIONS, SECTION 34.1 TECHNICAL DATA, PAGE GP-26

Replace 34.1.D with the following text:

"Computer and microprocessor software documentation including program design language or pseudo code listings, fully annotated source code, and machine level listings; and"

4.4. TECHNICAL PROVISIONS, SECTION 2.4 DEFINITIONS, PAGES TP-2-5 THROUGH TP-2-19

Add the following definitions to Section 2.4:

"Automated design tool - A software tool that aids in the synthesis, analysis, modeling, or documentation of a software design."

"Baseline - A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures."

"Code - To represent data or a computer program in a symbolic form that can be accepted by a processor."

"Compile - To translate a higher order language program into its relocatable or absolute machine code equivalent."

"Data Dictionary - A collection of names of all data items used in a software system, together with relevant properties of those items; for example, length of data item, representation, etc."

"Embedded Software - Software for an embedded computer system. In transit applications, embedded software is usually restricted to the firmware of read-only memory."

"Emulation - The imitation of all or part of one computer system by another, primarily by hardware, so that the imitating computer system accepts the same data, executes the same programs, and achieves the same results as the imitated system."

"Fault Tolerance - The built-in capability of a system to provide continued correct execution in the presence of a limited number of hardware or software faults."

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

"Flow Chart - A graphic representation of the definition, analysis, or solution of a problem in which symbols are used to represent operations, data, flow, and equipment."

"Machine Code - Instructions and data that are directly executable by a computer."

"Module - A software program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading."

"Program Design Language - A language with special constructs and, sometimes, verification protocols used to develop, analyze, and document a design."

"Pseudo Code - A combination of programming language and natural language used for computer program design."

"Requirements Analysis - The process of studying user needs to arrive at a definition of system or software requirements."

"Source Code - A computer program that must be compiled, assembled, or interpreted before being executed by a computer."

"Validation - The process of evaluating software at the end of the software development process to ensure compliance with software requirements."

"Verification - The process of determining whether or not the products of a given phase of the software development cycle fulfill the requirements established during the previous phase."

4.5 TECHNICAL PROVISIONS, SECTION 3.3.2 FAIL-SAFE DESIGN, PAGE TP-3-8

Add the following text as separate items in 3.3.2.:

"Computer-based equipment shall prevent latent software faults from causing unsafe conditions. Protec-

tion shall be provided against hardware failures, EMI, or unusual operating conditions that may cause the software to execute incorrectly."

"Vital logic functions implemented by computer-based equipment shall have protection features equivalent to those commonly employed for vital relay logic and fail-safe circuit design. Loss of power, component failures, damaged connections, wiring insulation failures, changes in memory, or similar failures shall not result in an unsafe condition."

4.6 TECHNICAL PROVISIONS, SECTION 3.5.2 ELECTROMAGNETIC COMPATIBILITY CONTROL PROGRAM, PAGE TP-3-28

Add the following text as Section 3.5.2.G:

"Submit for approval the techniques proposed to prevent electromagnetic interference of low-power control circuits, including analog circuits, digital logic, and microprocessors. Provide results of electromagnetic compatibility testing of electronic equipment on other rail transit systems and describe how microprocessor logic will react and recover in the event of momentary malfunctions caused by EMI."

4.7 TECHNICAL PROVISIONS, SECTION 3.6.2 SHIELDING, PAGE TP-3-29

Add the following text as Section 3.6.2.M:

"Inputs and outputs of electronic circuits shall be protected from electromagnetic interference by frequency-selective filters, transient suppression devices, or a combination of both where necessary. Outputs shall be protected against open-circuit and short-circuit conditions."

4.8 TECHNICAL PROVISIONS, SECTION 11.1.1 CITED REFERENCES, PAGE TP-11-3

Add the following reference:

"MIL STD-883, Test Methods and Procedures for Micro-electronics."

4.9 TECHNICAL PROVISIONS, SECTION 11.5.2 SEMICONDUCTORS, PAGE TP-11-14

Add the following text as separate items in 11.5.2:

"Microprocessor components and other integrated circuits

shall be screened to MIL STD-883 or an approved equivalent. Plastic packages shall not be used unless the equivalent is not available in a ceramic package. Use of plastic packages will require prior approval for each component application."

"Integrated circuit packages with more than 24 pins shall be mounted in sockets and not soldered directly to the printed circuit boards. For Automatic Train Control vehicle equipment, the sockets shall retain the integrated circuits under worst-case resonance conditions when subjected to the shock and vibration specified in Article 8.2.3. Sockets with a circular internal cross section shall be the preferred choice."

"Integrated circuits containing microprocessor firmware shall be mounted in sockets to permit removal for reprogramming. The integrated circuits shall be positively identified with the configuration revision of the embedded software. The microprocessor software shall include provisions to inhibit equipment operation unless a correctly configured set of integrated circuits containing the software are inserted into the correct sockets on the printed circuit boards".

4.10 TECHNICAL PROVISIONS, SECTION 13.6 FACTORY ACCEPTANCE TESTS, PAGE TP-13-10

Add the following text as Section 13.6.2 and renumber sections accordingly:

"Tests of Electronic Assemblies

"A. Acceptance testing shall be performed on electronic hardware at the Contractor's and each subsupplier's facility. Tests shall verify that equipment is in conformance with these specifications and demonstrate that each piece of equipment is operating properly prior to being shipped to the District.

"B. All assemblies that contain electronic components shall successfully pass a minimum 72-hour temperature-cycled burn-in test without failure.

1. Testing shall be performed with the electronic assemblies powered and connected to a simulator that shall duplicate the input and output conditions seen during normal revenue service. For processors, the testing shall be performed with a configuration of software that has been approved by the District.

2. The electronic assemblies shall be placed in an environment chamber and cycled continuously between 25 degrees F and 150 degrees F. Each cycle shall be 8 hours in duration, consisting of 3 hours and 40 minutes at 150 degrees F, followed by a 20 minute rate of change to 25 degrees F, followed by 3 hours and 40 minutes at 25 degrees F and a 20-minute return to 150 degrees F.

3. All outputs of the electronic assemblies shall be monitored continuously for anomalous operation."

4.11 TECHNICAL PROVISIONS, SECTION 14.5.2 GENERAL REQUIREMENTS, PAGE TP-14-4

Add the following text as Section 14.5.2.C:

"For the purposes of change control, the computer and microprocessor software baselines shall be established in the Final Design Review. Changes after the Final Design Review shall be documented in the form of Engineering Change Proposals and submitted for approval."

4.12 TECHNICAL PROVISIONS, SECTION 14.5.9 DESIGN AND CONFIGURATION REVIEW AND AUDITS, PAGES TP-14-6 AND TP-14-7

(1) Add the following text as Section 14.5.9.C.1.f:

"Include the design concepts and a requirements analysis for use of microprocessor-based logic; a separate analysis shall be provided for each subsystem in which microprocessors are proposed for use."

(2) Add the following text as Section 14.5.9.C.5:

"The data package shall include a discussion of the proposed maintenance approach to microprocessor hardware and software, including software documentation standards, fault-tolerance features, fault-isolation provisions, management of software changes, and software maintenance and automated design tools."

(3) Add the following text as Section 14.5.9.D.3:

"The design data submittal shall contain microprocessor documentation as follows:

- Description of the software architecture and the software modules which,

when combined, form the complete software architecture.

- Specifications for each of the software modules that fully define the module requirements including:
  - General operating procedures
  - Description of each module
  - Inputs and outputs
  - Logic flow diagrams
  - Data flow diagrams
  - Timing requirements and constraints
  - Safety criteria
  - Performance criteria
  - Memory budget size.
- Description of the microprocessor hardware components and second sources of component supply
- Description of the programming language to be used
- The Test Plan for the microprocessor hardware and software, including details of the verification and validation process to be used (CDRL)."

(4) Add the following text as Section 14.5.9.E.3:

"Microprocessor software documentation submitted for the FDR shall consist of:

- Flow charts or structure charts that give an overview of the processor software
- Completed algorithms expressed in program design language or pseudo code
- Input data definitions

- Output data definitions
- Interrupt structure definition
- Program parameters
- Diagnostic routines for processor self-test and subsystem self-test
- Error handling routines
- Data dictionary.

"The software documentation submitted at the FDR shall require approval before proceeding with coding. The approved FDR software documentation shall form the software baseline."

4.13 TECHNICAL PROVISIONS, SECTION 15.1.2 SYSTEMS ASSURANCE PROGRAM PLAN, PAGE TP-15-1

Add the following text as Section 15.1.2.E:

"Software Verification and Validation."

4.14 TECHNICAL PROVISIONS, SECTION 15.2 SYSTEM SAFETY PROGRAM, PAGE TP-15-5

Add the following text as Section 15.2.3:

"Quantitative Requirements

"A quantitative hazard analysis for safety-critical processor-based train control equipment shall be submitted. The quantitative analysis shall be based on fault tree analysis or an equivalent methodology. The quantification shall be based on component failure rates, built-in-test provisions, and a software reliability prediction. An alternative to the quantification of software reliability is a comprehensive software verification and validation process including testing of the software by rigorous test cases."

4.15 TECHNICAL PROVISIONS, SECTION 15.3.3 RELIABILITY ANALYSES, PAGE TP-15-8

Add the following text as Section 15.3.3.D:

"Predict the reliability of processor software from previous experience with identical or similar software in rail transit revenue service, or by a recognized reliability evaluation methodology."

4.16 TECHNICAL PROVISIONS, SECTION 15.5.2 PROGRAM IMPLEMENTATION, PAGE TP-15-12

Add the following text as Section 15.5.2.C:

"For microprocessor-based equipment, prepare a software quality assurance plan based on IEEE STD-730 or based on a comparable standard approved for use during the conceptual design review."

4.17 TECHNICAL PROVISIONS, SECTION 16.4 MICROPROCESSOR-BASED PRODUCTS, PAGE TP-16-5

Delete the existing text and replace with the following:

"The manuals for microprocessor-based equipment shall provide a complete set of the configured documentation developed in preliminary and final design and modified thereafter.

"In addition to the design documentation that includes pseudo code or program design language, system-level structure charts or flow charts, data flow diagrams, and the data dictionary, the manuals shall contain:

- A complete listing of fully configured and annotated source code.
- A full description of interrupt sequences and other protocols.
- A complete listing of machine code, indicating location in the microprocessor memory hardware.
- Memory maps and input-output maps.
- Full documentation of automated design tools, including in-circuit emulators, required to modify, compile, assemble, test, and evaluate the software. All development software used in the microprocessor software development shall be furnished and details of development hardware provided. The equipment required to reprogram the microprocessor memory shall be provided.

"Guidelines shall be provided for modification of the software, including limits and impacts of parameter modification, and for reconstruction of software self-checks such as checksums.

"Assistance in setting up software support systems shall be provided by the Contractor. In addition, recommendations shall be provided on minimum Contractor participation in software maintenance to preserve safety and reliability levels when firmware or software is modified."

4.18 TECHNICAL PROVISIONS, SECTION 16.6.2 SHOP-LEVEL EQUIPMENT, PAGE TP-16-8

Add the following as Section 16.6.2.E:

"Shop-level equipment shall isolate failures of microprocessor equipment to the circuit board level and to a component, or group of components, if practicable. A software development system with in-circuit emulation capability shall be provided if the bench or semiportable DTE cannot isolate failures to the same level as an in-circuit emulator."

4.19 TECHNICAL PROVISIONS, SECTION 17 CONTRACT DATA REQUIREMENTS, TABLE 17-1, PAGE TP-17-9

Add the following items:

"Microprocessor Software Verification and Validation Plan
Paragraph reference 15.5.2
Plan/10 copies
120 days/1 time/Approval required"

"Microprocessor Software Quality Assurance Plan
Plan/10 copies
Paragraph reference 15.1.2
120 days/1 time/Approval required"

5.0 RECOMMENDED CHANGES TO THE COMMUNICATIONS SPECIFICATION

5.0 RECOMMENDED CHANGES TO THE COMMUNICATIONS SPECIFICATION

The following changes to the Communications System Specification to strengthen the requirements for microprocessor software design, documentation, and support. They are based on the November 1985 Prefinal Submittal of the Communications Specification.

5.1 SPECIAL PROVISIONS, SECTION 1.0 STATEMENT OF WORK, PAGE SP-1

In Section 1.0.D, change the phrase "computer and microprocessor software" to "computer and microprocessor software documentation."

5.2 GENERAL PROVISIONS, SECTION 34.1 TECHNICAL DATA, PAGE GP-26

Replace Section 34.1.D with the following:

"Computer and microprocessor software documentation including program design language or pseudo code listings, fully annotated source code, and machine-level listings; and"

5.3 TECHNICAL PROVISIONS, SECTION 1.4.4 DEFINITIONS, PAGES TP-1-10 THROUGH TP-1-15

Add the following definitions:

"Algorithm - A finite set of well-defined rules that gives a sequence of operations for performing a specific task."

"Baseline - A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures."

"Code - To represent data or a computer program in a symbolic form that can be accepted by a processor."

"Compile - To translate a higher order language program into its relocatable or absolute machine code equivalent."

"Configuration Management - The process of identifying and defining the configuration items in a system, controlling the release and change of these items throughout the system life cycle, recording and reporting the status of configuration items and change requests, and verifying the completeness and correctness of configuration items."

"Data Dictionary - A collection of names of all data items used in a software system, together with relevant properties of those items; for example, length of data item, representation, etc."

"Flow Chart - A graphic representation of the definition, analysis, or solution of a problem in which symbols are used to represent operations, data, flow, and equipment."

"Machine Code - Instructions and data that are directly executable by a computer."

"Module - A software program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading."

"Program Design Language - A language with special constraints and, sometimes, verification protocols used to develop, analyze, and document a design."

"Pseudo Code - A combination of programming language and natural language used for computer program design."

"Source Code - A computer program that must be compiled, assembled, or interpreted before being executed by a computer."

"Validation - The process of evaluating software at the end of the software development process to ensure compliance with software requirements."

"Verification - The process of determining whether or not the products of a given phase of the software development cycle fulfill the requirements established during the previous phase."

5.4 TECHNICAL PROVISIONS, SECTION 5.3.2 TECHNICAL CHARACTERISTICS, PAGE TP-5-9

In Section 5.3.2.B, change the last sentence of the first paragraph to read as follows:

"The programming language and software documentation

shall be such as to require a minimum of time and labor to prepare and alter the programs."

5.5 TECHNICAL PROVISIONS, SECTION 9.7.1 DESIGN CHARACTERISTICS, PAGE TP-9-135

Add the following text at the end of 9.7.1:

"The software design shall be structured and top-down documentation shall be provided. Structure charts or top-level flow charts and program design language or pseudo code shall be provided. All source code in higher level language or assembly language shall be fully annotated."

5.6 TECHNICAL PROVISIONS, SECTION 13.7.2 SEMICONDUCTORS, PAGE TP-13-32

Add the following text as separate items in 13.7.2:

"Integrated circuit packages with more than 24 pins shall be mounted in sockets and not soldered directly to the printed circuit boards. Sockets with a circular internal cross section shall be the preferred choice."

"Integrated circuits containing microprocessor firmware shall be mounted in sockets to permit removal for reprogramming. The integrated circuits shall be positively identified with the configuration revision of embedded software. The microprocessor software shall include provisions to inhibit equipment operation unless a correctly configured set of integrated circuits containing the firmware are inserted into the correct sockets on the printed circuit boards."

5.7 TECHNICAL PROVISIONS, SECTION 17.2.2 MANAGEMENT PLAN, PAGE TP-17-2

Add the following text as Section 17.2.2.H:

"A schedule of software development and testing."

5.8 TECHNICAL PROVISIONS, SECTION 17.6.2 GENERAL REQUIREMENTS, PAGE TP-17-4

Add the following text as Section 17.6.2.C:

"Identify the design baseline for computer and microprocessor software and control the software changes through the verification and validation process."

5.9 TECHNICAL PROVISIONS, SECTION 17.6.9 DESIGN AND CONFIGURATION REVIEW, PAGE TP-17-10

Add the following text at the end of Section 17.6.9.E.3:

"The software documentation submitted at the FDR shall require approval before proceeding with the coding. The approved FDR software documentation shall form the software baseline.

"Changes after the Final Design Review shall be documented in the form of Engineering Change Proposals and submitted for approval."

5.10 TECHNICAL PROVISIONS, SECTION 21 CONTRACT DATA REQUIREMENTS, TABLE 21-1, PAGE TP-21-13

Add the following items:

"Software Verification and Validation Plan
Reference paragraph 17.6.9
Plan/10 copies
120 days/1 time/Approval required"

"Software Quality Assurance Plan
Reference paragraph 18.5.2
Plan/10 copies
120 days/1 time/Approval required"

"Software Configuration Management Plan
Reference paragraph 18.5.2
Plan/10 copies
120 days/1 time/Approval required"

6.0 RECOMMENDED CHANGES TO THE FARE COLLECTION SPECIFICATION

6.0 RECOMMENDED CHANGES TO THE FARE COLLECTION SPECIFICATION

The following changes to the Fare Collection Specification are recommended to strengthen the requirements for processor software design, documentation, and support. They are based on the October 1985 Prefinal Submittal of Contract No. A660.

6.1 REQUEST FOR TECHNICAL PROPOSAL, SECTION 2.0 GENERAL REQUIREMENTS, PAGE RFTP-10

Add the following text as a new Section after 2.0.D (reassign item identification letters accordingly):

> "Provide description of proposed software languages and the techniques for modifying application software to update fare structure and fare zones and to obtain ad hoc central computer reports. Describe the software compilers and assemblers that are required to develop and modify the software for each piece of processor-based equipment."

6.2 REQUEST FOR TECHNICAL PROPOSAL, SECTION 19.0 MANAGEMENT PROGRAM, PAGES RFTP-14 AND RFTP-15

(1) Expand the second sentence of Section 19.0 to read as follows:

> "The Proposer must demonstrate sufficient understanding of the management techniques required for proper implementation and control of the Work with special emphasis on Contract Data Requirements, control of systems engineering and design, development and verification of new processor software, modification of existing processor software, and planning of the integration planning for the integrated test program."

(2) Change section 19.0.I to read:

> "A description of the proposed test program including verification and validation of software. Use of block diagrams and brief narratives is acceptable."

6.3 BIDDING REQUIREMENTS, BID FORM A, PAGE BF-8

Add the following item:

"Description	Unit	Estimated Quantity
Software development system(s) for FCCC Computer and all other processor-based equipment	Lump Sum	1"

6.4 SPECIAL PROVISIONS, SECTION 1.0 STATEMENT OF WORK, PAGE SP-1

In Section 1.0.D, after the word "plans," add "computer and microprocessor software documentation."

6.5 GENERAL PROVISIONS, SECTION 34.1 TECHNICAL DATA, PAGE GP-26

Replace Section 34.1.D with the following:

"Computer and microprocessor software documentation including program design language or pseudo code listings, fully annotated source code, and machine level listings; and"

6.6 TECHNICAL PROVISIONS, SECTION 1.5.3 DEFINITIONS, PAGES TP-1-6 THROUGH TP-1-15

Add the following definitions:

"Algorithm - A finite set of well-defined rules that gives a sequence of operations for performing a specific task."

"Baseline - A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures."

"Code - To represent data or a computer program in a symbolic form that can be accepted by a processor."

"Compile - To translate a higher order language program into its relocatable or absolute machine code equivalent."

"Configuration Management - The process of identifying and defining the configuration items in a system, con- trolling the release and change of these

items through- out the system life cycle, recording and reporting the status of configuration items and change requests, and verifying the completeness and correctness of configu- ration items."

"Data Dictionary - A collection of names of all data items used in a software system, together with relevant properties of those items; for example, length of data item, representation, etc."

"Flow Chart - A graphic representation of the definition, analysis, or solution of a problem in which symbols are used to represent operations, data, flow, and equipment."

"Machine Code - Instructions and data that are directly executable by a computer."

"Module - A software program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading."

"Program Design Language - A language with special con- straints and, sometimes, verification protocols used to develop, analyze, and document a design."

"Pseudo Code - A combination of programming language and natural language used for computer program design."

"Source Code - A computer program that must be compiled,
assembled, or interpreted before being executed by a computer."

"Validation - The process of evaluating software at the end of the software development process to ensure com- pliance with software requirements."

"Verification - The process of determining whether or not the products of a given phase of the software development cycle fulfill the requirements established during the previous phase."

6.7 TECHNICAL PROVISIONS, SECTION 1.5.3 DEFINITIONS, PAGE TP-1-13

The definition of software includes "circuit diagrams," which are generally not classified as software. A suggested change is to replace "circuit diagrams" with "any associated documentation pertaining to the operation of a computer system." This change would cover circuit diagrams, flow charts, source code, etc.

6.8 TECHNICAL PROVISIONS, SECTION 9.5.1 REQUIREMENTS, PAGE TP-9-11

Add the following text at the end of 9.5.1:

"The software design shall be structured and top-down documentation shall be provided. Structure charts or top-level flow charts and program design language or pseudo code shall be provided. All source code in higher level language or assembly language shall be fully annotated."

6.9 TECHNICAL PROVISIONS, SECTION 16.3.1 SEMICONDUCTOR DEVICES, PAGE TP-16-4

Add the following requirements:

"Integrated circuit packages with more than 24 pins shall be mounted in sockets and not soldered directly to the printed circuit boards. Sockets with a circular internal cross section shall be the preferred choice.

"Integrated circuits containing microprocessor firmware shall be mounted in sockets to permit removal for reprogramming. The integrated circuits shall be positively identified with the configuration revision of embedded software. The microprocessor software shall include provisions to inhibit equipment operation unless a correctly configured set of integrated circuits containing the firmware are inserted into the correct sockets on the printed circuit boards."

6.10 TECHNICAL PROVISIONS, SECTION 19.2.2 MANAGEMENT PLAN PAGE, TP-19-1

Add the following text as Secton 19.2.2.H:

"Schedule of software development and testing."

6.11 TECHNICAL PROVISIONS, SECTION 19.6.2 GENERAL REQUIREMENTS, PAGE TP-19-4

Add the following text as Section 19.6.2.C:

"Identify the design baseline for computer and microprocessor software and control the software changes through the verification and validation process."

6.12 TECHNICAL PROVISIONS, SECTION 19.6.9 DESIGN AND CONFIGURATION REVIEW AND AUDITS, PAGE TP-19-7

(1) Add the following text as Section 19.6.9.D.3:

"Processor documentation submitted for the PDR shall consist of:

- Description of the software architecture and the software modules which, when combined, form the complete software architecture
- Specifications for each of the software modules that fully define the module requirements including:
  - General operating procedures
  - Description of each module
  - Inputs and outputs
  - Logic flow diagrams
  - Data flow diagrams
  - Timing requirements and constraints
  - Safety criteria
  - Performance criteria
  - Memory budget size
- Description of the processor hardware components and second sources of component supply
- Description of the programming languages to be used
- The Test Plan for the processor hardware and software including details of the verification and validation process to be used (CDRL)."

(2) Add the following text as Section 19.6.9.E.3:

"Processor software documentation submitted for the

FDR shall consist of:

- Flow charts or structure charts that give overview of processor software
- Completed algorithms expressed in program design language or pseudo code
- Input data definitions
- Output data definitions
- Interrupt structure definition
- Program parameters
- Diagnostic routines for processor self test and subsystem self test
- Error handling routines
- Data dictionary."

"The software documentation submitted at the FDR shall require approval before proceeding with coding. The approved FDR software documentation shall form the software baseline."

"Changes after the Final Design Review shall be documented in the form of Engineering Change Proposals and submitted for approval."

6.13 TECHNICAL PROVISIONS, SECTION 18.5.2, QUALITY ASSURANCE

Change Section 18.5.2.C to read as follows:

"For processor-based equipment, a software QA plan shall be submitted based upon IEEE STD-730 or based upon a comparable standard. The comparable standard shall be proposed for District approval no later than the conceptual design review. (CDRL) A separate software configuration management plan shall be submitted."

6.14 TECHNICAL PROVISIONS, SECTION 21.4 MANUALS, PAGE TP-21-2

Add the following requirements as Section 21.4.6 and renumber following sections accordingly:-

"Software Documentation and Maintenance

"The manuals for processor-based equipment shall

provide a complete set of the configured documentation developed in preliminary and final design and modified thereafter.

"In addition to the design documentation that includes pseudo code or program design language, system-level structure charts or flow charts, data flow diagrams, and the data dictionary, the manuals shall contain:

- A complete listing of fully configured and annotated source code
- A full description of interrupt sequences and other protocols
- A complete listing of machine code indicating location in the processor memory hardware
- Memory maps and input-output maps
- Full documentation of software development tools required to modify, compile, assemble, test, and evaluate the software.

"All development software used in the processor software development shall be furnished and details of any special development hardware provided. The equipment required to reprogram the processor firmware shall be provided.

"Guidelines shall be provided for modification of the software including limits and impacts of parameter modification, and for reconstruction of software self-checks such as checksums."

6.15 TECHNICAL PROVISIONS, SECTION 22 CONTRACT DATA REQUIREMENTS, TABLE 22-1, PAGE TP-22-8

Add the following items:

"Software Verification and Validation Plan
Reference paragraph 18.5.2
Plan/10 copies
120 days/1 time/Approval required"

"Software Quality Assurance Plan
Reference paragraph 18.5.2
Plan/10 copies
120 days/1 time/Approval required"

"Software Configuration Management Plan
Reference paragraph 19.6.9
Plan/10 copies
120 days/1 time/Approval required"

LA056480R

7.0 APPLICABLE DOCUMENTS

7.0 APPLICABLE DOCUMENTS

1. IEEE Standards Working Group (P1012), Standards for Software Verification and Validation Plans. [Note: This standard has not been released for use.]

2. IEEE Std-730, IEEE Standard for Software Quality Assurance Plans.

3. NBS Special Publication 500-93, Software Validation, Verification and Testing Technique and Tool Reference Guide.

4. Air Transport Association of American Specification 102, Specification for Computer Software Manual.

5. IEEE Std-828-1983, IEEE Standard for Software Configuration Management Plans.

6. NBS Special Publication 500-98, Planning for Software Validation, Verification and Testing.

7. IEEE Std-829-1983, IEEE Standard for Software Test Documentation.

8. IEEE Std-729-1983, IEEE Standard Glossary to Software Engineering Terminology.

9. NBS Special Publication 500-99, Structured Testing. A Software Testing Methodology Using the Cyclomatic Complexity Metric.

10. NBS Special Publication 500-75, Validation, Verification, and Testing of Computer Software.

11. FIPS PUB 101, Guidelines for Life Cycle Validation, Verification, and Testing of Computer Software.