ESD-TR-89-193

MTR-10454

Software Quality Assurance of the Groundwave Emergency Network: Final Report

By

A. M. Haley
T. E. Hutchinson

May 1989

Prepared for
Deputy Director for Strategic Command, Control & Communications Systems Program Office
Electronic Systems Division
Air Force Systems Command
United States Air Force
Hanscom Air Force Base, Massachusetts

Project No. 6730
Prepared by
The MITRE Corporation
Bedford, Massachusetts
Contract No. F19628-86-C-0001

Approved for public release;
distribution unlimited.

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

Lorelei A Krebs

LORELEI A. KREBS, Major, USAF
Deputy Program Manager, GWEN

FOR THE COMMANDER

GERALD G. IVERSON, Colonel, USAF
Director, Strategic Command,
Control & Communications
Systems Program Office

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) MTR-10454 ESD-TR-89-193	5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION The MITRE Corporation	6b. OFFICE SYMBOL *(If applicable)*	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS *(City, State, and ZIP Code)* Burlington Road Bedford, MA 01730		7b. ADDRESS *(City, State, and ZIP Code)*
8a. NAME OF FUNDING / SPONSORING ORGANIZATION Deputy Director (continued)	8b. OFFICE SYMBOL *(If applicable)* ESD/SZG	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F19628-86-C-0001

8c. ADDRESS *(City, State, and ZIP Code)*
Electronic Systems Division, AFSC
Hanscom AFB, MA 01731-5000

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
	6730		

11. TITLE *(Include Security Classification)*
Software Quality Assurance of the Groundwave Emergency Network: Final Report

12. PERSONAL AUTHOR(S)
Haley, A. M., Hutchinson, T. E.

13a. TYPE OF REPORT Final	13b. TIME COVERED FROM ________ TO ________	14. DATE OF REPORT *(Year, Month, Day)* 1989 May	15. PAGE COUNT 100

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
GWEN
Software Quality Assurance
Verification and Validation

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

MITRE carried out a Software Quality Assurance (SQA) effort for the Groundwave Emergency Network's Thin Line Connectivity Capability system during the early period of its development. The process included desktop analyses, simulations, and execution of the major portions of the software leading to formal qualification testing by the contractor. As a result, the contractor was alerted to potential omissions and incorrect implementations of the software, culminating in a final implementation that was essentially error-free. An important lesson learned from this type of effort is the need for early and thorough involvement of and interaction between contractor and SQA personnel.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Pamela J. Cunha	22b. TELEPHONE *(Include Area Code)* (617) 271-2844	22c. OFFICE SYMBOL Mail Stop D135

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete.

8a. for Strategic Command, Control & Communications Systems Program Office

ACKNOWLEDGMENTS

This document has been prepared by The MITRE Corporation under Project No. 6730, Contract No. F19628-86-C-0001. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, United States Air Force, Hanscom Air Force Base, Massachusetts 01731-5000.

The authors wish to acknowledge the following engineers for their contributions to the SQA effort: B. Barrett, D. Black, J. M. Cronin, M. W. Feeney, M. D. Gates, P. J. Nesky, and H. C. Reith, Jr.

We also wish to acknowledge the personnel of DSD Laboratories, Inc., of Wakefield, Massachusetts, and K. McClanahan, in particular, for their independent and timely assessment on the completeness and accuracy of GWEN software product specifications.

A special thanks to L. S. Meyer for her careful review and valuable suggestions, which greatly enhanced the content of this document, and to D. M. Reilly for sacrificing her personal time to provide word processing support.

EXECUTIVE SUMMARY

In 1984, MITRE initiated a Software Quality Assurance (SQA) effort for the mission-critical software of the Groundwave Emergency Network. Preliminary work was directed toward establishing a secure area for the "hands-on" examination of the various releases of the contractor-developed software. This laboratory, equipped with a VAX-11/730, microprocessor development workstations, and various tools for the examination of the software, allowed us to expand the software monitoring work that would normally have been conducted by MITRE.

The complement of GWEN software is comprised of 11 Computer Program Configuration Items (CPCIs) totaling approximately 30,000 lines of code in FORTRAN and 30,000 lines in Assembly language, and each line of code received a detailed examination. All specifications and test procedures were reviewed and a detailed desktop analysis of the code was performed for each CPCI as it was received. Several of the CPCIs were executed first in a microprocessor simulator residing on the VAX and then in emulation on the microprocessor development workstations. A test node station, designed by MITRE to be functionally identical to the node being built by the contractor, was constructed from commercial microprocessor components, and the software was also executed on this station.

For each CPCI release, we prepared a detailed assessment of its completeness and correctness down to the module level and then used this assessment to ensure schedule realism for the completion of software development. Because we had first executed or examined the code for all of the CPCIs, we were well prepared for the formal qualification testing of the software, having a full understanding of the test procedures and the expected results.

After assuring ourselves of the functional completeness of the software, we assessed the correctness and robustness of the algorithms. For example, we verified that the timing for packet decryption and for packet routing was sufficient and that the function could be accomplished within the real-time constraints of the GWEN system. We also tested the external interface portions of the algorithms to assure their correct operation. As part of this process, we verified the correct implementation of the TRANSEC algorithm and acquired a thorough understanding of the special use of the KG-84A cryptographic device in the GWEN system.

In some cases, such as the implementation of virtual circuits in the Network Control CPCI, we noted that early releases were incomplete, and during subsequent formal qualification testing, we noted that several features remained to be demonstrated or tested. After substantial work by the prime contractor, these deficiencies were corrected in the final version of the delivered software. During this SQA process, we discussed our findings directly and informally with the individual software developers. This direct and open communication allowed us to exchange information about the software and assist the contractor in identifying any "bugs," or shortfalls, in functionality.

Table 10 presents, on a CPCI-by-CPCI basis, the major findings from the desktop analyses of the software, while table 13 summarizes the memory usage of each CPCI. During the SQA effort, we tracked the increase in memory usage from release to release, and in several instances, we alerted the contractor to usage that would exceed the specification. To remedy this, the contractor used higher density memory boards for some CPCIs, and applied for and was granted a waiver in other cases.

During our examination of the CPCI for the Maintenance Terminal that is resident in an HP-85B desk-top computer, we noted that its use for station initialization and synchronization is time-consuming. Since the hardware is no longer manufactured, we believe it should be replaced by a newer, faster computer. As part of our SQA effort, we demonstrated the rehosting of this software to such a computer.

We also considered converting the portion of GWEN software in Assembly language, approximately 50 percent, to an approved higher-order language. Our analyses and calculations support such a conversion as both feasible and desirable.

As a result of the SQA conducted on GWEN, it is clear that early involvement of contractor, Government, and SQA personnel is essential. Of course, the appropriate paragraphs covering these activities must be included in the Statement of Work to the contractor. The increased interaction between the GWEN prime contractor and the MITRE SQA personnel proved to be very beneficial, since it resulted in a final implementation of the software that was essentially error-free.

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION 1

INTRODUCTION

1.1 PURPOSE

This report describes the work performed during MITRE's Software Quality Assurance (SQA) efforts on the Groundwave Emergency Network (GWEN) software development and qualification. The objective of the SQA effort was to provide a means of accomplishing in-depth testing and review of GWEN's mission-critical software prior to formal qualification testing by the contractor. The effort was initiated by MITRE's Survivable Communications Department (D95) in 1984, and was completed in January 1988.

An introductory review of the GWEN network components and software is followed by a description of the Software Quality Assurance facility (section 2) that was established to support the SQA undertaking. Section 3 outlines the methods used to evaluate the software and summarizes the findings. Conclusions and recommendations are given in section 4. Detailed discussions on analysis and testing of individual computer program configuration items are presented as appendices under separate cover.

1.2 COMPONENTS OF GWEN

The Groundwave Emergency Network is a packet-radio military communications system developed by the U.S. Air Force. Its mission is to provide survivable and enduring communications for the command and control of strategic forces in the continental United States. Approximately 30 military bases and eight command centers and sensors will be connected by the basic Thin-Line Connectivity Capability (TLCC) network.

The system employs low-frequency (LF) groundwave propagation between unmanned relay nodes (RNs) in a store-and-forward, packet-switched network. In addition to the LF relay nodes, there are two other classes of stations within GWEN: Input/Output (I/O) and Receive-Only (RO). The RO stations receive LF signals directly from the relay nodes. The I/O stations are connected to the two nearest RNs by two-way ultrahigh frequency (UHF) line-of-sight radio links. Finally, two subsystems support GWEN operation: the TRANSEC (Transmission Security) Rekey Station (TRS) and the GWEN Maintenance Notification Center (GMNC). Both subsystems are connected to the network through an I/O station at Headquarters, Strategic Air Command. The TRS supports network security while the GMNC monitors network operations, security, and maintenance status.

The Thin-Line Connectivity Capability is comprised of 56 relay nodes, eight Input/Output stations, 30 receive-only stations, the GWEN Maintenance Notification Center and the TRANSEC Rekey Station. For the Final Operational Capability (FOC), it is envisioned that there will be an increase in the total number of RNs and I/O and RO stations which will enhance the network's survivability and connectivity to all strategic forces. General Electric/RCA (formerly RCA) Government Communications Systems is the prime contractor for the TLCC phase of GWEN. Portions of the software were subcontracted to TRW. MITRE is the general system and Software Quality Assurance engineer supporting the Electronic Systems Division, SZG in the acquisition of GWEN.

The GWEN network components are presented in figure 1. As can be seen, the relay nodes are the backbone of the GWEN packet network. The message injection/reception and control stations comprise the message network. The GWEN components are further described below.

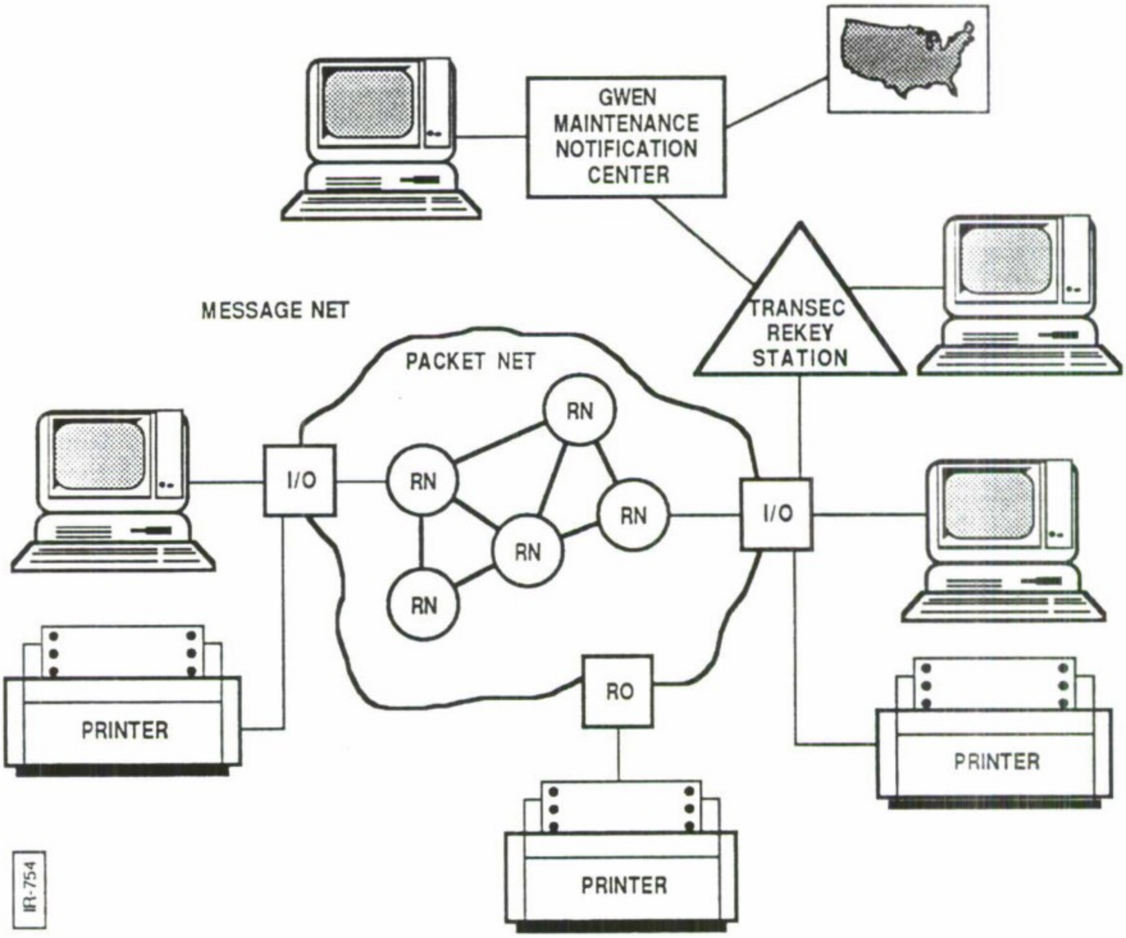


Figure 1. Network Components

1.2.1 Relay Nodes

In TLCC there are two types of relay nodes: nonaccess RNs and access RNs. Both types of relay nodes are located at unmanned, fixed sites. Nonaccess RNs receive packets from other RNs via LF radio links and in turn rebroadcast packets at LF to other RNs and to RO stations located within their areas of propagation. The LF portion of the relay nodes operates in a half-duplex mode; i.e., the LF receivers are blocked when the LF transmitters are active. LF signals transmitted from the RNs are broadcast at 2 kilowatts effective radiated power. They propagate using the groundwave, which is dependent on ground conductivity, and are potentially receivable by any correctly tuned receiver within the propagation distance. Access RNs have the additional capability to receive messages from I/O stations via UHF radio access links, to relay packets via LF radio links to other RNs, and to then deliver the messages to the I/O stations via UHF radio links. The UHF radio links operate in a full-duplex mode independent of the LF half-duplex mode of operation.

1.2.2 Input/Output Stations

At I/O stations, authorized commanders can initiate critical messages for transmission to other I/O and RO stations. Warning messages from sensors are also originated at the I/O stations. At the I/O, Network Control (NC) software converts the message into packets, determines the communications protocol, and TRANSEC-encrypts each packet. User data traffic injected at the GWEN I/O station can consist of messages addressed to a single I/O (point-to-point), to a group of I/Os (point-to-multipoint) or to all I/Os and ROs (broadcast). The point-to-point/multipoint traffic among I/Os is carried on virtual circuits employing hop-by-hop and end-to-end acknowledgments along paths automatically established by the Network Control software. Messages using the virtual circuits can be of varying lengths up to a maximum length of 880 characters (16 packets). The network transports these messages in fixed-length "packets" of 644 bits. Broadcast messages consist of a single packet only.

The critical messages can originate from sensor systems, from processors of other systems connected to GWEN via the external interface ports on the GWEN terminal at an I/O, or from manual inputs on keyboards of the GWEN system. Messages received by the I/O stations are delivered to these processors or are printed out for use by the GWEN operator. Conversational messages may also be originated at the I/O stations' keyboards.

The data portion of each packet is COMSEC-encrypted at the source I/O with a KG-84A and is not decrypted until it reaches its destination. Each packet is also TRANSEC-encrypted by every I/O station and relay node before it is transmitted/retransmitted.

1.2.3 Receive-Only Stations

The RO station receives I/O station-originated messages from the RNs over multiple LF links. The messages are printed out for use by the GWEN RO operator. ROs receive broadcast messages only.

1.2.4 GWEN Maintenance Notification Center (GMNC)

The GMNC is a subsystem that supports GWEN operation. It is dependent upon its colocated GWEN I/O station at HQ SAC for its message interface with the GWEN network. The GMNC originates automated and manual messages to monitor network performance. It also receives information from the stations and nodes on the status of security and physical conditions, subsystem components, and maintenance actions. The GMNC is able to generate statistical maintenance data and to monitor GWEN maintenance actions.

1.2.5 TRANSEC Rekey Station (TRS)

The TRS is also a support subsystem which is colocated at the HQ SAC I/O station. The TRS is used for both local and remote "rekeying" of TRANSEC encryption equipment at the GWEN nodes. The TRS software generates and maintains cryptovariables used by the GWEN system, and delivers them to the nodes and stations either by transmitting them over GWEN via the colocated I/O station (remote rekeying) or by placing the cryptovariables in KYX-15 devices for maintainer personnel to introduce into the nodes (manual rekeying). Besides rekey operations, the TRS supports reload operations. The TRS provides a method whereby selected Network Control reloadable site initialization parameters may be modified remotely over the network, eliminating the need to dispatch a maintainer to the site.

1.2.6 Maintenance Terminal

The Maintenance Terminal (MT) is support equipment used by maintenance personnel to initialize and synchronize each GWEN station when bringing them on-line. The MT consists of an HP-85B small personal computer that uses tape cartridges which contain site-specific data. The software for the MT was developed in BASIC and was regarded as a computer program component of the Built-in Test Equipment (BITE) program.

1.3 GWEN SOFTWARE

The GWEN software has a top-down, modular design. The system software is currently written in Assembly language (65 percent), FORTRAN (30 percent), and BASIC (5 percent). Two higher-order languages, FORTRAN and JOVIAL, were specified by the Government as acceptable for use as the primary application software language for GWEN. Of the developed software, only that developed in FORTRAN is compliant with the Government's approved higher-order languages. RCA's use of Assembly was driven by coding requirements to support binary operations, I/O instructions, interrupt control, register save/restore operations and other similar tasks which were not conducive to FORTRAN coding practices.

The GWEN software is comprised of 11 major computer programs, called Computer Program Configuration Items (CPCIs), that support the five network components. The following eight Computer Program Configuration Items support the relay nodes, I/O and RO stations:

- COMSEC Interface Processor (CIP)
- TRANSEC Control Processor (TCP)
- TRANSEC Front End (TFE)
- TRANSEC Variable Processor (TVP)
- Human-Machine Interface (HMI)
- Network Control Processor (NCP)
- Built-In Test Equipment (BITE)
- Maintainer Terminal (MT)

The remaining three CPCIs support the TRS and GMNC:

- Rekey Crypto Variable Controller (RCVC)
- Rekey Operator-Machine Interface (ROMI)
- GWEN Maintenance Notification Center (GMNC)

RCA subcontracted the development of the Human-Machine Interface and the Network Control CPCIs to TRW's Redondo Beach, California, facility; the GWEN Maintenance Notification Center CPCI was subcontracted to the TRW Colorado Springs, Colorado facility.

The GWEN software is implemented throughout the network in a distributed microcomputer arrangement. The software that executes all functions required by the RNs, I/Os, and ROs resides at the nodes and stations in

EMM/SESCO's SECS family of microcomputers. For the Maintainer Terminal CPCI, the software resides in an HP-85B device. The microcomputers are housed in RCA-designed chassis (drawers) within each node or station's equipment rack. Software for the GMNC subsystem, colocated at the SAC I/O station, runs on a VAX 11/730. The TRS subsystem, also colocated at SAC, runs on the EMM/SESCO SECS family of microcomputers.

Table 1 presents the function, locations, type of processor, and software developer for each of the 11 CPCIs. Also given are preliminary sizing statistics, including memory capacities and lines of source code written in FORTRAN (F), BASIC (B) and Assembly (A). Figure 2, the Computer Software Specification Tree, identifies the development and product specifications against which the individual CPCIs were evaluated, and illustrates their relationships to higher-level and adjacent specifications.

The MITRE SQA effort, centering around the 11 CPCIs, included extensive "desktop" review of the software and software documentation, as well as testing, simulation and execution of the major portions of the software during its development. The next section discusses MITRE's Software Quality Assurance task and the establishment of the MITRE SQA facility.

Table 1. GWEN CPCI Preliminary Statistics

CPCI	Function	Location	Processor	Memory	Lines of Code	Developer
HMI	HMI at I/O Station	I/O	SECS 86/10	128K EPROM 256K RAM	2,950F 3,025A	TRW
CIP	COMSEC Interface	I/O, RO	SECS 80/544	16K EPROM 16K RAM	3,125A	RCA
NC	Network Protocols	I/O, RO, RN	SECS 86/10	128K EPROM 256K RAM	2,900F 3,025A	TRW/RCA
TCP	TRANSEC Control	I/O, RO, RN	SECS 88/10	32K EPROM 4K RAM	1,580A	RCA
TVP	TRANSEC Variable Maintenance	I/O, RO, RN	SECS 88/10	32K EPROM 4K RAM	1,700A	RCA
TFE	LF/UHF Comm I/F	I/O, RO, RN	SECS 80/544	16K EPROM 16K RAM	1,470A	RCA
BITE	Status Monitoring & Reporting	I/O, RO, RN	SECS 80/10A	32K EPROM 4K RAM	3,000A	RCA
MT	Terminal Initialize & Synch	I/O, RO, RN	HP-85B	128K RAM	2,400B	RCA
GMNC	Status Display & Reporting	GMNC	VAX 11/730	2MB RAM	4,000F 8,060A	TRW
ROMI	Rekey Operator-Machine Interface	TRS	SECS 86/10	128K EPROM 256K RAM	2,000F 2,000A	RCA
RCVC	Rekey Crypto Variable Controller	TRS	SECS 86/10	128K EPROM 256K RAM	1,065F 1,130A	RCA

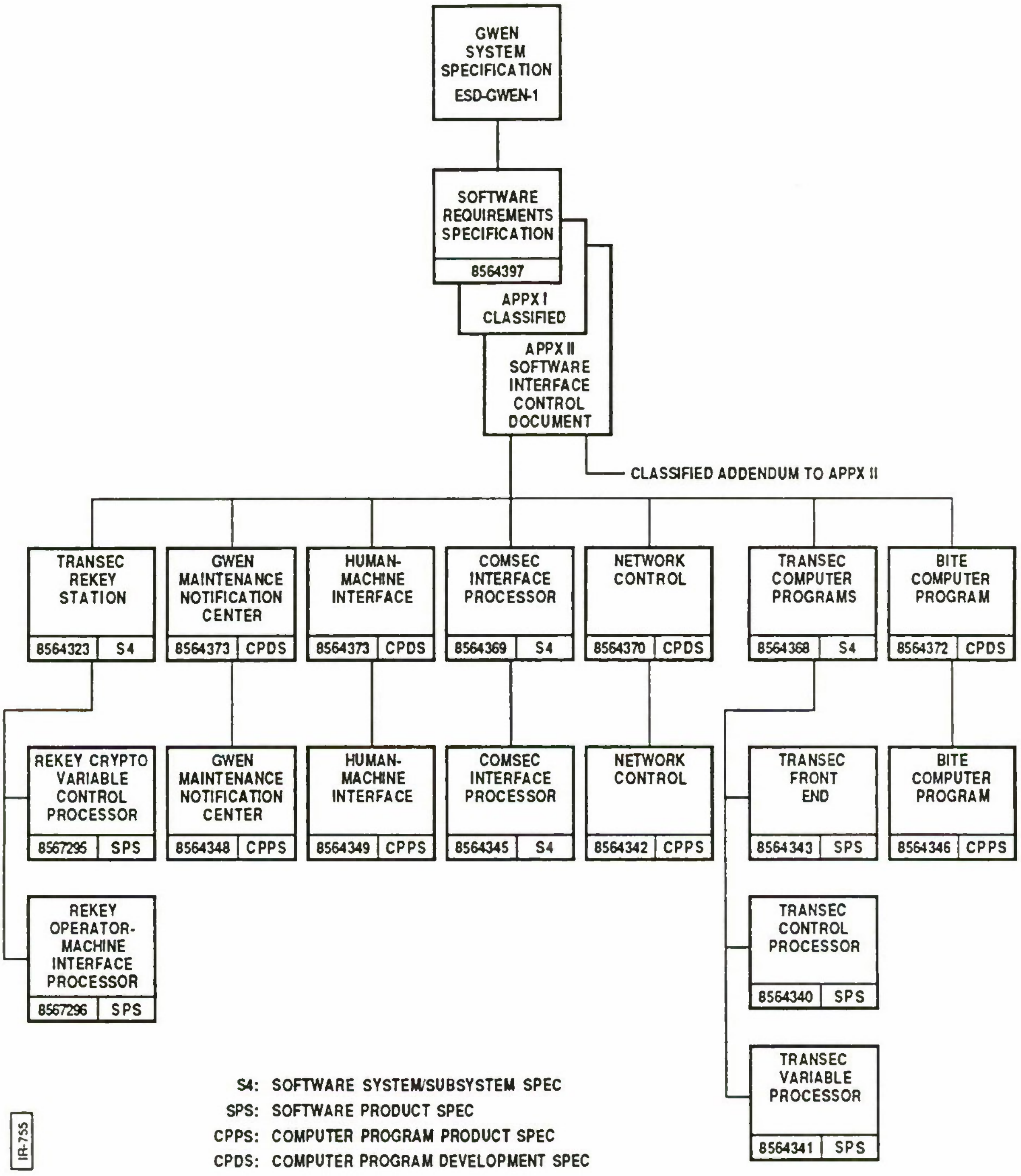


Figure 2. GWEN Computer Software Specification Tree

SECTION 2

SOFTWARE QUALITY ASSURANCE

2.1 MITRE SOFTWARE QUALITY ASSURANCE TASK

The GWEN Software Quality Assurance (SQA) effort was initiated at MITRE in May 1984 to provide verification and validation of the GWEN computer software. The definition of the MITRE Quality Assurance task was to:

- Provide extensive review of software documentation, document problem areas, identify solutions and track their resolution;
- Perform in-depth testing of the RCA/TRW software;
- Conduct an extensive review of mission-critical software algorithms;
- Attend and/or observe RCA/TRW technical reviews, unit testing, hardware/software integration, CPCI and system-level formal qualification testing.

In order to accomplish this task, MITRE would conduct, in the following sequence,

- A detailed review of the GWEN CPCI development and product specifications;
- An examination of software listings and Program Design Language (PDL);

- An evaluation of the contractor's COMSEC, TRANSEC, and physical security design;

- Individual line-by-line execution of the software;

- Functional testing of the code against the requirements;

- Functional testing of the code implementation;

- CPCI software testing on hardware.

The actual preparations for the SQA effort were initiated at the time the contract and SOW were generated for the Thin-Line Connectivity Capability phase of the GWEN program. Following contract award to RCA in September 1983, MITRE began to gather the necessary resources. This included the creation of a secure, "closed" area in H-Building at the MITRE/Bedford complex, and acquisition of computers which were compatible with the ones used by RCA for software development and hardware/software integration.

The MITRE SQA commitment included four contract engineers and two MITRE members of the technical staff (MTS). Contractors were assigned to the SQA facility full-time, while the MITRE engineers' involvement was in the area of 50 to 75 percent of their available time. The SQA commitment lasted several years, well beyond the effort normally expended by MITRE in monitoring software development integration and testing. The intensive hands-on effort lasted over two years. During the initial 12 months, the software development and the qualification testing of the individual CPCIs were completed. The SQA contract engineering support was then scaled down to two engineers for the next nine months and finally, one contract engineer for the remaining five months. Analysis of residual software issues identified during DT&E and IOT&E was performed by MITRE MTS.

MITRE's efforts in accomplishing the SQA tasks fell into four major areas of activity: performing desktop audits of individual CPCIs, compiling and simulating the CPCI software, establishing a nodal testbed for testing the software, and determining the maintainability of the software at the code level. Table 2 summarizes the scope of the SQA effort for each CPCI. Section 2.2 explains the verification and validation procedures that MITRE applied to its Software Quality Assurance efforts. Section 2.3 describes the test environment that was created to carry out various aspects of the tasks.

Table 2. SQA Task Efforts By CPCI

CPCI	Software Quality Assurance Task Effort			
	Desktop	Simulation	Node Test	Maintenance
NC	X	X	X	X
HMI	X	X	X	X
GMNC	X			X
BITE	X	X	X	X
MT	X		X	X
ROMI	X			X
RCVC	X			X
TCP	X	X		X
TVP	X	X		X
TFE	X	X	X	X
CIP	X	X	X	X

2.2 SOFTWARE QUALITY ASSURANCE PROCEDURES

For any contract, the function of quality assurance provides a planned and systematic set of actions designed to ensure that the system's software will conform to established requirements and standards. Software quality assurance involves numerous verification and validation (V&V) activities that both the software development and SQA contractors perform during the various stages of the system's evolution, from software design through formal qualification testing.

As a routine task in the design and development of the GWEN system, RCA followed all of the quality assurance procedures in accordance with relevant Government documents. MITRE, for its SQA effort, focused on a subset of these procedures that were relevant to the specific MITRE software validation tasks. The procedures used by MITRE included:

- Requirements List Generation - A process of extracting a list of requirements from requirement or design documents to be used as input for other SQA activities.

- Requirement Traceability Analysis - A process of verifying the traceability between two requirements or design documents by ensuring that all requirements in the lower-level documents have their basis in a higher-level document.

- Critical Algorithm Analysis - A process of evaluating selected algorithms to ensure conformance with system objectives.

- Interface Analysis - The process of evaluating the completeness and compatibility of system, subsystem, and module-level interface definitions.

- Traceability Analysis - A process of verifying that the test scenarios defined in the test procedures provide for complete and rigorous testing against the requirements.

- Database, Source Code, Program Design Language (PDL) and Development Test and Evaluation - For the database, a process of evaluating the requirements, design and physical characteristics. For the source code and PDL, a process of analysis to ensure adherence to applicable standards and coding practice, correct implementation of interfaces, the absence of abnormal return and error procedures, and efficient implementation of call sequence parameters. For development test and evaluation, a process of evaluating unit level integration, software component level integration, system level integration and operational levels of testing through the processed test plan/procedures analysis and review, test witnessing and test results evaluation.

- Design-to-Code Traceability - A process of verifying that the source code correctly implements the Computer Program Configuration Item (CPCI) as described in the requirements and detailed design documents.

- Formal Review and Audit Support - The overall V&V process of supporting formal reviews and audits of the CPCIs through the application of military standards, Department of Defense and Military Department regulations, and through the requirements of the Statement of Work.

- Document Discrepancy Report Preparation - A process of recording problems discovered during analysis of a document, identifying proposed solutions and tracking the disposition of the problem.

2.3 SOFTWARE QUALITY ASSURANCE TEST ENVIRONMENT

The MITRE SQA facility was configured and equipped to examine software execution at the module level, analyze critical timing constraints and determine compliance with the Government's developmental requirements. To this end, MITRE acquired two Hewlett-Packard (HP) microprocessor development systems (MDSs) with a full suite of 8080/8085, 8086, and 8088 emulators, a Tektronix 8550 microprocessor development system with a limited suite of 8080/8085 emulators, and a Digital Equipment Corporation VAX 11/730 computer system, First System (FS) software construction tools and Boston Systems Office (BSO) software development tools. Hardware required to build a three-chassis (TRANSEC/NCP, TIP I/O and BITE) node testbed was also acquired.

Ancillary support equipment for the facility included an HP-1631D logic analyzer, an HP multichannel internal software analyzer, an Analogic DATA 6000 digital storage scope, a Data I/O 29A Universal PROM programmer and several IBM PC and AT computers. The SQA facility allowed detailed examination of the RCA/TRW software operating in a simulated and in a GWEN hardware environment. Figure 3 provides a pictorial representation of the SQA process in relation to the associated MITRE task efforts.

2.3.1 Microprocessor Development Systems

Three microprocessor development systems were employed to provide complete software analysis of the CPCI under examination. By using the three MDS workstations within a basic configuration, MITRE engineers were able to monitor the individual processing of three CPCIs concurrently, as well as their node interaction with other CPCIs.

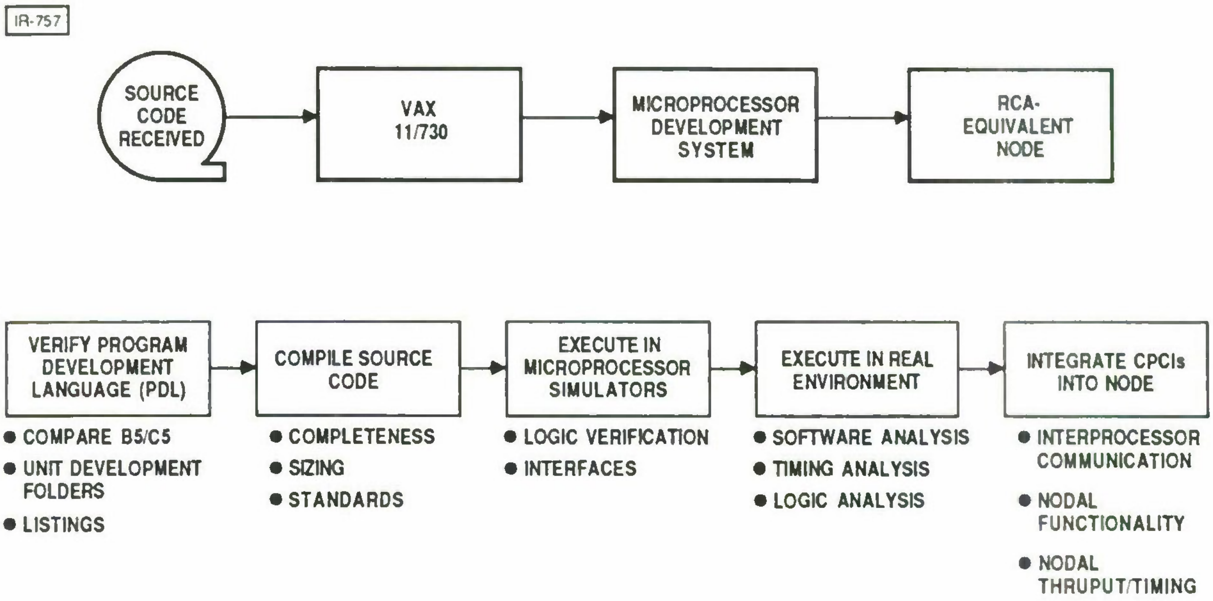


Figure 3. SQA Process/Task Relationship

The following items comprised the HP 64000 MDS software testing equipment provided in the GWEN SQA facility:

- HP 64000 MDS workstation with keyboard, 12-slot card cage, CRT and dual floppy disk drives;
- HP 8086 emulation subsystem which includes emulator controller, 8086 pod, 128K memory board, analysis board;
- HP 8088 emulation subsystem which includes emulator controller, 8088 pod, 64K memory board, analysis board;
- HP 8085/8080 emulation subsystem which includes emulator controller, 8085 and 8080 pods, analysis board;

- HP Software State Analyzer which provides the details of the software execution (e.g., timing, bus interaction, memory usage, CPU states, and interaction);

- HP 15-megabyte Winchester hard disk;

- Line printer.

The following items, which are similar in nature to the HP 64000 MDS, comprised the Tektronix 8550 MDS provided in the GWEN SQA facility:

- TEK 8500 MDS workstation with keyboard, a model 8301 microprocessor development unit with 32K of emulation memory;

- TEK 8501 data management unit;

- TEK 8080 and 8085 emulation control pods.

2.3.2 VAX 11/730 Computer System

The final major portion of the GWEN SQA facility was the Digital Equipment Corporation VAX 11/730 computer system, which consists of three megabytes of on-line RAM memory, a 121-megabyte fixed hard disk, a 10.4-megabyte removable hard disk, a nine-track tape drive, six video display terminals, one printer, application software which supported software creation from source code to executable format, and software for debugging and simulation of the GWEN CPCI processors. Additional information on the VAX-installed software applications is presented in section 3.

The interconnectivity of the microprocessor development systems with the VAX 11/730 computer system is presented in figure 4.

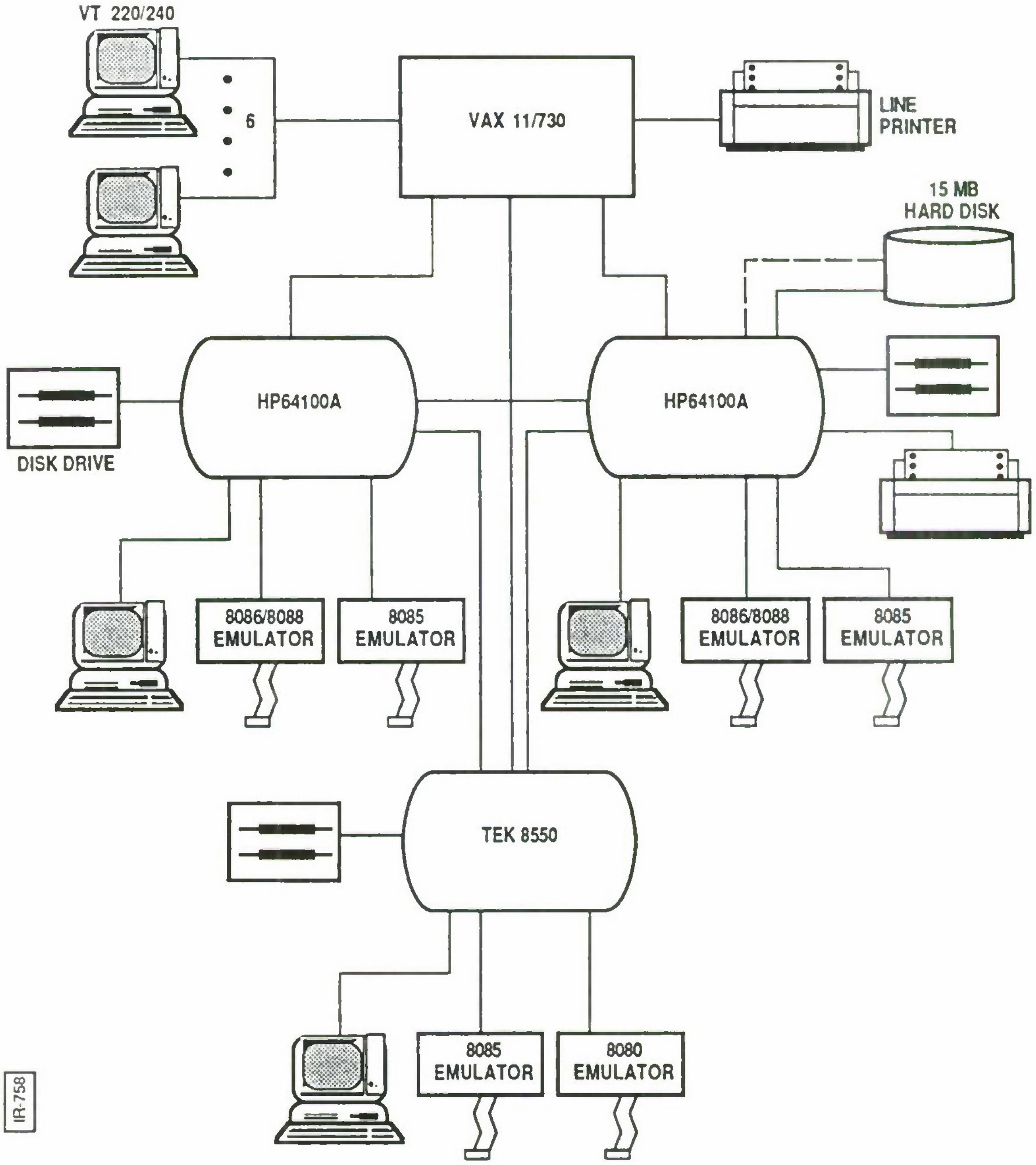


Figure 4. MITRE Software Quality Assurance Test Environment

2.3.3 Software Quality Assurance Node Testbed

Once the SQA microprocessor development systems and VAX facility were established, the next stage was to create a node testbed. Several alternative node configurations were reviewed. Based on the complexity of the software, the operator interaction and the network processing requirements, MITRE designed and configured three Intel-based chassis to be functionally identical to an RCA I/O station. The decision to build functionally identical units rather than using actual RCA chassis was driven by the non-availability of RCA engineering chassis to support the SQA effort. With the exception of the LF transmit/receive functions of the RN, the UHF transmit/receive functions of the I/O, and the GWEN Maintenance Notification Center and TRANSEC Rekey Station support subsystems, the entire station architecture was incorporated into the MITRE SQA node design (see figure 5).

The three chassis (TRANSEC/NCP, TIP I/O and BITE) were designed to be functionally equivalent to the RCA hardware. The development of the testbed provided MITRE with the capability for in-depth analysis of the COMSEC (Communication Security) interfaces, Network Control software, TRANSEC hardware/software and critical timing requirements of the GWEN relay nodes. The three chassis configurations, including the Intel boards and the corresponding RCA hardware boards, are presented in tables 3, 4, and 5 for the TRANSEC/NCP, TIP I/O, and BITE chassis, respectively. Figure 6 presents the chassis interfaces within the MITRE testbed node.

2.3.4 Software Quality Assurance Software Simulation

In two functional areas, TRANSEC and COMSEC Interface Logic, there were no commercial equivalents for the boards being manufactured by RCA to military specifications. Accordingly, to simulate the TRANSEC Control and Variable Processors and TRANSEC-KG functions, and the COMSEC Interface

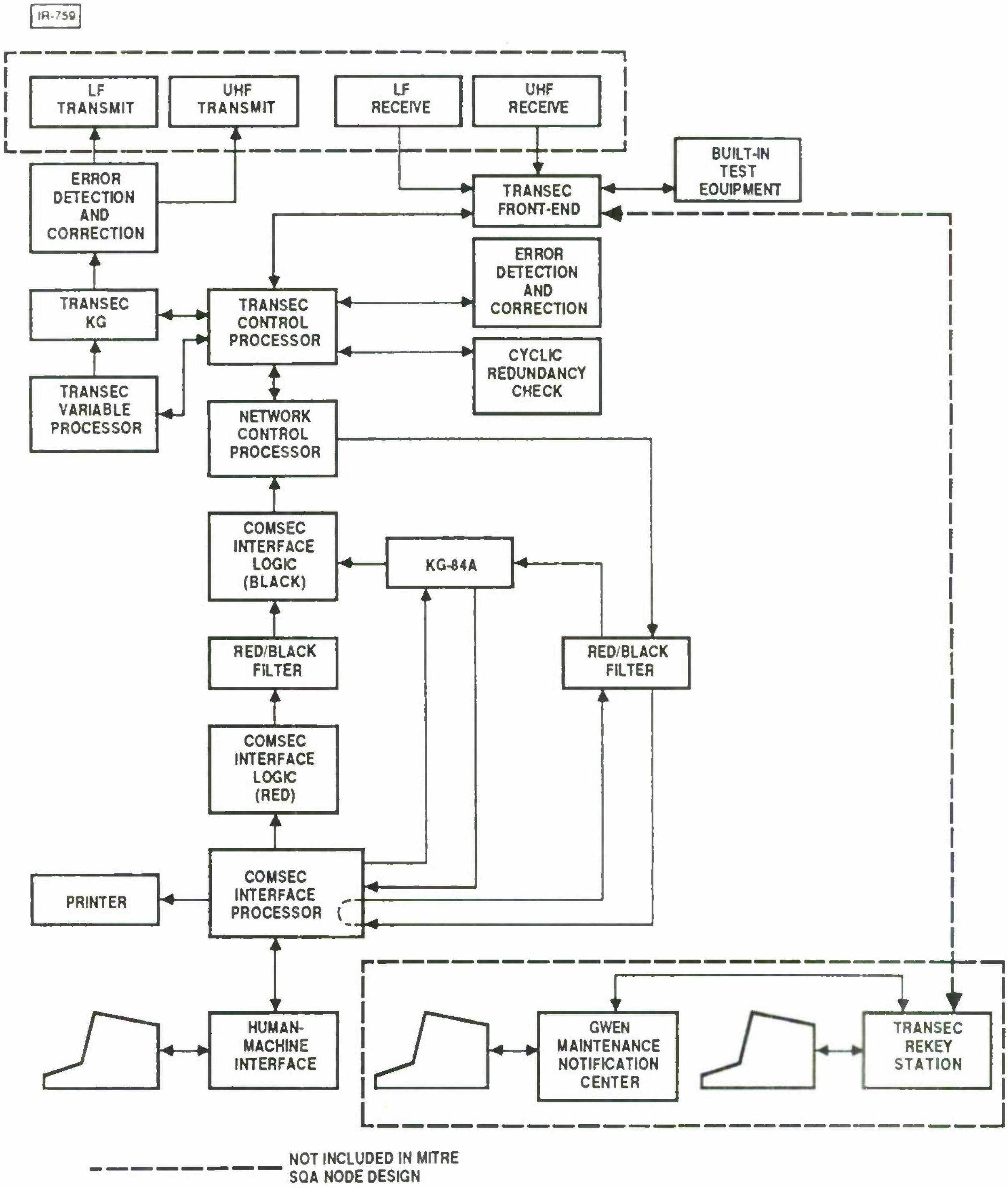


Figure 5. GWEN Station Architecture

Table 3. Chassis Construction: TRANSEC/NCP

Slot#	Description	RCA Board	MITRE Intel Board
1	-	-	-
2	EDAC	E-FXA	iSBC 88/25* w/iSBC 302
3	TVP	E-FXB	
4	TCP (MEMORY)	E-FXC	
5	TCP (KG)	E-FXD	
6	TCP (CPU)	E-FXE	
7	R/B FILTER	E-FXF	
8	BLACK I/O 1	E-FXG	
9	BLACK I/O 2	E-FXH	
10	TFE (CPU)	80/544	iSBC 544
11	TFE (I/O)	80/544	
12	NCP	86/10	iSBC 86/12A
13	NCP RAM	80/056A	iSBC 056A
14	NCP ROM	80/164	iSBC 464
15	NCP ROM	80/164	iSBC 464

---

* Simulation of boards required

Table 4. Chassis Construction: TIP I/O

Slot#	Description	RCA Board	MITRE Intel Board
1	Power Supply Equip.	(RCA)	-
2	CIP (CPU)	80/544	iSBC 544
3	CIP (I/O)	80/544	
4	HMI COMM	80/534-2	iSBC 534
5	HMI COMM	(not used)	-
6	-	-	-
7	-	-	-
8	HMI (expansion)	-	-
9	HMI RAM	80/056A	iSBC 056A
10	HMI ROM	80/164	iSBC 464 (2)
11	HMI (CPU)	86/10	iSBC 86/12A
12	CIL (RED)	(RCA)	-
13	R/B FILTER	(RCA)	-
14	CIL (BLACK)	(RCA)	iSBC 544*
15	-	-	-

---

* Simulation of boards required

Table 5. Chassis Construction: BITE

Slot #	Description	RCA Board	MITRE Intel Board
1	I/O STATUS	(RCA)	-
2	CLOCK GEN	(RCA)	-
3	-	-	-
4	BITE RAM/ROM	80/164	iSBC 464,028A
5	BITE (CPU)	80/10A	iSBC 80/10B
6	-	-	-
7	UHF MODEM	(RCA)	-
8	UHF MODEM	(RCA)	-
9	I/O UHF	(RCA)	-

Logic (CIL) function, we used Intel boards having similar hardware capabilities and MITRE-developed software. The TRANSEC hardware configuration, presented in figure 7, is comprised of four processors and the TRANSEC-KG. The TRANSEC simulation software was developed utilizing Intel 8086 Assembly language, which was compatible with the RCA-developed TVP, TCP, and TKG functions. The simulation software was developed on the VAX 11/730 system using First System's 8086 assembly and the BSO simulator. It provided full transmission/reception path cyclic redundancy check (CRC), encryption/decryption, error detection and correction (EDAC) functions and interface control to and from the NC and TRANSEC Front End (TFE) hardware.

The COMSEC Interface Logic hardware configuration, shown in figure 8, included two processors and the COMSEC interface module and KG-84A cryptographic device. The simulation software was developed with Intel 8085 Assembly language due to the hardware-intensive nature of the CIL functions. This simulation was necessary because RCA was manufacturing unique

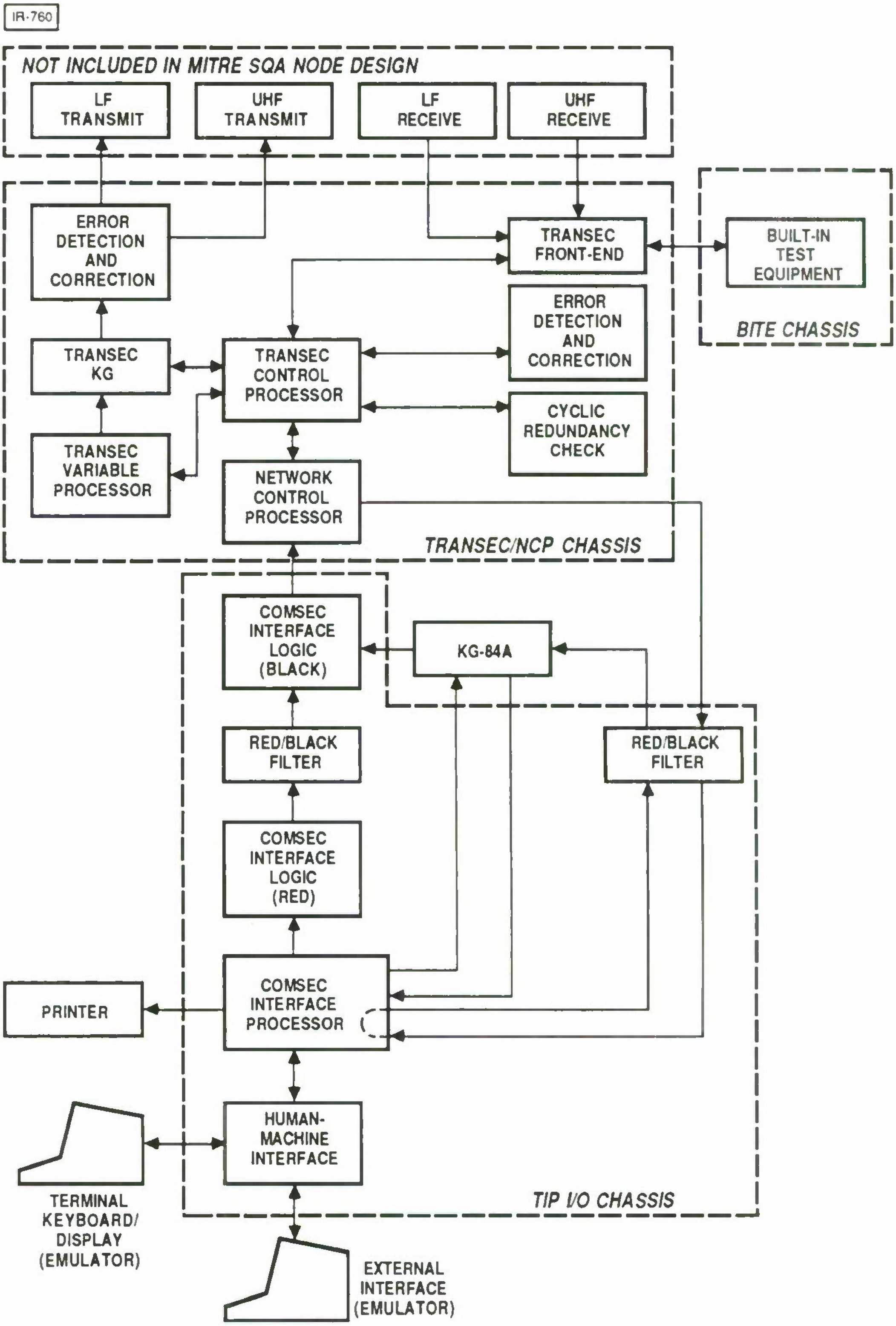


Figure 6. MITRE Node Configuration Chassis Interface Diagram

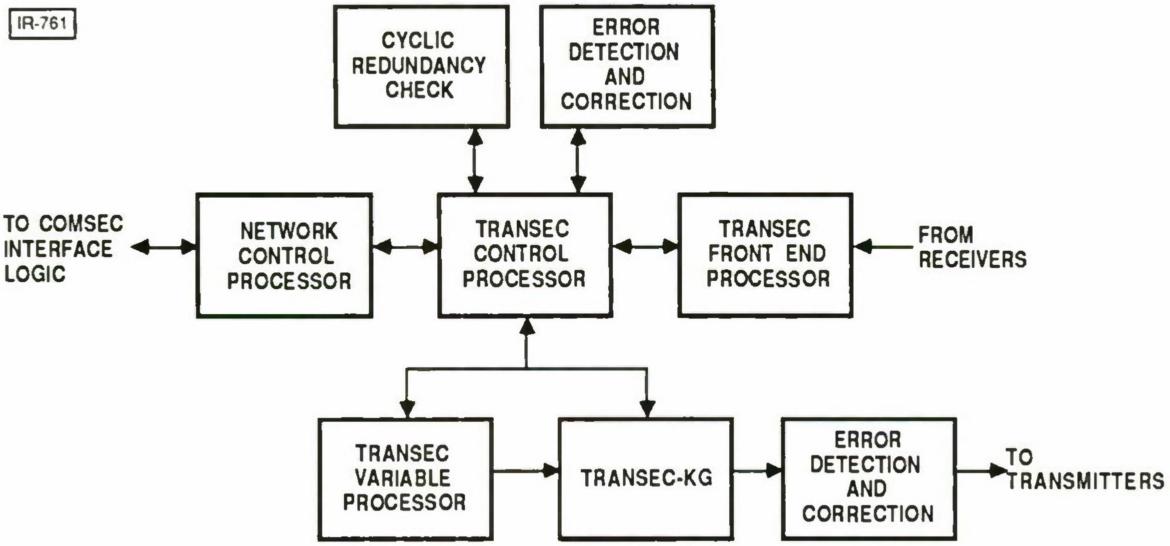


Figure 7. TRANSEC Configuration

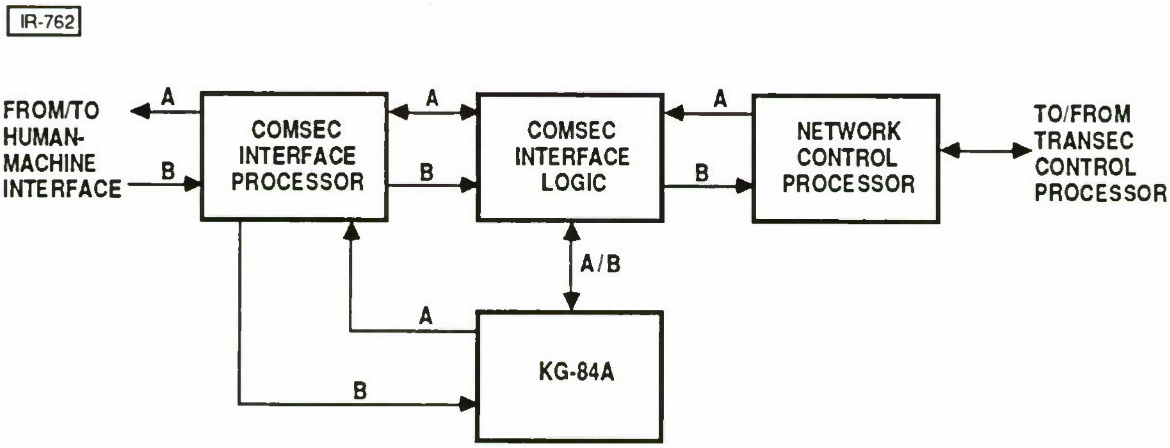


Figure 8. CIL System Configuration

boards to perform the CIL functions (red/black filter interface and message header bypass). Since Intel does not build similar CIL boards, MITRE used an Intel board with equivalent hardware capabilities and wrote a software simulation of the RCA CIL board.

The COMSEC Interface Logic acts as the communications interface between the COMSEC Interface Processor (CIP), the KG-84A cryptographic device and the Network Control Processor. Figure 8 illustrates the CIL system configuration processing flow. The data flow from the Network Control Processor to the Human-Machine Interface is reflected in the "A" data path. The reverse flow, Human-Machine Interface to the Network Control Processor, is reflected in the "B" data path.

In addition to the preceding functional areas, the HMI's Alpha/Graphic Plasma Display Terminal (PD-3500) was not available from the contractor to support the SQA effort. Software was developed to emulate, on an IBM PC, the PD-3500 display terminal and its interface to the HMI CPCI. The emulation software (System Resource Program (SRP)) provided a real-time operating shell which, when established in the PC-DOS environment, allowed execution of the HMI's menu displays, message origination/reception and off-line test data analysis. The SRP operating system contains a number of user-callable routines that perform functions normally associated with an operating system. These routines fall into six basic functional blocks: initialization, I/O processing, interrupt processing, semaphore event flag processing, time processing and system exit. In addition to the user-callable routines, there are a number of routines used by the SRP for task control and interrupt processing, which, when used together with DOS routines, make a complete environment for the execution of HMI's display terminal applications programs. Utilizing the SRP software functions, MITRE subsequently developed the External (System) Network Simulator Device Driver and the BITE Device Drivers. Both of these simulation driver programs made use of the real-time processing application derived from the SRP software.

Figure 9 summarizes the progression of tasks in the SQA facility. In order to compensate for delays in receiving the contractor-unique hardware used in the COMSEC Interface Logic, the TVP and the TCP, we developed the CIL simulation and investigated techniques for simulating operation of the TCP and TVP. The individual CPCIs were downloaded from the VAX onto the respective boards and combined (see figure) to form the BITE, TRANSEC/NCP and TIP drawers. The assemblage of the three drawers formed the MITRE node. This node was exercised to verify the correct performance of inter-CPCI interfaces and to demonstrate various nodal requirements. These tasks were carried out simultaneously with RCA's effort, represented in the upper portion of the figure. Our evaluations and results are the subject of section 3.

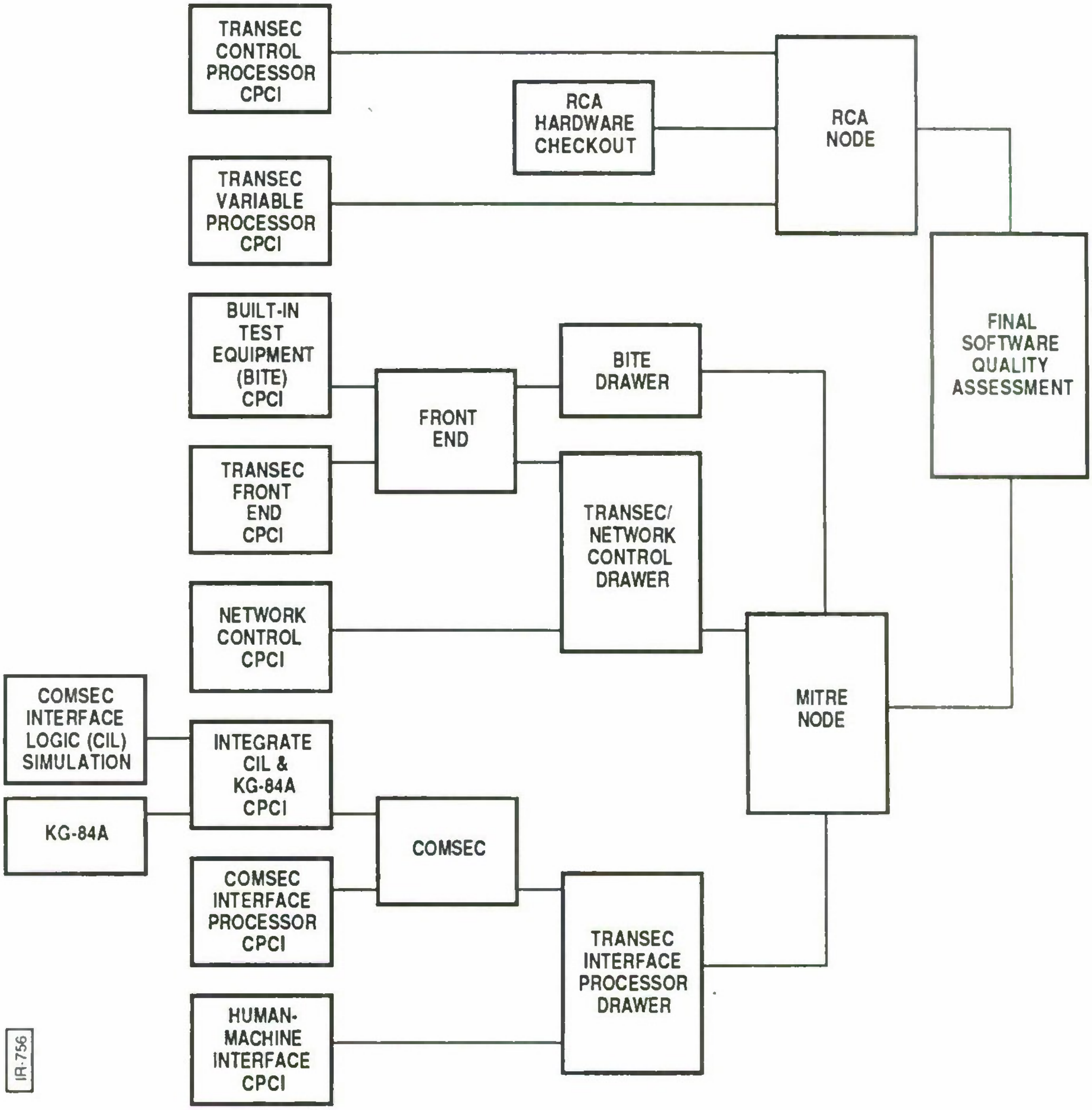


Figure 9. GWEN Software Quality Assurance Tasks

SECTION 3

SOFTWARE ANALYSIS METHODOLOGIES AND SUMMARY OF FINDINGS

3.1 SOFTWARE EVALUATION AND FINDINGS

The evaluation of the GWEN software for the SQA effort was divided into the following activities:

- Preliminary Statistical Analysis of Software Trouble Reports (STRs);
- Desktop Audits of Specifications, Source Code, Program Design Language (PDL), and Test Procedure Audits;
- VAX and Software Simulation Test Environment;
- MITRE Node Test Environment;
- Software Code Maintainability Analysis.

The level of the SQA effort completed for each individual CPCI was proportional to the RCA and TRW software development effort, the availability of software for the SQA effort, the level of risk each CPCI represented to the overall GWEN system, and the availability of SQA assets. As a minimum, all CPCIs received STR statistical analyses, desktop audits, and software code maintainability analyses. Due to their size, selected CPCIs (i.e., HMI and NC) required dedication of the VAX 11/730 in order to build and run them in simulation. Smaller CPCIs were built and run in simulation concurrently.

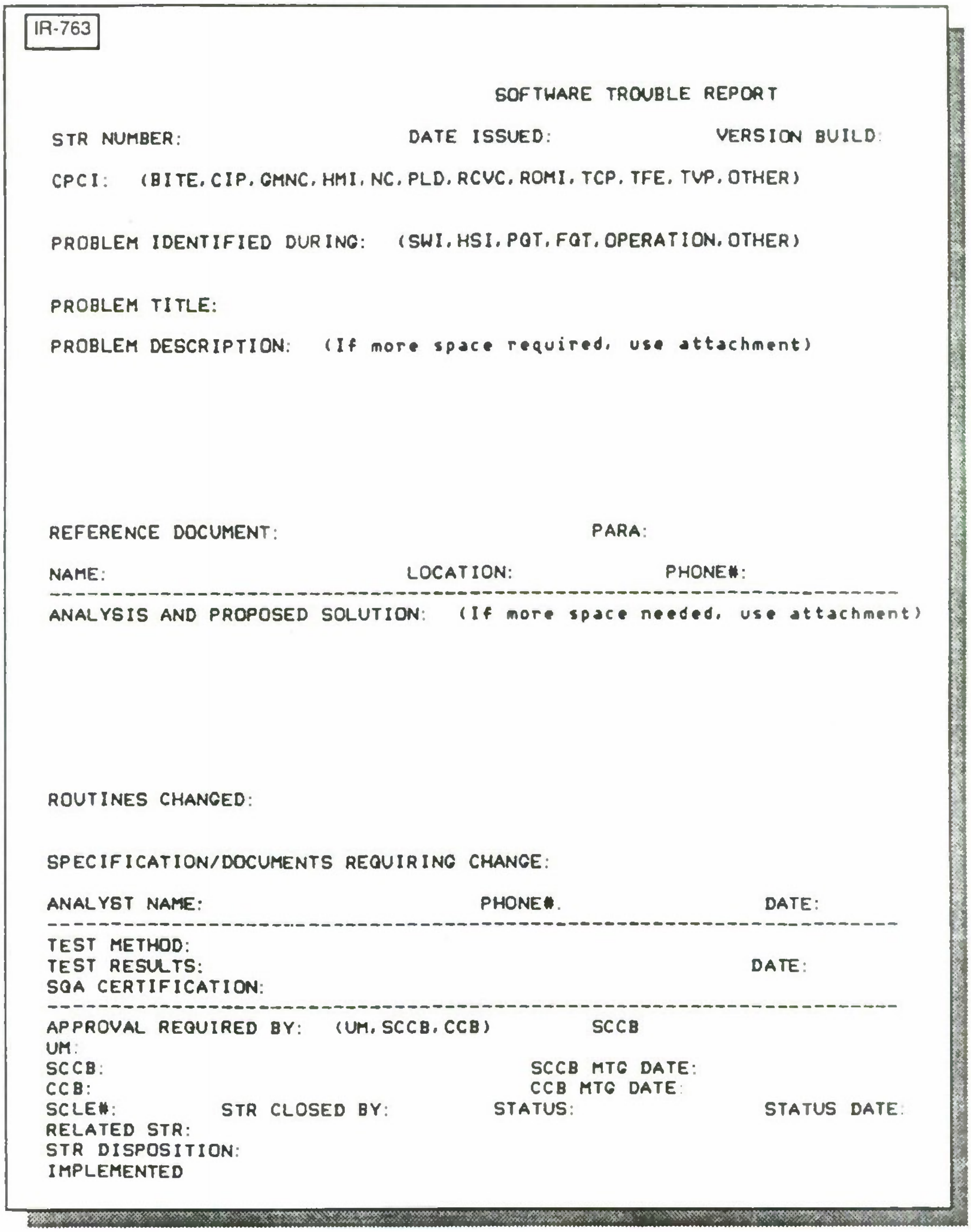
The individual SQA activities, with a summary of major findings, are presented in the following paragraphs. Detailed discussions and findings pertaining to the analysis of the individual CPCIs are presented as appendices to this report (published separately). To assist the reader, the appendix topics are listed below.

Appendix A	Computer Program Configuration Item Check List
Appendix B	COMSEC Interface Processor (CIP) CPCI
Appendix C	TRANSEC Processor (TCP, TFE, TVP) CPCIs
Appendix D	TRANSEC Rekey Station (ROMI, RCVC) CPCIs
Appendix E	Human Machine Interface (HMI) CPCI
Appendix F	Network Control Processor (NCP) CPCI
Appendix G	Built In Test Equipment (BITE)/Maintenance Terminal (MT) CPCI
Appendix H	GWEN Maintenance Notification Center (GMNC) CPCI

3.2 SOFTWARE TROUBLE REPORTS

3.2.1 Activities

The evaluation and analysis of RCA's Software Trouble Reports (STRs) was accomplished through detailed technical review of all STRs generated by RCA against GWEN software. The configuration control procedures for the processing of the STRs was additionally reviewed to determine the Government's role and responsibility in the resolution of the open issues. All STRs were maintained in a central library to provide a comprehensive baseline from which the statistical analysis was derived. Figure 10 shows a Software Trouble Report data sheet.



IR-763

SOFTWARE TROUBLE REPORT

STR NUMBER: DATE ISSUED: VERSION BUILD:

CPCI: (BITE, CIP, GMNC, HMI, NC, PLD, RCVC, ROMI, TCP, TFE, TVP, OTHER)

PROBLEM IDENTIFIED DURING: (SWI, HSI, PQT, FQT, OPERATION, OTHER)

PROBLEM TITLE:

PROBLEM DESCRIPTION: (If more space required, use attachment)

REFERENCE DOCUMENT: PARA:

NAME: LOCATION: PHONE#:

ANALYSIS AND PROPOSED SOLUTION: (If more space needed, use attachment)

ROUTINES CHANGED:

SPECIFICATION/DOCUMENTS REQUIRING CHANGE:

ANALYST NAME: PHONE#. DATE:

TEST METHOD:
TEST RESULTS: DATE:
SQA CERTIFICATION:

APPROVAL REQUIRED BY: (UM, SCCB, CCB) SCCB
UM:
SCCB: SCCB MTG DATE:
CCB: CCB MTG DATE:
SCLE#: STR CLOSED BY: STATUS: STATUS DATE:
RELATED STR:
STR DISPOSITION:
IMPLEMENTED

Figure 10. Software Trouble Report Data Sheet

3.2.2 Summary of Findings

The Software Trouble Report statistical analysis was completed as an integral part of the individual CPCI analysis. However, for purposes of this report, our findings and observations resulting from the STR analysis are presented as preparatory information to the CPCI assessments.

As of 20 December 1987, a total of 998 Software Trouble Reports had been generated by the contractor. The configuration control procedures employed by the contractor for processing STRs is documented in the GWEN Software Configuration Management Plan, COMSEC/TRANSEC, Revision 3, dated 25 January 1985, which was approved by the Government on 19 February 1985.

During our analysis of the STR configuration control procedures, we note that the Government was not a member of the Software Configuration Control Board (SCCB). The SCCB was delegated the responsibility for the evaluation and disposition of all proposed changes to approved software development specifications and to all software code which had been approved for software integration. As such, it should be stated that the final solutions which supported the closure of an STR represented only the contractor's position or objectives and did not reflect the Government's position, nor its approval or disapproval of the STR resolution. We further noted that the majority of the STRs were initiated upon the contractor's analysis of test results following testing at the module level, the component (CPM, CPC) level, the CPCI integration level, the Government Formal Qualification Test (FQT) level, and the GWEN hardware/software integration level. Nevertheless, we found that the STR methodology employed by the contractor provided the necessary configuration control and traceability of corrective actions for the software problems and discrepancies that were identified.

From January 1985 to January 1988, the number of Software Trouble Reports had a constant growth. Rather than try to analyze all the STRs, an intermediate software development period, April 1986 to October 1987, was selected for the purpose of this evaluation. This period coincides with the peak software development and testing phase. During the period of evaluation, a total of 625 Software Trouble Reports were originated. A breakdown of these STRs by status is shown in table 6.

By graphing the total number of STRs versus the open STRs over the period April 1986 to October 1987 (figure 11), the trend can be seen. The obvious trend is the increase in the total number of STRs; more importantly, however, is the almost constant value for the "open" STRs at any one time. While the number of open STRs remained around or below 50, the "approved" STRs were consistently at a level of approximately 100 STRs above the open STRs. This indicated that a conscious effort was being made by the software developer to keep the open STRs to a minimum and to take swift action on new STRs. It should be noted that SCCB approval of STRs indicates that the software fix had been implemented in the development versions of the software. For formal closure of these approved STRs by the SCCB, a satisfactory response from the hardware/software integration testing engineers was required.

The spread of the STRs across the CPCIs was also considered. Table 7 presents STR status data for the individual CPCIs as of 1 October 1987. As can be derived from this data and seen in figure 12, the CPCI having the greatest number of STRs (410) is the Network Control (NC). This high concentration was expected since the NC is the most complex CPCI. It interacts with all other CPCIs and possesses the greatest number of interfaces. Additionally, the TRW-developed NC virtual circuit (VC) software design was completely reworked and rewritten by RCA following the TRW delivery. Four other CPCIs, all having 50 or more STRs generated against them, were the HMI, TCP, CIP and TFE with 87, 69, 59 and 51 STRs, respectively. In the case of these four CPCIs, the STRs were generated mainly because of discrepancies identified during the hardware/software integration testing. The remaining six CPCIs accounted for less than 16 percent of all STRs

Table 6. Software Trouble Reports
(April 1986 to October 1987)

Date	Closed +	Open +	Approved +	Rejected +	Withdrawn =	Total
04/29/86	214	38	54	31	18	355
05/01/86	214	41	57	31	18	361
05/13/86	231	40	44	31	20	366
05/21/86	233	42	52	31	19	377
08/27/88	387	54	62	31	19	399
09/10/86	390	23	27	33	47	517
09/17/86	397	26	45	35	49	552
09/24/86	400	26	53	35	51	565
10/08/86	404	27	72	35	52	590
10/16/86	404	33	85	35	52	609
10/22/86	405	32	89	35	53	614
10/29/86	405	40	105	35	53	638
11/05/86	405	38	120	35	53	651
11/12/86	418	35	124	35	54	666
11/14/88	423	34	128	35	55	675
11/19/86	426	36	131	35	55	683
11/25/86	438	35	130	35	58	696
12/03/86	441	35	165	35	58	734
12/09/86	441	30	172	35	59	737
12/10/86	455	10	165	42	65	737
12/17/86	510	9	109	42	68	738
12/24/86	527	11	115	42	68	763
01/02/87	527	10	116	42	68	763
01/08/87	529	11	116	43	70	769
01/14/87	549	10	102	43	71	775
01/21/87	559	13	93	43	73	780
01/28/87	566	15	87	43	73	784
02/04/87	571	20	87	43	73	794
02/11/87	571	23	95	43	73	805
02/18/87	571	20	104	43	76	814
02/25/87	571	20	109	43	76	819
03/02/87	571	16	123	43	81	834
03/11/87	571	14	130	43	83	841
03/18/87	606	13	96	43	87	845
04/01/87	622	22	88	43	90	865
04/16/87	635	22	81	43	90	871
04/22/87	666	21	53	43	91	874
04/29/87	668	21	54	43	92	878
05/06/87	674	21	56	43	92	886
05/13/87	680	22	56	43	92	893
06/03/87	690	25	59	43	94	911

Table 6 (Concluded)

Date	Closed	+ Open	+ Approved	+ Rejected	+ Withdrawn	= Total
06/10/87	694	25	56	43	95	913
06/17/87	695	24	62	43	95	919
06/24/87	713	25	44	43	95	920
07/08/87	721	26	41	45	95	928
07/15/87	721	24	44	45	95	929
07/29/87	729	25	43	45	95	937
08/05/87	734	26	48	45	95	948
08/12/87	737	22	54	46	97	956
08/19/87	744	20	51	46	98	959
08/26/87	752	19	51	46	98	966
10/01/87	783	12	27	52	106	980

Table 7. Software Trouble Report Breakdown by CPCI
(as of 1 October 1987)

CPCI	Closed	+ Open	+ Approved	+ Rejected	+ Withdrawn	= Total	%
BITE	16	1	0	1	1	19	1.9
CIP	51	0	1	4	3	59	6.0
GMNC	18	6	7	6	5	42	4.3
HMI	65	1	2	4	15	87	8.9
MT	35	0	0	2	3	40	4.1
NC	342	2	3	18	43	410	41.8
ROMI	6	0	1	0	0	7	0.7
RCVC	23	0	6	0	5	34	3.5
TCP	58	0	0	9	2	69	7.0
TFE	48	0	0	1	2	51	5.2
TVP	19	0	0	0	2	21	2.1
Other						141	14.4
TOTAL						980	

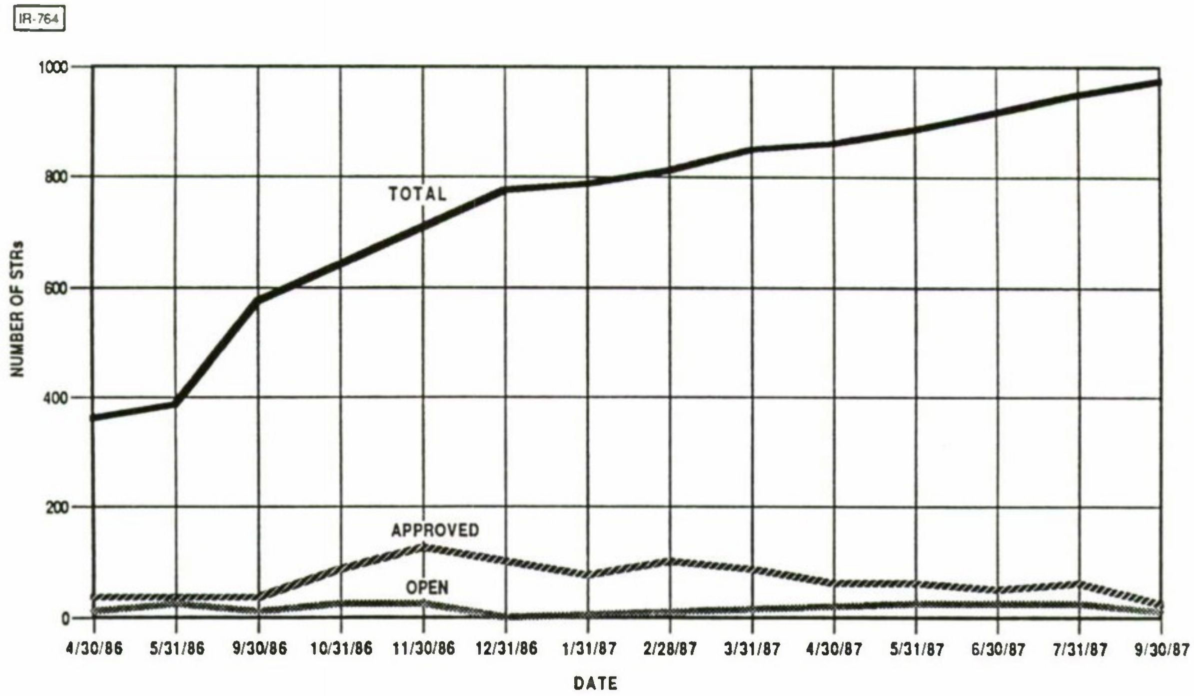


Figure 11. Software Trouble Report Statistics

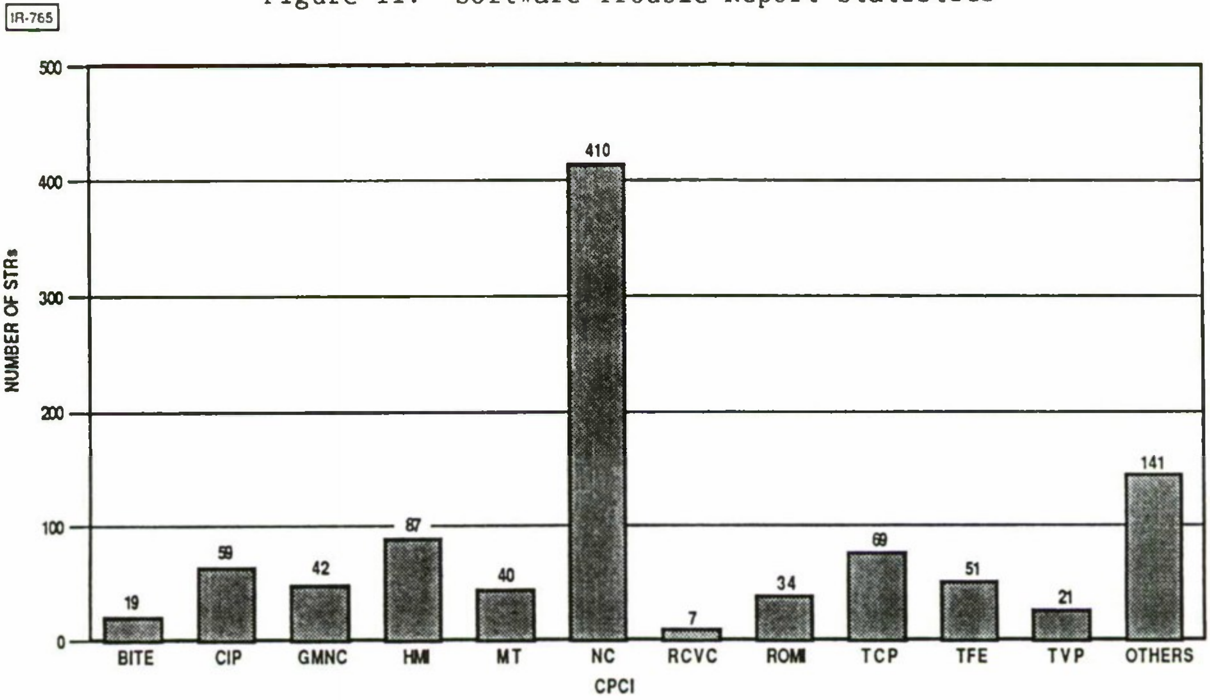


Figure 12. Software Trouble Reports by CPCI (as of 1 October 1987)

generated. The "Other" entry in table 7 refers to STRs that are related to the GWEN consolidated database necessary for correct station initialization.

Information was also gathered on the number of STRs originated on a monthly basis. This information, presented in table 8, reflects the period from January 1985 through September 1987. We note that the greatest number of new STRs were generated during the October through November 1986 period, which coincides with the redesign of the NC's virtual circuit function.

On an average, an STR was closed 73 days after it had been initiated. To obtain a greater understanding of closure time for an STR, the average number of days until closure, analyzed by quarter originated, was calculated. The average number of days to close out an STR ranged from a low of 18 to a high of 133. Table 9 provides this information.

Table 8. Software Trouble Report Monthly Origination Basis
(as of 1 October 1987)

Month	Number	Month	Number	Month	Number
Jan 85	3	Jan 86	20	Jan 87	26
Feb 85	5	Feb 86	22	Feb 87	48
Mar 85	4	Mar 86	25	Mar 87	25
Apr 85	21	Apr 86	52	Apr 87	12
May 85	31	May 86	50	May 87	30
Jun 85	56	Jun 86	44	Jun 87	23
Jul 85	21	Jul 86	36	Jul 87	16
Aug 85	13	Aug 86	31	Aug 87	23
Sep 85	33	Sep 86	49	Sep 87	16
Oct 85	15	Oct 86	72		
Nov 85	16	Nov 86	79		
Dec 85	25	Dec 86	38		

Table 9. Software Trouble Report Rate of Closure
(January 1985 - September 1987)

	Average Days Until	
Date	STR is Closed	by Quarter
Jan 1, 1985	18	
Feb 1, 1985	46	
Mar 1, 1985	115	60
Apr 1, 1985	27	
May 1, 1985	52	
Jun 1, 1985	86	55
Jul 1, 1985	100	
Aug 1, 1985	92	
Sep 1, 1985	82	91
Oct 1, 1985	118	
Nov 1, 1985	133	
Dec 1, 1985	70	107
Jan 1, 1986	85	
Feb 1, 1986	92	
Mar 1, 1986	77	85
Apr 1, 1986	79	
May 1, 1986	48	
Jun 1, 1986	77	68
Jul 1, 1986	63	
Aug 1, 1986	115	
Sep 1, 1986	113	113
Oct 1, 1986	79	
Nov 1, 1986	57	
Dec 1, 1986	38	58
Jan 1, 1987	123	
Feb 1, 1987	66	
Mar 1, 1987	87	92
Apr 1, 1987	55	
May 1, 1987	49	
Jun 1, 1987	51	52
Jul 1, 1987	37	
Aug 1, 1987	21	
Sep 1, 1987	18	25

Overall average - 73 days

3.3 DESKTOP CPCI AUDITS

3.3.1 Activities

As the first step in the SQA process, desktop audits were conducted on each of the Computer Program Components (CPCs) which comprise the individual CPCIs. The applicable documents upon which the audits were based, and which delineate the individual CPCI requirement baselines, are identified in the bibliography. The users' requirements are delineated in the individual CPCI development and product specifications: the Software Requirement Specification; the COMSEC Software System/Subsystem Specification and the Software Product Specification for COMSEC/TRANSEC-designated CPCIs; and the Computer Program Development Specification and the Computer Program Product Specification for non-COMSEC/TRANSEC-designated CPCIs. The user requirements were checked against the system-level requirements defined in ESD-GWEN-1 and the contractor's requirement qualification test matrices. The goal of cross-referencing the development requirements with the contractor's requirement qualification matrices was to provide traceability down to the qualifying test case.

The code itself, which had been loaded onto the VAX 11/730, was examined manually on a line-by-line basis and compared to the associated Program Design Language contained within the respective CPCI product specification. Checklists derived from MIL-STD-483 and ESD-TR-84-171 (Vol. III) for the Computer Program Product Specifications (CPPS), from NSAM 81-3 for the Software Product Specifications (SPS), and from ESD-GWEN-1 for the computer programming (design, construction, and sizing) requirements, were utilized to determine how well the individual CPCI product specifications and CPCIs themselves complied with the requirements of the standards and CDRL. The evaluation checklists used in reviewing the CPPS, SPS and the CPCI source code are presented in appendix A under separate cover. As part of the analysis of the COMSEC/TRANSEC CPCIs, the contractor's cryptographic principles and protective alarm software/hardware design, the TEMPEST security design, and the failure modes and related alarm notifications were assessed against the requirements of NACSIM 5100A, NACSEN 5112 and KAG 30A/TSEC.

3.3.2 Summary of Findings

Table 10 presents, on a CPCI-by-CPCI basis, the major findings from the MITRE review of the development and product specifications and our source code analysis. Complete descriptions of each CPCI audit are presented under separate cover in appendices B through H.

We found that all of the CPCIs contained numerous discrepancies between the PDL and code, errors in the programming logic and violations of conventional programming styles. Instances in which the source code disagreed with the in-line comments were also found by MITRE. Omissions, inconsistencies and/or erroneous data identified during the evaluation of the individual CPCI product specifications were recorded as part of our findings.

The final contractor-implemented designs for each of the CPCIs were found to contain minor deviations from the authenticated development specifications, including sizing of the RAM and ROM memories and conflicts with the criteria specified in the Government-approved documents. Also, the instructions and user manuals were incomplete and did not comply with the Government's instructions for their preparation and content. These minor deviations were generally resolved by the Government's granting of a waiver to the contractor. After technical interchange meetings with the contractor, relevant comments made by ESD/MITRE were resolved by the contractor, and the specifications, manuals, and other documents were approved.

Table 10. GWEN CPCI Desktop Audit Findings

CPCI	Function	Specifications	Developer/ Lines of Code	Key Findings
HMI	HMI at I/O Station	ESD-GWEN-1 CPDS No. 8564371E CPPS No. 8564371D FQT No. 8567316, Parts I, II & III CPPM No. 8564349-30 UM No. 8564349-28	TRW-CA 2950 FORTRAN 3025 Assembly	1. Design of message editor would not support user processing needs. 2. Absence of traffic flow control. 3. Status message processing excluded external systems. 4. Lack of error checking. 5. Insufficient annotations to explain lists of integer values for control sequences and ASCII integer values for screen messages. 6. Absence of RAM self-test certification upon initialization. 7. Erroneous programmed checksum algorithm (designed as 8-bit; hardware was 7-bit environment).
CIP	COMSEC Interface	ESD-GWEN-1 SRS NO. 8564397D S4 No. 8564369A SPS No. 8564345B FQT No. 8567322 CPMM No. 8564345-30	RCA-NJ 3125 Assembly	1. Inconsistency between code and PDL. 2. Poor code layout. Difficult to distinguish line labels from operands. Incorrect use of "ERROR" labels. 3. Software did not support input and output message throttling, resulting in CIP failure. 4. Design did not support processor congestion notification to source (i.e., NC and HMI).

Table 10. GWEN CPCI Desktop Audit Findings (Continued)

CPCI	Function	Specifications	Developer/ Lines of Code	Key Findings
NCP	Communication Protocols	ESD-GWEN-1 CPDS No. 8564370F CPPS No. 8564370B FQT No. 8564315 CPMM No. 8564370-30	TRW-CA and RCA 2900 FORTRAN 3025 Assembly	1. Incorrect implementation of Permanent Virtual Circuit (PVC) acknowledgment protocol. 2. No recovery for incomplete point-to-multipoint VC when not all destinations are reached. 3. Erroneous TRW software support design between the TRW prototype Intel 86/30 development and the RCA GWEN SESCO 86/10 hardware. 4. Traffic flow control mechanisms between CPCI and network not implemented. 5. Incorrect implementation of the VC "window of four" protocol for congestion notification to originating I/O station. 6. Many of the error-handling routines were stubbed off, resulting in the error handlers going into an infinite loop when called. 7. Absence of rules for assigning symbolic names and register usage conventions, sizing and memory allocation requirements not identified, packet structures and common files not described. 8. Conflict in fixed queue sizing between the CIP and NC for the message priority queues.

Table 10. GWEN CPCI Desktop Audit Findings (Continued)

CPCI	Function	Specifications	Developer/ Lines of Code	Key Findings
				9. Coding utilized by the software developer had been left in the modules. While beneficial from a developmental standpoint, no documentation/in-line comments were presented. Additionally, logger code which traces entries and exits of modules was found in the submitted code, and in the case of assembly routines causes the processor to halt. This code had to be removed. 10. Several routines contained data structures defined as DQ (quadwords). This construct is not supported by the Intel 86/88 programming environment except for the 8087 numeric co-processor. Utilization of the DQ construct could cause problems for later code modification. 11. Indirect recursive calls utilized in several routines are non-compliant with ANSI X3.9-1987. If used they would have to be restricted to recursive data structures which are not implemented within the NC.
TCP	Anti-Spoof Protection,	ESD-GWEN-1 SRS No. 8564397D	RCA-NJ 1580 Assembly	1. Inconsistencies in module-naming conventions.

Table 10. GWEN CPCI Desktop Audit Findings (Continued)

CPCI	Function	Specifications	Developer/ Lines of Code	Key Findings
	EDAC	S4 No. 8564368C SPS No. 8564340B FQT No.8567321A CPMM No. 8564340-30		2. Debugging code utilized in the development was found to have been stubbed off and comments removed. The debugging code should have been removed completely. 3. Hard-coded values were found to have been utilized instead of symbolic names.
TVP	Anti-Spoof, Crypto-Variable Control	ESD-GWEN-1 SRS No. 8564397D S4 No. 8564368C SPS No. 8564341B FQT No. 8567320 CPMM No. 8564341-30	RCA-NJ 1700 Assembly	1. Module preface block information required by ESD-GWEN-1 incomplete. 2. Several modules contain incorrect line labels. Additional line labels which address memory locations are never referenced in the code. 3. Numerous modules were found that never checked return status results. Others were found that always returned success status, regardless of the response. Recommended that a consistent method be implemented for all status processing.
TFE	LF/UHF/Comm Interface	ESD-GWEN-1 SRS No. 8564397D S4 No. 8564368C SPS No. 8564343A FQT No. 8567319 CPMM No. 8564343-30	RCA-NJ 1470 Assembly	1. Unit development folders identified as having incomplete code and PDL. 2. Code examination identified missing and incomplete comments, incomplete and stubbed off self-test routines, and no error recovery implemented.

Table 10. GWEN CPCI Desktop Audit Findings (Continued)

CPCI	Function	Specifications	Developer/ Lines of Code	Key Findings
				Initialization and synchronization algorithm found to be unnecessarily complex and difficult to maintain. 3. RAM and CPU self-test routines not compliant with requirements of ESD-GWEN-1. 4. Line loop usage did not conform to corresponding loop instructions.
BITE	Status Monitoring & Reporting	ESD-GWEN-1 CPDS No. 8564372C CPPS No. 8564346 FQT No. 8567318 CPMM No. 8564346-30	RCA-NJ 3000 Assembly	1. Preliminary versions of the code were limited due to stubbed off modules. Input transmission description (TD) tables were truncated to recognize only one report request. This severely reduced the amount of code which could be tested. 2. RAM test design limited to 1 kbyte. No error notification to the operator.
MT	Site Initialization and Maintenance	ESD-GWEN-1 CPDS No. 8564372C CPPS No. 8564346 FQT No. 8567318 CPMM No. 8163055-30 UM No. 8163055-28	RCA-NJ 2400 BASIC	1. Evaluation of hardware and software identified degraded (slow) processing and limited capability for upgrade/modification. Software language utilized is H-P-Enhanced BASIC. 2. Lack of documentation, commenting and structure in the written code.

Table 10. GWEN CPCI Desktop Audit Findings (Continued)

CPCI	Function	Specifications	Developer/ Lines of Code	Key Findings
				3. Poor programming practices, such as the use of "@" to join instructions in one statement, and the continous upward and downward movement of program flow, made the CPCs difficult to understand and follow. 4. Very weak PDL-to-code connection: unstructured programs, vague and incomprehensive line of PDL-to-code correlation.
GMNC	Status Display & Reports	ESD-GWEN-1 CPDS No.8564373B PIDS No. 8564392C CPPS No. 8564373B FQT No. 8567317 CPMM No. 8564348-30 UM No. 8564348-28	TRW-CO 4000 FORTRAN 8060 Assembly	1. System management functions of the GMNC VAX 11/730 were not identified. 2. No critical alarm notification to operator. 3. The TRW General-Purpose Interactive Display System (GIDS) was poorly documented. It was difficult to trace its interaction with the GMNC application-developed code. 4. Code used only for debugging had not been removed, but only stubbed off. This code had to be removed to facilitate life-cycle maintenance. 5. Misuse of variables to retrieve event start date and time fields.
ROMI	Rekey Operator Machine	ESD-GWEN-1 SRS No. 8564397D	RCA-NJ 2000 FORTRAN	1. Incorrect/incomplete software module documentation.

Table 10. GWEN CPCI Desktop Audit Findings (Concluded)

CPCI	Function	Specifications	Developer/ Lines of Code	Key Findings
	Interface	PIDS No. 8564334B S4 No. 8564323B SPS No. 8567296 FQT No. 8567576 (ROMI & RCVC) CPMM No. 8567295-30 (ROMI & RCVC) UM No. 8567296-28 (ROMI & RCVC)	2000 Assembly	2. VRTX service routines incompletely defined. 3. In Assembly routines, those variable modules declared EXTERNAL but not used needed to be removed. The PUBLIC statement needed a comment to identify whether a module is called by FORTRAN or Assembly.
RCVC	CRYPTO key generation & control	ESD-GWEN-1 SRS No. 8564397D S4 No. 8564323B PIDS No. 8564334B SPS No. 8567295 FQT No. 8567576 (RCVC & ROMI) CPMM No. 8567295-30 (RCVC & ROMI) UM No. 8567296-28 (RCVC & ROMI)	RCA-NJ 1065 FORTRAN 1130 Assembly	1. Detailed structure charts showing the interrelationship of the CPCI routines is missing. 2. Inconsistencies were identified in the naming convention for the interrupt service routine. 3. Incomplete header block information as required by ESD-GWEN-1. 4. Incomplete and inconsistent design of the error checking process. It was recommended that the code be revised to provide a consistent implementation of internal error checking.

Contractor documents relating to the COMSEC and TRANSEC security areas and their impact on acceptable software implementation were also reviewed. Since the contractor's submissions of these documents were behind schedule, and when initially submitted were incomplete, the Government was at risk. However, no significant design deficiencies were identified, and after discussions and technical interchange meetings with the contractor, the documents were finally approved. Detailed discussion of these issues may be found in appendices C and D in a separate report.

3.4 VAX AND MICROPROCESSOR DEVELOPMENT SYSTEM TEST ENVIRONMENT

3.4.1 Activities

Compiling and simulating the GWEN software in the SQA laboratory gave us insight into the sizing of each individual CPCI, its software execution, critical path and timing functioning, and its design. The review and validation of the GWEN TRANSEC algorithm and the packet processing algorithms in the NC received special attention.

Once the source code was received from the contractor and had been through a desktop audit, it was assembled into Intel-formatted Absolute code. A suite of First Systems products (FORTRAN and cross-assembler compilers; linker and locator) was used for compilation of the FORTRAN and Assembly code. The code output from the First Systems tools was then downloaded through the Boston System Offices (BSO) "Converter" for execution within the BSO "Simulator" on the VAX 11/730, or it was downloaded to the HP64000 Microprocessor Development System or the EPROM programmer for execution on the MITRE node hardware. The compilation and assembly of the FORTRAN source code and the Assembly source code utilizing the First Systems and BSO tools are presented in figures 13 and 14, respectively.

The Boston Systems Office software is a complete family of software development tools which were installed on the VAX 11/730 in the SQA facility. These tools included 8085 and 8086 cross-assemblers and symbolic debuggers, format and object file conversion utilities, cross-linkage (link and locate) editors and linkage librarian programs. The utilization of these software tools permitted easy transportability of the RCA-developed source programs into the MITRE SQA facility for analysis and testing, and full debugging of the operating software within the CPCI microprocessor application. Memory manipulation commands (both RAM and ROM), any instructions, program jumps, interrupts, and input/output instructions were easily tested and traced. Cross-linkage editors, load maps and formatted ASCII text were used to generate listings to verify the locations of the program sections and global symbols in the absolute load module which was developed during the linkage process.

Based on the GWEN software architecture presented in figure 5, MITRE developed three distinct configurations utilizing the Microprocessor Development Systems. Figures 15, 16, and 17 show the three configurations employed by MITRE during the SQA effort. In the figures, the location of the HP64000 workstation containing the software state analyzer identifies the CPCIs under which critical timing and throughput requirements were assessed: the Network Control Processor, the Human-Machine Interface Processor, and the TRANSEC Control Processor. With the two HP64000 and the Tektronix 8550 Microprocessor Development Systems, and an HP 1631D Logic Analyzer equipped with additional 8080 and 8085 emulators, MITRE had the capability to monitor the message flow and processing within the entire GWEN I/O node testbed. The TRANSEC simulation was developed and used to assess the strength of the RCA-developed TRANSEC algorithm. The COMSEC Interface Logic simulation was developed and used to test the operational functioning and integrity of the CRYPTO bypass mode of operation between the COMSEC Interface Processor and the Network Control.

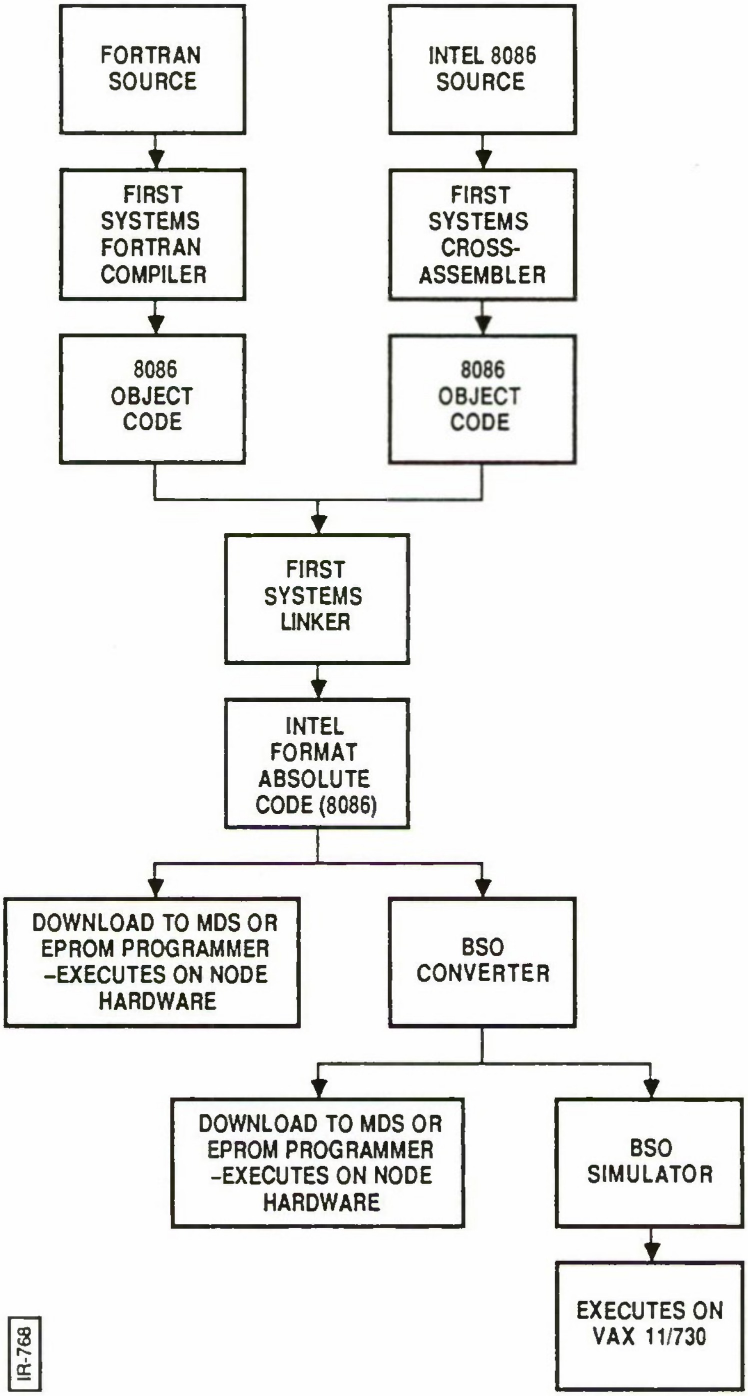


Figure 13. Compilation/Assembly of FORTRAN 8086 Code

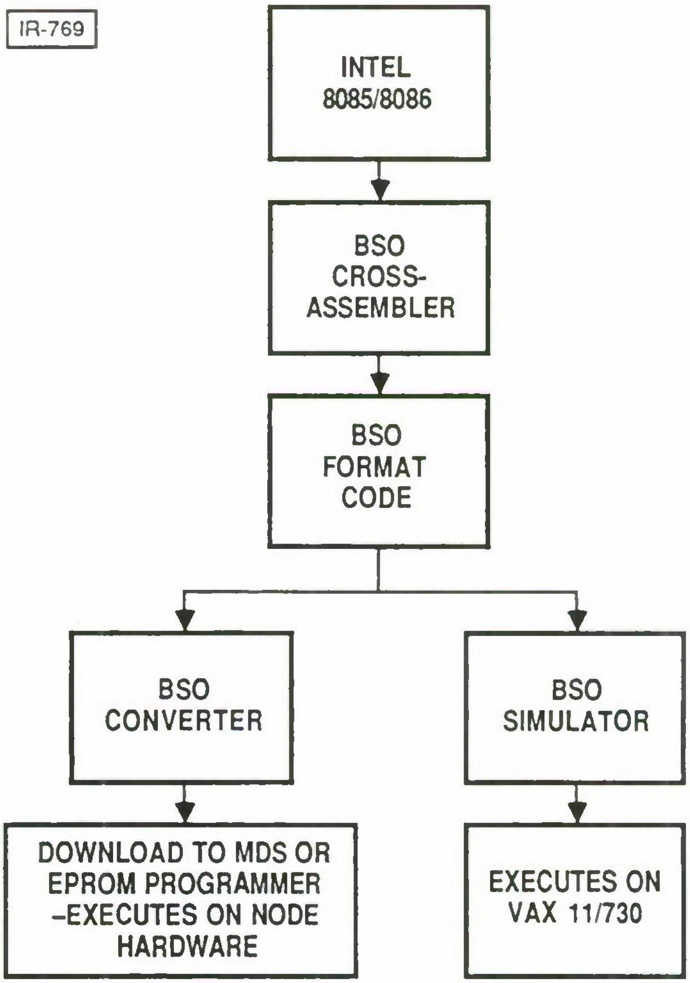


Figure 14. Assembly of 8085/8086 Assembly Code

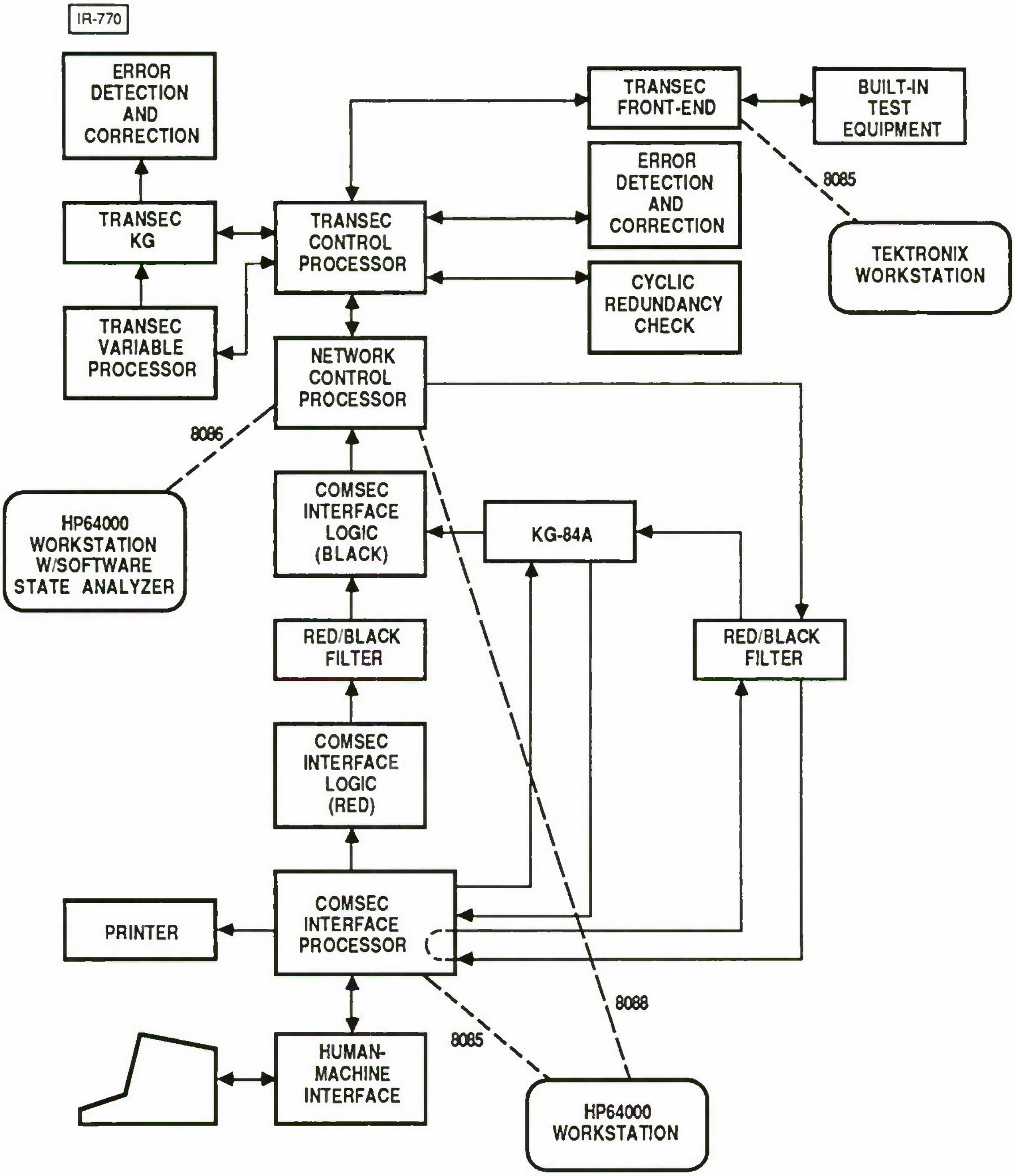


Figure 15. Microprocessor Development System Configuration 1

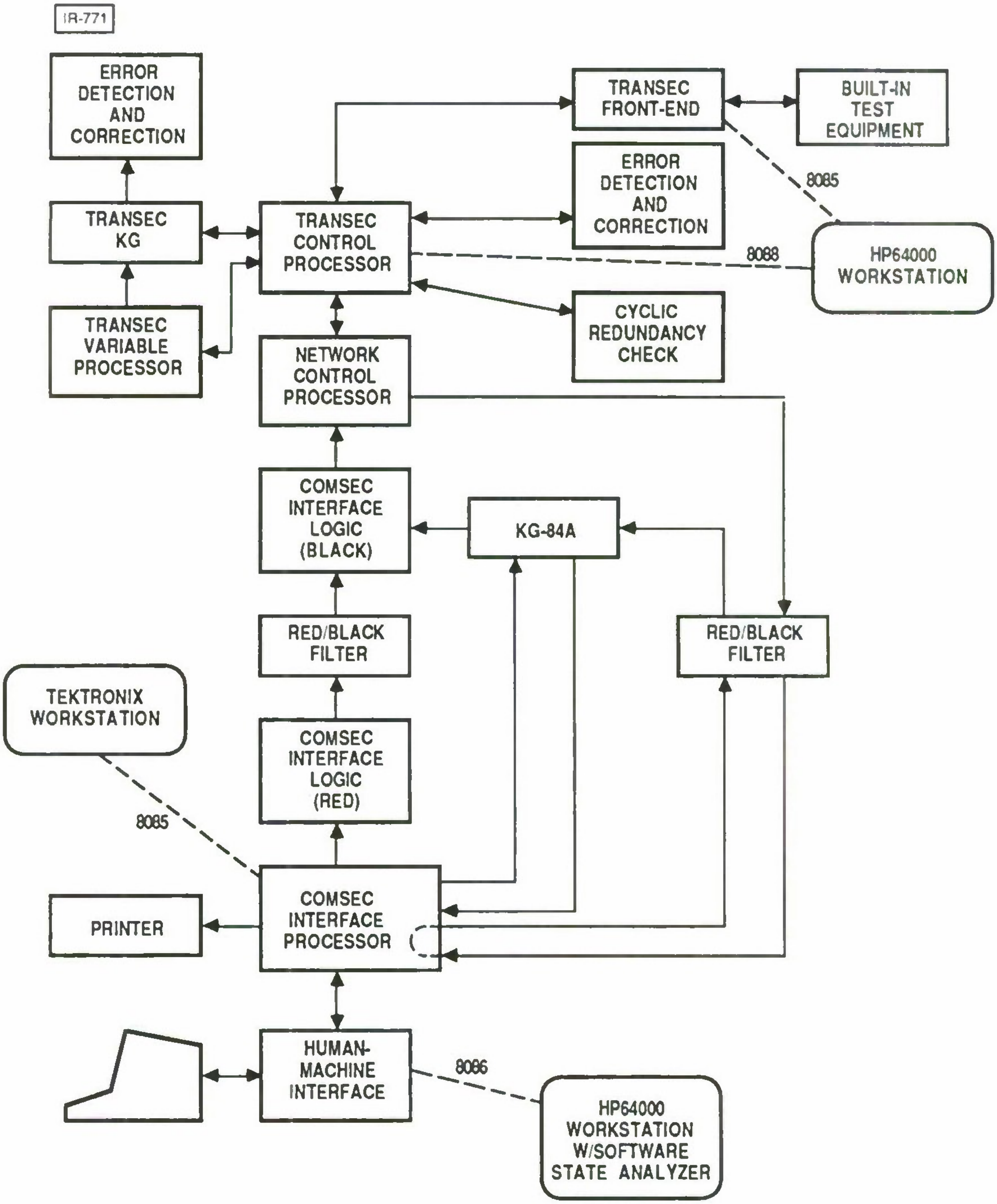


Figure 16. Microprocessor Development System Configuration 2

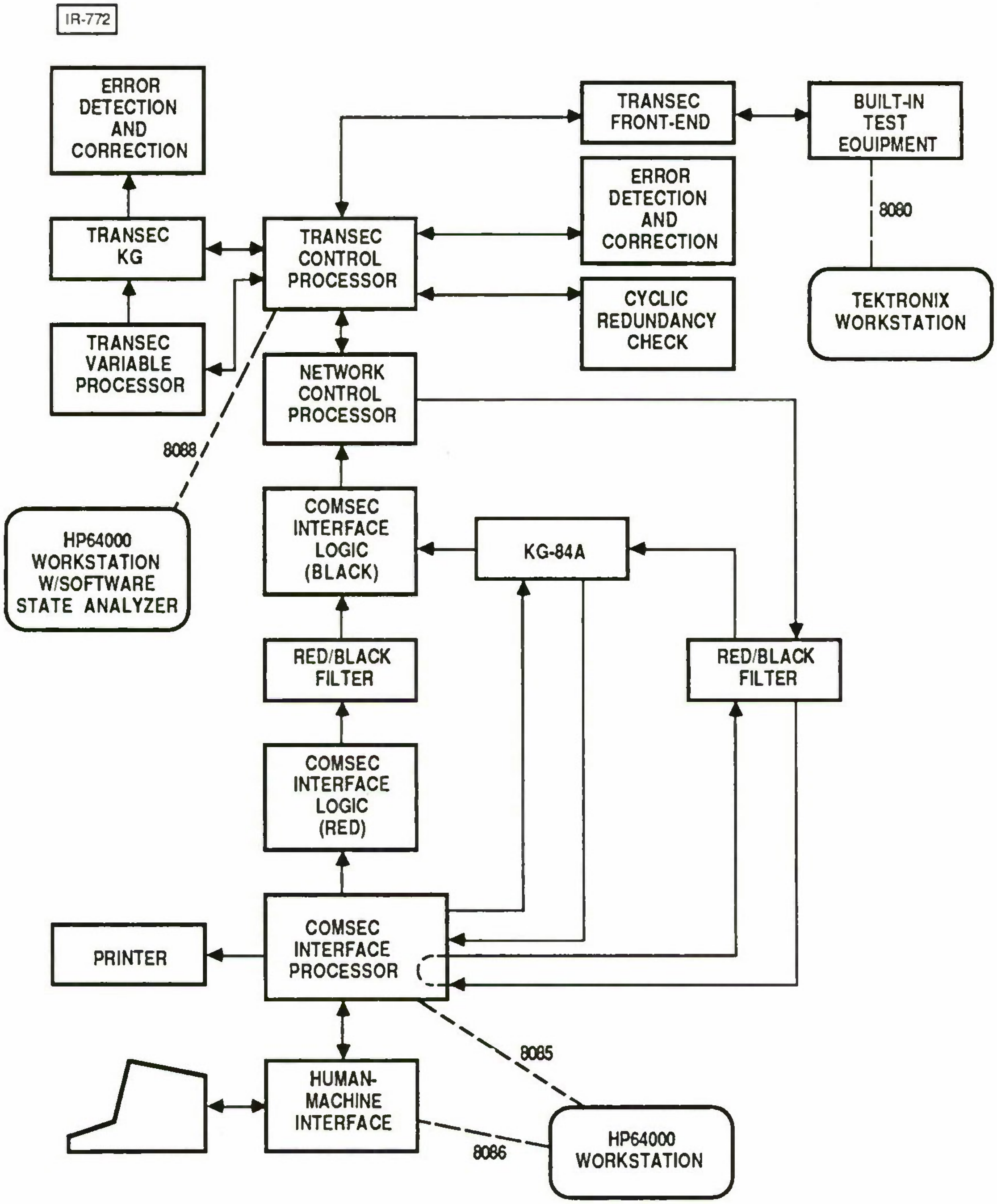


Figure 17. Microprocessor Development System Configuration 3

3.4.2 Summary of Findings

Detailed assessments of the GWEN software ROM and RAM utilization were prepared and provided to ESD for information and to the contractor for corrective action. Our initial findings are summarized in table 11. We identified potential deficiencies in the memory expansion capabilities within the CIP, TFE, TVP, NC and HMI CPCIs. Table 12 identifies the GWEN drawer board assignments which supported our conclusions. Presented in tables 13, 14, and 15 are the final ROM and RAM utilization figures which represent the contractor's solution to the preliminary memory sizing deficiencies. In addition to verifying ROM and RAM sizing requirements of the individual CPCIs, we identified several other problems within the NC CPCI which were forwarded to RCA for resolution. These problems included:

- Possible return of memory packet processing allocation to the caller without deallocating the buffer used to store the packet. This rendered that block of memory unusable and could have led to eventual depletion of the free memory pool.

- Incorrect modification of interrupt service routine prior to its being written to the Programmable Interrupt Timer (PIT).

- Use of stack pointers within the executive service routine, PIEM. Stack pointers are prone to errors caused by hardware interrupts, which require retrieval of objects from the stack that are out of the normal sequence (i.e., below the top of the stack).

Table 11. Preliminary ROM and RAM Memory Sizing
(April 1985)

CPCI	Used	Install	50% Req	Expand	100% Req	Maximum	Meets Req
ROM (Program) Memory Sizing							
BITE	7.71K	32K	11.57K	00K	23.14K	64K	No
CIP	11.90K	16K	17.85K	00K	35.70K	64K	Yes
TCP	10.45K	32K	15.67K	00K	31.34K	1000K	No
TVP	7.71K	32K	11.57K	00K	23.14K	1000K	No
TFE	8.31K	16K	12.47K	00K	24.94K	64K	Maybe
NC	98.85K	128K	148.28K	128K	298.56K	1000K	Yes
HMI	104.11K	128K	156.17K	128K	312.34K	1000K	Yes
RAM (Data) Memory Sizing							
BITE	0.55K	4K	0.82K	00K	1.64K	64K	No
CIP	7.17K	16K	10.75K	00K	21.50K	64K	Maybe
TCP	0.95K	4K	1.43K	00K	2.86K	1000K	No
TVP	1.50K	4K	2.25K	00K	4.50K	1000K	Maybe
TFE	1.03K	16K	1.53K	00K	3.08K	64K	No
NC	75.07K	256K	112.58K	00K	225.15K	1000K	No
HMI	220.00K	256K	330.00K	256K	660.00K	1000K	Yes

Used - Current assessment of the CPCI size.

Install - Memory installed on board that will accommodate final version of software.

50% Req - Requirement specified in ESD/GWEN-1 that memory capacity installed on board shall exceed amount occupied by final version of software by 50 percent.

Expand - Capability for additional memory expansion without board or drawer redesign.

100% Req - Requirement specified in ESD-GWEN-1 that the memory board shall be capable of supporting twice as much memory as the 50 percent requirement amount.

Maximum - Maximum memory the CPU processor can address (ROM and RAM together).

Meets Req - Quick indicator of whether CPCI is or is not meeting GWEN memory requirements.

Table 12. GWEN Drawer Board Assignments, Memory Expansion

Drawer	Slot	Board Description	Board Type
TIP I/O	1	Power Supply Equip	RCA Design
	2	CIP Processor	80/544
	3	CIP I/O	80/544
	4	HMI Communication	80/534-2
	5	HMI Communication	(not used)
	6	(spare)	
	7	(spare)	
	8	HMI Expansion	
	9	HMI RAM	80/056A
	10	HMI ROM	80/164
	11	HMI Processor	86/10
	12	CIL Red	RCA Design
	13	Red/Black Filter	RCA Design
	14	CIL Black	RCA Design
	15	(spare)	
TRANSEC/NC	1	(not used)	
	2	EDAC Board	E-FXA
	3	TCP Memory	E-FXB
	4	E-FXC	E-FXC
	5	TCP KG	E-FXD
	6	TCP Processor	E-FXE
	7	Red/Black Filter	E-FXF
	8	Black I/O #1	E-FXG
	9	Black I/O #2	E-FXH
	10	TFE Processor	80/544
	11	TFE I/O	80/544
	12	NC Processor	86/10
	13	NC RAM	80/056A
	14	NC ROM	80/164
	15	NC ROM	80/164
BITE	1	I/O Status Board	RCA Design
	2	Clock Generator	RCA Design
	3	(spare)	
	4	BITE RAM/ROM	80/164
	5	BITE Processor	80/10A
	6	(spare)	
	7	UHF Modem	RCA Design
	8	UHF Modem	RCA Design
	9	I/O UHF Board	RCA Design

Table 13. CPCI Memory Usage
(November 1987)

CPCI	Version	Used ROM : RAM	Maximum ROM : RAM	Installed ROM : RAM
BITE	7.05	2715 : 750	8K : 1K	4K : 1K
CIP	7.17	11911 : 6582	16K/32K : 16K/32K	16K/32K : 16K/32K
GMNC	5.11	NA : 139400	NA : 5MB	NA : 2MB
HMI	7.19	120072 : 243518	256K : 256K	192K : 256K
MT	7.40	NA : 108215	58K(1) : 438K	58K(1) : 128K(2)
NC	7.32	142470 : 128173	256K : 256K	192K : 256K
RCVC	6.03	39259 : 78219	256K : 256K	64K : 256K x2
ROMI	6.02	101698 : 91903	256K : 256K	128K : 256K
TCP	7.20	7192 : 992	32K : 4K	16K : 4K
TFE	7.20	6078 : 1112	16K/32K : 16K/32K	8/32K : 16/32K
TVP	7.17	4844 : 828	32K : 4K	8K : 4K

Used - Current assessment of the size of the CPCI.

Maximum - Maximum memory the CPU processor can address (both ROM and RAM together).

NA - Not Applicable.

(1) ROM is not available for use to the user.
(2) RAM consists of 64K internal and 64K external.

Note: Double-density boards are used only for HMI and NC.
Unit size is in bytes.

Table 14. ROM (Program) Memory Configuration
(November 1987)

CPCI	Chip Type (Quantity)	Used Memory	Installed: Expansion	Required 50% Exp : 100% Exp	Remarks
BITE	2716(2)	2715	4096 : 8194	4072.5 : 8145	
CIP	2764(2)	11911	16384 : 16384	17866.5 : 35733	1
	27256(1)	11911	32768 : 32768	17866.5 : 35733	*
GMNC		NA	NA : NA	NA : NA	
HMI	27128(12)	120072	196608 : 262144	180108 : 360216	1,2
MT		NA	NA : NA	NA : NA	
NC	27128(12)	142470	196608 : 262144	213705 : 427410	1,2
RCVC	27128(4)	39259	65536 : 262144	58889 : 117777	
ROMI	27128(8)	101698	131072 : 262144	152547 : 305094	1,2
TCP	2764(2)	7192	16384 : 32768	10788 : 21576	
TFE	2764(1)	6078	8192 : 16384	9117 : 18234	1
	27256(1)	6078	32768 : 32768	9117 : 18234	
TVP	2764(1)	4844	8192 : 32768	7266 : 14532	

Used - Current assessment of the size of the CPCI.

Installed - Available memory installed on board.

Expansion - Total memory available without board redesign.

Req 50% Exp - Requirement specified in ESD-GWEN-1 that memory capacity installed on board shall exceed amount occupied by final version of software by 50 percent.

Req 100% Exp - Requirement specified in ESD-GWEN-1 that the memory board shall be capable of supporting twice as much memory as the 50 percent requirement amount.

Remarks
1 - Board being upgraded to use 27256 bit memory chip to meet 100 percent requirement.
2 - 50 percent requirement met by populating entire board with current memory chip type.
* - Cannot meet 100 percent requirement.

Note: Unit size is in bytes.

Table 15. RAM (Data) Memory Configuration
(November 1987)

CPCI	Chip Type (Quantity)	Used Memory	Installed: Expansion	Required 50% Exp : 100% Exp	Remarks
BITE	6508(8)	750	1024 : 1024	1125 : 2250	*
CIP	6116(8)	6582	16384 : 16384	9873 : 19746	1
	64-02M(4)	6582	32768 : 32768	9873 : 19746	2
GMNC		1394000	2M : 5M	2091000 : 4182000	*
HMI	4164(32)	243518	262144 : 262144	365277 : 730554	3
MT		108215	131072 : 448512	162322.5 : 324645	4
NC	4164(32)	128173	262144 : 262144	192259.5 : 384519	3,5
RCVC	4164(32)	78219	262144 : 262144	117329 : 234658	3
ROMI	4164(32)	91903	262144 : 262144	137855 : 275710	3,6
TCP	6116(2)	992	4096 : 4096	1488 : 2976	
TFE	6116(8)	1112	16384 : 16384	1668 : 3336	
	64-02M(4)	1112	32768 : 32768	1668 : 3336	
TVP	6116(2)	828	4096 : 4096	1242 : 2484	

Used - Current assessment of the size of the CPCI.

Installed - Available memory supplied with board.

Expansion - Total memory available without board redesign.

Req 50% Exp - Requirement specified in ESD-GWEN-1 that memory capacity installed on board shall exceed amount occupied by final version of software by 50 percent.

Req 100% Exp - Requirement specified in ESD-GWEN-1 that the memory board shall be capable of supporting twice as much memory as the 50 percent requirement amount.

Remarks
1 - Use 64-02M chips to meet 100 percent requirement.
2 - Only 30K (30720) of the RAM is addressable.
3 - 44 chips/board: 32 for RAM & 12 for EDAC.
4 - 50 percent & 100 percent requirements met by installing additional (external) 64K memory modules.
5 - Slot #15 in TRANSEC/NC drawer used to provide an additional 256K to meet the 100 percent requirement.
6 - Extra slot, XA14, used to provide an additional 256K to meet the 100 percent requirement.
* - Cannot meet 50 percent and 100 percent requirements.

Note: Unit size is in bytes.

Utilizing the Boston System Office 8086 symbolic debugger running on the VAX 11/730, we accomplished a detailed analysis of the GWEN software queue structures. The VRTX executive program, together with the individual CPCI executive augmentation code written by RCA and TRW, provided the CPCIs with multitasking management, memory allocation, and intertask communications and synchronization. We found that in the RCA-developed CPCIs, the VRTX augmentation routines were consistent with the requirements of the CPCIs. TRW developed the HMI and NC CPCIs. We found that while the HMI used the executive service augmentation code properly, it was overused in the NC, with many nonessential routines and calls. Another point which we noted during our analysis of the queue structure was TRW's use of recursive calls within the module design. This implementation is not straightforward and made debugging and maintenance of the code more difficult. We discussed our concerns with RCA who agreed to restructure the TRW design to make it consistent with the NC requirements.

A detailed analysis of the RCA system timeline was completed in April 1986. The results verified that the NC does initiate the proper transfers to the TCP in the proper sequence and with relative timeliness, according to the RCA system timeline. Additional discussions on the RCA system timeline and its relationship to the MITRE measurements and findings are presented under separate cover in appendix F.

Throughput limits for input and output of messages within the BITE CPCI were never defined as a specific development requirement. MITRE's preliminary testing identified a potential problem with the BITE design, which uses a ring buffer as a temporary storage media. Under a stress condition ("stress" defined as the input and output of message traffic at a sustained processing rate of one message in each direction every 2/3 second), our preliminary testing identified a potential for the ring buffer to saturate, resulting in the overwriting of traffic on the ring. RCA was reluctant to test this requirement as it was not a specific development requirement. Accordingly, we wrote a program for use in the SQA facility which stressed the BITE, making it receive and originate a file of 100 messages at a sustained rate of one message in each direction every 2/3 second. The program was executed in the SQA facility, using the actual BITE code developed by RCA. Several scenarios of the test were run, all of which determined that the BITE performed properly and was able to handle the node's traffic as well as the maintainer terminal traffic under the stress conditions. Additional information related to this test effort is presented under separate cover in appendix G.

Taking the same instructions utilized by RCA in the development of the GWEN TRANSEC algorithm (derived from the classified appendix to the GWEN Statement of Work), we developed our own TRANSEC algorithm which was incorporated into the TRANSEC simulation and downloaded into the MDS environment. Test inputs for our simulator were obtained from RCA, along with copies of their TRANSEC algorithm test results. After executing our simulator with RCA's test inputs, a comparison was made of MITRE's and RCA's actual results. We found both sets of results to be identical, providing a high level of confidence that RCA had correctly implemented the algorithm and provided the required level of TRANSEC strength. To complete our assessment, our test results were compared to NSA's (the original test input supplier) expected results. These were also in agreement. Based on all results of the TRANSEC algorithm tests, we advised ESD that RCA's TRANSEC algorithm would provide the level of protection required by ESD-GWEN-1 and the GWEN contract.

The CRYPTO bypass mode of operation was tested on the MITRE-developed CIL simulation software. The simulation software, together with the COMSEC Interface Processor software, was downloaded from the VAX into the MDS environment. Using the MDS we verified that RCA's COMSEC Interface Logic design would allow message header bypass to the Network Control from the COMSEC Interface Processor, would properly validate the header portion of the test message and would maintain the integrity of the red/black message processing between the CIP, the KG-84A and the NC. By having used the CIL simulation to replicate the RCA CIL board design, we were able to advise ESD with a high level of confidence that RCA's software correctly implemented the message header validation and bypass, while retaining the integrity of the message red/black communication.

3.5 MITRE NODE TEST ENVIRONMENT

3.5.1 Activities

MITRE's GWEN node testbed environment is a replica of a GWEN Input/Output station, running GWEN software on commercially available Intel microcomputer boards. Some of the GWEN hardware was custom-designed by RCA (TRANSEC functions for encryption/decryption, red/black filtering, EDAC and CRC) and could not be constructed with off-the-shelf Intel boards. In these cases, MITRE used Intel-equivalent boards and simulated the custom RCA functions with MITRE-developed software.

The MITRE node configuration, presented in figure 18, shows the HMI (8086), CIP (8085), CIL simulator and the NC simulator in HP64000 emulation memory. Intel hardware boards were used for these processors, but the CPCI CPUs and associated memory were emulated by the HP64000 Microprocessor Development System to provide visibility into the processors during debug, integration and analysis. Both the HMI plasma display and the HMI external network simulator were simulated with MITRE-developed software executed on IBM personal computers.

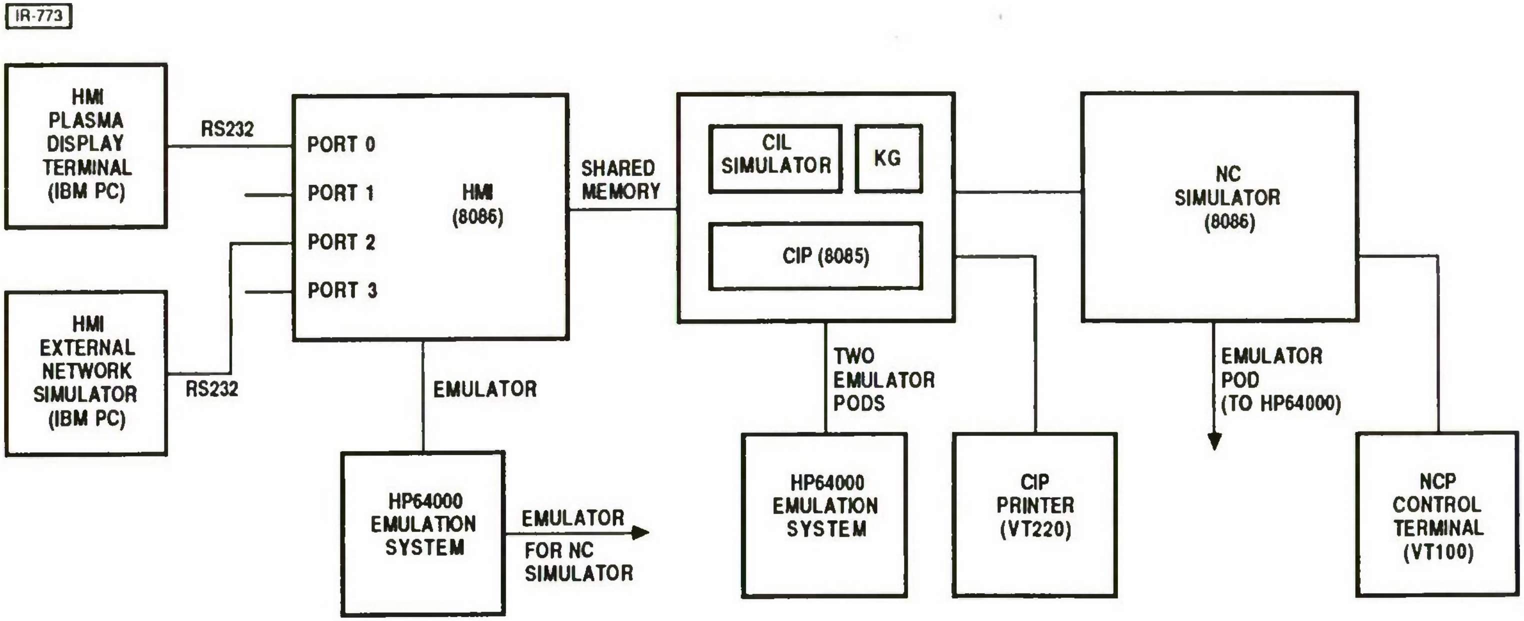


Figure 18. MITRE Node Hardware Configuration for HMI and NC CPCIs

The node test environment continued to evolve over the life of the SQA effort. The NC Driver Interface (NCDI) was developed and designed to operate on an Intel 86/12A single-board computer and act as the interface between the NC driver (IBM-PC) and the NC. The NC driver supported transfer of an interval's worth of packets to the NC and received an NC packet every 2/3 second. The driver displays and records in a log file all data communicated to and from the driver and the NC. NC software was burned into PROM, installed in an Intel 86/12A board and initialized as a Relay Node.

The BITE processor was integrated into the node. A switching matrix was developed which supported toggling of the LRU reporting status leads to the BITE. Utilizing this capability, the BITE was stressed to verify its throughput and reporting capabilities.

The results and findings of the above SQA efforts were then compiled, a report generated and the findings discussed with the GWEN Program Office and the contractor.

3.5.2 Summary of Findings

The following specific areas of concern within the HMI CPCI were tested with the test configuration presented in figure 18:

- Power-on processing and self-test;
- Site-specific data initialization;
- Communication between the COMSEC Interface Processor and the Human-Machine Interface.

We determined that the HMI's RAM control and internal processing test functions contained several deficiencies. On power-up the HMI did not access each of the RAM chips for proper initialization. This resulted in the EDAC system's indicating an error had occurred. We recommended to the contractor that the HMI software be modified to support proper RAM chip initialization.

A second issue uncovered during the node testing of the HMI was the limited site-specific data processing ability of the HMI processor. At a sustained rate of one packet from the NC every second, we found that the HMI processor was never able to complete initialization because the initialization data was being overwritten by the incoming data. By varying the input rate of initialization data (10 seconds per packet) to the HMI, we found that the HMI was able to initialize properly and consistently. Analysis of this problem revealed that although the HMI acknowledged (ACK) or did not acknowledge (NACK) each initialization packet from the maintenance terminal, the MT processing of the ACK or NACK requires only 0.5 to 1 second before the MT will originate the next packet. The HMI, on the other hand, requires between 2 to 6 seconds to fully process the received packet. The problem was discussed with RCA and resulted in RCA's incorporating a throttling scheme into the HMI initialization processing.

In the last area addressed in this test effort, communication between the CIP and HMI, we noted that upon receipt of a Transaction Descriptor (TD) from the CIP, the HMI validated the TD. It was found that if the TD was not valid, the message was discarded by the HMI with no notification to the CIP. With no mechanism to recover a discarded message or to recover an initialization packet count, the HMI would fail. Discussions with RCA resulted in the inclusion of full message ACK/NACK and recovery protocol between the HMI and CIP processors. Additional discussions and findings pertaining to the execution of GWEN software in the node testbed for the individual Computer Program Configuration Items are presented under separate cover in appendices B through H.

3.6 SOFTWARE CODE MAINTAINABILITY

3.6.1 Activities

The maintainability of the CPCI at the code level is based on three major points. First is simplicity of implementation. The program should be implemented in the most clear and concise manner possible. Long-term maintenance will be less time-consuming if the source code stays within established programming conventions. Unique data manipulations and complex algorithms, while functionally correct, may confuse the personnel responsible for software maintenance. Second, all supporting documentation should be clear, concise and complete. Code modifications and annotations must be reflected within the documentation. These code annotations should provide maintenance personnel with a primary source of information regarding the program's operation. Third is the use of modular top-down design. Implementing a single function per module allows a particular function to be changed without impacting the entire CPCI.

While addressing maintainability, we also performed a detailed review of the contractor-defined modification and verification procedures presented in each CPCI's Computer Program Maintenance Manual. Finally, as a part of the overall software maintainability assessment, we examined the feasibility of converting the majority of the GWEN software (implemented in Assembly by RCA) into an HOL. We examined the constructs in the delivered software and performed trial compilations with several FORTRAN and "C" compilers.

3.6.2 Summary of Findings

We found in general that the GWEN CPCIs could not be easily maintained. Although the contractor utilized a top-down design, the design documentation for the individual CPCIs (i.e., Computer Program Product and Software Product Specifications and the Computer Program Maintenance Manuals) were insufficient in terms of level of technical detail. Numerous modules exceeded sizing limitations imposed by the GWEN Statement of Work and ESD-GWEN-1. We also noted that the oversized multitasking modules were difficult to follow and/or modify since their interfaces and affected registers were not clearly recognizable.

Another concern we identified to the contractor was the use of "hard-coded" values for value range checks and loop counters. If these hard-coded values were to require a coding change, it would be both time-consuming and costly due to the necessity of a line-by-line code replacement.

In the HMI, the mailbox algorithm was poorly documented. Additionally, the integer values for the control sequence and ASCII values for the screen values were not adequately defined.

Within all CPCs, lack of a standard usage was noted in naming conventions to indicate digits and/or variable names as parameters. Module headers were incomplete and in some cases missing; nested assembly procedures were called from outside of the CPC.

The Computer Program Maintenance Manuals did not identify the required command files which are essential for the compilation of an individual CPCI source code into an executable file. Instructions to guide the Government programmers in making modifications of the source code were incomplete and would require in-depth knowledge of the contractor's software development practices. The procedures as reviewed needed extensive revision to be compliant with the Government requirements, and these revisions were required of the contractor.

In the MT CPCI, poor computer program design and inconsistency between the program's documentation and the code were major concerns. Programming language limitation of the selected hardware, the HP-85B, required the use of HP's Enhanced BASIC as the CPCI development language. This language is not an approved Higher-Order Language (HOL), cannot meet the Government's software design and construction requirements, and is poorly documented. However, since the operational and maintenance concept was based on the use of the MT as designed, a waiver for the use of BASIC was requested. In order to support the request and in response to concerns about the HP-85B, we sought alternate choices for the MT. We analyzed the current version of the MT software and converted it to run on a newer, faster personal computer, the HP-IPC. We demonstrated a significant increase in speed. We also used both computers to demonstrate improvements in software maintainability that would be available with either the HP-IPC in BASIC or a typical PC (for example, the IBM PC) in another HOL. While the waiver for the use of BASIC was granted, newer hardware may soon be selected by Government logistics personnel as a replacement for the HP-85B.

An additional concern was noted within the NC CPCI. In the Assembly routines, some of the IF constructs were coded with a conditional "jump" followed immediately by an unconditional jump. The coding of the unconditional jump was unclear. It was felt that in all of the routines where this coding practice was found a single conditional jump would have been sufficient, in addition to being more efficient, clear, and concise.

Within the GMNC, the General-Purpose Interactive Display System (GIDS) documentation for the GIDS utility library CPC did not define the GIDS flow and its interaction with the other GMNC CPCs. For maintainability, it was important that the GMNC program documentation present a comprehensive design to include fully defined inter-task and inter-processor descriptions.

We found in our HOL conversion feasibility study that the most suitable conversion language is "C", due to low code expansion and wide availability of production-quality compilers. These findings were also used to support a Pre-Planned Product Improvement (P^3I) plan for GWEN HOL software conversion that was done in December 1986.

Detailed discussions and findings pertaining to our analyses of individual Computer Program Configuration Item maintainability are presented under separate cover in appendices B through H.

SECTION 4

CONCLUSIONS AND RECOMMENDATIONS

4.1 CONCLUSIONS

The GWEN SQA effort provided a superior method for understanding the actual software design and documentation process. It provided an early and ongoing examination of the individual CPCI Program Design Language (PDL) and source code, and allowed comparison of the PDL with the code. However, it also reinforced the point that for a successful SQA effort, software must be received by MITRE and reviewed as soon as it has been unit-tested and placed under configuration control by the contractor. Doing this provides a major benefit of positive and open communication with the contractor and early feedback of findings from MITRE to the contractor. This reduces potential delay in the development schedule and/or the accrual of additional cost. In this case, the SQA effort provided MITRE personnel with solid experience on the adequacy of formal qualification test plans/procedures and a detailed knowledge of the hardware environment and the qualifying test scenarios, benefits that would be useful on any program.

The thrust of the SQA effort concentrated on verification of each CPCI's implementation completeness, correctness, readability, and maintainability. Completeness was determined via analyses which revealed that the source code was generally well-written and satisfied the requirements as delineated in the GWEN Software Requirements Specification, the Software System/Subsystem Specifications (for TRANSEC/COMSEC-designated CPCIs), and the Computer Program Development Specifications (for all other CPCIs). We verified the development requirements/source code analysis by correlating the source code at the Computer Program Component (CPC) level with the requirements traceability matrices of the TRANSEC/COMSEC-designated CPCIs'

Software Product Specifications and with the Computer Program Product Specification for all other CPCIs. Correctness was judged in terms of correlation with higher-level documents and consistency with the PDL. Modules were identified where discrepancies existed between the source code and the PDL. The readability evaluation revealed that many modules were deficient in header information.

Within the Network Control CPCI, we verified that the GWEN critical timing requirements could be met by the software. Our examination of the NC and HMI queue structures and their handling of messages identified an absence of a buffer overflow processing protocol that would ensure that high-precedence traffic would be delivered to the user. We additionally identified several potential problem areas within the queue structure design which, during periods of heavy traffic processing, could impact compliance with the Government's specified time constraints. We recommended that formal Government acceptance of the NC and HMI software include stress testing of the queue structures.

Several security concerns were identified within the TRANSEC CPCIs. The specific areas addressed were of a classified nature and related to the CPCIs' software design and hardware/software integration and qualification testing. RCA, in response to our SQA findings, revised their software design, their software/hardware integration effort, and expanded the qualification test procedures.

We tracked the memory allocation in hardware and software, identifying early on that the contractor's hardware design would not be compliant with the Government's memory expansion requirements. We recommended that the contractor develop a memory allocation work-around plan utilizing new higher-density boards.

Specific problem areas identified during the review of the individual CPCIs are addressed in detail in the respective CPCI appendices, to be published in a separate volume. These issues were discussed in detail with ESD and the contractor. Corrective actions were subsequently agreed upon between ESD, RCA and MITRE, and all have been fully implemented.

4.2 RECOMMENDATIONS

For future SQA efforts, early involvement of contractor, Government and SQA personnel is essential. The inclusion of Statement of Work paragraphs that identify the SQA effort, selection of the SQA group or contractor, definition of the scope of effort with supporting milestones, and the selection of the facilities to perform SQA need to be completed prior to the contractor's commencement of software development.

A specific concern for GWEN is the need for replacement of the Maintenance Terminal HP-85B hardware and source code as expeditiously as possible. The current hardware is no longer manufactured by the vendor and requires HP-copyrighted software to download and execute the contractor-developed MT CPCI.

Conversion of the GWEN software as a Pre-Planned Product Improvement (P^3I) to an approved Higher-Order Language (HOL) may be useful. Our calculations on the cost and schedule for this effort, previously documented in March 1986, support conversion to an HOL as both feasible and desirable.

GLOSSARY

ACK	Acknowledgment
ACSN	Advance Change/Study Notice
ADD	Algorithm Description Document
ADQ	Administrative Queues
AFOTEC	Air Force Operational Test and Evaluation Center
ASM	Assembly Language
ATP	Acceptance Test Procedure
BITE	Built-In Test Equipment
BP	Base Pointer
BSO	Boston Systems Office
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CIL	COMSEC Interface Logic
CILS	COMSEC Interface Logic Simulator
CIP	COMSEC Interface Processor
COMSEC	Communications Security
CONUS	Continental United States
COTS	Commercial Off-The-Shelf
CPC	Computer Program Component
CPCI	Computer Program Configuration Item
CPDS	Computer Program Development Specification (B-5)
CPM	Computer Program Module
CPMM	Computer Program Maintenance Manual
CPPS	Computer Program Product Specification (C-5)
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CRT	Cathode Ray Tube
CRVM	Cross-Reference Verification Matrix
CVP	Cryptographic Verification Plan
CVR	Cryptographic Verification Report
ECP	Engineering Change Proposal
EDAC	Error Detection and Correction
EPROM	Electrically Programmable Read-Only Memory
ESD	Electronic Systems Division
FIFO	First In, First Out
FOC	Final Operational Capability
FOR	FORTRAN
FQT	Formal Qualification Test
FS	First Systems
GIDS	General-Purpose Interactive Display System
GMNC	GWEN Maintenance Notification Center

GLOSSARY (Continued)

GVMSOS	GMNC'S GIDS VMS Operating System CPC
GWEN	Groundwave Emergency Network
GWPF	GMNC'S GIDS Word Processing Function CPC
HLT	Halt
HMI	Human-Machine Interface
HOL	Higher-Order Language
HP	Hewlett-Packard
HQ	Headquarters
I/O	Input/Output
ICD	Interface Control Document
IEC	Interstate Electronics Corporation
IEMATS	Improved Emergency Message Automated Transmission System
IOC	Initial Operational Capability
ITMCS	Inter-task Message Communications System
LF	Low Frequency
LF/CR	Line Feed/Carriage Return
LRU	Line-Replaceable Unit
MDS	Microprocessor Development System
MT	Maintainer Terminal
NACK	No Acknowledgment
NCDI	Network Control Driver Interface
NCP	Network Control Processor
NCPD	Network Control Processor Driver
NORAD	North American Air Defense Command
OJCS	Office of Joint Chiefs of Staff
P^3I	Pre-Planned Product Improvement
PDL	Program Design Language
PIDS	Prime Item Development Specification
PIT	Programmable Interrupt Timer
PLDC	Permanent Low Duty Cycle
PPI	Programmable Peripheral Interface
PQT	Preliminary Qualification Test
PROM	Programmable Read-Only Memory
QA	Quality Assurance
R/B	Red/Black
RAM	Random Access Memory

GLOSSARY (Continued)

RCVC	Rekey Crypto Variable Controller
RN	Relay Node
RO	Receive Only
ROM	Read-Only Memory
ROMI	Rekey Operator-Machine Interface
S^4	Software System/Subsystem Specification
SAC	Strategic Air Command
SCCB	Software Configuration Control Board
SFA	Security Fault Analysis
SOW	Statement of Work
SPO	System Program Office
SPS	Software Product Specification (C-5)
SQA	Software Quality Assurance
SRP	System Resource Program
SRS	Software Requirement Specification
STR	Software Trouble Report
TCP	TRANSEC Control Processor
TD	Transmission Descriptor
TEMSS	Test and Evaluation Message Signal Simulator
TEK	Tektronix
TFE	TRANSEC Front End
TIM	Technical Interchange Meeting
TLCC	Thin Line Connectivity Capability
TOD	Time of Day
TRANSEC	Transmission Security
TRS	TRANSEC Rekey Station
TTL	Transistor-Transistor Logic
TVP	TRANSEC Variable Processor
UHF	Ultrahigh Frequency
UM	Users Manual; Unit Manager
UUT	Unit Under Test
VC	Virtual Circuit
V&V	Verification & Validation
VRTX	Real-time operating system

BIBLIOGRAPHY

Standards:

MIL-STD-483	Configuration Management Practices for Systems, Equipment, Munitions and Computer Programs	21 Mar 1973
MIL-STD-1753	FORTRAN DOD Supplement to American National Standards X3.9-1978	9 Nov 1978
MIL-STD-52779A	Software Quality Assurance Program Requirement	1 Aug 1979
NACSEM 5112	Nonstop Evaluation Techniques	Apr 1975
NACSIM 5100A	Compromising Emanations Laboratory Test Standard Electromagnetics	Jul 1981
KAG-30A/TSEC	Compromising Emanations Standards for Cryptographic Equipment	Jan 1971
NSAM 81-3	NSA/CSS Software Product Standards Manual	26 Jul 1979

Other Applicable Documents:

X3.9-1978	Programming Language FORTRAN, American National Standards Institute, New York, NY	3 Apr 1978
ESD-GWEN-1	System Specification for Groundwave Emergency Network, Rev. A	23 Sep 1983
ESD-TR-84-171	Handbook for Evaluation and Life Cycle Planning for Software, Vol. III: Reviews, Audits and CPCI Specifications	1 Feb 1983
8564324	Algorithm Description Document for GWEN Thin Line Connectivity Capability System	7 Mar 1985
8564350B	Computer Program Timing and Sizing Data for GWEN Thin Line Connectivity Capability System	6 Mar 1986
8564397D	Software Requirement Specification for GWEN Thin Line Connectivity Capability (including change dated 16 Nov 1987)	9 Feb 1987

BIBLIOGRAPHY (Continued)

8564397D	Appendix I to SRS for GWEN Thin Line Connectivity Capability (including change dated 16 Nov 1987)	9 Feb 1987
8564397E	Appendix II to SRS for GWEN Thin Line Connectivity Capability	9 Oct 1987
8564397D	Addendum to Appendix II	20 Nov 1987
8564372C	Built-In Test Equipment Computer Program Development Specification (including change dated 30 Aug 1985)	19 Dec 1984
8564346	Built-In Test Equipment Computer Program Product Specification (including change dated 24 Sep 1987)	1 Apr 1985
8567318	Built-In Test Equipment Formal Qualification Test Procedures	13 Sep 1985
8564346-30	Built-In Test Equipment Computer Program Maintenance Manual	11 Sep 1987
8564369A	COMSEC Interface Processor System/Subsystem Software Specification	17 Dec 1984
8564345B	COMSEC Interface Processor Software Product Specification	25 Sep 1987
8567322	COMSEC Interface Processor Formal Qualification Test Procedures	17 Apr 1986
8564345-30	COMSEC Interface Processor Computer Program Maintenance Manual	15 Sep 1987
8564392C	GWEN Maintenance Notification Center Prime Item Development Specification (including change dated 20 Feb 1986)	26 Sep 1985
8564373B	GWEN Maintenance Notification Center Computer Program Development Specification for GWEN (Part I)	19 Dec 1985
8564373B	GWEN Maintenance Notification Center Computer Program Product Specification for GWEN Thin Line Connectivity Capability (Part II)	17 Sep 1987

BIBLIOGRAPHY (Continued)

8567317	GWEN Maintenance Notification Center Formal Qualification Test Procedures	14 Jan 1986
8564348-30	GWEN Maintenance Notification Center Computer Program Maintenance Manual	14 Sep 1987
8564348-28	GWEN Maintenance Notification Center Users Manual	30 Oct 1987
8564371E	Human-Machine Interface Computer Program Development Specification	12 Nov 1985
8564371E	Human-Machine Interface Computer Program Product Specification for GWEN TLCC	30 Mar 1988
8567316	Human-Machine Interface Formal Qualification Test Procedures (Part 1)	17 Jul 1986
8567316	Human-Machine Interface Formal Qualification Test Procedures (Part 2)	12 Aug 1986
8567316	Human-Machine Interface Formal Qualification Test Procedures (Part 3)	21 Oct 1987
8564349-30	Human-Machine Interface Computer Program Maintenance Manual	14 Sep 1987
8564349-28	Human-Machine Interface Users Manual	9 Sep 1987
8564372C	Maintainer Terminal Computer Program Development Specification (including change dated 30 Aug 1985)	19 Dec 1984
8564346	Maintainer Terminal Computer Program Product Specification (including change dated 24 Sep 1987)	1 Apr 1985
8567318	Maintainer Terminal Formal Qualification Test Procedures	13 Sep 1985
8163055-30	Maintainer Terminal Computer Program Maintenance Manual	14 Sep 1987
8163055-28	Maintainer Terminal Users Manual	14 Oct 1987

BIBLIOGRAPHY (Continued)

8564370G	Network Control Computer Program Development Specification (Part I)	10 Jun 1986
8564370B	Network Control Computer Program Product Specification (Part II)	5 Oct 1987
8564315	Network Control Formal Qualification Test Procedures	7 Oct 1987
8564370-30	Network Control Computer Program Maintenance Manual	23 Sep 1987
8564368C	TRANSEC System/Subsystem Software Specification (TRANSEC Control Processor, TRANSEC Variable Processor and TRANSEC Front End)	17 Dec 1984
8564340B	TRANSEC Control Processor Software Product Specification	15 Sep 1987
8567321A	TRANSEC Control Processor Computer Program Configuration Item FQT Procedures	11 Jul 1986
8564340-30	TRANSEC Control Processor Computer Program Maintenance Manual	14 Sep 1987
8564343A	TRANSEC Front End Software Product Specification	18 Sep 1987
8567319	TRANSEC Front End Computer Program Configuration Item Formal Qualification Test Procedures	21 Mar 1986
8564343-30	TRANSEC Front End Computer Program Maintenance Manual	11 Sep 1987
8564334B	TRANSEC Relay Station Prime Item Development Specification (including change dated 6 Nov 1987)	6 Feb 1987
8567295	TRANSEC Rekey Station Software Product Specification (Rekey Crypto Variable Controller)	25 Oct 1987

BIBLIOGRAPHY (Concluded)

8567296	TRANSEC Rekey Station Software Product Specification (Rekey Operator Machine Interface)	25 Oct 1987
8564323B	TRANSEC Rekey Station System/Subsystem Software Specification (Rekey Operator Machine Interface and Rekey Crypto Variable Controller)	30 Oct 1987
8567576	TRANSEC Rekey Station Formal Qualification Test Procedures (Rekey Operator Machine Interface and Rekey Crypto Variable Controller)	27 May 1987
8567295-30	TRANSEC Rekey Station Computer Program Maintenance Manual	24 Sep 1987
8567296-28	TRANSEC Rekey Station Users Manual	25 Sep 1987
8564341B	TRANSEC Variable Processor Software Product Specification	28 Sep 1987
8567320	TRANSEC Variable Processor Computer Program Configuration Item Formal Qualification Test Procedures	8 Dec 1986
8564341-30	TRANSEC Variable Processor Computer Program Maintenance	14 Sep 1987