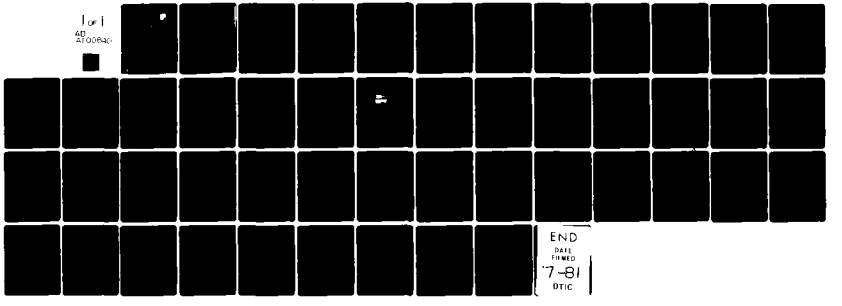
AD-A100 640 SPIRE CORP BEDFORD MA F/G 9/2
AUTOMATED TESTING OF MICROPROCESSOR CHARACTERISTICS IN RADIATIO--ETC(U)
APR 81 F L MILDER F19628-78-C-0118
UNCLASSIFIED RADC-TR-80-394 NL



AD A100640

LEVEL II

12

RADC-TR-80-394
Final Technical Report
April 1981

AUTOMATED TESTING OF MICROPROCESSOR CHARACTERISTICS IN RADIATION ENVIRONMENTS

Spire Corporation

Frederic L. Milder

DTIC
JUN 25 1981
S
D
E

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

DTIC FILE COPY

81 6 25 023

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-80-394 has been reviewed and is approved for publication.

APPROVED: Walter M Shedd

WALTER M. SHEDD
Project Engineer

APPROVED: Clarence D. Turner

CLARENCE D. TURNER
Acting Director, Solid State Sciences Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (ESR) Hanscom AFB MA 01731. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-80-394	2 GOVT ACCESSION NO. A100640	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AUTOMATED TESTING OF MICROPROCESSOR CHARACTERISTICS IN RADIATION ENVIRONMENTS		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 30 May 78 — 30 Jun 80 6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) Frederic L. Milder		8. CONTRACT OR GRANT NUMBER(s) F19628-78-C-0118
9. PERFORMING ORGANIZATION NAME AND ADDRESS Spire Corporation Patriots Park Bedford MA 01730		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62702F 46002023
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Electronic Technology (RADC/ESR) Hanscom AFB MA 01731		12. REPORT DATE April 1981 13. NUMBER OF PAGES 49
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Dr. Walter M. Shedd (RADC/ESR)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Automated test procedures
Microprocessor
Magnetic bubble memory
Automated testing hardware
Transient error detection
Portable test system

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Summary of tasks performed in compliance with contract requirements. In particular: initial results for automated test procedure for RCA CDP 1802 microprocessor; interfacing magnetic bubble memory to microcomputer and generating memory exercise procedures for radiation effects testing; setting up and testing basic elements of a centralized microprocessor automated testing facility; design and fabrication of portable, pulsed-exposure test system; adaptation of portable system to TMS 9900; software

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

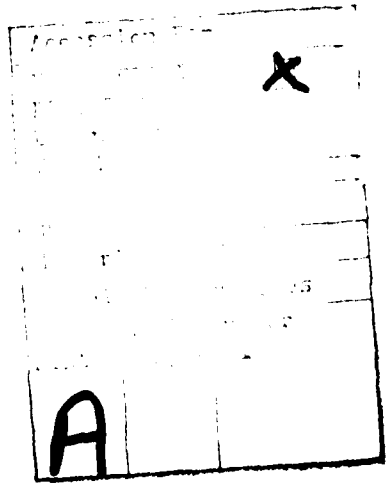
UNCLASSIFIED (Cont'd)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Item 20 (Cont'd)

development for running test system with TMS 9900.

TABLE OF CONTENTS



LIST OF FIGURES

EVALUATION

This is the final report under contract F19628-78-C-0118. The objective of this program was the development of efficient methods for the automated testing of microprocessors and other LSI circuits in radiation environments. The program has resulted in a test system capable of detecting both transient and permanent failures caused by radiation and also capable of recording the errors all under the control of a reference microprocessor. The system compares the results of the device under test with a reference device and thus is adaptable to a variety of circuits. The report covers the period from May 1978 - June 1980 and summarizes the design of the test system which is expandable up to 64 test lines. This effort was part of the radiation hardened technology program under TPO R4D.

Walter M Shedd

WALTER M. SHEDD
Project Engineer

SECTION 1
INTRODUCTION

This is the final report for contract number F19628-78-C-0118 (contract dates 30 May 1978 - 30 June 1980). The objective of this program as stated in the contract was to "perform an investigation to identify and develop efficient methods for the automated testing and characterization of radiation hardened microprocessors and integrated circuits that will be functioning as components of C^3". In this regard Spire has performed the following tasks:

1. Set up, interfaced and debugged a Texas Instruments 92K magnetic bubble memory unit for use with a TMS 9900/100M microcomputer;

2. Set up, tested, expanded and interfaced as necessary, major components for the development of radiation testing programs and procedures of three microprocessors which were of interest: RCA 1802, TI 9900 and Motorola 6800;

3. Developed test procedures based on limited, reasonable coverage in a specific logical sequence designed to test microprocessors for failure modes (after radiation damage) and identify the problem areas in the hardware.

4. Designed, built and partially debugged a microprocessor test unit which is capable of detecting and recording both transient and "permanent" upset errors in operating processors during either pulsed or continuous radiation exposure (8 channels currently, expandable to 64).

5. Designed and built the necessary interface hardware and developed the necessary software codes to enable a test of the TMS 9900 microprocessor on the above system.

Section 2 of this report covers the concepts developed in the course of the contract and explains how they contribute to the overall picture of the radiation testing of microprocessors and other LSI circuits. Section 3 summarizes the work performed during the first year of the contract. Section 4 reviews the past year. Finally, Section 5, contains recommendations for the directions of future work.

SECTION 2
TESTING CONCEPTS

The approach taken to the radiation hardness testing of microprocessors and LSI circuits was three pronged (see Figure 1). At the most basic level, to test any LSI circuit properly, no matter what the procedures, a familiarity with that circuit must be developed. This was most easily done through working with the circuits in question. Thus, one aspect of the contract was centered around setting up, interfacing, evaluating, expanding and becoming generally familiar with the following components:

RCA COSMAC development/evaluation system
RCA CDP 1802 based microcomputer
Tektronix 4024 display terminal
RCA floppy disk memory
Texas Instruments TMS 9900/100M microcomputer
Texas Instruments TMS 990/302 development system
Motorola 6800 based microcomputer
Tektronix S-3260 LSI test system
Texas Instruments 92K bubble memory (BKA 0203A)

The second prong of the approach was to develop the concept for testing procedures capable of finding faults in the hardware of functioning microprocessors induced by radiation damage. These procedures were to use the Tektronix S-3260 system as the testing apparatus. The concept developed was to examine the processor, one functioning area at a time, e.g. data bus buffer and data bus, then the data register, then the scratch and register, etc. Thus, for example, in checking the data register, one would be assured that any error was indeed from that register and not from the bus. In this concept one would basically "walk through" the processor from the outside in, that is, from the most I/O oriented hardware to the most internal registers and data paths. After identifying this technique as desirable, a program flow chart to partially test the RCA CDP 1802 microprocessor was developed to demonstrate the concept. The above work, except for that dealing with the TMS 9900/100M and the TMS 990/302, was performed during the first year of the contract. The second year was spent on the concepts and work described below.

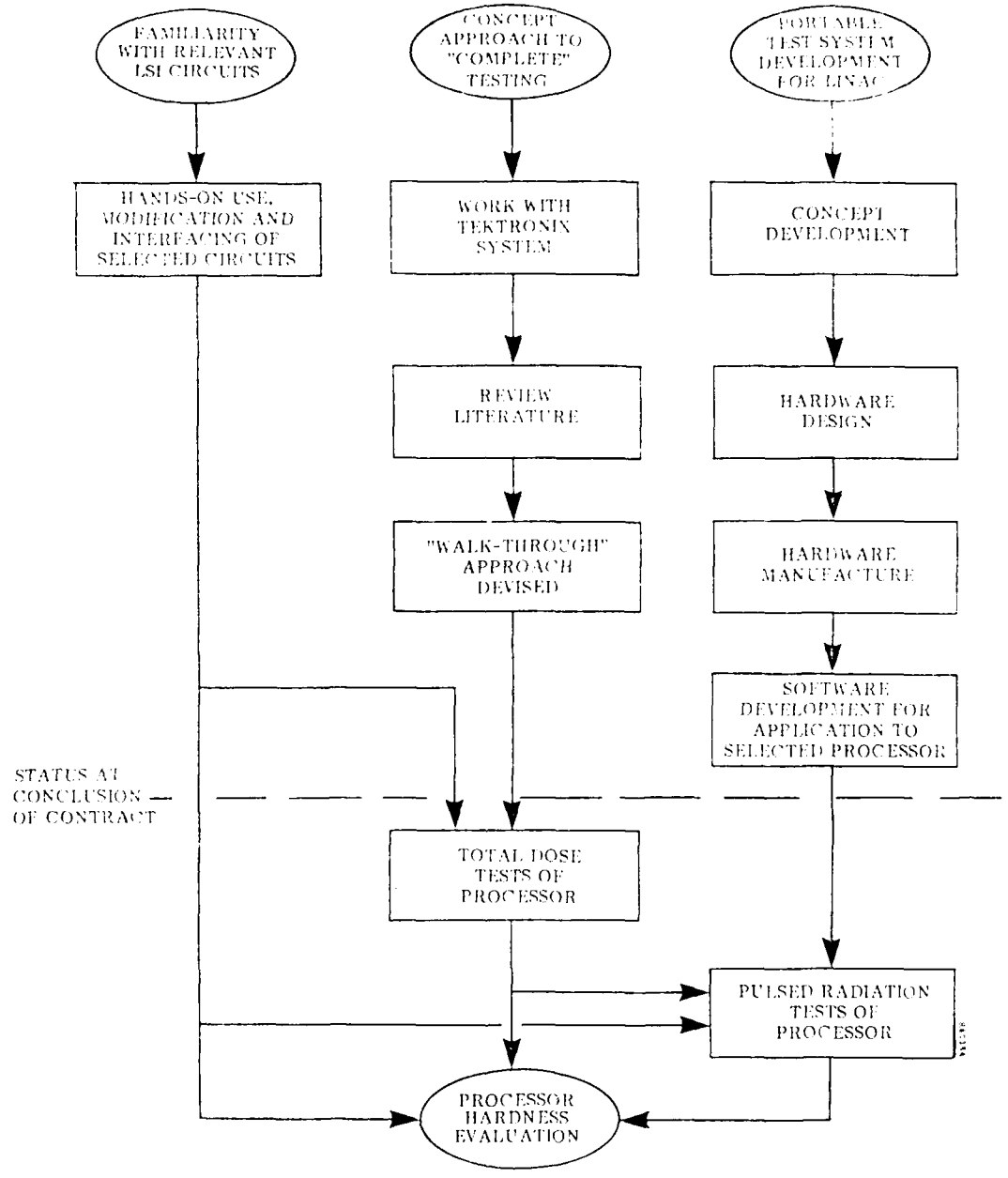


FIGURE 1. THREE-PRONGED APPROACH TO RADIATION TESTING OF MICROPROCESSORS

Since the S-3260 system is rather large and immobile, the above test procedure only has utility for permanent damage detection. That is, it is valuable for total dose failure tests wherein the device can be tested, exposed to radiation of a given quantity and then tested again. It is not useful for transient error detection or bit errors which do not involve permanent latchup, etc. Neither is it suited for testing for failures during pulsed radiation exposure, since at present it has no hookup to the LINAC test area. For this reason a third prong to the total testing approach was necessary.

To test a microprocessor in the pulsed environment at the LINAC, a portable hardware system capable of detecting and recording errors had to be developed. The design used is based on comparing a device under test (DUT) to a reference device (RD), pin for pin. Since the RD had to be a microprocessor itself, and since such a test system would naturally be run by some sort of smart logic, it was decided to use the RD as the controlling device as well. Thus, a comparator system was built, one which could compare 8 pin pairs (expandable up to 64) and detect logic errors transient errors, and parametric failures. However, the comparator system itself is not smart. Rather, the RD is incorporated into a microcomputer which controls the comparator system and provides the test sequence for the DUT as well. This concept is depicted in Figure 2. Included in this design is a good degree of flexibility in the comparator system. Thus, for each microprocessor type to be tested, the comparator system hardware is modified slightly (mostly cabling), and a new RD/microcomputer and DUT are integrated into the system. The comparator system, as described, was designed and built, and a TMS 9900 RD and DUT were chosen for the first testing. Debugging of the entire integrated system has not been completed to this date.

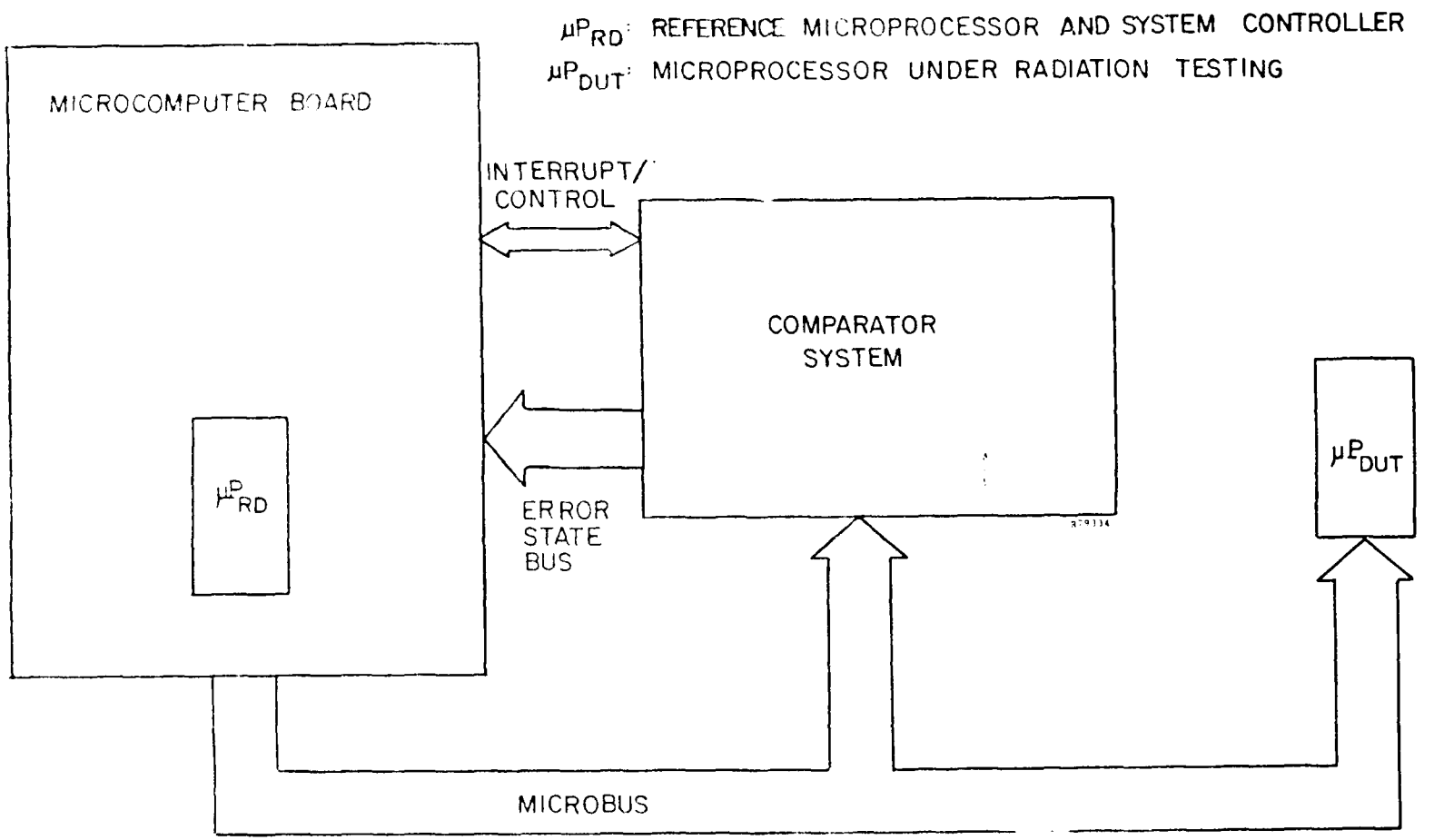


FIGURE 1. MICROPROCESSOR TEST SYSTEM CONCEPT USING REFERENCE DEVICE AS CONTROLLER

SECTION 3
REVIEW OF FIRST YEAR OF CONTRACT

As previously stated, the first year of the contract was spent in two endeavors. One was to gain familiarity with the detailed workings of several microprocessors, the Tektronix S-3260 system and a bubble memory through a hands-on development of the hardware and software. The details of this process need not be included here, since an insignificant amount of new knowledge was produced in this regard, although many minor problems had to be overcome in the area of component interfacing. This work is generally reviewed in more detail in AR-10059.

The second major endeavor was to define procedures to systematically test a microprocessor (RCA CDP 1802 in particular), and identify malfunctioning regions of the processor in a "comprehensive style" test (i.e., not exhaustive, but complete). The test was supposed to take a reasonable length of time on the S-3260 system. It was decided, through literature research and discussions, that it was best to use methods which rely heavily on concepts like commonality of data paths and of functions and subfunctions. For instance, if two registers of an on-chip register array can be shown to be able to be loaded (separately, via a common internal data bus) into the accumulator, and if data from one register can be shown to be properly added to the contents of the accumulator, then the proper addition of data from the second register into the accumulator need not be explicitly tested.

The subfunction approach recognizes the fact that execution of any of the instructions of the microprocessor repertoire involves sequential and parallel usage of some subset of the relatively small number of intermediate-level functional blocks or subfunctions. Typical of these functional blocks are, for example, address and data buffers, the ALU, scratch pad registers, the instruction register and decoder, the accumulator register, etc. Thorough testing of some register, for instance, by multiple, varied usage of any one instruction, and then testing all over again using another instruction, is an example of the high redundancy of the "exhaustive" approach, which can be readily eliminated.

The thorough testing of a register array brings up two further considerations concerning simplification. Ideally, a register array should be tested with all possible

(i.e. 2^{16} = 65,536) words, one register at a time (thereby ignoring possible, but hopefully very slight, interregister interactions). One simplifying assumption is that in a particular class (i.e. on-chip scratch pad) of registers, one is the same as any other as regards radiation effects and hardness. Accordingly, only one register of each class need by thoroughly tested. If different registers (or sets of registers) reside at different places on the chip, long access paths to some but not other registers may vitiate the assumption of identical registers.

A second, more extensive simplification concerns the ideal 65K-word test of one or more of the registers. Semiconductor memory manufacturers are faced with the same problem and resort to testing with various "bit patterns". Among the more than 100 such patterns developed are all one/all zeros, checkerboard, walking ones, surround/disturb, etc. A dozen or so well-chosen test words should provide fairly complete coverage. Added to the above concepts is the additional comment that the most reasonable approach in any one processor would be to start with the functional blocks which are closest to the I/O paths and to work "inward" from there.

As an example of these procedures, a partial test of the RCA CDP 1802 microprocessor which was specifically oriented toward execution on the Tektronix S-3260 automated LSI test system was flow charted. It is in the form of a "dialog" between the microprocessor and the LSI tester, typically in the form:

1. Microprocessor requests next instruction
2. S-3260 checks levels and supplies the next instruction
3. Microprocessor inputs this "fetched" instruction, decodes it and executes it
4. S-3260 supplies requisite "memory requests" and tests resulting output levels.

Preparation for generating this testing procedure involved analysis of available technical literature on the 1802 internal architecture — to establish data paths, major registers and functions, any temporary or "hidden" registers or "transparent buffers", etc. — as well as the explicit details of the complete instruction set, so as to determine what can be or must be determined, and in what order.

The resulting test procedure was included as Appendix I of the annual report AR-10059.

SECTION 4
REVIEW OF SECOND YEAR OF CONTRACT

One major effort during the second year concerned the conceptual design, detail design and fabrication of the portable, self-contained system capable of detecting and recording transient upsets in microprocessors. The system concept is to compare a device under test (DUT) to a reference device (RD) which is simultaneously controlling the test system. As the system concepts developed, the block design of the system changed from that given in Figure 2 to that shown in Figure 3. The more detailed design of Figure 3 and the design of the comparator system shown in Figure 4 contain provisions for the following characteristics: (1) parametric error detectors are included which will detect transient upsets or output level errors; (2) the reference microprocessor is used as both the reference device and the controller of the test system; (3) the system records each error as it occurs, resynchronizes the DUT and RD and proceeds with further testing; (4) all pins of the DUT are tested; (5) the test instruction sequence is contained in either ROM, RAM or both, and is thus completely flexible and not limited in length.

The main hardware hurdle was the buffering of the microprocessors in such a way that the reference microprocessor could serve both as the RD and as the system controller. This was accomplished by having a bidirectional buffer for the RD and a unidirectional buffer for the DUT. (See Figure 3.) The electrical termination of the two devices is, however, equivalent. Also, the problem of resynchronizing the microprocessors after an error occurrence without having to reenter a monitor program had to be dealt with. This was achieved in a microprocessor-independent fashion by the combined use of custom hardware and a well-defined software approach. The solutions to the above problems and details of the system will be discussed in the next subsection 4.1. Subsection 4.2 deals with the adaptation of the test system to the intended radiation testing of a TMS 9900 microprocessor through software. The test system control software developed and written (in assembly language code) for the 9900 was the second area of major effort during the last year of the contract.

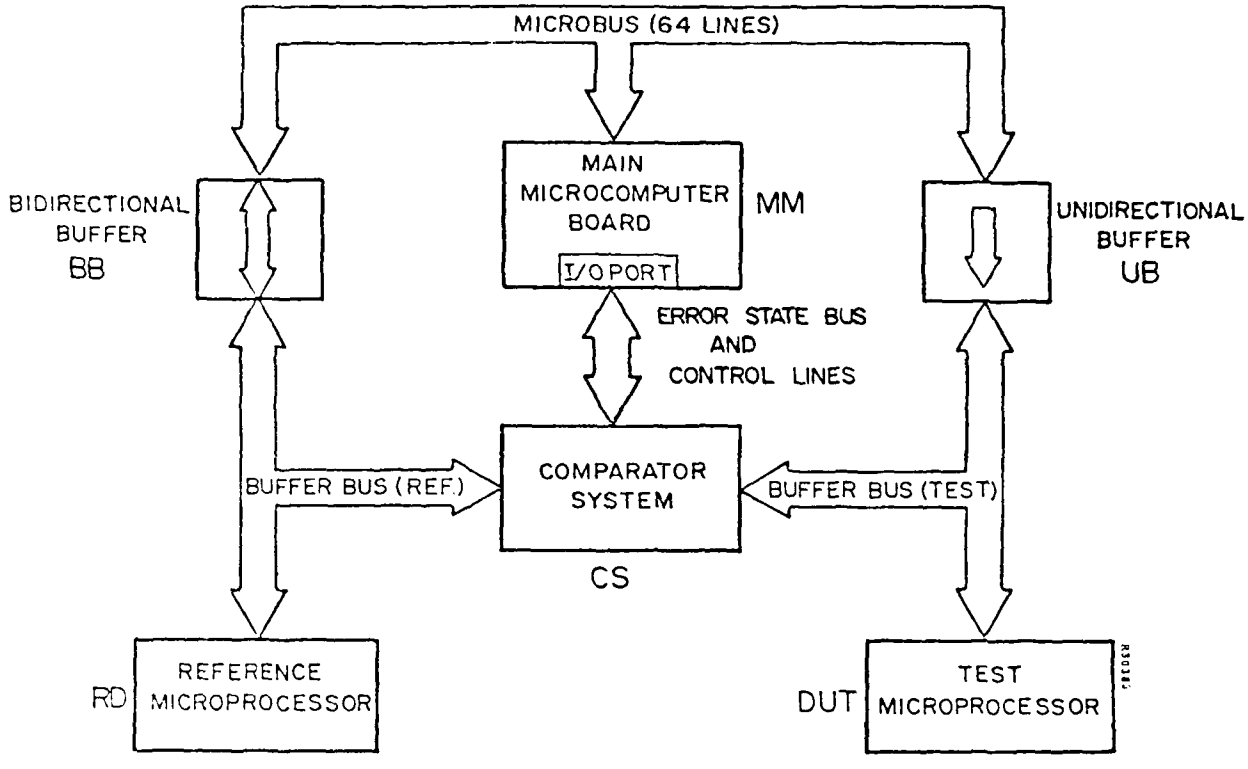


FIGURE 3. MICROPROCESSOR TEST SYSTEM BLOCK DIAGRAM

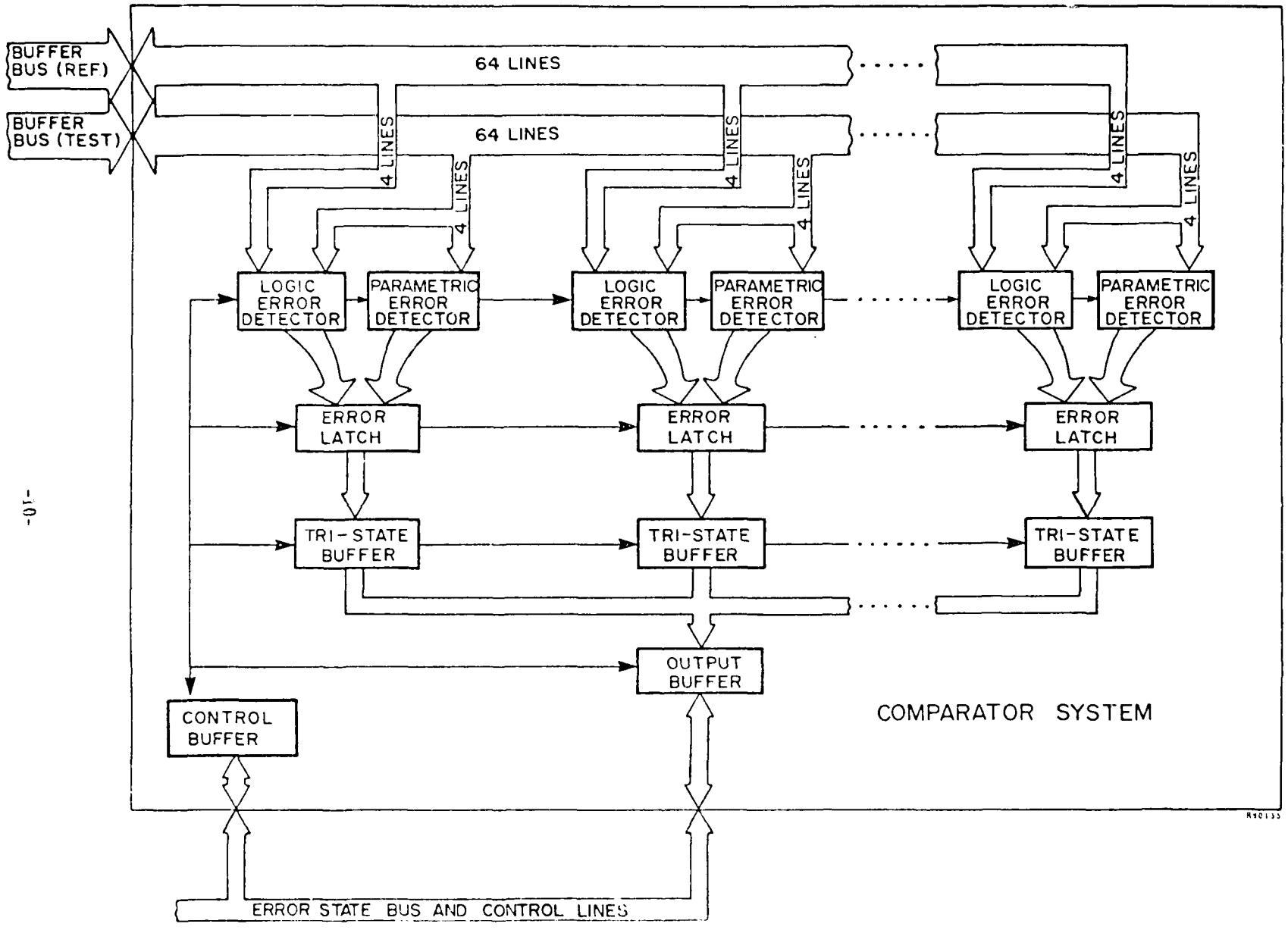


FIGURE 3. BLOCK DIAGRAM OF THE COMPARATOR SYSTEM

4.1 THE MICROPROCESSOR TEST SYSTEM HARDWARE

The hardware which comprises the microprocessor test system consists of six major elements: comparator system, reference microprocessor, test microprocessor, bidirectional buffer, unidirection buffer and main microcomputer board. Figure 5 shows this hardware, built for the system under this contract. Let us group these units into three functional groups. The first group, the RD, the bidirectional buffer (BB) and the main board (MM), make up a functioning microcomputer. Peripherals, support boards, additional memory, etc., can be added to the MM as desired in the usual fashion. The buffer is transparent to both the RD and the MM. This complete microcomputer is the controlling device for the system. The memory ROM and RAM contain (1) the monitor and monitor support codes, (2) the program codes for running the test system and (3) the actual codes designed to exercise the DUT microprocessor. The cables connecting these components are all 64 lines, so that any microcomputer on the market today can be handled. To convert the buffer from use with one microprocessor to use with another, a few control lines on the BB must be reconfigured. These lines control the breakup of the total 64 lines into input, output, power and I/O line groups, which of course change from processor to processor. The present BB is configured for a TMS 9900, since this was the major processor of interest for testing. The circuit diagram for the bidirectional buffer (Buffer Board A) and cables are included in Appendix I.

The unidirectional buffer (UB) and the DUT make up the second function group of the system. The UB is transparent for the DUT receiving input from the MM and also terminates the DUT output. Because the UB is one-way, however, the MM never receives any signals from the DUT. From the standpoint of the MM, the UB looks like a constant load. Thus, the DUT receives instructions, memory fetches, etc., from the MM in synchronization with the RD, but its outputs are ignored by the test system (except in the comparator). The UB is included in Appendix I as Buffer Board B.

The third functional group of the total system is comprised of the comparator system circuits. The block diagram was shown in Figure 4. As the DUT and the RD are functioning in parallel, the comparator system is comparing their outputs, pin for pin. The present system is built up for 8 lines only, but the design includes expansion to 64 lines. For a DUT response to be considered correct, it must pass two tests. First, it must be logically equal to the RD response. Thus, each line in the DUT is logically

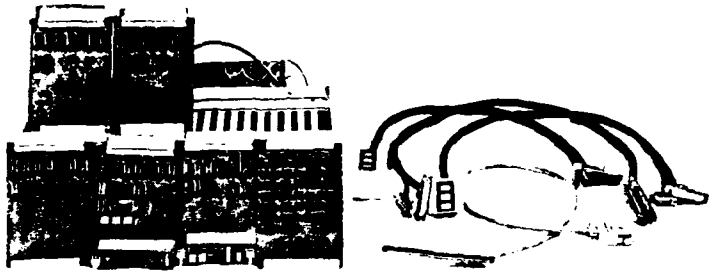
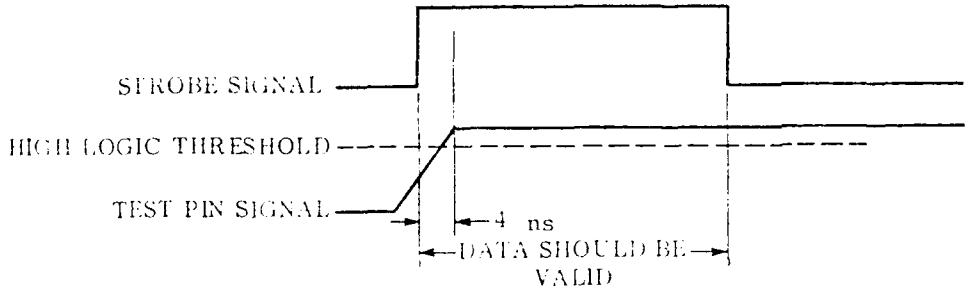


FIGURE 5. TEST SYSTEM HARDWARE: Power Supply, Card Cage With Backpanel Wiring, Cables, Processor Test Box, and Comparator, Buffer and Control Boards

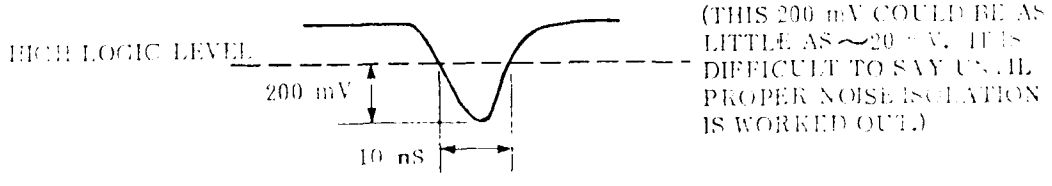
compared to the identical line in the RD. If the two are not in the same logic state for the entire time for which the data is supposed to be valid, a logic error is flagged for that line. The second test that must be passed is a parametric test. The logic level must be within certain voltage specifications for the entire data valid period for that line to pass its parametric error test. If not, a level error is declared for that line.

The possible combinations of level and logic errors which can be detected are as follows:

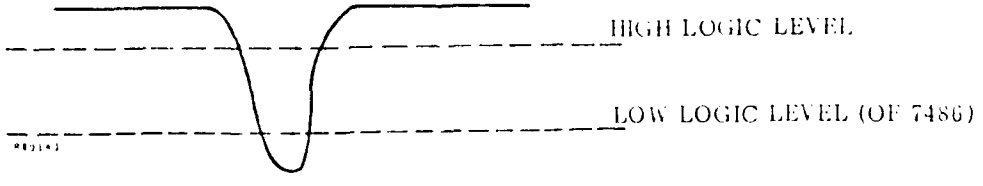
A. System will detect ±4 ns slew on normal logic pulse; e.g., if logic (high or low) is not present within 4 ns of edges of strobe, system declares a level error.



B. System will declare a level error if a "glitch" of 200 mV with 10 ns basewidth is present.

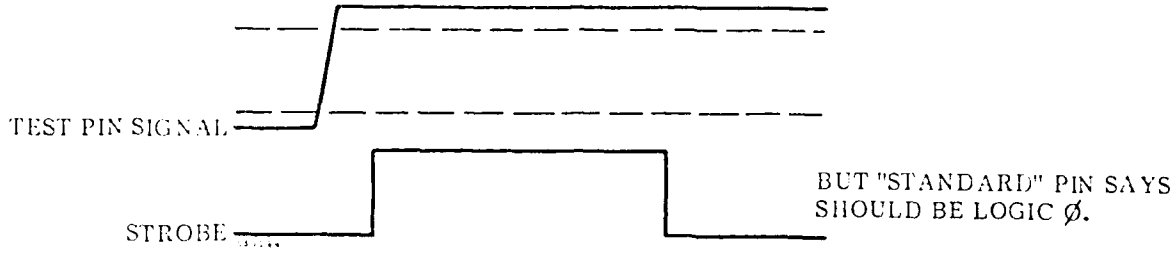


C. System declares level error and logic error if "glitch" reaches the opposite logic state.



D. System will declare a level error for a logic signal deviating from the valid regions by >20 mV for a large fraction of the strobe width.

E. System will declare a logic error if logic level is valid for the entire strobe width but is in disagreement with the "standard" logic state.



Once either a logic or a level error is flagged by the error detectors, that error condition and the error conditions of all the other lines being tested is latched, and a signal is sent to the control logic. The control logic then sends an interrupt to the microcomputer signalling that an error has occurred. At this point, all further signals to the comparator system are ignored, except control signals coming from the microcomputer.

The response of the computer is a set of sequential signals which turns on the tri-state buffers one at a time, along with the output buffer, so that the latched condition at the time of the error is placed on the I/O bus (for the acceptance by the computer) in one byte segments. Then the computer resynchronizes the processors (to be discussed), resets the comparator system and continues testing as desired. The entire circuits for the error detectors, latches and buffers for eight lines and the control functions are included in Appendix I as the comparator and control boards. The expansion to more pins is simply a matter of copying the detector/latch/buffer circuits as necessary and tying into the control logic (which is only a few lines).

Synchronization of the microprocessors at the initial powerup and after error detection proved to be a major difficulty. First of all, the TMS 9904 four-phase clock was not capable of driving two 9900 processors. Therefore, one 9904 was used to drive two additional 9904 clocks, one for each processor. This is shown in Figure 6. This is all right, except at powerup. After powerup, the relative phases of the two processor-driving 9904's is arbitrary. The two clocks must be brought into

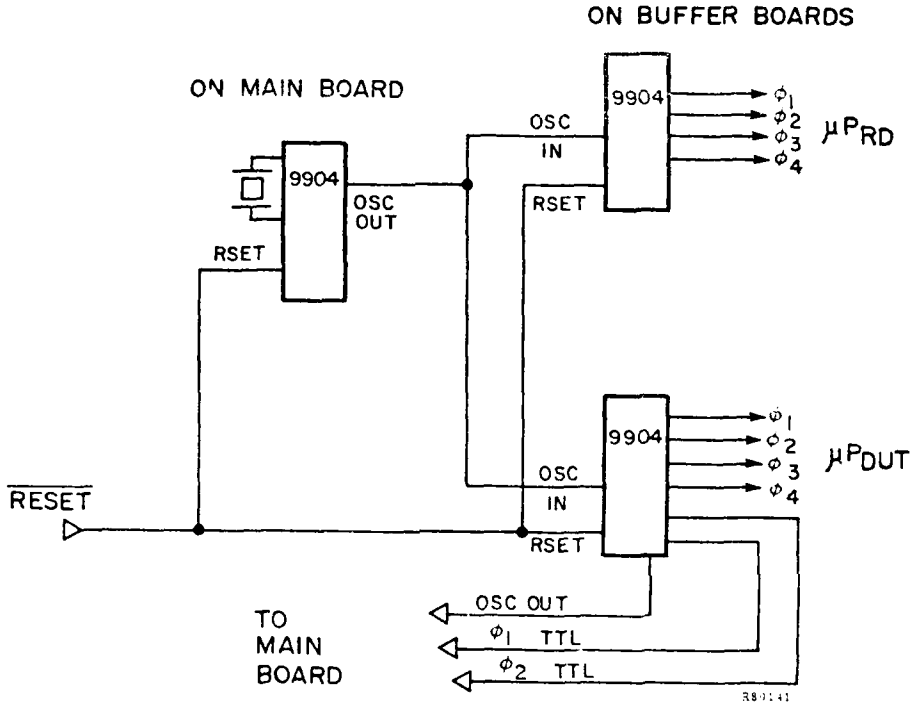


FIGURE 6. SOLUTION TO SEPARATE, SYNCHRONIZED CLOCK DRIVERS FOR THE RD AND THE DUT

synchronization. This problem was solved by having a gate between the primary clocks and the secondary clocks. Figure 7 shows the block diagram. The one-shots are adjusted so that every time the momentary closure is pushed, one clock pulse is removed from the OSC IN line going to one of the secondary clocks. This shifts the relative phase by 90^{o}. The synchronization is monitored by an LED, which turns off when the clocks are in phase. The circuit for the powerup synchronization is included in Appendix I, on the buffer boards.

The second part of the synchronization problem arises when the DUT makes an error. After this occurs, the DUT is in an unknown state. In order to resynchronize the processors, a reset must be used, since on most processors a reset signal is the only one which is always responded to on the next phase-one clock pulse, no matter what the processor status. However, upon reset, one does not want to go back to a monitor; one wants to continue testing. The solution was to have the processor initiate its own reset and simultaneously set a hardware flag on the control board. It then of course responds to its reset, but before it goes to the monitor, it checks the flag. If this flag is set, it does not go to the monitor, but rather, continues testing from the point of interrupt. The circuitry for this part of the control system is also included in Appendix I, on the control board.

More discussion on the details of the subunits is included in various quarterly reports.

4.2 SOFTWARE FOR THE TESTING OF THE TMS 9900

The software for the error detection, system control, error storage and test sequencing for the TMS 9900 was written during the latter part of the year, using the 990/302 development system. The code consists of four modules (SETUP, SYNC, ESCAPE, ERROR), the message list for I/O and an addition to the resident moniter called PSMON. The purposes of the modules are as follows:

SETUP - Does all of the initialization of the program: sets up workspaces, enters branch instructions into the interrupt vector locations, sets aside a stack in RAM for storage of the error conditions of the DUT and handles the proper entry or reentry into the test code. (The test code is that code, yet to be written, which will be the actual sequence of instructions that the DUT is being tested with during exposure.)

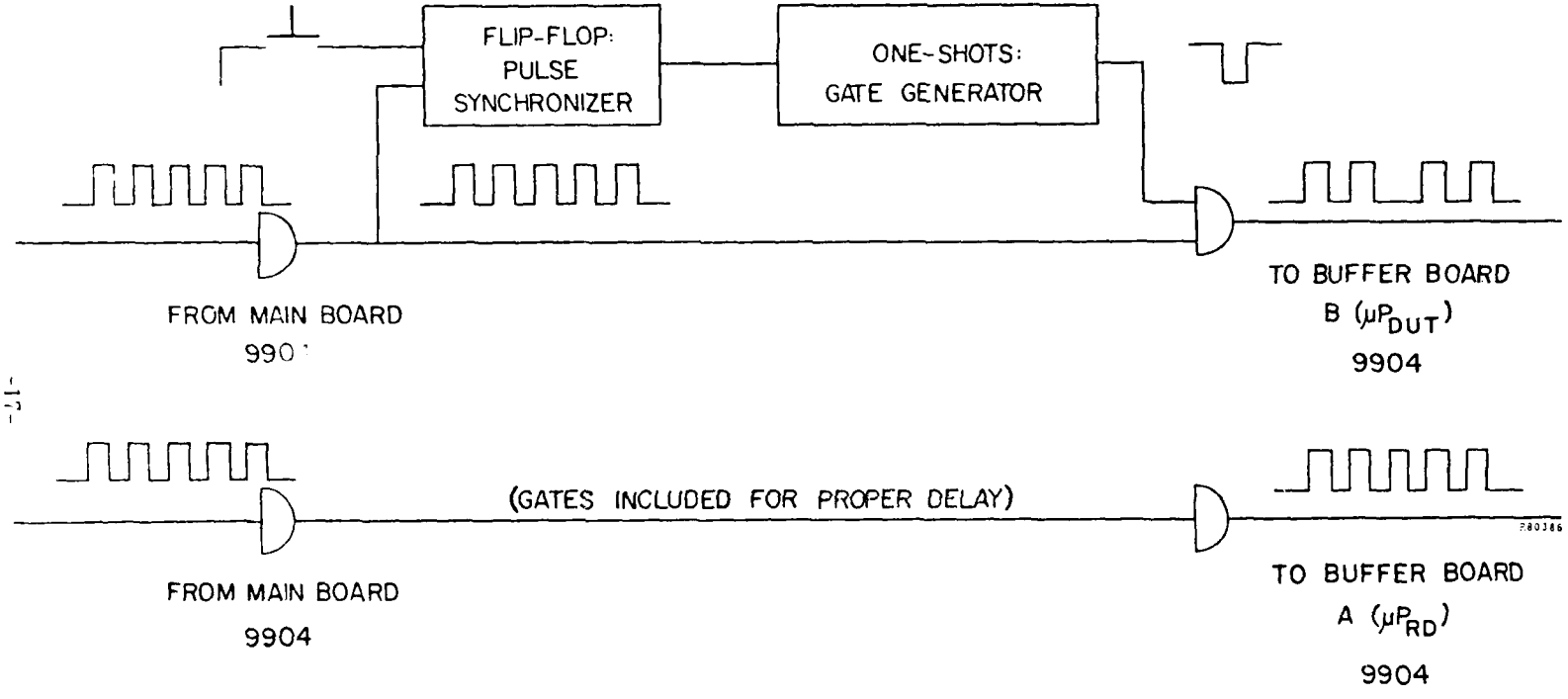


FIGURE 7. POWER-UP SYCHRONIZATION
A momentary connection removes one pulse from the OSC IN of the DUT 9904 clock. This shifts the phase by 90°. (The 9904 is a four phase clock.)

SYNC - Triggers the synchronization reset and setting of the self-reset flag. Upon return from the reset the program returns to the test code at the point of the error interrupt.

ESCAPE - Allows the escape from test code during execution: Hitting the ESC key causes a return to the monitor.

ERROR - Receives the interrupt when a DUT error occurs. It turns off the comparator system, stores the error condition one byte at a time on the error stack, tests for overrunning the allowed stack space, resynchronizes the processors through use of the SYNC module and returns.

The PSMON addition to the resident monitor precedes the first instructions normally executed by the monitor. This segment of code tests the flag which is set when the software issued reset is used. If the flag is set, the execution is transferred to the SYNC reentry point. If the flag is not set, execution is continued at the top of the usual monitor. This code addition is to be resident in the monitor EPROM (low addresses).

The memory map of the microcomputer for radiation testing is shown in Figure 8. The assembly language codes for the above modules are included as Appendix II.

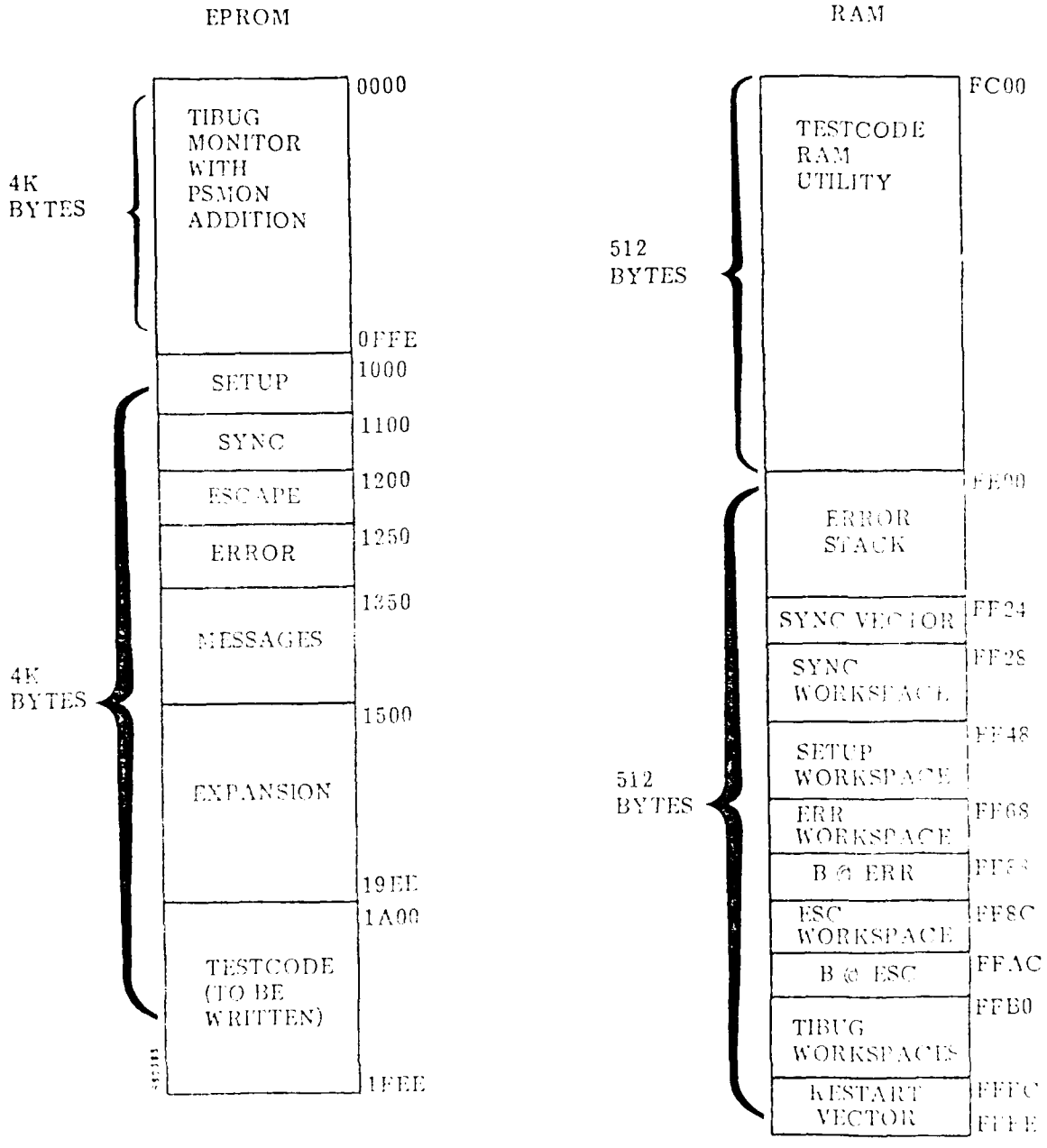


FIGURE 8. MEMORY MAP OF 990/100M DURING RADIATION TESTING

SECTION 5
RECOMMENDATIONS FOR FUTURE WORK

5.1 STATUS AT THE CONCLUSION OF THE CURRENT CONTRACT

The status of the test system hardware and software at the conclusion of the contract is as follows:

Hardware - All circuits for the testing of eight lines of a processor have been built. The adaptation of the buffer boards for the TMS 9900 has been completed, including the clock driver circuits. Cables for the 9900 board connectors have been fabricated. Board modifications of the 990/100M computer to accommodate the I/O, control and inclusion of the PSMON have been made. The monitor has been moved from 2708 EPROM's to 2716 EPROM's to allow for EPROM expansion with the relevant codes. The hardware is only partially debugged.

Software - All modules mentioned have been written and debugged. Further testing must await the complete debugging of the hardware because certain functions can only be tested with real I/O or with hardware emulation. The actual test code has not been written, and EPROM's have not been burned.

Summarizing, to bring the total system on line for testing eight lines of a TMS 9900 (four data lines, bidirectional; four address lines, output), the hardware must be debugged as a system, the software must be tested with the real hardware and the codes must be burned into EPROM.

5.2 FUTURE EFFORTS

Once the system is checked out, the next major task would be the writing of the test code for the actual radiation testing. This could be approached from two possible directions. One would be to follow the concepts devised in this contract of "walking through" the processor from the outside. To do this, one would have to become more familiar with the 9900 and to go through a careful study of the interal architecture. Alternatively, one could first take a more modest approach of defining small execution

loops using only a few instructions at a time and testing these loops individually. This technique has been successfully adopted by Tom Ellis of the Naval Weapons Support Center (IEEE Trans. NS-26, 4735 (1979)). Either way, with the completion of the system developed under this contract, the combination of being able to do parametric testing as well as logic testing during pulsed radiation expsoure should be a valuable and unique capability.

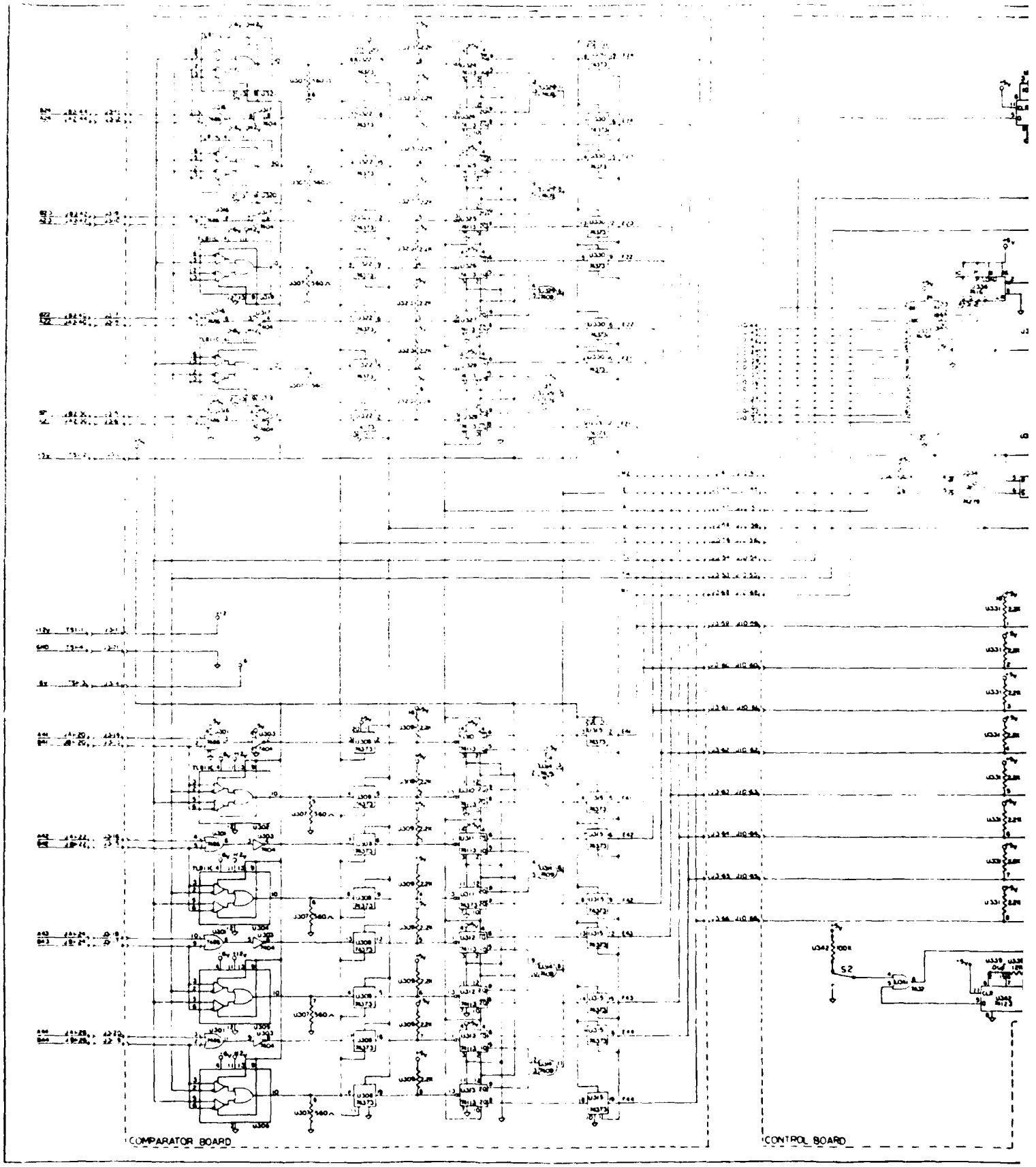
Further efforts should include work on the expansion of the system to at least 32 lines and intensive testing on the system susceptibility to interference from the LINAC. Additionally, one would want to be working on the software and hardware modifications necessary to test other processors as well as the minor modifications to the I^{2}L version of the 9900 (SBP 9900) and, of course, to actually go ahead with radiation testing.

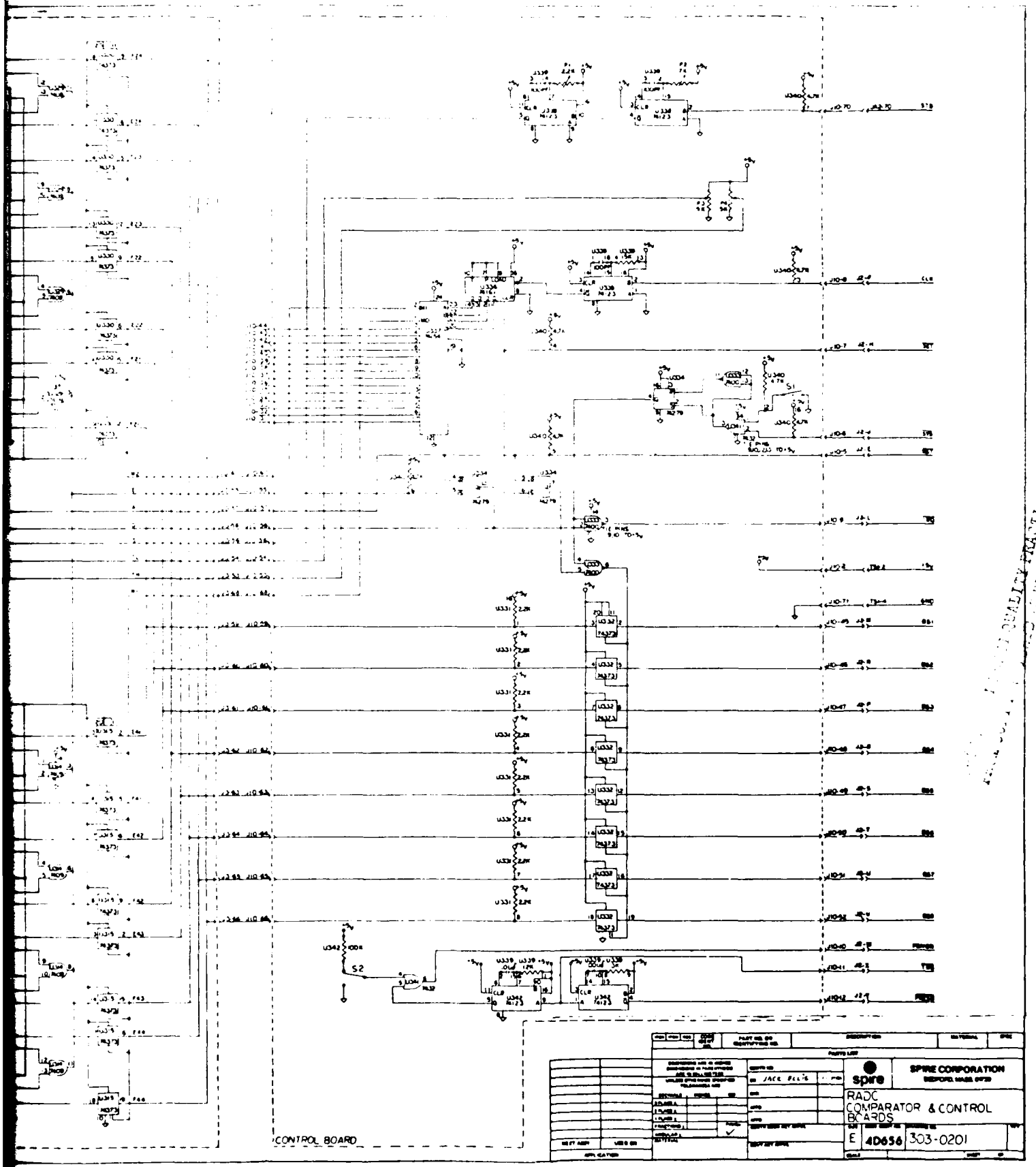
APPENDIX I

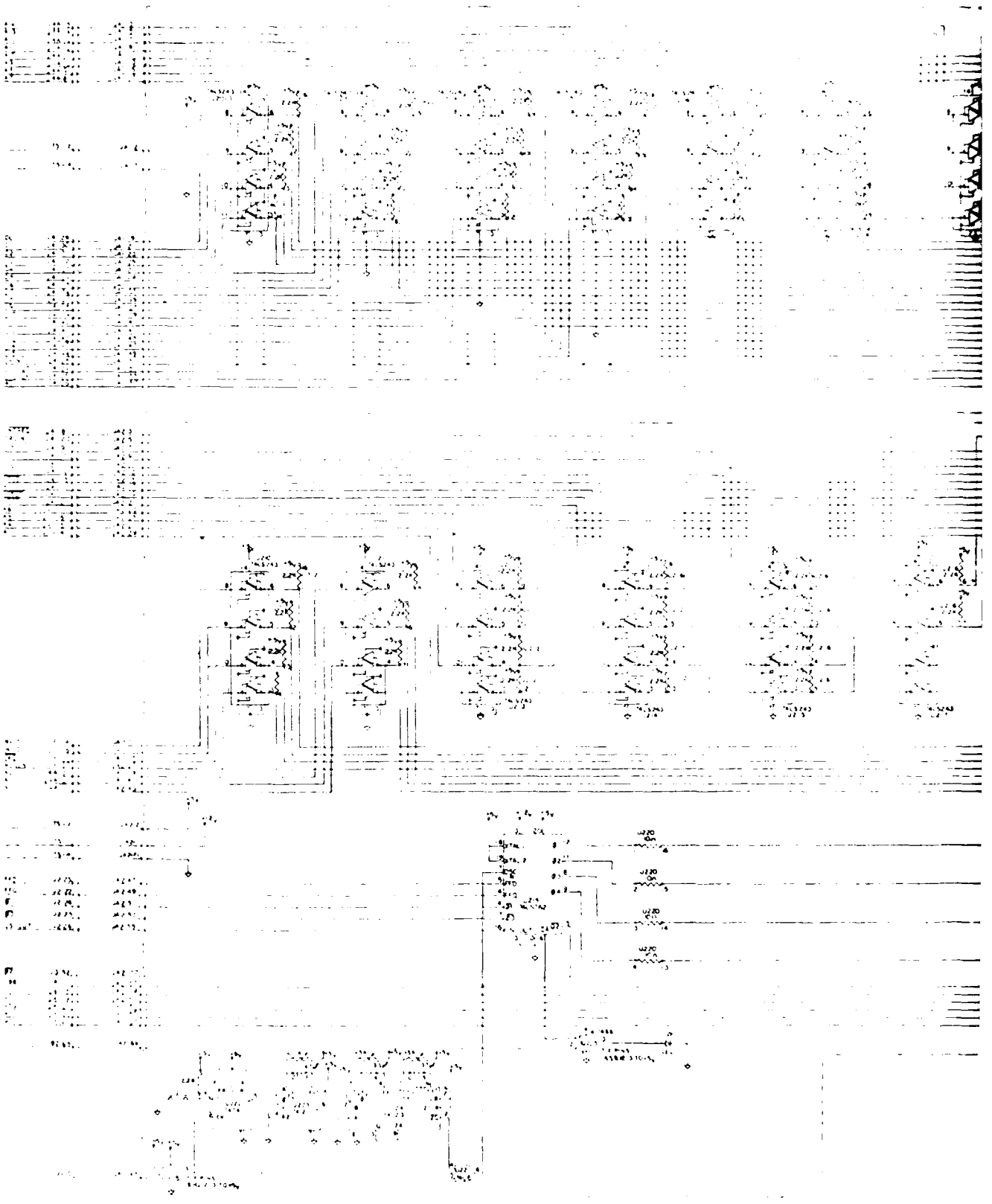
CIRCUIT DIAGRAMS FOR THE

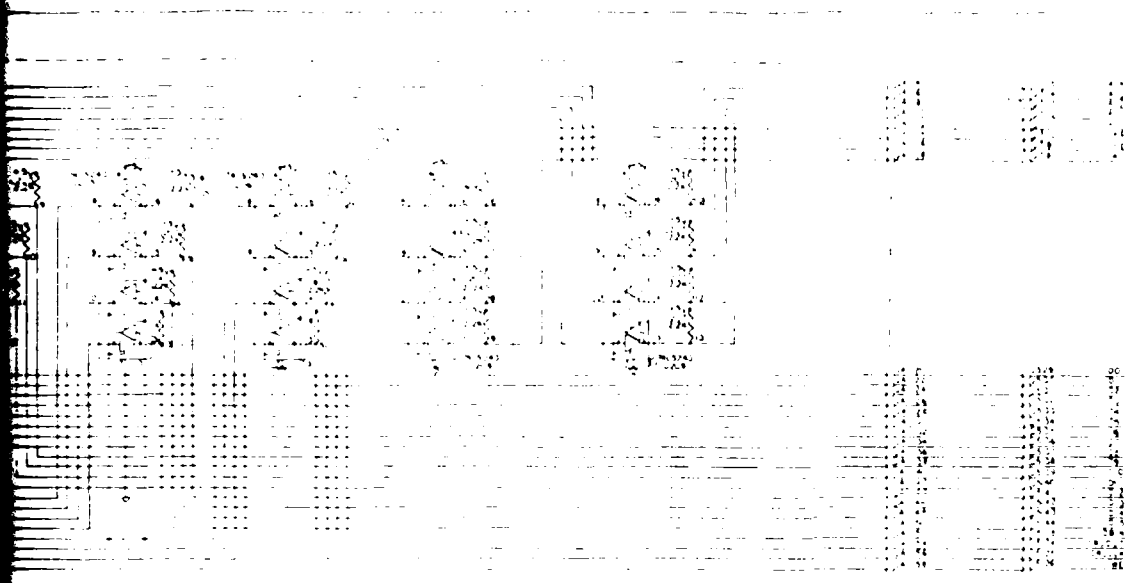
MICROPROCESSOR TEST SYSTEM

(Including adaptation for the TMS 9900)

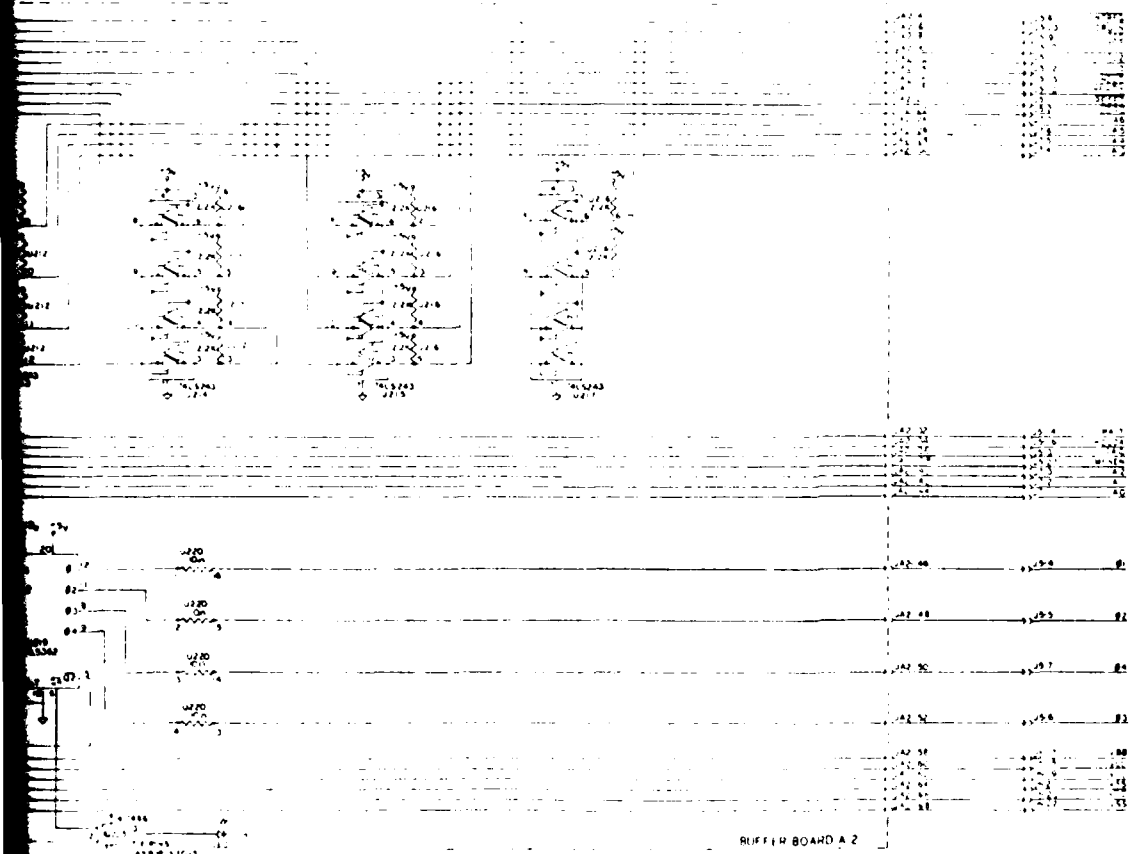

COMPARATOR BOARD
CONTROL BOARD


CONTROL BOARD
S1
S2
SPIRE CORPORATION
RADC
COMPARATOR & CONTROL
BOARDS
E
4D656
303-0201





BUFFER BOARD A



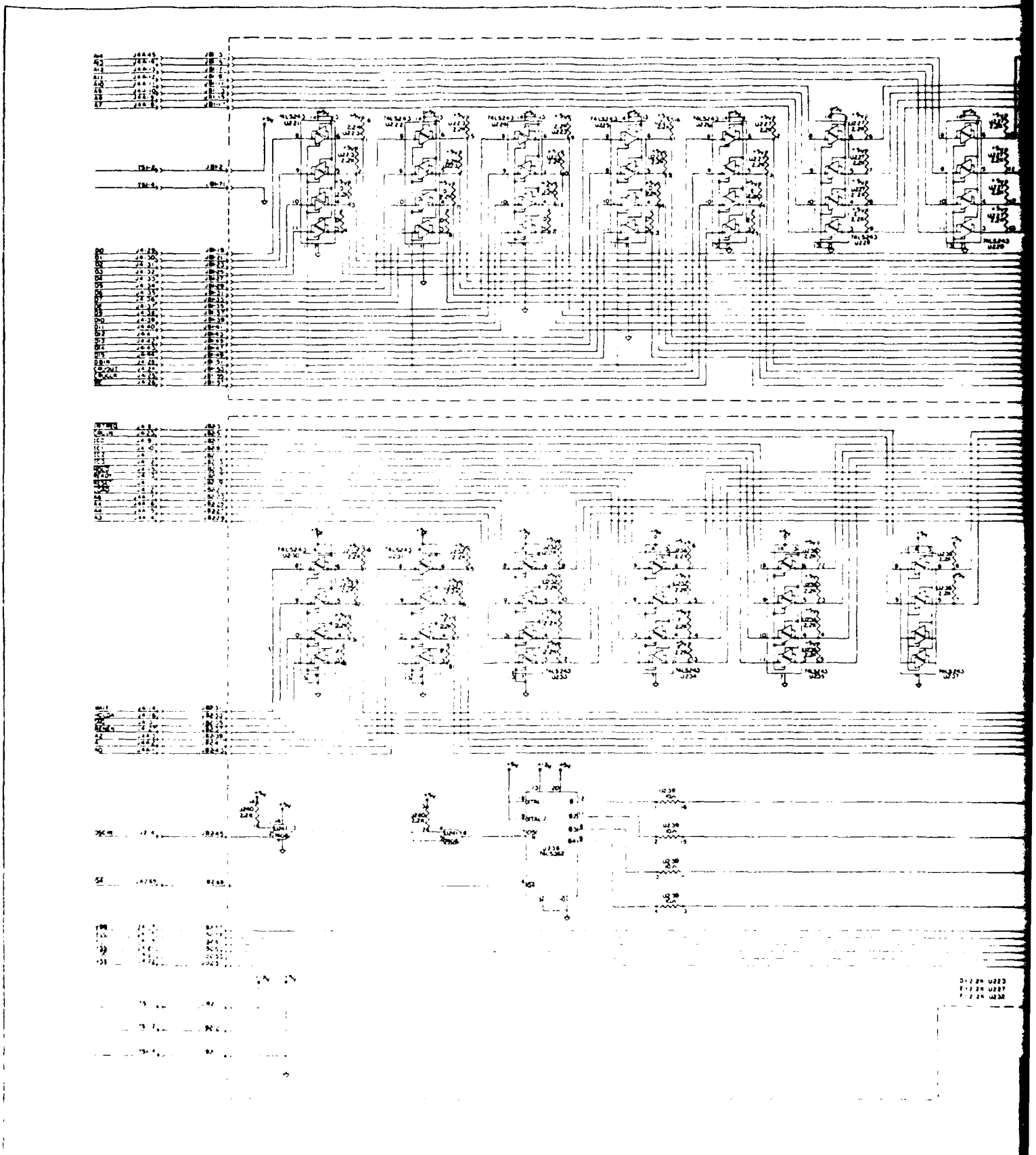
BUFFER BOARD A 2

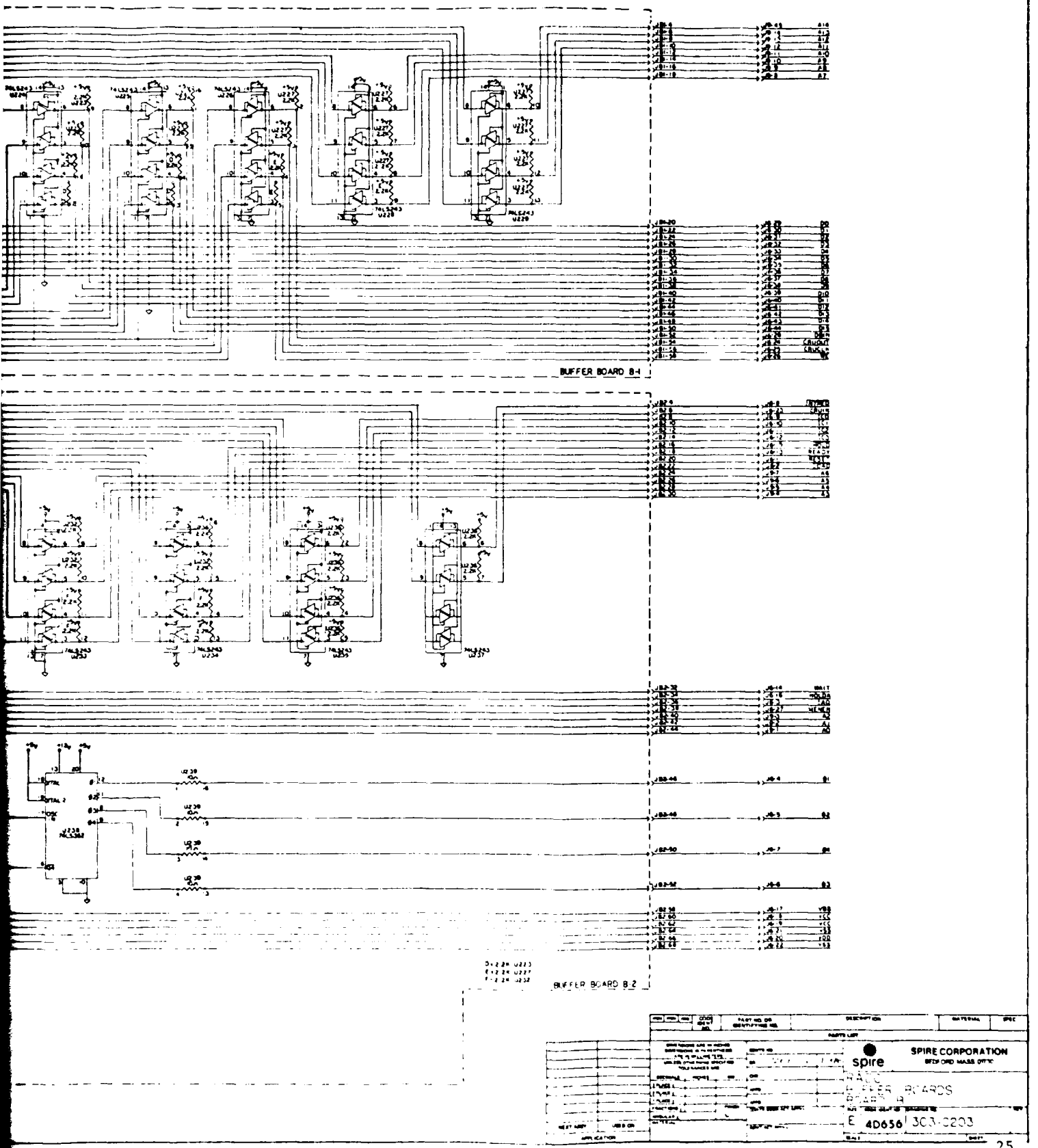
SPIRE CORPORATION

spire

BUFFER BOARDS

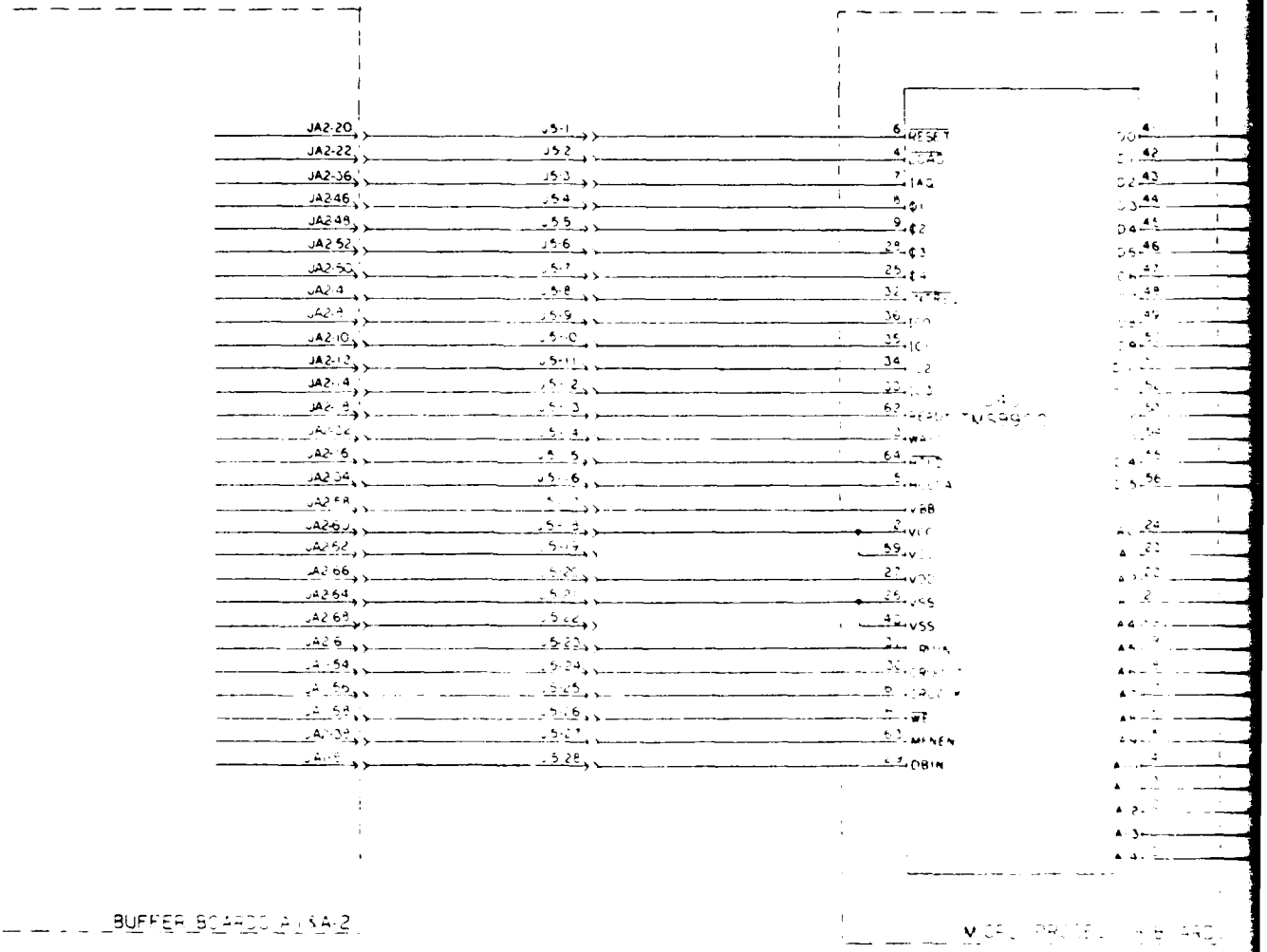
4D656 303 0202



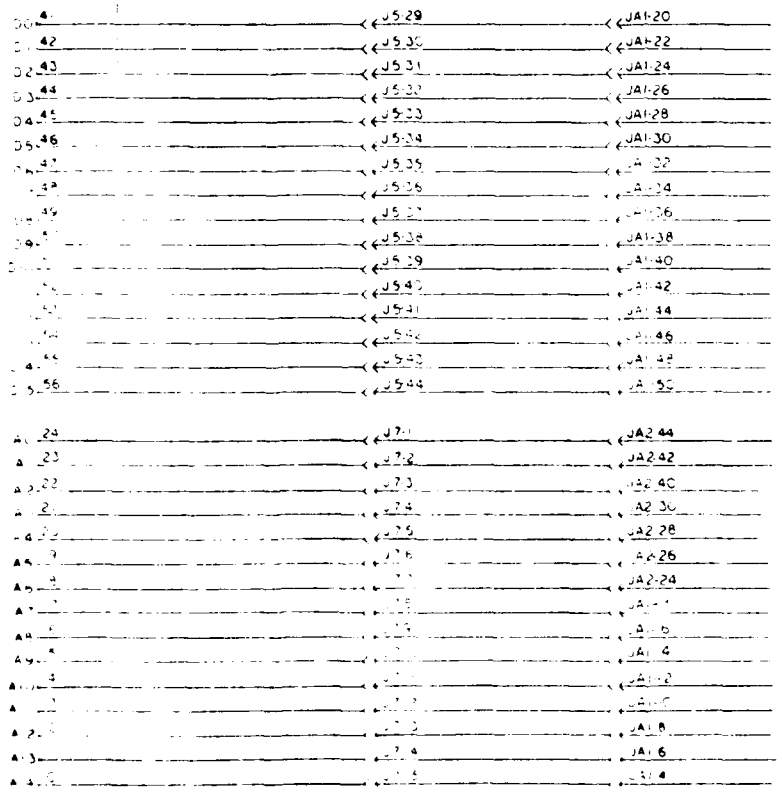

BUFFER BOARD B-1
BUFFER BOARD B-2
SPIRE CORPORATION
BUFFER BOARDS
4D656
303-0203

The information disclosed herein was originated by and is the property of Computer Products, Inc. and except as otherwise expressly granted to the United States Government Computer Products, Inc. reserves all patent, proprietary, design, use, sale, manufacturing and reproduction rights thereto.

NOTES



REVISIONS				
ZONE	LTR	DESCRIPTION	DATE	APPROVED



BUFFER BOARDS A1 & A2

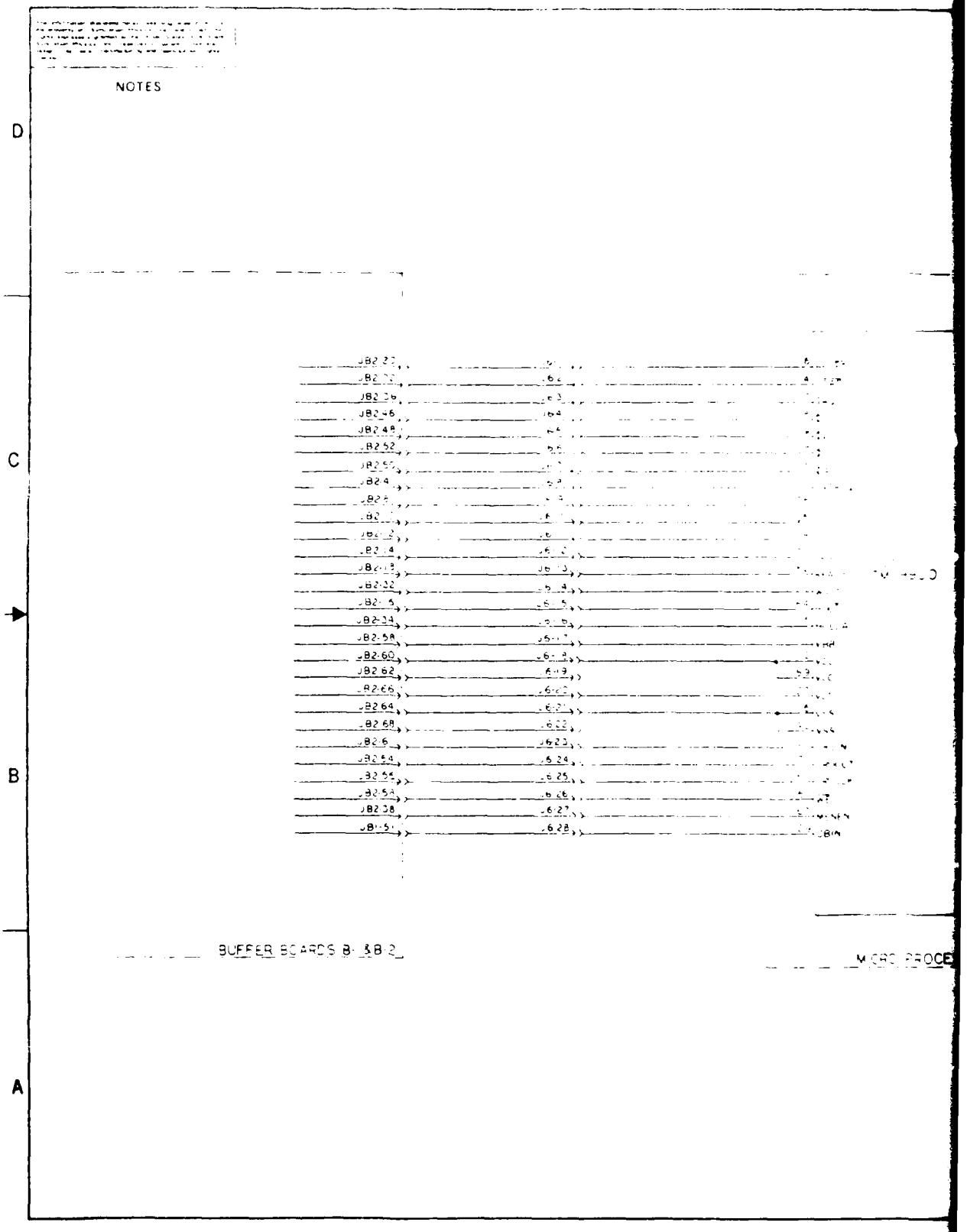
ITEM	QTY REQ	CODE IDENT NO.	PART NO OR IDENTIFYING NO	DESCRIPTION	MATERIAL	SPEC
				PARTS LIST		

DIMENSIONS ARE IN INCHES
DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS
UNLESS OTHERWISE SPECIFIED TOLERANCES ARE

CONTR NO

spire **SPIRE CORPORATION**
BEDFORD MASS 01730

D 4D656


NOTES
D
C
B
A
BUFFER BOARDS B-3B-2
MICRO PROCE

REVISIONS			
ZONE LTR	DESCRIPTION	DATE	APPROVED

D

C

B

A

J6-29	J81-30
J6-30	J81-32
J6-31	J81-24
J6-32	J81-26
J6-33	J81-28
J6-34	J81-20
J6-35	J81-22
J6-36	J81-34
J6-37	J81-36
J6-38	J81-38
J6-39	J81-40
J6-40	J81-42
J6-41	J81-44
J6-42	J81-46
J6-43	J81-48
J6-44	J81-50

J9-1	J82-44
J9-2	J82-42
J9-3	J82-40
J9-4	J82-38
J9-5	J82-36
J9-6	J82-34
J9-7	J82-32
J9-8	J81-2
J9-9	J81-5
J9-10	J81-4
J9-1	J81-1
J9-2	J81-3
J9-3	J81-8
J9-4	J81-6
J9-5	J81-4

BUFFER BOARDS B-1 & B-2

ITEM	ITEM	REQ	CODE IDENT NO.	PART NO. OR IDENTIFYING NO.	DESCRIPTION	MATERIAL	SPEC

PARTS LIST

DIMENSIONS ARE IN INCHES
DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS
UNLESS OTHERWISE SPECIFIED TOLERANCES ARE

DECIMALS INCHES MM
2 PLACE
3 PLACE
FRACTIONS
ANGULAR

CONTR NO
DR
CHK
APPD
APPD
CONTR DSGN ACT APPVL
GOVT ACT APPVL

NEXT ASSY | USED ON
APPLICATION

MATERIAL
FINISH

spire

SPIRE CORPORATION
BEDFORD, MASS 01730

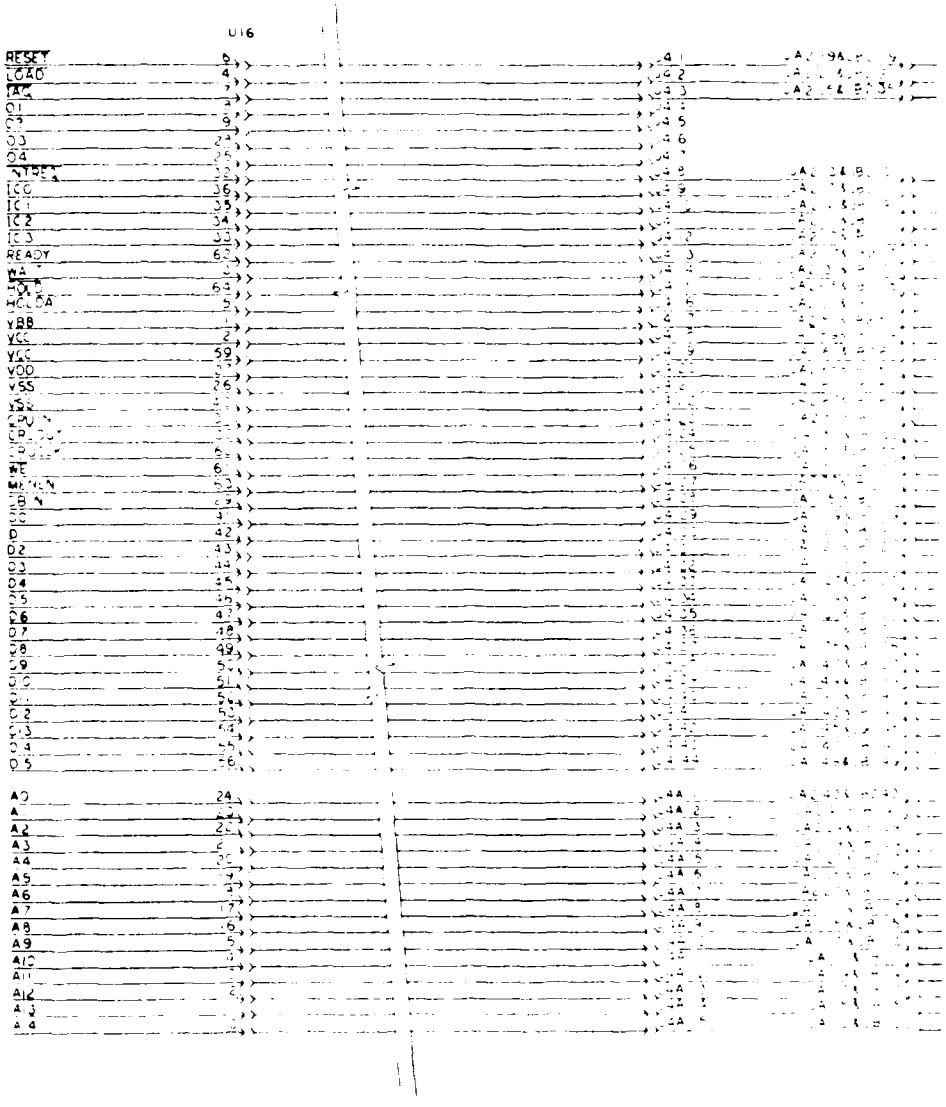
SUCCESSOR

SIZE	CODE IDENT NO	DRAWING NO	REV
D	4D656	[illegible]5	

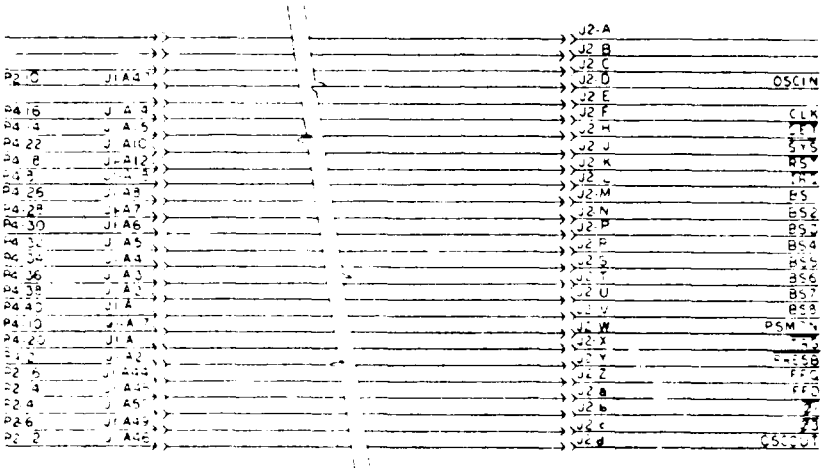
SCALE | SHEET 27

FROM COPY FURNISHED TO DOD

NOTES


U16
RESET
READY
D
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
A0
A
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12

REVISIONS			
ZONE	LTR	DESCRIPTION	DATE	APPROVED



ITEM	ITEM	REQ	CODE IDENT NO	PART NO OR IDENTIFYING NO	DESCRIPTION	MATERIAL	SPEC

PARTS LIST

DIMENSIONS ARE IN INCHES
DIMENSIONS IN PARENTHESIS ARE IN MILLIMETERS
UNLESS OTHERWISE SPECIFIED TOLERANCES ARE

QUANTITY

DRAWN
CHK
APPD
APPD

spire SPIRE CORPORATION
BEDFORD MASS 01730

SIZE D | CODE IDENT NO 4D656 | DRAWING NO | REV

NEXT ASSY | USED ON | APPLICATION

SCALE | WEIGHT | SHEET

APPENDIX II
ASSEMBLY LANGUAGE CODES FOR THE 990/100M CONTROL OF THE TEST SYSTEM

```
0010                    IDT   'MICRO'
0010        2400 CODE   EQU   2400          START ADDRESS OF CODE TO RUN MICRO
0020        FF63 SYAP   EQU   FF63          SYNC MAP SPACE
0030        FF43 IOAP   EQU   FF43          IOTAP MAP SPACE
0040        FF31 ECAP   EQU   FF31          ECC MAP SPACE
0050        FF53 EPAP   EQU   FF53          EPR MAP SPACE
0060        0460 BPR    EQU   0460          MACHINE CODE FOR BP
0070        0016 INT5   EQU   0016          INTERRUPT 5 VECTOR
0080        0012 INT4   EQU   0012          INTERRUPT 4 VECTOR
0090        FF24 VECT   EQU   FF24          SYNC VECTOR
0100        0004 BYT    EQU   4             NUMBER OF BYTES PER ERROR
0110        000A NUM    EQU   10            NUMBER OF ERRORS ALLOWED IN TEST
0120        4600 F      EQU   4600          ASCII F
0130        5200 P      EQU   5200          ASCII P
0140        1300 ECC    EQU   1300          ASCII ECC
0150        0120 CONT   EQU   0120          CONTROL BASE ADDRESS
0160        0130 INPT   EQU   0130          INPUT BASE ADDRESS
0170        0100 INTP   EQU   0100          INTERRUPT BASE ADDRESS
0180        0030 SER    EQU   0030          SERIAL BASE ADDRESS
0181        0000 P0     EQU   0
0182        0001 P1     EQU   1
0183        0002 P2     EQU   2
0184        0003 P3     EQU   3
0185        0004 P4     EQU   4
0186        0005 P5     EQU   5
0187        0006 P6     EQU   6
0188        0007 P7     EQU   7
0189        0008 P8     EQU   8
0190        0009 P9     EQU   9
0191        000A P10    EQU   10
0192        000B P11    EQU   11
0193        000C P12    EQU   12
0194        000D P13    EQU   13
0195        000E P14    EQU   14
0196        000F P15    EQU   15
0200                    D'OP' CNT+14
0210                    D'OP' STAD+11
0220 0000               *
```

```
0230 0000        *
0240 0000        *
0250 E000              AORG >E000          <<<< SETUP SEGMENT >>>>
0260 E000        *
0270 E000 0300 SETP    LIMI 0              TURN OFF ALL INTERRUPTS
     E002 0000
0280 E004 02E0         LWPI STWP           SETUP THE WORKSPACE
     E006 FF48
0290 E008 0205         LI   R5,SYNC        SYNC ENTRY POINT
     E00A E100
0300 E00C 0206         LI   R6,SYWP        ADDRESS OF SYNC WORKSPACE
     E00E FF28
0310 E010 0207         LI   R7,VECT        SYNC VECTOR (WP,PC)
     E012 FF24
0320 E014 CDC6         MOV  R6,*R7+        SYNC WORKSPACE INTO SYNC VECTOR
0330 E016 C505         MOV  R5,*R7         SYNC ENTRY POINT INTO SYNC VECTOR
0340 E018 0201         LI   R1,BRA         MACHINE CODE FOR B@>
     E01A 0460
0350 E01C 0202         LI   R2,ESC         ESCAPE ENTRY POINT
     E01E E200
0360 E020 0203         LI   R3,ERR         ERROR ENTRY POINT
     E022 E250
0370 E024 C120         MOV  @INT5,R4       INTERRUPT 5 VECTOR PC INTO R4
     E026 0016
0380 E028 CD01         MOV  R1,*R4+        B@> INTO FIRST INSTRUCTION WORD
0390 E02A C503         MOV  R3,*R4         ERR ENTRY INTO SECOND WORD
0400 E02C C120         MOV  @INT4,R4       INTERRUPT 4 VECTOR PC INTO R4
     E02E 0012
0410 E030 CD01         MOV  R1,*R4+        FIRST INSTRUCTION WORD
0420 E032 C502         MOV  R2,*R4         ESC ENTRY INTO SECOND WORD
0430 E034 0646         DECT R6
0440 E036 0646         DECT R6             R6 NOW HAS PROPER OVERRUN ADDRESS
```

```
0450 E038 2FA0 TOP   OUT  @CRLF
     E03A E350
0460 E03C 2FA0       OUT  @FR          OUTPUT INITIAL QUERRY
     E03E E354
0470 E040 2EC0       ECHO R0           GET RESPONSE WITH ECHO
0480 E042 2FA0       OUT  @CRLF
     E044 E350
0490 E046 0280       CI   R0,F         IS ANSWER AN F?
     E048 4600
0500 E04A 1304       JEQ  FRES         YES. THEN JUMP TO FRES
0510 E04C 0280       CI   R0,R         IS ANSWER AN R?
     E04E 5200
0520 E050 1314       JEQ  REST         YES. THEN JUMP TO REST
0530 E052 10F2       JMP  TOP          IF NEITHER, THEN MISTAKE. TRY AGAI
0540 E054 0208 FRES  LI   R8,ERWP      ERROR WORKSPACE POINTER INTO R8
     E056 FF68
0550 E058 C606       MOV  R6,*R8       OVERRUN ADDRESS INTO R0 OF ERR
0560 E05A 0201       LI   R1,BYT       NUMBER OF STORE BYTES PER ERROR
     E05C 0004
0570 E05E CA01       MOV  R1,@4(8)     PUT INTO R2 OF ERR
     E060 0004
0580 E062 0581       INC  R1           ROUND UP BYTES TO WORDS
0590 E064 0811       SRA  R1,1
0600 E066 0A11       SLA  R1,1
0610 E068 0202       LI   R2,NUM       NUMBER OF ERRORS ALLOWED IN STACK
     E06A 000A
0620 E06C 3381       MPY  R1,R2        NUMBER OF BYTES ALLOWED IN STACK I
0630 E06E C086       MOV  R6,R2        OVERRUN ADDRESS INTO R2
0640 E070 6083       S    R3,R2        START OF STACK INTO R2
0650 E072 CA82       MOV  R2,@2(8)     START OF STACK INTO R1 OF ERR
     E074 0002
0660 E076 CA82       MOV  R2,@20(8)    START OF STACK INTO R10 OF ERR
     E078 0014
0670 E07A 0420 REST  BLWP @VECT        SYNCHRONIZE MICROS
     E07C FF24
0680 E07E 0280       CI   R0,F         IS THIS A FRESHSTART?
     E080 4600
0690 E082 1602       JNE  PSU          NO. THEN RETURN AS FROM ESC
0700 E084 0460       B    @CODE        FREST START. SO GO TO CODE
     E086 E800
0710 E088 02E0 PSU   LWPI ESWP         ESC WORKSPACE
     E08A FF8C
0720 E08C 0380       RTWP
0730 E08E            .
```

```
0740 E03E        *
0750 E03E        *
0760 E100              AORG >E100          <<<< SYNC SEGMENT >>>
0770 E100        *
0780 E100 0200 SYNC    LI   R12,CONT       BASE ADDRESS FOR CONTROL
     E102 0120
0790 E104 1E00         SBZ  0              TRIGGER THE RESET
0800 E106 10FF WAIT    JMP  WAIT           WAIT HERE FOR THE RESET
0810 E108 02E0 REEN    LWPI SYWP           RELOAD WORKSPACE POINTER UPON REEN
     E10A FF28
0820 E10C 0200         LI   R12,DER        BASE ADDRESS FOR [illegible] OUTPUT
     E10E 0080
0830 E110 1D1E         SBO  18             ENABLE RECEIVER INTERRUPT
0840 E112 0200         LI   R12,CONT       BASE ADDRESS FOR CONTROL
     E114 0120
0850 E116 1D03         SBO  3              QUIESCENT CLK
0860 E118 1D02         SBO  2              QUIESCENT SET
0870 E11A 1E04         SBZ  4              RESET SYSTEM
0880 E11C 1D04         SBO  4
0890 E11E 1E01         SBZ  1              TURN ON SYSTEM
0900 E120 0200         LI   R12,INRR       INTERRUPT BASE ADDRESS
     E122 0100
0910 E124 1D04         SBO  4              ENABLE ESC AT 9901
0920 E126 1D05         SBO  5              ENABLE ERR AT 9901
0930 E128 0380         RTWP
0940 E12A        *
0950 E12A        *
0960 E12A        *
0970 E200              AORG >E200          <<<< ESCAPE SEGMENT >>>
0980 E200        *
0990 E200 0300 ESC     LIMI 0              TURN OFF ALL INTERRUPTS
     E202 0000
1000 E204 0200         LI   R12,CONT       BASE ADDRESS FOR CONTROL
     E206 0120
1010 E208 1D01         SBO  1              TURN OFF SYSTEM
1020 E20A 0400         CLR  R0             PREPARE R0
1030 E20C 0200         LI   R12,DER        SERIAL BASE ADDRESS
     E20E 0080
1040 E210 3600         STCR R0,8           GET INTERRUPT CHARACTER
1050 E212 1D12         SBO  18             CLEAR INTERRUPT
1060 E214 0280         CI   R0,>E00        IS IT AN ESC?
     E216 1E00
1070 E218 1301         JEQ  MON            YES. THEN JUMP TO MON
1080 E21A 0380         RTWP                NO. THEN RETURN
1090 E21C 06A0 MON     BL   @>30           GO TO MONITOR
     E21E 0030
1100 E220        *
```

```
1110 E220       *
1120 E220       *
1130 E250             AORG >E250          <<<< ERROR SEGMENT >>>>
1140 E250       *
1150 E250 0200 ERR    LI   R12,CONT       CONTROL BASE ADDRESS
     E252 0120
1170 E254 1D01        SBO  1              TURN OFF SYSTEM
1180 E256 2002        MOV  R2,R3          MAX BYTE COUNT INTO R3
1190 E258 C24E        MOV  R14,*R1+       STORE PC AT ERROR ON STACK
1200 E25A 0643        DECT R3             DECREMENT THE BYTE COUNT
1210 E25C 1E02        SBZ  2              SETUP THE READ
1220 E25E 1D02        SBO  2
1230 E260 020C READ   LI   R12,INPT       INPUT BASE ADDRESS
     E262 0130
1240 E264 3611        STCR *R1,8          READ ONE BYTE ONTO STACK
1250 E266 0581        INC  R1             INCREMENT ERROR STACK POINTER
1260 E268 0603        DEC  R3             DECREMENT BYTE COUNT
1270 E26A 1305        JEQ  DONE           GO TO DONE IF FINISHED
1280 E26C 020C        LI   R12,CONT       CONTROL BASE ADDRESS
     E26E 0120
1290 E270 1E03        SBZ  3              CLOCK TO NEXT BYTE
1300 E272 1D03        SBO  3
1310 E274 10F5        JMP  READ           GO GET NEXT BYTE
1320 E276 0581 DONE   INC  R1             ROUND UP ERROR STACK POINTER
1330 E278 0811        SRA  R1,1
1340 E27A 0A11        SLA  R1,1
1350 E27C C082        MOV  R2,R3          BYTE COUNT
1360 E27E 0583        INC  R3             ROUND IT UP
1370 E280 0813        SRA  R3,1
1380 E282 0A13        SLA  R3,1
1390 E284 A0C1        A    R1,R3          WILL WE OVERRUN ON NEXT ERROR?
1400 E286 8003        C    R3,R0
1410 E288 1B03        JH   OVER           IF YES - JUMP TO OVER
1420 E28A 0420        BLWP @VECT          SYCHRONIZE MICROS
     E28C FF24
1430 E28E 0380        RTWP                CONTINUE WHERE WE LEFT OFF
1440 E290 2FA0 OVER   OUT  @NOMO          OUTPUT OVERRUN MESSAGE
     E292 E378
1450 E294 0204        LI   R4,ESCWP       ESCAPE WORKSPACE POINTER TO R4
     E296 FF80
1460 E298 C90D        MOV  R13,@26(4)     SETUP PSEUDO ESC RETURN
     E29A 001A
1470 E29C C90E        MOV  R14,@28(4)
     E29E 001C
1480 E2A0 C90F        MOV  R15,@30(4)
     E2A2 001E
1490 E2A4 C04A        MOV  R10,R1         RETORE START OF STACK TO R1
1500 E2A6 06A0        BL   @>30           BRANCH TO MONITOR
     E2A8 0030
1510 E2AA       *
```

```
1320 E3AA       *
1330 E3AA       *
1340 E350             AORG >E350        <<<< MESSAGE LIST >>>>
1350 E350       *
1360 E350 0D0A CRLF   DATA >0D0A,0
     E352 0000
1370 E354 4652 FR     TEXT 'FRESH START OR RESTART? (F OR R)'
     E356 4553
     E358 4820
     E35A 5354
     E35C 4152
     E35E 5420
     E360 4F52
     E362 2052
     E364 4553
     E366 5441
     E368 5254
     E36A 3F20
     E36C 2846
     E36E 204F
     E370 5220
     E372 5229
1390 E374 0D0A        DATA >0D0A,0
     E376 0000
1400 E378 4E4F NOMO   TEXT 'NO MORE ROOM IN ERROR STACK.'
     E37A 204D
     E37C 4F52
     E37E 4520
     E380 524F
     E382 4F4D
     E384 2049
     E386 4E20
     E388 4552
     E38A 524F
     E38C 5220
     E38E 5354
     E390 4143
     E392 4B2E
1600 E394 0D0A        DATA >0D0A,0
     E396 0000
2000                  END
```

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

Printed by
United States Air Force
Hanscom AFB, Mass. 01731