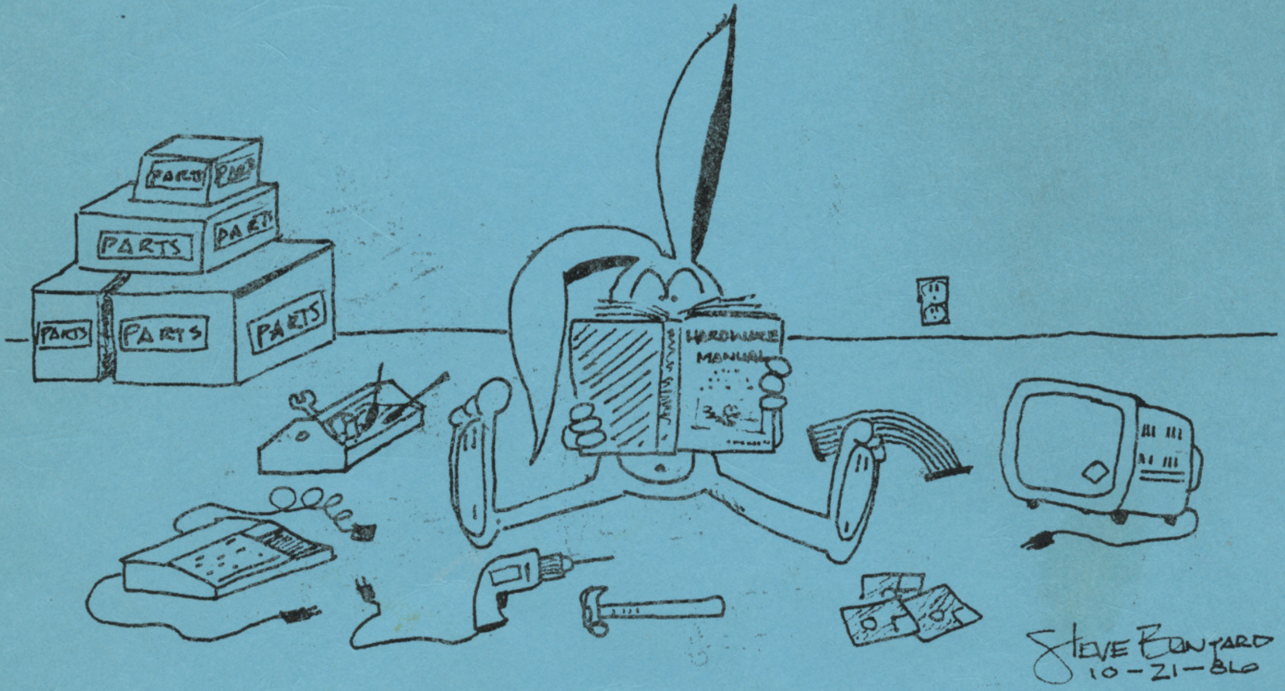
HARDWARE MANUAL

For The

TEXAS INSTRUMENTS

99/4A

HOME COMPUTER



Michael L. Bunyard, PE

HARDWARE MANUAL

For The

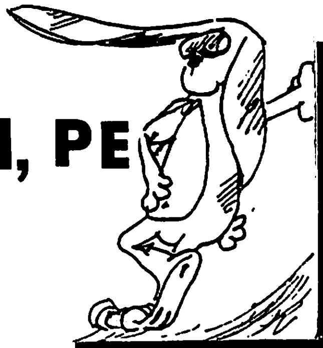
TEXAS INSTRUMENTS

99/4A

HOME COMPUTER

By

Michael L. Bunyard, PE



ABOUT THE AUTHOR

Mike Bunyard is a former Senior Member of the Technical Staff of Texas Instruments, Inc. In his 16 years of combined service at TI, he has participated in numerous projects in a System and Logic Design capacity. He is currently the owner of Angelo Microcomputer Systems.

Additionally, Mr. Bunyard has taught Logic Design on a university level, and has participated as an IEEE ad hoc Advisor for the Accrediting Board for Electronics Technology. He holds three patents, a BSEE from Texas Tech University, and a MSEE from Southern Methodist University.

SEMINAR FOR THE TI 99/4A

* ROUND TRIP PLANE FARE FOR TWO

* EXPENSES

    1. HOTEL
    2. MEALS
    3. RENT CAR
    4. $200 PER DAY

* GENERAL AUDIENCES

* TECHNICAL AUDIENCES

* AVAILABLE 8 HOURS

    A. WILL DISCUSS /4 HARDWARE W/ SLIDES

    B. 9900 HW & S/W CHARACTERISTICS W/SLIDES

    C. EXPANSION BOX CARDS

        1. RS232

        2. MEMORY EXPANSION

        3. DISK CONTROLLER

        4. P-CODE

* EXTENDED BASIC

* REPAIR TECHNIQUES

* GENERAL LOGIC DESIGN TECHNIQUES

* 9900 PROGRAMMING

* HARDWARE MANUAL OFFERED FOR $11.00

IMPORTANT NOTICE FOR TECHNICAL DATA WITHIN THIS MANUAL

It is important that one should read and understand the following paragraph before purchasing and/or using this Manual.

Angelo Microcomputer Systems has double checked the accuracy of the technical information contained in this Manual. To the best of our knowledge it is accurate, but Angelo Microcomputer Systems makes no warranty that the technical information in this Manual is correct. In no event shall Angelo Microcomputer Systems be liable to anyone for special, collateral, incidental, or consequential damages in connection with or arising out of the purchase or use of this Manual and the sole and exclusive liability of Angelo Microcomputer Systems, regardless of the form of action, shall not exceed the price of this Manual. Moreover, Angelo Microcomputer Systems shall not be liable for any claim of any kind whatsoever by any other party against the user of the material in this Manual.

Texas Instruments neither commissioned the writing of this Manual nor contributed to it. The author was not an employee of Texas Instruments at the time of the writing of this Manual.

TO OBTAIN ADDITIONAL COPIES OF THIS MANUAL:

SEND $19.94 PER COPY--CHECK OR MONEY ORDER TO

THE BUNYARD GROUP
P. O. BOX 53171
LUBBOCK, TX 79453

FOR YOUR CONVENIENCE AN ORDER BLANK HAS BEEN INCLUDED IN THE BACK OF THIS MANUAL.

TABLE of CONTENTS

SECTION 1 INTRODUCTION

1.1 PURPOSE 1- 1
1.2 ORGANIZATION 1- 1
1.3 ADDITIONAL DOCUMENTS 1- 1

SECTION 2 GENERAL SYSTEM DESCRIPTION

2.1 TI-99/4 HARDWARE ORGANIZATION 2- 1
2.2 MICROPROCESSOR AND PRIMARY MEMORIES 2- 1
2.3 VIDEO DISPLAY PROCESSOR 2- 1
2.3.1 VDP/DRAM Array Timing 2- 2
2.3.2 Video Amplifier 2- 2
2.3.3 Output Clocks 2- 2
2.3.4 Data Bus Connection 2- 3
2.4 SOUND GENERATOR 2- 3
2.5 GRAPHICS READ ONLY MEMORY 2- 3
2.5.1 GROM Address Register 2- 4
2.5.2 GROM Libraries 2- 4
2.5.3 ROM Array 2- 4
2.5.4 Data Prefetch 2- 5
2.5.5 GROM Control lines 2- 5
2.5.6 GROM READ DATA Command 2- 6
2.5.7 GROM WRITE ADDRESS Command 2- 6
2.5.8 GROM READ ADDRESS Command 2- 7
2.5.9 GROM Synchronization 2- 7
2.5.10 GROM Supply Voltages 2- 7
2.6 SYSTEM MEMORY MAP 2- 7
2.7 DATA BUS INTERFACE 2- 8
2.8 CRU I/O 2- 9
2.9 DATA BUS CONVERSION 2-10
2.10 COMMAND MODULE PORT 2-10
2.11 I/O PORT 2-11
2.12 INTERRUPT STRUCTURE 2-11
2.13 DATA BUS LOADING CONSIDERATIONS 2-11

SECTION 3 FUNCTIONAL BLOCK DESCRIPTIONS

3.1 PURPOSE 3- 1
3.2 CLOCK GENERATOR 3- 1
3.2.1 Oscillator Logic 3- 1
3.2.2 Phase Overlap Control 3- 1
3.2.3 Four Phase Clock Generator 3- 2
3.2.4 Synchronizing Function 3- 2
3.3 DATA BUS TRANSLATOR 3- 2

3.3.1 WRITE Cycle Operation 3- 3
3.3.2 READ Cycle Operation 3- 3
3.3.3 Data Bus Translator Timing and Control 3- 3
3.3.3.1 TMS 9900 READY Control 3- 4
3.3.3.2 WE- Generation 3- 5
3.3.3.3 Data Bus Driver Control 3- 5
3.4 SYSTEM RESET 3- 5
3.5 MEMORY SPACE DECODING 3- 6
3.6 Joystick Driver 3- 7
3.7 KEYBOARD 3- 7
3.8 CASSETTE INTERFACE 3- 7
3.8.1 Motor Drive Circuit 3- 7
3.8.2 Cassette WRITE Circuit 3- 8
3.8.3 Cassette READ Circuit 3- 8

SECTION 4 PITFALLS

4.1 SUMMARY OF PITFALLS 4- 1
4.2 SOUND GENERATOR CONSIDERATIONS 4- 2
4.3 VIDEO DATA PROCESSOR CONSIDERATIONS 4- 2
4.4 GROM CONSIDERATIONS 4- 2
4.5 SPEECH SYNTHESIZER CONSIDERATIONS 4- 3
4.6 ILLEGAL TMS 9900 INSTRUCTIONS 4- 3
4.7 TMS 9900 MEMORY ACCESSING 4- 4
4.8 TMS 9900 MOVE INSTRUCTION EXECUTION 4- 4
4.9 8-BIT DATA BUS CHARACTERISTICS 4- 5
4.10 ALPHA SHIFT LOCK KEY CHARACTERISTICS 4- 5
4.11 COMMAND MODULE PORT RESET CONSIDERATIONS 4- 5
4.12 CRUIN DATA LINE CONSIDERATIONS 4- 6
4.13 SYSTEM ROM CONSIDERATIONS 4- 6
4.14 SYNCHRONIZATION 4- 6
4.15 COMMAND MODULE PORT PIN ORDERING 4- 6
4.16 DISK DRIVE CONSIDERATIONS 4- 7
4.17 MEMORY EXPANSION MODULE CHARACTERISTICS 4- 7

SECTION 5 TMS 9900 H/W DESCRIPTION

5.1 TMS 9900 ORGANIZATION 5- 1
5.2 TMS 9900 PIN DEFINITION 5- 1
5.3 RETURN CURRENT CONSIDERATIONS 5- 2
5.4 ADDRESS BUS 5- 3
5.5 SIGNAL SETUP 5- 3
5.6 MEMORY CONTROL 5- 3
5.6.1 DBIN Control Signal 5- 4
5.6.2 Control Signal Observations 5- 4
5.6.3 READY and WAIT Signals 5- 4
5.6.4 HOLD and HOLDA Operation 5- 4
5.7 COMMUNICATION REGISTER UNIT DEFINITION 5- 5

5.8 INTERRUPT STRUCTURE 5- 5
5.8.1 Interrupt Level Control 5- 6
5.8.2 LOAD Interrupt 5- 6

SECTION 6 TMS 9900 INSTRUCTION SET

6.1 GENERAL COMMENTS 6- 1
6.2 TMS 9900 ARCHITECTURE 6- 1
6.3 REGISTER FILE 6- 1
6.4 BYTE ORIENTED INSTRUCTIONS 6- 2
6.5 OPCODE/OPERAND ORDERING 6- 2
6.6 INPUT/OUTPUT INSTRUCTIONS 6- 2
6.6.1 CRU Organization 6- 3
6.6.2 CRU Lines 6- 3
6.6.3 CRU Address Space 6- 3
6.6.4 CRU Relative Addressing Instructions 6- 3
6.6.5 Multi-bit CRU Instructions 6- 4
6.6.6 Tag Bit Pair Definition 6- 4
6.6.7 CRU Transfer Bit Length 6- 5
6.7 BIT MANIPULATION INSTRUCTIONS 6- 5
6.8 MEMORY TAG BITS 6- 6
6.9 LOGICAL INSTRUCTIONS 6- 6
6.10 RELATIVE JUMP INSTRUCTIONS 6- 7
6.10.1 Conditional Jump Instructions 6- 7
6.11 SHIFT INSTRUCTIONS 6- 8
6.12 COMPARE INSTRUCTIONS 6- 8
6.13 SUBROUTINE CALLING 6- 9
6.14 XOP INSTRUCTION 6- 9
6.15 CONTEXT SWITCHING 6- 9
6.15.1 External Context Switching 6-10
6.15.2 LOAD Operation 6-10
6.16 LOAD INTERRUPT MASK IMMEDIATE 6-10
6.17 ARITHMETIC INSTRUCTIONS 6-11
6.17.1 Unary Arithmetic Instructions 6-11
6.18 MULTIPLY AND DIVIDE INSTRUCTIONS 6-11
6.19 WORKSPACE REGISTER MANIPULATIONS 6-12
6.20 MOVE INSTRUCTIONS 6-12
6.21 SWAP BYTES INSTRUCTION 6-13
6.22 EXECUTE INSTRUCTION 6-13
6.23 UNSUPPORTED INSTRUCTIONS 6-13
6.24 SUMMARY 6-13

SECTION 7 THE PERIPHERAL EXPANSION BOX

7.1 PEB PURPOSE 7- 1
7.2 PEB POWER SUPPLY 7- 1
7.3 SOCKET PIN BUSSING 7- 2
7.4 Address Bus 7- 2

7.5 PEB CARDS 7- 2
7.6 CONSOLE INTERFACE 7- 2
7.6.1 Console End Interface 7- 2
7.6.2 PEB End Interface 7- 3
7.7 PERIPHERAL MEMORY SPACE ORGANIZATION 7- 3
7.7.1 Peripheral CRU Space Definition 7- 3

SECTION 8 32K BYTE MEMORY EXPANSION

8.1 MEMORY EXPANSION ORGANIZATION 8- 1
8.2 ADDRESS REQUIREMENTS 8- 1
8.3 DRAM FUNDAMENTALS 8- 2
8.3.1 Modes of Operation 8- 3
8.3.2 Address Sequencing 8- 3
8.3.3 Internal DRAM Bit Storage and Refresh 8- 3
8.4 MEMORY EXPANSION UNIT PLA 8- 3
8.5 MEX TIMING AND CONTROL 8- 5
8.6 MEMORY EXPANSION PRODUCT DIFFERENCES 8- 6
8.7 DIAGNOSTIC TEST PATTERNS 8- 6

SECTION 9 RS232 CARD

9.1 RS232 ORGANIZATION 9- 1
9.2 DEVICE SERVICE ROUTINE ROM 9- 1
9.3 PARALLEL PORT OPERATION 9- 1
9.4 CRU OPERATION 9- 2
9.5 UARTs 9- 3
9.6 INTERRUPT CONFIGURATION 9- 4
9.7 RS232 PORT CONFIGURATION JUMPERS 9- 4
9.8 RS-232 VOLTAGE LEVEL GENERATION 9- 4
9.9 RS-232 RECEIVERS 9- 4
9.10 UART RESET 9- 5
9.11 RS232 PROGRAMMABLE LOGIC ARRAY 9- 5
9.12 RS232 PORT PIN DEFINITION 9- 7
9.13 PARALLEL PORT PIN DEFINITIONS 9- 7
9.14 RS232 CARD AND MODULE COMMENTS 9- 8

SECTION 10 P-CODE CARD

10.1 P-CODE CARD DESCRIPTION 10- 1
10.2 DEVICE SERVICE ROUTINE ROMS AND GROMS 10- 1
10.3 P-CODE CRU 10- 1
10.4 P-CODE PROGRAMMABLE LOGIC ARRAY 10- 2
10.5 GROM CHIP INTERFACE 10- 4
10.6 P-CODE CARD DISABLE 10- 4
10.7 P-CODE CARD AND MODULE 10- 5

SECTION 11 DISK CONTROLLER CARD

11.1 DISC CONTROLLER ORGANIZATION 11- 1
11.2 DEVICE SERVICE ROUTINE ROMS 11- 1
11.3 FD1771 DISK CONTROLLER CHIP 11- 1
11.4 CRU ORGANIZATION 11- 1
11.4.1 Disk Controller CRU Status Port 11- 2
11.4.2 Disk Controller CRU Control Register 11- 2
11.5 PROGRAMMABLE LOGIC ARRAY DEFINITION 11- 2
11.6 CONSOLE READY CONTROL 11- 3
11.7 DRIVE KEEP-ALIVE 11- 3
11.8 INTERRUPT STRUCTURE 11- 4
11.9 DATA/CLOCK SEPARATOR 11- 4
11.9.1 Data/Clock Steering Logic 11- 5
11.9.2 Switching Logic 11- 5
11.9.3 Missing Clock Compensation 11- 5
11.9.4 Missing Clock Counter 11- 5
11.10 OSCILLATOR 11- 6
11.11 HEAD LOAD TIMING CONTROL 11- 6
11.12 DRIVE CONNECTED SENSE 11- 6
11.13 DRIVE PORT CONNECTOR DEFINITION 11- 6
11.14 CARD/MODULE DIFFERENCES 11- 7

APPENDIX A COMMAND MODULE PORT PIN DEFINITIONS A-1

APPENDIX B I/O PORT PIN DEFINITIONS B-1

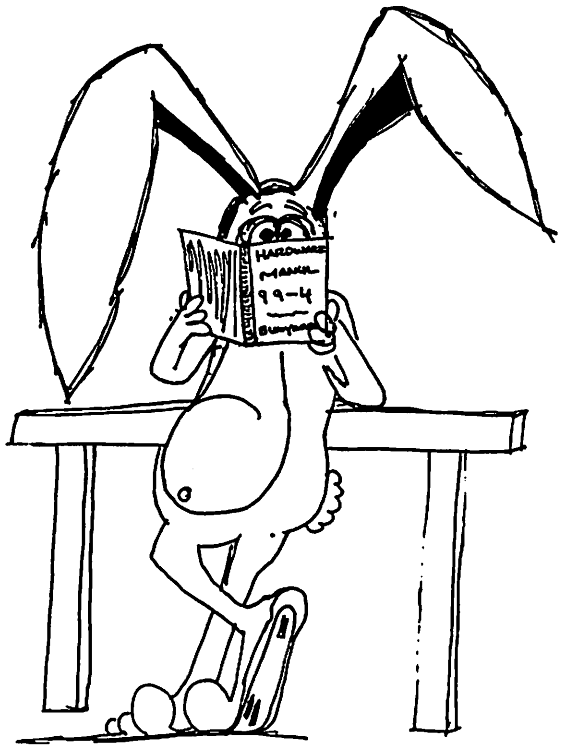
APPENDIX C PEB BACKPLANE PIN DEFINITION C-1

APPENDIX D A GROM SIMULATOR D-1

APPENDIX F TI-99/4A SCHEMATICS F-1

APPENDIX G EXTENDED BASIC COMMAND MODULE

G.1 ORGANIZATION . . . . G- 1
G.2 8K ROM PAGING . . . . G- 1
G.3 GROM OPERATION . . . . G- 1



LeT's Get STarTed!!!

CHAPTER 1

INTRODUCTION

1.1 PURPOSE

The purpose of this manual is to describe the TI-99/4 System hardware in detail. Short machine language programs are included to allow one to verify the results presented herein.

1.2 ORGANIZATION

Duplication of material is used in order to present a more complete discussion of the same subject at several different places in the manual; thus, one should not have to read cover to cover in order to gain sufficient information about one particular subject.

The basic topics to be covered are listed below.

* The TI-99/4(A) Console hardware in detail. Descriptive schematics are included.

* A description of known pitfalls when designing TI-99/4(A) hardware and software.

* A description of both the TMS 9900 uProcessor hardware characteristics and its instruction set.

* Peripheral Expansion Box Hardware description. No schematics are included for this part of the system.

* Some design examples for hardware interfaces for the TI-99/4(A).

1.3 ADDITIONAL DOCUMENTS

There are other documents available from Texas Instruments and other firms that may aid one in further investigation of the TI99/4A Home Computer. These are listed on the following page.

* TMS 9900 Microprocessor Data Manual.

* TMS 9901 Programmable Systems Interface Data Manual

* TMS9918A/TMS9928A/TMS9929A Video Display Processors Data Manual (the TMS 9918A is second sourced by Western Digital Corp.)

* TMS 9902A Asynchronous Communications Controller Data Manual

* TECHNICAL DATA MANUAL FOR THE TEXAS INSTRUMENTS HOME COMPUTER, and THE TI-99/4A CONSOLE AND PERIPHERAL EXPANSION SYSTEM TECHNICAL DATA, both of which were published and sold by Texas Instruments.

* MODEL 990 COMPUTER TMS 9900 MICROPROCESSOR ASSEMBLY LANGUAGE PROGRAMMER'S GUIDE, available from Texas Instruments, Inc. A later version of this is THE 990/99000 ASSEMBLY LANGUAGE REFERENCE MANUAL (Part no. 2270509-9701 *A, 15 November 1982).

* SN94624 (TMC9919) Sound Generator.

* TMS 4732 4K x 8 ROM.

* TMS 4764 8K x 8 ROM.

* TMS 4116-15 16K x 1 Dynamic RAM.

* MCM6810 128 x 8 Static RAM.

* The TTL Data Book (TI).

CHAPTER 2

GENERAL SYSTEM DESCRIPTION

2.1 TI-99/4 HARDWARE ORGANIZATION

Figure 2-1 is a basic block diagram of the TI-99/4 Home Computer. From an architectural point of view, the TI-99/4 is organized as a classical Von Neumann microprocessor system. Several custom memory chips access serially based on their own internal address counters. The implication of this access technique is that TMS 9900 MACHINE CODE MAY NOT BE EXECUTED DIRECTLY FROM THESE TYPES OF DEVICES. None of the advertised 16K Bytes of RAM may be used for executing machine level code. The data must be byte accessed with respect to the internal address counter in the TMS 9918 Video Data Processor (VDP) chip. The Graphics Read Only Memory (GROM) chips access in this manner also. The internal address counters must be reloaded whenever non-serial data are required.

2.2 MICROPROCESSOR AND PRIMARY MEMORIES

The TMS 9900 Microprocessor used in the 99/4 is a 16-bit machine that accesses a word oriented (16-bit) memory. Both a 128 word Static RAM (SRAM) and a 4K word ROM are provided for operating system use. There is no available SRAM for user programming unless SRAM data are moved temporarily to another memory, and then later restored. No system level software may be invoked until SRAM is restored.

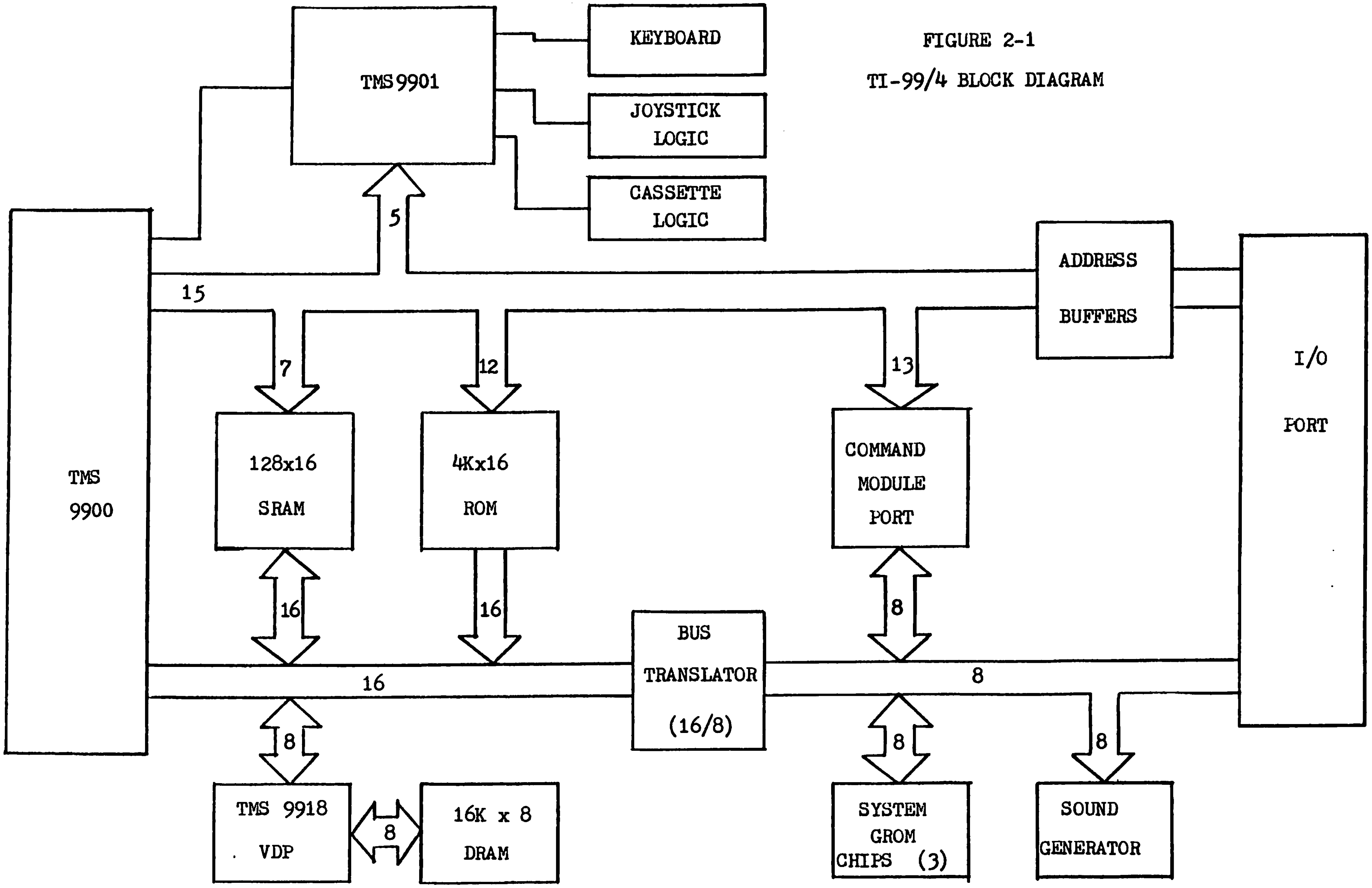
The SRAM pair is comprised of two Motorola MCM6810 128 x 8 RAMs. The ROM bank is made up of two TMS 4732 type ROMs to obtain the 4K x 16 organization. The TMS 9900 always accesses both memory chips in the two chip bank at the same time.

2.3 VIDEO DISPLAY PROCESSOR

The video interface is through a TMS 9918 VDP chip that has its own autonomous 16K byte RAM. This video RAM is made up of eight TMS 4116-15 Dynamic RAM (DRAM) chips connected in a rather strange manner. This is shown in Figure 2-2. The VDP chip output drivers

FIGURE 2-1

TI-99/4 BLOCK DIAGRAM



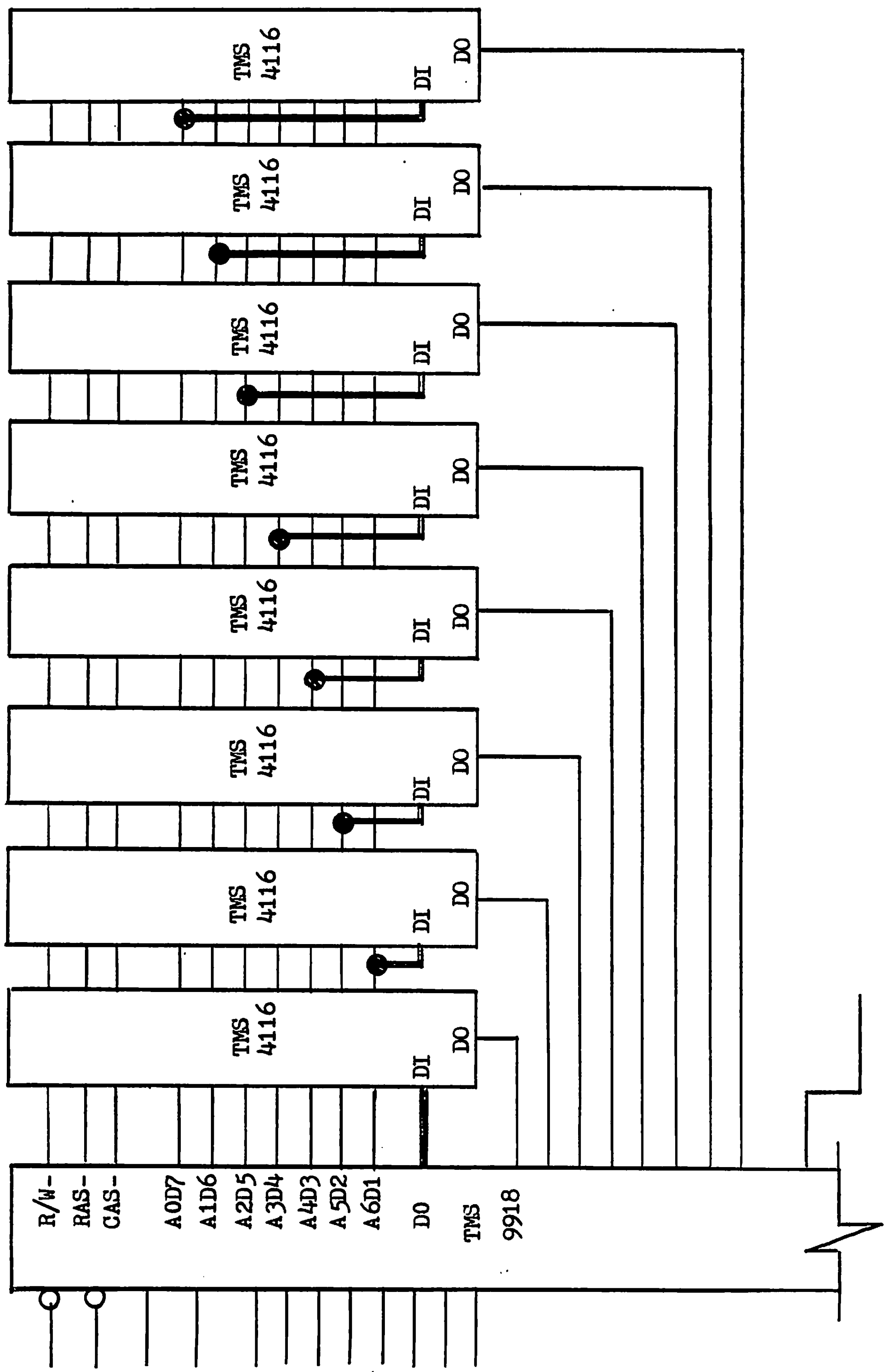


FIGURE 2-2
VDP DRAM CONNECTION

for the DRAM array are used for both WRITE data and address line functions. This type of arrangement requires a connection at the DRAM chip to join the DRAM "Data IN" pin to one of the seven DRAM address pins. Obviously, a different address line must be used on each of seven DRAMs for this connection. The eighth DRAM chip has a connection directly to the VDP chip for its Data IN pin. Apparently it was in this manner that the TMS 9918 bar designers eliminated the need for an additional seven output buffers which would have been required to support a bidirectional data bus for the DRAMs. A later version of the TMS 9918 did comprehend a bidirectional data bus to enable the use of the TMS 4416-20 (16K x 4) DRAM chip, but this was never incorporated into the TI-99/4. This VDP version was the TMS 9118.

2.3.1 VDP/DRAM Array Timing

A detailed look at the data sheets for both the TMS 9918 and the TMS 4116-15 indicates an exact match for worst case specifications. This is possibly a critical timing area. Early TI-99/4 models (with the small key keyboard) had four Inverters in the W- control line to the DRAM array. Additionally, it had a 68 pF capacitor from CAS- to Logic Ground. Thus, delays were provided in these signal lines.

2.3.2 Video Amplifier

The TMS 9918 VDP chip does not provide a 0 volt based video output. The earlier version large screen Zenith Monitors are not AC coupled to the TI-99/4 driving video; therefore, some means of providing DC restoration was required. This task is accomplished by a two stage transistor video amplifier. The second stage of this amplifier provides a low impedance drive for the video cable to the monitor or RF Modulator. The small screen monitors sold later were AC coupled, and will function quite nicely when driven directly from the VDP chip. See the schematics at the back of the Manual for more information on this area. Also, some degree of RFI suppression is provided by this amplifier circuit.

2.3.3 Output Clocks

The VDP chip also provides a clock output that is used for both the Sound Generator and the GROM chips. This clock is obtained by dividing the 10.73863 MHz clock by 24 to achieve a 447.443 KHz clock necessary for the GROM chips. The early model TI-99/4 Sound Generator required a 3.58 MHz clock, as is evidenced in the schematics contained in the early versions of the TECHNICAL DATA Manual for the TI-99/4 from Texas Instruments, Inc. Sometime later the Sound Generator Chip was modified internally to accept the 447 KHz clock.

2.3.4 Data Bus Connection

The VDP chip connects directly to the MSBY of the TMS 9900 Data Bus, and is READ at an address of >8800 for data and >8802 for status. The WRITE address for data is >8C00, and >8C02 is for control (register) information. The >400 difference in the READ and WRITE addresses is to eliminate an undesired READ of the chip when information is being written to it. This is fully discussed in the Chapter on PITFALLS, as well as in that for the TMS 9900 Hardware Description.

2.4 SOUND GENERATOR

The TMS 9919 Sound Generator is connected to the 8-Bit Data Bus, and it may only be written to. A READ from the Sound Generator will cause the system to hang up in the "NOT READY" state because the Sound Generator did not see a WRITE pulse on its WE- input. The Sound Generator will function correctly in the "Early Write" connection (its WE- input GROUNDed, and the CS- input made to be a function of the Write Enable control signal from the Data Bus Converter). This concept was utilized on the late 99/4A models with a Gate Array to provide the required Timing and Control. The Sound Generator responds at an address of >8400.

2.5 GRAPHICS READ ONLY MEMORY

The GRAPHICS READ ONLY MEMORY (GROM) chip is in essence a ROM chip with an internal address register. Additional logic is also included to allow the paralleling of up to eight GROM chips (there are three Console GROM chips, and they are connected in parallel). From the user's point of view, the ROM array is organized 6K x 8, and based at zero in an 8K byte space. The GROM chip is a synchronous chip that is clocked by a square wave 447.443 MHz clock obtained from the TMS 9918 VDP chip. The GROM chip runs partially closed loop with respect to the TMS 9900 driving it in the sense that the "READY" input to the TMS 9900 is controlled by the GROM chip "READY" output during GROM accesses. GROM READY simply indicates, when it is at a HIGH level, that the CPU dependent portion of the GROM cycle has been completed. It does NOT MEAN that the FULL GROM cycle has been completed. GROM CHIP SELECT MUST REMAIN INACTIVE (at a HIGH level) FOR AT LEAST 2.5 GROM CLOCK PERIODS BETWEEN SUCCESSIVE ACCESSES. This translates to 5.6 us for a 447.443 KHz GROM clock. The GROM chips for P-CODE in the Peripheral Expansion Box are clocked with a 375 KHz square wave clock; therefore, 6.67 us must be allowed from the end of one access to the beginning of the next. THIS IS A SOFTWARE RESPONSIBILITY. There is no hardware check.

2.5.1 GROM Address Register

The Address Register within the GROM chip is organized as a 3-bit Page Register concatenated to the Most Significant end of a 13-bit ROM Address Counter. The quantity in the Page Register is compared with additional page logic set at the mask level, and the result of that comparison is then ANDed with the external GROM Chip Select. This defines when that GROM chip is selected for gating data from a READ DATA operation to the System Data Bus. THIS IS THE ONLY UNIQUE OPERATION OF THE GROM CHIP IN A BANK OF PARALLEL GROM CHIPS. EVERYTHING ELSE IS PERFORMED IN UNISON BY ALL GROM CHIPS IN THE BANK THAT IS CHIP SELECTED BY THE EXTERNAL CHIP SELECT. It is just the gating of data from a READ DATA operation that is unique to that one chip. Of course, the mask programmable logic will be different on each of the eight possible chips in a bank or "LIBRARY". This Paging scheme allows for the direct paralleling of up to eight GROM chips.

2.5.2 GROM Libraries

Internal page decodes of 0, 1, and 2 are in the TI-99/4 Console itself. A maximum of five additional GROM chips may be installed in the Command Module unless additional logic is provided to further decode the GROM address space. If this is done, three additional banks of five GROM chips (maximum for each bank) are supported. The Console GROMs do not have the benefit of the additional decoding; therefore, they constitute the first three pages of each added bank or library. The GROM READY outputs from each new decoded bank (of five chips, maximum) must be multiplexed to feed the READY of the internal GROM chips because the GROM chip READY output is normally LOW.

2.5.3 ROM Array

The 13-bit Address Counter addresses a 6K x 8 array that is based at zero in the counter space. The response of the GROM chip in top 2K bytes is not defined. Observed operation of the GROM chip indicates that the Address Register is organized as a pair of 8-bit shift registers for ADDRESS LOADING and READING purposes. The MSBY is always handled first, and the LSBY second. It also appears that the MSBY is first loaded into the LSBY portion of the Address Register, and then shifted into the MSBY of the Address Register when the LSBY is transferred into the GROM chip. An Address Read operation is DESTRUCTIVE, and the LSBY gets duplicated in the MSBY of the Address counter after the MSBY has been read. Simple machine language diagnostic programs may be written to verify the operation of the GROM chips. See Appendix E for an example of this.

2.5.4 Data Prefetch

A data prefetch occurs automatically after all READ DATA operations, as well as after the second WRITE ADDRESS operation. This data prefetch effectively decreases the READ DATA access time for the CPU portion of the GROM access cycle. The GROM Address Counter is incremented after the prefetch occurs.

2.5.5 GROM Control lines

There are two control lines that define what type of operation the GROM chip is to perform once it is selected. The first, labeled M0, is used to select between the internal GROM Address Counter and the ROM data. It is connected to the system address bit A14. This is Least Significant Bit (LSB) of the TMS 9900 Address Bus, and is the LS WORD address bit. The second control bit in this pair is M1, and it is connected to the TMS 9900 Memory Data Bus Direction control bit "DBIN". This control input determines if a READ or a WRITE operation is to occur. The definition of each of the four possible combinations of M0 and M1 is as follows.

(A14) M0	(DBIN) M1	
0	0	NO OPERATION, as this is a "WRITE to the ROM". It is probably a good idea not to use this combination even though it is supposed to be a NOP.
0	1	READ DATA from the ROM. This data byte is always located one location less than the address obtained from a READ ADDRESS operation due to the data prefetch.
1	0	WRITE ADDRESS to the internal GROM Address Register. If this is the first WRITE ADDRESS operation following a READ DATA operation, the Most Significant Byte (MSBY) of the address must be on the GROM Data Bus. The LSBY must be written after the MSBY has been written.
1	1	READ ADDRESS from the internal GROM Address Register. The MSBY will be the first byte read (after either the WRITE ADDRESS or READ DATA operation). The MS 5-bits of the internal GROM Address Register is right justified on the GROM data bus, and the MS 3-bits represent the last GROM page written. The second Address Read operation in a row will obtain the LSBY of the GROM Address Counter. THE READ ADDRESS OPERATION IS A DESTRUCTIVE READ FROM THE GROM ADDRESS REGISTER.

2.5.6 GROM READ DATA Command

In an effort to further discuss the exact operation of the GROM chip, we will examine the READ DATA operation in more detail. The first data access of the ROM array is automatic, and occurs during the second WRITE ADDRESS operation after the LSBY of the address has been written into the GROM Address Counter. The GROM access cycle continues with a prefetch of the data at the address just loaded into the GROM Address Counter. Finally, the Address Counter is incremented to compensate for the pre-fetch operation, and the full cycle is completed. This is why the GROM Address Counter is one bigger than the value previously loaded into the Address Counter, plus the number of READ DATA accesses that have occurred since the Address Register was last loaded. THE READ DATA OPERATION INITIALIZES A FLAG IN THE GROM CHIP THAT IS USED IN ADDRESS REGISTER OPERATIONS. IF A READ DATA OPERATION IS NOT PERFORMED TO INITIALIZE THE GROM CHIP WHEN ITS STATE IS UNKNOWN, ERRONEOUS ADDRESS OPERATIONS MAY RESULT. It is a good practice to observe this procedure any time the state of the GROM chip is questionable.

It probably cannot be said too many times that ALL GROM CHIPS IN AN ARRAY THAT IS EXTERNALLY CHIP SELECTED RESPOND IN UNISON, BUT THAT ONLY THE CHIP IN THE ARRAY THAT HAS THE CORRECT PAGE NUMBER CAN GATE ITS DATA TO THE SYSTEM DATA BUS DURING A READ DATA OPERATION. Notice that this implies that ALL chips load their Address Registers in unison, and that ALL chips drive the System Data Bus during a READ ADDRESS operation. Additionally, ALL chips go through the data fetch during a READ DATA operation although only one chip in the accessed bank will actually gate ROM data to the System Data Bus.

2.5.7 GROM WRITE ADDRESS Command

The WRITE ADDRESS operation must only be performed after a READ DATA operation has occurred. The latter operation has initialized a flag used within the GROM chip to keep MSBY and LSBY Address Register operations separated. The GROM chip has no RESET pin to initialize this flag, and its state is unknown from Power Up. It is always best to perform a "dummy" READ DATA operation if in doubt.

The MSBY of an Address must be transferred to the GROM chip first, and the LSBY second. It appears that the MSBY is first saved in the LSBY half of the GROM Address Register, and then transferred to the MSBY portion as the LSBY is being loaded into the proper position.

Considerably more action occurs when the second Address byte is moved to the GROM Address Register. After the byte is actually written into the LSBY of the Address Counter, a data prefetch occurs. This is easily witnessed by observing the longer second GROM READY with an oscilloscope while loading the full GROM address into the GROM bank.

2.5.8 GROM READ ADDRESS Command

Reading the GROM Address Register presents no perils except for the fact that it is a destructive read out. The Address Register may be validly read as long as it has been previously loaded or a READ DATA operation has occurred. The MSBY is always read first, and the full address that is read will be one bigger than the number of DATA READ accesses due to the data prefetch during the latter portion of the second ADDRESS WRITE operation.

2.5.9 GROM Synchronization

The TMS 9900 accesses the GROM chips with respect to its 12 MHz clock, and the GROM chips are synchronous machines driven by the 447.443 KHz clock generated by the VDP chip. These two asynchronous systems are connected by synchronizing GROM READY before it is sensed by the TMS 9900. There is no synchronization on the GROM side of the system.

Between chip processing/fabrication and GROM clock skew from chip to chip, it is possible for chips in the GROM bank to be one GROM clock period out of step with each other. This problem is handled by paralleling all of the GROM "READY" outputs. This READY output is a normally LOW, open drain, type driver; therefore, this type of connection is allowable and effective.

2.5.10 GROM Supply Voltages

The GROM is a PMOS chip, and requires three bias voltages. These are +5, -5, and -.8 volts. The -.8V bias is strange in that the System Data Bus may be pulled down to -.6V or so during a GROM READ ADDRESS operation. This appears to be no problem, though. The current sinking capability of the -.8V is limited by a 270 Ohm resistor. See the schematics for more details on this. Notice also that the PG1992 is a loosely specified diode, and has a rather large allowable Vf spread.

2.6 SYSTEM MEMORY MAP

The system memory map is detailed in the following table.

Table 2-1 SYSTEM MEMORY MAP

>0000 Console ROM >1FFF	>2000 Memory Expansion >3FFF
>4000 Peripheral Space >5FFF	>6000 Command Module Space >7FFF
>8000 Console RAM >83FF	>8400 Sound Generator >87FF
>8800 VDP Read A14=0 is Data A14=1 is Status >8BFF	>8C00 VDP Write A14=0 is Data A14=1 is Address >8FFF
>9000 Speech Read >93FF	>9400 Speech Write >97FF
>9800 GROM Read A14=0 is Data A14=1 is Address >9BFF	>9C00 GROM Write A14-0 is Data A14=1 is Address >9FFF
>A000 Memory Expansion >FFFF	

2.7 DATA BUS INTERFACE

A resistor network controlled by DBIN is used to provide both pull-up and pull-down on the System Data Bus. The Pull-down operation is used when data is being READ by the TMS 9900. Pull-up is used when data MAY be WRITTEN by the TMS 9900. The "MAY" is used because the memory control signal DBIN from the TMS 9900 is HIGH only for a memory READ operation. It is LOW for memory WRITES and for all cycles not involving memory. The pull-down feature will help the rather poor bus driving capability of the GROM chips. The resistor network also satisfies the PMOS requirements of the Speech Synthesizer unit when it is connected in the system. Notice that this requires any interface logic to provide the current sourcing to drive this network up, and current sinking to pull it down.

2.8 CRU I/O

The TMS 9901 PROGRAMMABLE SYSTEMS INTERFACE chip is used to interface to the keyboard, the Joystick Port, and the Cassette Port. It also handles the synchronization of the interrupts to the TMS 9900.

The TMS 9901 is "block decoded" to reside within a 9-bit address space even though the TMS 9901 requires only a 5-bit space. It is considered to be based at >0000, but only Address Bus bits 3, 4, and 5 are included in the decode.

Additional information on the Keyboard Drive is contained in the schematic of the Keyboard. The CRU Map for the TMS 9901 in the Console is in the following table.

Table 2-1 CONSOLE TMS 9901 CRU MAP

>0000 9901 Control	>0002 External Interrupt	>0004 VDP Interrupt
>0006 9901 TMR Intr KBD Sense: M N , . / = Joystick FIRE	>0008 Keyboard Sense: H J K L ; Sp Bar Joystick LEFT	>000A Keyboard Sense: Y U I O P Entr Joystick RIGHT
>000C KBD Sense 6 7 8 9 0 Joystick DOWN	>000E Keyboard Sense: Alpha Lock FCTN 1 2 3 4 5 Joystick UP	>0010 Keyboard Sense: Shift A S D F G
>0012 KBD Sense CTRL Q W E R T	>0014 Keyboard Sense: Z X C V B	>0016 Not Used
>0018 HIGH Level	>001A thru >001E Not Used	>0020 thru >0022 Reserved
>0024 and >0026 KBD Drive P2 and P3	>0028 KBD Drive Enable 0 = Enabled	>002A KBD Alpha Lock Drive, 0=ON

Table 2-1 CONSOLE TMS 9901 CRU MAP, CONTINUED

>002C Cassette Motor Control #1, 1=ON	>002E Cassette Motor Control #2, 1=ON	>0030 Audio Gate 1=Audio Inhibit
>0032 Cassette Data Out Drive	>0034 Reserved	>0036 Cassette Data IN Sense
	>0038 thru >003E Not Used	

2.9 DATA BUS CONVERSION

The I/O and Command Module Ports feature an 8-bit Data Bus; therefore, some means of converting the 16-bit TMS 9900 Data Bus to 8-bits is required. The TMS 9900 is controlled by its "READY" input to lengthen each memory cycle from 667 ns to 2 us (six 1/3 us clock cycles). Each byte has a cycle time of 1 us, and the LSBY operation occurs first (A15 = 1). A READ access time of 500 + 333 = 833 ns is the minimum time per byte on the 8-bit data Bus. A detailed description of this circuit is found in the Functional Block Description Chapter.

2.10 COMMAND MODULE PORT

The Command Module Port is organized to support GROM libraries, an 8K Primary Memory space based at >6000, and the full CRU space. The CRU options were apparently never used, and were removed from the late model TI-99/4As with a Gate Array for Timing and Control. See Appendix A for the pin definitions of the Command Module Port. Notice from the system schematics that, with the exception of "A14", the NMOS TMS 9900 Address Bus serves this port. Excessive loading of these lines should be avoided. The GROM chips are on the Data Bus; therefore, loading considerations should be observed there also. The later versions of the TI-99/4A that had a gate array for timing and control isolated the Command Module Port Data Bus from that of the I/O Port.

2.11 I/O PORT

The I/O Port supports 36K bytes of the 64K byte memory space, the memory control signals, memory READY status, System RESET as an output, the TMS 9900 IAQ bit, the full CRU space, two interrupts, the 8-bit System Data Bus, and +/- 5V power for the Speech Synthesizer. All outputs have been buffered (DBIN is buffered through four levels).

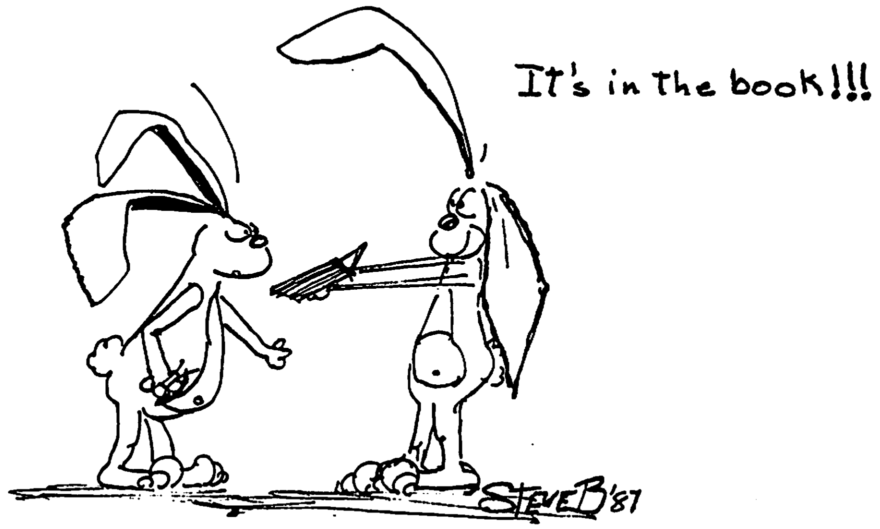
2.12 INTERRUPT STRUCTURE

Interrupt Level 1 is for use by all peripherals (if required), and the "LOAD" interrupt is available at the I/O Port for user implementation if desired. The LOAD interrupt may be used only if a Memory Expansion Unit is attached and loaded to provide the required two vectors. Appendix B lists the pin definitions for the I/O Port.

2.13 DATA BUS LOADING CONSIDERATIONS

The GROMs are connected to the 8-bit data bus that serves both the Command Module and I/O Ports; therefore, both drive and voltage level considerations must be observed when interfacing to this bus. The standard Texas Instruments interface from the Console to the Peripheral Expansion Unit buffers the Data Bus on both ends of the cable.

In the TI-99/4As with the Gate Array, both the GROM chips and the Command Module Data Bus were isolated from the I/O bus. Additionally, Logic Ground was substituted for the .8V bias on the GROM chips. Command Module Port WRITE information for both GROMs and its memory space will appear at the I/O Port on these models, though.



CHAPTER 3

FUNCTIONAL BLOCK DESCRIPTIONS

3.1 PURPOSE

The purpose of this Chapter is to describe in detail the major functional logic blocks of the TI-99/4. The depth of this discussion is suitable for use in repair of the 99/4 if the proper test equipment is also available.

3.2 CLOCK GENERATOR

The first block to be discussed is the clock generator for the TMS 9900 Microprocessor. There were two types used during the life of the 99/4A. The first one used was a SN74LS362N (TIM 9904AL), and this device required a 48 MHz crystal. Data sheets for the SN74LS362 were eliminated from later versions of the TI TTL DATA BOOK. The block diagram and pin out of the chip was kept, though, in Section 5 of that manual. Towards the end of the product life of the 99/4A, a version of the clock generator that functioned with a 12 MHz crystal was used. Most of the discussion here is for both chips with the exception of the manner in which non-overlapping clock phases are obtained in the 48 MHz version.

The TMS 9900 requires a 4 phase, 12V clock. A frequency of 3.0 MHz was chosen for the TI-99/4. This clock is independent of, and asynchronous to, the 10.73863 MHz VDP clock (and thus the 447.443 KHz GROM/Sound Generator clock). The clock generator has perhaps five major functional blocks.

3.2.1 Oscillator Logic

The oscillator section requires a parallel LC tank and a crystal for its operation. See the TI data sheets on this device for the type of crystal required, as well as for the L and C values.

3.2.2 Phase Overlap Control

The second function as described here is not present in the 12

MHz version of the chip. A divide by four twisted ring (Johnson) counter is used to develop a 75/25 active HIGH strobe to clock the 4-phase clock generator section. This is a 12 MHz strobe (48 divided by 4), and the 25% portion of the period is used for a disable time to insure the non-overlapping requirement. This is a 62.5 ns wide pulse if rise and fall times are ignored. It also leaves some 20.83 ns of dead time between phases. This strobe is buffered, and presented to the outside world as a 12 MHz system clock. It is not used by the 99/4. An Oscillator Input was provided at the point of decoding the divide by four counter, but the TI-99/4 does not use that signal either.

3.2.3 Four Phase Clock Generator

The third function is the 4-phase clock generator with buffering for both active HIGH 12V MOS and Active LOW TTL outputs. The generator itself is comprised of a two bit Twisted Ring counter. Three input NAND gates are used to decode the four possible states of the counter with an additional disable input to insure phase non-overlapping. The 12V buffers are sized to drive the large capacitive load of the TMS 9900 clock inputs, and coupled with the added voltage swing over that for TTL, they produce an output that lags that of the TTL buffers. This skew is specified to be typically 8 ns, but no minimum or maximum values are listed. The current return path to the driver from the TMS 9900 for the 12V buffers should be treated with care, as the rate of change of current with respect to time is quite high here. It does not take much inductance to generate a significant voltage spike at these voltage swings and transition times.

3.2.4 Synchronizing Function

The last clock chip function provides a single synchronized bit that may be used with the TMS 9900. The bit is synchronized with respect to the trailing edge of the TTL Phase 3-. This input is labeled as "FFD" on the Data Sheets, and the output is "FFQ". The TI-99/4 uses this circuit to generate the master reset to the Console and the I/O Port.

3.3 DATA BUS TRANSLATOR

The Bus Translator that is used to obtain an 8-bit System Data Bus from the 16-bit TMS 9900 Data Bus is the subject of the next discussion. The Bus Converter not only provides a data path translation, but it also provides a BYTE Address bit (A15). In addition, it modifies the WE- output of the TMS 9900, and adds WAIT states to the TMS 9900 memory access cycle by manipulating the TMS 9900 READY input.

3.3.1 WRITE Cycle Operation

From a common sense point of view, a WRITE operation by the TMS 9900 should require only combinational logic in the Data Bus routing. This is because storage is already built into the TMS 9900 that allows it to keep data on its 16-bit Data Bus during a Memory WRITE as long as its READY input is LOW. The LSBY path from the 16-bit bus is a SN74LS244 Bus Driver chip, and the path for the MSBY is a SN74LS245 Bidirectional Bus Driver. The reason for the '245 will be seen later. Figure 3-1 shows the data flow diagram for the Bus Converter. The LSBY will always be written first.

3.3.2 READ Cycle Operation

A READ operation from the 8-bit Data Bus is considerably more complicated than a WRITE because there must be storage for the first byte while the second is being fetched to assimilate the 16-bit word required by the TMS 9900. A SN74LS373 Octal Tristate Register is provided on the LSBY portion of the 16-bit bus for this storage, because that byte is fetched first. The MSBY is fetched second, and it is gated directly to the MSBY half of the TMS 9900 Data Bus by the SN74LS245 Bidirectional Bus Driver chip.

3.3.3 Data Bus Translator Timing and Control

The timing and control necessary for the Data Bus Translator is provided primarily by a SN74LS194A Shift Register. The S0, S1, and Clear inputs are manipulated to provide Shift Right, Hold, and Clear operations of the shift register. Only sections A, B, and C are used, and they are connected as a 3-bit Twisted Ring Counter. The count sequence, when counting, is 000>100>110>111>011>001>000 (Qa,Qb,Qc ordering). There is no chance for this counter to hang up cycling in the two invalid states, since the shift register "RESET" input is driven; thus, no additional logic was provided to detect and correct for either of the two invalid counts (010 or 101) found in a 3-bit Twisted Ring Counter. This design yields six TMS 9900 clock cycles for every 16-bit access (with no additional WAIT states); therefore, each byte operation is three 1/3 us clock cycles long (1 us per byte). The LSBY is operated on first, and A15 is equal to Qc'.

In many cases it is logically easier to say when an operation is NOT to occur rather that when it IS to occur. This is true for the Bus Converter, for it is NOT to operate for either internal ROM or RAM accesses. The Bus Converter IS to function for all other memory cycles, though. A 2-Input NAND gate is used to detect when either the ROM or RAM bank is selected by the TMS 9900 for an access cycle. When either of the active LOW ROM or RAM bank chip selects occurs, the output of the 2-NAND goes to a HIGH level. This output is "Active Low ANDed" with MEMEN- from the TMS 9900 in a 2-OR gate

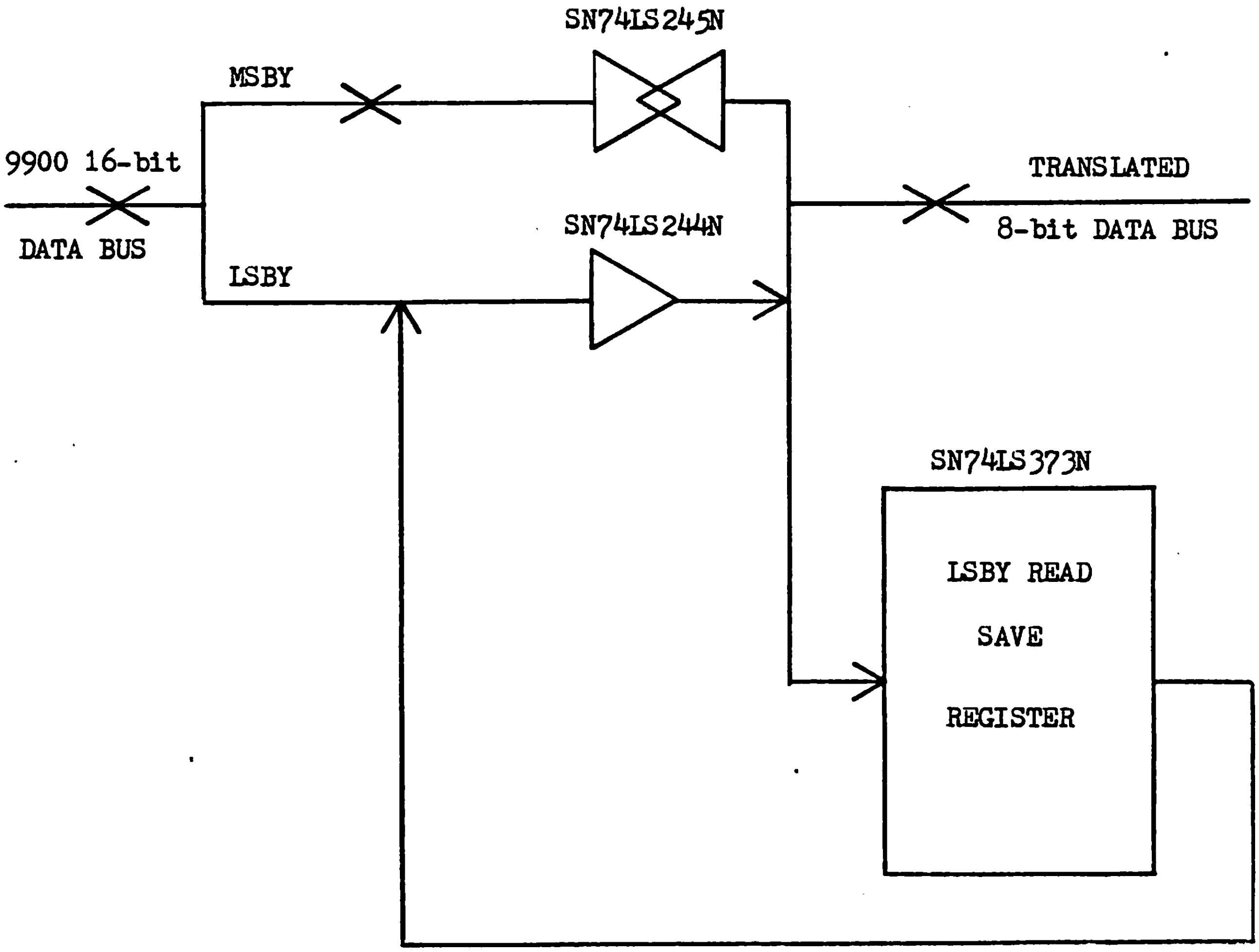


FIGURE 3-1
DATA BUS TRANSLATOR DATA FLOW DIAGRAM

to provide a logical expression of ((MEMEN)(RAM + ROM)')'. This expression feeds the S1 input of the Shift Register, and is then inverted and fed to the Active LOW CLEAR input of the Shift Register. Thus, the shift register is allowed to either SHIFT RIGHT or HOLD (if System Ready is LOW) when MEMEN- is true AND neither the RAM or ROM bank is chip selected.

For memory cycles requiring conversion to 8-bits, the timing is as per Figure 3-2. Notice from this Figure that A15 is the complement of Qc, and from the System Schematics that A15 is combined with CRUOUT from the TMS 9900 to form the 8-bit bus signal A15CRU. This is a similar signal to that on the Texas Instruments TMS 9995 Microprocessor. The TMS 9995 will not replace the TMS 9900 in the TI-99/4 for several reasons. One of the biggest ones is that the TMS 9995 accesses the MSBY first, and set-up times on GROMs, the VDP chip, and devices in the Peripheral Expansion Unit are not observed. All of these devices are accessed in the MSBY half of a memory cycle.

3.3.3.1 TMS 9900 READY Control

Once a Bus Conversion cycle has started, a 2-NAND gate is enabled to gate the decode of the Shift Register to the TMS 9900 READY status line. READY will be forced LOW (NOT READY) until the Shift Register count is Qa'QbQc (011). Qb is treated as a "Don't Care", and is not included in the decode logic. The Count of Qa'Qc occurs AFTER the TMS 9900 has sampled READY; therefore, the TMS 9900 does not see that its READY input is HIGH until the next Phase 1 time period (R=Qa'Qb'Qc). It will be one more clock cycle past the next Phase 1 before the TMS 9900 terminates a complete 16-bit memory cycle (R=Qa'Qb'Qc'). There may be more than one memory cycle needed by the TMS 9900, and if another is required, the Shift Register keeps right on counting in the same manner as it previously did. A Load Immediate type of TMS 9900 instruction will require two back-to-back memory cycles, and a Return With Workspace Pointer instruction requires three memory cycles before MEMEN- returns to a HIGH level. This mode of MEMEN- operation makes the TMS 9900 much more difficult to interface to Pseudo Static and Dynamic memory devices.

System Ready is used to add additional clock cycles into a byte access by causing the Shift Register to enter a HOLD mode if SYSRDY is LOW. There is no reason why that one byte in the pair could not have more cycles than its mate, although it seems to not to have ever been used this way. Notice from the TI-99/4 Schematics that System Ready is synchronized with a SN74LS74AN to obey the required set-up time with respect to the Phase 1 clock.

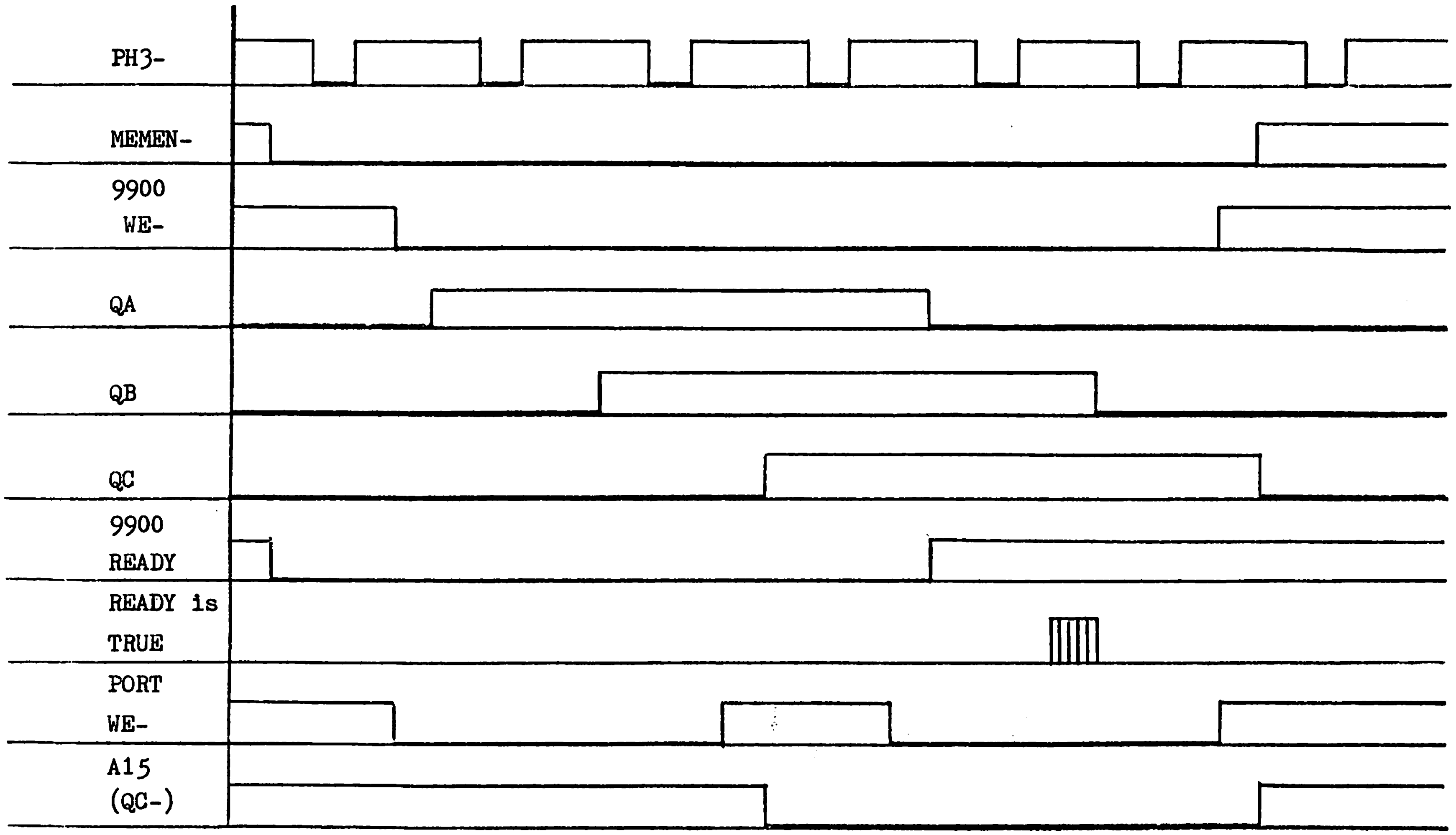


FIGURE 3-2

BUS TRANSLATOR TIMING DIAGRAM

3.3.3.2 WE- Generation

The WE- output from the TMS 9900 is modified by the Bus Converter logic to provide a WE- for each byte. The WE- generated for the 8-bit Data Bus goes inactive (HIGH) shortly before A15 goes from a HIGH level to a LOW. The 8-bit Data Bus WE- returns to its active level on the Trailing Edge of the next Phase 4 TTL clock. This is accomplished by using a SN74LS74AN Flip Flop (FF) and a 2-NAND gate. The FF provides the between byte inhibiting action when it is clocked RESET and then SET again. The equation for this is ((TMS 9900 WE)(FFQ))'.

3.3.3.3 Data Bus Driver Control

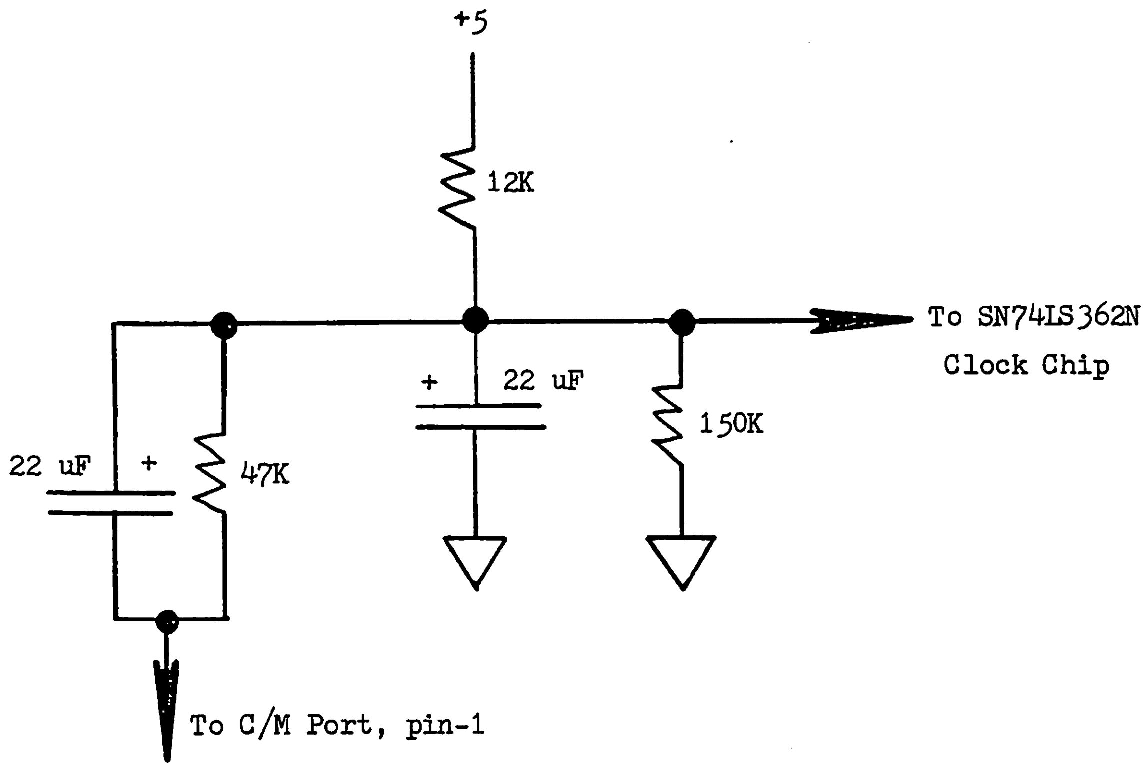
It may seem that the SN74LS245 Bidirectional Bus Driver ENABLE input must be active for the MSBY of all 8-bit memory cycles. Note from the TI-99/4 Block Diagram in Figure 2-1 that the VDP chip is connected directly to the TMS 9900 Data Bus; therefore, the '245 must NOT be enabled for a VDP READ operation even though the VDP is accessed with the Bus Converter functioning. A 2-NAND gate is included for this inhibit operation. If Qc of the Shift Register is at a HIGH level (A15 is at a LOW level, and thus a MSBY access is in progress AND that neither the RAM or the ROM is selected), AND the VDP is NOT selected for a READ access, then the '245 will be enabled. If this inhibit function were not present, both the '245 and the VDP would try to drive the TMS 9900 Data Bus at the same time for a VDP READ. The '245 would probably win this contest, although sometimes strange things happen during data bus conflicts. It makes no difference whether or not the '245 is enabled for a WRITE to VDP operation, as only the TMS 9900 is driving the Data Bus at this time.

The direction input of the SN74LS245N Bidirectional Data Bus driver is driven by the complement of the TMS 9900 memory direction control signal DBIN.

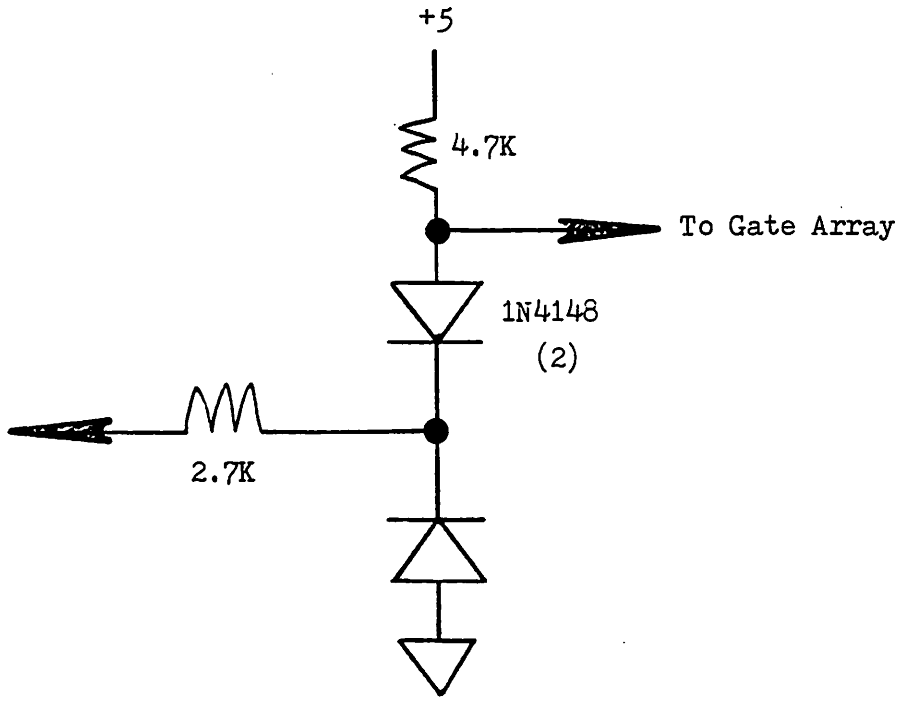
3.4 SYSTEM RESET

Another interesting circuit is the one that is used to reset the TI-99/4 when the Command Module is being plugged in. This circuit is drawn in Figure 3-3. Notice that there exists a capacitive path from pin 1 of the Command Module socket to Ground. Pin 1 of the earlier Command Modules is connected directly to -5V. A momentary short circuit path from -5V to Ground exists through the two 22 uF capacitors. Later Command Modules had a 100 Ohm series resistor to compensate for this. The TI-99/4A with a Gate Array had measures taken to eliminate this capacitive path.

Large shunt resistors with the 22 uF capacitors in this circuit



Most Commonly Used



Late Model TI-99/4A Machines

FIGURE 3-3

COMMAND MODULE RESET CIRCUITS

provide a several second time constant. This is why one must wait a short time period after first unplugging a Command Module and before plugging it back in to obtain a reliable reset the TI-99/4. Switching the power switch from ON to OFF, and back to ON again too rapidly will not reliably reset the Console for the same reason.

3.5 MEMORY SPACE DECODING

The Memory Space Decoding will be the next subject to be discussed. The TMS 9900 operates on a normal 16-bit, 500 ns memory READ access time (667 us cycle time), and the SN74LS138N decoder propagation delay eats into this access time somewhat. Couple this with the fact that a 1K Ohm resistor is in series with the 4K word ROM Chip Select line, and one may have problems using 450 ns EPROMs that have equal Address and Chip Select access times. For those wishing to do their own System ROMs in EPROMs, it will be a good idea to remove the 1K Ohm resistor from the circuit. This resistor was changed to 100 Ohms for late production TI-99/4As.

There are two SN74LS138N demultiplexers and one SN74LS03N Open-Collector 2-NAND gate used in the actual memory space decode. All on-board devices as well as the Peripheral space, the Command Module space, and the Speech Synthesizer space are decoded with this network. Two levels of '138 decoding are required for the Sound Generator, VDP, Speech Synthesizer, and GROM chips. A worst case 70 ns propagation delay may occur for some of these device decodes.

The first '138 demultiplexer in the network decodes the Most Significant (MS) three bits of the Address space (for 8K blocks), and is gated with the TMS 9900 memory control signal MEMEN-. The active LOW decode equations are as follows.

SYSTEM ROM = (MEMEN)(A0')(A1')(A2')

PERIPHERAL SPACE = (MEMEN)(A0')(A1)(A2')

C/M SPACE = (MEMEN)(A0')(A1)(A2)

PARTIAL SPACE = (MEMEN)(A0)(A1')(A2')

These four decodes are all that are used out of the eight available on the '138. The first three decodes should be well understood. The PARTIAL decode is used to decode the on-board SRAM, Sound Generator, VDP, Speech Synthesizer, and the GROM chips. The additional SRAM logic is comprised of three cascaded 2-OR gates to "Active LOW" AND the next three lower address bits with the PARTIAL SPACE decode. The second SN74LS138 and a 2-NAND are used to obtain the other decodes mentioned. This demultiplexer is enabled if:

(Address bit-5 is a "0" OR if a WRITE is active) AND

(Address bit-15 is a "0") AND

(it is enabled by the first decoder)

3.6 Joystick Driver

The Joystick driver circuit on the TI-99/4A consists of two NPN emitter followers, and nothing more than resistors connected between the +5V and -5V buses. The Joystick driver is required to sink current, but only a small amount can be handled in the sink resistor. The emitter follower on the TI-99/4A compensates for the series diode voltage drop in the joysticks by dropping the drive voltage to the joystick to about .6V below Logic Ground to regain TTL compatibility. See Figure 3-4 for these two circuits, and Figure 3-5 for the joysticks themselves.

3.7 KEYBOARD

The TI-99/4A keyboard has a double action Alpha Lock switch, and is driven by a NMOS output of the TMS 9901. All of the other keyboard column drivers are open collector LS TTL, and when the Alpha Lock switch was closed, some key closures required the LS TTL drivers to "over drive" the NMOS output (which they quite readily did). There was not nearly enough current sink capability in the NPN emitter follower (as a 1.5K resistor provided the current sinking) to over drive the NMOS. Therefore, the Joysticks are not functional when the Alpha Lock Switch is closed. The late model TI-99/4As that had a gate array for the timing and control eliminated this condition.

3.8 CASSETTE INTERFACE

The Cassette circuitry may be divided into three parts for this discussion.

3.8.1 Motor Drive Circuit

The motor control circuit is comprised of two NPN transistors, an optically coupled isolator, and a few discrete components. The half of the circuit driving the cassette remained the same from Console to Console, but the driving circuit for the OCI LED changed from a transistor driver to an open collector TTL device on late models. A transistor was added to the one in the OCI to provide a

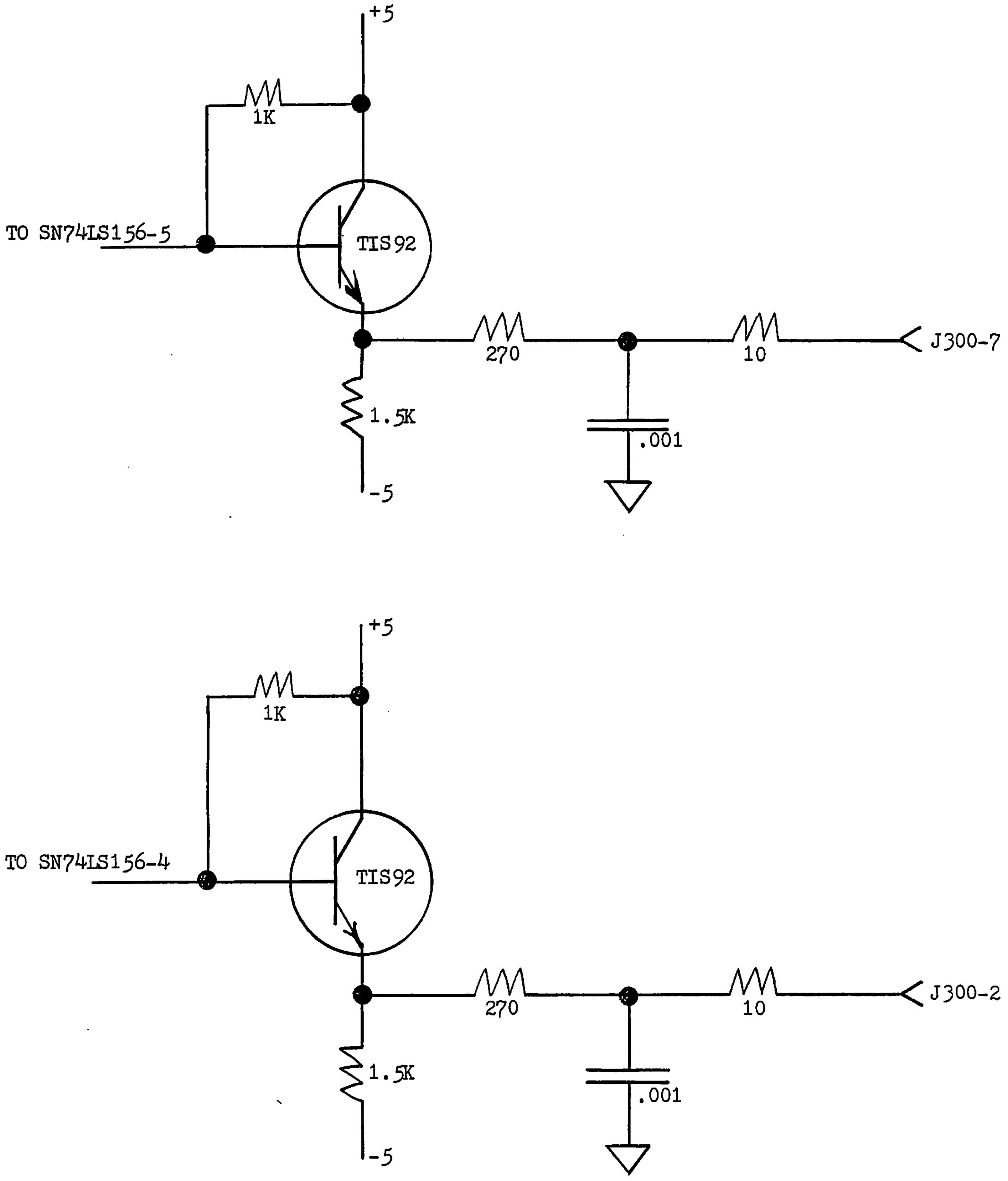


FIGURE 3-4

Joystick Drive Circuit

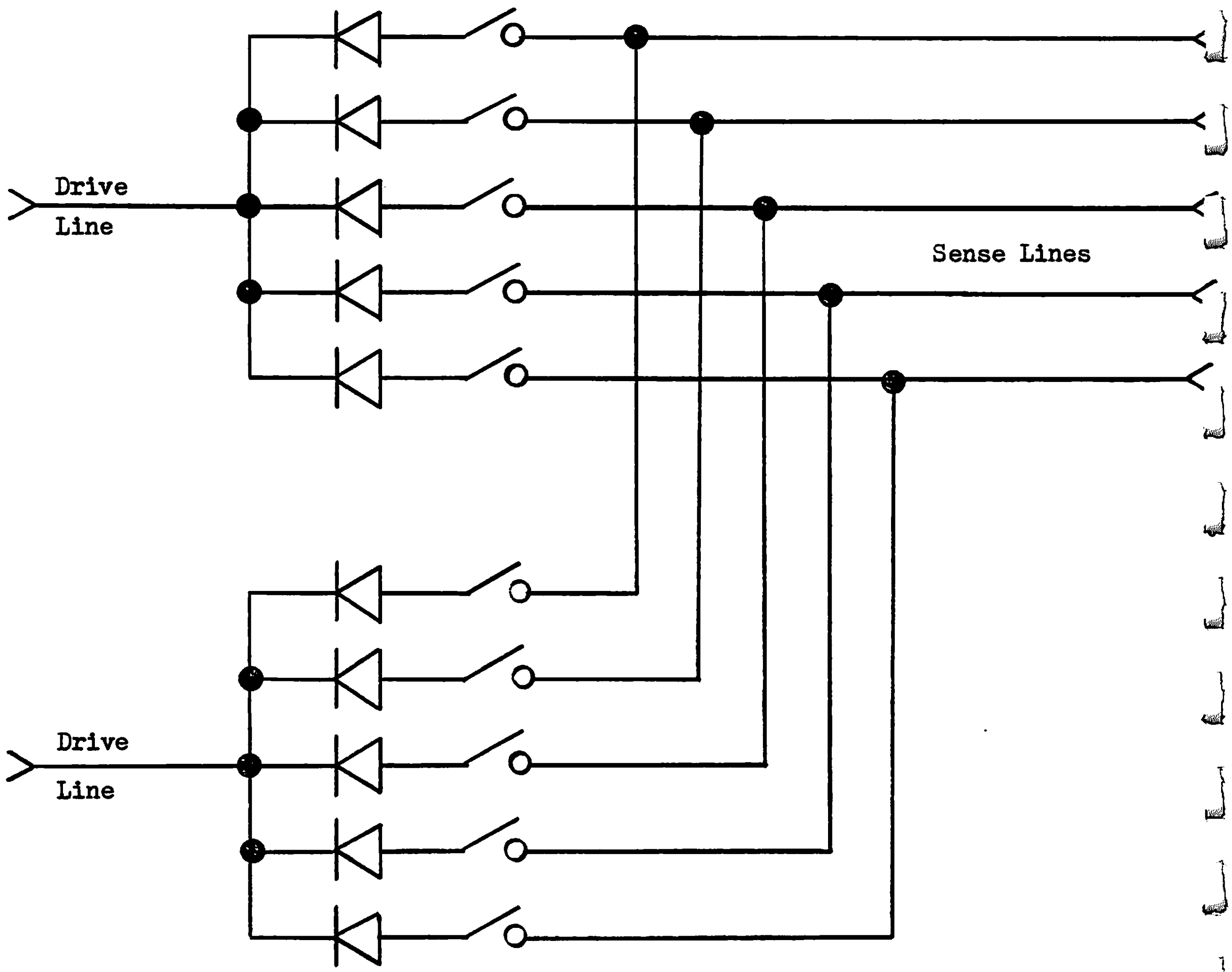


FIGURE 3-5
JOYSTICK CONFIGURATION

Darlington driver for the cassette motor. This connection causes a 1V minimum drop across the motor "switch". In some cases this will be enough drop to cause undesirable operation of the Cassette Drive (due to low motor voltage). The OCI is required provide the necessary common mode operating voltage between the Cassette Drive and the Console. The schematic of this is drawn in Figure 3-6.

3.8.2 Cassette WRITE Circuit

The WRITE circuit is shown in Figure 3-7, and is a somewhat confusing circuit. It appears to be a integrating circuit followed by a differentiating circuit (with different time constants), but a filter capacitor in the circuit obscures the differentiation. It appears that the capacitor on the output side was added for RFI purposes, but it is the same magnitude as that for the differentiation. A .01 uF capacitor is connected ahead of the differentiating circuit; thus, integration is provided ahead of differentiation. This is confusing to say the least.

3.8.3 Cassette READ Circuit

The READ circuit for the Cassette data is comprised of two 4558 operational amplifiers. The first is connected as an AC coupled inverting amplifier, and the second is connected as a comparator. The component values were changed at various times during the life of the TI-99/4, but the general circuit itself did not change appreciably. This circuit is shown in Figure 3-8.

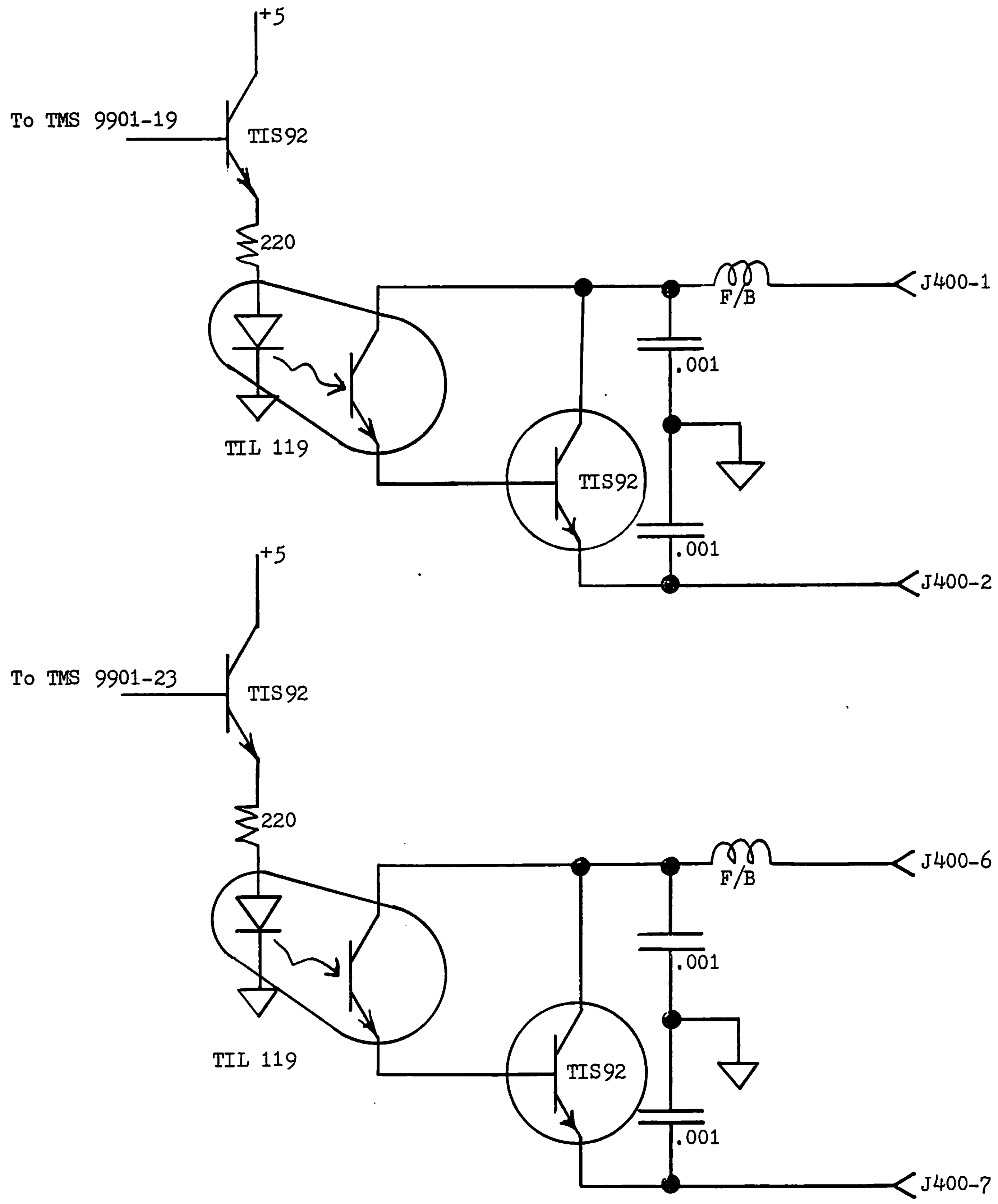


FIGURE 3-6

Cassette Motor Drive Circuit

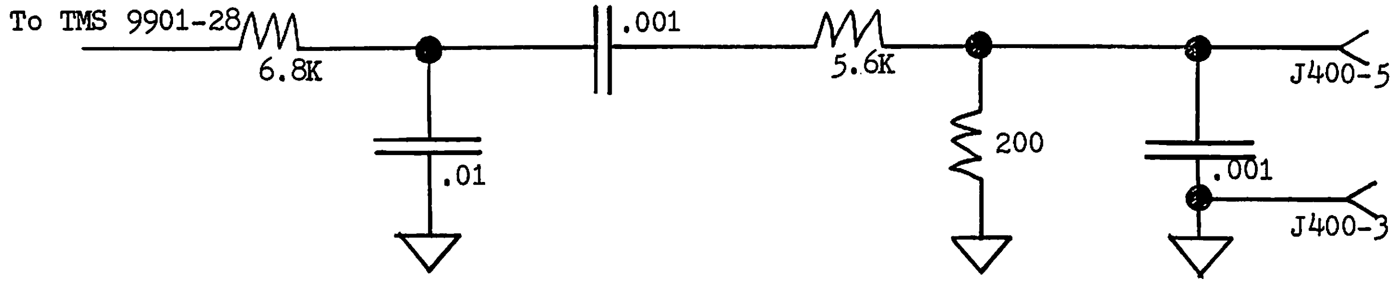
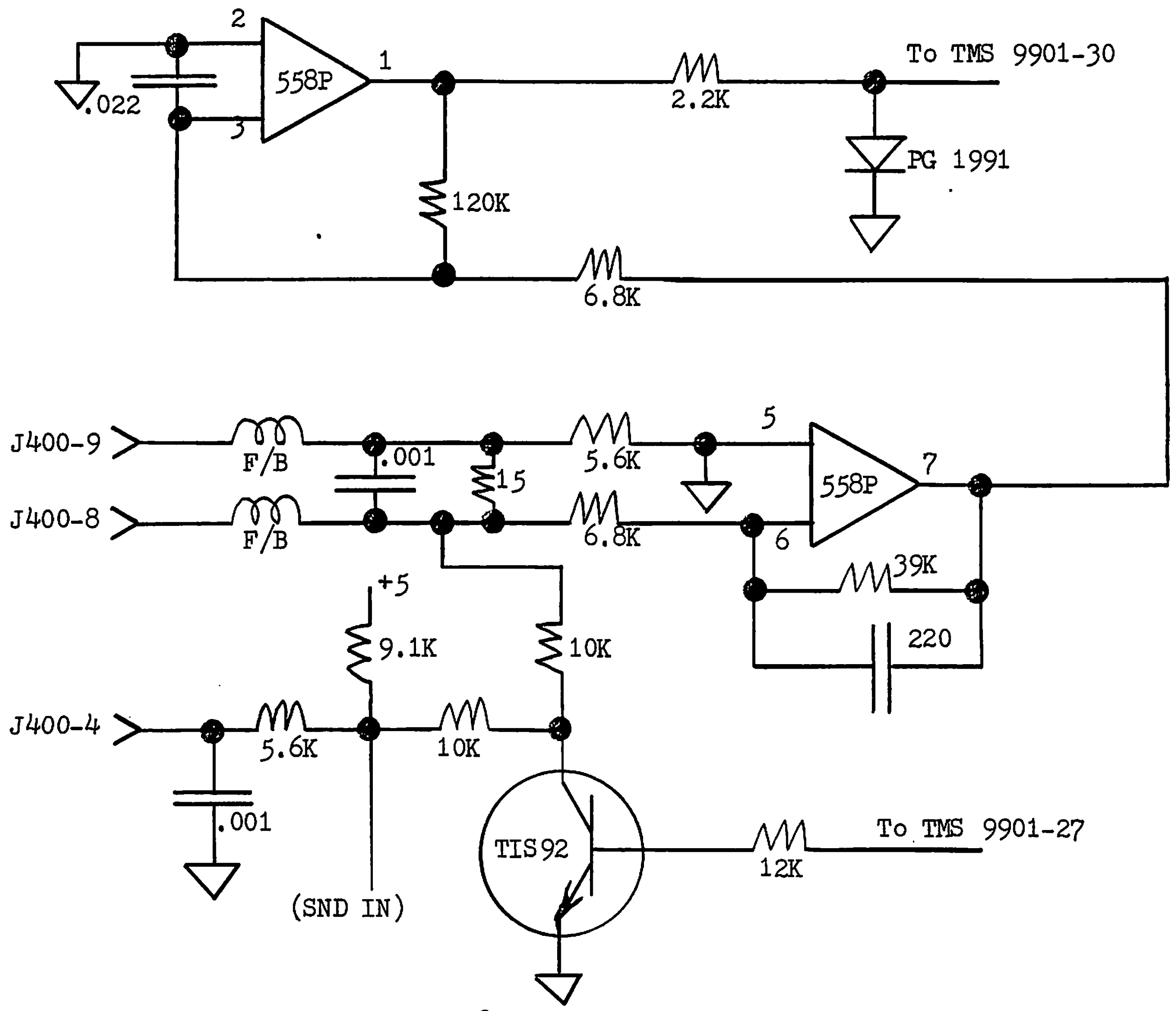


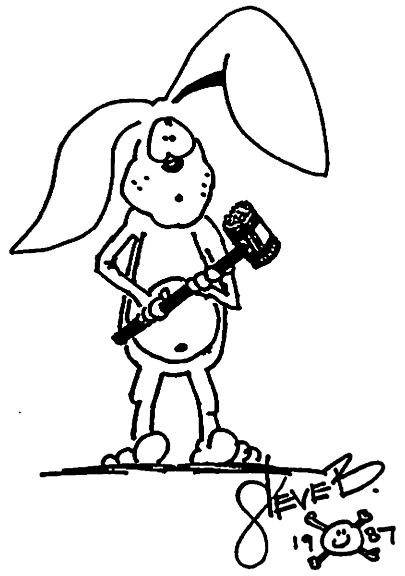
FIGURE 3-7

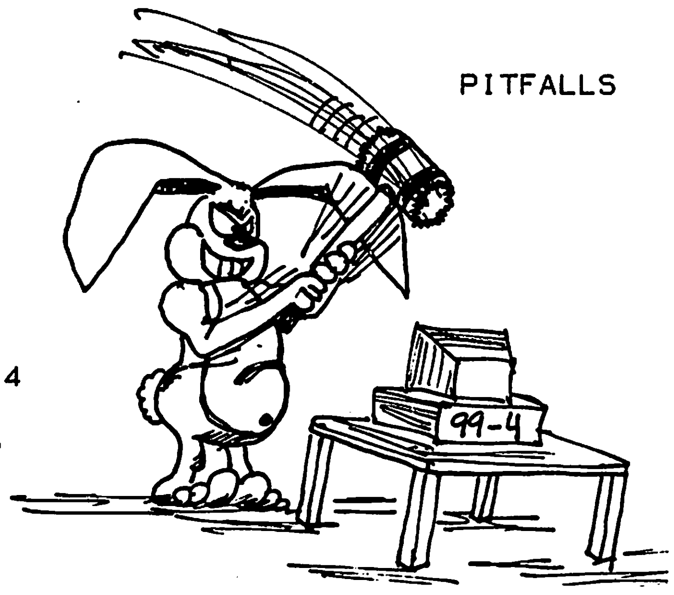
Cassette Write Circuit

FIGURE 3-8

Cassette Read Circuit







CHAPTER 4

PITFALLS

4.1 SUMMARY OF PITFALLS

There are several pitfalls in both designing equipment to interface with the TI-99/4, as well as in writing Assembly Language programs for it. These idiosyncrasies will be discussed in this Chapter. A brief description of each is listed below.

* Never READ from the Sound generator chip.
* Allow 8 us between successive accesses of the VDP chip.
* Allow 5.6 us between successive accesses of a GROM chip.
* Never READ from the Speech Synthesizer with code executing in the Memory Expansion Unit.
* Never use any of the TMS 9900 "EXTERNAL" instructions or the "IDLE" instruction.
* The TMS 9900 memory control signal MEMEN- does not go inactive between back-to-back (successive) memory accesses.
* The destination address of either a MOV or a MOVB instruction is READ from before being WRITTEN to.
* The P-MOS GROM chips are not TTL compatible.
* The Joystick drivers are not strong enough to over-drive the TMS 9901 driver for the ALPHA shift lock key.
* A resistor must be in series with the RESET pin on Command Modules.
* The "CRUIN" input signal to the TMS 9901 is unbuffered in the Console.
* 450 ns EPROMs may be too slow for use in the Console.
* Synchronization is necessary to interface between logic

that is clocked by a VDP derived clock and the TMS 9900.

* The pin out of the Command Module Port is not the same as the marking on the 36-pin connector.

* Disk Drives are "T" fed; therefore, two sets of terminating resistors are required.

* The "Side Car" Memory Expansion Module passes only Peripheral Memory Space data on the data bus.

4.2 SOUND GENERATOR CONSIDERATIONS

The Sound Generator chip (TMS9919 or SN94624N) is the first area we will examine, and this characteristic exists in all but the very latest TI-99/4As. DO NOT TRY TO READ FROM THE SOUND GENERATOR! The Sound Generator expects to see its WRITE input go LOW along with the Chip Select operation. The system will go "NOT READY", and hang up if WE- does not behave in the expected manner. There is no decoding logic provided to guard against a READ operation. This characteristic was eliminated on the late models with a Gate Array for decoding, etc.

4.3 VIDEO DATA PROCESSOR CONSIDERATIONS

The Video Data Processor (TMS9918 VDP) chip is another element to be approached with care. There must be sufficient time between one access and the next to allow it to complete the first task before taking on another. This applies to both the READ and WRITE Chip Selects. The Data Book specifies a 2 us inactive time after a register type of operation, and an 8 us inactive time after a VDP memory access. The former will present no problems to the programmer, but the latter will. Be sure to insert NOP instructions (>1000) between successive MOVB instructions to obtain this 8 us delay. Keep in mind that a MOVB to VDP transfers the byte during the latter portion of the instruction, and that a MOVB from VDP transfers that byte during the first portion of the instruction execution. These two instructions executed together out of the 16-bit memory will allow only a 3.5 us inactive time, because in essence, there is only an instruction access between the two VDP chip accesses.

4.4 GROM CONSIDERATIONS

The TMC0430 Graphics Read Only Memory chips require much the same type of consideration as does the VDP chip. They operate "kind of" closed loop with the TMS9900, but they too must be inactive 2.5 GROM clock cycles (about 5.6 us) between successive accesses. The inactive time applies to both data and register operations. When the inactive time is violated, the problem is one that may remain hidden for GROM Address Read operations in some systems because the GROM chips in the system are all connected in parallel. Enough may function correctly in one system (even though the inactive time has been violated), but not in another with GROMs of different production lots. Data operations will not function correctly in the abused chip, though. The same instruction scenario exists here for MOVB instructions accessing GROM chips. It is wise to insert one NOP instruction (>1000) between successive MOVB instructions accessing the GROM chips.

4.5 SPEECH SYNTHESIZER CONSIDERATIONS

Direct communication with the Speech Synthesizer from code in a Memory Expansion Unit will cause some interesting problems during a READ operation. The synthesizer chip is much like a TMS1000 uC. It must remove itself from the TI-99/4 system Data Bus with its own software even though the Console address decoding logic has deselected the chip. It will take at least 20 us for the chip to disable its data bus drivers; so, it will still be driving the Data Bus when the next instruction is fetched from the Memory Expansion. Therefore, this chip may not be generally accessed by code in the external Memory Expansion unit since both use the same data bus to the console. This is true of either the old "side car" peripheral or the Peripheral Expansion Box. You must transfer code from the Memory Expansion Unit into the console, and execute the code in the console. If the Console SRAM is to be used, an area must be saved to be restored later. No system level software may be used until SRAM is restored. In any case be sure that at least 20 us have elapsed from the READ access until control is returned to the code in the Memory Expansion. Experience has shown that strange things will happen if you do not! This is also a condition that some systems will tolerate, but others will not. The problem was discovered on systems that would not function correctly.

4.6 ILLEGAL TMS 9900 INSTRUCTIONS

There are several instructions that must not be executed in all but the latest TI-99/4A Consoles, and all are due to the lack of additional CRU space decoding logic in the Console to derive the CRU strobe CRUCLK-. The most significant three bits of the address bus were not included in the decode of CRUCLK*; therefore, one may not use any of the "EXTERNAL INSTRUCTIONS": IDLE, RSET, CKON, CKOF, and LREX. All of these instructions cause the TMS9900 to generate a

CRUCLK that would be interpreted by the TI-99/4 logic as a CRU associated operation even though it is not one. The result of this MAY be the changing of the internal TMS9901 or the TMS9902 UART(s) in the RS232 peripheral.

4.7 TMS 9900 MEMORY ACCESSING

All of the previous items have been related to software, but the following ones are related to hardware design. The first of these relates how the TMS9900 accesses memory. The memory control signal of the TMS9900 is MEMEN-. When this is active LOW, it indicates to the attached memory that a memory cycle is in progress. Furthermore, MEMEN- does not go inactive (to a HIGH level) between successive or Back-to-Back cycles; therefore, hardware must comprehend this if dynamic memories (DRAMs or Pseudo Static ROMs) are to be interfaced to the TI-99/4. This problem may not show up in all designs, but, for instance, if the Workspace is located in the same DRAM as the instruction code, a Load Immediate instruction will request two back- to-back 16-bit memory cycles. A Return with Workspace Pointer will cause three 16-bit back-to-back memory cycles to occur. Notice also that one 16-bit memory cycle translates to two 8-bit back-to-back memory cycles at both the I/O and the Command Module Ports. These two 8-bit cycles are performed without MEMEN- ever leaving the active state. The RTWP instruction causes six 8-bit back-to-back memory cycles to occur. A simple state counter may be implemented to comprehend this, or one of several other methods may be used in solving the problem of differentiating between successive memory cycles.

4.8 TMS 9900 MOVE INSTRUCTION EXECUTION

Another characteristic of the TMS 9900 that is handy to know prior to doing hardware design relates to the execution of both a Move Byte and a Move (word) instruction. Since the TMS 9900 is organized around a 16-bit Data Bus, and all accesses involve the full 16-bits, a MOVB instruction requires the destination address to be READ before a byte may be moved to that same destination address. This is because one of the two bytes at the destination address must not be altered when the opposite one is replaced, and the memory access involves both bytes at the same time. The TMS 9900 must first READ both bytes from the destination address, discard the byte to be replaced, concatenate the new byte with that which must not be altered, and THEN save them both in a 16-bit memory WRITE operation. The problem that can exist here is that the READ operation from the destination address will possibly disturb something in the chip being accessed. This is most certainly true with GROM chip accesses. That is the reason why there are different READ and WRITE addresses for both the GROMs and the VDP! THIS SAME THING HAPPENS

WITH WORD MOVES (the MOV instruction). It makes no difference what type of addressing is used, the destination address location is first read.

4.9 8-BIT DATA BUS CHARACTERISTICS

Another hardware restriction relates to both the command Module Port and the I/O Port. The GROM chips are connected directly to the 8-bit Data Bus. These are PMOS chips, and are not TTL compatible. To gain compatibility, a driven resistor network was included on the 8-bit system data bus to provide a pull-up function for WRITES, and a pull-down function for READs. Anything interfaced to this Data Bus must comprehend this loading as far as drive is concerned, and in addition must not disturb either the pull-up or pull-down operation. It must be also kept in mind that the GROM chips are relatively poor line drivers. A SN74LS245 Bus Driver was included on the Peripheral Expansion Box interface cable for increased drive levels.

4.10 ALPHA SHIFT LOCK KEY CHARACTERISTICS

An Alpha Shift Lock key was included on the TI-99/4A keyboard. A problem exists when the switch is closed (in the LOCKED position), and it relates to how the circuit is implemented. The Alpha Lock Key is directly driven from the TMS 9901 Interface chip. The driver here is a NMOS totem pole configuration rather than an open collector LS TTL type like the other Keyboard drivers are. This NMOS output presents no problem to the SN74LS156...the TTL output simply over powers the TMS 9901 NMOS output. The Joystick Drivers, on the other hand, are passive pull-down drivers which must take the Joystick drive voltage below Ground to compensate for a diode drop in the Joysticks themselves. THE JOYSTICK DRIVERS DO NOT HAVE ENOUGH DRIVE TO OVERPOWER THE TMS 9901 OUTPUT. Thus, the Joysticks will not function correctly if the Alpha Lock switch on the keyboard is in the LOCKED state. This characteristic was eliminated on the late consoles with a Gate Array for Timing and Control.

4.11 COMMAND MODULE PORT RESET CONSIDERATIONS

The Command Module Port is designed to allow one to reset the TI-99/4 when a Command Module is being plugged IN. This is accomplished when pin 1 on the Command Module is connected to the -5V power. It may be seen from the System Schematics that the Command Module pin-1 connects to Logic Ground in the console through two series 22 uF capacitors. This circuit will reset the console, but in the process, it momentarily shorts -5V to Ground. Later

Command Modules had a series 100 Ohm resistor to eliminate this direct short. This characteristic was eliminated on the later model consoles with a Gate Array for the Timing and Control.

4.12 CRUIN DATA LINE CONSIDERATIONS

The CRU Input line is one that should be treated with care. The TMS 9901 CRUIN output must drive all of the way down the cable and into the Peripheral Expansion Box in addition to driving the TMS 9900 CRUIN input in the console. There should be no appreciable capacitive load presented to this line, as it is already working pretty hard as it is.

4.13 SYSTEM ROM CONSIDERATIONS

If one undertakes replacing the TMS 4732 system ROMs in the console with EPROMs, care must be taken in selecting EPROMs that are fast enough. If the EPROM Chip Select input is also a power down control (commonly used to conserve power in these sizes and larger ones, too), 450 ns parts may be too slow. This is due to both the ROM Address Space decode and a series 1K Ohm resistor in the Chip Select line. This resistor was changed to a 100 Ohm value in late TI-99/4As, though.

4.14 SYNCHRONIZATION

There are two crystal oscillators used in the TI-99/4. One is for CPU use, and the other serves the video, Sound Generator, and the GROM chips. Care must be used in interfacing the latter devices (or any added to the VDP clock) to the TMS 9900, as the TMS 9900 requires PH1- set-up and hold times. A flip-flop is used in the console to synchronize System READY for the TMS 9900 READY input. The GROM chips can be out of synchronization with each other, but the parallel connection of their READY outputs accommodates the problem effectively.

4.15 COMMAND MODULE PORT PIN ORDERING

The pin ordering on the 36-pin GROM Port connector is "side reversed". Pin #1 marked on the connector is in reality the functional pin #2. This occurs on a paddle board and connector forming a right angle to go from the Console Main Logic PCB to the physical location of the Command Module plug in.

4.16 DISK DRIVE CONSIDERATIONS

The Disk Controller PCB in the Peripheral Expansion Box "T" feeds disk drives connected to it. If two or more drivers are connected, and if one of the two is in the Expansion Box, then both drives must have terminating resistors to allow the Controller Board to properly drive both drives. The terminating resistors must be larger than those furnished in the drive in order not to over load the drivers on the Disc Controller Card. Terminating resistors in the range of 510 ohms will suffice.

4.17 MEMORY EXPANSION MODULE CHARACTERISTICS

The old "Side Car" Memory Expansion Unit is a link in a "Daisy Chain" connection of Peripherals to the TI-99/4 Console. This peripheral has a bidirectional data bus driver that is enabled only for the Peripheral Address Space; therefore, TI requires the Memory Expansion Module to be the first peripheral connected to the Console. If the Speech Synthesizer is used, it must be connected between the Console and the Memory Expansion Module. This characteristic only causes problems to those who wish to monitor general transactions on the Peripheral Data Bus.

CHAPTER 5

TMS 9900 H/W DESCRIPTION

5.1 TMS 9900 ORGANIZATION

The TMS 9900 is a 16-bit machine that supports a 64K BYTE address space organized as 32K words. It has a 16-bit Data Bus. There are eight memory control signals, three Communications Register Unit (CRU) data/control lines, seven interrupt lines, and several other inputs. The four-phase clock inputs are the only +12V logic involved. An active LOW TTL level Phase 3 is provided on the TI-99/4 I/O Port.

5.2 TMS 9900 PIN DEFINITION

The pin definition (functionally grouped) is as follows.

PIN	FUNCTION
27	VDD, +12V POWER
2, 59	VCC, +5V POWER (PIN 59 FEEDS I/O BUFFERS)
26, 40	VSS, GROUND (PIN 40 FEEDS I/O BUFFERS)
1	VBB, -5V POWER
8	PH1, 12V PHASE 1 CLOCK
9	PH2, 12V PHASE 2 CLOCK
28	PH3, 12V PHASE 3 CLOCK
25	PH4, 12V PHASE 4 CLOCK
41	D0, MSB, DATA BUS
42	D1, DATA BUS
43	D2, DATA BUS
44	D3, DATA BUS
45	D4, DATA BUS
46	D5, DATA BUS
47	D5, DATA BUS
48	D7, DATA BUS
49	D8, DATA BUS
50	D9, DATA BUS

PIN	FUNCTION
51	D10, DATA BUS
52	D11, DATA BUS
53	D12, DATA BUS
54	D13, DATA BUS
55	D14, DATA BUS
56	D15, LSB, DATA BUS
24	A0, MSB, ADDRESS BUS
23	A1, ADDRESS BUS
22	A2, ADDRESS BUS
21	A3, ADDRESS BUS
20	A4, ADDRESS BUS
19	A5, ADDRESS BUS
18	A6, ADDRESS BUS
17	A7, ADDRESS BUS
16	A8, ADDRESS BUS
15	A9, ADDRESS BUS
14	A11, ADDRESS BUS
13	A12, ADDRESS BUS
12	A13, ADDRESS BUS
11	A14, ADDRESS BUS
10	A15, ADDRESS BUS
63	MEMEN-, ACTIVE LOW MEMORY ENABLE
29	DBIN, DATA BUS DIRECTION, HIGH = DATA INTO TMS 9900
61	WE-, ACTIVE LOW WRITE ENABLE (WRITE TO MEMORY)
7	IAQ, ACTIVE HIGH INSTRUCTION ACQUISITION STATUS
62	READY, ACTIVE HIGH MEMORY READY STATUS
3	WAIT, ACTIVE HIGH STATUS INDICATING MEMORY NOT READY ACKNOWLEDGED BY TMS 9900.
64	HOLD-, ACTIVE LOW REQUEST FOR TMS 9900 TO TRISTATE ITS ADDRESS BUS, DATA BUS, MEMEN-, DBIN, AND WE-.
5	HOLDA, ACKNOWLEDGE FROM TMS 9900 THAT IT HAS ACTED UPON THE HOLD REQUEST.
60	CRUCLK, CRU CLOCK USED IN COMPLEMENT FORM IN THE TI-99/4 TO INDICATE THAT THE ADDRESS BUS AND THE CRUOUT OUTPUT BIT ARE STABLE.
30	CRUOUT, CRU OUT DATA
31	CRUIN, CRU INPUT DATA
4	LOAD-, ACTIVE LOW NON MASKABLE INTERRUPT
32	INTREQ-, ACTIVE LOW INTERRUPT REQUEST
36	IC0, MSB OF INTERRUPT LEVEL CODE
35	IC1, INTERRUPT LEVEL CODE
34	IC2, INTERRUPT LEVEL CODE
33	IC3, LSB OF INTERRUPT LEVEL CODE
6	RESET-, ACTIVE LOW RESET LINE

5.3 RETURN CURRENT CONSIDERATIONS

Prudent design with the TMS 9900 allows for data and address bus currents to return to pin 40. Clock Ground return currents should return to pin 26. This simply means that bus driver chips should have a short and wide ground path to pin 40, and there should exist the same type of path from pin 26 back to the SN74LS362 clock driver chip. In addition, there should be the same short and wide ground path between pins 26 and 40. VCC pins 2 and 59 should be joined in the same manner.

5.4 ADDRESS BUS

The Address Bus serves both the 32K Word memory space and the 4K bit CRU space. The Most Significant three Address Bus bits are to be decoded to support six different instruction classes. The TI-99/4 did not decode these bits; therefore, it is illegal to use five of these six instructions. The valid class relates to CRU output operations, and is used by the LDCR, SBO, and SBZ instructions.

5.5 SIGNAL SETUP

ALL INPUTS TO THE TMS 9900 MUST BE SET UP AND STABLE BEFORE A PHASE 1 CLOCK OCCURS.

5.6 MEMORY CONTROL

The memory control by the TMS 9900 is rather unique, and is found only in TI u-Processors. The control signal MEMEN- is used to alert the outside world that there is a memory cycle starting when MEMEN- makes a transition to a LOW level. It is up to the external memory controller to step along with the TMS 9900 clock system to provide data at the appropriate time. This is no problem if fully static memory elements are used. Dynamic and Pseudo Static parts present a much more complicated problem, because more than one memory cycle may occur while MEMEN- is LOW. There are always two four-phase clock cycles if the READY input is at a HIGH level on the first Phase 1 clock following the time when MEMEN- goes to a low level. The TMS 9900 takes data on the very next Phase 1 after the Phase 1 it finds the READY input at a HIGH level. MEMEN- will stay at a low level for three consecutive accesses when executing a Return With Workspace Pointer instruction. Two back-to-back accesses will occur on a Load Immediate instruction. The normal memory (and CRU) cycle is 667 ns for a 3 MHz clock, but with Address Bus propagation delay and TMS 9900 input data set-up time, only 500 ns out of the 667 ns cycle time are left for access.

5.6.1 DBIN Control Signal

DBIN simply tells the direction of the TMS 9900 data bus. It is normally LOW, and makes a transition to a HIGH level for reading data at the same time MEMEN- goes active. In fact, the Address Bus, MEMEN-, DBIN, and IAQ (on an instruction fetch) ALL stabilize at the same time. WE- is the only signal that is bracketed by MEMEN-, DBIN, and the Address Bus.

5.6.2 Control Signal Observations

Some simple observations may be made about several of the memory control signals. IAQ may never be active if MEMEN- is inactive. The same is true for WE- with the additional constraint added that DBIN must be at a LOW level. A memory operation is always occurring when MEMEN- is LOW, and a CRU operation cannot occur at this time. A CRU operation, if it is supposed to occur, may do so only when MEMEN- is at its inactive (HIGH) level. More observations may be made with the other functional blocks of TMS 9900 signals.

5.6.3 READY and WAIT Signals

The READY input is used to add SINGLE CLOCK PERIOD times into a memory cycle. It is sampled by internal TMS 9900 logic on the first Phase 1 after MEMEN- goes LOW, and if READY is at a HIGH level, Data are taken on the following Phase 1. If READY is at a LOW level, the TMS 9900 sets the WAIT output signal on the following Phase 3. The TMS 9900 then samples READY on successive Phase 1 clocks until it finds READY at a HIGH level. Once it does, it takes data on the next Phase 1. The TMS9900 does not use READY on the Phase 1 it takes data. WAIT returns to a LOW level on the Phase 3 following the Phase 1 that READY was found to be at a HIGH level. It is important to understand that WAIT indicates the state of READY when it was sampled during a memory cycle. The required timing for READY is such that one MUST ask for it at a special time; therefore, one already knows what the state of WAIT must be. READY and WAIT may be connected together to provide 3 clock period memory cycles. READY has no control over TMS 9900 CRU operations.

5.6.4 HOLD and HOLDA Operation

The HOLD- input is used to cause the TMS 9900 to go into an inactive state that will allow Direct Memory Access by tri-stating the Address Bus, Data Bus, MEMEN-, DBIN, and WE-. The TMS 9900 will not enter a HOLD state until all successive memory accesses in an access cycle have completed if the HOLD request is made while MEMEN- is active. Once recognized, the TMS 9900 causes HOLDA to go active HIGH on the next Phase 3. Contrasted to the WAIT output, HOLDA is required since one is not sure when a HOLD will be granted. WAIT, on the other hand, is simply a frill that says READY was found to be at a LOW level. The timing for READY is such that one MUST ask for it at a special time; if it wasn't asked for at the proper time, you are now too late to do anything about it. HOLDA will go inactive on a Phase 3, also.

5.7 COMMUNICATION REGISTER UNIT DEFINITION

The Communication Register Unit (CRU) on the TMS 9900 is bit oriented; i.e., only single lines or bits may be operated upon at a time. A single line is dedicated for input data, and a single line is dedicated for output data. The TMS 9900 gives no warning to the outside world when it requires CRU data IN; therefore, the designer must decode the CRU address space for this on the basis of the Address Bus alone. MEMEN- could be included in its inactive state, but it is not necessary. There is a strobe indicating that the HIGH true data on CRUOUT is stable. This is the active HIGH signal CRUCLK, and the TI-99/4 provides this in complement form out the I/O Port. All but the latest TI-99/4s also provide this signal out the Command Module Port. CRU in the Command Module Port was never used on TI products. Data on CRUOUT is stable for the full duration of CRUCLK.

As the CRU Port is bit oriented, multiple bit operations must be accomplished in a serial fashion. The operation begins with a stable address on the Least Significant 13 Address lines (the MS three are set to 0). This is all that occurs for a READ operation (TEST and STCR are the only two), but CRUCLK occurs on a WRITE operation. The process takes two clock periods (667 ns @ 3MHz clock frequency) for each bit required by the instruction being executed. Successive cycles are characterized by an address one bigger than the previous one, and the data on CRUOUT may be different for each cycle on WRITE operations. Multiple bit operations are transparent to the programmer. CRU input cycles have an access time of 500 ns for a 3 MHz clock. Address Bus propagation delay out of the TMS9900 and CRUIN data set-up time for the Phase 1 sampling of data leave only 500 ns out of the 667 ns cycle time for access.

5.8 INTERRUPT STRUCTURE

The interrupt structure on the TMS9900 features a fully prioritized 16 level system, 15 of which are maskable. In addition there is another non-maskable interrupt which is separate from the other 16. The vectors associated with the 16 prioritized interrupts are located in the bottom end of memory, and the two for the non-maskable interrupt are in the top of the memory space.

5.8.1 Interrupt Level Control

The sixteen prioritized interrupts require five lines for their normal operation. Still another unique line is used to invoke Interrupt Level 0 apart from, but in addition to, the other five lines. The level of the interrupt is determined by the HIGH true value on IC0, 1, 2, and 3. Level 0 is the highest priority, and Level 15 the lowest. The active LOW signal INTREQ- indicates to the TMS 9900 that an interrupt is being requested. If the level being requested is less than (higher in priority) or equal to the level in the internal TMS 9900 interrupt mask register, the interrupt will be recognized and the mask will be set to one lower than the recognized level. Otherwise, it will not be recognized until the mask is changed by software, or the level on the IC lines lowers to the proper point. The RESET- input line also invokes a Level 0 interrupt when it makes a transition from active to inactive, as it is level sensitive rather than edge triggered. All operations cease when RESET- goes active.

All of the interrupt control lines must be synchronized before being presented to the TMS 9900.

5.8.2 LOAD Interrupt

The LOAD- input causes an identical type of response as did the other 16 levels, but it is non-maskable. The vectors associated with the 16 prioritized interrupts are located in the lower end of memory, and those for the LOAD interrupt are in the high end of memory. See the Chapter on the TMS 9900 instructions for more information on interrupts.

CHAPTER 6

TMS 9900 INSTRUCTION SET

6.1 GENERAL COMMENTS

In general the TMS 9900 instruction set is appealing for assembly language programming due to an enhanced instruction set rich in powerful addressing modes. Both Register Indirect and Indexing are supported along with Direct Addressing. Relative Addressing is supported for Conditional Jumping as well as for bit oriented CRU operations. The TMS 9900 is somewhat lacking when calling subroutines more than one level deep, as well as in I/O of more than one bit. It is hardily recommended that anyone seriously interested in assembly level language programming purchase a copy of MODEL 990 COMPUTER TMS 9900 MICROPROCESSOR ASSEMBLY LANGUAGE PROGRAMMER'S GUIDE from Texas Instruments, Inc. Your nearest TI Field Sales Office should be able to aid you in this. Also of interest is the TI-99/4A Software EDITOR ASSEMBLER package.

6.2 TMS 9900 ARCHITECTURE

The TMS 9900 Microprocessor is a register file machine that features an I/O structure somewhat similar to that on the TI-960 Mini-computer. The Register File, properly called a WORKSPACE, is located in memory (invariably RAM). The TMS 9900 has only three onboard registers, the Status Register, the Program Counter, and the Workspace Pointer. The latter points to the base WORD address of the sixteen word register file. Instructions are provided to manipulate this base address, as well as the contents of the Status Register.

6.3 REGISTER FILE

The 16-word Register File is advertised as a general purpose file, but there are several registers that are related to commonly used operations. Therefore, in "real life", only ten are truly general purpose, and one other does not allow Indexed Addressing. Registers 13, 14, and 15 are used any time the Workspace Pointer is changed other than by a Load Workspace Pointer Immediate instruction. Conditions that allow for changing the Workspace Pointer include: an interrupt, an Extended Operation (XOP) instruction, a Branch and Load Workspace (BLWP) instruction, and a Return with Workspace (RTWP) instruction. If changed, the old Workspace Pointer will be in R13, the old PC in R14, and the old Status will be in R15.

R12 is used as a CRU Base Register for the five types of CRU instructions, and R11 is used for the return address for subroutine calls. R0 may not be used for an Index Register, as this register is used to signify a Direct Addressing mode in the Opcode. As was stated before, all but Register 0 will function as a general purpose register. Experience has shown that programmers who treat them as such spend abnormally long time periods in the program debug stage, and woe be to the person who has to maintain those programs. One is far better off in reserving the five for their special purposes!

6.4 BYTE ORIENTED INSTRUCTIONS

There are several byte oriented instructions, and if an operation is to be performed on data in a register, it must always be in the MSBY location in the register. There is no bit in the instruction word for byte selection within a register; thus, the MSBY is defined as the only possibility for use as a byte operand within a register. This restriction is not required on byte operations specified by any of the memory addressing modes, and since the register file is actually in memory, an avenue exists for LSBY manipulations.

6.5 OPCODE/OPERAND ORDERING

When more than one Direct Address is to follow an Op Code, the SOURCE address will appear first, and the DESTINATION address second.

6.6 INPUT/OUTPUT INSTRUCTIONS

The INPUT/OUTPUT instructions are perhaps the most difficult of the instruction classes to understand. There are five instructions in this class: three for OUTPUT and two for INPUT operations. All five instructions address I/O with respect to R12 which functions as a "BASE" register for the CRU (Communications Register Unit) instructions.

6.6.1 CRU Organization

The CRU is bit oriented; i.e., only a single bit may be operated upon at a time. There are two instructions that support multiple bit I/O transfers. From a Hardware standpoint, the instruction produces a serial output (input) from the machine that is invisible to the programmer. These two instructions are Load Communications Register Unit (LDCR) and Store Communications Unit (STCR) for inputs and outputs, respectively. Both require the programmer to specify the number of bits to be transferred. There are pitfalls in using these two instructions, and we will discuss this aspect further into this Chapter. Instruction execution time increases two clock cycles per bit for every bit past the first one; therefore, execution time of multiple bit transfers may be quite long.

6.6.2 CRU Lines

The CRU is composed of a maximum 4096 unique input lines and 4096 unique Output lines. Any one of the Output lines may be set to either a LOW or a HIGH level with the SBZ or SBO instructions respectively. Any one of the Input lines may be sampled with a TB instruction. The level of the input line is copied into the "EQUAL" status bit in the Status Register, and changes of command may be accomplished with the JNE and JEQ Conditional JUMP instructions. The JNE jumps on a LOW CRU input bit level, and the JEQ jumps on a HIGH one.

6.6.3 CRU Address Space

The 15-bit TMS 9900 Address bus defines the "CRU Address Space." The Most Significant three bits are always set to zero by the CPU to signify that a CRU operation may be taking place. If a "CRUCLK" occurs, then a CRU OUTPUT operation is occurring. There is absolutely no indication that a CRU INPUT operation is either occurring or has taken place. The system hardware must always assume so, and gate data to the CRU single bit input bus.

The next lower twelve bits define a 4K CRU address space. This corresponds to Address Bus bits 3 thru 14. This same bit field is used in R12 (the CRU Base Register) to form a base or pedestal address that the CRU operation is to be performed with respect to. The Least Significant Bit of an Address (bit 15) is used in the TMS 9900 as a BYTE Pointer, and never leaves the inside of the machine. It has no meaning for CRU operations even though it exists in R12.

6.6.4 CRU Relative Addressing Instructions

Three of the CRU instructions allow for an eight bit 2's complement signed displacement from the base, and two instructions build exactly on the base address in R12. SBZ, SBO, and TB instructions comprehend a negative displacement of 128 CRU Address or bit locations, and 127 in the positive direction. These three instructions employ an eight bit Op Code in the MSBY of the instruction, and the signed displacement in the LSBY. The same is true for the relative JUMP instructions.

The following is a table illustrating the displacement feature.

Table 6-1 CRU RELATIVE ADDRESSING

HEX displacement	CRU Line Selected	CRU Addr, R12=>1A20
00	R12(bits 3 thru 14)	>1A20
02	R12(bits 3 thru 14)+2	>1A24
FF	R12(bits 3 thru 14)-1	>1A1E
EF	R12(bits 3 thru 14)-16	>1A00
10	R12(bits 3 thru 14)+16	>1A40

6.6.5 Multi-bit CRU Instructions

The remaining two CRU instructions allow more than one bit to be transferred, and in addition require the programmer to specify the memory address where data is to come from or be sent to by the CRU instruction. A 6-bit Op Code is utilized, and is followed by a 4-bit field to determine the number of bits to be transferred (1 to 16). A zero in the 4-bit field causes a 16-bit transfer. The next two bits are Tag Bits to define the type of Memory Addressing to be used. The last four bits are used to define a Register number. The format is as follows.

```
¦ Op Code (6) ¦ # of Bit X-fer (4) ¦ Tag (2) ¦ Reg (4) ¦
---------------------------------------------------------
```

6.6.6 Tag Bit Pair Definition. The Tag Bit definitions are listed in the following table. "EA" denotes Effective Address.

Table 6-1 DEFINITION OF TAG BITS FOR CRU USE

Tag Bits	Definition
00	Register Specified
01	Register Indirect, EA is in the Register specified
10	Direct Addressing if Reg field is zero
10	Indexed with respect to the Specified Register, but not R0

Table 6-1 DEFINITION OF TAG BITS FOR CRU USE, CONTINUED

11	Register Indirect with Auto-increment. EA is in the Register Specified, and after the address is used, it is incremented by either one or two depending if eight or less bits are involved in the transfer.

6.6.7 CRU Transfer Bit Length

The number of bits to be transferred in a CRU operation will determine the information format of the memory data. If 8-bits or less are to be transferred, the format requires data to be right justified in a byte. If the transfer is from one of the 15 registers normally available, then IT MUST BE FROM THE MSBY OF THAT REGISTER. An Auto-incrementing memory address will be incremented by one for 8-bits or less.

If nine or more bits are to be transferred, then that data must be formatted right justified in a word. The Auto-increment feature will cause an increment by two in this case.

Information received from the CRU will be formatted in this same manner upon the completion of the STCR instruction execution.

6.7 BIT MANIPULATION INSTRUCTIONS

There are four "Machine Programmer's delight" instructions for operations on both words and bytes. When a bit is to be turned ON in a word, it is customary to logically OR that word with another word that has the bit to be turned ON equal to a ONE, and all of the others set to zero. This is a template word. To turn that same bit OFF, a logical AND instruction must be used, and the template will be the compliment of that used in the OR instruction. Thus, to manipulate a single bit, two templates are required. All of the other bits in the word are unchanged when the word is operated on in this manner.

The TMS 9900 has two instructions that use the same template to change the phase of a bit. Instructions are included for both word and byte operations. They are Set Ones Corresponding, and Set Zeros Corresponding. The template will have a "ONE" in bit positions to be manipulated, and "ZEROs" in positions that are to be unchanged, whatever the bit value may be. Both the template and the word to be operated on may be anywhere in memory or in a register. If a byte is to be operated on in a register, it must be in the MSBY position of the register word. If the byte template is in a register, it must be in the MSBY position, also. The format for this instruction type is shown as follows.

```
¦Op Code (4)¦Des Tag (2)¦Des Reg (4)¦Src Tag (2)¦Src Reg (4)¦
-------------------------------------------------------------
```

6.8 MEMORY TAG BITS

The Tag Bit definitions are listed in the following table. "EA" denotes Effective Address.

Table 6-1 MEMORY TAG BITS

Tag Bits	Definition
00	Register Specified
01	Register Indirect, EA is in the Register specified
10	Direct Addressing if Reg field is zero
10	Indexed with respect to the Specified Register, but not R0
11	Register Indirect with Auto-increment. EA is in the Register Specified, and after the address is used, it is incremented by either one or two depending upon a byte or word operation.

6.9 LOGICAL INSTRUCTIONS

Additional Logical instructions are included past those previously discussed, and they operate on word sizes only. If two operands are required, the destination operand (and the result) must be in a Workspace Register. The AND and OR Immediate instructions may be used on Workspace Registers only. The "Unary" operations CLEAR, INVERT, and SET TO ONE all allow the full addressing range. The format for these three instructions is as follows.

```
¦ Op Code (10) ¦ Tag (2) ¦ Reg (4) ¦
------------------------------------
```

The Exclusive OR (XOR) instruction allows the full addressing range for the Source operand, but the Destination operand must be in a register. The XOR instruction is generally used for a complementing operation. Bit positions in the Destination operand that correspond to "1" bits in the Source operand will be complemented. Bits equal to zero in the Source operand cause no change in the corresponding Destination bits. Note that a quantity XORed with itself will be set to zero (cleared). The instruction format is as follows.

```
| Op Code (6) | Reg (4) | S Tag (2) | S Reg (4) |
-------------------------------------------------
```

6.10 RELATIVE JUMP INSTRUCTIONS

There are thirteen Relative Jump instructions in the TMS 9900 instruction set. These jumps are relative to the memory address two bigger than the location of the jump instruction, and the "displacement" in the Op Code is in WORD locations. The instruction is formatted such that the MSBY is the Op Code, and the LSBY is an 8-bit signed 2's Complement displacement. The following table shows the address of the next instruction to be executed for various jump displacements for the Unconditional Jump instruction Op Code. The Instruction format is shown below the examples.

Address	Op Code	Inst	Next Exe Addr	COMMENTS
>400E	>1000	JMP 0	>4010	NOP
>4002	>1001	JMP 1	>4006	
>4000	>10FF	JMP -1	>4000	JUMPS ON SELF
>4004	>10FD	JMP -3	>4000	

```
| Op Code (8) | Displacement (8) |
----------------------------------
```

6.10.1 Conditional Jump Instructions

With the exception of the Unconditional Relative Jump instructions, decisions are made on one or more of the Status Bits in the Status Register. At the risk of repetition, the JEQ and JNE instructions use the "EQUAL" Status Bit for the jump condition. The EQUAL bit is set to a ONE if the last operation involving a change of this bit found the result to be equal to ZERO, or if the last operation was a CRU Test Bit (TB) instruction, and the bit sensed was at a HIGH level. Note that in reality a comparison is a subtraction, and that the difference is sensed only for the Status Register. Refer to the TI literature on the TMS 9900 instruction set for more information on the Relative Jump instructions.

6.11 SHIFT INSTRUCTIONS

There are four shift instructions, and only registers may be shifted. The shift count may be set from 1 through 15 for the general case, but a shift count of 16 is possible. When the Shift Count field of a shift instruction is set to zero, the machine looks to the Least Significant four bits of Workspace Register 0 for the shift count. If those four bits are zeros, then the shift count is 16. If the Shift Count field is set to 0001, then a shift of one place occurs. The instruction execution time is a function of the number of places to be shifted.

Two of the shift instructions are Arithmetic shifts; i.e., the sign bit (bit-0) is held constant for Right Shifts. The OVERFLOW bit in the Status Register will be set for a Left Shift sign change or overflow condition. The Instruction format is shown as follows.

```
| Op Code (8) | Count (4) | Reg (4) |
-------------------------------------
```

6.12 COMPARE INSTRUCTIONS

The Compare instructions function for both Arithmetic and Logical comparisons. The Compare words or bytes instructions allow the full addressing modes for each operand. The result does not alter the Destination operand. The instruction format for this is shown as follows.

```
|Op Cd (4) | Des Tg (2) | Des Rg (4) | Src Tg (2) | Src Rg (4)|
---------------------------------------------------------------
```

If Direct Addressing is used for both operands, the Source Address will follow the Op Code word. The Destination Address will be the last word in the instruction sequence.

The Compare instruction offers a convenient way to add 4 to a register by executing (for R1) C *R1+,*R1+. Since this is a compare word type of instruction, R1 is incremented by 2 twice. This assumes that it will be no harm to read the contents of the address in R1 (which could possibly be any address in the entire memory space).

The two Compare Corresponding instructions are novel in that the Source word is a template of "ONEs" that is compared to a register. If the register has all of the same ones for a Compare Ones Corresponding or zeros for a Compare Zeros Corresponding, the EQUAL Status Bit is set. It makes no difference what level bits match the ZERO bits in the template. It is only the ONE bits in the template that count as far as the comparison goes. The full addressing range is available for the Source operand (template).

6.13 SUBROUTINE CALLING

Subroutine calls are made with a Branch and Link instruction. The Return Address is left in R11, and if a call to the next inner nesting level is required, the programmer MUST save R11, because the contents of R11 will most certainly be overwritten. To most people this is a minor inconvenience, though.

6.14 XOP INSTRUCTION

The TMS 9900 has sixteen Extended Operation (XOP) instructions that, in essence, function as Software Interrupts. They generate a context switch to obtain a new Workspace and Program Counter (if desired). The old Workspace, PC, and Status Register value are stored in Registers 13, 14, and 15 respectively, in the new Workspace. This is the link back to the old world. The new Workspace and PC are derived from a table in memory that is based at >0040. XOP 0 has its Workspace address located at >0040, and its PC value at >0042. XOP 1 has its Workspace at >0044, and its PC at >046, and so on for the other fourteen XOPs. In addition to the context change, a parameter is brought along in Register 11 of the new Workspace Register file. This parameter is obtained by any one of the five addressing modes available for memory addressing.

The Op Code for the XOP instruction is formatted as follows, and has the same format as does the LDCR and STCR instructions.

```
¦ Op Code (6) ¦ XOP # (4) ¦ Tag (2) ¦ Reg (4) ¦
-----------------------------------------------
```

The XOP # ranges from 0 thru >F. The source of the contents of the new R11 is obtained from the addressing defined by the Tag and Reg fields in the instructions.

6.15 CONTEXT SWITCHING

There are three more methods to cause a context change, but only one of these is an instruction. This is the Branch and Load Workspace Pointer (BLWP or "Bull Whip" as it is called in some circles). The Execution of the BLWP instruction causes a context change to obtain a new Workspace and Program Counter. The old Workspace, PC, and Status Register value are stored in Registers 13, 14, and 15 respectively, in the new Workspace. This is the link back to the old world. The new Workspace and PC are derived by any one of the five addressing modes available for memory addressing.

The BLWP has a ten-bit Op Code, and the standard 6-bit addressing field. This is shown below.

```
| Op Code (10) | Tag (2) | Reg (4) |
-------------------------------------
```

6.15.1 External Context Switching

There are three outside functions that can cause a context switch: 1) sixteen prioritized Interrupts, 2) the Reset operation, and 3) the LOAD operation. The Interrupts are labeled 0 thru 15, and 0 has the highest priority. A Level 0 interrupt may not be masked, and it is also the operation invoked when the TMS 9900 RESET- input pin is driven active LOW. The interrupt vectors are all based at >0000 in the memory space. The new address for the Workspace is located at >0000 for a Level 0 interrupt. The new PC value is located at >0002. The two vectors for the Level 1 interrupt are WS @ >0004, and PC @ >0006, and so on for the other interrupts. During the execution of the interrupt, the old Workspace, PC, and Status Register value are stored in Registers 13, 14, and 15 respectively, in the new Workspace. This is the link back to the old world.

6.15.2 LOAD Operation

The LOAD operation is non-maskable, and causes a context switch just as an interrupt does. The new Workspace value comes from >FFFC, and the New PC from >FFFE. A "Load on Reset" may be designed with external hardware to cause a LOAD operation to occur immediately after the Reset operation is completed (and before the first instruction can be fetched). Notice that the implication of this Reset is that three parameters will be stored displaced from the Workspace Address contained in memory location >0000. WOE BE TO THE PROGRAMMER THAT DOES NOT SUPPLY THIS VECTOR BECAUSE GARBAGE WILL BE STORED AT A GARBAGE ADDRESS! The LOAD operation is second in priority only to the Level 0 Interrupt (or the Reset operation).

6.16 LOAD INTERRUPT MASK IMMEDIATE

The Load Interrupt Mask Immediate instruction allows one to set the priority in masking interrupts of a lower priority. Selective masking of interrupts is not supported by the TMS 9900. The LIMI instruction has the classical form of the "Immediate" operation in that the next word following the Op Code is the value to be loaded into the interrupt mask. The Most Significant twelve bits are ignored, and the Least Significant four are placed, bit by bit, in the Least Significant four bits of the Status Register; i.e., the Interrupt Mask Field. The value in this field is the LOWEST priority interrupt that the TMS 9900 will respond to. If this field is 4, then interrupts 0, 1, 2, 3, and 4 will respond, but levels 5 thru 15 will not.

6.17 ARITHMETIC INSTRUCTIONS

The Arithmetic instructions requiring two operands are ADD, ADD BYTES, SUBTRACT, and SUBTRACT BYTES, and feature the full addressing capabilities for both operands. The Destination operand is replaced by the result of the operation. The instruction format is identical to that of the COMPARE instruction.

6.17.1 Unary Arithmetic Instructions

The unary arithmetic operations ABSOLUTE VALUE, DECREMENT BY ONE, DECREMENT BY TWO, INCREMENT BY ONE, NEGATE, and INCREMENT BY TWO all feature the full addressing capabilities, and the instruction format is identical to that of the unary logical instructions.

The ADD IMMEDIATE instruction allows register only operations with the immediate operand following the Op Code word. There is no Subtract Immediate instruction, but the same operation may be obtained by adding a negative number to a register. The Assembler will take care of the negation process for you. If no Assembler is available, try a Texas Instruments "Programmer" calculator. If neither of these is available, then: 1) take the binary positive number, 2) complement it, and finally 3) increment (add 1) the complemented number. This gives the desired negative number.

The Op Code is twelve bits long, and the Register field requires 4 bits. This is shown as follows.

```
| Op Code (12) | Reg (4) |
--------------------------
```

6.18 MULTIPLY AND DIVIDE INSTRUCTIONS

The Multiply and Divide instructions operate on unsigned 16-bit quantities. The "Multiplier" and the "Divisor" are both Source operands, and the full addressing capability is provided for this operand. The "Multiplicand" and the "Dividend" are destination operands, and must be in a register. In addition, the next bigger numbered register must be available for half of the result of the operation BECAUSE IT IS GOING THERE WHETHER YOU LIKE IT OR NOT! This restriction should not present a problem, though. It follows that one should not use the next larger register number than the Destination register for the source operand. The dividend will appear in the Destination register, and the remainder will appear in the next bigger register number for division. For multiplication the Most Significant word of the two word product appears in the Destination register. The format for these two instructions follows.

```
¦ Op Code (10) ¦ Tag (2) ¦ Reg (4)¦
------------------------------------
```

6.19 WORKSPACE REGISTER MANIPULATIONS

Provisions have been made to save the Workspace Pointer, but only in one of the current Workspace registers. There is no provision for loading either register from a Workspace register, though. The Op Codes are 12-bits wide, and the Register field is composed of 4-bits. The format is identical to that of the Add Immediate instruction.

The Workspace Register as well as any of the sixteen Workspace Registers may be loaded with an immediate instruction. They also are formatted identical to that for the Add Immediate instruction. There is no other way to load these registers, but a MOVE instruction may be used to load a Workspace Register with a little effort on the programmer's part. Storing the Workspace Pointer in one of the Workspace Registers gives one the address of R0. The address of R1 is two (byte locations) bigger than that for R0, that for R2 is four bigger than that for R0, and so on.

6.20 MOVE INSTRUCTIONS

There are two MOVE instructions, one for words, and one for bytes. The hardware design of the TMS 9900 causes a read operation on the destination address before the data is moved to it. This may present a hardware problem, but this effect is not obvious to the programmer. The Systems Engineer will be more interested in this characteristic, though. Both instructions allow the full addressing capabilities for both the Source and Destination addresses, and the instruction format is identical to that of the Compare instruction.

6.21 SWAP BYTES INSTRUCTION

An instruction with full addressing capabilities is provided to swap bytes in a word. This should be helpful for manipulating bytes in a register, as byte operations may be performed only on the MSBY in a Register. The instruction format is identical to that of both the Arithmetic and Logical Unary instructions.

6.22 EXECUTE INSTRUCTION

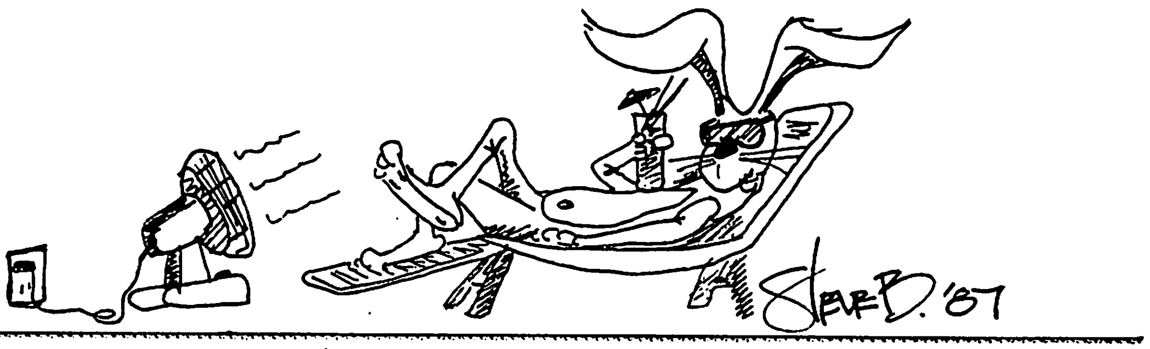
The last of the TMS 9900 instructions to be discussed is the EXECUTE instruction. One may generate an instruction to be executed, and then with the full addressing capability, specify the location of the word to be executed as a normal TMS 9900 Op Code. The format of this instruction is identical to that of both the Arithmetic and Logical Unary instructions.

6.23 UNSUPPORTED INSTRUCTIONS

There are several TMS 9900 instructions that the TI99-4 Console does not support. All of these cause the hardware signal "CRUCLK" on the TMS 9900 to occur. These instructions are: 1) IDLE, 2) RSET, 3) CKON, 4) CKOF, and 5) LREX. In assembly language programming the TI-99/4, only the IDLE instruction may be tempting to use. DO NOT USE THE IDLE INSTRUCTION. The possibility exists that either the TMS 9901 interface chip in the Console or the CRU Output register in one of the peripherals will be inadvertently changed since the system thinks that the CRUCLK generated is for a CRU operation when indeed it was only for the IDLE instruction.

6.24 SUMMARY

In summary, one should see that the TMS 9900 is indeed a powerful machine, and will be a pleasure to program.



CHAPTER 7

THE PERIPHERAL EXPANSION BOX

7.1 PEB PURPOSE

The Peripheral Expansion Box (PEB) provides a means of adding seven more functions to the TI-99/4 Home Computer. There are a total of eight card sockets on the backplane, but one of these is taken up in interfacing to the TI-99/4 Console.

A detailed description of each of the four PCBs marketed by Texas Instruments that plug into the PEB are included in this manual. Schematics for these boards are not.

7.2 PEB POWER SUPPLY

The power supply in the Peripheral Expansion Box provides a full wave, L/C filtered, raw DC power source for the cards that plug in the provided slots. It is up to the designer to include some type of regulator on the plug-in PCB when TTL logic is to be used. The PCBs designed by TI had Three Terminal regulators for +12 and +5 clamped in place by the "Clam Shells" that enclosed the PCB. These metallic Clam Shells provide RFI shielding as well as heat sinking for the 3-Terminal Regulators. Care must be taken in obtaining -5V from the -12V bus in that the power dissipation of the 3-Terminal Regulator may be more of a governing factor than the current required. The so called "Head Voltage" of the regulator will be about 12V for this situation.

One must also be careful when using 3-Terminal Regulators to include the required by-pass capacitors very close to the regulator itself. Consult the particular manufacture's data sheets for the correct capacitor size and other pertinent details.

Considering the rectifier sizes along with the power required for the internal disk drive, it seems reasonable to assume that a budget of .5 Amps per card slot is available from the +5V supply. It also seems that .25 Amps for +12V, and 50 ma for -12V is reasonable.

7.3 SOCKET PIN BUSSING

The back plane provides pin-to-pin connections for all eight sockets; i.e., there are no pins of unlike numbers connected to each other except for power and Ground bussing. Power pins are bussed together for like voltages. The connecting etch is on the bottom side of the PCB, and the top side is a solid Ground plane. There are neither active nor passive components on this PCB.

7.4 Address Bus

The PEB is configured with 19 address lines, but only the least significant sixteen are active. The Most Significant three (AMC, AMB, and AMA) are hardwired to a HIGH level on the Interface Board in the PEB. This fact leads one to believe that some other features were to be added at a later date. It also appears that these features were never added.

7.5 PEB CARDS

There were only four PCBs developed and marketed for the PEB, but several others were apparently used inside of Texas Instruments and several other places. Some of these others were: an EPROM Programmer, a 128K byte RAM, an IEEE 488 BUS Interface, and an Interval Timer. A third party (MYARC) developed a hard disk controller for the PEB. We will discuss only those marketed by Texas Instruments. They are: RS232, 32K byte Memory Expansion, P-Code, and the Single Density, Double Sided Disk Controller.

7.6 CONSOLE INTERFACE

The next function to be discussed is the interface back to the Console itself. Two PCBs that are hard wired to the 44-conductor shielded cable are included in this interface. One is located at the Console, and one plugs into the PEB (usually on the far left hand side). It should make no difference where it is plugged in, though. A Grounding lug is provided to insure good bonding of the cable ground and the PEB metal case.

7.6.1 Console End Interface

The interface for the Console end of the cable has a SN74LS245N Bidirectional Bus Driver, a small three terminal +5V regulator, a number of terminating resistors, and several capacitors. The '245 bus driver is used to isolate the Console 8-bit I/O Data Bus from the cable; thus, the devices that are connected to this bus internal to the Console do not have to furnish drive to the shielded cable load.

7.6.2 PEB End Interface

The PCB on the PEB end is responsible for providing buffering of the Address Bus, the Control Bus, and the Data Bus. Status lines such as READY and CRUIN are not buffered, but may have pull-up resistors included. Appendix C is a description of the backplane signals and their pin numbers.

7.7 PERIPHERAL MEMORY SPACE ORGANIZATION

The Device Service Routine (DSR) space allocated to peripherals is 8K bytes. This space is not nearly enough for all of the peripherals that may be connected to the Console; therefore, some mechanism of expanding this space is required. The CRU Output Bit at the base address of a peripheral is used as a page bit to enable memory oriented functions on each peripheral PCB. These CRU bases start at >1000, and increase in increments of >0100.

7.7.1 Peripheral CRU Space Definition

The Peripheral CRU space is noted in the following table.

Table 7-1 PERIPHERAL CRU SPACE

>1000 not used	>1100 Disk Controller
>1200 Modem	>1300 Primary RS232
>1400 not used	>1500 Secondary RS232
>1600 not used	>1700 HEX-BUS Controller
>1800 Thermal Printer	>1900 not used
>1A00 not used	>1B00 not used

Table 7-1 PERIPHERAL CRU SPACE, CONTINUED

>1C00 not used	>1D00 IEEE 488 Controller
>1E00 not used	>1F00 P-Code

CHAPTER 8

32K BYTE MEMORY EXPANSION

8.1 MEMORY EXPANSION ORGANIZATION

The 32K Byte Memory Expansion (MEX) PCB uses sixteen TMS4116 250 ns Dynamic Ram (DRAM) chips to provide the required Primary Memory storage; i.e., code may be executed directly out of the MEX. All of the Peripheral Expansion Box bus lines are buffered before use in the MEX with the exception of the signal READY, which drives only one unit load.

8.2 ADDRESS REQUIREMENTS

The MEX is designed to respond in two disjoint and unequal areas of the Console memory space. The first is an 8K byte block starting at >2000, and ending with >3FFF. The second is a 24K byte block that starts at >A000, and ends with >FFFF. As may be suspected, the address range is not in a pure binary order; therefore, some adjustment must be made on one or more of the address bits to obtain the required DRAM sequential binary addressing. This is illustrated in the Table below showing a segmentation of 8K blocks. Keep in mind that 15 address bits are required to address a 32K byte space.

Table 8-1 ADDRESS REQUIREMENTS

MS 3 Address Bits	HEX Address Range
001	>2000 thru >3FFF
101	>A000 thru >BFFF
110	>C000 thru >DFFF
111	>E000 thru >FFFF

Notice that there are not the same number of "ones" and "zeros" for bits 0 and 2 of the Most Significant three Address bits; thus, they cannot not form a binary sequence including 00, 01, 10, and 11 in any order. The order would have to be ascending if the addressing was for a ROM, unless code was adjusted accordingly. The order makes no difference for RAM, though. Address bits 1 and 2 provide the "01" combination twice, and "00" does not show up. It is necessary to generate a new bit to replace either Address bit 0 or 2. It makes no difference which one is replaced, and bit 2 was selected. One way to obtain this new bit is to use the NAND of Address bits 0 and 2. This was implemented in the PAL used to decode the response addresses, etc. This produces the ordering in the following table.

Table 8-1 ADDRESS MODIFICATIONS

Console Address	Generated 8K Block DRAM Address
001	01
101	00
110	11
111	10

From the preceding table and the PAL equations, one may see that Console 8K block base addresses >2000 and >A000 lie in one 16K bank, and >C000 and >E000 in the other.

8.3 DRAM FUNDAMENTALS

In order to better describe the operation of the MEX, a short discussion of DRAM fundamentals is in order. The following is meant to be a cursory, but straight forward description of the DRAM chips used in the MEX. These chips are of the address multiplexed variety of DRAMs, and have two chip select pins to accommodate each half of the 16K location address. The first is the ROW ADDRESS STROBE (RAS-), and the second is the COLUMN ADDRESS STROBE (CAS-). Both are active LOW, and a sequence of events is started within the chip on the falling edges of both. Large, but short term, transient currents appear on both the +12V and -5V pins following these negative going transitions. An adequate number of by-pass capacitors must be present to supply this short term current if reliable operation is to be obtained. The magnitude of these currents is 80 ma per chip, typical; therefore, a relatively low impedance supply for these voltages must exist. Catalogs from both Mostek and Intel discuss this subject in more detail.

8.3.1 Modes of Operation

There are several different modes of operation of these chips, especially from vendor to vendor, but the modes used in the MEX are common to all of the vendors. RAS- always occurs before CAS- in the MEX, and EARLY WRITE is the mode used for a WRITE operation (the DRAM "W" input is stable before CAS- occurs). RAS- only REFRESH is used to refresh the DRAMS.

8.3.2 Address Sequencing

In a DRAM access the first seven address bus bits are presented to the DRAM address bus (with both chip selects in the inactive state), and then, after sufficient set-up time, RAS- is driven active LOW. The row address must be held stable for a minimum hold time, and then the column address is multiplexed to the DRAM address pins. Most DRAMS have a small negative set-up time for this before CAS- is driven active LOW, but the MEX observes a positive set-up for additional operating margin. Once CAS- goes active, the address must be held stable for a minimum hold time. If Early Write is used, the "W" input to the DRAM will be in its stable state Before CAS- goes active. The MEX maintains the W- input stable for the entire access duration for both READs and WRITEs.

8.3.3 Internal DRAM Bit Storage and Refresh

The DRAM uses a capacitor to to hold the logic level of the bits stored; therefore, some mechanism must be provided for keeping the "imperfect" capacitors charged. Either a READ or a WRITE operation will do this, but a WRITE would require the proper data to be present at the Data IN pin of the DRAM. Practice has shown that this is a very inconvenient solution; so, the READ operation is invariably used for refresh. This is the case for the MEX. The READ operation in the DRAM reads information for many bits (normally 128, but not necessarily so) in parallel, and during the internal operation of the chip, all of these bits are automatically refreshed. With this in mind, it is only necessary to READ from each of the 128 (a 7-bit range) row addresses to completely refresh the entire DRAM chip. Additionally, most manufactures require that the entire 128 rows be refreshed in only 2 ms. In reality, data may not need refreshing at room temperature or below for several minutes in some chips. It is considerably shorter in others, though.

8.4 MEMORY EXPANSION UNIT PLA

A PAL12L6 Programmable Logic Array is used to obtain the address decoding for the two bank RAS- strobes, the generated address bit, the MEX access requests, and the inversion of the memory control signal "MEMEN-". It has inputs consisting of six PEB Address Bus bits, MEMEN-, the MS Refresh Address bit, Refresh Request, PCBEN, synchronized PEB Address bit 1, and Q2 of the state counter. The equations for this PAL are listed in the following table.

Table 8-1 MEX PAL EQUATIONS

PAL OUTPUT PIN	EQUATION
13	Q2*/A1S + REFRAS*/REFA7
14	PCBEN*MEMEN*AMC*AMB*AMA*/A1*A2 + PCBEN*MEMEN*AMC*AMB*AMA*A0*A1
15	Q2*A1S + REFRAS*REFA7
16	A0*A2
17	NOT USED
18	/MEMEN

The pin definitions of the PAL are listed in the following table. Monolithic Memories Inc. terminology is used.

Table 8-1 PAL PIN DEFINITIONS

1	PCBEN	11	REFRAS
2	AMC	12	REFA7
3	AMB	13	/RAS0
4	AMA	14	/MESEL
5	A0	15	/RAS1
6	A1	16	AP02
7	A2	17	not used
8	/MEMEN	18	MEMEN.B
9	Q2	19	A1S
10	Power, GND	20	Power, VCC

PCBEN is a signal obtained from the backplane that disables access requests when it is in the LOW state. The access request decoded from the PAL is used to enable the remote data bus driver at the Console end of the cable through a SN74LS125 on the MEX. This '125 is connected to affect an open collector gate. The access request decode is also synchronized with the Trailing Edge of the Phase 3 clock, and then presented to the two-bit state counter described in the next paragraph.

8.5 MEX TIMING AND CONTROL

A two-bit state counter is used to generate the timing and control for accessing the DRAMS. Separate logic is used for the DRAM Refresh requirements. Since Phase 3 of the TMS 9900 clock is the only phase furnished to the I/O Port, all timing is referenced to that clock. The TMS9900 generates timing off of the Phase 2 clock, and takes data (for a READ operation) during Phase 1. This sequence, coupled with the fact that the MEX is timed off of Phase 3, causes the DRAMs to be active for one clock phase past that required by the TMS9900. While this causes no terrible problems, one must latch both the Address bit that selects the required eight chip DRAM bank as well as DBIN; otherwise, data can be written at two addresses during one WRITE cycle. If DBIN were to change from a READ to a WRITE state during an access, it would be possible to WRITE to a READ address during this short overlap of clock phases. The address bit change condition happens only under certain (but definitely real) circumstances.

The two-bit state counter is responsible for generating the primary RAS for each access excluding refresh. The IDLE state is 00, and it counts from 00 to 01 to 11 to 00 to 01 to 11 and to 00 for a normal single word Console access. Remember that the Console requests an integral number of WORD accesses, and that a single word access translates to a pair of MEX byte accesses. The bit ordering is (Q1, Q2), and Q2 is the primary RAS. It is further encoded in the PAL to generate the proper bank RAS-.

Once the Primary RAS (Q2) becomes active, the FF controlling the DRAM address Multiplexer will be clocked ON by the next edge of the Phase 3 Clock. This will be the Trailing edge of Phase 3, and that clock phase has been delayed by two SN74LS244 gate delays past that buffered off of the PEB backplane. This is the manner in which an additional margin of RAS address hold delay is obtained.

CAS- is generated asynchronously off of the DRAM address multiplexer in such a manner that DRAM setup parameters cannot be violated. This is accomplished by using the delay of the tri-state DRAM multiplexer to clock a flip-flop (FF) ON through another delay (through a SN74LS125 non inverting gate). A pull-up resistor on the output of the multiplexer keeps that output at a HIGH level when the MUX is in the tri-state mode. CAS- is further delayed through another SN74LS125 and a 27 Ohm terminating resistor.

Refresh is controlled by a SN74LS161 HEX Counter that is clocked by the Trailing Edge of the Phase 3 Clock. The Ripple Carry output of this counter is run to the "D" input of a FF which is clocked by the Leading Edge of the Phase 3 Clock. This FF generates a 1/3 us Refresh cycle that is encoded in the PAL to provide a RAS- to the proper DRAM bank for refresh. The Trailing Edge of this FF is responsible for incrementing the seven-bit Refresh Address Counter. If the Console is not accessing any memory, or if it is, AND that memory element is in the NOT READY state, a refresh operation will occur every (16/3) us. Any console memory access with READY=TRUE causes the '161 counter to be loaded to >E or "1110". When the current Console memory cycle(s) completes, the counter will return to the count mode, and will count to >F or "1111". The Ripple Carry output will go active HIGH, and a refresh cycle will occur on the next edge of the clock.

The LED indicator is triggered with the rising edge of Q2 since this signal is not active during REFRESH cycles. A Retriggerable one-shot is used for this purpose.

There appears to be only one area of concern in running TMS 9900 machine language code out of the MEX, and it is related to READing the Speech Synthesizer. This device is responsible for disabling its own data bus drivers, and it does not get in a big hurry to do so. It takes this device about 20 us to disable its drivers once its chip select goes inactive. The solution to this problem is to save a segment of Console SRAM in the MEX, download code into the SRAM, execute the code in the SRAM with a 20 us delay included, and then return the SRAM image from the MEX to the Console. None of the system software is available for use until the contents of the SRAM have been restored.

8.6 MEMORY EXPANSION PRODUCT DIFFERENCES

There are some differences between the "Side Car" Memory Expansion Module and the Memory Expansion Card, but none are visible to the user. PAL logic is used in the MEX, and SSI/MSI is used in the Module. The timing is different in the module in that it generates the memory controls earlier than does the MEX, and uses a SN74LS373 Octal Latch to save the READ data until the proper time for the Console. The logic used to implement the Timing and Control in the Memory Expansion Module is considerably different than that used in the MEX. Additionally, the Module traps out all but the Peripheral DSR memory space from >4000 thru >5FFF, which it passes on through it. It buffers the Console Data Bus which isolates the GROM chips in the Console from modules past the Memory Expansion Module. This Data Bus buffering requires that all peripherals (with the exception of the Speech Synthesizer), be plugged into the Memory Expansion Module.

The Memory Expansion Module has its own internal Power Supply, and the MEX obtains its power from the PEB.

8.7 DIAGNOSTIC TEST PATTERNS

In writing assembly level diagnostics for the MEX, one should

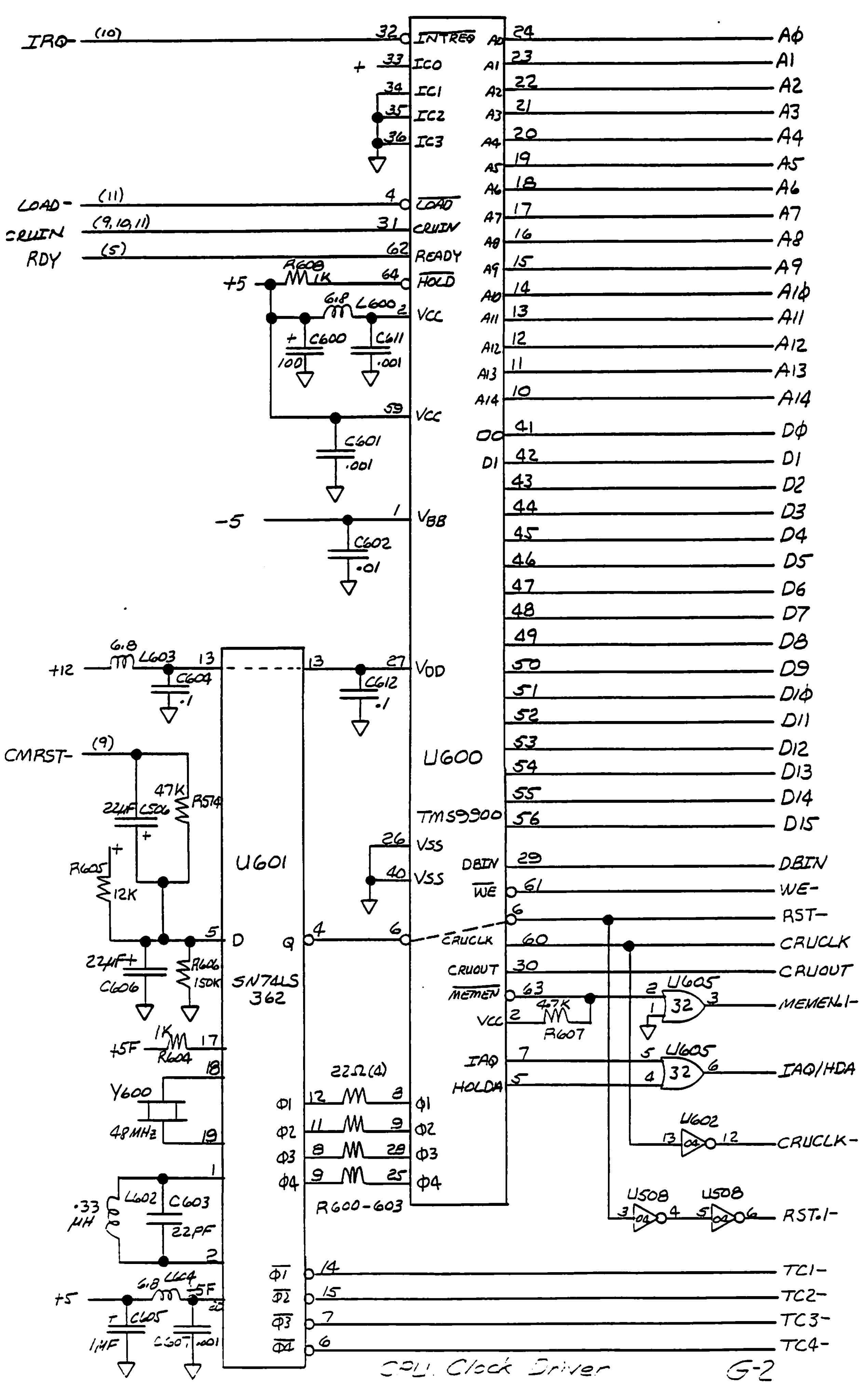
APPENDIX G

TI-99/4A SCHEMATICS

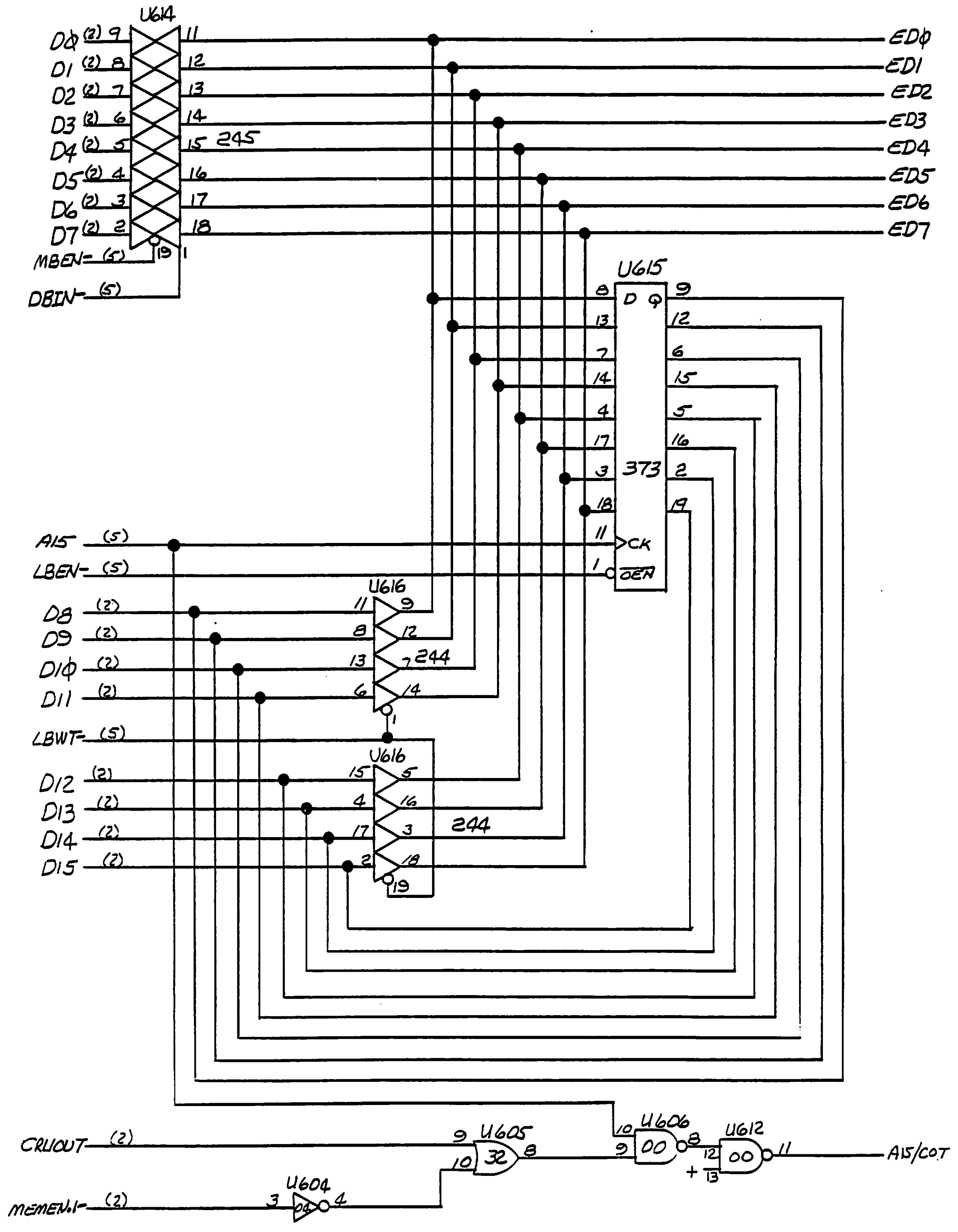
G.1 COMMENTS

The following set of schematics are functionally organized, and drawn on a size of paper that, when reduced, is still quite readable. In the interest of simplicity, most of the by-passing networks have been left off the drawings. These networks are generally "L" type LC networks, and were probably used for RFI suppression.

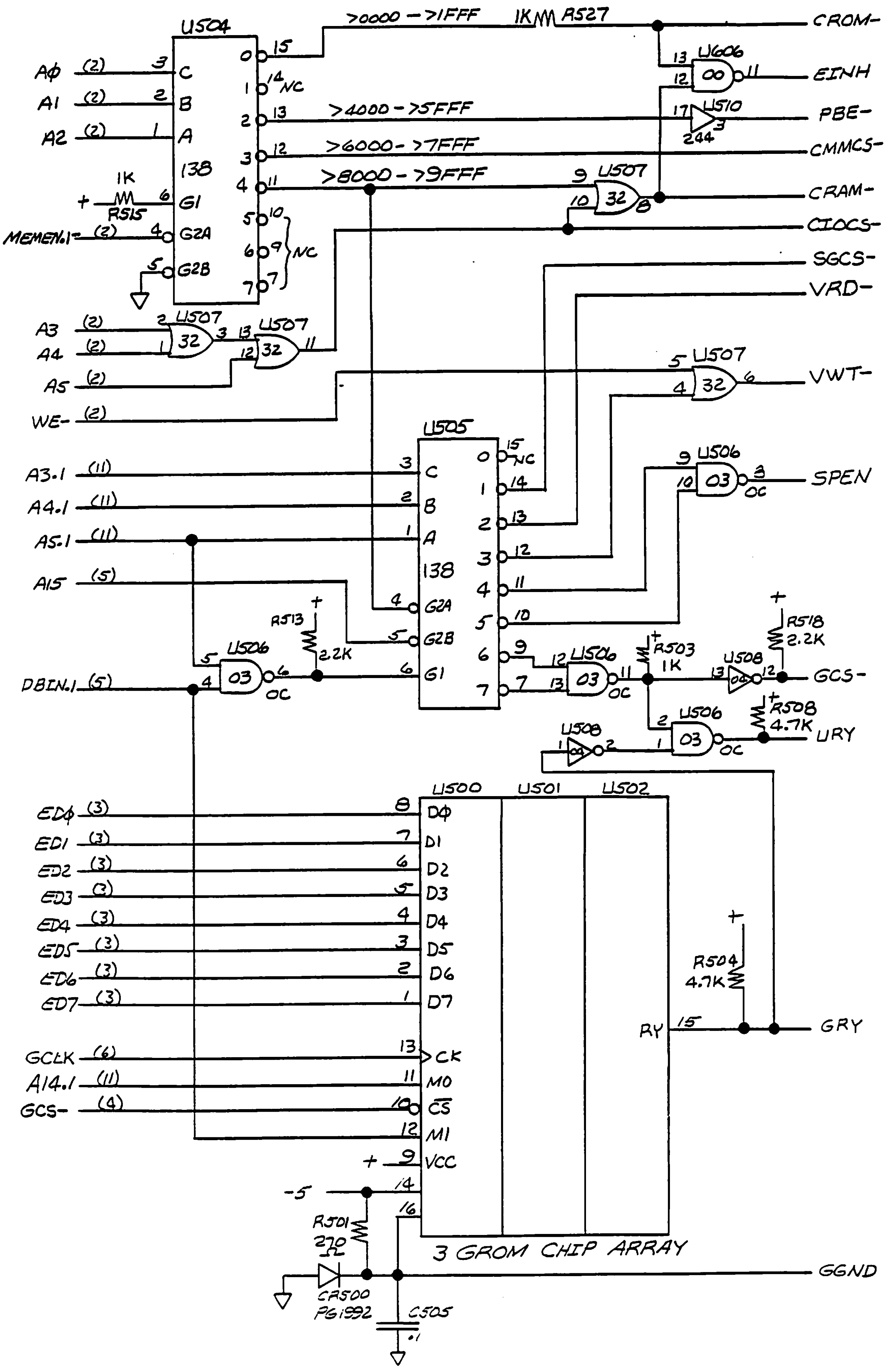
All inputs are drawn on the left side of the page, and outputs are on the right side. Page numbers are associated with the input signals to aid in tracing a signal source.



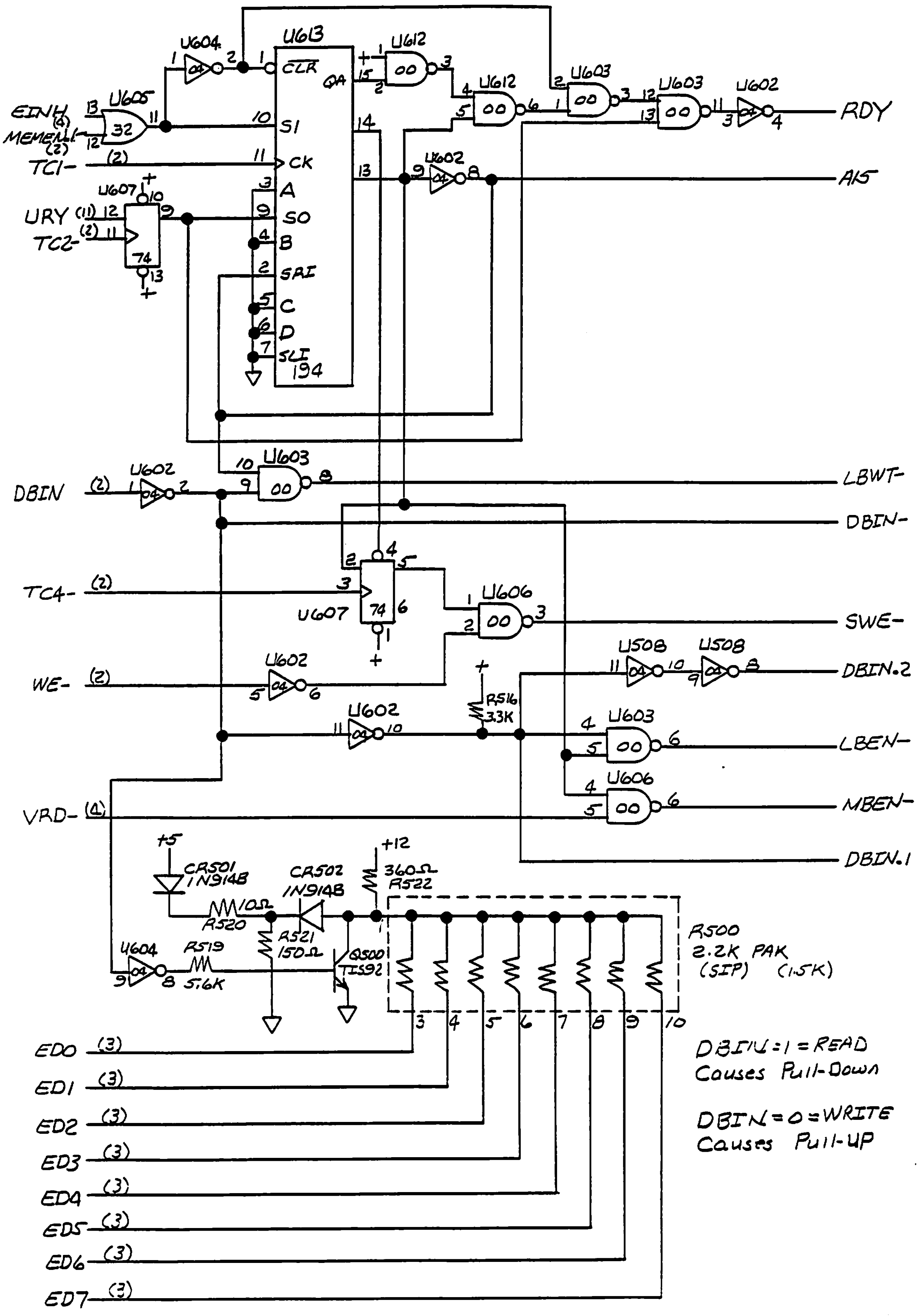
CPU Clock Driver G-2



BUS CONVERSION DATA PATH, A15/CRUOUT MUX



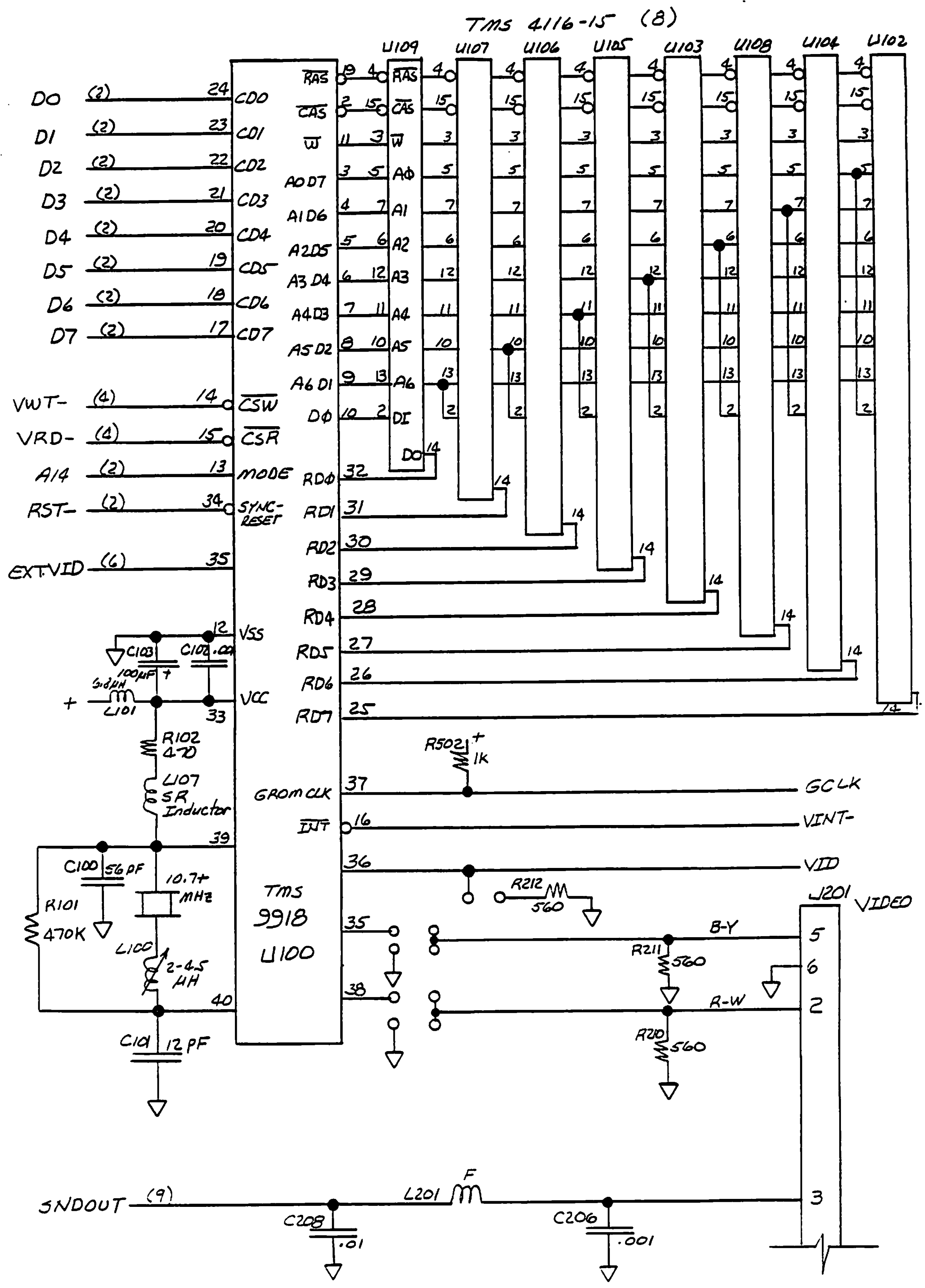
MEMORY/CRU SPACE DECODES, GROM ARRAY



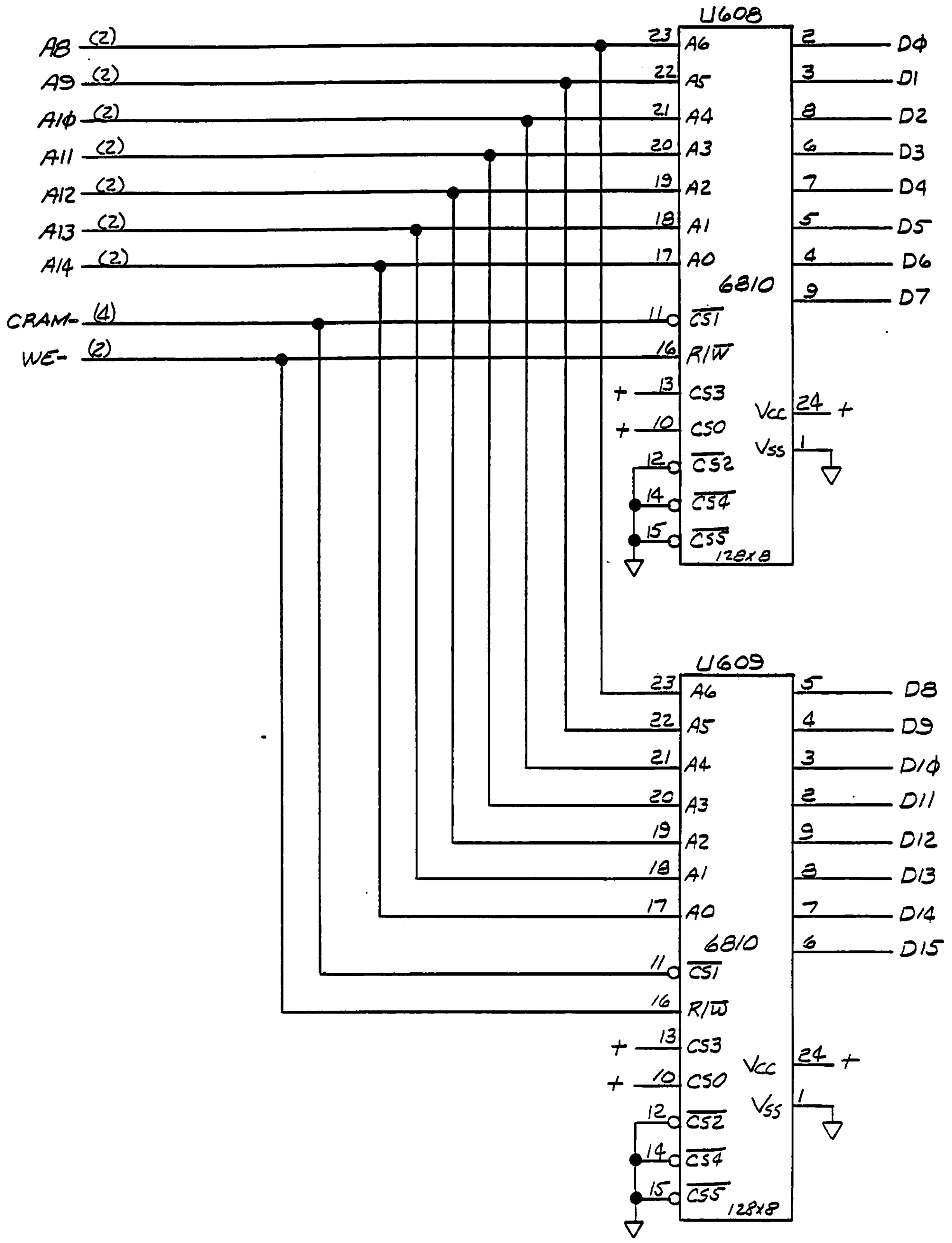
DBIN=1=READ
Causes Pull-Down

DBIN=0=WRITE
Causes Pull-UP

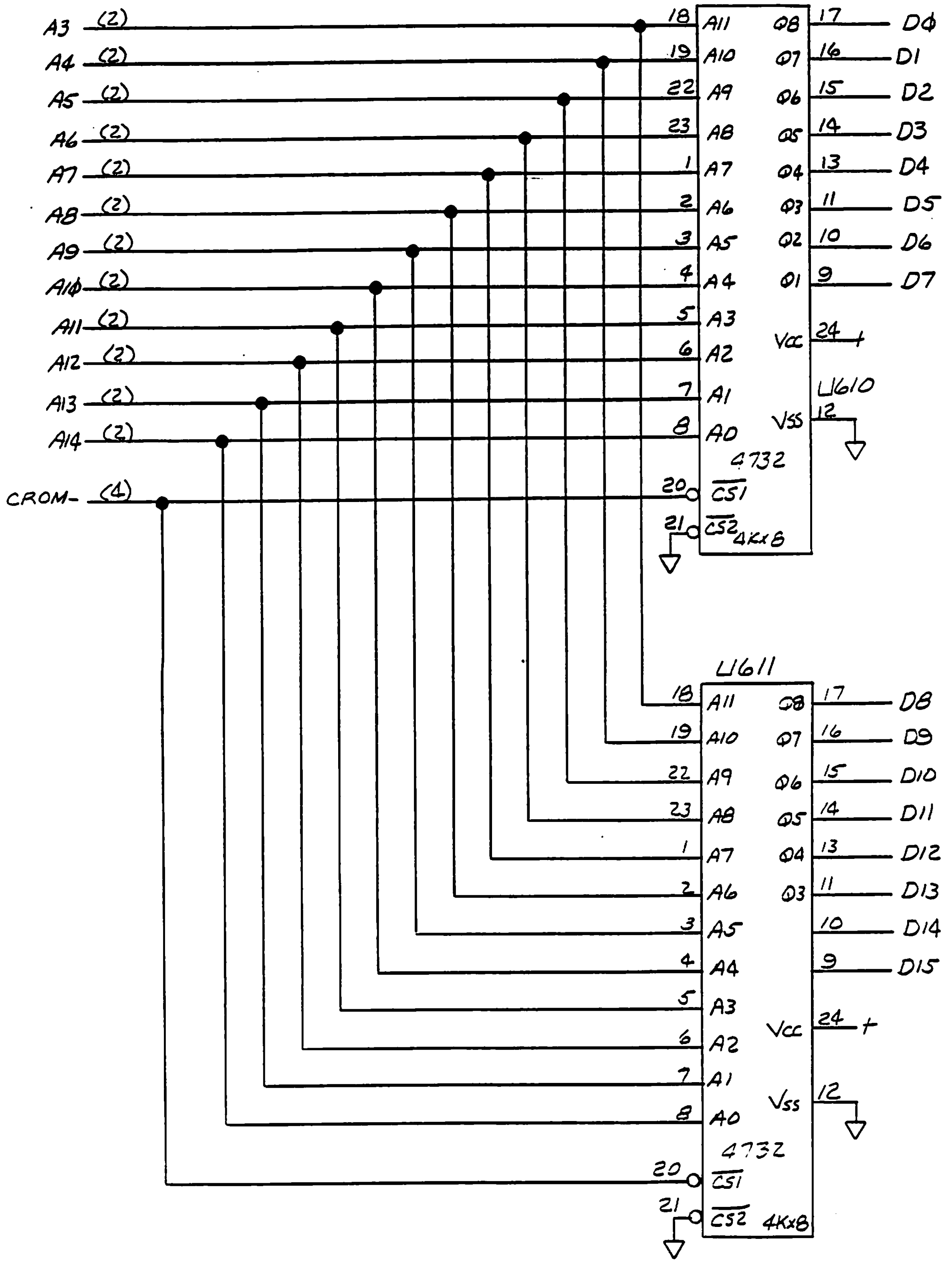
T&C - BUS CONVERTER, P/U - P/D Resistors G-5



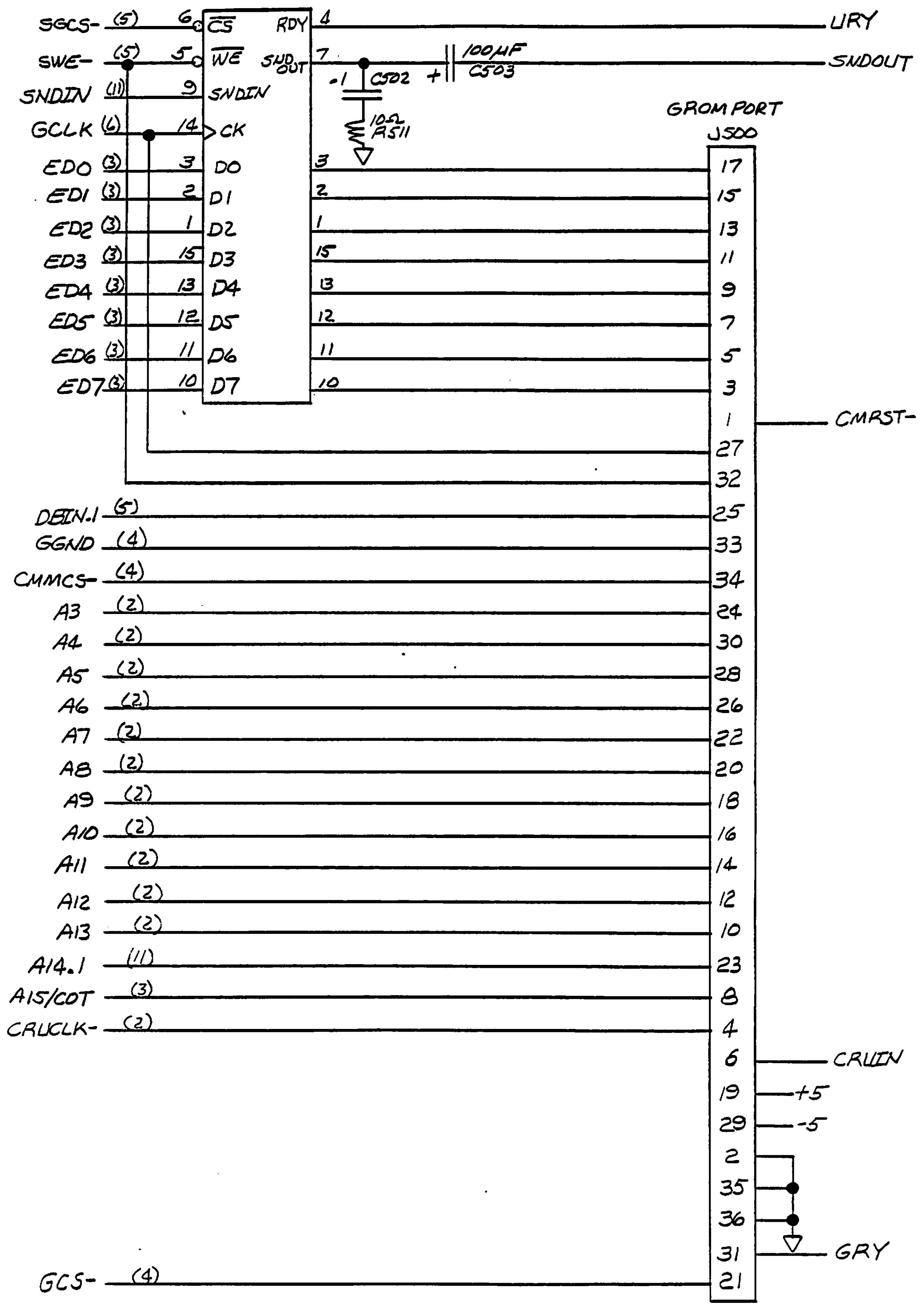
VDP, VIDEO RAM



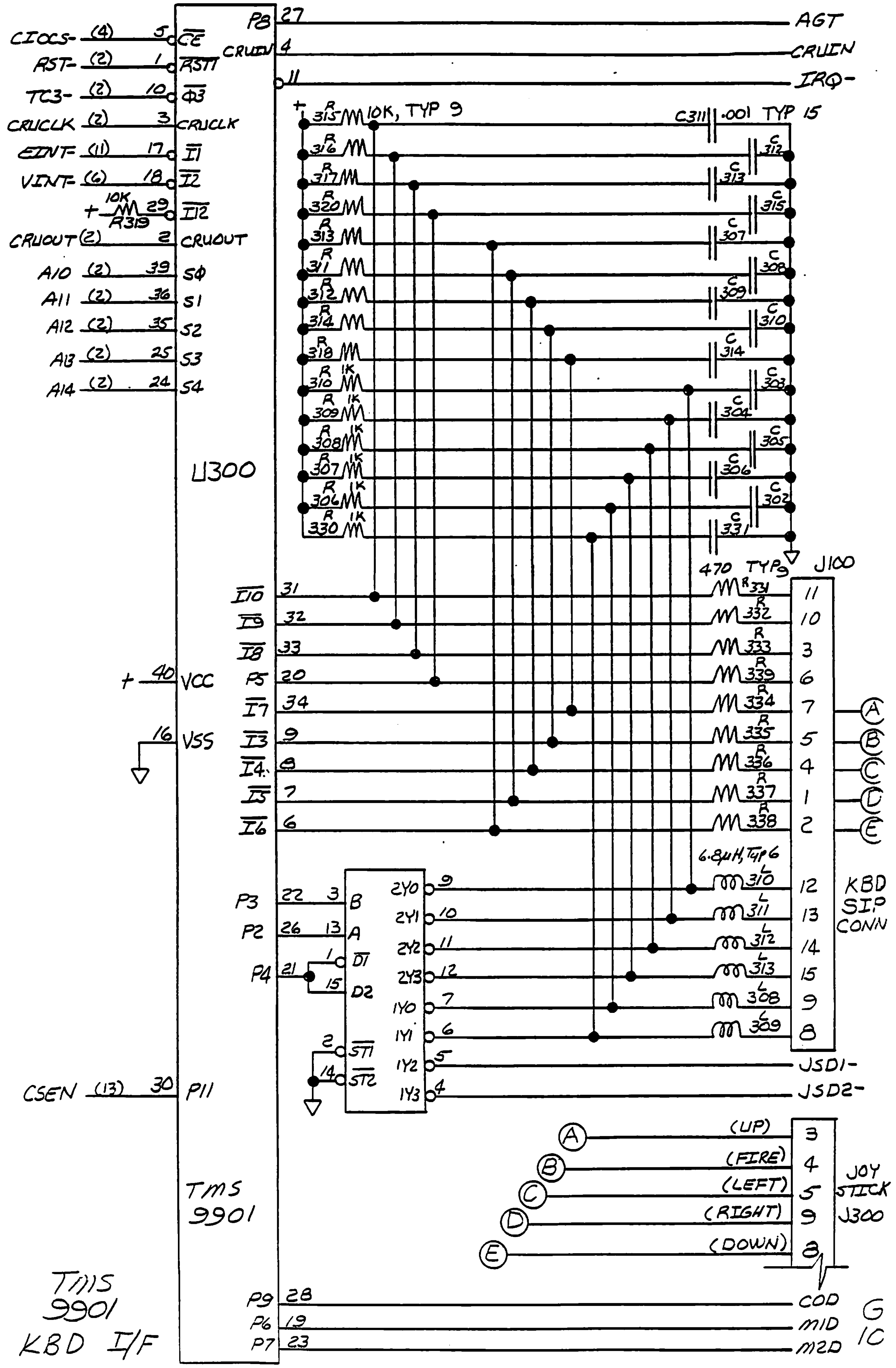
CONSOLE RAM (128X16)

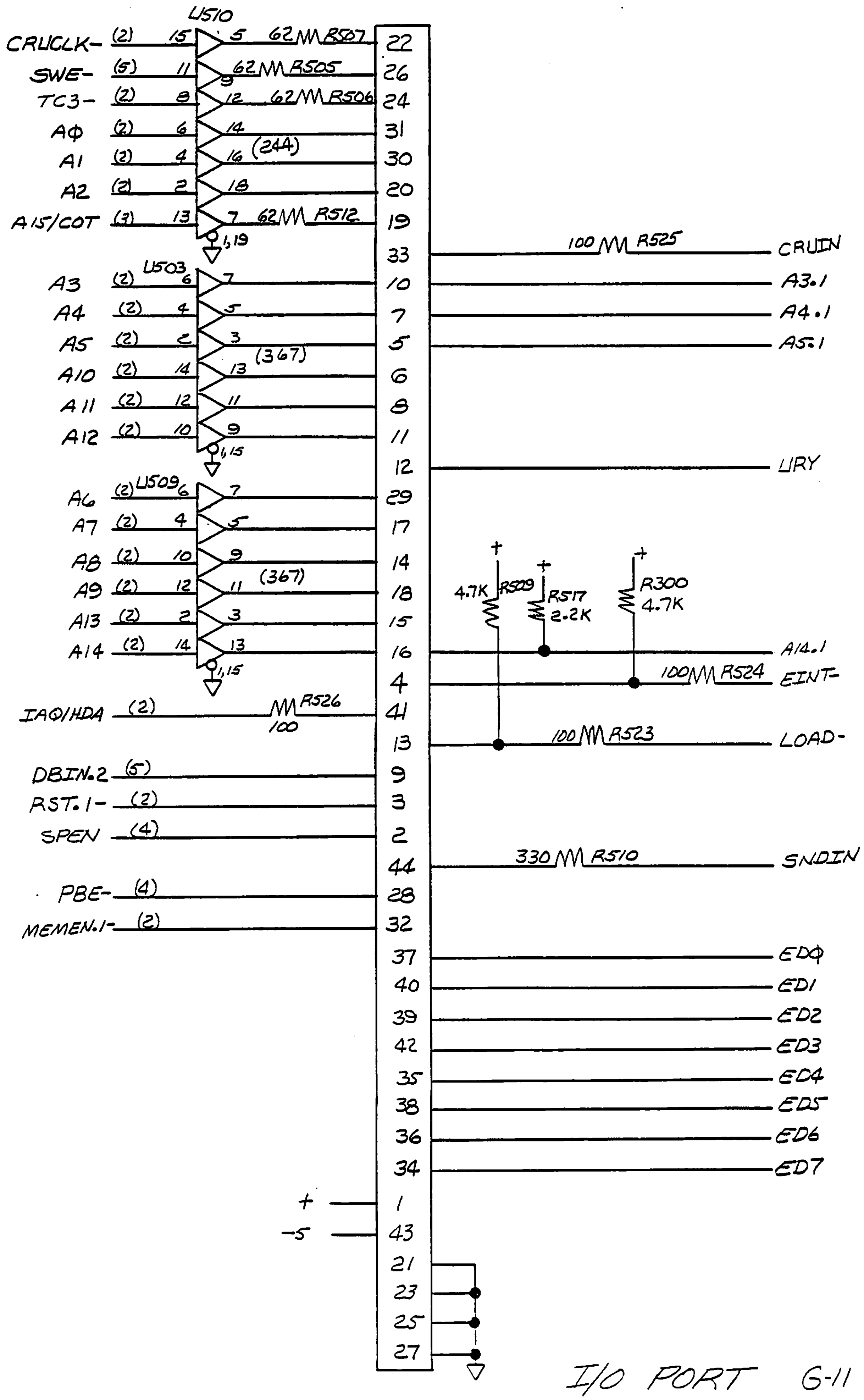


CONSOLE ROMS (4K x16)



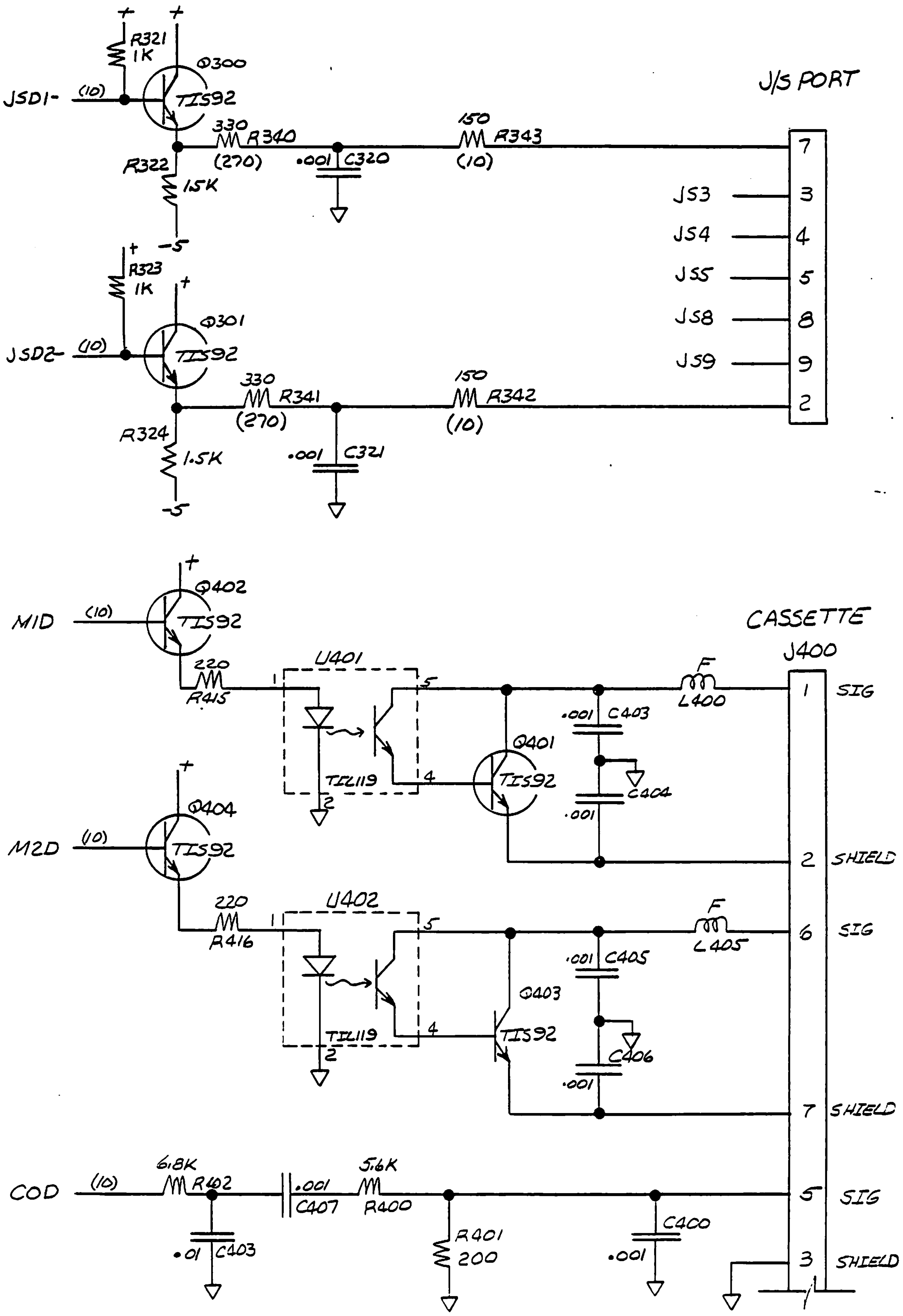
SOUND GEN, C/M PORT


CIOCS- (4) 5 CE
RST- (2) 1 RSTI
TC3- (2) 10 Φ3
CRUCLK (2) 3 CRUCLK
EINT- (11) 17 I1
VINT- (6) 18 I2
+ 10K R319 29 I12
CRUOUT (2) 2 CRUOUT
A10 (2) 39 S0
A11 (2) 36 S1
A12 (2) 35 S2
A13 (2) 25 S3
A14 (2) 24 S4
P8 27 AGT
CRUIN 4 CRUIN
11 IRQ-
R315 10K, TYP 9
C311 .001 TYP 15
U300
I10 31
I9 32
I8 33
P5 20
I7 34
I3 9
I4 8
I5 7
I6 6
+ 40 VCC
16 VSS
470 TYP 9
J100
KBD SIP CONN
6.8µH, TYP 6
P3 22 3 B
P2 26 13 A
P4 21 1 D1 15 D2
2 ST1 14 ST2
2Y0 9, 2Y1 10, 2Y2 11, 2Y3 12, 1Y0 7, 1Y1 6, 1Y2 5, 1Y3 4
JSD1-
JSD2-
CSEN (13) 30 P11
(UP) 3
(FIRE) 4
(LEFT) 5
(RIGHT) 9
(DOWN) 8
JOY STICK J300
TMS 9901
P9 28 COD
P6 19 MID
P7 23 M2D
G
IC
TMS 9901 KBD I/F

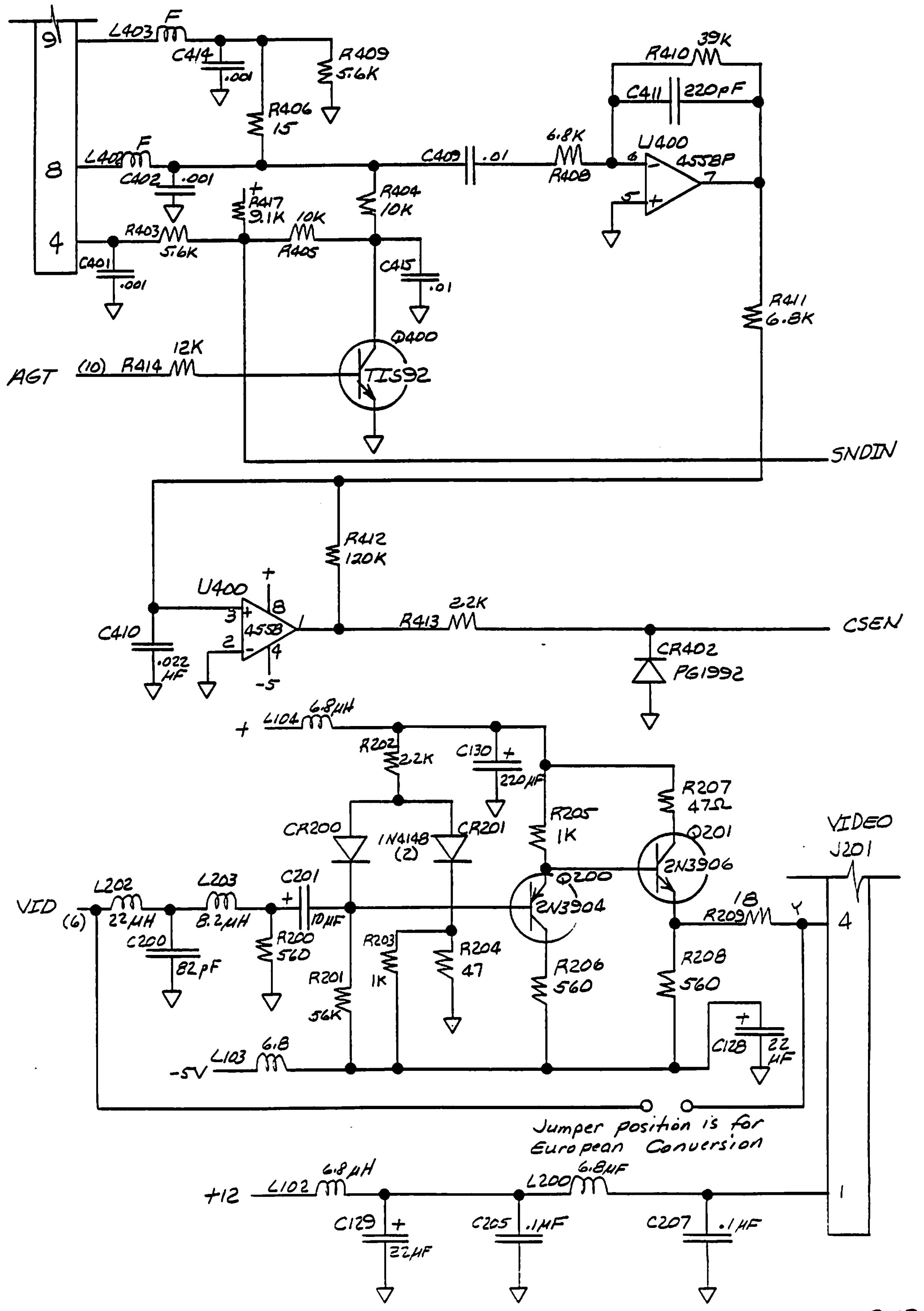


U510
CRUCLK- (2) 15 5 62 R507 22
SWE- (5) 11 9 62 R505 26
TC3- (2) 8 12 62 R506 24
AΦ (2) 6 14 31
A1 (2) 4 16 (244) 30
A2 (2) 2 18 20
A15/COT (3) 13 7 62 R512 19
1,19
33 100 R525 CRUIN
U503
A3 (2) 6 7 10 A3.1
A4 (2) 4 5 7 A4.1
A5 (2) 2 3 (367) 5 A5.1
A10 (2) 14 13 6
A11 (2) 12 11 8
A12 (2) 10 9 11
1,15
12 URY
U509
A6 (2) 6 7 29
A7 (2) 4 5 17
A8 (2) 10 9 14
A9 (2) 12 11 (367) 18
A13 (2) 2 3 15
A14 (2) 14 13 16 A14.1
1,15
4.7K R509
R517 2.2K
R300 4.7K
4 100 R524 EINT-
IAQ/HDA (2) R526 100 41
13 100 R523 LOAD-
DBIN.2 (5) 9
RST.1- (2) 3
SPEN (4) 2
44 330 R510 SNDIN
PBE- (4) 28
MEMEN.1- (2) 32
37 EDΦ
40 ED1
39 ED2
42 ED3
35 ED4
38 ED5
36 ED6
34 ED7
+ 1
-5 43
21
23
25
27

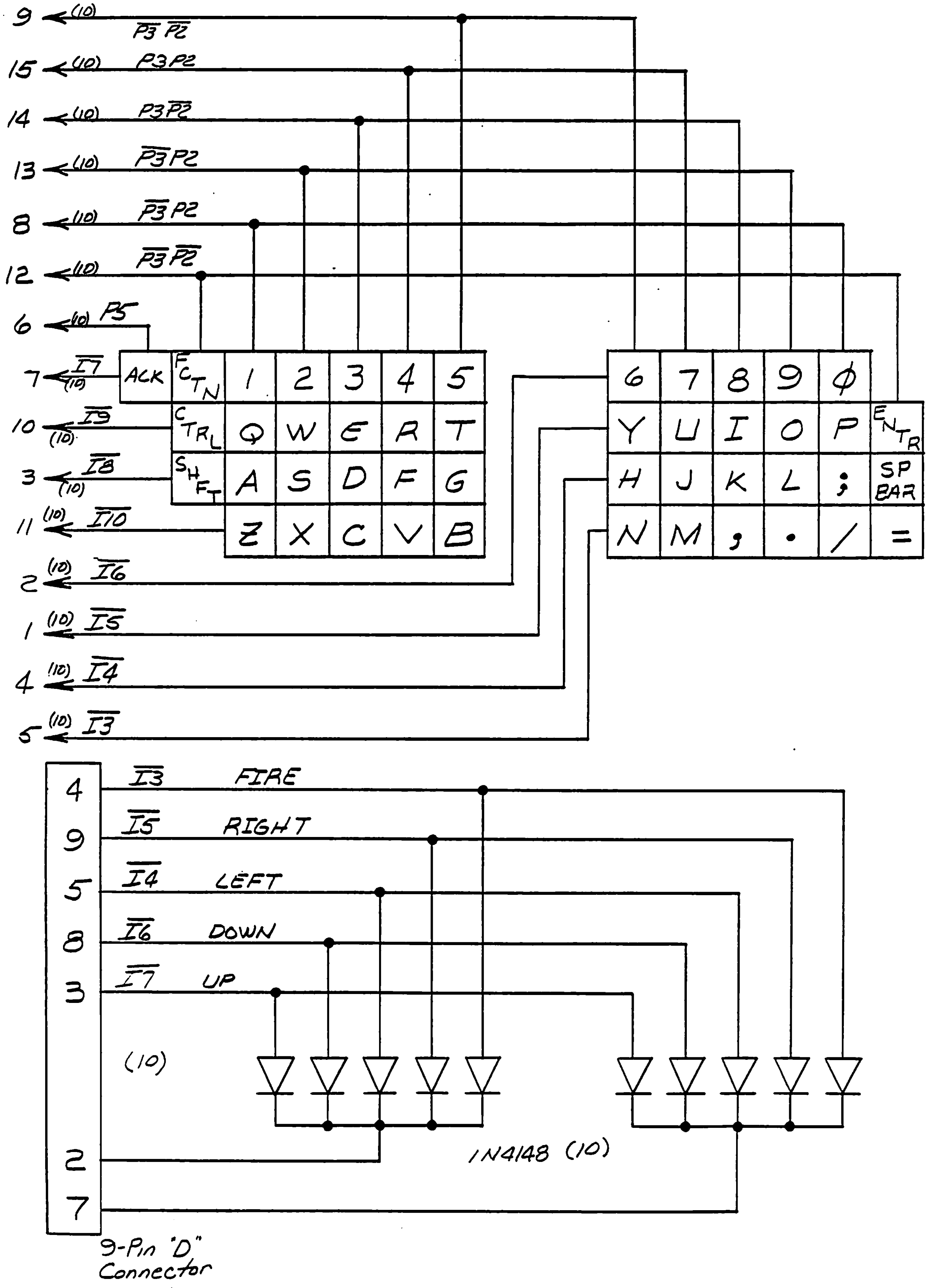
I/O PORT G-11



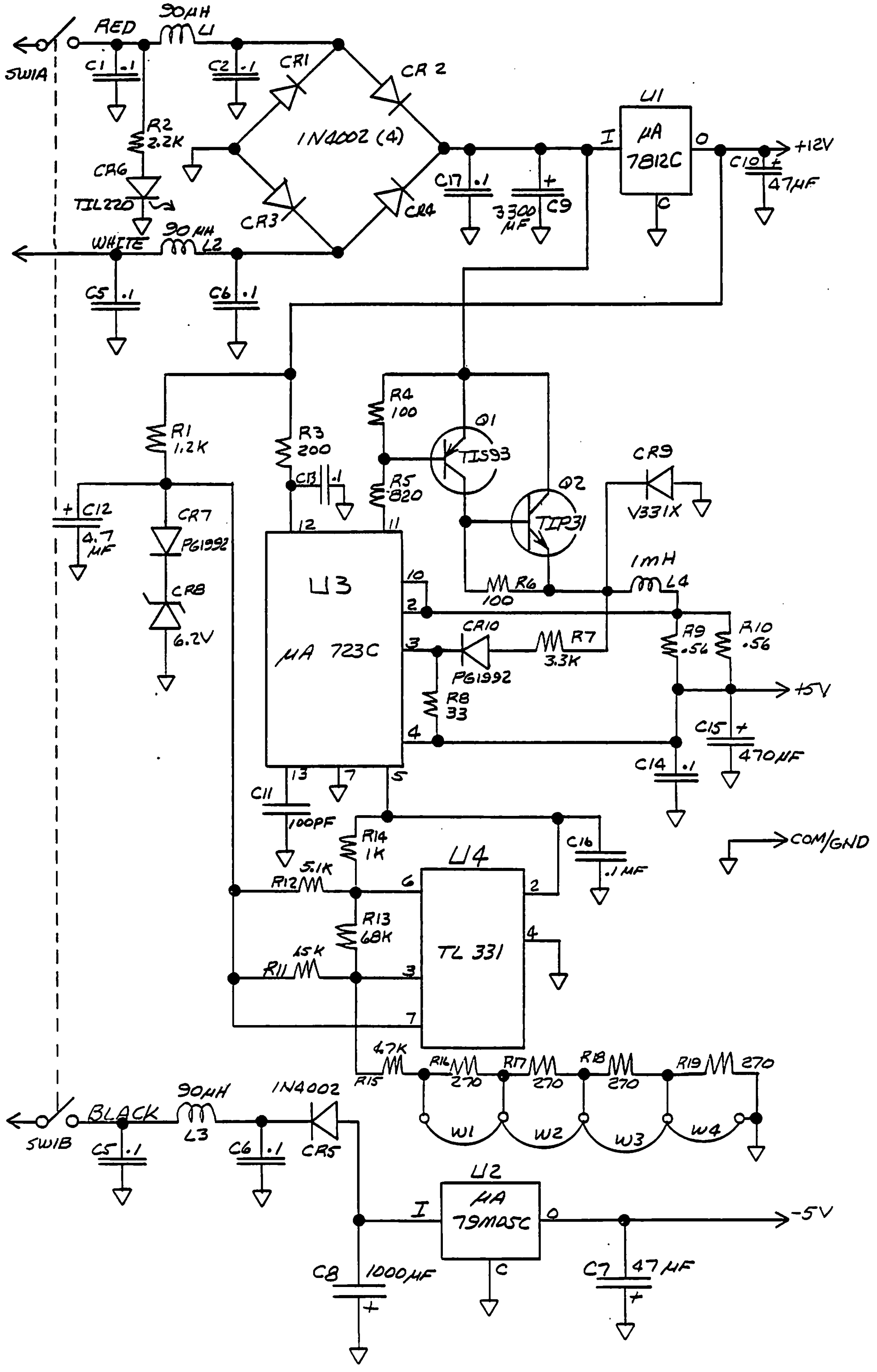
J/S DRIVE, CASSETTE MOTOR, WT



CASSETTE SENSE VIDEO RESTORE/BUFFER



KBD, JOYSTICKS



CONSOLE POWER SUPPLY

APPENDIX F

EXTENDED BASIC COMMAND MODULE

F.1 ORGANIZATION

The Extended Basic Command Module was manufactured in two basic versions. The production life of the first was short; therefore, consideration will be given only to the second version. It was organized around a 4-chip GROM bank, and a 12K ROM in the Command Module 8K memory space based at >6000. Paging was used to switch in the desired 4K half of the 8K ROM.

F.2 8K ROM PAGING

The 8K ROM is located in the upper half of the 8K ROM space, and must be paged in 4K blocks in order to fit. This is accomplished by WRITING to any address in the 8K space. If the WORD address is even (A14 = 0), the lower half of the 8K ROM will be enabled to respond at >7000. If the WORD address is odd (A14 = 1), then the upper half is able to respond.

The 4K ROM is based at >6000, and always responds there.

A SN74LS00 2-NAND and a SN74LS74A Flip-Flop comprise the paging control logic. The Schematic of this circuit is shown in Figure F-1.

F.3 GROM OPERATION

The GROM chips on the Extended Basic Command Module are connected in parallel, and respond with the Console GROM chips.

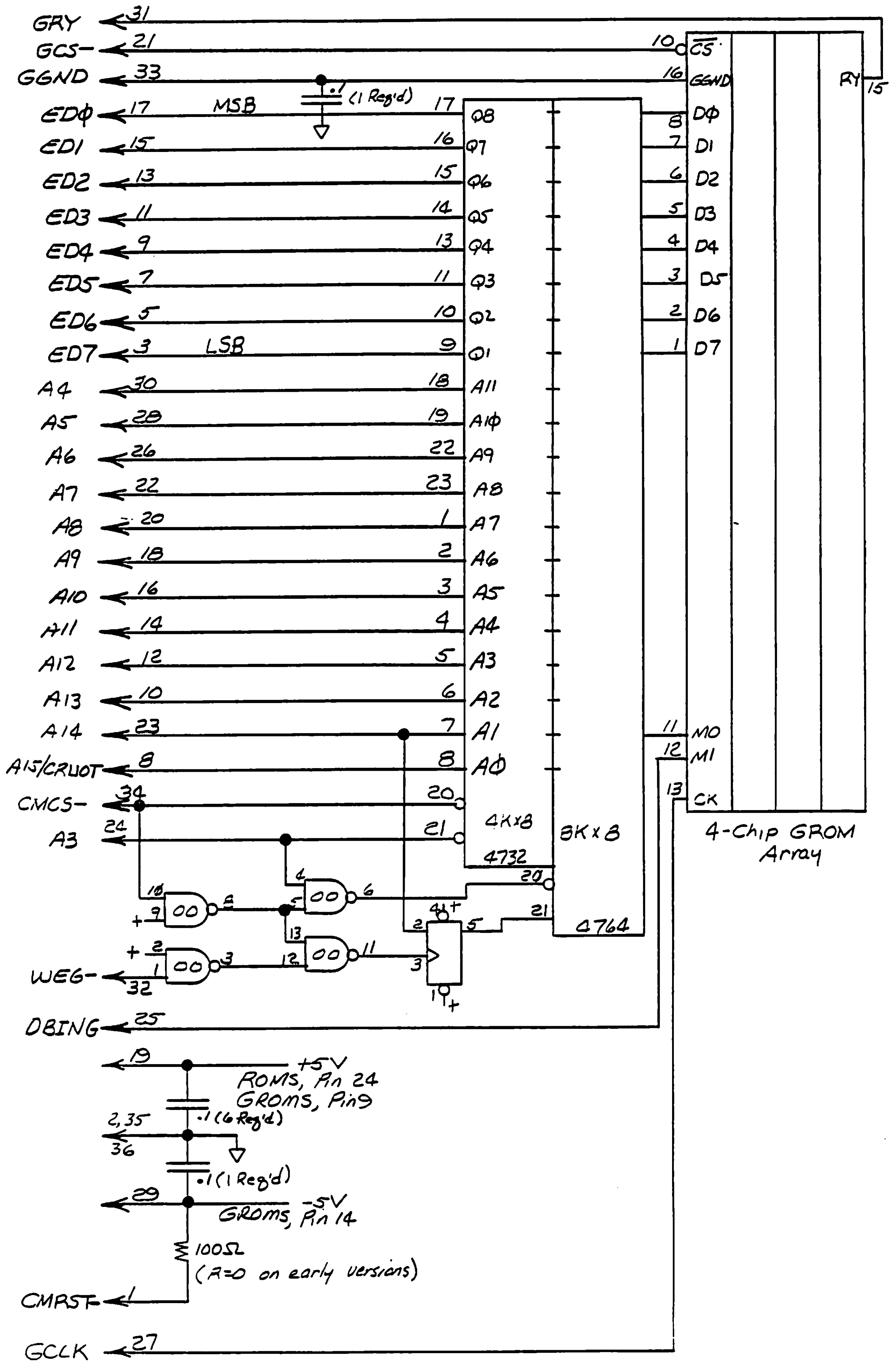


FIGURE F-1
EXTENDED BASIC C/M

use a random bit pattern for test patterns, execute code from the MEX, and have the Workspace Pointer in the MEX. Obviously, the test patterns must not overlay either the program for execution or the Workspace Pointer. The Workspace should be in a different bank of DRAM than is the test program for thorough testing. This allows a test of two successive word accesses, one in each bank, by using a Load Immediate instruction.

CHAPTER 9

RS232 CARD

9.1 RS232 ORGANIZATION

The RS232 Card features two RS-232C Ports and a single 8-bit Parallel I/O Port. The Parallel Port may be configured under software control to be either an Input Port, or an Output Port. Two Status lines and two Control lines, both CRU based, are included for use with the Parallel I/O Port. The two Control lines are set to a LOW level with a System Reset.

9.2 DEVICE SERVICE ROUTINE ROM

The DSR ROM on the RS232 Card is based at >4000 just as all TI-99/4 Peripheral DSR ROMs are, and it is a 4K byte ROM in the 8K possible peripheral space. The upper half of the DSR space is block decoded for the Parallel I/O Port. All 4K byte positions in the upper half of the DSR ROM respond as the Parallel I/O Port byte.

The 4K byte DSR ROM and the Parallel I/O Port respond only when the CRU "Page bit" is at a HIGH level. The Page Bit may be turned ON (set to a HIGH level) with either a SETO or a LDCR TMS 9900 instruction (a CRU operation).

9.3 PARALLEL PORT OPERATION

In the Input mode, the Parallel I/O Port is designed to insure that the data is stable on the internal Data Bus during the complete READ time period. A SN74LS373 Tristate Octal Latch is used for this function. A second '373 is used for the Output Register. This device type has buffered outputs; therefore, the Parallel I/O Port Output requires no further buffering before being presented to the outside world.

A MOVE BYTE operation from any address ranging from >5000 thru >5FFF gets Input data from the Parallel Input Port. Any MOVE BYTE operation to an address ranging from >5000 thru >5FFF puts data in the Parallel Port Output Register. This is true even if the port is configured as an Input Port. The data will appear at the Parallel I/O Port when the Port is reconfigured to be in the Output mode. Notice that the DSR ROM Page Bit must be ON before any memory space accesses of the RS232 PCB may occur.

9.4 CRU OPERATION

The RS232 Card differs from other peripheral cards in that it is designed to respond to one of two different CRU Base Addresses. This is apparently a factory option, and as can be best determined, the primary address of >1300 is the only version ever shipped. A simple change of a 100 Ohm resistor from one location to another causes the base address to be changed from >1300 to >1500. This resistor is labeled as R5 on the schematics. Notice that this is not a simple one bit address change, but a two bit change instead. This fact makes the equations for the decode more complicated, but it is hidden in the onboard PAL that takes care of all space decodes.

A LC network provides suppression on both of the two CRU Control outputs. The two CRU Status bits run directly into a SN74LS251 8/1 Multiplexer, and no RFI components are included in these runs.

The 4K byte DSR ROM responds from >4000 thru >4FFF when the CRU "Page bit" is at a HIGH level. The Page Bit may be turned ON (set to a HIGH level) with either a SETO or a LDCR TMS 9900 instruction (a CRU operation).

The Bidirectional Bus Driver on the Data Bus is controlled directly from the PAL device. It is enabled any time either the DSR ROM or Parallel I/O Port is being accessed.

The CRU space is fully decoded in that all Console address lines are used in the CRU space decoding. There are three areas of interest, and each are located >40 away from each other. The first is the general Status/Control CRU, and it is located on the CRU base address (either >1300 or >1500). The second function is the first or primary UART, a TMS 9902A, and it is based >40 above the CRU base of the card. The third function is the secondary UART, also a TMS 9902A, and it is based >80 above the card CRU Base. The >40 block located >C0 above the base is not used.

The Status CRU definition is defined in the following table.

Table 9-1 CRU STATUS DEFINITION

Bit #	Function
0	Spare bit
1	PIO PORT MODE CONTROL sense bit
2	PIO PORT Sense bit, 1=HIGH level
3	Spare PIO PORT Sense bit, 1=HIGH level
4	Flag bit sense, 1=set
5	CTS- sense, Primary RS232 Port, 0=active
6	CTS- sense, Secondary RS232 Port, 0=active
7	Card LED sense, 1=ON

The Control CRU definition is listed in the following table.

Table 9-1 CRU CONTROL DEFINITION

Bit #	Function
0	DSR ROM Page Bit, 1=active
1	PIO PORT MODE CONTROL, 1=INPUT
2	PIO OUTPUT "STROBE" bit, SETO=HIGH level
3	SPARE PIO OUTPUT STROBE bit, SETO=HIGH level
4	Flag bit, 1=set
5	CTS- control, Primary RS232 Port, 0=active
6	CTS- sense, Secondary RS232 Port, 0=active
7	Card LED control, 1=ON

9.5 UARTs

A pair of TMS 9902A UARTs is used to implement the RS-232C Port controllers. Complete information on these devices may be found in the Texas Instruments TMS 9902 Data Manual. The TMS 9902 is a UART especially developed for the TMS 9900 CRU concept. Aside from the fact that it must convert serial data from the TMS 9900 to a parallel format, and then back to serial again (but at a much lower bit rate) for the RS-232 Port(s), it is a rather pleasing device to work with. Among other things, it has the capability of looking at the level of the RS-232 serial data line directly, has a powerful interrupt generating capability, and has its own onboard programmable counter type of baud rate generator.

9.6 INTERRUPT CONFIGURATION

The RS232 Card is configured to use interrupts from both UARTs, and does so during normal operation. There are jumpers from drivers to the PEB Data Bus that are unused. Perhaps they were part of a "grander scheme" of sensing interrupts. It slightly resembles that used by the IBM 1800 computer to quickly sense which device(s) are causing an interrupt. The enable drive for gating this information to the backplane data Bus is no more than a pull-up resistor on the PEB Interface PCB. There is no reason why these jumpers need to be installed (even though they were at the factory) for normal operation.

9.7 RS232 PORT CONFIGURATION JUMPERS

A second set of jumpers has been provided to interchange sensing of pins 20 and 11 for the primary RS-232 Port, and pins 19 and 18 of secondary Port. Pins 20 and 19 have factory installed jumpers.

9.8 RS-232 VOLTAGE LEVEL GENERATION

A rather strange set of TTL to RS-232 translators are used. A TL082/TL084 Op Amp pair (six single amplifiers, total) is used for the Op Amp section, and a large number of discrete devices make up the rest of the circuit. The drivers are organized as the familiar inverting amplifier, but the "+" input is biased at 2.5V rather than ground. The only possible advantage of this circuit seems to be in limiting the output slew rate. RFI and RS-232C slew rate compliance result from a slower slew rate than that obtained from a device such as the SN75188 driver. Note that a large number of components will be eliminated for this device swap. An additional RC network that further slows down the wave fronts follows the Op Amp driver.

9.9 RS-232 RECEIVERS

The standard SN75189 RS232 Receivers are used to translate the received RS-232C levels to TTL levels required by the TMS 9902A UARTs. This type of receiver will function quite nicely when driven by TTL level signals; therefore, a balanced signal (with respect to Ground) is not required for proper operation. Low Power Shottky outputs have sufficed in the past for driving the SN75189 receiver.

9.10 UART RESET

There is no Hardware RESET available on the TMS 9902A UARTs. They must be reset by Software when that is required. One of these times is during Power-Up. At this time, the Console turns on the DSR ROM page bit for each peripheral, starting at a CRU Base Address of >1000, and looks at the header information in the DSR ROM. If Power-Up service for that Peripheral is required, control is transferred from the Console ROM to the DSR ROM of the peripheral involved. After the Power-Up reset routine has been executed, control is returned back to the Console, and the Page Bit is turned off before proceeding to the next peripheral.

9.11 RS232 PROGRAMMABLE LOGIC ARRAY

A Monolithic Memories PAL device is used to provide all of the required Memory and CRU space decoding. Equations for this are in the following table. Monolithic Memories, Inc. notation is used.

Table 9-1 PAL EQUATIONS

Pin #	Equations
13	(pin 1)*(pin 2)*A1*/A3*MEMEN*PCBEN
	this expands to
	/A0*A1*/A2*/A3*MEMEN*PCBEN*AMC*AMB*AMA*PAGE
14	(pin 1)*(pin2)*A1*MEMEN*PCBEN
	this expands to
	/A0*A1*/A2*MEMEN*PAGE*PCBEN
15	(pin 1)*(pin 2)*/A0*A1*A3*MEMEN*PCBEN
	this expands to
	/A0*A1*/A2*A3*MEMEN*PCBEN*AMC*AMB*AMA*PAGE

```
16     (pin 2)*(pin5)*/A1*A3*/A5*A6*/A8*/A9*/MEMEN*PCBEN*/RES

             +

       (pin 2)*(pin 5)*/A1*A3*A5*/A6*/A8*/A9*/MEMEN*PCBEN*RES

                          this expands to

       /A0*/A1*/A2*A3*/A4*/A5*A6*A7*/A8*/A9*/MEMEN*PCBEN*/RES

             +

       /A0*/A1*/A2*A3*/A4*A5*/A6*A7*/A8*/A9*/MEMEN*PCBEN*RES
17      (pin 2)*(pin5)*/A1*A3*/A5*A6*/A8*A9*/MEMEN*PCBEN*/RES

             +

        (pin 2)*(pin 5)*/A1*A3*A5*/A6*/A8*A9*/MEMEN*PCBEN*RES

                          this expands to

        /A0*/A1*/A2*A3*/A4*/A5*A6*A7*/A8*A9*/MEMEN*PCBEN*/RES

             +

        /A0*/A1*/A2*A3*/A4*A5*/A6*A7*/A8*A9*/MEMEN*PCBEN*RES
18      (pin 2)*(pin5)*/A1*A3*/A5*A6*A8*/A9*/MEMEN*PCBEN*/RES

             +

        (pin 2)*(pin 5)*/A1*A3*A5*/A6*A8*/A9*/MEMEN*PCBEN*RES

                          this expands to

         /A0*/A1*/A2*A3*/A4*/A5*A6*A7*A8*/A9*/MEMEN*PCBEN*/RES

            +

         /A0*/A1*/A2*A3*/A4*A5*/A6*A7*A8*/A9*/MEMEN*PCBEN*RES
```

The pin definitions of the PAL are given in the following table.

Table 9-1 PAL PIN DEFINITION

1	V1	11	/MEMEN
2	V2	12	PCBEN
3	A1	13	/DSRROM

Table 9-1 PAL PIN DEFINITION, CONTINUED

4	A3	14	/BUS DRVR ENA
5	V3	15	/PARALLEL I/O
6	A5	16	/CRU
7	A6	17	/UART0
8	A8	18	/UART1
9	A9	19	RES
10	GND, POWER	20	VCC, POWER

9.12 RS232 PORT PIN DEFINITION

The pin-outs of the RS-232C 25-pin "D" connector are listed in the table that follows.

Table 9-1 25-PIN D CONNECTOR PIN-OUTS

Pin #	Function	Pin #	Function
1	Equip't/Logic GND	12	DCD- from Sec UART
2	Data to Pri UART	13	Sec CTS-, CRUOUT
3	Data from Pri UART	14	Data to Sec UART
5	Pri CTS-, CRU OUT	16	Data from Sec UART
6	1.8K PU Res to +12V	19	DTR- to Sec UART
7	Logic ground	20	DTR- to Pri UART
8	DCD- from Pri UART		

9.13 PARALLEL PORT PIN DEFINITIONS

The connector for the Parallel I/O Port is a 16-pin, twin row, pin-header connector, and is organized as shown below.

```
----------¦   ¦-------
¦15 13 11 9   7 5 3 1¦
¦16 14 12 10  8 6 4 2¦
----------------------
      Front View
```

Table 9-1 PARALLEL I/O PORT PIN DEFINITION

Pin #	Function	Pin #	Function
1	Control Strobe	9	Data 0, MSB
2	Data 7, LSB	10	Sense Input
3	Data 6	11	Spare Sense Input
4	Data 5	12	1K PU Res to +5V
5	Data 4	13	10 Ohm PU Res to +5V
6	Data 3	14	Logic Ground
7	Data 2	15	Spare Control Strobe
8	Data 1	16	Logic Ground

9.14 RS232 CARD AND MODULE COMMENTS

In concluding the discussion of the RS232 Card, it might be appropriate to contrast the "Plug Up" (or "Side Car") RS232 Module with the RS232 Card. The RS232 Card has a Parallel I/O Port, uses Programmable Logic Array for all of its decoding, decodes the Most Significant three Address Bus Bits in the CRU decode, uses Op Amp TTL/RS-232 drivers, shares a single 25-pin D Connector with both RS-232 Ports, and had direct control over the control signal CTS-.

The RS232 Module had no Parallel I/O Port, used SSI/MSI logic for all space decodes, did not include the MS three Address bits in the CRU space decode, used SN75188 TTL/RS232 translators with shunt .001 uF capacitors on the outputs, had a 25-pin D connector for each RS-232 Port, and had its own on-board Power Supply.

CHAPTER 10

P-CODE CARD

10.1 P-CODE CARD DESCRIPTION

The P-Code Card is the simplest of the four peripheral cards, as it contains only ROM and GROM functions.

10.2 DEVICE SERVICE ROUTINE ROMS AND GROMS

The ROM portion is comprised of a 4K byte ROM based at >4000, and an 8K byte ROM paged to respond as a 4K byte ROM based at >5000. The MSB of the 8K byte ROM is obtained from an on board CRU controlled register. If that bit is at a LOW level, the lower half of the ROM responds at the >5000 base. If the CRU bit is ON, the upper half responds in the place of the lower one. The full 8K byte peripheral space is covered in this manner.

10.3 P-CODE CRU

There is only a CRU Output function used on the P-Code Card. It is split in the CRU address space, and is defined in the following table.

Table 10-1 CRU CONTROL DEFINITION

Displacement from >F000	Function
0	DSR ROM Page Bit, 1=enabled
1	Not used
2	Not used
3	Not used

Table 10-1 CRU CONTROL DEFINITION, CONTINUED

Displacement from >F800	Function
0	8K ROM Page bit, 0=Lower half
1	Not used
2	Not used
3	Optional LED Control, 1=ON

10.4 P-CODE PROGRAMMABLE LOGIC ARRAY

The GROM chips respond in the DSR ROM space; therefore, the ROM at the GROM address must be trapped, and disabled when GROM accesses occur. There are two address blocks that are trapped out of the DSR ROM space. One is for a GROM READ, and the other is for a GROM WRITE. The GROM READ trap is from >5BFC through >5BFF, and the WRITE trap is from >5FFC through >5FFF. A PAL 12L6 and several SSI gates are used to perform the required decoding. The action of the PAL is somewhat obscured by the support SSI logic; therefore, both the PAL and full logic equations will be given. The PAL equations are as follows.

Table 10-1 PAL EQUATIONS

PAL Pin	Equation
13	PCBEN*MEMEN*DBIN*A1*A3*V1*V2*/V4
	+
	PCBEN*MEMEN*DBIN*A1*A3*/A9*V1*V2
14	PCBEN*MEMEN*DBIN*A1*/A3*V1*V2
15	PCBEN*MEMEN*A1*V1*V2
16	PCBEN*MEMEN*DBIN*A1*A3*/A5*A9*V1*V2*V4
	+
	PCBEN*MEMEN*/DBIN*A1*A3*A5*A9*V1*V2*V4
17	PCBEN*/MEMEN*CRUCLK*/A1*A3*/A9*V2*V3
18	Not Used

Table 10-1 PAL EQUATIONS, CONTINUED

V1=(PAGE)*AMC*AMB*AMA	(PAL pin #1)
V2=/(/A0*/A2)	(PAL pin #2)
V3=A4*A5*A6*A7	(PAL pin #5)
V4=/(A4*A6*A7*A8*A10*A11*A12*A13)	(PAL pin #7)

The expanded equations are listed in the same order as those for the PAL, and are as follows.

Table 10-1 EXPANDED EQUATIONS

```
PAL Pin                          EQUATIONS
-------                          ---------
  13      PCBEN*MEMEN*DBIN*PAGE*AMC*AMB*AMA*/A0*A1*/A2*A3*

              /(A4*A6*A7*A8*A10*A11*A12*A13)    +

  14      PCBEN*MEMEN*DBIN*PAGE*AMC*AMB*AMA*/A0*A1*/A2*A3*/A9

  15      PCBEN*MEMEN*DBIN*PAGE*AMC*AMB*AMA*/A0*A1*/A2

  16      PCBEN*MEMEN*DBIN*PAGE*AMC*AMB*AMA*/A0*A1*/A2*A3*

               A4*/A5*A6*A7*A8*A9*A10*A11*A12*A13)    +

          PCBEN*MEMEN*/DBIN*PAGE*AMC*AMB*AMA*/A0*A1*/A2*A3*

               A4*A5*A6*A7*A8*A9*A10*A11*A12*A13)

  17      PCBEN*/MEMEN*CRUCLK*/A0*/A1*/A2*A3*A4*A5*A6*A7*/A9

  18      Not Used
```

As one may easily see, these equations are rather messy, and the PAL was most certainly a blessing to the Logic Designer.

The PAL pin definition is in the following table.

Table 10-1 PAL PIN DEFINITION

1	V1	11	/MEMEN
2	/V2	12	PCBEN
3	A1	13	/ROM8KEN
4	A3	14	/ROM4KEN
5	V3	15	/BUS DRVR ENA
6	A9	16	/GROM ENA
7	/V4	17	/CRUST
8	/CRUCLK	18	not used
9	DBIN	19	A5
10	GND, POWER	20	VCC, POWER

10.5 GROM CHIP INTERFACE

The GROM chips are not TTL compatible for HIGH logic levels, and some mechanism must be included to gain that property for the eight chip GROM array. Two methods are employed for this. The first utilized a CMOS Bidirectional Bus Driver to provide the 8-bit Data Bus compatibility. The second used open collector gates with pull-up resistors for the four control drivers. For more information on GROM chips, see the part of this Manual devoted to the Console GROM chips.

OR gating is provided to disable switching of the GROM bank control lines unless the bank is selected. This is an excellent method of lowering the system level noise that is generated by driving large capacitive loads.

The GROM clock was obtained by dividing the Phase 3 Clock by eight. This clock is somewhat slower than that used by the Console GROMs.

10.6 P-CODE CARD DISABLE

When the P-Code Card is plugged into the PEB, and the system comes up under that operating system, a fairly long boot time is required. A means of disabling the P-Code Card was provided to allow the system to come up under the normal operating system when that of P-COCE is not desired. This mechanism is a switch located on the "tongue" of the PCB, and the switch logically shorts the signal "PCBEN" at the PAL to Ground. This in turn disables the P-Code Card. An open collector non-inverting gate with a pull-up resistor on its output is connected in series with the signal PCBEN between the I/O connector and the PAL. The output of this gate also goes to one terminal of a SPST switch. The other switch terminal is Grounded. When the switch is closed, the P-Code Card is disabled.

A wire jumper option is provided to select which CRU Output Register bit turns ON the LED indicator. The LED is factory selected to come ON with the DSR ROM Page Bit.

10.7 P-CODE CARD AND MODULE

The only difference between the P-Code "Side Car" Module and the P-Code Card is the method by which decodes were made and the clock frequency of the GROM clock (which was slightly faster on the Module).

CHAPTER 11

DISK CONTROLLER CARD

11.1 DISC CONTROLLER ORGANIZATION

The Disk Controller Card supports up to three Double Sided Single Density disk drives. The controller chip is a Western Digital FD1771.

11.2 DEVICE SERVICE ROUTINE ROMS

Two 4K byte ROMs comprise the DSR Space. A 16 byte space is trapped out of the high end (>5FF0 through >5FFF) of the upper ROM to support the FD1771 Disk Controller chip (only four bytes are required). The first 4K byte ROM is based at >4000, and the second at >5000. Both ROMs and the FD1771 chip are enabled for accesses by the CRU page bit.

11.3 FD1771 DISK CONTROLLER CHIP

The FD1771 Chip Select pin is driven LOW when the WORD addresses from >5FF0 through >5FFE are active. Either the Read Enable or the Write Enable input pin will be driven when the MSBY of the WORD is accessed 1 us later. A15/COUT is the gating signal for the Read Enable pin, and both A15/COUT and WE* are ANDed together to drive the Write Enable Pin. The operation of this divide is discussed in the 1976 INTERFACE AGE October, November, and December issues.

11.4 CRU ORGANIZATION

The CRU is organized around an 8-bit input Status Port, and an 8-bit output Control Register. Both are based at >1100. Six of the Status lines are functional, one is connected to Logic Ground, and the other line is connected to VCC. All eight of the Control lines are functional.

11.4.1 Disk Controller CRU Status Port

The 8-bit Status Port is defined in the following table.

Table 11-1 DISK CONTROLLER CRU STATUS PORT

Bit #	Function
0	Head Load Sense from 1771
1	Drive #1 Connected Sense, 0=connected
2	Drive #2 Connected Sense, 0=connected
3	Drive #3 Connected Sense, 0=connected
4	Drive Ena Timeout Sense, 1=timed out
5	Logic GND (Sense=0)
6	VCC (Sense=1)
7	Side Select Sense, 0=side 2, 1=normal

11.4.2 Disk Controller CRU Control Register

The 8-line CRU Output Port is defined in the following table.

Table 11-1 CRU CONTROL REGISTER DEFINITION

Bit #	Function
0	DSR ROM/ FD1771 Page Enable, 1=enabled; LED control, 1=ON
1	Drive Keep-alive Clock
2	FD 1771 Access control, 1=Wait Enable
3	Head Load Timing input to FD 1771
4	Drive #1 Select, 1=selected
5	Drive #2 Select, 1=selected
6	Drive #3 Select, 1=selected
7	Side Select, 0=side 2, 1=normal

11.5 PROGRAMMABLE LOGIC ARRAY DEFINITION

The PAL for the Disk Controller PCB furnishes decodes for each of the two DSR ROMs, the FD1771 Chip Select, the Bi-directional Data Bus Driver (also functions as the Data Bus Buffer from the system), and a single decode for the CRU Status and Control ports. The PAL is augmented by a 3-NOR, a 3-NAND, and an 8-NAND to obtain an acceptable level of inputs and logical minterms. The 8-NAND is used to NAND Address Bus bits 4 through 11 together for FD1771 and upper ROM use, the 3-NAND to NAND the three most significant of the 19 Address Bus bits together (they are permanently connected to a HIGH level on the Interface Card to the Console), and the 3-NOR ANDs the negatives of Address Bus bits 4, 5, and 6. The equations are listed in the following table. Monolithic Memories Inc. symbology is used in the table.

Table 11-1 DISK CONTROLLER PAL EQUATIONS

LOROM = V1*/A00*A01*/A02*/A03*DSKPG*MEMEN*PCBEN

HIROM = V1*/A00*A01*/A02*A03*V3*DSKPG*MEMEN*PCBEN

BDRVR = V1*/A00*A01*/A02*DSKPG*MEMEN*PCBEN

WDEN = V1*/A00*A01*/A02*A03*/V3*DSKPG*MEMEN*PCBEN

CRU = /A00*/A01*/A02*A03*A07*V2*/MEMEN*PCBEN

Spare Output

1	V1 (3-NAND)	11	VCC (LOGIC)
2	A00	12	V2 (3-NOR)
3	A01	13	/LOROM
4	A02	14	/HIROM
5	A03	15	/BDRVR
6	DSKPG	16	/WDEN
7	A07	17	/CRU
8	/MEMEN	18	Spare Output
9	PCBEN	19	V3 (8-NAND)
10	GROUND (POWER)	20	VCC (POWER)

11.6 CONSOLE READY CONTROL

The System READY line is driven "Not Ready" only when data is being Written TO or Read FROM the Disk Drive. In this operation a data byte is required only every 64 us (the bit rate is 8 us). The Console is held in the Not Ready state from the time it requests another byte to the time the FD1771 is able to furnish that byte, or vice versa. This type of accessing only occurs if bit-2 of the CRU Control Register is set to a HIGH level.

A Drive Keep-alive time-out, a FD1771 Interrupt, or a FD1771 Data Request will cause the Console to go from NOT-READY to READY.

11.7 DRIVE KEEP-ALIVE

The Drive Keep-alive circuit is composed of one-half of a

SN74LS123 retriggerable One-shot that is triggered from bit-1 of the CRU Control register. The timing components, a 200K resistor and a 47 uF capacitor, provide a calculated 3.1 second pulse width from the last trigger applied. The One-shot must be strobed more often than this to maintain a constant motor on state. The Q' output of the One-shot may be sensed on bit-4 of the CRU Status input.

If the CRU Drive Select bit is set to a HIGH level (ON), and the Keep-alive One-shot is triggered, then the "DRIVE SELECT" line for that drive is active (at a LOW level).

The One-shot output is also provided in the logic used for controlling Console memory READY during data transfers to and from the FD1771 chip.

11.8 INTERRUPT STRUCTURE

The FD1771 Disk Controller chip is the only source of interrupts from this card to the console. There are two tristate gates connected to function as open collector devices that are included in the interrupt logic. The first gate drives the common interrupt line, and the second drives the Data Bus bit-0. The latter gate must be enabled by a signal called SENILB*. SENILB* is tied to a HIGH level on the Interface Card in the PEB; thus, data may never be gated to the Data Bus from this source.

11.9 DATA/CLOCK SEPARATOR

The Data/Clock Separator used is one that was described in the May 24, 1979 issue of ELECTRONIC DESIGN. The article starts on page 154, and the circuit is shown on page 155. The circuit description starts on page 157.

There are several purposes of the Data/Clock Separator. The most obvious is to separate the raw data stream from the Disk Drive into CLOCK pulses and DATA pulses for the FD1771 controller chip. Additionally, it must shape the raw data pulse width for the FD1771 chip, compensate for missing clock pulses, and resync to look for CLOCK pulses if too many DATA pulses occur (4 or more) with no CLOCK pulses placed in between. During the execution of the address mark, 3 successive missing CLOCK pulses occur normally.

FM data recording is used at a data rate of 125 KHz to yield a CLOCK period of 8 us. A 4 us displacement occurs between the leading edge (LE) of the CLOCK and the LE of the DATA pulse if a Data Pulse occurs.

11.9.1 Data/Clock Steering Logic

Steering logic is required to switch the raw data stream from the Data input of the FD1771 to its Clock input. This consists of a pair of SN74LS00 2-NAND gates that are connected to steer the raw Disk Data either to the FD1771 Clock or Data input. Control is accomplished such that the full pulse width of the disk data stream must occur before it is possible to change the steering logic. The logic is enabled by a One-shot that shapes the raw data from the drive to an approximate 500 ns pulse width for the FD1771 chip.

11.9.2 Switching Logic

The purpose of the Switching Logic is to provide the steering control signal to the 2-NAND gates for steering. Raw disk data is steered to the data output if a clock output has occurred within the last 6 us, or if less than four data outputs have occurred with no clock in between.

The Switching Logic is built around several 2-NAND gates, and a One-shot that is timed for 6 us.

11.9.3 Missing Clock Compensation

Up to 3 missing CLOCK pulses will occur normally in the generation of the address mark in the header area preceding the actual data. Steering must be generated to switch over to the DATA path even though no CLOCK occurred (to normally cause switching). There will always be a DATA pulse (by definition) just prior to (4 us) the missing clock. That DATA pulse is detected, and a pseudo CLOCK is later generated and fed to the switching logic just as a separated CLOCK pulse normally is. This pseudo clock is inhibited if a normal CLOCK occurs, and the pseudo CLOCK is not fed to the CLOCK pulse path to the FD1771 chip.

If 4 CLOCK pulses are missing, the Missing Clock Compensation is inhibited, and the Switching Logic is set up to steer the very next raw data pulse to the CLOCK pulse line. A second One-shot timed for 5.4 us and a "D" flip-flop provide this logic.

11.9.4 Missing Clock Counter

Still another missing clock function is required in the form of a Missing Clock Counter. This is for resynchronizing purposes if the separator accidentally sync's to DATA rather than CLOCK pulses in the raw data stream. This incorrect sync can be caused by the "write splice" area between the header information and the data information, or in the inter-record gap between sectors.

The Missing Clock Counter is INCREMENTED by separated DATA pulses and RESET by separated CLOCK pulses. If no separated CLOCK pulses occur between four separated DATA pulses, the counter "C" output is used to inhibit the Missing Clock Compensator from providing the pseudo CLOCK pulse to the Switching Logic. Three DFFs are used to count the steered raw data pulses to the Data output.

11.10 OSCILLATOR

A separate 8 mHZ oscillator is provided as a clocking agent for the FD1771 chip. The fundamental frequency is divided by eight before feeding the FD1771.

11.11 HEAD LOAD TIMING CONTROL

The Head Load timing is generated from bit-3 the CRU Output port.

11.12 DRIVE CONNECTED SENSE

Logic is provided to determine if a drive is connected. This function is implemented with a sense amplifier connected to look for the presence of a terminating resistor on the Drive Select pin of the Disk Drive. If a drive is connected, its terminating resistor causes the "-" input of the sense amplifier to be pulled away from Logic Ground, and past the "+" input of the sense amplifier. Thus, the output of the sense amplifier changes phase.

The outputs of the sense amplifiers are sensed on bits 1, 2, and 3 of the CRU Status Port.

11.13 DRIVE PORT CONNECTOR DEFINITION

The following table lists the pin definitions for the Disk Drive I/O Connector.

Table 11-1 DRIVE PORT CONNECTOR PIN DEFINITION

PIN	TYPE	FUNCTION
1		Logic GROUND
2		not used
3		Logic GROUND
4		not used

Table 11-1 DRIVE PORT CONNECTOR PIN DEFINITION, CONTINUED

PIN	TYPE	FUNCTION
5		Logic GROUND
6		not used
7		Logic GROUND
8	recve	Index Pulse from Drive, LOW TRUE PULSE
9		Logic GROUND
10	drive	Drive Select #1, LOW=SELECTED
11		Logic GROUND
12	drive	Drive Select #2, LOW=SELECTED
13		Logic GROUND
14	drive	Drive Select #3, LOW=SELECTED
15		Logic GROUND
16	drive	Motor control, LOW=ON
17		Logic GROUND
18	drive	Head Step Direction, LOW=STEP OUT
19		Logic GROUND
20	drive	Head Step pulse, LOW TRUE PULSE
21		Logic GROUND
22	drive	Write Data, LOW=TRUE
23		Logic GROUND
24	drive	Write Gate, LOW=WRITE ENABLE
25		Logic GROUND
26	recve	Track 00, LOW=TRACK 00
27		Logic GROUND
28	recve	Write Protect, LOW=PROTECTED
29		Logic GROUND
30	recve	Read Data, LOW=TRUE
31		Logic GROUND
32	drive	Side Select, 0=Opposite, 1=Normal
33		Logic GROUND
34		not used

11.14 CARD/MODULE DIFFERENCES

The PEB Card provides two functional enhancements over the original Disk Controller Module: 1) the capabilities of accessing a second side, and 2) determining if a drive is connected. A PAL was used in the Card to replace SSI logic in the address space decoding, and the address space trapped for the FD1771 is smaller in the Card. The Module has its own internal power supply, while the Card gets its power from the PEB.

APPENDIX A

COMMAND MODULE PORT PIN DEFINITIONS

Pin #	Label	Type	Description
1	CRS	S	C/M Reset (to -5V)
2	GND	OUT	Logic Ground
3	D7	I/O	Data Bus, LSB
4	CRUCLK-	OUT	Active LOW CRU Clock, (No-connect on late models)
5	D6	I/O	Data Bus
6	CRUIN	IN	CRU Input Data (No-connect on late models)
7	D5	I/O	Data Bus
8	A15/COUT	OUT	Address Bus LSB/CRU Output Data
9	D4	I/O	Data Bus
10	A13	OUT	Address Bus
11	D3	I/O	Data Bus
12	A12	OUT	Address Bus
13	D2	I/O	Data Bus
14	A11	OUT	Address Bus
15	D1	I/O	Data Bus
16	A10	OUT	Address Bus
17	D0	I/O	Data Bus, MSB
18	A9	OUT	Address Bus

Pin #	Label	Type	Description
19	VCC	OUT	+5V, regulated
20	A8	OUT	Address Bus
21	GCS-	OUT	Active Low GROM Chip Select
22	A7	OUT	Address Bus
23	A14	OUT	Address Bus
24	A3	OUT	Address Bus
25	DBIN	OUT	Data Bus Direction, 1=READ
26	A6	OUT	Address Bus
27	GCLK	OUT	GROM Clock (447.441 KHz)
28	A5	IN	Address Bus
29	-5V	IN	-5V, regulated
30	A4	OUT	Address Bus
31	GRDY	IN	GROM READY, 1=Ready
32	WE-	OUT	Active LOW Write Enable
33	GGND	OUT	-.7V GROM Ground, GND on late model
34	CMCS-	OUT	Active LOW C/M Memory Space CS
35	GND	OUT	Logic Ground
36	GND	OUT	Logic Ground

NOTE: The Address Bus, Memory Data Bus, and the CRU Data Bus levels are HIGH true.

APPENDIX B

I/O PORT PIN DEFINITIONS

Pin #	Label	Type	Description
1	+5V	OUT	+5V, regulated
2	SBE	OUT	Active HIGH Speech Synthesizer Ena
3	RESET-	OUT	Active LOW System Reset
4	EXTINT-	IN	Active LOW Interrupt
5	A5	OUT	Address Bus
6	A10	OUT	Address Bus
7	A4	OUT	Address Bus
8	A11	OUT	Address Bus
9	DBIN	OUT	Data Bus Direction, 1=READ
10	A3	OUT	Address Bus
11	A12	OUT	Address Bus
12	READY	IN	System READY, 1=Ready
13	LOAD-	IN	Active LOW Load Input
14	A8	OUT	Address Bus
15	A13	OUT	Address Bus
16	A14	OUT	Address Bus
17	A7	OUT	Address Bus
18	A9	OUT	Address Bus

Pin #	Label	Type	Description
19	A15/COUT	OUT	Address Bus LSB/CRU Data Out
20	A2	OUT	Address Bus
21	GND	OUT	Logic Ground
22	CRUCLK-	OUT	Active LOW CRU Output Clock
23	GND	OUT	Logic Ground
24	PH3-	OUT	Active LOW Phase 3 Clock
25	GND	OUT	Logic Ground
26	WE-	OUT	Active Low Write Enable
27	GND	OUT	Logic Ground
28	MBE-	OUT	Active Low Peripheral DSR Space Decode, not sent to PEB
29	A6	OUT	Address Bus
30	A1	OUT	Address Bus
31	A0	OUT	Address Bus, MSB
32	MEMEN-	OUT	Active LOW Memory Enable
33	CRUIN	IN	CRU Input Data
34	D7	I/O	Data Bus, LSB
35	D4	I/O	Data Bus
36	D6	I/O	Data Bus
37	D0	I/O	Data Bus, MSB
38	D5	I/O	Data Bus
39	D2	I/O	Data Bus
40	D1	I/O	Data Bus
41	IAQ/HDA	OUT	Active High Logical OR of TMS 9900 IAQ and HOLDA

Pin #	Label	Type	Description
42	D3	OUT	Data Bus
43	-5V	OUT	-5V, regulated
44	AUDIO	IN	Analog Audio Input to Console

NOTE: THE ADDRESS BUS, THE MEMORY DATA BUS, AND THE CRU BUS LEVELS ARE HIGH TRUE.

APPENDIX C

PEB BACKPLANE PIN DEFINITION

Pin #	Label	Type	Description
1		OUT	+5V Raw DC
2		OUT	+5V Raw DC
3	GND	OUT	Logic Ground
4	READY	I/O	System READY, 1=Ready
5	GND	OUT	Logic Ground
6	RESET-	OUT	System Reset, 0=Reset
7	GND	OUT	Logic Ground
8			Not used
9			Not used
10	AUDIO	OUT	Analog Audio Input for Console
11	RDBENA-	IN	Active LOW Cable Data Bus Driver Enable
12	PCBEN	IN	Active HIGH PCB Master Enable, 1=Enabled
13	HOLD-	IN	HOLD to Console, 0=Hold
14	IAQHDA	OUT	Logical OR of Console IAQ and HOLDA, 1=True
15	SENILA-	OUT	Pulled up on PEB I/F PCB
16	SENILB-	OUT	Pulled up on PEB I/F PCB
17	INTA-	IN	Active LOW Interrupt to Console
18	LOAD-	IN	Console LOAD input, 0=Load

Pin #	Label	Type	Description
19	D7	I/O	Data Bus, LSB
20	GND	OUT	Logic Ground
21	D5	I/O	Data Bus
22	D6	I/O	Data Bus
23	D3	I/O	Data Bus
24	D4	I/O	Data Bus
25	D1	I/O	Data Bus
26	D2	I/O	Data Bus
27	GND	OUT	Logic Ground
28	D0	I/O	Data Bus, MSB
29	A14	OUT	Address Bus
30	A15/COUT	OUT	Address Bus LSB/CRU OUT bit
31	A12	OUT	Address Bus
32	A13	OUT	Address Bus
33	A10	OUT	Address Bus
34	A11	OUT	Address Bus
35	A8	OUT	Address Bus
36	A9	OUT	Address Bus
37	A6	OUT	Address Bus
38	A7	OUT	Address Bus
39	A4	OUT	Address Bus
40	A5	OUT	Address Bus
41	A2	OUT	Address Bus
42	A3	OUT	Address Bus

Pin #	Label	Type	Description
43	A0	OUT	Address Bus, MSB
44	A1	OUT	Address Bus
45	AMB	OUT	Address Bus, pulled up on PEB I/F PCB
46	AMA	OUT	Address Bus, pulled up on PEB I/F PCB
47	GND	OUT	Logic Ground
48	AMC	OUT	Address Bus, pulled up on PEB I/F PCB
49	GND	OUT	Logic Ground
50	PH3-	OUT	Active LOW Phase 3 Console clock
51	CRUCLK-	OUT	Active LOW CRU Output Clock
52	DBIN	OUT	Console Data Bus Direction, 1=READ
53	GND	OUT	Logic Ground
54	WE-	OUT	Active LOW Console Write Enable
55	CRUIN	IN	CRU DATA to the Console
56	MEMEN-	OUT	Active LOW Console Memory request
57		OUT	-12V/-5V raw DC
58		OUT	-12V/-5V raw DC
59		OUT	+12V
60		OUT	+12V

NOTE: The Address Bus, Memory Data Bus and the CRU Data Bus levels are HIGH true.

APPENDIX D

A GROM SIMULATOR

We will define a GROM simulator by keeping in mind that the GROM chip is: a 6K x 8 ROM array, has an internal Address Counter, is connected in parallel with other GROM chips when in a bank, gates READ DATA to the data bus only when its page number agrees with the page register, steps together with all other GROM chips when connected in a bank, and is a synchronous machine as far as Timing and Control goes. Life will be considerably simpler if we make the assumption that we are to simulate only the GROMs on the outside of the TI 99/4 Console. This assumption means that we will not have to consider either a READ ADDRESS or WRITE DATA operation in the design of the Timing and Control. We will let the Console GROM chips supply the address when the Operating System desires it. This design will function in the system even though it is both a little less and a little more than the actual GROM chip. Our design will be for the first GROM outside of the console, or for GROM Page 3 (011 in Binary).

Figure D-1 shows a basic block diagram of the simulator for this design. We will use an 8K x 8 EPROM, and throw away the upper 2K. We will be able to access this unused area in the normal manner, but the TI-99/4 Operating System does not know this space exists.

The Address Counter must be fully synchronous for the COUNT and LOAD operations. The SN74LS161A, 163A, and 169A all satisfy this requirement. The latter device must be connected to count UP. We will choose the SN74LS161A because it was listed first, and for no other reason. Notice from Figure D-1 that the inputs of the counter are organized in a two byte connection. This comprehends the byte loading of the Address Counter. We must be able to shift from the LSBY Counter into the MSBY Counter when loading, because the MSBY is loaded FIRST in the GROM Address Load protocol. The Address Counter will be either LOADED or COUNTED. The Load occurs for the LOAD ADDRESS operation, and the Count for READ DATA. A simple state counter will be responsible for the correct operation of the Address Counter.

The 16-bit Address counter is divided into two fields; the Most Significant three bits for the GROM page, and the Least Significant thirteen for addressing the EPROM. Additional GROM banks (EPROMS) may be added by using/decoding the former field.

The EPROM will be a standard 8K x 8 EPROM such as the TI 5V

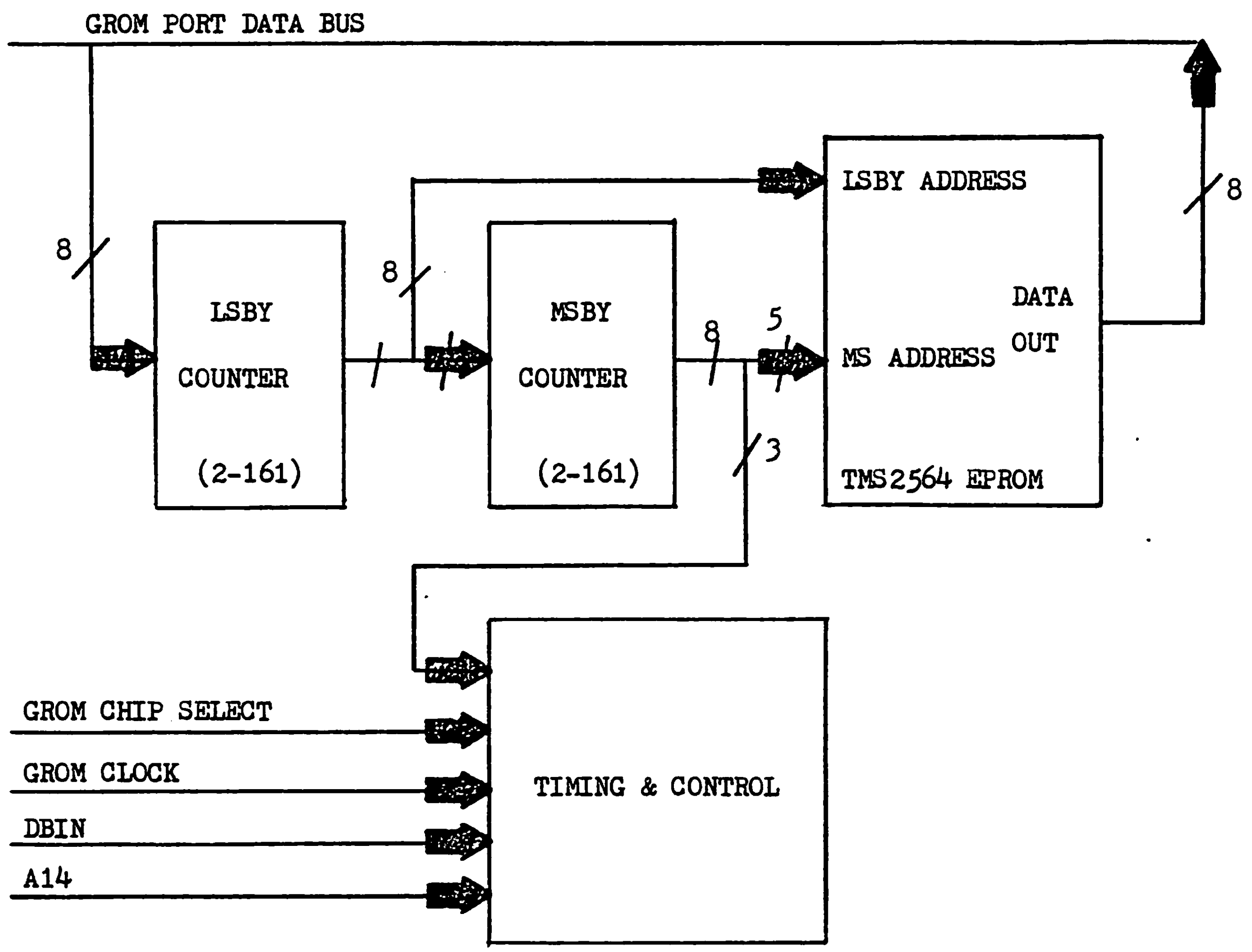


FIGURE D-1

GROM SIMULATOR BLOCK DIAGRAM

only TMS 2532. The Chip Select input will be derived by the Timing and Control section.

The Timing and Control is best implemented in Programmable Array Logic, but SSI logic will be used in the interest of availability and clarity. The design is centered around a two bit state counter clocked by the GROM Clock. Since the GROM cycle request will be from the TMS 9900 which is driven by an oscillator that is asynchronous to the GROM Clock, a synchronizing flip-flop will be required. The state counter will control both the LOAD and CLOCK inputs to the 16-bit Address Counter. The schematics for this design are shown in Figure D-3.

The conditions that cause the state counter to count from the IDLE state when GROM Chip Select is active (LOW) are: (DBIN)(A14′) for the DATA READ, and (DBIN′)(A14) for the ADDRESS LOAD to yield the logic equation DBIN(XOR)A14. This logic is ahead of the sync FF in order to provide double rail outputs for the "GO" condition.

The EPROM chip select is driven active for the logic equation:

(GROM Chip Select)(DBIN)(A14′)(AC0′)(AC01)(AC02)

This equation translates to a "GROM READ DATA" AND "GROM PAGE 3". If a larger EPROM bank is desired, one must only put a decoder on Most Significant 3-bits of the Address Counter to generate the required chip selects for the additional EPROMs. If there are to be no GROM chips in the system, the simulator will still function if a GROM chip is included for the READ ADDRESS operation only. Care must be taken for this case not to allow GROM Data to be read, though. Of course, more interesting Timing and Control and Data Bus Multiplexing could be added to allow an Address Read from our circuit.

Timing diagrams for the operation of the Simulator are in Figure D-2. Notice that the Counter is loaded early in the Address Load Cycle, and that the Address Counter is incremented late in the Data Read cycle.

The method shown here for simulating a GROM chip is simply one of many that may be designed. This one works, and will function correctly from machine to machine.

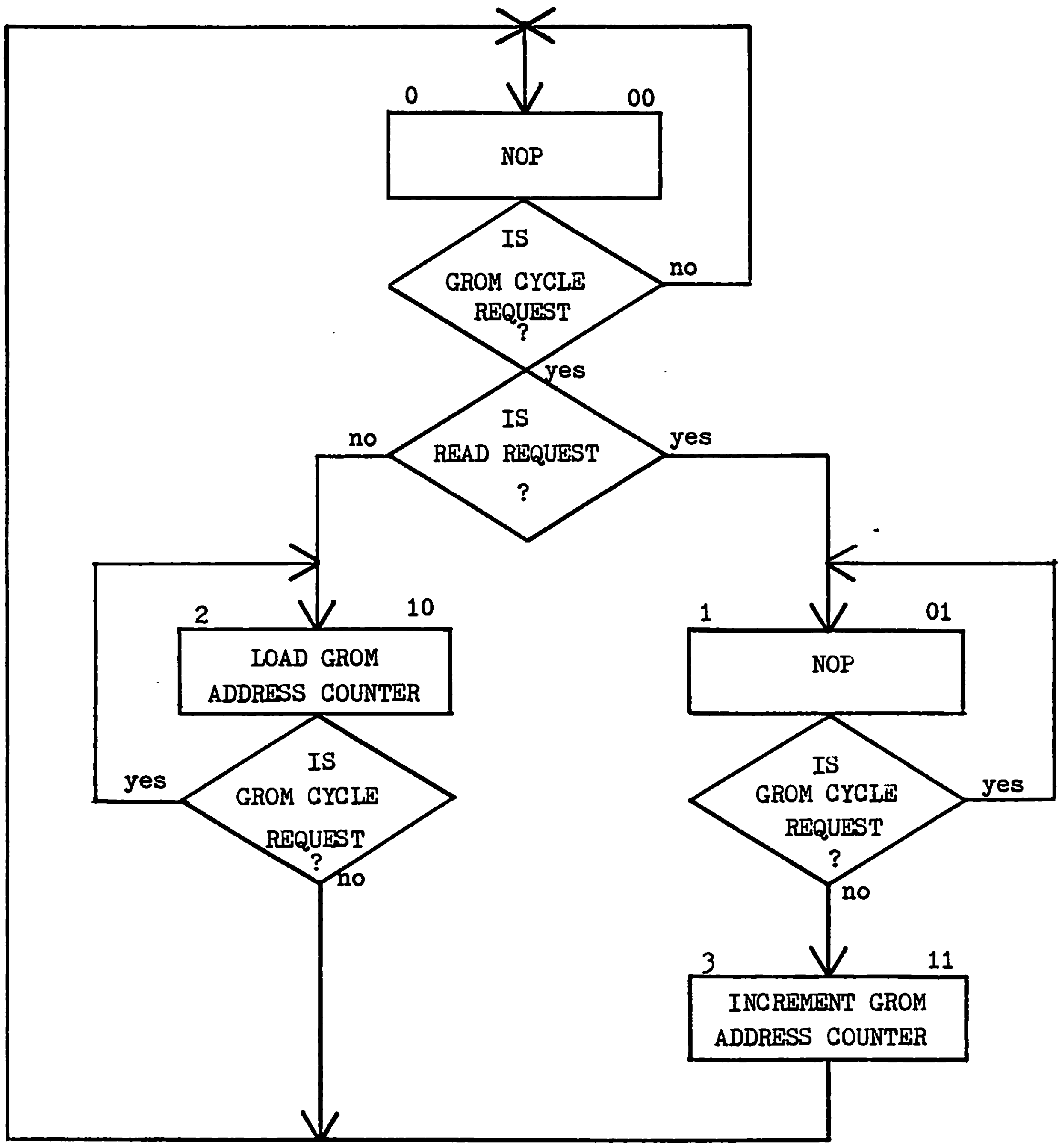
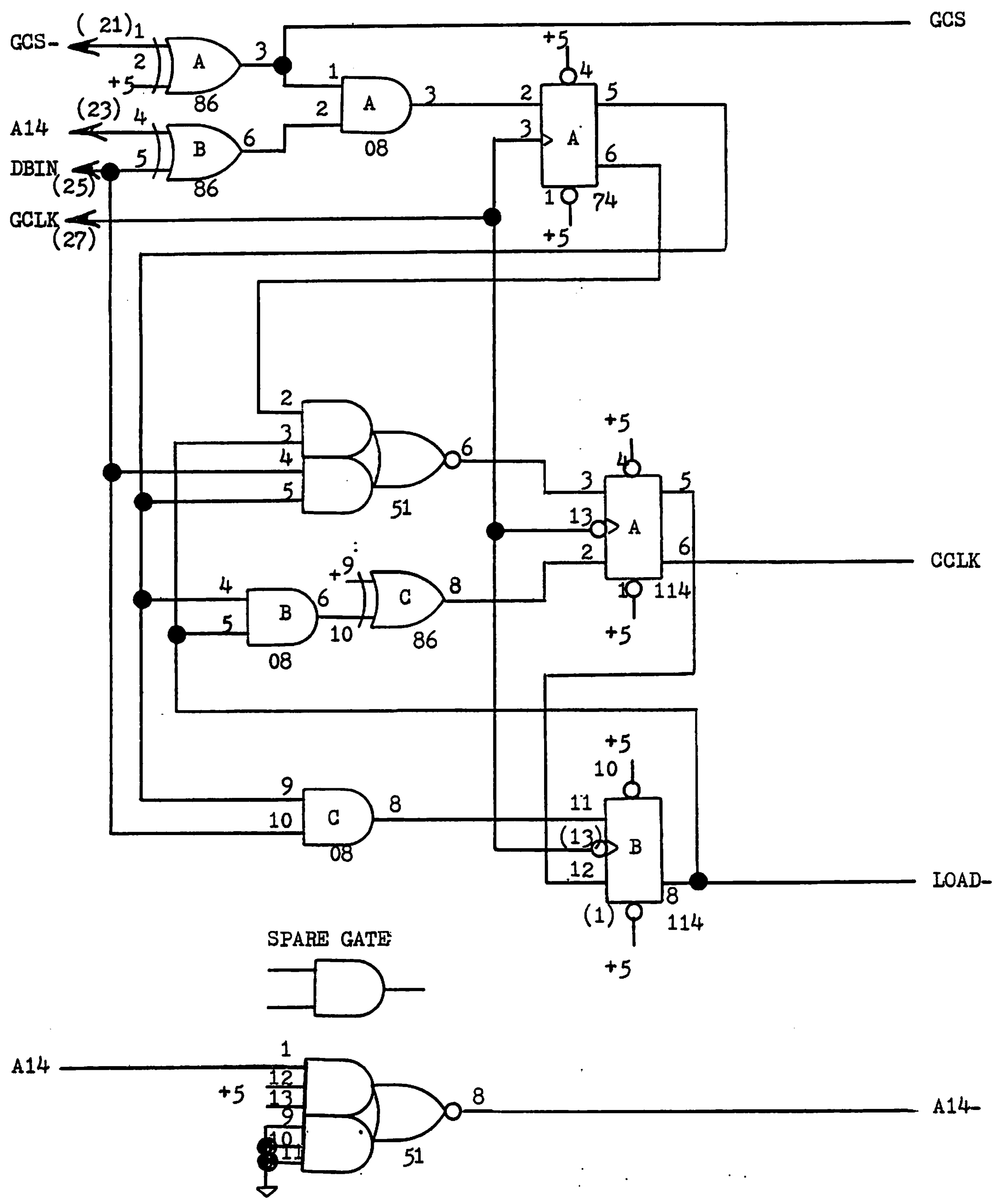


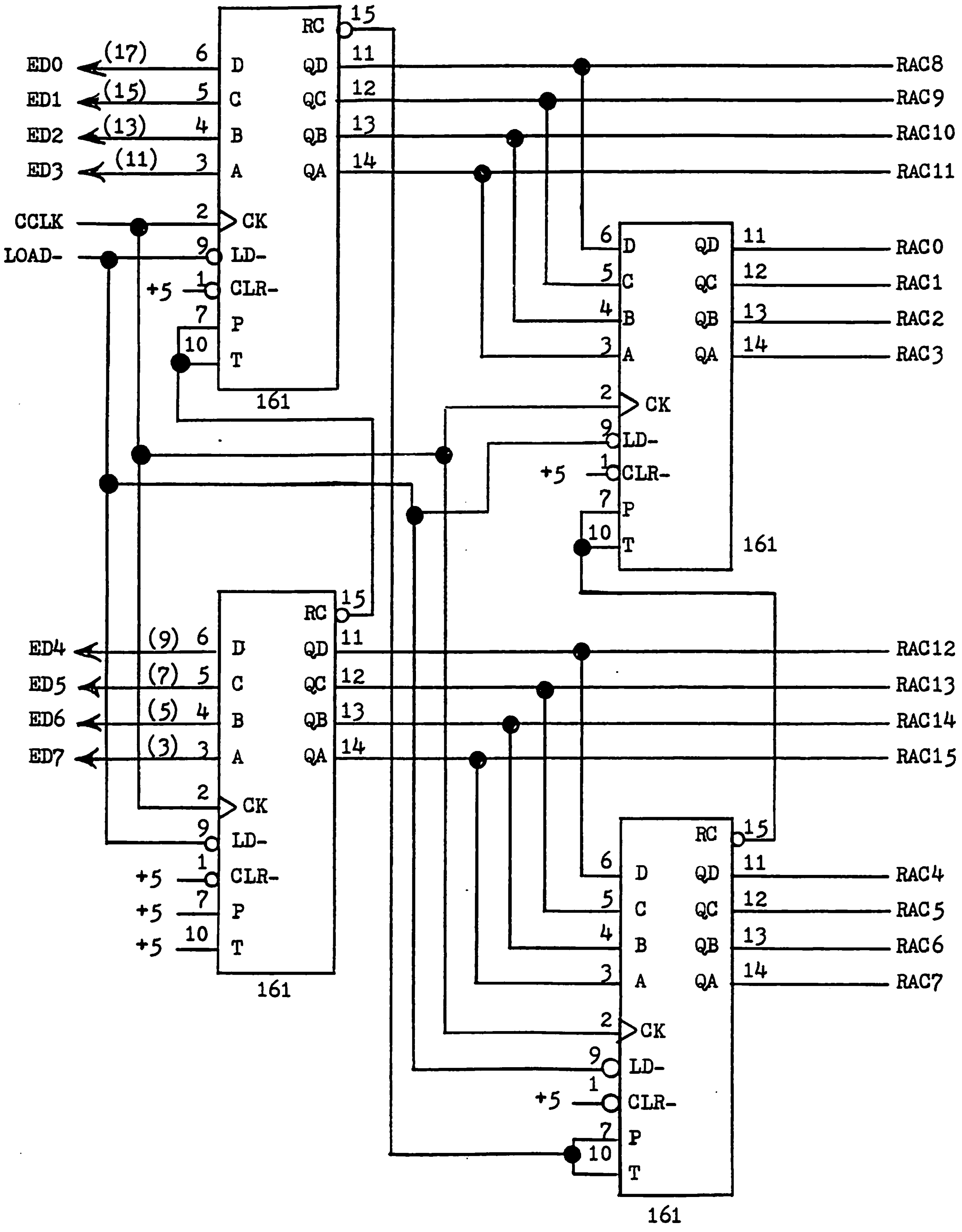
FIGURE D-2
GROM SIMULATOR STATE COUNTER FLOW CHART



TIMING & CONTROL

FIGURE D-3, CONTINUED

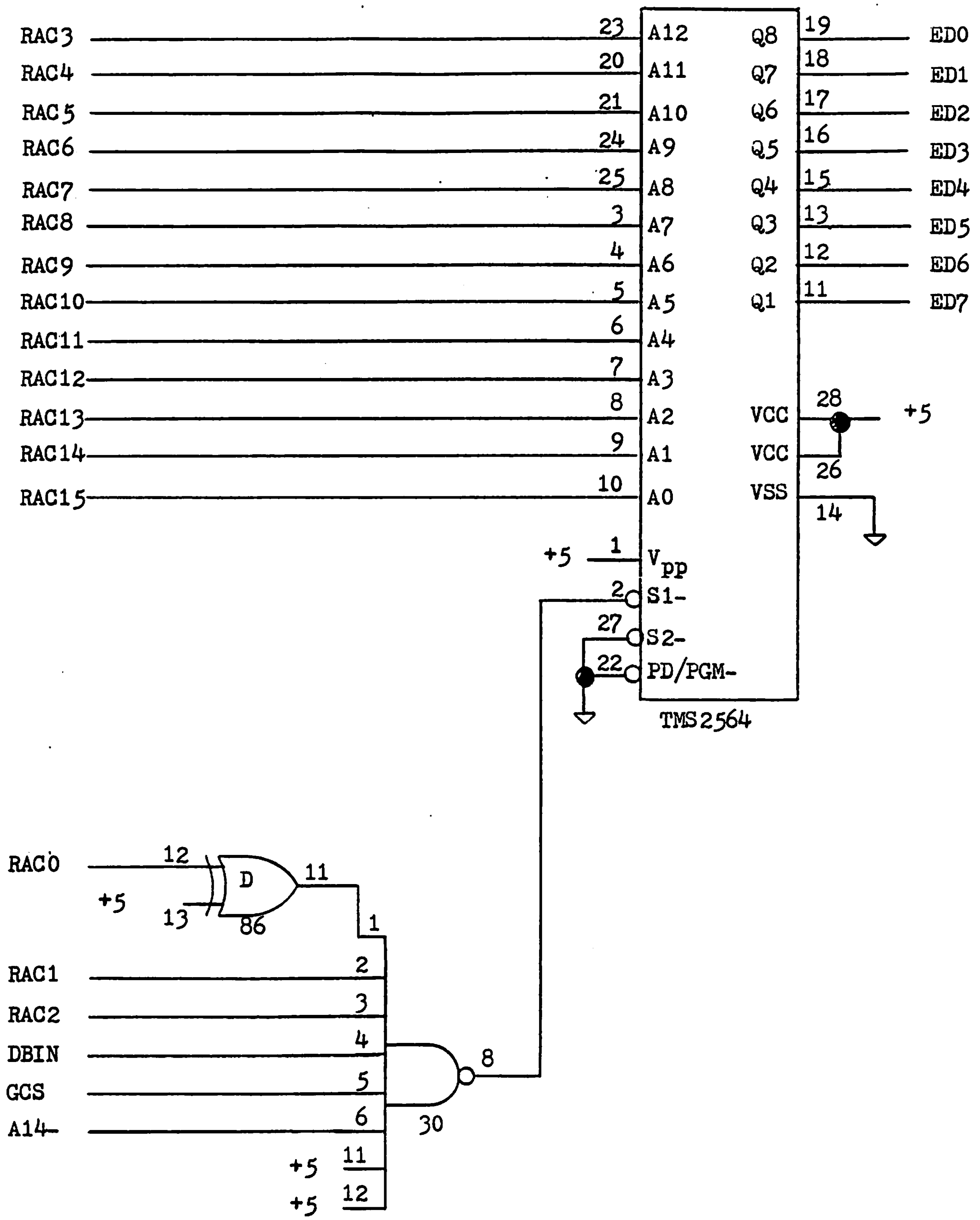
GROM SIMULATOR SCHEMATICS



ADDRESS COUNTER

FIGURE D-3, CONTINUED

GROM SIMULATOR SCHEMATICS



EPROM, CS- DECODE

FIGURE D-3, CONTINUED

GROM SIMULATOR SCHEMATICS

APPENDIX E

GROM CHARACTERIZATION PROGRAM

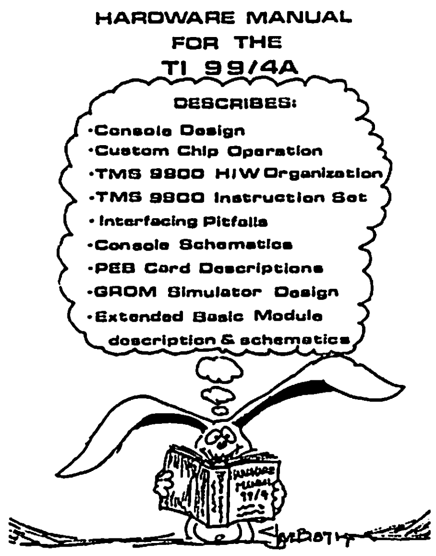
The following program was used to verify the operation of GROM chips.

```
    HEX
BASE ADDRESS    HEX
 DISPLACEMENT    OP CODE                  COMMENTS
 ------------    -------                  --------

      0          D060     MOVB @>9800, R1     DUMMY READ
      2          9800
      4          0201     LI R1,>2175         SET GROM ADDRESS
      6          2175
      8          D801     MOVB R1,@>9C02      SEND MSBY
      A          9C02
      C          06C1     SWPB R1             SET UP LSBY
      E          D801     MOVB R1,@9C02       SEND LSBY
     10          9C02
     12          06C1     SWPB R1             RESTORE ADDRESS
     14          0202     LI R2,>2000         SET UP SAVE ADDR
     16          2000
     18          DCA0     MOVB @>9800,*R2+    GET DATA BYTE
     1A          9800
     1C          DCA0     MOVB @>9802,*R2+    GET ADDR MSBY
     1E          9802
     20          DCAO     MOVB @>9802,*R2+    GET ADDR LSBY
     22          9802
     24          DCA0     MOVB @>9802,*R2+    GET ADDR ????
     26          9802
     28          DCA0     MOVB @>9802,*R2+    GET ADDR ????
     2A          9802
     2C          0420     BLWP @>0000         COLD START
     2E          0000
```

READ RESULTS

```
>2000 <READ DATA><MSBY ADDR>
>2002 <LSBY ADDR><LSBY ADDR>
>2004 <LSBY ADDR>< GARBAGE >
```



THE BUNYARD GROUP
AMS
P.O. BOX 53171
LUBBOCK, TX 79453

PLEASE SEND ME _____ COPIES OF THE TI-99/4A HARDWARE MANUAL AT $19.95 EACH ($24.00 U.S. FUNDS, FOR OUTSIDE USA ORDERS). SEND TO THE FOLLOWING NAME AND ADDRESS.

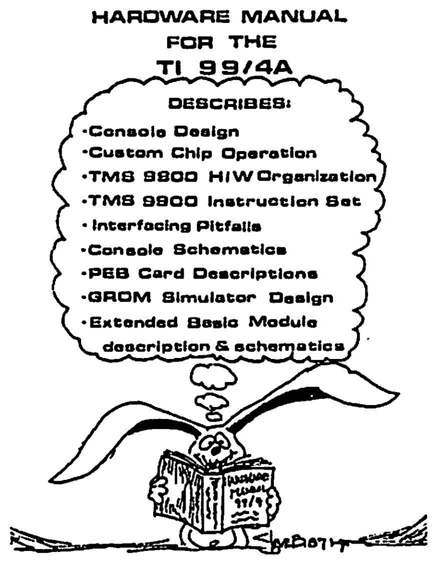
NAME: ______________________________

ADDRESS: ______________________________

CITY, STATE, ZIP ______________________

MAKE CHECKS OR MONEY ORDERS PAYABLE TO

THE BUNYARD GROUP.



THE BUNYARD GROUP
AMS
P.O. BOX 53171
LUBBOCK, TX 79453

PLEASE SEND ME _____ COPIES OF THE TI-99/4A HARDWARE MANUAL AT $19.95 EACH ($24.00 U.S. FUNDS, FOR OUTSIDE USA ORDERS). SEND TO THE FOLLOWING NAME AND ADDRESS.

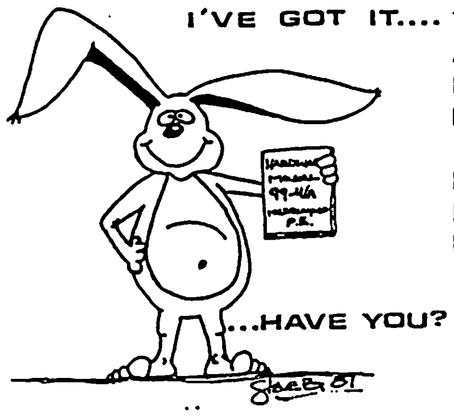
NAME: ______________________________

ADDRESS: ______________________________

CITY, STATE, ZIP ______________________

MAKE CHECKS OR MONEY ORDERS PAYABLE TO

THE BUNYARD GROUP.



HARDWARE MANUAL
FOR THE
TI 99 /4A

THE BUNYARD GROUP
AMS
P.O. BOX 53171
LUBBOCK, TX 79453

PLEASE SEND ME _____ COPIES OF THE TI-99/4A HARDWARE MANUAL AT $19.95 EACH ($24.00 U.S. FUNDS, FOR OUTSIDE USA ORDERS). SEND TO THE FOLLOWING NAME AND ADDRESS.

NAME: ______________________________

ADDRESS: ______________________________

CITY, STATE, ZIP ______________________

MAKE CHECKS OR MONEY ORDERS PAYABLE TO

THE BUNYARD GROUP.