Owner's Manual

Model 2116

16K Static RAM Module

California Computer Systems
TABLE OF CONTENTS

FEATURES ........................................................................ ii

CHAPTER 1 SETTING THE 2116 JUMPERS
  1.1 SETTING THE MEMORY GROUP ADDRESSES .......... 1-1
  1.2 SETTING THE BANK BYTE .................................. 1-2
  1.3 SETTING THE BANK PORT ADDRESS .................. 1-2
  1.4 SETTING MEMORY GROUP BANK-INDEPENDENCE ... 1-3
  1.5 SETTING THE BANK RESET JUMPER .................... 1-3
  1.6 SETTING THE PHANTOM JUMPER ......................... 1-3
  1.7 SETTING THE WAIT JUMPER ................................. 1-4
  1.8 EXAMPLES OF JUMPER SELECTION ...................... 1-5

CHAPTER 2 TESTING AND TROUBLESHOOTING THE 2116
  2.1 FRONT PANEL QUICK CHECKOUT ......................... 2-1
  2.2 DIAGNOSTIC TEST OVERVIEW ............................ 2-2
  2.3 PREPARING DRIVER ROUTINES ............................. 2-3
  2.4 SETTING UP FOR THE TEST ................................. 2-4
  2.5 LOADING THE DIAGNOSTIC ................................. 2-4
  2.6 RUNNING THE DIAGNOSTIC ................................. 2-5
  2.7 ERROR PRINTOUT INTERPRETATION ...................... 2-7
  2.8 SAMPLE MEMORY DIAGNOSTIC RUN ...................... 2-9
  2.9 MEMORY DIAGNOSTIC LISTING ............................ 2-10

CHAPTER 3 THEORY OF OPERATION
  3.1 MEMORY ......................................................... 3-1
  3.2 MEMORY ADDRESSING ....................................... 3-1
  3.3 BANK SELECTION ............................................. 3-3
  3.4 BANK-INDEPENDENCE ....................................... 3-4
  3.5 DATA BUFFERS ................................................ 3-4
  3.6 WAIT STATES .................................................. 3-5
  3.7 RESET .......................................................... 3-5

CHAPTER 4 TECHNICAL INFORMATION
  4.1 SCHEMATIC/LOGIC DIAGRAM .............................. 4-2
  4.2 ASSEMBLY COMPONENT LAYOUT ......................... 4-3
  4.3 PARTS LIST .................................................... 4-4
  4.4 ADDRESS/CHIP TABLE ...................................... 4-6

APPENDIX A LIMITED WARRANTY
FEATURES

Uses Popular 2114 Static RAMs
Available with 200, 300, or 450 nsec RAMs
Berg Jumpers Used for Selectable Features
4K Memory Blocks Individually Addressable to Any 4K Boundary
Bank Selection by Bank Port and Bank Byte
4K Blocks Individually Bank-Enabled
LEDs Indicate Board Active and Bank Active States
Wait State Jumper
Phantom Line Capability
Selectable Board-Enable/Disable on Reset
Operates on +8 Volts
Fully Buffered
Meets IEEE Proposed S-100 Signal Standards
Diagnostic Software Included
FR-4 Epoxy PC Board Solder-Masked on Both Sides
Silk Screen of Part Numbers and Reference Designations
CHAPTER 1

SETTING THE 2116 JUMPERS

The CCS 2116 is a 16K-byte static RAM board designed for use on the S-100 bus. Thirty-two 1K x 4-bit static RAM chips are arranged in columns of two in order to provide an 8-bit byte, and the sixteen 8-bit columns are divided into 4-column memory groups A through D. Each memory group is individually addressed and bank-enabled, and up to three memory groups can be buried to reconfigure the board to 4, 8, or 12K. The bank select feature, using a bank port and bank byte, is compatible with Alpha Micro and Cromemco as well as with other systems. Board Active and Bank Active states are indicated by LEDs.

To provide optimum compatibility with a variety of systems, CCS has equipped the 2116 with selectable addressing and several optional features. Selections are hard-wired with reliable, easy-to-use Berg jumpers. The addresses for each of the 4K memory groups, the bank port address and bank byte, and the bank-dependence or -independence of each memory group are jumper-set by the user to best suit his system. Phantom, Wait, and Reset features can be jumper-enabled as desired. Each jumper-selectable feature is discussed individually below. Further explanation can be found in Chapter 3, "Theory of Operation."

1.1 SETTING THE MEMORY GROUP ADDRESSES

In order to provide maximum flexibility in the location of the 2116's memory groups within a bank, CCS has made the addresses of the four memory groups jumper-selectable. The
jumper-set address for a memory group is compared with the high-order address lines A12-A15, and if the address matches, the memory group will be selected. Set the jumpers of each group to the binary equivalent of the high-order hex digit that specifies the 4K block of addresses in which you wish to locate the group. For example, the addresses of the block between 16K and 20K are 4000h-4FFFh, so you would locate a group in that block by setting its jumpers to 0100. Remember that A15 is the high-order binary digit, so you will set the binary addresses from right to left on the board.

The memory groups are fully prioritized, with A highest and D lowest. This allows you to give two (or more) memory groups the same address. Only the highest-priority group will be selected by that address; the RAMs of the other group(s) will be buried, permanently inaccessible and occupying no memory space until the address jumpers are reset. This allows you to configure the 2116 to 4, 8, or 12K without removing RAMs.

1.2 SETTING THE BANK BYTE

The bank-byte jumpers allow you to hardware-map the 2116 memory board to whichever of the eight memory bank levels 0-7 you choose. To select a bank level, jumper-set a 1 in the bit that corresponds to the desired bank level and jumper-set all other bits to 0. For example, to select bank 3 you would set bit D3 to 1 and DO-D2 and D4-D7 to 0. Remember that on the board high-order is to the right rather than the left.

You may cause the board to be activated with more than one bank by setting the jumpers corresponding to each desired bank to 1.

1.3 SETTING THE BANK PORT ADDRESS

In order to assign the 2116 to a bank, you must output the bank byte to the bank port. Most presently-marketed S-100 products using the bank port / bank byte scheme address the bank port at 40h. We recommend that you use this bank port address unless you have a strong reason for doing otherwise. Remember that A7 is the high-order bit; thus 40h is selected by setting jumper A6 to 1 and jumpers A0-A5 and A7 to 0.
1.4 SETTING MEMORY GROUP BANK-INDEPENDENCE

Each of the memory groups can be made independent of bank selection, causing it to be enabled whenever it is addressed regardless of which bank is active. This makes it possible, in time-sharing situations, for some groups to be commonly accessible while the remaining bank-dependent groups are reserved for individual users. To make a memory group independent, set its bank-dependence jumper to ME (Memory Enable). To make it bank-dependent, set the jumper to BE (Bank Enable).

1.5 SETTING THE BANK RESET JUMPER

If the Bank Reset jumper is set to B, all 16K of memory will be enabled each time the power is turned on or the system is reset. If the Reset jumper is set to A, the bank-dependent memory groups will be enabled only when the board's bank has been selected. Bank-independent memory groups will be enabled with each reset no matter which position the Bank Reset jumper is set to.

1.6 SETTING THE PHANTOM JUMPER

Setting the Phantom jumper to ON allows a device that generates a -PHANTOM signal to overlay portions of the 2116 memory. For example, CCS peripheral control boards generate Phantom signals when certain ROM locations are addressed; these locations contain code to drive the peripherals. If an identically-addressed location exists on the 2116 board, the Phantom signal will block the output from the 2116 of the contents of that location. This allows you to access the rest of the memory locations within the 4K block that contains the overlayed portion. Without Phantom capability the 2116 would not be able to locate a memory group in that block because the 2116 and the peripheral control board would both put data on the bus when a shared location was addressed.

Setting the Phantom jumper to OFF disables the -PHANTOM line.
1.7 SETTING THE WAIT JUMPER

The Wait jumper allows you to slow down your processor every time the board is addressed. This will be necessary if your processor allows less memory access time than your RAMs require.

If you have a 2116 with 200 nsec or 300 nsec RAMs, you should not need to enable the Wait feature for use with presently-available microprocessors. If you have the 450 nsec RAMs and a processor that operates at 4mHz you could, in theory at least, need to enable Wait. You should experiment, however; in most cases the 450 nsec RAMs will work successfully with a 4mHz processor without a Wait state.

Some Z-80 CPU boards, including the CCS 2810, provide a jumper-selectable Wait feature. Enabling this feature may be preferable to enabling the 2116 Wait feature. The 2116 Wait causes a Wait state to occur in every memory cycle in which the board is addressed; the CCS CPU Wait feature causes a Wait state to occur during the M1 cycle only. Because memory access time in the M1 cycle is half a clock cycle shorter than in the other machine cycles, a Wait state in this cycle effectively increases the time allowed for memory response without unnecessarily slowing the processor in other memory cycles. If you have memory boards operating at different speeds you probably will want to enable the Wait features as necessary on the slower memories rather than enable the processor Wait. This will allow you to operate at maximum speed with the faster memories. To find out what is best for your system, check your CPU manual and, if you're not sure, experiment.
1.8 EXAMPLES OF JUMPER SELECTION

The first diagram shows jumper settings for a basic CCS system consisting of a 2810 Z-80 CPU, a 2422 disk controller, and the 2116. The bank port address must be 40h. The board is activated with bank 0 as well as on start-up and reset. Memory is located between 0 and 16K. Phantom and Wait are disabled.

In the last diagram memory groups A and B are bank-independent and located in the last 8K of memory. Groups C and D reside in bank 2 between 12K and 20K. The bank port address is 40h. Only groups A and B are enabled on start-up and reset. Phantom and Wait are enabled.
2.1 FRONT PANEL QUICK CHECKOUT

(If your computer does not have a front panel, skip this section.)

Before powering on the computer, set the 2116 jumpers as follows:

The priority feature will cause Group A to be selected. Set the Front Panel Address Switches A0-A15 to the off position (0000H). Examine that address. Set the Data Switches D1-D7 to the off position and D0 to the on position (01H). Deposit (write) into memory and compare the Data readout with the switch settings. Now switch D0 to off and D1 to on, deposit into memory again, and compare the result with the switch settings. Continue the pattern of one Data Switch on and the rest off until all data bits have been checked. If any data does not match the switch settings,
isolate the malfunction with a logic probe or voltmeter before continuing.

After Group A has been checked, power down the computer and set the jumpers of groups B, C, and D to 1h.

Group B will be selected. Examine 1000H (A12 on, the rest off), and deposit the same data bytes as was done with Group A. Isolate and correct any malfunctions as they become apparent.

To check Group C, power down the computer and set the jumpers of groups C and D to 2h.

Examine 2000H (A13 on, the rest off), and test as with Groups A and B.

Finally, to test Group D, power down and set the jumpers of group D to 3h.

Examine 3000H (A12 and A13 on, the rest off), and test as before. When all malfunctions have been corrected, proceed to the next test.

2.2 DIAGNOSTIC TEST OVERVIEW

These memory diagnostics run on 8080 or Z-80 systems and provide a practical test of the 2116 memory board. Two diagnostics are provided: a walking bit test and a burn-in test. The routines have been written so that they do not require RAM other than the system stack and the RAM under test. The routines may be executed from either RAM or ROM.
Diagnostics in general can be divided into three classes: fault detection, fault isolation, and fault correction. These routines perform the fault detection and provide sufficient data for fault isolation. After a fault is isolated, correction is a practical matter.

Errors are displayed on the console device when they are detected. Two formats are used. The first, used by the burn-in test and the first stage of the walking bit test, shows errors as follows:

\[ xx \text{ yyyy zz} \]

Each character is a hexadecimal digit; \( xx \) is the bad data, \( yyyy \) is the address where the bad data occurred, and \( zz \) is what the data should have been.

The second stage of the walking bit test logs errors as follows:

\[ wwww \ xx \text{ yyyy zz} \]

Again, each character is a hexadecimal digit; \( wwww \) is the address where the error was found, \( xx \) is the bad data, \( yyyy \) is the address where data was last written, and \( zz \) is the last written data.

These error displays provide enough information for the problem to be isolated.

### 2.3 PREPARING DRIVER ROUTINES

Except for the system-unique input/output drivers, the memory test routines are capable of standing alone. The drivers must be provided by the user. Three routines are needed:

**CONIN**: Console input. This routine reads one ASCII character from the console keyboard and sets the parity bit (bit 7) equal to 0. The character is returned in the accumulator (A register).

**CONOUT**: Console output. This routine writes one ASCII character to the console display device. The character to be output is passed to CONOUT in the C register. If the console output device is sensitive to bit 7, then the user must set/reset bit 7 to what is needed in the CONOUT routine.
2-4 TESTING AND TROUBLESHOOTING

CONST: Console status. This routine reads the console input status. If data is not available, then the accumulator is set to 0 and the status flags must match. If data is pending, then a -1 (OFFH) should be returned in the accumulator (A register). The status flags must show at least a non-zero condition on the return.

After these routines have been prepared they must be loaded into memory. To allow the diagnostics to find them, three jump instructions are located at the front of the diagnostic: 0103H for CONIN, 0106H for CONOUT, and 0109H for CONST. The user should put the addresses of his I/O routines into these locations. See lines 51, 52, and 53 in the assembly listings.

2.4 SETTING UP FOR THE TEST

When you are ready to begin the test, set the jumpers as illustrated:

At this point you are ready to put the 2116 into the computer. Make sure that no other memory will respond to addresses in the range 4000H-0BFFFH.

2.5 LOADING THE DIAGNOSTIC

No special precautions are necessary. Use your standard method to load the routines. Load the diagnostic into your system at location 0100H. The diagnostic is small enough to fit into the first 1K of memory. It was assembled
assuming a 16K block of memory would be available starting at 0000H; if less memory is available, the only change necessary is to alter the stack location. The stack is currently initialized to 3F76H; a good alternate location would be 0100H.

2.6 RUNNING THE DIAGNOSTIC

Transfer control of the computer to location 0100H. The computer will type out:

DIAGNOSTIC:

You can now select which diagnostic you want. Current options are "C" for continuous burn-in or "W" for walking bit test. Any other selection will cause ????? to be displayed, after which "DIAGNOSTIC:" will again be printed. For the initial test, type in W. The computer will respond:

DIAGNOSTIC: WALKING BIT TEST
BLOCK SIZE:

Select a small block size initially. This way the read/write circuitry can be checked out without a flood of error printouts. A block size of 2 is suggested. To terminate entry, type in a space, a comma, or a carriage return. If you type in the wrong number, continue typing in until the last four digits are correct.

The computer will now ask for

BASE ADDRESS:

Type in the desired base address. (Note: The base address must be a multiple of 1024 (0400H). For the board setup suggested, a base address of 4000H is indicated.) At this time the diagnostic will do its test. On completion it will type out

TEST DONE
DIAGNOSTIC:

It is now ready for the next test. If errors were logged, see the troubleshooting section and correct the malfunction. Rerun the diagnostic until an error-free run is achieved.

Rerun the walking bit test with a block size of 1K (400H) and a base address of 4000H. Repeat the test,
increasing the base address in 1K (4000H) increments, until base address 7COOH has been tested. This tests all memory chips.

<table>
<thead>
<tr>
<th>BASE ADDRESS</th>
<th>CHIPS TESTED</th>
<th>MEMORY GROUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>4000H</td>
<td>U18, U35</td>
<td>A</td>
</tr>
<tr>
<td>4400H</td>
<td>U19, U36</td>
<td>A</td>
</tr>
<tr>
<td>4800H</td>
<td>U20, U37</td>
<td>A</td>
</tr>
<tr>
<td>4C00H</td>
<td>U21, U38</td>
<td>A</td>
</tr>
<tr>
<td>5000H</td>
<td>U14, U31</td>
<td>B</td>
</tr>
<tr>
<td>5400H</td>
<td>U15, U32</td>
<td>B</td>
</tr>
<tr>
<td>5800H</td>
<td>U16, U33</td>
<td>B</td>
</tr>
<tr>
<td>5C00H</td>
<td>U17, U34</td>
<td>B</td>
</tr>
<tr>
<td>6000H</td>
<td>U26, U43</td>
<td>C</td>
</tr>
<tr>
<td>6400H</td>
<td>U27, U44</td>
<td>C</td>
</tr>
<tr>
<td>6800H</td>
<td>U28, U45</td>
<td>C</td>
</tr>
<tr>
<td>6C00H</td>
<td>U29, U46</td>
<td>C</td>
</tr>
<tr>
<td>7000H</td>
<td>U22, U39</td>
<td>D</td>
</tr>
<tr>
<td>7400H</td>
<td>U23, U40</td>
<td>D</td>
</tr>
<tr>
<td>7800H</td>
<td>U24, U41</td>
<td>D</td>
</tr>
<tr>
<td>7C00H</td>
<td>U25, U42</td>
<td>D</td>
</tr>
</tbody>
</table>

TABLE 2.1

If errors are logged, replace the appropriate chip(s). The above table narrows any error to two chips. If the bad data is in the upper half of the byte, replace the lower-numbered chip (physically higher on the board). If the bad data is in the lower half of the byte, replace the higher-numbered chip. For example, the following error printout indicates chip 14 bad:

5C02 84 5C02 04

After a good run for all sixteen 1K increments, run the walking bit test with a block size of 16k (4000H).

At this point, invert the memory group address jumpers and run a 16K block starting at 8000H. This tests the group-select circuitry completely. The primary chips tested here are U2-U6.
When all walking bit tests run error-free, type in C for the continuous burn-in test. Specify a block size of 4000H and the appropriate base address (8000H if you follow the above procedure). Let it run for an hour or two to shake out the weak links (infant mortality). To terminate this test type in Control C. Errors, if any, will be printed out as they occur. The total number of errors will be printed out upon completion of the test.

2.7 ERROR PRINTOUT INTERPRETATION

Errors may show up in many forms. The table on the next page matches typical symptoms with probable causes. The best way to isolate a problem (and correct it at the same time) is to pull out a suspect part and replace it with a part that you know to be good. Then rerun the diagnostic and see if the problem is still present.

If a problem persists after all suspect parts are replaced, set up a controlled test condition and troubleshoot the problem with a logic probe or a voltmeter, using the logic diagram to identify test points.
### ERROR CONDITION

<table>
<thead>
<tr>
<th>Bad data=OFFH, all groups</th>
<th>a) bank select</th>
<th>U49, U56, U59</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>b) board select</td>
<td>U6, U56-59</td>
</tr>
<tr>
<td>Random data or all 0 data, all groups</td>
<td>bad write control</td>
<td>U53-54, U57-58</td>
</tr>
<tr>
<td>OFFH data, one group only</td>
<td>a) group A select</td>
<td>U5, U6, U7, U9</td>
</tr>
<tr>
<td></td>
<td>b) group B select</td>
<td>U4, U6, U7, U9</td>
</tr>
<tr>
<td></td>
<td>c) group C select</td>
<td>U3, U6, U7, U10</td>
</tr>
<tr>
<td></td>
<td>d) group D select</td>
<td>U2, U6, U7, U10</td>
</tr>
<tr>
<td>One address line hung (printout: good data, bad address)</td>
<td>address buffers</td>
<td>U50 (A0,1,4,5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>U51 (A2,3,6,7,12,15)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>U52 (A8-11,13-14)</td>
</tr>
<tr>
<td>One data line hung a) hung 0 (good address, bad data=0)</td>
<td>grounded data line</td>
<td>U53, U54, U55</td>
</tr>
<tr>
<td></td>
<td>b) hung 1 (good address, bad data=1)</td>
<td>a) open data line</td>
</tr>
<tr>
<td></td>
<td></td>
<td>b) data line shorted to +5V</td>
</tr>
<tr>
<td>Soft errors (random addresses and data, non-repeatable)</td>
<td>a) memory chip access time</td>
<td>Try setting Wait jumper ON and rerunning tests. Treat as a hard error and replace suspect parts.</td>
</tr>
<tr>
<td></td>
<td>b) heat-sensitive parts</td>
<td></td>
</tr>
<tr>
<td>Hard memory errors</td>
<td>bad memory chip</td>
<td>See TABLE 2.1 to identify chip.</td>
</tr>
</tbody>
</table>

#### TABLE 2.2
2.8 SAMPLE MEMORY DIAGNOSTIC RUN

DIAGNOSTIC: WALKING BIT TEST Typed in W
BLOCK SIZE: 30 Block may be any size
BASE ADDRESS: 300
BAD BASE ADDRESS:
BASE ADDRESS: 400
TEST DONE

DIAGNOSTIC: WALKING BIT TEST New test
BLOCK SIZE: 400
BASE ADDRESS: 400 Equal block size, base address
TEST DONE

DIAGNOSTIC: WALKING BIT TEST Larger block size test
BLOCK SIZE: 1000
BASE ADDRESS: 400
TEST DONE

DIAGNOSTIC: WALKING BIT TEST
BLOCK SIZE: 1800
BASE ADDRESS: 400
TEST DONE

DIAGNOSTIC: ???? Typed in 1
DIAGNOSTIC: WALKING BIT TEST Odd block size
BLOCK SIZE: 579
BASE ADDRESS: 400
TEST DONE

DIAGNOSTIC: CONTINUOUS BURNIN Typed in C
BLOCK SIZE: 3765 No parameter restrictions
BASE ADDRESS: 3D3
00 ERRORS
TEST DONE

DIAGNOSTIC: CONTINUOUS BURNIN
BLOCK SIZE: 3ABC
BASE ADDRESS: 3EF
00 ERRORS
TEST DONE

DIAGNOSTIC:
These routines are a highly-matured, well-thought-out set based on Intel's monitor. They provide a significant capability to converse with an 8080, 8085, or Z-80 based microprocessor system. The only registers altered are the accumulator and the pass register carrying active parameters upon entry to a routine. The stack is used extensively; sufficient space must be provided by the calling programs. The stack pointer is returned to its original place on exit unless an error was detected (SP=?) or parameters are returned on the stack. In the latter case, the stack is offset by 2 times the requested number of parameters and will be set right after these parameters are popped off the stack.

Register use conforms to ICOM and CP/M defined conventions: Output data is passed in the C register and input data is expected in the A register. These routines require CP/M-compatible CONIN and CONOUT routines as contained in the user's BIOS program, or CI and CO as in the ICOM Resident ROM.

TITLE '2114 MEMORY DIAGNOSTIC VER 1.1'

; Console input/output support routines

; These routines are a highly-matured, well-thought-out set based on Intel's monitor. They provide a significant capability to converse with an 8080, 8085, or Z-80 based microprocessor system. The only registers altered are the accumulator and the pass register carrying active parameters upon entry to a routine. The stack is used extensively; sufficient space must be provided by the calling programs. The stack pointer is returned to its original place on exit unless an error was detected (SP=?) or parameters are returned on the stack. In the latter case, the stack is offset by 2 times the requested number of parameters and will be set right after these parameters are popped off the stack.

; Register use conforms to ICOM and CP/M defined conventions: Output data is passed in the C register and input data is expected in the A register. These routines require CP/M-compatible CONIN and CONOUT routines as contained in the user's BIOS program, or CI and CO as in the ICOM Resident ROM.

; System linkages

; System linkages

1 0000 0003 CONIN EQU 0C003H
2 0100 C006 CONOUT EQU 0C006H
3 0100 C373 CONST EQU 0C373H
4 0100 C000 USER EQU 0C000H
5 0100 C38F03 JMP INIT
6 0103 C303C0 CONI: JMP CONIN
7 0106 C306C0 CONO: JMP CONOUT
8 0109 C373C3 CST: JMP CONST
9 010C C300C0 ERR: JMP USER
10 010F
Routine BLK prints one blank on the current console device.

Entry parameters: None

Stack usage: 4 bytes

Routine CONV converts a 4 bit binary number to its ASCII equivalent. The high-order 4 accumulator bits are lost.

Entry parameter: 4 bit binary number in lower half of accumulator

Exit parameter: ASCII character in (A)

Stack usage: 0 bytes

Routine CRLF prints an ASCII carriage return and line feed (in that order) on the console. It follows these with 4 blanks to create a left margin.

Entry parameter: None

Exit parameter: None

Stack Usage: 8 bytes

Routine DEPRT prints the contents of the (DE) register pair as a 4-digit hexadecimal number on the console.

Entry parameter: (DE) = 4 digit hex number

Exit parameter: None

Stack usage: 10 bytes

Alternate entry point if no CR, LF wanted
DEPRA:  MOV  A,D  ; Get high order byte
CALL  HEX2  ; Print 2 numbers
MOV  A,E  ; Get low order byte
; Alternate entry point to print (A) as two hex digits

HEX2:  PUSH  PSW  ; Save low order byte
RRC  ; Move high order nibble
RRC  ; to lower half of (A)
RRC
RRC
CALL  HEX1  ; Print the nibble
POP  PSW  ; Get low nibble back
; Alternate entry point to print low order nibble on console

HEX1:  CALL  CONV  ; Convert to ASCII
JMP  ECH1  ; Go print it

; Routine ECHO reads one character from the calling routine and then echoes it back. It is assumed that the console is in a full duplex mode.

ECHO:  CALL  CONI  ; Read a character
ECH1:  PUSH  B  ; Save (BC)
ANI  7FH  ; Strip off parity bit
MOV  C,A  ; Put character into (C)
POP  B  ; Restore (BC)
RET

; Alternate entry point for BLK routine

ECH2:  CALL  CONO  ; Output it

; Routine HLPRT prints the contents of the (HL) register as 4 hexadecimal digits on the console.

HLPRT:  CALL  CRLF  ; Print a (CR,LF)
HLPRA:  XCHG  ; Swap (HL), (DE)
CALL  DEPRA  ; Go print (DE)
XCHG  ; Unswap (HL), (DE)
RET

; Routine PCHK reads a character from the console and checks whether it is a valid delimiter (space, comma, or carriage return). If so, a zero is returned in the status flags. If the character is
a carriage return, the carry bit is set also. If it is not a delimiter, a non-zero, no-carry indication is required.

Routine PRM reads characters from the console and pushes them onto the stack. Multiple parameters may be read: values are delimited by a space or comma. If a carriage return is entered, PRM stops reading values and returns to the caller. Only the last 4 characters of a string are saved; to correct an error, type until the last four characters are correct. The caller may retrieve the values by popping them from the stack, last-entered character first.

Entry parameter: (C) = number of expected parameters

Exit parameters: (C) Parameters on stack:

If a bad value was entered, '?' is printed and control transferred to a user provided error handler. The stack pointer value is indeterminate and needs to be reset.

Stack usage: 4 + 2 = (C) bytes

Alternate entry point if only one parameter is desired.

Normal entry point

Set (HL) = 0

Get a character

Save input character

Check it and CVB

Not hex, see if delim

Multiply (HL) by 16
ROUTINE PRTWD PRINTS A CHARACTER STRING ON THE CONSOLE. DEPENDING ON THE ENTRY POINT, A CR AND LF MAY BE PRINTED FIRST. THREE FORMS OF MESSAGE-END DELIMITERS ARE ACCEPTED: BIT 7=1 IN LAST CHARACTER TO BE OUTPUT; ASCII ETX (CNTRL C) FOLLOWING THE LAST CHARACTER; OR A USER-SPECIFIED DELIMITER FOLLOWING THE LAST CHARACTER. IF THE LAST OPTION IS USED, (B) MUST HAVE THE DELIMITER ON ENTRY TO PHTA.

ROUTINE NIBBL STRIPS THE ASCII ZONE OFF A CHARACTER IN THE (A) REGISTER AND VERIFIES THAT IT IS A VALID HEX DIGIT. IF SO, THE BINARY VALUE IS RETURNED TO THE LOWER HALF OF THE A REGISTER; THE UPPER HALF IS SET TO ZERO. IF NOT, THE CARRY FLAG IS SET AND CONTROL RETURNED TO THE CALLER.

ENTRY PARAMETER: (A) = ASCII CHAR
EXIT PARAMETERS: See description above
STACK USAGE: None
TESTING AND TROUBLESHOOTING 2-15

When a bad memory cell is detected, this routine
subroutines for the memory diagnostics

Hardware diagnostics can be divided into 3 stages:
1) fault detection
2) fault isolation
3) fault correction

These routines automate the first stage only. See
the user's manual for guidelines for the second
stage. After the second step is completed, fault
correction should be no trouble.

When a bad memory cell is detected, this routine
2-16 TESTING AND TROUBLESHOOTING

Routine PARM reads in the desired test block size and block base address. Both parameters are pushed onto the stack.

is called to print the bad address, bad data, test address, and test data (in that order). With this error log, the fault isolation process can be conducted.

is called to print the bad address, bad data, test address, and test data (in that order). With this error log, the fault isolation process can be conducted.

Routine BREAK tests the console status to see if a character has been typed in. If so, it checks to see if it is an ASCII ETX (CNTRL C). If so, it types an "ABORT" message and returns control to the calling routine.

Alternate entry point when bad address is meaningless.

Routine BREAK tests the console status to see if a character has been typed in. If so, it checks to see if it is an ASCII ETX (CNTRL C). If so, it types an "ABORT" message and returns control to the calling routine.

; Routine PARM reads in the desired test block size and block base address. Both parameters are pushed onto the stack.

; Alternate entry point when bad address is meaningless.

; Alternate entry point when bad address is meaningless.

; Alternate entry point when bad address is meaningless.

; Alternate entry point when bad address is meaningless.

; Alternate entry point when bad address is meaningless.

; Alternate entry point when bad address is meaningless.

; Alternate entry point when bad address is meaningless.

; Alternate entry point when bad address is meaningless.

; Alternate entry point when bad address is meaningless.
Without errors, this diagnostic tests a 1K cell in approximately 2 seconds.

The base address, when asked for, must be on a 1K boundary or it will be rejected and another address asked for.

The operator can abort the test at any time by typing ETX (CTRL C) should too many errors be detected. Allowing the test to complete will ensure adequate data for thorough fault isolation.

Routine MADT performs a "Walking Bit" test on both the data and address lines of a 2114 pair at the same time. First, it zeros all cells in the specified block, then ensures that they are all zero. It tests each 1K section separately. Detected errors are logged on the console as they occur.

The operator can abort the test at any time by typing ETX (CTRL C) should too many errors be detected. Allowing the test to complete will ensure adequate data for thorough fault isolation.

Without errors, this diagnostic tests a 1K cell in approximately 2 seconds.

```
385 021B 213002  PARMA: LXI H,BAMSG ; Print BASE ADDRESS
386 021E CDAB01  CALL PRTWD ; message
387 0221 C36601  JMP PARM1 ; Get it and return
388 0224  ;
389 0224 424C4F43 BZMSG: DB 'BLOCK SIZE:'', '80H
0228 4B205349  022C 5A453A00
390 0230 42415345 BAMSG: DB 'BASE'
391 0234 20414444 ADMSG: DB ' ADDRESS:'', '80H
0238 52455353 023C 3AA0
392 023E  ; Routine MADT performs a "Walking Bit" test on both
393 023E  ; the data and address lines of a 2114 pair at the
394 023E  ; same time. First, it zeros all cells in the
395 023E  ; specified block, then ensures that they are all
396 023E  ; zero. It tests each 1K section separately.
397 023E  ; Detected errors are logged on the console as they
398 023E  ; occur.
399 023E  ; The base address, when asked for, must be on a 1K
400 023E  ; boundary or it will be rejected and another
401 023E  ; address asked for.
402 023E  ; The operator can abort the test at any time by
403 023E  ; typing ETX (CTRL C) should too many errors be
404 023E  ; detected. Allowing the test to complete will
405 023E  ; ensure adequate data for thorough fault isolation.
406 023E  ; Without errors, this diagnostic tests a 1K cell in
407 023E  ; approximately 2 seconds.
408 023E  ;
413 023E 217F02 MADT: LXI H,WBMSG ; Sign on
414 0241 CD0C02 CALL PARM ; Get parameters
415 0244 E1 MADTA: POP H ; Retrieve BASE ADDRESS
416 0245 D1 POP D ; Retrieve BLOCK SIZE
417 0246 7C MOV A,H ; Test for 1K boundary
418 0247 E603 ANI 3
419 0249 B5 ORA L
420 024A CA6002 JZ MADTB ; OK, jump
421 024D D5 PUSH D ; Save block size
422 024E 217B02 LXI H,BEMSG ; Reject base address
423 0251 CDAB01 CALL PRTWD
424 0254 213002 LXI H,BAMSG
425 0257 CDAE01 CALL PRTWA
426 025A CD1B02 CALL PARM ; Ask for another
427 025D C34402 JMP MADTA ; Test it again
428 0260  ;
429 0260 CD9902 MADTB: CALL ZTBK ; Zero the block
430 0263 D5 MADTC: PUSH D ; Save block size
431 0264 3E04 MVI A,4 ; Set 1K sections
432 0266 BA CMP D ; See if < 1K
433 0267 F26B02 JP MADTD ; Yes, test it
434 026A 57 MOV D,A ; No, set to 1K
435 026B CDBB02 MADTD: CALL WLKAD ; Test it
```
Testing and troubleshooting

Routine ZTBK zeros and tests for a contiguous block of memory. On entry, the (DE) register must have the block size and the (HL) register must have the base address. These values are restored to the registers on exit from the routine.

; Routine ZTBK zeros and tests for a contiguous block of memory. On entry, the (DE) register must have the block size and the (HL) register must have the base address. These values are restored to the registers on exit from the routine.

436 026E E1  POP  H  ; Get remaining size
437 026F 7D  MOV  A,L  ; Subtract tested size
438 0270 93  SUB  E
439 0271 6F  MOV  L,A
440 0272 7C  MOV  A,H
441 0273 9A  SBB  D
442 0274 67  MOV  H,A
443 0275 C8  RZ  ; Return if done
444 0276 EB  XCHG  ; (DE) = untested
445 0277 7D  XCHG  ; (HL) = previous increment
446 0277 09  DAD  B  ; Set new base address
447 0278 C36302  JMP  MADTC  ; Do it again
448 0278
449 027B 424144A0  BEMSG:  DB  'BAD', '80H
450 027F 57414C4B  WBMSG:  DB  'WALKING BIT TEST', '80H
451 0283 494E4720  TDMSG:  DB  'TEST DON', 'E'80H
452 0287 42495420
453 028B 54455354
454 028F A0
455 028F 54455354
456 0290 20444F4E 0294 20444F4E
457 0298 C5
458 0298
459 0299  ZTBK:  PUSH  D  ; Save block size
460 029A E5  PUSH  H  ; Save base address
461 029B 0E00  MOV  M,O
462 029D 71  ZTBKA:  MOV  A,E
463 029E 23  INX  H  ; Next address
464 029F 1B  DCX  D  ; Loop control
465 02A0 7B  MOV  A,E
466 02A1 B2  ORA  D
467 02A2 C9D02  JNZ  ZTBKA  ; Loop if not zeroed
468 02A5 E1  POP  H  ; Restore registers
469 02A6 D1  POP  D
470 02A7 D5  PUSH  D  ; Save parameters
471 02A8 E5  PUSH  H
472 02A9 7E  ZTBKB:  MOV  A,M  ; Read a cell
473 02AA B9  CMP  C  ; Same as written?
474 02AB C4DB01  CNZ  ADPRA  ; Log error if necessary
475 02AE CDF301  CALL  BREAK  ; See if abort wanted
476 02B1 23  INX  H  ; Next address
477 02B2 1B  DCX  D  ; Loop control
478 02B3 7B  MOV  A,E
479 02B4 82  ORA  D
480 02B5 C2A902  JNZ  ZTBKB  ; Loop if more to do
481 02B8 E1  POP  H  ; Restore base address
482 02B9 D1  POP  D  ; Restore block size
483 02BA C9  RET
484 02BB
Routine WLKAD walks a single high bit through each data bit of all addresses in a controlled manner. After a bit is written, all other locations are tested for zeros. When an error is detected, it is logged as described above. If excess errors occur, abort the test by typing CNTRL C.
Testing and Troubleshooting

Routine BRNIN continuously writes a sequence of non-zero numbers into a specified memory block and reads them back for comparison. If errors occur, they are logged on the console. A running error total is also maintained. The test may be terminated at any time with a CNTRL C; the error total at this time will be displayed on the console. The test data steps from 1 to 255 decimal, then repeats itself, always skipping 0.

BRNIN: LXI H, CBMSG ; Get message address
CALL PARM ; Write it, get parameters

BRNA: PUSH B ; Save data, error count
PUSH D ; Save block size
PUSH H ; Save base address
POP D ; Get block size
MVI C, 1 ; Seed the data
MVI B, 0 ; Initialize error count

BRNB: MOV M, C ; Write the data byte
INR C ; Advance data pattern
INR C ; Set to 1
Routines INIT and EXEC initialize the computer and monitor the console for a command. When a valid command is received, control is transferred to the appropriate routine.

```
595 0332 23  BRNC: INX  H  ; Go to next address
596 0333 1B  DCX  D  ; Do loop control
597 0334 7B  MOV  A,E
598 0335 B2  ORA  D
599 0336 C22C03  JNZ  BRNB
600 0339 E1  POP  H  ; Get base address
601 033A D1  POP  D  ; Get block size
602 033B C1  POP  B  ; Get data seed, error count
603 033C D5  PUSH  D  ; Restore them
604 033D E5  PUSH  H
605 033E 7E  BRND: MOV  A,M  ; Read data byte
606 033F B9  CMP  C  ; Check it
607 0340 CA4703  JZ  BRNE  ; Skip if OK
608 0343 04  INR  B  ; Error count
609 0344 CDBD01  CALL  ADPRA  ; Log the error
610 0347 0C  BRNE: INR  C  ; Change test data
611 0348 C24C03  JNZ  BRNF  ; Skip if not zero
612 034B 0C  INR  C  ; Reset to 1
613 034C 23  BRNF: INX  H  ; Next address
614 034D 1B  DCX  D  ; Loop control
615 034E 7B  MOV  A,E
616 034F B2  ORA  D
617 0350 C23E03  JNZ  BRND
618 0353 E1  POP  H  ; Reset base address
619 0354 D1  POP  D  ; and block size
620 0355 CD0901  CALL  CST  ; Time to quit
621 0358 CA2903  JZ  BRNA  ; No, do it again
622 035B CD0301  CALL  CONI  ; Get character
623 035E FE03  CPI  'C'-CNTL
624 0360  ; ETX (Cntl C)?
625 0360 C22903  JNZ  BRNA  ; No, continue
626 0363 CD1E01  CALL  CRLF
627 0366 78  MOV  A,B  ; Error count
628 0367 CD3301  CALL  HEX2  ; Print it
629 036A 217003  LXI  H,ERMSG  ; Get error message address
630 036D C3AE01  JMP  PRTWA  ; Print it and return to EXEC
631 0370  ;
632 0370 20455252  ERMSG: DB  'ERROR','S'+80H
633 0374 4F52D3  CBMSG: DB  'CONTINUOUS BURNIN',' '+80H
634 0377 434F4E54
635 037B 494E554F
636 037F 55532042
637 0383 55524E49
638 0387 4EA0
639 0389  ;
640 0389 219002  RETN: LXI  H,TDMSG  ; Print 'TEST DONE'
641 038C CDAB01  CALL  PRTWD
642 038F 314000  INIT: LXI  SP,STACK  ; Set stack pointer
643 0392 21AC03  EXEC: LXI  H,DIMSG  ; Print diag message
644 0395 CDAB01  CALL  PRTWD
```
LXI H, RETN ; Set up return address
PUSH H
CALL CONI ; Wait for command
CPI 'C' ; Continuous burn-in
JZ BRNIN
CPI 'W' ; Walking bit
JZ MADT
JMP QPRT

; DIMSG: DB 'DIAGNOSTIC:', '+80H
03B0 4E4F5354
03B4 49433AA0

; END

TOTAL ERRORS=00
CHAPTER 3
THEORY OF OPERATION

This chapter is intended for those users who want a more thorough understanding of the 2116 operation than they need to make the 2116 function in their systems. Used in conjunction with the logic diagram in Chapter 4, it should provide a sound understanding of the design and features of the board. Additional information, if desired, can be obtained from data sheets for the individual chips.

3.1 MEMORY

The 2116 uses 2114-type RAMs, which are fully static (i.e., they require no clock or refresh signals) and provide 4096 bits of storage organized 1024 x 4. Each RAM thus requires ten address inputs and four bi-directional data lines. A Chip Select input (\(-CS\)) provides for the selection of individual chips in a memory array. To prevent erroneous data from getting into the chip a R/W input inhibits the data input buffer when high. Thus data can be written to a memory chip only when both \(-CS\) and R/W are low. The 2116 controls \(-CS\) through the address decoders; R/W goes low when either \(-pWR\) or MWRITE is active.

3.2 MEMORY ADDRESSING

Addressing a specific memory location on the 2116 involves addressing a location on each chip while enabling only one two-chip column. Address lines A0-A15 enter the board and are inverted, A0-A9 addressing one location on
3-2 THEORY OF OPERATION

each chip through a common address bus. Chip selection is handled by a pair of 3-to-8 decoders. Each decoder selects one of eight columns, depending on the conditions of inputs A, B, and C. Inputs G1, G2A, and G2B determine whether a decoder will be enabled, G2A and G2B low and G1 high enabling a decoder.

Decoder enabling is controlled by the Address Select circuitry. Address bits A12-A15 are compared with the user-selected four-bit addresses of each of the four memory groups. -A12 through -A15 are paralleled into four quad open collector exclusive-OR gates. Each gate compares -A12, -A13, -A14, or -A15 with the corresponding bit of the memory group address. The output of each exclusive-OR gate in a memory group must be high for the memory group to be selected; one low output will pull the open collector output from that group low. All of the memory groups are ORed and the output is NANDed with the MEM line (high when sINTA, sINP, and sOUT are all low) to form the -SEL line. -SEL is the G2A input of the U10 and the G2B input of the U9. Thus if no memory group on the board is addressed, both chips are disabled by -SEL high.

If -SEL is low, the ORed outputs of groups A and B and groups C and D determine which decoder is enabled. U9's G2A is permanently pulled low. If the ORed output of groups A and B is high U9 is enabled through G1 and U10 is disabled through G2B. If the ORed output of groups A and B is low and the ORed output of groups C and D (U10's G1 input) is high, U10 is enabled and U9 is disabled.

Chip selection within the enabled decoder is determined by inputs A, B, and C. U9's C input is tied to the output of group A's memory-address-comparison circuitry; if group A is addressed, C is high, and one of the columns enabled by decoder outputs 4-7 will be selected. In the same way group C's memory-address-comparison circuitry determines which group U10 will select. Address lines A10 and A11 are the A and B inputs of the decoders, determining which of the four columns in a group will be selected.

The 2116 decoding scheme provides full prioritizing of the memory groups. If either or both of groups A and B are addressed, U9 is enabled and U10 is disabled; whether group C or D is addressed is irrelevant. Group selection by the decoders is determined by whether or not group A or C has been addressed, groups B and D being irrelevant. Thus group A has the highest priority, followed in order by groups B, C, and D. If two or more memory groups are given identical addresses, only the highest priority group will be selected when that address is received. The other groups will
effectively be buried; they will be unaddressable and will occupy no memory space.

3.3 BANK SELECTION

The CCS 2116 is bank-selectable by bank port address and bank byte. Thus it is fully compatible with Cromemco, Alpha Micro, and other bank port systems. IT IS NOT COMPATIBLE WITH ADDRESS-SELECT SYSTEMS SUCH AS IMSAI.

You assign the 2116 to a bank by jumper-setting the bank port address and the bank byte. To enable a bank during operation, the processor must address the bank port through the low order byte on the address bus and put the bank byte on the data bus. When the processor is in an I/O cycle (sOUT or sINP high), the 2116 compares the low-order byte on the address bus with the user-selected bank port address. If the two match, the 2116 compares the bank byte on the data bus with the user-selected bank byte. The bank-dependent memory groups are enabled or disabled according to whether or not the two bytes designate the same bank.

The 2116 compares -A0 through -A7 with the jumper-set bank port address using an open collector set of exclusive-OR gates. A pull-up resistor holds the output high unless a wrong address pulls the output low. The bank-address-comparison line is ANDed with the I/O line, and the resulting output is NANDed with inverted -pWR to form the BANK CLK line. This line clocks a D-type positive-edge-triggered flip-flop.

The bank address and I/O lines go high first. As long as -pWR is inactive (high, inverted low) the BANK CLK line is low. When -pWR goes active (low, inverted high) the BANK CLK line goes high, clocking the flip-flop. In the meantime the bank byte is written onto the data bus. A high signal on any of the data lines indicates that the corresponding bank is being selected (data lines D00-7 corresponding to banks 0-7). The bank byte signals are inverted for comparison with the user-selected bank byte. Jumper-selecting a bank connects the corresponding data line to the BANK DATA line; a low signal on that line pulls BANK DATA low. Other jumpers may also be connected and more than one bit of the bank byte on the data bus can be high; the open-collector output will be pulled low whenever a high-inverted-low data line is jumper-connected.
3-4 THEORY OF OPERATION

When the flip-flop is clocked by -pWR going low the condition of BANK DATA, the flip-flop's D input, determines the outputs Q and -Q. Q takes the value of D and -Q is D's complement. A low on the BANK DATA line resets Q to low, lighting the Bank Select LED. A high on the BANK DATA line sets Q, and therefore -BANK ENABLE, to high. -BANK ENABLE high is inverted to disable the memory groups that are jumper-set bank-dependent (see BANK-INDEPENDENCE below).

The processor can determine whether a bank has been selected by reading DIO at the bank port address. When pDBIN is active and the bank port has been addressed, the BANK READ ENABLE line is high. This line is NANDed with -Q, which is high when the 2116's bank has been selected. A low output from the NAND pulls DIO low, acknowledging to the processor that a bank has been enabled.

The flip-flop will be reclocked the next cycle in which the bank port address is received and the I/O line is high, at which point the new bank byte will be compared and Q and -Q set or reset depending on the BANK DATA line input to D. Until then the bank-dependent memory groups will remain enabled.

3.4 BANK-INDEPENDENCE

The 2116 allows you to make any memory group independent of bank selecting by setting a jumper so that the inverted -BANK ENABLE line is not connected to the memory-address-comparison circuitry of the memory group you want to make independent. This prevents that memory group's open collector output from being pulled low when the -Bank Enable line is active. The memory group will therefore be enabled whenever it is addressed, independent of which bank has been selected.

3.5 DATA BUFFERS

The DI and DO lines from the data bus are tied together to form the bi-directional data lines for the RAM chips. DIO-7 and DOO-7 are buffered by 3-State Bus Drivers. If the drivers are in the high-impedance state, the lines they drive are disabled. DOO-7 are disabled unless either -pWR or MWRITE is active (-WR line low). If the -WR line is low the buffer allows data to be written to the RAMs.
Read-enabling is more involved. Basically, if the Phantom jumper is off DIO-7 will be enabled whenever a memory group on the board is addressed and the processor is in a memory read cycle. If the Phantom jumper is on, a low on -PHANTOM will disable DIO-7. -PHANTOM is generated by another device in the system and allows that device to overlay identically-addressed memory locations on the 2116 board by preventing 2116 data from reaching the data bus. Thus data is read from the overlaying device only.

3.6 WAIT STATES

A Wait state is necessary when a peripheral device takes more time to complete a task than the processor normally allows. Because the 2116 is available with 200, 300, or 450 nsec Rams, and because processor speeds vary, the Wait feature on the 2116 has been made jumper-selectable. If the Wait jumper is set to on, pRDY will be pulled low whenever pSYNC goes high and the board is selected (-SEL low). This causes an extra clock cycle to be added to each memory read or memory write machine cycle during which the board is selected, thereby increasing the time that signals remain on the address and data busses. If the jumper is set to off the 2116 does not pull pRDY low and a Wait state does not occur unless it originates elsewhere.

3.7 RESET

The Reset jumper allows you to choose whether or not the 2116 will be enabled when the system is powered up or reset by determining which input of the bank-enable flip-flop will be controlled by pRESET. Pull-up resistors normally hold both the Preset and Clear inputs high, which they must be for the flip-flop to operate normally. The -pRESET line can be jumper-set so that either the Preset input or the Clear input is pulled low whenever the power is turned on or the system is reset. If the Reset jumper is set to position A, -pRESET active pulls Preset low, the flip-flop is set (Q high), and the bank-dependent memory groups are disabled. If the jumper is set to position B, -pRESET active pulls the Clear input low, the flip-flop is reset (Q low), and the bank-dependent memory groups are enabled.
4.2 ASSEMBLY COMPONENT LAYOUT
### 4.3 PARTS LIST

<table>
<thead>
<tr>
<th>QTY</th>
<th>REF</th>
<th>DESCRIPTION</th>
<th>CCS PART #</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>CAPACITORS</strong></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>C1,3,5,7,8</td>
<td>Tantalum, 4.7uf, 35 vdc, 20%</td>
<td>42084-54756</td>
</tr>
<tr>
<td>4</td>
<td>C2,4,6,9</td>
<td>Ceramic, .1uf, 50 vdc, 20%</td>
<td>42142-21046</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>RESISTORS</strong></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>R1,2</td>
<td>220 ohm, 1/4 w, 5%</td>
<td>40002-02215</td>
</tr>
<tr>
<td>1</td>
<td>R3</td>
<td>2.7K ohm, 1/4w, 5%</td>
<td>40002-02725</td>
</tr>
<tr>
<td>1</td>
<td>Z1</td>
<td>Resistor Network, SIP 2.7K ohm x 7</td>
<td>40930-72726</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>INTEGRATED CIRCUITS</strong></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>U14-29,31-46</td>
<td>MOS 2114 1Kx4 Static RAMS</td>
<td>31900-21142</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or 21143 (200nsec) or 21144 (450nsec)</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>U1,13,30,47</td>
<td>7805 +5v regulator</td>
<td>32000-07805</td>
</tr>
<tr>
<td>6</td>
<td>U2-5,11,12</td>
<td>74LS136 quad ex-OR:OC</td>
<td>30000-00136</td>
</tr>
<tr>
<td>1</td>
<td>U6</td>
<td>74LS32 quad 2-in OR</td>
<td>30000-00032</td>
</tr>
<tr>
<td>2</td>
<td>U7,8</td>
<td>74LS05 hex inverter:OC</td>
<td>30000-00005</td>
</tr>
<tr>
<td>2</td>
<td>U9,10</td>
<td>74LS138 octal decoder</td>
<td>30000-00138</td>
</tr>
<tr>
<td>1</td>
<td>U48</td>
<td>75452 dual NAND: OC</td>
<td>30300-00452</td>
</tr>
<tr>
<td>1</td>
<td>U49</td>
<td>74LS74 dual D flip-flop</td>
<td>30000-00074</td>
</tr>
<tr>
<td>3</td>
<td>U50-52</td>
<td>74LS04 hex inverter</td>
<td>30000-00004</td>
</tr>
<tr>
<td>3</td>
<td>U53-55</td>
<td>74LS367 hex bus driver</td>
<td>30000-00367</td>
</tr>
<tr>
<td>QTY</td>
<td>REF</td>
<td>DESCRIPTION</td>
<td>CCS PART #</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>------------------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>1</td>
<td>U56</td>
<td>74LS10 tri 3-in NAND</td>
<td>30000-00010</td>
</tr>
<tr>
<td>2</td>
<td>U57,58</td>
<td>74LS02 quad 2-in NOR</td>
<td>30000-00002</td>
</tr>
<tr>
<td>1</td>
<td>U59</td>
<td>74LS08 quad 2-in AND</td>
<td>30000-00008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IC SOCKETS</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>IC Socket, 8 PIN</td>
<td>58102-00080</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>IC Socket, 14 PIN</td>
<td>58102-00140</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>IC Socket, 16 PIN</td>
<td>58102-00160</td>
</tr>
<tr>
<td>32</td>
<td></td>
<td>IC Socket, 18 PIN</td>
<td>58102-00180</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MISCELLANEOUS</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td></td>
<td>Header Strip, 1x3</td>
<td>56004-01003</td>
</tr>
<tr>
<td>39</td>
<td></td>
<td>Berg Jumper</td>
<td>56200-00001</td>
</tr>
<tr>
<td>2</td>
<td>CR1,CR2</td>
<td>Diode, Light Emitting</td>
<td>37400-00001</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Heatsink, to 220</td>
<td>60022-00001</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Nut, hex, 6-32 &amp; lock washer (KEPS)</td>
<td>73006-32001</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Screw, Phillips head (SIMS), 6-32x3/8</td>
<td>71006-32061</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>PC Board</td>
<td>02016-00003</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Extractor, PCB Non-locking</td>
<td>60100-00000</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Roll Pin Extractor Mounting</td>
<td>60100-00001</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Owner's Manual</td>
<td>89000-02116</td>
</tr>
</tbody>
</table>
## 2116 Address/Chip Table

<table>
<thead>
<tr>
<th>X000-X3FF</th>
<th>X400-X7FF</th>
<th>X800-XBFF</th>
<th>XC00-XFFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH A</td>
<td>LOW</td>
<td>HIGH B</td>
<td>LOW</td>
</tr>
</tbody>
</table>
APPENDIX A

LIMITED WARRANTY

California Computer Systems (CCS) warrants to the original purchaser of its products that

(1) its CCS assembled and tested products will be free from materials defects for a period of one (1) year, and be free from defects of workmanship for a period of ninety (90) days; and

(2) its kit products will be free from materials defects for a period of ninety (90) days.

The responsibility of CCS hereunder, and the sole and exclusive remedy of the original purchaser for a breach of any warranty hereunder, is limited to the correction or replacement by CCS at CCS's option, at CCS's service facility, of any product or part which has been returned to CCS and in which there is a defect covered by this warranty; provided, however, that in the case of CCS assembled and tested products, CCS will correct any defect in materials and workmanship free of charge if the product is returned to CCS within ninety (90) days of original purchase from CCS; and CCS will correct defects in materials in its products and restore the product to an operational status for a labor charge of $25.00, provided that the product is returned to CCS within ninety (90) days in the case of kit products, or one (1) year in the case of CCS assembled and tested products. All such returned products shall be shipped prepaid and insured by original purchaser to:

Warranty Service Department
California Computer Systems
250 Caribbean Drive
Sunnyvale, California
94086
CCS shall have the right of final determination as to the existence and cause of a defect, and CCS shall have the sole right to decide whether the product should be repaired or replaced.

This warranty shall not apply to any product or any part thereof which has been subject to

(1) accident, neglect, negligence, abuse or misuse;

(2) any maintenance, overhaul, installation, storage, operation, or use, which is improper; or

(3) any alteration, modification, or repair by anyone other than CCS or its authorized representative.

THIS WARRANTY IS EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES EXPRESSED OR IMPLIED OR STATUTORY INCLUDING THE WARRANTIES OF DESIGN, MERCHANTABILITY, OR FITNESS OR SUITABILITY FOR USE OR INTENDED PURPOSE AND OF ALL OTHER OBLIGATIONS OR LIABILITIES OF CCS. To any extent that this warranty cannot exclude or disclaim implied warranties, such warranties are limited to the duration of this express warranty or to any shorter time permitted by law.

CCS expressly disclaims any and all liability arising from the use and/or operation of its products sold in any and all applications not specifically recommended, tested, or certified by CCS, in writing. With respect to applications not specifically recommended, tested, or certified by CCS, the original purchaser acknowledges that he has examined the products to which this warranty attaches, and their specifications and descriptions, and is familiar with the operational characteristics thereof. The original purchaser has not relied upon the judgement or any representations of CCS as to the suitability of any CCS product and acknowledges that CCS has no knowledge of the intended use of its products. CCS EXPRESSLY DISCLAIMS ANY LIABILITY ARISING FROM THE USE AND/OR OPERATION OF ITS PRODUCTS, AND SHALL NOT BE LIABLE FOR ANY CONSEQUENTIAL OR INCIDENTAL OR COLLATERAL DAMAGES OR INJURY TO PERSONS OR PROPERTY.

CCS's obligations under this warranty are conditioned on the original purchaser's maintenance of explicit records which will accurately reflect operating conditions and maintenance performed on CCS's products and establish the nature of any unsatisfactory condition of CCS's products. CCS, at its request, shall be given access to such records
for substantiating warranty claims. No action may be brought for breach of any express or implied warranty after one (1) year from the expiration of this express warranty's applicable warranty period. CCS assumes no liability for any events which may arise from the use of technical information on the application of its products supplied by CCS. CCS makes no warranty whatsoever in respect to accessories or parts not supplied by CCS, or to the extent that any defect is attributable to any part not supplied by CCS.

CCS neither assumes nor authorizes any person other than a duly authorized officer or representative to assume for CCS any other liability or extension or alteration of this warranty in connection with the sale or any shipment of CCS's products. Any such assumption of liability or modification of warranty must be in writing and signed by such duly authorized officer or representative to be enforceable. These warranties apply to the original purchaser only, and do not run to successors, assigns, or subsequent purchasers or owners; AS TO ALL PERSONS OR ENTITIES OTHER THAN THE ORIGINAL PURCHASER, CCS MAKES NO WARRANTIES WHATSOEVER, EXPRESS OR IMPLIED OR STATUTORY. The term "original purchaser" as used in this warranty shall be deemed to mean only that person to whom its product is originally sold by CCS.

Unless otherwise agreed, in writing, and except as may be necessary to comply with this warranty, CCS reserves the right to make changes in its products without any obligation to incorporate such changes in any product manufactured theretofore.

This warranty is limited to the terms stated herein. CCS disclaims all liability for incidental or consequential damages. Some states do not allow limitations on how long an implied warranty lasts and some do not allow the exclusion or limitation of incidental or consequential damages so the above limitations and exclusions may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.