Owner's Manual

Model 7424 Calendar/Clock Module

California
Computer
Systems

CALIFORNIA COMPUTER SYSTEMS

APPLE II™ CALENDAR/CLOCK MODULE

MODEL 7424

OWNER'S MANUAL

89001-07424

California Computer Systems
250 Caribbean Drive
Sunnyvale, CA 94086

TABLE OF CONTENTS

PREFACE

This manual is intended to provide as complete an understanding as possible of the hardware and software features of the CCS Model 7424 Calendar/Clock board. At the same time, we recognize that many APPLE owners want to be able to plug a board in and use it without having to wade through extensive discussions of hardware and software theory. For those of you in the latter category, Chapter 2 and Sections 1-4 of Chapter 3 provide all of the information necessary for the set-up, installation, and operation of the 7424. More curious users and those planning to write their own software will want to read the manual in its entirety.

A number of addresses referred to in the text depend on the number of the slot in which the 7424 is installed. We use "n" throughout the text to represent the slot number.

CHAPTER 1

THEORY OF OPERATION

The CCS Model 7424 Calendar/Clock Module is an addressable real time clock which counts seconds, minutes, hours, days-of-week, dates, months, and years. It automatically adjusts for leap year, and may be set for either a 12 hour (AM/PM) or 24 hour format. Normally operating on +5 volts from the APPLE II's power supply, the 7424 Calendar/Clock will continue time-counting functions down to +2.2 volts, allowing back-up batteries to preserve accurate time-keeping when the computer is powered down. Three on-board jumper-selectable drivers provide a range of capabilities without requiring a substantial time investment in software. For those users who want or need to create their own software, ample memory space is provided on-board.

1.1 THE 5832 CLOCK/CALENDAR CHIP

The heart of the 7424 Calendar/Clock Module is a 5832 Microprocessor Real-Time Clock/Calendar.

This device uses a 32.768 kHz crystal to count seconds, minutes, hours, days, months, and years. These counts are addressed one decimal digit at a time through inputs A0-A3, and are output in binary-coded-decimal (BCD) digits through D0-D3 when the READ input is high. A high to the WRITE input allows setting of the time data digit addressed through A0-A3. (See Table 3.1 for the address codes for each time data digit.) A high to the HOLD input maintains all counters in a static state, ensuring error-free reading and writing; accuracy is unaffected as long as HOLD is high for less than a second. A low to the CS input disables inputs and outputs, but does not affect time counting.

1.2 PROGRAM MEMORY

Three separate driver programs are available on-board in a 1Kx8 2708 EPROM (U1), each fitting into a 256-byte block. The fourth block is empty, and may be used to store user-generated programs. Jumpers A8 and A9 control address inputs A8 and A9 of the EPROM, allowing you to select which of the four 256-byte blocks will be enabled with the board (see Table 2.1).

Sockets are included on-board for the addition of two 256x4 RAMs or ROMs; the memory chips themselves must be provided by the user. Users who plan to create their own software for the 7424 may want to take advantage of the 256-byte ROM/RAM option rather than risk losing the three programs in the 2708 when burning in a fourth. If ROMs or RAMs are installed, the EPROM must be removed from the board.

1.3 SELECTION LOGIC

-I/O SEL and -DEV SEL, along with R/-W and A0, are the primary signals involved in the 7424's control logic. -I/O SEL low enables the Program EPROM. U9, a bi-directional data buffer, is enabled by a low on either -I/O SEL or -DEV SEL; when both inputs are high, a transistor shuts off power to U9 in order to conserve power. Direction of data transfer through U9 is determined by the R/-W line.

When -DEV SEL is low and A0 is high, the trailing (rising) edge of a low write pulse on R/-W clocks U8, the Clock Address Flip-Flops. Thus a write to an odd address between $C0(8+n)0 and $C0(8+n)F latches data from the 7424 data bus to the inputs of the 5832. Data bits D0-D3 address one of the 5832's BCD digits, D4 controls the Hold input, and D5 is tied to Chip Select. The digit addressed can be read at any even address between $C0(8+n)0 and $C0(8+n)F; Read is high when -I/O SEL is high and Write (see below) is low. A write to the same location changes the value of the digit addressed to the value on data lines D3-D0 (or to 00 in the case of seconds). When A0 is low, lows on -DEV SEL and R/-W force the Write input high (if the input line is jumper-enabled).

1.4 INTERRUPTS

The 7424 is capable of generating interrupts periodically. The PI (Programmable Interrupt) jumpers 1-4 allow interrupts every 1/1024 second (#4), every second (#3), every minute (#2), or every hour (#1). Pulses with these intervals are available at D0-D3 when A0-A3, CS, and READ are all high and HOLD is low. (The last condition is not necessary for the 1024 Hz square wave on D0.) The pulses on D1-D3 have a duration of 122.1 microseconds. D1 and D2 pulse low, while D3 pulses high.

Interrupts are enabled when three conditions are met: 1) data bus lines D0-D3 are high; 2) D6 is high; 3) neither -I/O SEL nor -DEV SEL is active. D0-D3 high cause the periodic pulses to be output by the clock. A low on D6, -DEV SEL, or -I/O SEL pulls the Interrupt Flip-Flop's Preset input low, disqualifying the clocking of the flip-flop and forcing -IRQ (the flip-flop's Q output) high. This prevents generation of an interrupt request by a read from or write to the 5832.

The Interrupt Flip-Flop is clocked by the Phase 0 clock signal (pin 40). If one of the PIE jumpers is installed, a periodic pulse from the 5832 will be clocked into the flip-flop, forcing -IRQ low. Running the 7424 driver resets the Interrupt Flip-Flop, removing the interrupt request.

CHAPTER 2

SET-UP AND INSTALLATION

2.1 BATTERIES

If you wish to take advantage of the battery back-up capability of the 7424, you will need to purchase and install batteries. The batteries are readily available and easily installed. Batteries that will work include:

Eveready	E675
Mallory	M675
Burgess	Hg-675

To install the batteries, simply lift the tops of the clips and insert the batteries. Make sure that the batteries are securely in place before you install the 7424 in your APPLE.

If you do not install batteries, you will need to set the 7424 each time you turn the power on, and will need to leave the Write Enable jumper permanently set to EN.

2.2 SETTING THE JUMPERS

Before you install the 7424 in your system, you will need to configure the jumpers for the options you desire. If you plan to use one of the programs in the EPROM, you must set the A8 and A9 jumpers as indicated in Table 2.1 below. A "0" indicates an installed jumper.

A9	A8	ROUTINE
0	0	CLOCK INPUT
0	1	TIME STRING
1	0	SCREEN DISPLAY
1	1	EMPTY

Table 2.1

If you plan to install a pair of 256x4-bit ROMs or RAMs for storing a custom driver program, you must remove the EPROM from the board and set the ROM/RAM jumper appropriately.

The Write Enale jumper enables or disables writing to the 5832. It is a good idea to enable the line only when you want to set the 5832. You will need to set the jumper to EN before first installing the board. After you have installed the board and set the time and date (see Section 3.1), we recommend that you turn off the power to the APPLE (batteries must be installed) and set the jumper to the opposite (disabling) position. This prevents accidental writing to the 5832. In normal situations, you should not need to set the 5832 again for the life of the battery, except to adjust for daylight-saving time.

If you wish to enable programmable interrupts, jumper-connect PI header pin pairs 1, 2, 3, or 4 for the period desired, as shown in Table 2.2 below.

JUMPER	INTERRUPT PERIOD
PI 4	1/1024 SECOND
PI 3	1 SECOND
PI 2	1 MINUTE
PI 1	1 HOUR

Table 2.2

2.3 INSTALLATION

Before you begin the installation procedure, turn the computer off and disconnect the power cord.

```
**************************************
*                                    *
*  WARNING:  Do not remove the cover *
*  of your computer if the power cord *
*  is plugged in.   You may injure   *
*  yourself or damage your computer. *
*                                    *
*b************************************
```

Place the computer directly in front of you. Put the palms of your hands on the back of the computer and curl your fingers around the rear edge. Gently but firmly pull up until you hear two distinct pops. Don't lift the cover any farther. Slide it to the rear to remove it from the computer. Inside, toward the rear of the computer, you will see eight 50-pin connecters. They are numbered 0 through 7 from left to right. Place the 7728 in any of these connectors, with the exception of #0, the leftmost; slot 0 does not have the 256-byte program area available. We suggest that you use slot 4 (allowing you to run programs written for Mountain Hardware's APPLE Clock™, which must reside in slot 4). Insert the card by holding it so that the component side of the card is to the right. Align the card edge into the chosen connector and gently push the card down until it is firmly seated. Replace the computer cover, and you are ready to try out the board.

2.4 CALIBRATION

All 7424 Calendar/Clock Modules are accurately set at the factory, but shipping vibrations may in some cases cause a board to be slightly fast or slow. Should you find that your board loses or gains time (from a few seconds up to a minute or two in 24 hours), you will need to adjust the variable capacitor C2, which fine-tunes the crystal-controlled clock frequency.

Most users who find that their calendar/clock modules need calibration will have to use the adjust-a-bit-and-check-the-results method. After determining the amount of time gained or lost per 24 hours, insert a small screwdriver blade into the slot at the top end of C2 and adjust slightly. Wait long enough to determine the effect of this adjustment, then readjust accordingly. Continue this process until you achieve the accuracy you desire.

For those of you who have access to a six-digit frequency counter, there is an easier way. Install PI jumper #4, then

```
        POKE 49281 + (16*n),111
```

to enable the 1 KHz square wave at PI4. Adjust C2 as described above so that the frequency of the wave at PI4 is 1024.00 ± .01 Hz.

The variable capacitor C2 also allows you to correct for another possible cause of diminished accuracy: crystal aging. Over the years, crystals undergo a very slight but detectable change in frequency. Since your 7424 should give you years of service, sometime in the future you will probably want to make a minor adjustment of C2.

CHAPTER 3

SOFTWARE

Three slot-independent drivers reside on-board the 7424 in a 1K EPROM, each routine occupying a 256-byte block. Which block will be enabled with the board depends on the A9 and A8 jumpers, as indicated in Table 2.1. User-written routines may be stored in the fourth block of the EPROM, or more safely in two 256x4 ROMs or RAMs.

3.1 SETTING THE CLOCK

When you have your 7424 set up and installed, you will want to set the clock/calendar to the correct date and time. You could do so by separately addressing each digit, but the task would be tedious. The BASIC program on the next page makes setting the clock easy and quick. Simply enter and run the program, answering the questions asked. (Be sure that the Write Enable jumper is set to EN.)

```
 10  REM    BASIC PROGRAM TO SET TIME
 20  REM  FIND OUT WHAT SLOT THE TIMER IS IN
 30  INPUT "WHAT SLOT IS THE CLOCK IN? ";S
 40  REM  COMPUTE SLOT BASE ADDRESS
 50  BASE = - 16256 + (16 * S)
 60  REM  SET YEARS
 70  INPUT "WHAT YEAR (0-99)? ";A$
 80  GOSUB 1000
 90  REM  CHECK FOR ENTRY ERROR
100  IF A1 = 1000 THEN 70
110  POKE BASE + 1,59: POKE BASE,A1
120  POKE BASE + 1,60: POKE BASE,A2
130  REM  SET MONTH
140  INPUT "WHAT MONTH (1-12) ? ";A$
150  IF VAL (A$) † 12 THEN 140
160  GOSUB 1000
170  IF A1 = 1000 THEN 140
180  POKE BASE + 1,57: POKE BASE,A1
190  POKE BASE + 1,58: POKE BASE,A2
200  REM  SET DAYS
210  INPUT "WHAT DAY (1-31) ? ";A$
220  IF VAL (A$) † 31 THEN 130
230  GOSUB 1000
240  IF A1 = 1000 THEN 210
250  POKE BASE + 1,55: POKE BASE,A1
260  POKE BASE + 1,56: POKE BASE,A2
270  REM  FIND HOUR FORMAT
280  INPUT "WHAT HOUR FORMAT (12 OR 24) ? ";B$
290  REM  SET HOURS
300  INPUT "SET HOURS TO ? ";A$
310  IF VAL (A$) † THEN 300
320  GOSUB 1000
330  IF A1 = 1000 THEN 300
340  IF B$ = "12" THEN INPUT "AM OR PM ? ";C$
350  IF B$ = "24" THEN A2 = A2 + 8:C$ = ""
360  IF C$ = "PM" THEN A2 = A2 + 4
370  POKE BASE + 1,52: POKE BASE,A1
380  POKE BASE + 1,53: POKE BASE,A2
390  REM  SET MINUTES
400  INPUT "SET MINUTES TO ? ";A$
410  IF VAL (A$) † 60 THEN 400
420  GOSUB 1000
430  IF A1 = 1000 THEN 400
440  POKE BASE + 1,50: POKE BASE,A1
450  POKE BASE + 1,51: POKE BASE,A2
460  INPUT "HIT RETURN TO START TIME.  SECONDS
     SET TO 00 ";A$
470  POKE BASE + 1,49: POKE BASE,00
480  POKE BASE + 1,0: END
900  REM  ROUTINE TO GET VALUES TO SET
     TIME AND DATA
1000  A1 = LEN (A$)
1010  IF A1 = 1 THEN A2 = 0:A1 = VAL (A$): RETURN
1020  IF A1 = 2 THEN A2 = VAL ( LEFT$ (A$,1)):
      A1 = VAL ( RIGHT$ (A$,1)): RETURN
1030  A1 = 1000:A2 = 1000:RETURN
```

3.2 CLOCK INPUT ROUTINE

The first routine loads calendar/ clock data into the first 17 locations of the keyboard input buffer. To use this routine in a BASIC program, four statements are necessary:

```
60  IN#n         (n=slot #)
70  INPUT X$
80  IN#0
90  PRINT X$
```

Line 60 changes locations $38 and $39, the APPLE's input vectors, to point to the 7424. Line 70, in which X may be any character, causes the data to be transferred to the input buffer. Input control is returned to the keyboard (slot #0) by line 80. Line 90 prints the data in the following format:

```
MM/DD hh;mm;ss.000
```

The above format ends with .000, which are dummy characters and will not change, in order to provide compatibility with the Mountain Hardware Apple Clock™, which counts thousandths of a second. Programs written for the Mountain Hardware clock in which fractions of a second are not critical will run with the CCS 7424.

3.3 TIME STRING ROUTINE

This program works only in APPLESOFT, which allows you to store up to 255 characters as a string. The 7424 Time String routine continually rewrites the correct hours, minutes, and seconds into an eight-character string. To use this routine, you must create a string TI$ with eight characters, including spaces. (Any characters can be used, so go ahead and express yourself.) You must enable periodic interrupts by installing one of the PI jumpers. Once it has been called and interrupt generation enabled, the routine will write the correct time into the string each time it interrupts.

To start the clock storing data in the TI$ location, CALL 49152 + (256 * n), the intialization entry point. The INIT subroutine will load the normal entry point of the routine into $3FE and $3FF, the interrupt vector addresses, so that when the processor is interrupted it will turn control over to the Clock Input routine. The routine searches the strings for TI$ and, if it finds it, transfers the correct time to the string's eight bytes of memory, then returns control to the processor. To read the time, all you need to do is type in a simple command, PRINT TI$. The time will be printed in the following format:

```
HH/MM/SS
```

If you set the clock to the 12 hour mode, the routine will not specify AM or PM.

There are two ways to stop this routine from interrupting your computer. One is to disable all interrupts by setting the Interrupt status flag. To do this, CALL 49405 + (256 * n). The other way is to disable interrupt request generation by the 7424. To do this, POKE a byte in which at least one of bits 0, 1, 2, 3, 5 and 6 is low ($00 works fine) to $C0(8+n)1. In either case, the routine can be re-initialized as described above.

Note: This routine's interrupts should be disabled whenever you are adding lines to or deleting lines from a program. The routine uses the APPLESOFT string pointers to find TI$ and when lines are added there is a period of time when the pointers do not reflect the actual locations of the strings.

3.4 SCREEN DISPLAY ROUTINE

This routine interrupts the processor to maintain the correct time in the upper right-hand corner of your CRT screen. To use it you must enable the 1 kHz interrupts by jumpering PIE 1. The display format is similar to the TI$ format except that AM and PM can be specified and colons replace the slashes.

HH:MM:SS AM

To use this program you must call the enabling routine at $CnD0:

```
CALL 49360+(256 * n)
```

The periodic interrupts will continually update the clock. Other programs may be run as long as they are compatible with the interrupts; however, anything written in the screen position reserved for the time display will be overwritten at the next interrupt. Interrupts can be disabled by a CALL to 49395 + 256 * n ($CnF3), or by a POKE to $C0(8+n)1 as described in Section 3.3.

3.5 WRITING YOUR OWN SOFTWARE

There are too many possibilities with a board like the CCS 7424 for us to make more than general comments about writing software for it. Certain firmware routines of your APPLE are very useful. The program listings included in this chapter show how some of them can be used, but you should already have a pretty good idea of what is available in your firmware if you are going to be doing any very complicated programming for the 7424.

You will also need to know how to communicate with the clock itself. This is actually fairly simple. To latch the address of a date or time digit, write $2x to $C0(8+n)1, where x is the code for the desired data as given in Table 3.1. (The format of the clock address byte is shown below. Bits 4 and 5 should be high to address the clock.) Data may be read at $C0(8+n)0. To enable interrupt request generation by the 7424, set the processor Interrupt flag and write $6F to $C0(8+n)1 before exiting. (Bit 6 high enables Interrupts when bits 0-5 are all high.)

BIT	7	6	5	4	3	2	1	0
5832 INPUT	-	INT	CS	HLD	A3	A2	A1	A0

5832 ADDRESS/DATA TABLE

ADDRESS				INTERNAL	DATA			
A0	A1	A2	A3	COUNTER	D0	D1	D2	D3
0	0	0	0	SECOND 1	X	X	X	X
1	0	0	0	SECOND 10	X	X	X	d
0	1	0	0	MINUTE 1	X	X	X	X
1	1	0	0	MINUTE 10	X	X	X	
0	0	1	0	HOUR 1	X	X	X	X
1	0	1	0	HOUR 10	X	X	a	b
0	1	1	0	DAY/WEEK	X	X	X	
1	1	1	0	DATE 1	X	X	X	X
0	0	0	1	DATE 10	X	X	c	
1	0	0	1	MONTH 1	X	X	X	X
0	1	0	1	MONTH 10	X			
1	1	0	1	YEAR 1	X	X	X	X
0	1	0	1	YEAR 10	X	X	X	X
1	1	1	1	REFERENCE	e	f	g	h

X. 0 or 1: BCD Digit
a. 0 for AM, 1 for PM
b. 0 for 12 hour, 1 for 24 hour format
c. 1 for 29 days in month 2 :write
d. Seconds reset to 00 by write to clock
e. 1024 Hz square wave
f. 1 Hz pulse
g. 1/60 Hz pulse
h. 1/3600 Hz pulse

Table 3.1

3.6 CLOCK INPUT ROUTINE LISTING

```
Cn00              *          CLOCK INPUT ROUTINE
Cn00              *
Cn00              *   THIS ROUTINE INPUTS DATA WHEN CALLED.
Cn00              *   THE SLOT NUMBER IS SIGNIFIED BY n.
Cn00              *
Cn00              *
Cn00              *
Cn00              BUF     EQU     $0200
Cn00              ADDR    EQU     $C081
Cn00              DATA    EQU     $C080
Cn00              *
Cn00                      ORG     $Cn00
Cn00              *
Cn00              *
Cn00-08           ENT     PHP                 Save P Register
Cn01-78                   SEI                 Disable CPU Interrupts
Cn02-98                   TYA                 Save Y
Cn03-48                   PHA
Cn04-20 CB FF             JSR     $FFCB       Get Slot Number
Cn07-BA                   TSX
Cn08-BD 00 01             LDA     $100,X
Cn0B-0A                   ASL     A           Multiply by 16
Cn0C-0A                   ASL     A
Cn0D-0A                   ASL     A
Cn0E-0A                   ASL     A
Cn0F-A8                   TAY                 Save in Y Reg
Cn10-A9 2A                LDA     #$2A        Get Months Tens
Cn12-99 81 C0             STA     ADDR,Y
Cn15-B9 80 C0             LDA     DATA,Y
Cn18-8D 00 02             STA     BUF+0       Input
```

```
Cn1B-A9 29          LDA    #$29      Get Months Ones
Cn1D-99 81 C0       STA    ADDR,Y
Cn20-B9 80 C0       LDA    DATA,Y
Cn23-8D 01 02       STA    BUF+1     Input
Cn26-A9 AF          LDA    #$AF      ASCII /
Cn28-8D 02 02       STA    BUF+2
Cn2B-A9 28          LDA    #$28      Get Date Tens
Cn2D-99 81 C0       STA    ADDR,Y
Cn30-B9 80 C0       LDA    DATA,Y
Cn33-8D 03 02       STA    BUF+3     Input
Cn36-A9 27          LDA    #$27      Get Date Ones
Cn38-99 81 C0       STA    ADDR,Y
Cn3B-B9 80 C0       LDA    DATA,Y
Cn3E-8D 04 02       STA    BUF+4     Input
Cn41-A9 A0          LDA    #$A0      ASCII Space
Cn43-8D 05 02       STA    BUF+5
Cn46-A9 25          LDA    #$25      Get Hours Tens
Cn48-99 81 C0       STA    ADDR,Y
Cn4B-B9 80 C0       LDA    DATA,Y
Cn4E-29 03          AND    $#03      Mask Format Flags
Cn50-8D 09 02       STA    BUF+6     Input
Cn53-A9 24          LDA    #$24      Get Hours Ones
Cn55-99 81 C0       STA    ADDR,Y
Cn58-B9 80 C0       LDA    DATA,Y
Cn5B-8D 0A 02       STA    BUF+7     Input
Cn5E-A9 23          LDA    #$23      Get Minutes Tens
Cn60-99 81 C0       STA    ADDR,Y
Cn63-B9 80 C0       LDA    DATA,Y
Cn66-8D 0C 02       STA    BUF+9     Input
Cn69-A9 22          LDA    #$22      Get Minutes Ones
```

```
Cn6B-99 81 C0            STA    ADDR,Y
Cn6E-B9 80 C0            LDA    DATA,Y
Cn71-8D 0D 02            STA    BUF+$A     Input
Cn74-A9 21               LDA    #$21       Get Seconds Tens
Cn76-99 81 C0            STA    ADDR,Y
Cn79-B9 80 C0            LDA    DATA,Y
Cn7C-8D 0F 02            STA    BUF+$C     Input
Cn7F-A9 20               LDA    #$20       Get Seconds Ones
Cn81-99 81 C0            STA    ADDR,Y
Cn84-B9 80 C0            LDA    DATA,Y
Cn87-8D 10 02            STA    BUF+$D     Input
Cn8A-A2 20               LDX    #$20       Set Up Index
Cn8C-A9 BF        LP1    LDA    #$BF       Make All ASCII
Cn8E-3D 00 02            AND    BUF,X
Cn91-9D 00 02            STA    BUF,X
Cn94-CA                  DEX
Cn95-10 F5               BPL    LP1        Next Character
Cn97-A9 BB               LDA    #$BB       Get ";"
Cn99-8D 08 02            STA    BUF+8
Cn9C-8D 0B 02            STA    BUF+$B
Cn9F-A9 AE               LDA    #$AE       Get "."
CnA1-8D 0E 02            STA    BUF+$E     Put After Seconds
CnA4-A9 B0               LDA    #$B0       Get 0
CnA6-8D 0F 02            STA    BUF+$F     Put 3 After "."
CnA9-8D 10 02            STA    BUF+$10
CnAC-8D 11 02            STA    BUF+$11
CnAF-A2 12               LDX    #$12       Put Length in X
CnB1-68                  PLA               Get Back Y
CnB2-A8                  TAY
CnB3-A9 8D               LDA    #$8D       Return CR to End
CnB5-8D 12 02            STA    BUF+$12    End String with CR
CnB8-28                  PLP
CnB9-60                  RTS               Return
```

```
Cn00                *                       TI$ ROUTINE
Cn00                *
Cn00                *  THIS PROGRAM CONTINUALLY REWRITES A STRING TI$
Cn00                *  WITH  HOURS, MINUTES, SECONDS  IN THE FORMAT
Cn00                *  HH/MM/SS.  SLOT NUMBER IS SIGNIFIED BY n.
Cn00                *
Cn00                *
Cn00                TEMP1   EQU    $4F8-$C0
Cn00                PNT     EQU    $69      STRING POINTER
Cn00                BUFF    EQU    $2F0     DATA BUFFER
Cn00                DATA    EQU    $C080    CLOCK DATA LOC
Cn00                CNTRL   EQU    $C081    CLOCK ADDR LOC
Cn00                IOSAVE  EQU    $FF4A    REGISTER SAVE ROUTINE
Cn00                IOREST  EQU    $FF3F    REGISTER STORE ROUTINE
Cn00                *
Cn00                        ORG    $Cn00
Cn00                *
Cn00                *
Cn00-58             STRT    CLI               Enable Interrupts
Cn01-08                     PHP               Save Status Word
Cn02-78                     SEI               Disable Interrupts
Cn03-2C CB FF               BIT    $FFCB      Set V Flag for Init Entry
Cn06-70 03                  BVS    FSLOT      Go Find Slot
Cn08-A5 45          INENT   LDA    $45        Intrpt Entry Point
Cn0A-B8                     CLV               Clear V for Intrpt Entry
Cn0B-48             FSLOT   PHA               Save Registers
Cn0C-98                     TYA
Cn0D-48                     PHA
Cn0E-8A                     TXA
Cn0F-48                     PHA
Cn10-20 CB FF               JSR    $FFCB      Get Slot Number
```

```
Cn13-BA                   TSX
Cn14-BD 00 01             LDA     $100,X
Cn17-AA                   TAX
Cn18-0A                   ASL     A            Multiply by 16
Cn19-0A                   ASL     A
Cn1A-0A                   ASL     A
Cn1B-0A                   ASL     A
Cn1C-9D 38 04             STA     TEMP1,X      Store Result
Cn1F-70 31                BVS     INIT         Branch if Init Entry
Cn21-A0 00                LDY     #$00         Clear Y Index
Cn23-B1 69        SRCH    LDA     (PNT),Y      Search for TI in String Table
Cn25-C9 54                CMP     #$54         Look for T
Cn27-D0 08                BNE     NXT          Branch If Not
Cn29-C8                   INY                  If T Found, See If Next Is I
Cn2A-B1 69                LDA     (PNT),Y
Cn2C-C9 C9                CMP     #$49+$80
Cn2E-F0 0C                BEQ     FND          String Found; Go Get Time
Cn30-88                   DEY
Cn31-98           NXT     TYA                  Go Find Next $
Cn32-18                   CLC                  Clear Carry for Add
Cn33-69 07                ADC     #$07         Point to Next String
Cn35-A8                   TAY                  Put New Index in Y
Cn36-90 EB                BCC     SRCH         Go Check Next String
Cn38-B0 1E                BCS     EXIT         Exit if TI$ Not Found
Cn3A-C8           FND     INY                  Point to Lentgh of String
Cn3B-A5 46                LDA     $46          Save 46 and 47 on Stack
Cn3D-48                   PHA
Cn3E-A5 47                LDA     $47
Cn40-48                   PHA
Cn41-C8                   INY                  Point to First Byte of $
Cn42-B1 69                LDA     (PNT),Y      Get Data Pointer
```

```
Cn44-85 46                STA     $46         Store It
Cn46-C8                   INY                 Point to Second Byte
Cn47-B1 69                LDA     (PNT),Y     Get Pointer
Cn49-85 47                STA     $47         Store It
Cn4B-BC 38 04             LDY     TEMP1,X     Store Y Index
Cn4E-50 1E                BVC     SDTA        Always Branch
Cn50-8E FF 03    INIT     STX     $3FF        Set Intrpt Vectors for Clock
Cn53-A9 08                LDA     #$08
Cn55-8D FE 03             STA     $3FE
Cn58-BC 38 04    EXIT     LDY     TEMP1,X
Cn5B-A9 6F                LDA     #$6F        Enable Clock Interrupts
Cn5D-99 81 C0             STA     CNTRL,Y
Cn60-70 06                BVS     SHT
Cn62-68                   PLA                 Restore 46 and 47
Cn63-85 47                STA     $47
Cn65-68                   PLA
Cn66-85 46                STA     $46
Cn68-68          SHT      PLA                 Restore Registers
Cn69-AA                   TAX
Cn6A-68                   PLA
Cn6B-A8                   TAY
Cn6C-68                   PLA
Cn6D-40                   RTI                 Return
Cn6E-8A          SDTA     TXA                 Store A Reg on Stack
Cn6F-48                   PHA
Cn70-A2 00                LDX     #$00        Set X Index to 0
Cn72-A9 25                LDA     #$25        Get Hours x 10
Cn74-99 81 C0             STA     CNTRL,Y
Cn77-B9 80 C0             LDA     DATA,Y
Cn7A-29 F3                AND     #$F3        Mask Format Codes
Cn7C-81 46                STA     ($46,X)     Store in String
```

```
Cn7E-E6 46              INC    $46        Next Position
Cn80-D0 02              BNE    S1         Incr Hi Byte If Lo Byte $00
Cn82-E6 47              INC    $47
Cn84-A9 24        S1    LDA    #$24       Get Hours x 1
Cn86-99 81 C0           STA    CNTRL,Y
Cn89-B9 80 C0           LDA    DATA,Y
Cn8C-81 46              STA    ($46,X)    Store in String
Cn8E-E6 46              INC    $46        Next Position
Cn90-D0 00              BNE    S2
Cn92-E6 46        S2    INC    $46        Skip space for /
Cn94-D0 04              BNE    S3         Incr Hi Byte if Lo Byte $00
Cn96-D0 C0        X1    BNE    EXIT       Passing through
Cn98-E6 47              INC    $47
Cn9A-A9 23        S3    LDA    #$23       Get Minutes by 10
Cn9C-99 81 C0           STA    CNTRL,Y
Cn9F-B9 80 C0           LDA    DATA,Y
CnA2-81 46              STA    ($46,X)    Store in String
CnA4-E6 46              INC    $46        Next Position
CnA6-D0 02              BNE    S4         Incr Hi Byte if Lo Byte $00
CnA8-E6 47              INC    $47
CnAA-A9 22        S4    LDA    #$22       Get Minutes x 1
CnAC-99 81 C0           STA    CNTRL,Y
CnAF-B9 80 C0           LDA    DATA,Y
CnB2-81 46              STA    ($46,X)    Store in String
CnB4-E6 46              INC    $46        Next Position
CnB6-D0 00              BNE    S5
CnB8-E6 46        S5    INC    $46        Skip Space for /
CnBA-D0 02              BNE    S6         Incr Hi Byte If Lo Byte $00
CnBC-E6 47              INC    $47
CnBE-A9 21        S6    LDA    #$21       Get Seconds x 10
CnC0-99 81 C0           STA    CNTRL,Y
```

```
CnC3-B9 80 C0                 LDA     DATA,Y
CnC6-81 46                    STA     ($46,X)      Store in String
CnC8-E6 46                    INC     $46          Next Position
CnCA-D0 02                    BNE     S7           Incr Hi Byte If Lo Byte $00
CnCC-E6 47                    INC     $47
CnCE-A9 20            S7      LDA     #$20         Get Seconds x 1
CnD0-99 81 C0                 STA     CNTRL,Y
CnD3-B9 81?C0                 LDA     DATA,Y
CnD6-81 46                    STA     ($46,X)      Store in String
CnD8-38                       SEC                  Go Back to Front of String
CnD9-A5 46                    LDA     $46
CnDB-E9 07                    SBC     #$07
CnDD-85 46                    STA     $46
CnDF-B0 02                    BCS     S8           Decr Hi Byte if Low Byte < 7
CnE1-C6 47                    DEC     $47
CnE3-A0 07            S8      LDY     #$07
CnE5-A9 3F            LP1     LDA     #$3F         Loop for Making ASCII
CnE7-31 46                    AND     ($46),Y
CnE9-91 46                    STA     ($46),Y
CnEB-88                       DEY
CnEC-10 F7                    BPL     LP1
CnEE-A9 2F                    LDA     #$2F         Get ASCII /
CnF0-A0 02                    LDY     #$2          Insert Between Digit Pairs
CnF2-91 41                    STA     ($46),Y
CnF4-A0 05                    LDY     #$5
CnF6-91 41                    STA     ($46),Y
CnF8-68                       PLA                  Restore A Reg
CnF9-AA                       TAX
CnFA-D0 9A                    BNE     X1
CnFC-78                       SEI                  Disable Interrupts
CnFD-60                       RTS                  Done
```

```
Cn00            *   SCREEN DISPLAY ROUTINE
Cn00            *
Cn00            * THIS ROUTINE INTERRUPTS TO REWRITE  TIME
Cn00            * ON CRT SCREEN.  SLOT NUMBER IS SIGNIFIED
Cn00            * BY n.
Cn00            *
Cn00            *
Cn00            BASE    EQU     $C080
Cn00            ADD     EQU     BASE+1
Cn00            SCRN    EQU     $0400
Cn00            IOSAVE  EQU     $FF4A
Cn00            IOREST  EQU     $FF3F
Cn00            *
Cn00            *       ORG     $Cn00
Cn00            *
Cn00            *
Cn00-A5 45      INTRP   LDA     $45         Get A Reg
Cn02-20 4A FF           JSR     IOSAVE      Save Registers
Cn05-BA                 TSX                 Get Slot Number from Stack
Cn06-BD 00 01           LDA     $100,X
Cn09-0A                 ASL     A           Multiply by 16
Cn0A-0A                 ASL     A
Cn0B-0A                 ASL     A
Cn0C-0A                 ASL     A
Cn0D-A8                 TAY                 Put Result in Y
Cn0E-A2 24              LDX     #$24        Set Screen Posit Index
Cn10-A9 20              LDA     #$20        Get Seconds
Cn12-99 81 C0           STA     ADD,Y
Cn15-B9 80 C0           LDA     BASE,Y
Cn18-29 BF              AND     #$BF        Make ASCII
```

```
Cn1A-9D 00 04        STA    SCRN,X     Display
Cn1D-CA              DEX               Next Space to Left
Cn1E-A9 21           LDA    #$21       Get Seconds
Cn20-99 81 C0        STA    ADD,Y
Cn23-B9 80 C0        LDA    BASE,Y
Cn26-29 BF           AND    #$BF       Make ASCII
Cn28-9D 00 04        STA    SCRN,X     Display
Cn2B-CA              DEX               Next Space to Left x 10
Cn2C-A9 BA           LDA    #$BA       Get :
Cn2E-9D 00 04        STA    SCRN,X     Display It
Cn31-CA              DEX               Next Space to Left
Cn32-A9 22           LDA    #$22       Get Minutes
Cn34-99 81 C0        STA    ADD,Y
Cn37-B9 80 C0        LDA    BASE,Y
Cn3A-29 BF           AND    #$BF       Make ASCII
Cn3C-9D 00 04        STA    SCRN,X     Display
Cn3F-CA              DEX               Next Space to Left
Cn40-A9 23           LDA    #$23       Get Minutes x 10
Cn42-99 81 C0        STA    ADD,Y
Cn45-B9 80 C0        LDA    BASE,Y
Cn48-29 BF           AND    #$BF       Make ASCII
Cn4A-9D 00 04        STA    SCRN,X     Display
Cn4D-CA              DEX               Next Space to Left
Cn4E-A9 BA           LDA    #$BA       Get :
Cn50-9D 00 04        STA    SCRN,X     Display It
Cn53-CA              DEX               Next Space to Left
Cn54-A9 24           LDA    #$25       Get  Hours
Cn56-99 81 C0        STA    ADD,Y
Cn59-B9 80 C0        LDA    BASE,Y
Cn5C-29 BF           AND    #$BF       Make ASCII
```

```
Cn5E-9D 00 04            STA   SCRN,X      Display
Cn61-CA                  DEX               Next Space to Left
Cn62-A9 25               LDA   #$25        Get Hours x 10
Cn64-99 81 C0            STA   ADD,Y
Cn67-B9 80 C0            LDA   BASE,Y
Cn6A-48                  PHA               Save on Stack
Cn6B-29 08               AND   #$08        12 or 24 Format?
Cn6D-F0 16               BEQ   AMPM        Branch if 12
Cn6F-68          AMPM1   PLA               Reget Hours x 10
Cn70-29 B3               AND   #$B3        Make ASCII
Cn72-9D 00 04            STA   SCRN,X      Display
Cn75-CA                  DEX               Next Space to Left
Cn76-A9 A0               LDA   #$A0        Get Space
Cn78-9D 00 04            STA   SCRN,X      Display It
Cn7B-A9 6F       EXIT    LDA   #$6F        Enable Clock Interrupts
Cn7D-99 81 C0            STA   BASE+1,Y
Cn80-20 3F FF            JSR   IOREST      Restore Registers
Cn83-40                  RTI               Return
Cn84-68          AMPM    PLA               Reget Hours x 10
Cn85-48                  PHA               Save on Stack
Cn86-29 04               AND   #$04        AM or PM?
Cn88-D0 04               BNE   PM          Branch if PM
Cn8A-A9 C1       AM      LDA   #$C1        Get A
Cn8C-D0 02               BNE   ST          Always Branch
Cn8E-A9 D0       PM      LDA   #$D0        Get P
Cn90-8D 26 04    ST      STA   SCRN+38     Display A or P
Cn93-A9 CD               LDA   #$CD        Get M
Cn95-8D 27 04            STA   SCRN+39     Display M
Cn98-38                  SEC
Cn99-B0 D3               BCS   AMPM1       Always Branch
```

```
                            ORG   $CnD0       Start-up Routine
CnD0-08                     PHP               Save A and X
CnD1-48                     PHA
CnD2-8A                     TXA
CnD3-48                     PHA
CnD4-20 CB FF               JSR   $FFCB       Get Slot
CnD7-BA                     TSX
CnD8-BD 00 01               LDA   $100,X      Store Slot High Address
CnDB-8D FF 03               STA   $3FF          in Interrupt Vector High
CnDE-0A                     ASL   A           Multiply Slot Number by 16
CnEF-0A                     ASL   A
CnE0-0A                     ASL   A
CnE1-0A                     ASL   A
CnE2-AA                     TAX               Store Result in X
CnE3-A9 00                  LDA   #$00        Set Interrupt Vector
CnE5-8D FE 03               STA   $3FE          Low to Zero
CnE8-A9 6F                  LDA   #$6F        Enable Clock Interrupts
CnEA-9D 81 C0               STA   #C081,X
CnED-68                     PLA               Restore Registers
CnEE-AA                     TAX
CnEF-68                     PLA
CnF0-28                     PLP
CnF1-58                     CLI               Enable Interrupts at Processor
CnF2-60                     RTS               Return
CnF3-78                     SEI               Disable Interrupts at Processor
CnF4-60                     RTS               Return
```

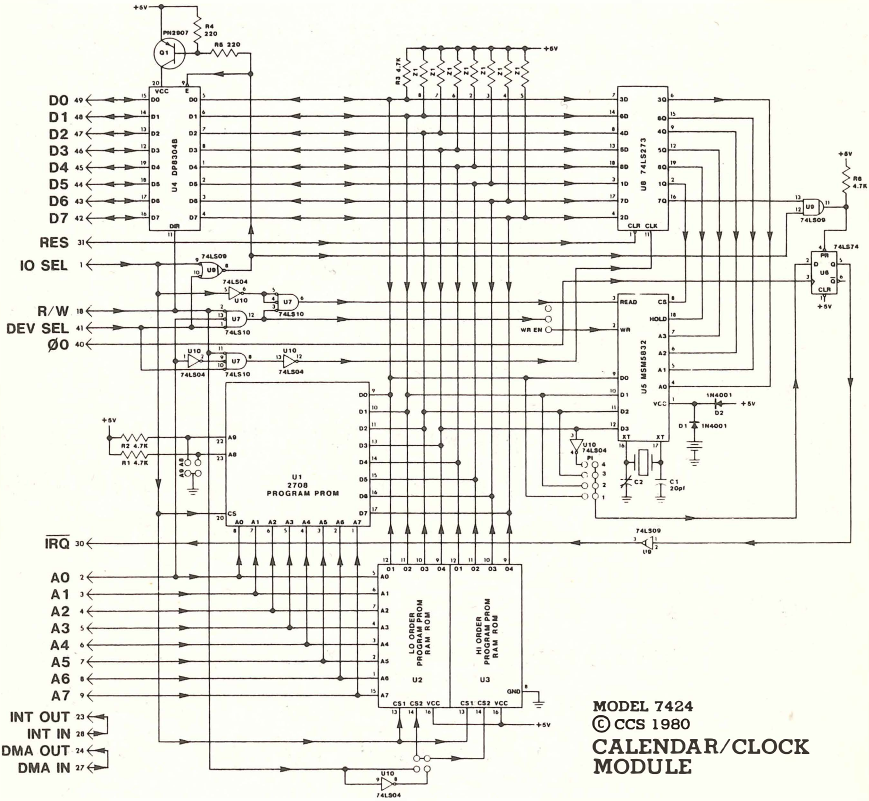
CHAPTER 4

TECHNICAL INFORMATION

4.1 SPECIFICATIONS

SIZE:	5" long x 2.75" high x .75" wide
WEIGHT:	less than 5 ounces
REQUIRED POWER:	+5V DC
CLOCK FEATURES:	Second, Minute, Hour, Day, Date, Month, Year Decimal Digits Separately Addressed and Set 12 and 24 Hour Formats Automatic Leap Year Adjustment Jumper-Selectable Periodic Interrupts 32.768 kHz Crystal Control Battery Back-up Maintains Timekeeping When System Is Powered Down
PROGRAM MEMORY:	2K bytes of PROM: Three 256-Byte Controller Programs 256 Bytes Unburned for User Program Circuitry for 256 Byte ROM/RAM
OTHER FEATURES:	Auto-Power-Down for High-Consumption DP8304B Interrupt Daisy Chain Support with Jumpered-Selectable IRQ Generation DMA Daisy Chain Pass-Through Component Silkscreen Glass Epoxy (FR-4) PC Board Gold-Plated Connector Fingers Solder Mask Both Sides of Board

4.2 SCHEMATIC/LOGIC DIAGRAM



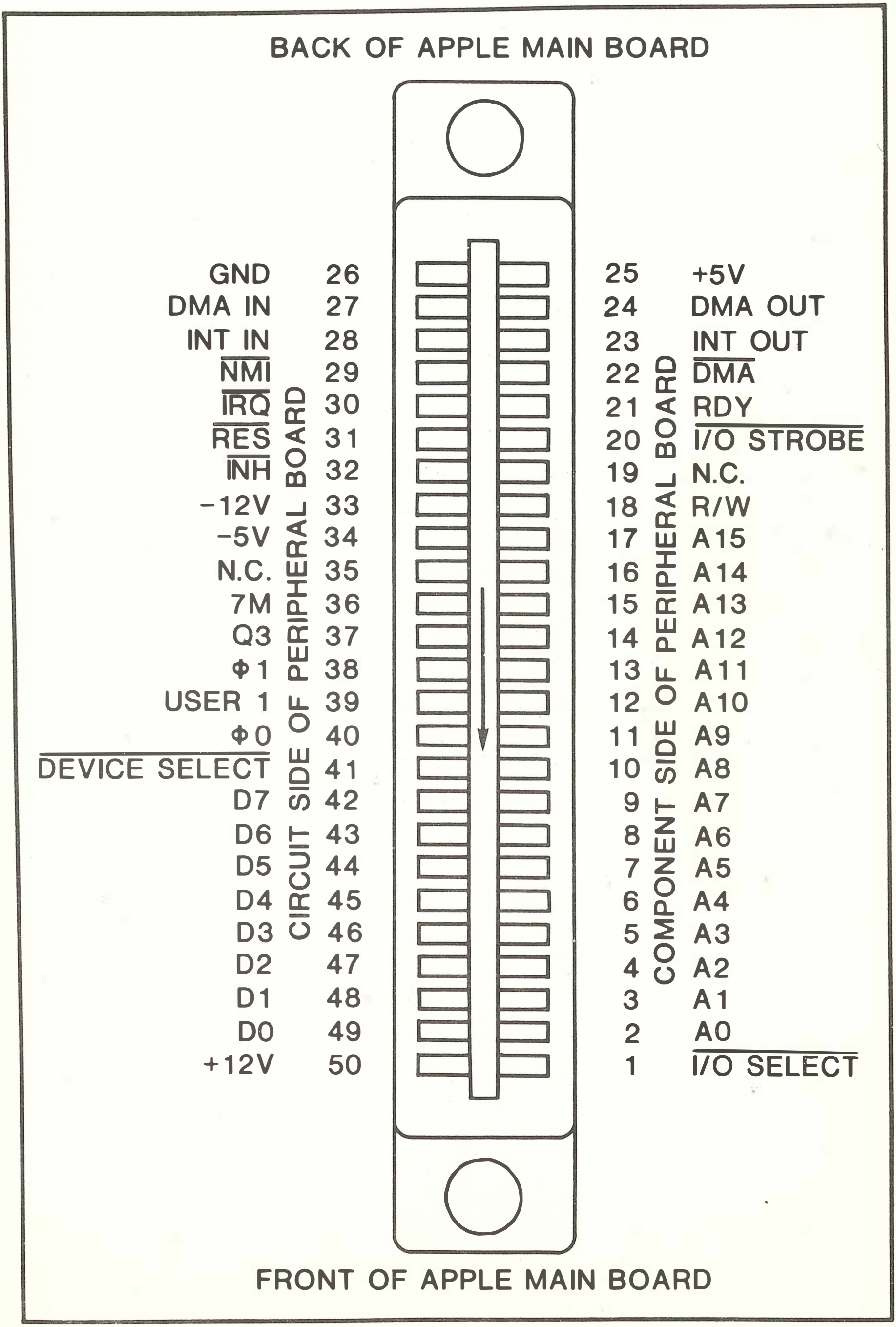
MODEL 7424
© CCS 1980
CALENDAR/CLOCK MODULE

4.3 PARTS LIST

QTY	REF	DESCRIPTION	CCS PART #
Integrated Circuits			
1	U1	2708 2K EPROM, burned	00000-07624
1	U4	DP8304B octal drvr/rcvr	30900-08304
1	U5	5832 clock/calendar	31000-05832
1	U6	74LS74 dual D flip-flop	30000-00074
1	U7	74LS10 tri 3-in NAND	30000-00010
1	U8	74LS273 oct D flip-flop	30000-00273
1	U9	74L309 quad 2-in AND	30000-00009
1	U10	74LS04 hex inverters	30000-00004
Capacitors			
1	C1	33pf, Mica, 500v, 10%	42215-53305
1	C2	5-25pf trimmer	42504-42500
1	C3	.1uf, mono, 50v, 20%	42034-21046
Resistors			
4	R1-3,6	4.7K ohm, 1/4W, 5%	40002-04725
2	R4,5	220 ohm, 1/4W, 5%	40002-02215
1	Z1	4.7K ohm x 7, sip, 20%	40930-74726

QTY	REF	DESCRIPTION	CCS PART #
Sockets			
4	XU6,7, 9,10	14 pin dip	58102-00140
2	XU2,3	16 pin dip	58102-00160
1	XU5	18 pin dip	58102-00180
2	XU4,8	20 pin dip	58102-00200
1	XU1	24 pin dip	58102-00240
Miscellaneous			
2	D1,2	Diode, IN4001-2	37000-41480
1	Q1	Transistor, PNP2907	36100-02907
1	X1	Crystal, 32.768 KHz	48033-27680
8		Hdr, 1 x 2, straight	56004-01002
1		Hdr, 1 x 3, straight	56004-01003
9		Berg jumper plugs	56200-00001
2		Battery clip	60015-00001
2		Battery cup	60015-00002
2		Bat clp insulator ring	60015-00003
2		Bat clp insulator sheet	60015-00004
4		Screw, 6-32 x 5/16 PPH	71006-32051
4		PEM nut,6-32	72606-32250
		Wire, bus, 22 AWG	51000-01220
1		PC Board, 7424 Rev A	07424-00002

4.4 APPLE II I/O CONNECTOR PINOUT

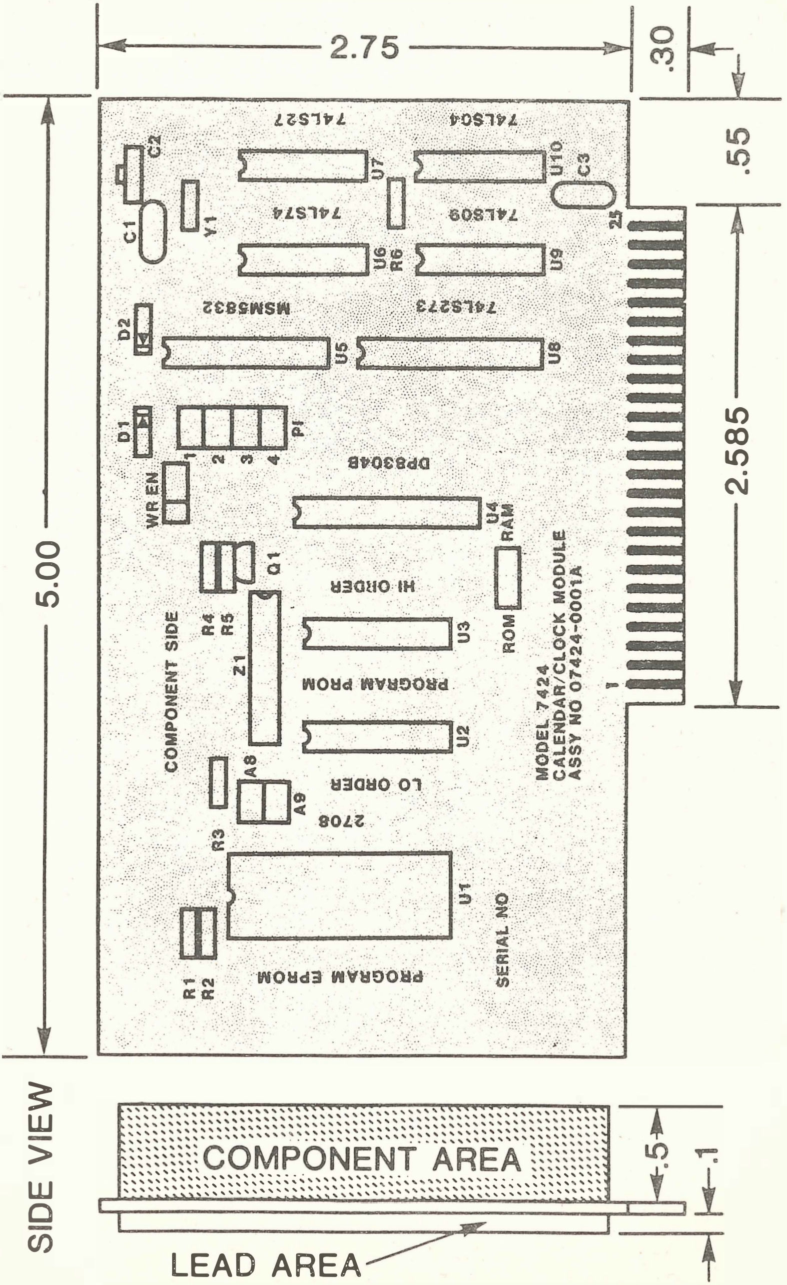


BACK OF APPLE MAIN BOARD

CIRCUIT SIDE OF PERIPHERAL BOARD			COMPONENT SIDE OF PERIPHERAL BOARD
GND	26	25	+5V
DMA IN	27	24	DMA OUT
INT IN	28	23	INT OUT
$\overline{\text{NMI}}$	29	22	$\overline{\text{DMA}}$
$\overline{\text{IRQ}}$	30	21	RDY
$\overline{\text{RES}}$	31	20	$\overline{\text{I/O STROBE}}$
$\overline{\text{INH}}$	32	19	N.C.
−12V	33	18	R/W
−5V	34	17	A15
N.C.	35	16	A14
7M	36	15	A13
Q3	37	14	A12
Φ1	38	13	A11
USER 1	39	12	A10
Φ0	40	11	A9
$\overline{\text{DEVICE SELECT}}$	41	10	A8
D7	42	9	A7
D6	43	8	A6
D5	44	7	A5
D4	45	6	A4
D3	46	5	A3
D2	47	4	A2
D1	48	3	A1
D0	49	2	A0
+12V	50	1	$\overline{\text{I/O SELECT}}$

FRONT OF APPLE MAIN BOARD

4.5 BOARD DIMENSIONS AND LAYOUT



APPENDIX A

LIMITED WARRANTY

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

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. 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 one (1) year. 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 preformed 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 orginal 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.