REAL TIME CLOCK/CALENDAR MODULE Model CLK-24C

FOR IEEE-696/S-100 COMPUTER SYSTEMS

USER'S MANUAL

DUAL SYSTEMS CORPORATION

DUAL

system reliability / system integrity

2530 San Pablo Avenue • Berkeley • CA 94702 • (415) 549-3854 • 172029 SPX

REAL TIME CLOCK/CALENDAR MODULE Model CLK-24C

FOR IEEE-696/S-100 COMPUTER SYSTEMS

USER'S MANUAL

DUAL SYSTEMS CORPORATION DUAL system reliability / system integrity

2530 San Pablo Avenue • Berkeley • CA 94702 • (415) 549-3854 • 172029 SPX

WARNING

Your Dual Systems CLK-24 non-volatile clock/calendar board is equipped with a new LITHIUM battery. This long-life battery will keep time/date intact for three years (minimum) from date of purchase. However, this battery will be RUINED if the CLK-24 board is placed on a metal surface or accidentally allowed to be shorted out by coming into contact with metal objects. There is a charge of $10.00 plus shipping for replacing of these batteries. THEREFORE, DO NOT ALLOW THE CLK-24 TO COME INTO CONTACT WITH METAL OBJECTS. WHEN THE BOARD IS NOT ACTUALLY INSTALLED IN A COMPUTER SYSTEM, IT SHOULD BE STORED IN ITS PROTECTIVE PINK ANTI-STATIC BAG.

Real Time Clock/Calendar Module
CLK-24
User's Manual

Appendices

0. INTRODUCTION

The Dual Systems model CLK-24 is an industrial grade real-time clock/calendar board that provides the date and time to any computer that meets the IEEE S-100 bus standard.

The heart of the CLK-24 is a new CMOS large-scale integrated circuit, the OKI MSM5832, which can be read by the computer to give the current time or date. The CMOS integrated circuit uses so little power that with one small battery it can keep time for over three years with the computer power off.

The CLK-24 is easy to use. The time and date may be set or read from either BASIC, assembly language, or any other language that allows access to the I/O ports. Programming examples are included in this manual which can be used as a framework for building systems using the CLK-24.

For critical timing applications or operating system requirements the CLK-24 can generate vectored interrupts on intervals of minutes, seconds, 1/64 second and 1/1024 second. This is useful if, for example, you wish to display the current time every second or minute, concurrent with normal system operation.

WARNING: Do not place the CLK-24 directly on a metal surface. To do this will result in a short circuit which will damage the battery, and void the warranty.

0-1. SPECIFICATIONS

FUNCTION: Precision real time clock and calendar with on-board battery backup.

BUS INTERFACE:
Plugs into virtually any S-100 type computer. Meets the IEEE 696/S-100 standard. Compatible with GODBOUT COMPUPRO, NORTH STAR HORIZON, IMSAI, ITHACA INTERSYSTEMS, IMS, and CROMEMCO , as well as the new 16 bit systems (e.g. Dual Systems CPU/68000.)

IEEE 696 BUS COMPATIBILITY:
Slave (D8) (I8) VI F5 T60 W (SH)

TIME KEEPING FUNCTIONS:
Provides current year, month, day, day of week, AM/PM, hour (12 or 24 hour format), minute, and second. Also provides jumper selectable interrupts every minute, second, 1/64th second or 1/1024th second. Each of these intervals can be jumpered to any of the eight interrupt levels, or NMI.

ACCURACY: ± 50 seconds per month. Controlled by 32.768 KHz tuning fork resonator with accuracy, in normal use, of better than ± 20 parts per million. On board trimmer is provided to adjust the clock accuracy.

ADDRESSING:
The CLK-24 is I/O mapped. It can be accessed by means of the 8080/8085/Z-80 IN and OUT instructions, or by the BASIC INP and OUT instructions. The board uses two consecutive I/O addresses which can be switch-selected to lie on any even boundary in the 256 byte I/O address space.

OPERATING POWER:
Operates from the +8 volt (nominal) line on the S-100 bus. Usable operating range is +7.5 to +14 volts. Current consumption is less than 250 mA during operation (typically 175mA).

STANDBY POWER:
Supplied by a 3-10 year sealed 3.6 volt lithium battery on the CLK-24 board. Current consumption is 30 uA maximum, 13 µA typical.

TIMEKEEPING DURATION WITH COMPUTER POWER OFF:
With factory supplied battery: three years minimum between battery replacements.

1. BOARD SETUP

1-1. UNPACKING:

Carefully remove the board from its protective antistatic bag. **Do not place the CLK-24 on any metal surface, as this may accidentally discharge the battery.**

1-2. PORT ADDRESS SELECTION:

The CLK-24 has two registers, the data register and the Control Register. Each is accessed through an I/O port. Notice that there is only one I/O address switch on the board. This switch sets the data port address. The Control port address is always the data port address + 1. The switch sets the data port to any even address (eg. 0, 2, 4, .. 252, 254). For example, if the data address is 240 (as delivered) the Control address is 241.

Before installing the CLK-24 it is important to make sure that the two addresses selected are not used by any other part of the computer system. Check the serial I/O, disk I/O, bank select, extended addressing port, and any other I/O mapped devices in your system to be sure the addresses do not conflict.

The CLK-24 is factory configured with a data port address of 240 decimal (F0 Hex) which should be compatible with most standard computer systems. To change this address it is necessary only to change the DIP switch settings on the board. Each switch selects one address bit (A7 to A1). When a switch is "ON" (UP position), that address bit is a logical 0. When a switch is "OFF" (DOWN), that bit is a logical 1. Figure 1 shows the arrangement and values of these switches.

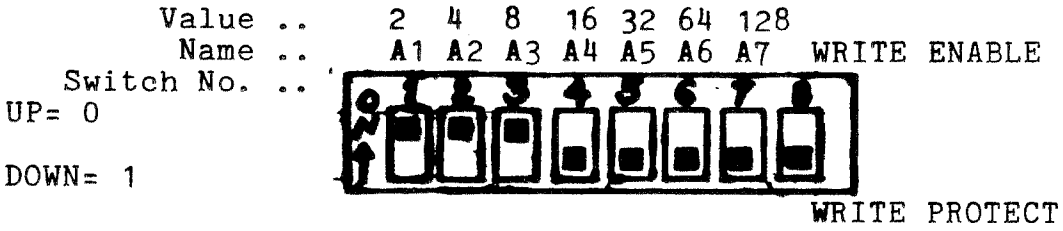


FIGURE 1.

Adding up the values of all the switches set for logical 1 gives the value of the base address. Example: **The board is** delivered with the switches set as shown in Figure 1. **The** base address is then 128 + 64 + 32 + 16 = 240 decimal.

1-3. WRITE PROTECT SWITCH:

The position 8 paddle on the DIP switch shown in Figure 1 is the Write Protect Switch. When the switch is in the UP (on) position, the CLK-24 will allow the time & date to be reset or changed. When the switch is in the DOWN (off) position the previously set time is protected from being changed. **To be safe, always keep the Write Protect Switch in the protect mode (DOWN), except when changing the time.**

1-4. INSTALLING THE CLK-24:

Once the data address has been set, the CLK-24 may be installed into the computer system. **The power MUST BE TURNED OFF, and left OFF for at least 30 seconds.** and then the board may be placed into any unused slot in the S-100 card cage. Be sure the gold tabs on the board are lined up with the socket on the motherboard and then carefully press the board all the way into place. **Be sure the battery does not accidentally touch the adjacent board.** If everything appears correct, turn the computer on.

2. HOW TO USE THE CLK-24

The time is read from or written to the CLK-24 one digit at a time. The Control port is used to set the mode of the CLK-24 and select which digit is to be read or written. The digits are read or written through the data port.

To READ a given digit, simply OUTPUT the Control Code for that digit to the Control Register and INPUT the digit from the data port.

To WRITE a given digit, simply OUTPUT the digit to the data port , then OUTPUT the Write Control Code (Control Code + 16) for that digit to the Control Register, lastly OUTPUT the CLEAR Code to the Control Register.

The following algorithm READS the time from the CLK-24:

1) OUTPUT Control Code for the desired digit (e.g. seconds units, seconds tens, etc.) to the Control port.
2) INPUT the digit from the data port.
3) REPEAT Steps 1 and 2 until all digits are read then OUTPUT the CLEAR Code.

The following algorithm WRITES the time to the CLK-24:

1) OUTPUT the desired digit to the data port.
2) OUTPUT the Control Code plus the Write Code to the Control port. (The Write Code is 16.)
3) OUTPUT the CLEAR Code to the Control port after EACH digit is written.
4) REPEAT Steps 1 through 3 until all digits are written.

The INPUT and OUTPUT functions for the CLK-24 can be executed using the appropriate input-output instructions for I/O ports. These are the INP and OUT instructions in BASIC, the IN and OUT instructions in 8080 assembly language, or the appropriate MOVE instructions in 68000 assembly language.

There is a restriction on the minimum amount of time which must elapse between setting the Control Port and reading the Data Port. This restriction is usually satisfied in high level languages, under a normal instruction set; but, in assembly language it might be necessary to program this delay. Please refer to Section 6 for further details.

The clock chip requires 150 µSec to either read or set the time. The CLK-24 incorporates circuitry to automatically suspend the operation of the computer for this period. The user is then unburdened of this task. See Section 6 for further details.

2-1. CONTROL CODES AND SPECIAL FUNCTIONS:

The Control Register Codes given below are used to select which digit of the date or time is to be read or written. Control Codes are also used to select the read or write mode. Table 1 summarizes these Contol Codes and Special Functions. Refer to Section 6-3 for an explicit description.

#	DIGIT	WHEN READING	WHEN WRITING
0	1 SECOND		
1	10 SECONDS		
2	1 MINUTE		
3	10 MINUTES		
4	1 HOUR		
5	10 HOURS	"AND" DIGIT WITH 3	ADD 4 FOR PM, 8 FOR 24 HOUR FORMAT.
6	DAY OF WEEK*		
7	1 DAY		
8	10 DAYS	"AND" DIGIT WITH 3	ADD 4 FOR LEAP YEAR
9	1 MONTH		
10	10 MONTHS		
11	1 YEAR		
12	10 YEARS		
16	WRITE		
64	CLEAR		
128	HOLD DISABLE (OPTIONAL)		
143	ENABLE INTERRUPTS		

* 0 is SUNDAY ... 6 is SATURDAY

TABLE 1. Control Register Codes, & Special Functions.

2-1-1. CONTROL CODES:

Recall that the Control Code must be sent to the Control Register for every digit, either before the digit can be read from the data port, or after the digit has been written to data port. The Control Code for the read functions is simply the Control Code for that particular digit. (e.g. reading the minutes digit of the time simply requires the Control Code to be 2.) To derive the Control Code for writing a digit, simply add the Control Code for the Write function to the Control Code for that particular digit. (e.g. setting the minutes digit of the time requires the Control Code to be 2+16 = 18.)

The Days of the week are encoded as follows: 0 is Sunday, 1 is Monday, ... , 6 is Saturday.

2-1-2. SPECIAL FUNCTIONS:

The CLK-24 provides the following Special Functions: Selection of either the AM/PM format or the 24 Hour format; and, recognition of a leap year. These functions are encoded in the data for two digits. These are the tens of Hours and tens of Days digit. Table 1 summarizes these functions.

The '10 Hours' and the '10 Days' digits serve multiple purposes. The additional information contained in the '10 Hours' digit is for selecting the 12 hour AM/PM format or the 24 hour format. The additional information contained in the '10 Days' digit is for selecting a leap year.

When INPUTTING the data for both of these digits 'AND' the data with 3, to mask out the extra information, making the value for that digit useable.

To determine whether the time is in the AM or the PM, the 12/24 hour data must be masked out. This is done by 'AND'ing the input data for the '10 Hours' digit with 7 when the data is first input, before doing anything else. Now if the remaiaining value

is greater than 3, the time is in the PM, otherwise it is in the AM. Next, to get the value of the '10 Hours' digit simply mask out all extra information by 'AND'ing the remaining data with 3, as above.

To determine whether the year is a leap year, simply compare the unmasked '10 Days' digit to 3. If it is greater than 3, then the year is a leap year, otherwise it isn't a leap year. Next, just 'AND' the data with 3, to mask out the leap year information.

When setting the time, to select the AM/PM format, 'ADD' four (4) to the '10 Hours' digit before OUTPUTTING it to the CLK-24. To select the 24 hour format, 'ADD' eight (8) to the '10 Hours' digit before OUTPUTTING it to the CLK-24. To select a leap year, add four (4) to the '10 Days' digit before OUTPUTTING it to the CLK-24. Following are some examples of how to use the special functions provided with these digits:

- to set the time such that the 10's of hours digit is 1, say 1:22 PM, the digit must be set with 1+4 = 5; but to set the time to 1:22 AM, the digit must simply be set with 4. The Control Code in both cases would be 5+16 = 21.

- to set the date to, say 1-21-82 (not a leap year), the 10's of days digit must simply be set with 2+0 = 2; but to set the date 1-21-84 (a leap year), this digit must be set with 2+4 = 6. The Control Code in both cases would be 8+16 = 24.

2-2. CLEAR CONTROL CODE

Most clocks that read one digit at a time have a potential source of error. Suppose it is 1:59. Your program reads the clock starting with 10's of hours ending with 1's of minutes. Suppose the time changes from 1:59 to 2:00 after the 1's of hours digit has been read. The time is now 2:00 and the program will procede to read the 10's and 1's of minutes. Both of those digits will now be 0. Thus the program will read the incorrect time of 1:00.

The CLK-24 has been designed to eliminate this problem.

Circuitry on board latches the time so that whenever you read the clock, the time is stable for 1/2 second. This does not affect the accuracy of the clock in any way. All readings within the 1/2 second will be consistent. After the 1/2 second the clock is again allowed to advance. If you wish to read the time more than once per second, then you must signal to the CLK-24 to update the time out of sequence. (You want the clock to advance BETWEEN complete date/time readings.) This is done with the CLEAR Control Code. If the CLEAR Control Code is not used, subsequent time readings within the 1/2 second interval will yeild the same result. The following algorithm will accurately read the time, even at a rate of more than once per second:

1) READ the entire time once.
2) OUTPUT CLEAR Control Code;
3) Jump to 1; (perform next entire read)

Similarly, most clocks that set time one digit at a time also have a potential source of error. Since many digits may need to the set, either one of the following extremes could cause the time to be incorrectly set:

1) If the time setting routine is too slow, then it is conceivable that the desired time will have advanced before the routine is finished setting the previously desired time.
2) If the time setting routine is too fast, then it is likely that the routine will not allow enough time for the actual write function to be completed.

The CLK-24 has been designed to eliminate these problems also. The CLEAR Control Code is very useful here because it produces an appropriate delay for the completion of a write function to the clock. The following algorithm will provide an optimum time setting procedure:

1) Prepare and OUTPUT next digit to Data Port;
2) OUTPUT the appropriate value to the Control Port;
3) OUTPUT the CLEAR Control Code;
4) Jump to 1; (prepare and 'write' next digit)

2-3. TIME SETTING PROTECTION

The CLK-24 has four features to prevent accidental writing of the time. The rightmost DIP switch (#8) is labelled WRITE ENABLE. When this switch is ON (up) the clock can be set. When it is OFF, the board is write protected. **We suggest that you leave the board in the WRITE PROTECTED (switch OFF) position all the time except on the rare occasions that the time must be set.**

Write errors are usually caused by microprocessors which execute random programs as the power falls (either when the computer is turned off or during a power fail). The CLK-24 is designed so that data can be written to the clock only AFTER the proper sequence of instructions are executed. This means that a single random OUT instruction cannot affect the clock, even if the clock is left with the WRITE PROTECT switch not in the protect mode. (See Section 1.3)

In addition, the CLK-24 monitors the S-100 power supply and detects when the power starts to drop. It also monitors the PWRFAIL* (powerfail) line on the bus which will go active when the power is about to fail (if your computer has this new IEEE-696/S-100 feature). When any of these conditions occur, the CLK-24 board is deselected.

3. PROGRAMMING IN BASIC

3-1. READING THE TIME IN BASIC

The following BASIC fragment will read the day of week (0 for Sunday to 6 for Saturday), and store this number in the variable "DAY".

```
100  OUT  241,6              :REM 6 IS CONTROL CODE FOR DAY OF WEEK
110  DAY = INP(240)          :REM READ DATA
```

This and all programming examples assume that the CLK-24 is configured with a base address of 240 (see section 1-2). In addition, all examples in BASIC are written in Microsoft BASIC-80 for CP/M.

Note that 10's of hours and 10's of days serve additional functions (i.e. AM/PM,leap year, etc.). The extra bits must be masked (removed) when the digit is read. For example, this program fragment reads the hours:

```
100 OUT 241,4              : REM SET CONTROL REG. FOR 1'S OF HOURS
110 UHOUR = INP(240)       : REM READ 1'S OF HOURS
120 OUT 241,5              : REM SET CONTROL REGISTER FOR 10'S HRS
130 THOUR = INP(240) AND 3  :REM STRIP AM/PM BIT
140 PRINT "HOURS= "; THOUR; UHOUR
```

The BASIC "AND" statement here removes all but the two least significant bits of the digit from the data port.

Some versions of BASIC do not provide the AND statement so you must use the INT function or the IF statement to do the same work. To remove the format and PM bits from the 10's of hours you can execute:

```
100  THOUR=THOUR - (4*INT(THOUR/4)) :REM MASK ALL BUT 2 LOW BITS
```

(The above is identical to THOUR=THOUR AND 3.)

If you know that only single extra bit may be set, then you can check for that bit explicitly. For example, if you read the 10's of days and want to remove the leap year bit then you can execute:

```
100 IF TDAY > 3 THEN TDAY = TDAY - 4    :REM REMOVE LEAP YEAR BIT
```

In most applications, however, you will be reading more than one digit at a time. A loop is a more efficient way to do this. For example, the following program reads the time and prints it out as fast as possible:

```
1000 CR=241 : DR=240              :REM DEFINE PORT ADDRESSES
1010 DIM T(6)                     :REM ARRAY TO HOLD THE DATA
1020 FOR I=0 TO 5                 :REM LOOP FOR ALL DIGITS
1030 OUT CR,I                     :REM SEND CONTROL CODE FOR DIGIT
1040 T(I)=INP(DR)                 :REM READ DIGIT FROM THE DATA PORT
1050 NEXT I
1060 OUT CR,64                    :REM SEND "CLEAR" CONTROL CODE
1070 T(5) = T(5) AND 3            :REM STRIP AM/PM BIT FROM HOURS
1080 PRINT T(5);T(4);":";T(3);T(2);":";T(1);T(0) :REM PRINT TIME
1090 GOTO 1020                     :REM DO AGAIN
```

Note that we have used the algorithm as described in section 2-2. Since this program will be reading the time faster than once per second, we output a CLEAR Control Code to the Control Register so that the next reading of the CLK-24 will yield a more recent time. The command used to send the CLEAR Control Code is:

```
1060      OUT  CR,64
```

Again, this is only needed in programs which read the time more than once per second, such as programs which repeatedly read the time while waiting for a certain reading.

3-1-1. SAMPLE READ PROGRAM:

The following is a more complete program which determines the time, date, and day of week. It also uses the status bit to determine AM/PM. In most programs which do any extensive processing of the data it is useful to store the data in a string or array before further processing. Storing the data simplifies the program structure and also has another advantage: If the clock is read continuously for more than 1/2 second then there is some chance that the time will change during the reading (see section 2-2.). If you store the data directly into an array then you can be sure that all digits will be read in less than 1/2 second regardless of what other processing or printing is done.

```
 1000 DIM D(12)                      :REM  AN ARRAY FOR TIME AND DATE
1010 CONTROL=241:DTA=240             :REM  SET NAMES OF CONTROL AND DATA PORT
1020 FOR I=0 TO 12                   :REM  LOOP FOR ALL 12 DIGITS OF INFO
1030 OUT CONTROL,I                   :REM  SET UP TO READ I,TH DIGIT OF DATA
1040 D(I)=INP(DTA)                   :REM  READ DIGIT AND SAVE IN ARRAY
1050 NEXT I
1055 OUT CONTROL,64                  :REM SEND "CLEAR" CONTROL CODE
1060 D(5)=D(5) AND 7                 :REM  STRIP 12/24 HR BIT (BIT 3)
1070 IF D(5)>3 THEN M$="PM" ELSE M$="AM" :REM CHECK BIT 2 FOR AM/PM
1080 D(5)=D(5) AND 3                 :REM  STRIP AM/PM BIT (BIT 2)
1090 D(8)=D(8) AND 3                 :REM  STRIP LEAP YEAR BIT
1100 P$=""                           :REM  SET PRINT STRING TO NULL
1110 FOR I=0 TO 12                   :REM  LOOP THROUGH ALL DIGITS
1120 P$ = HEX$(D(I)) + P$            :REM CONVERT TO ASCII AND ADD TO STRING
1130 REM  NOTE THAT EACH NEW CHAR IS ADDED AT LEFT END OF STRING
1140 NEXT I
1150 FOR I= 0 TO D(6)                :REM  COUNT UP TO DAY OF WEEK
1160 READ DAY$                       :REM  READ NAME OF EACH DAY
1170 NEXT I                          :REM  EXIT WHEN WE REACH THE RIGHT DAY
1180 REM  NOW PRINT DAY    MONTH/DAY/YEAR
1190 PRINT DAY$, MID$(P$,3,2) + "/" + MID$(P$,5,2) + "/" + LEFT$(P$,2),
1200 REM  NOW PRINT     HOUR:MIN:SEC   AM/PM
1210 PRINT MID$(P$,8,2) + ":" + MID$(P$,10,2) + ":" + RIGHT$(P$,2), M$
1220 RESTORE                         :REM  RESET READ DATA
1230 GOTO 1020                       :REM  READ DATA AND TIME AGAIN
1240 DATA "SUN","MON","TUES","WED","THURS","FRI","SAT"
```

3-2. SETTING THE TIME FROM BASIC

To set the day of week to Monday, use the following BASIC fragment:

```
100 OUT 240, 1          :REM SEND "MONDAY" TO DATA PORT
110 OUT 241, 16 + 6         :REM SEND TO CONTROL  "WRITE" (16) +
                                CODE FOR DAY OF WEEK (6)
120 OUT 241, 64         :REM SEND CLEAR CODE TO CONTROL PORT
```

Note the use of the CLEAR Control Code.

3-2-1. SAMPLE WRITE PROGRAM

The following program sets the time and date. Data statements and read statements are used for the prompts. Data is read from the terminal one digit at a time and stored in an array. Then a loop is used to send all the data to the clock.

```
10  CONTROL=241:DTA=240             :REM PORT ADDRESSES
20  DIM SET(12)                     :REM ARRAY TO HOLD DATA
30  FOR I= 2 TO 12                  :REM LOOP FOR ALL DIGITS BUT SECS
40  READ MES$                       :REM READ A PROMPT
50  PRINT MES$;                     :REM PRINT IT
60  INPUT SET(I)                    :REM INPUT ONE DIGIT
70  NEXT I
80  INPUT "AM? (Y OR N)",A$         :REM ASK IF AM OR PM
90  IF A$="N" THEN SET(5)=SET(5)+4 :REM SET PM BIT TO HOURS DATA
100 FOR I= 0 TO 12                  :REM LOOP TO WRITE ALL DIGITS
120 OUT DTA,SET(I)                  :REM SEND DIGIT TO DATA PORT
130 OUT CONTROL,16+I                :REM SEND WRITE CODE + CONTROL CODE
140 OUT CONTROL,64                  :REM SEND "CLEAR" CODE
150 NEXT I
160 DATA "MIN 1","MIN 10","HR 1","HR 10"  :REM DATA FOR TIME PROMPTS
170 DATA "DAY OF WK? (SUN=0 .. SAT=6)"    :REM DAY OF WEEK PROMPT
180 DATA "DAY 1","DAY 10","MO 1","MO 10","YR 1","YR 10" :REM DATE PROMPTS
```

4. PROGRAMMING IN ASSEMBLY LANGUAGE

4.1 SAMPLE PROGRAM IN 8080/8085/Z-80 ASSEMBLY LANGUAGE:

The following program reads a pair of digits from the CLK-24, converts the two digits from hex to BCD digits, then stores each pair of digits in a memory location as shown in the table below.

DATA	DEST.	CLOCK FUNCTION CODES
DAY	11H	7,8
YEAR	12H	11,12
HOURS	13H	4,5
MINUTES	14H	2,3
SECONDS	15H	0,1

TABLE 2.

These locations are unused by the CP/M operating system and are accessed by Sorcim's Pascal/M for the time functions. Hopefully they conform to some kind of standard.

The only critical timing consideration when programming the CLK-24 in assembly language is to be sure that you **allow at least 6 microseconds between outputting to the Control port and inputting a digit from the data port.**

```
        ORG     0100H
DATAR:  EQU     0240D           ; ADDRESS OF DATA REGISTER
CONTR:  EQU     0241D           ; ADDRESS OF CONTROL REGISTER

SECS:   MVI     B,00H           ; CLOCK FUNCTION FOR SECONDS
        LXI     H,0016H         ; MEMORY LOCATION FOR SECONDS + 1
        CALL    PAIR            ; GET SECONDS
REST:   CALL    PAIR            ; READ MINUTES
        CALL    PAIR            ; READ HOURS
        ANI     03FH            ; MASK PM AND 24 HR BITS
        MOV     M,A             ; SAVE CORRECT DIGITS
        MVI     B,11D           ; FUNCTION CODE FOR YEARS
        CALL    PAIR            ; READ YEARS
        MVI     B,7D            ; FUNCTION CODE FOR DAYS
        CALL    PAIR            ; READ DAYS
        ANI     03FH            ; MASK LEAP YEAR BIT
        MOV     M,A             ; SAVE CORRECT DIGITS
        CALL    PAIR            ; READ MONTHS
        RET                     ; DONE
```

```
*****************************************************************************
*                                                                           *
*     'PAIR' READS A PAIR OF DIGITS FROM THE CLOCK BOARD AND  RETURNS       *
*   WITH THE TWO DIGITS (IN BCD) IN THE ACUMULATOR.  THE ROUTINE ALSO       *
*   DECREMENTS THE HL PAIR AND THEN STORES THE PAIR OF DIGITS IN            *
*   MEMORY POINTED TO BY HL.  UPON ENTRY, THE B REG SHOULD HAVE THE         *
*   CLOCK CONTROL CODE FOR THE FIRST DIGIT.  THE HL PAIR SHOULD HAVE        *
*   ONE MORE THAN THE DESTINATION IN MEMORY OF THE PAIR OF DIGITS.          *
*   THE HL PAIR WILL BE DECREMENTED SO THAT UPON EXIT, IT WILL POINT        *
*   TO WHERETHE  DATA WENT.  ALSO, THE B REGISTER IS INCREMENTED            *
*   TWICE TO ANTICIPATE THE CLEAR CLOCK CONTROL CODE.                       *
*                                                                           *
*****************************************************************************

PAIR:   MOV     A,B             ; GET CLOCK CODE IN A
        OUT     CONTR           ; SET UP CLK-24 FOR READING DIGIT
        NOP                     ; WAIT FOR DATA VALID (6 USEC)
        NOP
        NOP
        NOP                     ; NEXT STATEMENT ALSO KILLS TIME
        INR     B               ; PREPARE FOR NEXT READ
        IN      DATAR           ; READ DIGIT
        MOV     C,A             ; SAVE DIGIT FOR LATER
        MOV     A,B             ; GET CLOCK CODE INTO A
        OUT     CONTR           ; SEND CONTROL CODE
        NOP                     ; ANOTHER DELAY
        INR     B               ; ANTICIPATE NEXT READ
        DCX     H               ; PREPARE FOR NEXT DESTINATION
        IN      DATAR           ; READ HIGH DIGIT
        RLC                     ; ROTATE HIGH DIGIT INTO HIGH 4 BITS
        RLC
        RLC
        RLC
        ORA     C               ; MERGE HIGH AND LOW DIGITS
        MOV     M,A             ; STORE INTO MEMORY
        RET
```

5. USING INTERRUPTS

Interrupts are used when sub-second precision is needed from the clock, or for real-time applications where polling (checking the clock continously until the right time is reached) is not appropriate. The CLK-24 can provide interrupts at 4 different rates: once every minute, every second, every 1/64th second or every 1/1024th second (.97 millisecond). The correct interval should be jumpered to the desired interrupt level. Pins are provided for connection to the S-100 interrupts VIO-VI7, as well as NMI. These pins may be found at the bottom left of the board, below U9 and U10. The "MIN" jumper has a dual function. It will provide either a 1 minute interrupt, or a 1/64 second interrupt, depending on the position of jumper J5, shown in figure 2b. Method of connection is demonstrated by the example in figure 2a.

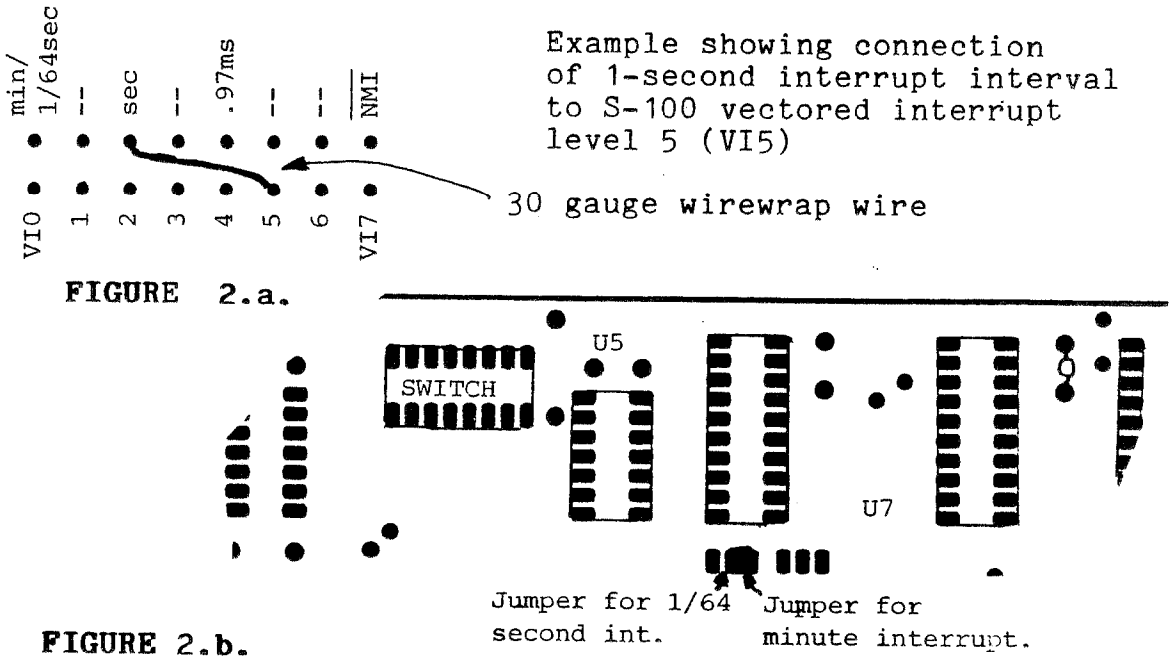


FIGURE 2.a.

FIGURE 2.b.

The use of interrupts is tied to the characteristics of the host CPU, and it is necessary to have some understanding of the behavior of the CPU on receipt of an interrupt request. When the interrupt is received, the processor finishes the current instruction, saves the current address (PC), and jumps to a specific memory location. Some sort of interrupt service routine **must** previously have been stored at that location.

The exact behavior, and especially the memory location to which execution jumps, depends on the CPU. Two examples are provided in the table below:

S-100 int	Host Processor Z-80	68000
VI7	jump to 38 (hex)	not supported
VI6	" " 30	not supported
VI5	" " 28	(IP1) jump to 64 (hex)
VI4	" " 20	(IP2) " " 68
VI3	" " 18	(IP3) " " 6C
VI2	" " 10	(IP4) " " 70
VI1	" " 8	(IP5) " " 74
VI0		(IP6) " " 78
NMI	jump to 66 (hex)	(IP7) " " 7C

TABLE 3.

Note that the space provided for each interrupt level is insufficient to hold an entire interrupt service routine. They should only contain a jump to the proper routine. For the Z-80, the actual machine code for the JMP instruction must be provided, along with the address. For the 68000 only the address (4 bytes) is to be supplied. The NMI is different from the other interrupts in that it is non-maskable (i.e., may not be inhibited). The CP/M operating system uses location 66, making NMI unavailable for those systems which use CP/M.

It is the responsibility of the interrupt service routine to make sure that any registers being used by the interrupted program are preserved, so that once the service routine is exited, execution can be resumed by the main program as if there had been no interrupt (the main program should not be aware of the occurrence of interrupts). The most common technique is to push everything that must be preserved onto the stack. It is very important, however, to make sure that everything pushed onto the stack is popped before the service routine is exited, since the information needed by the processor to return is generally on the stack as well.

To enable the CLK-24 interrupts, once the proper jumpers have been installed on the board, output the code 143 (8F hex) to the Control port. The service routine should acknowledge the interrupt (clear the interrupt line) by outputting the code 142 (8E hex) to the Control port. (See section 2-1 for details on the Control Codes).

The following sample program will demonstrate simple use of the CLK-24 interrupts. All that is done in this example is to increment a counter each time an interrupt is received. A possible use of this routine would be to time the execution of a program. The program would start up the clock, execute, and then, on termination, read the counter to see how many interrupts had occurred. The time would be the number of interrupts times the interrupt interval.

The following program is presented in two versions, one in 68000 assembly language, and the other in 8080 assembly (for use on the 8080 or Z-80). In each case, the clock's interrupt level has been jumpered to the S-100 VI5 pin, as in the example in the beginning of this section (see Figure 2).

5-1. SAMPLE INTERRUPT PROGRAM IN 68000 ASSEMBLY LANGUAGE:

```
*
*     CLK-24 interrupt service example (68000 processor)
*
*     ***INIT*** should be called when the interrupts are to be
*     started up, probably at the beginning of the program
*
*
IOPAGE     EQU $FFFF0000        * I/O locations for DUAL CPU/68000
CTL        EQU 241 + IOPAGE     * Clock control port
DATA       EQU 240 + IOPAGE     * Clock data port
ENINT      EQU $8F              * Enable interrupt code
DISABLE    EQU $8E              * Interrupt acknowledge code
VECTOR     EQU $64              * Interrupt jump location: 64 hex for
                                * 68000 IP1 (S-100 interrupt VI5)

*
*     ***INIT*** initializes the clock and the counter
*
INIT       MOVE.W    #0, COUNT       initialize counter to 0
           MOVE.L    #INTACK, VECTOR  store addr of service routine
           MOVE.B    #ENINT, CTL     enable interrupts
           JMP       MAIN            jump to main program

COUNT      DS.W      0               define storage area for count

*
*     ***INTACK*** services the interrupt from the CLK-24
*
INTACK     MOVE.B    #DISABLE, CTL   disable interrupts
           ADDQ.W    #1, COUNT       increment counter
           MOVE.B    #ENINT, CTL     enable clock interrupts
           MOVE.W    #0, SR          reset CPU interrupt mask
           RTS                       and return to caller
```

5-2. SAMPLE INTERRUPT PROGRAM IN Z-80 ASSEMBLY LANGUAGE:

```
*
*     CLK-24 interrupt service example (Z-80 processor)
*
*     ***INIT*** should be called when the interrupts are to be
*     started up, probably at the beginning of the program
*
*
DATA       EQU  241        ; clock data register
CTL        EQU  240        ; clock control register
ENINT      EQU  08FH       ; enable interrupt code
DISABLE    EQU  08EH       ; interrupt acknowledge code
VECTOR     EQU  028H       ; jump location for level 5 interrupt
JUMP       EQU  0C3H       ; machine code for JMP instruction

*
*     ***INIT*** initializes the clock and the counter
*
INIT:      XRA  A
           STA  COUNT          ; initialize counter to zero
           MVI  A,JUMP         ; load code for jump
           STA  VECTOR         ; store
           LXI  INTACK         ; load addr of service routine
           SHLD VECTOR+1       ; and store
           MVI  A,ENINT
           OUT  CTL            ; enable clock interrupts
           EI                  ; enable interrupts
           RET

COUNT      DW   0              ; define storage area for count

*
*     ***INTACK*** service the interrupt from the CLK-24
*
INTACK:    PUSH PSW            ; save status
           MVI  A,DISABLE
           OUT  CTL            ; acknowledge clock interrupt
           LDA  COUNT          ; get counter
           INR  A              ; increment
           STA  COUNT          ; and store
           MVI  A,ENABLE
           OUT  CTL            ; reenable clock interrupts
           POP  PSW            ; get status back
           RET
```

6. THEORY OF OPERATION

This section discusses, in more technical detail, the operation of the CLK-24. The information contained within will be useful for determining the Programming approach for applications where timing restrictions are strict.

6-1. CONTROL REGISTER:

The Control Register sets the mode of the CLK-24. It has the following bit definitions:

CONTROL REGISTER
(switch address+1)

D7	—	hold disable
D6	—	clear
D5	—	x
D4	—	write
D3	—	B3
D2	—	B2
D1	—	B1
D0	—	B0

"X" means "unused"

FIGURE 3.

READ/WRITE Bit:

When this bit is HIGH then the board WRITES the digit in the data latch to the clock. When this bit is LOW then the board is set for READING.

HOLD DISABLE Bit:

The hold disable bit allows more direct control over the clock board for faster reading. It disables the HOLD pin on the clock chip and disables the automatic wait state generation needed for the relatively SLOW clock chip.

The use of this feature is only suggested for assembly programs which are interrupt driven or for systems which cannot tolerate the 150 µSec wait states. Most assembly language programs should access the CLK-24 just as the BASIC programs outlined above.

The clock chip has a HOLD pin which keeps the data from changing between successive reads. Whenever the HOLD pin is set the clock chip requires a 150 µSec setup time before any other action can take place. Normally the CLK-24 board takes care of all timing on this pin (With the help of the CLEAR instruction.) The HOLD DISABLE bit is included to allow user programs to override the automatic HOLD generation and wait states. This should only be needed for interrupt driven programs where high speed is important.

When the Control Code is sent to the clock, there must be 6 uSec for the data to stabilize before attempting to read it. (see clock chip data sheet included). This is only a factor in assembly language.

Remember, there is some chance that the time will change between digit readings. Therefore, if you must use the HOLD DISABLE bit, the recommended procedure is to store all the read data in an array. Then read the digits again and compare them to the corresponding digits in the array. If there is a disagreement then start the procedure over. (See BASIC program in appendix A.)

After an interrupt the time is guaranteed to be stable for .5 mSec. If your interrupt response time is less than .5 mSec then the time can be read directly with no danger of error.

6-2. TIMING:

The clock chip HOLD pin keeps the clock from advancing when it is held high. This pin must be brought low at least once per second for at least 150 µSec to allow the clock to advance properly. Circuitry on the CLK-24 takes care of setting and resetting the HOLD pin automatically. Wait states are inserted by the CLK-24 during the 150 µSec interval.

6-2-1. READ CYCLE:

Assume the clock has not been accessed for **at least** one second. The clock HOLD pin is low and the clock advances normally. When a read command is received (output to Control port with read/write bit low) the HOLD pin is brought high. The CLK-24 generates wait states for 150 µSec to allow the HOLD pin to set up.

A read from the data port then reads directly from the clock chip. As the HOLD pin goes high, a monostable is set which will stay on for 1/2 second. Subsequent reads do not reset either the hold pin or the 1/2 second monostable. Note that we do assume that **the Control port is written to at least 6 µSec before the data is read.** This is a valid assumption in a high level language, because the instruction execution time for the high level languages is long enough to automatically program the needed delay; but, **in assembly language with a fast CPU it may be necessary to program this delay.** When 1/2 second has passed or the CLEAR command is executed, the monostable turns off. This brings down the HOLD pin and wait states are inserted to insure that the HOLD pin stays low for 150 µSecs.

6-2-2. **WRITE CYCLE:**

The HOLD pin must be high for a write to the clock to take place. When data is sent to the data port in preparation for a write, a flip-flop is set to enable writing. Then when the Control Register Code (with write bit set) is sent to the Control port, the write pin and the HOLD pin are raised. Wait states are generated for 150 µSecs to insure that the write has time to take place. The 1/2 second monostable is also set. The CLEAR Control Code will clear the 1/2 second monostable and set the CLK-24 up for another write. The HOLD pin will go low when the CLEAR Control Code is sent or when the 1/2 second monostable expires.

6-3. EXPLICIT DESCRIPTION OF CONTROL CODES AND DATA BITS:

F #	Control Code Bits C3 C2 C1 C0	Data Type	Data Bits D3 D2 D1 D0	Data Limits (Digit)	Notes
0	0 0 0 0	1 Seconds	B3 B2 B1 B0	0-9	Seconds are reset to zero whenever a write is executed to these registers.
1	0 0 0 1	10 Seconds	- B2 B1 B0	0-5	
2	0 0 1 0	1 Minutes	B3 B2 B1 B0	0-9	
3	0 0 1 1	10 Minutes	- B2 B1 B0	0-5	
4	0 1 0 0	1 Hours	B3 B2 B1 B0	0-9	
5	0 1 0 1	10 Hours	**F A/P** B1 B0	0-2	F=0 for 12 hour format, F=1 for 24 hour format. A/P=0 for AM, A/P=1 for PM
6	0 1 1 0	Day of Week	- B2 B1 B0	0-6	0=Sunday .. 6=Saturday
7	0 1 1 1	1 Day	B3 B2 B1 B0	0-9	
8	1 0 0 0	10 Day	- **L** B1 B0	0-3	L=1 for Leap year, Otherwise L=0.
9	1 0 0 1	1 Month	B3 B2 B1 B0	0-9	
10	1 0 1 0	10 Month	- - - B0	0-1	
11	1 0 1 1	1 Year	B3 B2 B1 B0	0-9	
12	1 1 0 0	10 Years	B3 B2 B1 B0	0-9	

B0,B1,B2,B3 are bits of binary representing the digit being read or written to.

F, A/P, and **L** are status bits which set the status of the clock when written, and allow the clock status to be determined during a read cycle. The Data Limits do NOT include these status bits.

Table 2
Description of Control Codes, & Data Bits

APPENDIX A. USING THE HOLD DISABLE PIN

This is another version of the complete date and time routine. This one uses the HOLD DISABLE bit so that the hold pin on the clock is never set and wait states are not generated. This is useful for certain systems that have dynamic memories or disk controllers which cannot tolerate 150 µSec wait states. This version reads the data twice and compares the first and second readings. If the time has changed, then the procedure starts over.

```
1000 DIM D(12)                      :REM  AN ARRAY FOR TIME AND DATE
1010 CONTROL=241:DTA=240            :REM  SET NAMES OF CONTROL AND DATA PORT
1020 FOR I=0 TO 12                  :REM  LOOP FOR ALL 12 DIGITS OF INFO
1030 OUT CONTROL,128+I              :REM  READ WITH HOLD DISABLE
1040 D(I)=INP(DTA)                  :REM  READ DIGIT AND SAVE IN ARRAY
1050 NEXT I
1060 FOR I= 0 TO 12                 :REM  LOOP FOR CHECKING
1070 OUT CONTROL,128+I              :REM  READ AGAIN FOR CHECKING
1080 IF D(I)  INP(DTA) THEN GOTO 1020 :REM IF DATA CHANGED THEN REPEAT
1090 NEXT I
1100 D(5)=D(5) AND 7              :REM  STRIP 12/24 HR BIT (BIT 3)
1110 IF D(5)  3 THEN M$="PM" ELSE M$="AM" :REM CHECK BIT 2 FOR AM/PM
1120 D(5)=D(5) AND 3              :REM  STRIP AM/PM BIT (BIT 2)
1130 D(8)=D(8) AND 3              :REM  STRIP LEAP YEAR BIT
1140 P$=""                        :REM  SET STRING TO NULL
1150 FOR I=0 TO 12                :REM  LOOP THROUGH ALL DIGITS
1160 P$=CHR$((D(I)AND15)+48)+P$:REM STRIP BITS 4-7 AND CONVERT TO ASCII
1170 REM  NOTE THAT EACH NEW CHAR IS ADDED AT LEFT END OF STRING
1180 NEXT I
1190 FOR I= 0 TO VAL(MID$(P$,7,1)) :REM  COUNT UP TO DAY OF WEEK
1200 READ DAY$                    :REM  READ NAME OF EACH DAY
1210 NEXT I                       :REM  EXIT WHEN WE REACH THE RIGHT DAY
1220 REM  NOW PRINT DAY    MONTH/DAY/YEAR
1230 PRINT DAY$,MID$(P$,3,2)+"/"+MID$(P$,5,2)+"/"+LEFT$(P$,2),
1240 REM  NOW PRINT     HOUR:MIN:SEC   AM/PM
1250 PRINT MID$(P$,8,2)+":"+MID$(P$,10,2)+":"+RIGHT$(P$,2),M$
1260 RESTORE                      :REM  RESET READ DATA
1270 GOTO 1020                      :REM  READ DATA AND TIME AGAIN
1280 DATA "SUN","MON","TUES","WED","THURS","FRI","SAT"
```

This routine can be executed by either a call from another program, or by executing the command:

```
GOSUB     1000
```

APPENDIX B. BATTERY REPLACEMENT

The long-life lithium battery in your CLK-24 should last between 3 and 10 years. Hence, you may never need to replace it.
If, however, the battery does require replacement (usually due to accidently placing the CLK-24 on a metal surface), it may be removed using a de-soldering tool, or de-soldering wick.

There are two alternate procedures for having the battery replaced:

1.) A new battery may be installed by soldering it in place of the old. The battery is of the following type

Electrochem Industries Part #BCX50

A replacement cell may be purchased from Dual Systems Control Corporation, for $15.00.

2.) The board may be returned to Dual, with a check for $15.00. It will then be promptly returned to you with a new battery.

APPENDIX C. NOTE TO USERS OF OLD CLK-24

If you have purchased a CLK-24 before May 12,1981, there are some differences between your old CLK-24 and the present version.

First, the operation of the HOLD pin has been made completely automatic. In keeping with this revision, what was previously the HOLD bit on the Control Register is now the HOLD DISABLE bit. That is, asserting this bit disables the automatic hold sequence.

As a result of this particular revision programs which could read the old CLK-24 can still read the the new CLK-24, However programs which set the old CLK-24 will NOT be able to set the new CLK-24.

APPENDIX D. IN CASE OF TROUBLE

If your CLK-24 does not work immediately check the following:

1.) Your program prints the time as FF/FF/FF or ??/??/??

The problem is probably that either the port address switch on the CLK-24 is set incorrectly or that your program is accessing the wrong port address.

2.) You cannot set the time at all.

Check switch #8 to make sure that it is in the 'WR' position. Check the port address. If you still cannot set the time install jumper J-2.

3.) If all else fails, gives us a call. Our Engineering Hotline is 415-549-3854.

OKI semiconductor

FEBRUARY 1981

MSM5832 MICROPROCESSOR REAL-TIME CLOCK/CALENDAR

GENERAL DESCRIPTION

The MSM5832 is a monolithic, metal-gate CMOS integrated circuit that functions as a real time clock/calendar for use in bus-oriented microprocessor applications. The on-chip 32,768 Hz crystal controlled oscillator time base is counted down to provide addressable 4-bit I/O data of SECONDS, MINUTES, HOURS, DAY-OF-WEEK, DATE, MONTH, and YEAR. Data access is controlled by 4-bit address, chip select, read, write and hold inputs. Other functions include 12H/24H format selection, leap year identification and manual ±30 second correction.

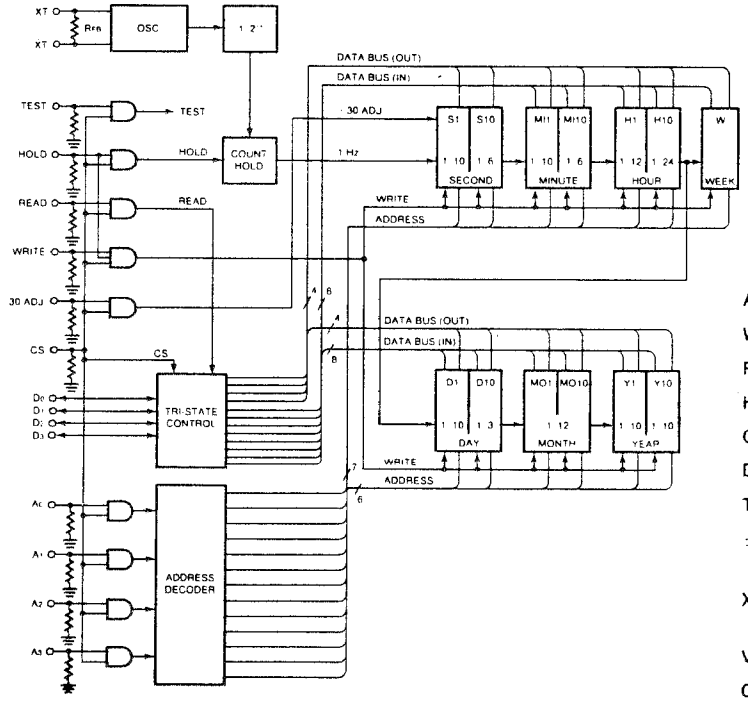
The MSM5832 normally operates from a 5 volt ±5% supply. Battery back-up operation down to 2.2 volts allows continuation of time keeping when main power is off. One test input facilitates rapid testing of the time keeping operations. The MSM5832 is offered in an 18-lead dual-in-line plastic (RS suffix) package.

FEATURES

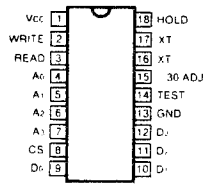
- Microprocessor bus-oriented
- TIME MONTH DATE YEAR DAY OF WEEK

  23:59:59 12 - 31 - 99 - 7
- 4-BIT DATA BUS
- 4-BIT ADDRESS
- Read, Write, Hold, Chip select inputs
- Interrupt signal outputs—1024, 1, 1/60, 1/3600 Hz
- 32.768 KHz crystal controlled operation
- Leap year register bit
- 12 or 24 hour format
- ±30 second error correction
- Single 5 volt power supply
- Back-up battery operation to V_{CC} = 2.2 V
- Low Power Dissipation
  90 μw Max. at V_{CC} = 3 V
  2.5 mw Max. at V_{CC} = 5 V
- High Density 300 mil 18-Pin Package

FUNCTIONAL BLOCK DIAGRAM



PIN CONFIGURATION



A_0 to A_3: Address Inputs

WRITE: Write Enable

READ: Read Enable

HOLD: Count Hold Enable

CS: Chip Select

D_0 to D_3: Data Input/Output

TEST: Test Input

±30 ADJ: ±30 Second Correction Input

XT & $\overline{XT}$: xtal oscillator connections

V_{CC}: +5 V Supply

GND: Ground

FUNCTION TABLE

FIGURE 1

ADDRESS INPUTS				INTERNAL COUNTER	DATA I/O				DATA LIMITS	NOTES
A_0	A_1	A_2	A_3		D_0	D_1	D_2	D_3		
0	0	0	0	S 1	*	*	*	*	0 ~ 9	S_1 or S_{10} are reset to zero irrespective of input data $D_0 \sim D_3$ when write instruction is executed with address selection
1	0	0	0	S 10	*	*	*		0 ~ 5	
0	1	0	0	MI 1	*	*	*	*	0 ~ 9	
1	1	0	0	MI 10	*	*	*		0 ~ 5	
0	0	1	0	H 1	*	*	*	*	0 ~ 9	
1	0	1	0	H 10	*	*	†	†	0~1 / 0~2	D_2 = "1" for PM D_3 = "1" for 24 hour format D_2 = "0" for AM D_3 = "0" for 12 hour format
0	1	1	0	W	*	*	*		0 ~ 6	
1	1	1	0	D 1	*	*	*	*	0 ~ 9	
0	0	0	1	D 10	*	*	†		0 ~ 3	D_2 = "1" for 29 days in month 2 D_2 = "0" for 28 days in month 2 (2)
1	0	0	1	MO 1	*	*	*	*	0 ~ 9	
0	1	0	1	MO 10	*				0 ~ 1	
1	1	0	1	Y 1	*	*	*	*	0 ~ 9	
0	0	1	1	Y 10	*	*	*	*	0 ~ 9	

(1) * data valid as "0" or "1"
blank does not exist (unrecognized during a write and held at "0" during a read)
† data bits used for AM/PM, 12/24 HOUR and leap year
(2) If D_2 previously set to "1", upon completion of month 2 day 29, D_2 will be internally reset to "0"

TYPICAL CHARACTERISTICS—Oscillator Frequency Deviations

Frequency Deviation *vs* Temperature

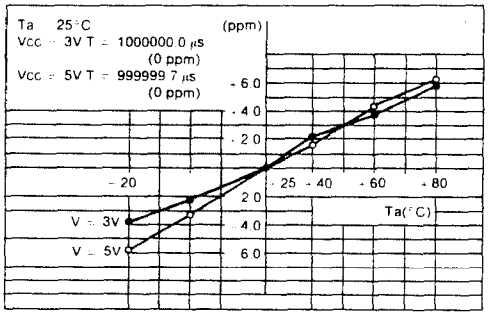


FIGURE 2

Frequency Deviation *vs* Supply Voltage

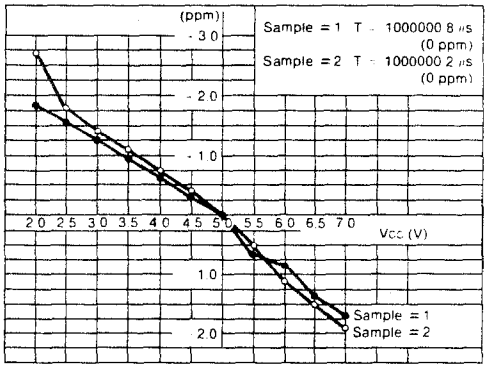
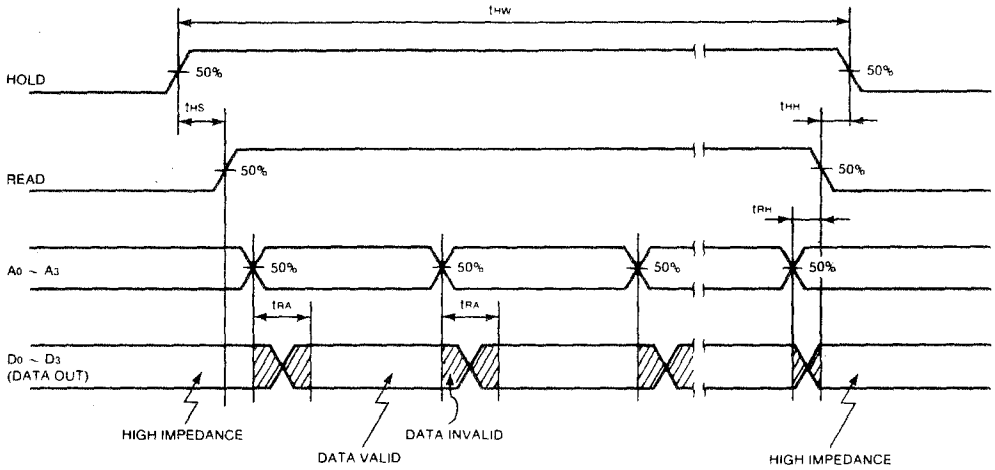


FIGURE 3

READ CYCLE **FIGURE 4**



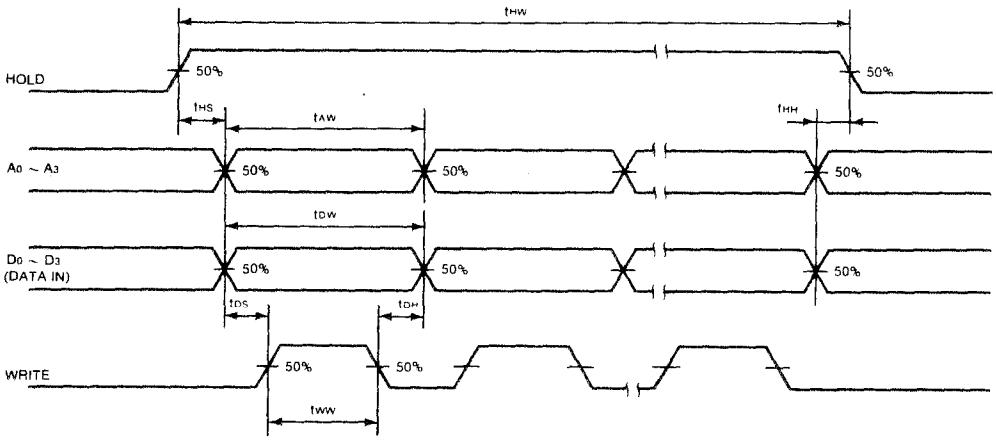
Notes: 1. A Read occurs during the overlap of a high CS and a high READ
2. Output Load: 1 TTL Gate, C_L = 50 pf and R_L = 4.7 KΩ
3. CS may be a permanent "1", or may be coincident with HOLD pulse

WRITE CYCLE

(V_{CC} = 5V ± 5%; Ta = 25°C)

Parameter	Symbol	Min.	Typ.	Max.	Unit
HOLD Set-up Time	tHS	150			μS
HOLD Hold Time	tHH	0			μS
HOLD Pulse Width	tHW			1	SEC
ADDRESS Pulse Width	tAW	1.7			μS
DATA Pulse Width	tDW	1.7			μS
DATA Set-up Time	tDS	0.5			μS
DATA Hold Time	tDH	0.2			μS
WRITE Pulse Width	tWW	1.0			μS

WRITE CYCLE **FIGURE 5**



Notes: 1. A WRITE occurs during the overlap of a high CS, a high HOLD and a high WRITE
2. CS may be a permanent "1", or may be coincident with HOLD pulse

FUNCTIONAL DESCRIPTION

A block diagram of the MSM5832 microprocessor real-time clock/calendar and a package connection diagram are shown on the first page. Figure 9 illustrates a method of interfacing between the clock/calendar circuit and a micro processor. Figures 9, 10 and 11 illustrate alternative standby power supply circuits. A function table listing relationships between address inputs, data input/output and internal counter selection is shown in Figure 1. Unless otherwise indicated, the following descriptions are based on the block diagram.

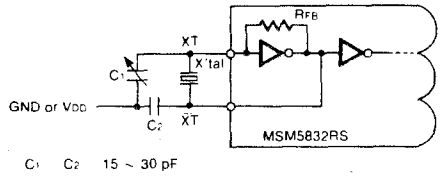
32.768 K Hz OSCILLATOR (pins 16 and 17): An internal inverting amplifier with feedback resistor, R_{FB}, is connected with a crystal and two capacitors *as shown in Figure 6* to form a stable, accurate oscillator—which serves as the precision time base of the circuit. Capacitors C1 and C2 in series provide the parallel load capacitance required for precise tuning of the quartz crystal. Typical oscillator performance as a function of ambient temperature and supply voltage is shown in Figures 2 and 3 respectively.

$A_0 \sim A_3$ (pins 4 ~ 7): Address inputs, used to select internal counters for read/write operations (see function table—*Figure 1*). A "1" is defined as V_{CC}; a "0" is GND. Pull-down to GND is provided by internal resistors.

$D_0 \sim D_3$ (pins 9 ~ 12): Data Inputs/Outputs, two-way bus lines controlled by READ and WRITE inputs. *As shown in Figure 7* external pull-up resistors of 4.7K or higher are required by the open-drain N-channel MOS outputs. D_3 is the MSB; D_0 is the LSB.

TEST (pin 14): Normally this input is unconnected—pull-down to GND is provided by an internal resistor—or connected to GND. With CS at V_{CC}, pulses to V_{CC} on the TEST input will directly clock the S1, MI10, W, D1 and Y1 counters, depending on which counter is addressed (W and D1 are selected by D1 address in this mode only). Roll-over to next counter is enabled in this mode.

OSCILLATOR CIRCUIT — FIGURE 6



C_1 C_2 15 ~ 30 pF

CHIP SELECT (pin 8): Connecting CS input to V_{CC} enables all inputs and outputs. Unconnected—pull-down to GND is provided by an internal resistor—or connecting CS to GND will disable HOLD, WRITE, READ, ±30 ADJ, $D_0 \sim D_3$, $A_0 \sim D_3$ and TEST.
As shown in Figure 9 CS can be used to detect system power failure by connecting system power (+ 5V) to CS, so that when system power is *on*, all inputs and outputs will be enabled, and when system power is *off*, all inputs and outputs will be disabled. The threshold voltage of CS is higher than all other inputs to insure correct operation of this function.

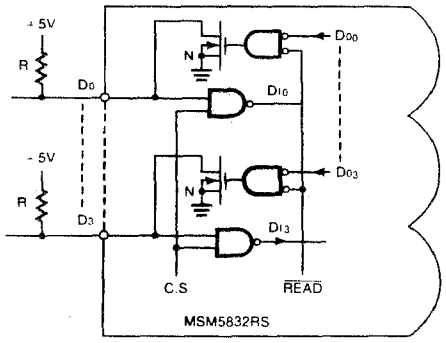
HOLD (pin 18): Switching this input to V_{CC} inhibits the internal 1Hz clock to the S1 counter. After the specified HOLD set-up time (150 μs), all counters will be in a static state, thus allowing error-free read or write operations. So long as the HOLD pulse width is less than 1 second, accuracy of the real time will be undisturbed. Pull-down to GND is provided by an internal resistor.

READ (pin 3): Read function *as shown in Figure 4* is enabled when READ is switched to V_{CC}. Pull-down to GND is provided by an internal resistor.

WRITE (pin 2): Write function *as shown in Figure 5* is enabled when WRITE is switched to V_{CC}. Pull-down to GND is provided by an internal resistor.

±30 ADJ (Pin 15): Momentarily connecting this input to V_{CC} (>31.25 ms) will reset seconds (S1, S10 counters and $2^{11} \sim 2^{15}$ frequency dividers) to 00; if seconds were 30 or more, one minute is added to the minutes (MI 1 counter) and if seconds were less than 30, the minutes are unchanged. Pull-down to GND is provided by an internal resistor.

DATA I/O CIRCUIT — FIGURE 7

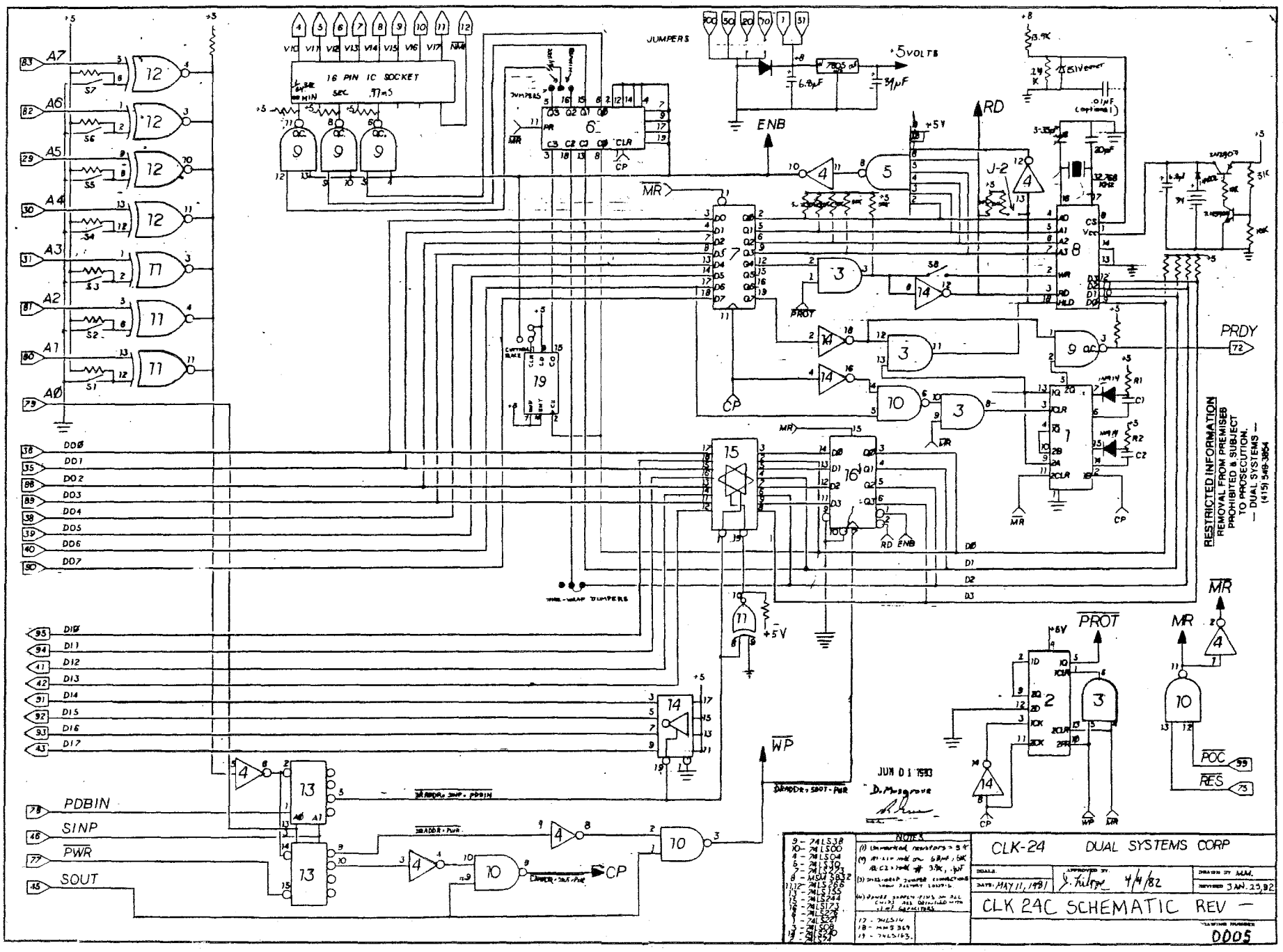


REFERENCE SIGNAL OUTPUT
Reference signals are available as outputs on $D_0 \sim D_3$ if CS, READ and $A_0 \sim A_3$ are at V_{CC}. Refer to *Figure 8* for specifics. *As shown in Figure 9* these signals may be used to generate interrupts for the microprocessor.

REFERENCE SIGNAL OUTPUTS — FIGURE 8

CONDITIONS	OUTPUT	REFERENCE FREQUENCY	PULSE WIDTH
HOLD = L	D_0 (1)	1024 Hz	duty 50%
READ = H	D_1	1 Hz	122.1 μs
C.S. = H	D_2	1/60 Hz	122.1 μs
$A_0 \sim A_3$ = H	D_3	1/3600 Hz	122.1 μs

(1) 1024 Hz signal at D_0 not dependent on HOLD input level.


A7
A6
A5
A4
A3
A2
A1
A0
S7
S6
S5
S4
S3
S2
S1
16 PIN IC SOCKET
JUMPERS
+5VOLTS
ENB
RD
MR
PROT
PRDY
CP
DD0
DD1
DD2
DD3
DD4
DD5
DD6
DD7
DI0
DI1
DI2
DI3
DI4
DI5
DI6
DI7
PDBIN
SINP
PWR
SOUT
WP
POC
RES
RESTRICTED INFORMATION
REMOVAL FROM PREMISES PROHIBITED & SUBJECT TO PROSECUTION.
— DUAL SYSTEMS —
(415) 549-3864
CLK-24 DUAL SYSTEMS CORP
CLK 24C SCHEMATIC REV —
DD05
DRAWN BY MM
DATE: MAY 11, 1981
REVISED JAN. 23, 82

APPENDIX G. PROGRAMMING SUMMARY

DATA PORT= ____ **(240 STANDARD)** "TIME"= ONE DIGIT OF DATA
CONTROL PORT= ____ **(241 STANDARD)**

CONTROL PORT CODES:

0	1 SECOND	
1	10 SECONDS	
2	1 MINUTE	
3	10 MINUTES	
4	1 HOUR	
5	10 HOURS	ADD 4 FOR PM, 8 FOR 24 HOUR FORMAT (AND WITH 3 WHEN READING)
6	DAY OF WEEK	0 IS SUNDAY .. 6 IS SATURDAY
7	1 DAY	
8	10 DAYS	ADD 4 FOR LEAP YEAR (AND WITH 3 WHEN READING)
9	1 MONTH	
10	10 MONTHS	
11	1 YEAR	
12	10 YEARS	
16	WRITE	
64	CLEAR CONTROL CODE	
128	HOLD DISABLE	

STANDARD READ PROCEDURE:

```
OUT CONTROL PORT, CONTROL CODE
TIME = INP (DATA PORT)
OUT CONTROL PORT, 64.............BETWEEN COMPLETE READ CYCLES IF
                                 MORE THAN ONE PER SECOND.
```

STANDARD WRITE PROCEDURE:

```
OUT DATA PORT, TIME
OUT CONTROL PORT, CONTROL CODE + 16
OUT CONTROL PORT,64.................AFTER EACH DIGIT IS WRITTEN
```

ENABLE INTERRUPTS:

```
OUT CONTROL PORT, 143
```

AWKNOWLEDGE INTERRUPTS:

```
OUT CONTROL,142 OR READING THE CLOCK.
```

WARRANTY

Dual Systems Corporation warrants the equipment covered hereby to be free from defects in material and workmanship for twelve (12) months from date of original shipment to purchaser. During this warranty period Dual Systems will repair or replace defective equipment FOB its place of business without charge to purchaser.

This warranty applies to defects arising out of normal use and service of the equipment as specified by Dual Systems. This warranty does not cover abnormal operation of the equipment, accident, alteration, negligence, misuse and repairs or service performed by other than Dual Systems' authorized representatives. Purchaser shall upon request by Dual Systems furnish reasonable evidence that the defect arose from causes placing a liability on Dual Systems.

The obligation of Dual Systems under this warranty is limited to repair or replacement of the defective equipment and is the only warranty applicable to the equipment. Dual Systems shall not be liable for any injury, loss or damage, direct or consequential, arising out of the use or inability to use the product. No changes in the warranty shall be effective without the prior approval in writing of both parties. This warranty and obligations and liabilities thereunder shall replace all warranties or guarantees express or implied including the implied warranty of merchantability.

Dual Systems Corporation
2530 San Pablo Avenue
Berkeley, California 94702

(415) 549-3854