Mountain Hardware, Inc.

LEADERSHIP IN COMPUTER PERIPHERALS

100,000 DAY CLOCK

OPERATING MANUAL

983-IBM-324

100,000 DAY CLOCK

OPERATING MANUAL

MOUNTAIN HARDWARE'S

100,000 DAY CLOCK

FOR S-100 BUS COMPUTERS

INTRODUCTION....................................1

USING THE CLOCK.................................2

- BATTERY INFORMATION.........................2
- SETTING THE FREQUENCY.......................3
- SETTING THE PORT ADDRESS....................3
- SETTING THE CLOCK...........................4
- READING THE CLOCK...........................5
- USING INTERRUPTS............................5

SOFTWARE ROUTINES...............................8

- SETTING THE CLOCK...........................8
- READING THE CLOCK...........................9
- CALENDAR ROUTINES...........................9
- INTERRUPT CONTROL SOFTWARE.................14

THEORY OF OPERATION............................20

- IMPLEMENTATION.............................21
- TROUBLE-SHOOTING...........................22

SCHEMATIC......................................24

PARTS LIST.....................................25

WARRANTY.......................................26

INTRODUCTION

Mountain Hardware's 100,000 Day Clock is an accurate time piece for your computer. It will keep track of time in 100 microsecond intervals, up to 100,000 days. Advanced Complimentary Metal Oxide Semiconductor (CMOS) circuit draws less than 2 mA, which allows the clock to be run off a 9-volt battery for up to four days while the computer is shut down or if AC power fails.

The Clock uses 15 I/O ports for the time plus one I/O port to set the interrupt function. Using DIP switches, the user can assign these ports to any 16 consecutive 8080/Z-80 ports. The Clock is easily set by entering BCD digits one at a time at each time port. The moment you enter the first digit, the Clock stops. Then you enter the remaining digits. The Clock starts again on the first "read" command. A "write protect" switch prevents the Clock from being accidentally stopped or changed.

By using the interrupt feature of the Clock, activities relating to time of day may be preformed at preprogrammed intervals without interfering with the normal operations of the computer. You may program interruptions on any change in a Clock digit; that is, at intervals of 100 microseconds, 1ms, 10ms, and so on to 1 hour, 10 hours, etc. The board can be easily used with most BASICs. However, with our Introl BASIC, time is especially simple to set, compare, check, display and print.

Two software packages are included that expand the capabilities of the Clock board. One package gives calendar information such as month, day, year and day of week. The other package allows multiple interrupts, at any time interval, or absolute time.

The 100,000 Day Clock board, because of all the features included on one board, will enhance the power of your computer and add to it the dimension of time.

USING THE CLOCK

The 100,000 Day Clock has been designed to work in virtually all S-100 computers. It will work with machines running at speeds up to 4 MHz, which means it will operate with the newer and faster microprocessors.

When handling the clock board, care must be taken to avoid static discharges on the board, as this can cause damage to the CMOS (Complimentary Metal Oxide Semiconductor) circuitry. Hold the board on the sides when handling, and store the unit on static-proof foam when out of the computer. CMOS has very high input impedances and properly placed fingers on the back side of the board can stop the clock or accidentaly change the time. General rule is to handle as little as possible, and then only on the sides.

BATTERY INFORMATION

The battery supplied with the clock board is a rechargeable NiCad battery that powers the clock when the computer is turned off, or when the power fails. The battery will continue to power the CMOS circuitry on the board and the correct time will not be lost. The battery has the capacity to run the clock for 4-5 days if it is fully charged.

To fully charge the battery, power must be applied to the clock for at least 4 days; as it is a slow charge. This is designed to maximize the life of the battery. As a general guideline with plenty of margin. let the battery charge two hours for every hour of use.

When the computer power is turned off, the board may be removed from the computer as long as the battery is in place and charged. The clock will continue keeping time. Hence, the board may be "time-shared" between two computers or set aside for awhile if its space is needed for another peripheral in machines with few expansion slots.

The battery's life should be several years and should be replaced if its performance drops significantly.

A 9 - 12 volt adapter may be plugged into J1. This will keep the battery charged and the clock running even when power is turned off from the computer. This will allow very long down times, and also keep the battery charged in the event power should go off in the building.

The adapters are available from Mountain Hardware.

SETTING THE FREQUENCY

Your 100,000 Day Clock has been factory assembled, burned in and tested. The 1.0000 MHz time base has been accurately set to within .001%. Vibrations or extreme temperatures can cause slight changes to the time base and may produce noticeable errors. If these errors are noticed, or if you desire to set this frequency more precisely for your environment, an accurate frequency counter and a small non-metallic screwdriver are required.

Connect the frequency counter with the ground lead to the screw on the regulator and the positive lead to Pin 10 of U6. Adjust C12 for a frequency as close to 1.000000 MHz as possible. Be sure the clock is at the same operating temperature as its normal environment.

SETTING THE PORT ADDRESS

The clock board occupies 16 port addresses on the S-100 system bus. Changing the switches labelled A4, A5, A6 and A7 can change the clock to respond to different port addresses. Table 1 shows the relationships between switch positions and addresses.

PORT ADDRESSES		SWITCH POSITION			
DECIMAL	HEX	A7	A6	A5	A4
0 - 15	0 - F	0	0	0	0
16 - 31	10 - 1F	0	0	0	1
32 - 47	20 - 2F	0	0	1	0
48 - 63	30 - 3F	0	0	1	1
64 - 79	40 - 4F	0	1	0	0
80 - 95	50 - 5F	0	1	0	1
96 - 111	60 - 6F	0	1	1	0
112 - 127	70 - 7F	0	1	1	1
128 - 143	80 - 8F	1	0	0	0
144 - 159	90 - 9F	1	0	0	1
160 - 175	A0 - AF	1	0	1	0
176 - 191	B0 - BF	1	0	1	1
192 - 207	C0 - CF	1	1	0	0
208 - 223	D0 - DF	1	1	0	1
224 - 239	E0 - EF	1	1	1	0
240 - 255	F0 - FF	1	1	1	1

TABLE 1

For port selection purposes, a 1 (one) means the switch is closed (on) a 0 (zero) means the switch is open (off).

Address your clock to a set of ports that are not presently used by your other peripherals. If possible, we recommend that you use ports 32-47 (20-2F HEX) to standardize with our software.

SETTING THE CLOCK

The following table shows the significance of each of the 16 ports assigned. Obviously, if you assigned your board with the switches set for 32-47 decimal, your lowest port address for the clock would be 32 and the highest would be 47.

PORT ADDRESS	PORT SIGNIFICANCE	TIME RANGE
Lowest	100 µs (microseconds)	0-9
Lowest +1	1 ms (milliseconds)	0-9
Lowest +2	10 ms	0-9
Lowest +3	100 ms	0-9
Lowest +4	1 second	0-9
Lowest +5	10 seconds	0-5
Lowest +6	1 minute	0-9
Lowest +7	10 minutes	0-5
Lowest +8	1 hour	0-9
Lowest +9	10 hours	0-2
Lowest +10	1 day	0-9
Lowest +11	10 days	0-9
Lowest +12	100 days	0-9
Lowest +13	1,000 days	0-9
Lowest +14	10,000 days	0-9
Highest	Interrupt port	0-14 (0-E Hex)

TABLE 2

Each digit of time is available by reading the proper port number. To set the clock, the write protect switch must be set in the WRITE position (the switch in the on position). When in the WRITE PROTECT position, it is impossible to write to the clock, except to set interrupts.

To set the clock OUT instructions are used to set the digits to the proper time. When an OUT instruction is executed to the clock, the clock stops. This allows you to set all the digits correctly without fear of the clock changing. Therefore, the recommended procedure is to set all the digits correctly for a minute or two ahead of time and wait for the real time to catch up with the clock. At the precise time, execute a read (INP) from one of the clock's ports. This starts the clock running.

As can be seen from the time range column in the last table, the clock keeps "normal" time for seconds, minutes and hours. The clock board will keep track of seconds up to 59 and then carry back to 00 instead of 60. The same is true of minutes. Hours count up to 23 and then carry back to 00. Days count from 0 through 99,999. The clock will need to be reset for time after 99,999 days, 23 hours, 59 minutes, 59.9999 seconds. The exception is, if you desire to change the clock for Daylight Saving time, if used in your area.

Once the time has been set, place the WRITE PROTECT switch back in the WP position. This places the clock in a state where the time cannot be accidentally changed.

READING THE CLOCK

The clock is read by using input commands from the ports addressed on the clock. See the chart on the previous page to find the significance of each port. The lower four bits of each word contain the BCD digit of time.

A potential problem can occur when reading the clock. If you are reading all the digits in order to find the time, the clock may change after you have read several but before you have read them all. The clock is changing every 100 microseconds. If the time is 23 hours, 59 minutes, 59.9999 seconds when you start reading and changes to 1 day, 00 hours, 00 minutes, 00.0000 seconds, a significant error will be found. Admittedly, this occurs rarely but can happen in lesser degrees to any port. Two solutions are available:

1. Read the clock twice. If the time has not changed then the time is correct. If it has changed, the later time is correct.

2. The clock board has a provision to tell the user that the time has changed since the clock was last read. Bit 7 of each input time (the bottom 4 bits contain the BCD digit of time) is a 1 (one) if the time has not changed since the last read. If it is a 0 (zero) then the time has changed by at least 100 microseconds since the last read. Consequently, the time can either be corrected with software or by re-reading the clock.

The speed of your computer determines how many instructions can be executed in 100 microseconds (when the clock has changed). For fast timing considerations, all of the digits should be read in order and stored before any manipulation is done. This will improve the chances of reading the clock without it changing. Of course, if you are not interested in 100 microsecond intervals the problem is not as significant and errors are less likely.

USING INTERRUPTS

The 100,000 Day Clock board also has the capability for generating hardware interrupts. The hardware supports one interrupt level. More can be handled in software.

The interrupt feature is enabled by sending a value of 1X in hexadecimal to the highest numbered clock port where X represents a digit from 0 to E. This corresponds to the clock digit whose transition will generate an interrupt.

In other words, if the clock is address for ports 20 through 2F Hex and a value of 14 (Hex) is sent to port 2F, then an interrupt will be generated every second since a 4 represents the "seconds" digit.

Number output to the highest port on the clock board		Interrupt will occur every:
DECIMAL	HEX	
16	10	100 microseconds
17	11	1 millisecond
18	12	10 millisecond
19	13	100 millisecond
20	14	1 second
21	15	10 seconds
22	16	1 minute
23	17	10 minutes
24	18	1 hour
25	19	10 hours
26	1A	1 day
27	1B	10 days
28	1C	100 days
29	1D	1,000 days
30	1E	10,000 days
0 - 15	00 - OE	CLEARS INTERRUPTS

TABLE 3

To clear the interrupt a 00 through OE (Hex) may be output to the interrupt port. Also hitting the reset button on your computer or powering up the computer will clear the interrupts.

The trace coming off of Pin 75 on the S-100 bus may be cut if it is desired to not have RESET clear the interrupt feature.

Several types of interrupts are available on the clock board. The unit is shipped set up for a re-start type interrupt. This means that whenever an interrupt occurs from the clock and the computer acknowledges it, an 8080-type re-start instruction is forced onto the S-100 bus. The switches labelled R1, R2 and R3 determine which re-start address is presented to the computer. When a re-start instruction is performed the computer calls an address as shown in Table 4.

RESTART ADDRESS		RESTART SWITCHES		
DECIMAL	HEX	R3	R2	R1
0	0000	0	0	0
8	0008	0	0	1
16	0010	0	1	0
24	0018	0	1	1
32	0020	1	0	0
40	0028	1	0	1
48	0030	1	1	0
56	0038	1	1	1

TABLE 4

Software can then be placed at the re-start location to service the interrupt.

The other type of interrupt on the 100,000 Day Clock board is a Vectored Interrupt. This is available for users of a vectored interrupt controller. To use this feature, the trace between I and PINT in the lower-left corner of the clock should be cut and a jumper placed between I and one of the VI pins labelled VIO-V17 and also in the lower-left corner of the clock board. For more information on the vectored interrupt see the details with your Vectored Interrupt board.

With all interrupts care must be taken to avoid conflicts between peripherals requesting an interrupt.

SOFTWARE ROUTINES

SETTING THE CLOCK FROM BASIC

Setting the clock is as easy as outputting the desired time digits to the corresponding digits of the clock.

Here is a program to do just that, setting the hours, minutes and seconds.

```
10  C = 32: REM CLOCK'S LOWEST PORT NUMBER
20  PRINT "ENTER TIME AS H,H,M,M,S,S"
30  INPUT H1,H0,M1,M0,S1,S0
40  OUT C+9, H1: OUT C+8, H0: OUT C+7, M1: OUT C+6, M0
50  OUT C+5, S1: OUT C+4, S0
60  X=INP(C): REM START CLOCK AGAIN
70  END
```

An assembly language program is just as simple. Here is one which sets the time to 12:30:15.

```
MVI A, 1       : Tens of hours
OUT CLK+9
MVI A,2        : Hours
OUT CLK+8
MVI A,3        : Tens of minutes
OUT CLK+7
MVI A,0        : Minutes
OUT CLK+6
MVI A, 1       : Tens of seconds
OUT CLK+5
MVI A,5        : Seconds
OUT CLK+4
INP CLK        : Start clock
```

READING THE CLOCK - PRINTING THE TIME

Since the clock stores the time on-board in the form of BCD digits, displaying the time is very easy. The lower 4 bits of each clock digit hold the actual information. Here is a basic program which prints the time.

```
10   C=32: REM CLOCK'S LOWEST PORT ADDRESS
20   FOR I = 9 TO 4 STEP - 1
30   D=INP(C+I): REM GET A DIGIT
35   D=D - INT (D/16)* 16 : REM REMOVE TOP 4 BITS
40   PRINT D;: REM PRINT DIGIT
50   IF I=8 OR I=6 THEN PRINT ":";
60   NEXT I
70   PRINT
80   END
```

CALENDAR ROUTINES

This software package was developed to enable you to translate the day information on the clock board (0 - 99,999 days) to date information in the form of month, day, year, day of week.

Using this software is simple:

1. Set location "CLKPRT" (4400) to the lowest port address of your clock board.

2. Call "READ" as a machine language subroutine. (Location 4200).

3. Read the returned information from RAM storage area.

MONTH	is the month (1 = JAN, 2 = FEB...) (Location 4402).
DATE	is the day (1 - 31) (Location 4403
YEAR	is the year (1978...) (locations 4404, 4405-Low, High).
DAY	is the day of week (0 = SAT, 1 = SUN...) (Location 4401).

The above addresses are given in hex and refer to the source listing of the calendar routines.

The calendar routine assumes that the DAYS digits of your clock board have been set to the number of days since December 31, 1977. That is, January 1, 1978 is DAY 00001.

That is all there is to it.

Here is a BASIC program to perform this task:

```
10   L =17408: REM START OF RAM STORAGE FOR CAL ROUTINES
20   POKE L, 32: REM CLOCK AT 32 (LOWEST PORT # ON CLOCK)
30   Y = USR (16896): REM CALL ROUTINE
40   PRINT PEEK (L+2); "/", PEEK (L+1); "/";
50   PRINT PEEK (L+4) + 256*PEEK (L+5)
60   END
```

This prints the date as MM/DD/YYYY.

```
                        ;; DATE ROUTINES FOR MOUNTAIN HARDWARE
                        ;; 100,000 DAY CLOCK/CALENDAR BOARD (S-100).
                        ;; JOHN C. SHEPARD \ 3-16-78

                        TITLE   'Calendar Software'
 4200 =         ROMBEG  EQU     04200H  ;BEGINNING OF ROM STORAGE
 4300 =         RAMBEG  EQU     04400H  ;BEGINNING OF RAM STORAGE
 4200                   ORG     ROMBEG

                ;; MAIN ROUTINE 'READ' HERE
                READ:
 4200 E5        READ1:  PUSH    H       ;SAVE REG'S
 4201 D5                PUSH    D
 4202 C5                PUSH    B
 4203 21BA07            LXI     H,1978  ;SET INITIAL YEAR
 4206 220444            SHLD    YEAR    ;STORE IN 'YEAR'
                ;;
                ;;      READ DIGITS FROM CLOCK BOARD INTO <HL>
                ;;      AS A BINARY NUMBER FROM 0 TO 32767.
                ;;
 4209 210000            LXI     H,0     ;CLEAR RESULT NUMBER
 420C 3A0044            LDA     CLKPRT  ;FIND OUT WHERE CLOCK LIVES
 420F C60E              ADI     0EH     ;ADD OFFSET TO LOOK AT DAYS
 4211 4F                MOV     C,A     ;KEEP IN C FOR INP ROUTINE
 4212 0605              MVI     B,5     ;FIVE DIGITS HATH CLOCK

 4214 CDB142    READ2:  CALL    INDO    ;READ CLOCK PORT
 4217 E60F              ANI     0FH     ;MASK ALL EVIL
 4219 54                MOV     D,H     ;SET DE=HL
 421A 5D                MOV     E,L
 421B 29                DAD     H       ;SET HL=HL*10
 421C 29                DAD     H
 421D 19                DAD     D
 421E 29                DAD     H
 421F 85                ADD     L       ;PLUS DIGIT FROM CLOCK
 4220 6F                MOV     L,A
 4221 D22542            JNC     READ3   ;TAKE CARE OF CARRY
 4224 24                INR     H
 4225 0D        READ3:  DCR     C       ;PREPARE TO GRAB NEXT DIGIT
 4226 05                DCR     B       ;MORE NUMBERS TO MUNCH?
 4227 C21442            JNZ     READ2   ;YEP.

 422A 220644            SHLD    DAYS    ;DONE READING CLOCK. SAVE.
                ;;
                ;;      COMPUTE NUMBER OF YEARS, ADD ON TO YEAR
                ;;      SEE HOW MANY DAYS INTO THIS YEAR
                ;;
 422D 010200            LXI     B,0002H ;LPFLG:=0, LPCNT:=2
 4230 116D01    ABBA1:  LXI     D,365   ;<DE>:=365
 4233 78                MOV     A,B     ;CHECK LPFLG
 4234 B7                ORA     A
 4235 CA3942            JZ      ABBA2   ;<DE> OK IF NOT LEAP YEAR
 4238 13                INX     D       ;ELSE MAKE <DE> = 366
 4239 E5        ABBA2:  PUSH    H       ;SAVE # DAYS LEFT
 423A 7C                MOV     A,H     ;COMPARE HL-DE
 423B 92                SUB     D
```

```
423C C24142              JNZ     ZIT1
423F 7D                  MOV     A,L
4240 93                  SUB     E
4241 F5         ZIT1:    PUSH    PSW        ;SAVE FLAGS
4242 7D                  MOV     A,L        ;<HL>:=<HL>-<DE>
4243 93                  SUB     E
4244 6F                  MOV     L,A
4245 7C                  MOV     A,H
4246 9A                  SBB     D
4247 67                  MOV     H,A
4248 F1                  POP     PSW        ;RESTORE FLAGS FROM COMPARE
4249 DA6642              JC      ABBA3      ;CARRY SAYS: DE>HL
424C CA6642              JZ      ABBA3      ;ZERO SAYS:  DE=HL
424F 0600                MVI     B,0        ;CLEAR LPFLG
4251 F1                  POP     PSW        ;CLEAR STACK FROM OLD <HL>
4252 E5                  PUSH    H          ;SAVE THIS <HL>
4253 2A0444              LHLD    YEAR       ;<YEAR>:=<YEAR>+1
4256 23                  INX     H
4257 220444              SHLD    YEAR
425A E1                  POP     H          ;RESTORE THIS <HL>
425B 0C                  INR     C          ;<LPCNT>:=<LPCNT>+1
425C 79                  MOV     A,C        ;IF (<LPCNT> AND 3)=0, THEN
425D E603                ANI     03H        ;       <LPFLG>:=1
425F C23042              JNZ     ABBA1
4262 04                  INR     B
4263 C33042              JMP     ABBA1
4266 D1         ABBA3:   POP     D          ;GET LAST <HL> VALUE INTO <DE>
                ;;
                ;;       <YEAR> NOW HAS CORRECT YEAR.
                ;;       NOW CALCULATE MONTH, DAY-OF-MONTH.
                ;;
4267 0E01                MVI     C,1        ;<MONTH>:=1 INITIALLY
4269 21FC42              LXI     H,MTBL     ;<HL> POINTS TO TABLE OF DAYS
426C D5         ABBA4:   PUSH    D          ;SAVE <DE> (DAYS)
426D 7E                  MOV     A,M        ;<DE>:=<DE>-<M(<MONTH>)>-
426E E5                  PUSH    H          ;       (IF <MONTH>=2, THEN
426F 6F                  MOV     L,A        ;        LPFLG, ELSE 0)
4270 79                  MOV     A,C
4271 FE02                CPI     2
4273 7D                  MOV     A,L
4274 C27942              JNZ     ABBA5
4277 80                  ADD     B
4278 6F                  MOV     L,A
4279 7A         ABBA5:   MOV     A,D        ;DO DE-0L COMPARE
427A B7                  ORA     A
427B C28042              JNZ     ZIT2
427E 7B                  MOV     A,E
427F 95                  SUB     L
4280 F5         ZIT2:    PUSH    PSW        ;SAVE FLAGS!!!
4281 7B                  MOV     A,E
4282 95                  SUB     L
4283 5F                  MOV     E,A
4284 7A                  MOV     A,D
4285 DE00                SBI     0
4287 57                  MOV     D,A
4288 F1                  POP     PSW        ;RESTORE FLAGS
```

```
4289 E1                 POP     H
428A DA9642             JC      ABBA7   ;CARRY SAYS <DE> TOO SMALL
428D CA9642             JZ      ABBA7   ;ZERO SAYS <DE> JUST RIGHT
4290 0C                 INR     C       ;<MONTH>:=<MONTH>+1
4291 23                 INX     H       ;POINT TO NEXT TABLE ENTRY
4292 F1                 POP     PSW     ;CLEAR STACK OF OLD <DAYS>
4293 C36C42             JMP     ABBA4   ;PROCESS MORE
4296 E1         ABBA7:  POP     H       ;GET LAST <DAYS>
4297 7D                 MOV     A,L     ;LOOK ONLY AT LO-ORDER BYTE
4298 320344             STA     DATE    ;WHICH IS DAY-OF-MONTH
429B 79                 MOV     A,C     ;LOOK AT <MONTH>
429C 320244             STA     MONTH   ;STORE IN APPROPRIATE PLACE
429F 2A0644             LHLD    DAYS    ;GET DAYS IN CLOCK
42A2 EB                 XCHG            ;PUT IN DE
42A3 210700             LXI     H,7     ;DIVIDE BY SEVEN
42A6 CDC642             CALL    DIVIDE  ;DO DIVIDE
42A9 7D                 MOV     A,L     ;GET LO REMAINDER
42AA 320144             STA     DAY     ;STORE DAY-OF-WEEK
                ;;
                ;;      PROCESSING FINISHED, RESTORE
                ;;      USER'S REGISTERS AND RETURN
                ;;
42AD C1                 POP     B       ;RESTORE REG'S
42AE D1                 POP     D
42AF E1                 POP     H
42B0 C9                 RET             ;RETURN TO CALLER

                INDO:                   ;GET INPUT FROM PORT C
42B1 E5                 PUSH    H       ; SAVE USER'S <HL>
42B2 CDB742             CALL    INDO2   ; GET BYTE FROM PORT
42B5 E1                 POP     H       ; RESTORE <HL>
42B6 C9                 RET             ; RETURN TO CALLER
42B7 2100C9     INDO2:  LXI     H,RET*256
42BA E5                 PUSH    H       ; PUSH 'NOP, RET' ON STACK
42BB 61                 MOV     H,C
42BC 2EDB               MVI     L,IN
42BE E5                 PUSH    H       ; PUSH 'IN <PORT>' ON STACK
42BF 210000             LXI     H,0
42C2 39                 DAD     SP      ; <HL>:=<SP>
42C3 F1                 POP     PSW     ; <SP>:=<SP>+2
42C4 F1                 POP     PSW     ; <SP>:=<SP>+2
42C5 E9                 PCHL            ; JUMP TO STACK AND RUN IT!
                ;;
                ;;      16-BIT DIVIDE. DIVIDE <DE> BY <HL>.
                ;;      RETURN <DE>=RESULT, <HL>=REMAINDER.
                ;;      CLOBBERS A,D,E,H,L,F/F'S
                ;;
42C6 C5         DIVIDE: PUSH    B
42C7 220844             SHLD    DVTMP1
42CA 210A44             LXI     H,DVTMP2
42CD 3611               MVI     M,11H
42CF 010000             LXI     B,0
42D2 C5                 PUSH    B
42D3 7B         DIVID2: MOV     A,E
42D4 17                 RAL
42D5 5F                 MOV     E,A
```

```
 42D6 7A                MOV     A,D
 42D7 17                RAL
 42D8 57                MOV     D,A
 42D9 35                DCR     M
 42DA E1                POP     H
 42DB C2E042            JNZ     DIVID3
 42DE C1                POP     B
 42DF C9                RET
 42E0 3E00     DIVID3:  MVI     A,0
 42E2 CE00              ACI     0
 42E4 29                DAD     H
 42E5 44                MOV     B,H
 42E6 85                ADD     L
 42E7 2A0844            LHLD    DVTMP1
 42EA 95                SUB     L
 42EB 4F                MOV     C,A
 42EC 78                MOV     A,B
 42ED 9C                SBB     H
 42EE 47                MOV     B,A
 42EF C5                PUSH    B
 42F0 D2F542            JNC     DIVID4
 42F3 09                DAD     B
 42F4 E3                XTHL
 42F5 210A44   DIVID4:  LXI     H,DVTMP2
 42F8 3F                CMC
 42F9 C3D342            JMP     DIVID2
               ;;
               ;;       TABLE OF DAYS IN EACH MONTH
               ;;       STARTING WITH JANUARY.
               ;;
 42FC 1F1C1F   MTBL:    DB      31,28,31
 42FF 1E1F1E            DB      30,31,30
 4302 1F1F1E            DB      31,31,30
 4305 1F1E1F            DB      31,30,31
               ;;
               ;;       ALL OF THE ABOVE MAY RESIDE IN ROM.
               ;;       THE FOLLOWING MUST RESIDE IN RAM.
               ;;
 4308 =        ROMEND   EQU     $       ;LAST ROM ADDRESS
 4400                   ORG     RAMBEG  ;RAM STARTS HERE

 4400 20       CLKPRT:  DB      20H     ;SET TO CLOCK ADDRESS
 4401          DAY:     DS      1       ;DAY-OF-WEEK (0=SAT,1=SUN..)
 4402          MONTH:   DS      1       ;MONTH-OF-YEAR (1=JAN,2=FEB..)
 4403          DATE:    DS      1       ;DAY-OF-MONTH (1 TO 31)
 4404          YEAR:    DS      2       ;YEAR (1978...)
 4406          DAYS:    DS      2       ;TEMP STORAGE FOR DAYS SINCE
                                                          12/31/77
 4408          DVTMP1:  DS      2       ;TEMP FOR 'DIVIDE'
 440A          DVTMP2:  DS      1       ;TEMP FOR 'DIVIDE'

 440B                   END
B>
```

INTERRUPT CONTROL SOFTWARE

The Mountain Hardware Time Interrupt Monitor/EXECUTIVE (TIME) is a software package designed to enable you to set up your computer to perform tasks while you use it for other things. You can program something to happen, say, every 15 seconds, or once after first waiting 13 days, 21 hours, 2 minutes and 12 seconds, or up to 255 days in 1-second increments. The use of the software is easy.

Here's the way it works. Every second the clock will generate an interrupt. This will be in the form of a RESTART instruction pushed on the bus. The 8080 will then go to the restart location and find a jump to the interrupt control software. The software then looks at a table you helped generate earlier that specifies a time when you wanted a certain task to be performed (you gave the task starting address). The time is in the form of an interval from now. For example, go do the task in 3 days, 2 hours, 10 minutes and 3 seconds. The task may be done once only or repeated after each time interval. If the proper time is not reached for task one, task two is looked at. Up to 20 tasks may be defined. If the time hasn't come for any of the tasks, a return from interrupt is performed. If time has come for a given task, that task is performed.

The following steps must be taken to use the interrupt software:

1. DISABLE INTERRUPTS.
2. PUT ØØH into each byte of the task table. (From Address 4ØF1 to 41FF would be adequate.)
3. PLACE A JUMP TO 4ØØØ at the restart location for which you have the clock board set up. (For a Restart 1, place C3 00 40 at Location 0008.) This causes a jump to 4000 whenever an interrupt occurs (every second using the MHI Interrupt Control Software).
4. DETERMINE WHAT YOUR TASK ADDRESS IS; i.e. where you want to do your task. Place the task software there; i.e. what it should do when the proper time comes. Do this for each task you may have up to 20 different tasks at 20 different times. Each task software may be at a different address.
5. LOAD in the Interrupt Control Software. (It may be re-assembled for any address if 4000 is not convenient.)

Steps 2-5 may be done in any order.

6. Now for each time a task is desired, do the following steps.

a. REG B = Hours
   Reg C = Days
   PUSH B
b. REG B = Seconds
   REG C = Minutes
   PUSH B
c. REG B = Task Address (Hi Byte)
   REG C = Task Address (Lo Byte)
   PUSH B
d. REG B = (0 = Static, 1 = Periodic)
   Static means interrupt occurs once after the time has elapsed.
   Periodic means the task is performed repeatedly after each time interval.
   REG A = (Task #, 0 - 19)
   Each task is given its own number in increasing order.
e. CALL SETTSK (4Ø63). This sets up Task Table.

7. Once all the tasks are set up, the procedure can be started by these steps:

a. REG A = Number of tasks; 0 = None)
   STA 40F1 (TSKTBL)
b. REG A = 1
   CALL SETALL (4ØBØ) Enables all tasks.
c. CALL ONJOB (40DC) Master Start.
d. REG A = 14
   OUTPUT REG A to the clock's highest port number.
   This starts one-second interrupts from the clock.
e. ENABLE INTERRUPTS.

8. Then RETURN, or go off and do something else. Every second an interrupt will occur which will cause a jump to the Interrupt Control Software. It will look at the times you listed and if that much time has gone by, the task software will be performed.

NOTE:

1. Your task software should contain a Return at its end. It need not save any registers.
2. Be sure that any software you are running has interrupts enabled if you want the clock to interrupt.
3. If you are using vectored interrupts, the instructions concerning Restart Locations should be changed appropriately for your vectored interrupt system.

The following routines will be helpful when using the Interrupt Control Software.

KILTSK (4ØA4)	Stops a task whose task number is in REG A.
LIVTSK (4ØAA)	Starts a task whose Task number is in REG A.
SETALL (4ØBØ)	Stops all task if REG A = Ø. Starts all tasks if REG A = 1.
RUNSTP (4ØE8)	Sets the Master Switch ON if REG A = 1. The Master Switch is handy if you have some of the tasks on, and some off, and then you want to temporarily turn off all interrupts. Using RUNSTP (or ONJOB/ OFFJOB) you can turn off the Master and then turn it back on, and only those tasks that were on will be active.
ONJOB (4ØDC)	Turns the Master Switch on.
OFFJOB (4ØE2)	Turns the Master Switch off.
SETTSK (4ØC3)	Routine to set up tasks in task table.

Mountain Hardware S-100 Clock Software - TIME
==

The MHI Timed Interrupt Monitor/Executive enables the user of MHI's Clock/Calendar board to perform background task scheduling based upon defined time intervals. Entry points to the software permit allocation of new tasks as well as run/stop control for a particular task or set thereof.
The executive maintains a table describing all on-line tasks. The format of this table is:

```
40F1   =-> Number of entries, 0 means none
40F2   =-> Master task-schedule-enable, 0 means disable
40F3   ++[ Task running (1), or stopped (0)
       ++[ Task periodic (1), or static (0)
       ++[ Scratch (days)
       ++[ Scratch (hours)
       ++[ Scratch (minutes)
       ++[ Scratch (seconds)
       ++[ Interval - Days
       ++[ Interval - Hours
       ++[ Interval - Minutes
       ++[ Interval - Seconds
       ++[ User task driver - Lo-order address
       ++[ User task driver - Hi-order address
```

The above '++[' sequence repeats for each task.

```
                        ;; T.I.M.E. -- M.H.I. CLOCK INTERRUPT SOFTWARE
                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

4000                            ORG     4000H

4000 F5         INTRPT: PUSH    PSW             ;; SAVE ALL REGISTERS
4001 C5                 PUSH    B
4002 D5                 PUSH    D
4003 E5                 PUSH    H
4004 21F140             LXI     H,TSKTBL        ;; POINT TO # OF ENTRIES
4007 7E                 MOV     A,M             ;; LOOK AT THIS NUMBER
4008 B7                 ORA     A               ;; ANY ENTRIES ???
4009 CA8E40             JZ      ENDJOB          ;; QUIT IF NONE FOUND
400C 32EE40             STA     NUMBER          ;; STORE THIS NUMBER
400F 23                 INX     H               ;; POINT TO RUN/STOP FLAG
4010 7E                 MOV     A,M             ;; LOOK AT IT
4011 B7                 ORA     A               ;; SET FLAGS
4012 CA8E40             JZ      ENDJOB          ;; IF STOPPED, QUIT
4015 23                 INX     H               ;; BUMP PAST THIS NONSENSE
4016 22EF40             SHLD    TSKPTR          ;; SAVE SECRET POINTER

4019 2AEF40     DOTASK: LHLD    TSKPTR          ;; GET TASK BLOCK POINTER
401C 7E                 MOV     A,M             ;; LOOK AT RUN/STOP
401D B7                 ORA     A               ;; RUNNING?
401E CA7740             JZ      NXTTSK          ;; IF NOT, TO NEXT TASK
4021 110500             LXI     D,5             ;; BUMP TO
4024 19                 DAD     D               ;;   SECONDS POSITION
4025 1E02               MVI     E,2             ;; # OF "59" COUNTS
4027 34         INC59:  INR     M               ;; COUNT THAT NUMBER
4028 7E                 MOV     A,M             ;; NOW LOOK AT IT
4029 FE3C               CPI     60              ;; ANY OVERFLOW
402B DA4040             JC      COMP            ;; NOPE, DO COMPARE
402E 3600               MVI     M,0             ;; YEP, CLEAR THIS NUMBER
4030 2B                 DCX     H               ;; NEXT IN LINE
4031 1D                 DCR     E               ;; MORE 59 COUNTS?
4032 C22740             JNZ     INC59           ;; DO TILL DONE
4035 34                 INR     M               ;; INC. HOURS
4036 7E                 MOV     A,M             ;; WHAT DID WE DO?
4037 FE18               CPI     24              ;; OVERFLOW?
4039 DA4040             JC      COMP            ;; IF NOT, DO COMPARE
403C 3600               MVI     M,0             ;; CLEAR HOURS
403E 2B                 DCX     H
403F 34                 INR     M               ;; DAYS:=DAYS+1
4040 2AEF40     COMP:   LHLD    TSKPTR          ;; FIND TASK BLOCK ADDRESS
4043 23                 INX     H
4044 23                 INX     H
4045 54                 MOV     D,H
4046 5D                 MOV     E,L             ;; DE => TEMP. TIME
4047 23                 INX     H
4048 23                 INX     H
4049 23                 INX     H
404A 23                 INX     H               ;; HL => SET TIME
404B 0604               MVI     B,4             ;; FOUR TO CHECK
404D 1A         CMPNXT: LDAX    D               ;; LOOK AT ONE VALUE
404E BE                 CMP     M               ;; COMPARE WITH FINAL
404F 13                 INX     D               ;; BUMP POINTERS
```


983-IBM-324

```
4050 23                 INX     H
4051 C27740             JNZ     NXTTSK          ;; MATCH FAILED - NEXT TASK
4054 05                 DCR     B               ;; DEC. # TO COMPARE
4055 C24D40             JNZ     CMPNXT          ;; DO TILL DONE
4058 7E                 MOV     A,M
4059 23                 INX     H
405A 66                 MOV     H,M
405B 6F                 MOV     L,A
405C 116140             LXI     D,RETPNT        ;; GET RETURN POINT
405F D5                 PUSH    D               ;; STICK IT ON THE STACK
4060 E9                 PCHL                    ;; OFF TO BOGUS LAND
4061 2AEF40    RETPNT:  LHLD    TSKPTR          ;; LOOK AT THIS TASK
4064 23                 DCX     H               ;; STOP STATIC TASK
4065 7E                 MOV     A,M
4066 B7                 ORA     A
4067 C26E40             JNZ     RETPT2
406A 2D                 DCX     H
406B 3600               MVI     M,0
406D 23                 INX     H
406E 1E04      RETPT2:  MVI     E,4             ;; CLEAR TEMP TIME SLOT
4070 23        RETPT3:  INX     H
4071 3600               MVI     M,0
4073 1D                 DCR     E
4074 C27040             JNZ     RETPT3

4077 3AEE40    NXTTSK:  LDA     NUMBER          ;; SEE HOW MANY YET TO DO
407A 3D                 DCR     A
407B CA8E40             JZ      ENDJOB          ;; QUIT IF DONE
407E 32EE40             STA     NUMBER
4081 2AEF40             LHLD    TSKPTR
4084 110C00             LXI     D,12
4087 19                 DAD     D
4088 22EF40             SHLD    TSKPTR
408B C31940             JMP     DOTASK

408E E1        ENDJOB:  POP     H               ;;FINISH
408F D1                 POP     D
4090 C1                 POP     B
4091 F1                 POP     PSW
4092 FB                 EI
4093 C9                 RET
               ;;
               ;; THE FOLLOWING ROUTINES PROVIDE FOR CONTROL
               ;; OF THE EXECUTIVE.
               ;;
4094 C5        GETADR:  PUSH    B
4095 2600               MVI     H,0
4097 6F                 MOV     L,A
4098 29                 DAD     H
4099 29                 DAD     H
409A E5                 PUSH    H
409B 29                 DAD     H
409C C1                 POP     B
409D 09                 DAD     B
409E 01F340             LXI     B,TSKTBL+2
40A1 09                 DAD     B
```

```
40A2 C1               POP     B
40A3 C9               RET

                 ;; STOP TASK IN A REG.
40A4 CD9440      KILTSK: CALL    GETADR
40A7 3600             MVI     M,0
40A9 C9               RET

                 ;; START TASK IN A REG.
40AA CD9440      LIVTSK: CALL    GETADR
40AD 3601             MVI     M,1
40AF C9               RET

                 ;; SET ALL TASKS TO A (RUN/STOP)
40B0 21F140      SETALL: LXI     H,TSKTBL
40B3 E601             ANI     1
40B5 46               MOV     B,M
40B6 04               INR     B
40B7 23               INX     H
40B8 23               INX     H
40B9 110C00           LXI     D,12
40BC 77          SETAL2: MOV     M,A
40BD 19               DAD     D
40BE 05               DCR     B
40BF C2BC40           JNZ     SETAL2
40C2 C9               RET

                 ;; SET A TASK TO A (RUN/STOP)
40C3 CD9440      SETTSK: CALL    GETADR
40C6 3600             MVI     M,0
40C8 23               INX     H
40C9 70               MOV     M,B
40CA 010A00           LXI     B,10
40CD 09               DAD     B
40CE D1               POP     D
40CF 3E03             MVI     A,3
40D1 C1          SETTS2: POP     B
40D2 70               MOV     M,B
40D3 2B               DCX     H
40D4 71               MOV     M,C
40D5 2B               DCX     H
40D6 3D               DCR     A
40D7 C2D140           JNZ     SETTS2
40DA EB               XCHG
40DB E9               PCHL
                 ;; ENABLE SOFTWARE CONTROL
40DC 21F240      ONJOB:  LXI     H,TSKTBL+1
40DF 3601             MVI     M,1
40E1 C9               RET
                 ;; DISABLE SOFTWARE CONTROL
40E2 21F240      OFFJOB: LXI     H,TSKTBL+1
40E5 3600             MVI     M,0
40E7 C9               RET
                 ;; SET MASTER ENABLE/DISABLE TO A
40E8 E601        RUNSTP: ANI     1
40EA 32F240           STA     TSKTBL+1



40ED C9               RET
40EE             NUMBER: DS      1
40EF             TSKPTR: DS      2
40F1             TSKTBL: DS      256
41F1 =           LASTLOC EQU     $

A>
```

THEORY OF OPERATION

The S-100 Clock communicates to the central processor through a block of 16 I/O ports. The address of this block is determined by the setting of S1 switches 2 through 5. These select the high order 4 bits of the eight-bit I/O address for each of the 16 ports.

The block address of each port, 0 through 15, corresponds to the address of a location in the 16X4 RAM at U29. Locations 0 through 14 of this RAM each hold one of the 4-bit BCD digits of the current time. The contents are as follows:

Loc	0	0-9	Hundreds of microseconds.
	1	0-9	Milliseconds.
	2	0-9	Tens of milliseconds.
	3	0-9	Hundreds of milliseconds.
	4	0-9	Seconds.
	5	0-5	Tens of seconds.
	6	0-9	Minutes.
	7	0-5	Tens of minutes.
	8	0-9	if Loc 9=0 or Loc 9=1
		0-3	if Loc 9=2, Hours.
	9	0-2	Tens of hours.
	10	0-9	Days.
	11	0-9	Tens of days.
	12	0-9	Hundreds of days.
	13	0-9	Thousands of days.
	14	0-9	Ten-thousands of days.
	15		Used for interrupts.

Thus, an input operation to block address 4 will return the current number of seconds in the lower four bits of A. The information in U29 is updated by the clock every 100 microseconds. A full clock read requires 15 input operations. Since it is possible for a clock tick (update) to occur between two of the input operations, a flag is included with the data to resolve any ambiguity. If the most significant bit of A comes back set after a clock input, then the clock has not ticked since the previous input.

The clock is set by output operations to the addresses of the time digits in the block of ports. S1 (Switch 1) must be closed to write enable the clock. Outputs to block addresses 0-14 cause the lower 4 bits of the AC to be written to the corresponding digit address in U29 and STOP THE CLOCK. The clock remains stopped until the processor reads any of the time digits. The clock will then tick 100 microseconds later and continue to update every 100 microseconds.

IMPLEMENTATION

The S-100 Clock operates on the principle of character processing. Every 100 microseconds the characters that represent the time are fetched from memory, processed, and returned. When the processing has ended, the time has been advanced by 100 microseconds. As in computer programming, the processing is no more than following a series of rules and making some simple decisions. The rules are as follows:

1. Start with Loc 0. (Hundreds of microseconds).
2. Fetch the digit and add 1 to it.
3. If the add carries, replace the digit with 0 and go to Step 2, else replace the digit with the sum and stop.
4. Carry is defined to be the number 10 or the number 6 if the address is either 5 or 7, (tens of seconds or tens of minutes) or the number 4 if the address is 8 (hours) and Loc 9 contains the number 2 (20 hours), or the number 3 if the address is 9 (tens of hours).

The time digits are processed in the 4-bit wide digital loop that begins at the output of U29, proceeds through the inverters in U28, is incremented at U18, latched or cleared at U21 and finally returned to U29. The inverting step is needed because the 74c89 at U29 gives an inverted output.

The address used in the processing is generated by the 4-bit counter U15. This address starts at 0 (hundreds of microseconds) and is incremented after each digit is fetched from U29. The address is transferred from U15 to U14 at the start of each digit cycle where it is latched and presented to the address lines of U29.

The carry logic is implemented with U19, U13, some gates in U1, U2, U11, U28 and a flip-flop in U20. U19 decodes the result of the increment at U18 and outputs a high level on Pin 5 if the result is a 3, Pin 2 if the result is 4, Pin 15 if the result is 6 and Pin 6 if the result is 10. U13 is used to determine which time digit is being incremented by decoding the address used at U29. The flip-flop at Pin 1 of U20 is set whenever a 2 is loaded into tens of hours.

Thus, if tens of seconds or tens of mintues is incremented from 5 to 6, Pin 3 of U12 will be high and Pin 15 of U19 will be high causing a low on Pin 4 of U2 and therefore a high on Pin 10 of U1 which is the carry line. If hours is incremented from 3 to 4 and tens of hours has a 2 in it, then Pin 1 of U20 will be high, Pin 4 of U19 will be high, and Pin 10 of U12 will be high. This causes Pin 9 of U1 to go low and again setting carry. If tens of hours is incremented from 2 to 3, Pin 11 of U3 will be low and Pin 12 of U28 will be low , causing a high on Pin 10 of U11 and therefore a low on Pin 11 of U11 and thereby a high on Pin 10 of U1, carry. If any digit is incremented from 9 to 10, pin 6 of U19 goes high and causes Pin 11 of U11 low, and again Pin 10 of U1 high, thus carry.

The clock timing is provided by the crystal oscillator with output at Pin 10 of U6. This is a 1 MHz signal which is always present during clock operation. The 1 MHz is divided by 100 in U3 and the outputs of U3 are added to generate a 1 microsecond pulse every 100 microseconds. This pulse initiates the incrementing of the time and is generated as long as Pins 7 and 15 of U3 remain low. These pins are pulled high by the SR flip-flop at Pin 3 of U2 if the processor does an output to a time digit and the clock is write enabled by S1-1. In going high, Pin 3 of U2 stops the clock from incrementing and clears U3 so that the next 100 microsecond period will start when Pin 3 goes low at the next intput to the processor of a time digit.

The 5-volt supply for the CMOS portion of the clock is regulated by the circuit at U32. This configuration uses the very low power Zener Effect of Q2, together with the micro-power op amp LM4250, to control the current through Q1. When the bus is powered, the clock uses the +16 volts as the input to the regulator. When the bus is powered down the regulator is switched to the on-board 9-volt battery. The S-100 Clock draws less than 2 Ma from its battery when the computer is powered down. If a rechargeable battery is used, R6 can provide charging current when the computer is on. If the clock is to be operated for very long times without bus power, an external 12-volt battery pack is recommended.

TROUBLE SHOOTING

The clock is one of the only boards that can be almost completely tested without the rest of the computer. The most useful tool in checkout is an oscilloscope. A frequency counter is also useful, but on essential.

The first step is to connect the 9-volt battery. The clock need not be connected to the S-100 bus. Measurements should be made to see that the supply voltage to the CMOS IC's is approximately 5 volts. Over-voltage cannot be worse than 9-volts won't hurt anything. If the voltage is too high, check into the voltage regulator circuit at U32. If the voltage is shorted to ground however, remove all the CMOS parts and check again. If there is still no +5, trace the regulator circuit.

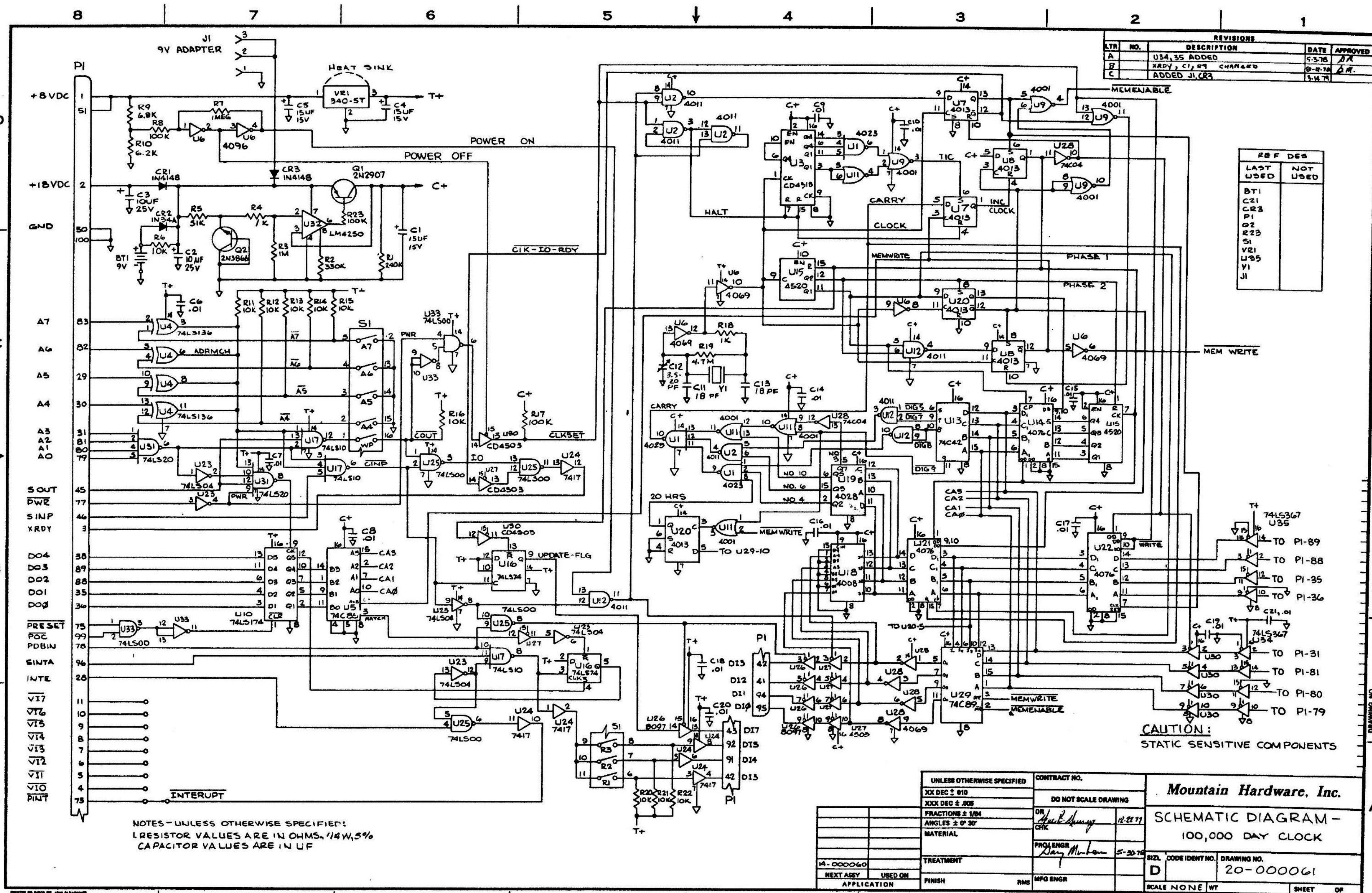
Once the 5 volts is OK to the CMOS parts, check the 1 MHz signal at Pin 10 of U6. If no signal, then remove all the CMOS logic except U6. If still no signal, then either U6 is bad or the crystal is bad.

After verifying the 1 MHz, check the state of Pin 3 of U2. If it is low, the clock should be running. If it is high, the clock has come up stopped. If it is high, poke it with a grounded wire and it should go low and start the clock. Once the clock is started, 1 microsecond pulses will be generated at Pin 3 of U9 every 100 microseconds.

The big picture is at Pin 12 of U15. This line is Phase 2 used in the digit increment process. The scope should show a 2 microsecond pulse every 100 microseconds. This is the primary signal but after 2 microseconds another pulse should be visible, but fainter. This is also true for the third and fourth pulse positions across the face of the scope. At the fifth position a pulse will flash every second and next to it one every 10 seconds, and so on.

If this picture is correct, then the incrementing and most all of the functions of the clock are functioning and it is time to put the board in the computer and run the testing software.

If a good frequency counter is available, the 1 MHz oscillator can be tuned to give good long-term accuracy. It is also possible to tune the oscillator by attaching a wire to Pin 10 of U6 and placing the wire close to a short-wave radio tuned to WWV at 5 MHz. The fifth harmonic of the clock will beat against WWV and allow very precise setting by tuning for the lowest beat note.



Mountain Hardware, Inc.
SCHEMATIC DIAGRAM - 100,000 DAY CLOCK
DRAWING NO. 20-000061
SIZE D
SCALE NONE
CAUTION: STATIC SENSITIVE COMPONENTS
NOTES - UNLESS OTHERWISE SPECIFIED:
1. RESISTOR VALUES ARE IN OHMS, 1/4W, 5%
CAPACITOR VALUES ARE IN UF
REVISIONS
A U34, 35 ADDED
B XRDY, C1, R9 CHANGED
C ADDED J1, CR3
REF DES
LAST USED
NOT USED
BT1 C21 CR3 P1 Q2 R23 S1 VR1 U35 Y1 J1
J1 9V ADAPTER
HEAT SINK
POWER ON
POWER OFF
CLK-IO-RDY
MEM WRITE
MEMENABLE
PHASE 1
PHASE 2
CARRY
CLOCK
HALT
INTERRUPT
UPDATE-FLG
20 HRS
14-000060
NEXT ASSY
USED ON
APPLICATION

983-IBM-324

PARTS LIST

R1	240 K
R2	330 K
R3, R7	1 M
R4, R18	1 K
R5	51 K
R6, R11-16 R20-R22	10 K
R8, R17, R23	100 K
R10	6.2 K
R19	4.7 M
R9	6.8K
BT1	9V Battery
CR1, CR3	IN4148
CR2	IN34A
Q1	2N2907A
Q2	RCA 2N3866
S1	8 Pos. Dip Switch
VR1	7805
Y1	1.0000 MHz Crystal
J1	Adapter Jack
C6-C10, C14-C21	.1 μF/25V
C2, C3	10 μF/25V
C4, C5, C1	15 μF/15V
C11	18 pF
C12	3.5-20 pF
C13	36 pF

(1) Set #6 hardware
(1) P.C. Card
(1) Manual
(1) 8-Pin Socket
(16) 16-Pin Sockets
(18) 14-Pin Sockets

U1	4023
U2, U12	4011
U3	4518
U4	74LS136
U5	74C85
U6, U28	4069
U7, U8, U20	4013
U9, U11	4001
U10	74LS174
U13	74C42
U14, U21, U22	4076
U15	4520
U16	74LS74
U17	74LS10
U18	4008
U19	4028
U23	74LS04
U24	7417
U25	74LS00
U26	74367
U27, U30	4503
U29	74C89
U31	74LS20
U32	LM4250
U33	74LS00
U34, U35	74LS367

WARRANTY

Your factory-built 100,000 Day Clock board is warranted against defects in materials and workmanship for a period of six (6) months from date of delivery. We will repair or replace products that prove to be defective during the warranty period, provided they are returned to Mountain Hardware. No other warranty is expressed or implied. We are not liable for consequential damages. We reserve the right to refuse to repair any product that in our opinion has been subjected to abnormal electrical or mechanical abuse. Products out-of-warranty are subject to a minimal service fee.

Please feel free to contact us if you have any questions or problems.

Mountain Hardware

Located in the Santa Cruz Mountains of Northern California, Mountain Hardware, Inc. is a computer peripheral manufacturer dedicated to the production of use-oriented high technology products for the microcomputer. On-going research and development projects are geared to the continual supply of unique, innovative products that are easy to use and highly complementary in a broad variety of applications.

300 Harvey West Boulevard
Santa Cruz, CA 95060
(408) 429-8600

100,000 DAY CLOCK ADDENDUM
6 JUNE, 1980

This is an addendum to the 100,000 Day Clock Manual. This addendum contains information about changes made to the circuit. The circuit has been changed so that it will work with most S100 bus systems. However, there is still a great variety in S100 systems. If you are having problems with your 100,000 Day Clock, then read this addendum carefully. If your problem persists after following the steps below, then feel free to contact Mountain Hardware.

If you are having a problem, the first thing to check is the XRDY and PRDY lines on the S100 bus. The 100,000 Day Clock uses the XRDY line (pin 3 on the S100 bus) to synchronize the Clock with the S100 bus. Some S100 systems use the PRDY line (pin 72 on the S100 bus) and not the XRDY line. You must find out which line your computer uses for the ready function. If your system uses the PRDY line, you must cut the trace to pin 3 and jumper that line to pin 72 on the card edge connector. Newer boards have a jumper area on the board for the XRDY and PRDY lines. The 100,000 Day Clock is shipped from the factory with the XRDY line in use.

The next thing to check is that the board is properly addressed. Carefully set the address switches and make sure that they are all the way up or down. Make sure they are set for the addresses that your program is using.

Changes have been made to the circuit. Refer to the schematic in he back of the manual. Resistors R11, R12, R13, R14 and R15 have been changed from 10K ohms to 2K ohms. Additionally, a 470 ohm resistor has been inserted in series with the signal coming from U17, pin 6 and going to both U25, pin 2 and U23 pin 9. If you cannot get the Clock to return a good time value on a 4Mhz system, you might try jumpering around the 470 ohm resistor on the back of the board. Another change to the circuit is a 270 pico-farad capacitor has been added between U16, pin 11 and U25, pin 8. The integrated circuit U17 has been changed from a 74LS10 to a 74L10. Again, if you have trouble with a 4Mhz system, you might try replacing the 74L10 at location U17 with a 74LS10. Also try removing the 270pf capacitor. Any one of these modifications (i.e., jumping the 470 ohm, the 74L10, and the 270 pf capacitor) or a combination of some or all of the modifications may solve the problem. We have included these parts installed on the board because of the wide variety of S100 systems available.

If you are going to use the interrupt feature of the 100,000 Day Clock, you must set the R0, R1, and R2 switches to the restart address, assuming you use 8080 interrupts. The 100,000 Day Clock will place the contents of those switches on the bus. The S100 bus is responsible for making the other five bits of the data bus high logic levels.

All of the changes have been made so that the 500,000 Day Clock will work on most of the many varieties of S100 bus systems on the market.

Dear Customer:

Thank you for your interest in Mountain Computer peripherals. Please take note of the additional pages with your manual. These pages include corrections for pages 3, 7, and 9 of the current manual plus a two-page addendum. We recommend that you keep these with your manual.

Thank You,
MOUNTAIN HARDWARE, INC.

SETTING THE FREQUENCY

Your 100,000 Day Clock has been factory assembled, burned in and tested. The 1.0000 MHz time base has been accurately set to within .001%. Vibrations or extreme temperatures can cause slight changes to the time base and may produce noticeable errors. If these errors are noticed, or if you desire to set this frequency more precisely for your environment, an accurate frequency counter and a small non-metallic screwdriver are required.

Connect the frequency counter with the ground lead to the screw on the regulator and the positive lead to Pin 10 of U6. Adjust C12 for a frequency as close to 1.000000 MHz as possible. Be sure the clock is at the same operating temperature as its normal environment.

SETTING THE PORT ADDRESS

The clock board occupies 16 port addresses on the S-100 system bus. Changing the switches labelled A4, A5, A6 and A7 can change the clock to respond to different port addresses. Table 1 shows the relationships between switch positions and addresses.

PORT ADDRESSES		SWITCH POSITION			
DECIMAL	HEX	A4	A5	A6	A7
0 - 15	0 - F	0	0	0	0
16 - 31	10 - 1F	1	0	0	0
32 - 47	20 - 2F	0	1	0	0
48 - 63	30 - 3F	1	1	0	0
64 - 79	40 - 4F	0	0	1	0
80 - 95	50 - 5F	1	0	1	0
96 -111	60 - 6F	0	1	1	0
112 -127	70 - 7F	1	1	1	0
128 -143	80 - 8F	0	0	0	1
144 -159	90 - 9F	1	0	0	1
160 -175	A0 - AF	0	1	0	1
176 -191	B0 - BF	1	1	0	1
192 -207	C0 - CF	0	0	1	1
208 -223	D0 - DF	1	0	1	1
224 -239	E0 - EF	0	1	1	1
240 -255	F0 - FF	1	1	1	1

TABLE 1

For port selection purposes, a 1 (one) means the switch is closed (on). A 0 (zero) means the switch is open (off).

Address your clock to a set of ports that are not presently used by your other peripherals. If possible, we recommend that you use ports 32-47 (20-2F HEX) to standardize with our software.

RESTART ADDRESS		RESTART SWITCHES		
DECIMAL	HEX	R0	R1	R2
0	0000	0	0	0
8	0008	1	0	0
16	0010	0	1	0
24	0018	1	1	0
32	0020	0	0	1
40	0028	1	0	1
48	0030	0	1	1
56	0038	1	1	1

TABLE 4

Software can then be placed at the re-start location to service the interrupt.

The other type of interrupt on the 100,000 Day Clock board is a Vectored Interrupt. This is available for users of a vectored interrupt controller. To use this feature, the trace between I and PINT in the lower-left corner of the clock should be cut and a jumper placed between I and one of the VI pins labelled VI0-VI7 and also in the lower-left corner of the clock board. For more information on the vectored interrupt see the details with your Vectored Interrupt board.

With all interrupts care must be taken to avoid conflicts between peripherals requesting an interrupt.

READING THE CLOCK - PRINTING THE TIME

Since the clock stores the time on-board in the form of BCD digits, displaying the time is very easy. The lower 4 bits of each clock digit hold the actual information. Here is a basic program which prints the time.

```
10      C=32:  REM CLOCK'S LOWEST PORT ADDRESS
20      FOR I = 9 TO 4 STEP - 1
30      D=INP(C+I): REM GET A DIGIT
35      D=D - INT (D/16)* 16: REM REMOVE TOP 4 BITS
40      PRINT D;: REM PRINT DIGIT
50      IF I=8 OR I=6 THEN PRINT ":";
60      NEXT I
70      PRINT
80      END
```

CALENDAR ROUTINES

This software package was developed to enable you to translate the day information on the clock board (0 - 99,999 days) to date information in the form of month, day, year, day of week.

Using this software is simple:

1. Set location "CLKPRT" (4400) to the lowest port address of your clock board.

2. Call "READ" as a machine language subroutine. (Location 4200).

3. Read the returned information from RAM storage area.

   MONTH is the month 1 = JAN, 2 = FEB...) (Location 4402).
   DATE is the day (1-31) (Location 4403).
   YEAR is the year (1978...) (Locations 4404, 4405 - Low, High).
   DAY is the day of the week (0 = SAT, 1 = SUN..) (Location 4401).

The above addresses are given in hex and refer to the source listing of the calendar routines.

The calendar routine assumes that the DAYS digits of your clock board have been set to the number of days since December 31, 1977 That is, January 1 1978 is DAY 00001.

That is all there is to it.

Here is a BASIC program to perform this task:

```
10  L = 17408: REM START OF RAM STORAGE FOR CAL ROUTINES
20  POKE L, 32: REM CLOCK AT 32 (LOWEST PORT # ON CLOCK)
30  Y = USR (16896): REM CALL ROUTINE
40  PRINT PEEK (L+2); "/", PEEK (L+3); "/";
50  PRINT PEEK (L+4) + 256*PEEK (L+5)
60  END
```

This prints the date as MM/DD/YYYY.