TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
INTRODUCTION		
	FEATURES	1
	INSTALLATION	1
	POINTS OF INTEREST	2
	CHECKOUT	2
CHAPTER 1		
	HOW TO PROGRAM A READ ROUTINE	5
	HOW TO READ THE DATE	9
	SUMMARY	11
CHAPTER 2		
	HOW TO WRITE A SET ROUTINE	12
	ADJUST THE TIME	13
	INTERRUPTS	14
	SUMMARY	14
CHAPTER 3		
	FUNCTIONAL DESCRIPTION	15
	DISCRIPTION OF OPERATION	16
	READ TIME AND DATE	16
	SET TIME AND DATE	17
	ADJUST TIME +-30 SECONDS	17
APPENDIX		
A	I/O PORTS	18
B	I/O PORT ADDRESS SELECT	19
C	CLOCK DEMO PROG. (BASIC LISTING)	21
D	CLOCK READ PROG. (ASSEMBLER LISTING)	24

INTRODUCTION

FEATURES

The ComputerWatch utilizes the popular MSM5832 real time CLOCK/CALENDAR chip designed for use in bus-oriented microprocessor applications. A 32.768 Hz crystal controlled time base will provide addressable 4-bit I/O data of SECONDS, MINUTES, HOURS, DAY OF WEEK, DATE, MONTH, YEAR. The data access is controlled by a 4-bit address, read, write, and hold inputs.

Features include:

- Time in Hours, Minutes, Seconds
- Program selectable 24 hour or 12 hour format
- Date in Month, Day, Year, Day of Week, Leap recognition
- Fast time and date setting
- +-30 Second adjust
- Four hard interrupts. 1024Hz (approx. 1 millisecond) 1 Hz, 1 minute, 1 hour
- Crystal controlled time base
- Latched input and output ports
- On board battery backup power
- Automatic power off sensing
- Simple programming interface

Applicational uses of the ComputerWatch features are limited only to the imagination. In any application where TIME or DATE is useful the ComputerWatch can be used. Applications would include, time or date stamping of reports, checks, letters, file updates, calculation of time intevals, recording measurements, ect. I am sure you can think of many more uses for the ComputerWatch.

INSTALLATION

Before installing the ComputerWatch, the address must be selected by setting the switches at location SW1 on the board. Look up the address you have selected in Appendix B and set the switches as shown. If possible use address range 128 - 131. This is the range used in the programming examples in this manual.

NOTE- THE COMPUTERWATCH USES FOUR I/O ADRESSES. BE SURE THAT THEY DO NOT OVERLAP OTHER DEVICE ADDRESS RANGES OR THE BOARD WILL NOT OPERATE PROPERLEY.

Ensure that all the ICs are properly installed with pin 1 toward the left side of the board. Inspect the board for shorts or damaged components. If damage is found return the board to place of purchase for replacement.

Carefully install the board in the system with power off, being certain that pins 1 and 50 line up with the ground pins on the S100 bus connector. Plugging the board in backwards will cause damage to the board and void your warranty.

POINTS OF INTEREST

There is a NI-CAD 3.6V battery mounted on the circuit board that will keep the clock circuit running when power is removed from the system. The battery will be charging whenever the system is powered on. The battery has sufficent capacity to keep the clock circuit running for months between chargings. It will take the battery about eight hours to reach full charge.

There is a trimmer capacitor located beside the clock chip U5. This capacitor is used to fine tune the clock time base circuit. The trimmer capacitor has been adjusted at the factory to the crystal frequency and should not be tampered with.

CHECK OUT

Appendix C contains a listing of a program that will check out all the time and date features of the ComputerWatch. It is highly recommended that you enter this program and become familiar with the features of the ComputerWatch. The rest of this section deals with the operation of this DEMO program and allows you to check out the operation of the board.

After you have entered the program and are sure there are no errors, RUN the program. The following will appear on the screen:

```
TYPE T TO DISPLAY TIME/DATE
TYPE R TO RESET THE TIME/DATE
TYPE A TO ADJUST THE TIME +-30 SECONDS
TYPE E TO END PROGRAM
```

Type T and then RETURN, from here on it will be assumed that you will press the RETURN key after each keyboard entry. The following display will be returned:

```
THE DATE IS M M / D D / Y Y
TODAY IS ( day of week )
THE TIME IS H H : M M : S S AM/PM
```

The time and date read should be somthing reasonable to your time. If you are in a time zone other than Pacific time you will want to reset the time and possibly the date. Before attempting to reset the date type T again and satisfy yourself the the board is working by comparing the first response with the second response. The time will have increased. To set the clock type R. The following message will appear.

```
YEAR TENS
```

Type the tens digit of the year or 8 for 81. The next message will be.

YEAR UNITS

Type the units digit of the year or 1 for 81. Continue typing the requsted information until the following question appears:

DAY OF WEEK, 0 = SUN, 1 = MON, ect.

The day of week is kept in the ComputerWatch as seven numbers. They are:

0 = SUNDAY
1 = MOMDAY
2 = TUESDAY
3 = WEDNESDAY
4 = THURSDAY
5 = FRIDAY
6 = SATURDAY

Type the number that applies to the day you want to enter. continue entering the requested information until the following question appears:

IS THIS LEAP YEAR

Your response should be a Y for yes or a N for no. The next question,

12 OR 14 HOUR FORMAT

should be answered with a 12 or 24. 12 hour format will act like a normal clock with AM OR PM notation. 24 hour format or military time will keep time up to 23:59:59 with 00:00:00 being midnight and 12:00:00 being noon time. The next question,

AM OR PM

you would respond to by typing AM or PM. If you had selected 24 hour format this question would not have appeared. The following message will be displayed:

```
THE DATE IS M M / D D / Y Y
TODAY IS ( day of week )
THE TIME IS H H : M M : S S AM/PM

TYPE T TO DISPLAY TIME/DATE
TYPE R TO RESET THE TIME/DATE
TYPE A TO ADJUST THE TIME +-30 SECONDS
TYPE E TO END THE PROGRAM
```

Now try the +-30 second adjust routine by typing A. The same screen as above will appear except that the minutes may have advanced by one and the seconds reset to zero. If the seconds are greater than 30 then one will be added to the minutes units digit and the seconds reset to zero. If the seconds are less than 30, then only the seconds would be reset.

You can exit the program by typing E.

The next two chapters will show you how to program the ComputerWatch so that you can interface it into your programs and applications.

Chapter 1 will show you how to program read operations and Chapter 2 will show you how to set or write time/date data to the ComputerWatch.

HOW TO PROGRAM A READ ROUTINE

The CLOCK/CALENDAR chip used in the ComputerWatch has 13 four bit registers. This means that all data stored in the clock chip registers is numeric. The 13 registers have the following meaning, as shown in TABLE 1.

REGISTER/ADDRESS	DATA DESCRIPTION	VALUES
0	Seconds Units Digit	0 TO 9
1	Seconds Tens Digit	0 TO 5
2	Minutes Units Digit	0 TO 9
3	Minutes Tens Digit	0 TO 5
4	Hours Units Digit	0 TO 9
5	Hours Tens Digit	0 TO 1
6	Day of Week Digit	0 TO 6
7	Day of Month Units Digit	0 TO 9
8	Day of Month Tens Digit	0 TO 3
9	Month Units Digit	0 TO 9
10	Month Tens Digit	0 TO 1
11	Year Units Digit	0 TO 9
12	Year Tens Digit	0 TO 9

TABLE 1 CLOCK REGISTER LAYOUT

If you examine TABLE 1 you will find that each digit of information has a register address. When programming you have to specify the address for each digit. FIGURE 1 is a chart that shows in a different form, the same information as TABLE 1. You will be using these charts from now on. Examine the chart and become familiar with it. As you progress, the charts will have more information added to them.

DATA DESCRIPTION	SECONDS		MINUTES		HOUR		DAY OF WEEK	DAY		MONTH		YEAR	
	UNITS	TENS	UNITS	TENS	UNITS	TENS		UNITS	TENS	UNITS	TENS	UNITS	TENS
REG	0	1	2	3	4	5	6	7	8	9	10	11	12
BITS 8	*		*		*			*		*		*	*
BITS 4	*	*	*	*	*		*	*		*		*	*
BITS 2	*	*	*	*	*	*	*	*	*	*		*	*
BITS 1	*	*	*	*	*	*	*	*	*	*	*	*	*

*ACTIVE BITS

FIGURE 1 PROGRAMMING CHART

The next thing you need to know is how to raise the HOLD signal. This signal is used to prevent the internal clock registers from being rolled over during a read or write operation. Without this feature, it could be possible to read a units digit of 9, and before the tens digit could be read the clock could count the units register causing a rollover to the tens register. The result would be nine units off. For example, a 49 would read as a 59. The use of the HOLD signal will eliminate this undesirable condition. It is, however, not necessary to raise the HOLD signal to read clock data. In fact it might be undesirable in some cases to do so. You will be given an example of one case later on in this chapter.

Assuming that you have already set the clock using the demo program you will now create a small program to read the seconds units digit. Type the following program in BASIC.

NOTE- Only two of the I/O port addresses are used. These two addresses will be refered to as the Address Port and the Data Port. Refer to Appendix B for the correct addresses for the range you have selected. In the following examples you will have to use different values for C1 and C2 if you have selected a address range other than 128 - 131.

```
010 C1 = 130 : C2 = 129
020 OUT C2,16
030 OUT C1,32
040 X = INP(C1)
050 PRINT X
060 OUT C2,0 : END
```

Run the program and see what happens. Run it a number of times. You will notice that the number displayed is changing. This is because you have been reading the seconds units digit. Its value should change every second.

EXPLANATION

The first instruction at line 010 sets the Address and Data Port I/O address into variable C1 and C2. Line 020 raises the HOLD signal by writing a 16 to the Data Port. Line 030 sets the address of the register you want to read into the Address Port. You must add 32 to the register value of the register address of the digit you want to read. In this case you wanted to read the seconds units digit, who's register address is 0. By adding 32 to 0 the actual value for the address would be 32. If, for example you want to read the hour tens digit, you would add 5 to 32 for an actual address of 37. The chart in figure 2 reflects this new information. Line 040 will read the data from the selected register from the Address Port and store it in variable X. Line 060 will lower the HOLD signal to allow the clock to continue counting by putting a 0 in the Data Port. The HOLD signal should never be held up longer than one second. There must be a delay of 150 microseconds after the HOLD is raised. When using BASIC, this delay will not be necessary due to the time BASIC takes to interpret the next instruction. If, however, you plan to use an assembler or machine language, then this delay must be provided.

DATA DESCRIPTION	SECONDS		MINUTES		HOUR		DAY OF WEEK	DAY		MONTH		YEAR	
	UNITS	TENS	UNITS	TENS	UNITS	TENS		UNITS	TENS	UNITS	TENS	UNITS	TENS
REG	0	1	2	3	4	5	6	7	8	9	10	11	12
+32	32	33	34	35	36	37	38	39	40	41	42	43	44
BITS 8	*		*		*			*		*		*	*
BITS 4	*	*	*	*	*		*	*		*		*	*
BITS 2	*	*	*	*	*	*	*	*	*	*		*	*
BITS 1	*	*	*	*	*	*	*	*	*	*	*	*	*

*ACTIVE BITS

FIGURE 2 PROGRAMMING CHART (expanded)

For a variation on the previous routine enter the following program, and then run it.

```
010 C1 = 130
020 OUT C1,32
030 X = INP(C1)
040 IF Y=X THEN 030
050 Y=X
060 PRINT X : GOTO 030
```

EXPLINATION

Notice in this program the HOLD signal was not raised, therefore it was not necessary to reference the Data Port (C2). The program only reads the seconds units digit and is not concerned about digit rollover. Line 040 checks to see if there was a change in the seconds digit. If there was a change, the new digit is printed and the varibale Y is given the new value.

The next step would be to read the digits from all the registers concerned with Hours, Minutes, and Seconds. Before you do this however, you need to learn about the option bits associated with the selection of 12/24 hour format and AM/PM. Refer to FIGURE 3.

The hours ten digit contains the information to determine what format is being used, 12 or 24 hour. It also contains the AM/PM information. Bit 4 is used to determine AM/PM and bit 8 is used for 12/24 hour format.

DATA DESCRIPTION	HOUR UNITS	HOUR TENS
REG	4	5
+32	36	37
BITS 8	*	24
BITS 4	*	AM/PM
BITS 2	*	*
BITS 1	*	*

BIT 8 ON 24 HOUR FORMAT
BIT 8 OFF 12 HOUR FORMAT
BIT 4 ON PM
BIT 4 OFF AM

FIGURE 3 TIME OPTION BITS

The following program will read the time digits and display them as H H : M M : S S AM/PM

```
010 C1 = 130 : C2 = 129
020 DIM T(6)
030 OUT C2,16
040 LET I=0
050 FOR D = 37 TO 32 STEP -1
060 OUT C1,D
070 T(I) = INP(C1)
080 I=I+1
090 NEXT D
100 B=T(0)
110 IF B>7 THEN P$ = " "
120 IF T(0)>7 THEN T(0) = T(0)-8
130 IF B<8 AND T(0)>3 THEN P$ = "PM"
140 IF B<8 AND T(0)<4 THEN P$ = "AM"
150 IF T(0)>3 THEN T(0) = T(0)-4
160 PRINT "THE TIME IS ";T(0);T(1);":";T(2);T(3);":";T(4);T(5);P$
170 OUT C2,0 : END
```

Check to be sure you entered the program correctly and then RUN it. The display on the screen should be:

THE TIME IS H H : M M : S S AM/PM

EXPLINATION

The array T has been set up for six elements at line 020. The FOR loop between 050 and 090 allows the digit register address's to be stepped down each time a digit is read from the Address Port. The digit is stored in array T and is subscripted by I. After all the time digits are read the IF statements between 110 and 150 check for 12 or 24 hour format and AM/PM. The PRINT statement will print the contents of the array T and insert a ":" between the hours minutes and seconds. You must strip the option bits before using them, otherwise the value you display will be wrong. The code at line 120 and 150 does this for the 8 and 4 bit. You may want to add some remarks to the program and save it for future use.

HOW TO READ THE DATE

Refer back to FIGURE 2 and look at register 6 of the Day of Week. The day of the week is stored in this register as one of six digits. For example:

0 = SUNDAY
1 = MONDAY
2 = TUESDAY
3 = WEDNESDAY
4 = THURSDAY
5 = FRIDAY
6 = SATURDAY

FIGURE 4 shows the option bits that you must be concerned with when working with the date digits.

DATA DESCRIPTION		DAY	
		UNITS	TENS
REG		7	8
+32		39	40
BITS	8	*	
	4	*	28/29
	2	*	*
	1	*	*

BIT 4 = 0 (28 DAYS IN FEB.)
BIT 4 = 1 (29 DAYS IN FEB.)

FIGURE 4 PROGRAMING CHART LEAP YEAR

Bit 4 of the Days Tens register controls whether or not there are 28 or 29 days in Febuary. If bit 4 is on then it is leap year and Febuary will have 29 days. If bit 4 is off then Febuary has 28 days. If it is leap year and Febuary 29th has passed the clock chip will automaticaly reset bit 4. The programming chart in FIGURE 5 has been updated with the information you have learned so far.

DATA DESCRIPTION	SECONDS UNITS	SECONDS TENS	MINUTES UNITS	MINUTES TENS	HOUR UNITS	HOUR TENS	DAY OF WEEK	DAY UNITS	DAY TENS	MONTH UNITS	MONTH TENS	YEAR UNITS	YEAR TENS
REG	0	1	2	3	4	5	6	7	8	9	10	11	12
+32	32	33	34	35	36	37	38	39	40	41	42	43	44
BITS 8	•		•		•	24		•		•		•	•
BITS 4	•	•	•	•	•	AM/PM	•	•	28/29	•		•	•
BITS 2	•	•	•	•	•	•	•	•	•	•		•	•
BITS 1	•	•	•	•	•	•	•	•	•	•	•	•	•

*ACTIVE BITS

FIGURE 5 PROGRAMMING CHART (updated)

Enter the following program.

```
010 C1 = 130 : C2 = 129
020 DIM T(7) : DIM W$(7)
030 DATA "SUNDAY","MONDAY","TUESDAY","WEDNESDAY","THURSDAY",
        "FRIDAY","SATURDAY"
040 FOR A = 0 TO 6 STEP 1
050 READ W$(A)
060 NEXT A
070 OUT C2,16
080 LET I=0
090 FOR D = 44 TO 38 STEP -1
100 OUT C1,D
110 T(I) = INP(C1)
120 I=I+1
130 NEXT D
140 IF T(4)>3 THEN T(4) = T(4)-4
150 PRINT "THE DATE IS ";T(2);T(3);"/";T(4);T(5);"/";T(0);T(1)
160 PRINT "TODAY IS "W$(T(6))
170 OUT C2,0 : END
```

Check the program to be sure all lines have been entered correctly and then RUN it. The display should look like this:

THE DATE IS (mm/dd/yy)
TODAY IS (day of week)

EXPLINATION

The basic difference between this routine and the previous time routines is the addition of the array W$ and the definition of the days of the week. Line 140 strips the 4 bit if it was set, from the days tens digit. If you dont strip this bit the digit that is displayed will be wrong.

An interesting exersise would be to combine the date and time routines into a single program. Try it. If you need help check the clock demo program in Appendix C.

SUMMARY

1. There are 13 clock registers.
2. All data stored in the clock is numeric.
3. Each digit of information has its own address.
4. The HOLD signal prevents counter rollover.
5. When using Assembler or Machine Languages 150 micrsecond delay is required after the HOLD signal is raised.
6. You must add a 32 to the register address when doing a read.
7. You must lower the HOLD signal after reading to enable the registers to continue counting.
8. You must examine the option bits when reading the hours tens digit.
9. You must strip the option bits when displaying the hours tens digit.
10. The day of the week is a value from 0 to 6.
11. The leap year bit is stored in the Day Tens 4 bit.
12. The leap year bit must be striped when displaying the days ten digit.
13. Only two I/O Port addresses are used. The Address Port and the Data Port.

CHAPTER 2

HOW TO WRITE A SET ROUTINE

In order to set or write data into the clock chip the following events must occur in the listed sequence.

1. Raise the HOLD signal by writing a 16 to the Data Port.
2. Write the register address to the Address Port.
3. Write the data plus 16 to the Data Port. The plus 16 is required to keep the HOLD signal active. For example, if you wanted to WRITE a 3, the value sent to the Data Output Port would be 3+16 or 19.
4. Write the address plus 16 to the Address Port. This will raise the write signal.
5. Write the register address to the Address Port. This will will lower the WRITE signal and complete the write operation.
6. Repeat steps 2, 3, 4, and 5 until all the desired registers are set.
7. Write a zero to the Data Port to lower the HOLD signal.

Enter and RUN the following program. It is an example of a simple set routine.

```
010 C1 = 130 : C2 = 129
020 OUT C2,16
030 INPUT "ENTER REG. ADDRESS 0-12";N
040 INPUT "ENTER DATA 0-9";T
050 OUT C1,N
060 OUT C2,T+16
070 OUT C1,N+16
080 OUT C1,N
090 OUT C1,N+32
100 D = INP(C1)
110 PRINT D
120 OUT C2,0 : END
```

This program will allow you to set a digit into any register and then it will read the data back and display it. Experiment with this program by entering different register addresses and data. Try loading the current time and date. Then use your read routine to read the data back to verify your settings.

NOTE- YOU CANNOT SET A VALUE INTO THE SECONDS REGESTERS. IF YOU WRITE DATA INTO THE SECONDS REGISTERS THEY WILL BE SET TO ZEROS.

Try writing a set routine similar to the clock demo program. If you have a problems refer to the listing in Appendix C. FIGURE 6 has been updated to reflect the new data necessary to set the registers.

DATA DESCRIPTION	SECONDS UNITS	SECONDS TENS	MINUTES UNITS	MINUTES TENS	HOUR UNITS	HOUR TENS	DAY OF WEEK	DAY UNITS	DAY TENS	MONTH UNITS	MONTH TENS	YEAR UNITS	YEAR TENS
REG	0	1	2	3	4	5	6	7	8	9	10	11	12
+32	32	33	34	35	36	37	38	39	40	41	42	43	44
+16	16	17	18	19	20	21	22	23	24	25	26	27	28
BITS 8	*		*		*	24		*		*		*	*
BITS 4	*	*	*	*	*	AM/PM	*	*	28/29	*		*	*
BITS 2	*	*	*	*	*	*	*	*	*	*		*	*
BITS 1	*	*	*	*	*	*	*	*	*	*	*	*	*

*ACTIVE BITS

FIGURE 6 PROGRAMMING CHART (updated)

ADJUST THE TIME

A feature of the ComputerWatch is the ability to adjust the time by +-30 seconds. If you find that the clock has gained or lost time you can adjust it with a +-30 second routine. This feature will advance the minutes units digit if the seconds are greater than 30 and reset the seconds to zero. If the seconds are less than 30 the minutes units digit will not be advanced and the seconds will be reset to zero.

To reset the time you must first raise the ADJUST signal. This signal must be active for 32 milliseconds. To complete the operation you must lower the ADJUST signal. Enter the following program.

```
010 C2 = 129
020 OUT C2,32
030 FOR A=1 TO 20 STEP 1
040 NEXT A
050 OUT C2,0 : END
```

EXPLINATION

Line 020 will write a 32 to the Data Port. This will activate the +-30 second ADJUST signal. Line 030 and 040 provide a time delay of 32 milliseconds and line 050 will write a zero to the Data Port. This will lower the ADJUST signal and complete the operation. Add this program to the ones you have completed and you will have a complete set of programs to operate the ComputerWatch.

INTERRUPTS

There are four interrupts provided by jumper options. The read signal and the address of the timers is detected external to the clock chip and enables the hard interrupt signals. These signals can then interrupt the system CPU if an option jumper is activated. The hard interrupts are only enabled during the time that the timer signals are being read from the clock chip, thus they can be controlled by software. The system CPU can input the timer signals when they are enabled at the clock chip. The four interrupts are:

1024 Hz (approx. 1 millisecond)
1 SECOND
1 MINUTE
1 HOUR

The 1024 Hz signal is a 50% duty cycle square wave and the 1 second, 1 minute, and 1 hour signals provide a low pulse for 122.1 microseconds at the indicated rate. The pulsed signals are normally high and go low momentarily (122.1us) when the indicated time interval has occured. The following jumpers must be installed if you want an interrupt at:

	PAD		$\overline{\text{IRQ}}$		$\overline{\text{NMI}}$
1024 Hz	D	TO	E	OR	F
1 SECOND	C	TO	E	OR	F
1 MINUTE	B	TO	E	OR	F
1 HOUR	A	TO	E	OR	F

To enable the timers you write a 47 to the Address Port. To disable the timers write a 0 to the Address Port.

NOTE- ANY READ OPERATION WILL DISABLE THE TIMERS. YOU MUST ENABLE THE TIMERS AFTER EACH READ OPERATION BY WRITTING A 47 TO THE ADDRESS PORT.

SUMMARY

1. You have to raise the HOLD signal and write the address of the register to be written in the Data Port.
2. You have to write the data +16 to the Data Port. The 16 keeps the HOLD signal active.
3. You have to write the address +16 to the Address Port to raise the WRITE signal. Then write the address again to lower the WRITE signal.
4. You should read the data back for verification.
5. You should lower the HOLD signal by writing a zero to the Data Port.
6. You adjust the time +-30 seconds by writing a 32 to the Data Port. A time delay of 32 milliseconds is required before writing a zero to the Data Port to lower the ADJUST signal.
7. To enable the timers write a 47 to the Address Port.
8. To disable the timers write a zero to the Address Port.
9. If timmers are being used be sure to enable them after each read operation.

FUNCTIONAL DESCRIPTION

The ComputerWatch is interfaced to the bus by two TTL latching type output ports and one tristate input port. The ComputerWatch responds to two of the four I/O port addresses selested. The output port latches are selected by address bit zero or one. When the ComputerWatch is selected for access and an output cycle (OUT) is taking place with address bit zero active (high) the data port latches data from the system bus. When the ComputerWatch is selected and an output cycle (OUT) is performed with address bit one active (high) the address port latches data from the system. The data port interfaces to the 4 bit bidirectional data bus of the clock chip by means of a tristate buffer. This buffer is disabled whenever a read of the clock chip is taking place, to allow the clock chip to drive its data onto the bus. The data port latches also interface two control signals to the clock chip. The 30 second +- adjust signal and the hold signal are controlled by the two most significant bits of the data latches. The address latches provide a four bit address to the clock chip for selection of the desired clock register. The two most significant bits of the address latches control the signals read and write to the clock chip. The read signal is made active after presenting an address to the clock chip. The clock chip will then present the selected data from the clock on the 4 bit data bus. This data may then be read by the system on an input (INP) operation.

When data is to be written to the clock chip, it is first written to the Data Port latches and the address is written to the Address Port latches. The write signal is then raised and lowered to complete the write to the clock chip.

Before reading or writing the clock chip the hold signal should be raised. When the read or write operation is completed the hold signal should be lowered to allow the clock to continue counting.

The interval timers are output by the clock when a read is performed to address 15 (hex F). The read signal and the address of the timers is detected external to the clock chip and enables the hard interrupt signals. These signals can then interrupt the system CPU if an optional hard interrupt is enabled (jumper installed). The hard interrupts are only enabled during the time that the timer signals are being read from the clock chip and thus can be controled by software. The system CPU can input the timer signals when they are enabled at the clock chip. The 1024 Hz signal is a 50% duty cycle square wave and the 1 second, 1 minute, and 1 hour signals provide a low pulse for 122.1 microseconds at the indicated rate. The pulsed signals are normally high and go low momentarily (122.1us) when the indicated tme interval has occured.

The 30 second +- adjust signal must be held active (high) for greater than 31.25 milliseconds in order to be effective. When this signal is made active and then lowered the clock will be adjusted. The seconds digits are set to zero and the divider chain is reset so that the next toggle will occur 1 second after the 30 second +- adjust signal is returned to the inactive state. The minutes are incremented by one if the seconds were equal to 30 or more when the adjust signal is raised. If the seconds were less than 30 at the time the adjust signal was raised then the minutes will not be changed.

The clock chip automatically detects power off and switches to standby mode. The interface circuits of the clock chip are disabled in standby mode and the power drain is substantially reduced (30uA max.). The clock will continue keeping time while the on-board battery supplies power. The clock will still operate properly when the battery voltage drops as low as 2.2 volts. The nominal battery voltage is 3.6 volts. The battery is charging whenever power is applied to the system. The time to fully charge a discharged battery is approximately 12 hours. A fully charged battery will keep the clock running for over two months without charging. It will even keep time when removed from the system.

The clock chip contains registers for the time of day in hours, minutes, and seconds. The date is stored as day of week, day of month, month, and year. The clock chip registers are organized as 13 four bit registers. Option bits are contained in the hours ten register. The two most significant bits of this regster are used to select 12 or 24 hour format and indicate AM or PM in the 12 hour format. An optional bit is contained in the day of month ten register for indicating when 29 days are needed in Febuary (leap year).

DISCRIPTION OF OPERATION

The ComputerWatch requires that events occur in a proper sequence and that proper timing is allowed for certain functions to occur. Following is a sequence of events and timing allowances that should be maintained when reading or setting the clock.

READ TIME AND DATE

1. Raise hold signal by writing 16 to the Data Port.
2. Wait 150 microseconds (not required for BASIC).
3. Activate read operation by writing to the Address Port the address of the desired register plus 32.
4. Wait 6 microseconds (not required for BASIC).
5. Input digit data. Note that certain registers have option bits encoded as part of the data. These bits must be examined for active options and masked from the data prior to using it for display purposes.
6. Repeat steps 3, 4, and 5 until all desired clock registers are read.
7. Write zero to the Data Port to lower the hold signal.
8. Write 47 to the Address Port to enable the interval timers.

SET TIME AND DATE

1. Raise hold signal by writing 16 to the Data Port.
2. Wait 150 microseconds (not required for BASIC).
3. Write address to the Address Port.
4. Write data plus 16 to the Data Port. The 16 is required to keep the hold signal active.
5. Write address plus 16 to the Address Port. This will raise the write signal.
6. Write the address to the Address Port. This will lower the write signal and complete the write operation.
7. Repeat steps 3, 4, 5, and 6 until all the desired clock register are set.
8. Write zero to the Data Port to lower the hold signal.
9. Write 47 to the Address Port to enable the interval

ADJUST TIME +-30 SECONDS

1. Write 32 to the Data Port. This will activate the +-30 second adjust signal.
2. Wait 32 milliseconds.
3. Write zero to the Data Port. This will lower the +-30 second adjust signal and complete the operation.

APPENDIX A

I/O PORTS

```
           ADDRESSS                                    DATA
             PORT                                      PORT

                       |BIT |                                  |BIT |
  READ SIGNAL>  A5|32  |           +-30 SECOND ADJUST> A5|32  |
 WRITE SIGNAL>  A4|16  |                  HOLD SIGNAL> A4|16  |
    DATA OUT }  A3| 8  |                             } A3| 8  |
       AND   }  A2| 4  |                  DATA IN    } A2| 4  |
    REG ADDR }  A1| 2  |                             } A1| 2  |
             }  A0| 1  |                             } A0| 1  |
             |     ------                                 ------
      _______|
  |
```

When A0 through A3 are active plus A5 the READ signal (decimal 47) the interrupts are enabled.

ADDRESS PORT

The Address Port is used for reading clock data. You must first write (OUT) the clock register address with the read signal active to the Address Port. The data will then be transfered from the selected clock register to the Address Port. To read the data from the Address Port to the system you would read (INP) the Address Port.

The Address Port also controls the write signal when data is being set into the clock by the Data Port.

The Address Port is also used for interrupt control. Interrupts are enabled when lines A0 through A3 are active and a read signal is present. You would write (OUT) a decimal 47 to the Adderss Port. Interrupt signals would then be placed at pads A, B, C, and D. They would remaine active until the next time a read was performed, then the interrupts would become disabled.

DATA PORT

To set data into the clock you must first write (OUT) the address of the clock register selected to the Address Port. You then write (OUT) the data to be set into the selected clock register to the Data Port. The next step would be to raise the write signal by writing (OUT) the register address plus the write signal to the Address Port. Next you would write the selected register address without the write signal to lower the write signal.

The Data Port also controls the hold signal and the +- 30 second adjust signal.

APPENDIX B

I/O PORT ADDRESS SELECT

DECIMAL ADDRESS RANGE	DATA PORT(C2) ADDR	ADDR PORT(C1) ADDR	SW1 A7 1-12	SW2 A6 2-11	SW3 A5 3-10	SW4 A4 4-9	SW5 A3 5-8	SW6 A2 6-7
0-3	1	2	X	X	X	X	X	X
4-7	5	6	X	X	X	X	X	
8-11	9	10	X	X	X	X		X
12-15	11	12	X	X	X	X		
16-19	17	18	X	X	X		X	X
20-23	21	22	X	X			X	
24-27	25	26	X	X	X			X
28-31	29	30	X	X	X			
32-35	33	34	X	X		X	X	X
36-39	37	38	X	X		X	X	
40-43	41	42	X	X		X		X
44-47	45	46	X	X		X		
48-51	49	50	X	X			X	X
52-55	53	54	X	X			X	
56-59	57	58	X	X				X
60-63	61	62	X	X				
64-67	65	66	X		X	X	X	X
68-71	69	70	X		X	X	X	
72-75	73	74	X		X	X		X
76-79	77	78	X		X	X		
80-83	81	82	X		X		X	X
84-87	85	86	X		X		X	
88-91	89	90	X		X			X
92-95	93	94	X		X			
96-99	97	98	X			X	X	X
100-103	101	102	X			X	X	
104-107	105	106	X			X		X
108-111	109	110	X			X		
112-115	113	114	X				X	X
116-119	117	118	X				X	
120-123	121	122	X					X
124-127	125	126	X					
128-131	129	130		X	X	X	X	X
132-135	133	134		X	X	X	X	
136-139	137	138		X	X	X		X
140-143	141	142		X	X	X		
144-147	145	146		X	X		X	X
148-151	149	150		X	X		X	
152-155	153	154		X	X			X
156-159	157	158		X	X			
160-163	161	162		X		X	X	X
164-167	165	166		X		X	X	
168-171	169	170		X		X		X
172-175	173	174		X		X		
176-179	177	178		X			X	X
180-183	181	182		X			X	

I/O PORT ADDRESS SELECT (CONT)

DECIMAL ADDRESS RANGE	DATA PORT(C2) ADDR	ADDR PORT(C1) ADDR	SW1 A7 1-12	SW2 A6 2-11	SW3 A5 3-10	SW4 A4 4-9	SW5 A3 5-8	SW6 A2 6-7
184 187	185	186		X				X
188-191	189	190		X				
192-195	193	194			X	X	X	X
196-199	197	198			X	X	X	
200-203	201	202			X	X		X
204-207	205	206			X	X		
208-211	209	210			X		X	X
212-215	213	214			X		X	
216-219	217	218			X			X
220-223	221	222			X			
224-227	225	226				X	X	X
228-231	229	230				X	X	
232-235	233	234				X		X
236-239	237	238				X		
240-243	241	242					X	X
244-247	245	246					X	
248-251	249	250						X
252-255	253	254						

X = Jumper installed or switch on.

APPENDIX C

CLOCK/CALENDAR DEMO PROGRAM

PROGRAM WRITTEN IN BASIC-E

```
    REM **************************************************
    REM *THIS PROGRAM WILL READ AND DISPLAY TIME/DATE,   *
    REM *RESET TIME/DATE, AND ADJUST TIME +-30 SECONDS. *
    REM **************************************************
    REM
    C1= 130 : C2= 129
    DIM T(13)
    DIM W$(7)
    DIM S$(13)
    DATA "SUNDAY","MONDAY","TUESDAY","WEDNESDAY","THURSDAY",\
         "FRIDAY","SATURDAY"
    DATA "YEAR TENS","YEAR UNITS","MONTH TENS","MONTH UNITS",\
         "DAY TENS"
    DATA "DAY UNITS","DAY OF WEEK 0 = SUN, 1 = MON, ECT"
    DATA "HOURS TENS","HOUR UNITS","MINUTE TENS",MINUTE UNITS"
    REM *********************
    REM * LOAD ARRAY W$     *
    REM *********************
    FOR A = 0 TO 6 STEP 1
    READ W$(A)
    NEXT A
    REM *********************
    REM * LOAD ARRAY S$     *
    REM *********************
    FOR A = 12 TO 2 STEP -1
    READ S$(A)
    NEXT A
310 PRINT "TYPE T TO DISPLAY DATE/TIME"
    PRINT "TYPE R TO RESET DATE/TIME"
    PRINT "TYPE A TO ADJUST THE TIME +-30 SECONDS"
    INPUT "TYPE E TO END PROGRAM";I$ : PRINT
    REM **************************
    REM * SELECT CORRECT ROUTINE *
    REM **************************
    IF I$ = "T" THEN 500
    IF I$ = "R" THEN 790
    IF I$ = "A" THEN 990
    IF I$ = "E" THEN STOP
    GOTO 310
    REM **************************
    REM * TIME/DATE DISPLAY      *
    REM **************************
500 OUT C2,16 : LET I = 0
    FOR D = 44 TO 32 STEP -1
    OUT C1,D : T(I) = INP(C1) : I = I+1
    NEXT D
```

```
    REM *****************************
    REM * CHECK FOR 12/24 & AM/PM *
    REM *****************************
    IF T(4)>3 THEN T(4) = T(4)-4
    B = T(7)
    IF B>7 THEN P$ = " "
    IF T(7)>7 THEN T(7) = T(7)-8
    IF B<8 AND T(7)>3 THEN P$ = "PM"
    IF B<8 AND T(7)<4 THEN P$ = "AM"
    IF T(7)>3 THEN T(7) = T(7)-4
    REM ****************************
    REM * PRINT ROUTINE            *
    REM ****************************
    PRINT "THE DATE IS ";T(2);T(3);"/";T(4);T(5);"/";T(0);T(1)
    PRINT "TODAY IS ";W$(T(6))
    PRINT "THE TIME IS ";T(7);T(8);":";T(9);T(10);":";T(11);T(12);P$
    PRINT : OUT C2,0
    GOTO 310
    REM *****************************
    REM *TIME/DATE SET ROUTINE      *
    REM *****************************
790 OUT C2,16 :
    FOR D = 12 TO 2 STEP -1
    PRINT S$(D) : INPUT N : T(D) = N
    NEXT D
830 INPUT "IS THIS LEAP YEAR";Y$
    IF Y$ = "Y" THEN T(8) = T(8)+4
    IF Y$ = "Y" THEN 880
    IF Y$ = "N" THEN 880
    GOTO 830
880 INPUT "12 OR 24 HOUR FORMAT";Y
    IF Y = 24 THEN T(5) = T(5)+8
    IF Y = 12 THEN 930
    IF Y = 24 THEN 980
    GOTO 880
930 INPUT "AM OR PM";X$
    IF X$ = "PM" THEN T(5) = T(5)+4
    IF X$ = "PM" THEN 980
    IF X$ = "AM" THEN 980
    GOTO 930
980 FOR D = 12 TO 2 STEP -1
    N = T(D) : GOSUB 985
    NEXT D
    OUT C2,0 : PRINT : GOTO 500
985 OUT C1,D : OUT C2,N+16 : OUT C1,D+16 : OUT C1,N : RETURN
    REM ***************************
    REM * ADJUST ROUTINE          *
    REM ***************************
990 OUT C2,32
    FOR A = 1 TO 20 STEP 1
    NEXT A
    OUT C2,0
    GOTO 500
    END
```

APPENDIX D

```
        TITLE   'TIME for the CompuTime ComputerWatch'

BASE    EQU     128         ;BASE OF CLOCK BOARD
PORT    EQU     BASE+2
BDOS    EQU     5           ;4205 FOR TRS-80, ETC.
BOOT    EQU     0           ;4200 FOR TRS-80, ETC.

        ORG     0100H
        LXI     SP,STACK

        MVI     A,6         ;DAY OF WEEK
        CALL    READ
        RLC                 ;*2 FOR TBL OFFSET
        LXI     H,DTBL      ;DAYTABLE
        MOV     E,A
        MVI     D,0
        DAD     D
        MOV     E,M
        INX     H
        MOV     D,M
        CALL    TEXTO

        CALL    CS          ;OUTPUT ", "

        MVI     A,9         ;GET MONTH UNITS DIGIT
        CALL    READ
        MOV     B,A         ;SAVE IN B
        MVI     A,10        ;GET MONTH TENS DIGIT
        CALL    READ
        MOV     A,B         ;GET THE UNITS BACK (DON'T SET FLAGS)
        JZ      SKIP        ;WAS 1-9 (JANUARY-SEPTEMBER)
        ADI     10          ;PLUS 10 IF (OCTOBER-DECEMBER)
SKIP    DCR     A           ;MAKE 0-11
        RLC                 ;*2 FOR TBL OFFSET
        LXI     H,MTBL      ;MONTHTABLE
        MOV     E,A
        MVI     D,0
        DAD     D
        MOV     E,M
        INX     H
        MOV     D,M
        CALL    TEXTO

        MVI     E,' '       ;PRINT A SPACE
        CALL    BYTEO       ;OUTPUT BYTE

        MVI     A,8         ;GET DAY TENS DIGIT
        CALL    READ
        ANI     3           ;STRIP LEAP YEAR BIT
        CNZ     ODGT        ;OUTPUT THE DIGIT (IF IT IS NON ZERO)

        MVI     A,7         ;GET DAY UNITS DIGIT
        CALL    READ
```

```
        PUSH    PSW
        CALL    ODGT    ;OUTPUT THE DIGIT
        POP     PSW

        CPI     1
        JZ      STER
        CPI     2
        JZ      NDER
        CPI     3
        JZ      RDER
THER    LXI     D,TH
        CALL    TEXTO
        JMP     PAST
STER    LXI     D,ST
        CALL    TEXTO
        JMP     PAST
NDER    LXI     D,ND
        CALL    TEXTO
        JMP     PAST
RDER    LXI     D,RD
        CALL    TEXTO

PAST    CALL    CS      ;OUTPUT ", "

        LXI     D,CENT  ;OUTPUT CENTURY (19TH)
        CALL    TEXTO

        MVI     A,12    ;YEAR TENS
        CALL    RDOD    ;READ AND OUTPUT DIGIT

        MVI     A,11    ;YEAR UNITS
        CALL    RDOD    ;READ AND OUTPUT DIGIT

        CALL    CS      ;OUTPUT ", "

        MVI     A,5     ;HOUR TENS
        CALL    READ
        PUSH    PSW
        ANI     3       ;STRIP 12/24, AM/PM BITS
        CALL    ODGT

        MVI     A,4     ;HOUR UNITS
        CALL    RDOD    ;READ AND OUTPUT DIGIT

        MVI     E,':'   ;OBVIOUSLY
        CALL    BYTEO

        MVI     A,3     ;MINUTE TENS
        CALL    RDOD    ;READ AND OUTPUT DIGIT

        MVI     A,2     ;MINUTE UNITS
        CALL    RDOD    ;READ AND OUTPUT DIGIT

        MVI     E,':'   ;OBVIOUSLY
        CALL    BYTEO
```

```
        MVI     A,1     ;SECONDS TENS
        CALL    RDOD    ;READ AND OUTPUT DIGIT

        MVI     A,0     ;SECONDS UNITS
        CALL    RDOD    ;READ AND OUTPUT DIGIT

        POP     PSW
        MOV     B,A     ;SAVE TMP
        ANI     8       ;24 HOUR MODE?
        JNZ     EXIT    ;DONE, FINISH

        MOV     A,B     ;RESTORE
        ANI     4       ;AM OR PM?
        JZ      AM      ;AM

PM      LXI     D,PMTXT
        JMP     FOO

AM      LXI     D,AMTXT
FOO     CALL    TEXTO

EXIT    JMP     BOOT

RDOD    CALL    READ    ;READ AND OUTPUT DIGIT

ODGT    ORI     30H     ;CONVERT TO ASCII
        MOV     E,A
        CALL    BYTEO
        RET

READ    ORI     20H     ;ADD IN OFFSET
        OUT     PORT    ;WE WANT THIS DIGIT!
        NOP ! NOP       ;A SHORT DELAY
        IN      PORT    ;READ THE DIGIT
        ORA     A       ;SET THE FLAGS
        RET

CS      LXI     D,CSTXT ;POINT TO ", "

TEXTO   MVI     C,9     ;STRING OUTPUT, ENDS WITH A '$'
        CALL    BDOS
        RET

BYTEO   MVI     C,2     ;BYTE OUTPUT, FROM REG. E
        CALL    BDOS
        RET

MTBL    DW      JAN
        DW      FEB
        DW      MAR
        DW      APR
        DW      MAY
        DW      JUN
        DW      JUL
```

```
        DW      AUG
        DW      SEP
        DW      OCT
        DW      NOV
        DW      DEC
JAN     DB      'January$'
FEB     DB      'February$'
MAR     DB      'March$'
APR     DB      'April$'
MAY     DB      'May$'
JUN     DB      'June$'
JUL     DB      'July$'
AUG     DB      'August$'
SEP     DB      'September$'
OCT     DB      'October$'
NOV     DB      'November$'
DEC     DB      'December$'

DTBL    DW      SUN
        DW      MON
        DW      TUE
        DW      WED
        DW      THU
        DW      FRI
        DW      SAT
SUN     DB      'Sunday$'
MON     DB      'Monday$'
TUE     DB      'Tuesday$'
WED     DB      'Wednesday$'
THU     DB      'Thursday$'
FRI     DB      'Friday$'
SAT     DB      'Saturday$'

ST      DB      'st$'
ND      DB      'nd$'
RD      DB      'rd$'
TH      DB      'th$'

CSTXT   DB      ', $'
CENT    DB      '19$'

AMTXT   DB      ' AM$'
PMTXT   DB      ' PM$'

        DS      16
STACK   EQU     $
        END
```

COMPUTER SYSTEMS INC.

"QUICK & TIMELY"

ERRATA SHEET

#1 FOR

S-100 CLOCK/CALENDAR BOARD
PART # QTC-CCS-A

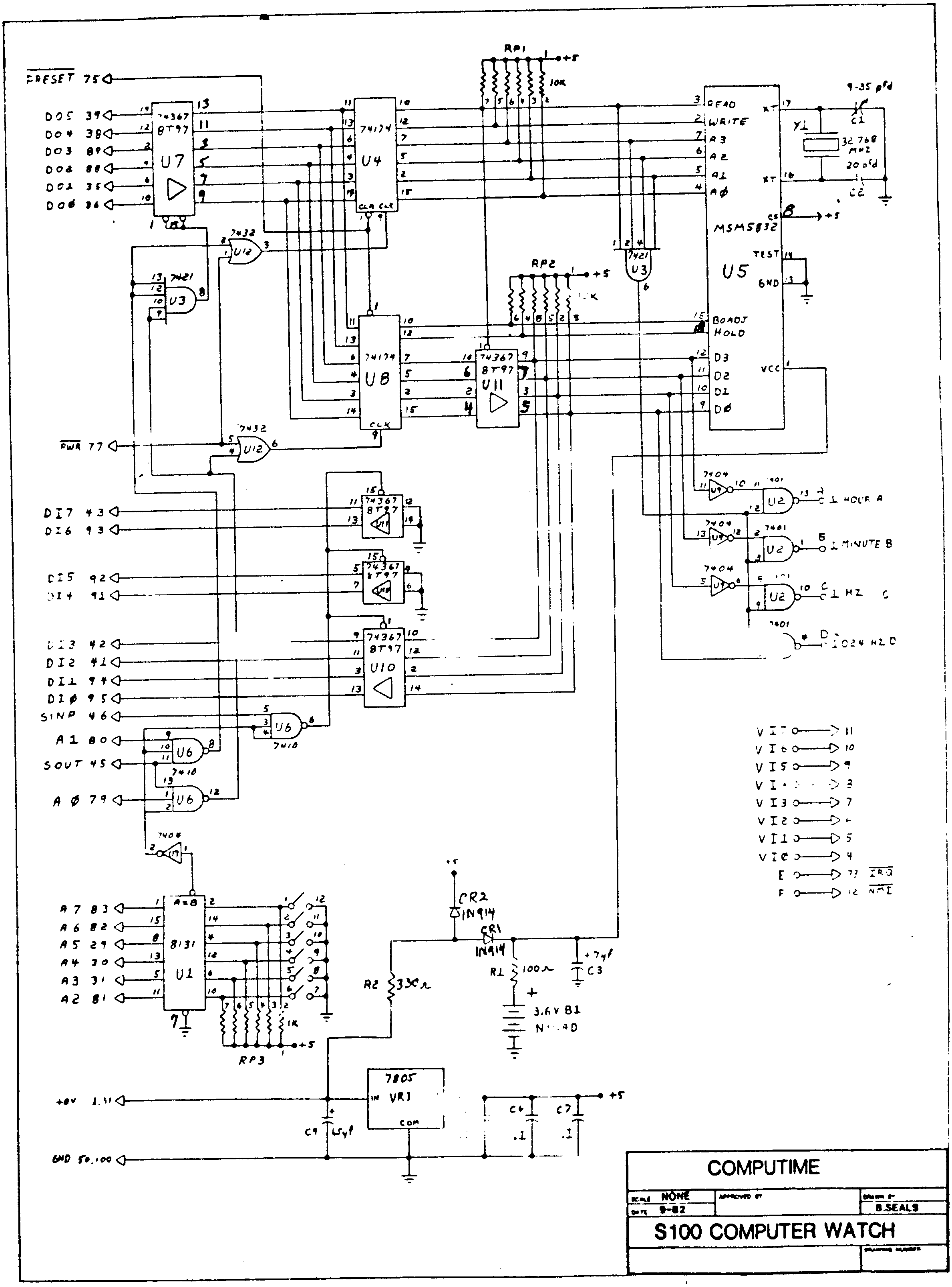
1. In order for the Clock/Calendar board to function in an 8080 system, it must have an integrated circuit (U12) on the board. Use either 74LS32 or 7432.

2. The SIP Resistor Networks are marked with a dot to indicate how they should be installed. The end with the dot should be on the left.

3. The Tantalum Capacitor can be anywhere from 1.5 MFD to 3.3 MFD.

4. The 5832 Clock Chip should be installed with care due to static electricity build-up. Try not to touch the pins on this IC.

3330 South 300 West, Salt Lake City, Utah . (801)487-8825
Toll Free 1-800-238-3100



PRESET 75
DO5 39
DO4 38
DO3 89
DO2 88
DO1 35
DO∅ 36
74367 8T97 U7
74174 U4
7432 U12
7421 U3
74174 U8
74367 8T97 U11
RP1
10K
RP2
MSM5832
U5
READ
WRITE
A3
A2
A1
A∅
BOADJ
HOLD
D3
D2
D1
D∅
VCC
TEST
GND
Y1
32 768 KHZ
C1
C2
9-35 pfd
20 pfd
PWR 77
DI7 43
DI6 93
DI5 92
DI4 91
DI3 42
DI2 41
DI1 94
DI∅ 95
SINP 46
A1 80
SOUT 45
A∅ 79
74367 8T97 U10
7410 U6
7404 U9
7401 U2
1 HOUR A
1 MINUTE B
1 HZ C
1024 HZ D
IRQ
NMI
A7 83
A6 82
A5 29
A4 30
A3 31
A2 81
8131 U1
RP3
1K
CR2 1N914
CR1 1N914
R1 100Ω
R2 330Ω
C3
3.6V B1
NICAD
7805 VR1
+8V 1,51
GND 50,100
C4
C5
C6
C7
.1
+5
COMPUTIME
SCALE NONE
DATE 9-82
DRAWN BY B.SEALS
S100 COMPUTER WATCH