The MM58174A Real Time Clock in a Battery Backed-Up Design Provides Reliable Clock and Calendar Functions

National Semiconductor
Application Note 359
Steve Munich
April 1991

The MM58174A Real Time Clock in a Battery Backed-Up Design Provides Reliable Clock and Calendar Functions

INTRODUCTION

National Semiconductor's MM58174A microprocessor real time clock is a reliable and economical solution to adding clock and calendar timekeeping to any system. This metal-gate CMOS circuit *(Figure 1)* will operate with a supply voltage as low as 2.2V, allowing easy implementation of battery back-up circuitry to maintain timekeeping year after year, even when the system's main supply fails. The MM58174A has counters for months, day of month, day of week, hours, minutes, seconds and tenths of seconds, as well as a register for automatic leap year calculations. Also included are periodic and single interrupt capabilities at 0.5, 5 and 60 second intervals.

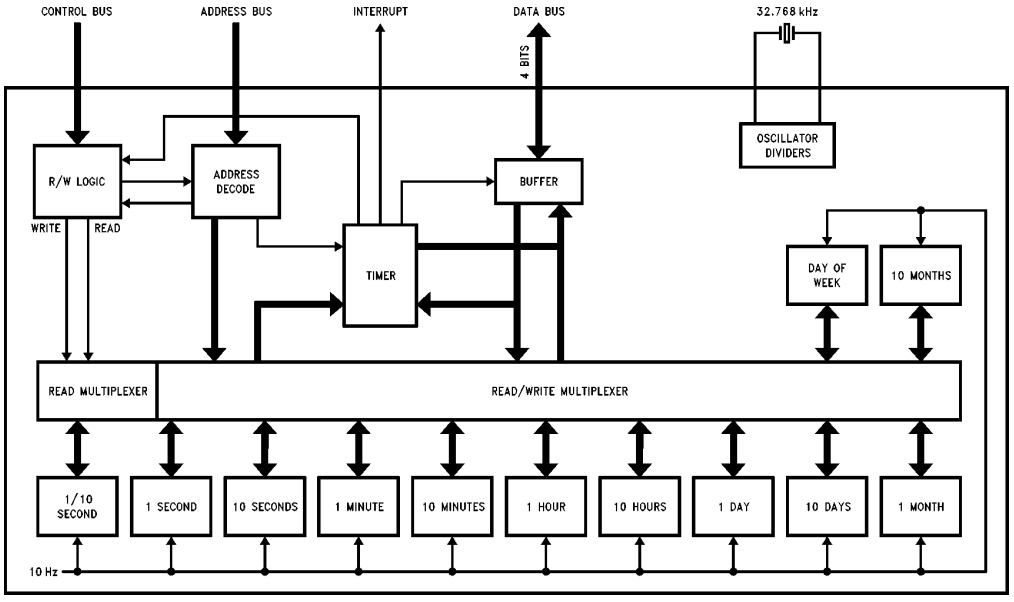
This application note will describe how to interface the MM58174A to microprocessors with battery backed-up circuitry. Included will be a functional circuit description, trouble-shooting hints, crystal oscillator adjustment and supplier information. Please refer to the data sheet for AC and DC electrical specifications and timing diagrams.

DESCRIPTION OF FEATURES

Reading and Writing the Time

The MM58174A has BCD counters for tenths of seconds through months, which are accessed by a 4-bit address as shown in Table I. Months through minutes registers can be read and written to. Tens of seconds, units of seconds and tenths of seconds registers can only be read and are reset to zero when counting is enabled by the start/stop flip-flop. When properly addressed, a nibble of data appears on the data pins DB0–DB3 when a read occurs, and data is accepted on these pins during a write. Any unused data pins will be ignored during a write operation (e.g., days of week uses only DB2 through DB0). To insure proper counter incrementation and accessing, all timing specifications must be observed. It is particularly important that the $\overline{RD}$ strobe width be less than 15 μs for the highest timekeeping accuracy, but never greater than 15 ms.

Address 13 is a write only leap year status register. Writing a ''1'' to DB3 at this address will cause the time 02/28 23:59 59.9 to roll over to 02/29 00:00 00.0 in one-tenth of a second. If a ''1'' is instead written to any other data bit, the rollover will go to 03/01 00:00 00.0 and the leap year will occur as shown in Table II.



TL/F/6169–1

FIGURE 1. Block Diagram

NSC800™ is a trademark of National Semiconductor Corporation.
Z80® is a registered trademark of Zilog Corporation.

RRD-B30M105/Printed in U. S. A.

TABLE I. Address Decoding for Internal Registers

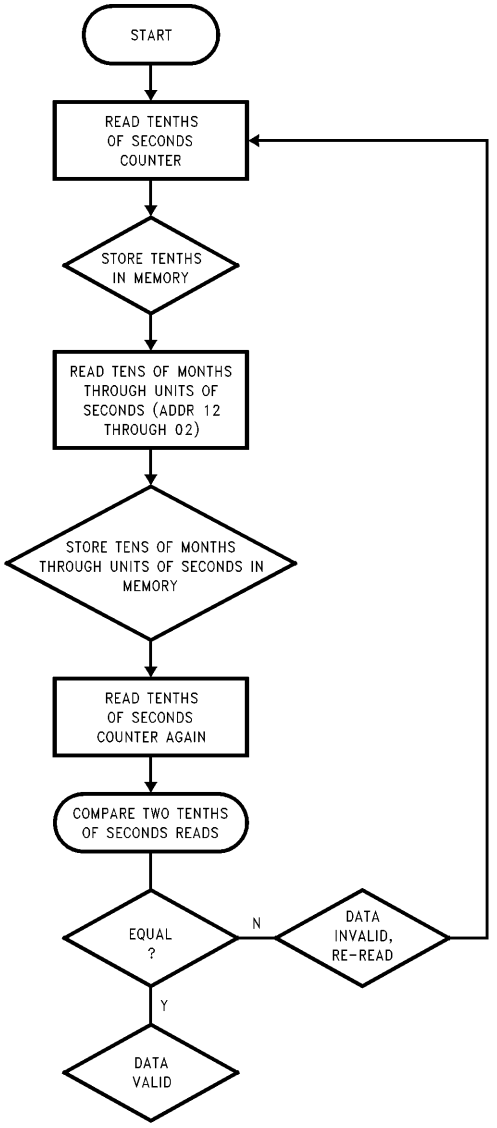
Selected Counter	Address Bits AD3	AD2	AD1	AD0	Mode
Test Only	0	0	0	0	Write Only
1 Tenths of Seconds	0	0	0	1	Read Only
2 Units of Seconds	0	0	1	0	Read Only
3 Tens of Seconds	0	0	1	1	Read Only
4 Units of Minutes	0	1	0	0	Read or Write
5 Tens of Minutes	0	1	0	1	Read or Write
6 Units of Hours	0	1	1	0	Read or Write
7 Tens of Hours	0	1	1	1	Read or Write
8 Units of Days	1	0	0	0	Read or Write
9 Tens of Days	1	0	0	1	Read or Write
10 Day of Week	1	0	1	0	Read or Write
11 Units of Months	1	0	1	1	Read or Write
12 Tens of Months	1	1	0	0	Read or Write
13 Years	1	1	0	1	Write Only
14 Stop/Start	1	1	1	0	Write Only
15 Interrupt	1	1	1	1	Read or Write

TABLE II. Years Status Register

Mode: Address 13, Write Mode				
	DB3	DB2	DB1	DB0
Leap Year	1	0	0	0
Leap Year-1	0	1	0	0
Leap Year-2	0	0	1	0
Leap Year-3	0	0	0	1

Detecting Changed Data

It is possible that during a sequential read of months through tenths of seconds a roll-over may occur. If the time at the start of the read is 23:59 59.5 and it rolls over to the time 00:00 00.0, the microprocessor could read back 23:50 00.00 or 23:00 00.0, etc. Wrong data could also be stored in the clock if the clock is running and is updated during a write (the start/stop flip-flop discussed in the next paragraph will help avoid invalid writes). The MM58174A has a data-changed flip-flop which indicates that a tenths of seconds roll-over has occurred. This flip-flop sets all the data lines high each time the tenths of seconds counter is updated. The ''F'' on the data lines is then cleared by the next low-to-high transition of any read strobe. In a sequential read of the counters, the tenths of seconds counter may change while the read strobe is low, but an ''F'' may never be seen before the read strobe comes high. Thus, the ''F'' may not be detected, although the experimental probability of this occurrence is approximately one in ten thousand reads, in the worst-case. It is essential to restart the whole sequence of reads, beginning from the tens of months register whenever an ''F'' is encountered on the data lines. A better procedure, outlined in the flowchart of *Figure 2*, would be to always begin each sequence of reads with the tenths of seconds register and end with this register. If comparing the two values read from this register shows them to be equal, the read is valid. If the compare yields two different values, repeat the same sequence of reads until the same value is read from the tenths of seconds register at the beginning and end of the sequence. It is advisable to use a machine code clock reading routine, or else the time to execute machine-interpreted code may be longer than one-tenth of a second, invalidating all sequential reads.



TL/F/6169–2

FIGURE 2. Flowchart to Detect Changed Data

Clock Accuracy

Two important factors affect the accuracy of the MM58174A. Any internal counter can jitter by −30.5 μs, meaning that the true count can be late by this amount. Also, whenever the clock is restarted (see next section), instead of holding a ''0'' in the tenths of seconds position for one-tenth of a second, the clock immediately jumps to a ''1''. So each time the clock is restarted, one-tenth of a second is lost. Accuracy would be maintained if the clock is restarted 0.1 second after the time reference's minutes change.

Starting and Stopping the Clock

Table I shows that address 14 accesses the start/stop flip-flop. A ''1'' on DB0 will start the clock. Writing a ''0'' to DB0 will stop it. This flip-flop is used for precise starting and stopping of the clock. It also prevents writing invalid data during a clock roll-over, as mentioned in the last paragraph. Before any sequence of writes, stop the clock. Restart it after the last write is completed.

Interrupts

The interrupt counter is controlled internally by three sequential flip-flops. By sending a sequence of read strobes to these flip-flops, the interrupt counter can be cleared or enabled. Initialization is necessary at power-up because these flip-flops can come up in any state. It is also necessary to re-initialize if an interrupt is not serviced within 16.6 ms. To initialize interrupts on the MM58174A, first write a ''0'' on the data lines at address 15, then read that address three times. The first read will clear any interrupts set. The second read insures that the counter is reset and the third read enables interrupts. Be sure to disable the microprocessor from accepting interrupts before initialization, because the act of writing ''0'' to address 15 may cause an interrupt.

Table III indicates which values turn on the 0.5, 5 and 60 second periodic or single interrupts. These values are written at address 15, as shown in Table I. To set a particular interrupt, a write need only occur once. Whenever an interrupt occurs, the signal at pin 13 falls from high to low.

TABLE III. Interrupt Selection and Status

Mode: Address 15, Read or Write Mode				
Function	**DB3**	**DB2**	**DB1**	**DB0**
No Interrupt	X	0	0	0
Int. at 60 Sec. Interval†	*	1	0	0
Int. at 5.0 Sec. Interval†	*	0	1	0
Int. at 0.5 Sec. Interval†	*	0	0	1

*0 for single interrupt (write), 1 for periodic interrupt (write).
X = Don't care (read)
†Add 16.6 ms to each time interval

To service the interrupt, read address 15 three times. This causes the interrupt output on pin 13 to return high and restarts the interrupt timer if periodic interrupts have been selected. The interrupt register may be read to see which interrupts have been set, but the MM58174A has no status bit indicating that the clock has sent out an interrupt. A version prior to the MM58174A had interrupt acknowledgement capability (the MM58174), so be sure to match data sheets with the correct parts. One final note about interrupts: they are not intended to be generated when the chip is in the sleep mode (see next paragraph). The MM58174A must be running with at least a 4V supply for interrupts to function.

Powering Down and Up

When the supply to pin 16 falls below 5V, timing becomes much more critical because propagation delays increase with a lowering of the power supply voltage. Note that the data sheet has timing specifications for 5V, and although the part is fully operational down to 4V, your design may not tolerate it. When the supply falls below 4V but stays above 2.2V, the MM58174A is in the sleep mode and only microamps are drawn from the battery. In this mode, the chip is not accessible by reading or writing, but time is being maintained.

On power-up from zero volts V_{CC}, one must make sure the chip is not in the test mode. This is done by writing a ''0'' to DB3 at address 0. It is advisable to do this even when coming out of the sleep mode. The test mode is mainly for production testing of the circuit.

There are several things to consider when designing the power-down circuitry. The basic functional requirements are to disable the chips before full power loss or malfunction, and to wait for V_{CC} to stabilize before enabling the chip on power-up. A desirable feature would be to allow the read or write in progress to complete. *Figures 3* and *4* include a typical power-down circuit which achieves these goals. In general, avoid using TTL since it is not rated below 4.5V. The power-down circuitry's signals to the MM58174A must not be allowed to deviate more than a diode drop above the clock's supply or below ground in order to avoid triggering SCR latch-up. Finally, be sure to use a PNP switch instead of a diode to disconnect the power supply from the battery. This will allow the MM58174A to see a V_{CC} closer to 5.0V coming from the main supply rather than 4.3V, enhancing timing requirements. See *Figures 3* and *4* and the next section for more information on design of power-down circuitry.

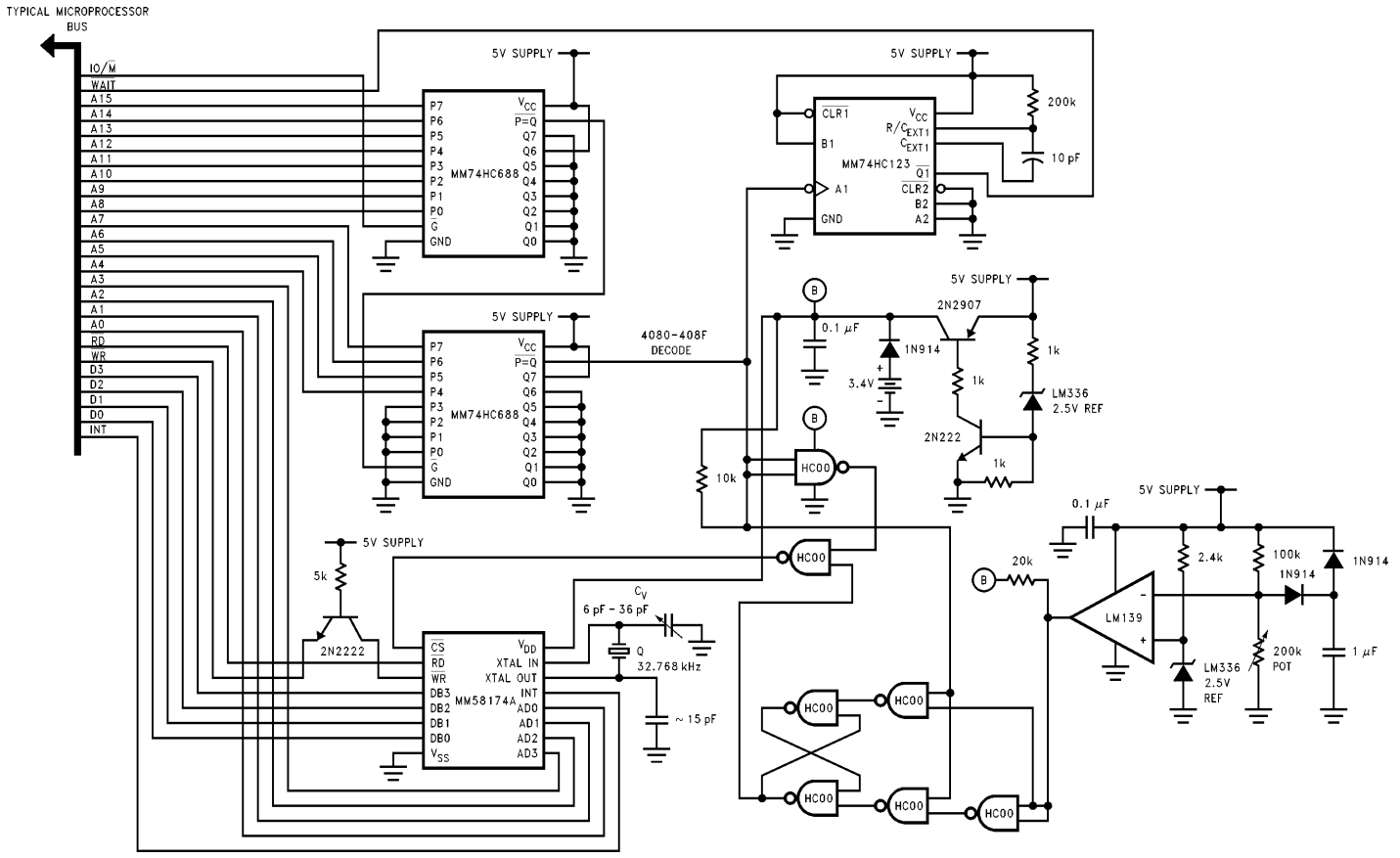
DESIGN IDEAS

Figures 3 and *4* show two possible ways of interfacing the MM58174A to a microprocessor; the former with wait stating and the latter eliminating wait states using the NSC810A RAM/IO timer as a peripheral interference adapter.

Real Time Clock Interface with Wait States

The design of *Figure 3* uses wait states to guarantee that the set-up and hold times of the MM58174A are satisfied. If one can afford to constrict his microprocessor throughput while accessing the MM58174A, this design has the advantages of simplified software and somewhat less expensive hardware. Decreased microprocessor throughput is usually not a consideration in most applications unless the clock is continuously being accessed for a real time display, while at the same time the processor is multiplexing the execution of other tasks.

The HC688s of *Figure 3* are used to fully decode the 4 bits of address space for the real time clock and to generate chip select and wait states. Each time an address between 4080H and 408FH appears on the address lines of the bus, the second of the cascaded HC688s generates a low strobe that allows the power-down circuitry to create a chip select, and also fires an HC123 one-shot configured to drive a 2 μs wait state onto the wait line of the microprocessor bus. For wait lines of the opposite polarity, the HC123's Q output could be used. A shift register may also be configured to give the proper access time delay.



Q = 32.768 kHz crystal (RCD, Saronix)

C_V = variable capacitor (Erie, Circuit Specialists)

FIGURE 3. Real Time Clock Interface with Wait States

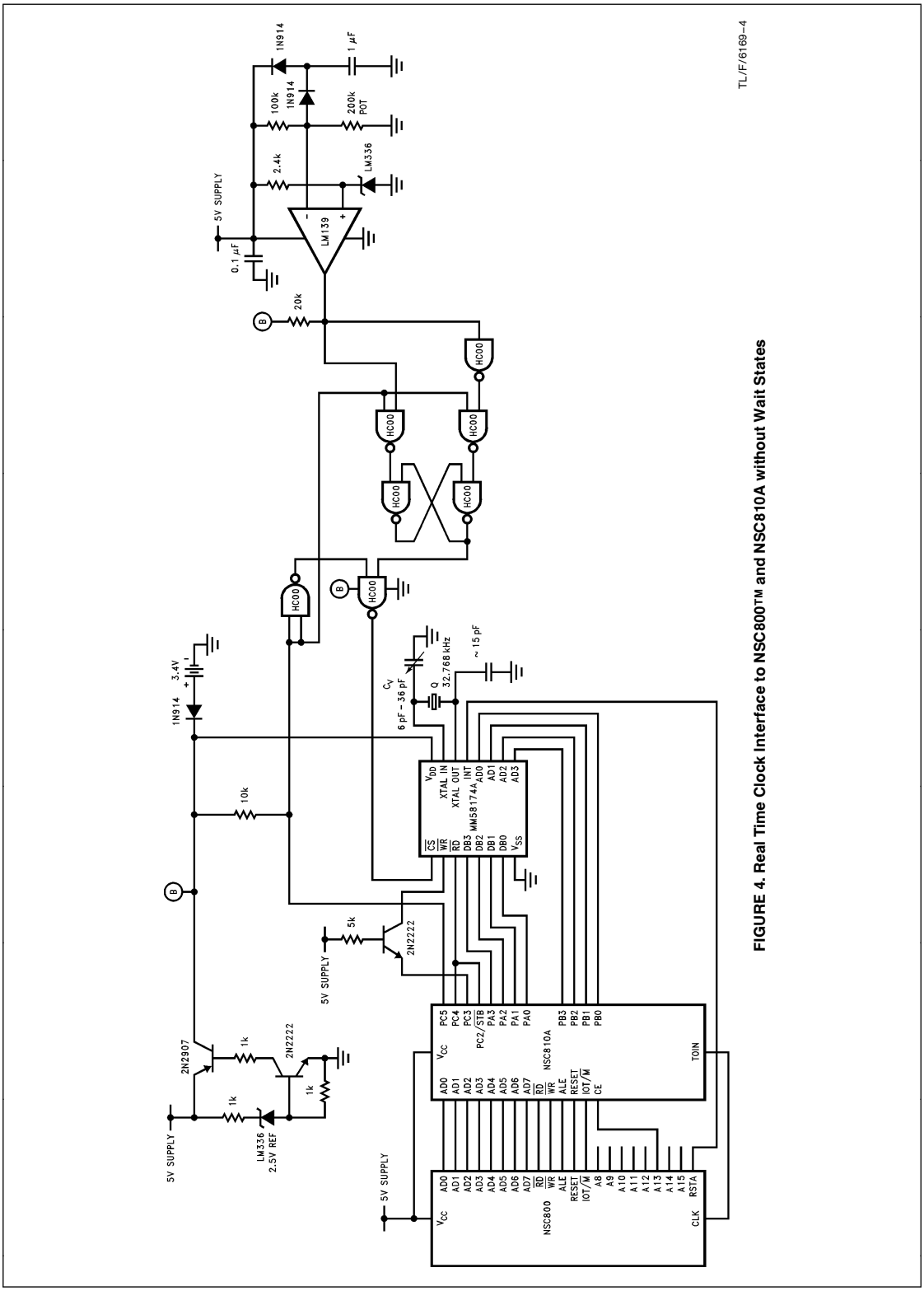


FIGURE 4. Real Time Clock Interface to NSC800™ and NSC810A without Wait States

Power is supplied to the parts from a 5.0V supply which is disconnectable by a PNP switch to a battery. When the main supply is on, the PNP in saturation brings the voltage at node B to about 4.8V. The diode near node B is back-biased to keep the battery from discharging and to protect it from damage by isolating it from 5V. If the main supply were to drift far enough downward, the diode would forward-bias, bringing node B to 0.7V below the battery voltage. Since the clock is now in the sleep mode, the only parts needed to be powered by the battery are the clock and the power-down circuitry. An NPN between the bus connection and the $\overline{WR}$ pin, as shown *Figure 3*, will reduce power consumption without inverting the signal into this pin. This is made necessary because both $\overline{CS}$ and $\overline{WR}$ pins on the MM58174A have internal pull-ups to V_{CC} which cause unnecessary current drain if either input were to become grounded. The NPN switch isolates the $\overline{WR}$ from ground, while the $\overline{CS}$ input is held high by the power-down circuitry.

Power-Down Circuitry Operation

The power-down circuitry of *Figures 3* and *4* consists of seven HC00 NAND gates and an LM139 low voltage comparator with an assortment of resistors, diodes and capacitors at the differential input.

With the 5.0V supply on, the assortment of diodes, resistors and capacitors at the comparator's differential input creates a low output. But when the supply is off, the battery pulls this output high through the 20 kΩ resistor. On power-up, after a short delay by the diodes and capacitors at the inverting input, the LM139's low level output enables a latch made from HC00 NAND gates to allow a chip select from the 'HC688 *(Figure 3)* or the NSC810A *(Figure 4)* to flow through to pin 1 ($\overline{CS}$) of the MM58174A.

As power from the 5.0V supply falls below 4.5V, the comparator's output immediately goes high. This threshold voltage is adjustable by the 200 kΩ potentiometer at the LM139's inverting input. A high output from the comparator to the NAND latch will disable chip selects to the MM58174A.

So as power begins to fail, this circuit will allow reads or writes to the MM58174A to go to completion if the chip is selected before the LM139's output goes high. It is assumed that the MM58174A's $\overline{CS}$ pin returns high before the supply falls to 4.0V (the minimum V_{CC} to access the chip). The length of the chip select strobe determines the limit of how fast the main power supply can drop from 4.5V to 4.0V. In situations where power failure detection is more critical, it is suggested that the comparator's output be connected to the microprocessor's highest priority interrupt so that the necessary software can be accessed.

This power-down circuitry has the advantage of proper operation in the presence of noise. With a slowly falling power supply in a noisy environment, the comparator's output may oscillate momentarily. This oscillation will have no bearing on the chip select signal to the MM58714A in this circuit because the HC00 latch only allows chip selects when the LM139 output is high, and it also does not alter their length once they begin. When the supply falls low enough to stop the comparator from oscillating, chip selects are locked out. One may consider the time that the comparator bounces as a delay before chip access is completely locked out as the standby mode is entered. If the cessation of comparator oscillation is desired, hysteresis can be added. A diagram of this can be found in the LM139 data sheet.

Real Time Clock Interface without Wait States Using a PIA

Figure 4 shows the details of a design using the NSC800™ CMOS microprocessor and the NSC810A peripheral interface adapter. This design has the advantages of lower chip count and the absence of wait states. Similar PIAs, such as the INS8255 or the 8155, could be used with some software adaptation. The power-down circuitry is operationally equivalent to that of *Figure 3*, except that in this design the chip select is created by the PIA. Only the essential connections between the NSC800 and the NSC810A are shown in *Figure 4*.

The NSC800 is an 8-bit CMOS microprocessor combining the features of the Intel 8085 and the Zilog Z80®. In this application 8085 code is used to manipulate the control strobes and handle interrupts as detailed in *Figures 5* through *9*. The interconnection between the NSC800 and the NSC810A is straightforward, except for the CE connection on the NSC810A. By tying CE to A13 of the NSC800, chip enabling occurs whenever an IN 2X or an OUT 2X instruction is executed, because the same port address appears on NSC800 lines AD0–AD7 as on A8–A16. Using 2X will raise A13 on the NSC800 high, where X represents a specific port address. This method of enabling is entirely optional. For more information on the NSC800 and NSC810A, refer to the NSC800 Microprocessor Family Handbook.

Software Description

The ports on the NSC810A are specially configured to control the data, address and control lines. The software allows the port signals to fulfill timing requirements. Port C is used to control the $\overline{WR}$, $\overline{RD}$ and $\overline{CS}$ lines, port B is used to control the address lines, and port A is used to read and write the data.

The NSC810A is configured into the strobed input mode in the read subroutine in order to get the shortest possible $\overline{RD}$ strobe. As stated previously, the read strobe must be under 15 μs to guarantee proper counter operation. The $\overline{RD}$ strobe is fed back to the PC2/$\overline{STB}$ input of the NSC810A in order to latch in the data from the MM58174A. The read subroutine of *Figure 5* begins by setting the port C direction. All bits are set for output, except PC2/$\overline{STB}$, which is set for input. Port B is set out and port A is set in. Next, all the control strobes from port C are set high using bit set. Before calling the read routine, the MM58174A address to be accessed was loaded into the NSC800's register B, and it is now sent out on port B. Bit clear is used to lower the $\overline{CS}$ strobe from PC5. The mode definition register is then written to for selecting the strobed mode of the NSC810A. Bit clear is used to lower the $\overline{RD}$ strobe from port C, and before it is raised again, a MOV instruction puts control values from port C into the accumulator in the shortest time possible. Using these three instructions, the read strobe is held low for about 5 μs. The rising edge of the $\overline{RD}$ strobe is fed into PC2/$\overline{STB}$ to latch the data into port A, and the IN instruction reads the data. Before exiting the read subroutine, the mode definition register of the NSC810A is again accessed to return the PIA's operation to the basic I/O mode. A wait loop may be added to the read subroutine or elsewhere in the code to limit the number of read strobes to less than 10,000 per second. This specification has been added because

more than 10,000 reads per second can slightly degrade timekeeping accuracy. The write subroutine of *Figure 6* uses the NSC810A ports in the same manner as the read routine; i.e., port A for data, port B for address and port C for control strobes. However, there is no need to latch the data in port A, so the basic I/O mode is used. The write subroutine uses the control strobes from port C by beginning with all three strobes high, manipulating $\overline{CS}$ and $\overline{WR}$ low, and finally bringing these port outputs high again. Before calling the write subroutine, the desired address to be accessed is to be stored in the NSC800's register B, and data stored in register A.

Figure 7 shows the necessary initialization code for the NSC800, NSC810A, and the MM58174A, which use the read and write subroutines. Of greatest importance is the code to insure that the clock is not in the test mode. Notice that a DI instruction is used to disable interrupts before a ''0'' is written to address 15. Also included is the code to initialize interrupts on the MM58174A. *Figure 8* shows the interrupt service routine, while *Figure 9* shows a method of time setting by first stopping the clock, then restarting it once the setting is complete.

```
                    ;DATA IS RETURNED INTO REG A
READ:   MVI A, 0FBH ;SET PORT C
        OUT 026H            ;DIRECTION
        MVI A, 0FFH ;SET PORT B
        OUT 025H            ;DIRECTION OUT
        MVI A, 00H  ;SET PORT A
        OUT 024H            ;DIRECTION IN
        MVI A, 038H ;SET PC3, PC4 & PC5 HIGH
        OUT 02EH            ;USING BIT SET
        MOV A, B    ;PUT ADDRESS IN A
        OUT 021H    ;ADDRESS OUT ON PORT B
        MVI A, 020H ;BIT CLEAR - PC5
        OUT 02AH            ;CHIP SELECT
        MVI A, 01H  SELECT STROBED
        OUT 027H            ;MODE
        MVI C, 030H ;GET READY
        MVI A, 010H ;BIT CLEAR - PC4
        OUT 02AH            ;RD STROBE
        MOV A, C    ;LATCH DATA IN PORT A
        OUT 020EH           ;& BRING STROBES HIGH
        IN 020H     ;GET DATA FROM PORT A
        ANI 0FH     ;MASK-OUT LOWER BITS
        MOV C, A    ;SAVE DATA
        MVI A, 00H  ;RETURN TO
        OUT 027H            ;BASIC I/O MODE
        MOV A, C    ;RECOVER DATA
        RET
```

FIGURE 5. Read Subroutine

```
                    BEFORE CALLING WRITE SUBROUTINE
                          ;STORE 174A ADDRESS IN REG B
                               ;AND DATA IN REG A
WRITE:  MOV C, A    ;SAVE DATA IN REG C
        MVI A, 0FBH
        OUT 026H    ;SET PORT C DIRECTION
        MVI A, 0FFH
        OUT 022H    ;SET PORT C HIGH
        OUT 024H    ;SET PORT A DIRECTION OUT
        OUT 025H    ;SET PORT B DIRECTION OUT
        MOV A, B    ;MOVE 174A ADDRESS TO REG A
        OUT 021H    ;ADDRESS OUT FROM PORT B
        MVI A, 020H ;CHIP SELECT — BIT CLEAR
        OUT 02AH            ;ON PC5
        MVI A, 08H  ;WRITE STROBE — BIT CLEAR
        OUT 02AH            ;ON PC3
        MOV A, C    ;RECOVER DATA FROM REG C
        OUT 020H            ;DATA GOES OUT
        MVI A, 0FFH ;SET PORT C
        OUT 022H            ;HIGH
        RET
```

FIGURE 6. Write Subroutine

```
        ORG 1200H       ;ORIGINATE @ 1200H
        LXI SP, 01FH    ;LOAD STACK POINTER
        MVI A, 00H      ;ENTER NON-TEST MODE
        MVI B, 00H           ;
        CALL WRITE                ;
        LXI H, VECTOR   ;"VECTOR" IS INTERRUPT SERVICE
        SHLD 1016H           ;ROUTINE @ 1016H
        MVI A, 04H      ;SET NSC800'S INTERRUPT CON-
        OUT 0BBH             ;TROL REGISTER FOR RSTA
        DI              ;DISABLE NSC800 INTERRUPTS
        MVI A, 00H      ;ENABLE INTERRUPTS ON 174A
        MVI B, 0FH           ;
        CALL WRITE                ;
        CALL READ                      ;
        CALL READ                           ;
        CALL READ                                ;
        EI              ;ENABLE NSC800 INTERRUPTS
```

FIGURE 7. Initialization

```
VECTOR: MVI B, 0FH
        CALL READ
        CALL READ
        CALL READ
        EI
        RET
```

FIGURE 8. Interrupt Service Routine

```
        MVI A, 00H       ;STOP CLOCK USING
        MVI B, 0EH            ;START/STOP FLIP-FLOP
        CALL WRITE

        (time setting code)

        MVI A, 01H
        MVI B, 0EH
        CALL WRITE
```

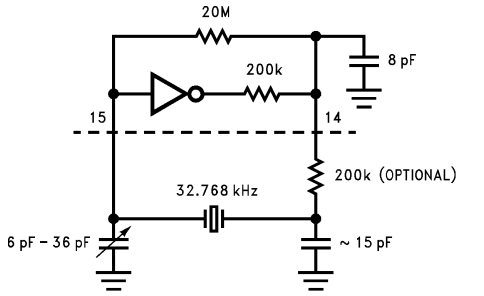
FIGURE 9. Recommended Procedure for Setting Time

Oscillator Design

The MM58174A is driven by a standard Pierce oscillator. *Figure 10* shows both the internal and external component sizes to be used. For crystals with a power rating of less than 1 μW, a 200 kΩ resistor, in series with the oscillator output, should be used to insure that the crystal is not overdriven. The typical gain for the internal inverter and internal 200 kΩ series resistor is 20 at 1 kHz input frequency and about 5 at 30 kHz. The oscillator may take from two to seven seconds to begin oscillating due to the high Q of the crystal.

Crystal Information

Choose one of the following crystal types: parallel resonant or tuning fork (NT CUT or XY BAR) with a Q $>$ 35,000 and a frequency of 32,768 kHz. The load capacitance required ranges from 9 pF to 13 pF. The maximum power rating is 20 μW. The choice of crystal accuracy and temperature coefficient are left to the user. Two crystals used in our lab are RCD's #RV-38 and Saronix's #NTF3238C.



TL/F/6169–5

FIGURE 10. Crystal Oscillator

Oscillator Adjustment and External Drive

A well-tuned oscillator for the MM58174A will have a frequency error of no more than $\pm$10 ppm. This would result in the clock being off by $\pm$5 minutes per year. This is a worst-case number, taking into account such factors as temperature variation ($-$40°C to $+$85°C) and supply variation (2.2V to 5.5V). The external oscillator components can also contribute to error and this should be taken into account by the user.

Adjusting the trimmer capacitor at pin 15 will minimize the oscillator error. But simply putting a scope probe on the crystal will load the oscillator with at least 10 pF, significantly altering the frequency. There are two good ways of isolating the probe from the oscillator. One method is to put the part in the test mode by writing a ''0'' to DB3 at AD0, then tune the signal at DB0 to 16,384.00 Hz using an accurate frequency counter. Another method would be to isolate the oscillator from the probe by adding an inverter to the small capacitance at pin 14. This would load the oscillator, but the input capacitance of the gate would not be affected by a probe at the output. The total capacitance on pin 14 should be kept near 15 pF.

To drive the oscillator from an external clock, connect the clock to pin 14 (crystal out) and tie pin 15 (crystal in) high.

CONCLUSION

The MM58174A can easily be interfaced to a microprocessor to bring the functions of a real time clock and calendar to any system. With a power-fail/back-up circuit, the system will be able to keep accurate time for years, independent of the system power supply.

The MM58174A Real Time Clock in a Battery Backed-Up Design Provides Reliable Clock and Calendar Functions

AN-359

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
1111 West Bardin Road
Arlington, TX 76017
Tel: 1(800) 272-9959
Fax: 1(800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408