CALCULATOR LEARNS TO KEEP TIME

INTRODUCTION

A number of interesting stopwatch and elapsed time functions can be implemented using the MM5736 calculator chip and a few packages of CMOS gates. This note describes six different circuits that are intended to stimulate thinking along these lines. The circuits to be described are listed below.

1. Stopwatch with 1/10 second resolution
2. Stopwatch with 1/100 second resolution
3. Stopwatch/calculator (1/10 second resolution)
4. Stopwatch/calculator (1/100 second resolution)
5. Stopwatch with 1/10 secs, secs, mins display
6. Interval timer with keyboard and alarm

With the exception of circuits 5 and 6 all of these circuits work in decimal fractions of seconds. They do not display in seconds and minutes. Circuit 6 displays minutes and tenths of minutes but not seconds. Circuit 5 displays tenths of seconds, seconds and minutes. It is anticipated that a number of applications can be satisfied by counting in only one unit, either seconds or minutes.

In all these circuits, the MM5736 calculator chip is used in the autosumming mode as a counting and display element. Application note AN-112 illustrates how to accomplish this counting. A thorough understanding of the calculator's operation as a counter can be gained from AN-112 and the MM5736 data sheet. Consequently, the emphasis in this note is on controlling the counter in such a way that useful timing functions are performed.

Two types of timebases are also described. The first, a CMOS RC oscillator, is depicted in all the circuits described but may not be stable enough for some applications. Consequently, a simple crystal controlled timebase is also described.

STOPWATCH WITH 0.1 SECOND RESOLUTION

The circuit in *Figure 1* provides the classic stopwatch functions of:

A) START
B) STOP
C) RESET

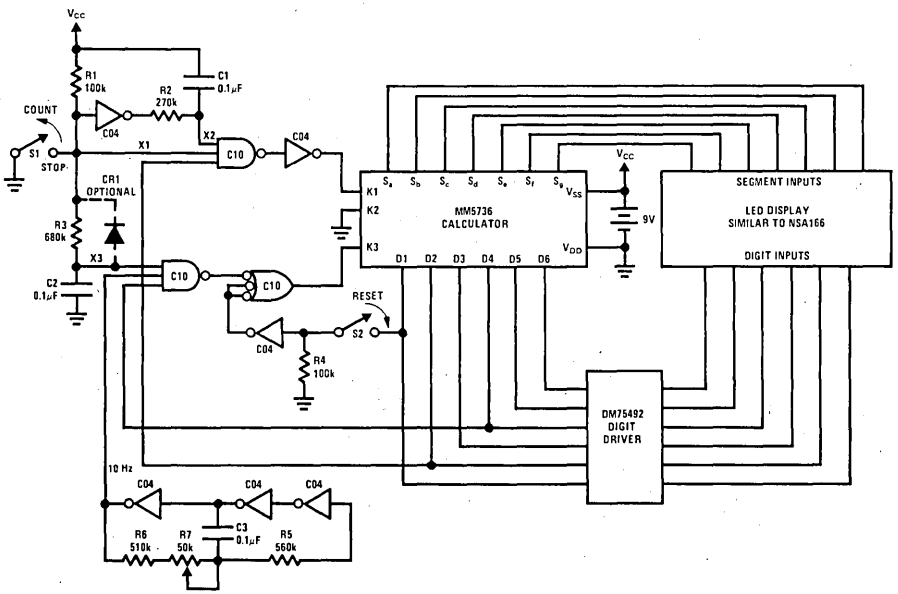

FIGURE 1. 1/10 Second Stopwatch

This implies that timing may continue after it has been stopped without resetting to zero. The display will be in tenths of seconds and seconds. Thus, 3 minutes 11 1/2 seconds would be displayed as 191.5 seconds.

Circuit Description

The RESET switch simply gates D1 into the K3 input of the calculator and clears it. Upon initial power up it will be necessary to activate RESET twice. From then on, only one RESET activation is necessary.

When the COUNT switch is activated, R1 pulls up signal X1. This makes all the inputs to the 3 input gate high and gates D2 into input K1. This will cause a 1 to be entered into the calculator. Signal X2 is delayed by R2 and C1 and will go low about 25 ms after X1 and shut off the D2 pulses being gated into input K1. About 40 ms after this, signal X3 (which is delayed through R3) goes high and D4 is gated into the K3 input at a 10 Hz rate. This causes repeated additions and results in the calculator counting at this rate. When the COUNT switch (S1) is returned to the STOP position the additions will be stopped about 60 ms later (after X3 is delayed through R3). This delay hardly seems objectionable since it is less than the resolution of the counter. However, purists may feel the addition of CR1 is necessary. This will cause the counting to stop immediately after S1 is returned to the STOP position since C2 will be discharged immediately.

The LED display shown differs from the NSA166 only in the placement of the decimal point.

STOPWATCH WITH 0.01 SECOND RESOLUTION

Figure 2 depicts a circuit that is identical to the one shown in *Figure 1* except that it has a resolution of 0.01 sec. This means the counter must run at a 100 Hz rate which is normally beyond the capability of the MM5736. However, as described in AN-112, a trick can be played with D6 and D4 that will double the effective counting rate of the calculator. This trick is accomplished by forcing D4 high during D6 time. SWD is a bilateral switch that connects D6 to D4. CR2 keeps D4 off D6. SWC is turned off during D6 time so the extra pulse on the D4 line will not get to either the LED display or the data entry logic.

The remainder of the circuit operates exactly like the one in *Figure 1* with the exception that some of the gating is implemented with the MM5616 switches.

STOPWATCH/CALCULATOR WITH 0.1 SECOND RESOLUTION

Figure 3 depicts a combination stopwatch and calculator. Stopwatch operation is the same as described earlier with the following exceptions: Transistors Q1 and Q2 (which are small GP switches) are used to switch either D2 or D4 to the calculator inputs. This allows the keyboard and the stopwatch logic to operate in parallel in what amounts to a "wire OR'ed" arrangement. Also there is no RESET switch in this circuit since the calculator's CLEAR key can be used to reset the time to zero.

Normal "four function" calculator operation is available when S1 is in the "CALCULATE" position.

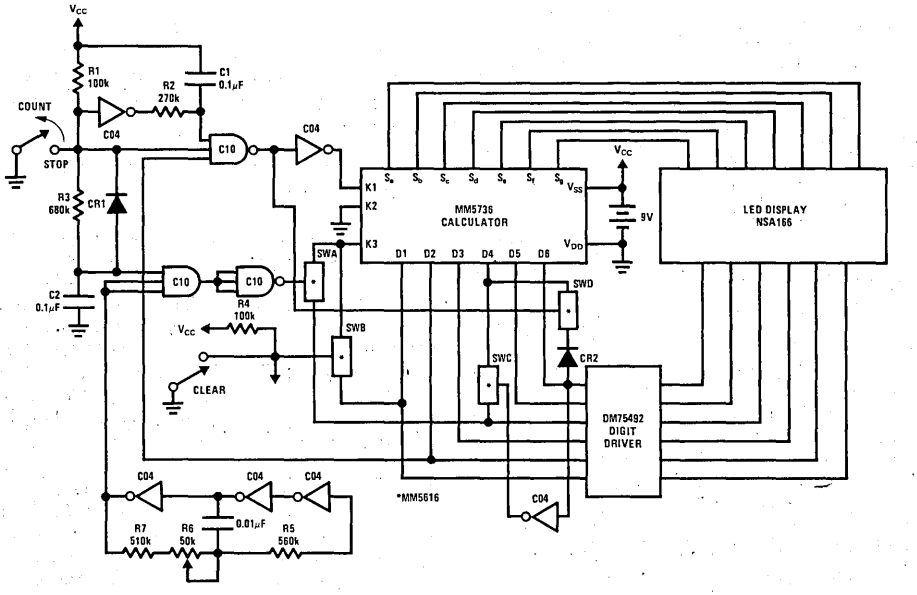

FIGURE 2. 1/100 Second Stopwatch

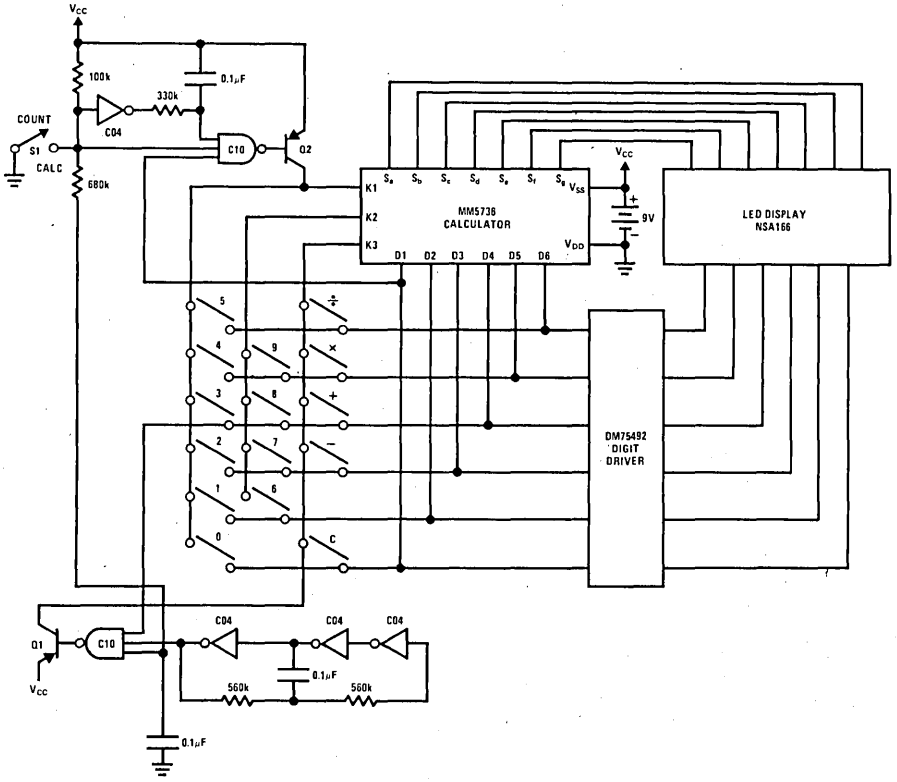

FIGURE 3. 1/10 Second Stopwatch/Calculator

STOPWATCH/CALCULATOR WITH 0.01 SECOND RESOLUTION

Figure 4 is just a souped up version of *Figure 3.* It will count at a 100 Hz rate giving a 0.01 second resolution to the stopwatch. Switch S2 now needs to be DPDT type. One pole of the switch provides the start-stop function and one pole is used to switch in the "speed-up" circuit involving D4 and D6. The additional gating keeps the extra pulse from reaching the display and keyboard.

STOPWATCH/CALCULATOR DISPLAYS MINUTES AND SECONDS

The conventional time keeping format of minutes and seconds can be obtained with the additional logic shown in *Figure 5.* This circuit provides a display of time up to 999 minutes, 59.9 seconds. But this requires a base sixty counting capability that is not inherent in the calculator chip. This conversion is accomplished by recognizing when the count has gone to 60.0 seconds and then quickly adding 40.0 to the count, thus giving an apparent base 60 carry. The sequence of operations required to do this is:

1. Recognize 6 in 3rd digit
2. Enter 3 into calculator
3. Enter 9 into calculator
4. Enter 9 into calculator
5. Enter + into calculator
6. Enter 1 into calculator
7. Enter + into calculator
8. Resume normal operation

This sequence leaves the calculator properly initialized with a "1" in it ready for more counting. This would not be the case if 400 was entered directly rather than as 399 + 1.

Circuit Description

The base 60 conversion is accomplished with a little controller that switches the Digit outputs to the proper calculator inputs through some FET switches. The sequencing it provided by an 8-bit counter and a decoder. If desired, the circuit could be re-implemented to use MM5616 quad switches rather than the MM552's shown. But, since the simplest device that will do this job is a MOS transistor, it was chosen in this particular case. It also lends itself to the negative going outputs of the MM74C42 decoder.

When the stopwatch is counting normally (rather than doing a base 60 conversion) the MSB (Q_D) of the sequence counter will be low which inhibits counting. It also turns on transistor Q1 which will allow the counter to be preset by the output of the gate that decodes a "6" according to the expression $S_e \cdot S_b$, which is a simplified version of the seven segment code for "6."

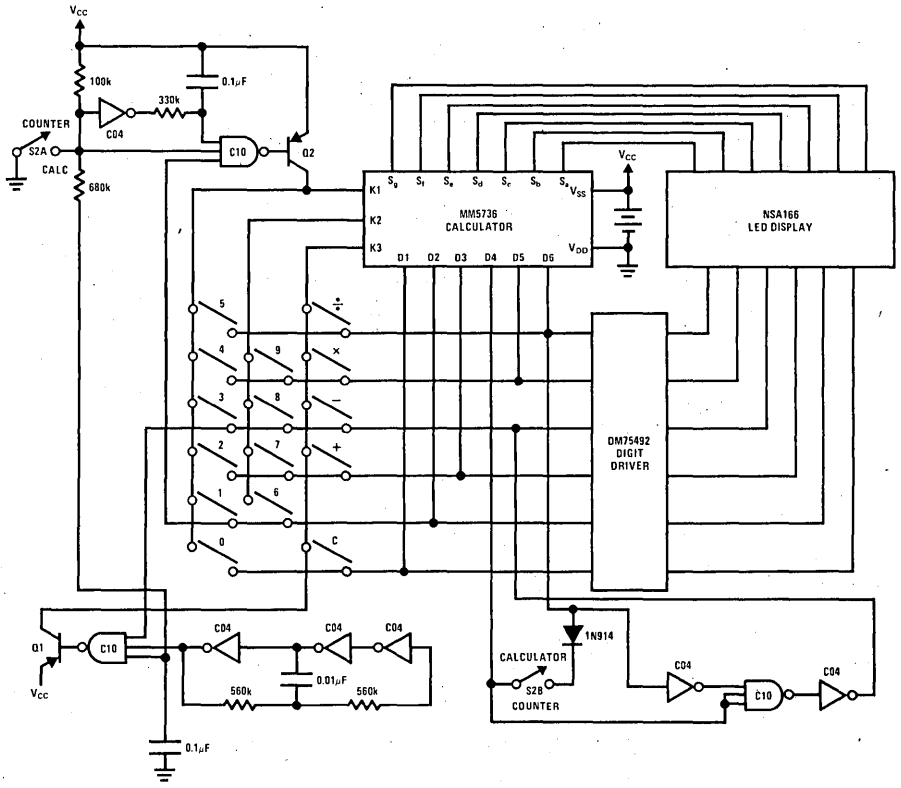

FIGURE 4. 1/100 Second Stopwatch/Calculator

Since the counter is clocked by D3, only a 6 in the 3rd digit will cause the counter to be present. This corresponds to a time of 60.0 seconds and signals the beginning of a base 60 conversion. The counter is preset to the state 1001 0000. Since the MSB is a 1, the counter's count enable term is enabled and its load term is disabled. It will now count word times on every D3.

Reference to AN-112 will reveal that with the calculator "speeded up" it is necessary to allow a digit output to be connected to the inputs for a minimum of 4 word times and then there must be at least 4 word times during which nothing is applied to the calculator inputs before the next entry is allowed. This timing is accomplished by Q_D of the low order counter. It toggles with a half period of 8 word times. This Q_D is connected to the D input of the decoder which is used as an enable input. When this signal is high, all outputs of the decoder are high and all the MOS transistors are off. When this signal is low the proper decoder output is low. So the first 4 bits of the counter provide timing and the next 3 bits provide the necessary sequence of entries. The last bit turns the sequence on or off. The sequence of entries is as described earlier and is implemented by transistors Q2–Q7.

Initialization

When S1 is first switched to the stopwatch mode, a burst of D2 pulses is gated into the K1 input by the one shot comprised of R2, C1 and the gate that drives Q8. This enters a "1" to get the calculator ready to count. A little later, Q9 will be turned on by the timebase oscillator at a 10 Hz rate and counting will begin.

Segment Drivers

Two DM8895 segment drivers are used in *Figure 5*. This is not absolutely necessary. The calculator can drive some displays directly. However, it is necessary to buffer both segment e and segment b to preserve proper logic levels for the CMOS decoding gates. This could be done by non-inverting CMOS buffers like the MM80C96 or 2 inverters in series. But if only S_e and S_b are buffered, there is no guarantee of segment intensity uniformity. Therefore, it is more desireable to buffer all segments. The DM8895 is a segment driver with internal current limiting resistors that are mask programmable. The DM75491 could also be used if external resistors are not objectionable.

"Speed Up" Circuit

Transistors Q11 and Q10 implement the "speed up" function in the same way as that described in *Figure 2* except that naked MOS transistors are used in place of the MM5616 CMOS switch.

AN INTERVAL TIMER WITH A KEYBOARD

An interval timer that can be programmed to time out long intervals can also be made using the calculator chip. The desired time interval is entered from a keyboard. When the interval is complete, a tone is emitted by a small speaker until the operator activates a RESET switch. The timer (as described) will handle intervals as long as 99999.9 minutes, which is about 69 1/2 days. This is probably too long an interval for an RC oscillator to be acceptable as a timebase. *Figure 6* shows an RC oscillator but it could be replaced by the crystal oscillator described later in this note. Counting speeds other than 0.1 minutes could be used as long as the counting speed of the calculator is not exceeded.

Circuit Description

As was the case for the stopwatch described in *Figure 5,* a small controller made from a counter and a decoder is used to switch Digit outputs to the proper K input to create the sequence of entries required. The counter is clocked by a 30 Hz oscillator whose output is also gated with all the Digit lines to create the proper "key down" and "key up" times.

There are two sequences of entries required: one for RESET and one for START, the beginning of the timing interval.

Reset Sequence

When the RESET switch is activated, it is debounced by a latch and differentiated by C1 to generate a positive going pulse that clears the MM74C193 controller counter and the sequence proceeds as follows:

1. **Reset Latches:** The "0" output of the decoder resets the zero decode latch and the buzzer latch.

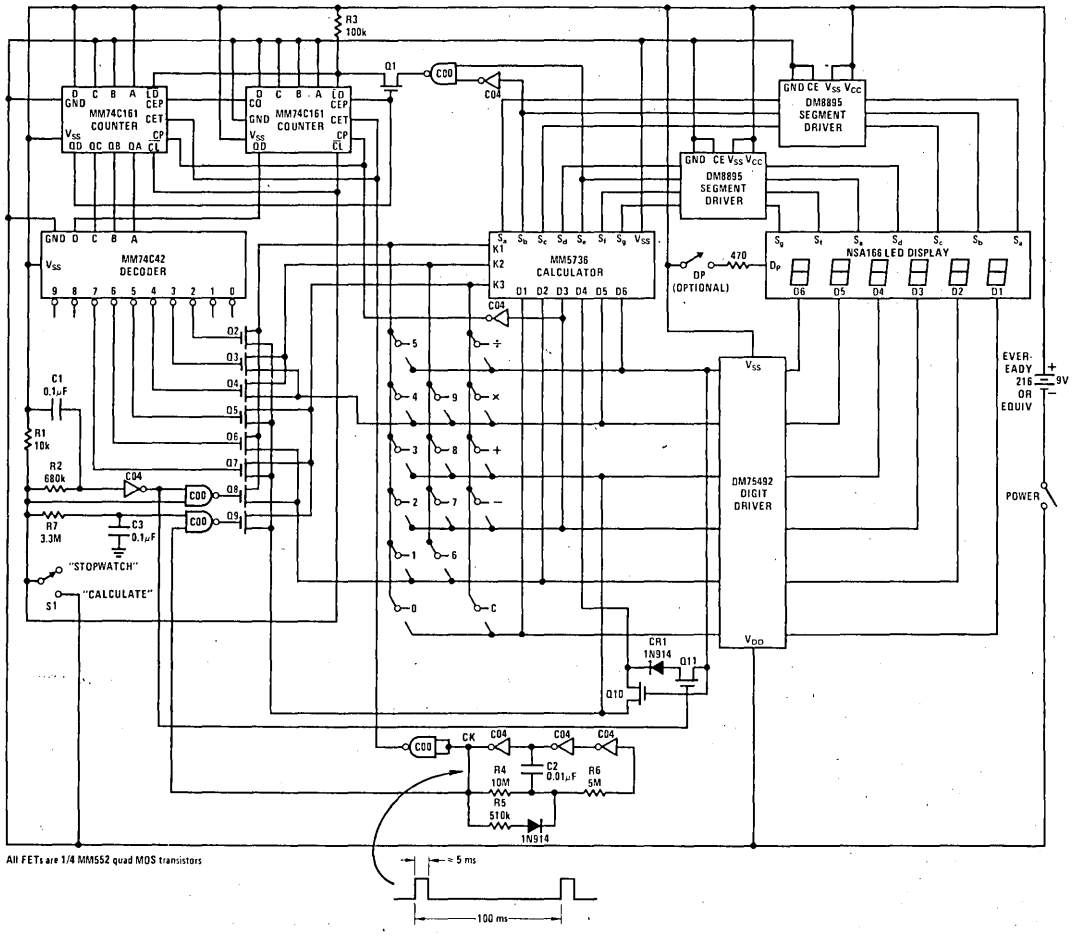

FIGURE 5. 1/10 Second, Seconds, Minutes Stopwatch/Calculator

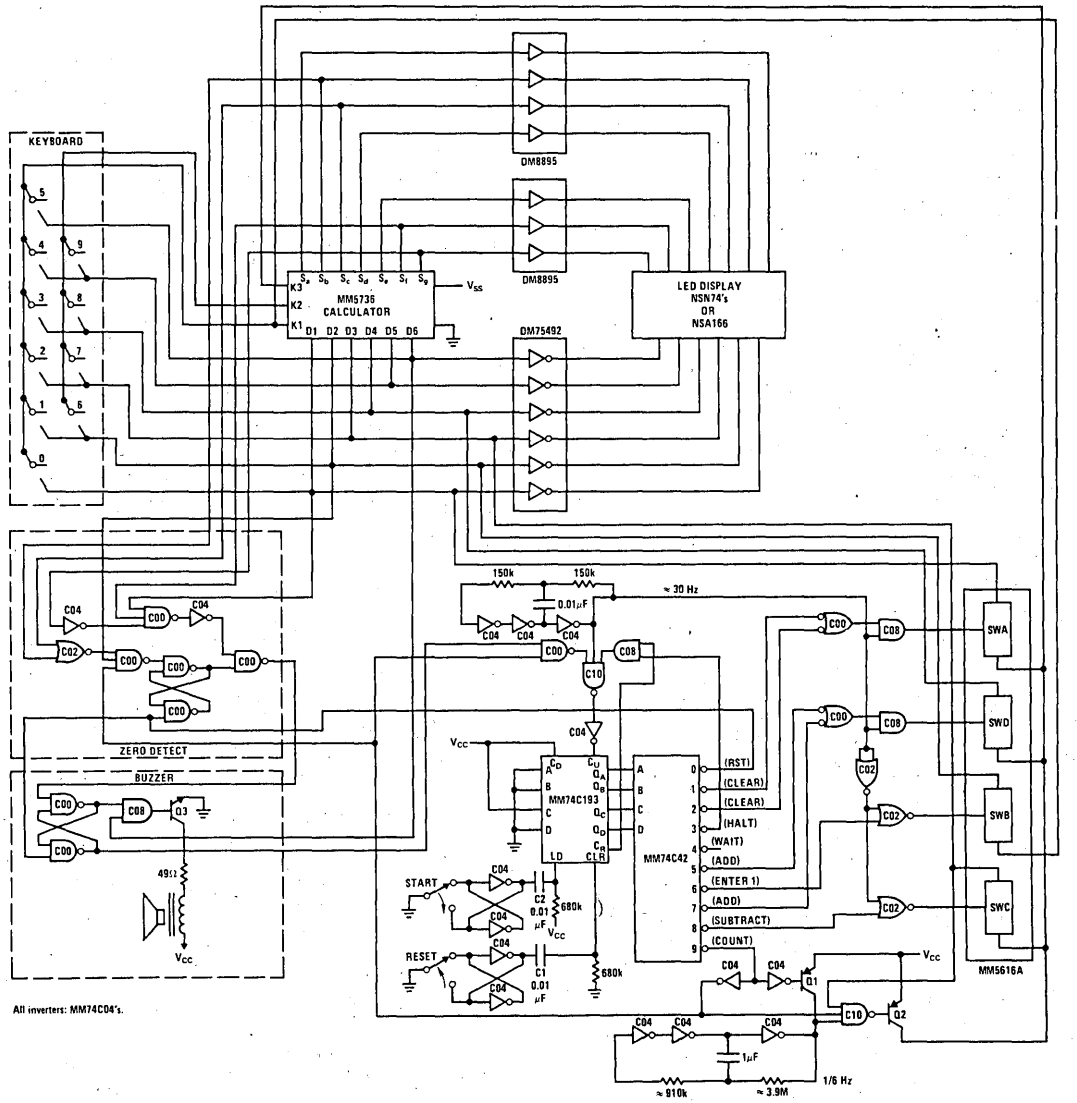

FIGURE 6. Interval Timer

2. **Clear Calculator**
3. **Clear Calculator:** Both outputs 1 and 2 of the decoder are "or'ed" and then gated to switch D1 into the K3 input of the Calculator to cause a clear. Two clears are necessary to insure that all registers are reset to zero.
4. **Halt:** Decoder output 3 forces count enable low and hangs up the counter.

Start Sequence

When the START switch is activated, C2 differentiates the latch output and generates a negative going pulse that loads the counter to state 4. Since this can happen at any time with respect to the 30 Hz clock, it is necessary to wait until the counter goes to the next count before trying to enter anything into the calculator. This is done to insure that a full cycle of the 30 Hz clock elapses during the time an entry is being made. The sequence then proceeds:

1. **Synchronize:** Decoder output 4 does nothing but insure that the first application of signals to the calculator will last for a complete interval.
2. **Add:** Decoder output 5 causes D4 to be gated into input K3 causing an add. This will enter (in normal Polish notation) the number already entered from the keyboard.

3. **Enter 1:** Decoder output 6 gates D2 into K1 to enter a 1. This is the number that will be repeatedly subtracted to make the total count down.
4. **Add:** This simply causes the 1 just entered to be added to the number that was entered from the keyboard. The total will now be one count higher than desired. Since this would shake up most users, the next step corrects this.
5. **Subtract:** Decoder output 8 causes a subtraction which decrements the display by 1 and brings it back to the correct reading.
6. **Count:** Decoder output 9 makes the controller halt and also turns transistor Q1 off. Q1 was initializing the timebase oscillator so the timer won't begin to count down prematurely. D3 is also gated into the base of Q2 which causes repeated subtractions at the timebase rate.

At this point the timer simply chugs away decrementing until it reaches zero. Time remaining to zero is continuously displayed. When zero has been detected, the controller's count enable term will go high and it will advance to state 15 at which time the "carry out" term will go high and inhibit any further counting. It will stay this way until the RESET button is activated.

Zero Decode Logic

A zero is detected by recognizing that a blank exists in digit 2 and a 0 exists in digit 1. A blank is decoded with the expression $S_b + S_c$ since one of these two segments is always on when any number is being displayed. When BLANK · D2 exists, a latch is set. Then when a zero is detected in digit 1 according to the expression $S_f \cdot \overline{S_g}$ the buzz latch is set. This gates D6 into the base of Q3 which turns the speaker on at about a 1 kHz rate with a 1/6 duty cycle and generates a buzz. The buzz latch will be reset during the RESET sequence.

OSCILLATORS

Two CMOS oscillators have been mentioned: one RC and one crystal controlled. These oscillators are analyzed elsewhere in National's applications literature (AN-118) so only a summary is given here.

RC Oscillator

An odd number of inverting gates (NAND, NOR, INVERTERS) will always oscillate if tied around on themselves as in *Figure 7.* Most beginning logic designers have discovered this fact of life by accident at one time or another.

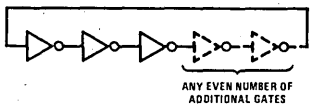

FIGURE 7. Odd Number of Gates Always Oscillates

Odd Number of Gates Always Oscillates

The oscillator will generate a square wave whose frequency will be determined by the propagation delay through the gates. All that remains to make this a useful oscillator is to control the frequency of oscillation. *Figure 8* depicts a simple and foolproof way to do this.

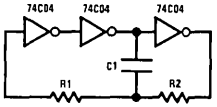

FIGURE 8. RC–CMOS Oscillator

The frequency of oscillation will be about f = 0.55/R2 C if R1 = R2. R2 has the most effect on frequency and in most applications it would be a pot. Stability of the oscillator as a function of time is dominated by the passive elements, especially at frequencies as low as 100 Hz or less. Variations in output drive capability of the CMOS will be swamped if R2 is 100k or more. Stability with respect to supply voltage in the range of voltages that can be used with the calculator chip (6.5–9.5V) is a function of frequency but the following is representative:

FREQUENCY	VARIATION (6.5–9.5V)
100 Hz	≈3%
10 Hz	≈0.5%

Empirically determined temperature drift of this oscillator due only to the CMOS is:

FREQUENCY	DRIFT
100 Hz	0.03%/°C (−15 → +50°C)
10 Hz	0.01%/°C

Crystal Oscillator

Figure 9 illustrates how to build a crystal oscillator using CMOS. This oscillator is also described in AN-118.

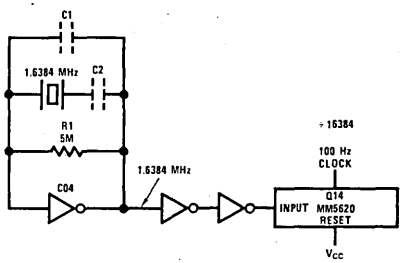

FIGURE 9. CMOS Crystal Oscillator and Divider for 100 Hz

The CMOS inverter is biased into its linear region by resistor R1. This dc path around the inverter ensures that the oscillator will start. C1 can be used to pull the crystal down and C2 to pull it up. The output of the oscillator is cleaned up by the next two inverters. This signal then is divided by 2^{14} or 16384 to yield the 100 Hz clock needed for the 0.01 second resolution timers.

The 0.1 second resolution timers could be obtained by using the dividing logic as suggested in *Figures 10 and 11.* The interval timer could use the 0.1 minute time base shown in *Figure 12.*

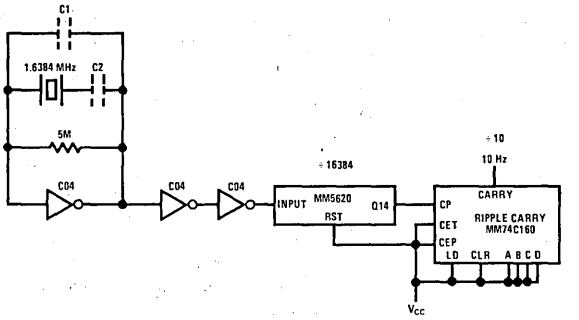

FIGURE 10. Divider for 10 Hz

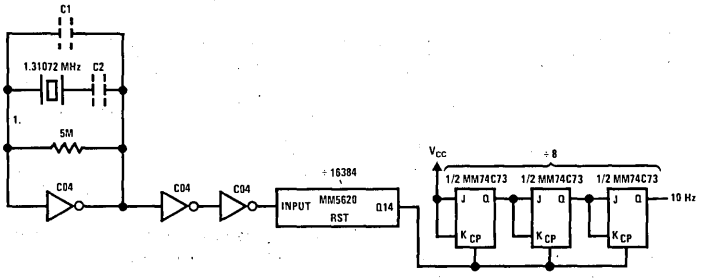

FIGURE 11. Alternate Divider for 10 Hz

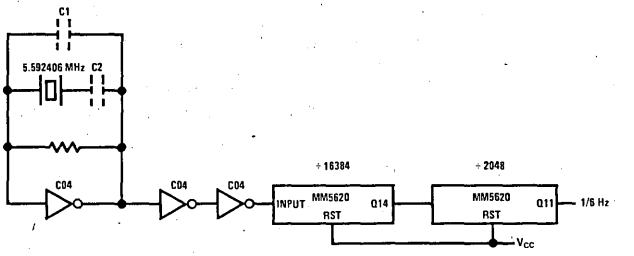

FIGURE 12. Divider for 1/6 Hz

SUMMARY

A rich variety of timing functions can be done digitally and many of these can be implemented with the MM5736 calculator chip. The MM5736 offers six decades of counting and display in one package and will yield low parts count solutions to many of these problems. It can be used in a variety of ways, it interfaces ideally with the 74C line of CMOS and consumes little power.