High-resolution frequency counter

Using a microprocessor enables the designer to provide better resolution at low frequencies and to add such useful features as sum, difference and ratio measurement.

STEPHEN THEOBALD

The circuit presented here was designed to overcome some of the limitations of conventional instruments using decimal dividing chains. Such instruments are generally characterized by poor resolution at low frequencies unless very long measuring periods are resorted to, and are rather inflexible.

Several specialized and, in most cases, highly-priced integrated circuits are available, offering nearly all the hardware necessary for building a complete counter, but these suffer from the same disadvantages. The basic design has for a long time now needed a little freshening up.

In line with modern practice, a powerful eight-bit microprocessor is at the heart of the circuit, resulting in fairly simple supporting hardware composed of a small number of inexpensive and readily-available 74-series devices. The computational abilities of the microprocessor make it possible to produce a very versatile instrument, capable of much more than just frequency measurement, but the extra facilities add almost nothing to the cost. The fact that the instrument is under program control also means that it can, if necessary, be tailored to particular applications.

The most important extras that the design offers are the sum, difference and ratio functions. The first two of these allow intermediate frequencies to be taken into account in radio receivers, regardless of whether the signal picked up lies above or below that of the local oscillator. It is only necessary to store the intermediate frequency in the memory, and the frequency to which the receiver is tuned can be read out directly by measuring the local oscillator frequency and applying the appropriate function. Again, the difference mode makes it an easy matter to determine drift in an oscillator. Simply make a measurement of the frequency, store the value and then switch over to the difference mode. Deviation will be displayed automatically. The last of these functions, the ratio of the input to some previously stored frequency, proves useful when working on frequency synthesis.

SPECIFICATION

Overall frequency range: 25μHz to 1GHz in three ranges, autoranging within each range.

Range function		Min.resolution	Accuracy
VLF[1] freq.	25μHz–3Hz	1 part in 10^4	± 1 digit
period	40ks–330ms	1 part in 10^5	± 1 digit
MF freq	0.9Hz–>30MHz	1 part in 10^4	± 1 digit
period	1.1s–<33ns	1 part in 10^4	± 1 digit
HF freq.	<25MHz–1GHz	4 parts in 10^6	± 1 digit
period	>40ns–1ns	1 part in 10^5	± 1 digit
Pulse	200ns–1s	200ns	+400ns, −0
	1s–50ks	1 part in 10^6	± 1 digit
Event[2]	0–9 999 999	–	–
Clock	0–9999.999s	1ms	±1ms
Freq. ratio[3]	10^{-9}–10^{+9}	1 part in 10^4	± 1 digit

Note 1. This range extends to 31Hz (period to 31ms), with lower resolution.
Note 2. The maximum pulse repetition rate should not exceed 10kHz.
Note 3. Ratio, sum and difference modes are not available on vlf ranges.

Input	Freq.range	Sensitivity	Input resistance	Max. input
VLF/MF amp.	0–40MHz	30mV pk-pk	1MΩ//25 pF	50V pk-pk
HF prescaler	25MHz–1GHz	10mV rms	50Ω approx.	3V pk-pk

MEASURING LOW FREQUENCIES

Let us first consider how to measure low frequencies very accurately, while keeping short the time required to do so. For frequencies not less than about 1Hz, we can use the hardware arrangement shown in **Fig.1**. The controller, which is the cpu and an input port, can enable, read, and reset the counter, and must also be able to measure elapsed time in short units of time, dt seconds.

Initially the counter is reset and then enabled (**Fig.2**). The controller synchronizes the start of the measuring period with a positive edge at the clock input, which it detects by sensing that the lsb of the counter changes from 0 to 1. Incoming pulses are now counted for a fixed length of time, Xdt. At this point the controller again notes the state of the lsb and continues to measure time for some variable period, xdt, until the next positive edge at the clock input causes the lsb to change state once more. Detecting this condition, the controller now disables the counter via the And gate and reads the count, which we shall denote as N+1. (Remember that the counter goes to 1 just before the measuring cycle starts.) N complete cycles of the input have now been received in a period of (X+x)dt seconds, and the frequency of the input is simply calculated as f=N/(X+x)dt Hz.

The degree of resolution obtainable depends on X, while the minimum duration of the measuring period to achieve a given resolution rests on the values of both X and dt. Worst case resolution occurs when the variable x assumes its minimum value of 1; that is, when f = N/(X+1)dt. The slightly lower frequency, f′, for which x becomes 2, is obviously f′ = N/(X+2)dt Hz. The difference between the two frequencies is the smallest change that the system can detect, and is given by

$$df = f - f' = N/(X+1)dt - N/(X+2)dt$$
$$= f/(X+2) \rightarrow f/X,\ X \gg 2$$

Resolution, which we define as df/f, is thus 1/X; and the shortest period necessary to achieve this value is Xdt. The design being presented here has values of X and dt of 10^5 and 10^{-5} respectively, permitting a frequency of 9.9998Hz to be distinguished from 9.9999Hz.

At any frequency, the maximum time which can elapse before a result can be calculated is given by Xdt+2/f seconds, i.e. 3 seconds at 1Hz, while the minimum time is Xdt+1/f seconds. The exact time depends on how long the system must wait for synchronization to occur. Although not shown here, a similar analysis for f″, the slightly higher frequency which can just be distinguished from f, yields the same results.

Synchronization is necessary up to at least 10kHz to ensure a minimum of five significant digits in the result. Above this point the system can revert to the more normal technique of just counting the input pulses, not bothering to synchronize to the input, provided that counting is performed for at least one second. This design continues to synchronize up to 25kHz because it was slightly easier to arrange it that way in the program.

The vlf range differs from the mf range in that the counter is not used. Instead, the controller reads the input directly and measures over just a single cycle of the input. This has the advantage that the start of the measuring period may be synchronized with either the positive or negative edge of the input, whichever occurs first, thus saving time and wear-and-tear on the operator when the input is very low in frequency. By maintaining dt at 10μs it is possible to attain five-figure precision up to √10Hz. Since it was desired that this range should extend to

a somewhat higher frequency, a switch from five-figure to four-figure readout occurs as f exceeds the 3Hz mark. A similar reduction to three-figure accuracy would be called for at frequencies greater than about 31Hz, so this figure was chosen as the upper limit of the vlf range. Aliasing occurs beyond this point, giving a readout of f/(INT(f/31.35)+1)Hz.

CIRCUIT ARRANGEMENT

Hardware (**Fig. 3**) is fairly straightforward except that the three highest address bus lines are used for control purposes. The range and function switches allow the cpu to determine which programs to call for a particular function as well as which input amplifier connection to select. The chosen input is applied to the input port either directly, for vlf measurements, or via the 12-bit binary counter. The pulse processing circuit, if selected, gates the clock oscillator with the input signal and sends trains of 5MHz pulses to the counter.

At the end of each measuring period, the processor disables the input and reads the input port. It then selects the 5MHz clock as the input to the counter and enables its own interrupt input, which then receives short pulses from the monostable each time Q_{11}, the counter's msb, changes from 0 to 1. These interrupts drive the multiplexing of the display while the cpu calculates the results. Thus the display is multiplexed at a rate in excess of 1.2kHz. Once the final result becomes available, the required display is stored in ram and the interrupt is disabled. The next measurement cycle is now undertaken while the cpu times the display drive internally, sending out the digits stored in memory to the output ports.

It is obvious that a 12-bit binary counter is inadequate for measuring signals higher than about 4kHz on its own, so the microprocessor detects overflow on Q_{11} by sampling this line via the port sufficiently often to ensure that the count is accurate, even at frequencies as high as 50MHz. In this way the counting chain is extended within the cpu.

In the main digital circuitry (**Fig 4, 5**), note that the 74150, which is a 16-line multiplexer used as an input port, is neither tri-state nor open-collector at its output. It is therefore connected to the data bus through a diode and pull-up resistor instead. The ram chosen for the design was the 2K 6116-LP3 since this is cheap and uses little power.

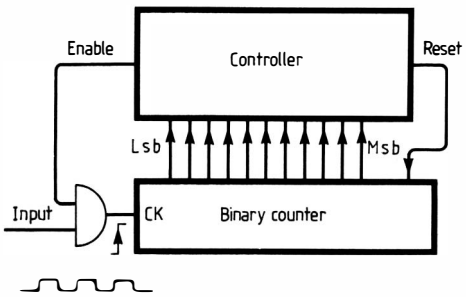


Fig.1. Basic hardware arrangement for measuring low frequencies.

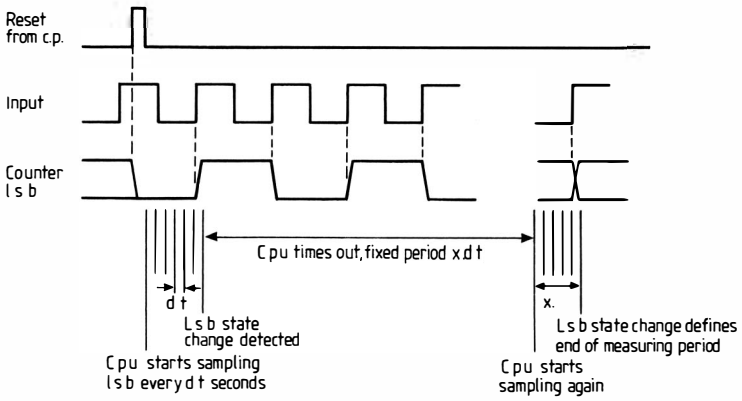


Fig.2. Sequence of events for measuring low frequencies.

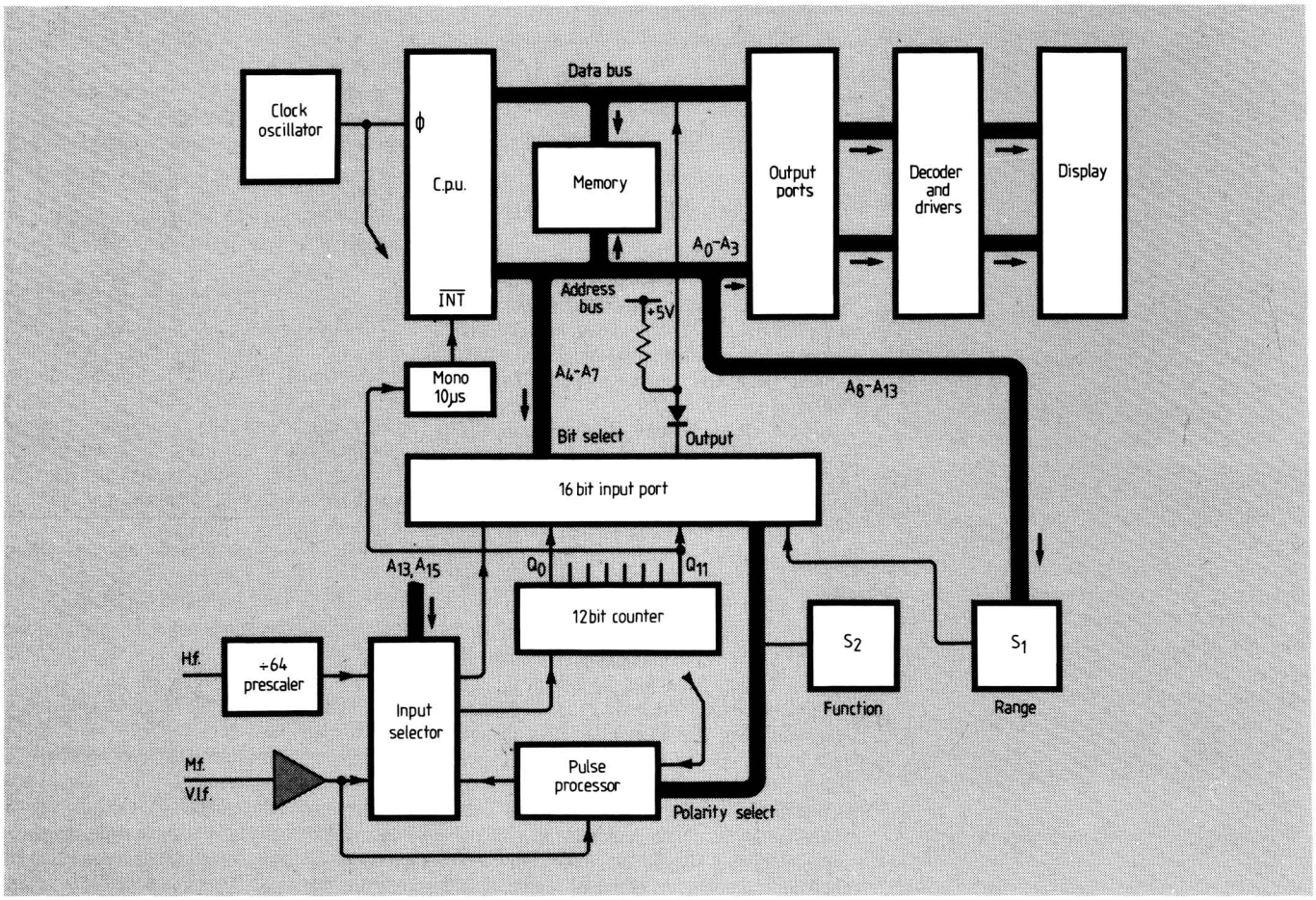


Fig.3. Block diagram of the high-resolution counter. The processor is a Z80.

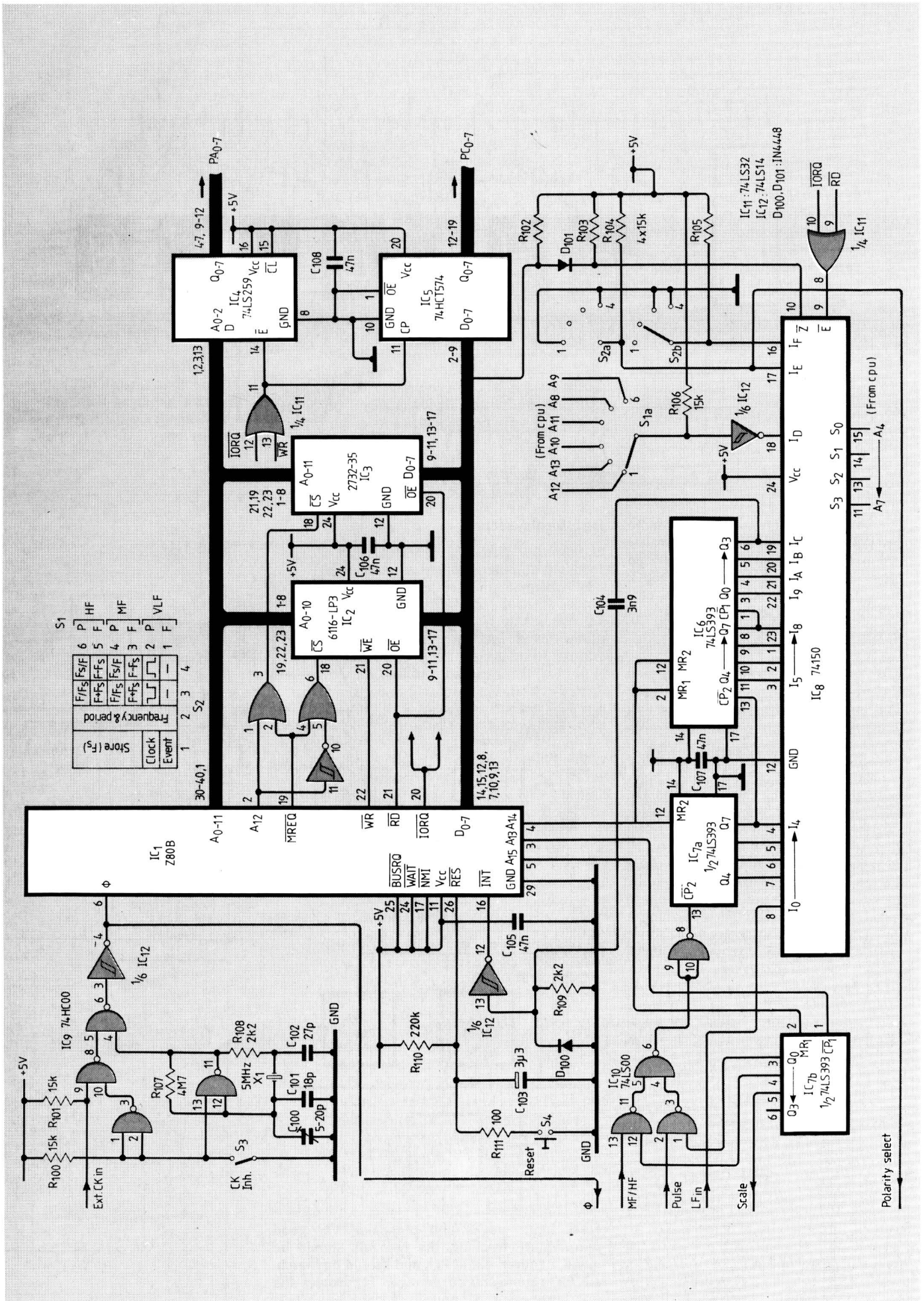


Fig.4. Digital section of the counter. A programmed eprom is available from the author – details at the end of this article.

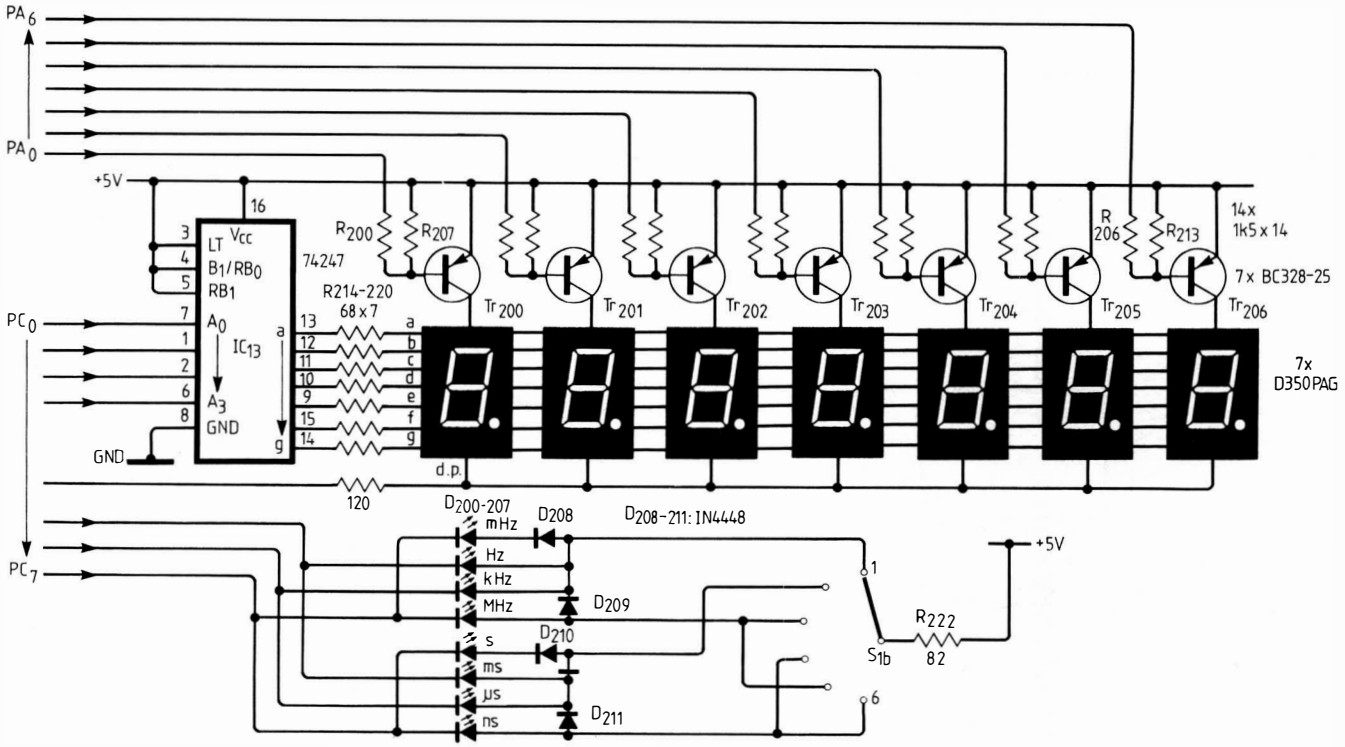


Fig.5. Display section.

However, the program requires no more than 128 bytes of ram and other, smaller memories may be used provided that cpu address lines A_0–A_5 plus A_8 are connected. Line A_8 became necessary as a result of a programming short-cut.

VLF/MF INPUT

Input amplifiers, selection network and pulse processor are all shown in **Fig. 6.** The vlf/mf amplifier has a frequency response extending from 0Hz to about 40MHz. The capacitor-resistor network feeding Tr_1 provides a high-impedance input for signals less than about 1V pk-pk of 1MΩ//20pF. Larger inputs are clipped by diodes D_1 and D_2, whereupon the input looks like a 1k resistor in series with 10pF. Relay Rl_a is used to select either ac or dc coupling. Potentiometer P_1 and the associated switched resistor chain allow an offset voltage to be applied to the gate of Tr_1 to balance out any direct voltage applied to the input when this is dc coupled. This arrangement also serves as a simple trigger control with capacitively coupled signals. The average value of low-level, narrow duty-cycle inputs may be such as to prevent reliable counting unless a small offset is superimposed. Signals are taken from the potential divider at the source of Tr_1 at about ground level and applied to one input of the 733 wideband video amplifier. The other input is connected to a similar tapping on the source of Tr_2. This transistor is only used to compensate for drift in the working point of Tr_1 with temperature variations. The 733 must not be overloaded too severely at high frequencies since its recovery time is then so great that it canot follow the input. Relay Rl_b is used to switch extra feedback into the circuit if necessary, to overcome this problem. The output of the 733 is differential and at ttl levels, so one of these outputs is fed to a Schmitt trigger to square up the signal before going further to the and routing network.

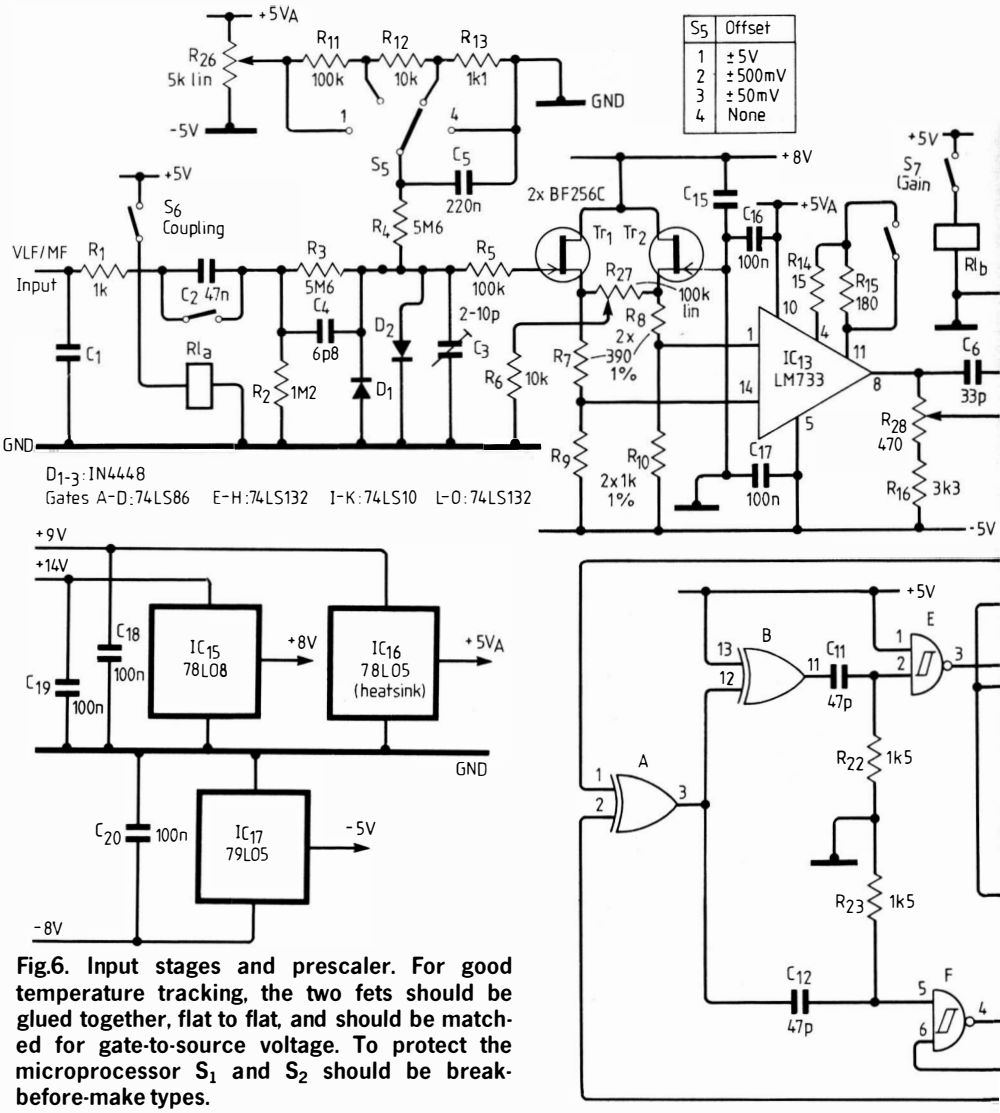
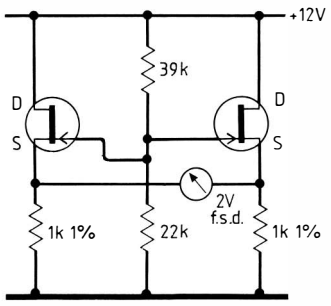
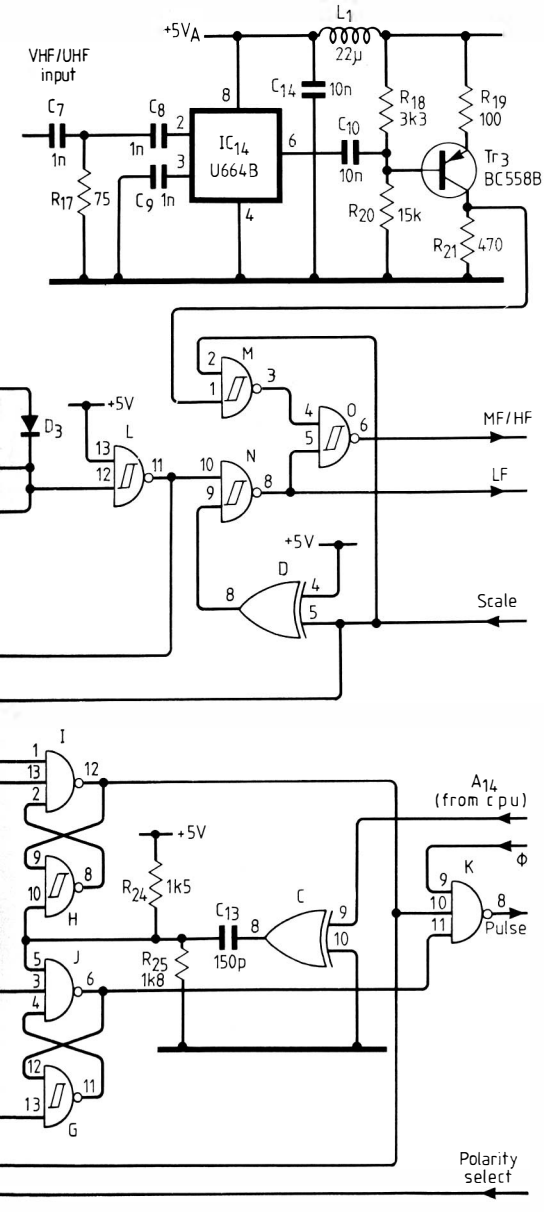


Fig.6. Input stages and prescaler. For good temperature tracking, the two fets should be glued together, flat to flat, and should be matched for gate-to-source voltage. To protect the microprocessor S_1 and S_2 should be break-before-make types.



Fets for the input amplifier should have well-matched gate-to-source voltages at a drain current of 5mA. Of eight devices tested by the author using the circuit above, two pairs were found to have gate-to-source voltages within 20mV of one another.

The input amplifier is best built in one long run down the printed circuit board to ensure good isolation between output and input.



The pulse processor receives this signal as its input and operates as follows. The cpu selects the pulse function by making IC_{7b} Q_0 and Q_2 high. It resets the two bistables formed from gates G, H, I, and J by pulsing A_{14} high for a moment, the falling edge providing the reset. The output of gate I is then low, while that of J is high. Output I blocks gates F and K, thereby hindering the clock from triggering the counter. Let us now assume that that the polarity select line to gate A is high. Gates A and B then work as two inverters and when the positive edge of an input pulse occurs at gate A it is passed unchanged to gate E, which generates a short pulse to alter the state of the H-I bistable, enabling K and allowing the clock oscillator through to the counter. This condition is maintained until the negative edge of the input pulse arrives, is inverted by A, and with the help of F, which was enabled when the H-I bistable flipped over, changes the state of the G-J latch so that the gate K is once more disabled, cutting off the 5MHz pulse train to the counter. The cpu senses this behaviour and calculates the pulse duration by multiplying the count by 200ns. Inputs of the opposite polarity are measured if the polarity select line to gate A is low, since this will then be non-inverting.

HF INPUT

The hf input is applied straight to the input of the U664B prescaler. This is a moderately-priced Telefunken device which has excellent sensitivity all the way up through the vhf and uhf bands to 1GHz. Take care not to confuse this with its stable-mate the U664BS, which is designed for self-oscillation in the absence of an input signal. The outputs of these devices are at ecl levels and so Tr_3 is used to translate them to ttl levels.

The cpu selects the source for the counter via the nand routing network, this being steered by Q_0, Q_1, and Q_2 of IC_{7b}, which is itself controlled by address lines A_{13} and A_{15}. The amplifiers are supplied from their own voltage regulators to prevent noise feed-through from the main digital circuitry causing problems.

THE PROGRAM

The program is very nearly 4K in length. In view of the very wide range of high-precision numbers that the arithmetical routines must handle, floating-point format was decided upon for internal representation. These have a 24-bit mantissa and an eight-bit exponent with no offset. Because of the small number of operations to be performed to calculate any result, this format allows working to an accuracy of seven significant digits, which is just about adequate. All calculations are properly rounded up or down to maintain precision.

For the sum and difference modes, the program slips into bcd arithmetic. This not only demands much less programming effort, it also safeguards against showing these results with spurious accuracy. Suppose that the signal being measured is 10MHz and that the stored frequency is 9.99998MHz. It is no good showing the difference between them as 20.000Hz to five not-very-significant-figures. Unlike floating-point arithmetic, bcd arithmetic works with a fixed decimal point, its position being determined by whichever of the two frequencies is the larger, and the zeroes which would then appear in the above example are displayed in the readout, which becomes 0.00002MHz.

To prevent the program getting stuck while trying to synchronize to an input which is suddenly removed half-way through a measuring cycle, a time-out limit has been included to abort the measurement after about one second.

DISPLAY

Great care has been taken to ensure smooth multiplexing of the display. The microprocessor is responsible for doing this whilst attending to a great many other chores in each cycle. Even quite small fluctuations in the rate of multiplexing cause surprisingly large amounts of flicker in the display, but the present version of the program has overcome this problem entirely.

Display format has been made as logical as possible within the limitations imposed by the 74247 decoder. When switched to a non-existent function, the program produces a display with only the decimal points lit. This is also the case with vlf and pulse functions until the first edge to which the program triggers occurs, whereupon zeroes are displayed in the frequency and pulse modes, and zeroes with decimal points in the period mode. This remains until the first meaningful result is obtained. Much the same system is used when overflow occurs in the ratio mode, which results if the divisor is zero. Ratios are shown as five-figure numbers followed by an exponent, the exponent being expressed in the multiple-of-three convention. An annoying shortcoming of the 74247 is its inability to show a minus sign and so, for want of better, the last decimal point of the readout has been pressed into service. A reading of 12.345 6. is therefore to be interpreted as meaning 12.345×10^{-6}. Use has been made of the same device in the frequency difference mode when the stored value is greater than the input frequency.

One point to note is that the clock function only measures time while the gate of Tr_1 is held higher than that of Tr_2 (use R_{26} to apply a bias if wished), and that the clock works cumulatively. It may be zeroed by resetting the instrument or choosing another function.

● A further article describing a method of interfacing the instrument with PCs is in preparation. keith@snook.eu

Software in eprom for this design is available from the author; readers should ask their bank to raise a cheque for 150 Danish kroner (about £13). If there is sufficient interest, he will also offer a set of three printed circuit boards. Enquiries can be forwarded to him via the editorial office: mark your covering envelope "Frequency counter".

High-resolution frequency counter

High-stability clock, display protection and PC interface for the instrument described in the January 1988 issue

STEPHEN THEOBALD

The clock oscillator described in the January issue was a standard design built around a single 74HC-type inverter. This circuit is reliable and simple, but its performance is far from ideal. HC type c-mos gates have input currents and capacitances which are both heavily voltage dependent when they are used as amplifiers with DC feedback, and the output from such an oscillator is easily seen to be a far cry from a pure sinewave. These factors, together with temperature-dependent propagation delay and the temperature coefficient of the external capacitors, make it seem unlikely that this simple oscillator will show good frequency stability. In an attempt to confirm this supposition, a few tests were carried out on similar oscillators built using c-mos gates and small ceramic capacitors. One oscillator showed a remarkably small temperature dependence of less than −2 Hz/°C in the region 10−40°C, while another was poorer at about −7 Hz/°C. A commercial oven-stabilized oscillator was used to drive the frequency counter while these tests were performed.

These figures are probably not particularly accurate, since the equipment used for temperature cycling was rather primitive (ice-cubes and large transistors mounted on a heatsink inside an insulated box) and the time-consuming nature of these experiments prevented further investigation. The results do, however, indicate that if the full seven-figure accuracy of the counter is to be realised, then some better clock source is required, bearing in mind too that the temperature rise inside the case of the counter can easily exceed 20°C even with good ventilation. It is at any rate wise to position the oscillator components as far from the power supply as possible, since this is the main heat source within the instrument.

When the clock-inhibit switch, S_3, is closed, the internal oscillator is disabled and the instrument can then be operated with an external clock source when greater precision is needed. An ovened crystal oscillator providing tight temperature control and using flow-temperature-coefficient capacitors should be capable of maintaining an accuracy of about 1 part in 10^6 or better. Suitable units can be built, which is difficult, or bought, which is expensive. They are also unfortunately rather bulky and consume a fair amount of power because the oscillator runs at an elevated temperature; 70° to 80°C seems to be typical. Long-term ageing effects will also mean that periodic, though infrequent, calibration will be necessary.

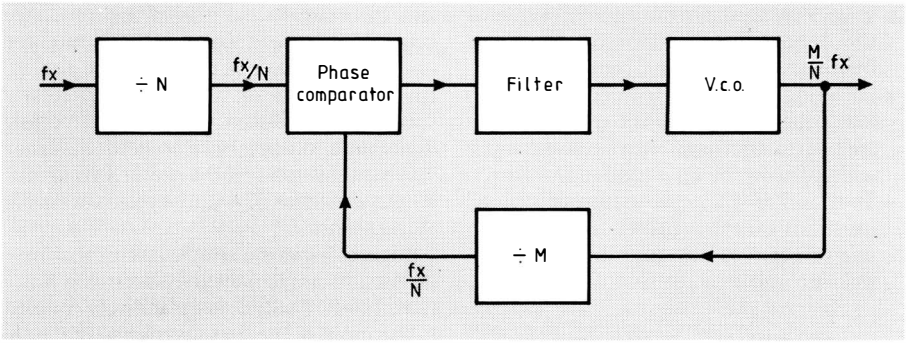


Fig.1. Phase-locked-loop clock source to provide 5MHz from unhelpful carrier frequencies.

Where the ultimate in accuracy is required, a clock source derived from, and phase-locked to one of the highly stable LF transmitters such as Rugby, Droitwich and Kalundborg (Denmark), provides the very best solution. These stations transmit frequencies accurate to better than 1 part in 10^{10}, which is at least two orders of magnitude more precise than we in fact need. Designs for suitable receivers have appeared in the pages of *Wireless World* and other electronic journals from time to time,[1,2] and these can supply the reference input to a phase-locked frequency multiplier. Integrated circuit manufacturers have taken most of the problems out of the design of phase-locked loops these days, and data sheets usually contain helpful advice, or even complete design examples.[3]

The transmitters mentioned above have slightly unfortunate carrier frequencies which cannot simply be multiplied up to exactly 5 MHz. This means that the carrier must first be divided down to some convenient frequency, say 5 or 10 kHz, before being multiplied. A block diagram of such a system is shown in **Fig. 1**, and it is clear that large division ratios may be required in the feedback path.

The microprocessor-based frequency counter can, however, simplify the design of the PLL stages owing to its in-built computing capability. Although designed to work with a 5 MHz clock, the instrument can be arranged to work with any clock frequency in the range of 4 to 6 MHz, provided that a few minor changes are made in the contents of the rom. When these changes are made, the input divider in Fig. 1 can be omitted, and only relatively small divisors appear in the feedback path. The changes necessary are the writing of a four-byte floating-point number and its reciprocal at certain addresses, and the resetting of a flag. The microprocessor uses the numbers to adjust the results obtained on the assumption of a 5 MHz clock if the flag is reset. The process is perhaps best explained by the use of an example. Consider that we wish to use the Rugby transmitter broadcasting on 60 kHz.

We first choose a convenient multiplier which will bring this frequency up to somewhere in the vicinity of 5 MHz. The figure of 80 is decided upon, which will increase the LF signal to 4.8 MHz.

$$X = 5/4.8 = 1.0416667 \text{ and } Y = 4.8/5 = 0.96$$

These are converted into binary form, giving

$$X = 0.1000\,01010101\,0101\,0101\,0101 \times 2^1$$

$$Y = 0.1111\,1010\,1110\,0001\,0100\,1000 \times 2^0$$

Note at this point that all working must be done to an accuracy of 8 decimal digits and 25 bits, the 25th bit being rounded up or down as necessary, and that X and Y both have a fractional part, F, such that $0.5 \leq F < 1$. X has been post-multiplied by 2 in the example to achieve this.

These are now converted into hexadecimal, four-byte values, the fourth byte being the power to which 2 has been raised. Thus

X′ = 85 55 55 01 and Y′ = FA El 48 00

These bytes are now written as a string in reverse order, 00 48 El FA 01 55 55 85, and stored in the rom at addresses OFFO to OFF7, in the given order. Doing this will also automatically reset the necessary flag. The bytes at these addresses are all FF originally.

Anyone considering using this technique should note the following points:

- The frequency decided upon must not lie below 2.5 MHz or above 6 MHz. Frequencies under 5 MHz will slow the instrument down, and 4 MHz is probably a usable lower limit.
- The 5 MHz crystal in the internal oscillator should be replaced by one having the same frequency as the external source.
- Lowering the clock frequency will also lower the maximum PRR which may be applied to the input in the event mode without pulses being missed.
- The counter will perform normally on all functions except the CLOCK function. This one function is interrupt driven and demands a 5 MHz oscillator.

PROTECTING THE DISPLAY

The multiplexed display of the frequency counter is run at a much higher current than the devices can stand during static operation. Each segment is driven with a current in excess of 30mA, but this is quite safe provided that each digit is only turned on for one eighth of the time. However, if the multiplexing action should cease for any appreciable length of time, there is a great danger that some of the segments may be damaged, resulting in reduced brilliance from that part of the display henceforth. If the worst comes to the worst, one of the digits may even burn out completely.

The instrument has been designed to ensure that this cannot happen during normal operation, but there is the ever-present risk of disturbances on the mains upsetting things. Noise spikes, for example, can quite easily propagate through the power supply and either cause the microprocessor to malfunction for a short time or cause incorrect data to appear on the busses. If the microprocessor, by this or any other means, gets out of step with its program, it is highly probable that the scanning of the display will stop with one of the digits turned on. The instrument has not shown itself to be unduly prone to such problems, but some means of protecting the display is obviously desirable.

A simple LC filter on the mains input will be able to reduce the effect of spikes and of course can be recommended, but this does not offer complete protection, nor will it guard against short-term drop-outs which can occur from time to time.

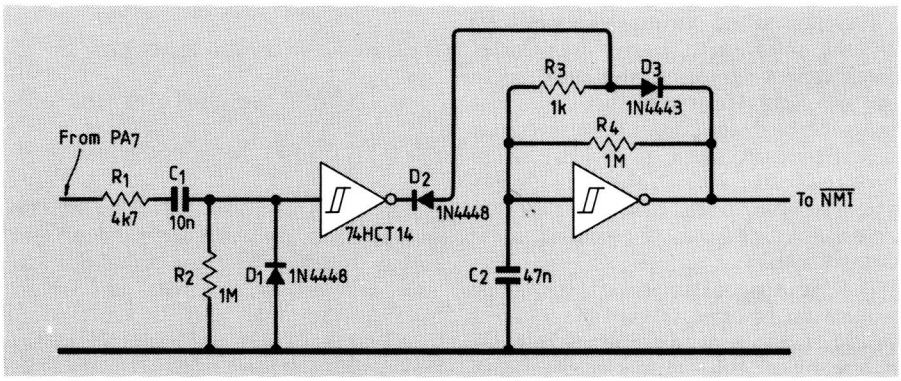
A better solution, one which guarantees full protection against all malfunctions caused by external influences, is a watch-dog circuit. This is standard equipment in a great many microprocessor designs nowadays, and is simple to implement. It works by monitoring some recurrent behaviour of the system and resetting the system to some well-defined state if the expected behaviour fails to occur. In the case of the frequency counter, it is clear that we should monitor one of the anode drive circuits of the display, the PA7 output of IC_4 being particularly suitable. This output does not drive any digit, but is multiplexed along with the other seven output lines all the same. A rectangular wave with an 87% duty-cycle is available at this point.

Figure 2 shows a simple circuit that may be used to implement the watch-dog function. The PA7 output is capacitively coupled to the input of the first inverter so that only dynamic multiplexing can affect the inverter. Diode D_1 restores the DC level of the waveform. So long as multiplexing takes place, the output of the first inverter will also be a square wave with the same frequency as the input. Each time that the inverter's output goes low it will discharge C_2 via D_2, maintaining the second inverter's input at a low level and its output high. This point is connected to the NMI input of the Z80B, which is active-low, so that in this condition the processor is not interrupted.

However, if the multiplexing should fail at any time, resulting in a static output from PA7, the input of the first inverter rapidly falls low, its output goes high, and D_2 disconnects it from the following circuitry. The second inverter is now able to function as a low-frequency oscillator, providing short negative pulses at its output which interrupt the microprocessor. The microprocessor will always respond to these interrupts, jumping to the routine starting at address 0066. This routine first of all blanks the display completely and then pulses the PA7 output rapidly up and down for about 200 ms. This then puts an end to the interrupting signals and the routine finishes by restarting the microprocessor at address 0000, which is the normal starting point of the program. Even prolonged and very irregular disturbances of the system will not prevent the instrument from eventually starting up normally again because the watch-dog will go on oscillating and interrupting the microprocessor until multiplexing is re-established. The chances of an AC signal appearing on the PA7 line if the program is not running correctly are very remote, perhaps almost impossible.

The inverters for the circuit could be the two spare ones in IC_{12}. This circuit was specified as a 74LS14 in the January article, but may be replaced by a 74HCT14. The NMI input was also shown connected to the +5V rail, but must of course be disconnected from this if the watch-dog circuit is to be incorporated in the instrument.

Fig.2. Watchdog circuit avoids damage to the display in case of multiplex failure.



INTERFACING THE COUNTER TO A COMPUTER

It is quite some time ago now that design work first started on the frequency counter, and no thought was then given to equipping the instrument with a computer interface. It should be possible to do this, however, providing not only for the transmission of measurements from the counter to a computer or data-logger, but also allowing the counter to be completely remote-controlled. Note that the interface design which is presented here has not been built, but it is straightforward and should work without difficulty. The design will not prove suitable for all computers, but enough information is given to allow the reader to adapt it to his/her needs.

The well-known 8255 24-bit I/O port has been chosen for the immediate interface to the computer. This device can operate in three main modes, mode 0 being chosen for this application. In this mode ports A and B operate as two independent 8-bit input or output ports, while port C operates as two 4-bit ports. In the diagram, **Fig. 3**, port A is output while B and C are inputs. It is not intended to deal with the detailed working and control of the 8255 here, and would-be users are recommended to refer to Intel's data sheet[4].

Let us consider port A first; this controls the functioning of the frequency counter. Bits A6 and A7 are used to switch the two relays governing gain and coupling in the preamplifier. Bits A4 and A3, clocked through the latch IC_2, replace switch $S2_a$/$S2_b$ in the diagram in the January issue, while the lowest three bits of the port and the associated circuits IC_3 and IC_4, an 8-input multiplexer and a 3-to-8 line decoder respectively, replace $S1_a$/$S1_b$. All of these switches should be removed from the counter entirely. The accompanying table in Fig. 3 shows which control word the computer should write to port A to obtain any particular function.

Synchronization of data transfers between two asynchronous systems is nearly always a difficult matter, as witnessed here by the presence of IC_5 and the use of bit 6 of port A. When a new function is to be selected in the

frequency counter, the appropriate control word should be sent to bits A0 – A4 with A5 low. It should then be sent again about 1 ms later with A5 high, and once again 1 ms later with A5 low. The clock signal for IC_5, a dual D-type flip-flop, is the same signal used to clock the display output ports. This clock occurs about every 820 microseconds. The net effect of the above procedure and the flip-flop delay line is to ensure that data is stable on A0-A4 before IC_2 is clocked from the pulse out of A5, at a time when the display is being refreshed; i.e. when the new data is not being examined by the counter. This allows the counter to change cleanly from one state to another without passing through any undesired states. If this procedure is not followed, it would be possible to overwrite the stored frequency in the counter's memory by accident.

If the counter is switched from one function to another, it reacts rapidly in most cases to perform the new function. This is not always the case however. In particular, moving horizontally in the upper four rows of the table of Fig. 3, it may take up to a second before the counter reacts, plus another second before a reliable reading becomes available. Moving vertically between rows 101 and 100 the reaction is very fast, and a valid reading is available within 10 ms. The same applies for vertical movement between rows 011 and 010, and for transitions in any direction in the lowest two rows.

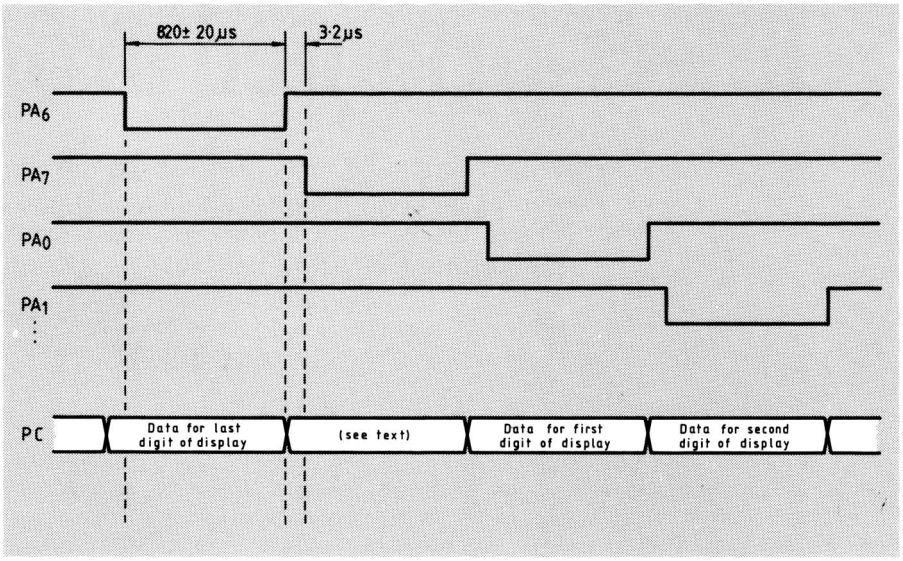


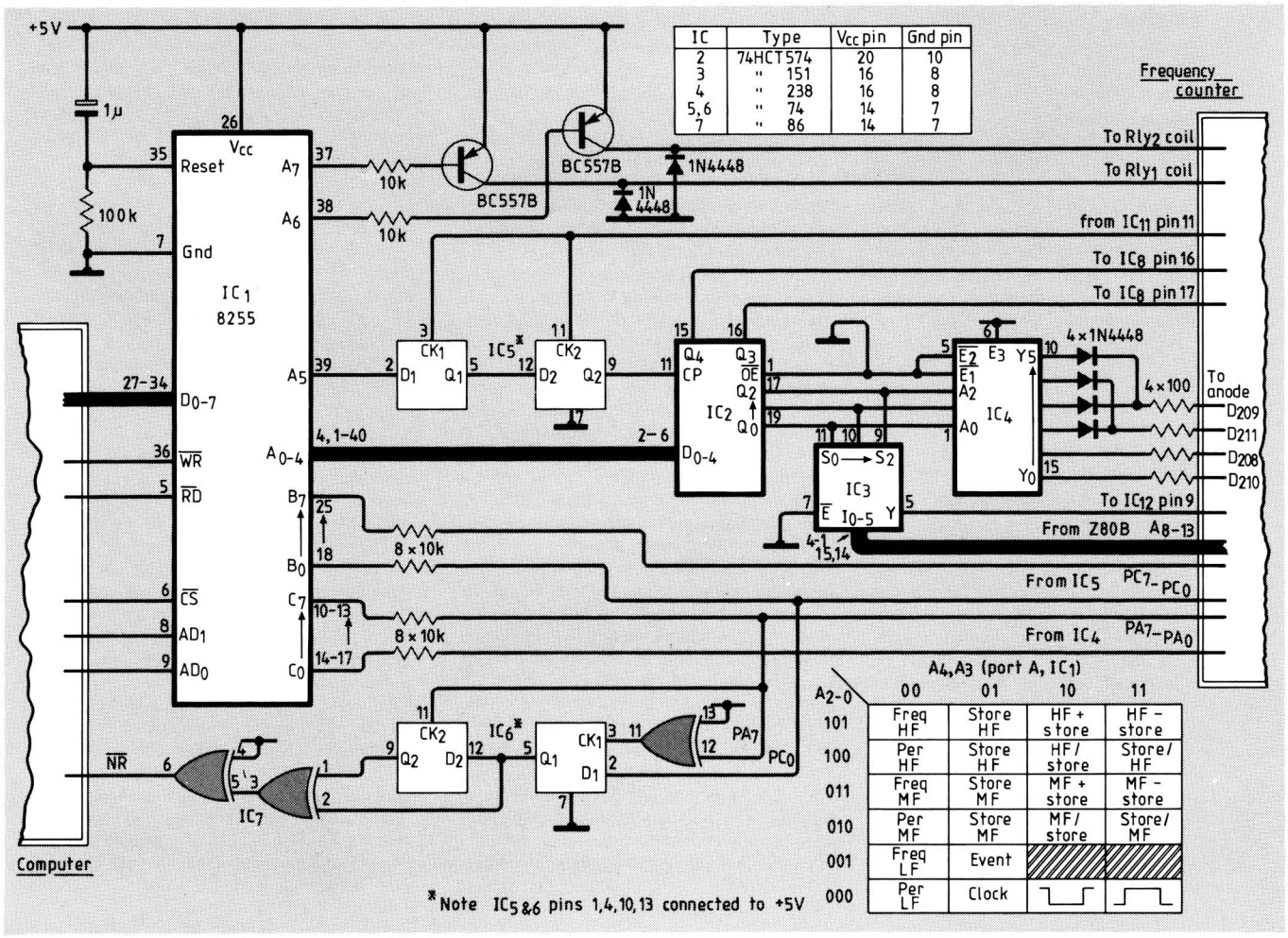
Fig.4. Timing of data on display output ports.

Ports B and C transmit the results from the display out onto the computer bus. Data received by the computer is obviously in the same format as that used by the display, and if the data must be mathematically manipulated, the computer will have to be programmed to convert the data to some more amenable form.

There are a couple of points worth noting in connection with the display. The first is illustrated in **Fig. 4**, which shows the timing of data on ports PA and PC in the counter. We see that the anode drives of the separate digits do not overlap, and that data is valid on PC 3.2 microseconds before the associated line of PA goes low. The second point concerns the data on PC. The lower nibble

continued on page 379

Fig.3. Interface between counter and computer.



IC	Type	Vcc pin	Gnd pin
2	74HCT574	20	10
3	" 151	16	8
4	" 238	16	8
5,6	" 74	14	7
7	" 86	14	7

A2-0	A4,A3 (port A, IC1) 00	01	10	11
101	Freq HF	Store HF	HF + store	HF - store
100	Per HF	Store HF	HF/ store	Store/ HF
011	Freq MF	Store MF	MF + store	MF - store
010	Per MF	Store MF	MF/ store	Store/ MF
001	Freq LF	Event		
000	Per LF	Clock		

contains the BCD value of the figure in the result being refreshed. The LSB of the higher nibble is low only if that digit has its decimal point lit. The remaining three bits indicate the range, which the computer must interpret as the exponent of ten by which the result must be multiplied. The highest bit will therefore mean a multipler of 10^6 (MHz) or 10^{-9} (ns), according to the function being measured. On the 1F and pulse ranges this bit indicates mHz and s, giving multipliers of 10^{-3} and unity.

Let us now look at the $\overline{NR}$ output. This is an extra output from the interface to the computer and goes low each time the counter has just made a new reading. The circuit consisting of $IC_{6,7}$ is basically a two-bit delay line. The input to the delay line is the LSB of port PC and the flip-flops are clocked by PA7, the anode drive for the last, non-existent digit. The circuit operates on the following principle.

Throughout any one complete measuring cycle the sequence of data put out on PC as PA sends its lines low in turn remains constant. When the counter obtains a new result at the end of a measurement, the new data sequence on PC is dependent upon the actual result obtained and can not be predicted. PA7 drives no digit however, and the microprocessor of the counter is able to guarantee that the data on the LSB of PC, when PA7 is active, will change state from one measuring cycle to the next. It is this transition that we detect.

The negative edge of PA7 clocks PC0 into the first flip-flop, and the positive edge copies it into the second. In the interval between these two edges then, the first flip-flop holds the new value of PC0, while the second holds the value from the previous refresh cycle of the display which occurred about 6.5 ms ago. The ex-Or gate connected to the output of the flip-flops can therefore compare the two samples and will give a high output if they are different. This is inverted by the next gate to produce $\overline{NR}$. $\overline{NR}$ is therefore about 820 microseconds in length, but this may be increased in steps of the same length by clocking the second of the flip-flops from the anode drive lines PA0, PA1, etc. The behaviour of the $\overline{NR}$ line is not to be counted upon in the EVENT and CLOCK modes of operation. The results here change too rapidly for the $\overline{NR}$ signal to have any meaning. Once the computer receives a pulse on $\overline{NR}$, perhaps on its interrupt pin, it may obtain the new result from the display in the course of the next 6.5 ms approximately, by reading each digit in turn via the interface.

Another solution to the interface problem which appears attractive is the use of one of the many single-chip microcontrollers that are on the market today. Some of these have a large number of I/O pins, 20 or more, allowing them to replace the 8255 in the above circuit. The part of the circuit built around IC_2, IC_3, IC_4 and IC_5 can be used unchanged, although it ought to be possible to do away with IC_5. To save on I/O pins, only one line of PA, say PA7, need be monitored. This may be used to synchronize data transfers into the microcontroller as each line of this port is low for a period of 800 – 840 microseconds in turn. The controller could also provide a serial rather than parallel link to the computer, allowing the counter to be placed very remotely from the computer. One controller which looks promising for this application is the 8400 from Phillips. This has 19 I/O pins and an in-built serial interface intended for I^2C bus working. It is available in a 28-pin piggy-back version which allows an eprom to be mounted directly on top of the controller. Although its working is rather leisurely when run with a clock frequency of 5 MHz, its instruction set, being very similar to that of the 8035/39 series of microprocessors, is also suitable for mathematics so that it could perform the necessary data conversion to binary, for example, if this was required. keith@snook.eu

References

1. J.M. Osborne. "High-standard, low frequency source". *Wireless World.* January 1973, pp. 20-21.
2. G. Jackson. "C-mos 60 kHz receiver". Circuit Ideas. *Wireless World.* May 1980, p. 78.
3. Phillips Limited. Integrated Circuits. Book IC06, 1988. High Speed Cmos. pp 755-782 and 1041-1070. www.keith-snook.info
4. Intel Corporation. Data sheet AFN-00744C.