DESIGNING A PC-BASED SOLUTION TO FREQUENCY GRADING

By

***G. N.ONOH & **E. C. OKAFOR**
**Department of Electrical and Electronic Engineering*
***Department of Computer Engineering*
Enugu State University of Science and Technology, Enugu

ABSTRACT

Frequency-to-code converters (FCC) appear in a number of applications, hence many techniques are available and in use in the design of existing instruments. They vary however in bandwidth and accuracy of measurements.

A PC-based FCC is presented here based on the technique of 'frequency grading'. The system incorporates a counter that slows down its pace automatically as the interval between the last pulse and the next one exceeds some predefined value. The PC-based solution -

- *effectively eliminates the restrictions ROM-based designs introduce at the high- and low-frequency ends of the useful frequency spectrum.*
- *directly delivers the results of measurements to other PC-based control and digital signal processing applications.*

Introduction

Many electronic instruments feature some form of frequency-to-code converter that essentially digitizes waveforms at varying frequencies (Kirianaki et al, [1] & [2]; Showden et al, [3]; Saha and Mazumder, [4]. Probably the simplest of these employ frequency counting technique to deliver a binary sequence proportional to the input frequency. Available digital frequency meters employ this or other techniques and also form the basic building block of equipment like heart-beat meter and others that measure pulse rate or frequency (Kirianaki et al, [1] & [5]; Saha and Mazumder, [6]. Wide range of measurement and resolution better than 1 beat/min is possible with frequency grading (Ottonello et al, [7].

The technique here does not just count 'cycles per second' but delivers a frequency-to-code converter for frequencies up to the ultra-sonic region. Based on a converter too, this system automatically slows down the pace of the clock to the counter as the time between the last pulse and the next one goes above a certain predetermined value. This allows for broadband operation (10Hz

- 100KHz) while keeping accuracy better than 1-percent. A microcomputer interface delivers the result of this operation to a PC for further processing and display.

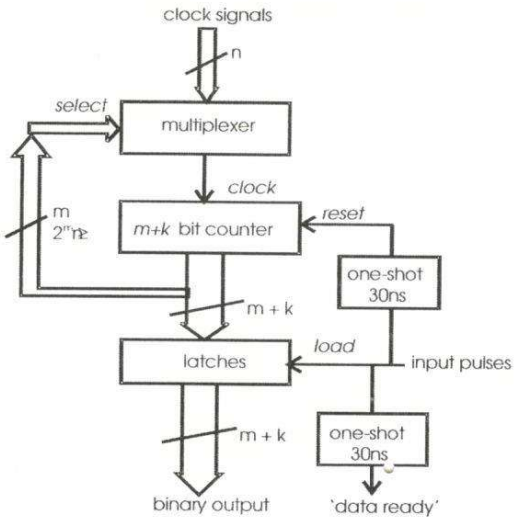

Figure 1: Operation System

Operation

Assume that the maximum possible value of the input frequency is $f_{in(max)}$. In operation, (Figure 1) this system is initially set up to receive pulses at intervals of $1/f_{in(max)}$. A counter is reset by the initial (*start*) pulse from the input signal. Its m-bit output (all logic 0) selects the highest of the n clock signals f_n available from the multiplexer to drive the counter. If *1-percent* accuracy is desired then $f_n = 100f_{in(max)}$.

Again, if no (*stop*) pulse arrives at the input after 2^K clock cycles have passed on the counter, the m outputs select another slower clock frequency for the counter. This drops the pace of counting for the next 2^K clock cycles and even further unless a (*stop*) pulse is received. In practice, the multiplexer may be fed from the output of an n-bit counter to produce clocks that are progressively halved.

The arrival of the stop pulse causes the latch to hold onto the current output of the counter. It is also delayed for a few nanoseconds to reset the counter and may be used as an interrupt to a computer system signaling that valid data is available.

Forcing a reset on the counter also initiates a new cycle with the multiplexer linking the counter's clock input again to the maximum frequency, f_n.

The outputs of the latch may well be used to address a PROM that has been carefully programmed as a 1/T look-up table (Koval and Yurish, [8]; Sergey et al, [9].

This option forces great restrictions on the useful range and accuracy of the system.

Control & Processing

The $(m+k)$-bit code read from the latch is proportional to the frequency of the input signal. However, this code is composite since it is derived from counting performed at varying rates. The control software must decompose this code and

work out the frequency by weighing in the different rates used by the counter.

Suppose we represent the n possible clock periods as t_n, t_{n-1}, t_{n-2}, ... where $t_n = 1/f_n$. Then assume that the binary code delivered by the counter and available at the PC interface is N. One can represent the period of the corresponding input signal T_N simply as:

$$T_N = T_{N1} + T_{N2} + T_{N3} + ...T_{Nn} \quad [1]$$

where $T_{Ni} = c_i t_{n-i}$, $1 \le c_i \le 2^K$, $1 \le i \le n$

Recall that the frequencies applied to the multiplexer are progressively halved, thus

$$t_{n-i} = 2^i t_n \quad [2]$$

and

$$T_{Ni} = 2^i t_n c_i \quad [3]$$

Now invoking [1] and [3]

$$T_N = \sum_{i=0}^{n-1} 2^i t_n c_i \quad [4]$$

In practice, $c_i = 2^K$ for at least $(n-1)$ times and assumes a fraction of this value just once. We can work out how many times c_i is equal to 2^K by evaluating $\delta = int(N/2^K)$. The fraction of N left (that is not up to 2^K) is given by $\mu = (N - \delta 2^K)$

Thus

$$T_N = \sum_{i=0}^{\delta-1} 2^i 2^K t_n + \mu 2^\delta t_n$$

$$= t_n\left[2^K \sum_{i=0}^{\delta-1}(2^i) + \mu 2^\delta\right] \quad [5]$$

where t_n = period of the maximum count frequency in seconds.

Figure 2 shows the algorithm that implements this. The procedure (coded in assembly language, C -language or any other) may be designed as a subroutine. It simply polls the interface; then generates and displays the frequency in Hertz. The algorithm initially checks that the input frequency $f_{in(max)}$ is not greater than 100KHz. One this is certain, it quickly evaluates equation [5]. Note that it avoids the use of any look-up table.

PC Interface

The architecture of this converter is shown in Figure 3. Sixteen clock signals are used for this example. These are generated using a 16-bit counter to divide down f. To maintain 1-percent accuracy within its entire range, the clock frequency required at any point f_{n-i} should be greater or equal to $100 \times f_{in}$.

As an example, if the system is to measure frequencies up to 100KHz at one-percent accuracy, then $f_{n(max)} = f/2 \ge 100$ x 100KHz = 10MHz.

Assuming $f_{n(max)}$ = 12.8MHz, then frequencies (f_{n-i}) divided down to 390.625Hz will be applied to the multiplexer.

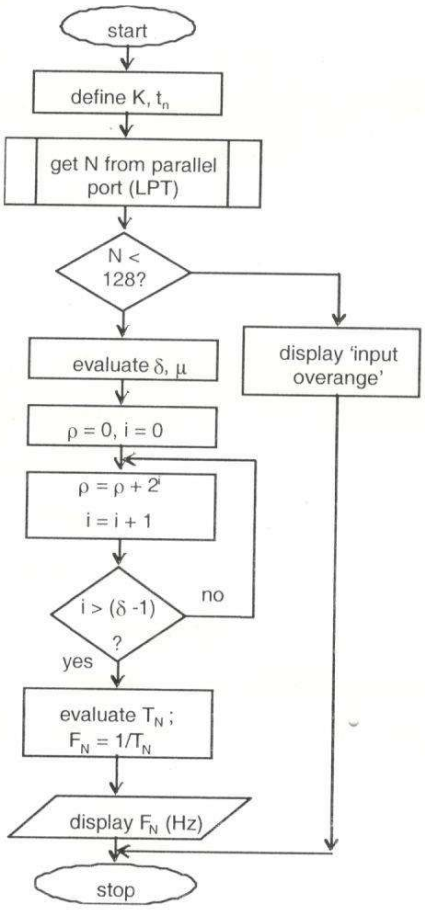

Figure 2: Logarithm Implementation

In other words, if A is the relative accuracy, f_{n-i} the frequency required to achieve A and T_{in} ($1/f_{in}$) the inter-pulse period of the input signal, then the absolute time resolution ΔT_{n-i} required when the instantaneous frequency is f_{in} is given by (Koval and Yurish, [8], Sergey et al, [10].

$$f_{n-i} = \frac{1}{\Delta T_{n-i}} = \frac{f_{in}}{A} = \frac{1}{AT_{in}} \qquad [6]$$

Cleared by an early pulse, the counter starts at this requency and slows down to the next lower frequency after every 256clock pulses provided no other (stop) pulse arrives. This continues to other frequencies depending on how long it takes for the

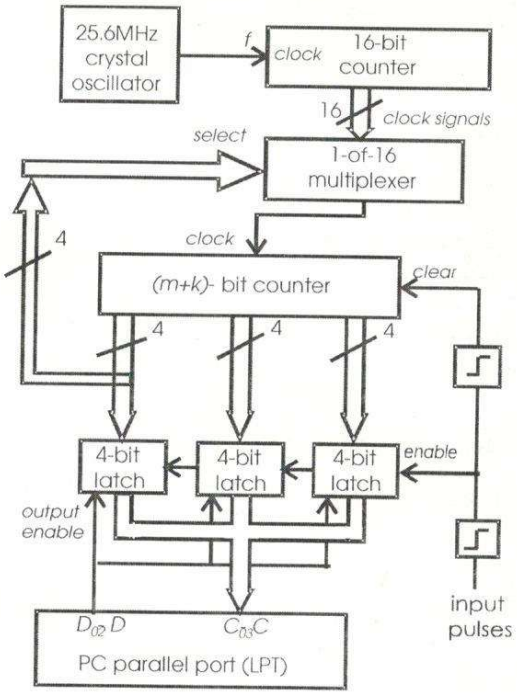

Figure 3: Converter's Architecture

arrival of the next (stop) pulse. When this happens, the 12-bit counter output is automatically stored in three 4-bit latches linked to the parallel port (LPT1) of a PC that has been configured as a standard parallel port (SPP) - the port operates in both compatible (8-bit output) and nibble (4-bit input) modes. Each nibble can be read from the status register of the port - probably at 0379H - with the appropriate enabling sequence sent to the data register at 0378H (Okafor, [11]; Mueller, [12].

Input signal period, t_{in} (μs)	12-bit count	Count frequency for the ranges f_{n-1} (KHz)	Output frequency (Hz)
10	128		100000
		12800	
20	256		16667
		6400	
60	512		7143
		3200	
140	768		3333
		1600	
300	1024		1613
		800	
620	1280		793.65
		400	
1260	1536		393.7
		200	
2540	1792		196.08
		100	
5100	2048		97.85
		50	
10220	2304		48.75
		25	
20460	2560		24.43
		12.5	
40940	2815		12.21
		6.25	
81900	3072		6.1

The control software subsequently concatenates these three nibbles as N and passes same to the conversion program (Figure 2). Figure 4 details the count/input-frequency relationship for the system of Figure 3.

Conclusion

The FCC presented provides broad-band response at very good accuracy levels. Frequency grading is essentially one of the most reliable techniques employed in the design of meters used in human and animal cardio-tachometry. The PC-based solution provides cost effective means increasing the bandwidth and accuracy even beyond the value in this presentation while allowing use of existing digital signal processing techniques in further analysis of inputs.

References

1) Kirianaki N.V., Yurish S.Y., Shpak N.O. (2003). "Programmable Measurement Converter Based on Advanced Methods for Phase Shift- and Frequency- to - Code Conversion", In Proceedings of IFAC Workshop on Programmable Devices and Systems (PDS' 2003), Ostrava, Czech Republic, 11-13 February, pp. 9-12.

2) Kirianaki Nickolay V, Yurish Sergey Y., Shpak Nestor o., Deynega Vadim P. (2002). "Data Acquisition and Signal Processing for Smart Sensors", John Wiley & Sons, Chichester, UK,.

3) Showden S., Simpson N. A. B., and Walker J. J. (2001), "A Digital System for Recording the Electrical Activity of the Uterus", IOP Journal on Physiological Measurement, Vol. 22, No. 4, pp. 749-753.

4) Saha A. R., Mazumder B. C. (1992). "A digital Phase-Locked Loop for Generating Frequency Discriminating Digital Words", Proceedings of IEEE, Vol. 70, No. 2, pp. 200-201.

5) Kirianaki N.V., Yurish S.Y., Shpak N.O. (2001). "Methods of Dependent Count for Frequency Measurements," Measurement, vol.29, issue 1, January 2001, pp.31-50.

6) Saha A. R., Mazumder B. C. (1991). "A digital Phase-Locked Loop for Generating Frequency Discriminating Codes and Frequency Multiplication", Proceedings of IEEE, Vol. 69, pp. 472-473,

7) Ottonello P., Basano L., Poggi A., Adezato L., Semino S., Ubaldi P., Viviani G. L. (1995). "An instrument for real-time spectral estimation of heart-rate variability signals" Computer Methods and Programs in Biomedicine, Vol. 47 No. 3, pp. 229-236.

8) Koval V.A., Yurish S.Y. (1997). "Precise Frequency Method of Measurement for Integrated Frequency-Output Temperature Sensors", In Proceedings of 3rd International Workshop on Thermal Investigations of ICs and Microstructures (THERMINIC'97), 21-23 September, 1997, Cannes, FRANCE, pp. 11-16.

9) Sergey Y. Yurish and Nikolay V. Kirianaki (2003), "Novel Duty-Cycle - to - Code Conversion Method", In Proceedings of 8th International Workshop on ADC Modelling and Testing (IWADC' 2003), Perugia, ITALY.

10) Sergey Y. Yurish, Nikolay V. Kirianaki and Ramon Pallas-Areny, (2004) "Low-Cost Precision Universal Programmable Frequency-to-Digital Converter", In Proceedings of IFAC Workshop on Programmable Devices and Systems (PDS'2004), Poland, Cracow, 18-19, pp.70-73.

11) Okafor E. C. (2002). 'Digital Devices and Applications' Immaculate Publishers Ltd Enugu Nigeria, pp 283-285, 454-457, 483.

12) Mueller Scott (2004). 'Upgrading and Repairing PCs' QUE Press, USA.