VERSATILE SYSTEM FOR FREQUENCY TYPE QUANTITIES MEASUREMENT

Liviu Breniuc [(1)], Constantin Sărmăşanu [(2)] , Mihai Creţu [(3)], Cristian Győző Haba [(4)]

[(1)] Department of Electrical Measurements, Faculty of Electrical Engineering, Iaşi, 6600 Romania
Phone (40) 278683 Fax (40) 130054 e-mail: lbreniuc@ee.tuiasi.ro
[(2)] Department of Electrical Measurements, Faculty of Electrical Engineering, Iaşi, 6600 Romania
Phone (40) 278683 Fax (40) 130054 e-mail: csarmas@ee.tuiasi.ro
[(3)] Department of Electrical Measurements, Faculty of Electrical Engineering, Iaşi, 6600 Romania
Phone (40) 278683 Fax (40) 130054 e-mail: mcretu@ee.tuiasi.ro
[(4)] Department of Electrical Engineering, Faculty of Electrical Engineering, Iaşi, 6600 Romania
Phone (40) 278683 Fax (40) 130054 e-mail: cghaba@ee.tuiasi.ro

Abstract - *In this paper we present a frequency-meter based on the method of reciprocal frequency-meter. For the implementation of the frequency-meter we have used an FPGA and a microcontroller. We have measured frequencies between 0.05 Hz and 10MHz. The measuring ranges are: 0.05Hz to 0.5Hz, 0.5Hz to 5Hz and 5Hz to 10MHz. The measurement times for the corresponding ranges are: 20 to 10s, 2 to 1s and 0.2 to 0.1s respectively. The base error of the frequency-meter is $2 \cdot 10^{-6}$.*

Keywords – Frequency-meter, FPGA, microcontroller.

1. INTRODUCTION

For high frequency signals, the frequency is measured using a frequency-meter type principle. Low frequency signals need a period-meter instrument to measure the frequency. In order to achieve reduced measurement errors is necessary the use of both principles. A solution that allows a constant error curve for the entire domain is the reciprocal frequency-meter. The block diagram of the reciprocal frequency-meter is shown in Fig.1.

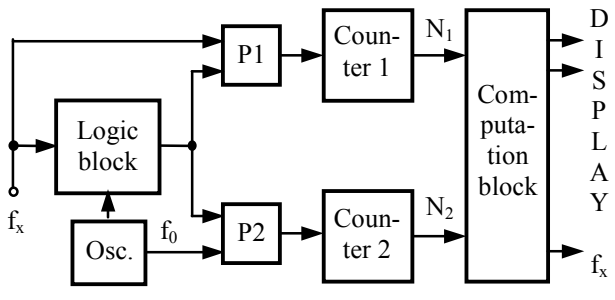

Fig.1 – The diagram of the reciprocal frequency-meter.

The frequency to be measured is f_x whereas f_0 is the reference frequency. The measuring time is approximately constant (about 1 second for example) and is multiple of the measurement signal period. During this time, which is established by the logic block, gates P1 and P2 are open and permit the passage of signals with frequency fx and f_0 respectively. In the same time, counter 1 counts N_1 pulses of T_x period ($T_x=1/f_x$) while counter 2 counts N_2 pulses of T_0 period. In this case T_x is given by (1):

$$T_x = \frac{N_2}{N_1} \cdot T_0 \tag{1}$$

Considering that N_2T_0 is approximately constant, the measurement error is also almost constant (the error on the N_1 number is zero). The expression of the measurement errors is given by (2):

$$\frac{\Delta T_x}{T_x} = \frac{1}{N_2} + \frac{\Delta T_0}{T_0} \tag{2}$$

Because the displaying of the signal period is not suggestive, the computation block will calculate the signal frequency, by inverting the period value as in (3):

$$f_x = \frac{N_1}{N_2} \cdot f_0 \tag{3}$$

The disadvantage of this scheme is the complexity. Practical possibilities of implementation using the 80C552 microcontroller are indicated in [1].

2. IMPLEMENTATION OF THE FREQUENCY-METER

For the implementation of the reciprocal frequency-meter we have used an XS40-010XL Board v1.4 from XESS Corporation. This board contains mainly the following components:

- Xilinx XC4010XL FPGA [2]
- 8051 microcontroller [3]
- 128kB SRAM memory used by the microcontroller as a data and program memory
- EEPROM memory, if existent, used for storing the configuration file (.bit) for the FPGA
- programmable oscillator, etc.

The detailed structure, the external level of interconnection of all components, the programming of the XS40-010XL and the way the XS Board can be connected to a PC is presented in [4].

ISBN 972-98115-4-7 © 2001 IT

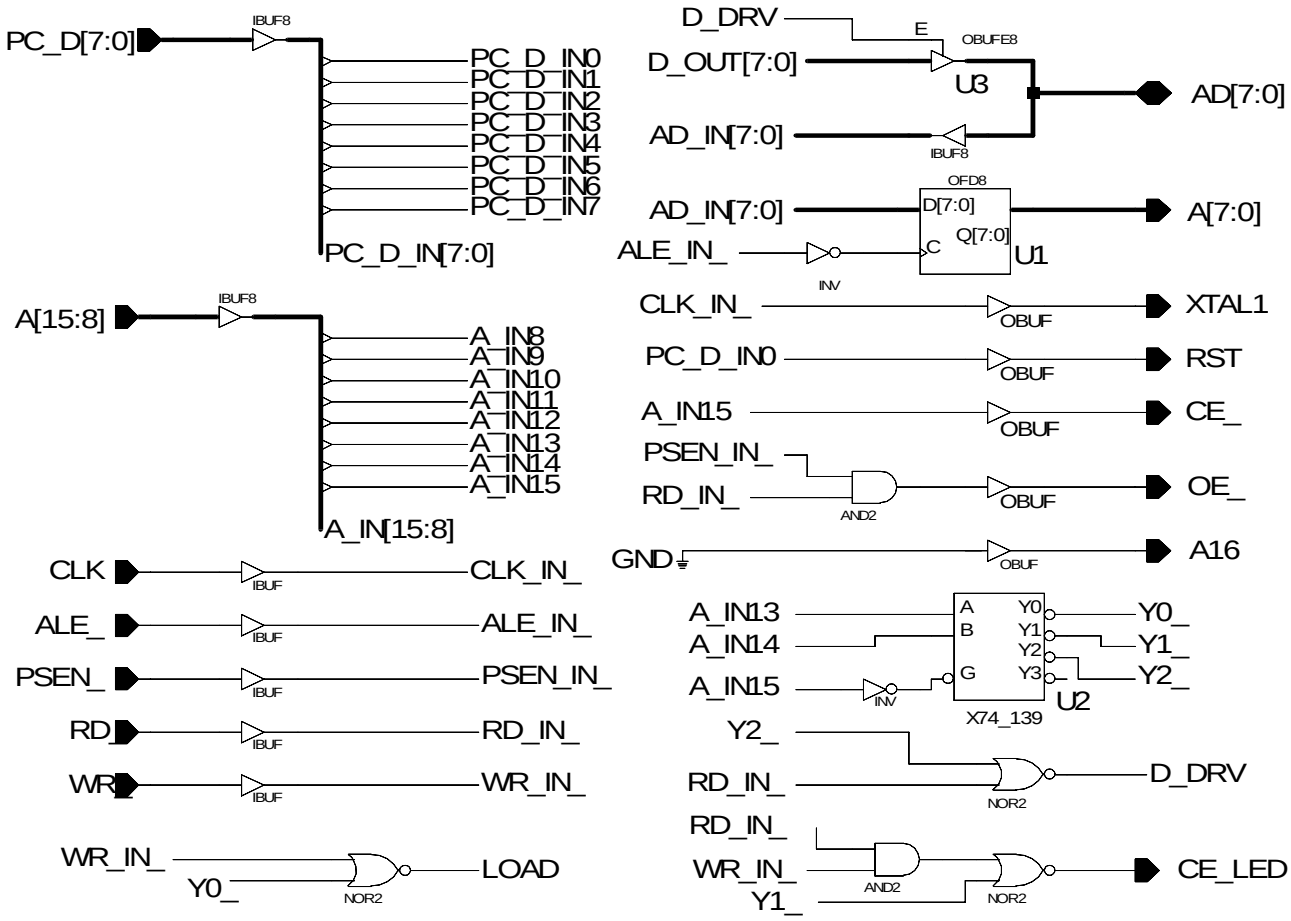

Fig.2 – The interconnections between the components (FPGA, microcontroller, SRAM, LCD display and PC parallel port.

The FPGA and the microcontroller can work independently or together. For the implementation of the frequency-meter we use both components.

The microcontroller controls the frequency-meter operation, determines the frequency range, receives the results of the measurement from the FPGA, computes the measured frequency and display the result on an LCD display. If an error occurs, the microcontroller displays an error message on the LCD display.

The FPGA circuit is used for the implementation of the logic blocks needed for the overall operation of the components (FPGA, microcontroller, SRAM, LCD display, PC parallel port connection) and for the implementation of the measurement function. Other information and programming examples of the FPGA circuit is given in [5,6,7].

In Fig.2 is depicted the schematic of the interconnections between the components (FPGA, microcontroller, SRAM, LCD display, PC parallel port connection). The pins indicated in Fig.2 are the pins of the FPGA circuit and the functions of these pins will be presented in the following.

PC_D[7:0] is the data bus of the PC parallel port which drive the internal signals PC_D_IN7,…, PC_D_IN0. Signal PC_D_IN0 is used to reset the microcontroller (pin RST).

Signals A[15:8] represent the upper byte of the address bus (microcontroller port P2) and drive internal signals A_IN15,…A_IN8.

The pin CLK is the input for the clock generated by the XS Board programmable oscillator. The programmed frequency of the oscillator is 10MHz. This signal is used to generate the internal clock CLK_IN_, which is transmitted also to the microcontroller (pin XTAL1).

From the microcontroler we input signals ALE_, PSEN_, RD_ AND WR_, with corresponding internal signals ALE_IN_, PSEN_IN, RD_IN_ and WR_IN_.

AD[7:0] represent the lines of port P0 of the microcontroller, i.e. the low byte of the address multiplexed with the data byte.

In order to demultiplex the address byte from the data byte we use the circuit U1 (circuit OFD8, 8 D-flip-flop).

A[7:0] represents the address lower byte which connects to the SRAM.

The SRAM has a capacity of 128kB and uses 17 address lines A16-A0. We have allocated for the microcontroller a total of 32kB in the address space 0000H÷7FFFH (A16=0, A_IN15 is connected to the validation pin CE_ of the SRAM memory). The enabling condition for reading the SRAM memory allocated to the microcontroller is obtained by computing the product between PSEN_IN_ and RD_IN_.

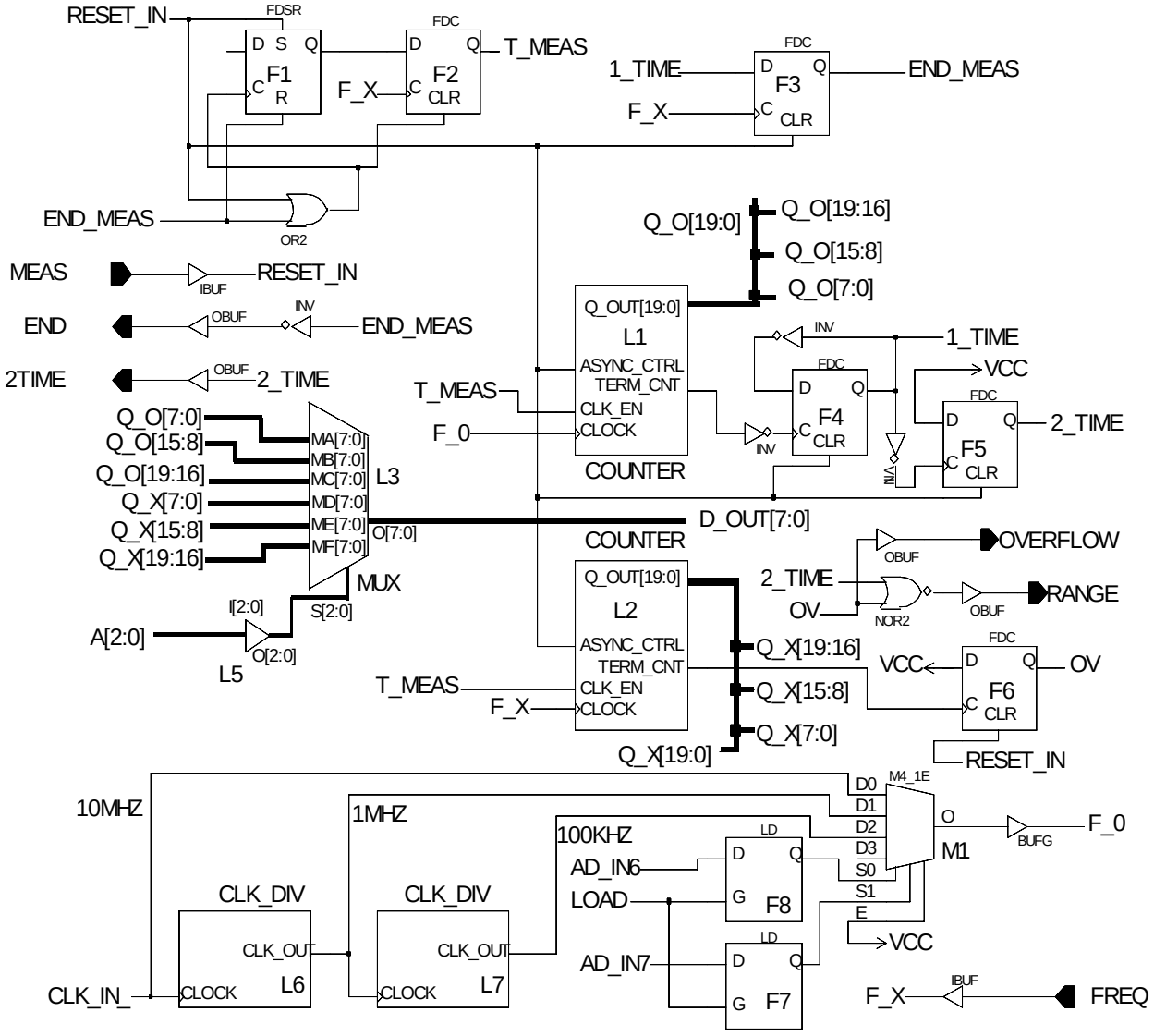

Fig.3 – Schematic of the measurement function.

In this way, this memory space can be seen as both program or data memory. The 8051 microcontroller makes no difference between the external memory and other external I/O devices.

In the schematic are depicted three selection lines Y0_ Y1_ and Y2_ (realized using circuit U2, a 2 to 4 decoder) which are active for the memory address spaces 8000H÷9FFFH, A000H÷B000H and C000H÷DFFFH respectively. By means of signal Y1_, synchronized with signals RD_IN_ and WR_IN_, we have created the enable signal CE_LED (active high) for an LM016 LCD display. Signal Y0_, synchronized with signal WR_IN_, is used to create an active low enable signal LOAD.

With the help of signal Y2_ synchronized with signal RD_IN_ we create the enable signal D_DRV (active high). With this signal will be enabled the output buffer U3 (OBUFE8) who receive input signals D_OUT[7 :0]. By means of this buffer and of course of pins AD[7 :0], the microcontroller can read from the FPGA circuit. In the Fig.3 it is showing the schematic implemented in the FPGA for the effective measurement function. This module is controlled by the microcontroller. The signal of unknown frequency to be measured is applied to the pin FREQ, the internal signal corresponding to this frequency is F_X. The internal reference frequency is F_0 and it is generated using the 10MHz clock signal CLK_IN_. This reference frequency can have the following values: 100kHz, 1MHz and 10MHz. The reference frequencies of 100kHz and 1MHz can be obtained using the dividers L6 and L7 which both have the division factor of 10. Selection of the value for the reference frequency F_0 is realized with the help of multiplexer M1. The control of the selection inputs of this multiplexer is realized by the microcontroller using flip-flops F7 and F8 and signals AD_IN7, AD_IN6 (the first two most significant bits of the data bus) and LOAD. The value of reference frequency F_0 determines the range of the frequency measurement.

The measurement starts when the microcontroller sets the line MEAS to logic HIGH (the corresponding internal signal is RESET_IN). In this way we realize the initialization of the module (setting flip-flop F1, resetting flip-flops F2÷F6, clearing counters L1 and L2). In this initial state, the microcontroller selects the reference frequency of 10 MHz establishing the default measurement range to the subrange of 5Hz÷10MHz. In the next step the microcontroller sets the line MEAS to logic LOW initiating the effective measurement process. The first rising edge of signal F_X toggles the output of flip-flop F2, signal T_MEAS, to logic HIGH. The signal T_MEAS enables now the clock inputs of counters L1 and L2.

The counter L1 and L2 are designed using LOGIBLOX feature of the Xilinx Foundation Series designing software and are configured as 20-bit binary counters. Outputs TERM_CNT of these counters are logic HIGH only when they reach the maximum counting number 1111 1111 1111 1111 1111 (1048575). The clock input of counter L1 receives signal F_0 while clock input of counter L2 receives signal F_X. After a period equal to $1048576 \cdot T_0 \approx 10^6 T_0$ (≈100ms) signal 1_TIME becomes logic HIGH. If during this period of time we overpass the counting capacity of counter L2, i.e. the frequency of the measured signal is greater than 10MHz, at the output TERM_CNT of counter L2 we will have a logic HIGH pulse. This pulse will toggle the signal at pin OVERFLOW to logic HIGH and the signal at pin RANGE to logic LOW. These two pins are connected to the microcontroller in the following way: pin OVERFLOW is connected to pin P1.0 and pin RANGE is connected to pin INT0 (external interrupt). The falling edge of the signal at pin RANGE will issue a microcontroller interrupt. The interrupt subroutine tests the state of signals at pins OVERFLOW and 2TIME and because signal at pin OVERFLOW is logic HIGH, the microcontroller will send to the LCD display an error message equivalent to "Signal exceeds upper limit of the measuring range". After that, it will bring the system to the initial state (MEAS to logic HIGH).

Suppose now that signal is within the measuring range. In this case, with the first rising edge of signal F_X, the output of flip-flop F3, i.e. signal END_MEAS, will become logic HIGH and signal T_MEAS will become logic LOW. When this happens, the counter L1 and L2 are blocked.

Because counter L2 is synchronized with the rising edges of signal F_X, the counting errors for this frequency is zero.

The same signal END_MEAS (negated) will release a microcontroller interrupt request (pin END is connected to input INT1 of the microcontroller). The interrupt subroutine will then read the content of counters L1 and L2. The output of the counters are connected to multiplexer L3 (six 8 bit-input and one 8 bit-output multiplexer).

The data from the FPGA will be read via addresses within C000H÷DFFFH (selection signal Y2_ and signal D_DRV will be activated, Fig.2).

In the next step, the microcontroller will compute the measured frequency and will display the result on the LCD display.

Problems arise at the lower limit of the measuring range. Signal 1_TIME remains logic HIGH for the same period of approximately 100ms. If during this period of time there is no rising edge of signal F_X, then signal 1_TIME will become logic LOW while signal 2_TIME will become logic HIGH. These signal values indicate that the measured signal has the frequency under the lower limit of the measuring range (5Hz). Total time since the beginning of the measuring process is twice 1_TIME (i.e. ≈200ms).

In this case, the signal at pin RANGE toggles to logic LOW issuing a microcontroller interrupt. Because signal 2_TIME is logic HIGH, the microcontroller will initialize the measuring system (MEAS becomes logic HIGH) and it will command the modification of the measurement range. The modification of the measurement range is realized by the selection of another value for the reference frequency F_0 that will be set to 1MHz. Subsequently, the measurement process will be the same to the one described before.

In this new situation, it is not possible to have an OVERFLOW pulse but, it is possible that signal 2_TIME become logic HIGH. This situation can arise only if the maximum measurement time $2 \cdot 10^6 T0$ (≈2 sec) has expired, i.e. the measured frequency is less than 0.5 Hz.

In order to measure frequencies less than 0.5 Hz, we have the third measurement range that is selected by setting the reference frequency F_0 to 100KHz. Within this range, the measurement time is between 10s to 20s and the lowest frequency that can be measured is 0.05Hz.

If the measuring frequency is less than 0.05Hz, the microcontroller will detect the corresponding conditions and will display an error message equivalent to "Signal frequency under the lower limit of the measuring range"

3. EXPERIMENTAL RESULTS AND CONCLUSIONS

For the implementation of the frequency-meter we have used a Xilinx XC4010XL FPGA and an 8051 microcontroller available on the XESS XS40 Board. The frequency-meter can measure frequencies within the range 0.05Hz÷10MHz.

In order to reduce the measurement time we have split the measuring range in three subranges:

- 5Hz÷10MHz (default range selected at the initialization of the system, measurement time is between 100ms ÷200ms)
- 0.5Hz÷5Hz (measurement time is from 1s to 2s)
- 0.05Hz÷0.5Hz (measurement time is from 10s to 20s)

The selection of the measuring range is done automatically by the microcontroller. When the measured signal has frequency outside the measurement range, the frequency-meter display the corresponding error messages.

The measurement error is constant and the same, no matter the value of the measured frequency.

The low frequency subranges can be set to 0.5÷1MHz and 0.05÷100kHz respectively but, this will increase the measurement time for high frequencies in the subranges without reducing measurement errors.

The basic frequency error is $2 \cdot 10^{-6}$ for a stability of the reference frequency of 10^{-6}.

REFERENCES

[1] L. Breniuc and M. Morosan, "Possibility of using the timer-counter system of the 80c552 microcontroller in measurement applications", in *Proc. of SIELMEC'97 World Congress 1997*, Chisinau, Moldavie, 16-18 October, 1997, pp. 185-188.

[2] ***, *The Programmable Logic,* Data Book 1999, Xilinx Corporation, http://www.xilinx.com.

[3] ***, *80C51-Based 8-Bit Microcontrolers*, Data Handbook IC20, 1998, Philips Semiconductors, http://www.semiconductors.philips.com

[4] L. Breniuc, A. Salceanu and C. Sarmasanu, "Digital Test Signal Generator Implemented With FPGA" in *Proc. of 11th IMEKO TC-4 Symposium 2001,* Lisbon, Portugal, 13-14 September, 2001, in press.

[5] D. Van den Bout, *The Practical Xilinx Designer Lab Book*, Upper Saddle River, New Jersey 07458, Prentice Hall, 1999, ISBN 0-13-021617-8.

[6] J. F. Wakerly, *Digital Design, Principle & Practices*, Upper Saddle River, New Jersey 07458, Prentice Hall, 2000, ISBN 0-13-082599-9.

[7] M. M. Mano and C. R. Kime, *Logic Computer Design Fundamentals*, Upper Saddle River, New Jersey 07458, Prentice Hall, 2000, ISBN 0-13-016176-4.