DIGITAL TEST SIGNAL GENERATOR IMPLEMENTED WITH FPGA

Liviu Breniuc[(1)], Alexandru Salceanu[(2)], Constantin Sarmasanu[(3)]

[(1)] Department of Electrical Measurements, Faculty of Electrical Engineering, Iassy, 6600 Romania
Phone (40) 32 278683 Fax (40) 130054 e-mail: lbreniuc@ee.tuiasi.ro
[(2)] Department of Electrical Measurements, Faculty of Electrical Engineering, Iassy, 6600 Romania
Phone (40) 32 278683 Fax (40) 130054 e-mail: asalcean@ee.tuiasi.ro
[(3)] Department of Electrical Measurements, Faculty of Electrical Engineering, Iassy, 6600 Romania
Phone (40) 32 278683 Fax (40) 130054 e-mail: csarmas@ee.tuiasi.ro

Abstract - *In this paper we present a digital generator which produces sine-wave signals with variable frequency. The shape of the signal is stored in memory. The modification of the frequency is achieved by changing the number of samples used to generate a period. The digital circuitry of the generator is implemented using a field programmable gate array (FPGA) and a microcontroller. Experimentally we have generated sine-wave signals with frequencies between 1÷100 kHz and analysed their content in harmonics. Finally we present some possibilities for optimising the generator.*

Keywords - Digital generator, fractional addressing, FPGA.

1. INTRODUCTION

In order to test electrical devices, we need variable frequency generators. Usually, this applications make use signals with variable frequency and amplitude. An interesting solution for digitally generating sinusoidal signals with variable frequency is presented in [1]. The block diagram of the generator is presented in Fig.1.

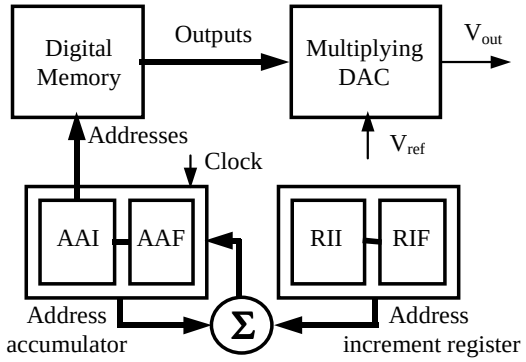

Fig.1 – Block diagram of the generator.

The generator consists of a memory block, a multiplying DAC, an address accumulator, an address increment register and an adder. The memory contains the sine waveform samples (N samples). The address accumulator has a fixed part (AAI) and a fractional part (AAF). The address increment register has also a fixed part (RII) and a fractional one (RIF). The memory address is calculated by adding the old content of the address accumulator with the content of the address increment register. Only the integer part of the address accumulator is sent to the memory. Considering that *d* is the value of the increment register, for *d=1* the sine wave is generated with *N* samples (frequency *f*), for *d=2* the frequency will be *2f* (only 0, 2, 4, 6,… samples are used), for d=3 the frequency will be *3f* (0, 3, 6, 9 samples are used), and so on. In this way, the maximum frequency which can be generated is *(N/2)f* (*N* even).

The problem arises when the fractional multiple frequencies of *f* have to be generated. For example, when *d=1.5* (fractional addressing), *1.5f* is generated (0, 1, 3, 4, 6…samples are used). Similarly, any other frequency can be obtained. The generated signal harmonic spectrum is studied in [1] and [2].

2. IMPLEMENTATION OF THE GENERATOR

The generator is made of a digital part and an analog part. For the digital part we have used an XS40-010XL Board v.1.4 from XESS Corporation [3]. The structure of this board and the interconnections between the components are indicated in Fig.2.

The board contains mainly the following components:

- XILINX XC4010XL FPGA[4]
- 8051 microcontroller
- 128 kB SRAM memory
- EEPROM memory
- programmable oscillator
- 7-segment LED display.

The board can be connected to a PC using a parallel port cable. On the board are also available a VGA port and a PS/2 port.

The FPGA integrated circuit can work independently and the bitstream configuration file (.bit file) can be stored into the EEPROM or can be downloaded directly to the FPGA from the PC using the parallel port.

The FPGA can also work together with an 8051 microcontroller in which case, the program for the microcontroller (.hex file) is downloaded to the SRAM also using the parallel port.

In the last case, a part of the functionality of the application can be implemented into the FPGA and the other

part can be implemented using the microcontroller. Generally, low-level functions of the application are implemented into the FPGA while the high-level functions are implemented using the microcontroller.

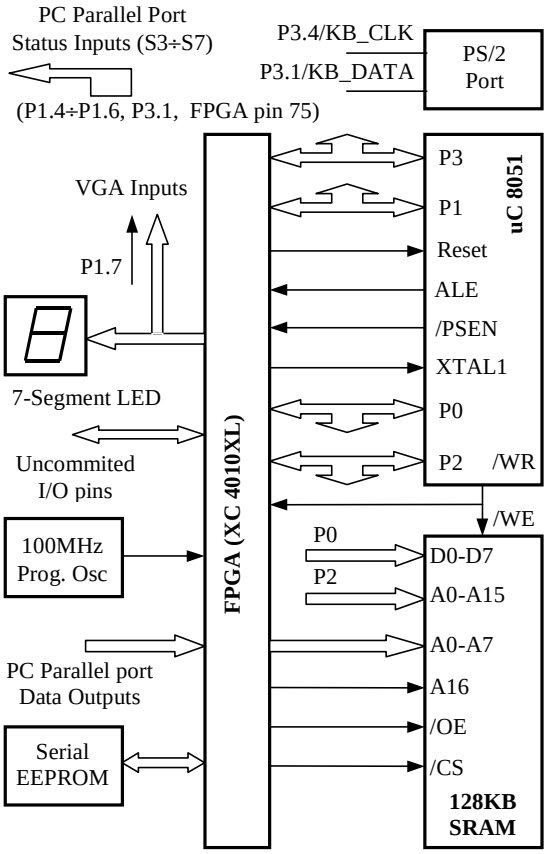

Fig.2 – Structure and interconnections of XS40 Board.

The basic design flow for building FPGAs+microcontroller applications is shown in Fig.3.

The XS40 board comes with a set of tools, XSTools, for interfacing the board to the PC. Also available are the GXSTools, a set of utilities that provide the Windows interface to XSTools [5]. The GXSTools package contains the following utilities:

- GXSTest: used to test if the XESS board works correctly,
- GXSSETCLK: programs the frequency of the on-board oscillator,
- GXSLoad: downloads the .bit and .hex files to the XESS board,
- GXSPort: sends logical signals to the board using the parallel port.

For the implementation of the digital part of the generator we have used both the FPGA and the microcontroller.

The microcontroller coordinates the operation of the generator by receiving from the PC the information regarding the frequency of the signal to be generated. It processes this information, sends it to the FPGA, and displays the information on a LCD display.

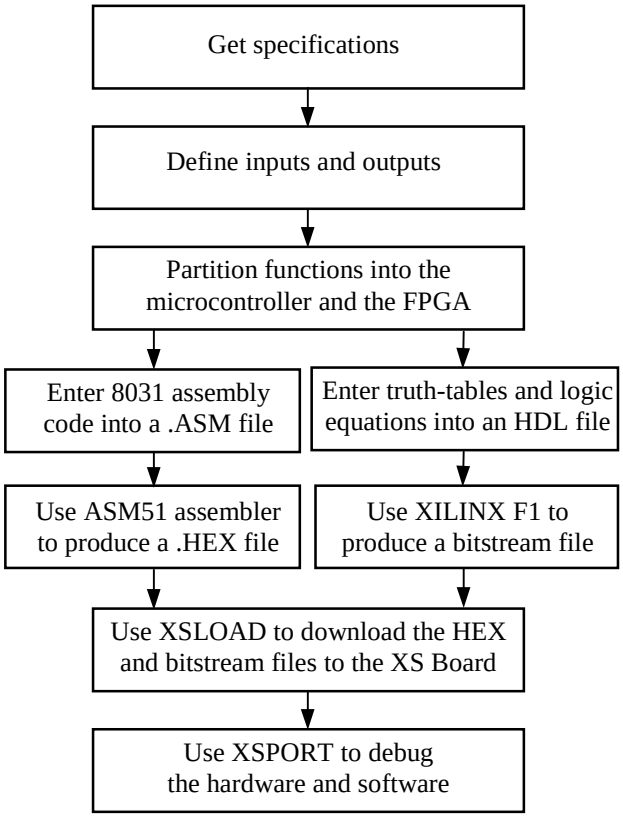

Fig.3 – The basic design flow for building FPGA+microcontroller applications.

The FPGA is used to implement the logic blocks which realise the interconnections between the board components and the functions which effectively generate the sine wave signals. The configuration file for the FPGA is obtained using the Xilinx Foundation version 1.5 software, [6].

In Fig.4 is depicted the schematic of the interconnections between the components on the XS40 Board specific to the digital generator which is implemented in the FPGA.

The pins indicated in the schematic in Fig.4 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 microcontroller we input signals ALE_, PSEN_, RD_ and WR_, with corresponding internal signals ALE_IN_, PSEN_IN_, RD_IN_, and WR_IN_.

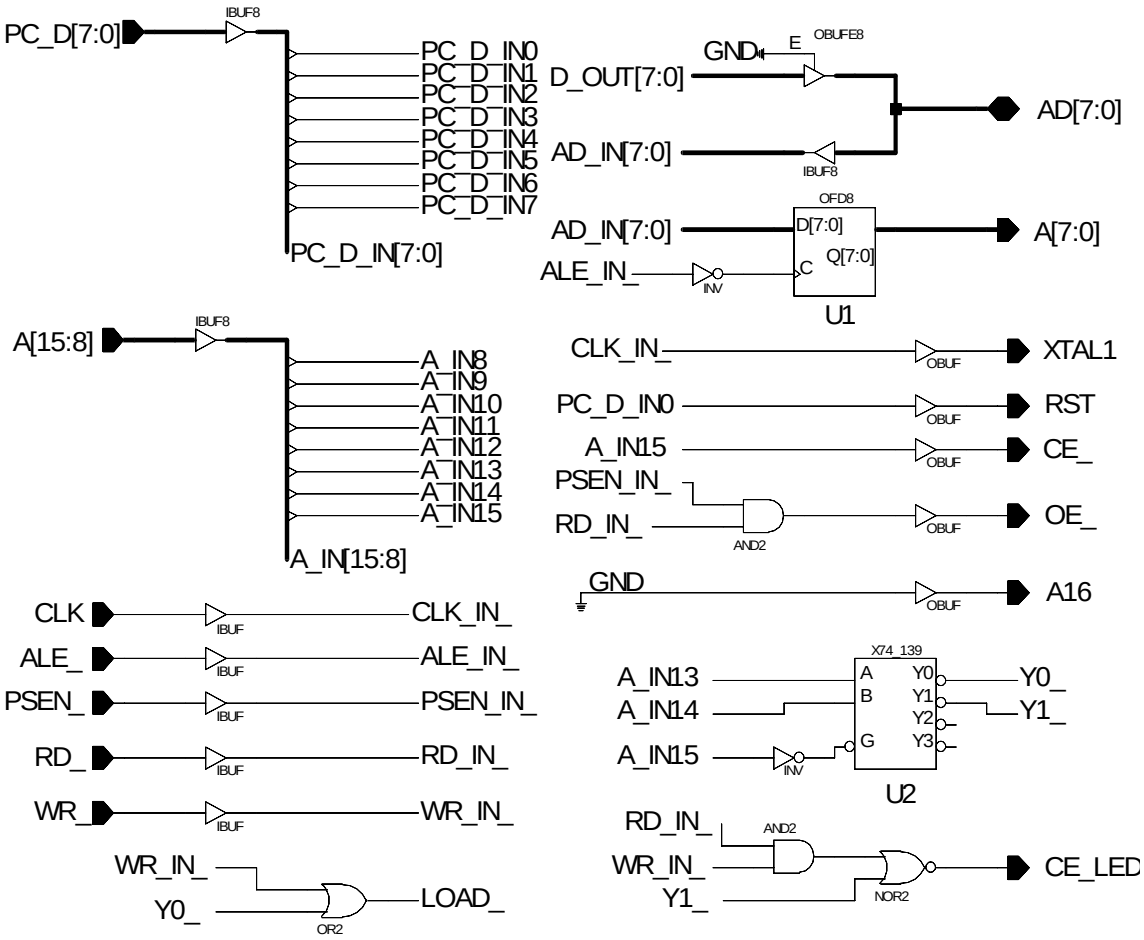

Fig.4 – The schematic of the interconnection between FPGA and microcontroller.

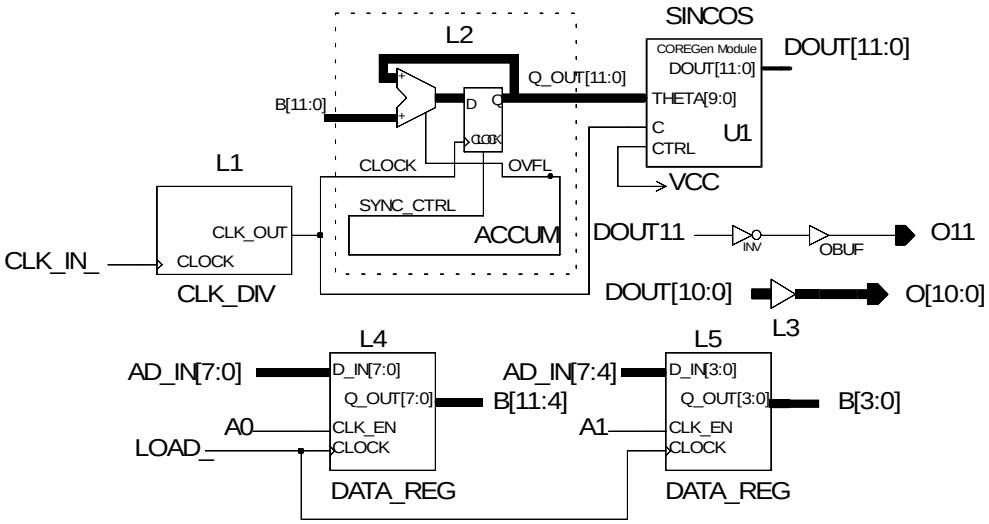

Fig.5 – Sine wave generator module.

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 lower address byte which connects to the SRAM.

The SRAM has a capacity of 128 kB and uses 17 address lines A16-A0. For the microcontroller we have allocated a total of 32 kB in the address space 0000H÷7FFFH (A16=0, A_IN15 is connected to the enable 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_. 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 two selection lines Y0_ and Y1_ (realised using circuit U2, a 2 to 4 decoder) which are active for the memory address spaces 8000H÷9FFFH and A000H÷BFFFH respectively. By means of signal Y1_, synchronised with signals RD_IN_ and WR_IN_, we have created the enable signal CE_LED (active high) for an LM016 LCD display. Signal Y0_, synchronised with signal WR_IN_, is used to create an active low enable signal LOAD_.

Fig.5 shows the sine wave signal generation block which is also implemented in the FPGA.

This block consists of circuits L1÷L5 of type LOGIBLOX and module U1 (SINCOS) created using Xilinx Core Generator software.

The SINCOS module, which is equivalent to a memory module, is used to store the shape of the sine wave. It has address lines THETA, a clock input C, a control line CTRL for the selection of the stored function and the data outputs DOUT. The selection of the stored function work as follows: when CTRL is high we have the sine function whereas when CTRL is low we have the cosine function. The module stores the samples corresponding to the first quarter of the period. Using the value of the first two address lines (the two most significant ones), by means of the internal logic, we can generate the full period of the sinusoidal signal. This technique simplifies the realisation of SINCOS block. In our case, the sinusoidal shape will be generated using 1024 samples (THETA[9:0]) with a 12 bit resolution (DOUT[11:0]).

The implementation of this block use 108 configurable logic blocks (CLB) (the number of the CLB is computed with the relation 8m+6+(m+1)/2, where m is the storing resolution of the sample – in our case m=12).

The addresses (THETA) can be given in either unipolar or bipolar two-complement binary data format. We have chosen the unipolar data format.

The values of the samples (DOUT) will always be generated in bipolar two-complement fractional binary format. In order to convert this format in bipolar offset binary format, the MSB (DOUT11) will be negated. The outputs applied to the DAC are O[11:0].

The addresses for the SINCOS module are generated by the accumulator module L2 (ACCUM). This module contains clock input, inputs (B) to feed a constant value and outputs (Q_OUT). The state of the outputs after the first clock cycle is obtained by adding the old outputs (previous clock cycle) with the value of the constant value applied to inputs B.

In our case, the outputs and the constant are 12 bit wide (D_OUT[11:0], B[11:0]). The block U1 receives only the 10 most significant bits of the output Q_OUT[11:0], i.e. the outputs Q_OUT[11:2]. The outputs Q_OUT[1:0] represent the fractional part of the address (this doesn't apply for block U1). Thus, the expression of the constant that is applied to inputs B[11:0] is b=b9b8…b0.b-1b-2.

For example: for b=1.00 (1) we obtain addresses 0,1,2,3,…; for b=10.00 (2) we obtain addresses 0,2,4,6, …; for b=11.00 (3) we obtain addresses 0,3,6,9,…; for b=1.10 (1.5) we obtain addresses 0,1,3,4,6,7,9,…; for b=1.01(1.25) we obtain addresses 0,1,2,3,5,6,7,8,10,… etc.

Depending on the value of the generated sinusoidal frequency, the microcontroller loads in registers L4 and L5 the value of constant b. The frequency f of the sine wave signal is given by (1):

$$f = \left(\frac{b}{N}\right) \cdot \left(\frac{f_{CLK\_IN\_}}{k}\right) \quad (1)$$

where: k is the division factor of the module L1 ($k=10$), N is the number of samples ($N=1024$), $f_{CLK\_IN\_}$ is the clock frequency ($f_{CLK\_IN\_}$=10MHz).

The analog part of the generator is depicted in Fig.6. The digital outputs O[11:0] are applied to the digital inputs of the AD7545 multiplying DAC. The U2 amplifier operates as a current-voltage converter. The amplifier U3 converts the voltage V_{DAC} from the output of the U2 amplifier to a bipolar voltage. The voltage V_{OUT} from the output of amplifier U3 is given by (2):

$$V_{OUT} = -V_{REF} - 2V_{DAC} \quad (2)$$

Taking into consideration that V_{DAC} is in the range $0 \div V_{REF}$, V_{OUT} will be in the range $-V_{REF} \div V_{REF}$ ($V_{REF}=+5V$).

3. EXPERIMENTAL RESULTS

The schematic presented in the previous section was implemented and tested experimentally. We have generated sine wave signals in the range 1÷100kHz, the number of samples was 8 to 1024 with a 12 bit resolution and with an amplitude of ± 5V.

For the study of the generated signal we have used a data acquisition system based on a National Instruments AT MIO 16X board. All data have been processed using MATLAB.

In Table I are given the level of harmonics *(n-1)f*, *(3n-1)f* and *(5n-1)f* (where *n* is the number of samples used for the generation of a period of the sinusoidal signal of frequency *f*). The other harmonics are under the level of the harmonics presented in Table I.

Table I

Harmonic	n							
	1024	512	256	128	64	32	16	8
(n-1)f	-69	-54	-53	-49	-35	-29	-23	-16
(3n-1)f	-92	-69	-68	-69	-55	-52	-51	-57
(5n-1)f	-83	-78	-66	-67	-70	-55	-53	-60

Analysing Table I we can see that for a reduced number of samples, the harmonic *(n-1)f* has a high value.

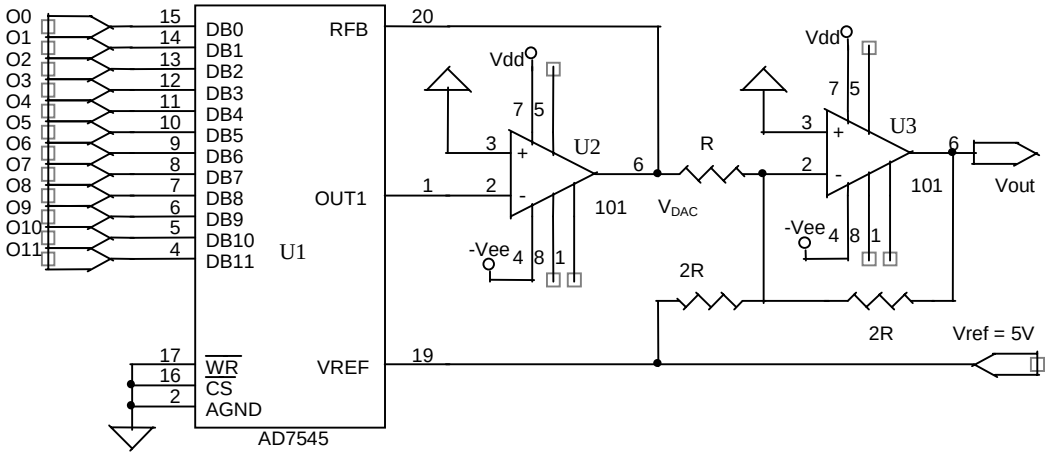

Fig.6 – The analog part of the generator.

For a sufficiently large number of samples, the harmonics have a low level and occur at high frequencies related to the fundamental (easy to filter).

The generator can be realised with digital outputs synchronised with well-determined moments of time related to the period of the sinusoidal signal (useful for phase sensitive detection).

There is also the possibility of generating two sinusoidal signals with same frequency and with a known phase relation between them (useful for the impedance measurement using the two generators method).

Coarse modification of the generator frequency can be obtained by modifying the clock frequency CLK_IN_ or changing the division factor of module L1 in Fig.5.

REFERENCES

[1] Y. C. Jenq, "Digital Spectra of Nonuniformly Sampled Signals – Digital Look-Up Tunable Sinusoidal Oscilators", *IEEE Trans. Inst. Meas*., vol. 37, No. 3, Sept. 1988, pp. 358-362.

[2] Y. C. Jenq, "Digital Spectra Of Nonuniformly Sampled Signals – Fundamentals and High-Speed Waveform Digitizers", *IEEE Trans. Inst. Meas*., vol. 37, No. 2, June 1988, pp. 245-251.

[3] ***, *XS40, XSP Board V1.4 User Manual*, XESS Corporation, http://www.xess.com.

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

[5] ***, GXSTOOLs *V3.0 User Manual*, XESS Corporation, http://www.xess.com.

[6] ***, *Foundation Series Quick Start Guide 1.5-0401762*, Xilinx Corporation, http://www.xilinx.com.