Development of Evaluation Kit Extension Modules for Analog-to-Digital and Digital-to-Analog Conversion

Ivanina Orlinova Petrova, Borislav Emilov Mihaylov and Slavy Georgiev Mihov

Abstract – **FPGA are becoming more and more widespread in techniques and machinery. In this paper, is made a description of the development of interface for Evaluation Kit Extension Modules for Analog-to-Digital and Digital-to-Analog Conversion. The digital interface is developed on VHDL and is tested in the kit's development environment. The developed extension modules for the FPGA evaluation kit, make a system with versatile applications such as digital filter and functional generator, etc.**

Keywords – **VHDL, ADC, DAC modules, interface**

I. INTRODUCTION

A novel approach in embedded system development is using Field Programmable Gate Array (FPGA) [1] for programming interface for Analog-to-Digital Converter (ADC) and Digital-to-Analog Converter (DAC) modules. If till now the possibility for programming has been limited, the imposing new trend allows fast reprogramming, as well as synthesizing of a file, which will be loaded in the FPGA. This process uses special-purpose software provided by the FPGA manufacturer. Everything mentioned makes this realization contemporary and compatible with current tendencies in electronic devices design.

Figure 1 shows block diagram of the developed system. It consists of Spartan 3A Evaluation Board (AvNet) [2][3] and expansion modules ADC and DAC.

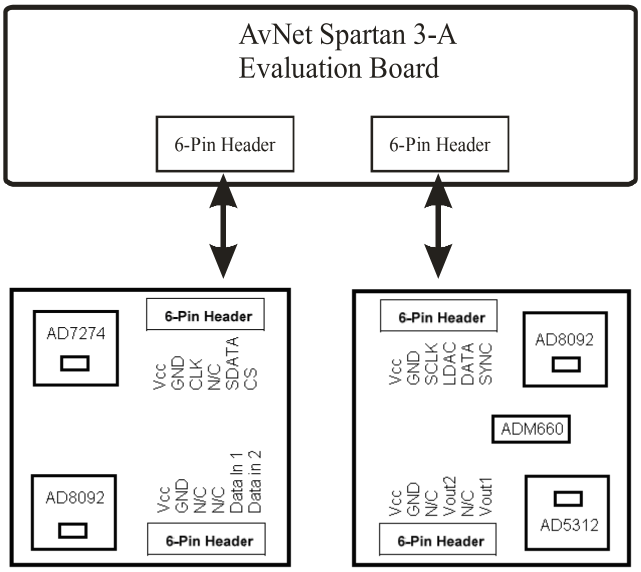


Figure 1 – Block diagram

On Figure 1 are shown expansion modules for Analog-to-Digital (AD) conversion and Digital-to-Analog (DA) conversion with corresponding input and output signals. For realization of the modules are used integrated circuits of Analog Devices.

II. DIGITAL INTERFACE

For analog-to-digital convertion is used integrated circuit AD7274 and for digital-to-analog converter – AD5312. ADC is 12-bit and DAC is 10-bit. For data interchange between the two modules, are taken the 10 Most Significant Bits (MSB) of the ADC. In this way steady communication between two modules is provided.

The two modules use specialized serial interface compatible with I^{2}C and SPI. The control signals of the ADC are: SCLK, CS, SDATA; The control signals of the DAC are: SCLK, SYNC, LDAC, DIN.

There are two possibilities for developing working cycle for ADC: cycle with 14-bit SCLK and cycle with 16-bit SCLK. It is chosen the first one – 14-bit SCLK cycle shown on Figure 2.

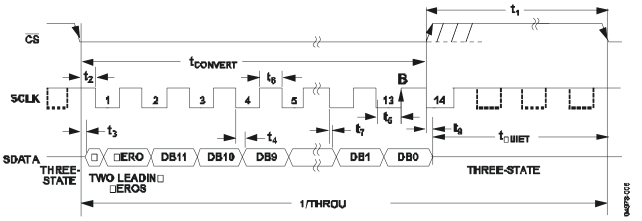


Figure 2 – ADC working cycle

The serial clock provides the conversion clock and controls the data transfer from the AD7274 during analog to digital conversion.

The CS signal initiates the data transfer and conversion process. The falling edge of CS puts the track-and-hold into hold mode and takes the bus out of three-state. The analog input is sampled and the conversion is initiated.

For the AD7274, the conversion requires completing 14 SCLK cycles. Once 13 SCLK falling edges have elapsed, the track-and-hold goes back into track mode on the next SCLK rising edge, as shown in Figure 2. If the rising edge of CS occurs before 14 SCLKs have elapsed, the conversion is terminated and the SDATA line goes back into three-state.

Serial interface for DAC is realized in the way shown on Figure 3. The input shift register is 16 bits wide (see Figure 4). Data is loaded into the device as a 16-bit word under the control of a serial clock input, SCLK. The timing diagram for this operation is shown in Figure 4. The 16-bit word consists of four control bits followed by 8, 10, or 12 bits of DAC data, depending on the device type. The first bit

I. Petrova, B. Mihaylov and S. Mihov are with the Department of Electronics, Faculty of Electronic Engineering and Technologies, Technical University – Sofia, 8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria, e-mail: smihov@tu-sofia.bg

loaded is the MSB (Bit 15), which determines whether the data is for DAC A or DAC B. Bit 14 determines if the reference input is buffered or unbuffered. Bit 13 and bit 12 control the operating mode of the DAC.

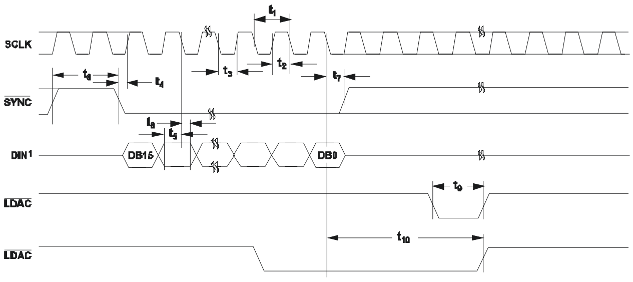
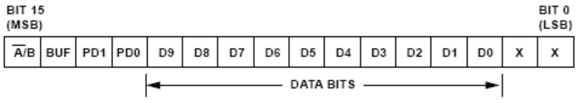


Figure 3 – DAC working cycle



BIT 15 (MSB)															BIT 0 (LSB)
A/B	BUF	PD1	PD0	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	X	X

Figure 4 – DAC input shift register

The SYNC input is a level-triggered input that acts as a frame synchronization signal and chip enable.

After the end of serial data transfer, data is automatically transferred from the input shift register to the input register of the selected DAC. If SYNC is taken high before the 16th falling edge of SCLK, the data transfer is aborted and the input registers are not updated.

III. Analog Modules

Attention is paid in developing the analog part of the modules. On figure 5 is shown the schematic solution for the AD converter.

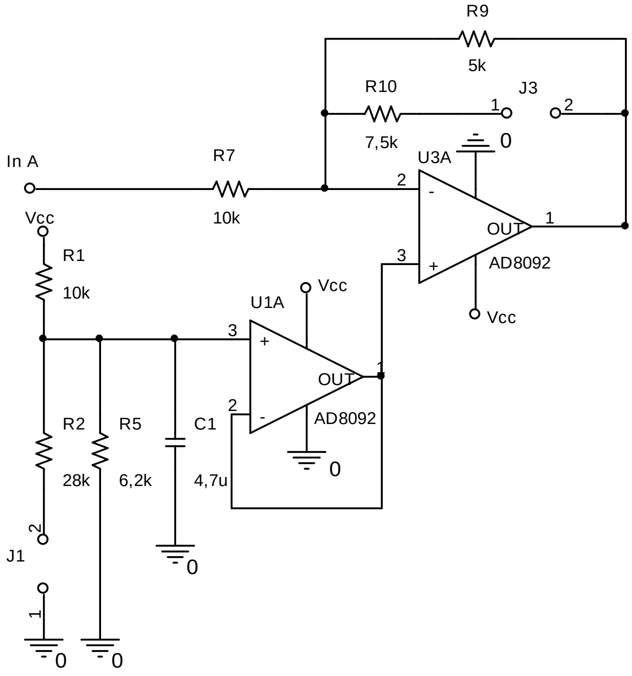


Figure 5 – Analog module of the AD converter

Input signal of AD7274 is within the range from 0 to 3V, according to the requirements of this project. This is achieved in the following way: the signal which is fed to the non-inverting input of the operational amplifier AD8092 (U_{3A}) is bipolar with amplitude ±3.3V. Jumper J_3 is open and the conversion gain is specified by resistors R_7 and R_9 with values respectively R_7 = 10kΩ and R_9 = 5kΩ. In the output of the operational amplifier is obtained signal with amplitude ±1.65V. In order to obtain unipolar signal in the range from 0 to 3.3V, bipolar signal must be shifted in positive direction. This is achieved by feeding reference voltage to the non-inverting input of the operational amplifier (U_{3A}).

Reference voltage is formed by feeding DC voltage with amplitude 3.3V to resistor couple R_1, R_2 and R_5, jumper J_1 is closed. Voltage in point 2 is 1/3 V_{cc}, and is fed to a buffer amplifier, realized as non-inverting operational amplifier with 100% negative feedback.

Principle of operation is analogous for the second channel of the AD converter.

On figure 6 is presented scheme decision of analog part of DA converter.

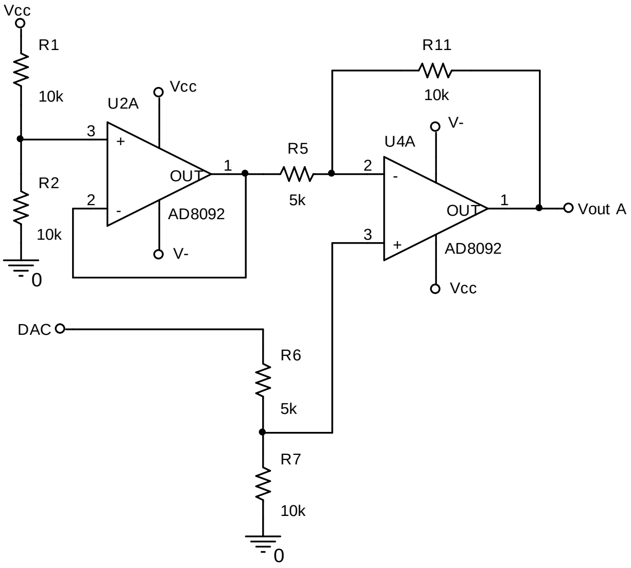


Figure 6 – Analog module of the DA converter

According to the requirements of the project, the output signal from the DA converter module should be bipolar. This is achieved in the following way: from integrated circuit AD5312 is obtained the output signal (produced by the programmable logic in the FPGA) within the range from 0 to 3.3V. This voltage can not be directly fed to operational amplifier AD8092 (U_{A4}), since its differential input voltage must not exceed 2.5 V (datasheet specification). Because of this, the amplitude of the signal is decreased to 2.2V (in control point U_4) using resistor couple R_6 and R_7, designed to produce in its middle point, voltage equal to 2/3 of the output one of AD5312 (control point U_2). The values of the resistors are R_6=5kΩ and R_7=10kΩ.

Output signal must be shifted in negative direction. This is achieved by feeding reference voltage to the inverting input of the operational amplifier AD8092 (U_{A4}). This reference voltage is obtained in the following way: DC voltage is fed to resistor couple R_1 and R_2. In point 1 is obtained 1/2 V_{cc}, because values of the resistors are in ratio 1 to 1. This voltage is fed to a buffer. This buffer is used to provide stable voltage in control point U_3. This voltage is fed to inverting input of AD8092 (U_{A4}) and is conversed with gain obtained by the ratio of resistors R_{11} and R_5 with values respectively $R_{11} = 10k\Omega$ and $R_7 = 5k\Omega$.

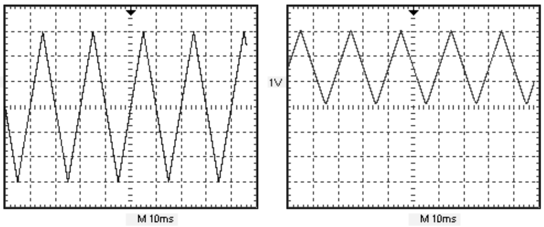


Figure 7 – ADC analog modules simulation

In order to prove the correct operation of the analog part of both modules is made a simulation testing in the environment MultiSim WorkBench. The simulation results for both input and output signals of the modules are shown on Figure 7 and Figure 8 (left and right respectively).

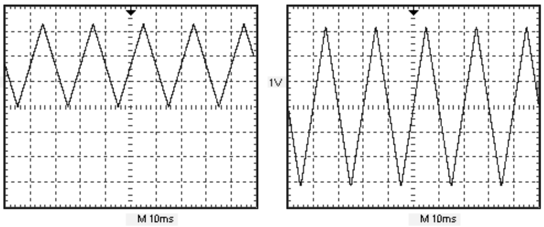


Figure 8 – DAC analog modules simulation

IV. VHDL INTERFACE

Major task in developing both modules is the program realization of the digital serial interface. This is done in ISE WebPack [4] environment (Xilinx) using VHDL [5]. At first is made a description of the modules as a black box (top-down approach [6]), defined are their input and output signals, and then is developed the behavioral description of each of the modules. Below is shown formal VHDL description of both modules.

```
entity AD7274 is
    Port (GCLK: in STD_LOGIC;
              CS : out  STD_LOGIC;
              DOUT : in  STD_LOGIC;
              CLK : out  STD_LOGIC;
ADC_DATA: out std_logic_vector(11 downto 0));
end AD7274;
```

```
entity AD5312 is
    Port ( SCLK : out  STD_LOGIC;
           LDAC : out  STD_LOGIC;
           DIN : out  STD_LOGIC;
           SYNC : out  STD_LOGIC;
           GCLK : in  STD_LOGIC);
end AD5312;
```

Follows behavioral description of the modules. VHDL code shows part of the description of DA converter. Specific for this module is the realization of a shift register. This is achieved in the following way:

```
D1<=D1(14 downto0)&'0';
process(GCLK)
       begin
       if rising_edge(GCLK) then
              counter<=counter + '1';
              D1<=D1(14 downto 0)&'0';
       end if;
```

In the behavioral description is made data interconnection of both modules (ADC and DAC data) as it shown with VHDL code below.

```
ADC: AD7274 PORT MAP
       (       GCLK => GCLK,
              CS => CS,
       DOUT=>DATAIN,
               CLK => CLK,
       ADC_DATA=>ADC_DATA
        );
```

Both modules (AD and DA) are intended to be used as a pair of input and output interfaces of the Spartan Evaluation Kit. The operation mode suggests converting bipolar analog input signal, processing its digital form by means of the programmable logic and converting it back to bipolar analog form in the output. Pairing the available 12 bits of the ADC output data and the 10 bits of the DAC input data is shown below with a VHDL code example, by trimming the excessive informative bits.

```
DAC_DATA<=ADC_DATA(11 downto 2);
```

V. SIMULATION RESULTS

The synthesized serial interface of the modules, is tested. This is done by creating a testbench in environment ISE WebPack. Testbench description is done in VHDL too. The paragraph below defines the GCLK signal that controls the transition of all important interface signals.

```
   GCLK_process :process
   begin
            GCLK <= '0';
            wait for GCLK_period/2;
            GCLK <= '1';
            wait for GCLK_period/2;
   end process;
```

Results from the simulation testing are shown on Figure 9 and Figure 10.

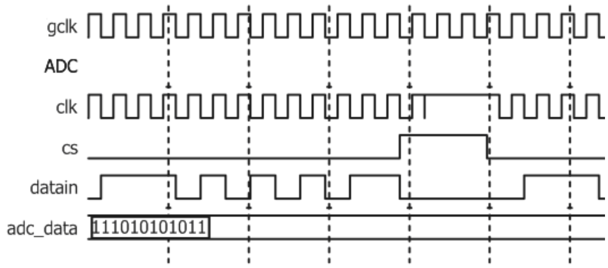


Figure 9 – ADC digital interface (VHDL realization)

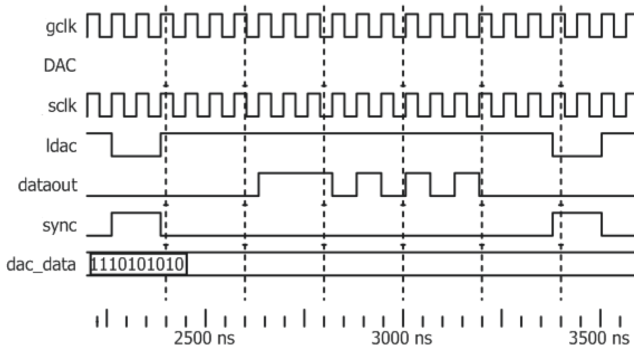


Figure 10 –DAC digital interface (VHDL realization)

As shown on Figure 8 signal SCLK is with the same frequency like signal GCLK. Signals LDAC and SYNC are controlled by a counter which controls the dataflow.

As a result of all coding, is generated a binary file of the developed serial interfaces between both modules (AD and DA) and the development kit, which is eventually loaded in the FPGA via a specialized software.

VI. Practical Results

Made is a practical testing of the developed system. The operation results are captured using digital oscilloscope and a functional generator. As shown on Figure 11, the functional generator supplies a sinusoid signal in range of ±2V (channel 1) in the input of the system. Channel 2 shows the signal in the output of analog part of AD converter in range from 0 ÷ 2 V.

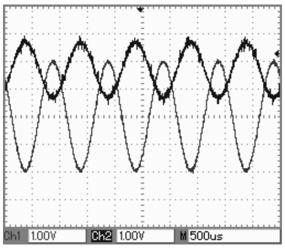


Figure 11 – ADC test

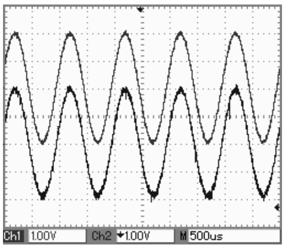


Figure 12 – System test

On Figure 12, at channel 1 of digital oscilloscope is shown signal from functional generator, channel 2 shows the signal in the output of DA converter in range of ±2V.

The so developed system is used for realization of the functional generator based on a lookup table. It generates four types of signals shown on (Figures 13 ÷16). Description of the functional generator is based on language VHDL. Initially in Matlab are generated four types of signals. Each signal is described with 1024 points. The points, describing all four signal forms are listed successively in a lookup table with 4096 lines. Its VHDL description is shown below.

```
type rom_table_type is array (addr_i_type) of
data_type;
        constant rom1_table: rom_table_type :=
               (0 => "1000000010",
               1 => "1000000101",
…              …       …       …       …
               4094 => "1000011001",
               4095 => "1000100010",
               others => (others => '0'));
```

By pressing button 1 from the lookup table are read first 1024 points, and on digital oscilloscope is shown sinusoid signal (shown on figure 13). The process of reading is shown in VHDL code below.

```
        process(BT1)
        begin
        if rising_edge(BT1) then
               inc1 <= inc1 + '1';
        end if;
```

By pressing the button again are read next 1024 points from the table and on oscilloscope visualize the next signal. In this way each pressing of the button causes reading the data points of the next signal.

Supplemental features that are already realized are: changing the frequency of the generated signals by skipping points from the lookup description and changing signal amplitude (by division with different coefficients).

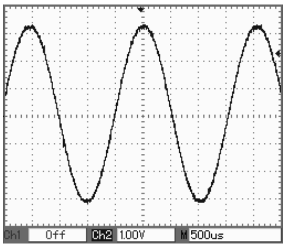


Figure 13 – Test Signal 1

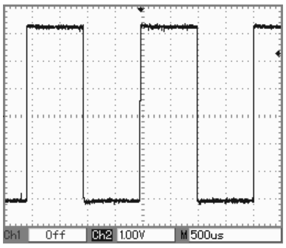


Figure 14 – Test Signal 2

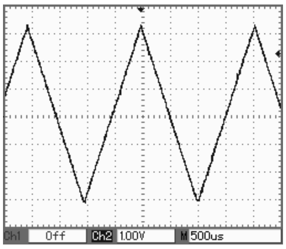


Figure 15 – Test Signal 3

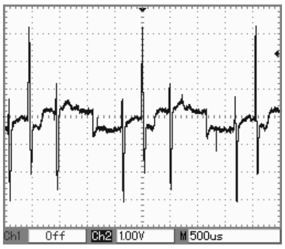


Figure 16 – Test Signal 4

The developed system is also used for realization of a digital FIR filter with 16 coefficients. In Matlab are generated the coefficients of the low-pass, high-pass and band-pass filters. These coefficients are stored in a lookup table, same as the described above. The type of the filter can be changed by simple push of a button.

VII. Conclusion

In this paper is made an analysis on the existing decisions for realization of interface for expanding modules. Depending on the analysis is suggested decision using VHDL description (bit file) and loading it in FPGA matrix through specialized software. Simulation testing has been made after the synthesis, which proves its efficiency. Practical results obtained when connecting the expansion modules to the evaluation board, proved the workability of the system. In this way the developed system can be used as a part of many other devices or as independent device with ability to change it operation depending on the VHDL programming file. This makes the system universal.

Acknowledgement

This research was supported by the NIS TUS funded project, No 091ni040-03/2009.

References

[1] www.xilinx.com – Xilinx Spartan-3A
[2] http://www.avnet.com/ – Spartan-3A Evaluation Kit
[3] N. Ivanov, *"Algebra of reprogramming devices"*, ISBN: 978-954-729-253-6, Atika, May 2007
[4] www.xilinx.com/ise/ – Xilinx ISE Web Pack
[5] P. Manoilov, *"Developing digital devices using extra large integrated circuits with VHDL"*, ISBN: 954-438-436-7, Technical Univertsity – Sofia, March 2007
[6] D. Smith, *"HDL Chip Design"*, ISBN:0-9651934-3-8, Doone Publications, 1996