Design of a UART –to- Bell202 converter SoC by using Altera-Intel FPGA

Agostino Giorgio, Francesco Paris

Digital Electronics Systems and Applications Lab, Dipartimento di Ingegneria Elettrica e dell'Informazione, Politecnico di Bari, Via E. Orabona, 4 70125 Bari – Italy

Corresponding author mail: *agostino.giorgio@poliba.it*

Abstract: The widespread use of Field Programmable Gate Array (FPGA) in the consumer electronics market makes designers integrate more and more solutions with these programmable logic devices. In this paper, the authors use a CYCLONE V 5CSEMA5FC31C6N Altera-Intel FPGA in order to implement a UART-to-Bell202 converter for the direct interfacing of an FPGA with an analogue line on the voice band, not still available for industrial and consumer electronic for communications applications. In particular, a transmitter is created with an Audio Shift Keying modulation without phase continuity, capable of converting UART signals coming from any UART transmitter (for example a personal computer, PC) into a modulated signal. The design has been performed using the Terasic DE1-SoC development board, while instrumental tests have been performed with a ZEROPLUS LAP-C logic states analyzer and a CML1102 oscilloscope.

Keywords: UART, Altera-Intel, FPGA, Oscilloscope

I. INTRODUCTION

The Bell202 protocol is a data communication method that uses an AFSK (Audio Shift Frequency Keying) digital modulation to transmit digital information through a transmission mean, such as a traditional analog telephone line.

The Bell202 protocol was created by AT&T and then it was standardized by the ITU-T V23 legislation, from which the data transmissions modems on analog V23 lines derive [1]. Currently, the Bell202 protocol, in its standard version, is used in VHF/UHF radio frequency communications with APRS communications (Automatic Packet Reporting System) to send digital data through repeating stations [2]. The improved version for industrial use of this protocol is the HART (Highway Addressable Remote Transducer) protocol, having an extensive spread because it is possible to perform data transmission by overlapping the digital data to the telephone (analog) one on the voice band without introducing distortion or leak of information [3].

The Bell202 protocol can be implemented in two different ways:

- Half-Duplex: AFSK modulation, having a Baud rate of 1200, a mark frequency of 1200 Hz and a space frequency of 2200 Hz;
- Full-Duplex: DBFSK modulation, having a transmission speed of 1800 Hz.

Currently the half-duplex configuration is widely used and, in particular, a serial digital signal is AFSK modulated.

As previously stated, the signal has a mark frequency of 1200 Hz and a space frequency of 2200 Hz; there is also a 384 Hz call tone, not properly involved in the serial data transmission.

Figure 1 shows the frequency band occupied by the protocol. This band also deals with signals for analogue telephone communications.

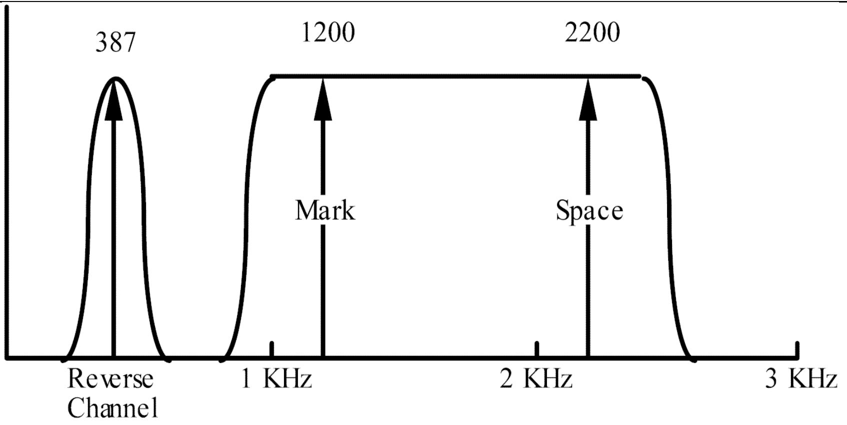


Fig. 1 - BELL202 protocol frequency band

The main fields of application range from 64 kByte standard telephone communications to VHF/UHF radio-communications for civil and military use.

Anyway, despite the importance of the UART-to-Bell202 converter and the significance research and design activity dealing with the FSK modulation and modulators [4 – 8], it is not still available an integrated, UART-to-Bell202 converter, suitable for industrial and consumer electronic for communications applications.

Therefore, this paper aims at describing the design of a UART –to- Bell202 converter implemented as system on chip (SoC) on Altera-Intel FPGA.

It has many very interesting features, giving many advantages and improvements for communications applications.

In fact, it allows directly interfacing an FPGA with an analogue line on the voice band.

Moreover, the designed converter allows the standalone, SoC implementation, of an AFSK modulator as an embedded system.

Furthermore, the designed converter is fully testable and therefore reliable.

Finally, the designed converter is suitable also for more performing processing system able to communicate on standard telephone lines.

Therefore, in section 2 the hardware part of the project is described in detail; in section 3 the firmware part of the project is detailed; in section 4 testing and debug procedure and results are described. Finally, conclusions and future developments are in section 5.

II. DESCRIPTION OF THE HARDWARE PART OF THE PROJECT

The project consists in the design and implementation of a UART-to-BELL202 converter, using an Altera-Intel CYCLONE V 5CSEMA5F31C6N FPGA mounted on the Terasic DE1-SoC development board [9]. It is useful to remind that the Altera-Intel Cyclone V FPGA is equipped with a microcontroller (soft processor), named NIOS II, and with a hard processor, named HPS, allowing the implementation of a complete embedded SoC. In this design it is used the NIOS II.

The development suite used is the Integrated Development Environment (IDE) Quartus powered by Altera [10 - 12], while the Eclipse environment is used to create the microcontroller's firmware [11].

Figure 2 shows the functional diagram of a Bell 202 modem for data transmission with UART protocol.

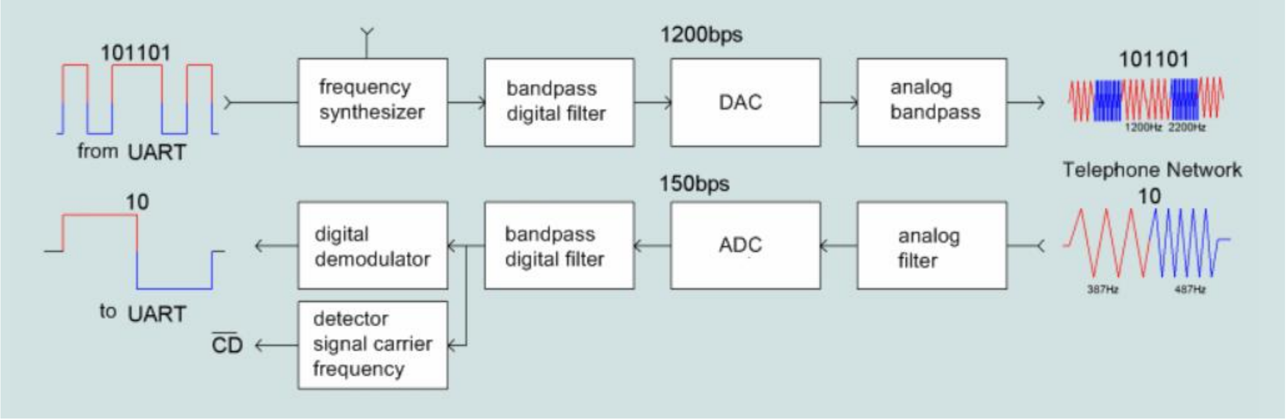


Fig. 2-Functional diagram of a UART-to-BELL202 converter

In order to convert a UART signal into a BELL202 standard signal, it is necessary to use an FSK modulation in audio band (AFSK).

The signal is modulated through an FSK modulator without phase continuity equipped with square wave oscillators to generate the mark and space frequencies.

The model of the implemented modulator is shown in figure 3

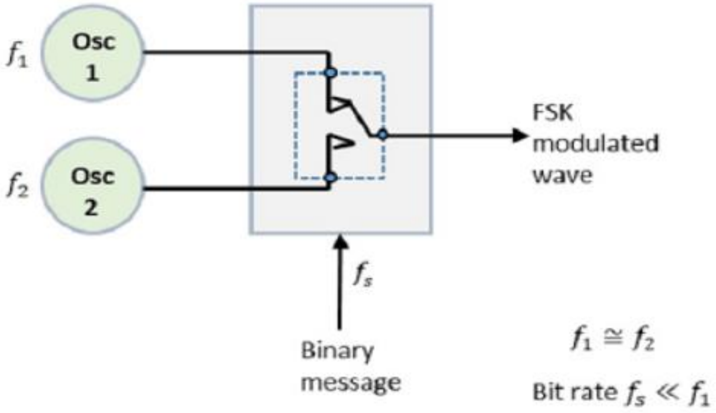


Fig. 3 - FSK modulator without phase continuity

The frequencies f_1 and f_2 are 1200 Hz (mark frequency) and 2200 Hz (space frequency), respectively, while binary data will be generated by a PISO (Parallel Input Serial Output) register, which will convert the parallel data obtained by a data-logging operation of the NIOS II microcontroller directly from the PC serial port.

The block diagram of the project is shown in figure 4

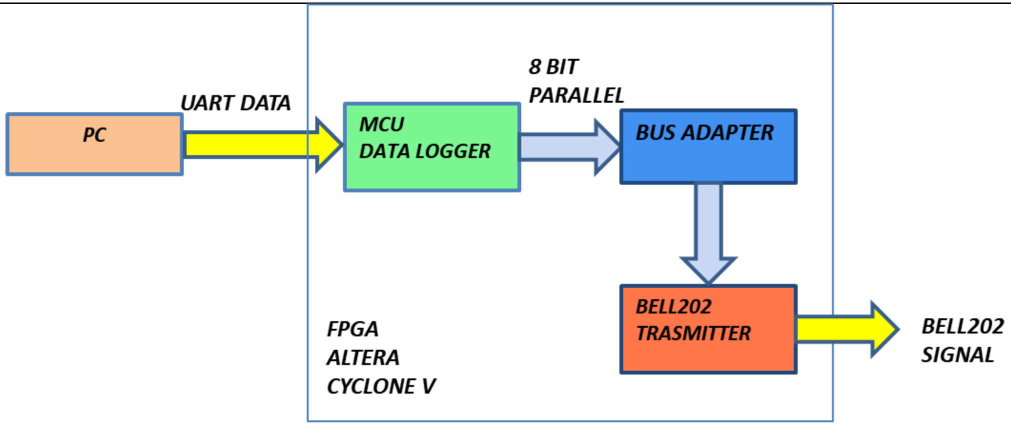


Fig. 4- Block diagram of the project

The schematic of the hardware design is shown in figure 5

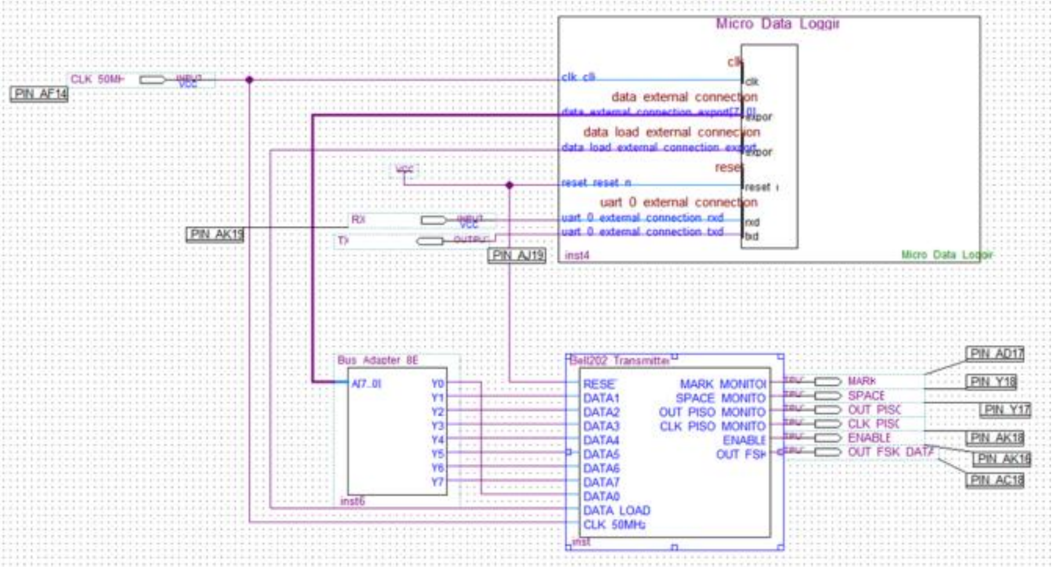


Fig. 5 - Schematic of the hardware part of the project

The role and the behavior of the 3 main blocks in figure 5, is explained in the following:

- MCU DATA LOGGER: its core is the NIOSII microcontroller (FPGA soft processor) executing instructions provided by a very compact firmware. It aims at receiving a string in ASCII code sent according to the UART protocol and rearrange it in parallel format. This block is also responsible for generating the control signals of the BELL202 transmitter, which allows the correct synchronization of the system;
- BELL202 TRASMITTER: it is the block performing the signal modulation. It has a parallel to serial converter inside, the control logic for the modulation and finally the oscillators necessary for the generation of the output FSK package of data;
- BUS ADAPTER: the data logger has the purpose to convert in parallel the data arriving from a UART transmitter. Since this data is packaged in an 8-bit array, it is necessary to convert the array into variables of wire type suitable for the FSK transmitter.

A. BELL202 transmitter

The BELL202 transmitter has been designed by functional blocks described in Verilog HDL and subsequently converted to symbol file suitable for the schematic entry of larger circuits in Quartus environment. Figure 6 shows the symbol, inputs and outputs of the Bell202 designed transmitter.

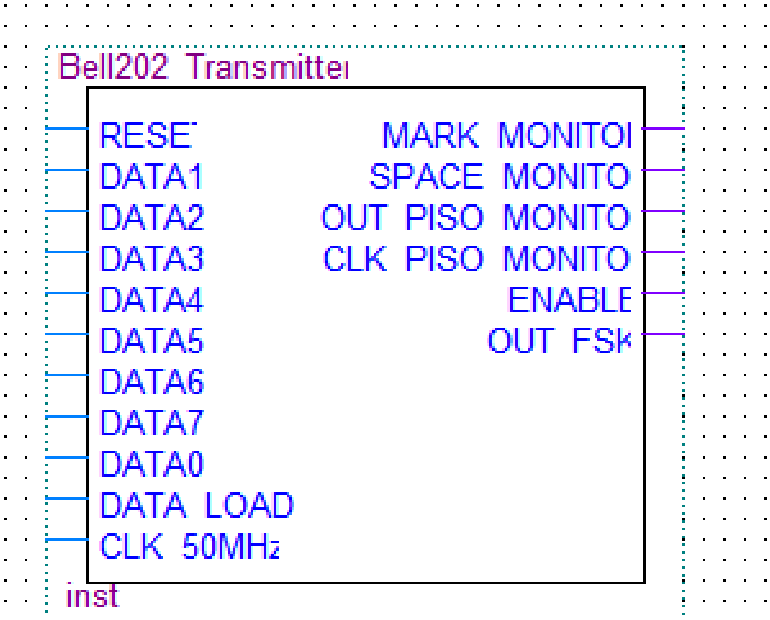


Fig. 6 - Transmitter circuit symbol

It follows the description of the block inputs of the transmitter:

- DATA0..DATA8 are the data inputs of the block, supplied by the NIOSII soft processor;
- DATA_LOAD is a block control signal that loads the data in the PISO register; the DATA_LOAD is an active high signal and its time duration must be shorter than the transmitter operating clock, otherwise the data will not be transmitted correctly;
- RESET is the input used to reset and initialize the block;
- CLK_50MHz is the general system clock and has the task of synchronizing all the blocks.

The block outputs of the transmitters are the following:

- OUT_FSK is the output from which the input UART signal is processed and modulated in AFSK. This is the output suitable to be sent to any radio or telephone transmission systems;
- ENABLE is the signal opening an 8-bit time window on the output line. Its duration is equal to 8 times the duration of the single bit (t = 0.00667 s, f = 150Hz). It can be used both as a verification signal and as a synchronization signal for any additional systems;
- MARK_MONITOR \ SPACE_MONITOR are signals suitable for instrumental verification of the frequencies generated for the mark and space bits of the FSK modulation;
- OUT_PISO_MONITOR is the signal suitable for the instrumental verification of data serialized by the PISO register;
- CLK_PISO_MONITOR is the instrumental verification signal of the PISO register clock.

The transmission module is composed of various functional blocks:

- PISO Register, shown in figure 7: it is a sliding register composed of a cascade of D-type flip-flops and a series of 2-bit multiplexers. The PISO register serializes the input data after acquiring it from the microcontroller block. Obviously the PISO has an extension of 8 bits, equal to the UART data package;

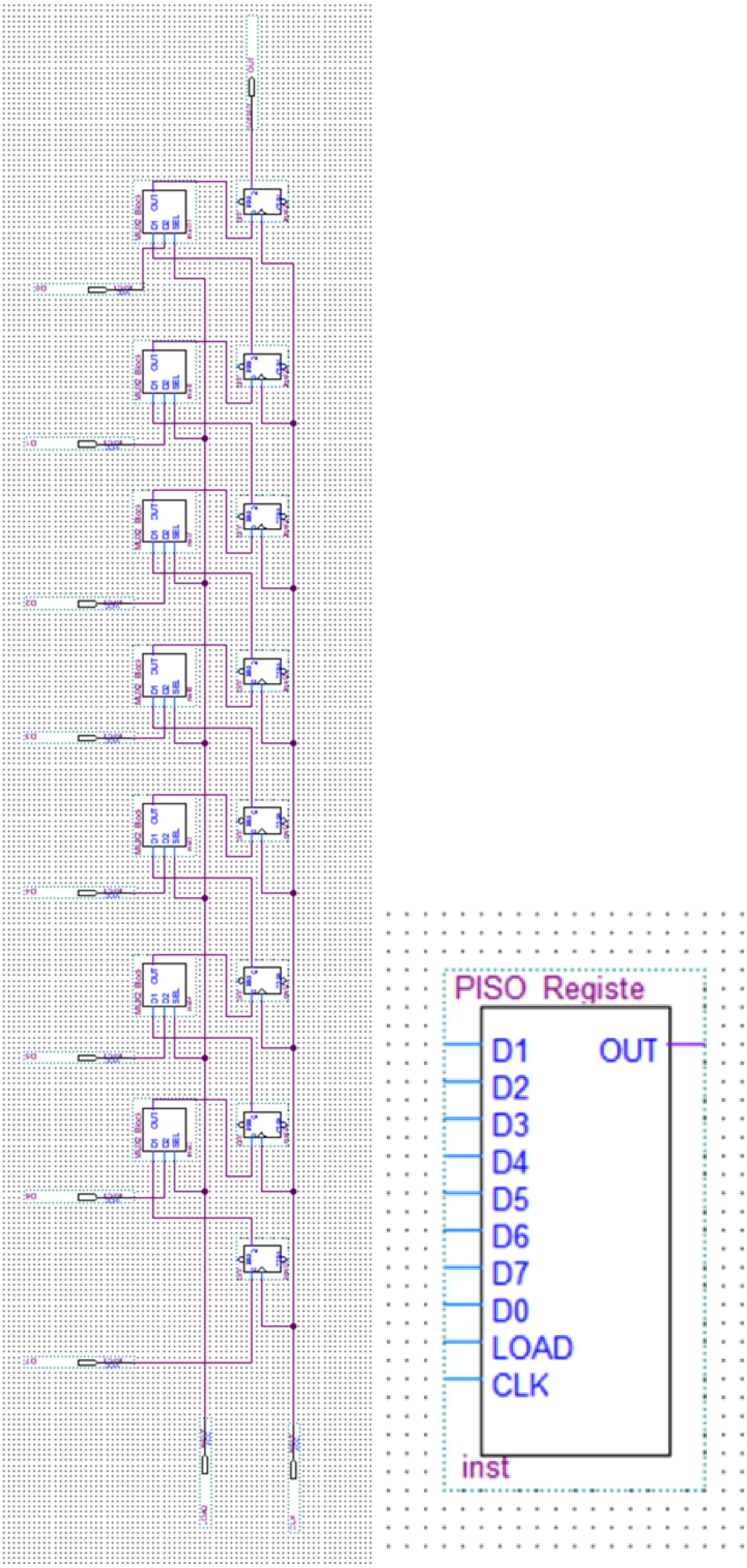


Fig. 7 – PISO register schematic and related symbol

- Clk_Divider: they are frequency dividers, shown in figure 8, generating square waves with a duty cycle of 50%. The frequency dividers have been designed in Verilog HDL (Hardware Description Language) [13] and can be set using the parametrical variable K, calculated as:

$$K = \frac{\frac{1}{f_{OUT}}}{\frac{1}{f_{ALTERA}}}$$

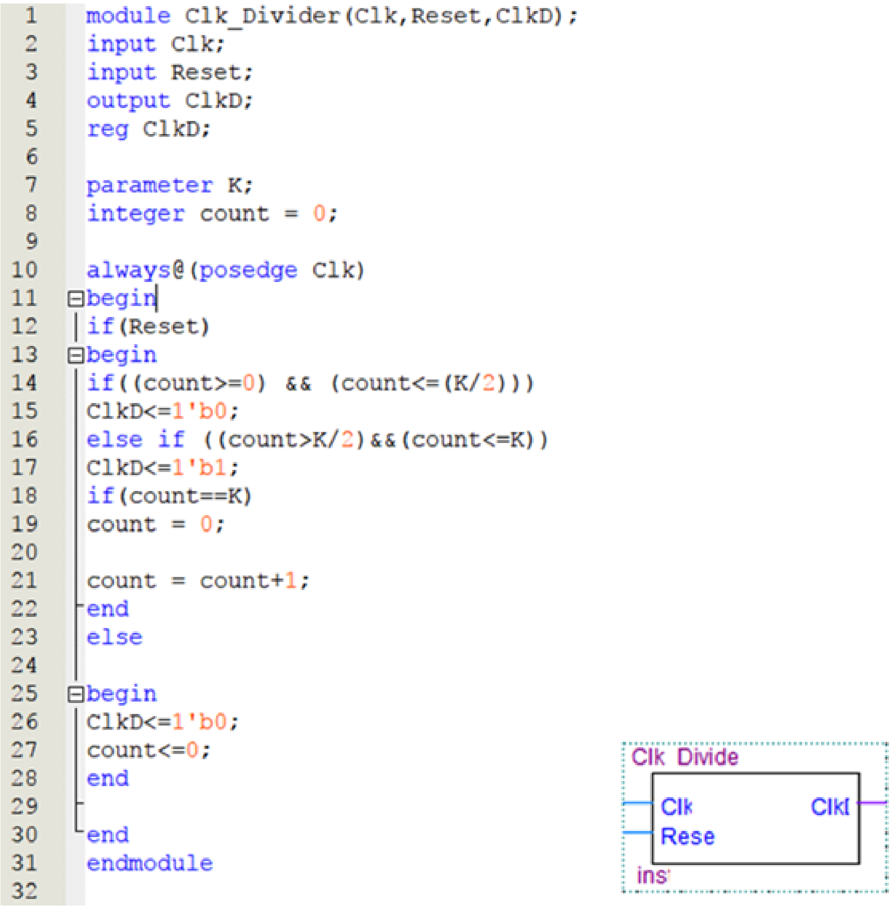
The frequency f_{ALTERA} is that generated by the clock powered by the DE1_SoC development board, which value is 50MHz.

In the described project, to generate the mark frequency, the K value is:

K = (1/1200) / (1/50 * 10 ^ 6) = 41167

While, for the space frequency, the K value is:

K = (1/2200) / (1 / 50 * 10 ^ 6) = 22727



```
module Clk_Divider(Clk,Reset,ClkD);
input Clk;
input Reset;
output ClkD;
reg ClkD;

parameter K;
integer count = 0;

always@(posedge Clk)
begin
if(Reset)
begin
if((count>=0) && (count<=(K/2)))
ClkD<=1'b0;
else if ((count>K/2)&&(count<=K))
ClkD<=1'b1;
if(count==K)
count = 0;

count = count+1;
end
else

begin
ClkD<=1'b0;
count<=0;
end

end
endmodule

```

Fig. 8 – Clk_Divider design in Verilog HDL and related symbol

- BUFFER_OUT: this block, which schematic is shown in figure 9, opens an 8-bit time window, in which the modulated signal can transit. Once the data have been transmitted, the output line is always placed at a low logical level.

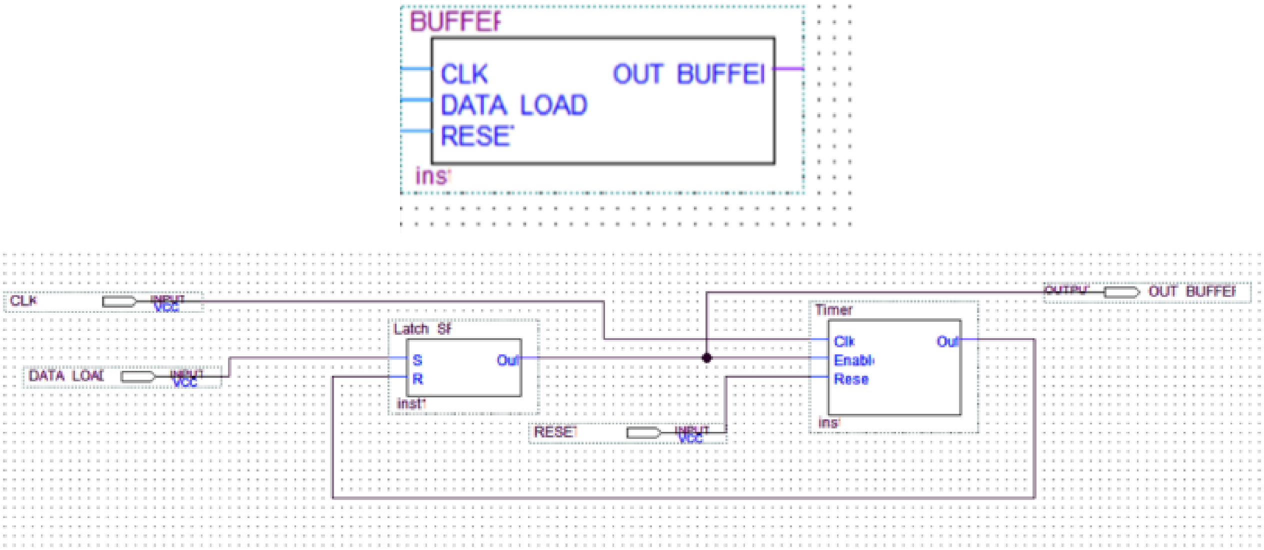
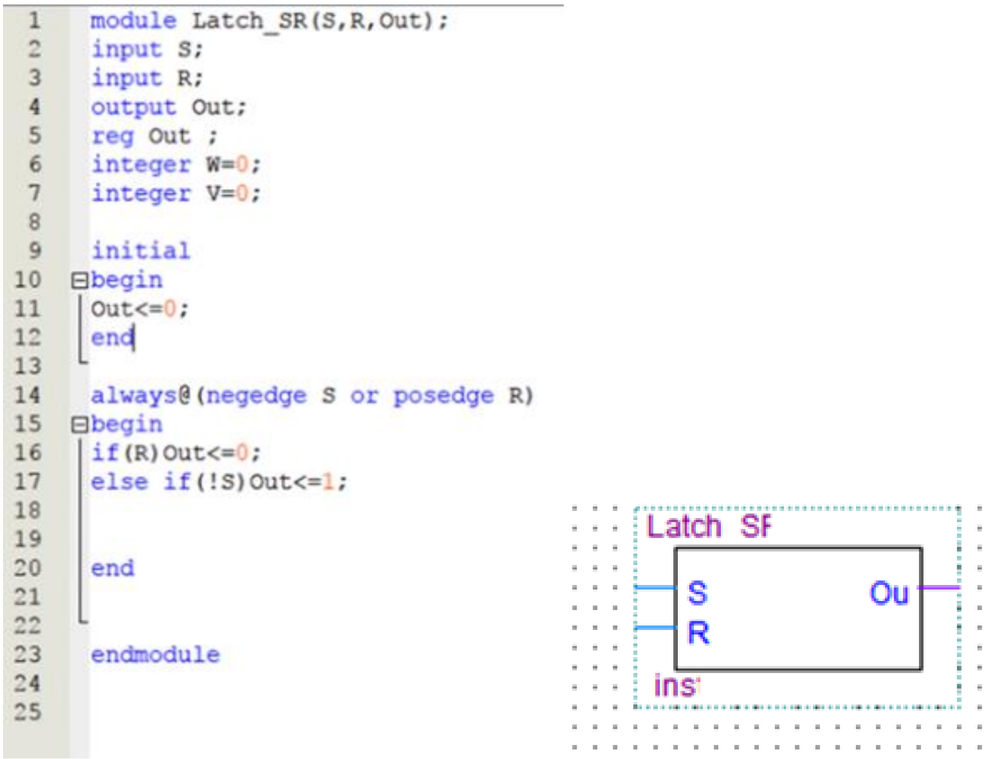


Fig. 9 – Buffer_OUT schematic and symbol file

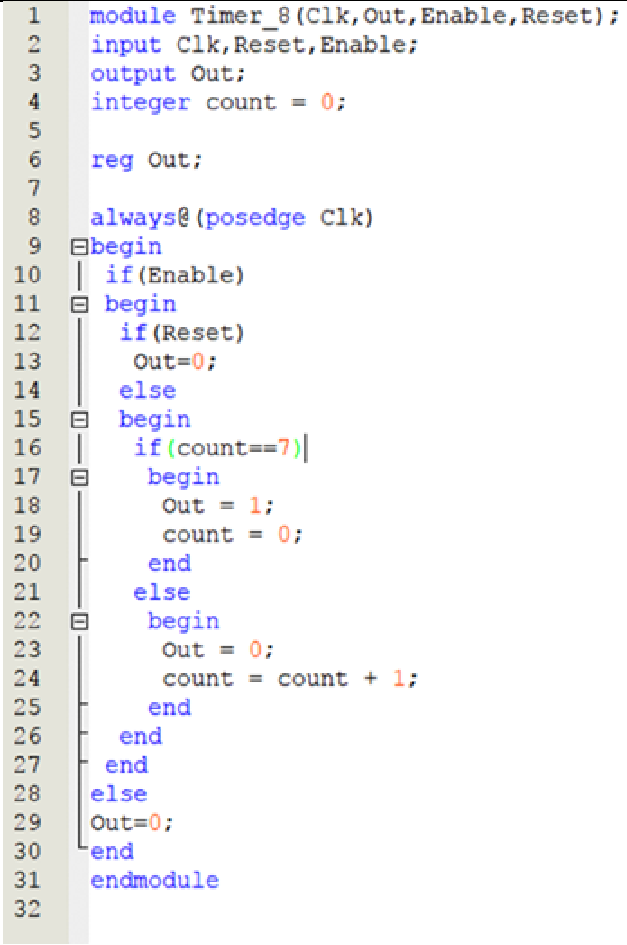
The design of the Set and Reset (SR) latch in Verilog HDL together with its symbol are shown in figure 10



```
module Latch_SR(S,R,Out);
input S;
input R;
output Out;
reg Out ;
integer W=0;
integer V=0;

initial
begin
Out<=0;
end

always@(negedge S or posedge R)
begin
if(R)Out<=0;
else if(!S)Out<=1;


end


endmodule
```

Fig. 10: SR latch design in Verilog HDL and related symbol

The design of the timer in Verilog HDL together with its symbol are in figure 11



```
module Timer_8(Clk,Out,Enable,Reset);
input Clk,Reset,Enable;
output Out;
integer count = 0;

reg Out;

always@(posedge Clk)
begin
 if(Enable)
 begin
  if(Reset)
   Out=0;
  else
  begin
   if(count==7)
    begin
     Out = 1;
     count = 0;
    end
   else
    begin
     Out = 0;
     count = count + 1;
    end
  end
 end
else
Out=0;
end
endmodule
```

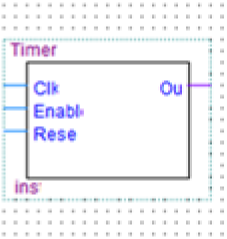


Fig. 11: Timer design in Verilog HDL and related symbol

2.A.1 MCU Data Logger

The UART-to-BELL202 converter needs an interface capable of communicating, through the UART protocol, with a processing system such as a PC. This task is performed by the MICRO DATA LOGGING block, using the NIOSII soft processor configured in the single FPGA chip used.

The block does not act only as UART interface, but is equipped also with some IN/OUT peripherals (PIO) able to generate all the control signals needed to drive the BELL202 transmitter.

The microcontroller block has been designed by using the QSYS tool, a facility integrated in Quartus environment, and the firmware was designed in Eclipse environment.

Figure 12 shows the circuit symbol of the block.

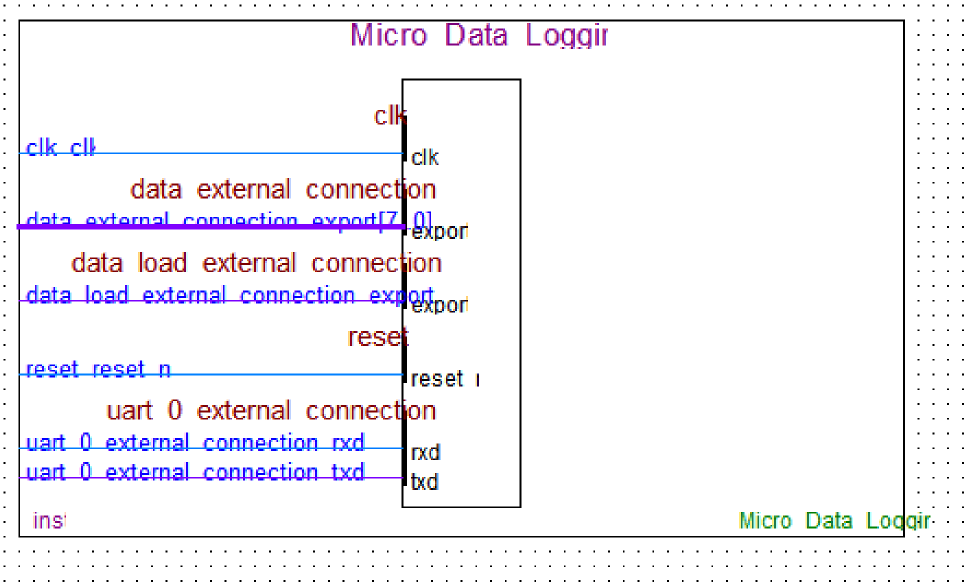

Micro Data Loggir
clk
clk clk
clk
data external connection
data external connection export[7..0]
expor
data load external connection
data load external connection export
expor
reset
reset reset n
reset r
uart 0 external connection
uart 0 external connection rxd
uart 0 external connection txd
rxd
txd
ins
Micro Data Loggir

Fig. 12 - Microcontroller block circuit symbol

The inputs of the module are the following:

CLK is the general system clock provided by the DE1-SoC board;

- Reset_n is the block reset signal, must always be placed at high logic level;
- Uart_0_external_connection_rxd is the reception line of the UART interface.

The outputs are described as follows:

- Data_external_connection_export [7: 0] are the 8 data bits that are sent to the transmitter block;
- Data_load_external_connection is the data load command bit of the transmitter;
- Uart_0_external_connection_txd is the output line of the UART interface.

Attention must be payed to the PIOs named Data and Data_load. These two outputs are respectively composed of 1 and 8 bits. The Data 8 bits are the data received by the UART and rearranged in parallel, while the Data_load bit is the control signal for loading data in the PISO register.

III. DESCRIPTION OF THE FIRMWARE PART OF THE PROJECT

The converter has the task to receive an input UART-protocol string and to modulate it in AFSK.

In order to complete the design, it has been used a PC as UART-protocol string generator, even if the designed converter can be interfaced with any device transmitting bits according to the UART protocol. Therefore, the string is supplied by a PC through a management software of the serial port (PUTTY, for example).

In order to implement the PC hardware connection to the system, a USB-TTL converter is used, as will be better detailed later.

In order to allow the NIOS II processor to perform this task, it has been designed a proper firmware, which flow diagram is shown in figure 12

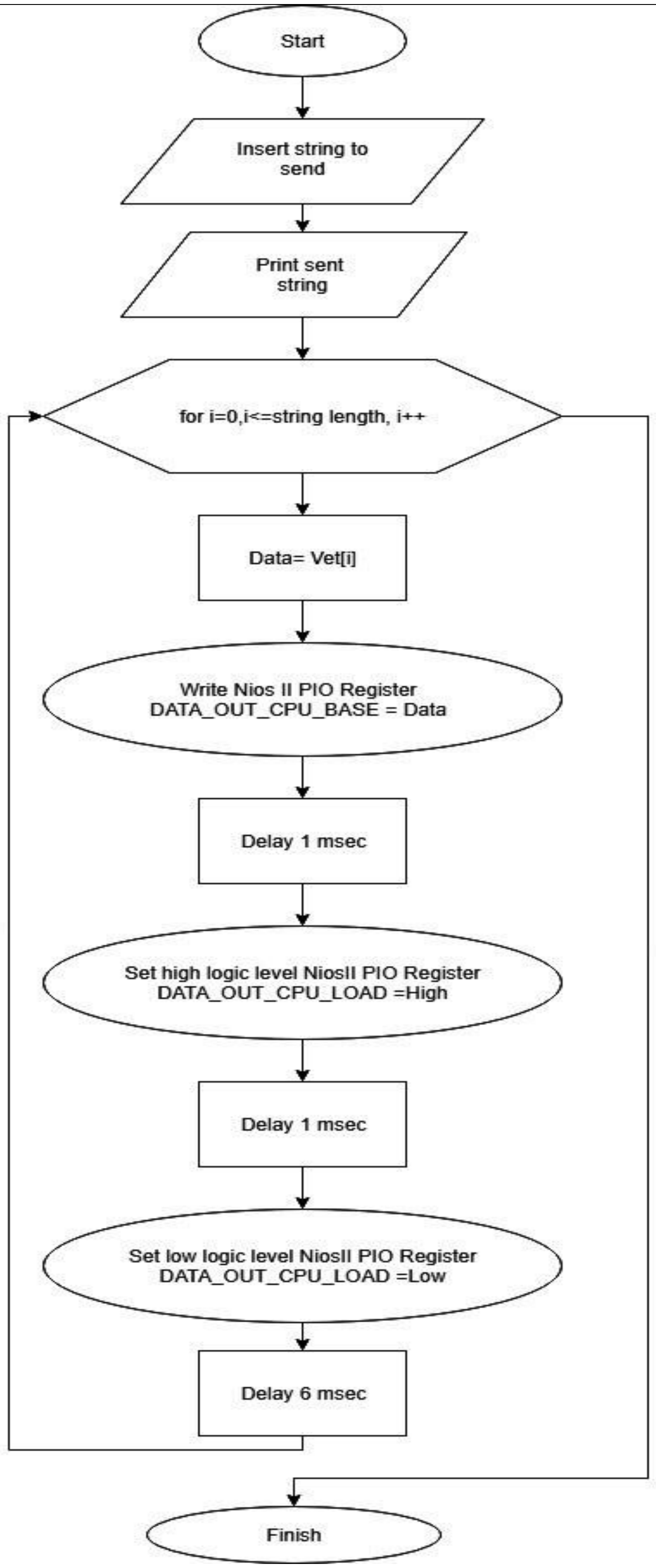

Start
Insert string to send
Print sent string
for i=0,i<=string length, i++
Data= Vet[i]
Write Nios II PIO Register DATA_OUT_CPU_BASE = Data
Delay 1 msec
Set high logic level NiosII PIO Register DATA_OUT_CPU_LOAD =High
Delay 1 msec
Set low logic level NiosII PIO Register DATA_OUT_CPU_LOAD =Low
Delay 6 msec
Finish

Fig. 12 – NIOS II firmware flowchart

The firmware asks the user to insert a string.

Then, it prints the data received and, through a “for” cycle, it sends one by one the characters of the string received from the UART on the PIOs.

At each cycle, the code puts the DATA_LOAD signal at high value, in order to enable the BELL202 transmitter to send data modulated in FSK.

The DATA_LOAD signal must have a shorter time extension than the baudrate period, otherwise the system will not work properly.

IV. FUNCTIONAL TESTS, FIRMWARE DEBUG AND INSTRUMENTAL CHECKS

Figure 13 shows the complete scheme of the converter, while figure 14 shows the pin assignment through the Quartus tool Pin Planner, necessary in order to implement the system on the DE1_SoC development board.

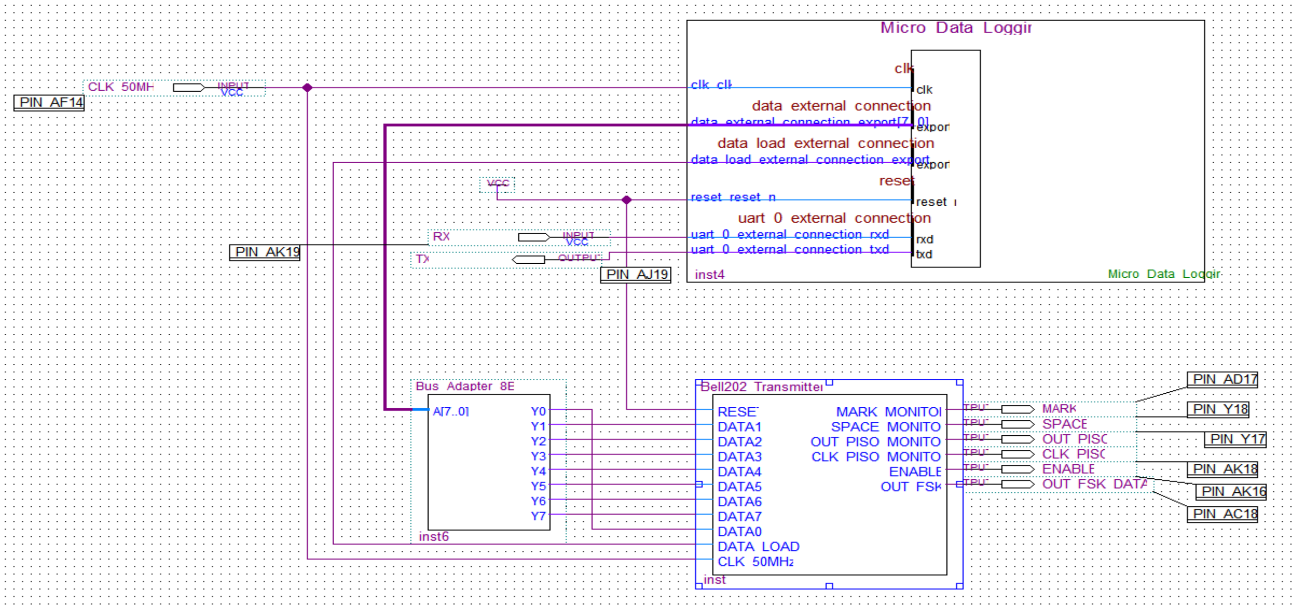
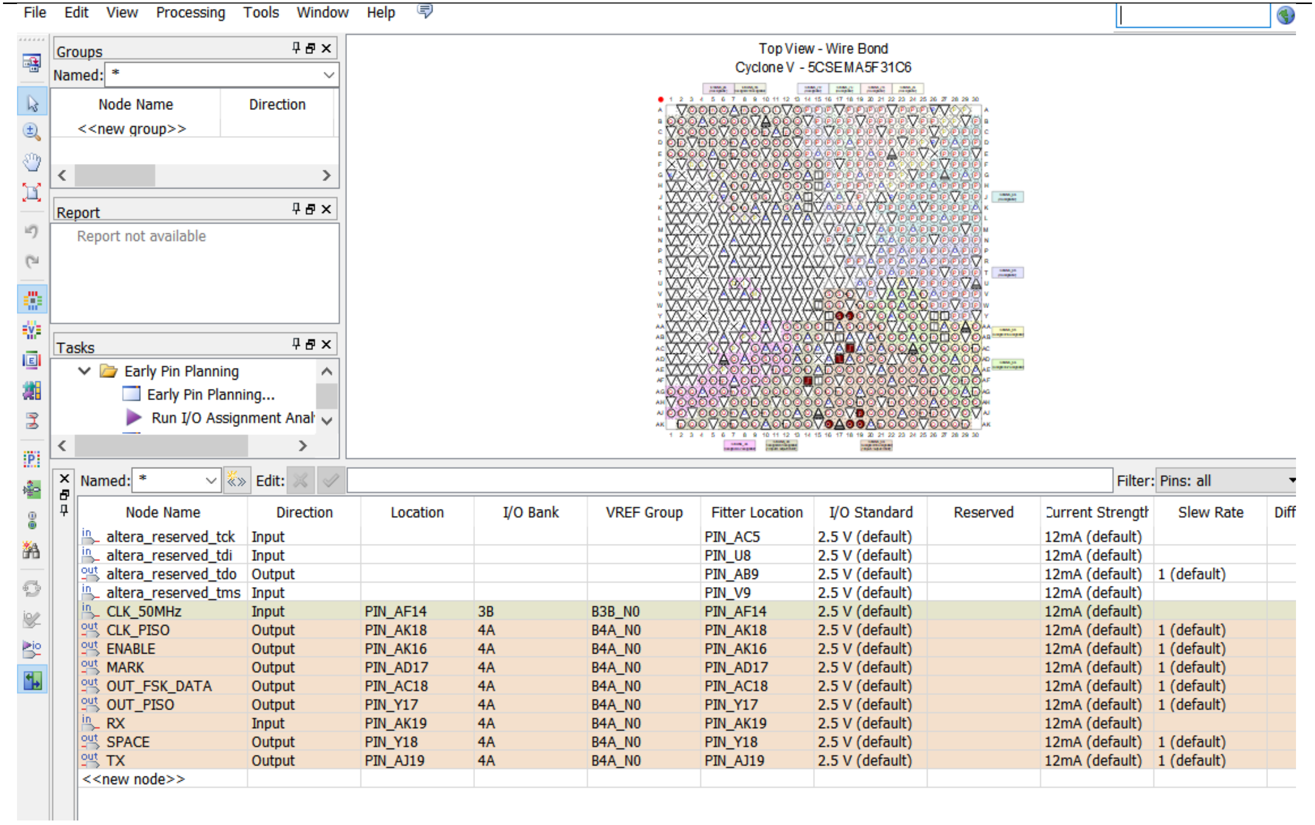


Fig. 13 – Schematic of the designed converter



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Diff
altera_reserved_tck	Input				PIN_AC5	2.5 V (default)		12mA (default)		
altera_reserved_tdi	Input				PIN_U8	2.5 V (default)		12mA (default)		
altera_reserved_tdo	Output				PIN_AB9	2.5 V (default)		12mA (default)	1 (default)	
altera_reserved_tms	Input				PIN_V9	2.5 V (default)		12mA (default)		
CLK_50MHz	Input	PIN_AF14	3B	B3B_N0	PIN_AF14	2.5 V (default)		12mA (default)		
CLK_PISO	Output	PIN_AK18	4A	B4A_N0	PIN_AK18	2.5 V (default)		12mA (default)	1 (default)	
ENABLE	Output	PIN_AK16	4A	B4A_N0	PIN_AK16	2.5 V (default)		12mA (default)	1 (default)	
MARK	Output	PIN_AD17	4A	B4A_N0	PIN_AD17	2.5 V (default)		12mA (default)	1 (default)	
OUT_FSK_DATA	Output	PIN_AC18	4A	B4A_N0	PIN_AC18	2.5 V (default)		12mA (default)	1 (default)	
OUT_PISO	Output	PIN_Y17	4A	B4A_N0	PIN_Y17	2.5 V (default)		12mA (default)	1 (default)	
RX	Input	PIN_AK19	4A	B4A_N0	PIN_AK19	2.5 V (default)		12mA (default)		
SPACE	Output	PIN_Y18	4A	B4A_N0	PIN_Y18	2.5 V (default)		12mA (default)	1 (default)	
TX	Output	PIN_AJ19	4A	B4A_N0	PIN_AJ19	2.5 V (default)		12mA (default)	1 (default)	
<<new node>>										

Fig. 14 - Pin assignment by using the Pin Planner tool in Quartus

Therefore, firstly the hardware part of the project has been compiled and uploaded on the DE1_SoC board. To this aim it is used the Programmer tool of Quartus. This programming procedure allows configuring the hardware of the FPGA according to the specific design.

In order to test the converter, the DE1-SoC needs a connection to a PC via a TTL-USB converter, as shown in figure 15. The connection pattern is shown in figure 16.

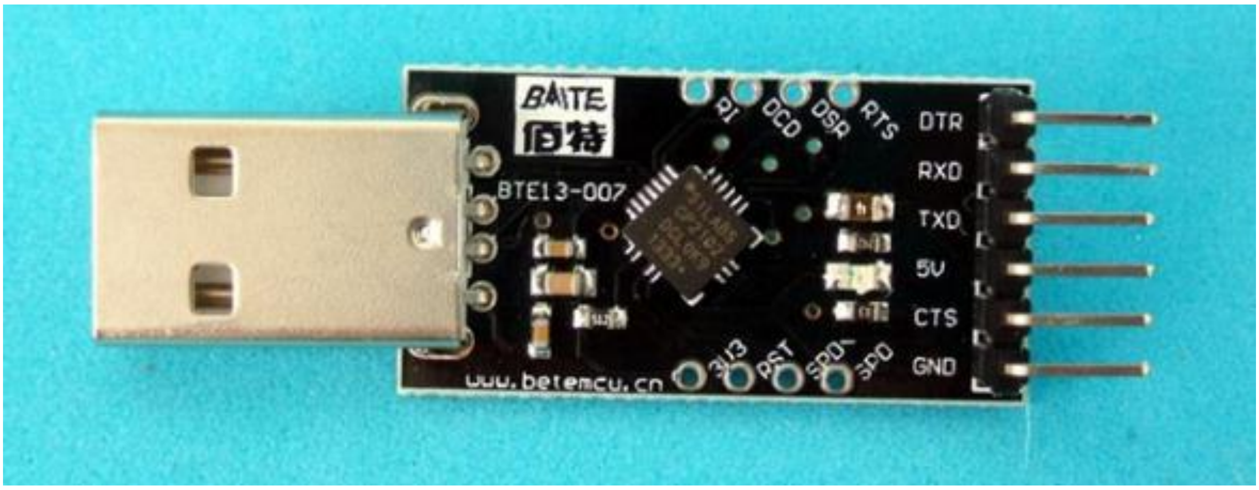


Fig. 15 - USB-TTL converter

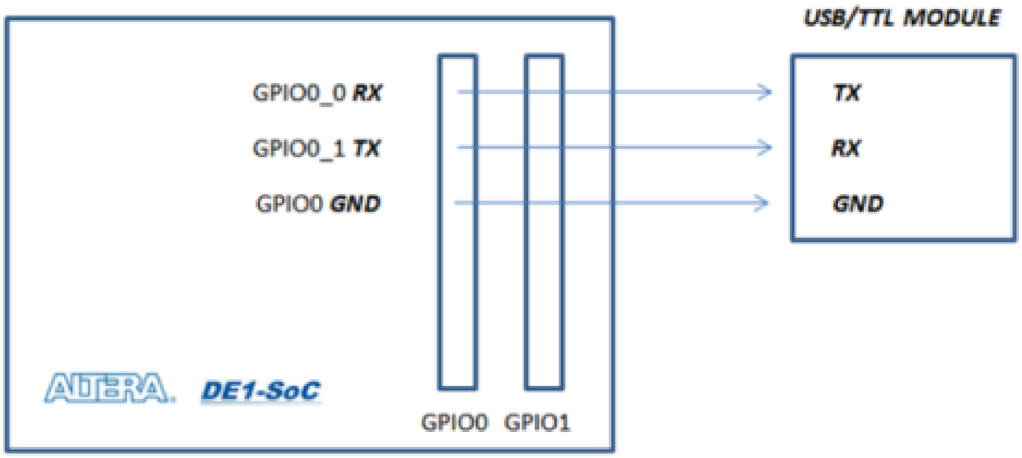


Fig. 16 - Connection scheme between the DE1_SoC and the USB-TTL converter

The photo of the system under test properly connected to a PC (notebook) is shown in figure 17.

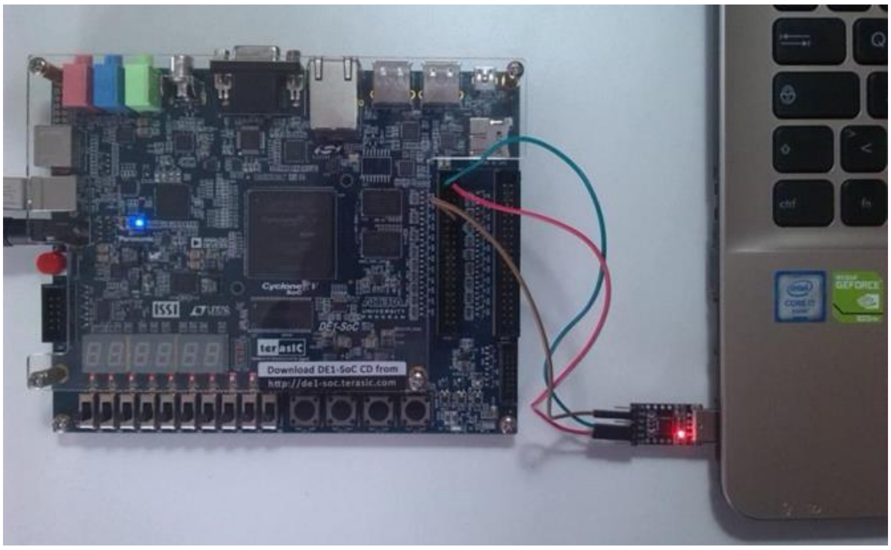


Fig. 17 – Photo of the system under test

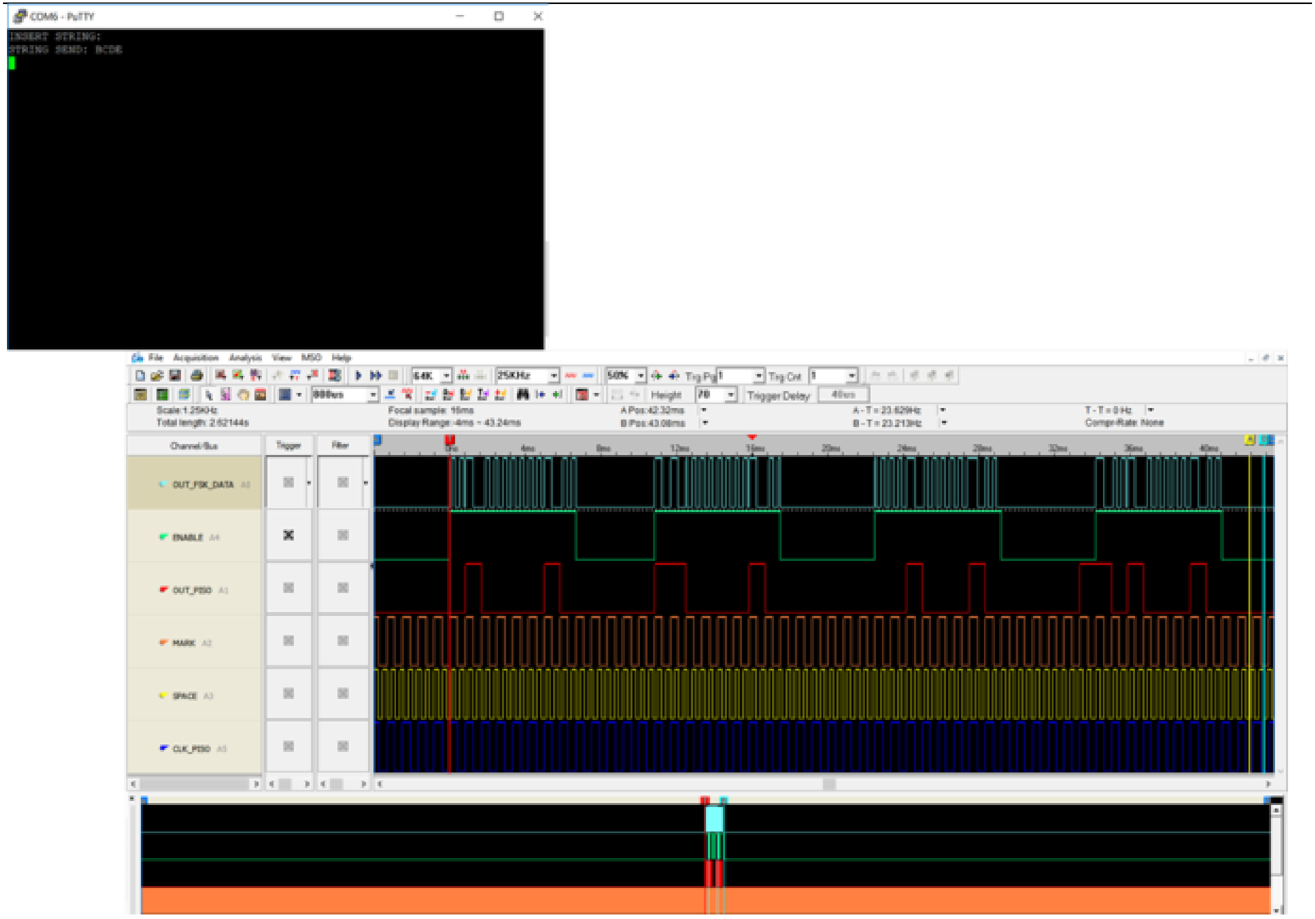
Then, a new BSP (Board Support Package) project has been created using the tool Eclipse, dealing with the firmware part of the project. The firmware is in C code.

Once the embedded system has been fully implemented on the DE1_SoC board, in its hardware and firmware parts, some debug procedures have been performed in order to check for the correct functioning.

The software interface used is the Putty serial terminal. The tools used for the experimental checks were the ZEROPLUS LAP-C logic states analyzer and the SIGLENT CML1102 digital oscilloscope.

In order to perform tests, it is necessary to open the Putty software. As Putty starts running, it is required to insert the COM port and the baud rate (115200).

After setting, the Putty terminal will require to write a string, also printed on screen. Figure 18 shows the Putty screen for inserting the "BCDE" string and the relative detection with logic analyzer.



```
INSERT STRING:
STRING SEND: BCDE
```

Scale:1.25KHz
Total length: 2.62144s
Focal sample: 16ms
Display Range: -4ms ~ 43.24ms
A Pos:42.32ms
B Pos:43.08ms
A - T = 23.629Hz
B - T = 23.213Hz
T - T = 0 Hz
Compr.Rate: None

Fig. 18 – System under test by using a logic analyzer and the Putty software

Figure 19 shows the verification carried out by using the logic analyzer when is sent the ASCII character "B", corresponding to the binary word 01000010. The names of the analyzed signals that correspond to the outputs of the BELL202 transmitter implemented in Quartus are at the left side of figure 19. In particular:

OUT_FSK_DATA is the output of the signal modulated in AFSK;

ENABLE is the signal that provides the measurement of the time window dealing with each 8-bit packet coming from the UART;

MARK / SPACE are the outputs of the mark and space frequency dividers, respectively;

OUT_PISO is the serial output generated by the PISO register;

CLK_PISO is the clock used by the register.

The measurement of the time intervals between the couple of cursors A-T and B-T are highlighted by a circle on the upper side of figure 19.

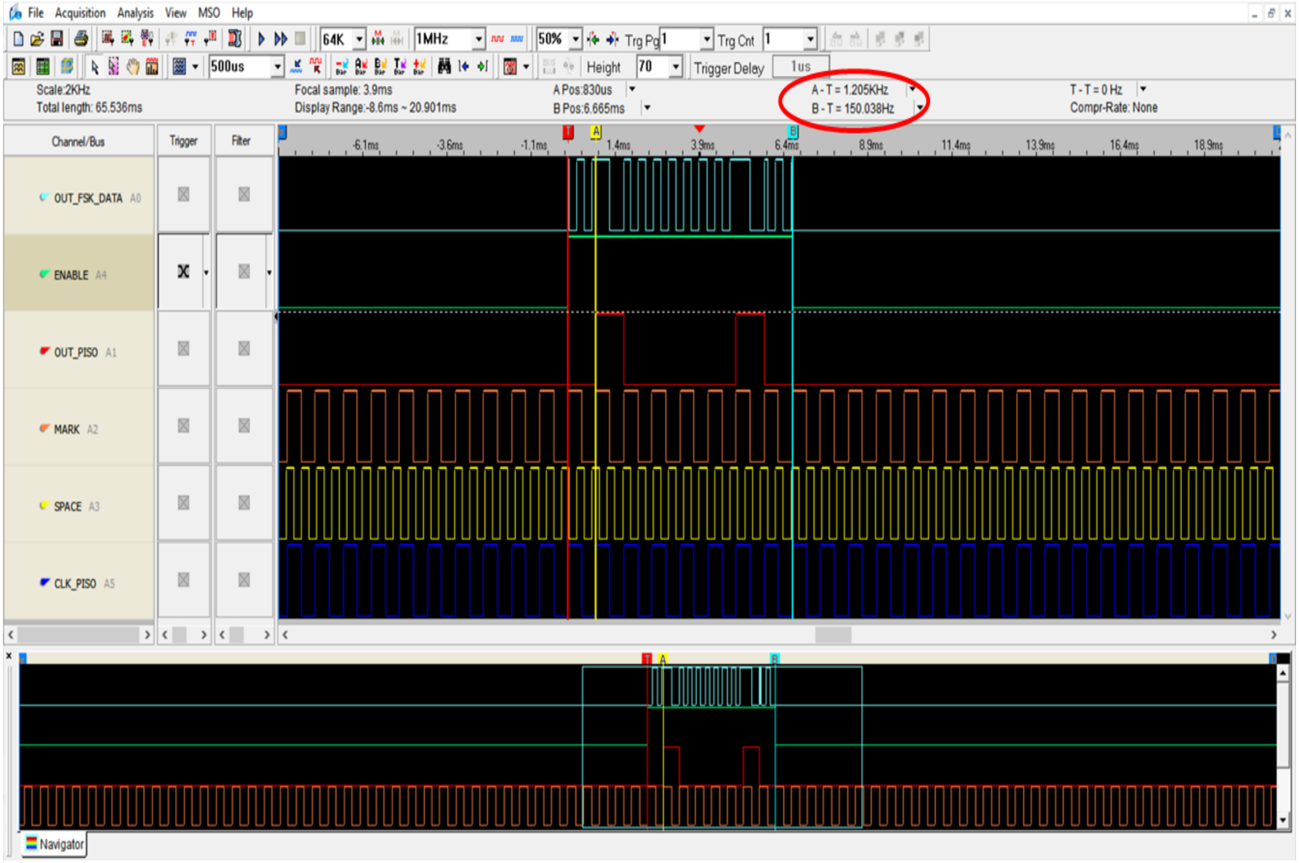
The A-T value is 1205 Hz as expected since the baudrate of the transmission is equal to 1200 baud, equivalent to 1200 Hz because each symbol is equal to one bit.

The B-T value is equal to the inverse of the period of the whole data package that is 0.0067 sec (150Hz).

Therefore, the correct synchronization and timing of the system are verified.

The right logic functionality of the system can be verified evaluating the PISO_OUT and OUT_FSK signals. In particular, the value of the PISO_OUT signal is equal to 01000010, which corresponds exactly to the binary ASCII code of the character sent that is "B" (01000010).

Figure 20 shows measurements carried out with a digital oscilloscope. The measure of the duration of the whole 8 bit data packet is equal to 0.0067 sec (ENABLE signal, lower side in figure 20). The signal at the upper side of figure 20 is the data packet modulated in AFSK.



64K
1MHz
50%
Trg Pg 1
Trg Cnt 1
500us
Height 70
Trigger Delay 1us

Scale:2KHz
Total length: 65.536ms
Focal sample: 3.9ms
Display Range:-8.6ms ~ 20.901ms
A Pos:830us
B Pos:6.665ms
A - T = 1.205KHz
B - T = 150.038Hz
T - T = 0 Hz
Compr-Rate: None

Fig. 19 – Testing of the system with the logic state analyzer sending the ASCII character "B" to the system (01000010)

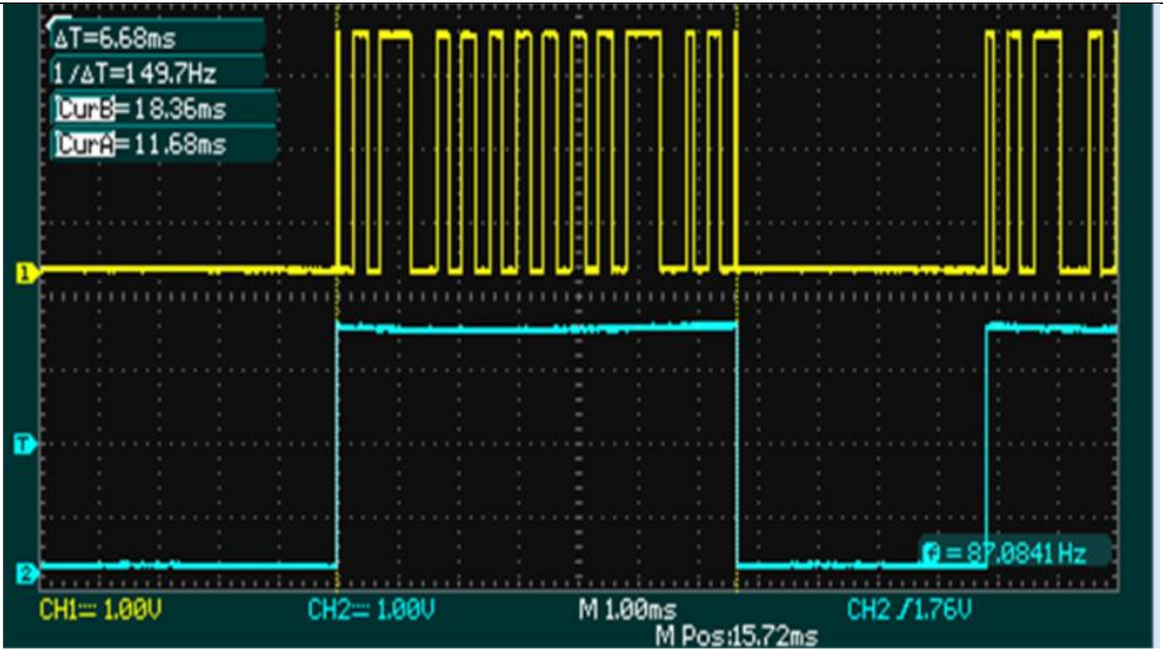


Fig. 20 - Time analysis of the FSK output and Enable output by sending to the system the ASCII character "B" (01000010)

The system is also capable to encode ASCII strings received from the PC keyboard, to calculate their length and then send one character at a time to the FSK modulator automatically, without any user’s intervention.

Figure 21 shows the detection of the ASCII string "BCDE" performed by using the logic states analyzer. It can be observed that the system sends four data packets of 8 bit each one, corresponding to the ASCII codes of the characters entered (01000010- 01000011-01000100-01000101). The PISO_OUT signal has the MSB firstly encoded, so its graphical display is overturned in comparison to its binary encoding.

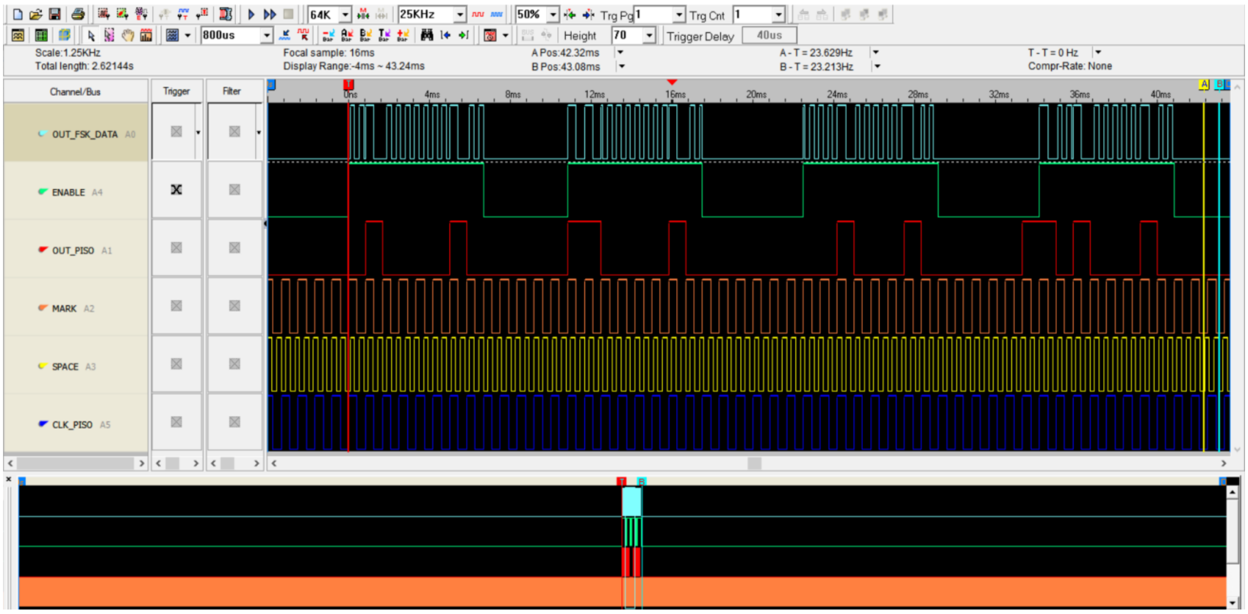


Figure 21 - Detection with logic state analyzers of the system functions by sending the ASCII string "BCDE" to the system (01000010- 01000011-01000100-01000101)

Figures 22 and 23 show the measurement with digital oscilloscope of the AFSK modulated string "BCDE", shown in figure 21. The correct functionality of the system is verified, once more.

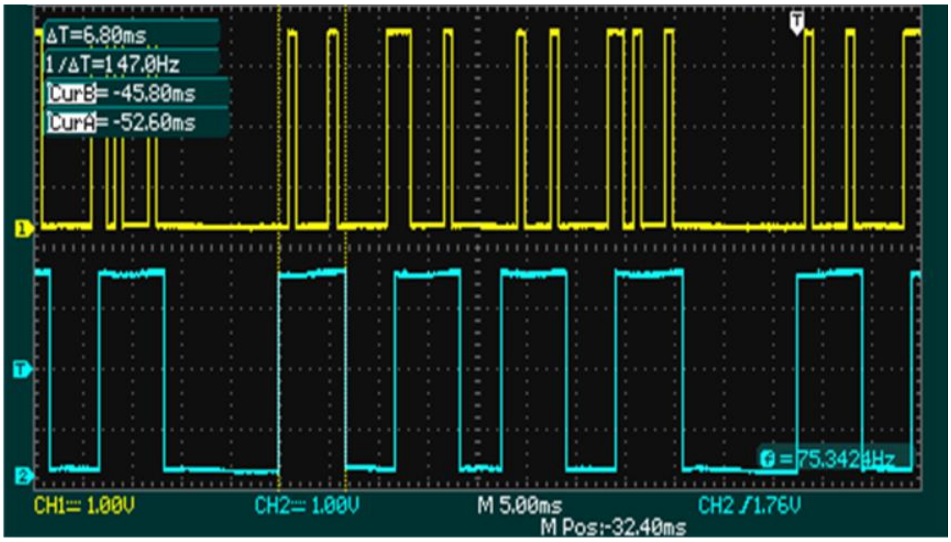


Fig. 22 - Time analysis of the FSK output and Enable output with sending to the system of the ASCII string "BCDE" (01000010- 01000011-01000100-01000101)

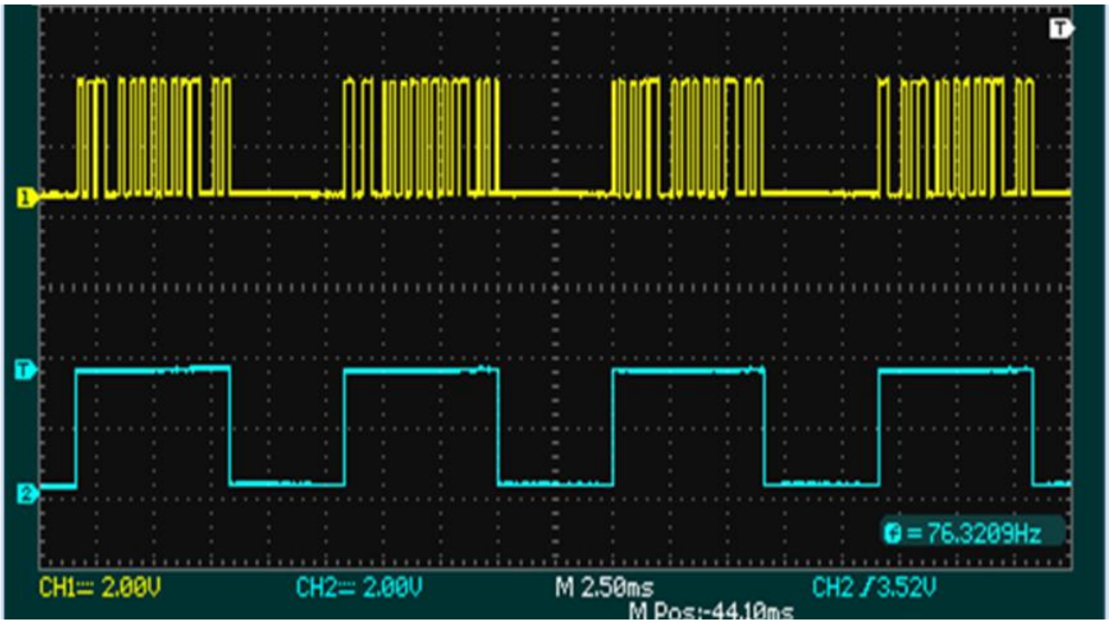


Fig. 23 - Time analysis of the FSK output and Enable output with sending to the system of the ASCII string "BCDE" (01000010- 01000011-01000100-0100010

V. CONCLUSIONS AND FUTURE DEVELOPMENTS

In this paper it has been described the design of a UART –to- Bell202 converter as SoC into Intel FPGA.

The converter has been implemented and successfullt tested on the DE1_SoC development board, powered by Terasic, and mounting a Cyclone V FPGA.

The designed converter is in fact a UART-AFSK modulator and it is easy to fit it to all data transmission needs with FSK without phase continuity. This is possible because the frequency dividers, that generate the frequencies of MARK and SPACE signals, are fully configurable: changing the parameter K, you can generate countless frequencies.

Furthermore, by adding an Altera PLL to the system, high-speed data transmissions can be supported.

The results show that it is possible through an FPGA, to obtain an audio band modulator on a digital carrier maintaining high precision in the frequency band occupation.

The results obtained also demonstrate the potential of these devices and in particular of the Soft CPUs powered by the Intel FPGA (Nios II). In fact, through a firmware modification, the transmission speed could be increased or interacted with electronic systems for analogue telecommunications.

The design also paves the way for integration on the same chip of digital systems that can be used for civil and military radio communications.

Conflict of interest: The authors declare that they have no conflict of interest.

Ethical statement: The authors declare that they have followed ethical responsibilities.

REFERENCES

[1] ITU-T V23 Recommendation 600/1200 Baud Modem Standardized use in general switched telephone network. International Telecommunication Union.

[2] Bob Bruninga "APRS International Organization" [Online]. Available: http://www.aprs.org/ .

[3] HART Communication Made Easy Analog Device USA 2012.

[4] Wu ChangFu "Analysis, Design and Implementation of FSK Modulate Systems" in Proceedings of the 2nd International Conference On Systems Engineering and Modeling (ICSEM-13) 2013.

[5] Haytham Azmi, Hamed Elsimary, M Ibrahim Youssef, Ahmad Safwat "FPGA based multi-standard configurable FSK demodulator" Integration the VLSI Journal 36(3):145-154.

[6] G.Bouzid H.Trabelsi "FPGA Implementation of FHSS-FSK modulator" 3rd International Conference on Design and Technology of Integrated System in Nanoscale 2008 IEEE.

[7] Ahmed Asim Ghouri "FPGA based FSK/PSK modulation" EDN Network December 11, 2014.

[8] Raytheon Co, Wayland Mass "The Performance of FSK Permutation Modulations in Fading Channels and Their Comparison Based on A General Method for the Comparison of M-ary Modulations" Vol. p.616-640 IEEE Transactions on Communication Technology;

[9] SoC FPGA ARM Cortex A9 MPCore Processor Advance Information Brief. Altera Corporation USA February 2012.

[10] Altera, Nios II processor Intel Corporation USA February 2017.

[11] A.Giorgio, F.Paris "Guida all'implementazione su FPGA di Sistemi Elettronici Digitali basati su NIOS II" 1st ed. GEDI Gruppo Editoriale S.p.A 2017 Italy.

[12] Pong P. Chu "Embedded SopC Design with NiosII Processor and Verilog Example" ed. 2012. WILEY & Sons Publication

[13] S.Palnitkar, "Verilog HDL:A Guide To Design and Sinthesys (2nd ed)" Prentice Hall, the 1st ed. March,3,2003.