ISSN 2229-5518

FPGA-Based Design of Controller for Sound Fetching from Codec Using Altera DE2 Board

A.R.M. Khan, A.P.Thakare, S.M.Gulhane

ABSTRACT— The trend in hardware design is towards implementing a complete system, intended for various applications, on a single chip. In order to implement the any speech application in Altera DE2 board a controller is designed to control the CODEC and acquire the digital data from it. This paper presents an experimental design and implementation of the controller using the specification given by the Philips for I2C protocol & DSP mode of operation of CODEC on cyclone-II EP2C35F72C6 FPGA in Altera DE2 board . A controller was designed using VHDL language, which performs the two operations: I2C protocol operation to drive the Wolfson Codec WM8731, sound fetching from Wolfson Codec WM8731 to FPGA in DSP mode. Altera Quartus II 9.0 sp2 web Edition is used for the synthesis of the VHDL logic on FPGA and ModelSim-Altera 6.5b (Quartus II 9.1) Starter Edition is used for the simulation of VHDL logic. Three modules have been created in the design: the I2C bus controller, virtual sound fetcher, and the clock module. The FPGA communicates with the Wolfson via the I2C (Inter-Integrated Circuit) protocol using two pins: 'SDIN' (the data line), and 'SCLK' (the bus clock). I2C bus controller modifies internal settings of Codec, de-mute the microphone input, boost the microphone volume, and change the default sound path (so that the microphone is given priority over other inputs). After the codec digitalizes the input it put the digital data on digital audio interface, to fetch the data on DACDAT of codec form digital audio interface DSP mode of operation of codec is used in the design. DACDAT is the formatted digital audio data stream with left and right channels multiplexed together. DACLRC (alignment clock) and BCLK (synchronization clock) is used to fetch the data on DACDAT this data can be use for any sound application. Clock module is design to generate different clock requirement for the controller.

Keywords— I^2C bus controller, Quartus II, ModelSim, Codec, Virtual Sound fetcher.

I. INTRODUCTION

FPGA becomes one of the most successful of today's technologies for developing the systems which require a real time operation. The term field Programmable highlights the customizing of the IC by the user, rather than by the foundry manufacturing the FPGA. Several researchers discussed the design of hardware systems. Numbers of these works were specialized in designing the controllers for different application, and were aim to get better control responses, FPGA are two dimensional arrays of logic blocks and flip-flops with an electrically programmable interconnection between logic blocks. The interconnections consist of electrically programmable switches which is why FPGA differs from Custom ICs, as Custom IC is programmed using integrated circuit fabrication technology to form metal interconnections between logic blocks. In an FPGA logic blocks are implemented using multiple level low fan in gates, which gives it a more compact design compared to an implementation with two-level AND-OR logic. FPGA provides its user a way to configure and these specifications. Altera DE2 board become one of the most widely development FPGA board which is used to development of FPGA design and implementations, the board offers a rich set of features that make it suitable for use in a laboratory environment for university and college courses, for a variety of design projects, as well as for the development of sophisticated digital systems. Software provided with the DE2 board features the Quartus® II Web Edition CAD system, and the Nios® II Embedded Processor. Also included are several aids to help students and professionals experiment with features of the board, such as tutorials and example applications. Traditionally, manufacturers of educational FPGA boards have provided a variety of hardware features and software CAD tools needed to implement designs on these boards, but very little material has been offered that could be used directly for teaching purposes. Altera's DE2 board is a significant departure from this trend.

II. DE2 DEVELOPMENT BOARD

Altera DE2 board become one of the most widely development FPGA board which is used to development of FPGA design and implementations [5]. The purpose of the Altera DE2 Development and Education board is to provide the ideal vehicle for learning about digital logic, computer organization, and FPGAs. It uses the state-of the-art technology in both hardware and CAD tools to expose students and professionals to a wide range of topics. The board offers a rich set of features that make it suitable for use in a laboratory environment for university and college courses, for a variety of design projects, as well as for the development of sophisticated digital systems. Altera provides a suite of supporting materials for the DE2 board; including tutorials, "ready-to-teach" laboratory exercises, and illustrative demonstrations [6] Fig. 1 gives the block diagram of the DE2 board. To provide maximum flexibility

for the user, all connections are made through the Cyclone II FPGA device. Thus, the user can configure the FPGA to implement any system design.

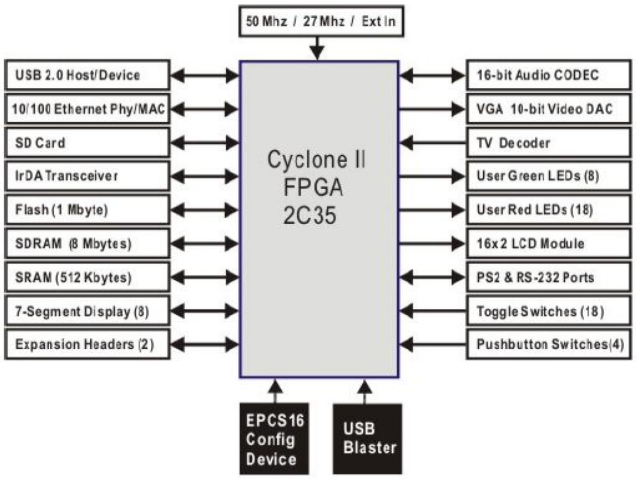

Fig. 1: Block Diagram of DE2 Board [6]

III. Structure Of Controller

In order to implement the any speech application controller is design in the Altera DE2 board shown in the Fig. 2 the design is broken down into modules. These are then mapped to combinational logic and finite-state machines (FSM); using the Quartus II software package .Three modules have been created: I²C bus controller, Virtual sound fetcher, Clock module.

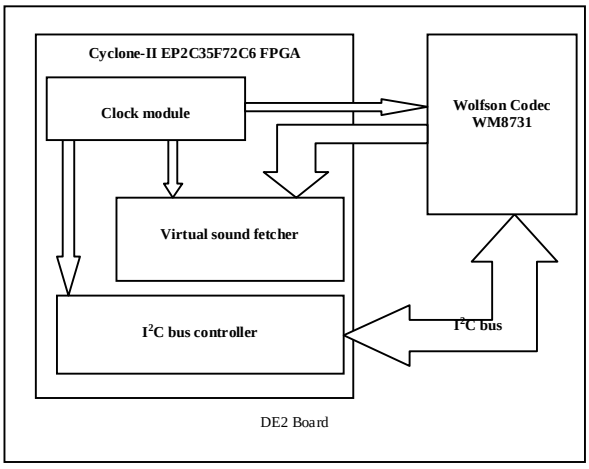

Fig. 2: Block Diagram of controller

A. *I²C bus controller*

The FPGA communicates with the Wolfson via the I²C (Inter-Integrated Circuit) protocol using two pins: 'SDIN' (the data line), and 'SCLK' (the bus clock) Fig. 3 shows the complete requirement of data transmission through I²C .Within the procedure of the I²C bus, unique situations arise which are defined as START (S) and STOP (P) conditions. START means a HIGH to LOW transition on the SDIN line while SCLK is HIGH & STOP means a LOW to HIGH transition on the SDIN line while SCLK is HIGH .The master always generates START and STOP conditions. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time after the STOP condition. The bus stays busy if a repeated START (Sr) is generated instead of a STOP condition. In this respect, the START (S) and repeated START (Sr) conditions are functionally identical. The S symbol will be used as a generic term to represent both the START and repeated START conditions, unless Sr is particularly relevant. Detection of START and STOP conditions by devices connected to the bus is easy if they incorporate the necessary interfacing hardware. The content s of the data line are sent in the same order as seen in Fig. 3 (after a start condition): 'RADDR', 'R/W', 'ACK','DATAB[15-9]', and 'DATAB[8-0]', which stand respectively for “base address”, “Read/Write”, “acknowledge”, “control address”, and “control data”. The clock for the control serial data input is SCLK. The maximum frequency for this signal is 400 kHz, the frequency used in this paper is f=50 MHz/128 = 390.625 kHz.

The data input signal is SDIN, it contains the information for the control interface [9]. The controller was reprogrammed using the 2-wire interface with the EP2C35F672C6 FPGA in Cyclone II board. After the start condition that is a falling edge on SDIN while SCLK is high. The following seven bits determines which device receives the data, the address depends on the CBS state (set to ground in this codec) so it is “0011010”. After this address, the bit R/W determines the direction of data transfer, in this case a ‘0’indicates ‘write’. The device recognizes the address and R/W by pulling SDIN low during the ninth clock cycle, acknowledging the data transfer. The control follows with two bit blocks (separated with another

acknowledge) the first block B [15:9] contains the control address bits, and the second block B [8-0] contains the control data bits. The stop condition after the data transfer is a rising edge on SDIN when SCLK is high. If a start condition is detected out of the sequence at any point in the data transfer then the device will jump to the idle condition. After a complete control operation, the audio codec returns to the idle state and waits for another start condition.[9],[10]

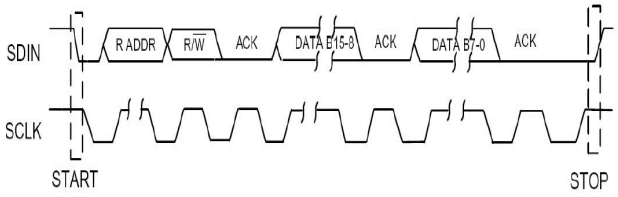

Fig. 3: Requirement of data transmission through I²C

B. Virtual Sound fetcher

The WM8731 or WM8731L (WM8731/L) are low power stereo CODECs with an integrated headphone driver. The WM8731/L is designed specifically for portable MP3 audio and speech players and recorders. The WM8731 is also ideal for MD, CD-RW machines and DAT recorders. The on-board digital to analogue converter (DAC) accepts digital audio from the digital audio interface. Digital filter de-emphasis at 32 kHz, 44.1 kHz and 48 kHz can be applied to the digital data under software control. The DAC employs a high quality multi-bit high-order oversampling architecture to again deliver optimum performance with low power consumption. The DE2 board provides high-quality 24-bit audio via the Wolfson WM8731 audio CODEC (enCOder/DECoder). This chip supports microphone-in, line-in, and line-out ports, with a sample rate adjustable from 8 kHz to 96 kHz. The WM8731 is controlled by a serial I²C bus interface, this is connected to pins on the Cyclone II FPGA. A schematic diagram of the audio circuitry is shown in Fig. 4, and the FPGA pin assignments are listed in Table 1.[6],[7]

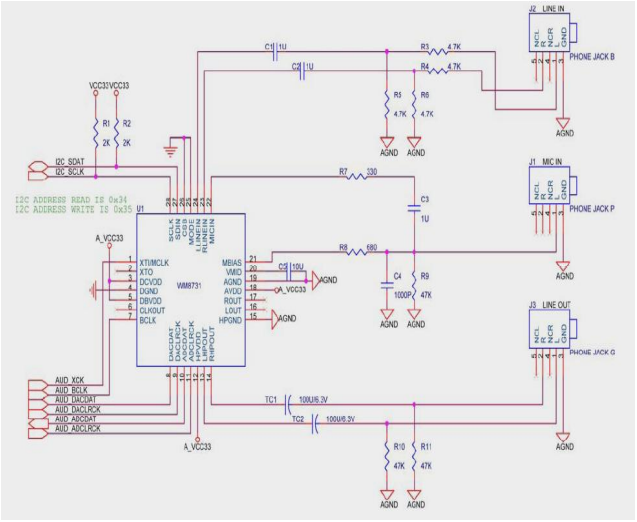

Fig. 4 Audio CODEC Schematic

TABLE.1
FPGA PIN ASSIGNMENTS

Signal Name	FPGA Pin No.	Description
AUD_ADCLRCK	PIN_C5	Audio CODEC ADC LR Clock
AUD_ADCDAT	PIN_B5	Audio CODEC ADC Data
AUD_DACLRCK	PIN_C6	Audio CODEC DAC LR Clock
AUD_DACDAT	PIN_A4	Audio CODEC DAC Data
AUD_XCK	PIN_A5	Audio CODEC Chip Clock
AUD_BCLK	PIN_B4	Audio CODEC Bit-Stream Clock
I2C_SCLK	PIN_A6	I2C Data
I2C_SDAT	PIN_B6	I2C Clock

1) Slave Mode of Operation: As a slave device the WM8731/L sequences the data transfer (ADCDAT, DACDAT) over the digital audio interface in response to the external applied clocks (BCLK, ADCLRC, DACLRC). Note that the WM8731/L relies on controlled phase relationships between audio interface BCLK, DACLRC and the master MCLK.[8]

2) Digital Audio Interfaces: WM8731/L may be operated in either one of the 4 offered audio interface modes. All four of these modes are MSB first and operate with data 16 to 32 bits. These are:

- Right justified
- Left justified

- I²S
- DSP mode

3) Digital Audio Interface Formats in DSP Mode: There are four digital audio interface formats accommodated by the WM8731/L. DSP Mode format is shown in the Fig. 5 below. DSP mode is where the left channel MSB is available on either the 1st or 2nd rising edge of BCLK (selectable by LRP) following a LRC transition high. Right channel data immediately follows left channel data. To accommodate system timing requirements the interpretation of BCLK maybe inverted, this is controlled vias the software. This is especially appropriate for DSP mode.[8]

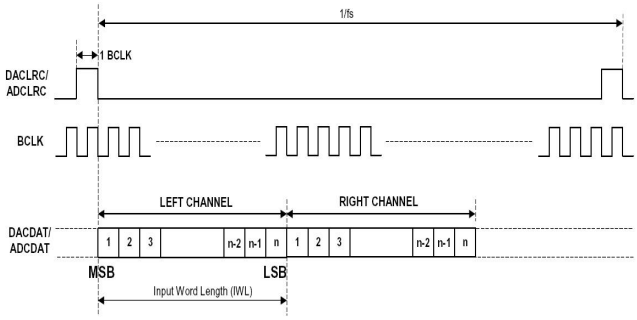

Fig. 5 DSP Mode format of CODEC

IV. Designing Of Controller

A. I²C bus controller

For implementing the controller, finite-state machine (FSM) is used. However, even without any knowledge in digital logic design, you can still very easily implement a FSM by writing VHDL code. VHDL is popular hardware description languages (HDL) for designing digital circuits. A finite-state machine is a sequential circuit that uses a finite number of states to keep track of its history of operations, and based on this history and its current inputs, determine what to do next. The state variable declared using the SIGNAL keyword is the state memory. It is of type STD_LOGIC_VECTOR, which is an 8-bit bit string. The PROCESS block specifies that whenever there is a change in either of the two signals, clk and rst, the statements inside the block will be executed in sequential order starting with the first line. We have an active high reset signal as specified in the IF statement that tests for the signal being a 1. When rst is deasserted, i.e., when rst is equal to 1, the module goes into the reset mode and outputs logic 1 value for both the SDIN and SCLK output signals. Furthermore, it assigns state x"00" as the initial state for when the FSM starts. x"00" is the syntax for the two hexadecimal digits 00. When rst is asserted, the ELSIF statement is executed.

The condition, Clock' EVENT AND Clock = '1', specified inside the ELSIF statement checks for a rising clock edge. So at every rising clock edge, the FSM will go to a new state and a new set of output signals will be generated [4] , To get 400kHz for I²C standard; every 2 cycles of 800kHz = 1 I²C cycle by using crystal frequency 50MHz/64 = 781kHz .

The clock for the control serial data input is SCLK maximum frequency for this signal is 400KHz that we are generating from 800KHz clock. The implementation of SDIN signal requires some special attention as it is bidirectional and open drain so to output the logic 1 on this line ,we need to set this to high impedance, to get high impedance we need to use tri-state output and assign to it a 'Z' value. The condition signal assignment statement use in VHDL program is [1],[4]

SDIN <= 'Z' WHEN SDIN01 ='1' ELSE '0'

B. Virtual Sound fetcher

Operating the digital audio interface in DSP mode allows ease of use for supporting the various sample rates and word lengths. The only requirement is that all data is transferred within the correct number of BCLK cycles to suit the chosen word length. In Slave mode, DACLRC and ADCLRC inputs are not required to have a 50:50 mark-space ratio. BCLK input need not be continuous.[8]

It is however required that there are sufficient BCLK cycles for each DACLRC/ADCLRC transition to clock the chosen data word length. The non-50:50 requirements on the LRCs is of use in some situations such as with a USB 12MHZ clock. Here simply dividing down a 12MHz clock within the DSP to generate LRCs and BCLK will not generate the appropriate DACLRC or ADCLRC since they will no longer change on the falling edge of BCLK.[8]

For example, with 12MHz/32kfs mode there are 375 MCLK per LRC. In these situations DACLRC/ADCLRC can be made non 50:50. The 12MHz Clock is generated using 50MHz crystal oscillator available in DE2 board.

V. Experimental Results

A. ModelSim- Altera 6.5b (Quartus II 9.1) (Results)

This is a program to simulate VHDL code. The VHDL files are complied, the input signals are forced to the

correspondent values and the evolution of the signals is shown in waves and list. The simulation result of the I²C Controller with start & stop condition is shown in Fig. 7 and simulation result of the complete operation virtual sound fetcher is shown in Fig. 8

B. Quartus II (9.0)Sp2 Web Edition (Results) :

This is the main program and it is used to build the projects. It has many tools to prepare VHDL or VERILOG structures and download them to the board. The assignment of the pins is done with the *Quartus II Assignment Editor*, after this the VHDL files can be compiled and downloaded to the board with the *Programmer tool.*[2]

In this project, JTAG programming (Joint Test Action Group) is used so the configuration bit stream is downloaded directly into the FPGA that will retain the configuration as long as the power is applied to the board. Pin assignment used in the design is as shown Table.1 .You can view the internal structure of the design netlist technology map viewer. To view the post-fitting view of the schematic, you must first perform a full compilation, and then choose the, either after fitting or after Analysis & Synthesis, with the technology map viewer command.

To view the post-mapping view of the schematic, you must first perform Analysis & Synthesis, and then choose the technology map Viewer (Post-Mapping) command.

Synthesized design of the I²C Controller.(RTL view) is as shown in Fig. 9 and synthesized design of virtual sound fetcher is shown in Fig. 10 and Fig. 11 shows complete Synthesized Design of virtual sound fetcher with I²C specification (RTL view).

C. Pin Assignment of FPGA

Quartus –II requires the pin assignment for the actual implementation of the design on FPGA in cyclone –II 475 I/O pins are available out of which 11 pin are used for the port mapping in the design. Cyclone –II classify the pins into I/O bank and the assignment editor in Quartus –II provide the I/O standard general function and special function of the assign pin detail of Quartus-II window is as shown in the Fig. 6 .[11]

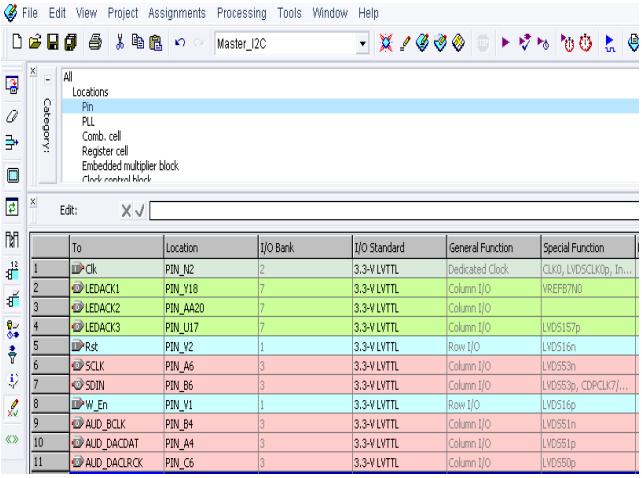

Fig. 6 FPGA Pin Mapping for the Design in Quartus-II

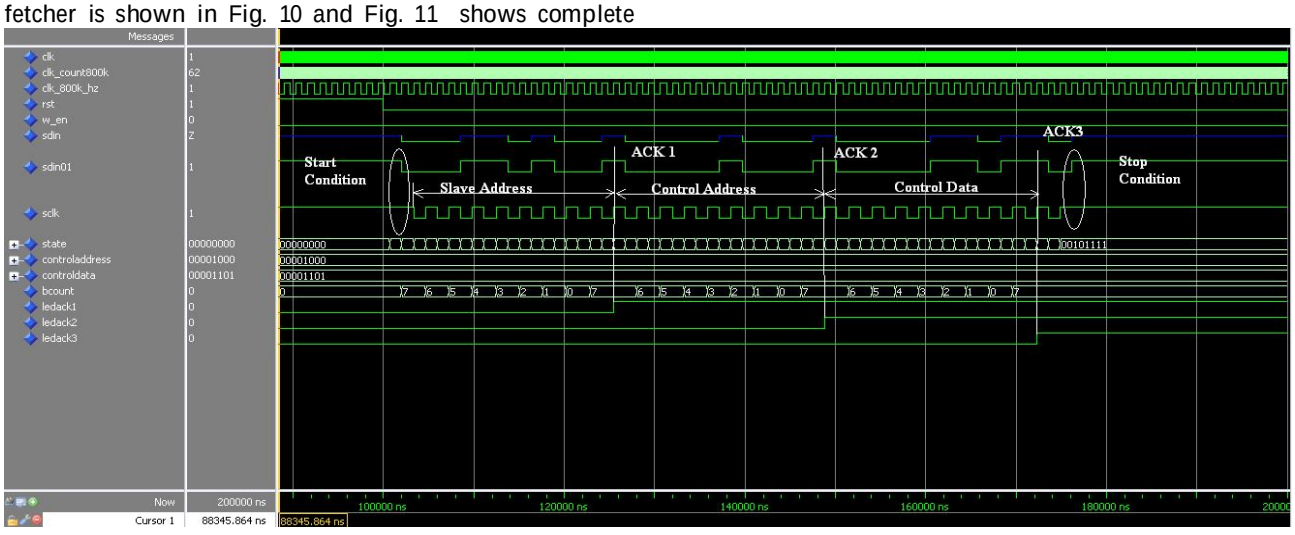

Fig. 7 Simulation result of the I²C Controller

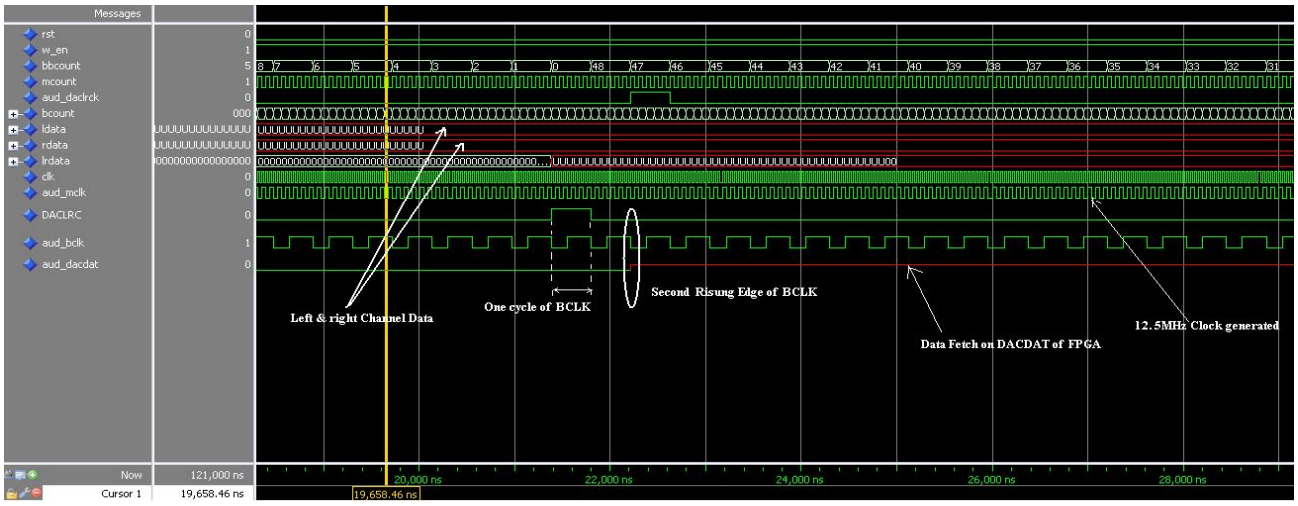

Fig. 8 Simulation result of the complete operation virtual sound fetcher

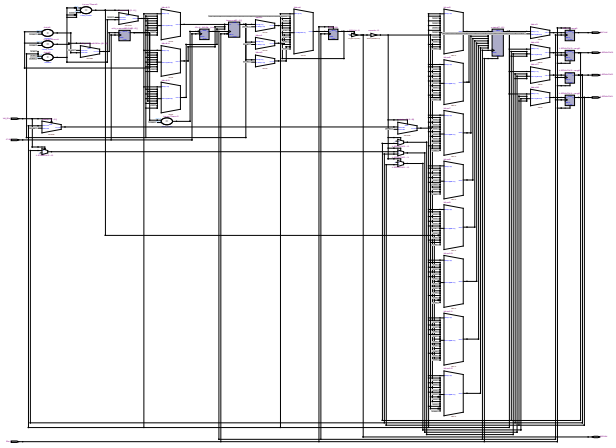

Fig. 9 Synthesized Design of the I²C Controller.(RTL view)

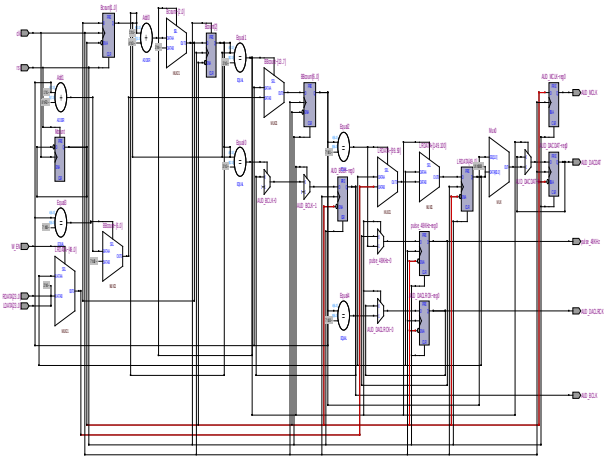

Fig. 10 Synthesized Design of virtual sound fetcher.(RTL view)

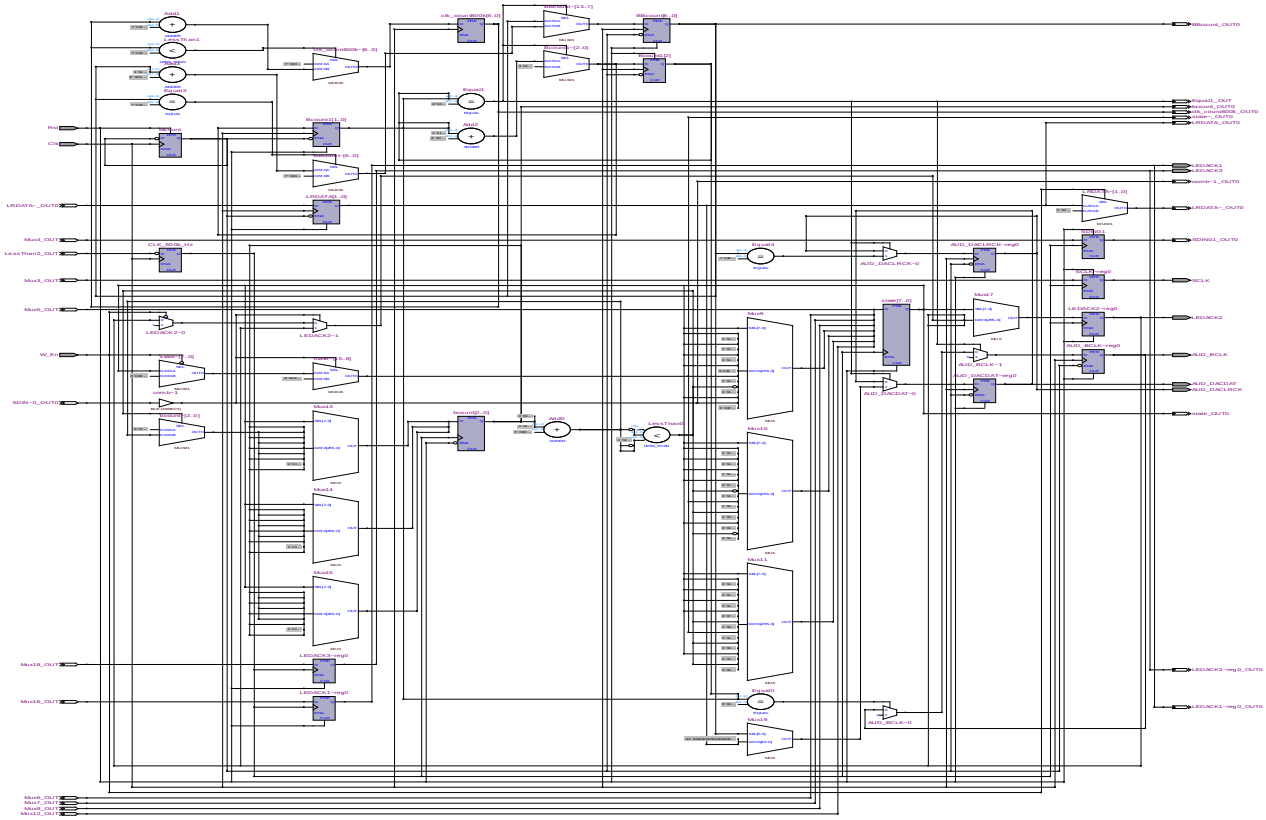

Fig. 11 Complete Synthesized Design of virtual sound fetcher with I²C specification (RTL view)

VI. Conclusions

The results of the simulations are shown in a paper, after checking the functionality with ModelSim, VHDL files is successfully synthesized in DE2 board with Quartus-II. This paper demonstrates the controller's operation by having its communication with Wolfson codec connected on the I²C bus as slave & fetching of the sound data that is left and right channel data of CODEC on the ADCDAT pin of the FPGA. Utilization of FSM reduces the synthesized hardware. Designed controller is mapped to FPGA of 90nm technology, out of 33,216 LEs, 76 logic elements are used in the design & only 11 I/O pins are used from 475 pins of FPGA. Hence this optimal design reduces the connection pin count for communication between the chips. The logic of the design is general enough so that anyone can use another compiler or another FPGA by changing hardware dependent pin mapping. You should also be able to use another chip instead of WM7831 that is used as the I²C slave. The designed controller also allows you to set address and the data switches at the bottom of the DE2 board to read and write from/to slave.

Acknowledgment

The authors would like to thank firstly, our GOD, and all friends who gave us any help related to this work. Finally, the most thank is to our families and to our country INDIA which born us.

References

[1] J Bhaskar, *A VHDL Primer,* Prentice Hall.

[2] Zeyad Assi Obaid, Nasri Sulaiman and M. N. Hamidon "FPGA-based Implementation of Digital Logic Design using Altera DE2 Board" *IJCSNS International Journal of Computer Science and Network Security*, VOL.9 No.8, July 2009

[3] Mohammed Y. Hassan and Waleed F. Sharif, " Design of FPGA based PID-like Fuzzy Controller for Industrial Applications", *IAENG International Journal*

of Computer Science, 34:2, IJCS_34_2_05, 17 November 2007.

[4] Enoch Hwang "Implementing an I²C Master Bus Controller in a FPGA for Maxim DS3232", *April 2008.*

[5] FPGA tutorial "Over view on FPGA", www.Tutorialreports.com, http://www.tutorialreports.com/computerscience/fpga/overview.ph

[6] "DE2 Development and Education Board" Altera DE2 Package, or [available online at] http://www.altera.com

[7] "DE2 Development and Education Board, user manual" Altera Version 1.42,2006, http://www.altera.com.

[8] "WM8731 Data sheet". Wolfson Microelectronics. 2004. PDF Document.

[9] *AN10216-01 I2C MANUAL*, Philips Semiconductors March 24, 2003

[10] Carlos Asmat , David López , Sanzo Kanwen "Speech Recognition Using FPGA Technology " June 18, 2007 , Design project laboratory, Department of Electrical Computer and Software Engineering

[11] José Ignacio Mateos Albiach "Interfacing a processor core in FPGA to an audio system" Master thesis performed in Electronics Systems LiTH-ISY-EX--06/3896--SE ,Linköping 2006