IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2019
Impact Factor: 4.819

Design 8051 Microcontroller in FPGA Using VHDL

Anupendra Singh[1] and Gajanand Sariyam[2]
Lecturer, Department of Electronics & Telecommunication[1,2]
Government Women's Polytechnic College, Jabalpur, Madhya Pradesh
anupendra1@yahoo.com and gajanandsariyam@gmail.com

Abstract: *This paper describes the design and implementation of a version of the 8051 microcontroller, one of the most commercially used microcontrollers in Altera DE-2 FPGA board equipped with Altera Cyclone-II FPGA using VHDL language with Quartus II software. The aim of this paper is to present the development of a simplified version of the 8051 microcontroller combining the basic components and the set of instructions executed in development time and the same dynamic reconfiguration at runtime. The microcontroller uses proposed more than a fixed set of instructions, with only one active at any given time and provide the ability to create new combination of instructions so that the microcontroller can incorporate them and use them in real time as if they were part of the fixed set of instructions. In this work comparative simulations are presented in relation to the time of execution and performance of traditional systems in relation to the microcontroller developed. In future work we will add more peripheral devices and instruction set.*

Keywords: 8051 microcontroller, VHDL, FPGA, Quartus II, Altera DE-2 board

I. INTRODUCTION

In recent years we have witnessed the growth of embedded systems design and telecommunications, which are mostly based on microcontrollers. Microcontrollers have features of a full computer system on a single chip, supporting activities of data acquisition, processing and control devices in industries and in complex projects involving electronics and computer [2]. However, the complexity of these devices has grown more and more the need for devices with better performance, flexibility to changing customer requirements, ability to adapt to operational changes and cost to ensure the competitiveness of new products [3].

The technological advances of programmable devices such as Field Programmable Gate Array (FPGA), allowed to design and develop high performance digital systems and capable of reconfiguring the hardware in order to optimize the features required to systems and control paradigms of the application [4].

II. CONTEXTUALIZATION

2.1 Reconfigurable Systems and FPGA

Many emerging applications in telecommunications and multimedia require flexibility even when implemented in view of the mutability patterns in the characteristics of systems and protocols during the life of equipment. There are two paradigms for solving these problems are fixed hardware solutions and solutions in software-programmable hardware.

The hardware solutions based are faster, but they are expensive and are not flexible because it does not allow you to change their characteristics after manufacture. Since the solutions based on software components are flexible, allow to easily correct mistakes and reuse components, but provide less than optimal performance and consume more power. An intermediate solution between these two paradigms are reconfigurable devices. The systems are reconfigurable hardware platforms with the ability to modify itself by software to better suit the application. Thus we have two basic types of reconfiguration:

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2019
Impact Factor: 4.819

1. Static reconfiguration, when the reconfiguration occurs in the development phase.
2. Dynamic Reconfiguration and / or Partial reconfiguration and when performed at runtime.

A reconfigurable architecture can also be hybrid, when they have a piece of hardware fixed and a reconfigurable, thus enabling even greater flexibility [2]. FPGAs are programmable circuits composed of a set of logic cells placed in the form of a matrix as shown in Fig.1. The logical structure varies from each manufacturer, but kept the main elements are: Configurable Logic Block (CLB) which is the logical unit of the FPGA, In / Out Block (IOB) that are responsible for the FPGA interfaces with the external environment and Switch Box (SB) which is the element responsible for the interconnection between the CLB routing through channels [2].

The field of reconfigurable computing has advanced extensively in the last decade using FPGAs as the basis for high performance reprogrammable systems. Many of these systems have achieved high levels of performance and demonstrated its applicability to solving a wide variety of problems [6].

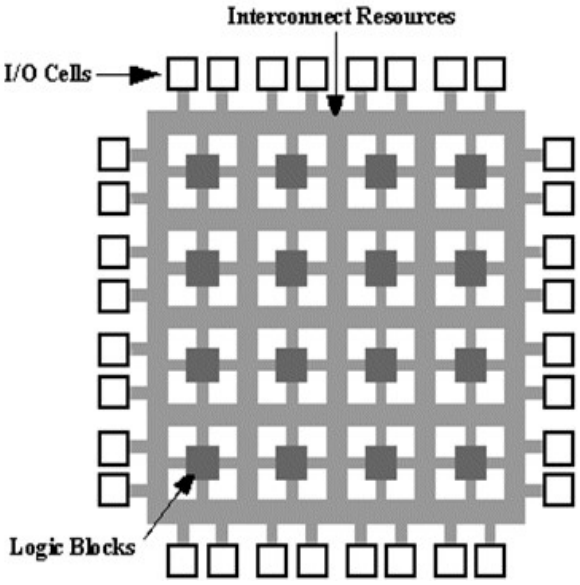

Figure 1: Component elements of the FPGA

A hardware description language describes what a system does and how it does. One of the most used languages in the description of hardware is the VHSIC Hardware Description Language (VHDL). A system described in language can be implemented in hardware allowing the use of FPGA in the field of your system, taking advantage of the code change at any time [7].

2.2 Microcontroller 8051

The Microcontroller 8051 is one of the most popular microcontrollers in the market. It is a simple 8-bit microcontroller, which can be found as a key element in the various equipment and systems. In Fig.2 we have the architectural model of a 8051microcontroller basics. The microcontroller 8051 is a CISC (Complex Instruction Set Computer) it has about 100 instructions and uses a typical 12MHZ clock.

The field of reconfigurable computing has advanced extensively in the last decade using FPGAs as the basis for high performance reprogrammable systems. Many of these systems have achieved high levels of performance and demonstrated its applicability to solving a wide variety of problems [6].

Microcontrollers are general purpose devices and in view of the requirements of modern systems for the speed, performance and fault tolerance, the time spent searching and instruction decoding is a bottleneck, because it directly influences the performance for solutions conventional microcontroller [8]. To resolve this issue were developed versions of the 8051 microcontroller on FPGA, using the properties and seeking to rewrite the necessary features that optimize applications.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2019
Impact Factor: 4.819

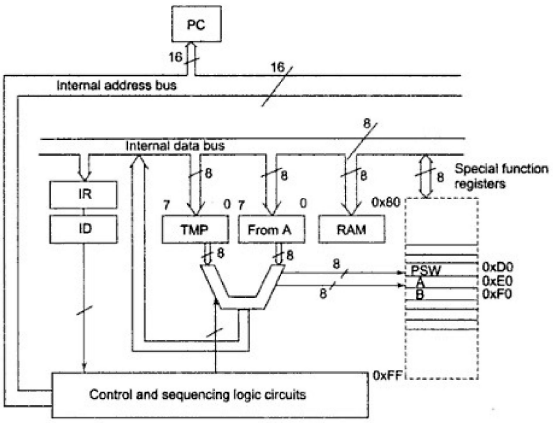

Figure 2: Basic Architecture of a 8051 microcontroller

2.3 Integration between the Microcontroller 8051 and FPGA

Currently there are commercial implementations of FPGA microcontroller 8051, focused on the processing of discrete tasks, with fixed set of instructions that cannot be modified to adapt to new situations processing. Some "core" of these are free and others as owners of Oregano mc8051, which provides all the components and peripherals of the microcontroller 8051 standard [5].

2.4 Features of DE-2 Board and QUARTUS II

DE-2 board is equipped with 16x2 LCD display, eight seven segment displays and LEDs (18 Red and 9 Green) that can be typically used to observe the outputs from the various output pins in the design. Available 18 toggle switches and four debounced switches can be used to provide external inputs, along with 50 MHz and 27 MHz on-board clocks. It also has an external SMA clock input. Figure 3 shows a complete schematic of Altera DE-2 board [11].

The expansion headers are used to hook-up the outputs to an external device, such as an oscillator for timing analysis. All these features have encouraged the migration of laboratory experiments, initially carried out using Xilinx XS40 FPGA board that offered considerably limited functionality as compared with Altera DE-2 board. Before using Altera DE-2 board, we can have created their designs in Mentor Graphics (UNIX) environment. It involved the use of Design Architect tool for schematic capture and QuickSim Pro for simulation. After a successful simulation, we followed a Xilinx-specific design flow to create a bit‖ file that could be downloaded onto Xilinx FPGA in Windows environment. Since the XS-40 board had an onboard 8031 microcontrollers, they could directly download it using serial communication with PC. Since 8031 does not contain on-chip ROM, the code had to be written onto on-chip RAM.

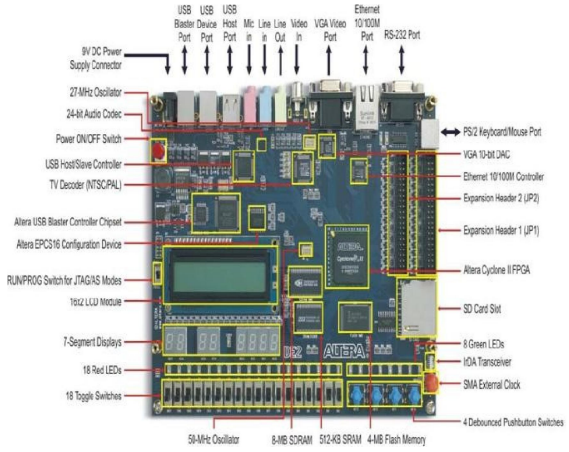

Figure 3: DE-2 Board

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2019
Impact Factor: 4.819

Altera's design and automation software ―Quartus-II‖ offers all the tools necessary for implementation of design. This eliminates the need to shift the OS environment every now and then. It has a Schematic-Capture module, a basic tool that enables to build custom digital circuits using a component library available with QuartusII. In addition, it can synthesize HDL designs (created using VHDL, Verilog or System Verilog) and it can also be used to integrate those with the designs created using schematic capture. Since DE-2 board does not have an on-chip microcontroller, we need to prototype one using its VHDL code. This can conveniently be done using Quartus-II that enables the user to create a schematic symbol for 8051 microcontrollers, which can be used as many number of times as needed. The HEX code is usually written inside Proceedings of 2007 Midwest Section Conference of American Society for Engineering Education ROM module in VHDL/Verilog and can be changed as per application. It can also be saved as a symbol and interfaced with the 8051 microcontroller using schematic capture. [8]

Quartus-II comes with a simulation tool that is used for functional verification of designs made using the above techniques. There are many other tools associated with Quartus-II which include Timing Quest timing analyzer, System on a Programmable Chip (SOPC) builder etc. The postsynthesis VHDL model is used to create all the necessary complex symbols to be given to them in the form of an EDIF (acronym for Electronic Design Interchange Format) File. This format is technology-specific and the one used here is Altera Cyclone –II EP2C35F672C6. EDIF can be read by Quartus-II and it can readily generate the symbol for the corresponding code. We build the glue logic using various tools in Quartus-II for prototyping in using Cyclone-II FPGA.[8]

III. Proposed Model

Our proposal is to build an 8051 microcontroller with FPGA dynamic reconfiguration of instruction, which may alter during the running time and a fixed instruction set implemented by means of fixed hardware development time. Our proposed 8051 Microcontroller Architecture as Fig. 3

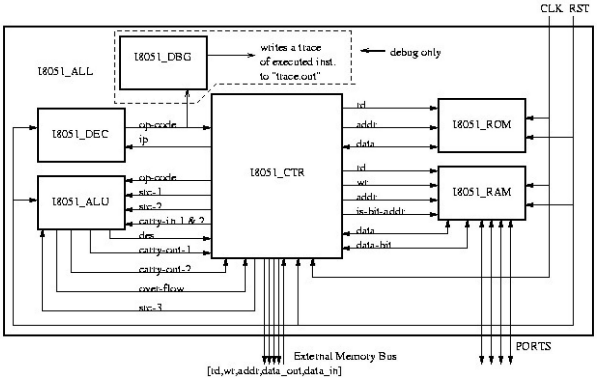

Figure 4: 8051 Microcontroller Architecture Proposed

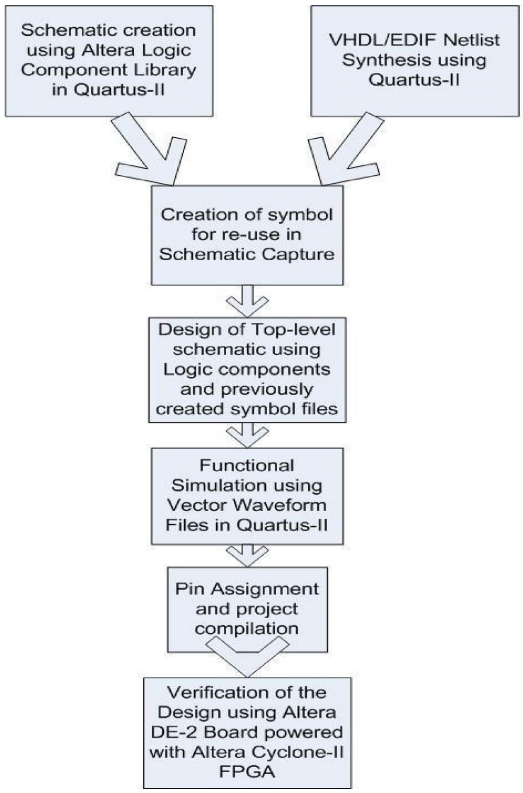

Figure 5: Design flow of proposed model

We follow a design procedure as Figure 5

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2019
Impact Factor: 4.819

File description of our model in given table 1.

File	Description
i8051_lib.vhd	Defines a package that is used in all the VHDL files of the 8051 model. This package defines commonly used constants.
i8051_alu.vhd	Model of an ALU that performs 8051 specific arithmetic. This model is described behaviorally as a combinational logic block.
i8051_dec.vhd	Model of a decoder that decodes the non-uniform 8051 instructions into uniform representations, i.e., enumerated codes. This model is described as a data-flow implementing a combinational logic block.
i8051_ram.vhd	Model of 128 bytes of RAM, specific to 8051, e.g., bit-addressable. This model is described behaviorally as a sequential logic block.
i8051_rom.vhd	Model of up to 64K bytes of ROM, specific to 8051. This model is automatically generated behaviorally, as a sequential logic block, using **i8051_mkr.c**.
i8051_ctr.vhd	Model of the core 8051 processor. This model is described behaviorally as a sequential logic block.
i8051_dbg.vhd	This entity is there for debugging only. Currently, it outputs a trace of each instruction that is executed.
i8051_all.vhd	Model of a complete 8051 micro-controller. This model structurally combines the above logic blocks.
i8051_xrm.vhd	Model of external SRAM that will interface to this 8051 micro-controller.
i8051_tsb.vhd an I/O	Model of a test-bench for the 8051 micro-controller. This model is described behaviorally as i8051_tsb.vhd an I/O-less sequential logic block. (It merely resets, then clocks the micro-controller forever.)
i8051_mkr.c	Program to convert an Intel 8051 HEX file into a ROM model, i.e., generates **i8051_rom.vhd**. You will need to compile this C/C++ file, say, **gcc -Wall i8051_mkr.c**, then run it with your HEX file as a command line argument to it, e.g., **a.out myfile.hex**.
syn_alu.inc,syn_dec.inc,syn_ram.inc,syn_rom.inc,syn_ctr.inc	Include files, specific to **dc_shell**, that will synthesize the above VHDL models down to gate.
zsim.scr,zsyn.scr	Script files that simulate/synthesize everything.

Table 1: File description

IV. RESULTS

After compilation on Quartus II version 4 of all VHDL file we obtain statistics as Fig.5, which incorporates the basic features of standard microcontroller.

Model	I/O Ports (bit)	Combinational Area (nand-gate)	Sequential Area (nand-gate)	Critical Path Length (ns)	Maximum Clock Speed (MHz)	Synth. Time (min)
I8051_ALU	50	2191	0	178	5.63	3
I8051_DEC	18	590	0	46.3	21.6	2
I8051_RAM	31	5237	8663	1.41	709	28
I8051_ROM*	23	1050	56	1.40	714	15
I8051_CTR	118	2202	998	1.77	565	17

Figure 5: Statistics result after compilation

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2019
Impact Factor: 4.819

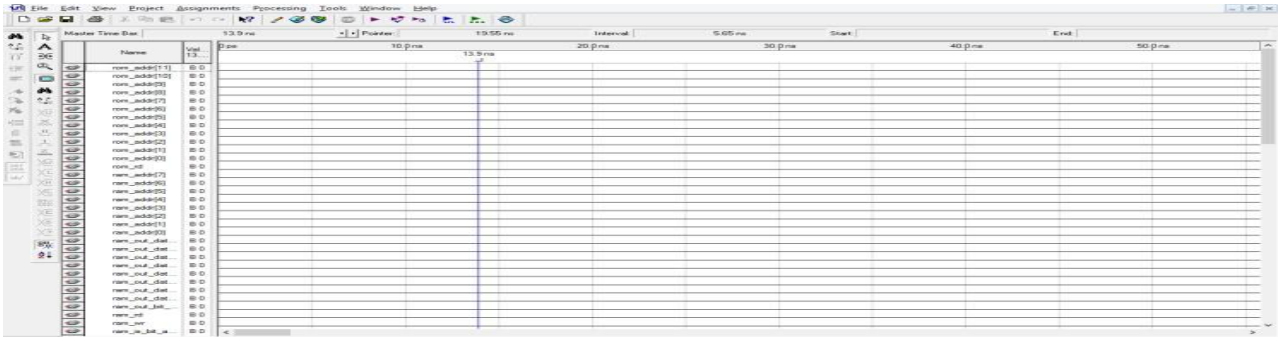

Figure 6: Simulation view of Quartus II

It will be used only a subset of the 8051 simplified instructions. Starting this core basics are all structured the architecture and functional blocks. Results of simulations were satisfactory and show a performance improvement over traditional solutions for core instruction set. we obtain RTL view structure as Fig. 7

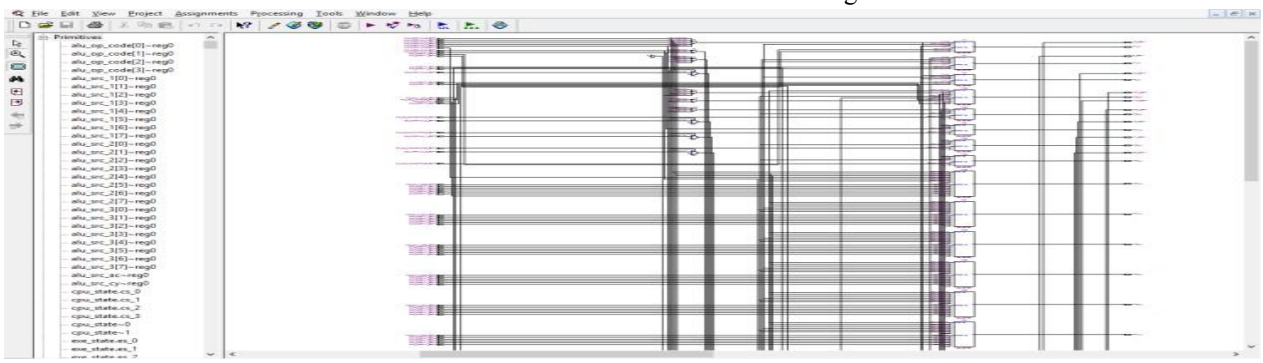

Figure 7: ***RTL view of Quartus II***

We also obtained a chip view in chip viewer of Quartus II as Figure 8. The last step will be to conduct tests using a development kit from Altera Cyclone II FPGA Starter Development, with the following features: 8MB SDRAM, 512KB SRAM, 4MB Flash, audio codec.

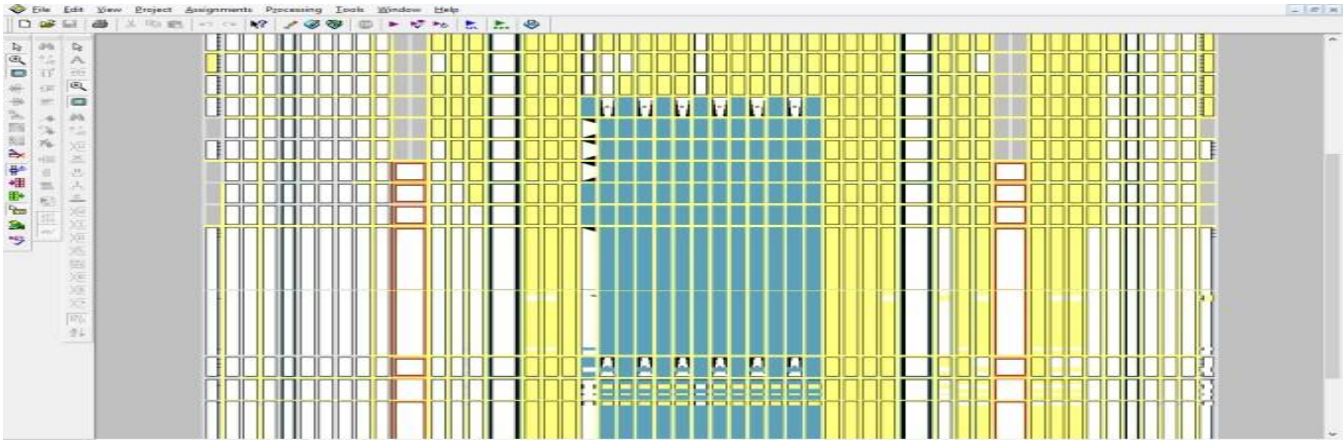

Figure 8: Chip view

V. Conclusions And Future Work

We design with the help of Quartus II software and implement in FPGA of basic features of 8051 microcontroller. In future work we intend to continue this same line of research and enhance the use of the core with reconfigurable instruction set processors for networked embedded in SOC capable of scheduling tasks. Furthermore you can optimize

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 12, Issue 1, December 2019
Impact Factor: 4.819

the instruction sets of fixed to better adapt to a wide range of different applications, allowing it to that use of the core is enhanced.

REFERENCES

[1]. CASTRO, F.S., ESKINAZI, R., Design of an 8051 Microcontroller in a FPGA with a Reconfigurable Instruction Set
[2]. ORDONEZ, E.D.M., PEREIRA, FD.., PENTEADO, C.G. and PERICINI, R.A. 2003. Projeto, Desempenho e Aplicações de Sistemas Digitais em Circuitos Programáveis (FPGA). São Paulo: Bless.
[3]. ZAQUEU, J.P.B and ESKINAZI, R. 2011. Implementação de uma versão Harvard/RISC reconfigurável do microcontrolador 8051 em FPGA. In IBERCHIP 2011.
[4]. CASSILO, L.A. 2005. Projeto e implementação em FPGA de um processador com conjunto de Instrução reconfigurável utilizando VHDL. Tese de Mestrado, Universidade Federal do Rio Grande do Norte, Natal.
[5]. CARRO, L.; PEREIRA, G.; SUZIN, A. Prototyping and Reengineering of Microcontroller-Based Systems. In: IEEE Rapid Systems Prototyping Workshop. Proceedings… June 1996.
[6]. SILVA, L.; LIMA, F.; CARRO, L.; REIS, R. Synthesis of the FPGA Version of 8051, UFRGS Microelectronics Seminar (12), June 6-7 : Porto Alegre, BRAZIL, pp. 115120, 1997.
[7]. MESQUITA, D.G. 2002. Contribuições para reconfiguração parcial, remota e dinâmica de FPGA. Tese de Mestrado, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre.
[8]. Waleed K. Al-Assadi, Mandar V. Joshi, Sagar R. Gosavi, and Daryl Beetner, Incorporating Altera FPGA Demo boards in Computer Engineering Labs
[9]. ALTERA Data Book 1999. Altera Corporation, www.altera.com, April, 2000.
[10]. A VHDL Primer , J. Bhasker , 3 rd edition PHI publication.
[11]. Altera DE-2 FPGA board Manual at www.altera.com/education/univ/materials/boards/DE2_UserManual.pdf
[12]. Altera Quartus-II Manual at www.altera.com/literature/manual/intro_to_quartus2.pdf