International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 08, August-2015

Field Programmable Gate Array Power Changeover System

Christian C Mbaocha
Department of Electrical/Electronic Engineering,
Federal University of Technology,
Owerri, Nigeria

Kalu Constance
Electrical Engineering Department,
University of Uyo,
Akwa Ibom State, Nigeria.

***Abstract*— The unreliability of power supply in developing countries inhibits the growth and development of both commercial and industrial sectors. Furthermore, these intermittent power outages result to damage of equipment and loss of business opportunities, hence the need to automate power changeover systems. This paper presents a cost effective approach to the design of an automatic power changeover system using Field Programmable Gate Array (FPGA) device for commercial and industrial usage. This device enhances the speed of operation of power changeover control system. ALTERA Quartus II development software was used to write codes and load into the FPGA. The Joint Test Action Group (JTAG) programming method was used and a switching time of less than 1second was realized.**

Keywords—Relays, FPGA, JTAG, switching, control unit.

I. INTRODUCTION

The worsening case of power instability in the developing countries spawns the need to automate electrical power generation or alternative sources of back-up power supply. It has been observed that over the years power instability has caused companies to lose huge sums of money as a result of the time lag between power failures and when power is restored. The major aim of this work is to exploit the advanced programmable logic device (PLD) facilities in bringing about automation of changeover process. Taking advantage of hardware parallelism, FPGAs exceed the computing power of digital signal processors (DSPs) by breaking the paradigm of sequential execution and accomplishing more per clock cycle. Controlling inputs and outputs (I/O) at the hardware level provides faster response times and specialized functionality to closely match application requirements. Each independent processing task is assigned to a dedicated section of the chip, and can function autonomously without any influence from other logic blocks. As a result, the performance of one part of the application is not affected when more processing is added [1]. Hence a FPGA Automatic changeover system proffer solution to the shortcomings of the already existing power changeover systems by performing power swap from public power to generator automatically and vice-versa. The earliest manual changeover switch box is operated whenever there is power failure and someone then changes the switch position to either ON or OFF as desired. [2]. Later, Automatic Changeover System with Electromechanical Relays (EMRs) was developed. Electromechanical relays (EMRs) have been used with other components such as logic gates, transistors, opto-couplers, microcontroller etc. Such control systems must be properly isolated from the relay to avoid the flow back of A.C signal into the control electronics [3]. Automatic Transfer Switch type was later developed but the limitations of this approach are more or less the same with automatic changeover system with electromechanical relays [4]. Engineers have made great effort in improving automatic changeover system using digital components and solid state relays with added flexibility and reliability to the existing system [5], [6]. A new type of automatic changeover system for industrial application is designed using FPGA. Field Programmable Gate Array (FPGA) is a recent breakthrough in the family of semiconductor devices. It can be reconfigured by the consumer to perform a desired task. FPGA is generically described as a programmable device with an internal array of logic blocks, surrounded by a ring of programmable input/output blocks, connected together via programmable interconnect. FPGA can be thought of as a two-layered device. This is shown in Figure 1.

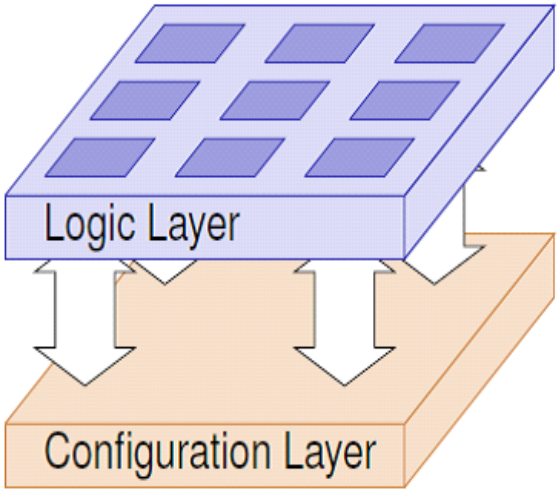

Figure 1.FPGA layers.

The Logic Layer consists of configurable logic blocks while the Configuration Layer holds in memory the FPGA "programs, controls the function computed on the logic layer and then allows partial reconfiguration at runtime.

A. FPGA Design Flow.

The standard FPGA design flow starts with the use of Schematic editor or a hardware description language (HDL), such as Verilog or VHDL [7]. The Schematic designs using schematic editor are easier to visualize whereas the use of HDL are for large hierarchical designs.

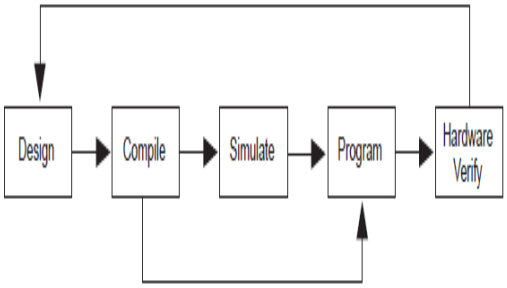

Figure 2. Design Flow Diagram.

The figure 2 shows the design flow. Here Schematic editor is used to create the digital circuit that is implemented in the FPGA. The flow then proceeds through compilation, simulation, programming, and verification in the FPGA hardware.

B. Design Entry.

This phase involves using schematic entry to create the logic to be implemented in the device. Pin assignments are made including pin placement information, and timing constraints that might be necessary for building a functioning design. Also library of parameterized modules (LPM) are added. Verilog HDL code is written to add desired logic blocks [8].

C. Compiling.

Compilation comes after creating the design. Compilation converts the design into a bit stream that can be downloaded into the FPGA. The most important output of compilation is an SRAM Object File (**.sof**), which is used to program the device. The software also generates other report files that provide information about the code as it compiles.

D. Program the Device.

After compilation and verification of design, next step is to program the FPGA on the development board. The SRAM Object File (sof) created in the above step is downloaded into the FPGA using the USB-Blaster circuitry on the board.

E. Verify in Hardware.

While verifying the design in hardware the runtime behavior of the FPGA hardware is observed and must be functioning appropriately.

II. SYSTEM DESIGN

The Figure3 shows the FPGA programming environment. It is the ALTERA Quartus II development software used to write codes and load into the FPGA. The software is compatible with Windows XP/Vista.

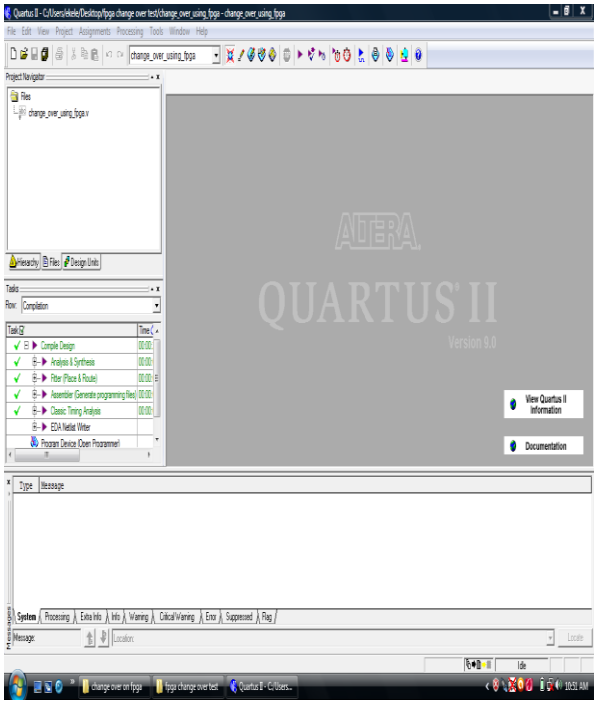

Fig 3: Figure showing the development environment/software.

F. Creating a New Project.

On the FPGA development software, navigate as shown in the diagram below to create a new FPGA project. The simplest way to create a project is from the **File** menu (**File » New » Project**).

G. Write the Code.

Type the Verilog code into the editor window. This code describes the system as a 4:1 multiplexer. Assigning the phases as inputs R, Y, B, G into the multiplexer, switching conditions are described. The Altera Software will generate hardware corresponding to the description given in the program. From the menu options, click tools > netlist viewers >RTL viewer to display generated hardware. The generated hardware is displayed as shown in figure 5.

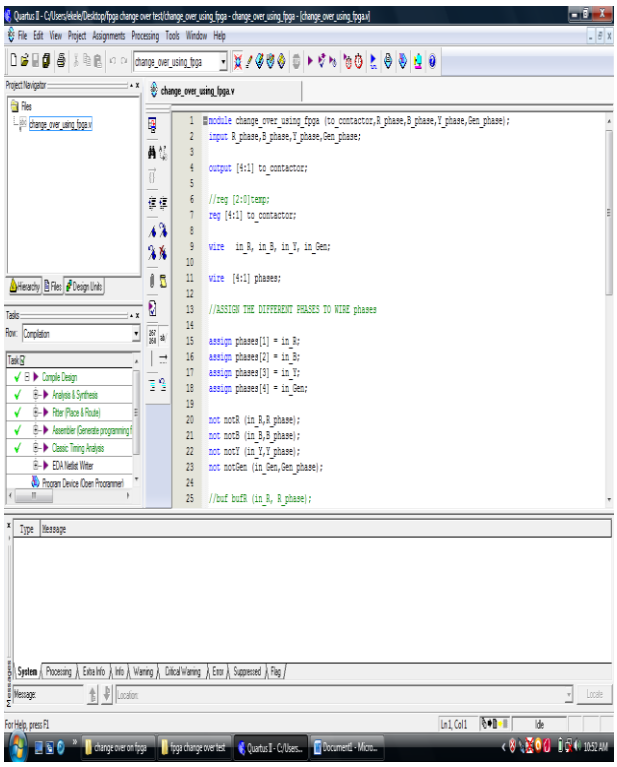

Figure 4: Showing the code that is written in the Development Environment.

H. Pin Assignments.

Pin assignments indicate where wires should be connected on the pins of the FPGA. Fig.6 indicate appropriate points in the hardware where connections are made using the interconnect wires. It should be noted that there are no predefined pin assignments. This helps in making pin assignments best suited for a circuit diagram to make routing of wires easier. The pins defined for the design are listed in Table I.

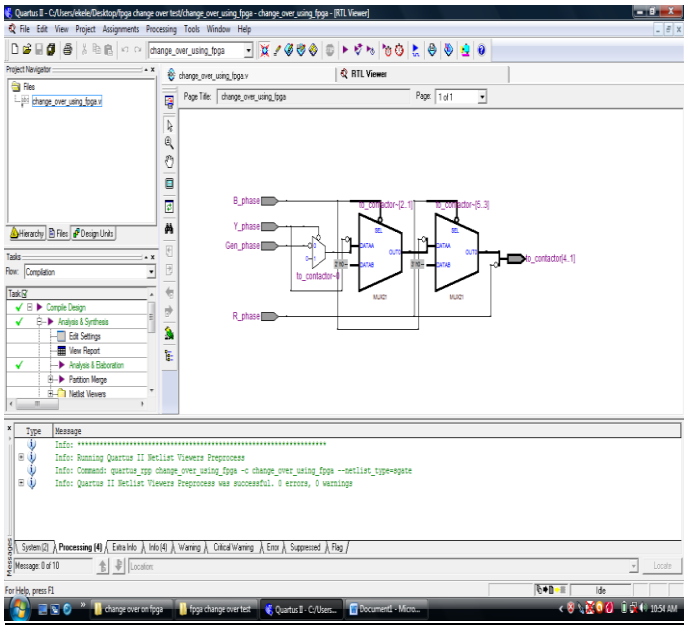

Figure 5: The generated hardware

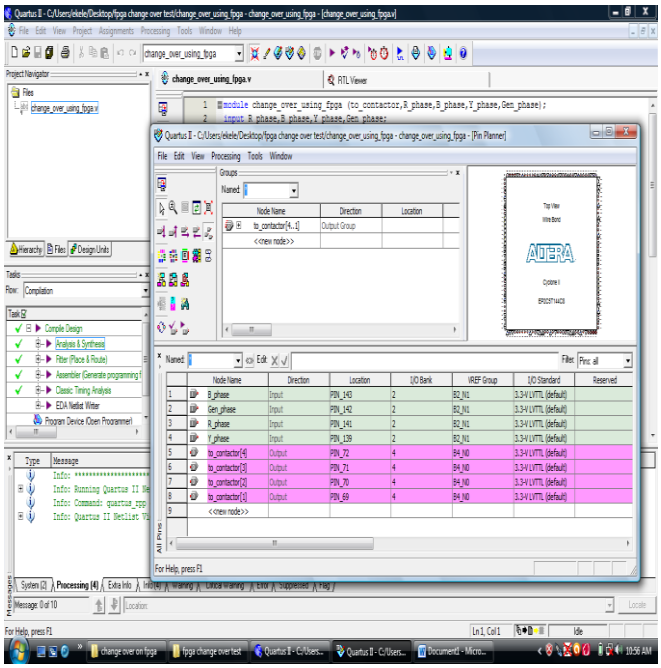

Figure 6: Showing the Assignment of Pins on the FPGA Body

TABLE I Table of Pin Assignments

	Node Name	Direction	Location
1	B_phase	Input	PIN_143
2	Gen_phase	Input	PIN_142
3	R_phase	Input	PIN_141
4	Y_phase	Input	PIN_139
5	to_contactor[4]	Output	PIN_72
6	to_contactor[3]	Output	PIN_71
7	to_contactor[2]	Output	PIN_70
8	to_contactor[1]	Output	PIN_69

The FPGA changeover system was implemented and results obtained is shown in Table II below

TABLE II Results

Phase 1	Phase 2	Phase 3	Gen	Phase priority	Input Voltage (V)	Output voltage (V)
ON	-	-	-	1	220	222~236
-	ON	-	-	2	220	229~238
-	-	ON	-	3	220	221-241
-	-	-	ON	Gen	220	219-241
ON	ON	-	-	1	220	221~236
ON	-	ON	-	1	220	221~236
ON	-	-	ON	1	220	221~236
-	ON	ON	-	2	220	229~240
-	ON	-	ON	2	220	229~240
-	-	ON	ON	3	220	220~242
ON	ON	ON	ON	1	220	219~238
-	-	-	-	-	-	-

From the Table 2, it can be seen that the system outputs at all times the status of the highest priority phase when more than one phase is on. Phase1 has the highest priority followed by phase 2 then phase 3. The Generator phase has the lowest priority and this means that if Generator is ON and any mains supply phase is On the Generator solenoid is de-energized. The condition of no power on any of the phases is used to generate a signal that energizes the Generator solenoid which automatically turns it on. Testing showed that the system has a switching time of less than 1sec.

II CONCLUSION

This work has demonstrated the application of FPGA in automatic power changeover system. This system adds intelligence to the power changeover and monitoring system by using digital logic programming method. The logic program was used to generate an interconnected logic gates that implements the logics in the written program. This method enhances the speed of operation while at same time reducing the component count. The work is carried out to compare the relative advantage of using FPGA in power changeover system over microcontroller based types. Granted, FPGA is bulky and expensive it is expected that future research will developed FPGA that is portable and cost effective through the application of VLSI technology. The paper will definitely benefit researchers and students in the areas of reliable switching device and automatic monitoring/changeover system.

The results obtained demonstrate that the Automatic Phase Change-Over Switch is relatively affordable and reliable. It is easy to operate, and it provides a high level of power supply when there are power outages. Finally, it reduces stress associated with manual change-over.

REFERENCES

[1] M. Thompson, (2012) "Introduction to FPGA Technology: Top 5 Benefits", National Instruments, USA.

[2] A. H. G. Al-Dhaher, (2004) "*Manual Changeover Switch Box*" Int. J. Engng Ed. Vol. 20, No. 1, pp. 52±60,

[3] Jonathan Gana Kolo, (Oct. 2007) "Technical Report- Design and Construction of an Automatic Power Changeover Switch", Au j.t. 11(2): 113-118.

[4] Ogbugo Arinze, (Jan. 2009). "*Changeover with Automatic Transfer Switch*", Academic Research International Journal vol 1. No. 3

[5] Mano, M. Morris and Kime, Charles R., (2004) "*Logic and Computer Design Fundamentals*, 3rd Edition'',Pearson Prentice-Hall.

[6] Hyde, Daniel C., (1997) "CSCI 320 Computer Architecture Handbook on Verilog Hdl".

[7] Brown, Stephen and Vranesic, Zvonko, "*Fundamentals of Digital Logic with Verilog Design"*, McGraw-Hill.

[8] Ciletti, Michael D., (2003 *) "Advanced Digital Design with Verilog Hdl"*, Prentice Hall.