ALTERNATIVE POWERED FIELD PROGRAMMABLE GATE ARRAY (FPGA) LIQUID DISPENSER: SOLUTION TO TROPICAL FARMLAND IRRIGATION AND FUMIGATION PROBLEM

E.A. Ugwu[1] and C.G. Ezekwe[2]

[1]Department of Computer Science, Enugu StateUniversity, Enugu; [2]Electronics Development Institute, Awka; National agency for Science and Engineering Infrastructure (NASENI), Federal Ministry of Science and Technology, NIGERIA.

[1]Ugwuedith2003@yahoo.com [2]norakingchi@yahoo.com

ABSTRACT

The research will be based on solar energy field programmable gate arrays (FPGA) liquid dispensing apparatus and methods for measuring various fluid parameters of the liquid. The liquid dispensing apparatus will comprise of containers having chambers for holding any liquid. This project will also focused on real time monitoring and control through network camera which triggers alarm in farm on intrusion by unwanted figures like birds, rodents, even unauthorised human etc. This system will be attached to alternative power supply (solar energy) thereby providing constant electrical energy to the farm. This project will be focused on using Very High Speed Integrated Circuits (VHSIC) Hardware Descriptive Languages (VHDL) and Verilog to design an application specific integrated circuit (ASIC) liquid dispenser controller system while targeting the device independent architecture (Altera cyclone II FPGA series) for synthesis, optimization and fitting to realize the design.

Keywords: Field programmable gate array (FPGA), liquid dispenser, VHSIC Hardware Descriptive Language (VHDL) and Verilog

INTRODUCTION

Hardware engineers are focusing on developing new digital system designs in languages like VHDL & Verilog, which are are consequently targeting CPLDs and FPGAs as well as their ASIC versions [1]. This project will be focused on using VHSIC Hardware Descriptive Languages (VHDL) and Verilog to design an application specific integrated circuit (ASIC) liquid dispenser control system with real time monitoring and control while targeting the device independent architecture (Altera cyclone II FPGA series) for synthesis, optimization and fitting to realize the design. This project will tend to achieve the required watering and implementation of other fluids like fumigatives, etc in our tropical farm lands as at when required. Also the project will be attached with real time monitoring and control through network camera which triggers alarm in the farm on intrusion by unwanted figures like birds, rodents, even unauthorised human etc . This system will be attached to alternative power supply (solar energy) thereby providing constant electrical energy to the system and the farm.

The VHSIC Hardware Descriptive Language (VHDL) has gained wide acceptance as a tool for hardware design and synthesis. VHDL is frequently used for two different goals:

i. Simulation of electronic designs, and
ii. Synthesis of such designs.

Synthesis is a process where a VHDL is compiled and mapped into an implementation technology such as an FPGA or an ASIC [1]. An ASIC is a custom- designed fixed-function device. The key advantage of VHDL, when used for systems design, is that it allows the behavior of the required system to be described (modeled) and verified (simulated) before

ISSN: 2223-9944, e ISSN: 2223-9553

synthesis tools translate the design into real hardware (gates and wires). Very High Speed Integrated Circuit Hardware DescripTION Language (VHDL) is a hardware description language with simple grammar. VHDL has strong behavior description ability; there might not be need for programmers to understand the structure of the hardware, design and characteristics of Field programmable gate array (FPGA). At present, VHDL gets support from numerous EDA Companies as the industrial standard hardware description language of IEEE[2].

State Machine VHDL Programming

A finite state machine is a mathematical model of computation used to design both computer programs and sequential logic circuits. It is conceived as an abstract machine that can be in one of a finite number of states. The machine is in only one state at a time; the state it is in at any given time is called the current state. It can change from one state to another when initiated by a triggering event or condition; this is called a **transition.** A finite state machine is defined by a list of its states and the triggering condition for each state [7]. Finite state machine (FSM) is an important means to realize high-reliability logic control with high efficiency in practical digital system design. The traditional state machine design method needs tedious state assignment, drawing state table, simplification of the next state equation etc. [3].

The state machine has Moore and Mealy model. The out coming signal of the state machine of Moore Model only relates to the current state; the out coming signal of the state machine of Mealy model not only relates to current state, but also relates to input signal. Design of this dispenser control module and the state transition diagram will be based on Moore model as shown as in Figure 1.

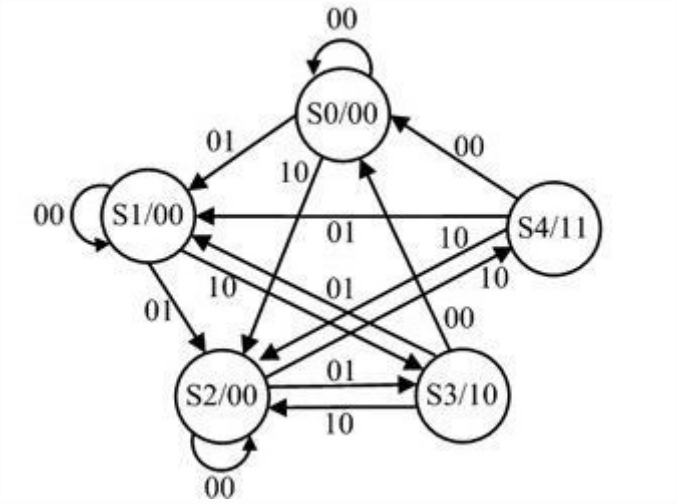

Figure 1: The VHDL Finite State Machine [2]

Field Programmable Gate Array (FPGA)

Field Programmable Gate Arrays popularly known as FPGAs is an alternative for implementation of digital logic in systems. They are prefabricated silicon chips that can be programmed electrically to implement any digital design [4]. The first static memory-based FPGA (commonly called as SRAM based FPGA) was proposed by Wahlstrom in 1967. This architecture allowed for both logic and interconnection configuration using a stream of configuration bits. Later on the first commercial modern-era FPGA was introduced by Xilinx in 1984. It contained the low classic array of Configurable Logic Blocks (CLBs) and inputs/outputs. The first FPGA contained 64 CLBs and 58 inputs and outputs. Today's

ISSN: 2223-9944, e ISSN: 2223-9553

modern FPGA now can contain approximately 330,000 logic blocks and around 1100 inputs and outputs. The basic architecture of FPGA consists of three major components:

i. programmable logic blocks which implements the logic functions,
ii. programmable routing (interconnects) to implement these functions and
iii. I/O blocks to make off-chip connections.

An illustration of typical FPGA architecture is shown in figure 2.

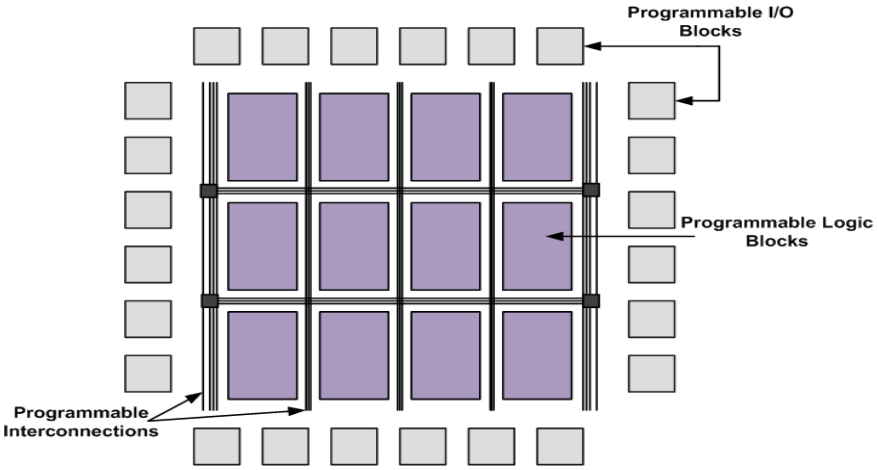

Figure 2: Internal Architecture of FPGA [4]

Programmable Logic

The purpose of programmable logic block in an FPGA is to provide the basic computation and storage elements used in digital systems. The basic logic element contains some form of programmable combinational logic, a flip-flop or latch and processor [5]. In addition to a basic logic block, many modern FPGAs contains a heterogeneous mixture of different blocks, some of which can only be used for specific functions, such as dedicated memory blocks, multipliers or multiplexers and configuration memory which is used throughout the logic block to control the specific function of each element within the block.

Programmable Interconnect

The programmable routing in an FPGA provides connections among logic blocks and I/O blocks to complete a user defined design**[4]**. It consists of multiplexers, pass transistors and tri-state buffers, which forms the desired connection. Generally, transistors and multiplexers are used within a logic cluster to connect the logic elements together while all three are used for more global routing structures. Several global routing structures have been used in FPGAs as: island-style, cellular, bus-based and registered (pipelined) architectures.

Programmable I/O

The media required to interface the logic blocks and routing architectures to the wide range of external components to FPGA are called I/O pads or programmable I/O**[4]**. The I/O pad and surrounding supporting logic circuitry forms I/O cells. These cells are important components of an FPGA and consume a significant portion (approximately 40%) of FPGAs area. The design of I/O programmable blocks is challenging as there is a great diversity in the supply voltage and reference voltage standards. One of the most important decisions in I/O architecture design is the selection of standards that will be supported. This involves carefully made trade-offs which can implement any digital functions. Supporting large number of standards can increase the silicon area required for I/O cells significantly. Additionally to the

ISSN: 2223-9944, e ISSN: 2223-9553

support of more number of standards pin capacitance, the capacitance may increase with more number of pins, which can limit the performance.

Over the time, the basic FPGA architecture has been further developed through the addition of more specialized programmable functional blocks. The special function blocks like:

i. embedded memory (Block RAMs),
ii. arithmetic logic (ALUs),
iii. multipliers, DSP-48 and
iv. Embedded microprocessors

The special function blocks have been added due to a frequent need of such resources for an application. The result is that many FPGAs have the heterogeneous mixture of resources than early FPGAs.

Hardware Description Language

Hardware description Languages includes VHDL, Verilog, System C and Handel-C are frequently used for FPGA programming. Behavioral, RTL and structural levels of description can be used inter-changeably in these languages. System C is a C++ based library used for modeling system level behavior. As the base language is C++, software processes can be more easily modeled than in a more traditional HDL. Synthesis tools for SystemC are emerging, but do not approach the maturity of VHDL or Verilog synthesis products. Handel-C is also a relatively new product in comparison to VHDL or Verilog. Handel-C follows the Communicating Sequential Process (CSP) model. Handel-C requires the designer to explicitly delineate parallel processing blocks within a process. It includes intrinsic for inter-process communication, as does SystemC 2.0.

RESEARCH OBJECTIVES

1. The system will aid to tackle draught problems in the farm by automatically watering only when needed in the farmlands.
2. It will tackle the problem of unwanted plant growth in the farm by automated introduction of fumigatives as at when required.
3. The problem of theft and intrusion will be addressed by the system by providing a central control and monitoring to all angle of the farm.
4. In the tropics farmland, we have existing ground irrigation systems through their capillary action draw up salt from the soil. these salt hinder the growth of crops. This system will be an overhead water dispenser thereby addressing this issue.
5. As water sits in irrigation channels, malarial mosquitoes bilharzia snails can breed [7] thereby causing health problems, the proposed system must address this problem by direct pumping of water from the source example bole holes, streams, lakes and rivers.
6. In the tropics farmlands especially rice and yam farm, our farmers suffer from the invasion of birds and rodents that can eat up the crops before harvest. the system will also address this problem by effective alarm system via network cameras that trggers off if birds or rodents invade.
7. Most of our tropical farm land are located in remote rural areas, the project is an electrical driven project and there must be need for constant electrification which will be provided through alternative energy source [solar system], thereby providing light in the farm.

ISSN: 2223-9944, e ISSN: 2223-9553

DESIGN PROCESS FRAMEWORK (FLOWCHART DESIGN)

The availability of computer-based tools has greatly influenced the design process in various design environments [1]. The flowchart in figure 3 depicts a framework for the design process that will be adopted in this work . In this work, it is the intension of the authors to develop an ASIC custom controller for a liquid dispenser system since the design can be optimized for a specific task; hence leading to better performance.

The process begins with the definition of the liquid dispenser controller system specifications. Then the general structure of the initial design is developed. The Quartus II Altera software was now used for simulation of the behavioral description. Any error is rectified before generating the prototype for a complete successful design.

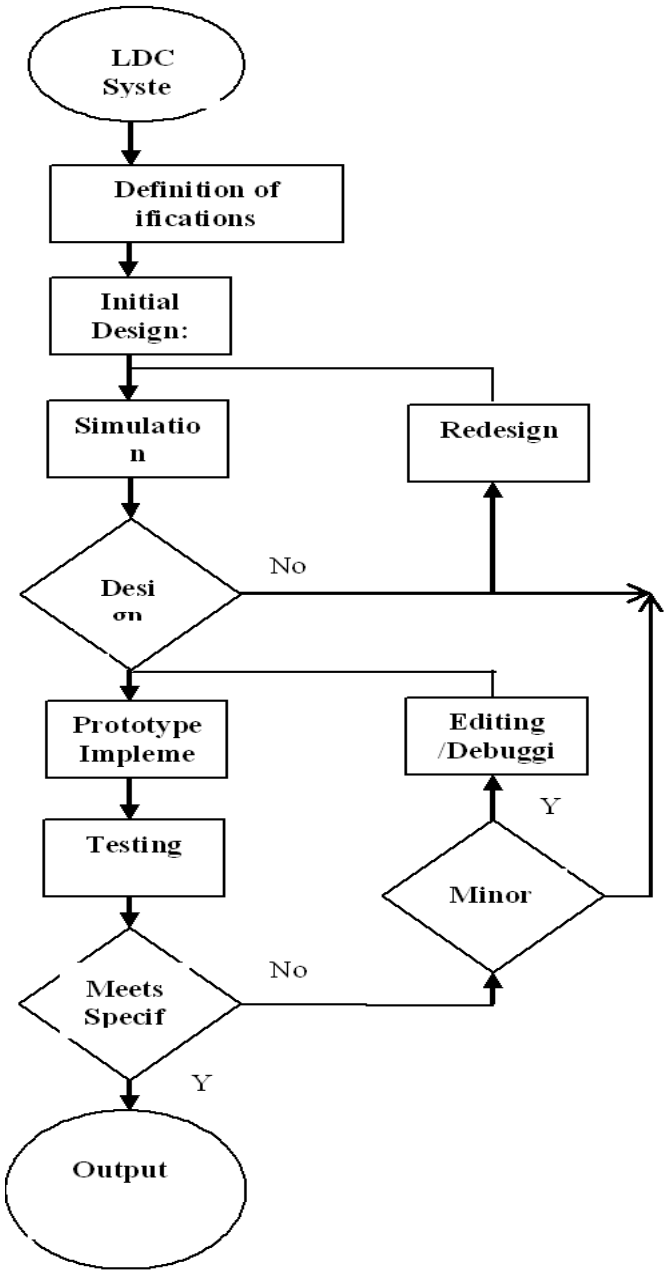

Figure 3: Design Process Framework

ISSN: 2223-9944, e ISSN: 2223-9553

METHODOLOGY

This project will be focused on using VHDL to design an application specific integrated circuit (ASIC) liquid dispenser controller/network/alarm system while targeting the device independent architecture (Altera cyclone II FPGA series) for synthesis,optimization and fitting to realize the design. These steps has to be taken;

1. The device-independent architecture that will be used for this work is FPGA cyclone FY37256-FY37256P208-154 NC and programmed with VHDL.
2. VHDL code for Entity, Architecture, Component and package declarations for the Liquid Dispenser Circuit/network/alarm system behavioral description will be developed.
3. Synthesizing the VHDL description for the Liquid Dispensing Circuit/network/alarm system that facilitates the development of the ASIC version of the system.
4. Writing VHDL Entity declaration and Architecture that instantiates the bin cans into a top level VHDL file via the logic Instantiation.
5. Running WARP on windows 7 platform to compile and synthesize the system design description.
6. Joint Electron Device Engineering Council (JEDEC) file will be generated for the chip mapping.

Designing Tools

The VHSIC Hardware Descriptive Language (VHDL) has gained wide acceptance as a tool for hardware design and synthesis. VHDL is frequently used for two different goals: simulation of electronic designs and synthesis of such designs. Synthesis is a process where a VHDL is compiled and mapped into an implementation technology such as an FPGA or an ASIC]. Other tools are:

i. Quartus II Altera software Altera
ii. Cadence Cypress warp EDA tools
iii. Mentor Graphics Synopsys
iv. Synplicity and Xilinx.

This work will use Quartus II Altera with the Cypress Warp EDA tool. After the design description in the EDA environment, the CAD tool performs the automatic tasks of optimizing the design logic circuitry to meet the objectives.

EXPECTED RESULTS

Essentially, there are two major bin can circuits and two systems (network and alarm). These circuits will be declared as components and their corresponding packages defined. Consequently, a top level description that instantiates the two bins can circuits will be implemented via the logic instantiation. After that, the low level bin can circuits and the high level logic instantiator will be compiled and synthesized into the device independent Altera cylone FPGA FY37256 file. This file will be burn and build into the physical FPGA device ready for construction.

DISCUSSION

This project will present VHSIC Hardware Descriptive Language (VHDL) approach as an efficient design technique for an ASIC Liquid Dispenser Circuit/network/alarm system. It will also outline an efficient design framework for the proposed system, unlike the traditional digital system development approaches; this technique will offer several advantages to the

ISSN: 2223-9944, e ISSN: 2223-9553

VLSI market segments particularly quick time to market, ASIC migration, flexibility, vendor independent compilation and synthesis among others. This is the future of digital revolution for hardware modeling of complex digital systems.

REFERENCES

[1]. Brown, S., Rose, J. (2005). Architecture of FPGAs and CPLDs: A Tutorial Fundamentals of Digital Logic with VHDL Design, 2nd Edition, 2005

[2]. IEEE Std 1076-1987. IEEE Standard VHDL Language Reference Manual. IEEE, 1989.

[3]. Lipsett, R., Schaefer, C., Usserty, C. (1989). *VHDL: Hardware Description and Design.* Kluwer Academic publishers.

[4]. Okafor, K. C., Oluwaseyi, A. O., Ezekwe, G. C. (2011). "Hardware Description Language:an efficient approach to device independent designs for VLSI market segment" Adaptive Science & Technology (ICAST), 2011 IEEE 3rd International Conference,

[5]. http:// www.google.com/wikipedia, 2013/finite state machine.html

[6]. http://www.google.com/acoustic liquid dispensing apparatus/patent US7426866.html

ISSN: 2223-9944, e ISSN: 2223-9553