

Compensation Power Design Framework For High Performance Asic Liquid Dispenser System (ALDS).

¹Jonathan U. Agber, ²Akura Mise, ³Kennedy C. Okafor

¹Department of Electrical and Electronics Engineering, Federal University of Agriculture, Makurdi, Benue State, Nigeria.

²Department of Electrical and Electronics Engineering, Federal University of Agriculture, Makurdi, Benue State, Nigeria.

³Department of Electronics Development Institute, ELDI Nat. Agency for Sc. and Eng Infrastructure (NASENI), Fed. Min. of Sc and Tech, Nigeria

-----ABSTRACT-----

In FPGA/CPLD based designs, the highest efficiency at all load conditions enables system designers to overcome thermal challenges, hence reaching out for a new level of system miniaturization. Besides, with the ever increasing power consumption trend in today's high-performance systems, managing the power within the system design budget is becoming as important as meeting the performance specification of the system design. Managing system power within the budget also has significant impact in maintaining overall system reliability. This paper presents ALDS system model with the synthesis results as well as discusses its energy efficiency and optimization for low power consumption, thus reducing thermal and power component cost as well as increasing overall system reliability.

KEYWORDS: Miniaturization, Performance, Budget, Reliability, ALDS

Date of Submission: 13.June .2013



Date of Publication: 10.July.2013

I. INTRODUCTION

ALDS is a VLSI system based device built on an Ultra 3700 CPLD series board developed in this research. ASIC is a custom-designed fixed-function device; an example of an ASIC would be a specially designed chip for a fax/modem [1]. The semiconductor industry's rapid migration to VLSI technologies (CPLD/FPGAs) produces benefits of performance and cost, but these benefits have an adverse effect on the power budget. Essentially, as transistor sizes decrease, leakage current, and hence static power increases. Dynamic power also increases with the increase in system speed and design density but in a more linear fashion [2]. Today, many designs have 50-50 static and dynamic power dissipation. According to International Technology Roadmap for Semiconductors (ITRS) projections, static power is increasing exponentially at every process node, and innovative process technologies are imperative [2]. With the adoption of FPGA/CPLDs in more system designs, power consumption within the entire system is becoming a critical part of the overall system budget. Consequently, leading FPGA vendors are adopting new techniques to mitigate the increasing static and dynamic power consumption. Xilinx uses triple oxide technology in its 65 nm process to reduce static power in Virtex™-5 FPGAs, while also providing embedded blocks and software tools to help system designers to further optimize power in high-performance system designs [2].

In this work, VHDL was used to develop the system code since being created once, a calculation block can be used in many other projects. However, many formational and functional block parameters were fine tuned (capacity parameters, memory size, element base, block composition and interconnection structure) for efficient performance. 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 synthesis tools translate the design into real hardware (gates and wires) [3]. Figure 1 shows the system architecture of ALDS ready for behavioural description for its entity and architecture in VHDL WARP [6] while Figure 2 shows the ALDS integration framework.

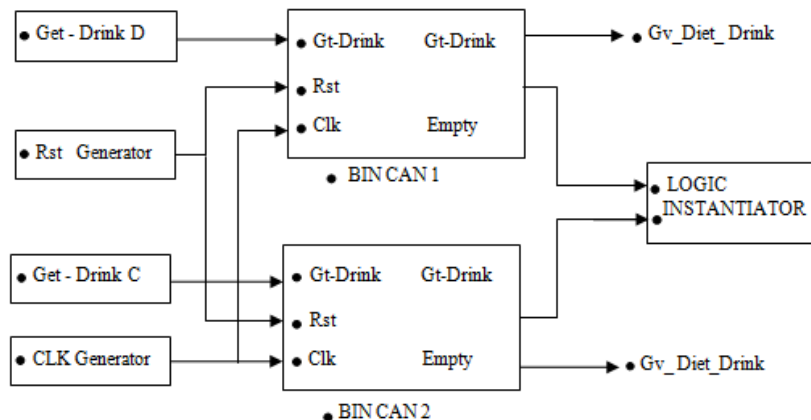


Figure 1: The system architecture of ALDS

II. METHODOLOGY- SYSTEM MODEL (PHASE 1)

The use of VHDL to design an application specific integrated circuit (ASIC) liquid dispenser system, while targeting the device independent architecture (Ultra 3700 CPLD series) for synthesis, optimization and fitting to realize the design is presented in the first phase of this work. ALDS model have two bin cans to dispense regular and diet drinks. The system dispenses a drink if the user activates a button for that drink and at least one can is available. A refill signal appears when both bins are empty. Activating a reset signal informs the system that the machine has been refilled and the bins are full as shown in Figure1. Figure 2 provides an anatomy of the report file generated by *Warp*. Starting from the design entry, design compilation to design verification, the report file generated by *Warp* has the same base name as the design VHDL.rpt extension. A report file is generated for every file that *Warp* compiles. If a design is split up among multiple files, the report file that will probably be most useful is the one for the top level design. The report file can be broken down into five main sections:

- Parser
- High Level Synthesis
- Synthesis and Optimization
- Technology Mapping and Optimization
- Fitting

The first operation is parsing. The parser, shown in Figure 3, verifies correct language syntax, correct semantics, correct instantiations of library components, and module generation. VHDLFE.EXE performs this operation for VHDL source files. The next step is high-level synthesis. This step reduces the high-level VHDL to an intermediate format called an expression tree. Synthesis and optimization are performed by TOPLD.EXE. There are two processes, synthesis and optimization, within TOPLD. The synthesis portion performs state machine synthesis. The optimization portion optimizes through substitution, removes unused logic at the gate or architecture level, and performs initial Boolean-level logic reduction. This step runs only when the top-level file in a device project is compiled. Technology Mapping and Optimization is performed by two programs: DSGNOPT.EXE and MINOPT.EXE. DSGNOPT optimizes the logic structures for the target architecture while MINOPT is called as a Boolean reduction engine. DSGNOPT chooses flip-flop type and equation polarity according to the options available in the target PLD to minimize product term usage. It also performs sum-splitting and global resource reduction, and finally converts all the equations into a format appropriate to the target architecture.



Figure 2: ALDS Full Integration Framework

Fitting is the compilation step that assigns the final equations and functions to physical hardware (ALDS) locations in the target device. This step runs only when the top-level file in a device project is compiled. The resulting file will be *filename.JED*. The fitter that is actually invoked depends on the type of device that is targeted. The Ultra37000 family invokes 37KFIT.EXE. The flow starts with design entry and synthesis in Leonardo Spectrum shown in Figure 4. The output is a VHDL netlist, which *Warp* then fits into a Cypress PLD. *Warp* generates a file for device programming, and VHDL or Verilog timing models are generated for post-synthesis simulation.

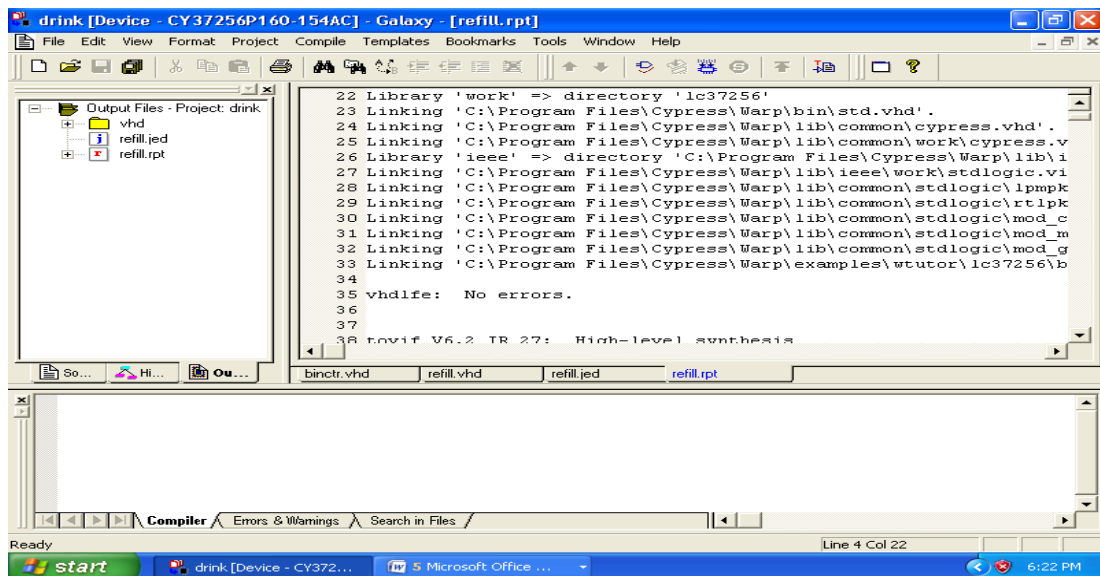


Figure 3: The Paser (Cypress CY37256P160 Jedec Fuse File: refill.jed)

Leonardo Spectrum

Exemplar Logic's Leonardo Spectrum uses flow tabs to guide users through the synthesis process. Additionally, process wizards lead users through the design flow. The four main tabs used while targeting a Cypress device are Technology, Input, Optimization and Output. Figures 5,6,7 and 8 shows the Leonardo spectrum EDA interfaces for device independent ALDS programming.

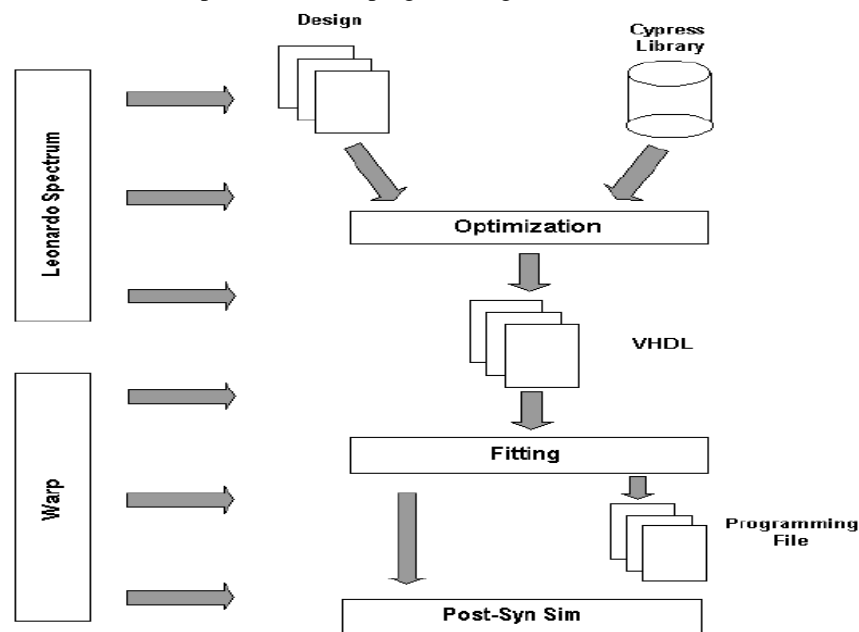


Figure 4: Cypress Logic Design Flow

III. ELECTRONIC DEVELOPMENT AUTOMATION TOOL (EDA)

Modern design of logic circuits depends heavily on Computer aided design tools [4]. Computer aided design tools in the VLSI market segments are produced by CAD vendors [4] like Altera, Cadence, Cypress, Mentor Graphics, Synopsys, Simplicity, and Xilinx. However, this paper developed the liquid dispenser controller (LDC) 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. Synthesis tools support VHDL, although some problems can arise with its use for synthesis [5]. Challenges such as modeling low-level devices, understanding if timing information is part of specification or if it belongs to the simulation model, defining the equivalence between logic levels and electrical levels, are some of the problems that may be encountered. Thus, synthesizing a circuit from a VHDL description is possible, but some limitations are imposed on the description.

IV. DEVICE INDEPENDENT DESIGN CONCEPT

Since VHDL supports multiple methods of design description, with one description for the liquid dispenser system (LDS), targeted device architecture (CPLD Ultra 37256-CY37256P208-154 NC) was selected to optimize the LDC system for resource utilization. The same VHDL code (design) can be used with any synthesis tool (compiler) and targeted to any vendor device (PLD, CPLD, or FPGA). Hence, with VHDL, device-independent design and portability allow for benchmarking a design using different device architectures and different synthesis tools. This, however, facilitates speedy design process and guarantees quick time-to-market at low cost. In this research, ALDS model is presented in Figure 1. The controller comprises two bin cans to dispense two drink brands (diet and coke). Each bin can holds three cans of drinks or more. The system dispenses a drink if the user activates a button for that drink. When both bin cans are empty, a refill signal appears. Pressing a reset signal tells the circuit that the machine has recycled and the bin cans are full again. The programming and modeling of the top level instantiation with the low level bin cans component explores completely the full capability of VHDL tool.

V. COMPLEX PROGRAMMABLE LOGIC DEVICES (CPLDS)

The CPLD architecture, (Ultra 37256-CY37256P208-154 NC) was used in this work. It is an In-system Reprogrammable™, with 256 microcells, 154MHz maximum frequency and a Temperature range 0°C to +70°C on a Plastic Quad Flat Package (PQFP). Because they offer high speeds and a range of capacities, CPLDs are useful for variety of applications, from implementing random glue logic to prototyping small gate arrays. One of the most common uses in industry at this time, and a strong reason for the large growth of the CPLD market, is the conversion of designs that consist of multiple SPLDs into a smaller number of CPLDs. CPLDs can realize reasonably complex designs, such as graphics controller, LAN controllers, UARTs, cache control, and many others. As a general rule-of-thumb, circuits that can exploit wide AND/OR gates, and do not need a very large number of flip-flops are good candidates for implementation in CPLDs.

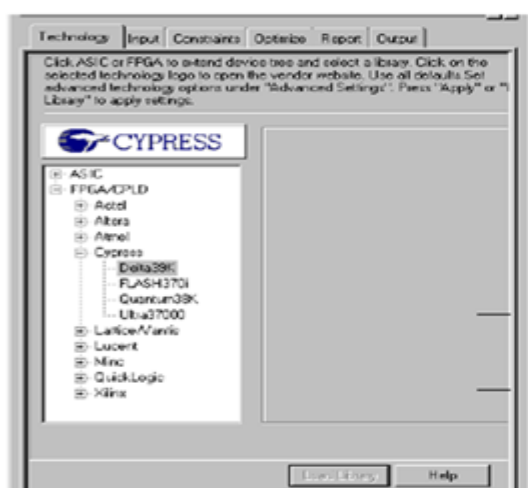


Figure 5: Technology tab in Leonardo Spectrum

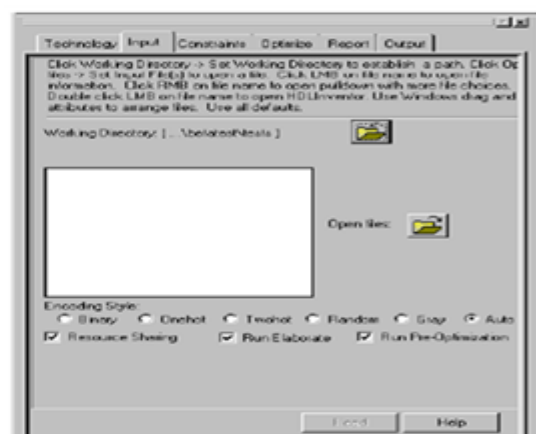


Figure 6: The Input tab in Leonardo Spectrum



Figure 7: The Optimization tab in Leonardo Spectrum

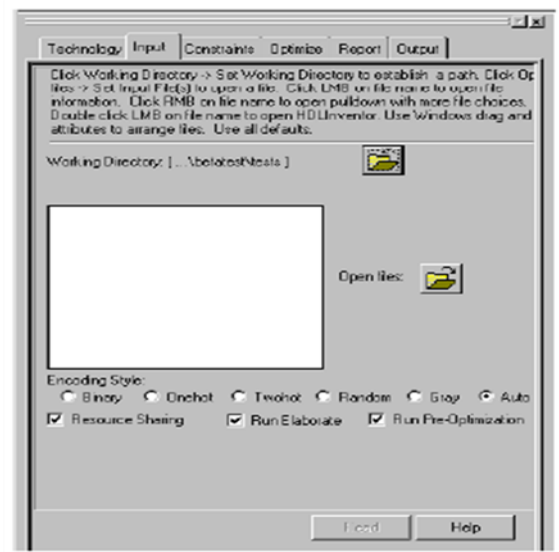


Figure 8: The Output tab in Leonardo Spectrum

A significant advantage of CPLDs is that they provide simple design changes through re-programming (all commercial CPLD products are re-programmable). With In-system Programmable CPLDs it is even possible to re-configure hardware (an example might be to change a protocol for a communications circuit) without power-down. Figure 9 shows a development board for experimentation of ALDS counter and Figure 10 shows the ALDS board interfacing for a prototype



Figure 9: System Development Board

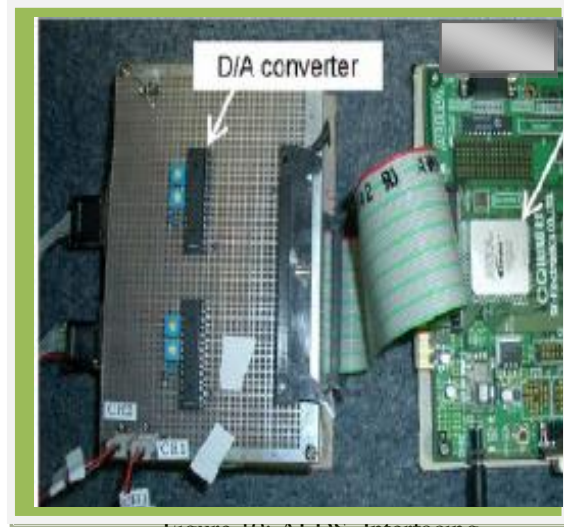


Figure 10: ALDS Interfacing

VI. COMPENSATION POWER DESIGN DISCUSSION FOR ALDS

With the increase in FPGA/CPLD performance functions and density, its power consumption is now a key design consideration and must meet certain industry standards for maximum allowable power. Because of these power limitations, ALDS chip functional density, can be limited by power consumption at the chassis board, or CPLD level. Also, power consumption is directly related to thermal consideration, which needs to be critically analyzed to keep the system working within the temperature specifications of the various components. The reliability decreases as some parts are operated at higher temperatures, so keeping the temperature lower is basically reasonable.

Essentially, there are two primary types of power consumption in FPGA/CPLDs: static and dynamic power. Static power is consumed due to transistor leakage while dynamic power is consumed by toggling nodes as a function of voltage, frequency, and capacitance. In our context, we understand both power types and how

each varies under different operating conditions so that they can be properly optimized to meet the design power budget. In static power drain, the leakage current starts to be fairly significant at 90 nm for ALDS core, and becomes even more challenging at 65 nm. To obtain higher performance from the transistor, the threshold voltage (V_T) of the transistor, which also increases leakage, is lowered. The variation in leakage and static power is about 2-to-1 between worst case and typical process. Static and leakage power are also strongly influenced by core voltage ($V_{CC_{INT}}$), with variations that are approximately the square and cube of $V_{CC_{INT}}$. Static power shows an approximate 15% increase with only a 5% increase in $V_{CC_{INT}}$. Leakage power is strongly influenced by junction (or die) temperature (T_J).

Dynamic power is the power consumed during switching events in the core or I/O of the ALDS core. The parameters for dynamic power computation includes the number of toggling transistors and traces, capacitance, and toggling frequency as shown in equation 1. Transistors, used for logic and programmable, interconnect between metal traces in the ALDS core. In context, the capacitance consists of transistor parasitic capacitance and metal interconnect capacitance.

The formula for dynamic power is given by [7]:

$$P_{DYNAMIC} = n C V^2 f \quad (1)$$

where:

n = number of toggling nodes

C = capacitance

V = voltage swing

f = toggle frequency

Voltage also has significant effect on dynamic power.

Thermal Considerations and Reliability

Consideration must be given to thermal management at both the component and system levels to ensure that all devices are operating within their specified temperature range and to maximize overall system reliability. The device's operating temperature (junction temperature) is a function of the device power and its ability to transfer the resultant heat to the surrounding environment via the component packaging, and the ambient temperature of the system. Reducing the device power consumption, therefore, has two significant benefits. First, it lowers system cost by enabling the use of less expensive thermal solutions to keep the device in its intended operating range. Second, reduced power means lower operating temperatures, which directly translates into improved component and system reliability. Additionally, the low power operation of the CPLD Ultera series also leads to a lower junction temperature. This in turn improves component and system reliability. This is best illustrated by examining the lifetime acceleration factor with respect to temperature. For example, compare two devices with only a 10°C difference in junction temperature:

• Device 1: $T_J = 76^\circ\text{C}$ (349°K)

• Device 2: $T_J = 86^\circ\text{C}$ (359°K)

Using a 0.75eV activation energy, the ratio of lifetime acceleration factors (AF) between Device 1 and Device 2 is given by Arrhenius equation [8] as:

$$AF = \frac{t_{f1}}{t_{f2}} = e^{\left[\frac{E}{k} \left(\frac{1}{T_1} - \frac{1}{T_2} \right) \right]} \quad (2)$$

where:

k = Boltzman's constant

E = Activation energy

T_1, T_2 = Maximum device junction temperature

Thus, a 10°C difference in junction temperature can result in a 2-times acceleration in time to failure. This acceleration illustrates the reliability advantage of lower temperature operation for any system component. Combined, the lower power device operation and the thermally efficient packaging technologies in Ultera devices facilitate cost-effective and high reliable designs.

Optimizing Designs for Power Consumption in ALDS CPLD Cores

To optimize the power consumption in any design, certain things can be done independent of the design contained within the core (CPLD/FPGA). Understanding the prevailing environment, e.g., operating temperature and core voltage, is therefore important.

Temperature Control

Controlling temperature not only helps with reliability but also reduces static power. A reduction in junction temperature from 100°C to 85°C reduces static power by approximately 20%, as shown in Figure 11. The static power of Ultra CPLDs is already reasonable. However, reducing it by another 20% is valuable because in some designs, the static power of the FPGA represents a sizeable portion (30-40%) of the total power budget. A reduction in junction temperature can be achieved by increased airflow and larger heat sinks. As observed, the reduction in junction temperature has an additional benefit of increasing reliability as depicted in Figure 11. Static power is a function of die temperature and also a function of how much power the device is consuming, the thermal properties of that device, and its package. Consequently, ALDS core chip ability to transfer the resultant heat to the surrounding environment, via the component packaging, is very important. Heat flows out of the die from the top of the CPLD and into the package balls and PCB, hence it is advisable to carry out a detailed study and understanding of the system model (PCB, FPGAs, heat sinks, airflow, and other components in a system).

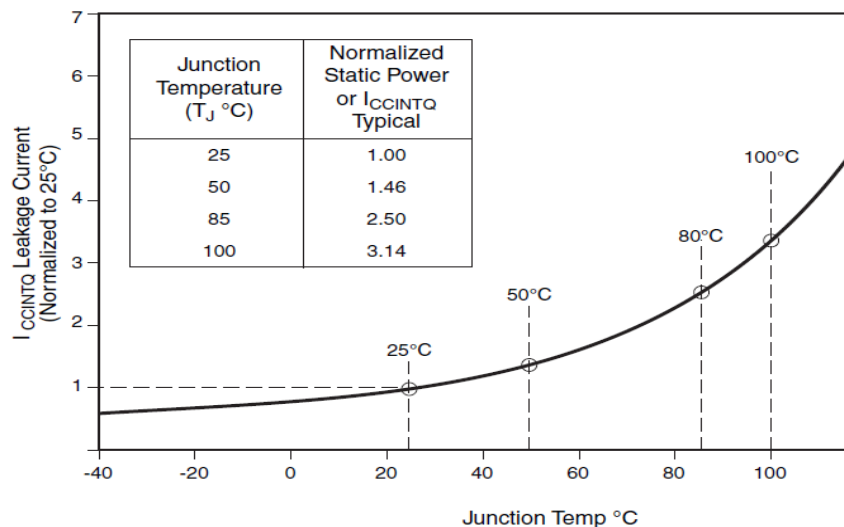


Figure 11: A metric Plot of ICCINTQ vs. Junction Temperature with Increase Relative to 25°C.

Use of Resets

It is a general practice to specify a global reset, many atimes an asynchronous global reset is used. In context, the use of a global reset results in less efficient and higher power ALDS design. This paper recommends the use of resets only when necessary for the proper function of the design and to use only synchronous resets.

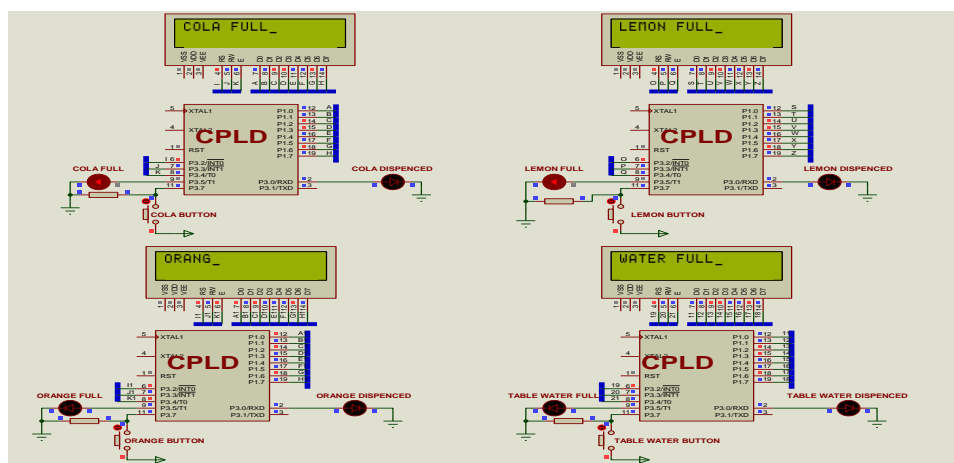


Figure 12: A protus Expanded schematics for ALDS

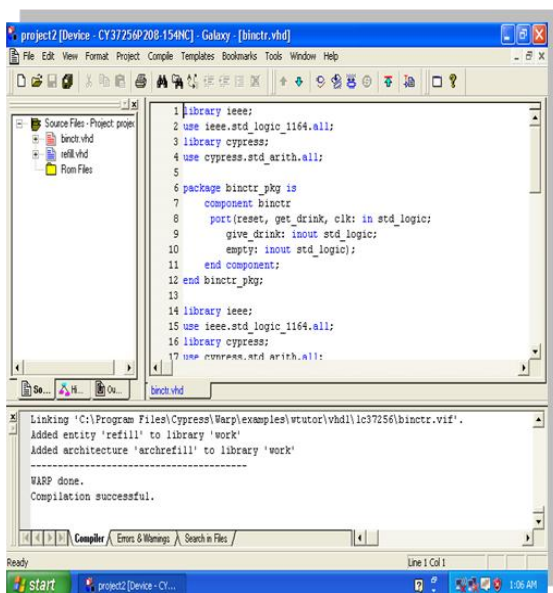
In ALDS, coding for a reset for any inferred register was avoided. It was also observed that coding a reset for CPLD is not necessary for initialization because there is already a dedicated reset for this purpose. So if global reset is coded solely for that purpose, the reset is redundant and consumes logic and routing resources

that could be made available to other functions of the design. This resource usage also indirectly contributes to the overall power budget; the description of the reset, particularly an asynchronous reset, can result in the consumption of more LUTs and register resources as well as less optimal routing choices due to the reset logic utilization. Designs that incorporate a global reset versus those that do not are found to consume on average 4% more registers, 6% more LUTs, and up to 18% more routing resources. This increase is attributed in part to greater Shift Register LUT.

VII. PROGRAMMING METRICS AND RESULTS

The device-independent architecture used for this work is CPLD Ultra 37256-CY37256P208-154 NC as stated earlier and programmed with VHDL. Synthesizing the VHDL description for the ALDS facilitates the development of the ASIC version of the system. For the ASIC ALDS, the programming methodology used includes:

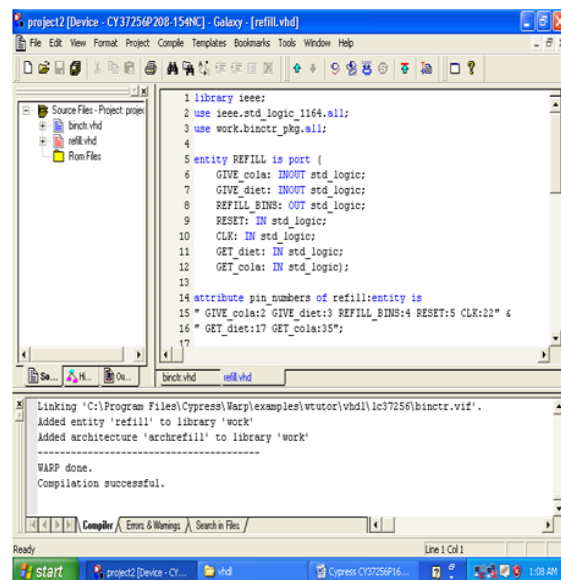
- a) Writing the VHDL code for Entity, Architecture, Component and package declarations for the LDC system behavioral description.
- b) Writing VHDL Entity declaration and Architecture that instantiates the two bin cans into a top level VHDL file via the logic Instantiator



```

1 library ieee;
2 use ieee.std_logic_1164.all;
3 library cypress;
4 use cypress.std_arith.all;
5
6 package binctr_pkg is
7     component binctr
8         port (reset, get_drink, clk: in std_logic;
9              give_drink: inout std_logic;
10             empty: inout std_logic);
11     end component;
12 end binctr_pkg;
13
14 library ieee;
15 use ieee.std_logic_1164.all;
16 library cypress;
17 use cypress.std_arith.all;
    
```

Figure 13: Low level BIN_CAN.vhd Synthesis code



```

1 library ieee;
2 use ieee.std_logic_1164.all;
3 use work.binctr_pkg.all;
4
5 entity REFILL is port (
6     GIVECola: INOUT std_logic;
7     GIVE_diet: INOUT std_logic;
8     REFILL_BINS: OUT std_logic;
9     RESET: IN std_logic;
10    CLK: IN std_logic;
11    GET_diet: IN std_logic;
12    GET_Cola: IN std_logic;
13
14    attribute pin_numbers of refill:entity is
15    " GIVE_Cola:2 GIVE_diet:3 REFILL_BINS:4 RESET:5 CLK:22" &
16    " GET_diet:17 GET_Cola:15";
17
    
```

Figure 14: High level REFILL.vhd Synthesis code, (Jedec File generated for CPLD)

- c) Running WARP [6] on windows 7 platform to compile and synthesize the LDC system design description.

From figure 1, there are two major bin can circuits for VHDL description. These circuits were declared as components and their corresponding packages defined. Consequently, a top level description that instantiates the two bin can circuits was implemented via the logic instantiator. After that, the low level bin can circuits and the high level logic instantiator were compiled and synthesized into the device independent Ultra CY37256 JEDEC file. Figures 4, 5, 6, 7, 8 show the results of synthesis and optimization processes for the LDC system in WARP. Warp is a [VHDL](#) low cost development system for [CPLD](#) by [Cypress Semiconductor](#) Corporation. Warp contains an interactive simulator ([Aldec](#)) and a compiler (Galaxy) [6]. Figure 12 shows an expanded diagram for four different kinds of drinks. Following the programming methodology explained above, Figure 13 shows a successful compilation of the low level BIN_CAN description. In this regard, a description of a bin can circuit in IEEE standard VHDL code that controls the operation of bin cans as components with predefined packages is made before the top level instantiation for the LDC system. Following the programming methodology explained above, Figure 14 shows a successful compilation of the high level BIN_CAN description. Essentially, a description of a top level circuit in IEEE standard VHDL code that instantiates the two bin cans for the LDC system was realized while Figures 15 and 16 show the linking process and report file generation from the logic instantiator.

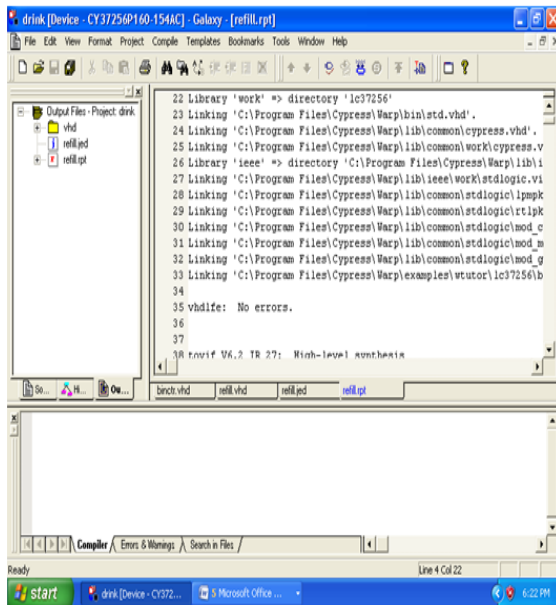


Figure 15: Compilation and optimization report file for Top level instantiator

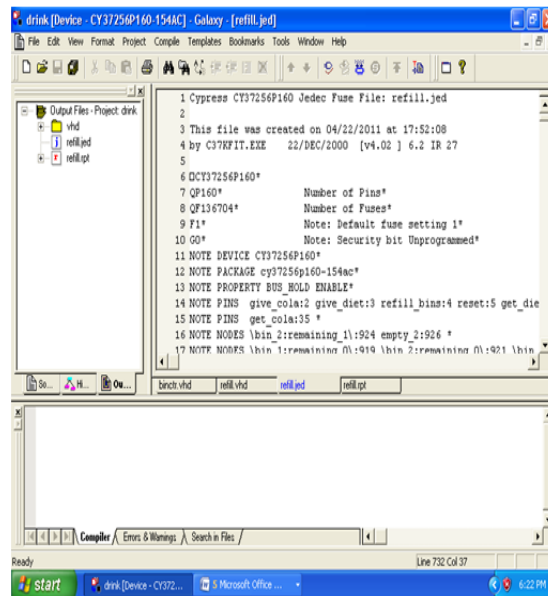


Figure 16: Cypress CY37256P160 Jedec Fuse File, REFILL.jed

CONCLUSION

This paper has presented ALDS model using Altera CPLD chip for optimized power drain. It outlined synthesis WARP integration framework for the system and presented the compilation results. The choice of I/O standards for various interfaces on the CPLD development board can have a large influence on power consumption. I/O standards that consume static power should be avoided when possible. This work is classified as a high-performance system, as such power consideration is indispensable. It has a direct impact on power supply and thermal component considerations. Also, the device temperature is directly tied to system reliability. It is imperative to know the system power budget and operating environment. Understanding where various forms of power consumption occur and allow for the adjustment of the ALDS core environment and design characteristics will minimize power consumption and meet the expected power budget. The effect is a stable and reliable product. By implementing the power design considerations in this work, one can further reduce CPLD power consumption to meet the design budget, thus decreasing overall system cost and enhancing system reliability.

REFERENCES

- [1] R. K. Dueck, Digital design with CPLD applications and VHDL....., 2005.
- [2] P. Abusaidi, M. Klein, and B. Philofsky, White Paper: Virtex-5 FPGAs- Virtex-5 FPGA System Power Design Considerations, WP285 (v1.0) February 14, 2008
- [3] K. C. Okafor, O. O. Akinyele and G. C. Ezekwe. Hardware Description Language (HDL): An Efficient Approach to Device Independent Designs for VLSI Market Segments. In *Proceeding of the 3rd IEEE international Conference on Adaptive Science & Technology (ICAST 2011)*, Abuja, 24th-26th, November 2011 .pp. 262 – 267.
- [4] VHDL MIPS Synthesis and Simulation using Altera CAD Tools, [online]: www.ece.gatech.edu/users/hamblen/ALTERA/mipsrgb. Visited on 8th April, 2013.
- [5] M. Crastes and A. Fonkoua. Hdls and logic synthesis. Technical report.
- [6] Warp (Cypress) - Wikipedia, the free encyclopedia, [online]: [http://en.wikipedia.org/wiki/Warp_\(Cypress\)](http://en.wikipedia.org/wiki/Warp_(Cypress)). Visited on 8th April, 2013.
- [7] Power Considerations in FPGA Design; A Lattice Semiconductor White Paper. February 2009, [online] <http://www.latticesemi.com/products/fpga/ecp3/lowpower.cfm> Visited on 8th April, 2013.
- [8] K. Laidler. The Development of the Arrhenius Equation. *J. Chem. Edu.*, 61(6), 1984, pp. 494.

AUTHORS PROFILE

1. Jonathan U. Agber received M.Sc. degree in Electromechanical Engineering from the Moscow Power Engineering Institute, Moscow, Russia in 1976, PhD degree in Electrical Engineering from the University of Newcastle-Upon-Tyne in the UK in 1985. He is a Senior Lecturer in the Department of Electrical/Electronics Engineering, The Federal University of Agriculture, Makurdi, Benue State, Nigeria. He is a registered Engineer and Member of the Nigerian Society of Engineers. His research interest includes CAD, simulation and control of incremental motion devices and software development.

2. Johnson Akura Mise is a lecturer with the Department of Electrical/Electronic Engineering at University of Agriculture, Makurdi. He received his M.Eng degree from the University of Nigeria Nsukka. His research interest includes semiconductor devices and nano-technology.
3. Okafor Kennedy. C. is a Senior R & D Engineer. He has B.Eng & M.Engr in Digital Electronics & Computer Engineering from the University of Nigeria Nsukka, (UNN). He is a Cisco expert and currently works with the ICT Department of Electronics Development Institute, ELDI. He is a member of IEEE, IAENG and NCS.