Low Power Configuration Logic Block Design using Asynchronous Static

Debarshi Datta, Partha Mitra, Avisek Sen

Abstract— Low power Configuration Logic Block (CLB) for FPGA is highly desirable in VLSI circuit and system. The CLB is the main block of any FPGA architecture. Each CLB block consists of three static LUT's for implementing NCL logic function. 27 fundamental NCL logic gates are implemented in each LUT. The proposed CLB has 10 inputs and 3 different outputs, each with resettable and inverting variations. There are two operating modes in each CLB, Configuration mode and Operation mode. The NCL FPGA logic element is simulated at the transistor level using 130nm TSMC CMOS process technology.

Index Terms— Configuration Logic Block (CLB), Field Programmable Gate Array (FPGA), Look Up Table (LUT), NULL Conventional Logic (NCL).

I. INTRODUCTION

There are large numbers of synchronous circuit used in semiconductor industry, there are major limiting factors to design approach, including clock distribution, increasing clock rate, decreasing feature size and excessive power consumption. It is well known asynchronous circuit require less power than synchronous circuit. At logic level, adopting NCL logic is a way to reduce power dissipation in System-on-Chip (SoC) as only portions of the system doing useful work will consume power. The NULL state in NCL logic is an inherent and automatic power idle state. The clock driving power eliminated. NCL systems operate entirely in terms of synchronized waveforms of level transition in one direction.

Field-Programmable Gate Arrays (FPGAs) use the high circuit densities in modern process to construct ICs, that, as their name suggests, are completely programmable even after a product is shipped or in the field. FPGA are relatively simple design, more predictable project cycle, low cost and field programmable. In FPGA design flow does not require time consuming floor plan, place and route, timing analysis because the design logic already synthesized to be placed onto a verified, characterized FPGA device. To reduce power dissipation of FPGA device, asynchronous based NCL logic are implemented on it. The circuit will become robustness and very low power.

The rest of this paper is organized as follows. Section II describes the NCL overview. After that, CLB design and implementation come from Section IV. Simulation and Result analysis is obtained from Section VI. Section VII concludes this paper.

Manuscript received on March, 2013.

Debarshi Datta, Electronics & Communications, Brainware Group of Institutions, Kolkata, India.

Partha Mitra, Electronics & Communications, Brainware Group of Institutions, Kolkata, India.

Avisek Sen, Electronics & Communications, Brainware Group of Institutions, Kolkata, India.

II. NCL LOGIC

NCL is a self-timed logic paradigm in which control is inherent in each datum. NCL follows the so-called weak conditions of Seitz's delay-insensitive signaling scheme that "all inputs of a combinational circuit must be null before all output become null" along with the condition that "all input of the circuit must be data before all outputs become data [1-2]. A major advantage of NCL is its delay-insensitivity, making timing analysis unnecessary.

NCL circuit consists of 27 fundamental gates. These 27 fundamental NCL logic gates perform al l functions with four or fewer number of inputs. All the NCL are state holding capacity. NCL uses threshold gates as its basic logic elements [3]. The primary type of threshold gates, shown in Figure 1, is the THmn gate, where $1 \leq m \leq n$. THmn gates have m inputs, where at least m of the inputs must be asserted before the output will become asserted. The gate threshold value m of a THmn gate is written inside of the gate.

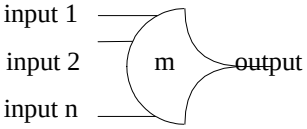

Figure 1: THmn threshold gate

Another type of threshold gate is weighted threshold gate that is denoted as $THmnWw_1w_2w_3\ldots w_R$. Weighted threshold gates have integer value $m \geq wR > 1$ applied to input R. Here $1 \leq R < n$, where n is the number of inputs; m is the threshold; and $w_1\ w_2,\ w_3 \ldots .\ w_R$, each > 1, are the integer weights of input 1, input 2, input R, respectively. For example, consider TH34W2 gate, whose n=4 inputs are labeled A, B, C and D as shown in Figure 2. The weight of A is 2. The threshold value of gate is 2, this represents that output to be asserted, either inputs B, C, D must all be asserted or inputs B, C, D. NCL threshold gates are designed with hysteresis state-holding capability, and such that all asserted inputs must be de-asserted before the output will be de-asserted. This ensures a complete transition of inputs back to NULL before asserting the output associated with the next wave front of input data. NCL gates may also include a RESET input to initialize the gate output to 0 or 1. Circuits' diagrams designate this by denoting a d or n after the threshold label inside the gate. A d represents that output rail is reset to data or 1 and n indicates that output is reset to NULL or 0.

Figure 2: TH34W2 weighted gate

Table I: 27 Fundamental Ncl Gates

NCL Gate	Boolean Function	Transistors (static)	Transistors (semi-static)
TH12	A + B	6	6
TH22	AB	12	8
TH13	A + B + C	8	8
TH23	AB + AC + BC	18	12
TH33	ABC	16	10
TH23w2	A + BC	14	10
TH33w2	AB + AC	14	10
TH14	A + B + C + D	10	10
TH24	AB + AC + AD + BC + BD + CD	26	16
TH34	ABC + ABD + ACD + BCD	24	16
TH44	ABCD	20	12
TH24w2	A + BC + BD + CD	20	14
TH34w2	AB + AC + AD + BCD	22	15
TH44w2	ABC + ABD + ACD	23	15
TH34w3	A + BCD	18	12
TH44w3	AB + AC + AD	16	12
TH24w22	A + B + CD	16	12
TH34w22	AB + AC + AD + BC + BD	22	14
TH44w22	AB + ACD + BCD	22	14
TH54w22	ABC + ABD	18	12
TH34w32	A + BC + BD	17	12
TH54w32	AB + ACD	20	12
TH44w322	AB + AC + AD + BC	20	14
TH54w322	AB + AC + BCD	21	14
THxor0	AB + CD	20	12
THand0	AB + BC + AD	19	13
TH24comp	AC + BC + AD + BD	18	12

A. Static NCL Library

The NCL Static Library consists of the static implementation of 27 fundamental NCL gates given in Table 1. The NCL threshold gates are designed with hysteresis state-holding capability, such that after the output is asserted, all inputs must be de-asserted before the output will be de-asserted. Therefore, NCL gates have both set and hold equations, where the set equation determines when the gate will become asserted and the hold equation determines when the gate will remain asserted once it has been asserted. The set equation determines the gate's functionality as one of the 27 NCL gates, as listed in Table 1, whereas the hold equation is the same for all NCL gates, and is simply all inputs ORed together. The general equation for an NCL gate with output Z is: Z = set + (Z*● hold), where Z* is the previous output value and Z is the new value. Take the TH23 gate for example. The set equation is AB + AC + BC, as given in Table 1, and the hold equation is A + B + C; therefore the gate is asserted when at least 2 inputs are asserted and it then remain asserted until all inputs are de-asserted. To implement an NCL gate using CMOS technology, an equation for the complement of Z (i.e. Z') is also required, which is general form is: Z' = reset + (Z*' ● set'), where reset is the complement of hold (i.e., the complement of each input, ANDed together), such that the gate is de-asserted when all inputs are de-asserted and remains de-asserted while the gate's set condition is false. For the TH23 gate, the reset equation is A'B'C' and the simplified set' equation is A'B' + B'C' + A'C'. Directly implementing these equations for Z and Z', after simplification of an NCL gate, as shown in Figure 3 for the TH23 gate. This requires the output, Z, to be feed back as input to the NMOS and PMOS logic to achieve hysteresis behavior. Due to the large transistor count they also dissipate more Power as compared to Semi- Static NCL gates.

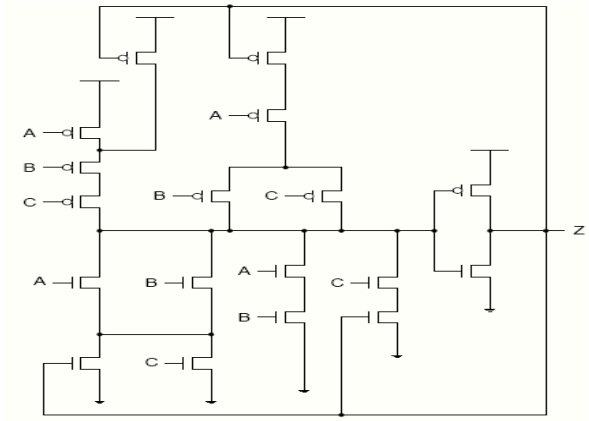

Figure 3: Static CMOS implementing of TH23 gate

$$Z = AB + BC + AC$$

III. CLB DESIGN AND IMPLEMENTATION

The proposed CLB design supports 10 logical input variables (A, B, C, D, E, F, G, H, Din_0, Din_1) and supports three different outputs (X, Y and Z). Each output comes with resettable and inverting variations. Figure 4 shows CLB block diagram. The CLB consists of: 3 X Static LUT, Decoder, Output Reset Logic, Output Inversion Logic, and Programmable Muxes. The CLB has two modes which are Configurations and Operating mode. In Configuration mode the 3 LUT's are programmed to implement different or similar functions and Output resets, Inversion logic along with the programmable multiplexers [muxes] are also configured. Once configured the CLB is ready for operation and should operate as the programmer wants it to.

A. Static LUT

The reconfigurable logic portion consists of a 5 bit address LUT, shown in Figure 5. The static LUT contains 27 NCL static fundamental gate and 28 muxes The gate inputs, A, B, C and D are connected to each of the 27 gates and the programmed Dp values decides which output pass to the LUT output through the MUX logic. Since all gate inputs (i.e. A, B, C and D) are connected to a series of NCL static gates, the LUT function output will be logic 1

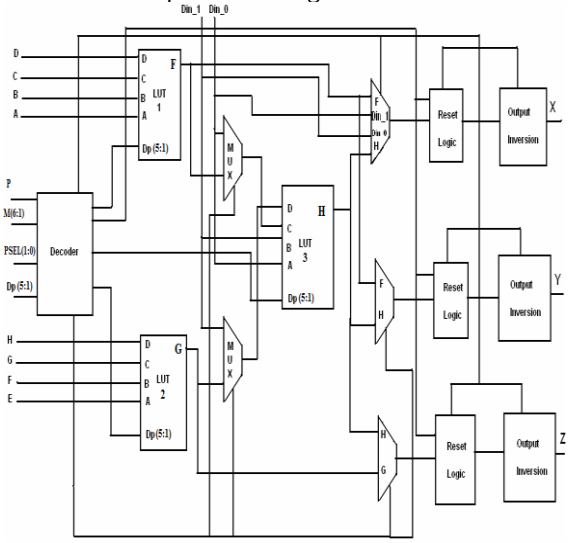

Figure 4: Static CLB Block Diagram

only when the selected gate's output is logic 1. The LUT is outputting logic 0 for Address. There no hysteresis logic as the Static gates are designed with internal feedback. To configure this LUT as a specified NCL gate, the LUT should be programmed with corresponding Dp for any set of inputs corresponding to the gate's set condition, shown in Figure 3. Take for example a TH23 gate, whose equations is AB + AC + BC.

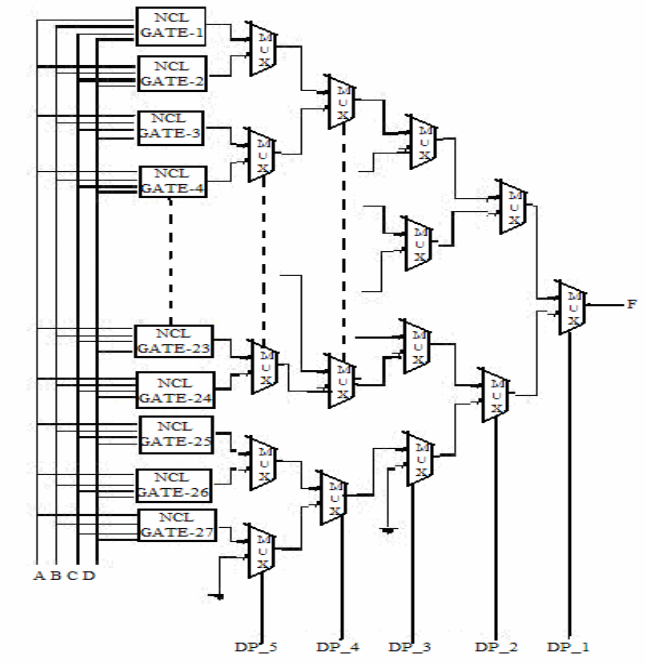

Figure 5: 4-input Static LUT

The LUT should be programmed with Dp ="00111". The LUT outputs logic 1 for the following four input patterns: ABC = 011, 101, 110 and 111 which correspond to setting condition of TH23 gate. The other four combinations ABC = 000, 001, 010 and 100 corresponding to logic 0 output for gate TH23 based on its previous state. For gates with less than four inputs, the unused inputs are not connected. Hence, for the TH23 gate, D would be unconnected to TH23 gate.

B. Decoder

The decoder operates in the Programming mode of the CLB i.e. when P is asserted. Since we have 3 LUT's each of which has a 5 bit programming input Dp, to reduce the number of CLB programming inputs we have designed a decoder. The decoder depending on the two bits input select line muxes the input Dp values to the corresponding LUT.

C. Output Reset and Inversion Logic, Programmable muxes

The proposed CLB has reset logic per output port. The Reset circuit is show below in Figure 6. The reset logic consists of a programmable latch and transmission gate MUX.

During the programmable phase when P is asserted (nP is de-asserted), the latch stores the value, Rv, that the gate will be reset to rst is asserted. rst is the MUX select input, such that when it is logic 0, the output of the PUPD function passes through the MUX to be inverted and output on Z; and when rst is logic 1, the inverse of Rv is passed through the MUX. In this way the CLB provides the user the option of resetting the output ports. The proposed CLB has inversion and hysteresis logic per output port. The Reset circuit is show above in Figure 6. The output inversion logic also consists of a programmable latch and transmission gate MUX.

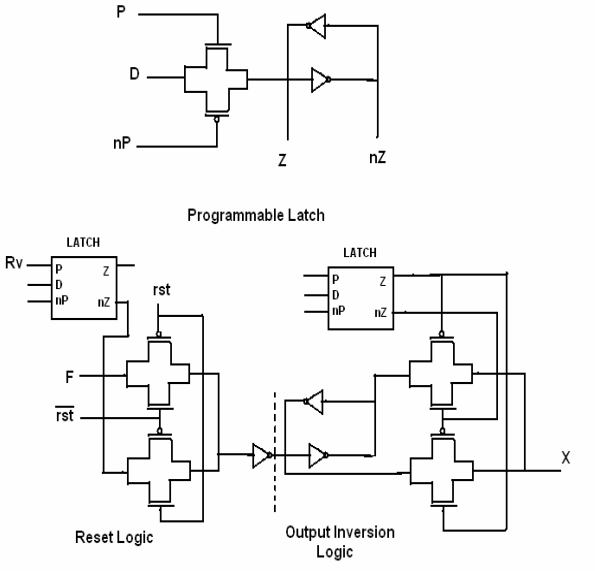

Figure 6: Output Inversion and Reset Logic

The programmable latch stores Inv during the programming phase, which determines if the gate is inverted or not. There is no need for hysteresis if the hysteresis logic is already implemented in each of the 27 fundamental Static gates used to make the LUT. The output and its inverted value are both fed as data inputs to the MUX, so that either the inverted or non-inverted value can be output, depending on the stored value of Inv, which is used as the MUX select input. The CLB design includes 5 programmable muxes so that the user has possible combinations at the outputs. During the programming phase i.e. signal P asserted and nP de-asserted when the Decoder select lines are "11", the select lines for these programmable muxes are driven. The user can have a combination Din_0, LUT1 output, LUT2 output, LUT 3 output at the output ports. For example, consider a case where LUT1 is configured as AND gate, LUT2 as OR gate, LUT3 as XOR gate and the user can have a combination of LUT1, LUT3 and LUT2 output at X, Y and Z by programming the muxes accordingly.

IV. SIMULATION AND RESULT

The proposed designed in this paper has been simulated using VCS compiler and synthesized using Synopsys Design Compiler under TSMC 130-nm CMOS process technology using 1.5V power supply [11]. The power consumption of single LUT is 2.465 X 10^{-9}Watts. The total CLB power dissipation is 8.465µW. The propagation delay is 1.71ns.

V. CONCLUSIONS AND FUTURE WORK

In this work, we have designed and implemented Asynchronous Static NCL library at all levels of abstraction, from Design schematic to cell area, using semi-static NCL gates. The CLB was configured for three different outputs and was successfully simulated and verified to be functionally correct. Furthermore, all of the other major components were implemented, simulated and verified at the transistor-level and physical-level. Future work includes optimizing the design for area and also calculates total delay.

Also the results can be verified for further complex configuration.

ACKNOWLEDGMENT

The authors would like to thank Advanced VLSI Design Laboratory, IIT Kharagpur for their co-operation and support.

REFERENCES

1. Indira P. Dugganapally, Waleed K. Al-Assadi, Tejaswini Tammina and Scott Smith, "Design and Implementation of FPGA Configuration Logic Block Using Asynchronous Static NCL," IEEE Region 5 Conference, pp. 1-6, 2008.
2. S. C. Smith, "Design of Logic Element for implementing an Asynchronous FPGA," IEEE Transaction on VLSI Systems, Vol. 15/6, June 2007.
3. S. Hauck, S. Burns, G. Borriello and C. Ebeling, " An FPGA for Implementing Asynchronous Circuits," IEEE Design & Test of Computer, Vol. 11, No. 3, pp 60-69, 1994.
4. R. E. Payne, "Self-Timed FPGA Systems," 5th International Workshop on Field Programmable Logic and Applications, pp. 21-35, 1995.
5. C. Traver, R. B. Reese and M. A. Thornton, "Cell Designs for Self-Timed FPGAs," 14th Annual IEEE International ASIC/SOC Conference, pp. 175-179, 2001.
6. J. Teifel, R. Manohar, " An Asynchronous Dataflow FPGA Architecture," IEEE Transactions on Computers, Vol. 53, No. 11, pp. 21-24, 2002.
7. C. G. Wong, A. J. Martin, and P. Thomas, "An Architecture for Asynchronous FPGAs," IEEE International Conference on Field Programmable Technology, pp. 170-177, 2003.
8. K. Meekins, D. Ferguson, M. Basta, "Delay Insensitive NCL Reconfigurable Logic," IEEE Aerospace Conference, Vol. 4, pp. 1961-1966, 2002.
9. D. H. Linder and J. H. Harden, "Phased logic: supporting the synchronous design paradigm with delay-insensitive circuitry," IEEE Transaction on Computers, Vol. 45/9, pp. 1031-1044, 1996.
10. Synopsys Design Compiler 2007 User Guide.

AUTHORS PROFILE

Debarshi Datta obtained his B.Tech in Electronics and Communication Engineering and M.Tech in Microelectronics and VLSI Technology under West Bengal University of Technology. He has over 5 years of experience in teaching and his area of interest includes Analog and Digital Communication, Low power VLSI circuit design and Digital Signal Processing.

Partha Mitra received his Bachelors' degree in electrical Engineering and M.Tech in Electronics and Communication Engineering .His research interests include Low power circuit design and Digital Signal Processing..

.

Avisek Sen received his B.Tech in Electrical and Electronics Engineering and M.Tech in Communication Engineering under West Bengal University of Technology. His research interests include Low power circuit design, Digital Signal Processing and Nanotechnology.