

Static Implementation Of A Null Convention Logic Based Exponent Adder

¹Anitha Juliette Albert and ²Seshasayanan Ramachandran

¹Research Scholar, Centre for Research,
Anna University, Chennai, Tamilnadu, India, 600025
anideni2002@gmail.com

²Associate Professor, Faculty of Information and Communication Engineering,
College of Engineering, Anna
University, Chennai, Tamilnadu, India, 600025
seshasayanan@annauniv.edu

Abstract

This paper focuses on developing a transistor level design and implementation of a 8 bit exponent adder, using the asynchronous Null Convention Logic paradigm. Exponent adder plays a significant role of adding the exponents during the process of multiplication of floating point numbers. The design is simulated in VHDL using 1.8V 0.18 μ m TSMC CMOS NCL process libraries. At the transistor level, the design is simulated and implemented using 2.5V 0.25 μ m CMOS process. The proposed Null Convention Logic exponent adder, realized using static version of NCL gates is compared with its equivalent synchronous version in terms of power and area. The proposed exponent adder offers 39% reduction in power, with a compromise of 26.9% increase in area when compared with its equivalent synchronous version.

Keywords: Null Convention Logic, adder, floating point, low power, asynchronous

1. Introduction

International Technology Roadmap for Semiconductors (ITRS) has predicted that asynchronous paradigms will dominate semiconductor industry on a very large scale. With the increasing demand for higher performance, greater complexity and decreased feature size, there will be a steady shift from synchronous to asynchronous design styles, in order to increase circuit robustness, decrease power and overcome clock related issues[1]. Delay insensitive, asynchronous Null Convention Logic (NCL) paradigm yield average delay performance rather worst case delay performance offered by bounded delay asynchronous and traditional synchronous paradigms[1].

This paper addresses the design and transistor level implementation of a NCL exponent adder. Addition of exponents is a significant operation involved in the process of multiplying floating point numbers. Hence, the proposed design can be used as a library component that performs the addition of 8 bit exponents and further subtracting the bias(127) from the addition result[2]. The NCL exponent adder was first designed at the gate level using standard NCL design techniques. NCL VHDL library developed with delays based on physical level simulations of static gates, designed using TSMC's 1.8V 0.18 μ m CMOS technology was used[3]. VHDL simulation of the design was performed to verify the functionality of the circuit. Secondly, the transistor level design of NCL threshold gates was performed using 2.5V 0.25 μ m generic CMOS technology in a schematic editor tool. Using, the basic NCL gates, the transistor level design of the NCL exponent adder was developed and simulated to obtain the power, delay and area. The performance metrics were compared with an equivalent synchronous design of an exponent adder.

The paper is structured as follows: Section II provides the background survey of NCL. Section III outlines the details about the conventional synchronous adder, used to add exponents of floating point numbers. Section IV describes the NCL exponent adder design and implementation at gate level and transistor levels of abstraction. Section V details the results and performance comparisons. Section VI concludes the paper.

2. NCL Literature

NCL circuits are delay insensitive, input complete, observable and possess hysteresis state holding capability. Delay insensitivity specifies that the NCL circuits operate correctly regardless of when the circuit inputs are available. No timing analysis is required. Delay insensitivity is achieved by the use of dual rail or quad rail signals. A Dual rail signal, D consists of two wires or rails D⁰ and D¹ [ref]. These signals take any value from the data set {DATA0, DATA1, NULL} as illustrated in Table 1[1]. The DATA1 state represents Boolean logic 1, DATA0 state represents Boolean logic 0, NULL state represents empty set which means that the data is not available. The two rails are mutually exclusive meaning that they cannot be asserted simultaneously. If assigned, it is called an illegal state[1].

Table 1: DUAL RAIL SIGNAL

	DATA0	DATA1	NULL	ILLEGAL
D ⁰	1	0	0	1
D ¹	0	1	0	1

NCL circuits are constructed using 27 fundamental gates with hysteresis[1] as illustrated in Table II. NCL circuits are comprised of 27 threshold gates with hysteresis [1]. The primary type of Threshold gates called as TH_mn gate is shown in Figure 1.

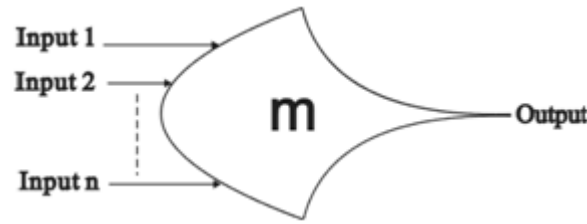


Fig.1 THmn Threshold gate

THmn gates have n inputs. At least m of the n inputs must be asserted before the output will become asserted. m is the threshold input of the gate. NCL gates have this distinct advantage of hysteresis state holding capability. Hysteresis ensures that all inputs transit back to NULL before asserting the output of next wavefront of input data[1]. Static NCL threshold gate is characterized by Set, Reset, hold0 and hold1 equations[1]. The Set equation determines the gate’s functionality as one of the 27 NCL fundamental gates and specifies when the gate will become asserted. The hold equation specifies when the gate will remain asserted once it has been asserted. Hold1 equation specifies the ORed operation of all inputs. Hold0 equation is the complement of set. Reset equation is the complement of hold equation(the complement of each input ANDed together).

The general equations 1 and 2 represent the output Z and complementary output Z’ of an NCL gate.

$$Z = \text{set} + (Z^- \cdot \text{hold1}) \tag{1}$$

$$Z' = \text{reset} + (Z^{-'} \cdot \text{hold0}) \tag{2}$$

where Z^- is the previous value of the output is, $Z^{-'}$ is the complement of the previous value of the output and Z is the new value.

For the th34w2 gate, the equations for the 4 transistor networks are represented in equations 3, 4, 5 and 6 respectively.

$$\text{set} = AB + AC + AD + BCD \tag{3}$$

$$\text{hold1} = A + B + C + D \tag{4}$$

$$\text{hold0} = (AB + AC + AD + BCD)' \tag{5}$$

$$\text{reset} = (A + B + C + D)' \tag{6}$$

The CMOS implementation of the static th34w2 gate is shown in Figure 2. Input-completeness illustrates that all outputs must not transit from NULL to DATA or DATA to NULL until all inputs have transited from NULL to DATA or DATA to NULL[1].

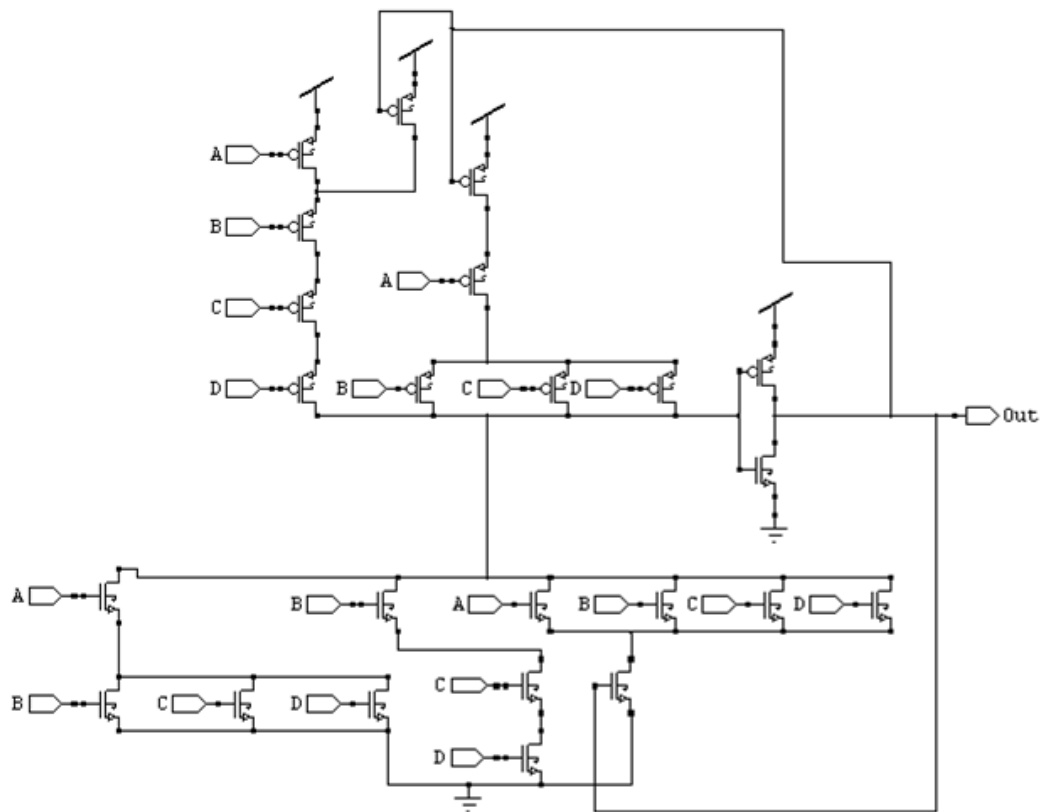


Fig.2 CMOS implementation of static th34w2 gate

3. Existing synchronous exponent adder

The existing synchronous exponent adder is an 8 bit ripple carry adder followed by a 9 bit subtractor[2], sandwiched between D-flip flops. The inputs and outputs, observed through these D-flip flops are controlled by a synchronous clock signal. The architecture of the exponent adder realized using synchronous architecture is shown in Figure3. One half adder and seven full adders are used to obtain the structural description of the exponent adder. The half adder and full adder are constructed using basic gates. This exponent adder performs the addition of 8 bit exponents: $(X + Y) - \text{Bias}$ [2].

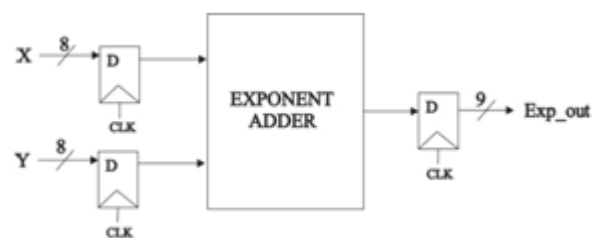


Fig. 3 Synchronous exponent adder

This paper focuses on structural and transistor level implementation of a NCL based exponent adder, to overcome the clock related issues as discussed in [1]. Hence the proposed NCL exponent adder, realized using static NCL gates, will dissipate much lower power when compared to its synchronous counterpart.

4. Design and implementation

The design flow utilized in the design and characterization of the NCL exponent adder is depicted in Figure 4. A gate level architecture of the 8 bit exponent adder was designed. The design was executed in two dimensions. First, a structural gate level description of the design was developed using VHDL. The NCL gates, used for developing the gate level design used gate delays based on physical simulations of the gates with TSMC's 1.8V 0.18 μm CMOS technology. The design was tested for functionality using a testbench that generates $2^8 \times 2^8 = 65536$ test vectors. Secondly, NCL static threshold gates based on 2.5V 0.25 μm CMOS technology libraries were developed using S-edit of Tanner EDA tool. Using the designed NCL threshold gates, transistor level description of the NCL exponent adder was developed. The power, average and area were estimated. Equivalently, a synchronous version of the exponent adder was developed. The proposed NCL exponent adder was compared with its equivalent synchronous version in terms of power, delay and area.

4.1 Design flow

The design flow utilized in the design and characterization of the NCL exponent adder is depicted in Figure 4. A gate level architecture of the 8 bit exponent adder was designed. The design was executed in two dimensions. First, a structural gate level description of the design was developed using VHDL. The NCL gates, used for developing the gate level design used gate delays based on physical simulations of the gates with TSMC's 1.8V 0.18 μm CMOS technology. The design was tested for functionality using a testbench that generates $2^8 \times 2^8 = 65536$ test vectors. Secondly, NCL static threshold gates based on 2.5V 0.25 μm CMOS technology libraries were developed using S-edit of Tanner EDA tool. Using the designed NCL threshold gates, transistor level description of the NCL exponent adder was developed. The power, average and area were estimated. Equivalently, a synchronous version of the exponent adder was developed. The proposed NCL exponent adder was compared with its equivalent synchronous version in terms of power, delay and area.

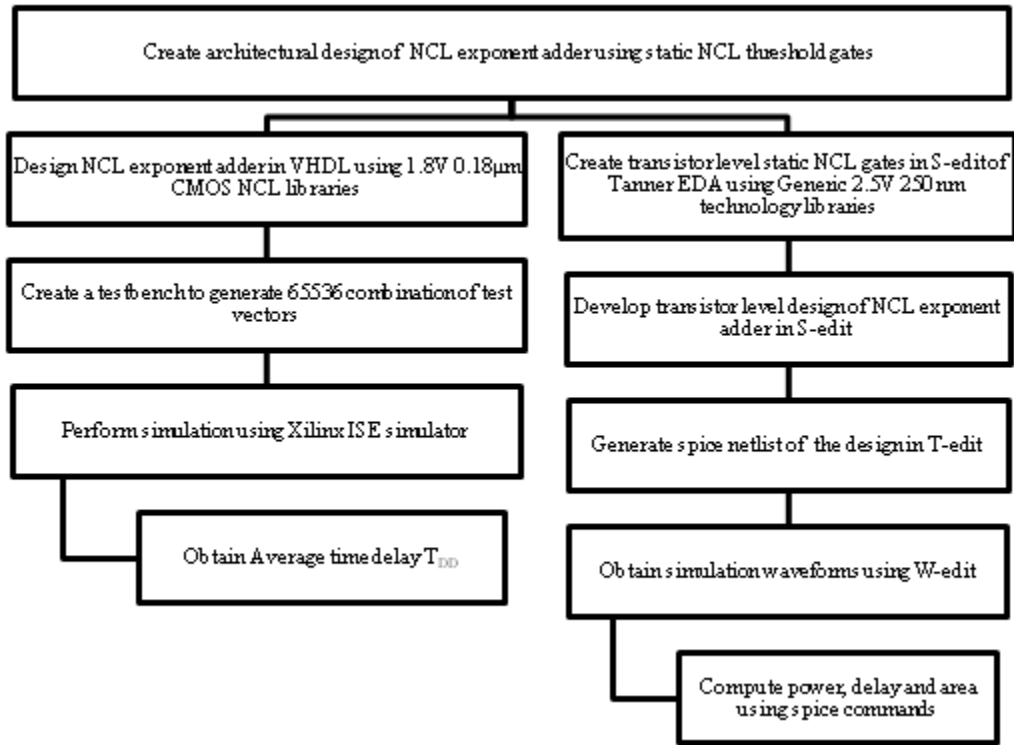


Fig.4 NCL Exponent adder design flow

2 Proposed architecture

The architecture of the proposed NCL exponent adder is presented in Figure 5. The block diagram of the NCL exponent adder consists of two blocks- NCL ripple carry adder, to add the exponents and a NCL subtractor, to perform subtraction of bias(127) from the result of NCL ripple carry adder. One NCL half adder and seven NCL full adders are cascaded to realize the NCL ripple carry adder. The NCL half adder and NCL full adder are input complete and consist of 7 and 12 gates respectively. The 8 bit inputs of the NCL exponent adder are the exponent bits of floating point numbers. The inputs X and Y are passed through the 16 bit DI register. The output of the 16 bit DI register is passed to the combinational part of the NCL exponent adder, whose 9 bit output is fed to the NCL subtractor[5]. Seven NCL one bit subtractors (NCL OS) and two NCL zero bit subtractors (NCL ZS) are cascaded to realize the NCL ripple borrow subtractor[5]. The result of the subtractor is passed through the 9 bit DI register. The output of the 9 bit DI register is the final output Exp_out of the exponent adder. The DI registers on the top and bottom of the combinational NCL exponent adder pass the DATA/NULL wavefronts through them. The full word completion component consists of 4 th33 gates. Its output is '1' (request for data), when all the inputs are '0'. Its output is '0' (request for NULL), when all inputs are '1'. The flow of wavefronts is controlled by the request and acknowledge signals k_i and k_o . Initially, upon Reset, all the DI registers are set to NULL. This forces the acknowledge signals (9 bit) k_o of the lower DI register to be '0', indicating the completion of NULL

wavefront. The output of the full word completion is ‘1’ which is given to the ki inputs (16 bit) of the top DI register, indicating the request of DATA wavefront from the input. Consequently, when the Reset is asserted low, the top DI register accepts the DATA wavefronts of X and Y. The DATA wavefront is processed by the combinational part and passed on to the lower DI register. This makes the ko bits of lower DI register to become ‘1’, indicating the completion of DATA wavefront. The output of the full word completion component is ‘0’, thereby requesting a NULL wavefront from the input side.

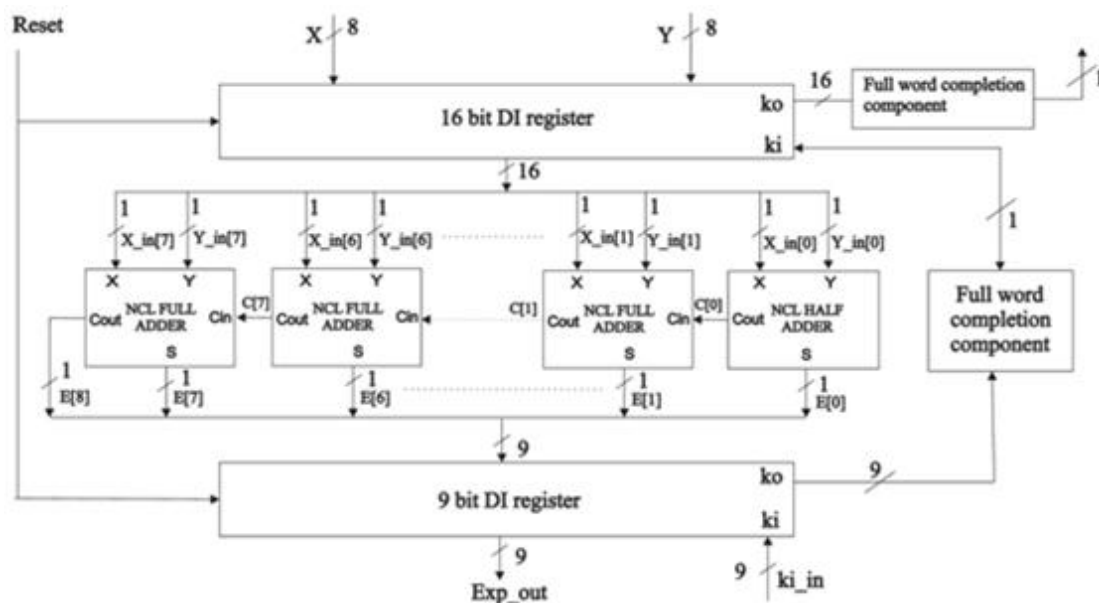


Fig.5 Architecture of NCL Exponent adder

4.3 Transistor level implementation

The static NCL threshold gates required to realize the NCL exponent adder are developed in Tanner EDA tool using 2.5V 0.25µm static CMOS technology libraries. We have created transistor level libraries for the static versions of all NCL gates using 2.5V 0.25µm technology. These NCL threshold gates were instantiated to obtain an NCL full adder, NCL half adder, NCL OS and NCL ZS. The gate level description of the NCL full adder, realized in Tanner EDA is depicted in Figure. An input complete NCL full adder is constructed using 2 th34w2 and 2 th23 gates are shown in Figure 6. The transistor level realizations of the NCL ripple carry adder and NCL subtractor, obtained by the instantiation of the sub modules are shown in Figures 7 and 8 respectively. The full word completion components required to interface the request and acknowledge signals are shown in Figure 5. Full word completion component1 is constructed using 3 th33 NCL gates. The output of this component is fed as request signal to the 16 bit DI register, which accepts the primary inputs. Full word completion component2 is constructed using seven th33 and two th22 NCL gates as shown in Figure 9. NCL half adder, 7 instances of NCL full adders, 7 instances of

NCL OS and 2 instances of NCL ZS and full word completion components are sandwiched between the DI registers to realize the complete transistor level description of the NCL exponent adder as shown in Figure 10.

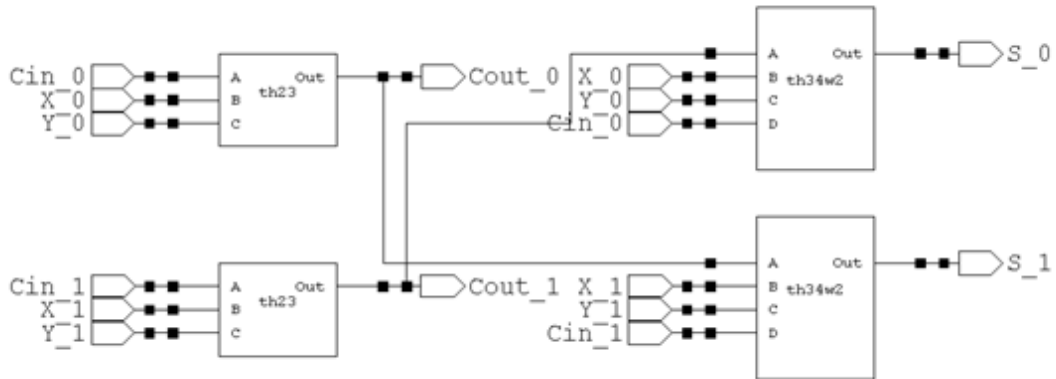


Fig.6 NCL full adder

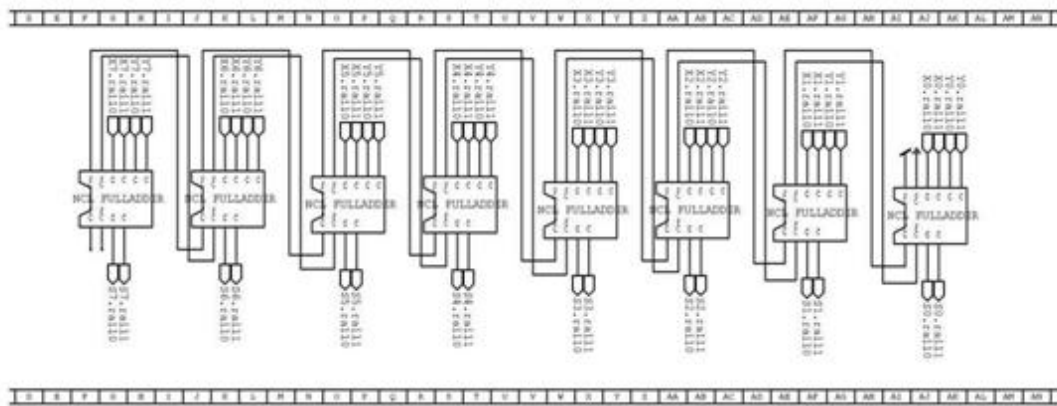


Fig.7 NCL ripple carry adder

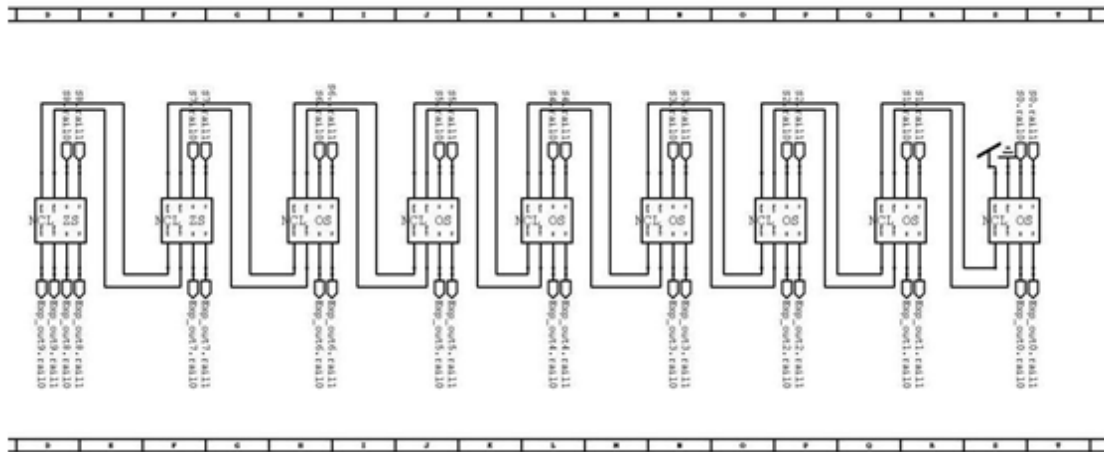


Fig.8 NCL subtractor

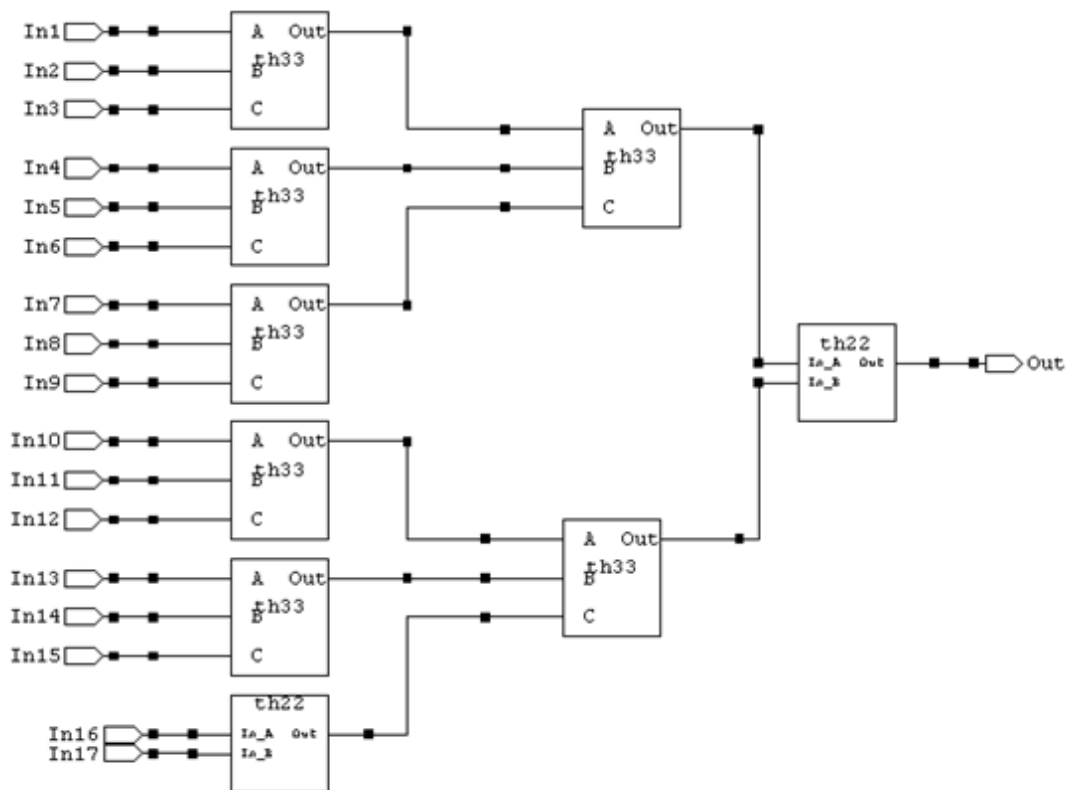


Fig.9 NCL completion component

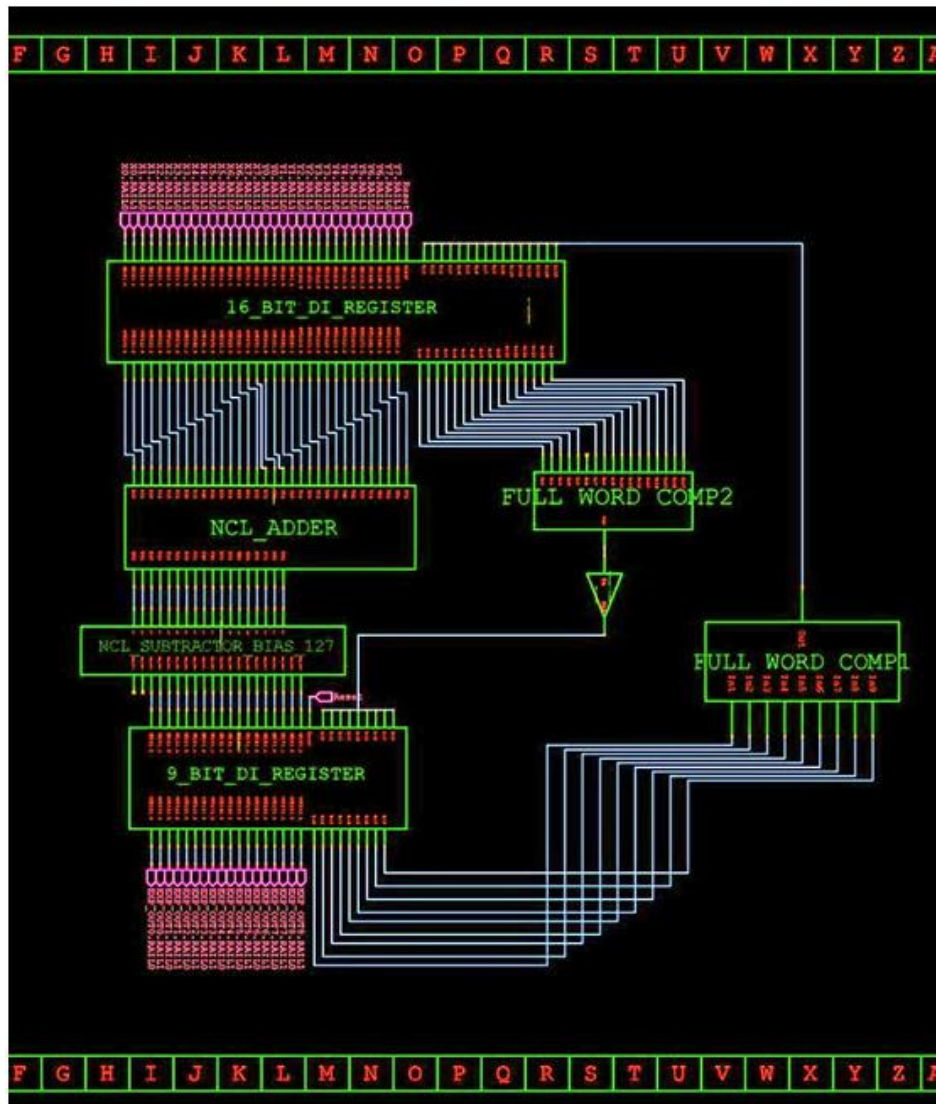


Fig.10 Schematic representation of NCL exponent adder

5. Results and discussions

The VHDL code of the 8 bit NCL exponent adder, sandwiched between the DI registers was developed using the basic threshold NCL gates. It was tested for functional performance using Xilinx ISE simulator. The testbench required to test the 8 bit NCL exponent adder was coded to generate $2^8 \times 2^8 = 65536$ test vectors. The waveform showing the VHDL simulation result is shown in Figure 11. We observe that the output alternates between the DATA and NULL wavefronts. The average delay time T_{DD} obtained was 4.96ns.

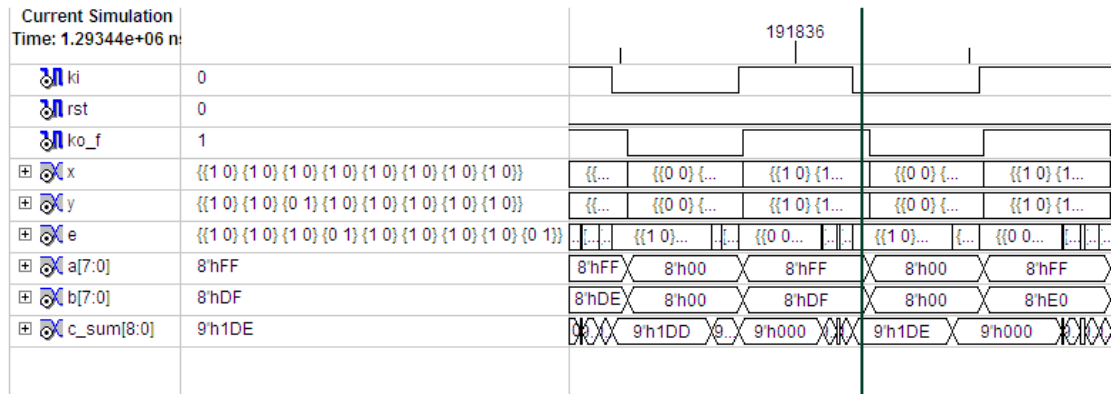


Fig.11 VHDL simulation result of NCL exponent adder

The transistor level design of the proposed adder, sandwiched between the DI registers was developed in S-edit of Tanner EDA tool using generic 250nm technology. The waveforms for the transistor level design of the NCL exponent adder was observed in Waveform editor. The input, output and handshake waveforms for a one bit full adder are shown in Figures 12, 13 and 14 respectively.

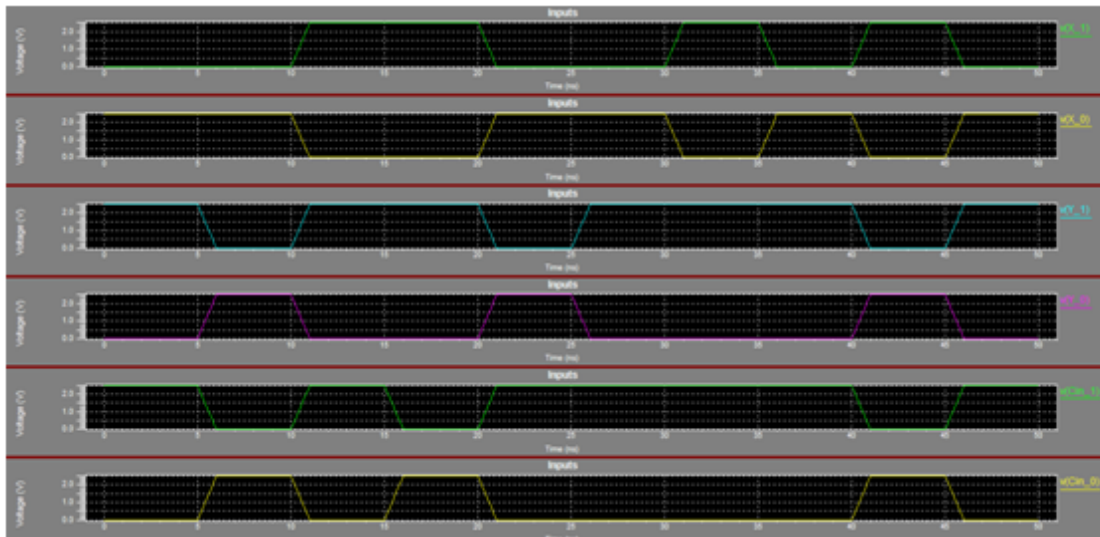


Fig.12 NCL exponent adder inputs-1 bit

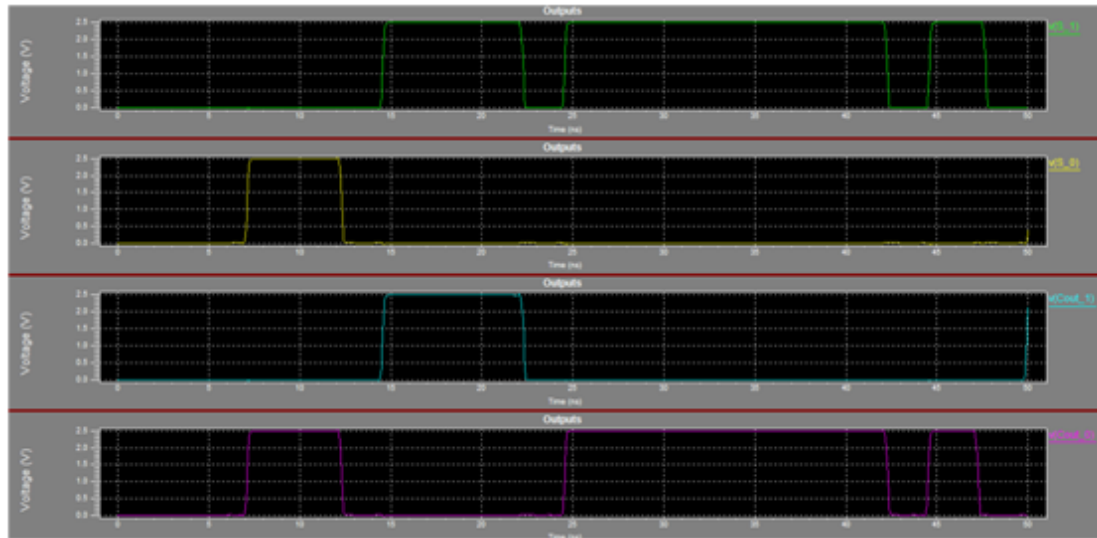


Fig.13 NCL exponent adder outputs-1 bit

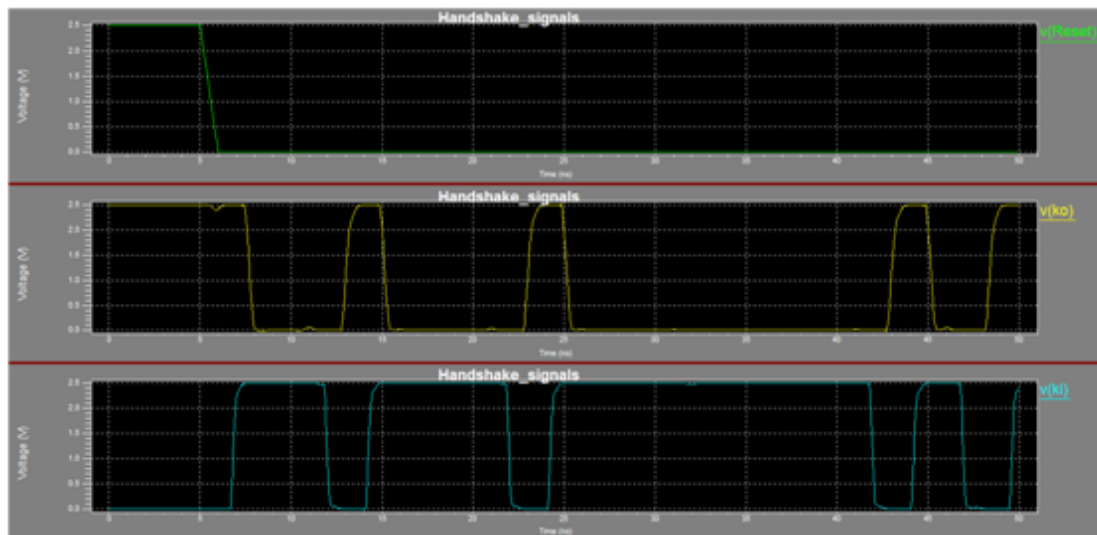


Fig.14 NCL exponent adder- handshake signals

From the waveforms, it is observed that the outputs transit between the DATA and NULL wavefronts in correspondence with the handshake signals k_i and k_o . Average delay T_{DD} of 4.12ns, obtained from the transistor level spice simulation of NCL exponent adder, is obtained as the arithmetic mean of cycle times corresponding to 65536 possible pair of input operands. The average power per operation was determined by performing the spice simulation for randomly selected three set of input operands. Total power of the three operations divide by three, yielded average power consumed per operation of the proposed adder to be 15.26mW. To compare the proposed design with its equivalent synchronous implementation, the synchronous

exponent adder was simulated with the clock period set to 4.12 ns (T_{DD} of exponent adder). This is done in order to make the synchronous exponent adder operate at the same speed as the NCL exponent adder. The results are summarized in Table II. From the figure 15, it is observed that power consumption of the NCL exponent adder is reduced by 48.18%, when compared with its equivalent synchronous implementation. The decrease in power consumption is achieved because of the absence of clock. The switching activity in the proposed adder is greatly reduced, since the circuit switches only when data is available. For asynchronous circuits, switching activity factor ranges between 0.1 and 0.5. Hence, power in NCL based designs is reduced considerably [6]. However, from the results, it is observed that the proposed adder offers 26.9% increase in area, when compared with its synchronous version. However, increase in area is a bottleneck in NCL systems[1]. This increase in area can be comprised, because in synchronous systems, clock distribution network will occupy a significant increase in area, which is totally devoid in NCL systems.

Table II : Comparison of NCL exponent adder and conventional synchronous exponent adder

	Power	Delay	Transistor count
NCL Exponent adder	15.12 mW	4.12 ns	1652
Synchronous Exponent adder	25.46 mW	4.12 ns	1206

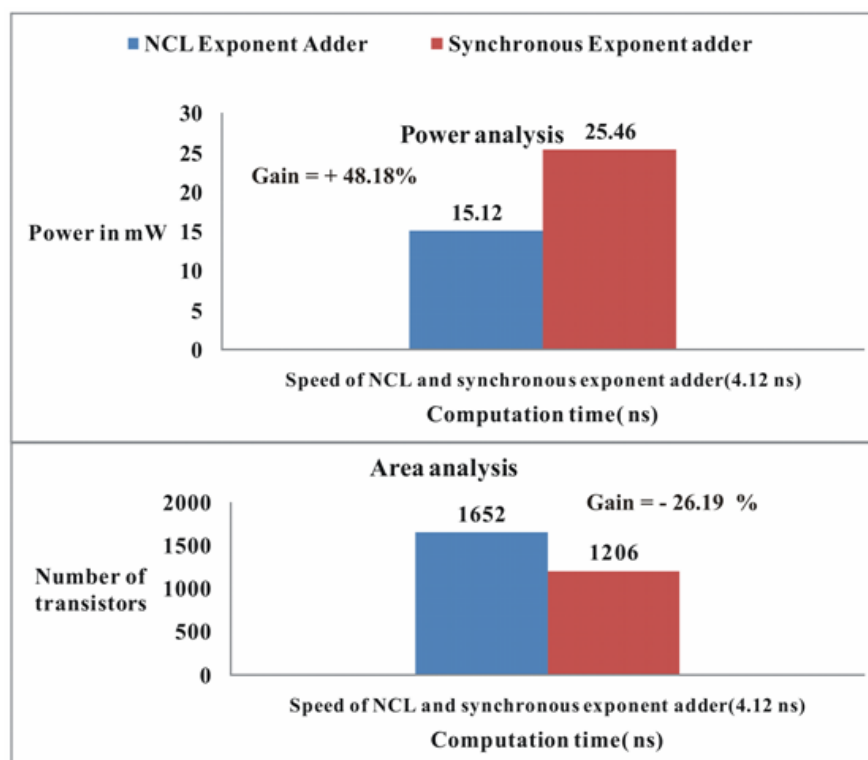


Fig 15 : Power and area analysis

6. Conclusion

A transistor level representation of NCL exponent adder was designed using static NCL gates. The proposed adder dissipated significantly 48.18% lesser power when compared with its equivalent synchronous version, developed using static CMOS design methodology. However, the area of the proposed adder was 26.19 % higher than its synchronous version. In future, the adder can be designed with modified static NCL gates proposed in [6], to obtain a significant increase in speed.

References

- [1]. Scott C. Smith and Jia Di, "Designing asynchronous circuits using NULL convention logic", Synthesis Lectures on Digital circuits and systems, vol.4, no.1, pp. 1-96, 2009.
- [2]. Mohamed Al-Ashrafy, Ashraf Salem, Wagdy Anis, "An efficient implementation of floating point multiplier", SIECPC, pp.1-5, 2011
- [3]. Dr.Scott C. Smith: http://www.ndsu.edu/pubweb/~scotsmit/CCLI_async.html.
- [4]. R. Sankar, V. Kadiyala, R. Bonam, S. Kumar, S. Mohan, F. Kacani, W. K. Al-Assadi, and S. C. Smith, "Implementation of Static and Semi-Static Versions of a Bit-Wise Pipelined Dual-Rail NCL 2s Complement Multiplier", Region 5 Technical Conference IEEE, pp.228-233, 2007
- [5]. Anitha Juliette Albert, Seshasayanan Ramachandran, "Null Convention floating point multiplier", The Scientific World journal, 2015[in press]
- [6]. Farhad A. Parsan and Scott C. Smith, "CMOS implementation of static threshold gates with hysteresis", VLSI-SoC IEEE, pp.41-45, October 2012.