THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

# A Novel Approach of Delay-Insensitive Null Convention Logic Microprocessor Design

**J. Asha Jenova**
Student, ECE Department, Arasu Engineering College, Tamilndu, India
**C. Reginamary**
Assistant Professor, ECE Department, Arasu Engineering College, Tamilndu, India

***Abstract:***
*Null Convention Logic (NCL) has the potential advantage of Delay insensitivity and Low Power when compared to other asynchronous VLSI design methodologies [1]. This paper has arisen focusing on exploiting the advantages of clock less NCL asynchronous design. The paper proposes the design of 8 bit Asynchronous Microprocessor using NCL Method: The Processor consists of various sub modules (like). We have proposed Half adder sub module using NCL. Results: We implemented a half adder where the data was sequenced synchronously and asynchronously, which yielded an average delay of 22.5ns, 7.5ns respectively. The same design methodology can be extended to the other sub modules of processor which can operate at a higher speed when compared with a synchronous processor. Performance analysis: We also made a comparative analysis of various asynchronous designs and NCL based circuit designs. Conclusion: These results provide a firm platform for designing Asynchronous Null convention logic based Microprocessor.*

***Keywords:*** *Delay Insensitivity, Average delay, Average Power Delay Product, Half Adder, VHDL.*

## 1. Introduction

High speed and high performance synchronous systems are dominant in VLSI industry. But the major disadvantages such as excessive clock skew, clock noise and larger power dissipation has led the way to the asynchronous world of VLSI. Asynchronous, clock-less circuits require less power, generate less noise and produce less Electro-Magnetic Interference (EMI) compared to their synchronous counterparts , without degrading performance[1]. NCL is a Delay Insensitive (asynchronous) paradigm [1]. It specifies that the circuit operates correctly regardless of when the circuit inputs are available. No timing analysis is required. Dual rail or Quad rail logic can be used in the design. This paper performs NCL design using Dual rail logic. A Dual rail signal, D consists of two wires or rails D0 and D1 [9]. These signals take any value from the data set {DATA0, DATA1, and NULL} as illustrated in Table 1[1].

| | DATA0 | DATA1 | NULL | ILLEGAL |
|---|---|---|---|---|
| $D^0$ | 1 | 0 | 0 | 1 |
| $D^1$ | 0 | 1 | 0 | 1 |

*Table 1: DUAL RAIL signal*

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].

The architecture for NCL systems consists of DI combinational logic integrated between DI registers as shown in Fig 1 [1]. The working of the NCL framework is similar to a synchronous system but with lesser power dissipation compared to clocked systems.

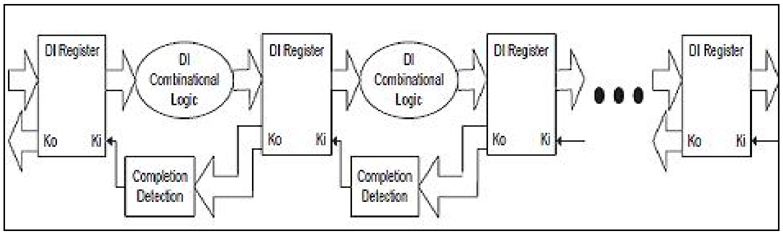

*Figure 1: NCL Architecture*

The gates used to realize an NCL circuit is not the traditional logic gates. Instead 27 Threshold NCL gates are constructed to realize the NCL circuits. The primary type of Threshold gates called as THmn gate is shown in Fig 2 [2]

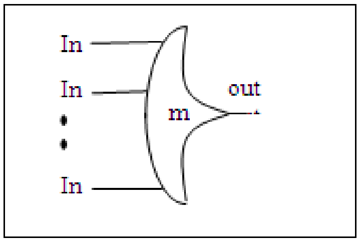

*Figure 2: THmn Threshold gate*

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 wave front of input data. Moreover NCL circuits must be both observable and input-complete inorder to achieve delay-insensitivity. 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].Observability illustrates that no orphans propagate through a gate[1]. An orphan is a wire that transitions during current DATA wave front but is not used in the determination of the output. A library of NCL threshold gates using static CMOS technology has been designed [2]. Using these gates asynchronous NCL modules are designed. The objective is to design an asynchronous MIPS Microprocessor using NCL. The block diagram is hierarchically decomposed until the leaf cells are reached. The NCL designs of individual cells are done. The cells are finally integrated to obtain the entire Asynchronous MIPS Microprocessor design. The Second chapter performs an analysis of existing asynchronous designs and NCL circuits. The third chapter focuses on proposed work of designing an NCL based MIPS Microprocessor and the evaluation parameter of the design. The Fourth chapter discusses about the results and discussions. The paper ends with the Conclusion and future work.

## 2. Analysis of Existing Asynchronous Designs : Litterature Review

Bounded-delay and delay insensitive models are the two major categories of Asynchronous circuits. Bounded-delay models, assume that delays in both gates and wires are bounded [1]. Delays are added based on worse-case scenarios to avoid hazard conditions [1]. This leads to extensive timing analysis of worse-case behavior to ensure correct circuit operation [1]. On the other hand, delay-insensitive circuits assume delays in both logic elements and interconnect to be unbounded [1]. An analysis of the existing Asynchronous designs is presented in Table 2.On analyzing all the methods represented in Table 2, Null Convention logic is found to be the optimized solution for performing asynchronous design for the following reasons:

- The NCL design flow follows the same procedure as the synchronous flow. Hence the design can be easily incorporated in the chip industry [1].
- Because of the property of DI, only minimal delay analysis is required [1].
- NCL have power, noise and EMI advantages when compared to the synchronous counterparts [1].

| ASYNCHRONOUS METHODOLOGY | DELAY MODEL | SIGNAL CONVENTION | ASYNCHRONOUS ELEMENTS | ANALYSIS |
|---|---|---|---|---|
| Micropipelines[3] | Bounded delay model | | Muller C elements | 24 stages Micropipeline Multiplier accepts 24 bit operand pairs before delivering its first product. |
| Self timed processor[4] | Delay Insensitive model | Double rail code to implement ternary logic | Delayed branches using Muller C elements | Hierarchical architecture of a self timed ST-RISC processor incorporating nine architectural and design features |
| Regular expression Recognizer[5] | Delay Insensitive model | Regular binary logic incorporated in 4 cycle signaling convention | Muller C element and 4 phase FIFO element | Self timed Regular expression recognizer replaced the clock in synchronous expression with 4 cycle signaling convention and required the generation of minterms |
| Self timed Boolean functions[6] | Delay Insensitive model | Double rail logic | Muller C element | Self timed Boolean circuits are implemented using Interval Temporal logic |

| | | | | |
|---|---|---|---|---|
| Phased Logic[7] | Delay Insensitive and Level Encoded Two-Phase Dual-Rail(LEDR) signals | | Phased logic gates | Phased logic circuitry was developed from its synchronous counterpart using an algorithm. Marked graph theory was used to reduce the timing complexity of synchronous designs. |
| Null convention Logic[1] | Delay insensitive and Dual rail, Quad rail signals | | NCL gates and Delay insensitive registers | NCL circuitry is entirely independent from its synchronous counterpart achieving Speed and Power benefits. |

*Table 2: Analysis of Existing Asynchronous Designs*

| NCL Circuits | Performance Metric |
|---|---|
| Multiply and Accumulate Unit | NCL based MAC [8] yielded 8.6ns average delay when compared with other self timed MAC's yielding 901ns [9], 90ns [10] and 24ns [11] average delay. |
| Self Timed Multipliers | Quad rail multiplier shows less power dissipation of 0.74nW when compared with 3.34nW Dual rail multiplier[12] |
| Self Timed Dividers | NCL based divider yielded an average cycle time of 64ns which is lower when compared with its synchronous counterparts.[13] |
| Arithmetic Logic Units | Pipelined version shows better area and delay performance when compared with non pipelined versions.[14] |
| Viterbi Decoder | 36.14mw(26 % power reduction when compared with synchronous counterparts)[15] |
| NCL based MIPS Microprocessor design | Proposed research area in this paper which promises Low power and delay insensitivity when compared with its synchronous counterparts. |

*Table 3: NCL BASED CIRCUIT DESIGNS*

## 3. Proposed Work

### *3.1. Design*

The design is partitioned into two top level designs: Controller and Data path. The Controller is decomposed into its sub modules such as the Control FSM (Finite State Machine) and ALU (Arithmetic Logic Unit) decoder. These are built from the leaf cells which are a binary of standard cells such as NAND's, NOR's and Inverters. The Datapath is decomposed into submodules such as 8-bit word slices which are in turn built from adders, multiplexers and flipflops. A complex circuit can thus be decomposed into smaller pieces. A general design hierarchy of the NCL based MIPS processor is shown in the Figure 3.

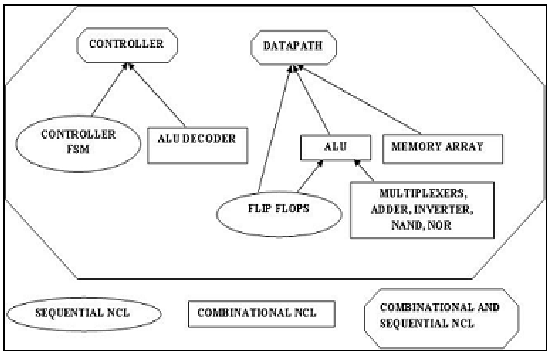

*Figure 3: Design Hierarchy of NCL based MIPS Microprocessor*

The NCL design is to be done starting from the leaf cells of the processor. It can be sequential or combinational. Controller Asynchronous State Machines (ASM) and Flip Flops are to be designed using Sequential NCL approach. ALU, decoder, Multiplexers, adders and logic gates are designed using Combinational NCL approach. The MIPS processor is thus a bundle of combinational and sequential NCL design. Half adder, which is one of the submodules of ALU, has been designed using NCL approach. The truth table of NCL based half adder is shown in Table 4.

| INPUTS | | | | OUTPUTS | | | |
|---|---|---|---|---|---|---|---|
| | | | | SUM | | CARRY | |
| $X^0$ | $X^1$ | $Y^0$ | $Y^1$ | $S^0$ | $S^1$ | $C^0$ | $C^1$ |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |

*Table 4: Truth Table of NCL based Half Adder*

By employing Kannaugh Map Techniques, the equations for the outputs $S^0$, $S^1$, $C^0$ and $C^1$ were realized.
$S^0 = X^0Y^0 + X^1Y1$
$S^1 = X^1Y^0 + X^0Y1$
$C^0 = X^0 + Y^0$
$C^1 = X^1\ Y^1$
The equations can be realized using TH24comp, TH24comp, and TH12 and TH22 NCL gates respectively. The NCL circuit of the half adder developed using the above procedure is shown in Figure 4

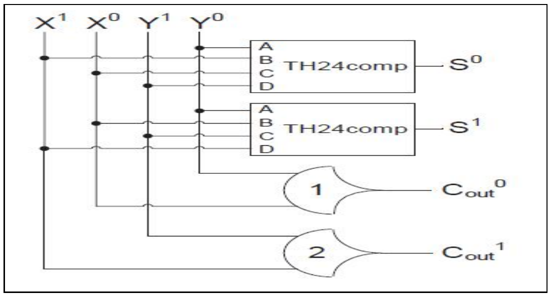

*Figure 4: NCL based Half Adder*

The data transfer from the inputs to the outputs of the NCL adder is done through the DI registers as shown in Figure 5. Data flow from inputs to outputs is sequenced through the DI registers in contrast with the half adder where data is sequenced with conventional clocked registers as shown in Figure 6.

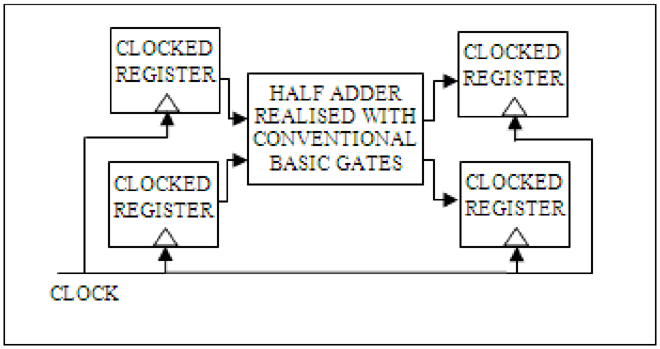

*Figure 5: Data sequencing in NCL based Half Adder*

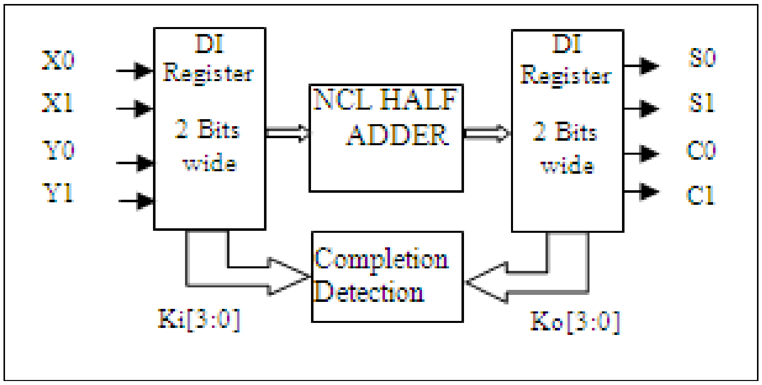

*Figure 6: Data sequencing in conventional Half Adder*

## 4. Results and Discussions

A comparison of Synchronous and NCL based Asynchronous designs of a Half Adder is performed using VHDL. A Combinational NCL based half adder was realised using VHDL. NCL libraries were used for realising the design. Data were passed into and taken out from the NCL Half adder through the DI registers. The NCL libraries developed using 0.18um technology [2] was used in the development of the VHDL code of NCL based half adder. The Synchronous version of half adder is designed by passing the data using Clocked Flip Flops. A clock of 50 MHz is used for simulation. The simulation was performed for 4 data set combinations. The Average delay time is computed by dividing the total simulation time by the number of input combinations. The simulation output for NCL based and Synchronous version of Half Adder are shown in Figure 7 and Figure 8 respectively.

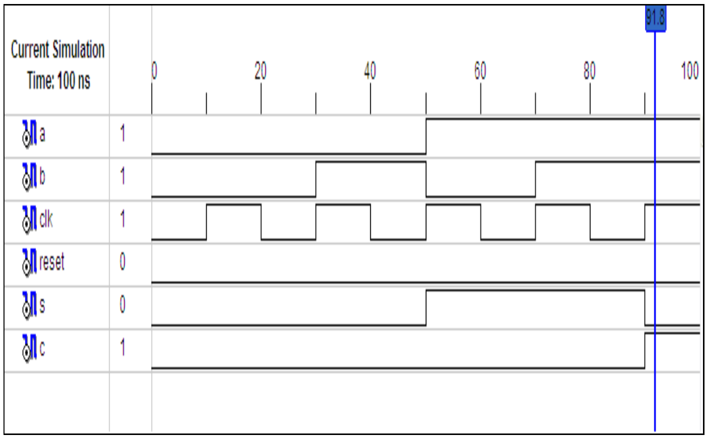

*Figure 7: Simulation output for Synchronous based Half Adder*

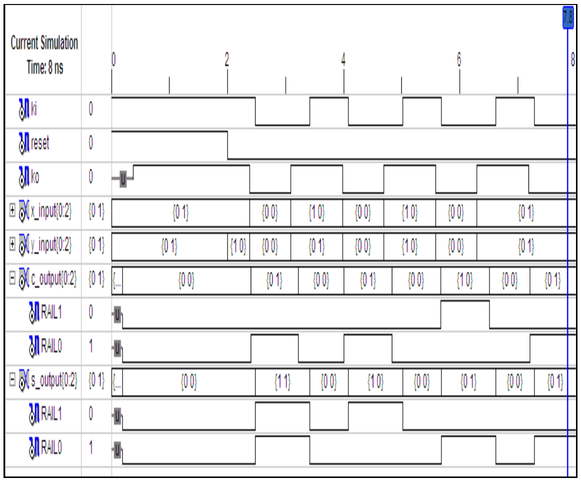

*Figure 8: Simulation output for NCL based Half Adder*

| Design | Average Cycle Time(Ns) |
|---|---|
| SYNCHRONOUS HALF ADDER WITH CLOCKED REGISTERS | 22.5 Ns |
| NCL HALF ADDER WITH DI REGISTERS | 1.75 S |

*Table 5: Results and analysis*

## 5. Conclusion

NCL has the potential advantage of offering delay insensitivity thereby offering lowest Power-Delay product. Hence a NCL based MIPS microprocessor design will present an efficient asynchronous design which will meet the latest needs of Asynchronous design world. This will present the first step in the design of Asynchronous NCL based processors.

## 6. References

1. Scott C. Smith and Jia Di,"Designing Asynchronous Circuits using NULL Convention Logic (NCL)",2009
2. Farhad A. Parsan and Scott C. Smith, "CMOS Implementation of Static Threshold Gates with Hysteresis: A New Approach " VLSI and System-on-Chip (VLSI-SoC), 2012 IEEE/IFIP 20th International Conference on 7-10 Oct. 2012,pg 41 – 45
3. I. E. Sutherland, "Micropipelines," Communications of the ACM, Vol. 32/6, pp. 720–738, 1989.
4. C. L. Seitz, "System Timing," An Introduction to VLSI Systems, Addison-Wesley, pp. 218–262,
5. T.S. Anantharaman,"A Delay Insensitive Regular Expression Recognizer," IEEEVLSITechnical Bulletin, Sept. 1986.
6. I. David, R. Ginosar, and M. Yoeli, "An Efficient Implementation of Boolean Functionsas Self-Timed Circuits," IEEE Transactions on Computers, Vol. 41/1, pp. 2–10, 1992. (2007) The IEEE website. [Online]. Available: http://www.ieee.org/
7. D. H. Linder and J. H. Harden, "Phased Logic: Supporting the Synchronous Design Paradigm with Delay-Insensitive Circuitry," IEEE Transactions on Computers, Vol. 45/9, pp. 1031–1044
8. S.C. Smith, "Development of a large word-width high-speed asynchronous multiply and accumulate unit", INTEGRATION, the VLSI journal 39 (2005) 12–28
9. C.D. Nielsen, A.J. Martin, "Design of a delay-insensitive multiply and accumulate unit", Twenty-Sixth Hawaii International Conference on System Sciences, vol. 1, 1993, pp. 379–388.
10. T. Tang, C. Choy, P. Siu, C. Chan," Design of self-timed asynchronous Booth's multiplier", Asian South-PacificDesign Automation Conference, 2000, pp. 15–16.
11. V.A. Bartlett, E. Grass, "A low-power concurrent multiplier-accumulator using conditional evaluation", Sixth IEEE International Conference on Proceedings of ICECS, vol. 2, 1999, pp. 629–633.
12. S. K. Bandapati, S. C. Smith, and M. Choi, "Design and Characterization of NULL Convention Self-Timed Multipliers," IEEE Design and Test of Computers: Special Issue on Clockless VLSI Design, Vol. 30/6, pp. 26-36, November-December 2003.
13. S. C. Smith, "Design of a NULL Convention Self-Timed Divider," International Conference on VLSI, pp. 447–453, June 2004.
14. S. K. Bandapati and S. C. Smith, "Design and Characterization of NULL Convention Arithmetic Logic Units," The 2003 International Conference on VLSI, pp. 178-184, June 2003.
15. Chien-Ching Lin, Yen-Hsu Shih, Hsie-Chia Chang, and ChenYi Lee,"Design of a Power-Reduction Viterbi Decoder for WLAN Applications", IEEE Transactions on Circuits and System-I: regular papers, 52(6), 321-328G