International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 9 (2015) pp. 23127-23136

http://www.ripublication.com

Gate Mapping and Performance Optimization of Asynchronous Circuits Using NCL

Ms. Neethu.B.S[1], Mr. D. Ruban Thomas[2]
Dept of ECE, Vel Tech Multi Tech Dr.Rangarajan Dr.Sakunthala Engineering College, Chennai
[1]neethubs3@gmail.com
[2]rubanthomas@veltechmultitech.org

Abstract

Clock distribution, increasing clock rates, decreasing feature size, excessive power consumption etc are the major limiting factors to synchronous design even though it dominates the current semiconductor design industry. In asynchronous design each component in the system is not governed by a clock signal so we can eliminate the problem of clock skew. NULL Convention Logic is an example of correct-by-construction asynchronous circuits, which have several advantages compared to their synchronous counterparts. Here we design a number of asynchronous circuits such as 1-bit full adder, register and 4-bit up counter by using different techniques in NCL and also incorporates 'the proposed grouping algorithm' given in [1] and thereby improving and comparing their performance. A 4x4 bus arbiter is then designed entirely in terms of NCL and then compared its performance with its asynchronous counterpart.

Keywords: NCL, arbiter, delay insensitive, up-counter, gate mapping

Introduction

The digital world is dominated by synchronous techniques from many decades due to ease in the design. But with clock speed in GHz ranges the global wire delays is increasing than the gate delay. With increase in wire delay problem related to clock, like clock skew, signal integrity, power, etc becomes more serious in synchronous techniques. Hence designers are evaluating new techniques to overcome wire delays.

In an asynchronous circuit [3] the clock signal is replaced with handshake signals or feedback paths for synchronization and communication between their components. The advantages of asynchronous circuits are: low power consumption due to zero standby power consumption since there is no clock signal, no clock distribution hence no problems of clock routing with minimum skew, high operating speed since it

depends on local latency and not on global latency and less circuit noise since no clock signal. Also Synchronous circuits are implemented using traditional Boolean logic which is not symbolically complete. It has time dependencies as well as symbolic value dependencies. Symbolic values dependency depends upon interconnect in logic and their truth table. Therefore designers are looking towards asynchronous techniques and/or self timed circuits for potential solution. Moreover Boolean logic is dependent on control logic i.e. time. The time dependent expression will depend on propagation of delay to determine correctness of data and control dependent factor depends on the interconnection of different gates to generate control signals. In late 1950's attempts were made to eliminate both the dependencies from the Boolean logic and make it symbolically complete. The attempts were Delay Insensitive [4] (DI) circuits.

Overview of NCL

NULL Conventional Logic [2] is a new technique developed for designing asynchronous circuits. NULL means there is NODATA or spacer between corresponding DATA. It indicates no input or output is present. NCL is theoretically complete and economically viable approach to delay insensitive circuits. NCL is not purely based on DI circuit method but on a class of DI called as quasi delay insensitive (QDI). QDI is designed using Isochoric Forks [5].

By using NCL we can design VLSI devices with reduced power, reduced noise margin, lower electro- magnetic interference and with greater ease. In NCL we eliminates the clocks, whereas retains the control information in the data path. The NCL design process is less complex than traditional asynchronous design since there is no need for worse case delay analysis and delay matching due to its delay insensitivity i.e . control is present with each datum. NCL allows a system to run at its maximum frequency independent of the input.

Traditional Boolean logic is symbolically incomplete in the sense that it is a time dependent as well as symbolic value dependent relationship. Here the output depends on the inputs as well as on time. Time dependent in the sense that it depends on the propagation delay of component elements, and the dependency on the interconnection of the logic gates and their logical behavior refers to the symbolic value dependency.

The symbolic completeness of NCL AND gate is given in fig.1. Here we have considered the concept of dual rail in which a signal D consists of two wires, D^0 and D^1, which may assume any value from the following DATA0, DATA1, and NULL. Let us assume 1nS delay for the output to become valid once the inputs are arrived. Let initially X is DATA1, Y is DATA0, and Z is DATA0. All the inputs must transition to NULL before applying the next set of inputs, which will cause the output to transition to NULL after 1nS. The next input set is applied after the transition of the output to NULL. The output Z will change to the next state corresponding to the inputs after 1nS, signaled by Z transitioning from NULL to DATA. And the output will remain in the same state until both inputs have transitioned to NULL, which is due to the hysteresis behavior of the threshold gate. That is here time is never

referenced to determine the validity of the output and the delay can just be considered as an arbitrary gate delay which does not affect the validity of Z.

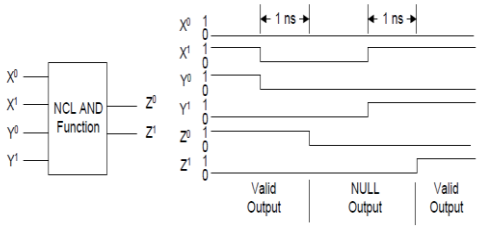

Figure 1: NCL AND Function: Z = X • Y And Associated Waveforms

NCL library includes 27 gates with which we can realize any expression with 4 or fewer variables. The symbolic completeness of inputs in NCL logic design is implemented using state holding threshold gates with hysteresis [6]. The hysteresis behavior is such that, once the set function of the gate is satisfied and its output is asserted, it remains asserted until all its inputs are de asserted. The behavior that the output becomes data if at least M of N inputs have become data refers to the threshold behavior.

This paper work includes the designing of a full adder, an asynchronous register and an up-counter using NCL, in which the proposed grouping algorithm in [1] has been incorporated. Here we have chosen dual rail as well as quad rail encodings which yield the minimum of two wires per bit. We have chosen dual rail optimization for designing the full adder and register, whereas quad rail optimization for the counter. The up-counter is designed by using two different methods which are 'quad rail increment circuit with reduced minterm' and 'quad rail increment circuit employing complex gates' [7]. The performances of the two realizations are then compared using Xilinx ISE simulator [9]. An arbiter is then designed entirely in terms of NCL in which the concept of NCL counter and priority encoder has been used. The performance of the same is then compared with simple 4x4 asynchronous bus arbiter.

Table 1: Library of The Standard NCL Gates

Threshold Gate	**Set Function**	**No. of Transistors**
TH12	A+B	6
TH22	AB	12
TH13	A|B|C	8
TH23	AB+AC+BC	18
TH33	ABC	16
TH23w2	A+BC	14
TH33w2	AB+AC	14
TH14	A+B+C+D	10
TH24	AB+AC+AD+BC+BD+CD	26
TH34	ABC+ABD+ACD+BCD	24

TH44	ABCD	20
TH24w2	A+BC+BD+CD	20
TH34w2	AB+AC+AD+BCD	22
TH44w2	ABC+ABD+ACD	23
TH34w3	A+BCD	18
TH44w3	AB+AC+AD	16
TH24w22	A|B|CD	16
TH34w22	AB+AC+AD+BC+BD	22
TH44w22	AB+ACD+BCD	22
TH54w22	ABC+ABD	18
TH34w32	A+BC+BD	17
TH54w32	AB+ACD	20
TH44w322	AB+AC+AD+BC	20
TH54w322	AB+AC+BCD	21
THXOR0	AB+CD	20
THAND0	AB+BC+AD	19
TH24comp	AC+BC+AD+BD	18

A. Dual rail optimizations

In dual rail optimizations, the input as well as the output is divided into two with two wires D^0 and D^1. The two wires assume one value among DATA0, DATA1 and NULL. In the case of NCL also we can construct K-map for each output as in traditional Boolean circuits. The zeros in the K-map refer to each signals rail0 line and the ones refer to rail1 line of the same. The expressions thus obtained can be mapped to any gate from the 27 NCL macros. This method will always generate two-level logic, where we can OR the minterms together to produce the desired outputs.

B. Quad rail optimizations

Quad rail optimization is actually similar to that of dual rail optimizations which may sometimes lead to more efficient designs. In this case instead of two as in the previous case there must be four wires D^0, D^1, D^2 and D^3 which may assume one value from DATA0, DATA1, DATA2, DATA3 and NULL. Two dual-rail signals yield the same five logic states as one quad-rail signal. Quad-rail optimization follows the same steps as in dual-rail optimization whereas in the K-map it also contains 2s and 3s corresponding to rail2 and rail3 respectively.

1) Quad-rail increment circuit with reduced minterm

Initially the NCL circuit is optimized by using reduced minterm expressions for all four rails of both outputs, S0 and S1, the low order two bits and the high order two bits, respectively, derived from the K-maps. The derived equations can then be directly mapped to TH1n and THnn gates to produce the reduced minterm model.

2) Quad Rail Increment Circuit Employing Complex Gates

Here we map the factored expressions to the full 27 macros in Table 1, reducing both the number of gates and the number of logic levels.

Application of NCL To Various Circuits

A. Application to Full adder

In case of the full adder, after constructing the K-map from the truth table given in table 2, we can obtain the expressions for the functions $C0^0$ and $C0^1$ directly. Whereas in case of the sum functions S^0 and S^1, we have to consider the carry as an intermediate term. From the K-map it is seen that both the carry output functions $C0^0$ and $C0^1$ can directly map to TH23 gate and the sum functions which are based on inputs X, Y, Ci and the intermediate output Co can directly map to TH34W2 gates. There is no significance for the proposed grouping algorithm in [1], since here all the functions can directly be mapped to the gates from the NCL library. Finally the circuit diagram (RTL view) for 1-bit NCL full adder is shown in fig.2.

Table 2: Truth Table of Full Adder In NCL

A^0	A^1	B^0	B^1	C_i^0	C_i^1	S^0	S^1	C_o^0	C_o^1
D	X	D	X	D	X	D	X	D	X
D	X	D	X	X	D	X	D	D	X
D	X	X	D	D	X	X	D	D	X
D	X	X	D	X	D	D	X	X	D
X	D	D	X	D	X	X	D	D	X
X	D	D	X	X	D	D	X	X	D
X	D	X	D	D	X	D	X	X	D
X	D	X	D	X	D	X	D	X	D

B. Application of NCL to register

The asynchronous register shown in Fig.3 is a dual rail register. In the register realization TH22 gates are used, and the proposed grouping algorithm in [1] can be used if required. An NCL register simply resembles the feedback hysteresis behavior of NCL threshold gates. The gate passes a DATA or a NULL value according to the request and will continue to assert the same until all of its inputs are changed.

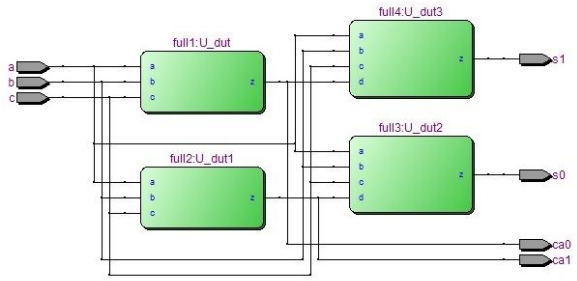

Figure 2: NCL full adder

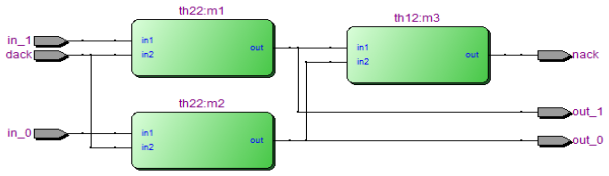

Figure 3: NCL Register

C. Application of NCL to up-counter

Here the input as well as the output is divided into four rails since we are employing quad rail optimization. The actual inputs are X0, X1, and Inc and the outputs are S0 and S1. To increment the count by 1 give Inc=1, and on the other hand to keep the count unchanged keep Inc=0.

1) Quad Rail Increment Circuit with Reduced Minterm

The equations for S00, S01, S02, S03, S10, S11, S12 and S13 are derived from the K-map. The expressions for S00, S01, S02 and S03 can be directly mapped to one of the gates from NCL gate library. Where as in the case of the output S1 we can use the proposed grouping algorithm in [1] for grouping and map it to the 27 NCL macros from NCL gate library. The logic diagram of corresponding NCL technique is given in fig.4.

2) Quad Rail Increment Circuit Employing Complex Gates

Here we added two don't care terms and derives the expressions for S0 and S1. Thereby they can be mapped to TH24comp gates. Then follows the same steps as before. The logic diagram for quad rail increment circuitry employing complex gates is shown in fig.5.

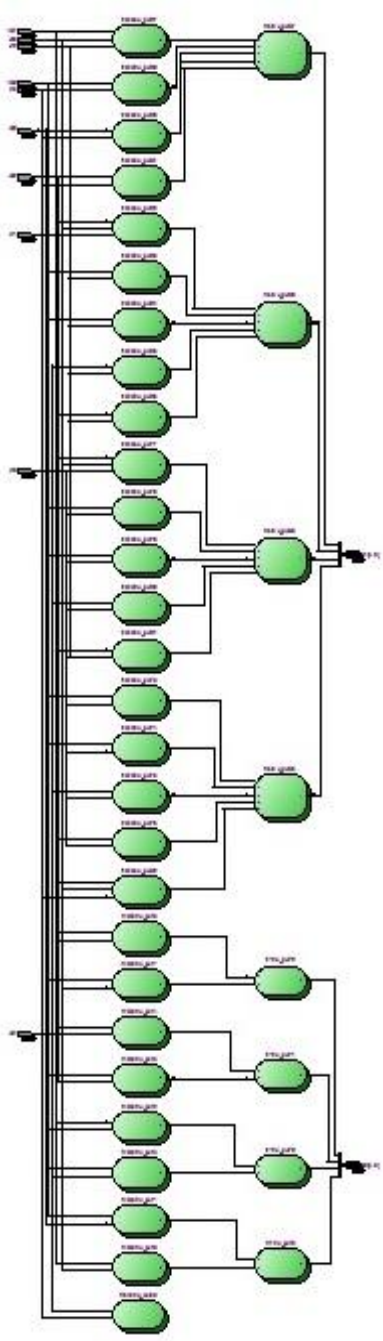

Figure 4: Quad Rail Increment Circuit with Reduced Minterm

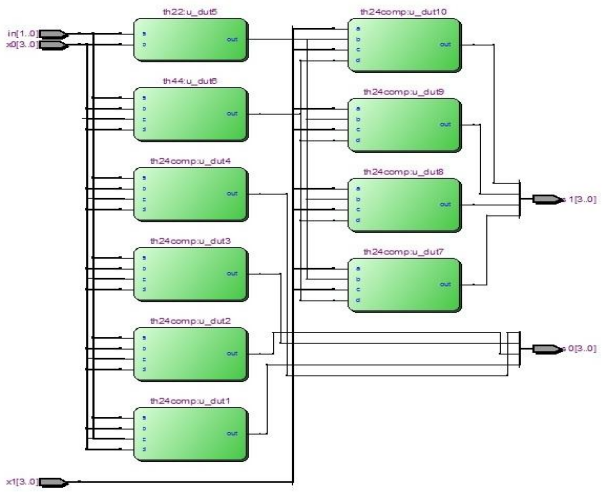

Figure 5: Quad-rail increment circuit employing complex gates

D.4x4 Bus Arbiter

As the number of bus masters increases in a single chip, we goes for an arbiter. Bus arbiter is a device which is used in multi-master bus system to decide which bus master will be allowed to control the bus for each bus cycle. Or in other words a bus arbiter conflicts when multiple bus masters request a bus in the same cycle. A 4x4 bus arbiter arbitrates between 4 asynchronous inputs. The logic diagram of bus arbiter includes counter as well as priority encoder. Here we have used dual rail optimization to design a 4x4 bus arbiter [8] entirely in terms of NCL and used the proposed grouping algorithm in [1] for grouping and mapping the output expressions to the gates from NCL library.

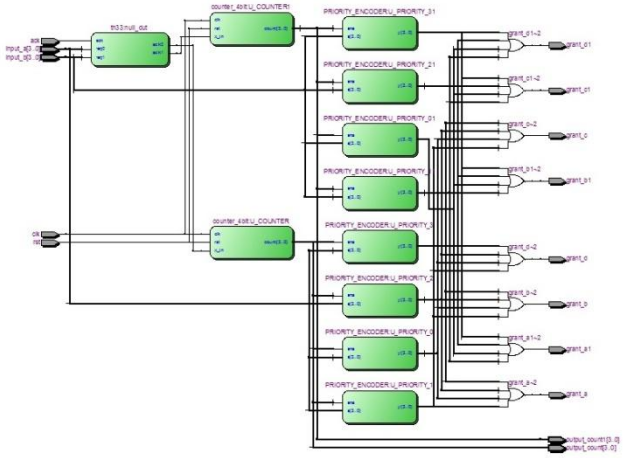

Figure 6: 4x4 Bus arbiter

Simulation and Results

The outputs of 1-bit Full Adder and 4-bit register designed entirely in NCL, by using dual rail optimization method is shown, also the performance comparison of 4-bit up counter using 'increment circuit using quad rail reduced minterm expression' and 'quad rail increment circuit using complex gates' is given. Finally an arbiter circuit has been designed entirely in terms of NULL Convention Logic and its performance is then compared with its asynchronous counterpart, and its comparison is given.

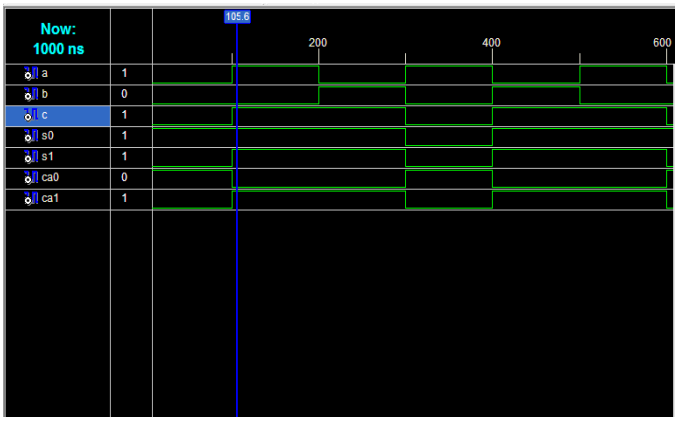

Figure 8: Full Adder output

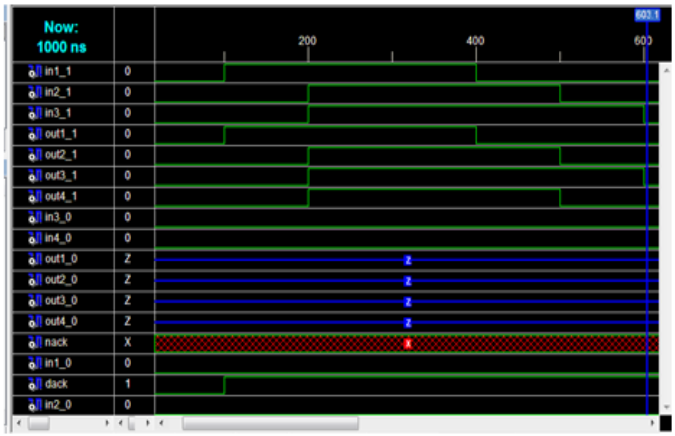

Figure 9: Register Output

Table 3: Performance Comparison of Arbiter

Parameter	**NCL Arbiter**	**Asynchronous Arbiter**
Power dissipation(mW)	68.43	69.88
Delay(nS)	6.236	15.46
Total logic elements	21	42

Table 4: Performance Comparison of Up-Counters

Parameter	Quad rail Increment circuit with reduced minterm	Quad rail increment circuit employing complex gates
Total logic elements	11	10
Delay(nS)	6.161	6.051
Power dissipation (mW)	68.77	68.77

Conclusion

In this paper basic asynchronous circuits are being designed using the various techniques in NCL with incorporating the proposed grouping algorithm given in [1]. Which are dual rail optimization for full adder and asynchronous registers and quad rail optimization for up-counter. The up-counter is designed by using two different techniques such as 'Quad rail increment circuit with reduced minterm' and 'quad rail increment circuit employing complex gates' and obtaining the result such that the former exhibits better performance in terms of area and delay. Then a 4x4 bus arbiter is designed entirely in terms of NCL and its performance is compared with simple asynchronous arbiter concluding that the former is better than the later.

References

[1] Farhad A. Parsan, Waleed K. Al-Assadi, Scott C. Smith,"Gate Mapping Automation for Asynchronous NULL Convention Logic Circuits" in IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 22, No. 1, January 2014

[2] K. M. Fant and S. A. Brandt, "NULL convention logic: A complete and consistent logic for asynchronous digital circuit synthesis," in Proc. Int. Conf. Appl. Specific Syst., Archit. Process., Aug. 1996, pp. 261–273.

[3] S. Hauck, "Asynchronous design methodologies : An overview," Proceedings of the IEEE, vol. 83, pp. 69–93, January 1995.

[4] P. Patra and D. Fussell, "Building-blocks for designing DI circuits," Department of Computer Science, University of Texas at Austin, Tech. Rep., November 1993.

[5] K. V. Berkel, "Beware the isochronic fork," Integr. VLSI J., vol. 13, no. 2, pp. 103–128, Jun. 1992.

[6] G. E. Sobelman and K. Fant, "CMOS circuit design of threshold gates with hysteresis," in Proc. IEEE Int. Symp. Circuits Syst., vol. 2. Jun. 1998, pp. 61–64.

[7] Scott Christopher Smith, "Gate and throughput optimizations for NULL convention self-timed digital circuits", Spring term 2001.

[8] Integrated circuits , data sheet,74F786,4-bit asynchronous bus arbiter, Philips semiconductors, 1991 Feb 14.

[9] Xilinx web page. www.xilinx.com