01 Aug 2007

Clock-Free Nanowire Crossbar Architecture Based on Null Convention Logic (NCL)

Ravi Bonam

Shikha Chaudhary

Yadunandana Yellambalase

Minsu Choi
Missouri University of Science and Technology, choim@mst.edu

Recommended Citation

R. Bonam et al., "Clock-Free Nanowire Crossbar Architecture Based on Null Convention Logic (NCL)," *Proceedings of the 7th IEEE International Conference on Nanotechnology (2007, Hong Kong, China)*, pp. 85-89, Institute of Electrical and Electronics Engineers (IEEE), Aug 2007.
The definitive version is available at https://doi.org/10.1109/NANO.2007.4601146

Proceedings of the 7th IEEE
International Conference on Nanotechnology
August 2 - 5, 2007, Hong Kong

Clock-Free Nanowire Crossbar Architecture based on Null Convention Logic (NCL)

Ravi Bonam, Shikha Chaudhary, Yadunandana Yellambalase and Minsu Choi
Dept of ECE, University of Missouri-Rolla, MO, USA

Abstract— **There have been numerous nanowire crossbar architectures proposed till date, although all of them are envisioned to be synchronous (i.e., clocked). The clock is an important part in a circuit and it needs to be connected to all the components to synchronize their operation. Considering non-deterministic nature of nanoscale integration, realizing them on a nano wire crossbar system would be quite cumbersome. Unlike the conventional clocked counterparts, a new clock-free crossbar architecture is proposed to resolve the issues with clocked counterparts in this paper, where the use of clock is eliminated from the architecture. This has been done by implementing delay-insensitive logic encoding technique called Null Convention Logic (NCL). A delay-insensitive full adder has been implemented on the proposed architecture to demonstrate the feasibility in this paper.**

Index Terms— **Nanowire crossbar, Asynchronous computing, Null convention logic (NCL), Manufacturability, Robustness, Scalability, Defect & fault-tolerance.**

I. Introduction

The end of photolithography as the driver for Moore's Law is predicted within seven to twelve years and nanotechnologies are emerging that are expected to continue the technological revolution. Recently, numerous nanoscale logic devices have been proposed based on nanoscale components such as CNTs and SiNWs; computing architectures are also being proposed using them as primitive building blocks.

One of the most promising nanotechnologies is the crossbar-based architecture, a two-dimensional array (i.e., nanoarray) formed by the intersection of two orthogonal sets of parallel and uniformly-spaced nanometer-sized wires, such as carbon nanotubes (CNTs) and silicon nanowires (SiNWs). Experiments have shown that such wires can be aligned to construct an array with nanometer-scale spacing using a form of directed self-assembly and the formed crosspoints of nanoscale wires can be used as programmable diodes, memory cells or FETs (Field-Effect Transistors); therefore, nanoscale logic devices can be realized.

Nanowire crossbars offers both an opportunity and a challenge. The opportunity is to achieve ultra-high density which has never been achieved by photolithography. The challenge is to make them *simple enough to be manufactured and reliable enough to be used in everyday computing applications*, since high-density systems consisting of nanometer-scale elements assembled in a bottom-up manner are likely to have many imperfections (much higher raw fabrication defect densities, as high as 10%, are expected [1, 2]) and parametric variations. A computing system designed on conventional design basis and top-down lithographic manufacturing would not be practical. Ultra-high density fabrication could potentially be very inexpensive if researchers can actualize a chemical self-assembly, but such a circuit would require laborious testing, repair and reconfiguration processes, implying significant overhead costs. Also, all reconfigurable computing architectures based on nanowire crossbars are commonly envisioned to be used for synchronous circuits and systems. Thus, a clock distribution network should be fabricated along with nanowire crossbars and precise timing control should be practiced to avoid all timing-related faults induced by physical design parameter variations caused by nanoscale non-deterministic assembly.

In order to be a viable nanotechnology, the nanowire crossbar-based systems should be:

1) Structurally simple and scalable enough to be fabricated by bottom-up manufacturing technique,
2) Robust enough to tolerate extreme parametric variations,
3) Defect and fault-tolerant enough to overcome the extreme defect densities, aging factors and transient faults, and
4) Able to support at-speed verification and reconfiguration.

Unlike the conventional clocked counterparts, the proposed research is to propose a new asynchronous architecture for carbon nanotube (CNT) and silicon nanowire (SiNW)-based reconfigurable nanocomputing systems and to address aforementioned issues in doing so.

The proposed asynchronous nano-architecture is based on a delay-insensitive data encoding and self-timed logic encoding scheme - therefore, it is totally clock-free. Thus, no clock distribution network is needed and all failure modes related to the timing will be also eliminated. Potential benefits from the proposed asynchronous architecture includes enhanced manufacturability, scalability, robustness and defect and fault-tolerance.

II. Preliminaries and Review

A. Null Convention Logic for Asynchronous Logic

Most of the traditional Boolean circuits that we have been using are clock driven. The clock is one of the most important parts of the circuit and is also a parameter determining the speed and

1-4244-0608-0/07/$20.00 © 2007 IEEE.

performance of the circuit. All the devices in a circuit have to be connected to the clock; hence the clock network is quite cumbersome. The traditional Boolean circuits do not check for input completion at the time of evaluating an expression i.e. whether all the inputs have arrived to start computation of the expression. Hence the Traditional Boolean circuits are symbolically incomplete in terms of evaluating expressions as they are dependent on the clock.

Null Conventional Logic integrates data and control into a single signal thus yielding inherently clock less, delay insensitive circuits and systems [5]. This technology uses two states, DATA and NULL, which are used for synchronizing and I/O control. DATA wave front contains the data that has to be processed by the combinational circuit. The Null wave front is a non-data value used to reset the logic gates in the circuit and is also used as a delimiter between two DATA wave fronts [5]. Circuits communicate with each other using local hand shakes which provide synchronization. The concept of global clock is removed and this in turn removes the clock network that has to be circulated inside the circuit. The removal of clock reduces the power consumption and the circuit becomes data driven (i.e. data is processed as soon as it is available). In the DATA combinational evaluation period the combinational circuitry processes the data passed on by the register and the results are stored in the successive register. The successive register generates the Request for NULL signal in the DATA completion Acknowledgement period and propagates the signal to the previous register. The previous register will then transfer a NULL to the combinational circuitry which is evaluated during the NULL combinational evaluation period. The evaluated result is passed to the successive register which then generates a Request for DATA signal.

If the output of a particular gate is NULL, it does not change until and unless all the inputs to the gate are DATA. When all the inputs receive DATA then the output changes to data and remains asserted as long as all the inputs do not change to NULL. This attribute of the threshold gates helps in achieving input completeness feature enabling the circuits to function without the clock [7]. To achieve this property the inputs to the gates are to be encoded using an encoding scheme. In a dual rail encoding scheme, each bit is represented using two rails.

According to the representation in the Table 1 the combination of rails (rail1,rails0) represents a single Boolean value. The value ”00” is regarded as NULL state which resets the circuit and does not represent any Boolean value. The value ”11” is an undefined expression in the dual rail encoding scheme.

NCL uses symbolic completeness [11] of expression to achieve self-timed behavior. A symbolically complete expression is defined as an expression that only depends on the relationships of the symbols present in the expression without a reference to the time of evaluation.

This is achieved by keeping the following conditions in mind [11]:

1) The input-completeness criterion, which NCL circuits must maintain in order to be self-timed, requires that the outputs of a circuit may not transition from NULL to DATA until all inputs have transitioned from NULL to DATA or vice versa.
2) In circuits with multiple outputs, outputs that are dependent on arrived inputs can make transition, but all outputs can change only when all inputs arrive which eliminates the possibility of a data cycle and null cycle overlapping.
3) No orphans may propagate through a gate. An orphan is defined as a wire that transitions during the current DATA wavefront, but is not used in the determination of the output. Orphans are caused by wire forks and can be neglected through the isochronic fork assumption, as long as they are not allowed to cross a gate boundary. This observability condition ensures that every gate transition is observable at the output.

The primary advantages of the use of NCL for the proposed clock-free nano-architecture are as follows:

1) Circuits are less complex and are large circuits can be designed in a bottom-up manner and integrated directly without any trouble of synchronizing each module [5].
2) In clock-driven circuits, major part of power is consumed by the clock and its network. By removing the clock from the circuit, cumulative power consumption decreases [5].
3) The use of NCL makes the circuit insensitive to delay and the circuits operate at the rate of the flow of data. The circuits can be called as delay insensitive and self timed circuits [5, 7].
4) The circuits become more reliable than the clocked circuits as the problems caused due to clock such as clock skew, race conditions etc. are eliminated [5].

There are 27 threshold gate macros that are implemented in NCL. The significance of these 27 NCL gates is that any possible expression involving two or three or four variables can be implemented using these functions. Inversion can be implemented by interchanging the rail1 and rail0 in case of the dual rail encoding scheme.

B. Asynchronous Crossbar Architecture and its Advantages

Using the normal crossbar architecture is similar to the conventional Boolean circuits i.e. clock has to be circulated throughout the circuit for synchronizing various blocks. The normal crossbar circuit cannot decide when to receive or release data, therefore a clock must be added to control the flow of input and output. In contrast, the asynchronous crossbar architecture would be data driven; Instructions are acted upon the moment they are available and output is available the moment it is completed. This architecture employs discrete threshold gates

[5] that recognize only certain simultaneous combinations of values. Each of the gates act as ”synchronization nodes”, making the circuit as a whole symbolically complete. The data state follows the Null state and is processed by the gates and output is passed on to a register. The register contains completion circuitry that enables synchronization and checks the state of the output and generates an appropriate signal indicating the previous register to send the complementary state i.e. if the circuit is processing a Null state then the register on arrival of the output will send a request for data signal requesting for data to the previous register. The primary Advantages of the Asynchronous architecture would be, a. Manufacturability: Asynchronous architecture significantly increases the manufacturability of the nanowire crossbar systems in large scale manufacture. Manufacturing of these kinds of circuits would be easier compared to their clocked counterparts. Clocked synchronous architectures are difficult to map on crossbars architectures as they require complex placement and routing algorithms. In case of Asynchronous crossbar architecture, discrete blocks of crossbars can be used to map gates onto them and there is no need of a global synchronous signal to coordinate all the blocks. All clock-related hardware components can be removed from the overall hardware design. Circuits would be less complex and easier to design. b. Scalability: The overall circuit is self time i.e. timing information is integrated with data in the encoding. As the timing of each circuit is handled locally, Scalability of these circuits would be higher. The timing complexity remains the same even though the size of the circuit gets larger i.e. the time taken for any particular computation will not change on the basis of the size of the circuit. c. Robustness: Due to non-determinism of the directed self-assembly paradigm, nanowire crossbar circuits are anticipated to exhibit large variations in physical parameters. Since any physical variation in an electrical parameter may have its own negative effect on the timing behavior of the circuit, being able to design delay-insensitive circuits (i.e., correct operation of the circuit is independent of the timing) is a significant capability and it would greatly increase the robustness of the circuit to design parameter variations. As explained in Null Conventional logic for asynchronous logic subsection, there is no delay in processing data due to clock cycles as in clocked synchronous circuits, instead data would be processed as and when it is available. d. Defect and Fault Tolerance: As NCL circuits have a definite flow patter i.e. DATA or NULL and vice versa the output can be checked if it is a data or null. In addition to the complete removal of all timing-related failure modes, testing complexity is reduced in that stuck-at-1 faults simply halt the circuit, since the NCL circuit cannot make a transition from DATA to NULL. Also, in case of dual-rail encoding, 11 is considered as an invalid code. So, any permanent or transient fault that results in this invalid codeword can be eventually detected. Only stuck-at-0 faults and some other transient faults need to be exercised with applied patterns. Design time and risk as well as circuit testing requirements are expected to be decreased because of the elimination of the complexity of the clock with its critical timing issues.

In this paper we are going to implement Null Conventional Logic on Nano wire crossbar architecture to realize ”Asynchronous Crossbar Architecture”. We also show the implementation of a full adder using the new crossbar architecture and discuss feasibility of a Multi-bit adder.

III. Proposed Architecture

A. Programmable Gate Macro Block (PGMB)

The basic unit of the proposed architecture is a programmable gate macro block (PGMB). Each block is made of an AND plane and a OR plane formed by the diode crossbars. Vertical nano wires with pull up resistors form product terms and horizontal wire with pull down resistor add them using OR logic. It also has a feedback loop which drives the output back to an input wire. The maximum number of inputs to any threshold gate is 4 and along with this it needs a feedback to implement any of the 27 threshold gates [7]. For example, Figure 1 shows the implementation of TH23 gate in the programmable gate macro block. The output of the TH23 gate is given by the logic Z = AB+BC+CA + (A+B+C)Z*. Z* is the previous output of the TH23 gate which is fed back to the input nanowire.

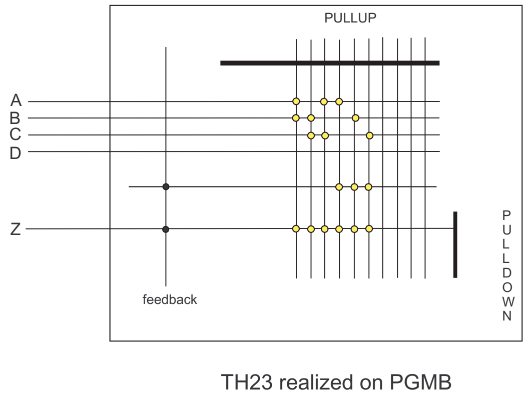

Fig. 1. TH23 gate realized on PGMB.

B. Physical Structure

The new architecture consists of array of PGMBs which are interconnected in the form of 2D grid structure. These blocks are surrounded by nano wires which are used to route the signals inside the grid structure. The PGMB’s input and output nano wires cross these routing wires forming programmable cross points. By programming these cross points we can route the signals to any of the programmable gate macro blocks. The input stage consists of programmable resistor cross points formed by the micro wires and nano wires. By programming relevant cross points we can route the signals to the required PGMB. Each block can in turn be programmed to implement any of the threshold gates [7]. These blocks can tap the input signals by programming corresponding cross point formed by the nano wire column carrying input signal and nano wire row which is an input to the macro block. The output of the implemented threshold gate [7] can be routed to the other gates in the similar fashion. Thus the number of columns of nano wires between programmable macro blocks determines the amount of cross points available for routing signals. This number has to be sufficient to route all the required inputs and outputs to the macro blocks. The number of rows and columns of PGMBs in the grid are limited by the amount of signal degradation caused by the propagation. Before the complete degradation of the signal, a buffering stage can be implemented to restore the strength.

IV. Implementation of Asynchronous Single-Bit Full Adder

A full adder can be implemented using threshold gates as shown in Figure 2 [7]. Now, let us implement 1 bit full adder using the proposed architecture. The 1 bit full adder can be implemented using two th23 gates and two th34w2 gates as shown in the Figure 3. This requires 3 input bits a, b for addition and c for carry encoded in dual rail logic. These bits are represented by a0, a1, b0, b1, c0 and c1 respectively. By programming required cross points at the input cross bar these signals routed to the programmable gates.

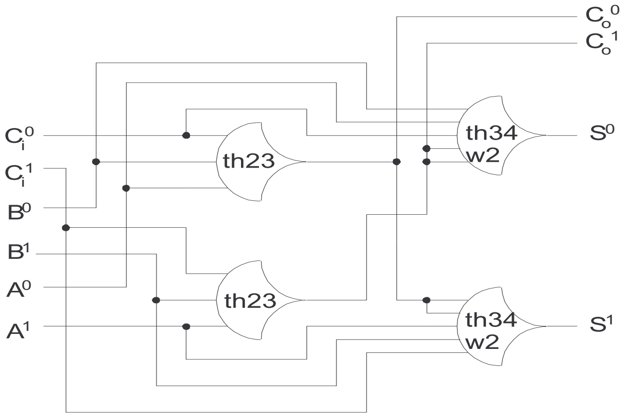

Fig. 2. NCL 1-bit full adder using threshold gates.

The complete implementation of the 1 bit full adder is shown in the Figure 3. The blocks present in row 1 and columns 1, 2 are programmed as th23 gates and blocks in row 2 and columns 1, 2 are programmed as th34w2 gates. The th23 gates require 3 inputs and therefore 1 input row is unused where as in th34w2 all the 4 input rows are used. The realized threshold gates on PGMB are shown in the Figures 1 and 4. Next we have to route required signal into the corresponding input rows. Outputs from the threshold gates should also be routed to the input of other gates or to the output block. This can be achieved by programming routing cross points and using free nano wires.

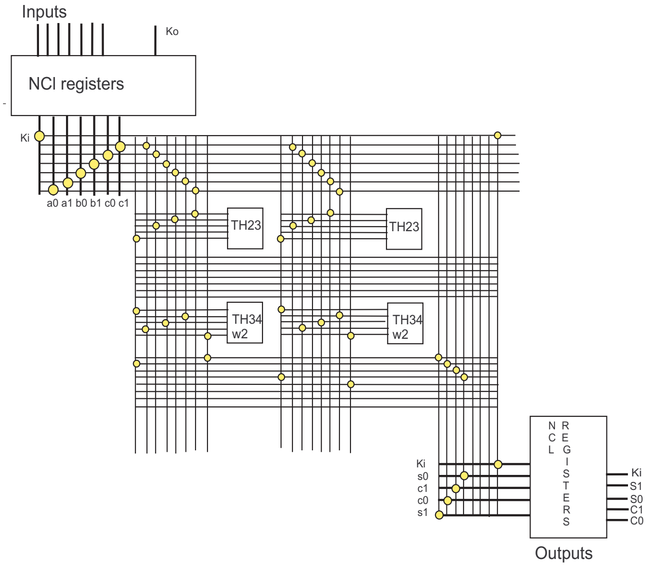

Fig. 3. 1-bit adder implemented on the proposed asynchronous architecture.

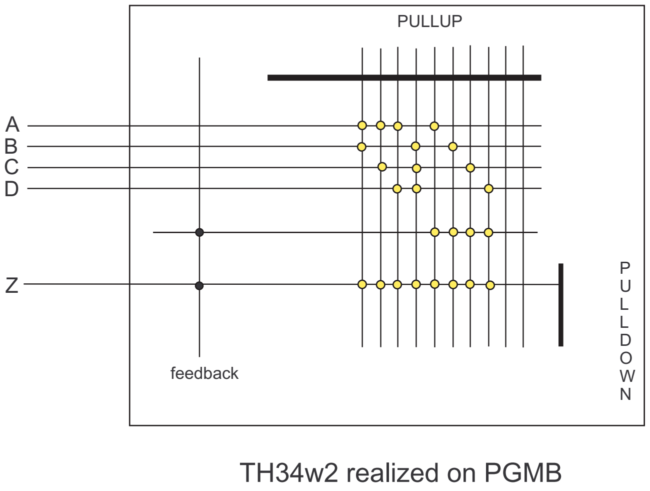

Fig. 4. TH34w2 gate realized on PGMB.

The NCL register stage consists of two TH22 and a TH12 gates that are used to generate a handshaking signal that helps in synchronizing the circuit. There are two kinds of signals, request for data and request for null, generated by the registers that are passed on to the previous register. The Ki (input from successive stage) and Ko (output to previous stage) are the handshaking signals and Do, D1 are input data rails and Q0, Q1 are the output rails. The single bit register stage is shown in the Figure 5.

V. Conclusion

In this paper, we have proposed a new clock-free nanowire crossbar architecture based on delay-insensitive logic known as Null Convention Logic. The complex clock distribution

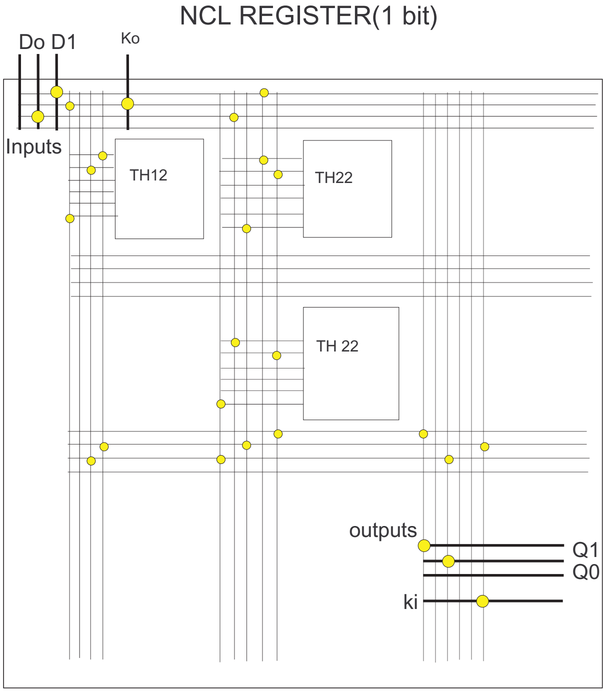

Fig. 5. 1-bit NCL register realized on the proposed architecture.

network can be removed from the hardware and many clocking-related failure modes can be intrinsically eliminated by the proposed clock-free architecture. To demonstrate the feasibility, delay-insensitive full adder design has been implemented on the proposed clock-free architecture. Our future direction is to develop automated design optimization tools, testing schemes and defect-tolerant logic mapping techniques for the proposed architecture.

References

[1] J. Huang, M. B. Tahoori and F. Lombardi, ”On the defect tolerance of nano-scale two-dimensional crossbars”, IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, pp. 96-104, Oct 2004.

[2] M. Jacome, C. He, G. de Veciana, and S. Bijansky, ”Defect tolerant probabilistic design paradigm for nanotechnologies”, IEEE/ACM Design Automation Conference (DAC), pp. 1-6, 2004.

[3] Jiangtao Hu et. al., ”Chemistry and Physics in One Dimension : Synthesis and properties of Nanotubes and Nanowires,” Acc. Chem. Res. , Vol. 32, pp. 435-445, 1999.

[4] Matthew M. Ziegler and Mircea R.Stan, ”Design and analysis of Crossbar Circuits for Molecular Nanoelectronics,” IEEE Nanotechnology Conference, pp. 323-327, 2002.

[5] Karl M. Fant, Scott A. Brandt, ”NULL Convention Logic : A Complete and Consistent Logic for Asynchronous Digital Circuit Synthesis,” IEEE International Conference on Application-Specific Systems, Architectures and Processors, pp. 261-273, 1996.

[6] R. Smith and M. Ligthart, ”High-Level Design for Asynchronous Logic,” Design Automation Conference, pp. 431-436, 2001.

[7] S. C. Smith, R. F. DeMara, J. S. Yuan, D. Ferguson, and D. Lamb, ”Optimization of NULL Convention Self-Timed Circuits,” Integration, The VLSI Journal, Vol. 37, No. 3, pp. 135-165, 2004.

[8] D. Whang, S. Jin and C. M. Lieber, ”Large-Scale Hierarchical Organization of Nanowires for Functional Nanosystems,” Japanese Journal of Applied Physics, Vol. 43, No. 7B, 2004.

[9] Y. Cui and C. M. Lieber, ”Functional nanoscale electronic devices assembled using silicon nanowire building blocks,” Science, Vol. 291, pp. 851-853, 2001.

[10] Nicolas A. Melosh, Akram Boukai, Frederic Diana, Brian Geradot, Antonio Badolato, Pierre M. Petroff, James R. Health, ”Ultrahigh-Density Nanowire Lattices and Circuits,” Science, Vol. 300, pp. 112-115, 2003.

[11] S. Smith, R. DeMara, J. Yuan, M. Hagedorn and D. Ferguson, ”Delay-Insensitive gate-level pipelining,” Integration, the VLSI journal, Vol. 30, pp. 103-131, 2000.