

Designing NULL Convention Combinational Circuits to Fully Utilize Gate-Level Pipelining for Maximum Throughput

Scott C. Smith

University of Missouri – Rolla, Department of Electrical and Computer Engineering
133 Emerson Electric Co. Hall, 1870 Miner Circle, Rolla, MO 65409
Phone: (573) 341-4232, Fax: (573) 341-4532, E-mail: smithsco@umr.edu

Keywords: Asynchronous logic design, delay-insensitive circuits, speedup, Threshold Combinational Reduction (TCR), NULL Convention Logic (NCL)

Abstract

Since the NULL Convention Logic (NCL) paradigm is delay-insensitive, NCL combinational circuits cannot be partitioned indiscriminately when pipelining, as can clocked circuits. Instead, these circuits must be partitioned into stages, such that each stage is input-complete with respect to all of its inputs and delay-insensitivity is maintained. Therefore the selected architecture for an NCL circuit may vary depending on whether or not the given circuit is to be pipelined. For example, the combinational circuit that has the shortest critical path may not necessarily result in optimal throughput when pipelined.

This paper presents a method called Threshold Combinational Reduction for Pipelining (TCRP), to design NCL combinational circuits that will maximize throughput when pipelined. This method is demonstrated on the design of a 4-bit \times 4-bit quad-rail multiplier and a dual-rail Booth2 multiplier, showing that both designs have an additional increased throughput after pipelining, when their combinational circuitry is designed using TCRP rather than the previous Threshold Combinational Reduction (TCR) method [1].

1. NCL Overview

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 [2]. Like other delay-insensitive logic methods, the NCL paradigm assumes that forks in wires are isochronic [3]. Various aspects of the paradigm, including the NULL (or spacer) logic state from which NCL derives its name, have origins in Muller's work on speed-independent circuits in the 1950s and 1960s [4].

1.1 Delay-Insensitivity

NCL uses symbolic completeness of expression to achieve delay-insensitive behavior. A symbolically complete expression depends only on the relationships of the symbols present in the expression without reference to their time of evaluation [5]. In particular, dual- and quad-rail signals or other mutually exclusive assertion groups (MEAGs) can incorporate data and control information into one mixed-signal path to eliminate time reference. For NCL and other circuits to be purely delay-insensitive, assuming isochronic wire forks [3], they must meet the input-completeness and observability criteria [6].

Completeness of input requires that all the outputs of a combinational circuit may not transition from NULL to DATA until all inputs have transitioned from NULL to DATA, and that all the outputs of a combinational circuit may not transition from DATA to NULL until all inputs have transitioned from DATA to NULL. In circuits with multiple outputs, it is acceptable, according to Seitz's weak conditions [2], for some of the outputs to transition without having a complete input set present, as long as all outputs cannot transition before all inputs arrive. Observability requires that no *orphans* may propagate through a gate [7]. 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 [3], as long as they are not allowed to cross a gate boundary. This *observability* condition, also referred to as *indicatability* or *stability*, ensures that every gate transition is observable at the output; which means that every gate that transitions is necessary to transition at least one of the outputs. Furthermore, when circuits use the bit-wise completion strategy with selective input-incomplete components, they must also adhere to the completion-completeness criterion

[6], which requires that completion signals only be generated such that no two adjacent DATA wavefronts can interact within any combinational component.

Most multi-rail delay-insensitive systems [2, 5, 8], including NCL, have at least two register stages, one at both the input and the output. Two adjacent register stages interact through request and acknowledge lines, K_i and K_o , to prevent the current DATA wavefront from overwriting the previous DATA wavefront by ensuring that the two are always separated by a NULL wavefront.

1.2 Logic Gates

NCL differs from other delay-insensitive paradigms [2, 8], which use only one type of state-holding gate, the C-element [4]. A C-element behaves as follows: when all inputs assume the same value, the output assumes this value; otherwise, the output does not change. On the other hand, all NCL gates are state-holding. NCL uses threshold gates as its basic logic elements [9]. The primary type of threshold gate is the TH_{mn} gate ($1 \leq m \leq n$). TH_{mn} gates have n inputs. At least m of the n inputs must be asserted before the output becomes asserted. Because NCL threshold gates are designed with *hysteresis*, all asserted inputs must be deasserted before the output is deasserted. Hysteresis ensures a complete transition of inputs back to NULL before asserting the output associated with the next wavefront of input data. NCL threshold gates may also include a *reset* input to initialize the output. Circuit diagrams designate resettable gates by either a D or an N appearing inside the gate, along with the gate's threshold. D denotes the gate as being reset to logic 1; N , to logic 0.

2. Previous Work

Previous work includes a method called Threshold Combinational Reduction (TCR) [1] for designing arbitrary NCL combinational circuits, and a method called Gate-Level Pipelining (GLP) [10] for optimally pipelining an NCL combinational circuit. However, the throughput achieved when pipelining an NCL combinational circuit is highly dependent on the design of the circuit's individual components (i.e. the full-adder in the case of a multiplier), since the components cannot be arbitrarily divided without compromising delay-insensitivity. The combinational circuit that has the shortest critical path may not necessarily provide for optimal throughput when pipelined. This may occur when larger components, with more gate delays than the rest of the components, are used to reduce the critical path of the circuit as a whole. While this increases the throughput for the non-pipelined circuit, the larger components will yield a larger combinational stage delay when pipelined, thus limiting the throughput increase for the entire circuit.

TCR [1] produces combinational circuits with minimal critical path delay, while GLP [10] optimally partitions a TCR combinational circuit in order to maximize throughput. However, neither of these papers address the issue of designing a combinational circuit, which is specifically intended to be pipelined. This design may be different than the minimal critical path design produced by TCR, since the objective is to minimize the maximum component delay rather than to minimize the critical path delay.

3. Combinational Design for Pipelining

In this paper a method called Threshold Combinational Reduction for Pipelining (TCRP) is developed, to be used specifically for producing combinational circuits that are to be subsequently pipelined in order to maximize their increased throughput. TCRP follows the same steps as TCR [1]; however the main objective is slightly different. The objective when a combinational circuit is to be pipelined is to minimize the maximum component delay, rather than to minimize the critical path delay. Therefore the main difference between TCR and TCRP is how to select the component size. In TCR, larger components are normally better, since they combine more logic into a single component, which usually reduces the critical path delay for the circuit as a whole, but may yield components with more than two gate delays. TCRP on the other hand tries to minimize component delay, at the expense of increased critical path delay. TCRP partitions circuits into logic blocks that can be implemented in two or less gate delays, whenever possible, since most NCL circuits can be partitioned this way without substantially increasing the critical path delay for the non-pipelined circuit.

4. Application to the Quad-Rail Multiplier

One example of using TCRP to design the combinational circuitry, instead of TCR, to maximize throughput for the pipelined design, is the 4-bit \times 4-bit quad-rail multiplier [11]. This circuit multiplies the multiplicand, MD , with the multiplier, MR , both of which consist of two quad-rail signals, to produce the parallel product, P , which consists of four quad-rail signals. Using TCR to design the circuit, results in the partitioning shown in Figure 1, where Partition 1 produces the partial products, while Partitions 2, 3, and 4 sum the partial products in carry-save fashion. This design yields a critical path of 8 gate delays and is optimally pipelined with GLP by inserting asynchronous registers between Partitions 1 and 2 and between Partitions 2 and 3, and utilizing bit-wise completion, which results in 3 stages. The bit-wise completion strategy only sends the

completion signal for bit b in register _{i} back to those bits in register _{$i-1$} that took part in the generation of bit b [6].

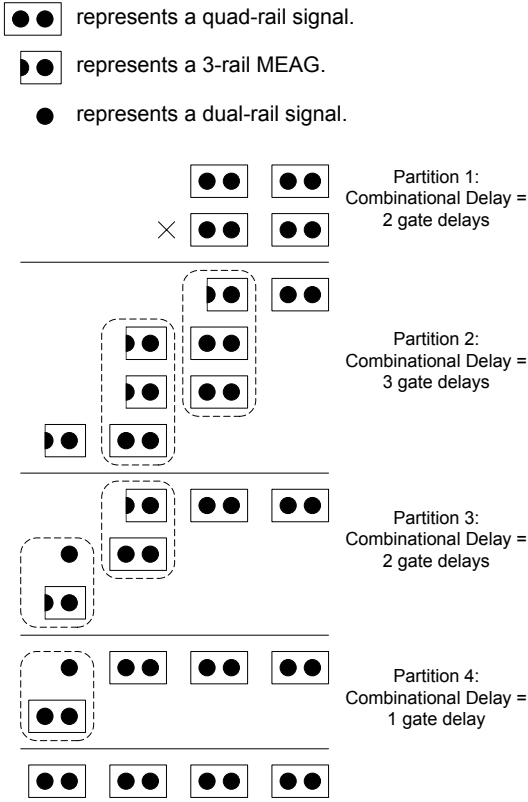


Figure 1. Partitioning of the TCR quad-rail multiplier.

The required components are characterized in Table 1, resulting in Stage 1 having a worse-case combinational delay of 2 gates and a completion delay of 1 gate, and Stages 2 and 3 each having a worse-case combinational delay of 3 gates and a completion delay of 1 gate. In this table Q3 represents a quad-rail signal of range 0-3, Q2 represents a 3-rail MEAG ranging from 0-2, and D represents a dual-rail signal of range 0-1. The outputs of the multiplier component, Q33mul, consist of a higher

order partial product, PPH , as well as a lower order partial product, PPL . The outputs of the various adders consist of a Sum signal, and $Carry$ signal when necessary. The multiplier component was ensured to be input-complete by adding additional terms to the equation for PPL^0 , such that both inputs, A and B , are required even when either A or B is logic 0. The various adder circuits are all inherently input-complete. Stage 2 cannot be further partitioned without compromising delay-insensitivity.

The maximum cycle time for a stage is estimated as two times the maximum number of gate delays in the stage's combinational logic plus two times the number of gate delays in the stage's completion logic, to account for both the DATA and NULL wavefronts; thus the cycle time for Stage 1 is 6 gate delays, the cycle time for Stage 2 is 8 gate delays, and the cycle time for Stage 3 is 8 gate delays. The maximum cycle time for the entire pipeline is then estimated as the global maximum of the maximum cycle times for each of the stages; thus it is 8 gate delays for this design. Simulating the TCR pipelined design using a static CMOS implementation and a $0.5\mu\text{m}$ process operating at 3.3V, resulted in an average cycle time, T_{DD} , of 6.22 ns. T_{DD} is calculated as the arithmetic mean of the cycle times corresponding to all 256 possible pairs of input operands.

Applying the TCRP algorithm to design the 4×4 quad-rail multiplier results in the repartitioning of the carry-save adder tree, as shown in Figure 2, such that the worst-case adder delay is only 2 gates, as shown in Table 2. This design yields a critical path of 9 gate delays and is optimally pipelined with GLP by inserting asynchronous registers between each partition, and utilizing bit-wise completion, which results in 5 stages. Stages 1, 2, 3, and 4 each have a worse-case combinational delay of 2 gates and a completion delay of 1 gate, while Stage 5 has a combinational delay of 1 gate and a zero completion delay. The maximum cycle time for the entire pipeline is therefore reduced to only 6 gate delays. Simulating the TCRP pipelined design resulted in $T_{DD} = 4.93$ ns, a speedup of 1.26 over the TCR pipelined design.

Table 1. TCR quad-rail multiplier component characterization.

Component Type	Input Types	Output Type		Output Gate Delays	
		Carry / PPH	Sum / PPL	Carry / PPH	Sum / PPL
Q33mul	Q3, Q3	Q2	Q3	1	2
Q332add	Q3, Q3, Q2	Q2	Q3	3	3
Q322add	Q3, Q2, Q2	D	Q3	2	3
Q32add	Q3, Q2	D	Q3	2	2
Q2Dadd	Q2, D	N/A	Q3	N/A	1
Q3Dadd	Q3, D	N/A	Q3	N/A	1

Table 2. TCRP quad-rail multiplier component characterization.

Component Type	Input Types	Output Type		Output Gate Delays	
		Carry / PPH	Sum / PPL	Carry / PPH	Sum / PPL
Q33mul	Q3, Q3	Q2	Q3	1	2
Q33add	Q3, Q3	D	Q3	2	2
Q32add	Q3, Q2	D	Q3	2	2
Q3DDadd	Q3, D, D	D	Q3	2	2
Q2DDadd	Q2, D, D	N/A	Q3	N/A	2
Q3Dadd	Q3, D	N/A	Q3	N/A	1

- represents a quad-rail signal.
- represents a 3-rail MEAG.
- represents a dual-rail signal.

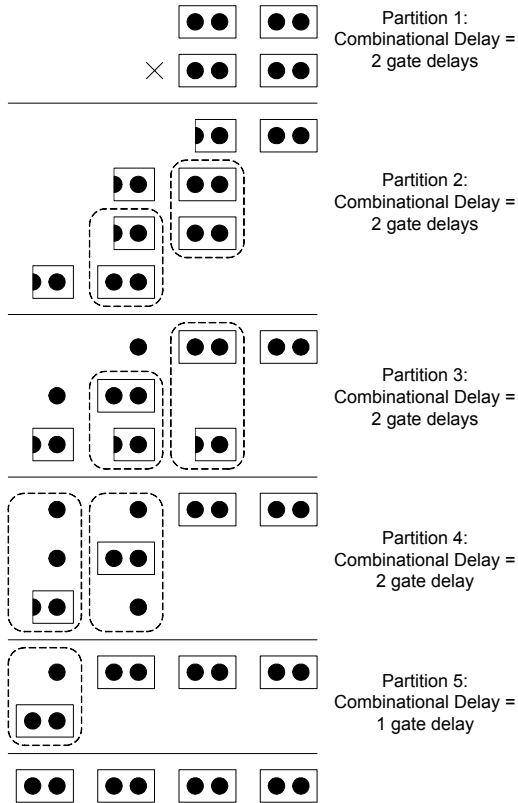


Figure 2. Partitioning of the TCRP quad-rail multiplier.

5. Application to the Booth2 Multiplier

Another example where TCRP maximizes throughput for the pipelined design when TCR does not is for a dual-rail NCL Booth2 multiplier. Using TCR to design the standard partial product, PP , generation circuitry results

in a component with 3 gate delays that cannot be further partitioned without compromising delay-insensitivity, as shown in Figure 3. MR_2 , MR_1 , and MR_0 represent the group of three multiplier bits, while MD_i and MD_{i-1} represent the multiplicand bits necessary to generate PP_i . Since the rest of the multiplier consists of full-adders and half-adders to sum the partial products, which have 2 gate delays and 1 gate delay, respectively, the 3 gate delays in the partial product generation stage will limit the throughput increase for the entire circuit when pipelined.

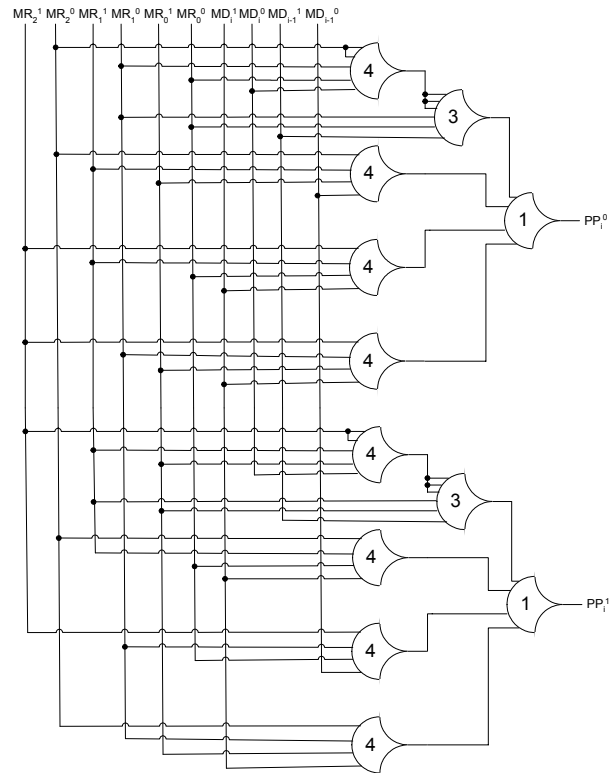


Figure 3. TCR partial product generation component for the Booth2 multiplier.

On the other hand, using TCRP to design the Booth2 multiplier results in a 2-stage partial product generation

circuit. The first stage recodes each group of three multiplier bits into a sign bit, S , to determine if PP_i is positive or negative, and two select bits, Sel_i and Sel_0 . When $Sel = 00$, $PP_i = +0$, regardless of S ; when $Sel = 01$, $PP_i = \pm MD$, depending on S ; when $Sel = 10$, $PP_i = \pm 2MD$, depending on S ; and when $Sel = 11$, $PP_i = -0$, regardless of S . This stage requires 2 gate delays, as shown in Figure 4. The second stage utilizes the recoded groups of multiplier bits, along with the multiplicand, to generate each partial product bit, as shown in Figure 5. This second stage also requires 2 gate delays. An asynchronous register can then be inserted between these two stages without compromising delay-insensitivity in order to decrease the pipeline's overall worst-case combinational stage delay from 3 gate delays, as in the TCR design described above, to only 2 gate delays; thus resulting in a faster pipelined design. Note that this repartitioning of the partial product generation circuitry does increase the critical path by 1 gate delay for the non-pipelined design. Also note that both the TCR and TCRP partial product generation circuits for the Booth2 multiplier are not input-complete with respect to all of their inputs. However, this does not compromise delay-insensitivity because input-completeness can be ensured for the circuit as a whole, through the special partial product generation circuits, such as those that generate the MSB and LSB of each partial product, without increasing the partial product generation delay.

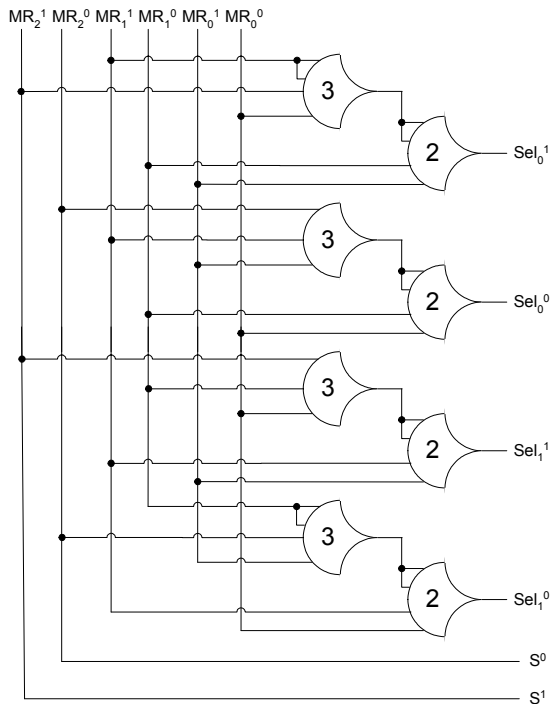


Figure 4. First stage of TCRP partial product generation component for the Booth2 multiplier.

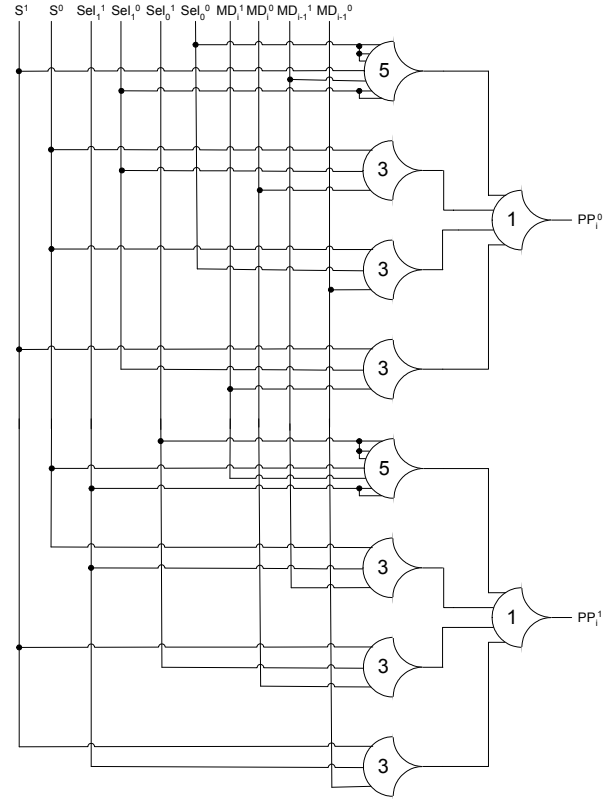


Figure 5. Second stage of TCRP partial product generation component for the Booth2 multiplier.

6. Conclusions

A method called Threshold Combinational Reduction for Pipelining (TCRP) has been developed for designing NCL combinational circuitry that will be subsequently pipelined, in order to maximize the increased throughput. TCRP follows the same circuit construction steps as TCR [1], but with a slightly different design objective: to minimize the maximum component delay, instead of minimizing the critical path delay, as does TCR.

TCRP was applied to design the combinational circuitry for a 4-bit \times 4-bit quad-rail multiplier and for a dual-rail Booth2 multiplier, both of which yielded faster pipelined designs when utilizing TCRP instead of TCR. For the 4-bit \times 4-bit quad-rail multiplier, TCRP resulted in an increase of 1 gate delay in the critical path. For larger word-width quad-rail multipliers the critical path delay would increase even more when using TCRP versus TCR; however, TCRP would always produce faster pipelined circuits. For the dual-rail Booth2 multiplier, TCRP also resulted in an increase of 1 gate delay in the critical path. For larger word-width dual-rail Booth2 multipliers the critical path would only increase by 1 gate

delay, regardless of the size of the input operands. Furthermore, TCRP would again always produce faster pipelined circuits.

References

- [1] 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*, accepted for publication, December 2003.
- [2] C. L. Seitz, "System Timing," in *Introduction to VLSI Systems*, Addison-Wesley, pp. 218-262, 1980.
- [3] C. H. (Kees) van Berkel, M. Rem, and R. Saeijs, "VLSI Programming," *1988 IEEE International Conference on Computer Design: VLSI in Computers and Processors*, pp. 152-156, 1998.
- [4] D. E. Muller, "Asynchronous Logics and Application to Information Processing," in *Switching Theory in Space Technology*, Stanford University Press, pp. 289-297, 1963.
- [5] K. M. Fant and S. A. Brandt, "NULL Convention Logic: A Complete and Consistent Logic for Asynchronous Digital Circuit Synthesis," *International Conference on Application Specific Systems, Architectures, and Processors*, pp. 261-273, 1996.
- [6] S. C. Smith, "Completion-Completeness for NULL Convention Digital Circuits Utilizing the Bit-wise Completion Strategy," *The 2003 International Conference on VLSI*, pp. 143-149, June 2003.
- [7] A. Kondratyev, L. Neukom, O. Roig, A. Taubin, and K. Fant, "Checking delay-insensitivity: 10^4 gates and beyond," *Eighth International Symposium on Asynchronous Circuits and Systems*, pp. 137-145, 2002.
- [8] J. Sparso and J. Staunstrup, "Design and Performance Analysis of Delay Insensitive Multi-Ring Structures," *Twenty-Sixth Hawaii International Conference on System Sciences*, Vol. 1, pp. 349-358, 1993.
- [9] G. E. Sobelman and K. M. Fant, "CMOS Circuit Design of Threshold Gates with Hysteresis," *IEEE International Symposium on Circuits and Systems (II)*, pp. 61-65, 1998.
- [10] S. C. Smith, R. F. DeMara, M. Hagedorn, and D. Ferguson, "Delay-Insensitive Gate-Level Pipelining," *Integration, The VLSI Journal*, Vol. 30/2, pp. 103-131, October 2001.
- [11] 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.