

Material Implication in CMOS: A New Kind of Logic

Elkim Roa, Wu-Hsin Chen and Byunghoo Jung
Electrical and Computer Engineering Department
Purdue University, West Lafayette, IN USA
{elkim, chen279, jungb@purdue.edu} *

ABSTRACT

For more than seventy years, all the development in digital electronics have been founded on Shannon's work based on the fact that Boolean logic operators, OR, AND and NOT, can form a computationally complete logic framework. We propose a new paradigm in logic circuit design using material implication logic operators, different from the traditional logic gates in implementation and operation. In this paper we present early evidences, with experimental silicon results, showing that this new logic framework significantly improves performance, power and speed, over an equivalent conventional-logic framework in CMOS. This new computing paradigm would enable the continuance of increasing computing functionality and performance with decreasing cost in silicon technologies.

Categories and Subject Descriptors

B.6 [Logic Design]: Design Styles - Combinatorial Logic

General Terms

Performance, Design, Measurement, Theory

Keywords

Material Implication, High-Speed Logic

1. INTRODUCTION

A hundred years ago, the first volume of Whitehead and Russell's monumental work Principia Mathematica was published [1]. The book presented how the truths of math could be derived from logic operations. They described four fundamental logic operations, three of them used 27 years later by Shannon [2], and a fourth logic operation called material implication, $A \text{ IMP } B$ (also denoted as $A \rightarrow B$), which reads A implies B , and is equivalent to $(\text{NOT } A) \text{ OR } B$. Russell and Whitehead emphasized the relevance of this logic

*E. Roa is also with U. Industrial de Santander in Colombia.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2012, June 3-7, 2012, San Francisco, California, USA.

Copyright 2012 ACM ACM 978-1-4503-1199-1/12/06 ...\$10.00.

operation by showing that only the IMP and NOT operations are enough to form a computationally complete logic basis. Because of the fact that Boolean logic operations, OR, AND, and NOT, form a computationally complete logic basis and they can be easily implemented using switching devices, modern digital electronics have been founded on Boolean algebra, often referred to as switching algebra. In addition, the great achievement in modern electronics seems to indicate no need for extra logic functions. Consequently, the fourth fundamental logic operator, material implication ($A \text{ IMP } B$), has been ignored during last a few decades and disappeared in many electronics textbooks (still plays an important role in logic theory textbooks). A year ago, a letter in the journal Nature, presented that memristors can naturally execute the material implication operation, and inherently provides logic-in-memory [3]. However, these devices were fabricated using two layers of platinum-wire separated by an active layer of TiO_2 , requiring a different process platform than current silicon process. Furthermore, reported speed performance was considerably low.

In this research, we will show that CMOS circuits can also be used to perform material implications, IMP and NIMP (negated IMP). They are unit gates, which are not based on traditional Boolean logic operation, $(\text{NOT } A) \text{ OR } B$. After presenting their effectiveness for high-speed and low power operation in a frequency divider design, we will discuss about their potential extension to a complete logic basis, which would affect modern computing paradigm if successful.

2. IMPLICATION LOGIC IN CMOS

A set of logic functions is called complete if and only if any possible logic function can be composed by a combination of functions of the set. The operation logic NAND by itself, or NOR by itself, can be used to construct any Boolean logic, making the NAND and the NOR complete operations. It has been proved that the following set of pairs are complete: $\{\text{OR}, \text{NOT}\}$, $\{\text{AND}, \text{NOT}\}$, $\{\text{IMP}, \text{NOT}\}$, $\{\text{IMP}, \text{FALSE}\}$, $\{\text{NIMP}, \text{NOT}\}$, $\{\text{NIMP}, \text{TRUE}\}$, $\{\text{IMP}, \text{XOR}\}$, $\{\text{NIMP}, \text{XNOR}\}$ and $\{\text{NIMP}, \text{IMP}\}$. As a consequence, we can take any pair and develop a complete logic system with their respective algebra. Interestingly, the operation cells, IMP and NIMP, are common in most of the pairs. The truth table for the basic material implication operation is indicated in Figure 1a, meaning that if A is true, then the output after the implication operation will be B . The stateful nature of the implication operation, automatically provides a stateful logic operation if the operation can incorporate a memory. For example, implementing the IMP and NOT op-

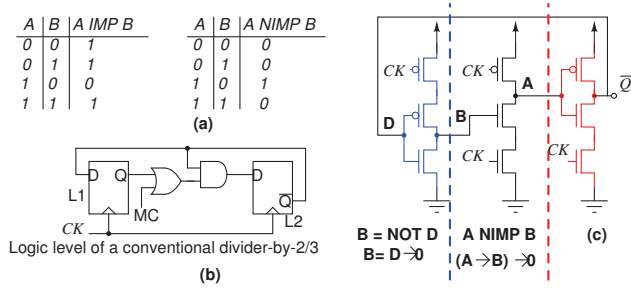


Figure 1: (a) Truth table of IMP and NIMP. (b) Conventional frequency divider-by-2/3. (c) Frequency divider-by-2/3 using NIMP operation.

erations in such a way that they behave as memory devices at the same time, enables them to recursively synthesize all the logic operations, as shown in Table 1.

Table 1: Boolean operations using material implication and false operations.

NOT A	$A \rightarrow 0$
A OR B	$(A \rightarrow 0) \rightarrow B$
A NOR B	$((A \rightarrow 0) \rightarrow B) \rightarrow 0$
A AND B	$(A \rightarrow (B \rightarrow 0)) \rightarrow 0$
A NAND B	$A \rightarrow (B \rightarrow 0)$
A XOR B	$(A \rightarrow B) \rightarrow ((B \rightarrow A) \rightarrow 0)$
A XNOR B	$((A \rightarrow B) \rightarrow ((B \rightarrow A) \rightarrow 0)) \rightarrow 0$

3. DESIGN EXAMPLE

The operations NAND, NOR, OR, AND and NOT, have been used so far as the key gates to synthesize most of the digital logic in CMOS. However IMP and NIMP operators, can provide a shortcut function to optimize certain logic operations at circuit level. We present a dual-modulus divide-by-2/3 shown in gate level at Figure 1b as an example. This is an appropriate example considering it involves combinatorial and sequential operators. This circuit is commonly used in frequency synthesizers where speed performance and power consumption are key features. The conventional wisdom is to synthesize the circuit using the traditional switching logic concepts proposed by Shannon. For instance, recently published prescalers have proposed to add the AND and OR logic gates within the flip-flops to reduce the number of stages in cascade, reducing the path delay [4]. However, in this approach, parasitic capacitances at internal nodes increase, requiring greater current driving capability and consequently more power consumption and slower speed.

Alternatively, the logic gate that includes AND and OR can be replaced by a dynamic NIMP to provide the same functionality. The NIMP operation is achieved by shifting the DC level of the clock signal connected to the flip-flop shown in Figure 1c. By doing this, the operation performed at node A is $(A \rightarrow B) \rightarrow 0$, which is the same as A NIMP B. The complete performed operation within the flip-flop is explained in Table 2, where 0+ and 1+ indicate the new logic values with the clock level shifted up. As a result, the final operation is $(A \rightarrow (D \rightarrow 0)) \rightarrow 0$, where B is $D \rightarrow 0$. This operation is equivalent to A AND D, which is the

Table 2: NIMP operation explained.

A	B	$ACK=0$	$ACK=1$	$ACK=1+$	$ACK=0+$
0	0	1	0	0	0
0	1	1	0	0	0
1	0	1	1	1	1
1	1	1	0	0	0

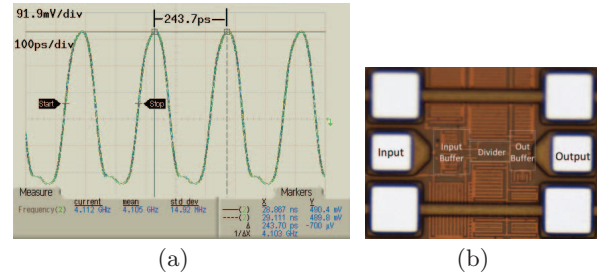


Figure 2: (a) Measured output in divide-by-3 operation mode. (b) Microphotograph of divider-by-2/3.

required function to have the circuit operating in divide-by-3 mode. The divide-by-2 operation mode is performed by the using the normal clock DC level, where the circuit operates as a flip-flop connected in feedback mode. Consequently, a divider-by-2/3 is designed using just one flip-flop by considering the NIMP and NOT operations without requiring extra logic gates. Experimental results indicate a 2.5 folds of speed improvement and a 4 folds of power reduction compared to the traditional AND and OR based frequency dividers presented in [4]. The measured output of the frequency prescaler in divide-by-3 operation mode, is presented in Figure 2(a) given an input frequency of 12.3GHz. The prescaler is implemented in $0.13\mu\text{m}$ 1.2V CMOS technology. Figure 2(b) shows a chip microphotograph.

4. DISCUSSION

We have introduced new implication logic cells significantly different from those conventionally used in Boolean logic cells. The initial evaluation based on a reconfigurable 2/3 bit counter shows that, these new logic cells are effective in achieving high-speed and low-power operation and have a great potential to form a highly efficient complete functional logic library based on {IMP, NOT}, {IMP, FALSE}, {NIMP, NOT}, {NIMP, FALSE}, or {NIMP, IMP} pairs. If successful, it will open a plethora of new applications and approaches for logic synthesis for future high-speed and low-power computing systems.

5. REFERENCES

- [1] A. Whitehead and B. Russell, *Principia Mathematica*. Cambridge at the University Press, 1910.
- [2] C. Shannon, "A Symbolic Analysis of Relay and Switching Circuits," Master's thesis, MIT, 1937.
- [3] J. Borguetti et al., "'Memristive' Switches Enable 'Stateful' Logic Operations Via Material Implication," *Nature*, vol. 464, pp. 873–876, April 2010.
- [4] M. Khrisna et al., "Design and Analysis of Ultra Low Power True Single Phase Clock CMOS 2/3 Prescaler," *IEEE Transaction on Circuits and Systems I*, vol. 57, no. 1, pp. 72–82, Jan. 2010.