Low Power VLSI Circuit Implementation using Mixed Static CMOS and Domino logic with Delay Elements

Rajib Kar, Durbadal Mandal, Gaurav Khetan, Sunil Meruva
Department of Electronics and Communication Engineering
National Institute of Technology, Durgapur, West Bengal, INDIA
rajibkarece@gmail.com

Abstract— **The advent of dynamic CMOS logic, more precisely domino logic, made them widely used for the implementation of low power VLSI circuits. However, the main drawback of this logic is the non implementation of inverted logic. To implement the inverted logic, it is required to duplicate the logic circuit up to that part with inverted inputs. This obviously results the increase in area, delay as well as the power dissipation of the circuit. On the other hand, it is very simple to realize the circuit with both the inverted and non-inverted logic using static CMOS implementation. In this paper, this problem is addressed with the realization of the circuit which requires the implementation of inverted logic using mixed static and domino logic. To show the efficiency of the proposed model, a simple example like implementation of high fan-in NAND gate cascaded with AND gate is considered. With the comparison of all the three logics with a fixed fan-in of 7, 8 and 9 for both the gates, on an average 69.7% improvement is achieved in Power Delay Product (PDP), 11.4% improvement in area in terms of transistors using mixed logic implementation over static logic implementation and 68.64% improvement in PDP and 28.4% improvement in area over dynamic CMOS implementation when designed in 180nm technology.**

Keywords- Low Power VLSI; Static CMOS; Domino Logic; Mixed CMOS; Power Delay Product

I. INTRODUCTION

For the implementation of low-power and high-speed VLSI circuits, dynamic CMOS in particular domino logic is the logic of choice [1]. However, domino logic has many inherent limitations like charge leakage, charge sharing, clock skew etc. [2]. The main disadvantage in implementing domino logic is that it can implement only non-inverting logic. The requirement of implementation of inverted logic forces the designer to duplicate the entire circuit before that inverter with opposite polarities of inputs which increases the number of gates in the circuit which in tern increases the power dissipation and delay of the entire circuit. Hence the efficiency of domino logic is challenged if the circuit requires the implementation of more intermediate inverters. The implementation of static CMOS is efficient as it can implement both the inverted and non-inverted logic. However static CMOS logic is slower than dynamic logic and suffers from large area and high short circuit power dissipation [6]. On the other hand dynamic CMOS logic has less transistor count and zero short circuit power dissipation. Now considering the advantages of both logic styles, in this paper, novel circuit architecture, using mixed static and dynamic CMOS logic has been proposed.

Very few attempts have been made to implement a given circuit using mixed logic. Out of which an approach called two phase static-domino design [4-5] used two out of phase clocks, master and slave flip flops. In the first domino evolution phase, domino logic and static CMOS logic gets evaluated but the output of static CMOS logic is fed to domino logic only in the second phase of evaluation. The presence of two clocks in this design results in inevitable problems like clock skew and clock routing overheads. The requirement of mid-cycle latches between two domino phases often degrades the performance of the circuit. In another approach [3], instead of static inverter in a domino gate, complex static gate has been used, which follows strict sequence of domino gate-static gate-domino gate, called DS domino gate. Whenever the intermediate inverter implementation was necessary, [3] used either a dual output domino gate or dual output static gate. This approach requires more number of parts to be implemented using static CMOS which degrades the circuit performance in terms of area, power dissipation and delay.

In this paper, a given circuit is realized using mixed static CMOS logic and domino logic which need not be a sequence of Domino gate - Static gate - Domino gate [3] and the output of the Static gate can be fed to the next gate in every evolution phase not like in [4-5]. The parts of the circuit which requires the implementation of inverted logic is implemented using static CMOS logic where as others using domino logic. So this requires less transistor number compared to both the logics which in tern reduces the delay and power dissipation of the circuit.

The rest of the paper is organized as follows: Section 2 discusses the design issues in implementation of mixed static-domino logic, section 3 describes the proposed model of implementation using mixed logic and section 4 shows the schematics of the circuits using all the three logic styles designed in 180nm technology. Section 5 discusses the simulation results and finally section 6 concludes the paper.

978-1-4673-0102-2/11/$26.00 ©2011 IEEE

II. Static Domino Interface Constraints

Though the domino logic requires less area and dissipates less power, the design of the circuit using domino logic is more complex compared to static CMOS logic. In this section, the timing constraints are discussed that must be satisfied in order to mix the static CMOS logic and domino logic and realize the given circuit using mixed static-domino logic efficiently. In contrast to static CMOS logic, as each domino gate is clocked, certain set up and hold constraints are imposed. If the output node of the domino logic is erroneously discharged by an incorrect input in evaluation phase, it cannot be recharged until next precharge phase. So, in order to design an efficient mixed logic, there should be a harmony between domino input signals, the output signals of the domino gate and the static input signals, output signals of the static CMOS gate. The timing constraints are explained in the following sub-sections which need to be followed, namely when domino input signal fed to domino gate and static input signal to domino gate

A. Domino Input Signal to Domino Gate

For correct operation of the gate, the following set up and hold constraints must be followed when a domino input signal is fed to domino gate. Input signal must rise before CLOCK falls i.e. before the end of evaluation cycle. This ensures that the domino gate evaluates before the end of evaluation cycle. Input signal must fall before CLOCK rises i.e. the data input must precharge before the start of next evaluation cycle. This constraint ensures that domino gate output is precharged before the end of precharge cycle. Further domino input signal falling must be held till the dynamic output node settles. The domino logic gate precharges when the clock is low and evaluates when the clock is high. This implies that the output of every domino gate pre-charges before the evaluation cycle and it conditionally rises during evaluation cycle. Thus when the domino input signal is fed to domino gate, the above timing constraints are naturally satisfied as long as the cascaded sequence of the domino gates are evaluated within the evaluation phase.

B. Static Input Signal to Domino Gate

The following timing constraints should be followed when a static input signal i.e. output signal from static CMOS gate is fed to domino gate. Input signal must be glitch free. It is not expected that output from static gate satisfies the set up and hold constraints which are necessary for domino gate. The input signal from static gate is inherently glitchy i.e. it may switch multiple times before settling. So, when a glitchy input signal is fed to domino gate, it may erroneously discharge the output of the domino gate during the evaluation cycle instead of keeping it high. Once the output of the domino gate discharges, there is no such mechanism available that can precharge before the next precharge cycle. Static input signal must be available during the evaluation cycle. The static signal coming from the static gate does not follow the precharge and evaluation characteristics of the domino signal i.e. it does not reset and conditionally rises in every clock cycle. Thus, the rise and fall transitions of the static input signal may happen in any of the two cycles (precharge and evaluation) and may erroneously discharge the dynamic node causing the incorrect evaluation of the domino gate. So, for the domino gate to evaluate correctly, the static signal should also follow the same set up and hold timing constraints as domino signal. Thus, a reset or fall transition of static signal should occur and settle before the onset of the domino evaluate cycle.

III. Proposed Mixed Signal Logic Design

Keeping in mind the above timing constraints, both the logics are mixed. The architecture is implemented using domino logic when it needed non-inverted logic and static CMOS logic when it needed inverted logic. In this section, it is explained how the static gate is made to satisfy the timing constraints using a block diagram. Some delay elements are used too wherever it is required to maintain the synchronization and to get glitch free output.

Fig. 1 is the block diagram of the static gate that satisfies the timing constraints and gives the glitch free signal as output which is fed to a domino gate. The working principle of this block is expatiated below. This static gate consists of Pull-Up Network (PUN) and Pull-Down Network (PDN) with extra two transistors to satisfy the timing constraints. One of the extra transistors is NMOS which is connected between Source V_{dd} and PUN. The other is PMOS transistor connected between output of the gate and ground terminal as shown in Fig. 1. Both the transistors have same clock. When the clock is low, the NMOS is off and PMOS is on, which makes the output to discharge to zero. So, in pre-charge phase the output will be low which is considered to be the desired condition as an input to the next domino gate. When the clock goes high, evaluation phase begins and the desired output is available, as NMOS transistor is on and PMOS transistor is off. In this way, the static gate gives the correct input to the domino gate and the synchronization is maintained.

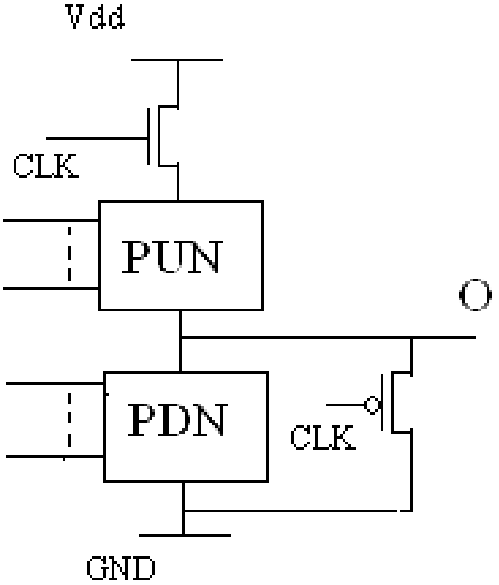

Figure 1. Static Gate Block Diagram

IV. Schematics Using Proposed Mixed Logic

In this section, the difference in models between three logics namely, static CMOS, domino and mixed logic are discussed in brief using the schematics drawn in 180nm technology for the standard circuit. The examples of NAND gate cascaded with AND gate is considered with high fan-in of 7, 8 and 9, respectively.

A. *Using Static CMOS Logic*

The following schematic (Figure 2) is the implementation of 7 input NAND gate cascaded with 7 input AND gate using static CMOS logic.

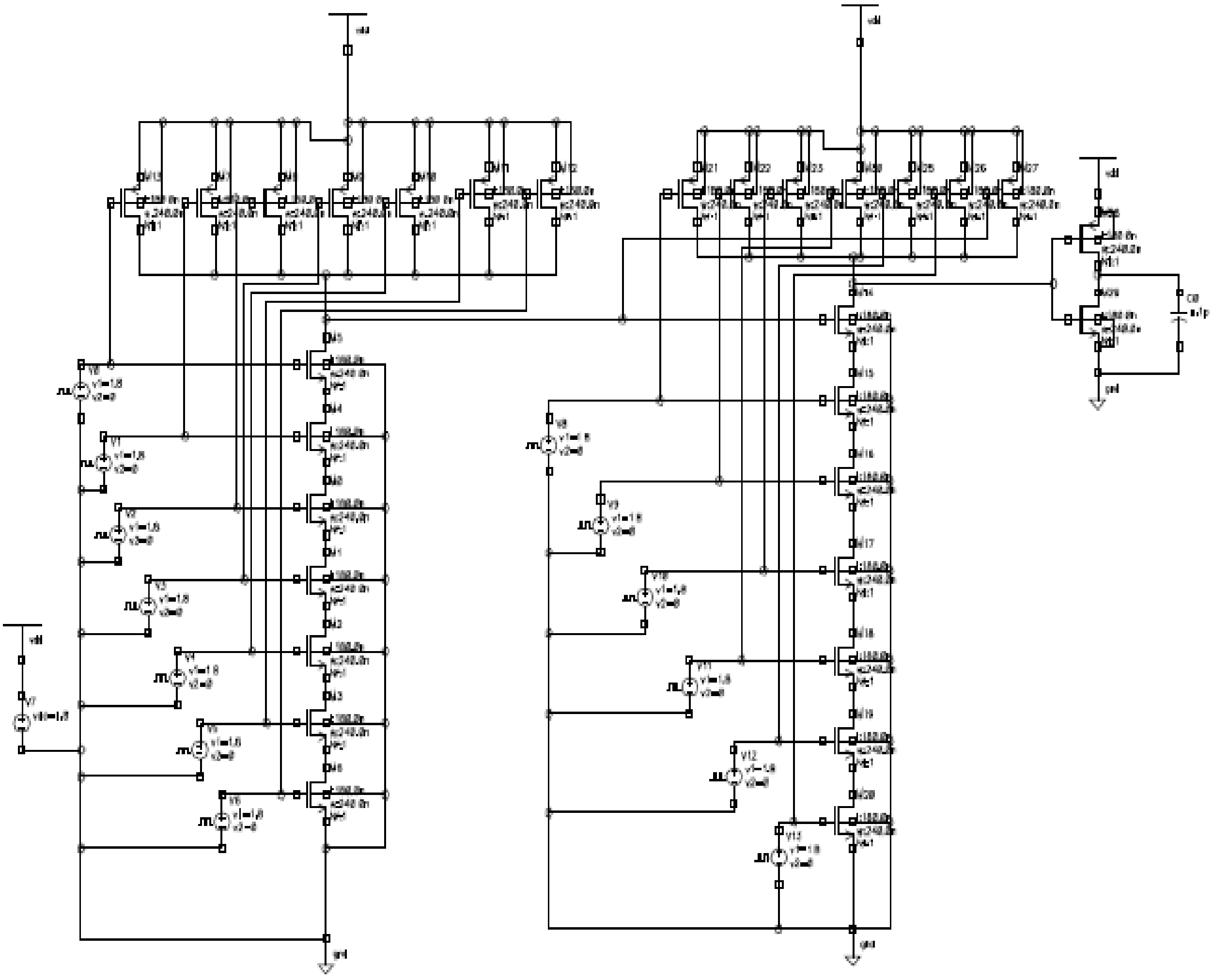

Figure 2. Implementation Using Static CMOS Logic

B. *Using Domino Logic*

This schematic shown in Fig.3 is the implementation of 7 input NAND gate cascaded with 7 input AND gate using domino logic. Since this circuit requires the implementation of inverted logic, the entire circuit before that inverter is duplicated, with opposite polarities of inputs as shown below.

This results in the increases of the transistor count which in turn increases the power dissipation and delay of the circuit. Unlike the keeper transistors in [8-9], in which power dissipation of the circuit has been reduced but it requires the sacrifice of area a normal keeper transistor is used to mitigate the problem of charge sharing, charge leakage and proper delay is provided to each and every input which is slightly greater than propagation delay of the previous stage [7] to maintain the synchronization of the circuit.

C. *Using Mixed Logic*

Since the circuit demands for the implementation of inverted logic, the part of the circuit requiring the implementation of inverter can be replaced with proposed static gate as shown in Fig.4. It is clear from the schematic that number of transistors required to implement using this logic is less compared to both static CMOS logic and domino logic. The part of the circuit in rectangular box is the static part following timing constraints.

Proper delay is given to the inputs and clock, based on the propagation delay of the previous elements of the circuits to maintain the synchronization of the circuit and for efficient functioning of the logic.

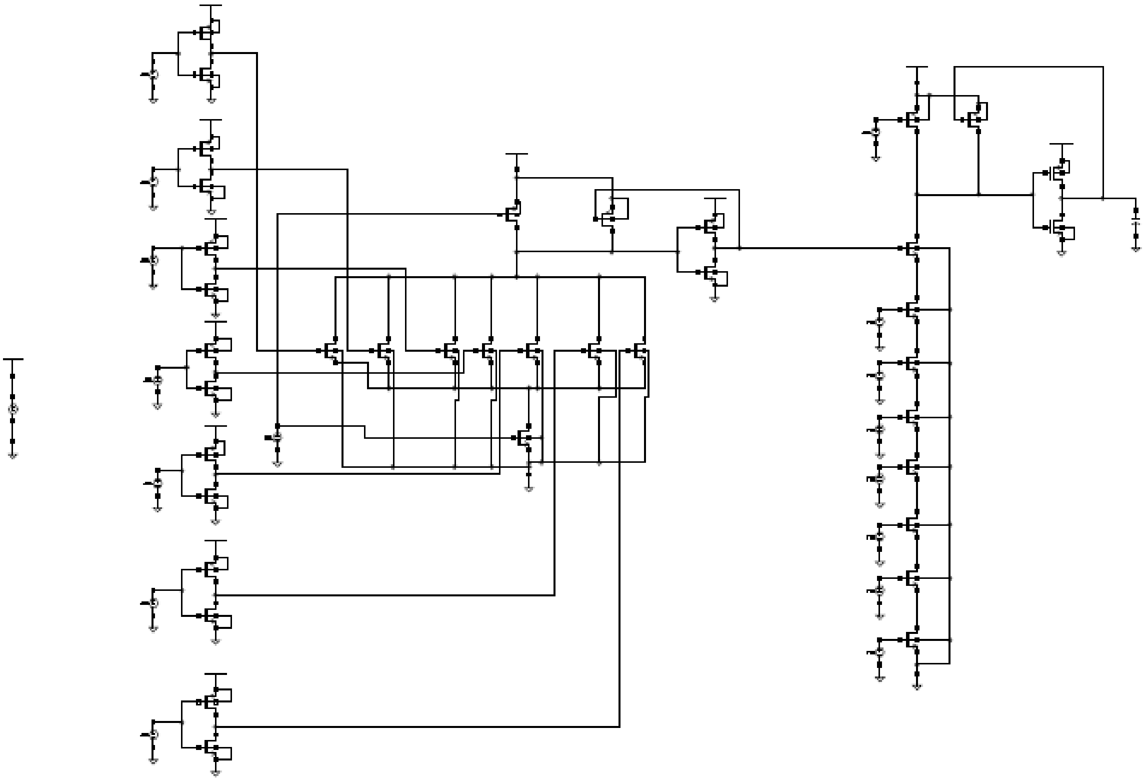

Figure 3. Implementation Using Domino logic

V. Simulation Results

The simulations are performed in 180nm technology node for all the three logics using same process parameters. The input to each circuit is pulsated in such a way that it covers all the possible inputs. The delay is calculated in ns, power dissipation in mW and area in terms of transistors. The following tables give the % of improvement in delay, area, power and Power Delay Product (PDP) for the mixed logic over static and domino logic.

VI. Conclusions

Despite many difficulties in implementing domino logic, we implemented mixed logic effectively bounding to timing constraints. Using this mixed logic the problem of implementation of inverted logic can be eradicated. With this logic we achieved on an average 11.4% improvement in area, 69.7% improvement in PDP over static logic and 28.4% improvement in area, 68.64% improvement in PDP over domino logic for the standard examples. This model can be applied for implementation of any intermediate inverted logic and it need not be a strict domino gate – static gate – domino gate and this can be extended for implementing not only an inverter but also other inverting gates.

TABLE I. % Improvement of Mixed Logic over Static Logic

Fan-in	**% Improvement in**			
	Area	*Delay*	*Power Dissipation*	*PDP*
7	6.6	52.36	13.15	58.6
8	11.76	54.2	64.39	83.7
9	15.7	49.8	33.94	66.8

TABLE II. % Improvement of Mixed Logic over Domino Logic

Fan-in	**% Improvement in**			
	Area	*Delay*	*Power Dissipation*	*PDP*
7	26.3	2.1	57.40	58.3
8	28.5	5.1	70.13	71.6
9	30.4	6.2	74.38	75.9

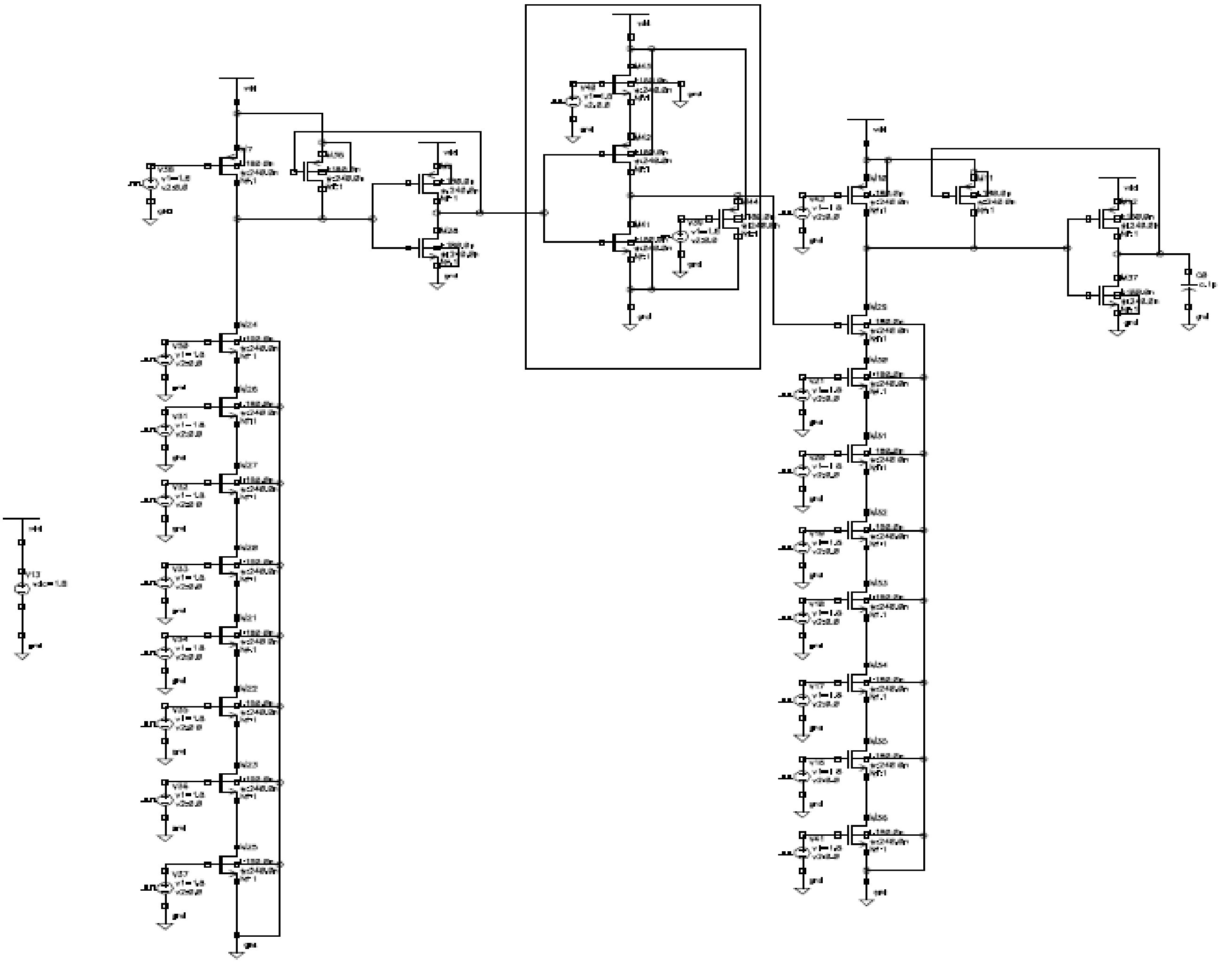

Figure 4. Implementation Using Mixed Logic

REFERENCES

[1] R.H. Krambeck, C.M. Lee and H. Law, High-speed compact circuits with CMOS, IEEE Journal of Solid-State Circuits June.

[2] Ki-Wook Kim, C. L. Liu and Sung-Mo Kang. Implication Graph based Domino Logic Synthesis, IEEE International Conference on Computer-Aided Design, Digest of Technical Papers, Nov 1999, pp. 111 – 114.

[3] T. Thorp, Gin Yee and C. Sechen, Domino logic synthesis using complex static gate, IEEE International Conference on Computer-Aided Design, Digest of Technical Papers, Nov 1998 pp. 242-247.

[4] Ruchir Puri, Design issues in mixed static-domino circuit implementations, IEEE International Conference on Computer-aided design, proceedings, October 1998 pp.270-275.

[5] Min Zhao and Sachin S. Sapatnekar, Timing-driven Partitioning for Two-Phase Domino and Mixed Static/Domino Implementations, IEEE International Conference on Computer-aided design, Digest of Technical Papers, November 1999 pp. 107-110.

[6] A. Bellamour, and M. I. Elmasri, Low Power Digital VLSI Design: Circuits and Systems, Kluwer Academic Press, 1995.

[7] Amir Amirabadi, Ali Afzali-Kusha, Yousof Mortazavi and Mehrdad Nourani, Clock Delayed Domino Logic With Efficient Variable Threshold Voltage Keeper, IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 15, No. 2, February 2007, pp. 125-134.

[8] Farhad Haj Ali Asgari, and Majid Ahmadi, Jonathan Wu, Low Power High Performance Keeper Technique for High Fan-in Dynamic Gates, Circuit Theory and Design, 2009. ECCTD 2009 pp. 523-526.

[9] Jinhui Wang, Wuchen Wu, Na Gong and Ligang Hou, Domino Gate with Modified Voltage Keeper, 11th Int'l Symposium on Quality Electronic Design 2010 pp. 443-446.