

3-1-1997

REDUCING GLITCHING AND LEAKAGE POWER IN LOW VOLTAGE CMOS CIRCUITS

Zhanping Chen

Purdue University School of Electrical and Computer Engineering

Liqiong Wei

Purdue University School of Electrical and Computer Engineering

Kaushik Roy

Purdue University School of Electrical and Computer Engineering

Follow this and additional works at: <http://docs.lib.purdue.edu/ecetr>

Chen, Zhanping; Wei, Liqiong; and Roy, Kaushik, "REDUCING GLITCHING AND LEAKAGE POWER IN LOW VOLTAGE CMOS CIRCUITS" (1997). *ECE Technical Reports*. Paper 85.

<http://docs.lib.purdue.edu/ecetr/85>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

REDUCING GLITCHING AND
LEAKAGE POWER IN LOW VOLTAGE
CMOS CIRCUITS

ZHANPING CHEN
LIQIONG WEI
KAUSHIK ROY

TR-ECE 97-6
MARCH 1997



SCHOOL OF ELECTRICAL
AND COMPUTER ENGINEERING
PURDUE UNIVERSITY
WEST LAFAYETTE, INDIANA 47907-1285

Reducing Glitching and Leakage Power in Low Voltage CMOS Circuits Using Multiple Threshold Transistors

Zhanping Chen, Liqiong Wei, and Kaushik Roy
Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907-1285
Ph: 765-494-3372
Fx: 765-494-3371
e-mail: {zhanping, liqiong, kaushik}@ecn.purdue.edu

Area: Design & Test, Computer-Aided Design, and Technology

Contact person: Kaushik Roy
Ph: 765-494-2361
Fx: 765-494-3371
e-mail: kaushik@ecn.purdue.edu



Reducing Glitching and Leakage Power in Low Voltage CMOS Circuits Using Multiple Threshold Transistors

Abstract

The need for low power dissipation in portable computing and wireless communication is making design communities accept ultra low voltage CMOS processes. With the lowering of supply voltage, the transistor thresholds (V_{th}) have to be scaled down to meet the performance requirements. However, such scaling can increase the leakage current through a transistor, thereby increasing the leakage power. It can also be noted that in static CMOS circuits, the paths converging to any internal gate may have different propagation delays. The delay mismatch of different paths causes spurious transitions. Such transitions increase the power dissipation due to the switching component of current. In this paper we present a novel algorithm to balance different paths of a design converging to logic gates using *multiple threshold transistors* such that both power dissipation due to spurious transitions and leakage current are minimized. Leakage power is reduced due to the use of high threshold transistors in the non-critical paths. Results for ISCAS benchmark circuits show that the glitching power can be minimized by more than **30%** using three different threshold voltages. The practicality of multiple threshold designs using dual-gated SOI (Silicon-On-Insulator) technology is also discussed.

1 Introduction

With the growing use of portable and wireless electronic systems, reduction in power consumption has become one of the main concerns in today's VLSI circuit and system design.

For a CMOS digital circuit, power dissipation includes three components [8]: switching power dissipation ($P_{switching}$), short-circuit power dissipation ($P_{short-circuit}$), and static leakage power dissipation ($P_{leakage}$). The average power dissipation can be expressed by,

$$\begin{aligned} P_{average} &= P_{switching} + P_{short-circuit} + P_{leakage} \\ &= \alpha C_L V_{DD}^2 f_{clk} + I_{sc} V_{DD} + I_{leakage} V_{DD} \end{aligned} \quad (1)$$

where α is the switching activity (average number of switching per clock period), C_L is the load capacitance, f_{clk} is the clock frequency, I_{sc} is the direct-path short circuit current, $I_{leakage}$ is the leakage current, and V_{DD} is the supply voltage. Lowering supply voltage is obviously the most effective way to reduce the power consumption. With the scaling of the supply voltage, the transistor threshold voltages should also be scaled in order to satisfy the performance requirements. Unfortunately, such scaling leads to the increase of the leakage current through a transistor [5]. Therefore, the leakage power cannot be ignored for low voltage low power circuit designs.

Among the three sources of power dissipation, switching current (which charges or discharges load capacitances of logic gates) produces a majority of the power dissipation. Switching includes functional transitions and spurious transitions (glitches). Due to the delay of each gate, paths arriving at one internal gate may have different propagation delays. Therefore, a gate exhibits multiple spurious transitions before settling to the correct logic level. In typical combinational logic circuit, spurious transitions account for between **10%** and **40%** of switching power. In some circuits such as combinational adders [2, 6], the power dissipation caused by glitches can even be as high as **70%** of the total power. In order to reduce the power dissipation caused by spurious transitions, the delays of different paths converging to each logic gate should be roughly equal. One method to balance such paths is to selectively add buffers in small delay paths [7, 9]. However, the added circuitry not only increases overall circuit area but also consumes power itself. Therefore, the costs of additional buffers may offset the reduction of spurious transitions [3]. Gate re-sizing [1] is another method to balance paths. By replacing a gate with a logically equivalent but smaller cell, the delay of that gate can be changed. However, with the reduction of transistor size, direct path current will increase. This in turn increases the power component due to direct path current. Also, in custom design, transistors in non-critical paths are normally small and hence, gate re-sizing may not be that effective.

In this paper, we present a new method to reduce power dissipation under performance constraints using *multiple threshold transistors*. Since the delay of each gate is a function of the threshold voltage, by increasing the threshold voltages of those gates in non-critical paths, the propagation delays along different paths converging to the same gate can be balanced so that spurious transitions are minimized. The performance will be maintained because it is determined by

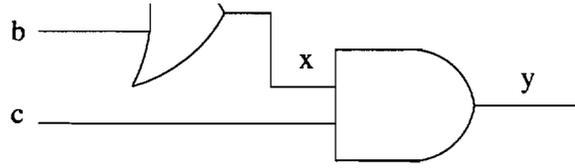


Figure 1: Glitching in static CMOS circuit

the critical path. Choosing higher threshold voltages for the transistors in non-critical paths can also reduce leakage current in such paths. Therefore, *multiple* threshold *transistors* technique can be efficient in reducing power dissipation of low voltage CMOS circuits.

The rest of the paper is organized as follows. In section 2, necessary definitions are introduced. Principles regarding the use of multiple threshold transistors to balance paths are discussed in section 3. Section 4 describes a heuristic algorithm to properly assign different threshold voltages to different transistors. The pseudo-code of our program is also given in this section. The practicality of multiple threshold designs using different technologies is discussed in section 5. Section 6 presents the implementation details and experimental results on ISCAS benchmark circuits. Finally, conclusions are given in Section 7.

2 Definitions

A combinational circuit can be represented as a directed acyclic graph. Each node in the graph corresponds to a logic gate in the circuit while each edge corresponds to a path. In this paper, we have used the terms "node" and "gate" interchangeably. In this section we will give the necessary concepts.

2.1 Spurious transition

In a CMOS circuit, due to the delay mismatch along different fan-in paths of each internal gate and primary output, a gate may have unexpected transitions (glitches or spurious transitions) before settling to the correct logic level. Figure 1 is a simple circuit used to show how a glitch occurs when two or more paths having different delays converge to a logic gate. For simplicity, the two gates are assumed to have unit delay. All the primary inputs are assumed to change simultaneously. Because of the finite delay of each logic gate, a spurious transition occurs at the primary output y before it settles to the expected logic value (logic "Low"). The corresponding waveform is shown in Figure

2.2 Propagation delay

The propagation delay of node x , denoted as $t_p(x)$, defines how quickly the output responds to a change in its input. The propagation delay of a path π_j , denoted as $Pd(\pi_j)$, is the sum of the

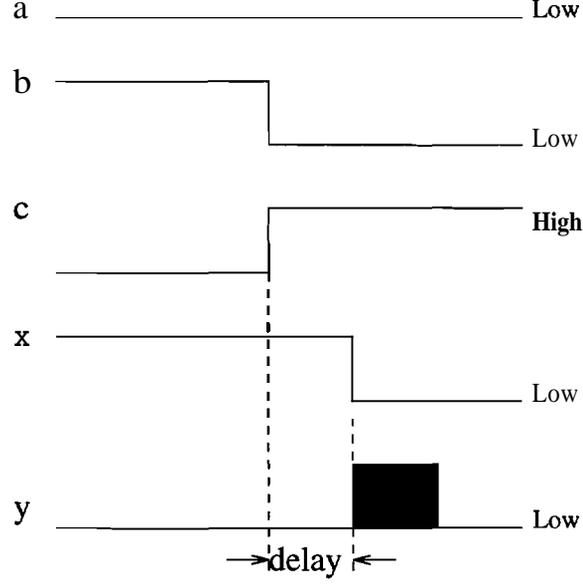


Figure 2: Wave form to show how a glitch occurs

propagation delays $t_p(i)$ of each node i along this path. It can be expressed as follows,

$$Pd(\pi_j) = \sum_{i=1}^{m_j} t_p(i) \quad (2)$$

where m_j is the number of nodes along path π_j .

For a node x , among all the paths arriving at this node (fan-in paths), there exists a path (or paths) which has a maximum propagation delay value $MAX(x)$ and a path (or paths) which has a minimum propagation delay value $MIN(x)$.

$$MAX(x) = \max\{Pd(\pi_j)\} \quad (3)$$

$$MIN(x) = \min\{Pd(\pi_j)\} \quad (4)$$

where $Pd(\pi_j)$ is the propagation delay of each path π_j arriving at node x . The propagation delay along a fan-out path π_k of node x can be calculated as follows,

$$Pd(\pi_k) = t_p(x) + MAX(x) \quad (5)$$

For each primary input x , $MAX(x) = 0$, $MIN(x) = 0$, and $t_p(x) = 0$. For each node x in level 1 (level of a node is equal to the maximum of the level of its fan-in nodes plus 1; level of all primary inputs being 0), $MAX(x) = 0$, $MIN(x) = 0$. Using equation 3, 4, and 5, level by level, we can calculate $MAX(x)$ and $MIN(x)$ associated with each node x , and the propagation delay associated with each path.

2.3 Imbalance measure \mathcal{M}_D

Since a spurious transition results from the delay mismatch of different fan-in paths, no new spurious transition can occur at a logic gate which has only one fan-in node. Such a node can only propagate glitches generated at its fan-in node.

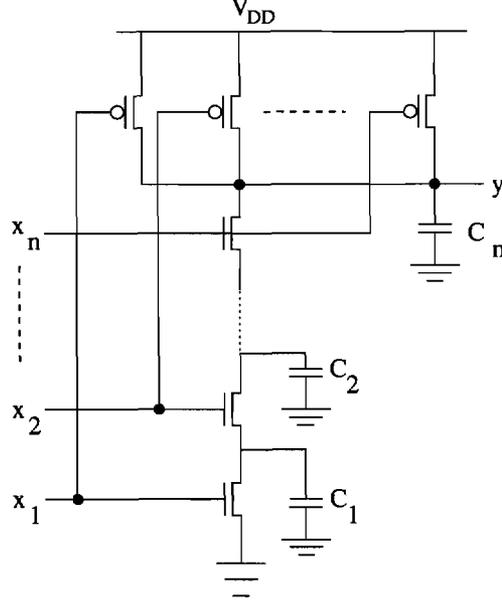


Figure 3: An n-input NAND

For a node x which has two or more fan-in nodes, the difference between the two values $MAX(x)$ and $MIN(x)$ has an important effect on the glitches generated at this node. Therefore, we define imbalance measure $\mathcal{M}_{\mathcal{D}}$ as the sum of the difference between the two values $MAX(x)$ and $MIN(x)$ associated with each node:

$$\mathcal{M}_{\mathcal{D}} = \sum_{all\ nodes} \delta(x) \quad (6)$$

where $\delta(x) = MAX(x) - MIN(x)$.

2.4 Glitching power measure $\mathcal{M}_{\mathcal{P}}$

Spurious transitions are unnecessary transitions. However, they will increase the power dissipation due to switching. Equation 1 indicates that switching power is proportional to the load capacitance. Therefore, we define *glitching* power measure, denoted by $\mathcal{M}_{\mathcal{P}}$, as a measure of the power dissipation caused by glitches,

$$\mathcal{M}_{\mathcal{P}} = \sum_{all\ nodes} \delta(x) C(x) \quad (7)$$

where $C(x)$ is the load capacitance of node x . For low power design, $\mathcal{M}_{\mathcal{P}}$ must be reduced in order to reduce total power consumption.

3 Multiple threshold transistors technique

In this section, we first explain how to use the Elmore delay model [4] to obtain the delay of each gate, with the capacitance of each internal node in a logic gate taken into consideration. Then we discuss the relationship between propagation delay and threshold voltage.

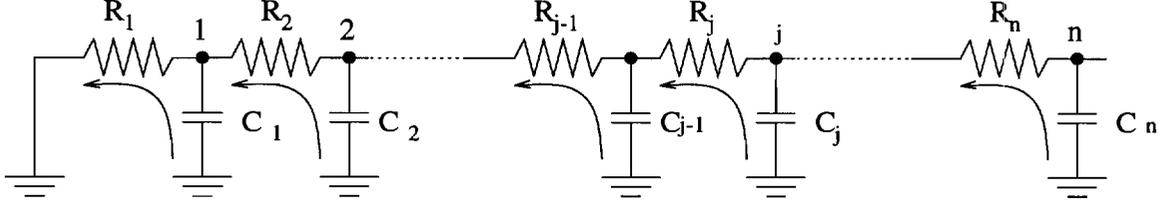


Figure 4: Equivalent pull-down network of n-input NAND gate

3.1 Elmore delay model

Let us look at the n-input NAND gate which is shown in Figure 3. It can be analyzed using an equivalent RC network. Each MOS transistor has an equivalent on-resistor R_j , and each internal node of the n-input NAND gate has a capacitance C_j . The transistors in the pull-down network are in series. Therefore, t_{pHL} is dominant.

The equivalent RC network of the pull-down network is shown in Figure 4. The worst case occurs when each C_j needs to be discharged. Based on the Elmore delay model, the worst case delay t_{pw} of the pull-down network is given by,

$$\begin{aligned} t_{pw} &= R_1 C_1 + (R_1 + R_2) C_2 + \cdots + (R_1 + R_2 + \cdots + R_j) C_j + \cdots + (R_1 + R_2 + \cdots + R_n) C_n \\ &= \sum_{j=1}^n C_j \sum_{k=1}^j R_k \end{aligned} \quad (8)$$

The best case occurs when only capacitance C_n needs to be discharged and all other capacitances have already been discharged. The best case delay t_{pb} of the pull-down network is given by,

$$\begin{aligned} t_{pb} &= (R_1 + R_2 + \cdots + R_{n-1} + R_n) C_n \\ &= C_n \sum_{j=1}^n R_j \end{aligned} \quad (9)$$

The capacitance of each internal node j (j varies from 1 to $n - 1$) in the n-input NAND gate is given as follows,

$$C_j = 2 C_{dn} \quad (10)$$

where C_{dn} is the diffusion capacitance of a NMOS transistor. The capacitance of internal node n is,

$$C_n = f_{anout}(C_{gp} + C_{gn}) + n C_{dp} + C_{dn} \quad (11)$$

where C_{dp} is the diffusion capacitance of a PMOS transistor, C_{gp} and C_{gn} are the gate capacitances corresponding to a PMOS transistor and an NMOS transistor respectively, f_{anout} is the fan-out number of the gate, and n is the number of fan-in gates. Assuming that a PMOS transistor and an NMOS transistor have the same diffusion capacitance C_d and the same gate capacitance C_g , we have,

$$C_j = 2C_d \quad (j = 1, 2, \cdots, n - 1) \quad (12)$$

and

$$C_n = 2f_{anout} C_g + (n + 1)C_d \quad (13)$$

Assuming that each NMOS transistor has the same on-resistance R , the worst-case delay and the best-case delay of the pull-down network are simplified as follows,

$$\begin{aligned} t_{pw} &= R_1 C_1 + (R_1 + R_2) C_2 + \dots + (R_1 + \dots + R_{n-1}) C_{n-1} + (R_1 + \dots + R_n) C_n \\ &= R(2C_d) + 2R(2C_d) + \dots + (n-1)R(2C_d) + nR(2f_{anout} C_g + (n+1)C_d) \\ &= R(2C_d) \frac{n(n-1)}{2} + nR(2f_{anout} C_g + (n+1)C_d) \\ &= R(2n^2 C_d + 2n f_{anout} C_g) \end{aligned} \quad (14)$$

and

$$\begin{aligned} t_{pb} &= (R_1 + \dots + R_n) C_n \\ &= R(n(n+1)C_d + 2n f_{anout} C_g) \end{aligned} \quad (15)$$

The real t_{pHL} varies from t_{pb} to t_{pw} . For simplicity, we take the average of the two,

$$t_{pHL} = \frac{t_{pw} + t_{pb}}{2} \quad (16)$$

Substituting equation 14 and 15 into 16, we have,

$$t_{pHL} = R\left(\frac{3n^2 + n}{2} C_d + 2n f_{anout} C_g\right) = R C_{avg} \quad (17)$$

where C_{avg} is equal to $(\frac{3n^2+n}{2} C_d + 2n f_{anout} C_g)$, and is determined by input number n , diffusion capacitance C_d , and gate capacitance C_g .

For an n -input NOR gate, the n PMOS transistors in the pull-up network are in series. Therefore, t_{pLH} is dominant. Following the same procedure, we get similar expressions. Complex gates such as XOR and XNOR can be viewed as a combination of NAND gates and NOR gates. Therefore, based on the Elmore delay model we can obtain the propagation delay of each gate in a logic circuit.

3.2 Relationship between delay and threshold voltage

Equation 17 indicates that the propagation delay of a CMOS gate is proportional to the equivalent on-resistance R of the transistors in a logic gate. Although the on-resistance R depends on the operation point and varies during the switching transient, we still can make a reasonable approximation by using a fixed value. This value is the average of the resistances at the end points of the transitions. Consider a CMOS inverter. The on-resistance R of the NMOS transistor is given by,

$$\begin{aligned} R &= \frac{R_{NMOS} |_{V_{out}=V_{DD}} + R_{NMOS} |_{V_{out}=V_{DD}/2}}{2} \\ &= \frac{1}{2} \left(\frac{V_{DS}}{I_D} |_{V_{out}=V_{DD}} + \frac{V_{DS}}{I_D} |_{V_{out}=V_{DD}/2} \right) \\ &= \frac{V_{DD}}{k_n (V_{DD} - V_{Tn})^2} + \frac{V_{DD}}{k_n (2(V_{DD} - V_{Tn})V_{DD} - (\frac{V_{DD}}{2})^2)} \end{aligned} \quad (18)$$

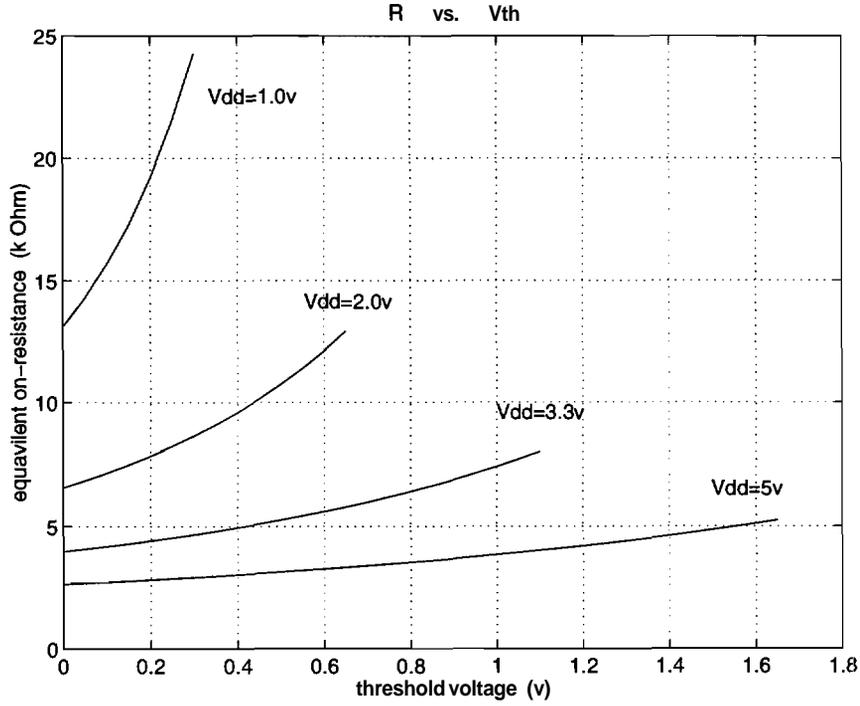


Figure 5: Relationship between R and V_{th}

where V_{Tn} is the threshold voltage of the NMOS transistor. k_n is the *gain factor* of a NMOS transistor. It is equal to the product of *process transconductance parameter* k' and the (W/L) ratio.

Figure 5 shows the relationship between equivalent on-resistance R and threshold voltage V_{th} with $k_n = 8.0 \cdot 10^{-5} A/V^2$ and $W/L = 1.8/1.2$. Given a fixed supply voltage V_{DD} , the on-resistance R changes with threshold voltage. Therefore, different threshold voltages result in different propagation delays of a gate.

In our method, the transistors in non-critical paths will be assigned to higher threshold voltages

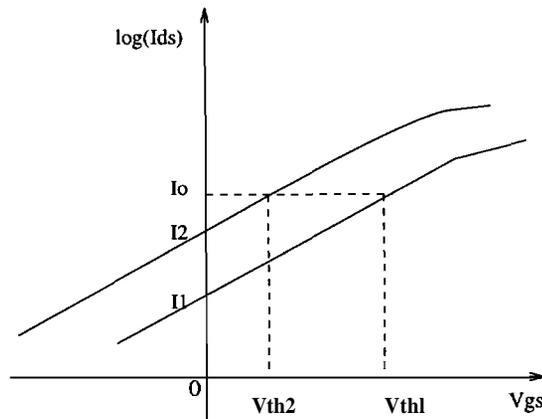


Figure 6: Relationship between leakage current and threshold voltage

in order to balance paths. Figure 6 shows the relationship between leakage current and different threshold voltages. If threshold voltage increases from V_{th2} to V_{th1} , the leakage current at $V_{gs} = 0$ will decrease from I_2 to I_1 . Obviously, leakage currents through those transistors in non-critical paths will be reduced. This makes our method suitable for low voltage low power design.

4 Algorithm to properly choose different threshold transistors

In this section, we first show how to levelize a circuit. Then we define the concept of delay cost. Based on this concept, a heuristic algorithm to properly assign different threshold voltages to different gates is proposed. Since the algorithm traverses each node level by level, its complexity is $O(n)$, where n is the total number of logic gates in a circuit. We also give the pseudo-code of our algorithm.

4.1 Circuit levelizing

For a combinational circuit, the distance of a node from primary inputs can be embodied in its logic level. The level of each primary input is defined to be 0. The level of a node x , denoted as $l(x)$, can be calculated by the following equation:

$$l(x) = 1 + \max_j \{l(j)\} \quad (19)$$

where j is the fan-in node of node x and varies from 1 to n , and n is the total number of fan-in nodes of node x . The following procedure is used to levelize each node of a circuit:

```

Circuit levelizing function levelize()
  For each node  $x$ 
    If (it is not a primary output)
      Set  $l(x) = 0$ 
  For (each primary input  $x$ )
    Add node  $x$  into queue  $Q$ 
  While ( $Q$  not empty) {
    Remove node  $x$  from  $Q$ 
    For each fan-out node  $y$  of node  $x$  {
      If ( $y$  is not a primary output)
        If ( $1 + l(x) > l(y)$ ) {
           $l(y) = 1 + l(x)$ 
          Add node  $y$  into queue  $Q$ 
        }
      }
    }
  }

```

4.2 Algorithm to assign different threshold voltages to different nodes

Section 2 shows that the sum of $\delta(x)$ associated with each node x is the *imbalance* measure of a circuit. Reducing $\delta(x)$ results in a reduction of \mathcal{M}_D . This in turn will reduce power dissipation caused by spurious transitions. This is the basic idea of our heuristic algorithm. Our method works

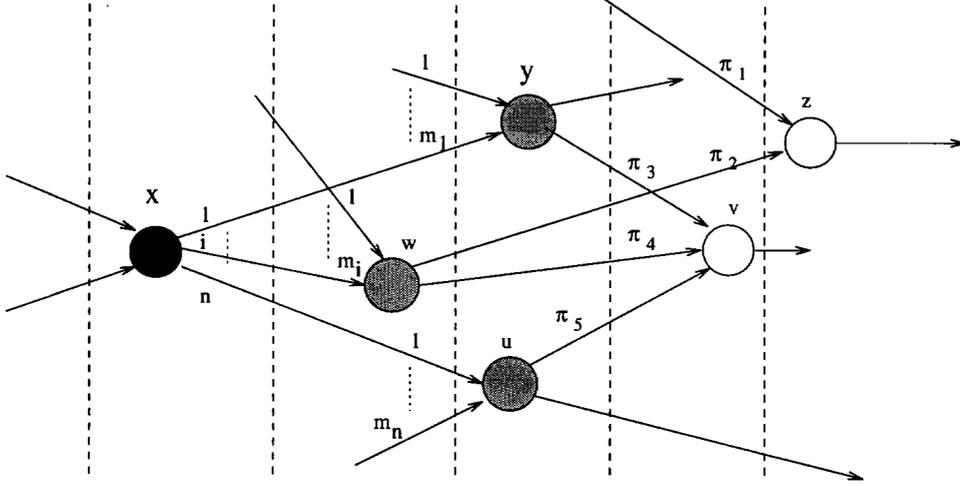


Figure 7: Delay cost

as follows. Starting from the nodes in level one, the algorithm checks each node level by level. If changing the threshold voltage of the gate will reduce the imbalance measure of its fan-out gate(s), the gate will be assigned to a different threshold voltage.

It should be noted that node x may have more than one fan-in path with the same $MIN(x)$. Let us look at Figure 7. Node a and v are the two fan-out gates of node w . The fan-out paths of node w are path π_2 and π_4 . Paths π_1 and π_2 are the fan-in paths of node a . For node v , the fan-in paths are π_3 , π_4 , and π_5 . Let us assume that paths π_4 and π_5 have the same propagation delay $MIN(v)$ which is less than the propagation delay of path π_3 , which is $MAX(v)$. Let us also assume that node w is the node being checked. The heuristic method checks each gate only once. If we change the threshold voltage of node w , the propagation delay difference between path π_3 and path π_4 may become smaller. Since node u is in the level one higher than node w , node u cannot be checked before node w . The propagation delay of path π_5 still remains at the old value $MIN(v)$. Therefore, $\delta(v) = MAX(v) - MIN(v)$ doesn't change. Nevertheless, if we change the threshold voltage of gate w , the next time when node u is checked, only path π_5 will have the value $MIN(v)$ for node v . Node u still may be assigned to a different value. Therefore, $\delta(v)$ will be reduced. According to the definition of glitching power measure in section 2.4, this in turn will reduce glitching power. But if we do not change the threshold voltage of node w , it is impossible for node u to be assigned to a different delay value.

To avoid the case mentioned above, we introduce another concept called delay cost $\mathcal{C}(x)$ for node x . The delay cost is used as a criterion to determine whether a node should be assigned to a different threshold voltage. Again let us look at Figure 7. Suppose node x (black circle) is the node being checked with n fan-out nodes. Each fan-out path i of node x has the same propagation delay value $Pd(i) = MAX(x) + t_p(x)$. Each fan-out node (grey circle) of node x has m_i fan-in paths, each corresponding to propagation delay of $Pd(ij)$. The delay cost $\mathcal{C}(x)$ of node x is defined as the sum of propagation delay differences between one fan-out path of node x and all other fan-in paths

of each fan-out node of node x (such as node y , w , and u),

$$C(x) = \sum_{i=1}^n \sum_{j=1}^{m_i} |Pd(i) - Pd(ij)| \quad (20)$$

For node w , its delay cost $C(w)$ is

$$C(w) = |Pd(\pi_1) - Pd(\pi_2)| + |Pd(\pi_3) - Pd(\pi_4)| + |Pd(\pi_4) - Pd(\pi_5)| \quad (21)$$

where $Pd(\pi_i)$ ($i=1, 2, \dots, 5$) is the propagation delay of path π_i .

Assume that reducing the delay cost of node x leads to reducing $\delta(y)$, where y is the fan-out gate of node x . The following is the pseudo-code of our program. First levelize the circuit using the function *levelize()* provided in section 4.1. Assign an initial threshold voltage to each gate of the simulated circuit. Using the Elmore delay model, obtain the delay of each node. Calculate $MIN(x)$, $MAX(x)$, $\delta(x)$, and *delay cost* $C(x)$ associated with node x , and the propagation delay of each path. Starting from the nodes in level one, the program traverses each node level by level to recalculate all the parameters above. Check the *delay cost*. Assign to the node a threshold voltage which can minimize its *delay cost*. Obviously, the complexity of this method is $O(n)$, where n is the total number of logic gates in a circuit.

Different Threshold Voltages Assignment

levelize()

Initialize each node

 Compute propagation delay of node x corresponding to initial V_{th}

 Level by level, calculate $MAX(x)$, $MIN(x)$, and $\delta(x)$ associated with node x

 Calculate initial *imbalance measure* $\mathcal{M}_{\mathcal{D}}$ and *glitching power measure* $\mathcal{M}_{\mathcal{P}}$

For each node x

Do {

 Modify $MAX(x)$, $MIN(x)$, and $\delta(x)$ associated with node x

 If (fan-out number of node x == 1)

 If (x is not in the only fan-in of its fan-out y)

 Compute delay cost *delay cost* $C(x)$

 Assign to node x the V_{th} which can minimize $C(x)$

 Else

 Compute *delay cost* $C(y)$ based on different V_{th} of node x

 Assign to node x the V_{th} which can minimize $C(y)$

 Else {

 For each fan-out y of node x

 Compute delay cost $C(x)$ based on different V_{th}

 Assign to node x the V_{th} which can minimize $C(x)$

 }

 calculate changed *imbalance measure* $\mathcal{M}'_{\mathcal{D}}$ and *glitching power measure* $\mathcal{M}'_{\mathcal{P}}$

}

5 Discussion about multiple threshold transistors technique

The threshold voltage V_{th} of a MOS device is given by:

$$V_{th} = V_{th0} + \gamma(\sqrt{V_{SB}} - 2\phi_F - \sqrt{2\phi_F}) \quad (22)$$

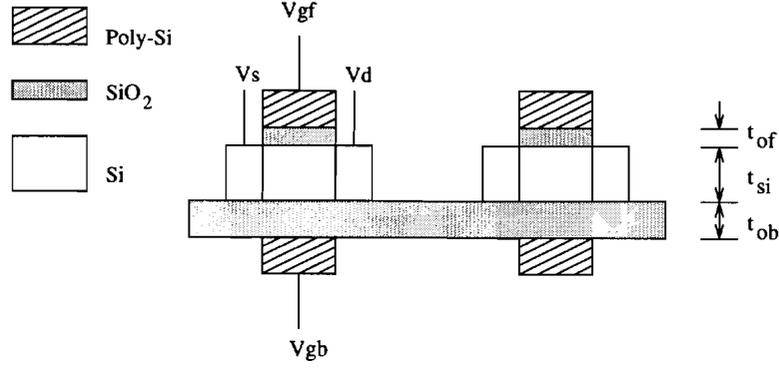


Figure 8: Dual-gated SOI device cross section

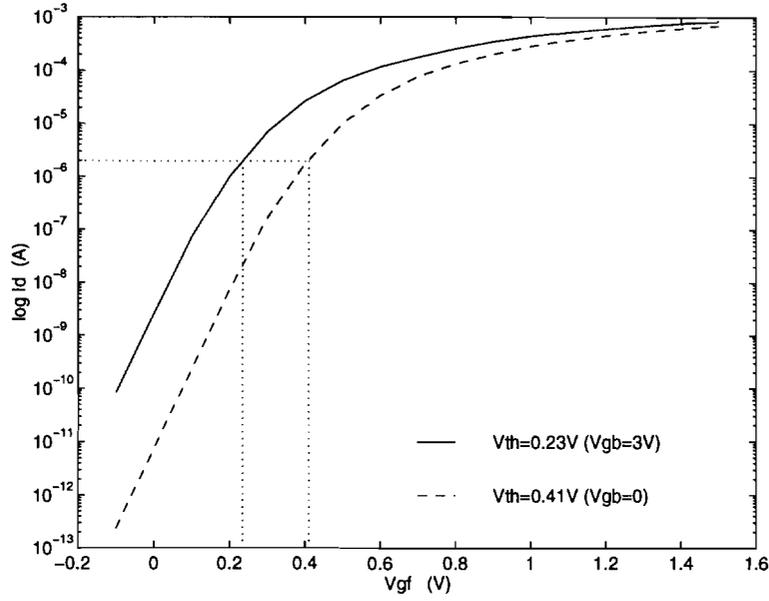


Figure 9: Dual-gated SOI $I_d - V_g$ characteristics for different back gate bias

where.

$$V_{th0} = \phi_{ms} - \frac{Q_{ox}}{C_{ox}} - \frac{Q_I}{C_{ox}} - 2\phi_F - \frac{QB}{C_{ox}} \quad (23)$$

ϕ_{ms} is work-function difference, Q_{ox} is fixed charge sitting at the oxide-silicon interface, Q_I is threshold voltage adjusting implanted impurities, γ is body-effect coefficient, and ϕ_F is Fermi potential. V_{SB} is the voltage difference between source and substrate. Equation 22 and 23 indicate that different threshold voltages can be developed using different Q_I or controlling V_{SB} .

SOI (Silicon On Insulator) technology is an efficient way to realize multiple threshold voltages. An SOI circuit has better performance than a bulk silicon circuit because of the smaller parasitic capacitance and lower leakage current as a result of the steeper subthreshold swing. Moreover, SOI devices are isolated naturally. It is very easy to control the body bias in order to obtain multiple threshold voltages. As for dual-gated SOI devices, since the front gate and the back gate surface potentials are strongly coupled to each other, the front gate threshold voltage can be controlled by

Table 1: Results of $\mathcal{M}_{\mathcal{D}}$ and $\mathcal{M}_{\mathcal{P}}$ for ISCAS benchmarks using two V_{th} with the normalized delay of 1 and 1.5

Circuit Chosen	PI's #	Gate #	Level #	imbalance measure $\mathcal{M}_{\mathcal{D}}$			glitching power measure $\mathcal{M}_{\mathcal{P}}$		
				$\mathcal{M}_{\mathcal{D}}$	$\mathcal{M}'_{\mathcal{D}}$	%	$\mathcal{M}_{\mathcal{P}}$	$\mathcal{M}'_{\mathcal{P}}$	%
C432	36	160	17	249078	195105	21.7	8146150	8113760	0.4
C499	41	202	11	59988	50256	16.2	2095680	2047840	2.3
C880	60	357	23	217992	173079	20.6	4990110	4919575	1.4
C1355	41	514	23	340632	310992	8.7	923160	918296	0.9
C1908	33	880	40	414492	335781	19.0	14654440	13185920	10
C2670	233	1161	32	332328	275967	17.0	8219670	783581	4.7
C3540	50	1667	47	562962	487548	13.4	16335700	15891105	2.7
C5315	178	2290	49	1067484	840435	21.3	34948320	34034205	2.6
C6288	32	2416	124	3261486	2628696	19.4	83092130	81788690	1.6
C7552	207	3466	43	891168	824379	7.5	28583600	27172165	4.9

back bias [10]. Figure 8 shows the dual-gated SOI device structure. V_{gf} and V_{gb} are the front and back gate bias, while t_{of} , t_{Si} , and t_{ob} are the front gate oxide thickness, silicon film thickness, and back gate oxide thickness, respectively.

We simulated a dual-gated SOI NMOSFET using SOI-SPICE. Results are shown in Figure 9 where W/L is 10/0.5, t_{Si} is 500Å, t_{of} is 70Å, t_{ob} is 1000Å, and V_{ds} is 0.5V. By changing the back-gate bias of the dual-gated SOI device from 3V to 0, its threshold voltage will increase from 0.23V to 0.41V. Therefore, using dual-gated SOI technology, multiple threshold transistors technique can be implemented easily.

6 Experimental results

The method to reduce both glitching power and leakage power using multiple threshold transistors has been implemented in C under the Berkeley SIS environment. In this section we will analyze the results of the multiple threshold transistors technique. In this paper we only show the results of using two and three different threshold voltages.

Section 4.2 shows that our algorithm first assigns each node the same initial threshold voltage. The delay of a gate corresponding to such an initial threshold voltage is $t_{p,ini}$. After being checked, if a node is assigned to a different threshold voltage, its delay will be changed to $t_{p,dif}$. The normalized delay is obtained as $t_{p,dif}/t_{p,ini}$. Table 1 shows the results of using two threshold voltages with normalized propagation delay of 1 and 1.5. Results indicate that the imbalance measure $\mathcal{M}_{\mathcal{D}}$ can be reduced by more than 20% for some circuits, and the glitching power measure $\mathcal{M}_{\mathcal{P}}$ decreases by less than 10%. Table 2 shows the results corresponding to normalized delay 1 and 2. In this case, reduction of *glitching* power measure can be improved almost by a factor of two for most circuits compared with the former case. However, this varies for different circuits.

Table 3 lists the results of using three threshold voltages with normalized delay of 1, 1.3, and 1.7. Results indicate that the imbalance measure $\mathcal{M}_{\mathcal{D}}$ can be reduced by more than 40% for

Table 2: Results of $\mathcal{M}_{\mathcal{D}}$ and $\mathcal{M}_{\mathcal{P}}$ for ISCAS benchmarks using two V_{th} with the normalized delay of 1 and 2.0

Circuit Chosen	PI's #	Gate #	Level #	imbalance measure $\mathcal{M}_{\mathcal{D}}$			glitching power measure $\mathcal{M}_{\mathcal{P}}$		
				$\mathcal{M}_{\mathcal{D}}$	$\mathcal{M}'_{\mathcal{D}}$	%	$\mathcal{M}_{\mathcal{P}}$	$\mathcal{M}'_{\mathcal{P}}$	%
C432	36	160	17	249078	193572	22.3	8146150	8060630	1.0
C499	41	202	11	59988	49632	17.3	2095680	2000000	4.6
C880	60	357	23	217992	170976	21.6	4990110	4866216	2.4
C1355	41	514	23	340632	320556	9.2	9231600	9134320	1.1
C1908	33	880	40	414492	325470	22.7	14654440	11780790	19.6
C2670	233	1161	32	332328	269586	18.9	8219670	7525810	8.4
C3540	50	1667	47	562962	476886	15.3	16335700	15483110	5.2
C5315	178	2290	49	1067484	821676	23.0	34948320	33164170	5.1
C6288	32	2416	124	3261486	2694630	17.4	83871710	83092130	0.9
C7552	207	3466	43	891168	798108	10.4	28583600	26118780	8.6

Table 3: Results of $\mathcal{M}_{\mathcal{D}}$ and $\mathcal{M}_{\mathcal{P}}$ for ISCAS benchmarks using three V_{th} with the normalized delay of 1,1.3, and 1.7

Circuit Chosen	PI's #	Gate #	Level #	imbalance measure $\mathcal{M}_{\mathcal{D}}$			glitching power measure $\mathcal{M}_{\mathcal{P}}$		
				$\mathcal{M}_{\mathcal{D}}$	$\mathcal{M}'_{\mathcal{D}}$	%	$\mathcal{M}_{\mathcal{P}}$	$\mathcal{M}'_{\mathcal{P}}$	%
C432	36	160	17	332104	249426	24.9	6516920	6190044	5.0
C499	41	202	11	79984	58152	27.3	1676544	1477536	11.9
C880	60	357	23	290656	181730	37.5	3992088	31045650	22.2
C1355	41	514	23	454176	388328	14.5	7385280	6913184	6.4
C1908	33	880	40	552656	375038	32.1	11723552	8856872	24.5
C2670	233	1161	32	443104	306728	30.8	6575736	5262410	20.0
C3540	50	1667	47	750616	559040	25.5	13068560	10972942	16.0
C5315	178	2290	49	1423312	836196	41.3	27958656	20225082	27.7
C6288	32	2416	124	4348648	3195080	26.5	66473704	59818892	10.0
C7552	207	3466	43	1188224	830324	30.1	22866880	16471176	28.0

Table 4: Results of $\mathcal{M}_{\mathcal{D}}$ and $\mathcal{M}_{\mathcal{P}}$ for ISCAS benchmarks using three V_{th} with the normalized delay of 1,1.5, and 2.0

Circuit Chosen	PI's #	Gate #	Level #	imbalance measure $\mathcal{M}_{\mathcal{D}}$			glitching power measure $\mathcal{M}_{\mathcal{P}}$		
				$\mathcal{M}_{\mathcal{D}}$	$\mathcal{M}'_{\mathcal{D}}$	%	$\mathcal{M}_{\mathcal{P}}$	$\mathcal{M}'_{\mathcal{P}}$	%
C432	36	160	17	249078	133260	46.5	4887690	4584086	6.2
C499	41	202	11	59988	37672	37.2	1257408	950784	24.4
C880	60	357	23	217992	116656	46.5	2994066	2108546	29.6
C1355	41	514	23	340632	255800	24.9	5538960	4692880	15.3
C1908	33	880	40	414492	254434	38.6	8792664	6336010	27.9
C2670	233	1161	32	332328	203338	38.8	4931802	3729274	24.4
C3540	50	1667	47	562962	383674	31.8	9801420	7796078	20.5
C5315	178	2290	49	1067484	597600	44.0	20968992	14281194	31.9
C6288	32	2416	124	3261486	2045526	37.3	49855274	43445152	12.9
C7552	207	3466	43	891168	571266	35.9	17150160	11715988	31.7

some circuits, and the glitching power measure \mathcal{M}_p decreases by more than 25%. Therefore, a three threshold transistors technique can reduce glitching power much more than a two threshold transistors technique.

We also simulate the circuits using another three threshold voltages corresponding to normalized delay of 1, 1.5, and 2.0. (See Table 4). Results show that the imbalance measure \mathcal{M}_p can be reduced by more than 45% for some circuits, and the glitching power measure \mathcal{M}_p can be reduced by more than 30%.

Our heuristic method in section 4 indicates that a gate in a non-critical path can be assigned to a higher threshold voltage. Obviously, this will reduce the leakage power consumed by the devices in non-critical paths while the performance is maintained by the critical paths. Therefore, the total power dissipation will be reduced further.

7 Conclusion

In this paper we propose a new technique to reduce power dissipation due to spurious transitions and leakage current for low voltage circuits using multiple threshold transistors. A heuristic algorithm to properly choose different threshold transistors is presented. Results for ISCAS benchmark circuits indicate that multiple threshold transistors technique can effectively reduce power dissipation caused by glitches. Using three threshold transistors, the glitching power measure can be reduced by about 30%. Our results show that using three different threshold voltages produces even more improvement than two threshold transistors. Our method will assign higher threshold voltages to the transistors in non-critical paths if they can meet the cost function requirement. Therefore, the leakage power dissipated by those transistors will be reduced. The performance is maintained by the transistors in the critical paths. Hence, this method is very suitable for low voltage low power CMOS circuits.

References

- [1] R. I. Bahar, G. D. Hachel, E. Macii, and F. Somenzi, "A Symbolic Method to Reduce Power Consumption of Circuits Containing False Paths," *IEEE Intl. Conf. of Computer Aided Design*, November 1994, pp. 368-371.
- [2] A.P. Chandrakashan, S. Sheng, and R. Brodersen, "Low Power CMOS Digital Design," *IEEE Trans. on Solid-State Circuits.*, vol. 27, No. 4, April, 1992, pp. 473-483.
- [3] S. Devadas and S. Malik, "A Survey of Optimization Techniques Targeting Low Power VLSI Circuits," *ACM/IEEE 32nd Design Automation Conf.*, 1995, pp. 242-247.
- [4] E. Elmore, "The Transient Response of Damped Linear Networks with Particular Regard to Wideband Amplifiers," *Journal of Applied Physics*, January 1948, pp. 55-63.

- [5] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematu, and J. Yamada, "1-V Power Supply High-Speed Digital Circuit Technology with Multithreshold-Voltage CMOS," *IEEE Journal of Solid-State Circuits*, vol. 30, No. 8, August, 1995, pp. 847-853.
- [6] F.N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," *IEEE Trans. on VLSI Systems*, Dec. 1994, pp. 446-455.
- [7] S. Pullela, N. Menezes, and L. T. Pillage, "Skew and Delay Optimization for Reliable Buffered Clock Trees," *Proc. of IEEE Intl. Conf. on CAD*, 1993, pp. 556-562.
- [8] Jan M. Rabaey, "Digital Integrated Circuits," New Jersey: Prentice-Hall, 1996.
- [9] J. G. Xi and Wayne W. M. Dai, "Buffer Insertion and sizing Under Process Variations for Low Power Clock Distribution," *ACM/IEEE Design Automation Conf.*, 1995, pp. 491-496.
- [10] I. Y. Yang, C. Vieri, A. Chandrakasan, and D. A. Antoniadis, "Back gated CMOS on SOIAS for dynamic threshold voltage control," *IEEE IEDM*, 1995, pp. 887-880.