An Improved Logical Effort Model and Framework Applied to Optimal Sizing of Circuits Operating in Multiple Supply Voltage Regimes

Xue Lin, Yanzhi Wang, Shahin Nazarian, Massoud Pedram
Department of Electrical Engineering, University of Southern California, CA USA
E-mail: {xuelin, yanzhiwa, snazaria, pedram}@usc.edu

Abstract

Digital near-threshold logic circuits have recently been proposed for applications in the ultra-low power end of the design spectrum, where the performance is of secondary importance. However, the characteristics of MOS transistors operating in the near-threshold region are very different from those in the strong-inversion region. This paper first derives the logical effort and parasitic delay values for logic gates in multiple voltage (sub/near/super-threshold) regimes based on the transregional model. The transregional model shows higher accuracy for both sub- and near-threshold regions compared with the subthreshold model. Furthermore, the derived near-threshold logical effort method is subsequently used for delay optimization of circuits operating in both near- and super-threshold regimes. In order to achieve this goal, a joint optimization of transistor sizing and adaptive body biasing is proposed and optimally solved using geometric programming. Experimental results show that our improved logical effort-based optimization framework provides a performance improvement of up to 40.1% over the conventional logical effort method.

Keywords

Sub/near-threshold, logical effort, delay optimization

1. Introduction

Aggressive voltage scaling from the conventional super-threshold region to the sub/near-threshold region has shown effectiveness in reducing power consumption in digital circuits [1][2][3]. It is especially beneficial for applications such as wireless sensor processing and RFID tags where performance is not the primary concern. The operating frequency of sub/near-threshold logic is much lower than that of regular strong-inversion logic ($V_{DD} > V_{th}$) due to the smaller transistor current, which consists mostly of leakage current. Authors of [4][5] derived analytical expressions of the optimum V_{DD} to minimize energy consumption, i.e., the minimum energy point (MEP), and showed that the MEP for CMOS circuits typically occurs in the near-threshold region.

Logical effort based delay calculation relies on the computation of the logical effort and parasitic delay of logic gates [6]. The logical effort and parasitic delay of a logic gate are independent of the transistor sizes in the logic gate. The first step is the derivation of the sizes of NMOS and PMOS transistors for a minimum-size inverter that achieves equal rise and fall times and then using this template inverter to derive the values of the logical effort and parasitic delays of more complex logic gates (e.g., NAND and NOR gates.) However, the extension of the logical effort method to the sub/near-threshold region requires taking into account the different characteristics of MOS transistors in the sub/near-threshold region compared with the strong-inversion region. The authors of [7] extended the logical effort method to the subthreshold region by (i) using circuit simulation to determine the optimal width ratio between NMOS and PMOS transistors for the template inverter and (ii) providing an analytical sizing method of stacked transistors based on the case when the drive currents in all the stacked transistors are equal, which is not the actual worst case.

The conventional MOSFET models are expressed in a piecewise fashion with a breakpoint at or near the threshold voltage V_{th}, separating the super-threshold region where the α-power law model [8] is applied and the subthreshold region where the exponential dependency model [5] is applied. Researchers provided the EKV model [9] that is continuous and continuously differentiable for the near-threshold region. However, this model is difficult to provide back-of-the-envelope insights and is hard to work with analytically. Therefore, we propose a simple empirical model that is an enhancement over [10] for the transistors operating in the sub/near-threshold regime, taking into account the DIBL (drain induced barrier lowering) effect that is phenomenal in short-channel transistors. This model results in an average of 3.9% and 2.1% inaccuracy for NMOS and PMOS transistors, respectively, compared with HSpice simulation results.

Based on the accurate transregional model, we extend the logical effort calculation and optimization framework to the near-threshold regime. Different from [7], we provide an analytical sizing method for the NMOS and PMOS transistors in the template inverter with equal rise and fall times, and validate the sizing results through HSpice simulation. Furthermore, we propose an analytical sizing method for the stacked transistors based on the worst-case rise and fall times, showing higher accuracy than the stack sizing method proposed in [7]. We calculate the logical effort and parasitic delay values of different CMOS logic gates operating in the near-threshold regime.

Many burst-mode applications require high performance for brief time periods between extended sections of low performance operation [11]. Digital circuits supporting such burst-mode applications should work in both near-threshold region and super-threshold region (for brief time periods.) It would be difficult for a logic gate to achieve equal rise and fall times simultaneously in both regions even if certain techniques such as adaptive body biasing [12] are exploited. Therefore, we turn to logic gates with asymmetric rise and fall times. We derive the logical effort and parasitic delay values of any arbitrarily sized gates (possibly with asymmetric rise and fall times) with body biasing in both near-threshold and super-threshold regions. Using the

extension of the logical effort-based optimization framework, we perform a joint optimization of gate sizes and body biasing voltages for circuits so that they can operate robustly with the minimum *weighted delay*. The optimization problem is formulated as a geometric programming problem and therefore can be solved optimally in polynomial time complexity [13]. This delay optimization framework can possibly be applied to size the critical paths in large scale circuits. Experimental results of HSpice simulation using 32nm Predictive Technology Model (PTM) [14] show that the proposed improved logical effort-based optimization framework provides a performance improvement of up to 40.1% over the conventional logical effort method under different load capacitance requirements. To our best knowledge, this is the first work that derives logical effort and parasitic delay values of near-threshold logic gates with and without body biasing.

The rest of this paper is organized as follows. Section 2 presents the proposed enhanced transregional model for the transistors operating in the sub/near-threshold regime. Section 3 provides the proposed template inverter sizing, stack sizing and logical effort calculation methods. Section 4 presents our joint near- and super-threshold delay optimization methodology. Experimental results and conclusion are presented in Section 5 and Section 6, respectively.

2. Transistor model in the sub- and near-threshold regimes

The drain current I_{ds} of NMOS transistors operating in the subthreshold regime obeys an exponential dependency on the gate drive voltage V_{gs} and drain-to-source voltage V_{ds}, given by:

$$I_{ds} = \mu C_{ox} \frac{W}{L} (m-1) v_T^2 \cdot e^{\frac{V_{gs}+\lambda V_{ds}-V_{th}}{m \cdot v_T}} \left(1 - e^{\frac{-V_{ds}}{v_T}}\right), \quad (1)$$

where μ is the mobility, C_{ox} is the oxide capacitance, m is the subthreshold slope factor, λ is the DIBL coefficient, and v_T is the thermal voltage $\frac{kT}{q}$. Given a specific technology node (e.g., the 32 nm PTM), we can rewrite the subthreshold model in Eqn. (1) as:

$$I_{ds} = I_0 W \cdot e^{\frac{V_{gs}+\lambda V_{ds}-V_{th}}{m \cdot v_T}} \left(1 - e^{\frac{-V_{ds}}{v_T}}\right), \quad (2)$$

where I_0 is a technology-dependent parameter.

Figure 1 (green dots) plots the simulated curve of I_{ds} v.s. V_{DD} (we set $V_{gs} = V_{ds} = V_{DD}$) on a semi-logarithmic scale for an NMOS transistor with a threshold voltage V_{th} of approximately 0.35 V. One can observe that the curve is nearly straight for $V_{DD} < V_{th}$, corresponding to the exponential I-V relationship in the subthreshold region. The curve rolls off when $V_{DD} > V_{th}$. We extend the transregional model over [10] that accounts for the DIBL effect and unifies the transistor current model for both sub- and near-threshold regions. The transregional model fits the drain current I_{ds} as

$$I_{ds} = I_0 W \cdot e^{\frac{(V_{gs}+\lambda V_{ds}-V_{th})-\alpha\cdot(V_{gs}+\lambda V_{ds}-V_{th})^2}{m \cdot v_T}} \cdot \left(1 - e^{\frac{-V_{ds}}{v_T}}\right), \quad (3)$$

where α is an empirical fitting parameter. We fit the values of parameters I_0, α, and m from HSpice simulation results.

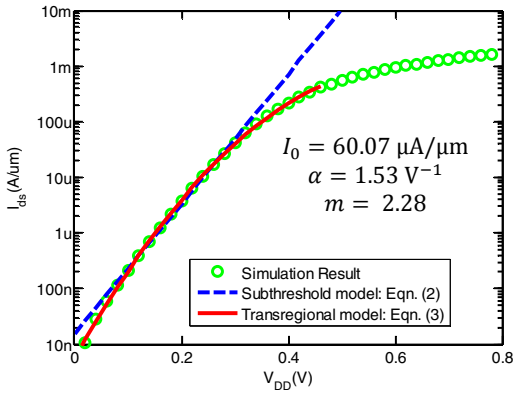

Figure 1. I_{ds} v.s. V_{DD} of an NMOS transistor from simulation, subthreshold model and transregional model.

Over the sub- and near-threshold operation range of 0.10 V to 0.45 V ($V_{th} = 0.35$ V for NMOS transistors), the transregional model (the red curve in Figure 1) results in an average error of 3.9% and a maximum error of 8.0%. However, the subthreshold model (the blue curve in Figure 1) is only valid in the subthreshold region. Over the subthreshold operation range of 0.10 V to 0.3 V, it results in an average error of 8.1% and a worst-case error of 19.0%. We can observe from Figure 1 that the transregional model is even more accurate compared with the subthreshold model in the subthreshold region. A PMOS transistor fits the same empirical model Eqn. (3) with average and worst case errors of 2.1% and 5.8%, respectively.

3. Logical effort of logic gates in the near-threshold regime

Logical effort based delay calculation [6] is a simple and effective way to both estimate and optimize the delay of CMOS circuits. It relies on the computation of the logical effort and parasitic delay values of logic gates. The logical effort and parasitic delay values are independent of the transistor sizes in a logic gate. The logical effort and parasitic delays of logic gates (e.g., NAND, NOR) are derived based on a reference template. The reference template is typically an inverter with minimum size NMOS and PMOS transistors such that the rise and fall times are equalized. More specifically, the gate delay is modeled as $d = ghb + p$, where g is the *logical effort*, h is the *electrical effort*, b is the *branching factor* that accounts for off-path capacitance, and p is the *parasitic delay*. Logical effort is defined as the ratio of the input capacitance of a gate to that of an inverter delivering the same amount of output current (related to its resistance.) The electrical effort represents the ratio of output capacitance to input capacitance; the ghb product is called the stage effort; and the parasitic delay is defined as the delay of a gate driving no load. This final value is set by the parasitic junction capacitance.

3.1 Template inverter sizing

In super-threshold region, the desirable ratio of PMOS width (W_P) to NMOS width (W_N) for achieving equivalent driving currents is approximately 2.7:1 (obtained through HSpice simulations using the 32nm PTM), due to a joint contribution of the mobility difference between charge carriers in PMOS and NMOS devices and the velocity saturation effect [8]. However, the characteristics of MOS transistors in the super-threshold region and in the sub/near-threshold region are significantly different. In the strong-inversion regime, the drain current is a first or second-order function of the MOS terminal voltages, whereas it is given by Eqn. (3) in the sub/near-threshold regime. Hence, the W_P: W_N ratio for an inverter to achieve equal rise and fall times becomes different in the sub/near-threshold region.

Let $V_{gs} = V_{ds} = V_{DD}$, and then we have $e^{\frac{-V_{ds}}{v_T}} = e^{\frac{-V_{DD}}{v_T}} \approx 0$. From Eqn. (3), we derive the following desirable W_P: W_N ratio in the sub- and near-threshold regions:

$$\frac{W_P}{W_N} = \frac{I_{0,N} \cdot e^{\frac{\left(V_{DD}+\lambda_N V_{DD}-V_{th,N}\right)-\alpha_N\cdot\left(V_{DD}+\lambda_N V_{DD}-V_{th,N}\right)^2}{m_N\cdot v_T}}}{I_{0,P} \cdot e^{\frac{\left(V_{DD}+\lambda_P V_{DD}-|V_{th,P}|\right)-\alpha_P\cdot\left(V_{DD}+\lambda_P V_{DD}-|V_{th,P}|\right)^2}{m_P\cdot v_T}}}. \quad (4)$$

Using the 32 nm PTM [14], the derived W_P: W_N ratios under different V_{DD} values for the subthreshold region, the near-threshold region and the super-threshold region are listed in Table 1. We observe from Table 1 that the W_P: W_N ratio is 0.75 for an inverter operating in the near-threshold region (0.3V), which is in the middle between 0.49 for the subthreshold region and 2.67 for the super-threshold region. We also provide the HSpice simulation results of the corresponding W_P: W_N ratios that achieve symmetric rise and fall times. We observe from Table 1 that the analytical results match the simulation results very well, with a maximum deviation of 4.2%.

Table 1. W_P: W_N ratios under different V_{DD} values.

V_{DD} (V)	0.2	0.3	1.0
W_P: W_N ratios (calculated)	0.49	0.75	–
W_P: W_N ratios (simulated)	0.47	0.77	2.67

3.2 Optimal stack sizing

In order to develop the near-threshold logical effort framework, we need to find the stack sizing factor in the near-threshold regime. We use the 2-input NAND gate as an example to derive the stacking factor for NMOS transistors. Please note that a stack of more than two transistors may not be favored in near-threshold circuits because of performance degradation. As shown in Figure 2, the stack sizing factor for NMOS transistors (i.e., ρ) is defined as the ratio of the width of NMOS transistors in NAND gate to the width of NMOS transistor in the template inverter, such that the pull down network in NAND gate has the same current driving strength as the NMOS transistor in the template inverter.

The worst case fall delay of the NAND gate can be obtained when input B is always high and input A makes a transition from low to high. The discharging process can be separated into two phases. During the first phase, V_X drops to a relative low value $V_{X,tar}$, while V_{OUT} remains near to V_{DD}. At the beginning of phase one, I_A is much larger than I_B, due to (1) $V_{gs,B} \approx 0$, (2) $V_{sb,B} > 0$, and (3) almost no DIBL effect for NMOS B. As V_X is decreasing, I_B increases exponentially and I_A decreases slowly. Therefore, before I_B matches I_A, the node X loses more charge than node OUT. In addition, the capacitance at node X is typically smaller than that at node OUT. As a result, V_X drops to a relative low value $V_{X,tar}$ rapidly, while V_{OUT} remains near to V_{DD} during phase one. The second phase starts when I_B matches I_A and V_{OUT} starts to drop rapidly.

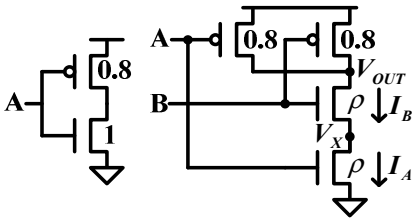

Figure 2. Stack sizing of NMOS transistors.

Then the stacking factor can be calculated from

$$I_0 W \cdot e^{\frac{(V_{DD}+\lambda V_{DD}-V_{th})-\alpha\cdot(V_{DD}+\lambda V_{DD}-V_{th})^2}{m\cdot v_T}}\left(1-e^{\frac{-V_{DD}}{v_T}}\right) = I_0 W\rho \cdot e^{\frac{(V_{DD}+\lambda V_{X,tar}-V_{th})-\alpha\cdot(V_{DD}+\lambda V_{X,tar}-V_{th})^2}{m\cdot v_T}} \cdot\left(1-e^{\frac{-V_{X,tar}}{v_T}}\right) \quad (5)$$

where the left hand side of the equation is the current of the NMOS transistor in the template inverter, and the right hand side of the equation is the current of the pull down network in the NAND gate when I_A matches I_B. We need to derive $V_{X,tar}$ before we can calculate ρ using Eqn. (5). According to the near-threshold current model, we have

$$I_A = I_0 W\rho \cdot e^{\frac{V_{DT,A}-\alpha\cdot V_{DT,A}^2}{m\cdot v_T}}\left(1-e^{\frac{-V_{X,tar}}{v_T}}\right) \approx I_0 W\rho \cdot e^{\frac{V_{DT,A}-\alpha\cdot V_{DT,A}^2}{m\cdot v_T}}, \quad (6)$$

$$I_B = I_0 W\rho \cdot e^{\frac{V_{DT,B}-\alpha\cdot V_{DT,B}^2}{m\cdot v_T}}\left(1-e^{\frac{-(V_{DD}-V_{X,tar})}{v_T}}\right) \approx I_0 W\rho \cdot e^{\frac{V_{DT,B}-\alpha\cdot V_{DT,B}^2}{m\cdot v_T}}, \quad (7)$$

where

$$V_{DT,A} = V_{DD} - \left(V_{th} - \lambda V_{X,tar}\right), \quad (8)$$

$$V_{DT,B} = V_{DD} - V_{X,tar} - \left(V_{th} - \lambda(V_{DD} - V_{X,tar}) + \gamma V_{X,tar}\right). \quad (9)$$

Note that λ is the DIBL coefficient and γ is the body effect coefficient. After the approximations in Eqns. (6) and (7) are made, $I_A = I_B$ implies $V_{DT,A} = V_{DT,B}$. Then $V_{X,tar}$ is obtained as

$$V_{X,tar} = \lambda V_{DD}/(1 + 2\lambda + \gamma). \quad (10)$$

With the calculated $V_{X,tar}$ value, we use Eqn. (5) to calculate the stacking factor.

Table 2 summarizes the stacking factor for NMOS transistors in the 2-input NAND gate using our proposed method, HSpice simulation, and the method in [7] under different near-threshold voltages. It shows that our proposed method is more accurate than the method in [7], because we use a more accurate current model and the DIBL effect is more significant in short channel devices. The stacking factor for PMOS transistors in a 2-input NOR gate can also be characterized with the same method.

Table 2. Stack sizing factor calculation.

V_{DD} (V)	The proposed	HSpice	Reference [7]
0.3	2.28	2.16	2.42
0.4	1.89	2.00	2.85

Using $\rho = 2.2$, we simulate the delay of a 2-input NAND gate by HSpice. The results are shown in Table 3. The rise delay is almost equal to the fall delay for both cases in Table 3. Therefore, the NAND gate can be approximated as a symmetric gate (i.e., the gate with $g_f = g_r$ and $p_f = p_r$.) Also, we derive the logical effort and parasitic delay for the worst case scenario, which means a larger parasitic delay value. For example, in the NAND gate if we do not consider the worst case, the parasitic delay is $p = (0.8 \times 2 + 2.2)/1.8$. However, the worst case parasitic delay should be calculated using Elmore delay method as:

$$p = \frac{0.5R(C_X + C_d) + 0.5RC_d}{1.8RC_g} = 6.0/1.8, \quad (11)$$

where C_X is the capacitance at node X in Figure 2, R is the resistance of NMOS transistor in the inverter in Figure 2.

Table 3. Delay of a 2-input NAND gate.

	Worst case B=1, A changes	A=1, B changes
Rise delay	42.7622 ps	37.1325 ps
Fall delay	42.7125 ps	36.8844 ps

3.3 Logical effort calculation

Based on the results from the previous sections, we derive the logical effort and parasitic delay values for different types of gates operating in the near-threshold, super-threshold and subthreshold regions, respectively. We size the template inverters in different operation regions for equal rise and fall times. Figure 3 compares the logical effort and parasitic delay values of standard logic gates operating in the near-threshold region ($V_{DD} = 0.3$ V), in the strong-inversion region ($V_{DD} = 1.0$ V) and in the subthreshold region ($V_{DD} = 0.2$ V). And we derive the logical effort and parasitic delays for the worst case, similar to (11). The NMOS and PMOS sizes in the near-threshold region for equal rise and fall times are more balanced than those in the subthreshold region or the super-threshold region under the 32nm PTM. However, this conclusion is technology-dependent in general.

3.4 Simulation results

We build a FO4 inverter chain (chain 1) using the inverters sized for near-threshold operation (Figure 3 (a)) and build a FO4 inverter chain (chain 2) using the inverters sized for super-threshold operation (Figure 3 (b)). Then we simulate the delay of the two inverter chains under near-threshold operation ($V_{DD} = 0.3$ V) using HSpice. Table 4 summarizes the normalized delays, showing that delay of the FO4 inverter chain can be reduced by 3.8%~34.0% by using the new logical effort calculation for near-threshold operation.

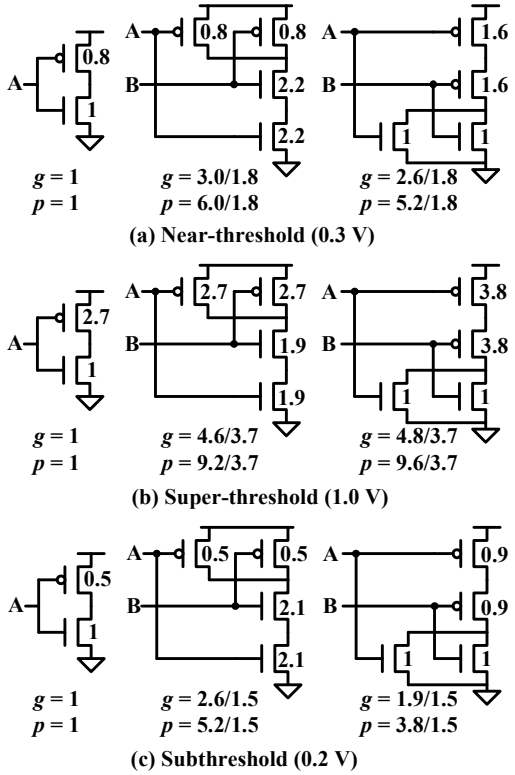

Figure 3. Sizing of standard logic gates.

An n-input AND function can be constructed using the NAND-NOR structure, where 2-input NAND/NOR gates are used to implement the AND function. We build an n-input AND function (AND 1) using NAND and NOR gates sized for near-threshold operation and build an n-input AND function (AND 2) using the NAND and NOR gates sized for super-threshold operation. AND 1 and AND 2 have the same input capacitance and load capacitance, respectively. Then we simulate the delay of them under near-threshold operation using HSpice. Table 5 summarizes the normalized delays, showing that the delay of AND function can be reduced by 18.6%~20.2% using the new logical effort calculation for near-threshold operation.

Table 4. Normalized delay of inverter chains.

Stage Num.	2	3	4	6	8
Chain 1	0.660	0.663	0.715	0.962	0.705
Chain 2	1	1	1	1	1

Table 5. Normalized delay of n-input AND function.

	16-input AND	64-input AND
AND 1	0.798	0.814
AND 2	1	1

4. Logical effort of asymmetric gates and its application

Many burst-mode applications require high performance for brief periods between extended sections of low performance operation [11]. Digital circuits supporting such burst-mode applications should work properly (with acceptable delay) on both near-threshold region and super-threshold region. A gate can hardly achieve equal rise and fall times simultaneously in both regions even if certain techniques such as adaptive body biasing are used. A circuit optimally designed for super-threshold operation may not have optimal delay in the near-threshold region, and vice versa. Therefore, we turn to gates with asymmetric rise and fall times. In this section, we will first derive the logical effort and parasitic delay values of arbitrarily sized (possibly asymmetric) logic gates with body biasing for both near-threshold and super-threshold regions. Next, we will introduce a joint optimization of gate sizing and body biasing on a super buffer so that it can have the minimum *weighted delay* during its operation (both near-threshold and super-threshold.) This framework can be applied to critical path sizing for large scale circuits given the input and load capacitance requirements.

4.1 Logical effort of asymmetric gates

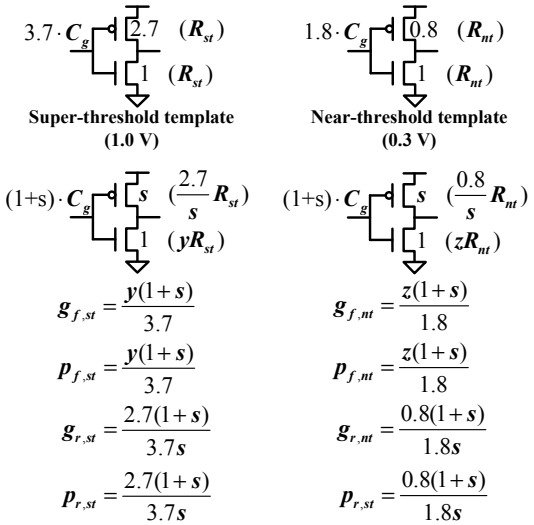

Figure 4. Logical effort and parasitic delay values of an inverter in the super- and near-threshold regions.

Consider an inverter with an NMOS transistor with the size of one and a PMOS transistor with the size of s. We assume that forward or reverse body biasing is only applied to the NMOS transistor in order to reduce the area required for routing extra signals, and of course, the body biasing voltage is the same for all the gates in the same circuit block. Let C_g and C_p denote the gate capacitance and diffusion capacitance of the unit-sized NMOS transistor, respectively. We have $C_g \approx C_p$ for the 32 nm PTM. Let R_{st} and R_{nt} denote the effective resistances of the unit-sized NMOS transistor (i.e., size = 1) in the super-threshold and near-threshold regimes, respectively, when body biasing voltage is not applied. Let $V_{bs,st}$ and $V_{bs,nt}$ denote the applied body biasing voltages on the NMOS transistors operating in the super-threshold and near-threshold regions, respectively. The body biasing voltage values can be positive (forward body biasing) or negative (reverse body biasing). Let $y \cdot R_{st}$ and $z \cdot R_{nt}$ denote the effective resistances of the unit-sized NMOS transistors in the super-threshold and near-threshold regimes, respectively, when the effect of body biasing is considered. The values of y and z are functions of body biasing voltages $V_{bs,st}$ and $V_{bs,nt}$, respectively.

For such an inverter with specific values of s, y and z, we derive its logical effort and parasitic delay values for both super-threshold and near-threshold regions. As shown in Figure 4, we use different template inverters for super-threshold and near-threshold delay calculation. Let $g_{f,st}$ denote the falling logical effort of the inverter in the super-threshold region (with respect to the super-threshold template inverter.) Let $g_{f,nt}$ denote the falling logical effort of the inverter in the near-threshold region (with respect to the near-threshold template inverter.) We calculate $g_{f,st}$ using:

$$g_{f,st} = \frac{yR_{st} \cdot (1+s)C_g}{R_{st} \cdot 3.7C_g} = \frac{y(1+s)}{3.7} \qquad (12)$$

And we have $p_{f,st} \approx g_{f,st}$. Similarly, we can define and calculate the other logical effort and parasitic delay values, as summarized in Figure 4.

4.2 Joint optimization of a super buffer

We use the super buffer as an example to demonstrate our joint optimization framework that can achieve the minimum *weighted delay* for circuits working on both near-threshold and super-threshold regions given the input and load capacitance requirements. Let F denote the portion of cycles when the super buffer operates in the near-threshold region, and thereby, $1 - F$ is the portion of cycles when the super buffer operates in the super-threshold region. The weighted delay of the super buffer is defined by

$$F \cdot \frac{delay_{nt}}{delay_{nt,ref}} + (1-F) \cdot \frac{delay_{st}}{delay_{st,ref}} \qquad (13)$$

where $delay_{nt}$ is the delay of the super buffer in near-threshold operation, and $delay_{st}$ is the delay of the super buffer in the super-threshold operation. In Eqn. (13), we normalize $delay_{nt}$ by $delay_{nt,ref}$, which is the delay of a super buffer specially designed for the optimal delay in near-threshold operation (without body biasing). Similarly, $delay_{st}$ is normalized by $delay_{st,ref}$, which is the delay of a super buffer specially designed for the optimal delay in super-threshold operation (without body biasing). If we define the weighted delay as $F \cdot delay_{nt} + (1-F) \cdot delay_{st}$ instead of Eqn. (13), minimizing the weighted delay of the super buffer is almost equivalent to minimizing the

delay of the super buffer for near-threshold operation only, since $delay_{nt}$ is orders-of-magnitude larger than $delay_{st}$.

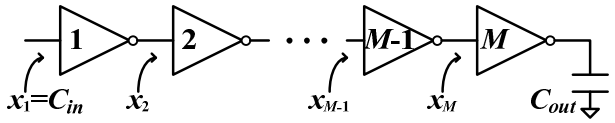

Figure 5. An *M*-stage super buffer.

Figure 5 shows a super buffer with M stages and each inverter has a W_P: W_N ratio of s: 1 (the inverter discussed in Section 4.1.) The input capacitance and output capacitance are given by C_{in} and C_{out}, respectively. Let x_i $(1 \le i \le M)$ denote the input capacitance of the i-th inverter in the super buffer. We have $x_1 = C_{in}$ and $x_{M+1} = C_{out}$. Let $D_{r,st}$ and $D_{f,st}$ denote the rise and fall delays, respectively, of the super buffer in super-threshold operation with respect to the super-threshold template inverter. Let $D_{r,nt}$ and $D_{f,nt}$ denote the rise and fall delays, respectively, of the super buffer in near-threshold operation with respect to the near-threshold template inverter. These values can be calculated as follows based on the logical effort and parasitic delays derived in Section 4.1.

$$D_{r,st} = \frac{y(1+s)}{3.7}\sum_{i=1}^{\frac{M}{2}}\left(\frac{x_{2i}}{x_{2i-1}}+1\right) + \frac{2.7(1+s)}{3.7s}\sum_{i=1}^{M/2}\left(\frac{x_{2i+1}}{x_{2i}}+1\right) \quad (14)$$

$$D_{f,st} = \frac{2.7(1+s)}{3.7s}\sum_{i=1}^{\frac{M}{2}}\left(\frac{x_{2i}}{x_{2i-1}}+1\right) + \frac{y(1+s)}{3.7}\sum_{i=1}^{M/2}\left(\frac{x_{2i+1}}{x_{2i}}+1\right) \quad (15)$$

$$D_{r,nt} = \frac{z(1+s)}{1.8}\sum_{i=1}^{\frac{M}{2}}\left(\frac{x_{2i}}{x_{2i-1}}+1\right) + \frac{0.8(1+s)}{1.8s}\sum_{i=1}^{M/2}\left(\frac{x_{2i+1}}{x_{2i}}+1\right) \quad (16)$$

$$D_{f,nt} = \frac{0.8(1+s)}{1.8s}\sum_{i=1}^{\frac{M}{2}}\left(\frac{x_{2i}}{x_{2i-1}}+1\right) + \frac{z(1+s)}{1.8}\sum_{i=1}^{M/2}\left(\frac{x_{2i+1}}{x_{2i}}+1\right) \quad (17)$$

Then, the joint optimization for the minimum weighted delay is formulated as follows.

Given: F, input capacitance C_{in} and load capacitance C_{out}.

Find: The optimal values of M, x_i's, y, z, and s.

Minimize: The weighted delay

$$F \cdot D_{nt} + (1-F) \cdot D_{st} \quad (18)$$

Subject to the following constraints:

$$D_{r,nt} \le D_{nt} \quad (19)$$

$$D_{f,nt} \le D_{nt} \quad (20)$$

$$D_{r,st} \le D_{st} \quad (21)$$

$$D_{f,st} \le D_{st} \quad (22)$$

The body biasing voltage constraints:

$$y_{min} \le y \le y_{max} \quad (23)$$

$$z_{min} \le z \le z_{max} \quad (24)$$

The balancing constraints for each inverter:

$$BC_{st,min} \le \frac{ys}{2.7} \le BC_{st,max} \quad (25)$$

$$BC_{nt,min} \le \frac{zs}{0.8} \le BC_{nt,max} \quad (26)$$

Note that D_{nt} is the delay of the super buffer in near-threshold operation with respect to the near-threshold template inverter, and D_{st} is the delay of the super buffer in super-threshold operation with respect to the super-threshold template inverter. Therefore, minimizing Eqn. (18) is equivalent to minimizing Eqn. (13). The joint optimization problem is a geometric programming problem if M is given, which can be transformed into a standard convex optimization problem and therefore can be solved optimally with polynomial time complexity [13]. We use an outer loop to determine the optimal M value in the optimization.

This joint optimization framework can also be applied to other circuit structures with little modification. We will show the effectiveness of this framework on both a super buffer structure and an n-input AND function in next Section.

5. Experimental results

We test our joint optimization framework with the super buffer structure. We perform simulations using the 32nm PTM. The supply voltage in super-threshold operation is 1.0 V, and the supply voltage in near-threshold operation is 0.3 V. For different given F, input capacitance C_{in} and load capacitance C_{out} values, we find the optimal values of M, x_i's, y, z, and s using the convex optimization toolbox. Then we simulate the optimal super buffer using HSpice with the optimal values of M, x_i's, y, z, and s. x_i's and s correspond to the transistor sizes in the super buffer, and y and z are translated into body biasing voltages in super-threshold and near-threshold regions, respectively. Note that we use one body biasing voltage for all the NMOS transistors in the super buffer for super-threshold operation and another body biasing voltage for all the NMOS transistors in the super buffer for near-threshold operation. Baseline 1 is a super buffer (without body biasing) designed for the minimum delay in super-threshold operation, with the same given C_{in} and C_{out}. Baseline 2 is a super buffer (without body biasing) designed for the minimum delay in near-threshold operation, with the same given C_{in} and C_{out}. Then we compare the weighted delay of the super buffer optimized with the proposed method, and the weighted delays of Baseline 1 and Baseline 2. The results are summarized in Table 6. The weighted delays of Baseline 1 and Baseline 2 are normalized

by the weighted delay of the proposed optimal super buffer with adaptive body biasing. The proposed optimal super buffer achieves a maximum of 40.1% ($1-1/1.67$) reduction in weighted delay compared to the baselines.

Table 6. The normalized weighted delay values of the optimal super buffer, Baseline 1 and Baseline 2.

Simulation Setup			Weighted Delay		
C_{in}	C_{out}	F	Optimal	Base 1	Base 2
3	10000	0.8	1	1.22	1.38
3	8000	0.8	1	1.29	1.36
3	6000	0.7	1	1.29	1.39
3	4000	0.7	1	1.40	1.32
3	1000	0.7	1	1.58	1.26
3	100	0.7	1	1.67	1.23
3	100	0.3	1	1.38	1.20

We further test our joint optimization framework with the n-input AND function, which has a NAND-NOR structure, and 2-input NAND/NOR gates are used to implement the AND function. For given F, C_{in} and C_{out} values, we find the optimal values of x_i's, y, z, and s using the convex optimization toolbox for the n-input AND function. Note that the number of stages M in the AND function is determined by the input number n, and therefore M is no longer an optimization variable here. Baseline 1 is an n-input AND function designed for the minimum delay in super-threshold operation (without body biasing.) Baseline 2 is an n-input AND function designed for the minimum delay in near-threshold operation (without body biasing.) Then we compare the weighted delay of the AND function optimized with the proposed method with the weighted delays of the baselines. The results are summarized in Table 7. The weighted delays of Baseline 1 and Baseline 2 are normalized by the weighted delay of the proposed optimal AND function with adaptive body biasing. The proposed optimal AND function achieves a maximum of 31.0% ($1-1/1.45$) reduction in weighted delay compared to the baselines.

Table 7. The normalized weighted delay values of the optimal AND function, Baseline 1 and Baseline 2.

	Optimal	Baseline 1	Baseline 2
64-input AND	1	1.45	1.31
256-input AND	1	1.24	1.21

6. Conclusion

This paper presented a new logical effort calculation and optimization framework for digital circuits operating in both near- and super-threshold regions. Modification of the traditional logical effort method is required because the characteristics of MOS transistors operating in the near-threshold region are very different from those in the strong-inversion region. This paper thus introduces: (i) an enhanced analytical transregional transistor model with high accuracy for the sub/near-threshold region, and the application of such model for sizing the template inverter with equal rise and fall times; (ii) an accurate sizing of the transistors in a stack considering the worst case; (iii) logical effort and parasitic delay calculation. We subsequently utilize the derived near-threshold logical effort method for delay optimization for circuits operating in both near- and super-threshold regimes. We perform a joint optimization of transistor sizing and adaptive body biasing for a super buffer and an n-input AND function, utilizing the geometric programming method.

Experimental results on 32nm Predictive Technology Model shows that our improved logical effort-based optimization framework provides a performance improvement of up to 40.1% over the conventional logical effort method.

7. Acknowledgements

This research is sponsored in part by grants from the PERFECT program of the Defense Advanced Research Projects Agency and the Software and Hardware Foundations of the National Science Foundation.

8. References

[1] R. Dreslinski, M. Wiekowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: reclaiming Moore's law through energy efficient integrated circuits," *Proc. of IEEE*, vol. 98, no. 2, pp. 253 - 256, Feb. 2010.

[2] D. Markovic, C. Wang, L. Alarcon, T. Liu, and J. Rabaey, "Ultralow-power design in near-threshold region," *Proc. of IEEE*, vol. 98, no. 2, pp. 237 – 252, Feb. 2010.

[3] A. Wang and A. Chandrakasan, "A 180 mV FFT processor using subthreshold circuit techniques," *IEEE International Solid-State Circuits Conference* (ISSCC), pp. 292 – 293, Feb. 2004.

[4] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "The limit of dynamic voltage scaling and insomniac dynamic voltage scaling," *IEEE Trans. on VLSI*, vol. 13, no. 11, pp. 1239 – 1252, Nov. 2005.

[5] B. H. Calhoun, A. Wang, and A. Chandrakasan, "Modeling and sizing for minimum energy operation in subthreshold circuits," *IEEE J. Solid-State Circuits*, vol. 40, no. 9, pp. 1778 – 1786, Sept. 2005.

[6] I. Sutherland, B. Sproull, and D. Harris, Logical Effort: Designing Fast CMOS Circuits. San Francisco, CA: Morgan Kaufmann, Jan. 1999.

[7] J. Keane, H. Eom, T.-H. Kim, S. Sapatnekar, and C. Kim, "Subthreshold logical effort: a systematic framework for optimal subthreshold device sizing," in *Design Automation Conference* (DAC), 2006.

[8] T. Sakurai and R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584 - 594, April 1990.

[9] C. Enz, F. Krummenacher, and E. Vittoz, "An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications," *Analog Integrated Circuits and Signal Processing*, vol. 8, no. 1, pp. 83 – 114, July 1995.

[10] D. M. Harris, B. Keller, J. Karl, and S. Keller, "A transregional model for near-threshold circuits with application to minimum-energy operation," in *International Conference on Microelectronics* (ICM), 2010.

[11] B. H. Calhoun, A. Wang, N. Verma, and A. Chandrakasan, "Sub-threshold design: the challenges of minimizing circuit energy," *ISLPED*, 2006.

[12] J. W. Tschanz, J. T. Kao, S. G. Narendra, R. Nair, D. A. Antoniadis, A. P. Chandrakasan, and V. De, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE J. Solid-State Circuits*, Nov. 2002.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[14] W. Zhao and Y. Cao, "New generation of Predictive Technology Model for sub-45nm early design exploration," *IEEE Transactions on Electronic Devices*, vol. 53, no. 11, pp. 2816 – 2823, Nov. 2006.