A Schmitt-Trigger and Transistor Sizing based Optimization in Dynamic CMOS Circuits

Kumar Yelamarthi, Central Michigan University, Mt Pleasant, MI 48859

Abstract

One of the significantly used circuit style in high-performance VLSI systems is dynamic CMOS. With its principal advantage of implementing the evaluation logic only in pull down network, it offers a significant performance boost when compared to its static CMOS counterpart. However, the rising magnitude of circuits implemented on a chip, along with shrinking device size and process variations have increased the complexity of implementing dynamic CMOS circuit efficiently. Answering this challenge, this paper proposes an Schmitt-trigger and transistor sizing based optimization for dynamic CMOS circuits. This method operates through i) identifying significance of each timing path, and updating sizes of each transistor in the path; ii) operating the evaluation network and feedback keeper at lower supply voltage to reduce charging and discharging at dynamic nodes; and iii) using a Schmitt trigger to restore the low voltage swing at the dynamic node to normal levels at the design output. When tested through implementation on a IBM 90nm CMOS process, the proposed method has shown an improvement in worst-case delay by 40.64%, delay uncertainty by 49%, delay sensitivity by 28%, power consumption by 36%, and energy-delay-product by 77% when compared to their initial performances.

Introduction

Traditionally, the goal of CMOS circuit designers has been to obtain the best trade-off between delay and power consumption. As CMOS technology continues to scale towards the nanometer regime, millions of transistors are densely packed to increase the system functionality. Such advancements of dense packaging and shrinking device sizes led to the advent of further challenges in power density, leakage power, parametric yield, and process parameter variations. Furthermore, performance degradation of circuits due to leakage, poor noise margins, dense packaging results in design performance variability, and loss of parametric yield [1,2]. The ratio of this performance variability to the nominal values has been increasing with the shrinking device size towards 32nm [3]. While at the same time, process parameter variations have a strong impact on circuit performance-in terms of speed, dynamic power, leakage power, and even possibly causing functional failure [4].

Of the many challenges in the timing optimization of dynamic CMOS circuits, variation in delay (uncertainty) due to process variations is the most prominent due to parameter variations within and across dies. As reported in [5], circuits implemented using 180nm CMOS technology have shown about 30% variation in timing performance, and a 20x variation in leakage current. Along with shrinking CMOS device sizes, the magnitude of these performance variations is predicted to increase significantly [6,7].

Microprocessors and digital signal processors are typically designed using dynamic CMOS circuits due to their speed and compactness in designs with wide fan-in [8]. However, dynamic CMOS circuits suffer from dynamic leakage power and sub-threshold leakage current, limiting the overall design performance. Identifying this challenge, numerous methods were proposed in the recent years. While many of these methods increase system performance in one factor, it is often at the cost of sacrificing others. Transistor sizing is one efficient method that allows designers to move away from the conventional method of minimizing a single factor, instead focusing on optimizing several concurrently. For instance, through transistor sizing: a) several sources of leakage current and power consumption due to glitches can be minimized; b) propagation delay due to device parasitic capacitances can be reduced; and c) sufficient noise margins can be maintained.

Addressing these challenges and based on our previous work [9], this paper presents an Schmitt-trigger and transistor sizing based optimization method for dynamic CMOS circuits while accounting for process variation and parametric yield.

Delay Uncertainty due to Process Variations

Process variations are categorized into two types: die-to-die variations and within-die variations. While within-die variations influence the mean of delay distribution, die-to-die variations influence the variance, also an important factor in performance optimization [10]. Consider the situation presented in Fig. 1, where curve-A denotes the delay distribution of a pre-optimized circuit. When the circuit is optimized only for mean delay, delay distribution curve-B is formed, where a significant portion of the circuits...
operate fast, but with higher leakage current. On the other hand, when the circuit is optimized for only delay variance, delay distribution curve-C is formed, with lower delay variance, but no positive impact on mean delay. As the objective of circuit optimization is reducing both circuit delay and leakage current, accounting for a linear combination of mean and standard deviation ($\mu + \sigma$) improves the overall performance, resulting in curve-D, where $\mu$ is the mean delay, and $\sigma$ is the standard deviation of delay distribution. Alongside, some other parameters to be considered during performance optimization are delay uncertainty ($\Delta = T_{\text{max}} - T_{\text{min}}$) and delay sensitivity ($\delta = \sigma / \mu$), where $T_{\text{min}}$ and $T_{\text{max}}$ are the minimum and maximum delays due to process variations.

While significant research was performed to identify and reduce the impact of process variations, majority either deal with statistical variation, and are not optimal for circuits with large number of process variation, or require these variations to be correlated, which is often not the case in CMOS circuits [11]. With the rising number and magnitude of process variations, along with limited simulation time available, the ideal solution is to pick a set of parameters, find the worst-case path in each case, and take a combination of these paths to obtain performance characteristics ($\mu, \sigma, \Delta, \delta$). Some parameters that are to be considered during this process include junction capacitance, overlap capacitance, gate length and width, threshold voltage, gate oxide thickness, mobility [12].

**Previous Work**

Due to the significance of timing optimization in high performance circuits, numerous methods utilizing transistor sizing have been proposed [11,13-16], but are primarily targeted towards static CMOS and transistors with multiple threshold voltages. One of these prominent works is the Timed Logic Synthesizer (TILOS) [13] that sizes transistors iteratively based on the significance of the timing path, with the cost of larger device size, and higher power consumption. MINFLORTRANSIT as proposed in [14] operates based on iterative generation of directed acyclic graphs in every step of timing optimization. One of the other predominately used transistor sizing method is the Logical Effort [11], requiring computation of every input capacitance, a cumbersome task for circuits with interconnected timing paths, and also this method operates at the cost of increased area [11]. One of the other recently proposed work is the simple exact algorithm [16] that operates by grouping series and parallel transistors for iterative parametric sweep analysis. While this method works for dynamic CMOS circuit, it does not always converge to an optimal solution. Also, the simulation time increases at a quadratic rate in proportion to size of the design. Most importantly, none of these algorithms account for process variations.

On the other hand, numerous methods [5,7,17-20] have been proposed to limit the impact of process variations on CMOS circuits. The Adaptive Body Biasing (ABB) method presented in [5,19] operates by computing bias voltages specific to a die, thus reducing the variation in frequency. However, this method suffers from the limitation of not addressing the intra-die variations, and increased leakage power due to reduction in threshold voltages. While programmable keepers proposed in [20] addresses impact from both intra-die and inter-die variations, its significance is limited by additional keeper hardware, and noise imposed on dynamic node through coupling capacitances [17].

**Schmitt-Trigger Based Pseudo-pMOS Feedback Keeper**

Power consumption, an important metric to evaluate circuit performance is represented as in (1), where $f$ is the clock frequency, $C_L$ is the average switched capacitance per clock cycle, $V_{dd}$ is the supply voltage, $I_{short}$ is the short circuit current, and $I_{leak}$ is the leakage current. While minimizing $V_{dd}$ reduces power consumption, it is typically at the cost of higher worst-case delay ($\tau$) as in (2) where, $V_T$, $\beta$, and $\alpha$ are threshold voltage, gate transconductance, and velocity saturation index respectively.

$$P = fC_LV_{dd}^2 + fI_{short}V_{dd} + I_{leak}V_{dd}$$
$$\tau = C_LV_{dd}/\beta(V_{dd} - V_T)^\alpha$$

Addressing this challenge, Fig. 2 presents a new architecture using an Schmitt trigger based pseudo-pMOS feedback keeper for optimization of delay and power consumption at the same time. This technique operates on two basic principles: (i) the evaluation network and feedback keeper are operated at a reduced supply voltage $\Phi V_{dd}$ ($\Phi$ is a technology dependent factor in the range of 0.6 - 0.8) to lower the amount of charging and discharging of each dynamic node thus reducing power consumption; and (ii)
using a Schmitt trigger to restore the logic levels from $\Phi V_{dd}$ to $V_{dd}$ for a smooth operation with other designs on the system chip. During the pre-charge phase, the dynamic node ‘$D$’ is charged to a logic-high voltage of $\Phi V_{dd}$ rather than $V_{dd}$. Then, during the evaluation phase, voltage at this dynamic node will be either retained at $\Phi V_{dd}$ or decreased to 0 depending on the state of the inputs. When the dynamic node is at its logic-high ($\Phi V_{dd}$), the Schmitt trigger will discharge the gate output ‘$Q$’ to 0, and when the dynamic node is at its logic-low (0), it will charge the gate output ‘$Q$’ to $V_{dd}$. Accordingly, this method will reduce circuit delay and power consumption as charging the dynamic node from 0 to $\Phi V_{dd}$ is much faster and consumes less power than charging it from 0 to $V_{dd}$.

### Pseudo-pMOS Feedback Keeper

As dynamic circuits suffer from charge leakage on the dynamic node ($D$), and the proposed architecture operates on dual supply voltages, an efficient keeper design is necessary to assure optimal design performance. While a simple feedback keeper transistor is easy to implement, the fundamental challenge is choosing the optimal transistor size to obtain a balance between power consumption and signal contention.

Fig. 2 shows a Pseudo-pMOS inverter feedback keeper [21] with a weak CMOS inverter employed to generate the feedback signal. This keeper at the dynamic node-$D$ has at least two advantages: (i) faster feedback response time that is independent of external gate load, and (ii) autonomy to independently optimize feedback keeper. However, the keeper $I$-$V$ characteristics are susceptible to slight changes in transistor parameter variations. As the keeper current quickly drops with the voltage level at the dynamic node-$D$, the most important parameter is the size ratio of pull-up and pull-down transistors ($K_1$ & $K_2$).

#### Schmitt Trigger

Consider that gate output ‘$Q$’ is at high ($V_{dd}$) and the dynamic node ‘$D$’ is at low (0). As transistors $T_1$ and $T_2$ are off and $T_3$ is on, source of $T_3$ will float to $V_{dd}$. $V_{th1}$. As long as $V_D < V_{th1}$, $V_X$ remains at $V_{dd}$. When then voltage at dynamic node ‘$D$’ increases, $T_1$ begins to turn on and the voltage $V_X$ starts to decrease. This high switching point voltage is defined when (3) satisfies or when $T_2$ starts to turn on. When $T_2$ turns on, output ‘$Q$’ will start to reduce towards ground, causing $T_3$ to start turning off. This in turn causes $V_X$ to reduce, turning $T_2$ on even more. This continues until $T_3$ is totally off, $T_1$ and $T_2$ are on. Accordingly, when (3) is a valid, current in $T_1$ and $T_3$ is essentially the same as in eq. (4).

$$V_D = V_{th2} + V_X = \Phi V_{dd}$$

(3)

$$\frac{1}{2} \left( V_{th1} - V_{th2} \right)^2 = \frac{1}{2} \left( V_{dd} - V_X - V_{th3} \right)^2$$

(4)

As the sources of $T_2$ and $T_3$ are connected together, $V_{th2}=V_{th3}$, and the increase in threshold voltages from the body effect is the same for both transistors. Combining (3) and (4) yields (5), where, $S$ and $L$ are the transistor size width and length respectively. With same minimum gate length (5) can be simplified to (6), to find sizes of $T_1$ and $T_3$.

Based on (6) and the general design rule [22], size of $T_3$ and $T_4$ can be found as in (7) and (8) respectively.

$$\frac{\beta_s}{\beta_s} = \frac{S_3}{L_s} \cdot \frac{\Phi V_{dd} - V_{th1}}{V_{dd} - \Phi V_{dd}}$$

(5)

$$\frac{S_1}{S_3} = \left( \frac{V_{dd} - \Phi V_{dd}}{\Phi V_{dd} - V_{th1}} \right)^2$$

(6)

$$S_2 \geq S_1 \ or \ S_3$$

(7)

$$S_4 = \frac{\mu_p}{\mu_n} \times S_2$$

(8)

#### Transistor Sizing

The initial step in keeper optimization is identifying the unique timing discharge paths ($p_i = \{T_i, T_{i+1},...\}$) in the evaluation network. The timing discharge paths in Fig.2 are $p_2 = \{T_5, T_6, T_8\}$ and $p_3 = \{T_7, T_8\}$. Later, each transistor in the design is assigned a weight ($\omega$) in the range of 0.05-0.5 based on its distance from the dynamic output node ‘$D$’.
For example, the circuit in Fig. 2 is comprised of three transistor stacks based on their distance from the output. Stack-1, closest to the dynamic output includes \( T_8 \). Stack-2 includes \( T_6, T_7 \). Stack-3 includes \( T_c \). Accordingly, transistors in these stacks are assigned weights of 0.1, 0.2, and 0.4 respectively. Next, the keeper sizing factor \( M_p \) for each path-\( p \) is found through (9), where \( S_i \) and \( w_i \) are size and weight of each transistor in a timing path. Later, widths of keeper transistors \( K_0, K_1 \) and \( K_2 \) are found using (10) and (11), where, \( f \) is a technology dependent initial sizing factor between 1.1 - 1.5.

\[
M_p = \max(S_i \times w_i, S_{i+1} \times w_{i+1}, \ldots) \tag{9}
\]

\[
K_1 = \sum_{i=1}^{p} M_p \tag{10}
\]

\[
K_2 = K_0 = f \times K_1 \tag{11}
\]

**LBMP for Timing, Power, and Parametric Yield**

Delay, the performance metric of a circuit depends on the size and number of transistors in evaluation network. While increasing size of transistors in one path might reduce its path-delay, it might inadvertently increase the delay of its neighboring path due to capacitive load from the channel-connected transistors. To demonstrate this optimization complexity further, a 2-b Weighted Binary-to-Thermometric Converter (WBTC) as used in high-performance binary adders is presented in Fig. 3, and a Load Balancing of Multiple Paths (LBMP) optimization algorithm is presented in Fig. 4.

Consider two neighboring timing paths: \{\( T_{28}, T_7, T_8, T_{12}, T_{18} \}\} and \{\( T_{28}, T_0, T_4, T_{11}, T_{15}, T_{16} \}\} in Fig. 3., where the first path has a much higher delay compared to the second. To reduce its delay, sizes of all transistors in the path are slightly increased. While doing so reduced delay of the first path, it increased delay of the second path significantly, due to capacitive load from channel-connected transistors. In this case, \{\( T_4, T_{11}, T_{15}, T_{16} \}\} in the second path are channel-connected to \{\( T_7, T_8 \}\}, \{\( T_{15}, T_{16} \}\} in the first path respectively. Increasing sizes of transistors in the first path increased the capacitive load on the second path, and thus its delay. Accordingly, astute care must be taken while choosing the sizing factor of every transistor in a design with multiple channel-connected transistors and timing paths.

As leakage charge and power consumption decreases overall design performance, the first step in the proposed optimization algorithm is inserting the pseudo-pMOS feedback keeper and Schmitt trigger at each of the dynamic output nodes, and sizing their respective transistors per (7) and (8). Later, each transistor in the design is assigned a weight \( w \) in the range of 0.05-0.5 based on its distance from the dynamic output node, in a similar fashion presented in the previous section. Next step in the optimization algorithm is identifying significance of each transistor in the circuit based on the number of timing paths it is present in, and denoting this number as repeats \( r \). Once the repeat and weight profiles are obtained, all transistors widths \( S \) are initially sized with \( S_i = S_{\text{min}} \times f^\psi \) for faster optimization convergence, where \( f \) is a technology dependent initial sizing factor between 1.1 -1.5, and \( \psi \) is the relative position of transistor from output.

Later, process variations are induced in the circuit design with updated transistor sizes to obtain delay distribution of all timing paths. From these delay profiles, all transistors in the top 20% critical paths are clustered to set-\( c \), and the respective widths are updated by (12), where \( r_j \) and \( w_j \) are repeat and weight of transistor-\( j \). Next, to limit the impact of capacitive load from channel-connected transistors, all 1st order connections to transistors in set-\( c \) are identified and clustered to set-\( d \), and their respective widths are updated by (13) where \( r_k \) and \( w_k \) are repeat and weight of transistor-\( k \). Next, sizes of transistors in the pseudo-pMOS feedback keeper are updated using (10) and (11), and the footer size is updated using (14).

\[
S_{\text{new}} = S_{\text{old}}\left[1+\left(r_j w_j / (1+r_j)\right)\right] \tag{12}
\]

\[
S_{\text{new}} = S_{\text{old}}\left[1-\left(r_k w_k / (1+r_k)\right)\right] \tag{13}
\]

Next step in the optimization algorithm is sizing the evaluation network footer transistor, \( T_f \). All the 1st order connections to \( T_f \) are identified, and its size \( S_f \) is updated.
Given: Dynamic CMOS circuit with ‘m’ transistors and ‘n’ timing paths

Inputs: 1) List of timing paths
2) Supply voltage levels (V_{dd}, \Phi V_{dd})

Objective: Minimize path-delay distributions

\[
\max(d_1, d_2, d_3, \ldots, d_d)
\]

subject to: \( S_{\min} \leq S_i \leq S_{\max} \) for \( i = \{1, 2, 3, \ldots, m\} \)

Output: Optimized transistor sizes \( \{S_1, S_2, \ldots, S_d\} \)

---

1: Compute repeat (r_i) and weight (w_j) for all transistors \( i = \{1, 2, 3, \ldots, m\} \)
2: Insert a Pseudo-pMOS feedback keeper and Schmitt trigger at each of the dynamic CMOS output node
3: Change the supply voltage for entire circuit, except the Schmitt trigger, from \( V_{dd} \) to \( \Phi V_{dd} \)
4: Compute the sizes of transistors in the Schmitt trigger for switching of supply levels through

\[
S_i/S_j = [(V_{dd} - \Phi V_{dd})/(\Phi V_{dd} - V_{th})]^2
\]

\( S_i \geq S_j \) or \( S_j \)

\( S_i = (\mu_i/\mu_j) \times S_j \)
5: Initialize all transistor sizes in evaluation network with \( S_j = S_{\min} \times f^{\sigma_j} \)
6: do { 7: induce process variations & obtain path-delay distribution \( \{d_1, d_2, d_3, \ldots, d_d\} \) of all paths 8: sort and rank all paths based on their \( delay(S_i) \) 9: list \( set-c = \{T_1, T_2, T_3, \ldots, T_d\} \) transistors from top 20% paths 10: for \( j \leftarrow 1 \) to \( c \) 11: \( S_j = S_j \left[ 1 + (r_j w_j / (1 + r_j)) \right] \) 12: end for 13: identify \( 1_n \) order connection transistors ‘set-d = \{T_1, T_2, \ldots, T_d\}’ to ‘set-x transistors 14: exclude ‘set-x transistors from ‘set-y’ if any 15: for \( k \leftarrow 1 \) to \( d \) 16: \( S_k = S_k \left[ 1 - (r_k w_k / (1 + r_k)) \right] \) 17: end for 18: for all unaltered transistors, \( S_{new} = S_{old} \)
19: for all dynamic output nodes, update the pseudo-pMOS feedback keeper transistor sizes

\[
M_p = \max(S_i \times w_j, S_{i+1} \times w_{i+1}, \ldots)
\]

\[
K_1 = \sum_{j=1}^n M_p
\]

\[
K_2 = K_3 = f \times K_1
\]
20: update the size \( S_j \) of footer transistor \( T_j \) using

\[
S_j = \sum_{i=1}^n S_i \times w_i
\]
21: } while // delay converges to an acceptable value

---

\[
S_j = \sum_{i=1}^n S_i \times w_i
\]  
(14)

**Performance Optimization from LBMP Algorithm**

Performance optimization of the proposed LBMP optimization algorithm is demonstrated through a 2-b Weighted Binary-to-Thermometric Converter (WBTC) as shown in Fig. 3. With its complex 34 timing paths, and channel loading on neighboring paths, this 2-b WBTC as used in high-performance adders is an ideal circuit to test effectiveness of an optimization algorithm. Through initial simulation, the initial worst-case delay of the circuit was found to 550 ps from path-1. Per the LBMP algorithm presented in Fig. 4, a Schmitt trigger based Pseudo-pMOS feedback keeper was inserted at each of the dynamic CMOS output nodes, and the supply voltage was changed from \( V_{dd} \) to \( \Phi V_{dd} \). Later, based on technology parameters and eq. (6)-(8), sizes of transistors \( \{T_1, T_2, T_3, \ldots, T_d\} \) in the Schmitt Trigger are changed from 120nm (minimum size) to 157 nm, 120 nm, 120 nm, and 480 nm respectively.

Next step in the algorithm is updating the evaluation network transistor sizes. In order to do, transistors in the top 20% critical paths from initial simulations were found, and their initial sizes were increased by a ration \( \delta \). For instance, sizes of transistors \( T_{22}, T_{11}, T_4, T_0 \) in path-1 were updated from 120nm to 120x1.1 = 132nm, 120x1.1^2 = 145nm, 120x1.1^3 = 160nm, 120x1.1^4 = 176nm respectively. Similarly, sizes of transistors in all the remaining 20% critical paths were updated, and process variations were induced to obtain updated delay distribution profiles. In a similar manner, through further updates of transistor sizes per the proposed LBMP optimization algorithm, the worst-case delay of 2-b WBTC reduced to 322 ps, accounting for an improvement of 41.48%.

To demonstrate the effectiveness of proposed LBMP algorithm, further analysis is performed using multiple performance metrics. As reducing std. dev (\( \sigma \)) in delay distribution is equally important compared to mean (\( \mu \)), Fig. 5 shows a comparison of delay distribution of pre- and post-optimized 2-b WBTC with low supply voltage, where the worst-case delay (\( \mu + \sigma \)), and delay uncertainty (\( \Delta \)) reduced from 550 ps to 322 ps, and from 210 ps to 65 ps, an improvement of 41.48% and 70% respectively. Furthermore, the deterministic nature of the LBMP optimization algorithm is demonstrated through Fig. 6, where worst-case delay (\( \mu + \sigma \)), delay uncertainty (\( \Delta \)), and delay sensitivity (\( \delta \)).
have always converged towards an optimum value. As power consumption optimization is equally, if not more important, Fig. 7 shows the optimization efficiency of the proposed algorithm where, the average power reduced from 28.3 µW to 22.02 µW, accounting for an improvement of 22.2%.

Additionally, the proposed LBMP optimization algorithm was applied to several ISCAS benchmark circuits [23], and results shown an average improvement in delay and power consumption by 40.64% and 36.43% respectively as presented in Table 1. In addition to optimization of worst-case delay as in Fig. 8, the algorithm has also demonstrated an optimization in delay uncertainty and delay sensitivity by 49.64% and 28.12% as in Fig. 9 and Fig. 10 respectively. Furthermore, the proposed algorithm has reduced the average power consumption by 36.43% without any penalty in delay as presented in Table 2.

While power-delay-product is a performance metric often used, considering it to assess design performance is questionable as energy can be reduced by operating the designs slowly at a lower supply voltage [11]. The energy-delay product however is less prone to such inconsistencies [24]. Accordingly, the energy-delay products in pre- and post-optimized designs are computed, and normalized data is presented in Fig. 11, to demonstrate an improvement by 77%.

Moreover, parametric yield of the benchmark circuits at different supply voltages has been studied with simulation results (obtained from Cadence Spectre) presented in Table 3. When the supply voltage in the pre-optimized designs was reduced from 1.2 V to 0.8 V (minimum supply voltage per the 90nm CMOS technology library used), the measured parametric yield in the 2-b WBTC quickly reduced towards 54%. However, the same circuits when optimized using LBMP algorithm, has shown an improvement in parametric yield to 100%.
Table 1: Delay improvements in benchmark circuits

<table>
<thead>
<tr>
<th>Design</th>
<th>Pre-opt Delay (ps)</th>
<th>Post-opt Delay (ps)</th>
<th>Delay Improvement (%)</th>
<th>Area Increase (%)</th>
<th>Uncertainty Improvement (%)</th>
<th>Sensitivity Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-b WBTC</td>
<td>550.6</td>
<td>322.2</td>
<td>41.48</td>
<td>60.2</td>
<td>69.12</td>
<td>63.06</td>
</tr>
<tr>
<td>74181-ALU</td>
<td>351.1</td>
<td>207</td>
<td>41.04</td>
<td>53.1</td>
<td>47.72</td>
<td>18.67</td>
</tr>
<tr>
<td>c2670-CLA</td>
<td>369.3</td>
<td>266.4</td>
<td>27.86</td>
<td>58.3</td>
<td>33.56</td>
<td>17.72</td>
</tr>
<tr>
<td>c3540-CC5</td>
<td>247.2</td>
<td>152</td>
<td>38.51</td>
<td>36.2</td>
<td>31.78</td>
<td>14.2</td>
</tr>
<tr>
<td>c3540-CC8</td>
<td>724.3</td>
<td>348.2</td>
<td>51.92</td>
<td>53.8</td>
<td>42.82</td>
<td>17.8</td>
</tr>
<tr>
<td>c3540-CC9</td>
<td>316.1</td>
<td>243.2</td>
<td>23.06</td>
<td>45.1</td>
<td>44.62</td>
<td>12.5</td>
</tr>
<tr>
<td>c5315-CalP2</td>
<td>312.2</td>
<td>201.6</td>
<td>35.42</td>
<td>36.5</td>
<td>37.43</td>
<td>13.5</td>
</tr>
<tr>
<td>c5315-CB4</td>
<td>788.2</td>
<td>303</td>
<td>61.55</td>
<td>31.2</td>
<td>60.68</td>
<td>47.77</td>
</tr>
<tr>
<td>c5315-GLC4_2</td>
<td>476.6</td>
<td>282</td>
<td>40.83</td>
<td>61.1</td>
<td>71.16</td>
<td>52.87</td>
</tr>
<tr>
<td>c5315-CB4</td>
<td>476.6</td>
<td>282</td>
<td>37.5</td>
<td>66.3</td>
<td>46.55</td>
<td>17.92</td>
</tr>
<tr>
<td>c5315-GLC17</td>
<td>228.3</td>
<td>119</td>
<td>47.87</td>
<td>48.7</td>
<td>57.21</td>
<td>22.19</td>
</tr>
<tr>
<td>c7552-GLC5_1</td>
<td>497.3</td>
<td>282</td>
<td>40.79</td>
<td>60.8</td>
<td>71.16</td>
<td>52.87</td>
</tr>
<tr>
<td>c7552-GLC20</td>
<td>249.2</td>
<td>148</td>
<td>40.6</td>
<td>35.2</td>
<td>31.58</td>
<td>14.6</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td>40.64</td>
<td>49.73</td>
<td>49.64</td>
<td>28.12</td>
</tr>
</tbody>
</table>

Table 2: Power consumption improvements in benchmark circuits

<table>
<thead>
<tr>
<th>Design</th>
<th>Pre-Opt Power (uW)</th>
<th>Post-Opt Power (uW)</th>
<th>Power Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-b WBTC</td>
<td>28.3</td>
<td>22.02</td>
<td>22.19</td>
</tr>
<tr>
<td>74181-ALU</td>
<td>16.7</td>
<td>10.52</td>
<td>37</td>
</tr>
<tr>
<td>c2670-CLA</td>
<td>17.3</td>
<td>8.63</td>
<td>50.11</td>
</tr>
<tr>
<td>c3540-CC5</td>
<td>3.04</td>
<td>1.91</td>
<td>37.17</td>
</tr>
<tr>
<td>c3540-CC8</td>
<td>38.19</td>
<td>24.12</td>
<td>36.84</td>
</tr>
<tr>
<td>c3540-CC9</td>
<td>37.65</td>
<td>25.26</td>
<td>32.9</td>
</tr>
<tr>
<td>c5315-CalP2</td>
<td>7.12</td>
<td>4.88</td>
<td>31.46</td>
</tr>
<tr>
<td>c5315-GLC4_2</td>
<td>5.36</td>
<td>3.06</td>
<td>42.91</td>
</tr>
<tr>
<td>c5315-CB4</td>
<td>5.3</td>
<td>3.02</td>
<td>43.01</td>
</tr>
<tr>
<td>c7552-GLC34_4</td>
<td>9.12</td>
<td>6.77</td>
<td>25.76</td>
</tr>
<tr>
<td>c7552-GLC17</td>
<td>2.99</td>
<td>1.97</td>
<td>34.11</td>
</tr>
<tr>
<td>c7552-GLC5_1</td>
<td>5.3</td>
<td>3.02</td>
<td>43.01</td>
</tr>
<tr>
<td>c7552-GLC20</td>
<td>3.04</td>
<td>1.91</td>
<td>37.17</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td>36.43</td>
</tr>
</tbody>
</table>

Conclusion

In this paper, an LBMP optimization algorithm was proposed for performance improvement in timing, power, and parametric yield. The efficiency of the proposed algorithm was demonstrated through three key features of: (i) proficiently balancing the capacitive load on critical paths from neighboring timing paths; (ii) implementing the updated evaluation network at reduced supply voltage to lower the charging and discharging; and (iii) using a Schmitt trigger to restore the reduced voltage swing to normal levels at the design output.

Past work in timing optimization focuses on reducing delay and effect of process variations by sacrificing power consumption. This work suggests that by employing Schmitt trigger based pseudo-pMOS feedback keeper and operating the evaluation network at a lower supply voltage leads to faster circuits, while at the same time results in energy savings with a moderate increase in circuit area. To demonstrate the efficiency of the proposed algorithm, numerous benchmark circuits were tested to obtain an average improvement in worst-case delay by 40.64%, delay uncertainty by 49%, delay sensitivity by 28%, power consumption by 36%, and energy-delay-product by 77%.
Table 3: Parametric yield optimization at different supply voltages

<table>
<thead>
<tr>
<th>Design</th>
<th>Pre-/Post-Opt Yield (%)</th>
<th>Vdd=1.2</th>
<th>Vdd=1.1</th>
<th>Vdd=1.0</th>
<th>Vdd=0.9</th>
<th>Vdd=0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-b WBTC</td>
<td>100/100</td>
<td>100/100</td>
<td>100/100</td>
<td>98/100</td>
<td>82/100</td>
<td>55/100</td>
</tr>
<tr>
<td>74181-ALU</td>
<td>100/100</td>
<td>100/100</td>
<td>100/100</td>
<td>95/100</td>
<td>86/100</td>
<td>60/100</td>
</tr>
<tr>
<td>c2670-CLA</td>
<td>100/100</td>
<td>100/100</td>
<td>100/100</td>
<td>96/100</td>
<td>65/100</td>
<td>32/100</td>
</tr>
<tr>
<td>c3540-CC5</td>
<td>100/100</td>
<td>100/100</td>
<td>100/100</td>
<td>97/100</td>
<td>70/100</td>
<td>65/100</td>
</tr>
<tr>
<td>c3540-CC8</td>
<td>100/100</td>
<td>100/100</td>
<td>100/100</td>
<td>95/100</td>
<td>78/100</td>
<td>60/100</td>
</tr>
<tr>
<td>c3540-CC9</td>
<td>100/100</td>
<td>100/100</td>
<td>100/100</td>
<td>97/100</td>
<td>81/100</td>
<td>55/100</td>
</tr>
<tr>
<td>c5315-CalP2</td>
<td>100/100</td>
<td>100/100</td>
<td>100/100</td>
<td>97/100</td>
<td>73/100</td>
<td>41/100</td>
</tr>
<tr>
<td>c5315-GLC4_2</td>
<td>100/100</td>
<td>100/100</td>
<td>100/100</td>
<td>94/100</td>
<td>85/100</td>
<td>76/100</td>
</tr>
<tr>
<td>c5315-CB4</td>
<td>100/100</td>
<td>100/100</td>
<td>100/100</td>
<td>90/100</td>
<td>69/100</td>
<td>52/100</td>
</tr>
<tr>
<td>c7552-CGC34_4</td>
<td>100/100</td>
<td>100/100</td>
<td>100/100</td>
<td>92/100</td>
<td>52/100</td>
<td>30/100</td>
</tr>
<tr>
<td>c7552-CGC17</td>
<td>100/100</td>
<td>100/100</td>
<td>100/100</td>
<td>94/100</td>
<td>83/100</td>
<td>69/100</td>
</tr>
<tr>
<td>c7552-GLC5_1</td>
<td>100/100</td>
<td>100/100</td>
<td>100/100</td>
<td>90/100</td>
<td>69/100</td>
<td>52/100</td>
</tr>
<tr>
<td>c7552-CGC20</td>
<td>100/100</td>
<td>100/100</td>
<td>100/100</td>
<td>97/100</td>
<td>76/100</td>
<td>65/100</td>
</tr>
<tr>
<td>Average</td>
<td>100/100</td>
<td>100/100</td>
<td>100/100</td>
<td>95/100</td>
<td>75/100</td>
<td>54/100</td>
</tr>
</tbody>
</table>

Figure 9. Delay uncertainty improvement

Figure 10. Delay sensitivity improvement

Figure 11. Energy-Delay-Product comparison in pre- and post-optimized designs

REFERENCES


KUMAR YELAMARTHI received his Ph.D. degree from Wright State University in 2008. He is currently an Assistant Professor of Electrical Engineering at Central Michigan University, Mt Pleasant, MI. His research interest is in the area of RFID, embedded systems, robotics, integrated circuit optimization, and engineering education. He has served as a technical reviewer for several IEEE/ASME/ASEE international conferences and journals, and has published over 75 articles in archival journals and conference proceedings. He is a member of Tau Beta Pi engineering honor society, and Omicron Delta Kappa national leadership honor society. He may be reached at kumar.yelamarthi@cmich.edu