Research Article

Dynamic CMOS Load Balancing and Path Oriented in Time Optimization Algorithms to Minimize Delay Uncertainties from Process Variations

Kumar Yelamarthi¹ and Chien-In Henry Chen²

¹ School of Engineering and Technology, Central Michigan University, Mt Pleasant, MI 48859, USA

² Department of Electrical Engineering, Wright State University, Dayton, OH 45435, USA

Correspondence should be addressed to Kumar Yelamarthi, kumar.yelamarthi@cmich.edu

Received 2 June 2009; Revised 19 October 2009; Accepted 3 December 2009

Academic Editor: Ethan Farquhar

Copyright © 2010 K. Yelamarthi and C.-I. H. Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The complexity of timing optimization of high-performance circuits has been increasing rapidly in proportion to the shrinking CMOS device size and rising magnitude of process variations. Addressing these significant challenges, this paper presents a timing optimization algorithm for CMOS dynamic logic and a Path Oriented IN Time (POINT) optimization flow for mixed-static-dynamic CMOS logic, where a design is partitioned into static and dynamic circuits. Implemented on a 64-b adder and International Symposium on Circuits and Systems (ISCAS) benchmark circuits, the POINT optimization algorithm has shown an average improvement in delay by 38% and delay uncertainty from process variations by 35% in comparison with a state-of-the-art commercial optimization tool.

1. Introduction

The performance improvement of microprocessors has been driven traditionally by dynamic logic and microarchitectural improvements [1] and can be further enhanced through circuit design and topology organization. Dynamic logic is an effective logic style in terms of timing and area when compared to its static counterpart due to (1) the absence of requirement for design implementation in complementary PMOS logic, and (2) the use of a clock signal in its implementation of combinational logic circuits. In general, CMOS dynamic logic uses fast NMOS transistors in its pull-down network. Its delay is dependent on the number and size (width) of transistors in the NMOS critical path. This paper presents an NMOS transistor sizing optimization for a faster operation.

Static logic is slower because it has twice the loading, higher thresholds, and actually uses slow PMOS transistors for computation. Dynamic logic has been predominantly used in microprocessors, and their usage has increased the timing performance significantly over static CMOS circuits

[1, 2]. However, timing optimization of dynamic logic is challenging due to several issues such as charge sharing, noise-immunity, leakage, and environmental and semiconductor process variations. Also, with dynamic circuits consuming more power over static CMOS, an optimal balance of delay and power can be achieved at the architectural level through effective partitioning of design into a mixed-static-dynamic circuit style [3].

Process variations introduce design uncertainties at each step of process development, design, manufacturing, and test. The ratio of these process variations to the nominal values has been increasing with the shrinking device size towards 32 nm [4], causing an impending requirement to account for process variations during timing optimization. They need to be taken into account during the design phase to make sure that performance analysis provides an accurate estimation [5].

One of the challenges in timing optimization of CMOS logic is delay uncertainty (Δ) from process variations, $\Delta = T_{\max} - T_{\min}$, where $T_{\max}$ and $T_{\min}$ are the maximum and minimum delays of a timing path. In the 180 nm CMOS

technology, these process variations have caused about 30% variation in chip frequency, along with 20X variation in current leakage [6]. The magnitude of intradie channel length variations has been estimated to increase from 35% of total variations in 130 nm to 60% in 70 nm CMOS process and variation in wire width, height, and thickness is also expected to increase from 25% to 35% [7]. In CMOS 65 nm process, the parameters that affect timing the most are device length, threshold voltage, device width, mobility, and oxide thickness [8]. For process variation sensitive circuits such as SRAM arrays and dynamic logic circuits, these process variations may result in functional failure and yield loss [7].

Addressing the challenges of timing optimization and delay uncertainty from process variations, this paper presents a timing optimization algorithm for dynamic circuits, and a timing optimization flow for mixed-static-dynamic CMOS logic. The proposed algorithm and flow are validated through implementation on several benchmark circuits and a 64-b binary adder in 130 nm CMOS process. This paper is an extension of our previous work [9] and is organized as follows. Section 2 presents previous work on transistor sizing (width) optimization for timing and process variation minimization. Section 3 presents the proposed transistor sizing optimization for CMOS dynamic logic. Section 4 presents implementation and results of several ISCAS benchmark circuits. Based on this timing optimization algorithm for dynamic circuits, Section 5 presents a timing optimization flow for mixed-static-dynamic CMOS logic and its implementation and results of several ISCAS benchmark circuits. Finally, conclusion is presented in Section 6.

2. Previous Work

Methods of automating transistor sizing for timing optimization were proposed in [3, 10–16], but many of them focus on static CMOS circuits and technologies using multiple threshold voltages. TImed LOgic Synthesizer (TILOS) [10] presented an algorithm of iteratively sizing transistors by a certain factor in the critical path. The algorithm is not a deterministic approach, as it does not guarantee convergence in timing optimization. MINFLOTTRANSIT [11] is another algorithm proposed for transistor sizing based on iterative relaxation method but requires iterative generation of directed acyclic graphs for every step of timing optimization. Computation of “Logical Effort” is the other method proposed for timing optimization [16]. However, it has two limitations. First, it requires estimation of input capacitance, of which circuits with complex branches or multiple paths have difficulty in accurate estimate. Second, it optimizes timing at the cost of increased area [17].

Methods to mitigate the effect of process variations in CMOS circuits were proposed in [6, 7, 18–23]. These methods deal with statistical variations and are not optimal for designs with large number of parameter variations [24]. A technique called Adaptive Body Biasing (ABB) was presented in [6, 22] to compensate for variation tolerance. The ABB technique is implemented in postsilicon where each die receives a unique bias voltage, reducing the variance of frequency variation. However, this method does not minimize

intradie variations, as each block in the design requires a unique bias voltage. Another limitation is the increasing leakage power, caused by the reduction of threshold voltage. Programmable keepers were proposed to compensate for process variations in [23]. This method works for designs with large number of parallel stacks (similar to the NOR gates). However, it requires additional hardware to program the keeper transistors for other designs.

Research has shown that intradie variations primarily impact the mean delay, and interdie variations impact the variance of delay [18]. As timing optimization should consider both interdie and intradie variations, both mean and variance should be accounted for. In addition to optimizing path delay, other parameters affected by process variations that need to be considered and reduced are delay uncertainty ($\Delta = T_{\max} - T_{\min}$) and sensitivity ($\delta = \sigma/\mu$), where $T_{\max}$ and $T_{\min}$ are the maximum and minimum delays, μ is the mean delay, and σ is the standard deviation of delay distribution.

3. Transistor Sizing Optimization of Dynamic Circuits

The delay of dynamic circuit is highly dependent on the number and size (width) of transistors in the critical path. Increasing width of transistors in a path will increase the discharging current and reduce the output pull-down path delay. However, increasing width of transistors to reduce one path delay may increase the capacitive load of channel-connected transistors on other paths and substantially increase their delays. This complexity increases along with the number of paths present in the circuit. A 2-b Weighted Binary-to-Thermometric Converter (WBTC) that is used in high-performance binary adders [25] shown in Figure 1 is used as an example to explain the path delay optimization complexity while considering process variations.

Figure 1 highlights two timing paths: path-A ($T_{28}-T_7-T_8-T_{12}-T_{18}-T_{32}$) and path-B ($T_{28}-T_0-T_4-T_{11}-T_{15}-T_{16}-T_{31}$). A test was performed to optimize path-A by gradually increasing widths of T_7 , T_8 , T_{12} , and T_{18} . It was observed that the delay of path-A reduced by 4%, but delay of path-B increased by 9.3%. This is a result of transistors on path-B being channel-connected to transistors on path-A. For instance, T_4 and T_{11} are channel-connected to T_7 and T_8 , and T_{15} and T_{16} are channel-connected to T_{12} and T_{18} . Increasing widths of T_7 , T_8 , T_{12} , and T_{18} in path-A causes the capacitive load of T_4 , T_{11} , T_{15} , and T_{16} to increase and therefore increase delay of path-B. This circuit example illustrates that increasing widths of transistors on one critical path increases capacitive load and delay of the other critical paths.

Conventionally, a path delay is denoted by the mean (μ) of its delay distribution, which accounts only for intradie variations. As interdie variations are equally important, its standard deviation (σ) is as important and should be considered as well. Consider the delay distribution of two paths of WBTC shown in Figure 2. Path-B has a high mean delay, while path-A has a high standard deviation. Typically, path-B would be chosen as the critical path for timing optimization as it has the highest mean delay (μ). Optimizing the design by increasing width of transistors on path-B may

FIGURE 1: 2-b Weighted Binary to Thermometric Converter.

FIGURE 2: Delay distribution of two paths in 2-b WBTC.

reduce the mean delay (μ) but may not reduce its standard deviation (σ) as well. However, by considering both the mean and the standard deviation (i.e., $\mu + \sigma$), path-A would be chosen as the critical path to be optimized. As both interdie and intradie variations are equally important, the proposed timing optimization algorithm ranks critical paths based on the sum of the path mean delay and its standard deviation, ($\mu + \sigma$). The Load Balance of Multiple Paths (LBMPs) timing optimization algorithm proposed for transistor sizing of dynamic circuits while considering process variations is presented in Figure 3.

Consider the circuit with a series of n channel-connected NMOS transistors, $T_1, T_2, \dots, T_n$, in Figure 4. While T_n conducts the discharge current of the load capacitance C_L , T_1 conducts the discharge current from a total capacitive load, $C_{\text{total}} = C_L + \dots + C_3 + C_2 + C_1$, which is substantially large when n increases. So, the discharge time of T_1 , the transistor near Gnd, is longer than T_n , the transistor near output. Accordingly, accounting for the discharge times increasing from T_n to T_1 , the transistor sizes are made progressively larger, starting from a minimum-size transistor at T_n to reduce the total discharge time of the pull-down path (out, $T_n, \dots, T_2, T_1$). The width of the next to the last transistor is scaled up by a factor. In the proposed LBMP algorithm we assign a *weight* (used for transistor sizing) in the range of 0.05–0.5 to each transistor relative to its distance from the output for this reason. For instance, the 2-b WBTC in Figure 1 is comprised of seven transistor stacks relative to their distance from the output. Stack-1, closest to the output, includes transistors $T_3, T_{10}, T_{16}, T_{21}, T_{25}$, and T_{27} . Stack-2 includes transistors $T_6, T_{13}, T_{18}, T_{23}$, and T_{26} . Stack-3 includes transistors T_2, T_9, T_{15}, T_{20} , and T_{24} . Stack-4 includes transistors T_5, T_{12}, T_{17} , and T_{22} . Stack-5 includes transistors T_1, T_8, T_{14} , and T_{19} . Stack-6 includes transistors T_4 and T_{11} . Stack-7 farthest from the output includes transistors T_0 and T_7 . Accordingly, transistors in stacks 1–7 are assigned *weights* of 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, and 0.5, respectively. For designs with different number of stacks, *weights* of transistors

FIGURE 3: LBMP Transistor sizing algorithm.

in stacks are evenly distributed in the range 0.05–0.5 relative to its distance from the output; *weight* of 0.05 is assigned to stack of transistors closest to the output, and a *weight* of 0.5 is assigned to stack of transistors farthest from the output.

As increasing the width of transistor that appears in the most number of paths reduces overall delay, the number of paths a transistor is present in is computed and denoted by “*repeats*”. The initial step in LBMP algorithm is to size transistors on every path with a fixed ratio, for example, 1.1 for faster optimization convergence [26]. After the *repeats* and the *weights* of all transistors are computed, simulations are performed to obtain delay distribution of each path. The transistors on the top 20% critical paths are grouped to *set-x*, and their widths are increased and calculated by (1):

$$\text{New_Size} = \text{Old_Size} \left(1 + \left(\frac{\text{Repeats}}{1 + \text{Repeats}} \right) \times \text{Weight} \right). \quad (1)$$

FIGURE 4: Transistor sizes are made progressively larger.

As the delay of critical path is dependent on the capacitive load of channel-connected transistors, reducing this capacitive load reduces the overall delay. The 1st-order connection transistors in *set-x* are identified and grouped to *set-y*. Then, transistors in *set-x* are excluded from *set-y* to form *set-z*. For each transistor in *set-z*, it is checked if the transistor is present in *set-x* of previous iteration. If so, its width is decreased and calculated by (2) and (3). If not, its width is decreased and calculated by (4). Once new transistor widths are determined, simulations are performed to locate the new critical paths. This algorithm is repeated until a convergence of an optimal solution is obtained:

$$\text{TempNew} = \text{Old_Size} \left(1 - \left(\frac{\text{Repeats}}{1 + \text{Repeats}} \right) \times \text{Weight} \right), \quad (2)$$

$$\text{New_Size} = \frac{\text{Old_Size} + \text{TempNew}}{2}, \quad (3)$$

$$\text{New_Size} = \text{Old_Size} \left(1 - \left(\frac{\text{Repeats}}{1 + \text{Repeats}} \right) \times \text{Weight} \right). \quad (4)$$

4. Optimization of Delay, Uncertainty, and Sensitivity from Process Variations

A 2-b Weighted Binary-to-Thermometric Converter (WBTC) used in high-performance binary adders was shown in Figure 1 [25]. This circuit is used as an example to illustrate the complexity of transistor sizing optimization. With less than 50 transistors, the 2-b WBTC has 34 timing paths, and of which path delays change dramatically with different transistor sizes.

The timing paths of the 2-b WBTC are shown in Table 1, and transistor *repeat* and *weight* profiles are shown in

FIGURE 5: Delay Distribution of 2-b WBTC before Optimization.

FIGURE 6: Delay Distribution of 2-b WBTC after Optimization.

Table 2. Prior to optimization, the worst-case delay of 2-b WBTC was 355 psec from path-1. The top 20% critical paths are path-1, 2, 5, 8, 26, and 29. Widths of all transistors in these critical paths are initially increased by a ratio of 1.1 to their initial values. For example, the sizes of transistors (T_{22} , T_{11} , T_4 , and T_0) in path-1 are increased to $160 \times 1.1 = 176$ nm, $160 \times 1.1^2 = 193$ nm, $160 \times 1.1^3 = 213$ nm, $160 \times 1.1^4 = 234$ nm, respectively. After initial transistor sizing, process variations are considered in simulations in which delay distribution of each path is obtained. Then, transistor sizes are updated using (1)–(4), and simulations are performed to obtain a new critical path order. After several iterations of the LBMP timing optimization algorithm, the worst-case delay of 2-b WBTC is reduced and finally converged to 157 psec, accounting for a 55.77% improvement.

Efficiency of the LBMP algorithm is further illustrated through reduction in delay uncertainty (Δ). Figures 5 and 6 show the normalized delay distribution of the 2-b WBTC before and after optimization, respectively. It is clearly evident that delay uncertainty has reduced and distribution has been narrowed significantly in the optimized design.

FIGURE 7: Delay Impact from Variation in Junction Capacitance.

FIGURE 8: Delay Impact from Variation in Oxide Thickness.

With major contributors towards delay uncertainty being gate length, channel width, capacitance, supply voltage, and threshold voltage [5], timing analysis was performed to categorize the impact of each. Figure 7 shows the reduction in delay uncertainty of 2-b WBTC from 14% to 8% due to variation in zero-bias junction capacitance.

Kinget in his work on device mismatch [27] has shown that variance in delay distribution of ΔV_T is dependent on device area, $W \times L$ as shown in (5), where W is the transistor width, L is the transistor channel length, and A_{VT} is the proportionality constant (a technology dependant value). Experimental results of delay impact from variation in oxide thickness are shown in Figure 8 where increasing device area after transistor sizing reduces the delay uncertainty of the 2-b WBTC from 24% to 15%. The other research [5] shows that a drop in supply voltage degrades cell timing at a quadratic rate; a 5% drop in total rail-to-rail voltage may result in a 15% timing degradation. Figure 9 shows the delay uncertainty of the 2-b WBTC before and after optimization using the LBMP algorithm. It is observed that a 20% drop in total rail-to-rail voltage (from 1.0 V to 0.8 V) results in a 4% variation in timing (much less than 15%), which

TABLE 1: Timing Paths in 2-b WBTC.

Path No.	Transistors	Path No.	Transistors
1	$T_{28}, T_0, T_4, T_{11}, T_{22}, T_{26}$	18	$T_{28}, T_7, T_8, T_{12}, T_{18}$
2	$T_{28}, T_7, T_{11}, T_{22}, T_{26}$	19	$T_{28}, T_7, T_{11}, T_{15}, T_{18}$
3	$T_{28}, T_{19}, T_{22}, T_{26}$	20	$T_{28}, T_7, T_{11}, T_{17}, T_{21}$
4	T_{28}, T_{24}, T_{26}	21	$T_{28}, T_7, T_{11}, T_{20}, T_{21}$
5	$T_{28}, T_0, T_4, T_{11}, T_{17}, T_{23}$	22	$T_{28}, T_{14}, T_{15}, T_{18}$
6	$T_{28}, T_0, T_4, T_{11}, T_{20}, T_{23}$	23	$T_{28}, T_{14}, T_{17}, T_{21}$
7	$T_{28}, T_0, T_4, T_{11}, T_{22}, T_{25}$	24	$T_{28}, T_{19}, T_{20}, T_{21}$
8	$T_{28}, T_7, T_{11}, T_{17}, T_{23}$	25	$T_{28}, T_0, T_1, T_5, T_{13}$
9	$T_{28}, T_7, T_{11}, T_{20}, T_{23}$	26	$T_{28}, T_0, T_4, T_{11}, T_{15}, T_{16}$
10	$T_{28}, T_7, T_{11}, T_{22}, T_{25}$	27	$T_{28}, T_7, T_8, T_9, T_{13}$
11	$T_{28}, T_{14}, T_{17}, T_{23}$	28	$T_{28}, T_7, T_8, T_{12}, T_{16}$
12	$T_{28}, T_{19}, T_{20}, T_{23}$	29	$T_{28}, T_7, T_{11}, T_{15}, T_{16}$
13	$T_{28}, T_{19}, T_{22}, T_{25}$	30	$T_{28}, T_{14}, T_{15}, T_{16}$
14	T_{28}, T_{24}, T_{25}	31	$T_{28}, T_0, T_1, T_2, T_6$
15	$T_{28}, T_0, T_4, T_{11}, T_{15}, T_{18}$	32	$T_{28}, T_0, T_1, T_5, T_{10}$
16	$T_{28}, T_0, T_4, T_{11}, T_{17}, T_{21}$	33	$T_{28}, T_7, T_8, T_9, T_{10}$
17	$T_{28}, T_0, T_4, T_{11}, T_{20}, T_{21}$	34	$T_{28}, T_0, T_1, T_2, T_3$

TABLE 2: Repeat and Weight Profiles of 2-b WBTC.

Repeats	Near GND				Near Vdd		
16		T_{11}					
12	T_0, T_7						
8		T_4					
6				T_{17}, T_{22}	T_{15}, T_{20}	T_{23}	T_{21}
4			T_1, T_8, T_{14}, T_{19}			T_{18}, T_{26}	T_{16}, T_{25}
2				T_5, T_{12}	T_2, T_9, T_{24}	T_{13}	T_{10}
1						T_6	T_3, T_{27}
Weight	0.5	0.4	0.3	0.2	0.15	0.1	0.05

FIGURE 9: Delay Uncertainty from Variation in Supply Voltage.

further illustrates that the LBMP algorithm is less sensitive to variation in supply voltage:

$$\sigma^2(\Delta V_T) = \frac{A_{VT}^2}{W \cdot L}. \quad (5)$$

Another benchmark used to validate the algorithm is a 4-b Unity Weight BTC (UWBTC) that is used in high-performance digital-to-analog converters, as shown in Figure 10. Along with an increase in the number of transistors, the number of timing paths to be considered is also increased to 83. Prior to optimization, the 4-b UWBTC had a worst-case delay of 152 psec. Through iterative optimization using the LBMP algorithm, the worst-case delay of 4-b UWBTC was reduced to 103 psec, an improvement of 33%. Furthermore, the LBMP algorithm was also implemented on several ISCAS benchmark circuits of which the ratio of the number of critical paths to the number of transistors is shown in Table 3. Through implementation and verification in 130 nm CMOS process, in Table 4 the LBMP algorithm has shown an average delay reduction by 47.8%, uncertainty reduction by 48%, power increase by 13%, and an area increase by 39.8%. The delay convergence profiles of these circuits are shown in Figure 11.

As delay in general can be reduced by increasing power consumption [28], power-delay product (PDP) is a key evaluation parameter to compare the design performance among different circuit structures. Table 5 shows the PDP of benchmark circuits before and after optimization. Through

FIGURE 10: 4-bit Unity Weight BTC.

TABLE 3: Characteristics of Benchmark Circuits.

Design	# Inputs	# Outputs	# Paths	# Transistors
CCT-2	8	6	6	7
2-b WBTC	4	6	34	28
4-b UWBTC	4	15	83	83
74181-CLA	10	6	18	24
74181-E Mod	8	6	6	7
C2670-CLA	24	1	15	39
C3540-CC5	7	1	4	7
C3540-CC8	7	3	17	35
C3540-CC9	8	3	22	47
C3540-UM12-7	9	1	24	50
C5315-CalP2	6	1	7	14
C5315-GLC4.2	8	1	5	8
C5315-CB4	9	1	5	9
C7552-GLC5.1	8	1	4	9
C7552-CGC34.4	9	4	14	9
C7552-CGC17	6	1	3	5
C7552-CGC20	7	1	4	7

TABLE 4: Optimization results from the LBMP Algorithm.

Design	Initial delay (psec)	Final delay (psec)	Delay reduction (%)	Uncertainty reduction (%)	Power increase (%)	Area increase (%)
CCT-2	226	109	52	46.8	8.6	53.1
2-b WBTC	355	157	55	48.6	2.3	65.5
4-b UWBTC	152	103	33	63.6	14.0	12.6
74181-CLA	209	103	51	46.8	22.3	53.1
74181-E Mod	225	110	51	47.3	9.4	54.2
C2670-CLA	391	206	47	52.7	14.4	58.5
C3540-CC5	144	77	46	61.6	13.2	43.3
C3540-CC8	427	216	50	53.5	14.2	62.1
C3540-CC9	341	202	41	64.3	18.4	32.1
C3540-UM12-7	485	178	63	43.7	n/a	33.4
C5315-CalP2	230	137	39	40.0	2.7	9.2
C5315-GLC4.2	197	122	38	40.4	18.4	12.2
C5315-CB4	243	134	45	48.3	15.2	32.3
C7552-GLC5.1	196	95	52	34.3	17.5	62.8
C7552-CGC34.4	468	161	65	67.7	9.5	51.8
C7552-CGC17	136	84	39	35.0	14.2	21.3
C7552-CGC20	144	78	46	24.4	13.4	19.3
Average (%)			47.8	48.1	13.0	39.8

optimal sizing of transistor widths, the proposed LBMP timing optimization algorithm has reduced the PDP by an average of 40.17%.

The other electronic performance measurement associated with timing optimization is delay sensitivity (∂) due to process variations. Traditionally, CMOS device switching speed improves at a lower temperature due to increase in mobility. However, Negative Bias Temperature Instability (NBTI) effects may degrade the device switching speed

over time via threshold voltage shifts in PMOS transistors [29, 30], even at a lower temperature. The delay sensitivities of several ISCAS benchmark circuits, due to process variations at different temperatures are reported in Figure 12. It is observed that all circuits after timing optimization have a very little difference in the delay sensitivity reduction for different temperatures. The LBMP timing optimization provides consistent delay sensitivity at different temperatures.

TABLE 5: Power Delay Product optimization results from the proposed algorithm.

Design	Initial PDP (fWs)	Final PDP (fWs)	PDP Reduction (%)
CCT-2	3.57	1.87	47.49
2-b WBTC	13.66	6.185	54.74
4-b UWBTC	6.61	5.11	22.57
74181-CLA	5.04	3.04	39.55
74181-E Mod	3.55	1.90	46.46
C2670-CLA	2.97	1.75	41.07
C3540-CC5	1.09	0.66	38.78
C5315-CalP2	7.47	4.57	38.78
C5315-GLC4.2	3.68	2.70	26.48
C5315-CB4	4.61	2.93	36.43
C7552-CGC34.4	16.42	6.19	62.26
C7552-CGC17	1.63	1.15	28.97
C7552-CGC20	1.09	0.67	38.70
Average (%)			40.17

FIGURE 11: Delay Convergence using LBMP Algorithm.

5. Timing Optimization of Mixed-Static-Dynamic Circuits

Conventionally, synthesis tools perform design and optimization using static CMOS logic [31, 32]. It is not uncommon for the synthesis tools to not find an acceptable solution in terms of timing. This challenge can be answered through utilizing the advantage of fast timing in dynamic logic. Dynamic logic has smaller gate capacitances compared to their static CMOS counterparts, which accounts for a significant speedup [3, 33]. With static and dynamic logic having their respective advantages of low power and low delay, an optimal balance can be obtained by partitioning the design to use both static and dynamic logic in an effective manner.

FIGURE 12: Delay Sensitivity Reduction.

At the architecture level, a common limitation in most design optimization flows is the limited accountability for process variations. Typically after placement and route, if a design fails to meet the timing constraints, optimization flow is reiterated. Even after several iterations, design may still not meet the timing constraint and miss the time-to-market window. The process variation-aware Path Oriented IN Time (POINT) optimization algorithm proposed in Figure 13 answers these challenges of timing optimization and also accounts for process variations. Utilizing the LBMP algorithm proposed in Section 3, the POINT optimization algorithm partitions the design to effectively utilize both dynamic and static CMOS logic to meet the timing constraints.

Initially, a high-level description of a design is input to Synopsys Design Compiler (SDC) [31] for synthesis and optimization. The optimized designs from SDC are considered as the initial case for POINT optimization flow. Following synthesis and optimization, static timing

FIGURE 13: POINT Optimization Algorithm.

FIGURE 14: ISCAS benchmark–C3540 (8-b ALU).

FIGURE 15: Architecture of M5/UM5.6 in C3540.

FIGURE 16: Delay Convergence using POINT Optimization Flow.

analysis (STA) is performed using Synopsys PrimeTime (SPT) to identify the critical paths. Also, the critical timing modules identified by the number of occurrences, and delay significance on the critical paths are reported and dynamic circuits of the same are designed. Using the LBMP algorithm, iterative transistor sizing optimization for timing is performed on these dynamic circuits.

With the updated design comprising of dynamic logic circuits, clock tree design and timing verification is performed. After the design is verified for clock signal timing constraints, incremental STA is performed to verify for timing convergence. The algorithm is iteratively repeated towards convergence of acceptable solution. Following the timing convergence through iterations, the final mixed-static-dynamic circuit design is exported for placement and route.

The POINT optimization algorithm is verified through implementation on several ISCAS benchmark circuits, including C3540, an 8-b ALU as shown in Figure 14 [34]. Initial synthesis and optimization was performed using SDC, and static timing analysis was performed using SPT

TABLE 6: POINT Optimization Flow results.

Design	# Inputs	# Outputs	# Gates	Delay reduction (%)	Uncertainty reduction (%)
74181	14	8	74	43	14
c2670	233	140	1193	39	32
adder64	130	65	1491	61	63
c3540	50	22	1669	32	40
c5315	178	123	2406	29	32
c7552	207	108	3512	26	31
Average (%)				38	35

FIGURE 17: Timing Profiles using POINT Optimization Algorithm.

[35]. For the design in hierarchical format (synthesis and optimization was performed at block level, and design flatten option was disabled), the critical path delay was found to be 3.6 nanoseconds. The critical modules and the critical paths obtained from STA are highlighted in Figure 14. Based on the STA report, it is shown that the ALU Core-M5 with a delay of 1.24 nanoseconds is the timing critical module with the most number of worst-case paths. Figure 15 shows the schematic of UM5_6 from ALU Core-M5 with the critical paths highlighted; the submodules labeled CC5 and CC9 are timing critical with delays of 0.5 nanoseconds and 0.61 nanoseconds respectively.

With timing optimization being the primary goal in this stage, submodules CC5 and CC9 in M5/UM5_6 of C3540 are designed in dynamic logic, and timing optimization is performed using the LBMP algorithm. With dynamic circuits optimized using LBMP algorithm, the delay of CC5 was reduced from 0.5 nanoseconds to 0.07 nanoseconds, and delay of CC9 was reduced from 0.61 nanoseconds to 0.20 nanoseconds, respectively. After the first iteration of the POINT optimization flow, the critical path delay of C3540 was reduced from 3.6 nanoseconds to 2.8 nanoseconds. Further iterations of POINT optimization flow reduced the critical path delay from 3.6 nanoseconds to 2.4 nanoseconds, as shown in Figure 16. In addition to reducing the delay by 33%, the delay uncertainty due to process variations is also reduced by 40% as shown in Figure 17.

Similarly, the process variation-aware POINT optimization was implemented on other benchmark circuits and timing optimization results are presented in Table 6, where

both delay and uncertainty from process variations were reduced by an average of 38% and 35%, respectively, over initial designs optimized with Synopsys Design Compiler.

6. Conclusion

In this paper a process variation-aware timing optimization of dynamic logic and a timing optimization flow for mixed-static-dynamic CMOS logic have been presented. Solutions addressing further design challenges are presented by (1) considering delay uncertainties from process variations and (2) developing a process variation-aware Path Oriented IN Time (POINT) optimization algorithm for mixed-static-dynamic logic.

Through implementation and verification of several benchmark circuits in 130 nm CMOS process, the process variation-aware timing optimization algorithm has shown by average a delay reduction by 47.8%, an uncertainty reduction by 48%, a power increase by 13%, and a power-delay-product reduction by 40%. Validated through implementation of mixed-static-dynamic logic on a 64-b adder and several ISCAS benchmark circuits, the POINT optimization algorithm has demonstrated an average improvement in delay reduction by 38% and delay uncertainty reduction from process variation by 35%.

7. Acknowledgment

The authors wish to thank the anonymous reviewers for insightful comments to improve the quality of presentation.

References

- [1] D. H. Allen, S. H. Dhong, H. P. Hofstee, et al., "Custom circuit design as a driver of microprocessor performance," *IBM Journal of Research and Development*, vol. 44, no. 6, pp. 799–822, 2000.
- [2] P. E. Gronowski, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allmon, "High-performance microprocessor design," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 5, pp. 676–685, 1998.
- [3] M. Zhao and S. S. Sapatnekar, "Timing-driven partitioning and timing optimization of mixed static-domino implementations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 11, pp. 1322–1336, 2000.
- [4] L. Zhang, *Statistical timing analysis for digital circuit design*, Ph.D. dissertation, December 2005.
- [5] P. McGuinness, "Variations, margins, and statistics," in *Proceedings of the International Symposium on Physical Design*, pp. 60–67, Portland, Ore, USA, April 2008.
- [6] J. Tschanz, K. Bowman, and V. De, "Variation-tolerant circuits: circuit solutions and techniques," in *Proceedings of Design Automation Conference*, pp. 762–763, 2005.
- [7] P. S. Zuchowski, P. A. Habitz, J. D. Hayes, and J. H. Oppold, "Process and environmental variation impacts on ASIC timing," in *Proceedings of IEEE/ACM International Conference on Computer Aided Design (ICCAD '04)*, pp. 336–342, November 2004.
- [8] S. B. Samaan, "The impact of device parameter variations on the frequency and performance of VLSI chips," in *Proceedings of IEEE/ACM International Conference on Computer Aided Design (ICCAD '04)*, pp. 343–346, 2004.
- [9] K. Yelamathi and C.-I. H. Chen, "A path oriented in time optimization flow for mixed-static-dynamic CMOS logic," in *Proceedings of the 51st Midwest Symposium on Circuits and Systems*, pp. 454–457, Knoxville, Tenn, USA, August 2008.
- [10] J. P. Fishburn and A. E. Dunlop, "TILOS: a posynomial programming approach to transistor sizing," in *Proceedings of IEEE International Conference on Computer Aided Design (ICCAD '85)*, pp. 326–328, Santa Clara, Calif, USA, 1985.
- [11] V. Sundararajan, S. S. Sapatnekar, and K. K. Parhi, "Fast and exact transistor sizing based on iterative relaxation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 5, pp. 568–581, 2002.
- [12] M. Borah, R. M. Owens, and M. J. Irwin, "Transistor sizing for minimizing power consumption of CMOS circuits under delay constraint," in *Proceedings of the International Symposium on Low Power Design*, pp. 167–172, Dana Point, Calif, USA, April 1995.
- [13] S.-O. Jung, K.-W. Kim, and S.-M. Kang, "Transistor sizing for reliable domino logic design in dual threshold voltage technologies," in *Proceedings of the 11th Great Lakes Symposium on VLSI (GLSVLSI '01)*, pp. 133–138, West Lafayette, Ind, USA, March 2001.
- [14] Z. Luo, "General transistor-level methodology on VLSI low-power design," in *Proceedings of the ACM Great Lakes Symposium on VLSI (GLSVLSI '06)*, pp. 115–118, Philadelphia, Pa, USA, April 2006.
- [15] A. R. Conn, I. M. Elfadel, W. W. Molzen Jr., et al., "Gradient-based optimization of custom circuits using a static-timing formulation," in *Proceedings of Design Automation Conference*, pp. 452–459, June 1999.
- [16] I. Sutherland, B. Sproull, and D. Harris, *Logical Effort: Designing Fast CMOS Circuits*, Morgan Kaufmann, San Francisco, Calif, USA, 1999.
- [17] N. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, Addison Wesley, Boston, Mass, USA, 3rd edition, 2004.
- [18] K. A. Bowman, S. G. Duvall, and J. D. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 2, pp. 183–190, 2002.
- [19] M. Orshansky, *Increasing Circuit Performance through Statistical Design Techniques*, Closing the Gap between ASIC & Custom, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.
- [20] D. Burnett, K. Erington, C. Subramanian, and K. Baker, "Implications of fundamental threshold voltage variations for high-density SRAM and logic circuits," in *Proceedings of the Symposium on VLSI Technology*, pp. 15–16, Honolulu, Hawaii, USA, June 1994.
- [21] K. Takeuchi, T. Tatsumi, and A. Furukawa, "Channel engineering for the reduction of random-dopant-placement-induced threshold voltage fluctuation," in *Proceedings of the IEEE Electron Devices Meeting (IDEM '97)*, pp. 841–844, Washington, DC, USA, December 1997.
- [22] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *Proceedings of Design Automation Conference*, pp. 338–342, 2003.
- [23] C. H. Kim, K. Roy, S. Hsu, R. Krishnamurthy, and S. Borkar, "A process variation compensating technique with an on-die leakage current sensor for nanometer scale dynamic circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 6, pp. 646–649, 2006.
- [24] L. Scheffer, "The Count of Monte Carlo," in *Proceedings of the ACM/IEEE International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems (TAU '04)*, February 2004.
- [25] F. Maloberti and C. Gang, "Performing arithmetic functions with the Chinese abacus approach," *IEEE Transactions on Circuits and Systems II*, vol. 46, no. 12, pp. 1512–1515, 1999.
- [26] B. Fu, Q. Yu, and P. Ampadu, "Energy-delay minimization in nanoscale domino logic," in *Proceedings of the 16th ACM Great Lakes Symposium on VLSI (GLSVLSI '06)*, pp. 316–319, Philadelphia, Pa, USA, April 2006.
- [27] P. R. Kinget, "Device mismatch and tradeoffs in the design of analog circuits," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 6, pp. 1212–1224, 2005.
- [28] W. Wolf, *Modern VLSI Design: IP-Based Design*, Prentice Hall, Upper Saddle River, NJ, USA, 4th edition, 2008.
- [29] B. Lasbouvaygues, R. Wilson, N. Azemard, and P. Maurine, "Timing analysis in presence of supply voltage and temperature variations," in *Proceedings of the International Symposium on Physical Design*, pp. 10–16, 2006.
- [30] W. Wang, S. Yang, S. Bhardwaj, et al., "The impact of NBTI on the performance of combinational and sequential circuits," in *Proceedings of the 44th Annual Design Automation Conference*, pp. 364–369, 2007.
- [31] Synopsys Design Compiler, <http://www.synopsys.com/>.
- [32] Cadence Encounter, <http://www.cadence.com/>.
- [33] R. Puri, "Design issues in mixed static-dynamic circuit implementation," in *Proceedings of International Conference on Computer Design*, pp. 270–275, 1998.

- [34] M. C. Hansen, H. Yalcin, and J. P. Hayes, “Unveiling the ISCAS-85 benchmarks: a case study in reverse engineering,” *IEEE Design and Test of Computers*, vol. 16, no. 3, pp. 72–80, 1999.
- [35] Synopsys PrimeTime, <http://www.synopsys.com/>.