

Glitch-Optimized Circuit Blocks for Low-Power High-Performance Booth Multipliers

Anuradha Chathuranga Ranasinghe¹, Graduate Student Member, IEEE, and Sabih H. Gerez²

Abstract—This article presents a novel implementation scheme of the essential circuit blocks for high-performance, full-precision Booth multipliers leveraging a hybrid logic style. By exploiting the behavior of parasitic capacitance of MOSFETs, a carefully engineered design style is employed to reduce dynamic power dissipation while improving the glitch immunity of the circuit blocks. The circuit-level techniques along with the proposed signal-flow optimization scheme prevent the generation and propagation of spurious activities in both partial-product and adder-tree stages. Two full-precision Booth multipliers built from proposed strategies were compared to the state-of-the-art versions known from literature by means of extensive post-layout simulations in 65-nm CMOS technology. The proposed versions on average demonstrated up to 10% and 30% power savings in general.

Index Terms—Alternative logic styles, array multipliers, Booth multipliers, CMOS, glitch reduction, spurious activities, Wallace tree, XOR-XNR.

I. INTRODUCTION

MULTIPLIERS are essential components of digital hardware, ranging from deeply embedded system-on-chip (SoC) cores to GPU-based accelerators. As they are often critical for system performance, a great emphasis was placed on their performance improvement in the past few decades [1]–[7]. While performance remains important, the high demand for battery-powered ubiquitous systems has promoted low-power operation to a primary design goal [8]. However, the majority of proposed high-performance multipliers suffers from increased capacitive loads and spurious activities due to their complex combinatorial modules and unbalanced reconvergent paths which could turn the multiplier to be the dominant source of power dissipation.

The Radix-4 modified Booth encoding (MBE) scheme is often preferred in high-performance multipliers due to its minimized delay and silicon area. Booth encoding reduces the number of partial products required to be added by approximately twofold compared to non-Booth versions. Moreover MBE is incorporated with various adder-tree-reduction schemes such as Wallace [2], optimized Wallace-tree (OWT) [9], [10], Dadda [3], Braun's [4] and three-dimensional

minimization (TDM) [11]–[14] to speedup the partial product addition. OWT scheme along with carry-save propagation is known for logarithmic delay reduction of the adder-tree which is composed of either full-adders [14]–[17] or 4-to-2 compressors [18]–[20]. The latter is preferred for a regular adder-tree implementation.

Despite faster operation, the fitness of MBE for energy efficiency has been questioned due to its complex encoding-decoding circuitry and higher spurious activities. This fact is especially prominent when the input operands are in 2's complement notation and have a smaller dynamic range. Therefore, alternative multiplier schemes such as Baugh-Wooley [5], [21]–[23], sign magnitude (SM) [24], [25], and gray coding (GC) [26] have been proposed. The Baugh-Wooley implementation utilizes a 2-input AND array for partial product generation (PPG), which is simpler in logic and was shown to be ~25% more power efficient at a slightly higher delay [23] when compared to Booth version. SM and GC, on the other hand, leverage the number representation to lower the signal transitions at the expense of a format conversion logic at both ends of the multiplier. SM implementations [24], [25] have reported up to 90% and 50% reduction in switching activity whereas GC [26] reports 45% of power reduction compared to MBE. However, the applications where the input operands rapidly change across the entire word length scarcely benefit from these techniques. Besides, when the timing constraints are stringent, the conversion circuits in the critical path make these implementations slower and even more power-hungry due to the gate upsizing.

The Booth multiplier has also been subjected to structural and gate-level optimizations in literature. A more regular partial product array [15], [16], [27] was proposed to minimize the extra adder rows for carry summation. The approach in [27] has improved the multiplier performance by 25% when compared with the conventional implementations. Kang and Gaudiot [15] presents a fast 2's complement generation circuit to reorganize the partial product array by removing the subsequent carry-in terms. The work in [16] proposes a less hardware-intensive mechanism to achieve the same goal. These approaches have achieved up to 5%–9.1% improvements in performance while reporting 15%–33% of power savings for an 8-bit version, respectively. As alternatives to OWT, leap-frog (LFR) [28], [29], and left-to-right [7], [17] structures were proposed to alleviate the sum-carry imbalance. Despite their feasible layouts, the incurred area and delay overhead is not negligible. Alternatively, the optimized circuits presented in [7] and [16] demonstrate more balanced data paths and an efficient partial-product array structure that

Manuscript received May 5, 2020; accepted June 21, 2020. Date of publication July 27, 2020; date of current version August 26, 2020. This work was supported in part by the Dutch NWO Applied and Engineering Sciences program *ZERO: Towards Energy Autonomous Systems for IoT* and in part by Dialog Semiconductor B.V., The Netherlands. (Corresponding author: Anuradha Chathuranga Ranasinghe.)

The authors are with the Chair of Computer Architecture for Embedded Systems, University of Twente, 7522 NB Enschede, The Netherlands (e-mail: rnsburg@gmail.com).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TVLSI.2020.3009239

1063-8210 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

outperform other higher level implementations. Row and column bypassing [30], [31], dynamic operand interchanges [32], [33] were also considered to exploit the multiplier input asymmetry for low power. These techniques are questionable in general cases as the extra circuit overhead is a heavy burden. More recent approaches [34]–[41] exploit the accuracy and the number representation for energy efficiency. Among them, only [41] can be found relevant to the scope of this work, and it employs the same circuits presented in [7] and [16].

This work proposes a novel transistor-level implementation of the essential circuit blocks of Booth multipliers aiming to lower dynamic power dissipation. By presenting a comprehensive study on the contribution of parasitics and the spurious activities, a careful design strategy and an optimized interconnect scheme are presented. The energy efficiency of the proposed approach is evaluated against previous full-precision Booth multipliers. The remainder of this article is structured as follows. Section II carefully analyzes the sources of dynamic power dissipation in Booth multipliers and the significance of MOSFET parasitic capacitance on the circuit style. The functionality of the novel circuitry and its merits are illustrated in Sections III and IV. An optimal device sizing strategy for presented circuits is elaborated in Section V. Section VI presents the performance evaluation and finally, the conclusions are drawn in Section VII.

II. SOURCES OF DYNAMIC POWER DISSIPATION

The switching of parasitics is the dominant power source of PPG of the MBE. In terms of transistor density of PPG, MBE [16] requires approximately 40% more transistors than non-Booth versions [23] and eventually results in more transitions in PPG. In addition to that, both PPG and adder-tree are prone to spurious (redundant) switching activities resulting in wasted power. The spurious switching is primarily attributed to the different arrival times of the input signals to the adder-tree. It propagates from the first row to the latter rows of the adder-tree, where the amount of spurious switching gradually increases. The significance of both aspects is evaluated in this section. Note that this article's evaluation is based on 65-nm bulk CMOS technology.

A. Behavior of Parasitic Capacitance in MOSFETs

Fig. 1 illustrates the MOSFET parasitic behavior from 40- and 65-nm technology libraries, respectively. Note that PMOS is ratioed with respect to the minimum sized NMOS device for equal driving strengths. C_g , C_d , C_s , and C_b represent the parasitic capacitance at corresponding MOSFET terminals. During the rise/fall time period t_1 – t_2 of the complementary control signals, each device transits from cutoff region to saturation. As such, the channel formation imposes a non-linear time-variant behavior on all parasitic capacitances. The average current consumed during this transition period at the k th terminal can be expressed as

$$i = \frac{1}{(t_2 - t_1)} \int_{t_1}^{t_2} C_k dt \cdot \frac{dV_k(t)}{dt} \quad (1)$$

where $dV_k(t)/dt$ is the slew rate of the signal at terminal k . The total capacitance seen at the gate terminal is expressed

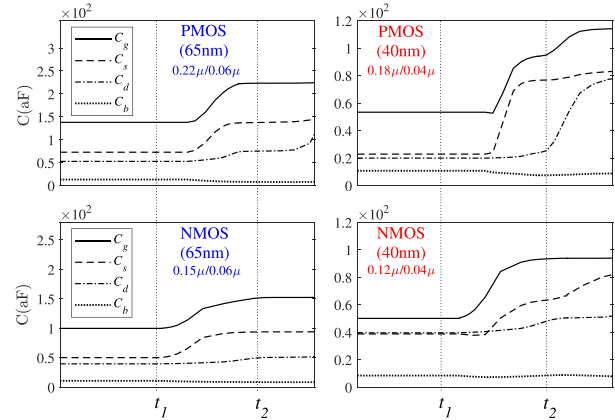


Fig. 1. MOSFET parasitic capacitance behavior during the switching.

as [42], [43]

$$\begin{aligned} C_g &= C_{gs} + C_{gd} + C_{gb} & C_{g_on} &= 2(WL_o C_{ox} + WLC_{ox}/3) \\ C_{g_off} &= 2WL_o C_{ox} + WLC_{ox} C_{dep}/(WLC_{ox} + C_{dep}) \end{aligned} \quad (2)$$

where C_{ox} , L_o represent the oxide capacitance and the gate overlapped length of the MOSFET. C_{g_off} and C_{g_on} are the equivalent gate capacitances in cutoff and saturation regions. Depletion capacitance C_{dep} is relatively smaller in nonsaturation, so that $C_{g_off} < C_{g_on}$. Similarly

$$\begin{aligned} C_{s_off} &= WL_o C_{ox} + WL_S C_j + (2L_S + W)C_{jsw} \\ &\quad + WC_{jsw_c} \\ C_{s_on} &= 2WLC_{ox}/3 + C_{s_off} & C_{d_on} = C_{d_off} = C_{s_off} \end{aligned} \quad (3)$$

C_j , C_{jsw} , and C_{jsw_c} correspond to the junction bottom plate and sidewall capacitances. L_S represents the sidewall length. It should be noted that both L_S and L_o are much smaller than the gate length L . The junction capacitances can be minimized by sharing the common drain–source areas between adjacent devices in cell layouts. Simulation results confirm the aforesaid behavior of the parasitics and the dominance of the gate parasitic capacitance in both technology nodes. Therefore, the cell topologies of the least number of gate parasitics and of smaller geometries are ideal for dynamic power reduction regardless of the technology node in use.

B. Spurious Activity Generation

The dominant source of spurious activities in a multiplier was attributed to the sum-carry imbalance of the adder-tree [7], [11]–[14], [28], [44]. However, a considerable amount of these activities also stems from PPG. This can be further elaborated by referring to the top-down structure of the improved Booth multiplier [16] (8 bit) shown in Fig. 2(a). $PP_{i,j}$, C_i , \bar{S}_i , and τ_i in PPG, represent the partial product, negative carry-in, sign-extension and LSB terms of each row, respectively. The adder-tree can be in one of the presented routing schemes. The final adder is typically realized by a faster adder such as a carry-lookahead (CLA) or a carry-propagation (CPA) adder.

For an $M \times N$ multiplier, the encoder–decoder signal arrangement for PPG is depicted in Fig. 2(b). Fig. 2(c) depicts the contribution of spurious activities from both PPG and adder-tree of an 8- and 16-bit conventional Booth multipliers [27]. Note that these adder-trees were constructed utilizing full adders. The activities were captured in an analog SPICE

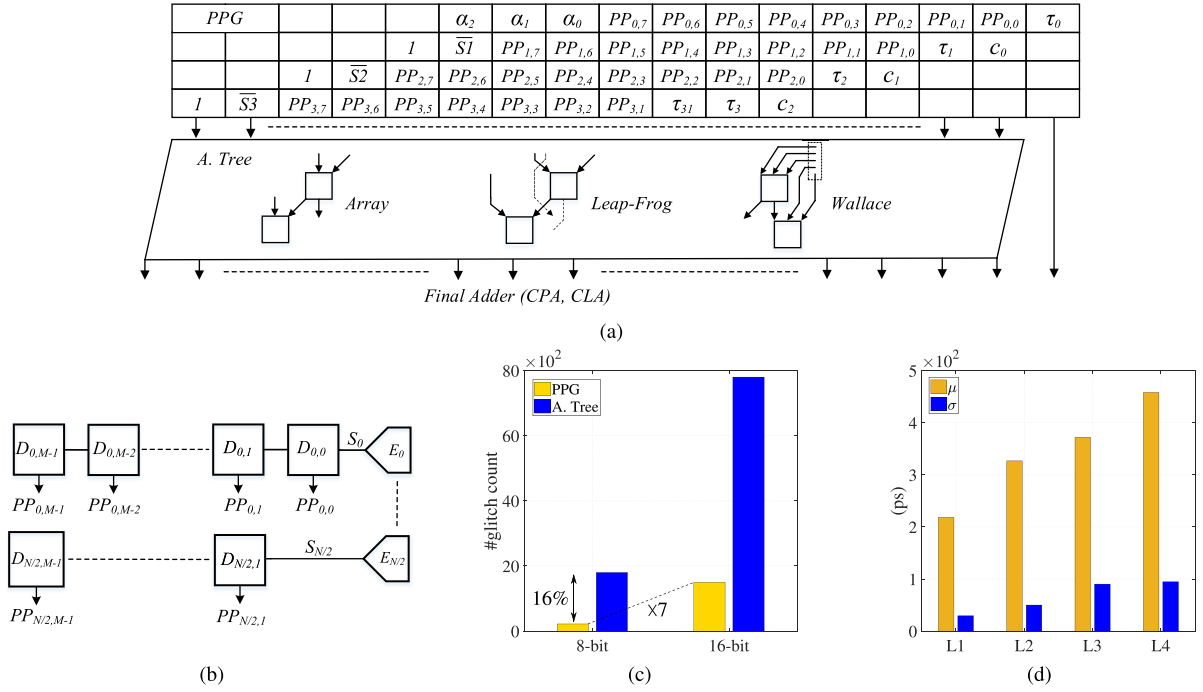


Fig. 2. (a) Top-down structure of the 8-bit Booth multiplier [16]. (b) PPG stage. (c) Spurious activities contributed by PPG versus Adder-Tree of the multipliers. (d) Delay variation across the Adder-Tree rows L1–L4 of 16-bit version (μ – maximum delay difference, σ – standard deviation of delays).

environment by monitoring the narrow pulses that cross the 50% level of V_{DD} . As depicted, the glitches that emanate from the PPG block of the 8-bit version is $\sim 16\%$ of the total glitch count and this becomes prominent in the 16-bit version ($\sim 7\times$) due to the imbalance of the accumulated capacitive loads along the encoder signal lines. Moreover, the encoder (E_0 – $E_{N/2}$) driving strength required for large operand widths is higher due to the high fan-out nature of the signals S_0 – $S_{N/2}$. Hence, the delay mismatch among the signals arriving at decoder loads (D_0 – D_{M-1}) is inevitable. In the worst case, these glitches could propagate till the final adder row.

The rest of the spurious activities originates from the adder cells owing to two reasons: the mismatch of the adder cell input capacitance and intracell sum-carry delay imbalance. The delay variation of the arrival signals at different levels of a 16-bit multiplier adder-tree (1.2 V, at 250 MHz) is depicted in Fig. 2(d). L1–L4 represent the adder-tree levels, whereas μ corresponds to the maximum delay observed in the signal arrival at each level. σ represents the standard deviation of the delays. With the aid of Elmore [45] delay model, the arrival time to a CMOS adder cell input can be related to the inertial delay τ_D of the cell as follows:

$$V_{out}(t) = V_{DD}(1 - e^{-t/R_{eq}C_{eq}})$$

$$\tau_D = R_{eq}C_{eq} \ln\left(\frac{V_{DD}}{V_{DD} - V_{th}}\right) \quad (4)$$

where the total parasitic time constant $R_{eq}C_{eq}$ is given by

$$R_{eq}C_{eq} = R_{M1}C_{M1-Poly} + (R_{M1} + R_{Poly})C_L \quad (5)$$

R_{M1} , R_{Poly} , $C_{M1-Poly}$ represent the extrinsic metal-1, polysilicon interconnect parasitic resistances and metal-1-poly via capacitance, respectively. C_L corresponds to the intrinsic capacitive load seen at the adder cell input, according

to (2). In 65-nm technology, typically $R_{Poly} \geq 60R_{M1}$ while $C_L(i.e., C_g) \geq 4C_{M1-Poly}$ per unit area. The V_{th} of the transistors which switch, is assumed to be the minimum compliance voltage for the full adder input so that the input signal should be stable after τ_D to excite the input transistors properly. If the PPG outputs are synchronized and sufficiently strong in driving strength, the first row (L1) of the adder-tree becomes relatively less prone to the arrival mismatch, as depicted in Fig. 2(d). According to (4) and (5), the arrival time of the PPG signals to full adders mainly depends on the intrinsic parasitic elements as the encoder–decoder blocks are typically placed near to the adder-tree. The subsequent stages of the adder-tree are susceptible to larger variations as the intracell sum-carry delay dominates in L2–L4. The intercell sum-carry delay has been addressed to some extent in [7] and [28] with the aid of different routing schemes. However, the complexity of these schemes is relatively higher and the spurious activities remain. Alternatively, the latch-based adder-tree [44] is a promising way to counteract this issue, yet the gain of the implementation could be less favorable for high-performance multipliers.

III. NOVEL CIRCUITS FOR MBE

As observed in earlier research, a proper choice of intermediate signals in the interface between Booth encoding and decoding offers opportunities for logic optimization. Fig. 3(a)–(d) illustrates the traditional implementations of MBE circuits found in the literature. Note that only the full-swing circuit topologies were considered in this study. Fig. 3(a) (BED13) depicts a hybrid implementation of encoder–decoder circuits which require 36 and 10 transistors [46], respectively. This non-CMOS implementation reports the least number of transistors for the decoder

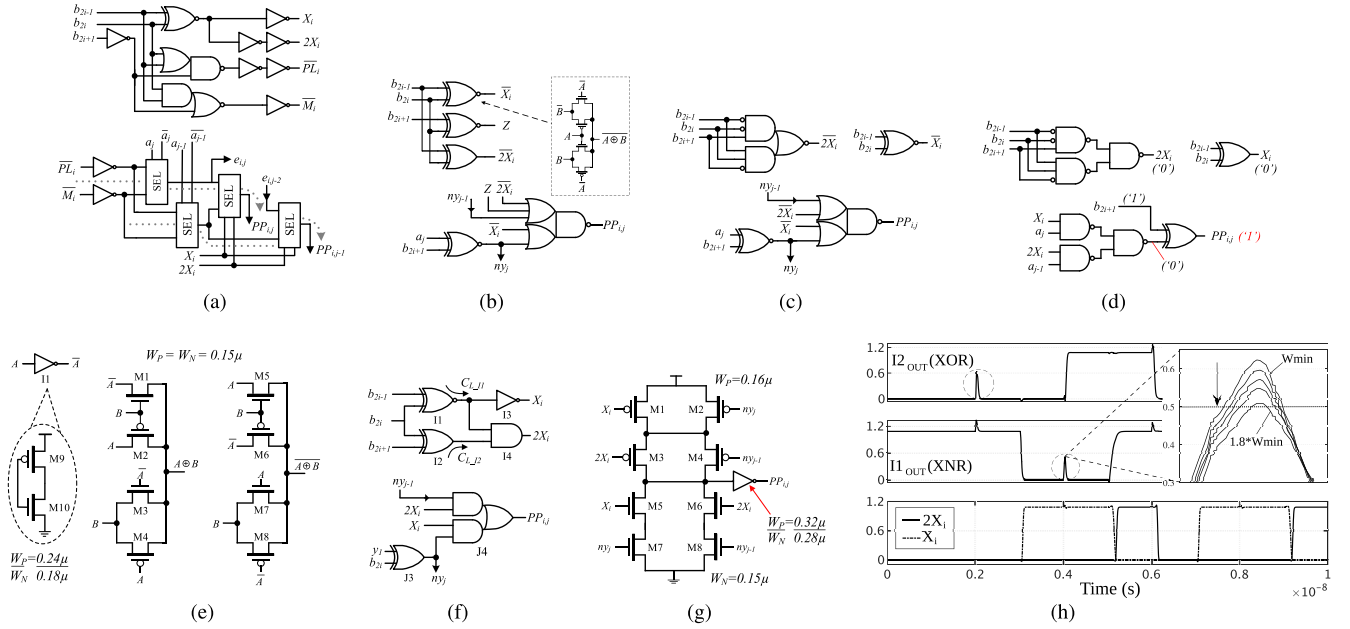


Fig. 3. Various Booth encoder–decoder implementations. (a) BED13 [46]. (b) BED20 [27]. (c) BED22 [7], [16], [41]. (d) Erroneous Booth circuits in [17]. (e) 6T-XOR/XNR circuits of this work ($W_{M1-M8} = 0.15\mu$). (f) Proposed encoder–decoder circuits (BED18). (g) AO22 ($J3$) of the decoder ($W_{M1-M4} = 0.16\mu$, $W_{M5-M8} = 0.15\mu$).

block among the presented. However, there are a few issues that emanate from this implementation. First, the unbuffered selector circuit which is denoted by *SEL* (composed of four pass transistors), forms cascaded resistive paths from decoder inputs to the outputs as highlighted in Fig. 3(a). This results in an asymmetry in the driving loads to the *SEL* blocks for different input combinations and therefore different arrival times. Secondly, the routing congestion across the decoding blocks in Fig. 3(a) is relatively higher and increases the interconnect parasitics across the PPG.

The circuits shown in Fig. 3(b) (BED20) [27] uses transmission gate pairs for encoders leading to a faster operation in PPG. However the unbuffered encoder outputs become transparent to the hazards induced by the circuit itself. The additional wiring and higher capacitive loading at the decoder leads to a higher power consumption in PPG at the same time. The arrangement in Fig. 3(c) (BED22) [7], [16] is the most optimized version in terms of transistor count and signal synchronization. The XORs which produce $ny_{j-1}-ny_j$ are shared among the decoders and the AOI22 cell provides balanced loads to the encoder signals. Therefore, it was also preferred for the truncated multiplication in [41]. The unique Booth circuits presented in [17] and [44] are not considered for the evaluation due to functional failures when all the encoder inputs ($b_{2i-1}-b_{2i+1}$) are at logic “1” [see Fig. 3(d)]. The proposed MBE circuits in this work are shown in Fig. 3(e)–(g).

The essential leaf cell of the proposed circuitry is depicted in Fig. 3(e). This XOR/XNR arrangement results in fewer number of gate capacitances when compared to any other full-swing implementations [47]–[49]. Despite this merit, it suffers from the delay asymmetry between the signal paths. If, for example, in the circuit of Fig. 3(e), when both inputs change from $0 \rightarrow 1$, *M1* of the XOR drives the output for a

short period of time due to the inertial and propagation delays of the inverter and as a result, a glitch appears at the XOR output. The inversely proportional relationship between the inertial and propagation delays limits the liberty of device sizing. As such, the direct interfacing of these XOR/XNR outputs to high fan-out nets could only worsen the spurious activities in PPG.

Fig. 3(f) (BED18) illustrates the proposed encoder–decoder circuit blocks in this article. I_1 and I_2 of the encoder block are directly constructed from Fig. 3(e) by sharing b_{2i} . This requires only one inverter for the input b_{2i} . I_3 and I_4 generate X_i-2X_i while providing output buffering to these signals. The buffering capacitance of I_3 and I_4 along with the resistive paths of I_1 and I_2 now acts as a low-pass filter and absorbs possible glitches produced at XOR/XNR outputs as depicted in Fig. 3(h). These observations were captured at the worst-case corner (SS, $V_{DD}-10\%$ and 125°C) operation. SS (TT) denotes slow–slow (typical–typical) corner process parameters for NMOS/PMOS devices, respectively. From (4), the condition to satisfy this filtering requirement is given by

$$\begin{aligned} t_{(A \rightarrow \bar{A})} - \tau_{D_B} &< \tau'_{\text{XOR-Vth}} \\ \tau_{D_INV} + \tau_{pd_INV} - \tau_{D_B} &< \tau'_{\text{XOR-Vth}} \end{aligned} \quad (6)$$

where

$$\tau'_{\text{XOR-Vth}} = R_{D1} (C_{d1_on} + C_{\text{OFF_XOR}} + C_{L_J2}) \ln \left(\frac{V_{DD}}{V_{DD} - V_{th}} \right).$$

$t_{A \rightarrow \bar{A}}$ is the total delay for input *A* to reach its complement \bar{A} . τ_{D_B} and $\tau'_{\text{XOR-Vth}}$ represent the inertial delay of *M1* and the time for \bar{A} to reach V_{th} of the MOSFET at load C_{L_J2} . τ_{D_INV} and τ_{pd_INV} represent the inertial and propagation delays of the inverter *I1*. $C_{\text{OFF_XOR}}$ is the total off-state parasitic capacitance at the XOR node and R_{D1} is the equivalent drain resistance of *M1*. Up-sizing of the inverter is not practical here as it increases the inertial delay τ_{D_INV} . Instead, this condition can

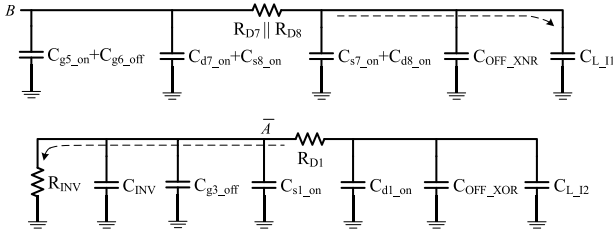


Fig. 4. Worst-case delay mismatch between of XOR-XNR circuits [from Fig. 3(e)].

be met by fine tuning R_{D1} , C_{L_I1} and C_{L_I2} . This corresponds to the width adjustment of pass transistors of XOR/XNR circuits and the input transistors of I_3 and I_4 . When standard device sizing in Fig. 1 is applied, the required width for $C_{L_I1} - C_{L_I2}$ was found to be $\sim 1.8 \times$ of the minimum drawn width (W_{\min}). Consequently, the glitch peak will never reach V_{th} as depicted in Fig. 3(h). The effect of these adjustments to the overall delay is negligible because the width adjustment of I_3 and I_4 ultimately reduces their propagation delays. A more elaborate analysis on transistor sizing will be given in Section V.

Furthermore, the worst-case delay mismatch between I_1 and I_2 outputs occurs when $b_{2i} = b_{2i+1} = 1$. The equivalent RC circuits for the paths of XOR-XNR for this scenario are shown in Fig. 4. Assuming $b_{2i-1} = "1"$ in this state, the paths correspond to $M1$ of XOR and $M7$ and $M8$ of XNR in Fig. 3(e) become activated. The effective parasitic drain resistance during this period can be expressed as follows [45]:

$$R_D \approx \frac{3}{4} \frac{V_{DD}}{\mu C_{ox} \frac{W}{L} (V_{DD} - V_{th})^\alpha (1 - \frac{1}{3} \lambda V_{DD})} \quad (7)$$

where λ is the channel length modulation parameter. Note that R_D is calculated for 50% rise-fall time. Since the NMOS and PMOS pass transistors of both circuits are sized for equal driving strengths, $R_D \approx R_{D_NMOS} \approx R_{D_PMOS}$. For simplicity, the source resistance of the preceding driving stage is assumed to be smaller for all inputs, so that the effect of $C_{g5_on} + C_{g6_off}$ and $C_{d7_on} + C_{s8_on}$ is negligible for XNR. For the propagation delays from inputs to the outputs of I_1 and I_2 , (4) and (5) can be rewritten to (at 50% of V_{DD})

$$\tau_{pd_XNR} = 0.69(R_{D7} || R_{D8})(C_{s7_on} + C_{d8_on} + C_{OFF_XNR} + C_{L_I1}) \quad (8)$$

$$\tau_{pd_XOR} = 0.69\{R_{INV}(C_{INV} + C_{g3_off} + C_{s1_on}) + (R_{INV} + R_{D1})(C_{d1_on} + C_{OFF_XOR} + C_{L_I2})\} \quad (9)$$

From (8) and (9), τ_{pd_XOR} evidently becomes larger due to the series R_{INV} and R_{D1} . However, interfacing the faster path to the both inputs of I_3 and I_4 as shown in Fig. 3(f) alleviates this timing mismatch ($C_{L_I1} > C_{L_I2}$).

The XOR $J3$ in decoder block is constructed by combining the XNR circuit in Fig. 3(e) with an inverter. In addition to glitch filtering, this satisfies the delay matching between ny_j and the rest of the decoder input signals. The inputs to the decoder are connected to the equally sized NMOS-PMOS pair in AO22 ($J4$) cell which reasonably provides equal loads for all the input signals. Similar to the encoder, the buffering capacitance introduced by AO22 in Fig. 3(g), filters out any possible glitch in the decoder block. Moreover, the output buffering relaxes the sizing of $M1-M8$ of AO22. This property is not

available in OAI22 of Fig. 3(c) and hence OAI22 requires wider transistors despite the fewer number of devices. If the regular PPG scheme presented in Fig. 2(a) is adopted for an 8-bit multiplier, the implementations in Fig. 3(a)-(c) require an average of 13, 20, and 22 transistors per block for PPG, respectively, while the proposed one needs 18.

IV. MULTIPLIER ADDER-TREE OPTIMIZATION

A. Balanced Full Adder Design

Full adders are the basic building blocks of the multiplier adder-tree. The most prevalent, rail-to-rail static full adder implementations are shown in Fig. 5(a)-(e). For a fair comparison, the buffered versions of the original implementation are considered. The blue arrow line indicates the critical path of each full adder. Fig. 5(a)-(c) [50]-[52] requires a minimum of 22 transistors (including the inverters for the input signals that have not been drawn). The numbers for Fig. 5(d)-(f) are 26, 28, 26, respectively. Fig. 5(a) (RFL22) [50] utilizes a simultaneous, six transistors XOR-XNR circuit which is delimited by a dashed line in 5(a). Despite its compactness, the regenerative feedback paths introduced by this circuit results in slower transitions. In addition, the cascaded transmission gates worsen the sum-carry generation (SCG), thereby making the outputs more susceptible to glitches. In Fig. 5(b) (TFA22) [51], the Sum output (S) is produced faster when input $C = "1"$, compared to other input combinations. Besides, the late arrival of XOR-XNR signals to the SCG could introduce glitches at output S . By contrast, the control signals to the transmission gates in Fig. 5(c) [52] (BFA22) are reasonably synchronized except its input signals, i.e. early arrival of input C when $XOR_{0 \rightarrow 1}$ is a potential scenario for glitch generation at output S . Similar to RFL22 and TFA22, HFA26 in Fig. 5(d) [49] suffers from asymmetric path delays despite its faster operation. Fig. 5(e) (CMOS28) [44] represents the traditional CMOS full adder which is reasonably immune to glitches. The proposed full adder (PBFA26) is illustrated in Fig. 5(f).

This arrangement differs from the others in two aspects. First, the internal signals are capacitively terminated at the SCG stage and the gate capacitances of the transmission gate pairs in SCG absorb possible glitches similar to Booth circuits. Secondly, the synchronization of all signals to SCG is achieved by incorporating a low-overhead intracell delay element [44] depicted by $M1-M4$ of Fig. 5(f). $M1$ and $M4$ provide the required delay to the input C through their drain-source parasitics C_d/C_s which are smaller than C_g . Since C_g of both $M1$ and $M4$ are not switched, its parasitic contribution to the full adder dynamic power is significantly lower when compared to an inverter-based delay elements. Hence, the arrival of \bar{C} can be independently controlled without a significant overhead. The equivalent RC circuit for $M1-M4$ for condition $C_{10 \rightarrow 1}$ is shown in Fig. 6. Similar to (8) and (9), the synchronization delay required for input C can be expressed as follows:

$$\tau_{pd_C \rightarrow C1} = 0.69\{R_{D1}(C_{d1_on} + C_{s2_on}) + (R_{D1} + R_{D2}) \times (C_{d2_on} + C_{d3_off} + C_{INV})\} \quad (10)$$

With appropriate device sizing to $M1(M4)$, the required delay can be obtained with the minimal impact to the loading of

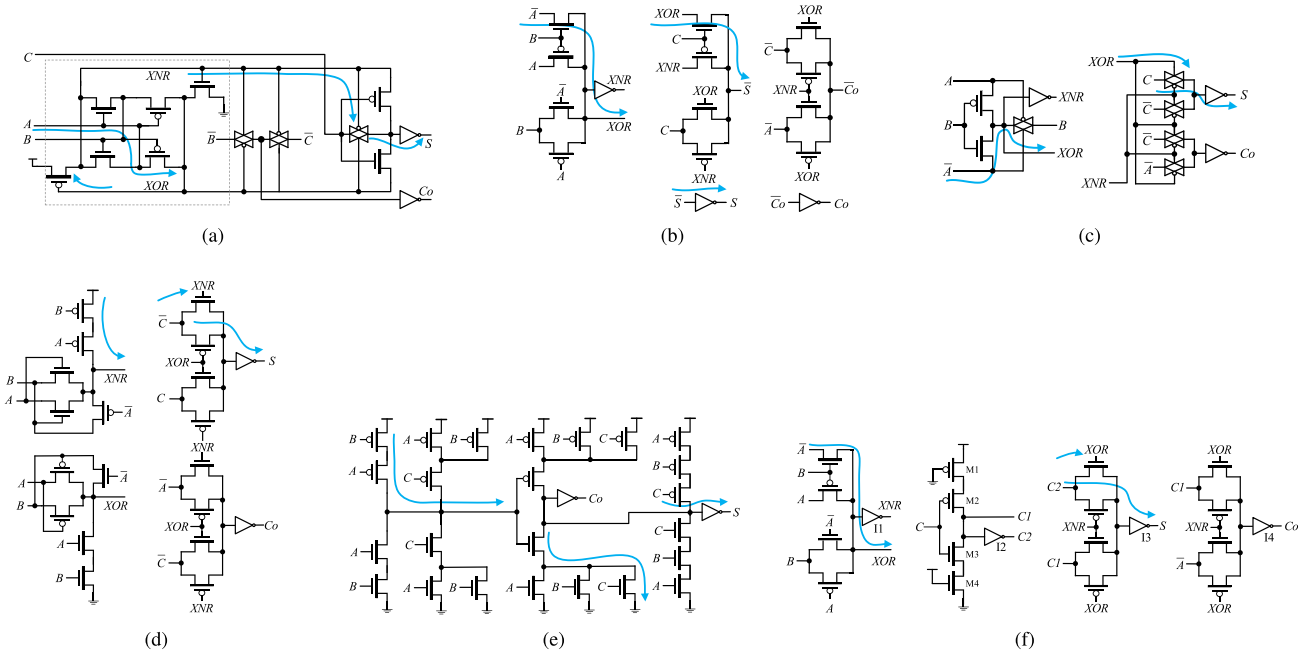


Fig. 5. Various low-power, full-swing full adders. (a) RFL22 [50]. (b) TFA22 [51]. (c) BFA22 [52]. (d) HFA26 [49]. (e) CMOS28 [44]. (f) Proposed (PBFA26).

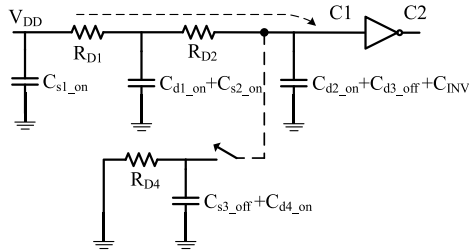


Fig. 6. Equivalent RC circuit for M1–M4 of Fig. 5(f) when $C1_0 \rightarrow 1$.

the inverter $M2$ and $M3$, such that $\tau_{pd_C \rightarrow C1} \approx \tau_{pd_A,B \rightarrow xor} \approx \tau_{pd_A,B \rightarrow xnr}$.

B. Optimized Interconnect Network

If the conventional full-adders or 4-to-2 compressors are utilized, care must be taken to synchronize the sum-carry signals with the aforementioned techniques (i.e., TDM, LFR). In addition to the reduction schemes (i.e., OWT or Array), if the proposed full adder (PBFA26) is adopted, the signal probability can be exploited to lower both spurious activities and dynamic power of the adder-tree. It is apparent from Fig. 5(f) and (2) and (3) that the transitions at inputs A and B of PBFA26 are internally driving a higher gate capacitance than at input C . Moreover, the total capacitance excited by input $A|_{B=0}$ is slightly higher than input $B|_{A=0}$. This is also true for both inputs when their corresponding reference signals are at logic “1.” More importantly, the worst-case input capacitances seen at inputs A and B ($\approx FO2$ – $FO3$) is moderately higher than input C ($\approx FO1$), so that the predriver at input C always consumes less power. Note that $FO2$ refers to a fan-outs of 2. These facts justify that $P_A > P_B > P_C$ where P_j is the average power consumed due to the transitions at input j . From the standard Radix-4 MBE table [16], the switching

Algorithm 1 : Parasitic-Aware Signal Routing (PASR)

Input: $\rho_{i,n}$ of input list $i_n \in \{PP_{ij}, \tau_{i0}, \tau_{i1}, c_i, \bar{S}_i, \alpha_i, S_{ij}, Co_i\}$; $n \in \mathbb{N}; 0 \leq n < 3$
Output: Optimum i_n assignment to FA_i inputs
 Connect i_n of $\max\{\rho_{i,n} : n=0, \dots, 2\}$ to pin C
 Update the list i_n ; $0 \leq n < 2$
if $\rho_{i,0} \neq \rho_{i,1}$ **then**
 Connect i_n of $\max\{\rho_{i,n} : n=0, 1\}$ to pin A
 Connect last i_n to pin B
else
 Connect i_n of $\max\{\gamma_{i,n,0} : n=0, 1\}$ to pin B
 Connect last i_n to pin A
end if

Fig. 7. Greedy algorithm for signal flow optimization in adder-tree.

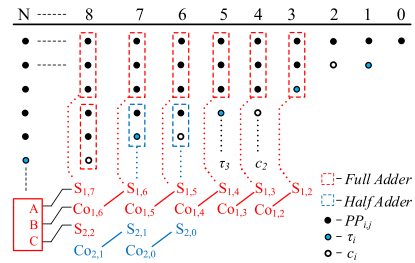


Fig. 8. OWT-carry-save Scheme and PASR for the adder-tree, with reference to Fig. 2(a) ($S_{1,7}$ or $S_{2,2} \rightarrow$ input C and $Co_{2,1} \rightarrow$ input B of PBFA26 as $\rho_S > \rho_{Co}$).

probabilities ρ_i of PPG signals in Fig. 2(a) and sum-carry pairs (ρ_S, ρ_{Co}) in the adder-tree can be generalized in the order: $\rho_S > \rho_{Co} > \rho_{PP} > \rho_{\tau_{i1}} > \rho_{c_i} > \rho_{\tau_{i0}}$ while ρ_{α_i} and $\rho_{\bar{S}_i}$ being the lowest [16]. If the switching information is readily available, a greedy algorithm can be developed for the adder-tree routing as shown in Fig. 7. Note that $\rho_{i,n}$ and $\gamma_{i,n,0}$ represent the toggle count of the input n and the number of occurrences

Algorithm 2 : Optimal Transistor Sizing**Steps 1-4 for 6T-XOR/XNR CUT in Fig. 3(e):**

- 1: **Input:** W_n of the transistors of CUT; $n \in \mathbb{N}$; $1 \leq n \leq 10$
- 2: **Output:** List of W_n for the optimal PDP of CUT
- 3: Assign W_{min} to W_n of all NMOS transistors.
- 4: Assign $W_{min} \times \frac{\beta_{NMOS}}{\beta_{PMOS}}$ to W_n of all PMOS transistors.
- 5: **do**
- 6: Simulate the circuit and compute PDP
- 7: **if** $T_{CD} >$ next critical path **then**
- 8: Up-size W_n of M_n in XOR and XNR critical paths;
- 9: $n_{XOR}=\{1, 9-10\}$, $n_{XNR}=\{6\}$
- 10: **else**
- 11: Balance the τ_{PD} of all paths by adjusting W_n of
- 12: M_n ; $n_{XOR}=\{2-4\}$, $n_{XNR}=\{5, 7-8\}$
- 13: **end if**
- 14: **while** PDP = desired PDP_{min}

Steps 4 and 6 for BED18 CUT in Fig. 3(f):

- 15: **Input:** W_n of the transistors of I3 / I4; $n \in \mathbb{N}$; $1 \leq n \leq 8$
- 16: **Output:** List of W_n to satisfy glitch suppression and the driving strength requirements.
- 17: Assign W_{min} to W_n of all NMOS transistors.
- 18: Assign $W_{min} \times \frac{\beta_{NMOS}}{\beta_{PMOS}}$ to W_n of all PMOS transistors.
- 19: **do**
- 20: Up-size W_n for NMOS/PMOS pairs in the list of M_n ;
- 21: $n=\{1-8\}$ to fine-tune C_{L_I1} and C_{L_I2} .
- 22: Obtain $W_{D_{min}}$ and improve driving strength.
- 23: **while** $(\tau_{XOR-V_{th}} > t_{A \rightarrow \bar{A}} - \tau_{D_B})$ **and** $(\tau_{PD_XOR} \approx \tau_{PD_XNR})$

Steps 5 for PBFA26 CUT in Fig. 5(f):

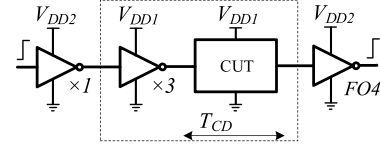
- 24: **Input:** W_n of the transistor M_n ; $n \in \mathbb{N}$; $1 \leq n \leq 4$
- 25: **Output:** List of W_n for M1-M4 SCG synchronization.
- 26: First, synchronize $\tau_{pd_AB \rightarrow xor}$ and $\tau_{pd_AB \rightarrow xnr}$ by adjusting XOR/XNR transistors.
- 27: Assign W_{min} to W_n of all NMOS transistors.
- 28: Assign $W_{min} \times \frac{\beta_{NMOS}}{\beta_{PMOS}}$ to W_n of all PMOS transistors.
- 29: **do**
- 30: Fine-tune W_n of M1-M2 and M3-M4 for equal rise/fall delays at C1 and C2.
- 31: delays at C1 and C2.
- 32: **while** $(\tau_{pd_C \rightarrow C1} \approx \tau_{pd_AB \rightarrow xor} \approx \tau_{pd_AB \rightarrow xnr})$

Fig. 9. Optimal device sizing algorithm for CUTs.

of logic “0” at input n , respectively. If the toggle rates of input signals are comparable, signals of higher γ_{i,n_0} can be interfaced to input B , so that the parasitics of the transmission pair in the XOR stage remain in off-state in most cases. The application of parasitic-aware routing scheme (PASR) in an OWT-carry-save adder-tree is illustrated in Fig. 8. Numbers $0-N$ represent the bit positions of the adder-tree partial products [Fig. 2(a)]. $S_{i,j}$ and $Co_{i,j}$ represent the full/half adder outputs accumulated in carry-save and PASR fashion.

V. TRANSISTOR SIZING FOR OPTIMUM PERFORMANCE

Several strategies for optimal device sizing to obtain minimum power-delay product (PDP) have been presented

Fig. 10. Simulation test setup used for power-delay measurements of all circuits (T_{CD} : critical path delay).

in [49] and [53]. The method proposed in [49] is more comprehensive and considers the dependency of adjacent devices as well. In this work, this dependency has been explicitly formulated in Sections III and IV. In addition to smaller PDP, the conditions for glitch-free operation has also been studied in this work. Considering all this information, the device sizing procedure for major building blocks in this work has been summarized in Fig. 9. Note that CUT and T_{CD} in Fig. 9 represent the circuit under test and the critical path delay of CUT, respectively. The MOSFET current gain β is given by $0.5 \mu C_{ox} (V_{DD} - V_{th})^2$ for typical operation and $L = L_{min}$ (60 nm).

VI. EVALUATION AND COMPARISON

This section evaluates both cell level and top level merits of the proposed techniques against the state of the art. All the evaluations are based on the post-layout simulations in 65-nm bulk CMOS process technology. The simulation test setup for the circuits is depicted in Fig. 10 (CUT). The power consumption of the input drivers is excluded from the total power for better accuracy. The circuit blocks were implemented in Cadence Virtuoso environment and the SPICE simulations were carried out in Cadence Spectre tool.

A. Booth Encoder-Decoder Performance

The baseline circuits presented in Fig. 3(a)–(c) were considered for this evaluation. Device sizing is performed with respect to the minimum drawn W/L values (0.15/0.06 μm) and using the algorithm in Fig. 9. For a fair comparison, the Booth encoder/decoders were arranged to generate an even number of partial product bits in all simulations, satisfying BED13 requirement [46]. Fig. 11 summarizes the figures of merits of these circuits when each encoder circuit drives two decoders at different operating conditions. The stimuli to the CUT consisted of a uniformly distributed random (UDR) test sequence of 5000 patterns. The propagation delay of the circuit was measured from the inputs (i.e., b_{2i}) of the encoder to the output (PP_{ij}) of the decoder covering the critical path of each circuit.

In this arrangement, the encoder and decoder loading conditions are minimal and correspond to FO2 and FO4, respectively. As can be seen, the proposed BED18 version reports the lowest power consumption across a wider supply voltage range. This corresponds to 22.7% of reduction at typical supply voltage and 16.7% at near- V_{th} voltage levels. When it comes to the propagation delays, BED20 outperforms BED18 thanks to its transmission gates. As depicted in Fig. 11(b), the maximum performance improvement of BED20 over BED18 can be seen at 0.6 V which is $\sim 9\%$.

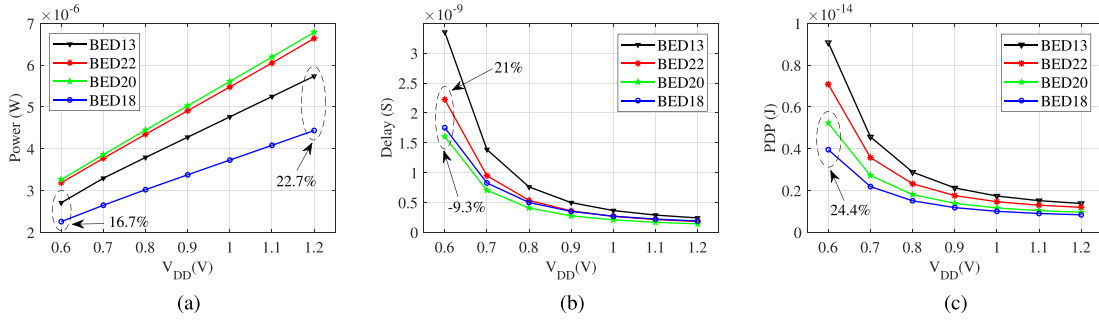


Fig. 11. Power (at 250 MHz), delay, and PDP of the Booth circuits: Fig. 3(a)–(c) and (e) in typical conditions (TT/at 25 °C/encoder-FO2/decoder-FO4).

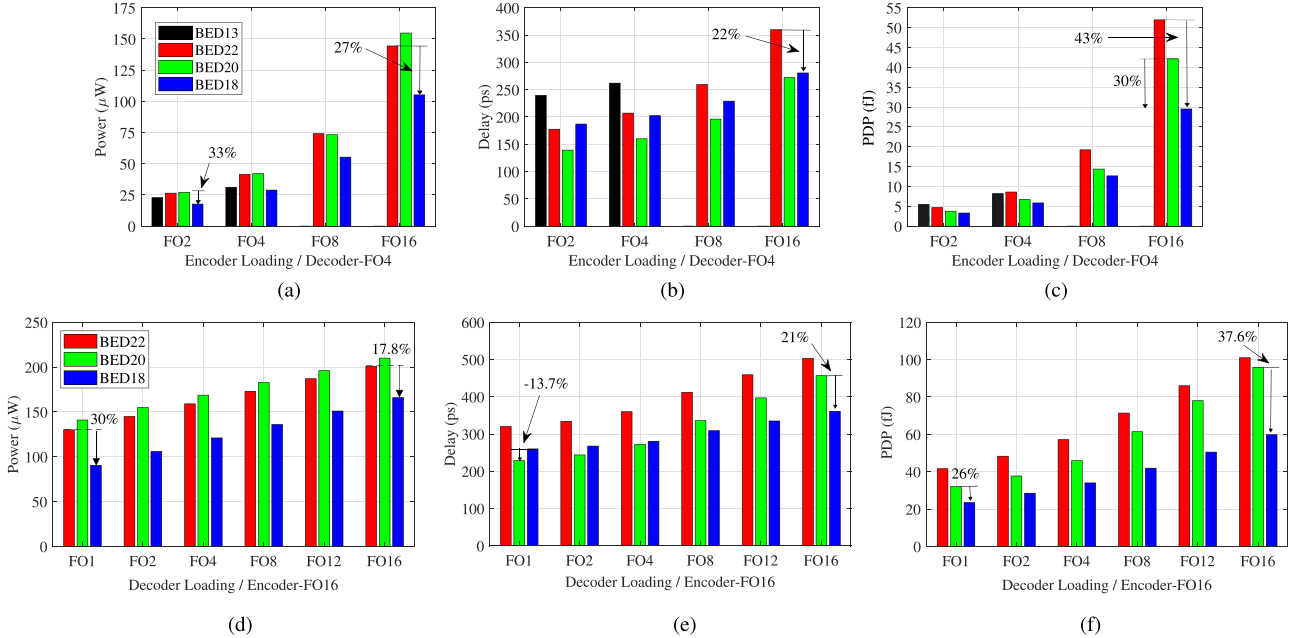


Fig. 12. Power, delay, and PDP of Booth circuits (1.2 V/TT/1 GHz/at 25 °C). (a)–(c) Different encoder loading conditions. (d)–(f) Different decoder loading conditions.

However, the larger switching capacitance and extra wiring of the BED20 circuit overshadows its higher performance and consumes ~24% more energy than BED18 at 0.6 V as shown in Fig. 11(c).

Power, delay, and PDP of these circuits against different loading scenarios under typical operating conditions ($V_{DD} = 1.2$ V/at 1 GHz/TT/at 25 °C) are illustrated in Fig. 12(a)–(f). This depiction reflects the effect of the loading conditions on the integrity of the encoder/decoder signals at a higher performance level. Here FO4, FO8, and FO16 represent the loading conditions of a single encoder in 4-, 8-, and 16-bit multiplier arrangements. Fig. 12(a)–(c) evidently suggests that the proposed BED18 has the most energy-efficient figures compared to all baselines. Interestingly, the propagation delay of BED20 degrades at FO16 and becomes comparable to BED18. This is caused by the higher input capacitive load of BED20 decoder (OAI32 cell). Under the same condition, the delay of BED22 increases by 22% compared to BED18. The lack of driving strength of encoder (AOI33) as well as the accumulated input loads of decoder cell (AOI22) were found to be the causes of this. Notably, at 1 GHz, BED13 fails to sustain its operation when the encoder loading \geq FO8. This

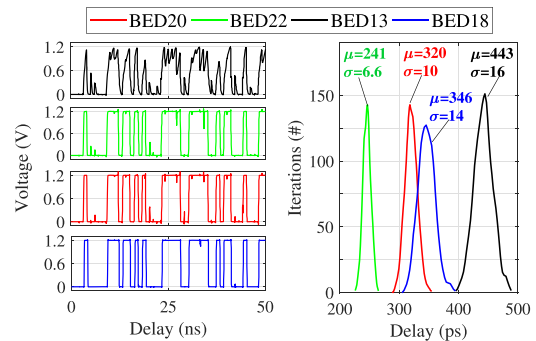


Fig. 13. PP_{ij} waveforms (encoder-FO16/decoder-FO4) and process/mismatch variations in Booth circuits (1.08 V/SS/at 125 °C/encoder-FO2/decoder-FO4).

is mainly due to the cascaded and imbalanced resistive paths of the *SEL* blocks as foreshadowed in Section III.

Fig. 12(d)–(f) depicts the circuit performance against different decoder loading conditions when the encoder load is FO16 under the same operating conditions as was in Fig. 12(a)–(c). BED18 circuit outperforms all other circuits in both power and delay figures under heavily loaded conditions. This advantage is evidently seen in its energy

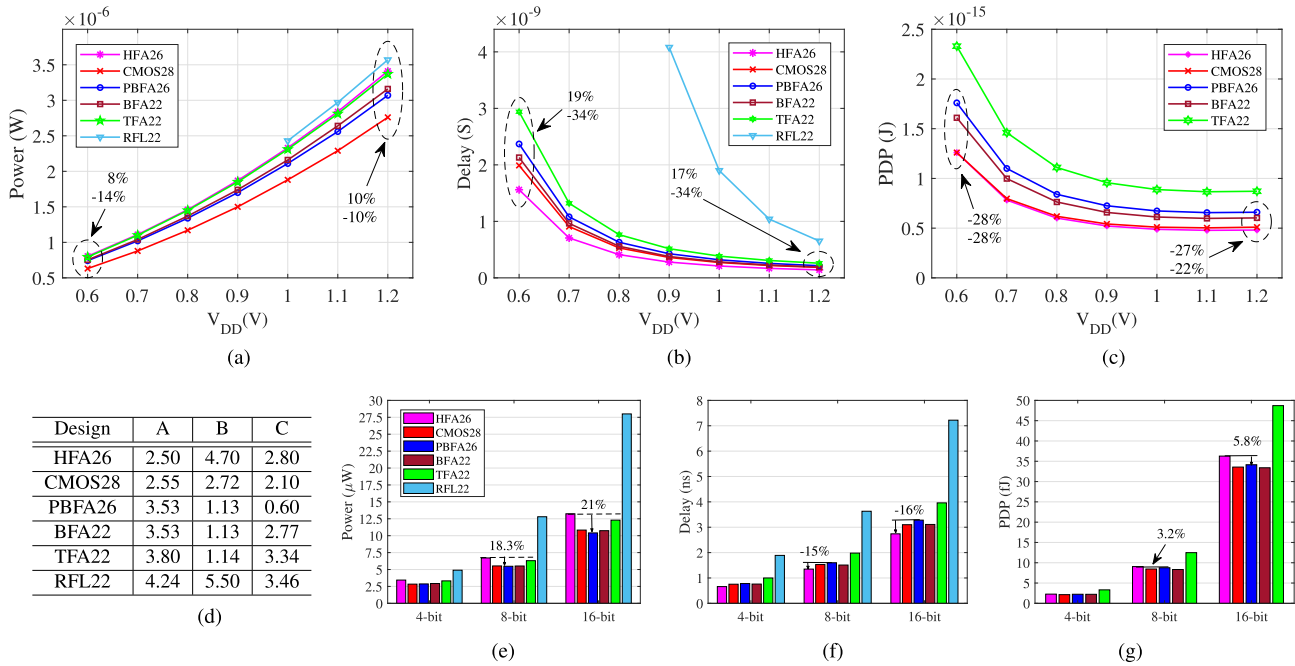


Fig. 14. (a)–(c) Power (at 320 MHz), delay, and PDP of full adders (TT/at 25 °C/ C_L =FO4 at S and C_o). (d) Worst-case input capacitance in fF. (e)–(g): Worst-case power, delay, and PDP of cascaded full adders under the same operating conditions.

consumption, reporting $\sim 26\%$ – 37% reduction compared to BED20. Furthermore, this indicates that the driving strength of the proposed decoder is more than sufficient, even though smaller devices were used. The output signals of the Booth circuits for Fig. 12(c) loading condition are illustrated in Fig. 13 (left). As shown, the degradation of BED13 output is prominently seen in heavily loaded conditions while all other circuits perform reasonably well.

Fig. 13 (right) also outlines the process and mismatch variations on the Booth circuits based on 1000 Monte Carlo iterations (MCIs). In this case, the loading condition is minimal and corresponds to the FO2 condition of Fig. 11(b). A similar behavior in delays can be observed in both graphs. BED13 reports the largest delay with the highest standard deviation. The variation of BED22 is slightly better than BED18 due to the fewer number of transistors in BED22 decoder. Even though BED20 has more transistors in its decoder, the reduced encoder critical paths result in less variations. Nonetheless, its higher power consumption overshadows this merit.

B. Full Adder Evaluation

All the full adders presented in Fig. 5 are considered to demonstrate their immunity to the spurious activities along with the typical figures of merit. To highlight the geometry independent qualities of each, all NMOS and PMOS devices were equally sized to minimum drawn values ($W_{NMOS} = W_{min}$, $W_{PMOS} = \beta W_{NMOS}$, where $\beta = 1.5$ and $W_{min} = 0.15 \mu\text{m}$). This results $0.15/0.060 \mu\text{m}$ for NMOS and $0.22/0.060 \mu\text{m}$ for PMOS. This sizing suits for $M1$ – $M4$ of PBFA26 as well. A full adder in the adder-tree always drives a single input of another full adder cell which is typically less than a FO4 load. Hence minimum-sized devices can be used. Fig. 14(a)–(c)

illustrates power, delay and PDP of each circuit against V_{DD} under typical conditions (TT/at 25 °C/ C_L =FO4). The power consumption of the full adders is the average power observed for their all 56 possible input transitions [49], [53] at 320-MHz operation.

As was mentioned, the RFL22 is the slowest design [Fig. 14(b)]. When the supply voltage goes below 1 V, the circuit fails to meet the constraints. It consumes more power at the same time in all operating points leading to impractical PDP values. Hence, the fitness of RFL22 in this scope is questionable. Interestingly, the standard CMOS28 full adder cell reports the lowest power consumption which is 14% and 10% lower than proposed PBFA26 at near- V_{th} and nominal supply levels. PBFA26 is slightly better ($\sim 3\%$) than BFA22 even though the latter requires only 22 transistors. And BFA22 is followed by TFA22, HFA26, and RFL22 consuming 9%, 10%, and 14% more power than PBFA26, respectively, at 1.2 V. When it comes to the propagation delays, PBFA26 only outperforms TFA22 (by 17%–19%) and RFL22. Notably HFA26 even outperforms CMOS28 and becomes 34% faster than PBFA26 in both supply domains. Similarly, these merits lead CMOS28 and HFA26 to be more energy efficient than proposed PBFA26 as shown in Fig. 14(c) in standalone simulation. However, this is not the case for cascaded and multiplier arrangements which will be discussed later in this section.

Fig. 14(d) summarizes the worst-case capacitance of each full adder input observed for all possible input combinations. RFL22 reports the largest capacitance values for all three inputs. In HFA26, input B sees the largest capacitance which is $\sim 1.8\times$ of other inputs. Inputs A and B of PBFA26 and BFA22 have the same values as their input arrangement is similar. The input A of these full adders tends to see a

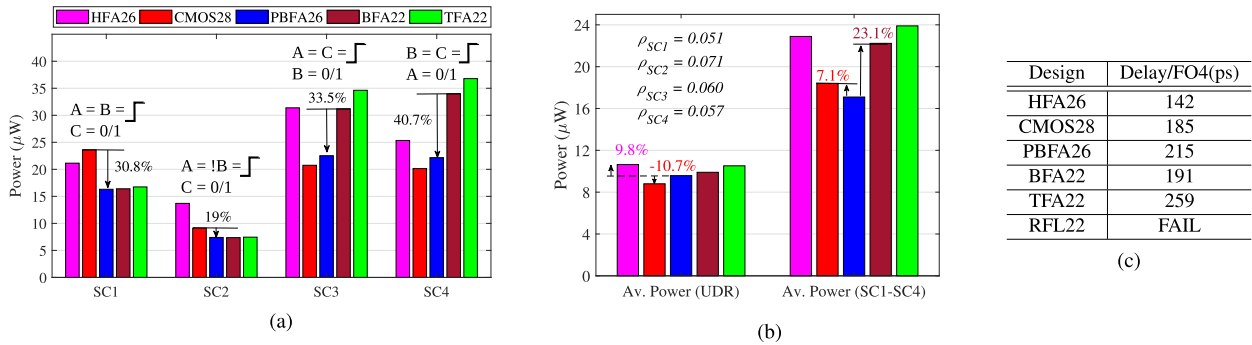


Fig. 15. Full adders. (a) Average power: glitchy input scenarios (SC1–SC4). (b) Average power: UDR versus SC1–SC4. (c) Propagation delays at 1.2 V (1.2 V/1 GHz/TT at 25 °C/CL = FO4 at S and Co).

larger capacitance than the CMOS28 due to the slow rise/fall short-circuit transitions at the input stage. This phenomena worsens in TFA22, RFL22, and HFA26 (input B). As was predicted in Section IV-B, input C of PBFA26 sees the smallest input capacitance and this will be leveraged to reduce the dynamic power dissipation in the PASR algorithm. The probability of a full adder output to drive a higher capacitive load in the adder-tree is relatively lower in PBFA26 design. Although the individual PBFA26 is slower, this probability should lead to an improvement in PDP for a cascaded circuit.

Fig. 14(e)–(g) illustrates the effect of cascading of full adders in 4-, 8-, and 16-bit arrangements. The full adders can be cascaded in six different modes [49]. Among these, the propagation delay of the slowest mode of each arrangement is depicted in Fig. 14(f) and used for PDP in Fig. 14(g). Fig. 14(a)–(d) evidently suggests that PBFA26 is more power efficient than HFA26 and this is similarly observed in cascaded modes leading to 18%–21% savings in 8- and 16-bit modes. Interestingly, the most power efficient design CMOS28 (10.83 μW) consumes slightly higher power ($\sim 3.88\%$) than PBFA26 (10.4 μW) in 16-bit cascaded mode. HFA26 outperforms all other designs in propagation delays and as depicted in Fig. 14(f) it is $\sim 16\%$ faster than PBFA26 in 8-/16-bit modes. When it comes to PDP, CMOS28 (33.6 fJ) and BFA22 (33.4 fJ) become comparable to PBFA26 (34.1 fJ) while PBFA26 is 6% more energy efficient than HFA26 (36.3 fJ). The main reason for this slight improvement of PBFA26 in cascaded mode can be related to its lower input capacitance and balanced data-paths in general. TFA22 and RFL22 were found to be the worst designs among the full adders.

Fig. 15(a) depicts the power consumption of full adders (except RFL22) for specific input scenarios (SC1–SC4) which lead to self-emancipated glitches in these adders. The nature of these scenarios are illustrated in Fig. 15(a). For instance, scenario SC1 implies that the inputs A and B of the full adder have simultaneous transitions while input C is at a constant logic level. SC2 is similar to SC1 except the transitions at inputs A and B are in opposite directions. As can be seen, CMOS28 is sensitive to SC1 and SC2 due to its inherent XOR/XNR path imbalance. In both of these scenarios, PBFA26, BFA22, and TFA22 perform well as their XOR/XNR paths are more balanced. In SC1, CMOS28 power consumption (23.6 μW) can go up to 30.8% of PBFA26’s

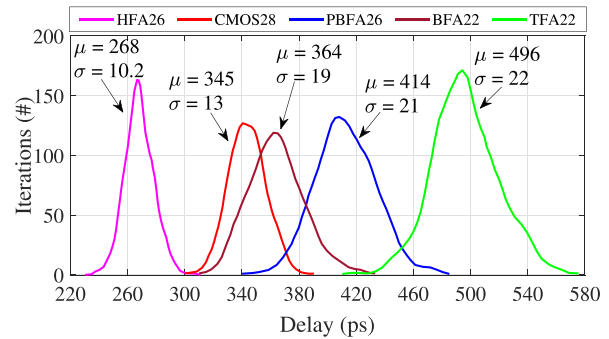


Fig. 16. Full adder delays against PVT variations at SS corner (1.08 V/SS/at 125 °C/CL-FO4).

consumption (16.31 μW). However, this behavior flips in SC3 and SC4. Instead of CMOS28, power consumption of BFA22 (31.2–33.9 μW) increases by 33%–40% compared to PBFA26 (22.52–22.16 μW) due to the poor synchronization between carry-in (C) and other two inputs. TFA22 shows the highest power consumption for SC3 and SC4. Recall that the only structural difference between PBFA26 and BFA22 is the intracell delay element which demonstrates its impact to the power consumption in SC3 and SC4 scenarios.

Fig. 15(b) compares the average power consumption between a UDR stimuli (of 1000 patterns) and SC1–SC4. Obviously, the probability of the occurrence of SC1 and SC2 ($\rho_{SC1} - \rho_{SC1}$) is much lower in UDR and therefore CMOS28 (8.7 μW) is shown to be 10.7% more power efficient than PBFA26 (9.75 μW). Even though the probabilities of SC3 and SC4 are similarly lower, PBFA26 is generally more power efficient than HFA26 (10.81 μW) and BFA22 (9.94 μW) as was observed in Fig. 14(a). This corresponds to 9.8% and 2% of savings, respectively. On average, proposed PBFA26 (17.1 μW) consumes 7.1% and 23% less power than CMOS28 (18.4 μW) and BFA22 (22.2 μW) in specific scenarios such as SC1–SC4. Fig. 15(c) summarizes the propagation delays at 1.2-V/1-GHz operation where RFL22 fails to survive. The propagation delays of the other designs closely follow the attributes of Fig. 14(b).

Fig. 16 summarizes the PVT variations of full adders at worst case corner (1.08 V/SS/125 °C) operation based on 1000 MCIs. HFA26 demonstrates better delay variations compared to other designs. HFA26, CMOS28, and TFA22 reported four samples beyond $\mu \pm 3\sigma$ limits while

PBFA26 and BFA22 reported only 3. RFL22 (not shown) demonstrated the worst case performance of $\mu = 2.7$ ns/ $\sigma = 0.683$ ns while having ten outliers beyond $\mu \pm 3\sigma$ limit. PBFA26 has shown 12% higher μ than BFA22 due to the variations introduced by its intracell delay element. Their σ values are somewhat comparable.

C. Multiplier Performance

This section presents the figures of merit of the proposed glitch-optimized circuit blocks in the multiplier integration. Based on the conclusions drawn in Sections III and IV, this article proposes two multiplier structures Prop-W, Prop-LFR with OWT, and LFR reduction schemes for full-precision operation. The sum-carry interconnections of the adder-tree were arranged according to the generalized version of the PASR algorithm presented in Section IV-B. The proposed Booth encoder/decoder in Fig. 3(f) (BED18) and the full adder in Fig. 5(f) (PBFA26) are utilized in these designs. For the comparison, six baselines were constructed for 16- and 32-bit versions based on OWT, Array, and LFR schemes utilizing BED22 in Fig. 3(c) and the most promising full adders in Fig. 5(d) (HFA26) and Fig. 5(e) (CMOS28). These two full adders produce C_o output faster than output S and therefore, they are well suited for OWT, Array, and LFR schemes. The 8-bit version scarcely benefits from the Array and LFR schemes so that it is only limited to OWT. The OWT and Array versions were further optimized with TDM. Applying TDM to LFR is not considered as the gain is expected to be minimal [28], [29]. Moreover, the traditional “Array-Only” versions are not considered due to their inferior performance [7]. The regular PPG structure in Fig. 2(a) [16] and a two-level CLA adder as the final adder were utilized in all variants which can be summarized as follows.

- 1) *Base-W1(TDM)*: Baseline with OWT, TDM schemes and BED22, CMOS28 circuits.
- 2) *Base-W2(TDM)*: Baseline with OWT, TDM schemes and BED22, HFA26 circuits.
- 3) *Base-AR1(TDM)*: Baseline with Array, TDM schemes and BED22, CMOS28 circuits.
- 4) *Base-AR2(TDM)*: Baseline with Array, TDM schemes and BED22, HFA26 circuits.
- 5) *Base-LFR1*: Baseline with LFR schemes and BED22, CMOS28 circuits.
- 6) *Base-LFR2*: Baseline with LFR schemes and BED22, HFA26 circuits.
- 7) *Prop-W*: Proposed version with OWT, PASR schemes and BED18, PBFA26 circuits.
- 8) *Prop-LFR*: Proposed version with LFR, PASR schemes and BED18, PBFA26 circuits.

1) *Experimental Setup*: The cells have been characterized with intracell parasitics for the physical design experiment. All the multipliers were fully placed and routed in Cadence Innovus digital environment to account for both gate-level and interconnect-level parasitics. To preserve the logic structure of the design, the optimizations in the physical design tool were disabled. Due to the cumbersome nature of the transistor level simulations on the entire design, the power consumption was observed in two steps. First, the power consumption

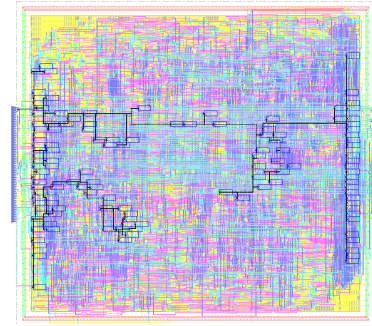


Fig. 17. 32-bit Prop-W multiplier (clock network highlighted).

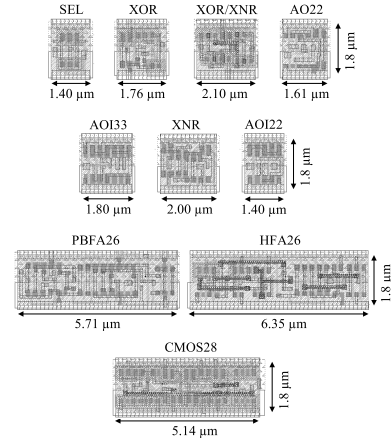


Fig. 18. SEL of BED13, XOR-XOR/XNR-AO22 of BED18, AOI33-XNR-AOI22 of BED22 and full adders.

of each design with intracell parasitics was observed by transistor-level simulations in Cadence Virtuoso analog environment. This is essential as the impact of the spurious activities can be accurately reflected. Secondly, the impact of the interconnect parasitics to the total power consumption was quantified in Synopsys PrimeTime environment by running relatively larger test patterns. Finally, the numbers obtained from transistor-level simulations were scaled up by the average percentage of the interconnect power contribution obtained from PrimeTime.

In the analog environment, each design was simulated against a UDR test sequence of 5000 input patterns, given by P1. Besides, 16-bit versions were simulated against P2, a realistic data set that was extracted from the JPEG decoding benchmark (*djpeg*) application from MediaBench Suite [54]. Typically in lossy JPEG decompression, 8-bit sampled data with recommended scaling requires only 16-bit wide variables and constants. The extracted sequence includes 50000 input patterns, which represent 16-bit multiplication data involved in inverse discrete cosine transform (IDCT) calculations for 8-bit JPEG image samples. Tables I–III summarize the figures of merit of the different multiplier versions considered in this evaluation. Note that these figures only correspond to the combinatorial portion of the multipliers. *Delay* and *Inter. Power* columns represent the critical path delay and the contribution of interconnect parasitics to the total power consumption respectively. The physical layout of the 32-bit Prop-W and the cell layouts of the proposed circuits are shown in Figs. 17 and 18.

TABLE I
8-bit MULTIPLIER PERFORMANCE (1.2 V, 500 MHz, TT, 25 °C)

Scheme	Input	Power _{tot} (μW)	Power _{leak} (μW)	Delay (ns)	Inter. Power	Core Area (μm ²)
Base-W1(TDM)	P1	344 (17.4%)	0.068	1.32 (6.0%)	11.0%	1043 (0.7%)
Base-W2(TDM)	P1	353 (19.4%)	0.074 (8.1%)	1.24	9.2%	1085 (4.5%)
Prop-W	P1	284	0.069 (1.4%)	1.42 (12.7%)	8.5%	1036

TABLE II
16-bit MULTIPLIER PERFORMANCE (1.2 V, 250 MHz, TT, 25°C)

Scheme	Input	Power _{tot} (μW)	Power _{leak} (μW)	Delay (ns)	Inter. Power	Core Area (μm ²)
Base-W1(TDM)	P1	703 (12.2%)	0.204 (3.4%)	2.25 (2.2%)	15.2%	3224 (0.8%)
	P2	507 (10.5%)				
Base-W2(TDM)	P1	718 (14.0%)	0.240 (17.9%)	2.20	13.0%	3471 (7.8%)
	P2	520 (12.7%)				
Prop-W	P1	616	0.228 (13.6%)	2.39 (7.9%)	14.0%	3294 (2.9%)
	P2	454				
Base-AR1(TDM)	P1	770 (20.0%)	0.197	2.50 (12.0%)	15.6%	3198
	P2	546 (16.8%)				
Base-AR2(TDM)	P1	767 (19.6%)	0.240 (17.9%)	2.21 (0.5%)	13.0%	3473 (7.9%)
	P2	551 (17.6%)				
Base-LFR1	P1	716 (13.8%)	0.204 (3.4%)	2.42 (9.0%)	15.3%	3253 (1.7%)
	P2	515 (11.8%)				
Base-LFR2	P1	744 (17.2%)	0.240 (17.9%)	2.38 (7.6%)	14.0%	3481 (8.1%)
	P2	534 (15.0%)				
Prop-LFR	P1	710 (13.2%)	0.228 (13.6%)	2.75 (20.0%)	15.5%	3294 (2.9%)
	P2	479 (5.2%)				

TABLE III
32-bit MULTIPLIER PERFORMANCE (1.2 V, 200 MHz, TT, 25 °C)

Scheme	Input	Power _{tot} (μW)	Power _{leak} (μW)	Delay (ns)	Inter. Power	Core Area (μm ²)
Base-W1(TDM)	P1	2462 (9.5%)	0.590	4.16 (3.6%)	18.6%	10930 (1.5%)
Base-W2(TDM)	P1	2558 (12.9%)	0.750 (21.3%)	4.01	15.7%	12024 (10.4%)
Prop-W	P1	2228	0.712 (17.1%)	4.22 (5.0%)	15.0%	11310 (4.8%)
Base-AR1(TDM)	P1	3093 (27.9%)	0.631 (6.5%)	4.92 (18.5%)	18.0%	10771
Base-AR2(TDM)	P1	3082 (27.7%)	0.797 (26.0%)	4.55 (11.9%)	15.1%	11862 (9.2%)
Base-LFR1	P1	2757 (19.2%)	0.651 (9.4%)	4.41 (9.1%)	17.4%	10940 (1.5%)
Base-LFR2	P1	2887 (22.8%)	0.815 (27.6%)	4.64 (13.6%)	17.2%	12022 (10.4%)
Prop-LFR	P1	2687 (17.1%)	0.765 (22.9%)	4.95 (19%)	14.2%	11166 (3.5%)

2) *Power Consumption*: Evidently, the Prop-W version is the most power-efficient design among all the multipliers for both P1 and P2 patterns. This figure is more prominent in 8-bit versions (17%–20% against Base-W1 and Base-W2) as Prop-W significantly benefits from the reduced parasitics and spurious activities in Booth circuits as well as the adder-tree. Recall that the proposed Booth encoder/decoder circuit (BED18) was already shown to be the most energy-efficient design in Section VI-A. In 16- and 32-bit versions, the growth of the adder-tree surpasses the complexity of Booth circuits. Therefore, the power efficiency gradually decreases to 12%–14% for 16 bit and 10%–13% for 32-bit versions when compared with Base-W1 and Base-W2, respectively. An exception can be observed for other baselines. Notably, Prop-W outperforms Array-TDM and LFR versions by nearly 20%–30%. This implies that, even with the imbalanced full adders, the combined effort of OWT and TDM of Base-Wx(TDM) has severed the baseline versions reasonably well in suppressing spurious activities. Needless to say that the self-emancipated glitches for CMOS28 and HFA26 [Fig. 15(a)] still remain in these designs. Even

though the single CMOS28 full adder was found slightly more power-efficient when compared with the proposed PBFA26 full adder [Fig. 14(a)], the efficacy of PBFA26 is pronounced in the multiplier integration. Thanks to OWT and PSR schemes, the parasitic behavior has been efficiently exploited to address the spurious activities. Even though Prop-LFR benefits from LFR and PSR schemes, it only outperforms baseline Array(TDM) and LFR versions. More specifically, Prop-LFR 32-bit version is 11.3%–11.6% and 0.8%–5.3% power efficient when compared to Base-ARx(TDM) and Base-LFRx 32-bit versions, respectively.

The leakage power of the multiplier is related to the worst-case leakage power of its cells. The lowest leakage was observed for the multiplier variants of CMOS28 full adder cells. This is obvious as the stacked CMOS devices in CMOS28 produce less leakage. This figure is followed by the PBFA26 and finally the variants of HFA26. Despite the similar transmission-gate structure between PBFA26 and HFA26, the signal synchronization provided by the intracell delay element [Fig. 5(f)] of PBFA26 further reduces the leakage currents when the transmission pairs switch in SCG.

For instance, Base-W1(TDM) of CMOS28 cells shows 17% and 21% better leakage figures than Prop-W of PBFA26 and Base-W2(TDM) of HFA26 for 32-bit versions.

Another interesting observation of the experiment is the layout complexity. Even though the Wallace and OWT schemes were considered notorious for layout power [7], [17], [28], [29], [44], the contribution of layout parasitics of OWT and PASR for the high-performance operation was found negligible compared to other schemes. The interconnect power values across OWT, Array and LFR schemes are almost comparable. An equal effort was put to optimize the layouts of all the cells by limiting the intracell routing to metal-1 (ME1) layer. Moreover, the pins of the cells were placed carefully on the ME1-ME2 routing grid, so that the routing tool has sufficient room to access the pins. These facts, as well as the stringent timing constraints (i.e., for 32-bit multipliers), are indeed the cause for these comparable interconnect figures. More specifically, the 32-bit version of Base-W1(TDM) required 96 nets in ME6 layer while Base-W2(TDM) and Prop-W required 26 and 19 nets, respectively. Hence a negligible increase of 3% in interconnect power of Base-W1(TDM) can be observed. The metal use is lowest in 32-bit Prop-LFR version in which only ME1–ME5 nets have been used.

3) *Propagation Delays*: The proposed designs based on OWT and PSR are slightly slower than other OWT versions and are indeed faster than the Array(TDM) and LFR versions. The multipliers of HFA26 full adder are the fastest variants and they are followed by CMOS28 variants. As was highlighted in Section VI-B, both HFA26 and CMOS28 full adders are faster than PBFA26 [Fig. 14(b)]. However, when the full adders are in cascaded mode (i.e., 8 and 16 bit), the asymmetry of the input capacitive loading and other parasitic effects average out the total delay of the cascaded chain [as was in Fig. 14(f)] and therefore the delay difference becomes negligible. Similarly, the performance of Prop-W version improves for higher word lengths. Moreover, the proposed Booth circuit (BED18) has always been faster than the baseline (BED22) for wider loading conditions as was depicted in Fig. 12(b) and (e). Hence the partial products ($PP_{i,j}$) of the proposed versions arrive faster at the front-end of the adder-tree. For instance, 8-bit Base-W2(TDM) is 12.7% faster than Prop-W and it reduces to 5% for 32-bit version mode. Naturally, the OWT with TDM or with PSAR outperforms all Array(TDM) and LFR versions pertaining to its logarithmic delay reduction. Prop-LFR is the slowest version among all the variants.

4) *Core Area*: The multiplier variants of HFA26 full adder consume relatively a larger core area while CMOS28 variants are the smallest. More specifically, for a given word length, Base-AR1(TDM) of CMOS28 full adders, reported the smallest core area. This is 2.9% and 4.8% area efficient than the proposed versions for 16- and 32-bit versions, respectively. Base-W1(TDM) and Prop-W are comparable in 8-bit versions. The continuous diffusion connections of the standard CMOS style (of CMOS28) is the main reason for this observation. On the contrary, both HFA26 and PBFA26 based on alternative logic styles require a relatively larger layout area due to the diffusion breaks of the cells. Note that in addition to these

functional cells, the final layout contains physical-only cells (well continuity and metal fillers) as well.

VII. CONCLUSION

This article has proposed and investigated glitch-optimized circuit blocks for high-performance Booth multipliers aiming to reduce the dynamic power dissipation caused by the parasitics and spurious activities. The proposed strategy incorporates circuit-level techniques with a PASR to achieve this goal. Therefore, the proposed approach is an excellent choice for high-performance, energy-constrained multiplication at the expense of a slightly higher delay. The efficacy of the proposed strategies has been verified by the extensive postlayout simulations carried out in 65-nm process technology. Two versions of the multiplier structures (Prop-W, Prop-LFR) comprising these circuit blocks, have been compared to highly optimized array and tree versions of the multipliers comprised of the state-of-the-art building blocks in literature. From the postlayout simulations, it was concluded that the proposed versions are on average 10%–30% more power efficient compared to the baselines.

ACKNOWLEDGMENT

The authors would like to thank Bert Helthuis from CAES Group for providing technical assistance. Some of the experimental data reported in this article can be accessed at DOI: 10.21227/aeqk-7j60.

REFERENCES

- [1] A. D. Booth, "A signed binary multiplication technique," *Quart. J. Mech. Appl. Math.*, vol. 4, no. 2, pp. 236–240, 1951.
- [2] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electron. Comput.*, vol. EC-13, no. 1, pp. 14–17, Feb. 1964.
- [3] L. Dadda, "Some schemes for parallel multipliers," *Alta Frequenza*, vol. 34, no. 5, pp. 349–356, Mar. 1965.
- [4] E. L. Braun, *Digital Computer Design: Logic, Circuitry, and Synthesis*. New York, NY, USA: Academic, 2014.
- [5] C. R. Baugh and B. A. Wooley, "A two's complement parallel array multiplication algorithm," *IEEE Trans. Comput.*, vol. C-100, no. 12, pp. 1045–1047, Dec. 1973.
- [6] D. Hampel, K. E. McGuire, and K. J. Prost, "CMOS/SOS serial-parallel multiplier," *IEEE J. Solid-State Circuits*, vol. SSC-10, no. 5, pp. 307–313, Oct. 1975.
- [7] Z. Huang and M. D. Ercegovac, "High-performance low-power left-to-right array multiplier design," *IEEE Trans. Comput.*, vol. 54, no. 3, pp. 272–283, Mar. 2005.
- [8] J. Prummel *et al.*, "A 10 mW Bluetooth low-energy transceiver with on-chip matching," *IEEE J. Solid-State Circuits*, vol. 50, no. 12, pp. 3077–3088, Dec. 2015.
- [9] J. Fadavi-Ardekani, "M×N Booth encoded multiplier generator using optimized Wallace trees," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 1, no. 2, pp. 120–125, Jun. 1993.
- [10] N. Itoh, Y. Naemura, H. Makino, Y. Nakase, T. Yoshihara, and Y. Horiba, "A 600-MHz 54×54-bit multiplier with rectangular-styled Wallace tree," *IEEE J. Solid-State Circuits*, vol. 36, no. 2, pp. 249–257, Feb. 2001.
- [11] V. G. Oklobdzija, D. Villeger, and S. S. Liu, "A method for speed optimized partial product reduction and generation of fast parallel multipliers using an algorithmic approach," *IEEE Trans. Comput.*, vol. 45, no. 3, pp. 294–306, Mar. 1996.
- [12] P. F. Stelling, C. U. Martel, V. G. Oklobdzija, and R. Ravi, "Optimal circuits for parallel multipliers," *IEEE Trans. Comput.*, vol. 47, no. 3, pp. 273–285, Mar. 1998.
- [13] A. A. Farooqui and V. G. Oklobdzija, "General data-path organization of a MAC unit for VLSI implementation of DSP processors," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 2, May/Jun. 1998, pp. 260–263.
- [14] N. Petra, D. De Caro, V. Garofalo, E. Napoli, and A. G. M. Strollo, "Truncated binary multipliers with variable correction and minimum mean square error," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 6, pp. 1312–1325, Jun. 2010.

- [15] J.-Y. Kang and J.-L. Gaudiot, "A simple high-speed multiplier design," *IEEE Trans. Comput.*, vol. 55, no. 10, pp. 1253–1258, Oct. 2006.
- [16] S.-R. Kuang, J.-P. Wang, and C.-Y. Guo, "Modified booth multipliers with a regular partial product array," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 5, pp. 404–408, May 2009.
- [17] W. Yan, M. D. Ercegovac, and H. Chen, "An energy-efficient multiplier with fully overlapped partial products reduction and final addition," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 11, pp. 1954–1963, Nov. 2016.
- [18] J. Mori *et al.*, "A 10 ns 54×54 b parallel structured full array multiplier with $0.5 \mu\text{m}$ CMOS technology," *IEEE J. Solid-State Circuits*, vol. 26, no. 4, pp. 600–606, Apr. 1991.
- [19] N. Ohkubo *et al.*, "A 4.4 ns CMOS 54×54 -b multiplier using pass-transistor multiplexer," *IEEE J. Solid-State Circuits*, vol. 30, no. 3, pp. 251–257, Mar. 1995.
- [20] C.-H. Chang, J. Gu, and M. Zhang, "Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 10, pp. 1985–1997, Oct. 2004.
- [21] L.-D. Van and J.-H. Tu, "Power-efficient pipelined reconfigurable fixed-width Baugh-Wooley multipliers," *IEEE Trans. Comput.*, vol. 58, no. 10, pp. 1346–1355, Oct. 2009.
- [22] T. K. Callaway and E. E. Swartzlander, "Power-delay characteristics of CMOS multipliers," in *Proc. 13th IEEE Symp. Comput. Arithmetic*, Jul. 1997, pp. 26–32.
- [23] M. Sjalander and P. Larsson-Edefors, "High-speed and low-power multipliers using the Baugh-Wooley algorithm and HPM reduction tree," in *Proc. 15th IEEE Int. Conf. Electron., Circuits Syst.*, Aug. 2008, pp. 33–36.
- [24] M. Zheng and A. Albicki, "Low power and high speed multiplication design through mixed number representations," in *Proc. Int. Conf. Comput. Design VLSI Comput. Process. (ICCD)*, 1995, pp. 566–570.
- [25] V. G. Moshnyaga and K. Tamaru, "A comparative study of switching activity reduction techniques for design of low-power multipliers," in *Proc. Int. Symp. Circuits Syst. (ISCAS)*, vol. 3, Apr. 1995, pp. 1560–1563.
- [26] E. Costa, S. Bampi, and J. Monteiro, "A new architecture for 2's complement gray encoded array multiplier," in *Proc. 15th Symp. Integr. Circuits Syst. Design*, 2002, pp. 14–19.
- [27] W.-C. Yeh and C.-W. Jen, "High-speed booth encoded parallel multiplier design," *IEEE Trans. Comput.*, vol. 49, no. 7, pp. 692–701, Jul. 2000.
- [28] S. S. Mahant-Shetti, P. T. Balsara, and C. Lemonds, "High performance low power array multiplier using temporal tiling," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 7, no. 1, pp. 121–124, Mar. 1999.
- [29] K. S. Chong, B. H. Gwee, and J. S. Chang, "Low energy 16-bit Booth leapfrog array multiplier using dynamic adders," *IET Circuits, Devices Syst.*, vol. 1, no. 2, pp. 170–174, 2007.
- [30] J.-N. Ohban, V. G. Moshnyaga, and K. Inoue, "Multiplier energy reduction through bypassing of partial products," in *Proc. Asia-Pacific Conf. Circuits Syst.*, vol. 2, 2002, pp. 13–17.
- [31] M. C. Wen, S. J. Wang, and Y. N. Lin, "Low-power parallel multiplier with column bypassing," *Electron. Lett.*, vol. 41, no. 10, pp. 581–583, May 2005.
- [32] O. T.-C. Chen, S. Wang, and Y.-W. Wu, "Minimization of switching activities of partial products for designing low-power multipliers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 3, pp. 418–433, Jun. 2003.
- [33] M. Fujino and V. G. Moshnyaga, "Dynamic operand transformation for low-power multiplier-accumulator design," in *Proc. Int. Symp. Circuits Syst. (ISCAS)*, vol. 5, 2003, p. 5.
- [34] S.-K. Chen, C.-W. Liu, T.-Y. Wu, and A.-C. Tsai, "Design and implementation of high-speed and energy-efficient variable-latency speculating booth multiplier (VLSBM)," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 10, pp. 2631–2643, Oct. 2013.
- [35] Y.-H. Chen, "An accuracy-adjustment fixed-width booth multiplier based on multilevel conditional probability," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 1, pp. 203–207, Jan. 2015.
- [36] X. Cui, W. Liu, X. Chen, E. E. Swartzlander, and F. Lombardi, "A modified partial product generator for redundant binary multipliers," *IEEE Trans. Comput.*, vol. 65, no. 4, pp. 1165–1171, Apr. 2016.
- [37] H. Jiang, J. Han, F. Qiao, and F. Lombardi, "Approximate radix-8 booth multipliers for low-power and high-performance operation," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2638–2644, Aug. 2016.
- [38] E. Antelo, P. Montuschi, and A. Nannarelli, "Improved 64-bit radix-16 booth multiplier based on partial product array height reduction," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 2, pp. 409–418, Feb. 2017.
- [39] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi, "Design of approximate radix-4 Booth multipliers for error-tolerant computing," *IEEE Trans. Comput.*, vol. 66, no. 8, pp. 1435–1441, Aug. 2017.
- [40] Z. Zhang and Y. He, "A low-error energy-efficient fixed-width Booth multiplier with sign-digit-based conditional probability estimation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 2, pp. 236–240, Feb. 2018.
- [41] J. Ding and S. Li, "A modular multiplier implemented with truncated multiplication," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 11, pp. 1713–1717, Nov. 2018.
- [42] B. Razavi, *Design of Analog CMOS Integrated Circuits*. New York, NY, USA: McGraw-Hill, 2002.
- [43] Y. Tividis and C. McAndrew, *Operation and Modeling of the MOS Transistor*, vol. 2. Oxford, U.K.: Oxford Univ. Press, 1999.
- [44] K.-S. Chong, B.-H. Gwee, and J. S. Chang, "A micropower low-voltage multiplier with reduced spurious switching," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 2, pp. 255–265, Feb. 2005.
- [45] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*, vol. 2. Englewood Cliffs, NJ, USA: Prentice-Hall, 2002.
- [46] G. Goto *et al.*, "A 4.1-ns compact 54×54 -b multiplier utilizing sign-select Booth encoders," *IEEE J. Solid-State Circuits*, vol. 32, no. 11, pp. 1676–1682, Nov. 1997.
- [47] X. Wu and F. Prosser, "Design of ternary CMOS circuits based on transmission function theory," *Int. J. Electron.*, vol. 65, no. 5, pp. 891–905, Nov. 1988.
- [48] M. Suzuki *et al.*, "A 1.5-ns 32-b CMOS ALU in double pass-transistor logic," *IEEE J. Solid-State Circuits*, vol. 28, no. 11, pp. 1145–1151, Nov. 1993.
- [49] H. Naseri and S. Timarchi, "Low-power and fast full adder by exploring new XOR and XNOR gates," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 8, pp. 1481–1493, Aug. 2018.
- [50] M. Vesterbacka, "A 14-transistor CMOS full adder with full voltage-swing nodes," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, Oct. 1999, pp. 713–722.
- [51] N. Zhuang and H. Wu, "A new design of the CMOS full adder," *IEEE J. Solid-State Circuits*, vol. 27, no. 5, pp. 840–844, May 1992.
- [52] N. H. E. Weste and K. Eshraghian, "Principles of CMOS VLSI design: A systems perspective," *STIA*, vol. 85, 1985, Art. no. 47028.
- [53] A. M. Shams, T. K. Darwish, and M. A. Bayoumi, "Performance analysis of low-power 1-bit CMOS full adder cells," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 1, pp. 20–29, Feb. 2002.
- [54] C. Lee, M. Potkonjak, and W. H. Mangione-Smith, "MediaBench: A tool for evaluating and synthesizing multimedia and communications systems," in *Proc. 30th Annu. Int. Symp. Microarchitecture*, 1997, pp. 330–335.



Anuradha Chaturanga Ranasinghe (Graduate Student Member, IEEE) received the B.Eng. degree (First Class) in electronic engineering from Sheffield Hallam University, Sheffield, U.K., in 2013, and the M.Sc. degree (Laudatur) in integrated circuits and systems from the University of Turku, Turku, Finland, in 2016. He is currently working toward the Ph.D. degree at the CAES Group, University of Twente, Enschede, The Netherlands.

He briefly worked as an Electronic Design Engineer with Tengri Aero Industries (Pvt) Ltd., Colombo, Sri Lanka, from 2016 to 2018. His expertise and research interests include mixed-signal IC designing, standard cells, and system-on-chip development focusing on ultralow-power operation.



Sabih H. Gerez received the M.Sc. degree (Hons.) in electrical engineering and the Ph.D. degree in applied sciences from the University of Twente, Enschede, The Netherlands, in 1984 and 1989, respectively.

He has been an Assistant Professor with the University of Twente since 1990 (part-time starting from 2001), focusing on research and education in the fields of implementation of digital signal processing, digital integrated circuit design, and design automation. From 2001 to 2009, he was employed by the Cordless Telephony Division of National Semiconductor (called Sitel Semiconductor after 2005 and currently part of Dialog Semiconductor). Since 2009, he runs his business Bibix which offers consultancy services in his mentioned fields of interest. He has authored the book *Algorithms for VLSI Design Automation* (Wiley, 1998).