

Low Transition Encoder and Adder based High Performance Modified Booth Multiplier

Ruksana M, Mr. Rahul M Nair

ECE Department, Nehru College of Engineering & Research Centre, Pampady, Thrissur, Kerala

Abstract: *Multipliers are essential components of digital hardware, ranging from deeply embedded system on-chip (SoC) cores to GPU-based accelerators. The proposed method uses a novel implementation scheme with the essential circuit blocks for high-performance booth multiplier. 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 partial-product. Further, a low transition adder is used to reduce the switching activity of partial product addition. The proposed adder separates the target designs into two parts, i.e., the most significant part and least significant part (MSP and LSP), and turns off the MSP when it does not affect the computation results to save power. Booth multipliers built from proposed strategies were compared to the state-of-the-art versions known from literature and achieve better results in terms of power and performance.*

Keywords: *Booth multiplier, partial product addition, glitch optimized circuit, low transition adder.*

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. While performance remains important, the high demand for battery-powered ubiquitous systems has promoted low-power operation to a primary design goal. However, the majority of proposed high-performance multipliers suffer 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, optimized Wallace-tree (OWT), Dadda, Braun's and three-dimensional minimization (TDM) to speed up 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 or 4-to-2 compressors. 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 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, sign magnitude (SM), and gray coding (GC) 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 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 format conversion logic at both ends of the multiplier. SM implementations 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 was proposed to minimize the extra adder rows for carry summation. The approach has improved the multiplier performance by 25% when compared with the conventional implementations. Kang and Gaudiot presents a fast 2's complement generation circuit to reorganize the partial product array by removing the subsequent carry-in terms. The work 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) and left-to-right 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 demonstrate more balanced data paths and an efficient partial-product array structure that outperform other higher level

implementations. Row and column by passing dynamic operand interchanges 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 exploit the accuracy and the number representation for energy efficiency. Among them, only 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.

II. SOURCES OF DYNAMIC POWER DISSIPATION

The switching of parasitic 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.

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

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 nonlinear time-variant behavior on all parasitic capacitances. The average current consumed during this transition period at the k th terminal can be expressed as

$$C_g = C_{gs} + C_{gd} + C_{gb} \quad C_{g_on} = 2(WL_o C_{ox} + WLC_{ox}/3)$$

$$C_{g_off} = 2WL_o C_{ox} + WLC_{ox} C_{dep}/(WLC_{ox} + C_{dep})$$

$dV_k(t)/dt$ is the slew rate of the signal at terminal k

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

$$C_{s_off} = WL_o C_{ox} + WL_s C_j + (2L_s + W)C_{jsw}$$

$$+ WC_{jsw_c}$$

$$C_{s_on} = 2WLC_{ox}/3 + C_{s_off} \quad C_{d_on} = C_{d_off} = C_{s_off}$$

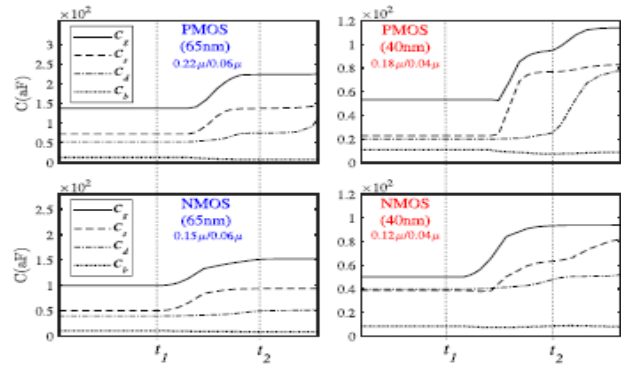


Fig.1 MOSFET Parasitic Capacitance Behavior during the Switching

C_j , C_{jsw} , and C_{jsw_c} correspond to the junction bottom plate and sidewall capacitances. L_s represent 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.

III. 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 , 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 environment by monitoring the narrow pulses that cross the 50% level of V_{DD} .

8-bit version is ~16% of the total glitch count and this becomes prominent in the 16-bit version (~7×) due to the imbalance of the accumulated capacitive loads along the encoder signal lines. Moreover, the encoder ($E0-EN/2$) driving strength required for large operand widths is higher due to the high fan-out nature of the signals $S0-SN/2$. Hence, the delay mismatch among the signals arriving at decoder loads ($D0-DM-1$) is inevitable. In the worst case,

these glitches could propagate till the final adder row.

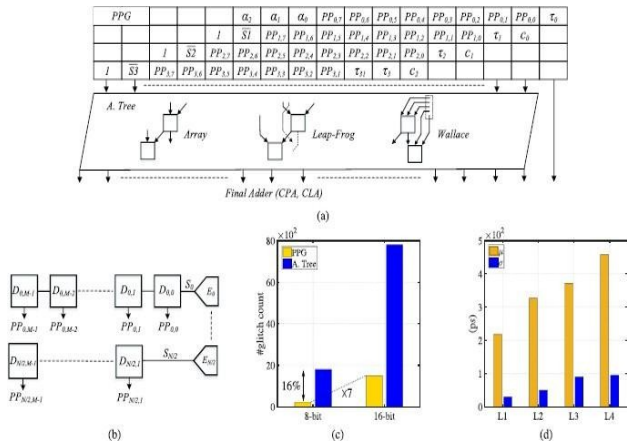


Fig.2 (a) Top-down structure of the 8-bit Booth multiplier [16]. (b) PPG stage. (c) Spurious activities contributed by PPG versus Adder- Tree of the multipliers. (d) Delay variation across the Adder-Tree rows L1–L4 of 16-bit version (μ – maximum delay difference, σ – standard deviation of delays).

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 τD of the cell as follows:

$$V_{out}(t) = V_{DD}(1 - e^{-t/Req Ceq})$$

$$\tau D = Req Ceq \ln \left(\frac{V_{DD}}{V_{DD} - V_{th}} \right)$$

where the total parasitic time constant $Req Ceq$ is given by

$$Req Ceq = RM1 CM1-Poly + (RM1 + Rpoly) CL$$

$RM1$, $RPoly$, $CM1$ -Poly represent the extrinsic metal-1, polysilicon interconnect parasitic resistances and metall-poly via capacitance, respectively. CL corresponds to the intrinsic capacitive load seen at the adder cell input, according to (2). In 65-nm technology, typically $RPoly = 60RM1$ while CL (i.e., Cg) $\approx 4CM1$ -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 τD 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 addertree. 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.

IV. NOVEL CIRCUITS FORMBE

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 block among the presented. However, there are a few issues that emanate from this implementation.

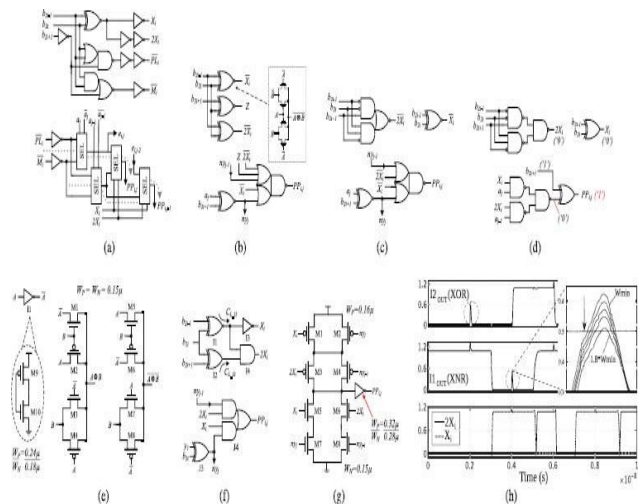


Fig. 3 Various Booth encoder–decoder implementations. (a) BED13 [46]. (b) BED20 [27]. (c) BED22 [7], [16], [41]. (d) Erroneous Booth circuits in [17]. (e) 6T-XOR/XNR circuits of this work ($WM1-M8 = 0.15\mu$). (f) Proposed encoder–decoder circuits (BED18). (g) AO22 ($J3$) of the decoder ($WM1-M4 = 0.16\mu$, $WM5-M8 = 0.15\mu$).

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], 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 "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]:

$$R_D \approx \frac{3}{4} \frac{V_{DD}}{\mu C_{ox} \frac{W}{L} (V_{DD} - V_{th})^\alpha (1 - \frac{2}{9} \lambda V_{DD})}$$

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, $R_{D_RD_NMOS_RD_PMOS}$. 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 $I1$ and $I2$, (4) and (5) can be rewritten to (at 50% of V_{DD})

$$\begin{aligned} \tau_{pd_XNR} &= 0.69(R_{D7}||R_{D8})(C_{s7_on} + C_{d8_on} + C_{OFF_XNR} + C_{L_{I1}}) \\ \tau_{pd_XOR} &= 0.69(R_{INV}(C_{INV} + C_{g3_off} + C_{s1_on}) \\ &\quad + (R_{INV} + R_{D1})(C_{d1_on} + C_{OFF_XOR} + C_{L_{I2}})) \end{aligned}$$

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 $I3$ and $I4$ as shown in Fig. 3(f) alleviates this timing mismatch ($CL_{I1} > CL_{I2}$). 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.

V. 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 XOR0_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. With appropriate device sizing to $M1(M4)$, the required delay can be obtained with the minimal impact to the loading of $M1$ – $M4$ of Fig. 5(f). $M1$ and $M4$ provide the required delay to the input C through their drain–source parasitic 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 C can be independently controlled without a significant overhead. The equivalent RC circuit for $M1$ – $M4$ for condition $C10_1$ is shown in Fig. 6. Similar to (8) and (9), the

synchronization delay required for input C can be expressed as follows:

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

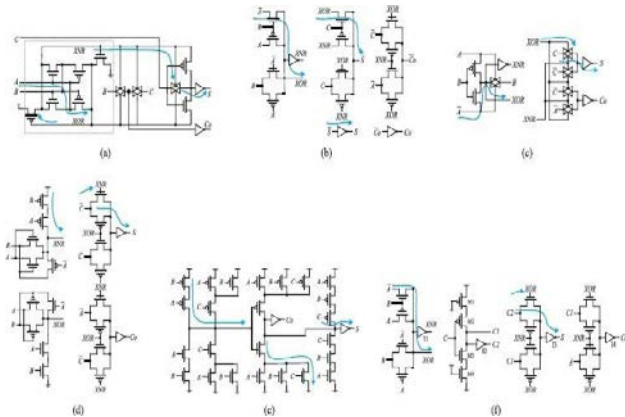


Fig.4 Various low-power, full-swing full adders. (a) RFL22 [50]. (b) TFA22 [51]. (c) BFA22 [52]. (d) HFA26 [49]. (e) CMOS28[44]. (f) Proposed (PBFA26).

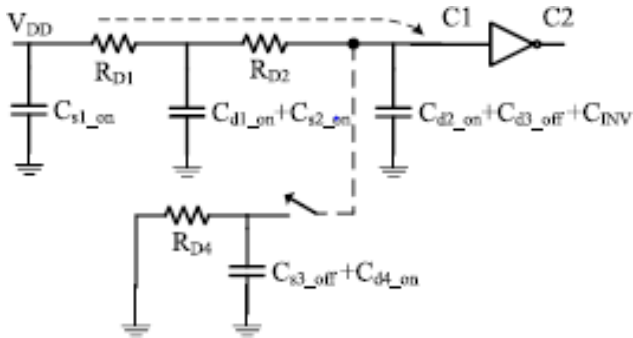


Fig.5 Equivalent RC circuit for M1–M4 of Fig. 5(f) when C_{10_1} . the inverter $M2$ and $M3$, such that $\tau_{pd_C_C1} = \tau_{pd_A,B_xor} = \tau_{pd_A,B_xnor}$.

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 ($_{FO2}$ – $_{FO3}$) are moderately higher than input C ($_{FO1}$), 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 $PA > PB > PC$ 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 $i_n \in \{PP_{ij}, \tau_{i0}, \tau_{i1}, c_i, \bar{S}_i, \alpha_i, S_{ij}, Co_j\}$; $n \in \mathbb{N}$; $0 \leq n < 3$
Output: Optimum i_n assignment to FA; 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.6 Greedy Algorithm for Signal Flow Optimization in Adder-Tree.

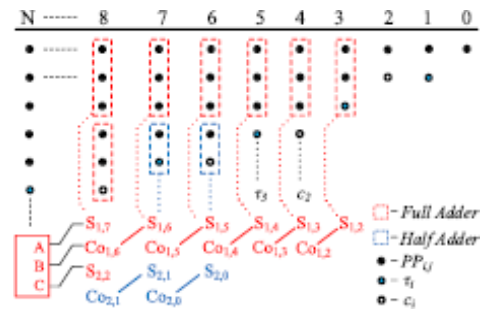


Fig.7 OWT-carry-save Scheme and PASR for the adder-tree, with reference to Fig. 2(a) ($S_{1,7}$ or $S_{2,2}$ _input C and $Co_{2,1}$ _input B of PBFA26 as $\rho_S > \rho_{Co}$) probabilities ρ_i of PPG signals in Fig. 2(a) and sum-carry pairs (ρ_S, ρ_{Co}) in the adder-tree can be generalized in the order: $\rho_S > \rho_{Co} > \rho_{PP} > \rho_{\tau1} > \rho_{ci} > \rho_{\tau10}$ while ρ_{ai} and ρ_{Si} being the lowest [16].

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_{LJ1} and C_{LJ2} .
- 22: Obtain W_{D_min} and improve driving strength.
- 23: **while** ($\tau_{XOR_Vth} > t_{A \rightarrow \bar{A}} - \tau_{D_B}$) **and** ($\tau_{PD_XOR} \approx \tau_{PD_XNR}$)

Fig. 8 Optimal Device Sizing Algorithm for Cuts

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 γ_{i,n_0} represent the toggle count of the input n and the number of occurrences 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 parasitic 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 C_{oi}, j represent the full/half adder outputs accumulated in carry-save and PASR fashion

VI. PROPOSED SPST

Besides the explanations presented in our former studies [14],[15], this paper provides further illustrations of the proposed SPST as described in the following sections.

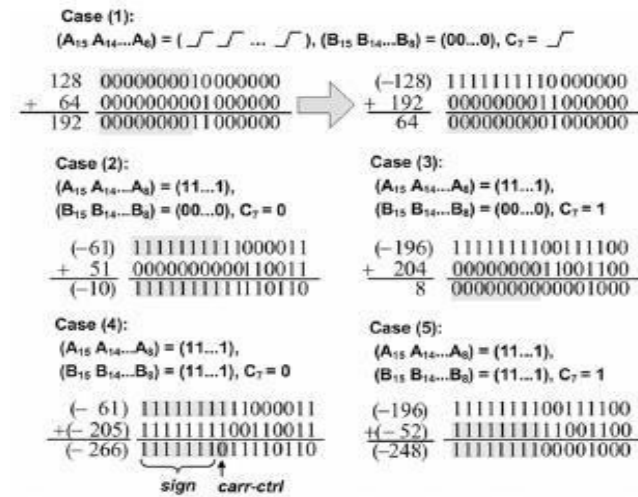


Fig. 9 Spurious Transitions in the Multimedia/DSP Computations

A. Theoretical Analysis and Logic Derivation

To illustrate the reason of those spurious signal transitions shown in Fig. 1, we explore five cases of 16-bit additions as shown in Fig. 3. The cases of exchanging the operands A and B in additions lead to the same spurious transitions with those shown in Fig. 3. Hence, there is probably no other case beyond these five based on this design. The first case illustrates a transient state in which spurious transitions of carry signals occur in the MSP, although the final result of the MSP is unchanged. Meanwhile, the second and third cases describe situations involving one negative operand adding another positive operand without and with carry-in from the LSP, respectively. Moreover, the fourth and fifth cases demonstrate the addition of two negative operands without and with carry-in from the LSP, respectively. In those cases, the results of MSP are predictable; therefore, the computations in MSP are useless and can be neglected. Eliminating those spurious computations not only can save the power

consumption inside the adder/subtractor in the current stage but also can decrease the glitching noises which cause powerwastage inside the arithmetic circuits in the next stage. From the analysis of Fig. 3, we are motivated to propose the SPST that separates the adder/subtractor into two parts and then latches the input data of the MSP whenever they do not affect the computation results. The SPST can be expanded to be a fine-grain scheme in which the adder/subtractor is divided into more than two parts. However, the hardware complexity of the augmented circuits such as the detection-logic unit, the data latches, and the SE unit increases dramatically. Based on an adder/subtractor example, we actually find that the power expense caused by the augmented circuits is larger than the power reduction in a tripartitioned scheme. This is the reason we propose a bipartitioned SPST scheme in this paper. To know whether the MSP affects the computation results in the bipartitioned SPST scheme, a detection-logic unit must be used to detect the effective input ranges. The Boolean logical equations shown as follows express the behavioral principles of the detection-logic unit:

$$A_{MSP} = A[15:8] \quad B_{MSP} = B[15:8] \quad (1)$$

$$A_{and} = A[15] \times A[14] \times \dots \times A[8] \quad (2)$$

$$B_{and} = B[15] \times B[14] \times \dots \times B[8] \quad (3)$$

carr-ctrl	$C_{LSP}, A_{and}, A_{nor}$							
	000	001	011	010	100	101	111	110
B_{and}, B_{nor}	00	0	0	0	0	0	0	0
	01	0	0	0	1	0	1	0
	11	0	0	0	0	0	0	0
	10	0	1	0	0	0	0	1

(a)

sign	$C_{LSP}, A_{and}, A_{nor}$							
	000	001	011	010	100	101	111	110
B_{and}, B_{nor}	00	0	0	0	0	0	0	0
	01	0	0	0	1	0	0	0
	11	0	0	0	0	0	0	0
	10	0	1	0	1	0	0	1

(b)

Fig. 10 Representations of (a) carr-ctrl signal and (b) sign signal in terms of KARNAUGH maps

$$A_{nor} = \overline{A[15] + A[14] + \dots + A[8]} \quad (4)$$

$$B_{nor} = \overline{B[15] + B[14] + \dots + B[8]} \quad (5)$$

$$close = \overline{(A_{and} + A_{nor}) \times (B_{and} + B_{nor})} \quad (6)$$

$$\begin{aligned} carr-ctrl &= \overline{C_{LSP}} \times \overline{A_{and}} \times A_{nor} \times B_{and} \times \overline{B_{nor}} \\ &+ \overline{C_{LSP}} \times A_{and} \times \overline{A_{nor}} \times \overline{B_{and}} \times B_{nor} \\ &+ C_{LSP} \times \overline{A_{and}} \times A_{nor} \times B_{and} \times B_{nor} \\ &+ C_{LSP} \times A_{and} \times \overline{A_{nor}} \times B_{and} \times \overline{B_{nor}} \\ &= \overline{C_{LSP}} \times (\overline{A_{and}} \times B_{and} + A_{and} \times \overline{B_{and}}) \\ &\times (A_{and} \times B_{and} + A_{and} \times B_{nor} + A_{nor} \times B_{and} \\ &+ A_{nor} \times B_{nor}) + C_{LSP} \\ &\times (A_{and} \times B_{and} + \overline{A_{and}} \times \overline{B_{and}}) \\ &\times (A_{and} \times B_{and} + A_{and} \times B_{nor} + A_{nor} \times B_{and} \\ &+ A_{nor} \times B_{nor})(\overline{C_{LSP}} \oplus A_{and} \oplus B_{and}) \\ &\times (A_{and} + A_{nor}) \times (B_{and} + B_{nor}) \quad (7) \end{aligned}$$

$$\begin{aligned}
 sign &= \overline{C_{LSP}} \times (\overline{A_{and}} \times A_{nor} \times B_{and} \times \overline{B_{nor}} + A_{and} \\
 &\quad \times \overline{A_{nor}} \times \overline{B_{and}} \times B_{nor} + A_{and} \times A_{nor} \\
 &\quad \times B_{and} \times \overline{B_{nor}}) \\
 &+ C_{LSP} \times A_{and} \times \overline{A_{nor}} \times B_{and} \times \overline{B_{nor}} \\
 &= \overline{C_{LSP}} \times (\overline{A_{and}} \times B_{and} + A_{and}) \\
 &\quad + C_{LSP} \times A_{and} \times B_{and} \\
 &= \overline{C_{LSP}} \times (A_{and} + B_{and}) + C_{LSP} \times A_{and} \times B_{and} \quad (8)
 \end{aligned}$$

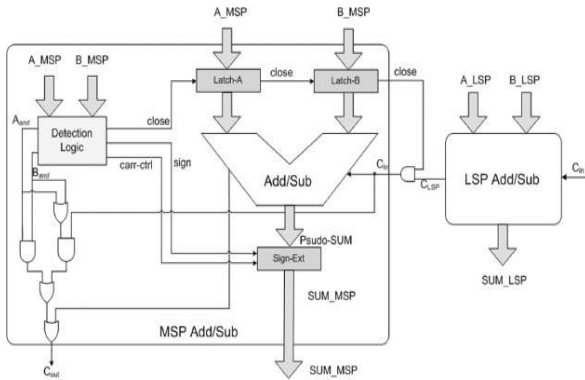


Fig. 11 Low-Power Adder/Subtractor Design Example Adopting the Proposed Spst.

B. Realization Issues of the Proposed SPST

Fig. 5 shows a 16-bit adder/subtractor design example adopting the proposed SPST. In this example, the 16-bit adder/subtractor is divided into MSP and LSP between the eighth and the ninth bits. Latches implemented by simple AND gates are used to control the input data of the MSP. When the MSP is necessary, the input data of MSP remain unchanged. However, when the MSP is negligible, the input data of the MSP become zeros to avoid glitching power consumption. The two operands of the MSP enter the detection-logic unit, except the adder/subtractor, so that the detection-logic unit can decide whether to turn off the MSP or not. Based on the derived Boolean equations (1) to (8), the detection-logic unit of SPST is shown in Fig. 6(a), which can determine whether the input data of MSP should be latched or not. Moreover, we propose the novel glitch-diminishing technique by adding three 1-bit registers to control the assertion of the *close*, *sign*, and *carr-ctrl* signals to further decrease the transient signals occurred in the cascaded circuits which are usually adopted in VLSI architectures designed for multimedia/DSP applications. The timing diagram is shown in Fig. 6(b). A certain amount of delay is used to assert the *close*, *sign*, and *carr-ctrl* signals after the period of data transition which is achieved by controlling the three 1-bit registers at the outputs of the detection-logic unit. Hence, the transients of the detection-logic unit can be filtered out; thus, the data latches shown in

Fig. 5 can prevent the glitch signals from flowing into the MSP with tiny cost. The data transient time and the earliest required time of all the inputs are also illustrated in Fig. 6(b). The delay should be set in the range of, which is shown as the shadow area in Fig. 6(b), to filter out the glitch signals as well as to keep the computation results

correct. Based on Figs. 5 and 6, the timing issue of the SPST is analyzed as follows.

1) *When the detection-logic unit turns off the MSP:* At this moment, the outputs of the MSP are directly compensated by the SE unit; therefore, the time saved from skipping the computations in the MSP circuits shall cancel out the delay caused by the detection-logic unit.

2) *When the detection-logic unit turns on the MSP:* The MSP circuits must wait for the notification of the detection-logic unit to turn on the data latches to let the data in. Hence, the delay caused by the detection-logic unit will contribute to the delay of the whole combinational circuitry, i.e., the 16-bit adder/subtractor in this design example.

3) *When the detection-logic unit remains its decision:*

The detection-logic unit should be a speed-oriented design. When the SPST is applied on combinational circuitries, we should first determine the longest transitions of the interested cross sections of each combinational circuitry, which is a timing characteristic and is also related to the adopted technology. The longest transitions can be obtained from analyzing the timing differences between the earliest arrival and the latest arrival signals of the cross sections of a combinational circuitry. Then, a delay generator similar to the delay line used in the DLL designs [16], [17], comprising several invertors and some capacitors, can be used to generate a proper delay to control the “*close*,” “*sign*,” and “*carr-ctrl*” signals. Fig. 7 shows the data-controlling components of the SPST, where Fig. 7(a) shows the design of the data latch. The SE circuits can be intuitively implemented by multiplexers to compensate for the sign signals of the MSP, as shown in Fig. 7(b). The input data of the SE circuits are pseudosummations (PS) from the MSP adder/subtractors. In this paper, we further explore two more approaches besides using multiplexers to optimize the SE circuits. One approach uses simple OR gates, as shown in Fig. 7(c). The other adopts Complementary Pass transistor Logics (CPLs) [18], as shown in Fig. 7(d). Both of these approaches can help realize the needed SE circuits. From the performance and overhead comparisons, fully discussed in Section IV, we decide to adopt the CPL circuits in our design.

VII. CONCLUSION

The proposed work investigated on glitch-optimized circuit blocks for high-performance Booth multipliers aiming to reduce the dynamic power dissipation caused by the parasitic and spurious activities. The proposed strategy incorporates a low transition adder to reduce the switching activity of partial product addition. The adder separates the target designs into two parts, and turns off the MSP when it does not affect the computation results to save power. Therefore, the proposed approach is an excellent choice for high-performance, energy-constrained multiplication. The

efficacy of the proposed strategies has been synthesized using Xilinx ISE. From the results, it was concluded that the proposed versions are on average reduce up to 55%–65% of delay and 50% of area reduction along with power reduction.

VIII. FUTURE SCOPE

Further booth architecture can extend in to Multiprecision multiplier to reduce power and area.

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] Z. Huang and M. D. Ercegovac, "High- performance low-power leftto- right array multiplier design," *IEEE Trans. Comput.*, vol. 54, no. 3,pp. 272–283, Mar.2005.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] 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/June. 1998, pp. 260–263.
- [13] 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
- [14] 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.
- [15] 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.
- [16] 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.
- [17] J. Mori *et al.*, "A 10 ns 54×54 b parallel structured full array multiplier with 0.5 μm CMOS technology," *IEEE J. Solid-State Circuits*, vol. 26,no. 4, pp. 600–606, Apr.1991.
- [18] N. Ohkubo *et al.*, "A 4.4 ns CMOS 54×54-b multiplier using passtransistor multiplexer,"*IEEE J. Solid-State Circuits*, vol. 30, no. 3,pp. 251–257, Mar. 1995.
- [19] 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.
- [20] L.-D. Van and J.-H. Tu, "Power-efficient multipliers," *IEEE Trans. Comput.*, vol. 58, no. 10,pp. 1346–1355, Oct. 2009.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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.

- [30] 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.