IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

A Low Power Radix-4 Booth Multiplier with Pre-Encoded Mechanism

Yen-Jen Chang[1], Member, IEEE, Yu-Cheng Cheng[1], Shao-Chi Liao[1], and Chun-Huo Hsiao[1]

[1] Department of Computer Science and Engineering, National Chung Hsing University, Taichung, Taiwan

Corresponding author: Yu-Cheng Cheng (e-mail: co7835g@gmail.com).

This work was supported in part by Ministry of Science and Technology, Taiwan, under Project MOST 104-2221-E-005-025-

ABSTRACT The radix-4 Booth algorithm is widely used to improve the performance of multiplier because it can reduce the number of partial products by half. However, numerous additional encoders and decoders would cause the power consumption of the Booth multiplier to be considerable. In this paper, a new radix-4 Booth pre-encoded mechanism is proposed to reduce the power consumption of the Booth multiplier. The proposed design can effectively reduce the power of the Booth multiplier dissipated in the redundant activities by disabling the Booth encoders and decoders from unnecessary working. Particularly, since the control signals are generated early at the pipeline input register before the multiplier, the performance of our design is better than the traditional Booth multiplier. Based on the TSMC 40 nm technology, the simulation results show that the proposed pre-encoded mechanism can reduce the dynamic and static power by 45% and 65%, respectively, compared to the traditional 16-bit radix-4 Booth multiplier. Compared to the previous designs, the proposed design keeps the feature of race-free and has lower power consumption. Even compared to the approximate design, the proposed design has better power efficiency and can provide the exact products.

INDEX TERMS Booth algorithm, low power multiplier, power efficiency, partial product, radix-4 Booth multiplier

I. INTRODUCTION

Many digital signal processing (DSP) and machine learning applications are heavily dominated by multiplication [1]-[4], e.g., more than 90% convolutional neural networks (CNN) computations are occupied by multiply-accumulate (MAC) operations [5], [6]. Therefore, the multiplier is an important component in various hardware platforms. The conventional multiplication includes three major phases [3], [7]-[9]. (1) Two inputs (multiplier and multiplicand) are multiplied to generate the partial products (PPs). (2) Reducing the PPs' matrix into two rows by partial product reduction schemes (3) The final carry propagated addition of the remaining two rows of PPs. Particularly, the second phase plays a significant role in power consumption, cost, and overall performance [7]-[9]. Then, the radix-4 Booth algorithm can improve the performance of multiplication because the radix-4 Booth multiplier can reduce the number of PP rows by half [3], [10], [11].

The authors of [11] provided a simple and intuitive encoding/decoding method to implement the radix-4 Booth algorithm; reference [12] provided a modified sign extension structure to reduce the cost and to improve performance. In [16], the author found that the traditional radix-4 Booth implementations [13]-[15] may result in unnecessary glitches of PPs. Thus, the author of [16] proposed the glitch-free Booth encoder and partial product generator to eliminate the unnecessary glitches of the radix-4 Booth multiplier. However, these traditional designs still suffer from high power consumption and high cost of Booth encoders and decoders. Thus, the authors of [17] proposed a high performance and low cost radix-4 Booth decoder. The decoder of [17] also keeps the advantage of race-free and its cost is less than the cost of [16]. Research [18] developed a *neg/two/one-nf* generator (encoder and decoder) to reduce the glitches in the second phase of multiplication. This *neg/two/one-nf* generator has less encoded signals and its signal paths are more balanced than other schemes. In [19], in order to reduce the cost of the Booth encoder and decoder, the authors proposed a novel modified Booth encoder (NMBE) scheme that is based on the pass transistor logic (PTL).

For some error-tolerant applications, the approximate circuits can be employed to achieve low power, low circuit complexity, and high performance [9], [20]-[22]. The traditional radix-8 Booth algorithm can generate fewer PPs

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3003684, IEEE Access

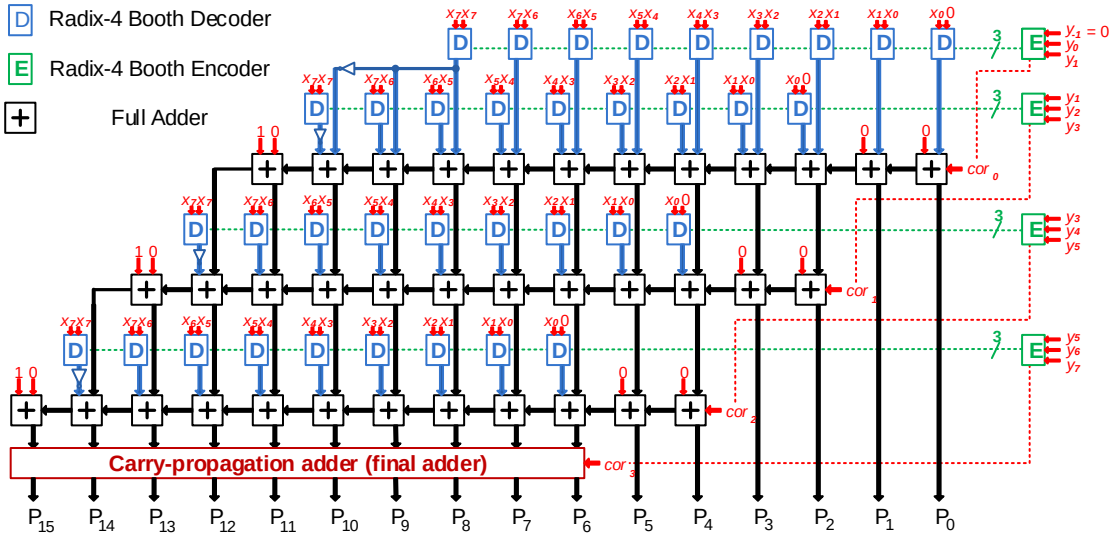

FIGURE 1. **The traditional 8×8 radix-4 Booth multiplier with the modified sign extension structure.**

than the radix-4 Booth algorithm, but it needs additional adders to process the operation of odd multiples of the multiplicand. Therefore, the approximate 2-bit adder [20] was proposed to generate the triple multiplicand with no carry propagation to improve the performance. In [21], the approximate radix-4 Booth multipliers were proposed by using their approximate Booth encoders and approximate Wallace tree structure. In [22], three approximation techniques for the radix-4 Booth multiplier were proposed and these designs can reduce the logic complexity of the PP generator.

In this paper, we propose a radix-4 Booth multiplier with pre-encoded mechanism to improve the power efficiency of multiplication. One specific feature of the radix-4 Booth algorithm is that when the continuous three bits of multiplier Y (y_{2i+1}, y_{2i}, y_{2i-1}) have the same values, the corresponding PPs will be 0. This feature inspired us to find the "0X" case earlier to reduce the unnecessary switching activities of the radix-4 Booth encoders and decoders. Thus, we propose the pre-encoded scheme to detect the "0X" case before every multiplication. When the "0X" case occurs, the proposed pre-encoder will turn off the Booth encoders and decoders to save power, and set the corresponding PPs to 0 directly before the starting of multiplication. The proposed design is simulated with Taiwan Semiconductor Manufacturing Company (TSMC) 40 nm technology. Compared with the traditional design [16] and the related designs [18], [19], [22], the simulation results show that the proposed design outperforms these designs in terms of transistor count, delay, and power consumption.

The rest of this paper is organized as follows. Section II reviews the traditional radix-4 Booth multiplier and the related works. Section III describes the low power radix-4 Booth multiplier with pre-encoded mechanism in detail. Section IV shows the simulation results of the proposed design. Section V offers a brief conclusion of this paper.

TABLE I
RADIX-4 BOOTH ALGORITHM

y_{2i+1}	y_{2i}	y_{2i-1}	M_i	Operation on X
0	0	0	0	0X
0	0	1	+1	+1X
0	1	0	+1	+1X
0	1	1	+2	+2X
1	0	0	-2	-2X
1	0	1	-1	-1X
1	1	0	-1	-1X
1	1	1	0	0X

II. TRADITIONAL RADIX-4 BOOTH MULTIPLIER AND RELATED WORKS

Multiplication is a basic arithmetic operation; many DSP and machine learning applications are highly multiply-intensive [1]-[4]. Therefore, the power consumption and performance issues of the multiplier are important. However, the traditional array multiplier generates a lot of PPs ($n \times n$ multiplication has n PP rows) and accumulates all PPs to get the final product; it consumes huge power and is not power efficiency. Then, the Booth algorithm (radix-2 Booth algorithm) has been proposed to improve the performance of the multiplication [23]; the radix-4 Booth algorithm (also called modified Booth algorithm) [24] can reduce the number of PP rows by half to facilitate the multiplication. In this section, we introduce the traditional radix-4 Booth algorithm and the related works.

A. TRADITIONAL RADIX-4 BOOTH ALGORITHM

The radix-4 Booth algorithm is a powerful method to improve the performance of multiplication and applies to two's complement operands. The radix-4 Booth algorithm partitions the multiplier Y ($y_{n-1}y_{n-2}\ldots y_0$) into overlapping groups of contiguous three bits. Each group is encoded and then decoded with multiplicand X ($x_{n-1}x_{n-2}\ldots x_0$) to generate the corresponding PPs. The n-bit multiplication of radix-4 Booth algorithm can be expressed as follows:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3003684, IEEE Access

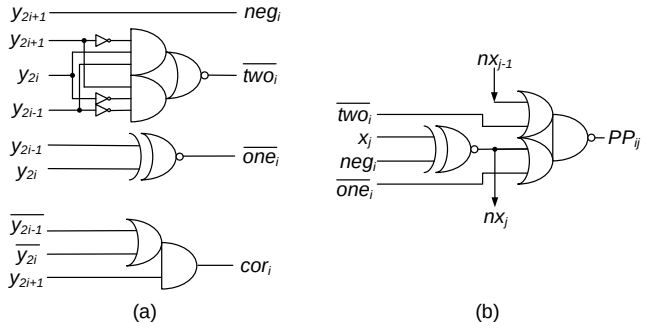

FIGURE 2. ***neg/two/one-nf*** **generator [18]. (a) encoder; (b) decoder.**

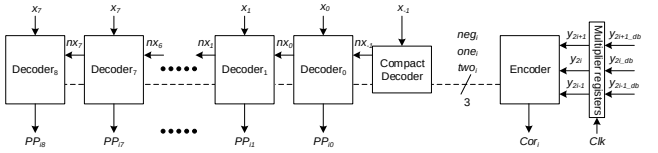

FIGURE 3. **One 8-bit PP row of the** ***neg-first*** **scheme.**

$$\begin{aligned} X \times Y &= X \times \left(-y_{n-1}2^{n-1} + \sum_{i=0}^{n-2} y_i 2^i\right) \\ &= X \times \sum_{i=0}^{\left(\frac{n}{2}\right)-1} (-2y_{2i+1} + y_{2i} + y_{2i-1}) 2^{2i} \\ &= X \times \sum_{i=0}^{\left(\frac{n}{2}\right)-1} M_i 2^{2i} \\ &= \sum_{i=0}^{\left(\frac{n}{2}\right)-1} M_i 2^{2i} X \end{aligned} \quad (1)$$

multiplicand X and multiplier Y are n bit two's complement numbers. According to the continuous three bits of multiplier Y (y_{2i+1}, y_{2i}, y_{2i-1}), the radix-4 Booth algorithm can generate the corresponding coefficient M_i. The coefficient M_i has five possible values (± 1, ± 2, or 0) as shown in Table I. Refer to (1), the radix-4 Booth algorithm can reduce the number of PP rows by half, and one specific feature is that the corresponding PPs will be 0s when the continuous three bits of multiplier Y (y_{2i+1}, y_{2i}, y_{2i-1}) ($M_i = 0$) have the same values. As shown in Table 1, the continuous three bits of multiplier Y (y_{2i+1}, y_{2i}, y_{2i-1}) have the same values that indicate the "0X" case. Notice that y_{-1} is always 0 when i equals to 0.

Although the radix-4 Booth algorithm can reduce the number of PPs, the cost of the radix-4 Booth multiplier is still high. Reference [12] provided a modified sign extension structure which can reduce the number of PPs and the cost further. Fig. 1 shows the traditional radix-4 Booth multiplier with the modified sign extension structure. Each PP row has one Booth encoder and n+1 Booth decoders. Booth encoders and decoders generate the PPs; full adders and carry-propagation adder add all PPs to get the final product. However, the unnecessary glitches and switching activities occur in the traditional radix-4 Booth multiplier because of the unbalanced signal paths of radix-4 Booth encoders and decoders. That leads to high power consumption [16]-[18]. The overall power consumption of CMOS can be defined as

$$P = aCV_{dd}^2 f + V_{dd} I_{leakage} \quad (2)$$

TABLE II
THE *NEG-FIRST* RADIX-4 BOOTH ALGORITHM [18]

y_{2i+1}	y_{2i}	y_{2i-1}	Operation on X	neg_i	two_i	one_i	cor_i
0	0	0	+0X	0	0	0	0
0	0	1	+1X	0	0	1	0
0	1	0	+1X	0	0	1	0
0	1	1	+2X	0	1	0	0
1	0	0	-2X	1	1	0	1
1	0	1	-1X	1	0	1	1
1	1	0	-1X	1	0	1	1
1	1	1	-0X	1	0	0	0

The first and second terms are dynamic and static power consumption, respectively. Where a is the switching activity parameter, C is the total capacitance load, V_{dd} is the supply voltage, f is the operating frequency, and $I_{leakage}$ is the leakage currents [25][26]. According to (2), the power consumption can be reduced by minimizing the switching activities.

B. RELATED WORKS

The radix-4 Booth encoder and decoder of [11] is the most common implementation of the radix-4 Booth algorithm, and the authors of [13]-[15] provided more compact implementations. However, the author of [16] found these designs have unnecessary glitches caused on PPs. Therefore, the glitch-free Booth encoder and partial product generator (decoder) have proposed to reduce the unnecessary glitches. In [16], the propagation delay from inputs X and Y to PP is only two units (one XOR/XNOR and the complex gate for output), and all paths almost have the same propagation delay. In [17], the authors proposed a high performance and low cost radix-4 Booth decoder that also keeps the advantage of race-free. Especially, the cost of [17] is less than the cost of [16].

In [18], the authors proposed a *neg/two/one-nf* (*nf* means *neg-first*) generator as shown in Fig. 2, and the *neg-first* radix-4 Booth algorithm is shown in Table II. The *neg-first* radix-4 Booth algorithm is the three-signal scheme; signals one_i, two_i, and neg_i can indicate that (y_{2i+1}, y_{2i}, y_{2i-1}) belongs to which case. The signal neg_i is for negation operation and cor_i is the correction bit for negative operation. The *neg-first* means that the negation operation is done before the selection between "1X" and "2X". For generating PP, the *neg/two/one-nf* decoder adopts the OR-AND-INV (OAI) gate, and all input signals of this OAI gate almost arrive at the same time (about 1 XNOR gate delay) as shown in Fig. 2. Therefore, the signal paths are more balanced than other schemes and the glitches can be reduced. However, in the case of "-0X", $neg_i = y_{2i-1}$ may lead to more switching activities in the XNOR gate since signals one_i and two_i can set PP to 0 regardless of neg_i. Note that the *neg-first* design needs one more compact decoder to generate the first *nx* signal and this decoder no needs to generate the PP as shown in Fig. 3. Therefore, each PP row of the *neg-first* design has one encoder and n+2 decoders.

The NMBE scheme [19] which is based on the PTL has less cost than traditional design. But, this scheme suffers from the problems of weak 1/strong 0 and the signal paths are not balanced. Research [22] provided three approximation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3003684, IEEE Access

techniques for radix-4 Booth multipliers and one of them is called approximate Booth multipliers models 1 (ABM-M1). ABM-M1 is composed of the exact partial product generators (radix-4 Booth encoder and encoder) and the approximate 2-signal partial product generators (called PPG-2S). Take 8-bit ABM-M1 with $m = 4$ for example, the PPs with a significance less than 4 are generated by PPG-2Ss and the remaining PPs are generated by the exact partial product generators. ABM-M1 can provide useful results with the low area-power product.

III. THE PROPOSED RADIX-4 BOOTH MULTIPLIER WITH PRE-ENCODED MECHANISM

In this paper, we propose a pre-encoded mechanism to reduce the power consumption of the radix-4 Booth multiplier. As mentioned above, the unnecessary switching activities make multiplier consume more power, and the PPs must be 0s in the "0*X*" case. For that reason, we propose a pre-encoded mechanism to find the "0*X*" case earlier to reduce the unnecessary switching activities of the radix-4 Booth encoders and decoders. When detects the "0*X*" case, the proposed pre-encoded mechanism can turn off Booth encoders and decoders immediately to save power.

The architecture and the timing chart of the proposed pre-encoded mechanism are shown in Fig. 4 and Fig. 5, respectively. As shown in Fig. 4, the proposed pre-encoded mechanism is composed of multiplicand and multiplier registers, additional proposed pre-encoders, and radix-4 Booth multiplier (includes adders, the proposed low cost Booth encoders and decoders). The signals $x_{n-1\_db}…x_{0\_db}$ ($_{db}$ means data bus) and $y_{n-1\_db}…y_{0\_db}$ denote the multiplicand X and multiplier Y on the data bus, respectively; $x_{n-1}…x_0$ and $y_{n-1}…y_0$ denote the outputs of the multiplicand and multiplier registers. As shown in Fig. 5, the multiplicand X and multiplier Y would be set on the data bus during the data setup time before the multiplication (multiplication phase). The proposed mechanism can detect the "0*X*" case during this setup time (denotes as pre-encode phase). In order to detect the "0*X*" case before the multiplication, the proposed design needs additional pre-encoders. If the "0*X*" case occurs, the proposed pre-encoders will immediately turn off the corresponding Booth encoders and decoders, and the corresponding PPs will be set as 0s to reduce the switching activities. In contrast, the Booth encoders and decoders will work as usual. Because the pre-encoders process the "0*X*" case already, the proposed Booth encoders and decoders only need to process the "$\pm$1*X*" and "$\pm$2*X*" cases, then, the cost of our design can be reduced. Accordingly, the proposed pre-encoded mechanism has less cost than the other designs.

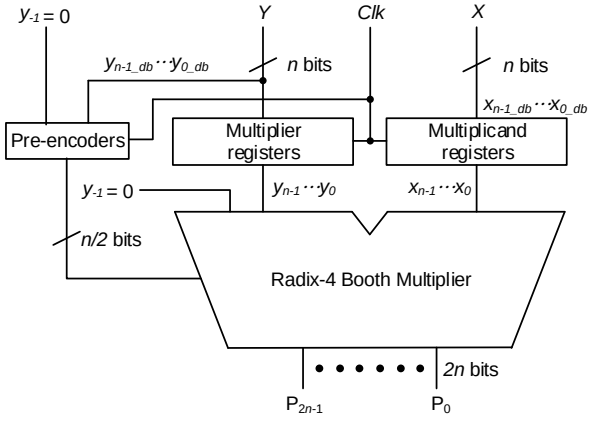

FIGURE 4. The architecture of the proposed radix-4 Booth multiplier with the pre-encoded mechanism.

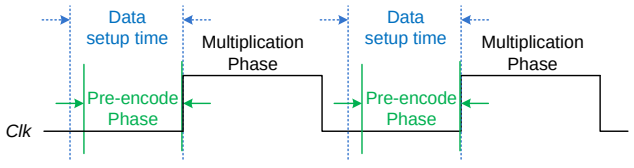

FIGURE 5. The timing chart of the proposed pre-encoded mechanism.

A. THE PROPOSED PRE-ENCODER

The proposed pre-encoded mechanism needs pre-encoder to detect the "0*X*" case in the pre-encode phase. Table III shows the proposed pre-encoded radix-4 Booth algorithm. The proposed pre-encoded mechanism has three encoded signals. Signal $zero_i$ which is generated by the proposed pre-encoder can determine the continuous three bits of multiplier Y on the data bus are the same or not. Signals neg_i and ot_i are generated by the proposed encoder. Signal neg_i is for the negation operation; signals neg_i and ot_i are for the remaining cases ("$\pm$1*X*" and "$\pm$2*X*"). Note that signal $zero_i$ has the highest priority, cor_i is the correction bit for negative operation, and d means "don't care".

According to Table III, when y_{2i+1}, y_{2i}, and y_{2i-1} on the data bus have the same value, the "0*X*" case occurs, thus, the signal $zero_i$ will be 1. The equation of $zero_i$ can be written as

$$zero_i = (\overline{y_{2i+1}} \cdot \overline{y_{2i}} \cdot \overline{y_{2i-1}})_{db} + (y_{2i+1} \cdot y_{2i} \cdot y_{2i-1})_{db} \quad (3)$$

Where $_{db}$ means that the y_{2i+1}, y_{2i}, and y_{2i-1} arrive at the data bus in the pre-encode phase. According to (3), the proposed pre-encoder can be implemented with the AND-OR-INV (AOI) gate and a latch-like circuit as shown in Fig. 6(a). Fig. 6(b) shows the proposed pre-encoder at the transistor level. In the pre-encode phase, when the multiplier Y arrives on the data bus, the pre-encoder starts to detect the "0*X*" case. The signal $zero_i$ is generated to control the proposed low cost encoder

TABLE III
THE PROPOSED PRE-ENCODED RADIX-4 BOOTH ALGORITHM

y_{2i+1}	y_{2i}	y_{2i-1}	Operation on X	$zero_i$	neg_i	ot_i	cor_i
0	0	0	0*X*	1	0	*d*	0
0	0	1	+1*X*	0	0	1	0
0	1	0	+1*X*	0	0	1	0
0	1	1	+2*X*	0	0	0	0
1	0	0	-2*X*	0	1	0	1
1	0	1	-1*X*	0	1	1	1
1	1	0	-1*X*	0	1	1	1
1	1	1	0*X*	1	1	*d*	0

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3003684, IEEE Access

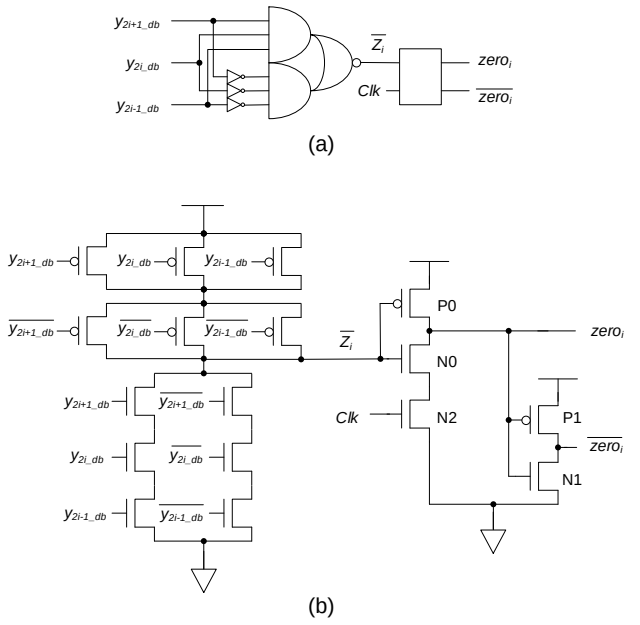

FIGURE 6. The proposed pre-encoder. (a) gate level; (b) transistor level.

and decoders (introduce in section III-B) of the *i*th PP row; the encoder and decoders work as usual or not in the multiplication phase according to the signal $zero_i$.

1) "Non-0*X*" cases: In the pre-encode phase, if the y_{2i+1_db}, y_{2i_db}, and y_{2i-1_db} are not the same, the signal $zero_i$ will be set to 0 by the proposed pre-encoder. Therefore, the proposed encoder and decoders of the *i*th PP row will work as normal to generate the corresponding PPs in the multiplication phase.
2) "0*X*" case: In the pre-encode phase, if the y_{2i+1_db}, y_{2i_db}, and y_{2i-1_db} are the same, the signal $zero_i$ will be set to 1 by the proposed pre-encoder. Thus, the proposed encoder and decoders of the *i*th PP row will be powered off to reduce power consumption, and the corresponding PPs will be set to 0s directly in the pre-encode phase. In the multiplication phase, these gated encoder and decoders are no need to work since they are already turned off and the corresponding PPs are already set to 0s.

Note that the latch-like circuit shown in Fig. 6 is added to prevent the probably happened unnecessary switching activities when "0*X*" case changes to "non-0*X*" case. Fig. 7

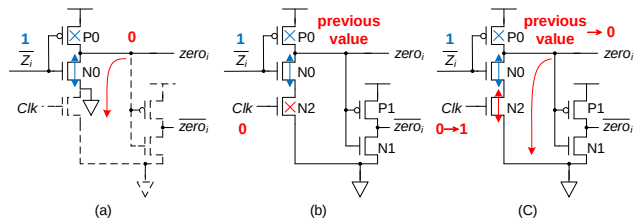

FIGURE 7. The difference between the proposed pre-encoder without and with the latch-like circuit. (a) without the latch-like circuit; (b) with the latch-like circuit in the pre-encode phase; (c) with the latch-like circuit in the multiplication phase.

shows the latch-like circuit operations in this special situation. As shown in Fig. 7(a), when "0*X*" case changes to "non-0*X*" case (signal $\overline{Z}_i$ changes from 0 to 1) in the pre-encode phase, the pre-encoder without the latch-like circuit generates $zero_i$ as 0 immediately that makes encoder and decoders turn on to do the redundant multiplication of the previous time ("0*X*" case). To avoid this redundant multiplication, the latch-like circuit is required. As shown in Fig. 7(b), when "0*X*" case changes to "non-0*X*" case in the pre-encode phase, *Clk* is 0, and N2 is turned off. The pre-encoder with the latch-like circuit makes $zero_i$ keep the previous value at a high voltage level to turn off the encoder and decoders, then, the redundant multiplication can be avoided. However, until *Clk* changes from 0 to 1, N2 is turned on and the pre-encoder with latch-like circuit makes $zero_i$ as 0 through N0 and N2 to power on the encoder and decoders; the "non-0*X*" multiplication starts normally in the multiplication phase.

B. THE PROPOSED LOW COST RADIX-4 BOOTH ENCODER AND DECODER

Because the "0*X*" case has been processed by the proposed pre-encoder, the encoder and decoder only need to process the remaining cases ("$\pm 1X$" and "$\pm 2X$"). For that reason, the costs of radix-4 Booth encoder and decoder can be reduced. In this paper, we propose the low cost radix-4 Booth encoder and decoder as shown in Fig. 8 and Fig. 9 to reduce the power consumption further. The gating techniques (power gating and ground gating) and the low cost of our design can reduce the dynamic and also static power consumption effectively. In particular, the issues of reducing static power consumption become more and more important when technology progresses [26][27]. Table IV shows a summary of the proposed pre-encoded radix-4 Booth algorithm. Signal $zero_i$ can be used as the control signal of the gating transistors that

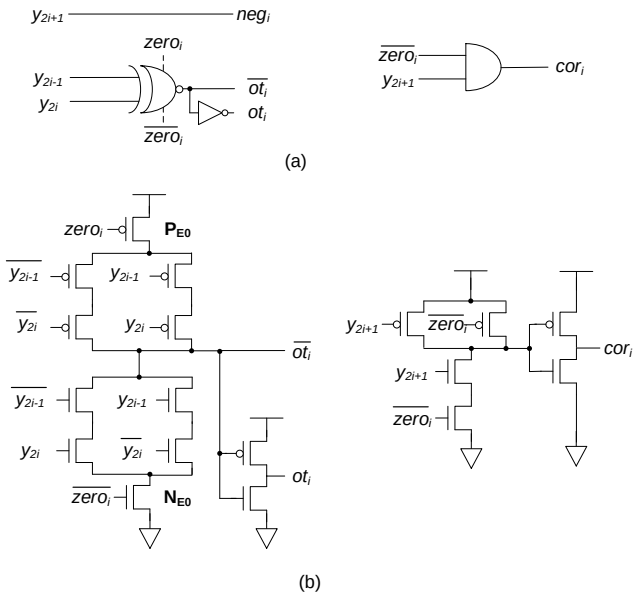

FIGURE 8. The proposed low cost encoder. (a) gate level; (b) transistor level.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3003684, IEEE Access

TABLE IV
THE SUMMARY OF THE PROPOSED PRE-ENCODED RADIX-4 BOOTH ALGORITHM SIMPLIFIED FROM TABLE III

Case	$zero_i$	ot_i	cor_i	PP_{ij}
$0X$	1	*d*	0	0
$\pm 1X$	0	1	neg_i	$neg_i \oplus x_j$
$\pm 2X$	0	0	neg_i	$neg_i \oplus x_{j-1}$

are added in the proposed encoder and decoder. When $zero_i$ is 1, the proposed encoder and decoder will be gated to reduce power consumption. In contrast, the proposed encoder and decoder will work as usual.

Fig. 8(a) and Fig. 8(b) show the details of the proposed encoder. Like the traditional design, the inputs y_{2i+1}, y_{2i}, and y_{2i-1} of the proposed encoder are the outputs of the corresponding multiplier registers. Note that the inputs of the proposed pre-encoder are the multiplier *Y* on the data bus since the pre-encoder needs to detect the "0*X*" case earlier in the pre-encode phase. Because of this pre-encoder, the proposed encoder only needs to generate signals neg_i and ot_i. According to Table III, the expressions of neg_i and ot_i are

$$neg_i = y_{2i+1} \tag{4}$$

$$ot_i = y_{2i-1} \oplus y_{2i} \tag{5}$$

Signal neg_i is equal to y_{2i+1} and signal ot_i can be generated by an XOR gate according to (4) and (5), respectively. The XOR gate can be implemented with the CMOS logic [28]. To reduce the power consumption in the "0*X*" case, the XOR gate of the proposed encoder adopts the gating techniques. As shown in Fig. 8(b), the gating transistor P_{E0} is added between the power supply and the XNOR logic, and is controlled by $zero_i$. The gating transistor N_{E0} is added between GND and the XNOR logic, and is controlled by $\overline{zero_i}$. Signals $zero_i$ and $\overline{zero_i}$ are generated by the proposed pre-encoder as introduced above.

1) "Non-0*X*" cases: In the pre-encode phase, if the y_{2i+1}, y_{2i}, and y_{2i-1} are not the same, the signal $zero_i$ is 0. Thus, the gating transistors P_{E0} and N_{E0} of the proposed encoder will be turned on. The proposed encoder works as normal to generate the corresponding encoded signals in the multiplication phase.
2) "0*X*" case: In the pre-encode phase, if the y_{2i+1}, y_{2i}, and y_{2i-1} are the same, the signal $zero_i$ is 1. Thus, the gating transistors P_{E0} and N_{E0} of the proposed encoder will be turned off to reduce the dynamic power consumption and leakage currents. In the multiplication phase, the proposed encoder still stays in the standby mode to save power.

According to Table III, the expression of correction bit cor_i can be written as

$$cor_i = y_{2i+1} \cdot \overline{zero_i} \tag{6}$$

Based on (6), the correction bit can be generated by an AND gate that is simpler than the circuitry in [18]. The correction bit cor_i is 0 in the "0*X*" case; otherwise, the value of cor_i depends on y_{2i+1} (neg_i) as summarized in Table IV.

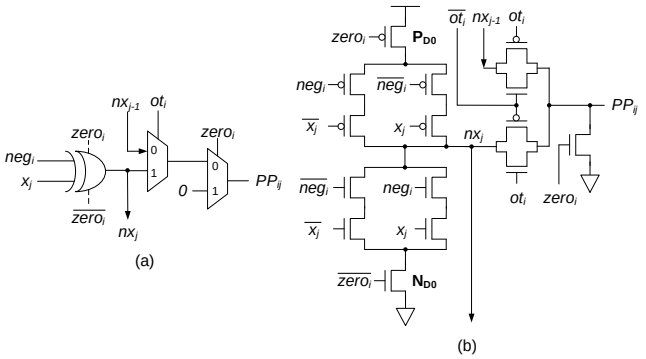

FIGURE 9. **The proposed low cost decoder. (a) gate level; (b) transistor level.**

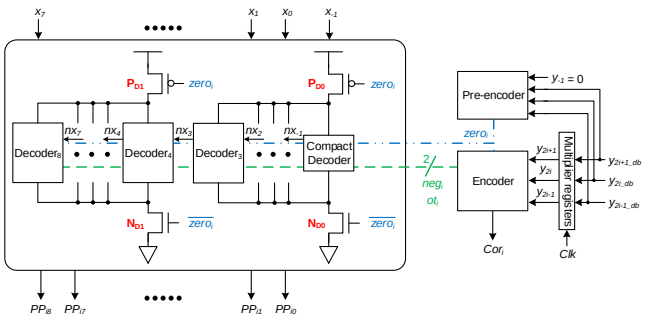

FIGURE 10. **One 8-bit PP row of the proposed pre-encoded mechanism.**

Fig. 9(a) and Fig. 9(b) show the details of the proposed decoder. Like the traditional design, the input x_j of the proposed decoder is connected with the output of the corresponding multiplicand register. According to the encoded signals of the proposed pre-encoder and encoder, the proposed decoder can generate the corresponding PP. As introduced before, the PP should be 0 when the "0*X*" case occurs. The PP of "+1*X*" is x_j and the PP of "-1*X*" is $\overline{x_j}$. The PP of "+2*X*" is x_{j-1} and the PP of "-2*X*" is $\overline{x_{j-1}}$. Table IV summarizes the PP value of each case; the PP values of "$\pm 1X$" and "$\pm 2X$" can be defined as (7) and (8), respectively.

$$PP_{\pm 1X} = neg_i \oplus x_j = nx_j \tag{7}$$

$$PP_{\pm 2X} = neg_i \oplus x_{j-1} = nx_{j-1} \tag{8}$$

Based on Table IV, the expression of PP value can be written as

$$PP_{ij} = zero_i \cdot 0 + \overline{zero_i} \cdot \left(ot_i \cdot nx_j + \overline{ot_i} \cdot nx_{j-1}\right) \tag{9}$$

According to (9), the proposed decoder can be composed of an XOR gate and two multiplexers (MUXs). The XOR gate is implemented with the CMOS logic [28] and its output is shared with the neighbor decoder. The first MUX which is controlled by ot_i is implemented by the transmission gates (TGs); the second MUX which is controlled by $zero_i$ can be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3003684, IEEE Access

TABLE V
COMPARISONS FOR THE RADIX-4 BOOTH ALGORITHMS

	The traditional design [16]	The *neg-first* design [18]	The NMBE design [19]	The ABM-M1 design [22]	The proposed design
Features of Algorithm	• 4 encoded signals: *neg*, *X1*, *X2P*, and *ZP* • traditional glitch-free design • $n + 1$ decoders per row • did not adopt gating techniques	• 3 encoded signals: *neg*, *two*, and *one* • *neg-first* • has "-0*X*" case • $n + 2$ decoders per row • did not adopt gating techniques	• 3 encoded signals: *n*, *s*, and *z* • based on PTL • strong 0/ weak 1 • $n + 1$ decoders per row • did not adopt gating techniques	• 3 encoded signals: *neg*, *two*, and *zero* • composed of the exact and approximate PP generators • $n + 1$ decoders per row • did not adopt gating techniques	• 3 encoded signals: *zero*, *neg*, and *ot* • *zero* pre-encoded • no "-0*X*" case • $n + 2$ decoders per row • adopts gating techniques
Pre-encoder					• 17 transistors • generates signal *zero* in pre-encode phase • one PP row needs one pre-encoder
Encoder	• 34 transistors	• 30 transistors	• 20 transistors	• 40 transistors	• 18 transistors (with gating transistors)
Decoder	• 40 transistors	• 16 transistors • signal *nx* is shared with the neighbor	• 18 transistors	Exact • 20 transistors Approximate (PPG-2S) • 14 transistors	• 15 transistors (with gating transistors) • signal *nx* is shared with the neighbor
Propagation paths	• balance • race-free design	• balance • race-free design	• unbalance	Exact • unbalance Approximate (PPG-2S) • unbalance	• balance • race-free design

implemented easily by an NMOS. Like the proposed encoder, the proposed decoder adopts the gating techniques. As shown in Fig. 9(b), the gating transistor P_{D0} which is controlled by $zero_i$ is added between the power supply and the XOR logic; the gating transistor N_{D0} which is controlled by $\overline{zero_i}$ is added between GND and the XOR logic.

1) "Non-0*X*" cases: In the pre-encode phase, the signal $zero_i$ is 0 when the y_{2i+1}, y_{2i}, and y_{2i-1} are not the same. In the multiplication phase, the gating transistors P_{D0} and N_{D0} of the proposed decoder are turned on to generate the signal nx_j as normal. Therefore, the proposed decoder can generate the corresponding PP according to the encoded signals ot_i and $zero_i$.
2) "0*X*" case: In the pre-encode phase, the signal $zero_i$ is 1 when the y_{2i+1}, y_{2i}, and y_{2i-1} are the same. Since $zero_i$ is 1, the PP will be set to 0 directly in the pre-encode phase. In addition, the gating transistors P_{D0} and N_{D0} of the proposed encoder are turned off to reduce the dynamic power consumption and leakage currents. In the multiplication phase, the PP is 0, and the proposed decoder still stays in standby mode to save power.

As shown in Fig. 10, each PP row of the proposed pre-encoded mechanism has one pre-encoder, one encoder, and $n + 2$ decoders (10 decoders for $n = 8$) that is similar to [18]. Clearly, the decoders account for the majority of the circuitry cost of each PP row. To minimize the cost of the proposed decoders, the additional gating transistors can be shared. As shown in Fig. 10, every five decoder shares a set of gating transistors. Take 8-bit multiplication ($n = 8$) for example, there are 10 decoders including the compact decoder in each PP row, and these decoders can be divided into two groups. In the first group, $decoder_0$ to $decoder_3$ and the compact decoder share the gating transistors P_{D0} and N_{D0}. Notice that the compact decoder is implemented only by the XOR gate with the gating transistors because the compact decoder no needs to generate the PP. In the second group, $decoder_4$ to $decoder_8$ share the gating transistors P_{D1} and N_{D1}. When $zero_i$ is 0, these two groups work as usual to generate the corresponding PPs. When $zero_i$ is 1, these two groups will be turned off to reduce power consumption and set PPs of this row to 0s immediately.

Table V summarizes the comparisons for the proposed pre-encoded design and the related designs [16], [18], [19], [22]. The traditional design [16] is the four-signal scheme and have balanced signal propagation paths. But, the cost of traditional design is higher than the others. The NMBE design [19] is the three-signal scheme and based on the PTL. The cost of the NMBE design [19] is low, but the signal paths are not balanced. Moreover, the NMBE design [19] suffers from the problems of weak 1/strong 0 that makes the static power increase. In order to reduce the circuit complexity, the ABM-M1 design [22] adopted the approximate decoders (PPG-2S) in some least-significant bits. The encoder of [22] has 50 transistors which is larger than the others since this encoder needs to encode for the exact and approximate decoders. The ABM-M1 design [22] can only be used in the error-tolerant applications. Same as the *neg-first* design [18], the proposed design has three encoded signals and balanced signal propagation paths. However, the *neg-first* design [18] has the "-0*X*" case that may lead to more switching activities in XNOR gates of decoders. For *n*-bit multiplication of design [18] and the proposed design, one PP row needs one encoder and $n + 2$ decoders (one additional pre-encoder for the proposed design). Obviously, the proposed design has less cost than the others even though the proposed design needs the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3003684, IEEE Access

additional pre-encoder. Because of the pre-encoder, the proposed design can turn off the encoder and decoders to save power in the "0*X*" case. Therefore, we expect that the proposed design has the better power efficiency than the other designs.

IV. SIMULATION RESULTS

In this paper, the related works [16], [18], [19], [22], and the proposed pre-encoded mechanism are simulated by using TSMC 40 nm CMOS technology. The supply voltage is 1.0V, the clock frequency is 100 MHz, and the simulation is done by HSPICE tool. We simulate generating one PP row of *n*-bit multiplication (n = 8 or n = 16); we provide the power consumption, the performance, and transistor count (TC) to show the effectiveness of the proposed pre-encoded mechanism. We also provide the overall comparisons of multipliers to prove the superiority of the proposed design over the related works.

A. FUNCTIONALITY

To verify the feasibility and correctness of the proposed pre-encoded mechanism, the TSMC 40 nm technology is used to simulate two specific scenes with HSPICE. One specific scene is changing from "0*X*" case to "non-0*X*" case and the other is changing from "non-0*X*" case to "0*X*" case. Fig. 11 and Fig. 12 show the waveforms of these two specific scenes. The black solid line is the clock signal *Clk*, the blue solid line is the encoded signal $zero_i$, and the green solid line is the decoder output PP_{ij}. The dotted red and solid purple lines indicate y_{2i_db} for the data bus and the register output y_{2i}, respectively.

Fig. 11 shows the waveforms of "0*X*" case changing to "non-0*X*" case. We choose $i = 1$ and $j = 1$ to describe the waveforms in detail. Suppose that the multiplicand X is fixed at 01010101 and the continuous three bits of multiplier Y (y_3, y_2, y_1) change from (0, 0, 0) to (1, 1, 0), that is, "0*X*" case changes to "-1*X*" case. In the beginning, (y_3, y_2, y_1) = (0, 0, 0), thus, $zero_1$ is 1 and PP_{11} is 0. When y_{3_db} and y_{2_db} are set to 1 (y_3 and y_2 are still 0) in the pre-encode phase, the pre-encoder output $zero_1$ keeps the high voltage to prevent unnecessary switching activities as introduced in section III-B until *Clk* changes from 0 to 1. When *Clk* changes from 0 to 1 (the multiplication phase), registers store the new data from the data bus, and $zero_1$ changes from 1 to 0 to make the encoder and decoders work as normal. PP_{11} changes from 0 to 1 because y_3 (neg_1) is 1, x_1 is 0, and $zero_1$ is 0.

Fig. 12 shows the waveforms of "non-0*X*" case changing to "0*X*" case. We choose $i = 1$ and $j = 1$ to describe the waveforms in detail. Suppose that the multiplicand X is fixed at 01010101 and the continuous three bits of multiplier Y (y_3, y_2, y_1) change from (0, 1, 1) to (0, 0, 0), that is, "+2*X*" case changes to "0*X*" case. In the beginning, (y_3, y_2, y_1) = (0, 1, 1), therefore, $zero_1$ is 0 and PP_{11} is 1 ($y_3 = 0$ and $x_0 = 1$). When y_{2_db} and y_{1_db} are set to 0 (y_2 and y_1 are still 1) in the pre-encode phase, the pre-encoder output $zero_1$ is set to 1 directly to turn off the encoder and decoders. Notice that PP_{11} is also set to 0 immediately in the pre-encode phase because the pre-encoder detects the "0*X*" case. When *Clk* changes from 0 to 1 (the multiplication phase), registers store the new data from the data bus, and the encoder and decoders are turned off already to reduce power consumption. Fig. 11 and Fig. 12 prove the correctness and feasibility of the proposed pre-encoded design.

B. POWER CONSUMPTION ANALYSIS

The proposed pre-encoded design is compared with the related works [16], [18], [19], [22]; we simulate generating one PP row with 8-bit multiplication and 16-bit multiplication for each design. The continuous three bits of multiplier Y (y_{2i+1}, y_{2i}, y_{2i-1}) change from one pattern to each pattern. We simulate these pattern switches with the 100 MHz clock frequency and provide the average power consumption for each pattern. The power consumption of the proposed design is composed of the power consumption of pre-encoder, encoder, and decoders.

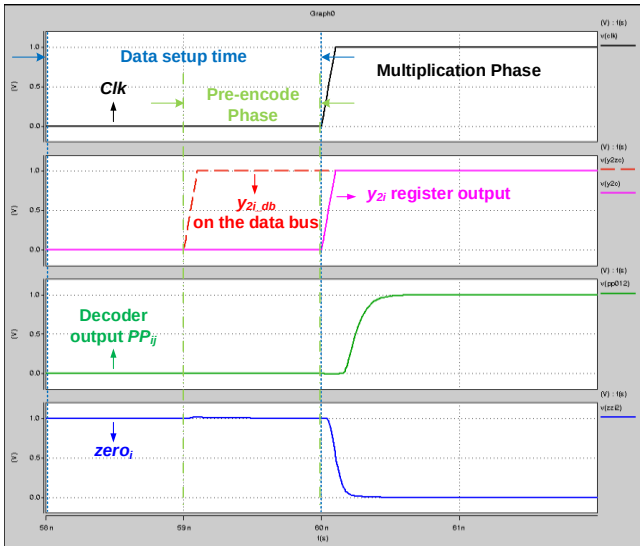

FIGURE 11. The waveforms of "0*X*" case changing to "non-0*X*" case.

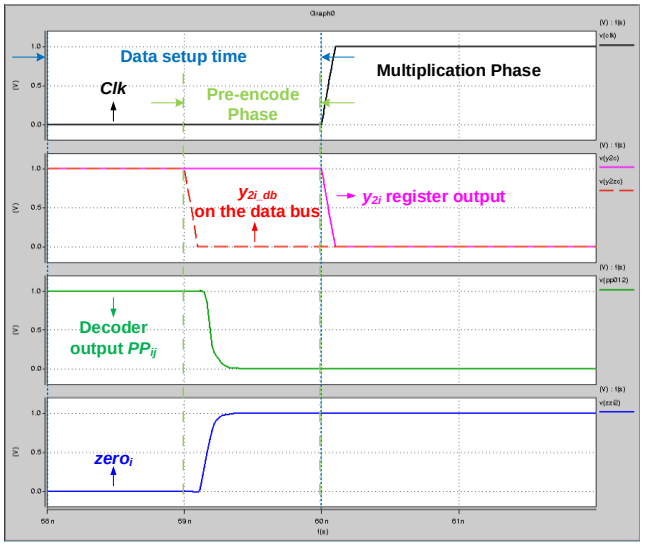

FIGURE 12. The waveforms of "non-0*X*" case changing to "0*X*" case.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3003684, IEEE Access

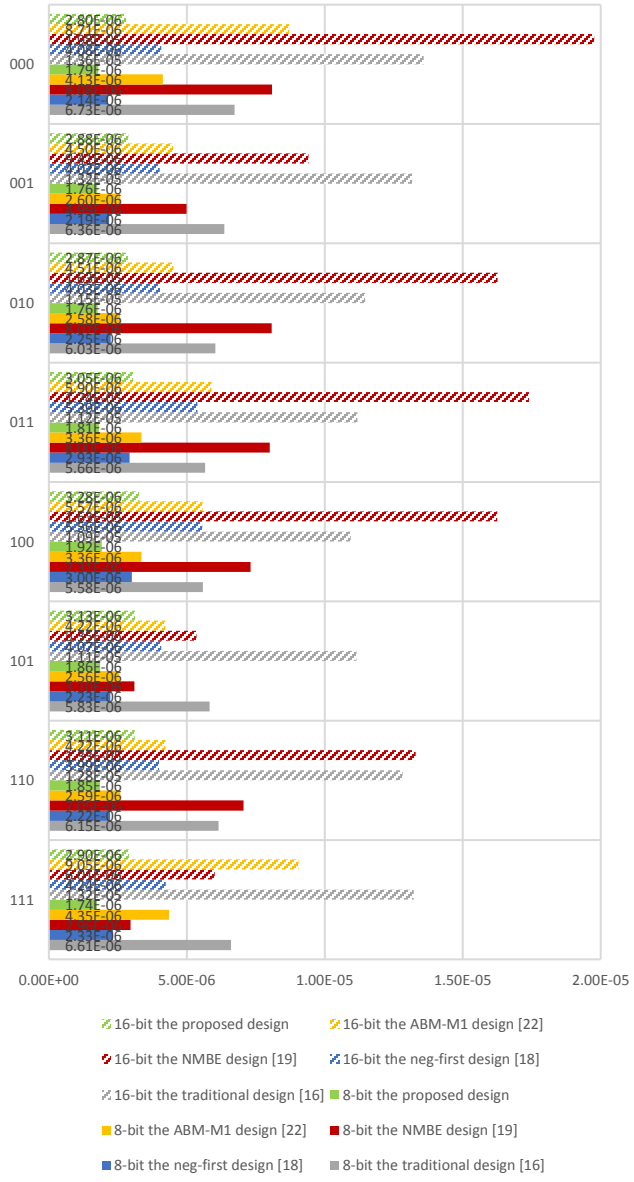

FIGURE 13. **The average dynamic power consumption for each pattern switch.**

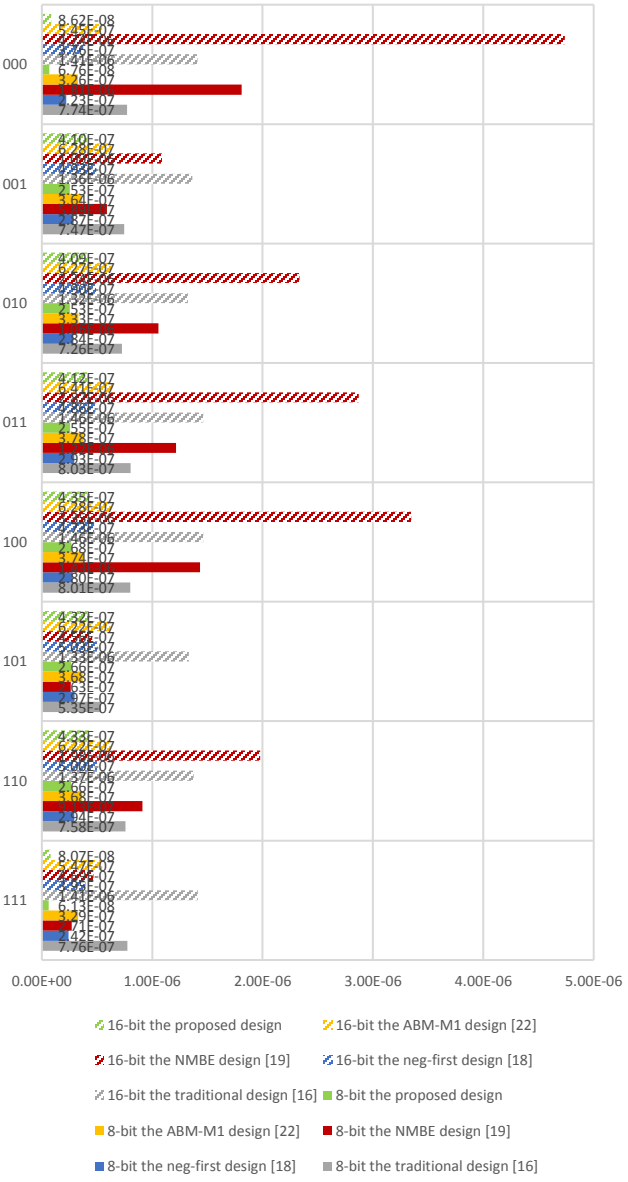

FIGURE 14. **The static power consumption for each pattern.**

The power consumption of each related work is composed of the power consumption of encoder and decoders. Note that we simulate one row of the ABM-M1 design [22] with the approximation factor $m = 4$ ($m = 8$) for the 8-bit multiplication (16-bit multiplication). When $m = 4$ ($m = 8$), one row of the ABM-M1 design [22] is composed of one encoder, five (nine) exact decoders, and four (eight) PPG-2Ss.

Fig. 13 shows the average dynamic power consumption for each pattern switch. For instance, "000" in Fig. 13 represents the average power consumption for changing from each pattern to pattern "000". Obviously, the dynamic power consumption of the proposed design is less than the power consumption of the other designs for each pattern switch because the proposed design has the advantages of less cost and race-free. The dynamic power consumption of the NMBE design [19] is larger than the traditional design [16] in some cases (especially "000") because of the strong 0/weak 1 problems. Table VI summarizes the average dynamic power consumption of pattern switches for generating one PP row. The NMBE design [19] has the largest dynamic power consumption due to the drawbacks of the poor voltage level. The results of the ABM-M1 design [22] in Table VI are simulated with approximation factor $m = 4$ (8-bit) and $m = 8$ (16-bit). For 8-bit (16-bit) case, compared to the traditional design [16] and the *neg-first* design [18], the proposed design can reduce the dynamic power consumption by 70.4% (75.4%) and 24.9% (32.1%), respectively; compared to the approximate design [22], the proposed design can reduce the dynamic power consumption by 43.3% (48.6%) and provide precise products.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3003684, IEEE Access

TABLE VI
COMPARISONS OF THE RADIX-4 BOOTH DESIGNS FOR GENERATING ONE PARTIAL PRODUCT ROW

	One PP row		The traditional design [16]	The *neg-first* design [18]	The NMBE design [19]	The ABM-M1 design [22]	The proposed design
8-bit	Average dynamic power (*W*)	Pre-encoder					2.69E-07
		Encoder	9.95E-07	6.42E-07	1.28E-06	9.76E-07	2.93E-07
		Decoders	5.12E-06	1.77E-06	4.91E-06	2.21E-06	1.25E-06
		Total	6.12E-06	2.41E-06	6.20E-06	3.19E-06	1.81E-06
	Average static power (*W*)	Pre-encoder					2.96E-08
		Encoder	7.12E-08	6.03E-08	1.13E-07	8.69E-08	3.32E-08
		Decoders	6.69E-07	2.15E-07	8.30E-07	2.68E-07	1.49E-07
		Total	7.40E-07	2.75E-07	9.44E-07	3.55E-07	2.11E-07
	Delay (ns)		0.0795	0.0791	0.1087	Exact: 0.0774 Approximate: 0.0745	0.0629
	Transistor count (TC)		394	182	182	196	164
16-bit	Average dynamic power (*W*)	Pre-encoder					3.05E-07
		Encoder	1.46E-06	9.06E-07	1.54E-06	1.22E-06	3.38E-07
		Decoders	1.07E-05	3.52E-06	1.14E-05	4.62E-06	2.36E-06
		Total	1.22E-05	4.42E-06	1.30E-05	5.84E-06	3.00E-06
	Average static power (*W*)	Pre-encoder					3.21E-08
		Encoder	8.34E-08	6.89E-08	1.52E-07	9.84E-08	3.63E-08
		Decoders	1.31E-06	3.96E-07	2.01E-06	5.09E-07	2.69E-07
		Total	1.39E-06	4.64E-07	2.16E-06	6.08E-07	3.37E-07
	Delay (ns)		0.1142	0.0944	0.1399	Exact: 0.0980 Approximate: 0.0962	0.0749
	Transistor count (TC)		714	310	326	332	272

Fig. 14 shows the static power consumption for each pattern. Take "000" for example, the multiplier Y (y_{2i+1}, y_{2i}, y_{2i-1}) is fixed at 000 to get the static power consumption of "000" pattern. As shown in Fig. 14, in patterns "000" and "111", the proposed design can save more static power consumption since the proposed pre-encoder can turn off the encoder and decoders in these patterns (the "0*X*" case). The static power consumption of the NMBE design [19] is larger than the traditional design [16] in some cases (especially "000") because of the drawbacks of the poor voltage level. Table VI summarizes the average static power consumption of the eight patterns for one PP row. The NMBE design [19] still has the largest static power consumption. For 8-bit (16-bit) case, compared to the traditional design [16] and the *neg-first* design [18], the proposed design can reduce the static power consumption by 71.5% (75.8%) and 23.3% (27.4%), respectively; compared to the approximate design [22], the proposed design can reduce the static power consumption by 40.6% (44.6%).

C. PERFORMANCE AND COST ANALYSES

The performance and cost analyses for one PP row are summarized in Table VI. The worst case delay is the metric to evaluate the performance. For example, the worst case delay of the *neg-first* design [18] will occur when signal $\overline{two_i}$ changes from 0 to 1 or 1 to 0. The worst case delay of the proposed design will occur when the encoded signals change from "0*X*" to "non-0*X*" case. As shown in Table VI, the NMBE design [19] has the longest delay due to the weak drivability of signals. For 8-bit (16-bit) case, compared to the traditional design [16] and the *neg-first* design [18], the delay reductions of the proposed design are 20.9% (34.4%) and 20.5% (20.7%), respectively. As shown in Table VI, the worst case delay of the ABM-M1 design [22] is contributed by the exact decoder. Compared to the ABM-M1 design [22], the delay reduction of the proposed design is 18.7% (23.6%).

Table VI also provides the transistor count (TC) to evaluate the costs of each design. For one *n*-bit PP row, different from the other designs, both of the *neg-first* design [18] and the proposed design need $n + 2$ decoders (one of them is the compact decoder). Take 8-bit for example, the *neg-first* design [18] needs 182 transistors for one PP row which is composed of one 30T encoder, one 8T compact decoder, and nine 16T decoders. The proposed design needs 164 transistors for one PP row which is composed of one 17T pre-encoder, one 18T encoder, one 8T compact decoder, nine 13T decoders, and four shared gating transistors (two groups). As shown in Table VI, the traditional design has the largest TC; the other designs can effectively reduce the costs and the proposed design has the least TC.

D. COMPARISON OF RADIX-4 BOOTH MULTIPLIERS

In order to show the effectiveness of our design, we also provide the comparisons of the radix-4 Booth multipliers as shown in Table VII. For a fair comparison, each design adopts the same adder array for PP accumulation as shown in Fig. 1. Only the multiplier of the ABM-M1 design [22] is the approximate multiplier, and the approximation factor $m = 4$ for 8-bit multiplication ($m = 8$ for 16-bit multiplication). Take 16-bit multiplication with $m = 8$ for example, the PPs with a significance less than 8 are generated by PPG-2Ss and the remaining PPs are generated by the exact decoders.

Compared to the traditional design [16], the other designs can reduce the dynamic power consumption and TC. However, the NMBE design [19] is the worst in terms of static power consumption and delay because of the drawbacks of the poor

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3003684, IEEE Access

TABLE VII
COMPARISONS OF THE *N*-BIT RADIX-4 BOOTH MULTIPLIERS

	8-bit Radix-4 Booth Multiplier					16-bit Radix-4 Booth Multiplier				
	Average dynamic power (*W*)	**Average static power (*W*)**	**Delay (ns)**	**PDP (f*J*)**	**Transistor count (TC)**	**Average dynamic power (*W*)**	**Average static power (*W*)**	**Delay (ns)**	**PDP (f*J*)**	**Transistor count (TC)**
The traditional design [16]	4.24E-05	4.63E-06	0.4135	17.53	2772	1.96E-04	1.68E-05	0.7818	153.23	9972
The *neg-first* design [18]	2.53E-05	2.41E-06	0.4131	10.45	1924	1.16E-04	8.27E-06	0.7620	88.39	6740
The NMBE design [19]	3.92E-05	1.28E-05	0.4427	17.35	1924	1.86E-04	2.94E-05	0.8075	150.20	6868
The ABM-M1 design [22]	3.24E-05	2.90E-06	0.4114	13.33	2040	1.50E-04	9.92E-06	0.7656	114.84	7180
The proposed design	2.38E-05	1.78E-06	0.3969	9.45	1852	1.08E-04	5.84E-06	0.7424	80.18	6436

voltage level that increases the leakage currents and delay. Because the proposed pre-encoded design has the advantages of race-free, conditional power gating, and low cost, the proposed design is the best design in terms of dynamic power, static power, delay, power-delay product (PDP), and TC as shown in Table VII. For 8-bit (16-bit) case, compared to the traditional design [16], the proposed design can reduce the dynamic power and static power by 43.9% (44.9%) and 61.6% (65.2%), respectively, with 4.0% (5.0%) performance improvement. Compared to the *neg-first* design [18], the proposed design can reduce the dynamic power and static power by 5.9% (6.9%) and 26.1% (29.4%), respectively. Compared to the approximate design [22], the proposed design can provide the exact results, and can reduce the dynamic power and static power by 26.5% (28.0%) and 38.6% (41.1%), respectively. Obviously, the proposed design has the lowest PDP results. Fig. 13, Fig.14, Table VI, and Table VII prove the superiority of the proposed design compared to the other designs.

V. CONCLUSION

In this paper, a low power radix-4 Booth pre-encoded mechanism has been proposed to reduce the unnecessary switching activities of encoders and decoders in the "0*X*" case. The proposed pre-encoded mechanism can detect the "0*X*" case earlier and adopts the gating techniques. When the "0*X*" case occurs, the encoder and decoders will be turned off immediately by the proposed pre-encoder to reduce the power consumption and leakage currents. The simulations are done by HSPICE with the TSMC 40 nm technology and the results show that the proposed design can provide significant reductions in power consumption, delay, and transistor count compared with the state-of-the-art encoded designs. For 16-bit multiplication, compared to the traditional radix-4 Booth multiplier, the proposed pre-encoded mechanism has 35% reduction in transistor count, 5% improvement in performance, and can reduce dynamic and static power consumption by 45% and 65%, respectively. Compared to the *neg-first* design and the NMBE design, the proposed design has better performance, less transistor count, and lower power consumption. Even compared to the approximate design, the proposed design can provide precise products, and can achieve 28% dynamic power reduction and 41% static power reduction.

REFERENCES

[1] Gang-Neng Sung, Yan-Jhih Ciou, and Chua-Chin Wang, "A Power-Aware 2-Dimensional Bypassing Multiplier Using Cell-Based Design Flow," in *Proc. 2008 IEEE International Symposium on Circuits and Systems*, May 2008, pp. 3338–3341.

[2] Guan-Lin Jiang, Tung-Chi Wu, and Yen-Jen Chang, "Low Power Multiplier with Alternative Bypassing Implementation," in *Proc. International Conference on Embedded Systems and Applications (ESA)*, July 2012, pp. 77-82.

[3] Xiaoping Cui, Weiqiang Liu, Xin Chen, Earl E. Swartzlander Jr., and Fabrizio Lombardi, "A Modified Partial Product Generator for Redundant Binary Multipliers," *IEEE Transactions on Computers*, vol. 65, no. 4, pp. 1165–1171, Apr. 2016.

[4] Duncan J. M. Moss, David Boland, and Philip H. W. Leong, "A Two-Speed, Radix-4, Serial–Parallel Multiplier," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 1–9, Dec. 2018.

[5] Bert Moons and Marian Verhelst, "An Energy-Efficient Precision-Scalable ConvNet Processor in 40-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 4, pp. 903–914, Apr. 2017.

[6] Jihyuck Jo, Suchang Kim, and In-Cheol Park, "Energy-Efficient Convolution Architecture Based on Rescheduled Dataflow," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp. 4196–4207, Dec. 2018.

[7] Chip-Hong Chang, Jiangmin Gu, and Mingyan Zhang, "Ultra Low-Voltage Low-Power CMOS 4-2 and 5-2 Compressors for Fast Arithmetic Circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 10, pp. 1985-1997, Oct. 2004.

[8] Peiman Aliparast, Ziaadin D. Koozehkanani, and Farhad Nazari, "An Ultra High Speed Digital 4-2 Compressor in 65-nm CMOS," *International Journal of Computer Theory and Engineering*, vol. 5, no. 4, pp. 593-597, Aug. 2013.

[9] Yen-Jen Chang, Yu-Cheng Cheng, Yi-Fong Lin, Shao-Chi Liao, Chun-Hsiang Lai, and Tung-Chi Wu, "Imprecise 4-2 Compressor Design Used in Image Processing Applications," *IET Circuits, Devices & Systems*, vol.13 , no. 6, pp. 848-856, Sep. 2019.

[10] Jongsu Park, San Kim, and Yong-Surk Lee, "A Low-Power Booth Multiplier Using Novel Data Partition Method," in *Proc. 2004 IEEE Asia-Pacific Conference on Advanced System Integrated Circuits*, Aug. 2004, pp. 54–57.

[11] Issam S. Abu-Khater, Abdellatif Bellaouar, and M. I. Elmasry, "Circuit techniques for CMOS low-power high-performance multiplier'', *IEEE Journal of Solid-State Circuits*, vol. 31, no. 10, pp.1535-1546, Oct. 1996.

[12] Miloš D. Ercegovac and Tomás Lang, *Digital Arithmetic*, 1st ed., Los Altos, CA, USA: Morgan Kaufmann Publishers, 2003.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3003684, IEEE Access

[13] Paul J. Song and Giovanni De Micheli, "Circuit and Architecture Trade-offs for High-speed Multiplication", *IEEE Journal of Solid-State Circuits*, vol. 26, no. 9, pp. 1184-1198, Sep. 1991.
[14] Gensuke Goto, Tomio Sato, Masao Nakajima, and Takao Sukemura, "A 54 x 54-b Regularly Structured Tree Multiplier", *IEEE Journal of Solid-State Circuits*, vol. 27, no. 9, pp. 1229- 1235, Sep. 1992.
[15] Norio Ohkubo, Makoto Suzuki, Toshinobu Shinbo, Toshiaki Yamanaka, Akihiro Shimizu, Katsuro Sasalu, and Yoshinobu Nakagome, "A 4.4 ns CMOS 54 x 54-b Multiplier Using Pass-Transistor Multiplexer", *IEEE Journal of Solid-State Circuits*, vol. 30, no. 3, pp. 251-257, Mar. 1995.
[16] Rafael Fried, "Minimizing energy dissipation in high-speed multipliers," in *Proc. 1997 International Symposium on Low Power Electronics and Design*, Aug. 1997, pp. 214-219.
[17] Wen-Chang Yeh and Chein-Wei Jen, "High-Speed Booth Encoded Parallel Multiplier Design," *IEEE Transactions on Computers*, vol. 49, no. 7, pp. 692-701, July 2000.
[18] Zhijun Huang and Miloss D. Ercegovac, "High-Performance Low-Power Left-to-Right Array Multiplier Design," *IEEE Transaction on Computers*, vol. 54, no. 3, pp. 272-283, Mar. 2005.
[19] Ravindra P. Rajput and M.N. Shanmukha Swamy, "High Speed, Efficient Area, Low Power Novel Modified Booth Encoder Multiplier for Signed-Unsigned Number", *Advances in Intelligent Systems and Computing*, vol.464, pp. 321-333, Apr. 2016.
[20] Honglan Jiang, Jie Han, Fei Qiao, and Fabrizio Lombardi, "Approximate Radix-8 Booth Multipliers for Low-Power and High-Performance Operation", *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2638–2644, Aug. 2016.
[21] Weiqiang Liu, Liangyu Qian, Chenghua Wang, Honglan Jiang, Jie Han, and Fabrizio Lombardi, "Design of Approximate Radix-4 Booth Multipliers for Error-Tolerant Computing", *IEEE Transactions on Computers*, vol. 66, no. 8, pp. 1435–1441, Aug. 2017.
[22] Suganthi Venkatachalam, Elizabeth Adams, Hyuk Jae Lee, and Seok-Bum Ko, "Design and Analysis of Area and Power Efficient Approximate Booth Multipliers", *IEEE Transactions on Computers*, vol. 68, no. 11, pp. 1697–1703, Nov. 2019.
[23] Andrew D. Booth, "A Signed Binary Multiplication Technique", *The Quaterly Journal of Mechanics and Applied Mathematics*, vol.4, no. 2, pp. 236-240, 1951.
[24] O. L. Macsorley, "High-Speed Arithmetic in Binary Computers," *Proceedings of the IRE*, vol.49, no.1, pp. 67-91, Jan. 1961.
[25] Jan M. Rabaey, Anantha Chandrakasan, and Borivoje Nikolic, *Digital Integrated Circuits*, 2nd ed., Upper Saddle River, NJ, USA: Prentice Hall, 2002.
[26] Nam Sung Kim, Todd Austin, David Blaauw, Trevor Mudge, Krisztián Flautner, Jie S. Hu, Mary Jane Irwin, Mahmut Kandemir, and Vijaykrishnan Narayanan, "Leakage Current: Moore's Law Meets Static Power," *IEEE Computer*, vol. 36, no. 12, pp. 68-75, Dec. 2003.
[27] Semiconductor Industry Association, International Technology Roadmap for Semiconductors (ITRS), 2015 [Online]. Available: http://www.itrs2.net/
[28] Shinichi Nishizawa, Tohru Ishihara, and Hidetoshi Onodera, "Analysis and comparison of XOR cell structures for low voltage circuit design," in *Proc. International Symposium on Quality Electronic Design (ISQED)*, Mar. 2013, pp. 703-708.

Yen-Jen Chang (M'02) received the Ph.D. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2003. He joined the faculty of the Department of Computer Science and Engineering, National Chung Hsing University, Taichung, Taiwan, in 2004, where he is currently a Professor. He has served as the department chair since Aug. 2019. His research interests include VLSI design, low-power memory design, embedded system, and SoC design.

Yu-Cheng Cheng received the B.S., M.S., and Ph.D. degrees in computer science and engineering from National Chung Hsing University (NCHU), Taichung, Taiwan, in 2011, 2013, and 2019, respectively.

His research interests include computer and microprocessor architecture, digital integrated circuit design, low-power memory design, and approximate circuit design.

Shao-Chi Liao received the B.S. degree in physics from National Changhua University of Education (NCUE), Changhua, Taiwan, in 2009, and M.S. degrees in computer science and engineering from National Chung Hsing University (NCHU), Taichung, Taiwan, in 2015. He is currently a testing engineer at Winbond Electronics Corporation.

His research interests include digital integrated circuit design, low-power integrated circuit design, and Booth multiplier.

Chun-Huo Hsiao received the B.S. degree in computer science from National Taichung University of Education (NTCU), Taichung, Taiwan, in 2018. He is currently pursuing the M.S. degree in Department of Computer Science and Engineering at National Chung Hsing University (NCHU).

His research interests include digital integrated circuit design, Booth multiplier, and approximate circuit design.