

Low - error Fixed-Width Modified Booth Multipliers for DSP applications

A. Poli Reddy¹

M.Tech, ECE Dept, PBR VITS, Kavali, AP-INDIA.

K.V. Yateendranath²

Assoc Prof, ECE Dept, PBR VITS, Kavali, AP-INDIA.

Abstract: In this paper, a single compensation formula of adaptive conditional-probability estimator (ACPE) applied to fixed-width Booth multiplier is proposed. Based on the conditional probability theory, the ACPE can be easily applied to large length Booth multipliers (such as 32-bit or larger) for achieving a higher accuracy performance. To achieve great results between accuracy and area cost, the ACPE provides varying column information w to adjust the accuracy with respect to system requirements. Furthermore, the ACPE Booth multipliers are applied to two-dimensional (2-D) discrete cosine transform (DCT) to evaluate the system performance.

Keywords--- Booth Encoder, Partial product generator, Fixed-width multiplier, Modified Booth multiplier, Compression tree, Carry Look ahead Adder.

I. INTRODUCTION

The main Objective of this paper is to reduce the number of partial products and truncation error by using the modified booth multiplier. Where multipliers are always the fundamental arithmetic unit and significantly influence the system's performance and power dissipation, the modified Booth encoding which reduces the number of partial products by factor of two through performing the multiplier recoding has been widely adopted in parallel multipliers. In DSP Applications

Modified Booth Multipliers plays an Vital role for getting good performance and little truncation error. To perform n -bit multiplication in previous days we have to do n -partial product terms and also we need large number of adder cells. For the sake of this concept the system having large propagation delay and more power consumption. This multiplication process getting different innovations according to the different trends. In those some of them are mentioned below.

The Existing Method named as Direct Truncated Fixed width Multiplier (DTFM)- As the name indicates Truncation operation will done in this method but the resultant product having large truncation error. Here we have to large number of logic gates, buffer gates and adder cells then automatically it requires large power source.

Significant hardware complexity reduction and power saving can be achieved by directly removing the adder cells of standard multiplier for the computation of the N least significant bits of $2N$ -bit output product. However, a huge truncation error will be introduced to direct-truncated fixed-width multiplier (DTFM).

To effectively reduce the truncation error, various error compensation methods, which add estimated compensation value to the carry inputs of the reserved adder cells, have been proposed. The error compensation value can be produced by the constant scheme or the adaptive scheme.

Logic Utilization	Used
Number of Slice Flip Flops	34
Number of 4 input LUTs	208
Logic Distribution	
Number of occupied Slices	109
Number of Slices containing only related logic	109
Number of Slices containing unrelated logic	0
Total Number of 4 input LUTs	208
Number of bonded IOBs	68
IOB Flip Flops	54
Number of GCLKs	1
Number of GCLKIOBs	1
Total equivalent gate count for design	2,021
Additional JTAG gate count for IOBs	3,312

Fig: Required Components for DTFM

Drawbacks of DTFM is High hardware complexity due to number of slices usage, More partial product array, More power consumption

II FIXED WIDTH MODIFIED BOOTH MULTIPLIER

The proposed high-accuracy fixed width multiplier comprises of (i) Booth Encoder (ii) Partial Product generator and (iii) Compression Tree comprising of Carry Look Ahead Adder. The proposed module outperform the existing module by hardware complexity, power consumption and performance speed. The number of hardware is reduced in each of the existing module by the proposed system. The existing system uses large number of hardware that is full adder and half adder for the final product generation whereas the proposed system uses four fixed width Carry Look Ahead Adder

This modified booth multiplier is to produce at most $n/2+1$ partial products, In Fixed Width Modified Booth Multiplier (FWMBM) we have to observe some Special parameters below, The booth encoder circuit takes the input multiplicand X and produces mul, shift and two com signal which is used by the partial product generation circuit to produce the partial product bits of one dimensional array.

X_{i+1}	X_i	X_{i-1}	Action
0	0	0	$0 \times Y$
0	0	1	$1 \times Y$
0	1	0	$1 \times Y$
0	1	1	$2 \times Y$
1	0	0	$-2 \times Y$
1	0	1	$-1 \times Y$
1	1	0	$-1 \times Y$
1	1	1	$0 \times Y$

Fig: Booth Encoder Table

By using this encoding table we have to follow some steps in multiplication process as follows

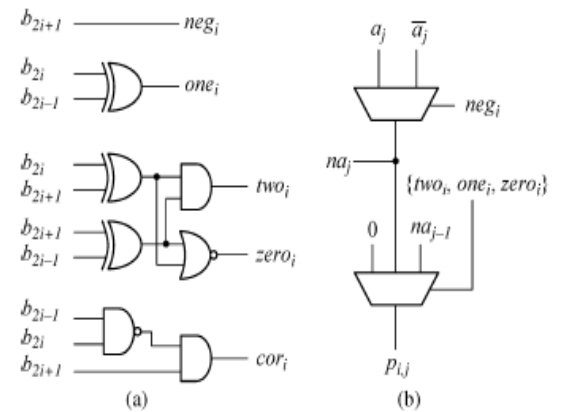


Fig. 1. (a) Modified Booth encoder. (b) Partial product generation circuit.

Algorithm:

1. Pad the LSB with one zero.
2. Pad the MSB with 2 zeros if n is even and 1 zero if n is odd.
3. Divide the multiplier into overlapping groups of 3-bits.
4. Determine partial product scale factor from modified booth 2 encoding table.
5. Compute the Multiplicand Multiples
6. Sum Partial Products

Let us consider the multiplication operation of two n-bit signed numbers are $X = x_{n-1}, x_{n-2} \dots x_0$ (multiplicand) and $Y = y_{n-1}, y_{n-2} \dots y_0$ (multiplier). The two's complement representations of X and Y can be expressed as follows:

$$X = -x_{n-1}2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i,$$

$$Y = -y_{n-1}2^{n-1} + \sum_{i=0}^{n-2} y_i 2^i.$$

The functional model design of the booth multiplier consists of first, booth encoder for encoding the multiplicand/multiplier. Second, if the multiplicand and multiplier are of n-bits, partial product generator generates (0.....n/2-1) n/2 number of partial product bits which are a one dimensional array. Third, compression tree consists of 9-bit, 12-bit and 16-bit carry Look Ahead generator to produce final output product.

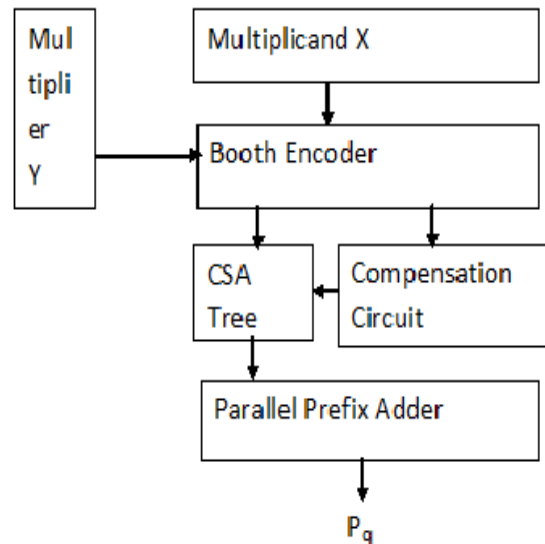


Fig: Block diagram of Booth Multiplier

The above is Block Diagram of Booth Multiplier, Where Multiplier is Y and Multiplicand is X. Both X and Y is connected to the Booth Encoder. The purpose of Booth Encoder is used to generate the partial products. The compensation circuit used to reducing the errors. The Carry Save Adder (CSA) used to adding the partial products and saving the carry value. The parallel prefix adder is used to adjusting stage and also adding the partial products. The final product is getting through the p_q .

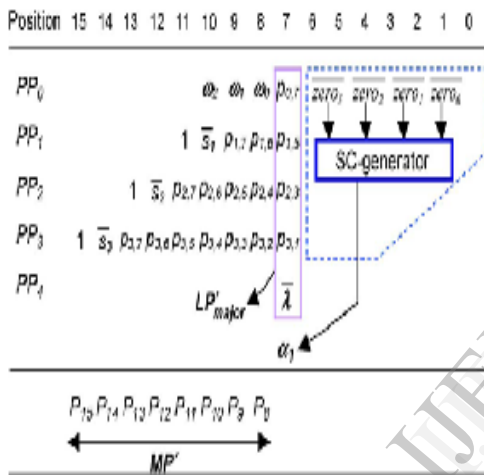


Fig: Partial Product Generator:

The above figure illustrates the final partial product matrix of proposed fixed width modified booth multiplier for n=8. In it, all the partial product bits in LP'_{minor} are removed and replaced by the SC generator. In addition, the carries generated by LP'_{minor} are also replaced by the outputs of SC generator

The Compression tree Compared to previous technique, the proposed error compensation circuit can achieve a tiny mean error and a significant reduction in mean square error. The smaller mean error and mean square error represent that the error distribution is more symmetric to and centralized in the error equal to zero (denoted as zero error).

SC-Generator:

The term SC-Generator is nothing but signal conditioning generator, the purpose of SC-generator is to check whether it is more number of 0's or 1's are present in the network.

Circuit Explanation:

The inputs of SC-Generator are \overline{zero} for $0 \leq i \leq n/2-1$ and it will generate m-output bits $\alpha_1, \alpha_2, \dots, \alpha_m$ as shown in above fig. Where $m = (n/2-1)/2$ and $s(\Phi)$ will be equal

0 0 0 0 0 1 0 0 0	8
x 0 0 0 0 1 0 1 0 0	20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0xY
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	1xY
1 0 0 0 0 0 0 0 0 1 0 0 0	1xY
0 0 0 0 0 0 0 0 0 0 0	0xY
0 0 0 0 0 0 0 0 0	0xY
0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0	160

to $\alpha_1 + \alpha_2 + \dots + \alpha_m$. Due to the subtraction operation in $[(R-1)/2]$, it is difficult to generate $\alpha_1, \alpha_2, \dots, \alpha_m$ by adder cells directly. Instead of adder cells, the proposed SC-generator is composed of a sorting network based on the following observation. We assume that zero for $0 \leq i \leq n/2-1$ can be sorted and the sorted outputs are p_j for $0 \leq j \leq n/2-1$. Moreover, if the largest bits are gathered to the less significant positions, then $\alpha_k = p_{2k}$ for $1 \leq k \leq m$. That is the problem of designing a SC generator can be translated into the design of a sorting network that sorts $Zero_{n/2-1} \dots zero_1 zero_0$ into $p_{n/2-1} \dots p_1 p_0$ and $\alpha_k = p_{2k}$

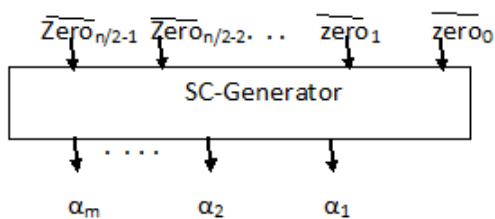


Fig: Error compensation circuit

There are 2 kinds of well known comparison based sorting networks, the bitonic and the odd-even merge sorting networks suited to hardware implementation. Since the odd-even merge sorter has the same number of compare levels as the bitonic sorter but requires fewer comparators, thus we adopt and simplify the odd-even merge sorting network to realize the SC-generator. These

sorting networks are composed of appropriately connected comparators. Each comparator takes in 2 input bits and either passes them directly or switches them. With inputs a and b, the outputs max(a, b) and min(a, b) of comparator correspond to $(a \vee b)$ and $(a \wedge b)$ respectively. After sorting for $n=8, \alpha_i$ are equal to the sorted outputs p_0, p_1, p_2, p_3 . The logic gates only for producing these outputs can be removed to simplify the SC-generator. Besides the sorting networks can be further simplified by using NAND, NOR, AND-OR INVERTER (AOI), and OR-AND INVERTER (OAI) gate for $n=8$ respectively. The SC generator for different n can be constructed in a similar fashion.

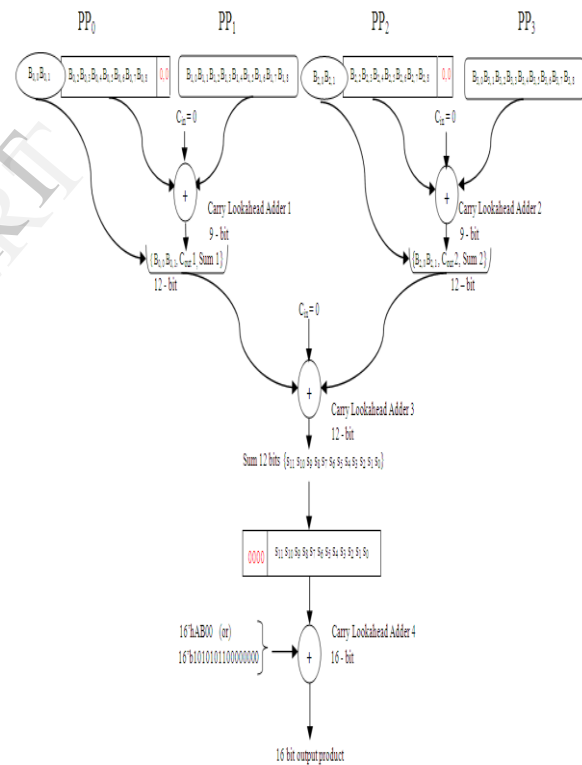


Fig. Compression Tree using Carry Look Ahead Adder

The compression tree is the one which adds the individual partial product bits generated by partial product generator using Carry Look ahead Adder and produce the final output product bits.

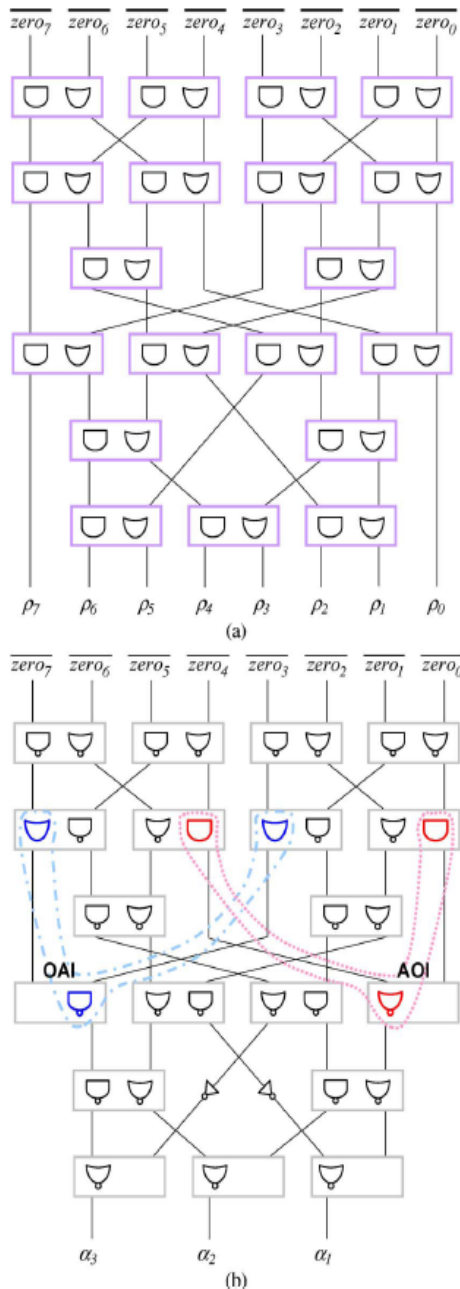


Fig. 7. (a) The odd-even merge sorting network for N=16. (b) The proposed SC-generator for N=16.

Here Low power consumption Architecture reduces the area of registers used in conventional multiplier are the main factors.

Applications:

1. Real-time signal processing
2. Audio signal processing
3. Video/image processing
4. Digital Signal Processing

III Experimental Results of Proposed Method

By using XILINX software for synthesis of DTFM and FWMBM to compare the area and power using VerilogHDL – Behavioral structure –

The area depends on number of components used in the circuit.

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Notes)
Number of 4 input LUTs	74	1,536	4%	
Logic Distribution				
Number of occupied Slices	39	768	5%	
Number of Slices containing only related logic	39	39	100%	
Number of Slices containing unrelated logic	0	39	0%	
Total Number of 4 input LUTs	74	1,536	4%	
Number of bonded IOBs	32	98	32%	
IOB Flip Flops	9			
Number of GCLKs	1	4	25%	
Number of GCLKIOBs	1	4	25%	
Total equivalent gate count for design	561			
Additional JTAG gate count for IOBs	1,584			

Fig : Required Components for FWMBM

Power requirement also depends on the value of electrical components.

Power summary:	I(mA)	P(mW)
Total estimated power consumption:		25
Vccint 1.80V:	10	18
Vcco33 3.30V:	2	7
Inputs:	0	0
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	10	18
Quiescent Vcco33 3.30V:	2	7

Fig : Power Analysis for FWMBM

In power comparison to execute an instruction in FWMBM it requires only 25mW power but for the same operation

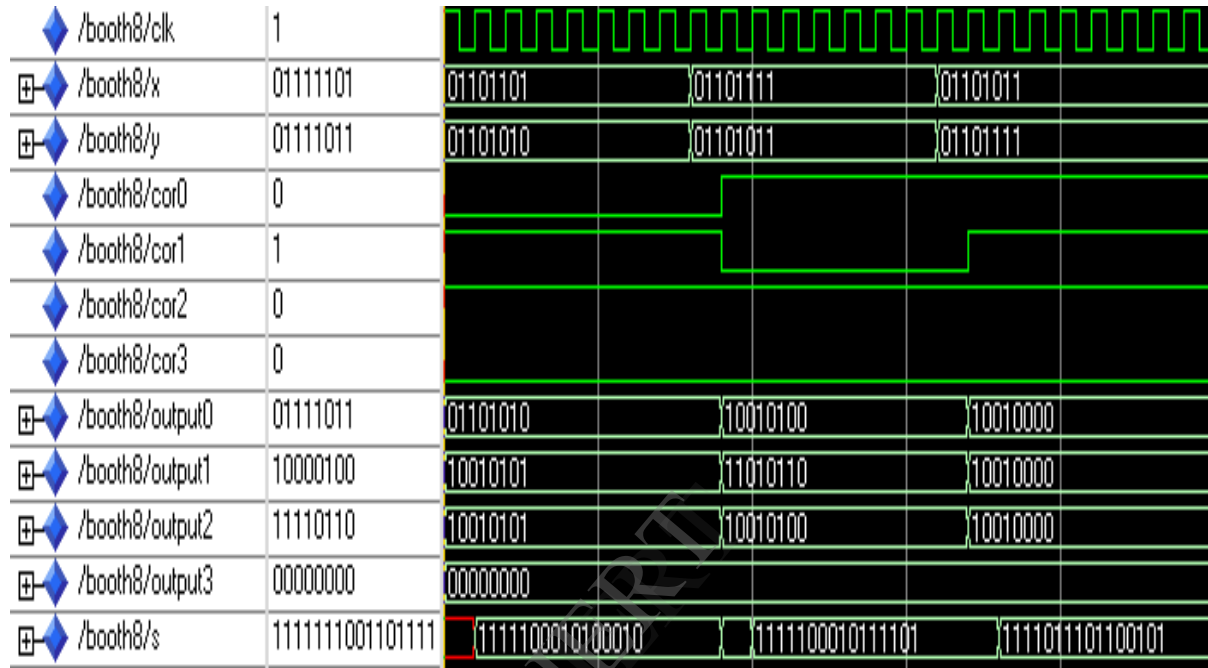
DTFM requires 218nW power it is one of the drawback of DTFM.

DTFM.....218 mW

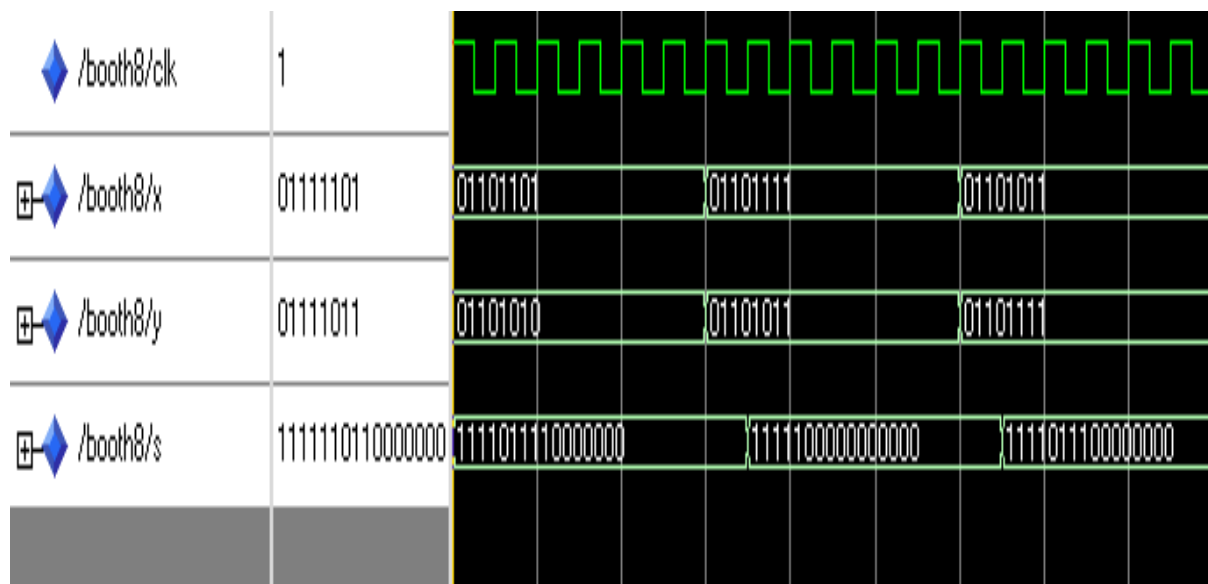
FWMBM... 25 mW

By using Model Sim we can perform simulation operation here we can observe the time delay . error compensation, and truncation error,mean & mean square errors

Direct truncated fixed width multiplier:



Fixed Width Modified Booth Multiplier:



The above Simulation waves shows the Output Product term By using DTFM and FWMBM

IV CONCLUSION

In this paper, a high accuracy fixed width modified booth multiplier has been proposed. In the proposed multiplier, the partial product matrix of booth multiplication was slightly modified and an effective error compensation function was derived accordingly. This compensation function makes the error distribution be more symmetric to and centralized in the error equals to Zero, leading the fixed width modified booth multiplier to very small mean and mean square errors. In this paper reduce number of partial products by using the booth encoding and also we can apply this concept for different 'n' values.

Presently technology used towards the reduced power consumption and area occupation which is very important in certain applications. Hopefully the present multiplier architecture may lead to the advanced technology in minimizing the area and reducing the power consumption.

References:

- [1] 52 IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, VOL. 19, NO. 1, JANUARY 2011 High-Accuracy Fixed-Width Modified Booth Multipliers for Lossy Applications Jiun-Ping Wang, Shiann-Rong Kuang, Member, IEEE, and Shish-Chang Liang
- [2] CHIP Implementation Center, CIC, Taiwan, CIC Referenced Flow for Cell-Based IC Design, Document no. CIC-DSD-RD-08-01, 2008.
- [3] M.-A. Song, L.-D. Van, and S.-Y. Kuo, "Adaptive low-error fixed width Booth multipliers," *IEICE Trans. Fundamentals*, vol. E90-A, no.6, pp. 1180–1187, Jun. 2007.
- [4] M. J. Schulte and E. E. Swartzlander, Jr., "Truncated multiplication with correction constant," in *Proc. VLSI Signal Processing*, VI, New York, 1993, pp. 388–396.
- [5] S. S. Kidambi, F. El-Guibaly, and A. Antoniou, "Area-efficient multipliers for digital signal processing applications," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 43, no. 2, pp. 90–94, Feb. 1996.
- [6] J. M. Jou, S. R. Kuang, and R. D. Chen, "Design of low-error fixedwidth multipliers for DSP applications," *IEEE Trans. Circuits Syst. I, Exp. Briefs*, vol. 46, no. 6, pp. 836–842, June 1999.
- [7] L. D. Van, S. S. Wang, and W. S. Feng, "Design of the low error fixedwidth multiplier and its application," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 47, no. 10, pp. 1112–1118, Oct. 2000.