VLSI Design, 2002 Vol. 14 (3), pp. 287–298

Taylor & Francis
Taylor & Francis Group

Low-power Application-specific Parallel Array Multiplier Design for DSP Applications

SANGJIN HONG[a,*], SUHWAN KIM[b] and WAYNE E. STARK[b]

[a]*Department of Electrical and Computer Engineering, State University of New York at Stony Brook, Stony Brook, NY 11794-2350, USA;* [b]*Department of Electrical Engineering and Computer Science, University of Michigan at Ann Arbor, Ann Arbor, MI 48105-2122, USA*

(Received 11 October 2000; Revised 14 January 2001)

Digital Signal Processing (DSP) often involves multiplications with a fixed set of coefficients. This paper presents a novel multiplier design methodology for performing these coefficient multiplications with very low power dissipation. Given bounds on the throughput and the quantization error of the computation, our approach scales the original coefficients to enable the partitioning of each multiplication into a collection of smaller multiplications with shorter critical paths. Significant energy savings are achieved by performing these multiplications in parallel with a scaled supply voltage. Dissipation is further reduced when conventional array multiplier is modified disabling the multiplier rows that do not affect the multiplication's outcome. We have used our methodology to design low-power parallel array multipliers for the Fast Fourier Transform (FFT). Simulation results show that our approach can result in significant up to 76% power savings over conventional array multipliers on 64-coefficient FFT computation.

Keywords: Low-power design; Multiplier architecture; Coefficient optimization; Fast Fourier transform; Digital signal processing; Voltage scaling

INTRODUCTION

Many common Digital Signal Processing (DSP) functional units such as Finite Impulse Response (FIR) filters and Fast Fourier Transform (FFT) modules perform extensive sequences of multiply-and-accumulate computations. Multipliers tend to be the most dissipative elements in these computations, and power-efficient multipliers are therefore essential for the design of low-power DSP hardware.

In this paper we present a low power design methodology that manipulates the bit-patterns of a DSP computation's coefficients to increase the energy efficiency of the multiplications performed. Specifically, our approach scales the values of the given coefficients to derive a representation that enables the partitioning of the original multiplication into several, smaller multiplications that can be performed in parallel. The critical path of each small multiplication is shorter than the original one. The coefficient values are chosen so that a quantization error bound for the overall computation is satisfied. Therefore, the supply voltage can be reduced to decrease power dissipation while maintaining any given throughput and quantization error constraints. Additional reductions in power dissipation are achieved by disabling the portion of multiplier that do not affect the final result of the multiplication. For high-speed operation, the multiplier can be pipelined to achieve a low worst-case critical delay. Our design methodology focuses on multiplications where the number of coefficients is larger than the number of multipliers available in which case previously proposed techniques [1–4], that hardwire the coefficient bits to reduce circuit complexity and power are not applicable. Throughout the paper, we assume that it is possible to select arbitrary voltage in the multiplier design even though it may incur a problem of voltage compatibility in some implementation.

The remainder of this paper has six sections. In the second section we review previous research on power reduction approaches applied to multiplier design. The third section describes our methodology for designing low-power multipliers. The fourth section discusses

*Corresponding author. Tel.: +1-631-632-1160. Fax: +1-631-632-8494. E-mail: snjhong@ece.sunysb.edu

ISSN 1065-514X print/ISSN 1563-5171 online © 2002 Taylor & Francis Ltd
DOI: 10.1080/10655140290011087

implementation details, including the selective disabling of rows, and an analysis of the worst-case delay and power-dissipation of the multiplier. In the fifth section, we present results from the application of our methodology to the design of a multiplier for the 64-coefficient FFT. Numerical results show that the low-power multipliers proposed in this paper can be used to cut the power requirements of this computation in half without degrading its throughput or quantization error. Our contributions are summarized in the fourth section.

BACKGROUND ON LOW-POWER MULTIPLIERS

In this section, we describe previous research on low-power design for DSP multiplications. Specifically, we describe architectural-based approaches separately from coefficient-based ones, even though the distinction is not clear.

Architectural Approaches

To first order, the average power required to perform a DSP operation of type i is given by the expression

$$P_i = C_{\text{eff}} \times V_{\text{supply}}^2 \times f_{\text{clock}}, \tag{1}$$

where C_{eff} is the average effective capacitance switched per operation of type i (corresponding to adder, multiplexor, etc.), V_{supply} is a single operating supply voltage, and f_{clock} is the clock frequency which is fixed in many DSP and dedicated applications. Thus, given the fixed data sampling rate f_{clock}, an important design goal is to minimize the product term $C_{\text{eff}} \times V_{\text{supply}}^2$. The average effective switching capacitance C_{eff} is the product of load capacitance and its switching activity ratio [5]. Most of power reduction techniques attempt to reduce the power dissipation of multipliers by changing the power-affecting parameters of Eq. (1) in circuit, logic and architecture level [6–11].

The power dissipation of a multiplication operation is directly proportional to the switching activity of all internal node capacitances of the multiplier. The internal node capacitance switching activity depends on the multiplier input values (data samples and coefficients), thus the multiplier may be designed to eliminate any unnecessary switching activity by deactivating any circuitry that is not needed for the evaluation [12]. This approach may increase the physical capacitance due to the introduction of additional circuitry for deactivation control, but the effective switching capacitance C_{eff} may be reduced.

Power dissipation can also be reduced by reducing the supply voltage V_{supply} which influences the power dissipation quadratically. However, to maintain a given throughput, the delay penalty must be compensated since the multiplier delay T_{d} depends on the supply voltage as:

$$T_{\text{d}} = \frac{C_{\text{load}} \times V_{\text{supply}}}{I} \propto \frac{C_{\text{load}} \times V_{\text{supply}}}{(V_{\text{supply}} - V_{\text{th}})^2}, \tag{2}$$

where I is the supply current, V_{th} is the threshold voltage of the transistors, and C_{load} is the load capacitance. In recent short channel device, exponent of $(V_{\text{supply}} - V_{\text{th}})$ in the above equation is about 1.3. The proportionality factor comes from the given processing technology and is assumed to be fixed. In order to compensate for the delay penalty at the architectural level, a simple solution is to parallelize and/or pipeline the multiplier [12–14].

A conventional $M \times N$ multiplier can be duplicated with additional control circuitry to perform the multiplications at a half of the execution frequency. In this structure, the combined effect on the power equation is identical but the voltage scaling technique can be easily employed resulting in the reduction of the power dissipation. For example, let C_{before}, V_{before}, and f_{before} be the effective switching capacitance, supply voltage, and execution frequency of the original multiplier and let C_{after}, V_{after}, and f_{after} be the corresponding parameters for the parallel multiplier where, $C_{\text{after}} = 2C_{\text{before}} + C_{\text{overhead}}$ and $f_{\text{after}} = f_{\text{before}}/2$. Assuming C_{overhead} is negligible if the supply voltages $V_{\text{before}} = V_{\text{after}}$, the no power reduction has been achieved. Since the voltage scaling reduces supply voltage of the parallel multiplier due to reduced throughput requirement so that $V_{\text{after}} < V_{\text{before}}$, the power reduction achieved by the parallelizing the multiplier is a factor of $V_{\text{after}}^2/V_{\text{before}}^2$. Thus, this architectural change primarily reduces the operating frequency f_{clock} which results in the reduction of the operating supply voltage V_{supply} at the cost of increase in the switching capacitance C_{load}. One of the drawbacks of a large degree of parallelism is that the overhead complexity in terms of the overhead switching capacitance C_{overhead} may become larger than the power reduction gained by reducing the f_{clock} and V_{supply}^2 [15].

In the case where the scaled supply voltage reaches the minimum allowable voltage level V_{min} set by the processing technology, any further parallelization and/or pipelining do not improve power efficiency.

Coefficient Optimization

Even though substantial research has been devoted to low-power multiplier design at the circuit and logic level alone, power reduction can be enhanced by properly designing the coefficient bit patterns. The coefficient multiplication is very common in DSP where the multiplication requires random data samples and known set of coefficients. The class of the multiplication can be broadly categorized as fixed- and variable-coefficient multiplications. In the fixed-coefficient multiplication, the number of coefficients is equal to the number of multipliers available. Thus, the coefficients can be effectively hardwired into each multiplier. This type of multiplications is often encountered in DSP. One main drawback is that the complexity of

the overall architecture may increase significantly when the number of coefficients is very large. In the variable-coefficient multiplication, the number of coefficients is larger than the number of multiplier available. Each multiplier uses different coefficient in every multiplication. The variable-coefficient multiplication is often incorporated in the time-multiplexed multiplication such as in large transform size FFT or FIR filters and its coefficients are usually stored in a non-volatile memory.

Power reduction strategies for variable-coefficient multiplication differ from those for fixed-coefficient multiplication. In the fixed-coefficient multiplications, previous known approaches include minimization of nonzero coefficient bits by using either signed-magnitude or canonic signed digit number representations [5,12]. The minimization of the number of nonzero bits is very effective in reducing power, since the number of nonzero bits in a coefficient corresponds to the number of rows of adders when array multiplier is adopted. In addition, further minimization of nonzero bits is achieved by perturbing the coefficients by multiplying a scale factor [1–4,16]. So when the fixed coefficients are hardwired into the circuits, any unnecessary circuitry corresponding to zero coefficient bits can be removed. Moreover, the architecture-driven voltage scaling technique can be easily employed since the critical path influenced by the depth of the multiplier is reduced. However, in the variable-coefficient multiplications, a simple minimization of nonzero coefficient bits cannot be directly incorporated since the coefficient values cannot be hardwired.

Due to the linearity possessed by both FIR or FFT, scaling the output preserves the filter characteristics in terms of filtering performance, but the results in an overall magnitude gain equal to the scale factor. For scale factor α, the equations for FIR or FFT, respectively, can be written as

$$\alpha X[k] = \alpha \sum_{p=0}^{P-1} x[p].h[p-k] = \sum_{p=0}^{P-1} x[p].\alpha h[p-k] \quad (3)$$

$$\alpha X[k] = \alpha \sum_{p=0}^{P-1} x[p].W_P^{kp} = \sum_{p=0}^{P-1} x[p].\alpha W_P^{kp} \quad (4)$$

Thus the coefficients of the scaled FIR or FFT are given by $\alpha h[k]$, αW_P^{kp}, respectively. For the purpose of discussion, we will denote the coefficients as $h[k]$ to represent both FIR and FFT coefficients. This coefficient perturbation technique with the scaling factor α is widely used in digital filter designs [3,4]. Given an allowable range of scaling, an optimum scaling factor α can be found such that the total number of ones in the binary representations of $\alpha h[k]$ coefficients is less than the total number of ones in the binary representation of the $h[k]$ coefficients. This can save power in hardwired coefficient multiplication with an array multiplier.

Differential coefficient method [16] is another approach to reducing computational complexity and hence the power dissipation of DSP applications. The main goal is to reduce the Hamming distance between the consecutive coefficients so that the internal node switching is minimized. This is achieved by using various orders of differences between the coefficients in conjunction with stored pre-computed results rather than using the coefficients directly. The results in [16] indicate that for certain type of DSP applications this technique enables significant power reduction.

Another approach is to code the data samples and the coefficient differentially simultaneously so that the overall switching is low. Significant power saving can be achieved with this approach in cases where the Hamming distance between the consecutive data samples or the coefficient is very small. Thus, this method is most beneficial for data samples with high temporal correlation.

POWER REDUCTION METHODOLOGY

In this section, we highlight our power reduction methodology for multiplication using a fixed set of coefficients. We first describe how to reduce power dissipation by *coefficient clustering*. We then describe *coefficient partitioning* which can further reduce power dissipation by partitioning the original multiplication into several smaller ones.

Clustering

The *cluster width C* of an *N*-bit coefficient *Y* is defined as the distance between the first and the last nonzero bits in *Y*. Clustering confines the maximum among the cluster widths of the multiplication coefficients to a value $C_{\max}$ smaller than *N*. Consequently, the effective bit-width of the coefficient is reduced from *N* to $C_{\max}$. From a power dissipation standpoint, clustering results in two main benefits. First, the multiplier can be designed to reduce switching activity by ignoring the bit positions outside the cluster. Second, the supply voltage required to meet a given throughput requirement can be reduced, since the worst-case critical path among all multiplications is decreased. These benefits are applicable to most multiplier architectures but especially to array multipliers.

The effect of reducing *C* on the worst-case delay of the multiplier is illustrated in Fig. 1. As *C* decreases, the depth of the multiplier also decreases. The output of the $M \times C$ multiplier is shifted by an amount *S* equal to the width of zeros following the cluster (*S* controls the shifter). Since power dissipation reduces quadratically with the supply voltage, the reduction of *C* is intuitively expected to result in decreased dissipation primarily due to voltage supply scaling rather than due to the reduction in switching activity.

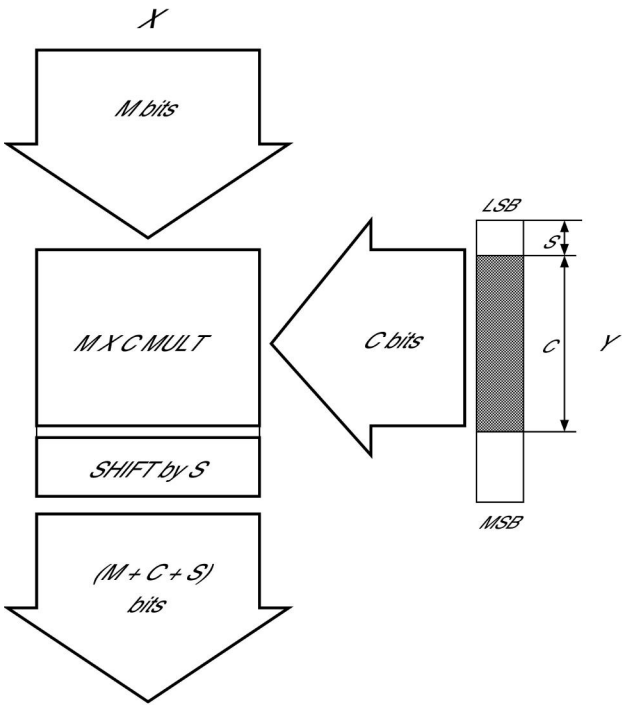

FIGURE 1 Effect of cluster width C on multiplier structure.

Partitioning

A particularly effective approach to lowering energy consumption is to operate at the lowest possible power supply voltage. At lower supply voltages, however, the individual circuit elements run slower, and circuit performance degrades. One way to maintain throughput at reduced voltages is to reduce the input-to-output delay by parallelization. In this subsection, we describe coefficient partitioning, a technique that can be used to parallelize coefficient multiplications.

Coefficient partitioning divides an N-bit coefficient $Y = y_{N-1}y_{N-2}\ldots y_0$ into an N_a-bit coefficient Y_a and an N_b-bit coefficient Y_b, where $Y_a = y_{N-1}y_{N-2}\ldots y_{N-N_a}$, $Y_b = y_{N_b-1}y_{N_b-2}\ldots y_0$, and $N = N_a + N_b$.

For the partitioned coefficient, we have

$$C_a + C_b \leq C, \tag{5}$$

$$0 < C_b < K < N - C_a - 1 < N - 1, \tag{6}$$

where C_a is the cluster width of Y_a, C_b is the cluster width of Y_b, C is the cluster width of Y, and K is the partition point. Inequalities (5) and (6) imply that coefficient partitioning can decrease coefficient cluster widths.

After coefficient partitioning, the original multiplication $Z = X \times Y$ is rewritten as $Z = X \times (Y_a \times 2^{N_b-1} + Y_b) = (X \times Y_a) \times 2^{N_b-1} + (X \times Y_b)$. Thus, the original $M \times N$ multiplication is turned into an $M \times N_a$ multiplication and an $M \times N_b$ multiplication which can be performed in parallel. These two parallelized multiplications can employ the coefficient clustering discussed previously to further reduce complexity as shown in Fig. 2. The depth of each resulting multiplication $X \times Y_a$ and $X \times Y_b$, which is C_a and C_b, respectively, is much less than that of the multiplication $X \times Y$. Two shifters and an additional adder are required to generate the final product. The output of the $M \times C_a$ multiplier is shifted by a number

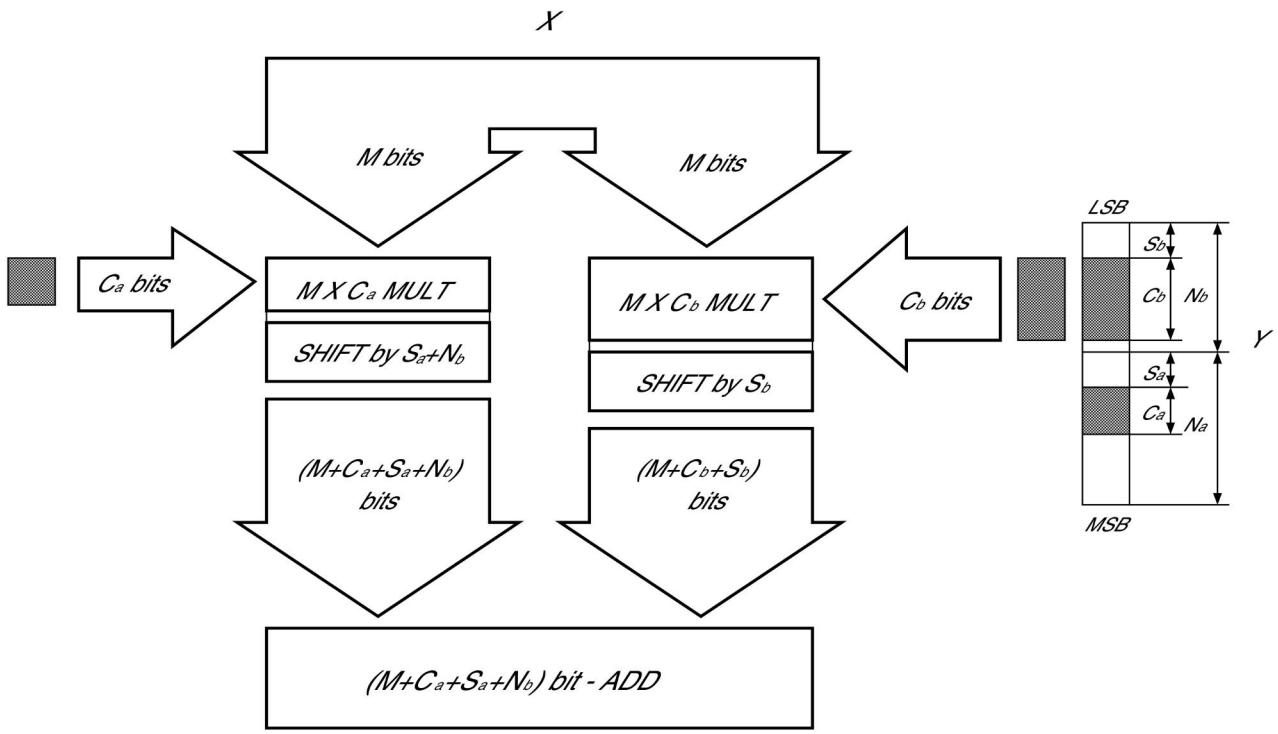

FIGURE 2 Parallel multiplication by coefficient partitioning.

of bits equal to the sum of N_b and the number S_a of zero bits between the LSB of Y_a and the LSB of the cluster. Similarly, the output of the $M \times C_b$ multiplication is shifted by the number S_b of zero bits preceding the cluster of Y_b. The outputs of the two partitioned multipliers must be shifted properly and added to correctly represent the final product. The power dissipation of the two shifters and the additional adder is offset by the savings that can be achieved due to the strictly smaller cluster widths C_a and C_b of the multiplications $X \times Y_a$ and $X \times Y_b$. These multiplications can also be pipelined to achieve the high-speed operation with low-power consumption.

In general, the delay of a multiplier increases with the maximum cluster width of its coefficients. Thus, the *optimal cluster partitioning problem* is to select the partition value K such that the maximum critical delay of two clusters is minimized. The computation of an optimum partition can thus be a particularly challenging problem from a computational standpoint, since the solution space is not convex.

Coefficient partitioning can be applied to generate more than two smaller multipliers. In this case, however, the dissipation overhead of the additional shifters and adders may reduce the gains of parallelization. Even though multi-way partitioning could still be beneficial for larger coefficient bit-widths, we only investigate 2-way partitioning in this paper. A heuristic algorithm for coefficient partitioning for example of FFT will be explained in the "Case study: FFT multiplier design" section. This is an reasonable class of partitions in practice, since the number of bits used is in the range 12–24 in most DSP applications.

LOW-POWER MULTIPLIER DESIGN

The degree of spatial and temporal correlation between the architectural attributes and the coefficient bit-patterns can significantly affect the power dissipation of the multiplier. In this section, we describe a multiplier design whose worst-case delay and switching activity are highly correlated with the cluster widths and the number of ones in the coefficients.

Conventional Array Multiplier

A generic $M \times N$ array multiplier operates by computing partial products in parallel and by shifting and accumulating the partial products. The multiplier width corresponds to the word length M of the data samples. The multiplier depth corresponds to the word length N of the coefficients. The switching activity of this multiplier is poorly correlated with the coefficient. In particular, if the k-th bit of a coefficient is 0, the k-th row of adders does not need to be activated. The partial product of the previous adder rows can simply be shifted and bypassed to the next row of adders. In this case, the function of the k-th row of adders is simply a one-bit shift of the partial products. The adders of a conventional array multiplier corresponding to zero coefficients are still switching, however, even though the addition is not required. The increased switching activity of the internal node capacitance results in unnecessary power dissipation.

Array Multiplier With Reduced Switching Activity and Critical Path

In order to reduce the power dissipated by the multiplier, the conventional array multiplier structure is modified by incorporating deactivation circuitry and bypass logic so that one-bit shifts can be performed without unnecessarily switching adder cells.

Figure 3 shows a 4 × 4 parallel multiplier with reduced switching activity (RSA) that uses the adder cells shown in Fig. 4. Two multiplexors added to the original adder cells route partial product terms to bypass the adders when the corresponding bits are zero. Three-state buffers are incorporated at the inputs of the adder cells to avoid any switching activity in the adder cells which are bypassed.

An L-bit adder, called a *vector-merge adder*, is used to generate the final result of the multiplication. The actual word-size of L may be less than the actual word-size of the product of the multiplication, which is determined by the applications depending on accuracy requirement (i.e. L MSBs are generated). For a given coefficient Y, the critical path delay $t_{\text{mult}}(Y)$ of the RSA multiplier is given by the expression

$$t_{\text{mult}}(Y) = Mt_{\text{carry},o} + (C_{\text{nz}} - 1)(t_{\text{carry},m} + t_{\text{mux}}) + ((C - 1) - (C_{\text{nz}} - 1))t_{\text{mux}} + t_{\text{and}}, \quad (7)$$

where $t_{\text{carry},o}$ and $t_{\text{carry},m}$ are the propagation delays between the input and output carry of the conventional and the RSA adder cells, respectively, t_{mux} is the delay of a multiplexor, t_{and} is the delay of an AND gate, and C_{nz} is the number of nonzero bits in the cluster of Y. The worst-case critical delay is defined, when C_{nz} is equal to C.

The critical delay and the power dissipation of our RSA multiplier are decreased by selecting coefficient representations with small cluster widths (i.e. small effective multiplier size) and a small number of nonzero bits in the cluster (i.e. fewer switching adders). Another approach for reducing the propagation delay of our RSA array structure is to reduce $Mt_{\text{carry},o}$ by using a fast adder implementation for the merge adder such as a carry-select or a carry-lookahead structure.

The RSA multiplier architecture presented in the above reduces the power dissipation over the conventional array multiplier by reducing the C_{eff} and the V_{supply}. However, the worst-case delay is still limited by the cluster width. For a high speed operation, the multiplier will not allow effective voltage scaling.

In order to reduce the power dissipated by the RSA multiplier by allowing the voltage scaling at the high operating speed requirement, the RSA multiplier structure

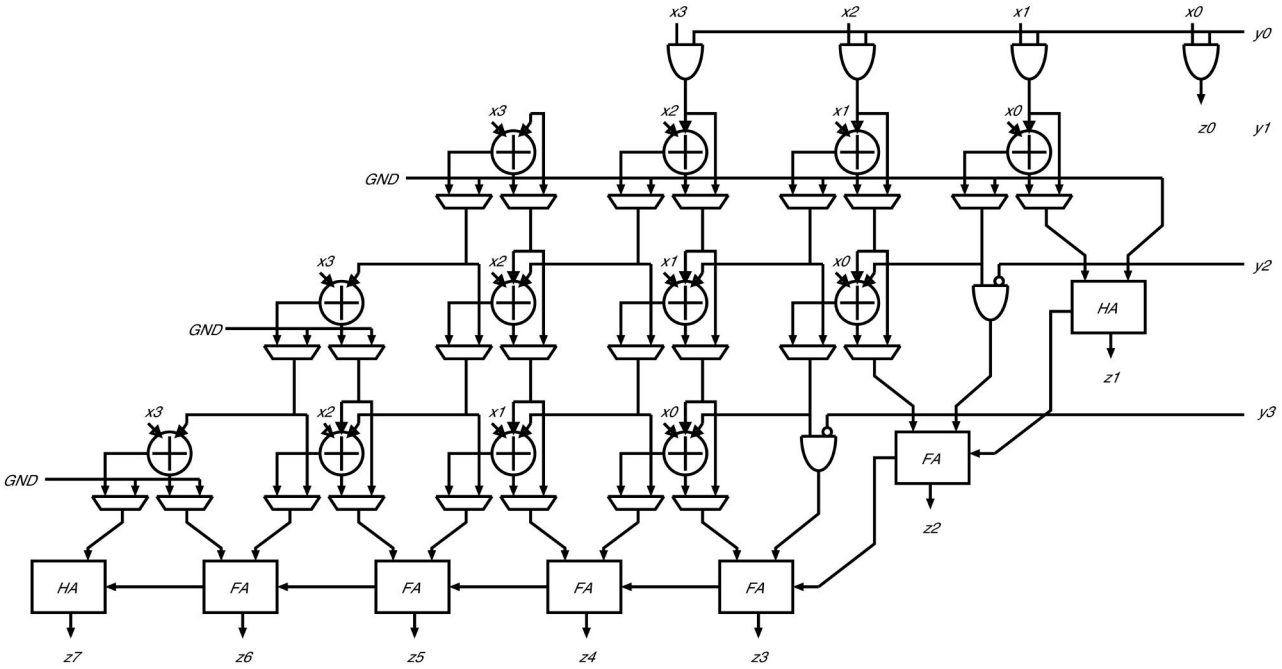

FIGURE 3 A 4 × 4 multiplier with reduced switching activity and critical path.

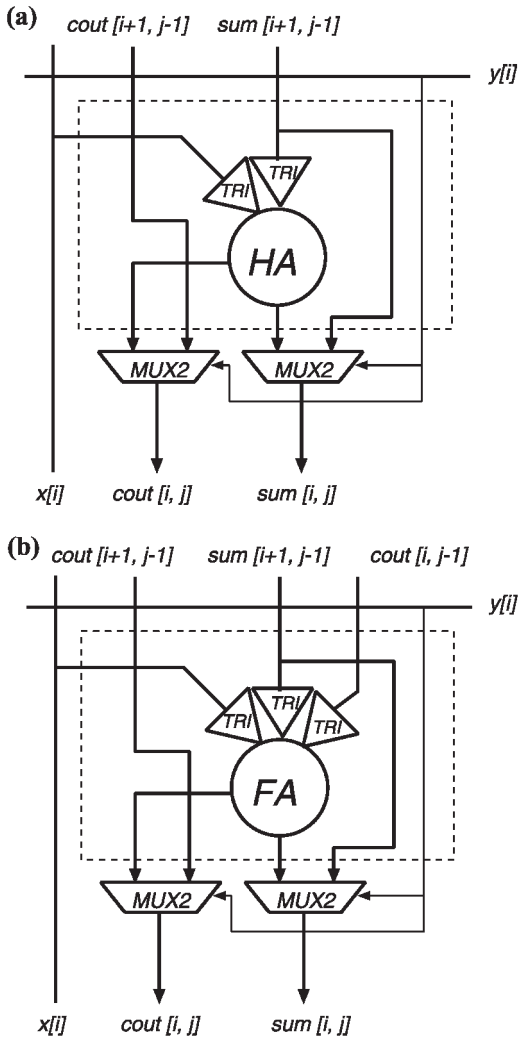

FIGURE 4 Reduced switching activity adder is shown in dotted box. (a) A half adder cell and (b) a full adder cell incorporated in the multiplier shown in Fig 3.

is modified again by pipelining. Additional circuitry latches, are incorporated into the adder cell at the output.

Figure 5 shows a 4 × 4 RSA pipeline multiplier that uses the RSA adder cells shown in Fig. 6. Three one-bit registers are added to the RSA adder cells with multiplexers.

For a given coefficient Y, the worst-case critical path delay $t_{\text{mult}}(Y)$ of the pipelined RSA multiplier is given by the expression

$$t_{\text{mult}}(Y) = t_{\text{carry},m} + t_{\text{mux}} + t_{\text{latch}}, \tag{8}$$

where $t_{\text{carry},m}$ are the propagation delays between the input and output carry of the RSA adder cells, t_{mux} is the delay of a multiplexor, and t_{latch} is the setup-time and clock-to-output of a latch. The number of stage is equal to the cluster width C.

The critical delay of the pipelined RSA multiplier is decreased by reducing the number of stages and consequently the power dissipation can be achieved through scaling the voltage at the high operating speed.

Throughput and Power-dissipation

The RSA multiplier in this paper was evaluated and characterized in terms of its throughput and dissipation using HSPICE simulation with RC-parameters extracted from standard cell layout using a 0.35-μm CMOS standard process technology. Figure 7(a)shows worst-case delay of the RSA multiplier when the supply voltage was varied from 1.2 to 3.3 V. The curves in the graph correspond to the two-way partitioned RSA multiplier with cluster widths $C = 3, 4, 5, 6$. Critical delay and average switching power dissipation of an individual RSA

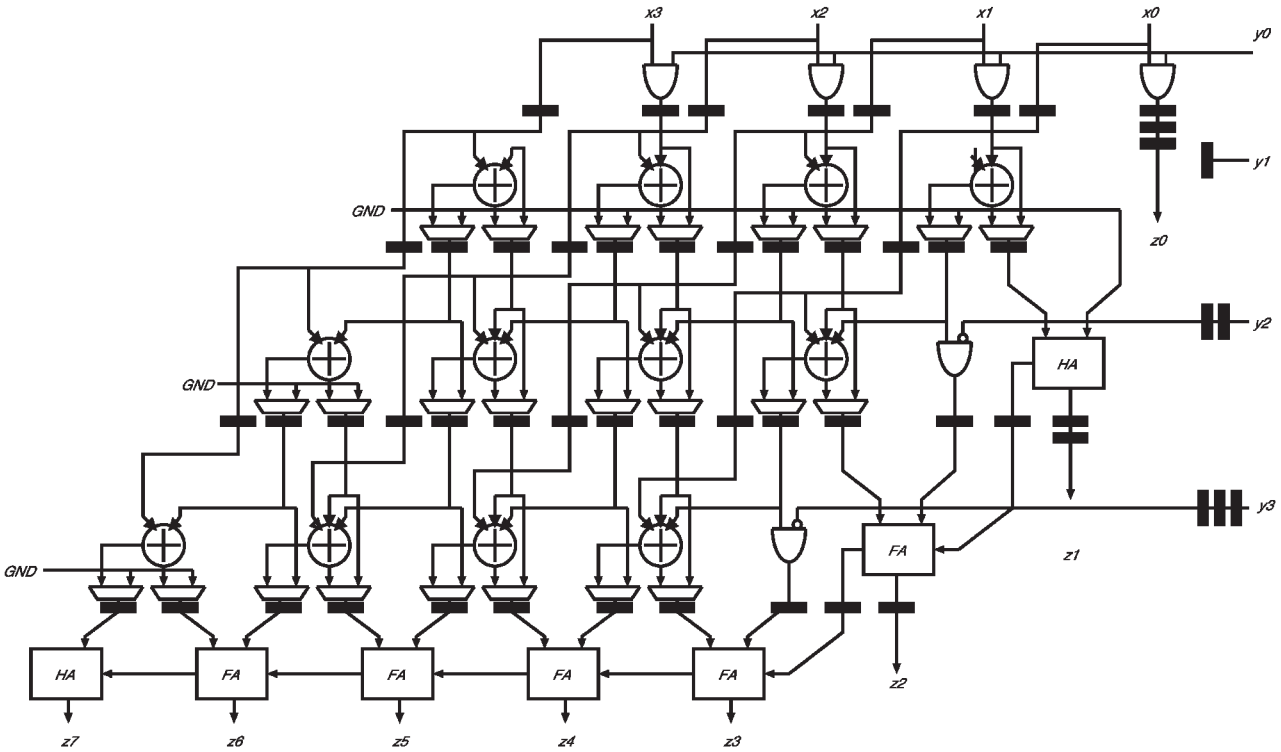

FIGURE 5 A pipelined 4 × 4 multiplier with reduced switching activity.

adder have been increased approximately by 15 and 60%, respectively, due to additional circuitry. The multiplier delay of the RSA multipliers are compared with 12 × 12 conventional array multiplier. It is evident in the figure that the critical delay of the RSA multiplier with cluster width of six is comparable to the delay of the conventional array multiplier due to the additional circuitry. Also shown in the figure are pipelined version of the RSA multiplier. Figure 7(b) shows the normalized average power-delay product of the RSA multiplier as well as the pipelined version of the RSA multipliers. As shown in these figures, the RSA multipliers generally have much lower power-delay product than that of the conventional array multiplier due to its lower critical delay and power dissipation. Even though the additional circuitry is incorporated into the pipelined RSA multiplier, its power-delay product is the lowest among the multipliers considered mainly because of its small delay. When the coefficients are optimized by partitioning, our RSA multipliers can result in significant power savings.

CASE STUDY: FFT MULTIPLIER DESIGN

In this section, we describe our low-power design methodology through the design of multipliers for the 64-coefficient FFT. First, we present our coefficient optimization algorithm. We subsequently give HSPICE simulation results from multiplier designs that were obtained using our algorithm.

Coefficient Optimization

The P-point FFT, including quantization noise and some scaling factor α, is given by the equation

$$\hat{X}[k] = \frac{1}{\alpha}\sum_{p=0}^{P-1} \alpha(X[p] + e_d[p])(W_P^{kp} + e_w[p]), \qquad (9)$$

where W_P^{kp} is the pth coefficient, $X[p]$ is the input data sample, $e_w[p]$ is the error due to twiddle-factor coefficient approximation, and $e_d[p]$ is the input sample quantization error.

Given a set $\mathcal{W}$ of infinite-precision twiddle-factor coefficients W_P^{kp}, $p = 0, 1, \ldots, P-1$, an error-ratio bound δ, a delay bound $T_{\max}$, and a cluster width bound C_{target}, our optimization process returns an encoded set of P twiddle-factor coefficients $Y_p = \alpha W_P^{kp}$, $p = 0, 1, \ldots, P-1$ such that $\max_p \{C_{a,p}\} \le C_{\text{target}}$, $\max_p \{C_{b,p}\} \le C_{\text{target}}$, the multiplier critical delay is less than $T_{\max}$, the coefficient quantization error ratio is less than δ, and the multiplier dissipation is minimal.

The pseudo-code of our heuristic coefficient optimization algorithm is given in Fig. 8. This algorithm comprises three nested loops. The outer loop steps through the possible scaling factors α. The middle loop steps through the possible partition points K. The inner loop steps through the possible number representations f (two's complement or canonic signed digit representations). For each α, K, and f, the algorithm partitions the encoded coefficients with respect to the current partition point. It subsequently compares its power dissipation,

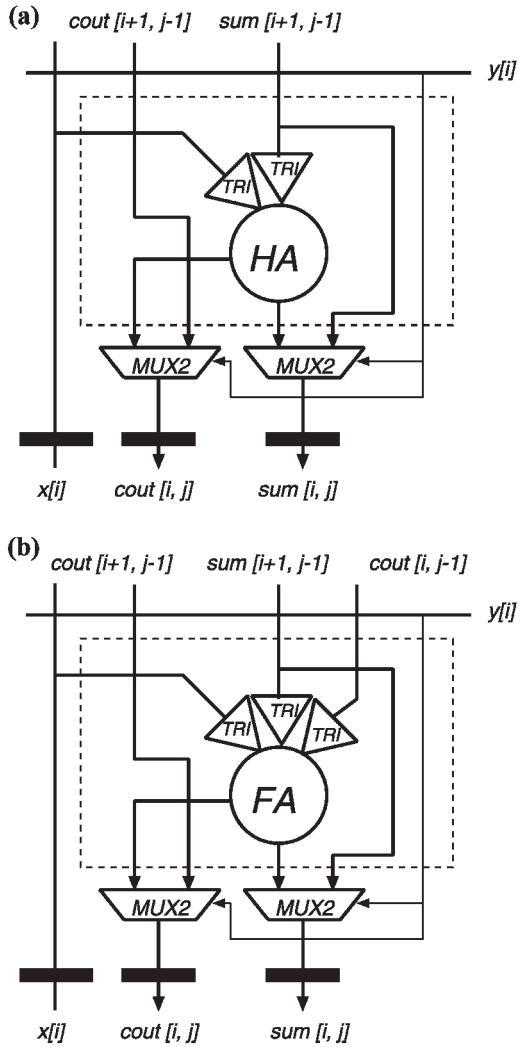

FIGURE 6 Reduced switching activity adder is shown in dotted box. (a) A half adder cell and (b) a full adder cell incorporated in the pipeline RSA multiplier shown in Fig. 5.

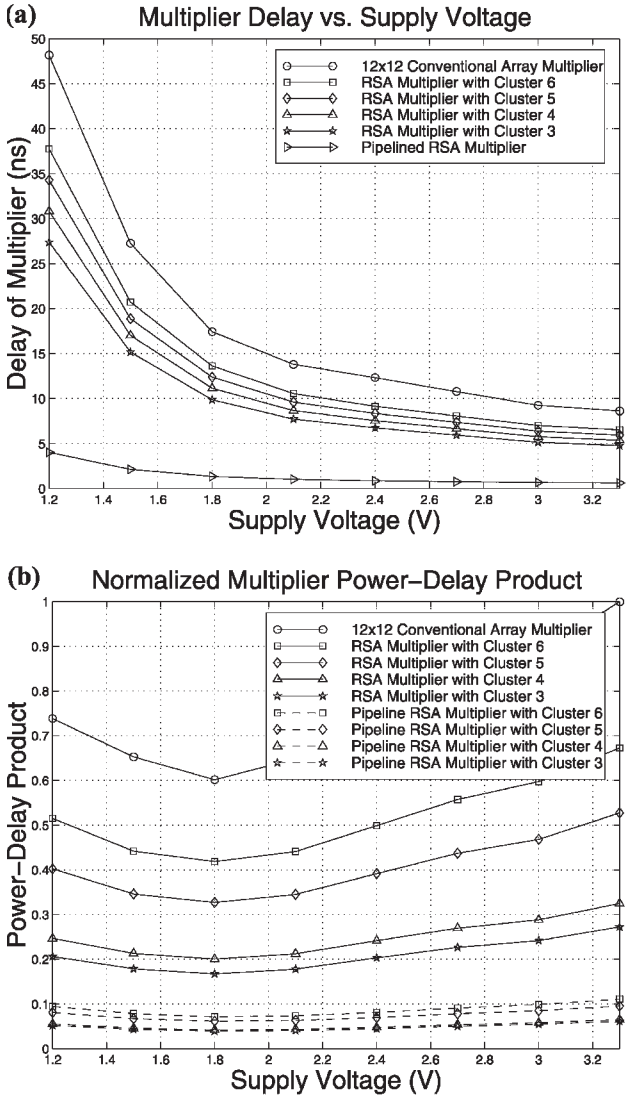

FIGURE 7 Part (a) gives the worst-case delay of the RSA multipliers as a function of the supply voltage. Part (b) gives its normalized average power-delay product as a function of the supply voltage.

delay, and error ratio with the best previous encoding of a scaled coefficients set. The minimum supply voltage for which the delay bound $T_{\max}$ is not exceeded is determined by Eqs. (8) and (9) in its main body. For the minimum supply voltage, the power constraint is checked by simulation or using the expression

$$\text{power}\,(Y) = \frac{\sum_{p=0}^{P-1} C_{p,\text{nz}}}{PN_p} C_{\text{load}} V_{\text{supply}}^2, \tag{10}$$

where $C_{p,\text{nz}}$ is the number of nonzero bits in the coefficient Y_p and N_p is the coefficient width and C_{load} is the output load capacitance. The error ratio constraint δ is checked using the expression

$$\frac{\sum_p (\tilde{Y}_p - Y_p)^2}{\sum_p \tilde{Y}_p^2} < \delta, \tag{11}$$

where $\tilde{Y}_p$ is the unquantized coefficient value. The best

COPT$(\mathcal{W}, \delta, T_{max}, C_{target})$

1. Initialize *temp* to an encoded set $\mathcal{Y}$
2. with $delay(temp) \le T_{max}$ and $error(temp) \le \delta$
3. for $(\alpha = 1.0;\ \alpha \ge 0.5;\ \alpha = \alpha - \Delta\alpha)$
4. for $(K = C_{target};\ K \le N - C_{target};\ K = K + 1)$
5. for each number representation f
6. $\mathcal{Y} = \{f(Y_0), f(Y_1), \ldots, f(Y_{P-1})\}$
7. Partition coefficients in $\mathcal{Y}$ with respect to K
8. if $power(\mathcal{Y}) < power(temp)$
9. and $delay(\mathcal{Y}) < delay(temp)$
10. and $error(\mathcal{Y}) < error(temp)$
11. then $temp \leftarrow \mathcal{Y}$
12. return $\mathcal{Y}$

FIGURE 8 Heuristic algorithm COPT for coefficient partitioning.

TABLE I Minimum error ratio δ as a function of cluster width $C_{a,\max}$ (row) and $C_{b,\max}$ (column)

	3	4	5	6
3	6.38×10^{-6}	8.87×10^{-7}	4.41×10^{-7}	4.51×10^{-8}
4	1.29×10^{-6}	1.56×10^{-7}	4.73×10^{-8}	2.05×10^{-8}
5	4.08×10^{-7}	1.08×10^{-7}	1.75×10^{-8}	1.75×10^{-8}
6	2.76×10^{-8}	1.38×10^{-8}	8.32×10^{-9}	6.37×10^{-9}

solution encountered until any given iteration is stored in the variable *temp* which can be initialized to any possible encoding of the coefficients.

Numerical Results

We have applied our algorithm to the design of a low-power multiplier for the 64-coefficient FFT. One multiplier is used to compute FFT which requires 64 coefficients. Both data and coefficient are 12-bit. Two number representations were considered: Two's complement and canonic signed digit.

Table I gives the minimum error ratio δ for various maximum cluster width pairs $C_{a,\max}$, $C_{b,\max}$. For reference, δ for 12-bit quantization is about 2×10^{-7}. Depending on the error ratio specification, the appropriate multiplier with small cluster widths is incorporated for low power applications.

Two separate comparison are performed. First of all, the effectiveness of our coefficient optimization methodology in array multipliers with and without the switching activity reduction feature is compared with a conventional array multiplier under identical quantization error bound δ. Second, our design methodology is applied to the other well-known multipliers, independent of the multiplier architecture for reducing the power dissipation.

In the simulation, 2-way partition with a cluster width (effective coefficient width) of 4 is used for the multipliers with the coefficient optimization and the coefficient width of 12 is used for the multipliers without the coefficient optimization. These two provide identical quantization error ratio bound $\delta < 2 \times 10^{-10}$. For both coefficient-optimized and unoptimized multipliers, the data width is fixed at 12. The entire FFT was performed assuming a random input data sample. These multipliers are designed and synthesized with 0.35 μm CMOS. However, no low-level circuit optimization is considered.

For all three multipliers considered in the first comparison, the worst-case critical delay of the multipliers and the normalized average power-delay product of the multipliers during the 64-coefficient FFT operation are shown in Fig. 9 as a function of the operating supply voltage ranges from 1.2 to 3.3 V. The maximum computation speed of the FFT operation is limited by the worst-case critical delay of the multipliers. In Fig. 9(a), the partitioned multipliers with the coefficient optimization are generally faster than that the conventional array multiplier since the critical delay paths of the optimized multipliers is shorter for a given operating supply voltage.

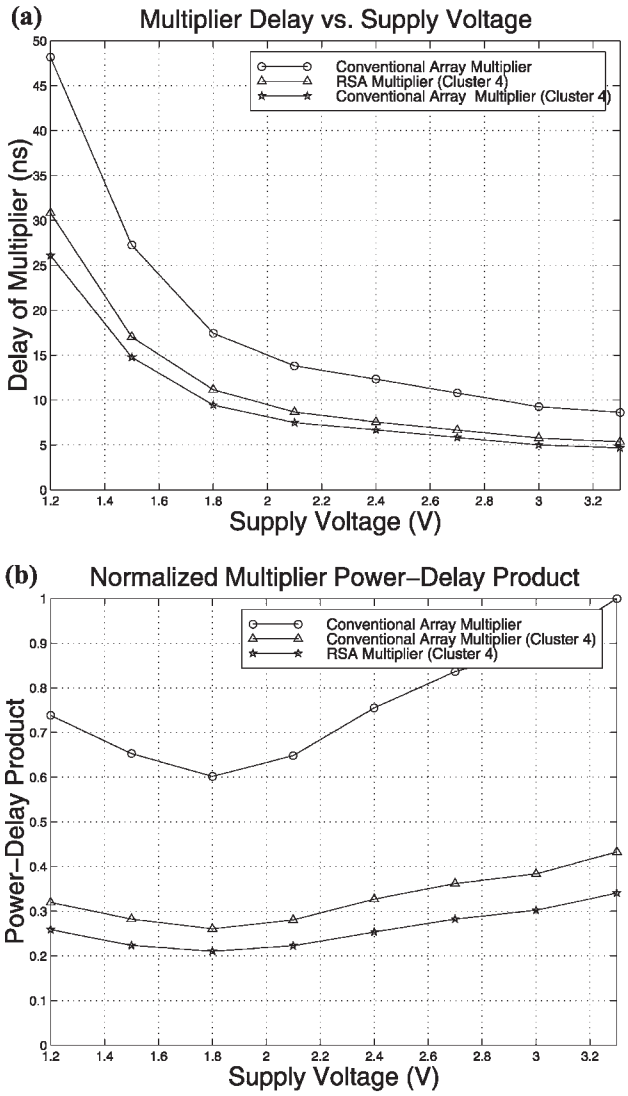

FIGURE 9 Three multipliers are considered: conventional 12 × 12 array multiplier, 2-way partitioned RSA multiplier with cluster width 4, and 2-way partitioned conventional array multiplier with cluster width 4. Part (a) gives the worst-case delay of the multipliers as a function of the supply voltage. Part (b) gives their normalized average power-delay product for 64-coefficient FFT operation as a function of the supply voltage.

However, between the coefficient optimized multipliers, the delay of the RSA multiplier is longer than that of the conventional array multiplier. This is expected since additional circuitry introduced in the RSA multiplier for row disabling contributed to the increase in the worst-case critical delay. In Fig. 9(b), the conventional array multiplier has the largest average power-delay product for a given operating supply voltage. The switching capacitance and activity make the difference in power dissipation among coefficient optimized multipliers. Even though, the optimized conventional array multiplier has smaller delay, the optimized RSA multiplier has lower average power-delay product than the optimized conventional array multipliers mainly due to the less effective switching capacitance.

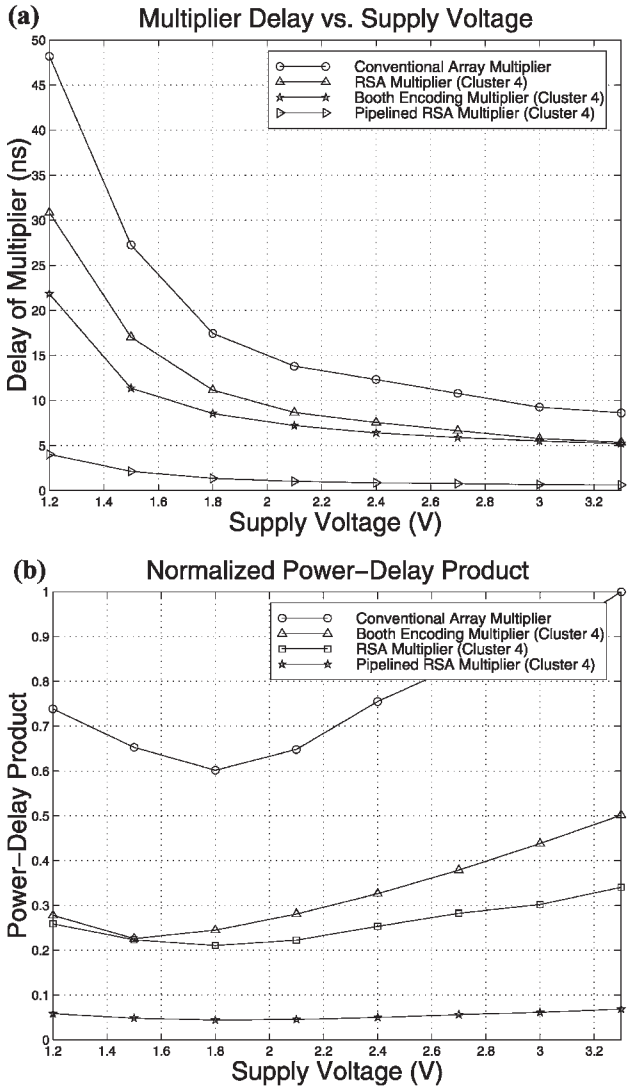

FIGURE 10 Four multipliers are considered: conventional 12 × 12 array multiplier, 2-way partitioned RSA multiplier with cluster width 4, 2-way partitioned pipelined RSA multiplier with cluster width 4, and 2-way partitioned Booth encoding multiplier with cluster width 4. Part (a) gives the worst-case delay of the multipliers as a function of the supply voltage. Part (b) gives their normalized average power-delay product for 64-coefficient FFT operation as a function of the supply voltage.

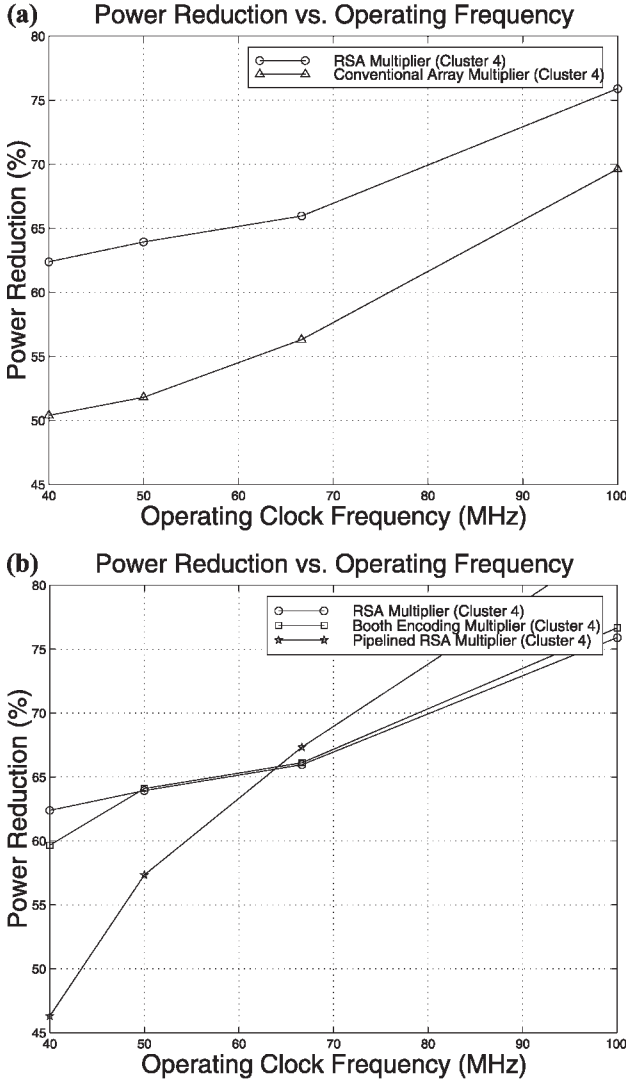

FIGURE 11 (a) Relative power savings over 12-bit conventional array multiplier as a function of throughput when using our multiplier design methodology to implement a 64-coefficient FFT computation. (b) Relative power savings over 12-bit conventional array multiplier as a function of throughput when using our multiplier design methodology on Booth encoding multiplier and pipelined RSA multiplier to implement a 64-coefficient FFT computation.

Similar analysis is performed in the second comparison. All multipliers use optimized coefficients except the baseline conventional array multiplier. In Fig. 10(a), the Booth encoding multiplier with the coefficient optimization is generally faster than the RSA multiplier. This is expected since the Booth encoding multiplier has only half the number of partial products to sum. Moreover, the RSA multiplier has many additional circuitry that contributes to the increase in the delay. The pipelined RSA multiplier is the fastest among the multipliers considered. In Fig. 10(b), the conventional multiplier has the largest average power-delay product among the multipliers and the RSA multiplier has the smallest. The pipelined RSA multiplier also consumes considerable amount of power compare to the RSA multiplier but its power-delay product is much smaller due to the speed.

Figure 11(a) shows the power dissipation savings achieved with our multiplier design methodology (for $C_{a,\max}$ and $C_{b,\max}$ set to 4) over the use of conventional array multipliers. The curves in this graph give power reduction as a function of operating clock frequency. The supply voltage of the multipliers has been scaled appropriately. In general, relative savings increase as operating clock frequency increases. Because voltage scaling dominates at high operating clock frequency resulting in a significant power reduction. As the required clock frequency decreases, most of the power savings come from the reduced switching activity of the multiplier. For example, 76 and 63% power reductions by the optimized RSA multiplier were achieved at 100 and 40 MHz, respectively. The graph also shows that the coefficient optimization alone saves a significant amount

of power. For example, 69 and 50% power reductions by the optimized conventional array multiplier were achieved at 100 and 40 MHz, respectively. The power reduction gap between the two curves are narrower at the high operating speed than low operating speed. Since the worst-case critical delay is smaller for the multiplier without row-disabling, the benefit of low switching is compensated by the amount of supply voltage scaled. However, as the required operating speed becomes slower, all multipliers operates at the lowest supply voltage which results in no supply voltage scaling and the power reduction due to the reduced switching activity dominates.

In Fig. 11(b), voltage scaling dominates at high operating frequency resulting in a significant power reduction. This is evident for the pipelined RSA multiplier since it has the lowest worst-case delay hence a significant reduction can be achieved by the voltage scaling. However, at low operating speed, the power reduction is not as significant since power reduction by the switching capacitance dominates. Even though the worst-case critical delay of the Booth encoding multiplier is smaller than the RSA multiplier so that the amount of voltage scaled is larger with the Booth encoding multiplier given a operating speed, the actual power is compensated by the switching activity difference. In summary, the effectiveness of power saving is different for different region of the operating frequency. At low operating frequency, the pipelined RSA multiplier is the worst among the multipliers considered, but it is the best as the operating clock frequency increases beyond 65 MHz.

CONCLUSION

In this paper we have presented a novel methodology for designing low-power multipliers by coefficient optimization and architectural modifications. Specifically, the coefficients are scaled and encoded to achieve minimum bit clustering and partitioning. In conjunction with a reduced switching activity (RSA) multiplier that uses selective row disabling, this coefficient optimization scheme can result in significantly more efficient DSP computations over conventional array multipliers. In simulation with the 64-coefficient FFT, our methodology reduced dissipation in half, in comparison with a conventional array implementation, without any throughput or error rate degradation. Our design methodology can be applied to every DSP computation that performs multiplications with a set of coefficients. Many well-known design approaches at the device and the circuit level as well as coefficient level such as differential encoding the coefficient can be applied to the multiplier obtained by our methodology for additional gain in the power efficiency.

Acknowledgements

This research was supported in part by a Department of Defense Research & Engineering (DDR&E) Multidisciplinary University Research Initiative on "Low Energy Electronics Design for Mobile Platforms", managed by the US Army Research Office under Grant No. DAAH04-96-1-0377, and also supported in part by a Young Investigator Award from the US Army Research Office under Grant No. DAAG55-97-0395.

References

[1] Mehendale, M., Sherlekar, S.D. and Venkatesh, G. (1995) "Coefficient optimization for low power realization of FIR filters", *Proceedings of IEEE Workshop on VLSI Signal Processing* **January**, 352–361.

[2] Mehendale, M., Venkatesh, G. and Sherlkar, S.D. (1995) "Techniques for low power realization of FIR filters", *Proceedings of IEEE Asia and South Pacific Design Automation Conference*, 447–450.

[3] Samueli, H. (1989) "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients", *IEEE Transactions on Circuits and Systems* **36**(7), 1044–1047.

[4] Lim, Y.C. and Rarker, B.R. (1983) "FIR filter design over a discrete powers-of-two coefficient space", *IEEE Transactions on Acoustics, Speech, Signal Processing* **ASSP-31**, 583–591.

[5] Chandrakasan, A.P. and Brodersen, R. (1996) Low Power Digital CMOS Design (Kluwer Academic Publishers, Dordrecht).

[6] Abu-Khater, I.S., Bellaouar, A. and Elmasry, M.I. (1996) "Circuit techniques for CMOS low-power high-performance multipliers", *IEEE Journal of Solid State Circuits* **31**(10), 1535–1546.

[7] Poornaiah, D.V., Ananda-Mohan, P.V., Nishitani, T. and Parhi, K.K. (1995) "Novel VLSI multi-bit coded multiplier and multiplier–accumulator architectures for DSP applications", *VLSI Signal Processing* **VIII**, 542–551.

[8] Wu, A., Ng, C.K. and Tang, K.C. (1998) "Modified Booth pipelined multiplication", *Electronics—Letters* **34**(12), 1179–1180.

[9] Dawoud, D.S. (1997) "Modified Booth algorithm for higher radix fixed-point multiplication", *Proceedings of IEEE South African Symposium on Communications and Signal Processing*, 95–100.

[10] Rao, V.M. and Nowrouzian, B. (1997) "A novel approach to the design and hardware implementation of high-speed digit-serial modified-booth digital multipliers", *Proceedings of IEEE International Symposium on Circuits and Systems*, 1952–1955.

[11] Goto, G., Inoue, A., Ohe, R., Kashiwakura, S., Mitarai, S., Tsuru, T. and Izawa, T. (1997) "A 4.1-ns compact 54*54-b multiplier utilizing sign-select booth encoders", *IEEE Journal of Solid State Circuits* **32**(11), 1676–1682.

[12] Chandrakashan, A.P. and Brodersen, R. (1995) "Minimizing power consumption in digital CMOS circuits", *Proceedings of the IEEE* **83**(4), 498–523.

[13] Chandrakasan, A., Potkonjak, M., Mehra, R., Rabaey, J. and Brodersen, R. (1992) "Minimizing power using transformations", *Proceedings of IEEE International Conference on Computer-Aided Design*, 300–303.

[14] Jou, S.J., Chen, C.Y., Yang, E.C. and Su, C.C. (1997) "A pipelined multiplier–accumulator using a high-speed, low-power static and dynamic full adder design", *IEEE Journal of Solid State Circuits* **32**(1), 114–118.

[15] Hong, S., Kim, S., Papaefthymiou, M.C., Stark, W.E. "Power-complexity analysis of pipeline VLSI FFT architecture for low energy wireless communication applications". *Proceeding of IEEE Midwest Symposium on Circuits and Systems*, Las Cruse, New Mexico, 1999.

[16] Sankarayya, N., Roy, K. and Bhattacharya, D. (1997) "Algorithms for low power FIR filter realization using differential coefficients", *International Conference on VLSI Design* **June**, 174–178.

Authors' Biographies

Sangjin Hong received the BS and MS degrees in EECS from the University of California at Berkeley and the PhD in EECS from the University of Michigan at Ann Arbor. He is currently an assistant professor in the department of

Electrical and Computer Engineering at the State University of New York at Stony Brook. Before joining the department, he was a research fellow at the University of Michigan at Ann Arbor. He also worked at Ford Aerospace and Communications Corporation and at Samsung Electronics Corporation in Korea as a technical Consultant. His current research interests are in the areas of low power multimedia DSP and wireless communication system design, power/performance evaluation, and design methodology.

Suhwan Kim received the BS and MS degrees in Electrical Engineering and Computer Science from Korea University, Korea, in 1990 and 1992, respectively. He received the PhD degree in electrical engineering from the University of Michigan, Ann Arbor, in 2001. From 1993 to 1997, he was with LG Electronics, Korea, where he designed several multimedia systems-on-chip (SOCs) including an MPEG2 CODEC for audio, video, and system. Currently, he is a member of Research Staff with IBM T.J. Watson Research Center, Yorktown Heights, NY. His research interests encompass high-performance and low-power circuits and technology, low-energy design methodologies for high-performance VLSI signal processing, and CAD tools for VLSI. Mr Kim has received the 1994 IEEE Korea section's Best Student Paper Award.

Wayne E. Stark received the BS (with highest honors), MS, and PhD degrees in electrical engineering from the University of Illinois, Urbana in 1978, 1979, and 1982, respectively. Since September 1982 he has been a faculty member in the Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor where he is currently Professor. From 1984 to 1989 he was Editor for Communication Theory of the IEEE Transactions on Communication. He was selected by the National Science Foundation as a 1985 Presidential Young Investigator. His research interests are in the areas of coding and communication theory, especially for spread-spectrum and wireless communication networks. Dr Stark is a member of Eta Kappa Nu, Phi Kappa Phi and Tau Beta Pi and a Fellow of the IEEE.