Journal of VLSI Signal Processing 38, 101–113, 2004

Power Reduction Technique in Coefficient Multiplications Through Multiplier Characterization

SANGJIN HONG AND SHU-SHIN CHIN
Department of Electrical and Computer Engineering, Stony Brook University-SUNY, Stony Brook, NY 11794-2350, USA

SUHWAN KIM AND WEI HWANG
Department of Low Power Circuit Technology, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA

Received July 29, 2002; Revised April 11, 2003; Accepted May 21, 2003

Abstract. This paper presents a multiplier power reduction technique for low-power DSP applications through utilization of coefficient optimization. The optimization is implementation dependent in that the multipliers are assumed to be designed in either ASIC or full-custom architectures for general purpose multiplication. The paper first describes a model characterizing the power consumption of the multiplier. Then the coefficient optimized made based on this model. This methodology is applicable to multiplications requiring a large set of coefficients and random data sets. We can accurately estimate the actual power dissipation of the multipliers using the characterization technique. The coefficient optimization based on the power model can save as much as 34.02%.

Keywords: low-power multiplier, coefficient optimization, power modeling, power weight factor

1. Introduction

Low-power system design has become an important issue as more functional blocks are being integrated onto a single chip. Many digital signal processing systems such as wireless communications, signal processing, and image processing systems use extensive multiply and accumulate operations. In such applications, multiplications comprise a significant portion of the overall operations and tend to be the most power dissipative. Thus, power-efficient use of multipliers is therefore essential for the design of low-power DSP hardware.

Many power reduction strategies have been proposed for low-power multiplier design including reduction of supply voltage and clock speed, use of signed-magnitude arithmetic and differential data encoding, parallelization or pipelining of operations, and tuning of input bit-patterns to reduce switching activity [1–4]. In the context of DSP with a given set of coefficients, substantial research has been devoted to the topic of manipulating the coefficients to reduce power dissipation. Earlier work in this area focused on techniques for transforming the coefficients' binary representation to minimize the computations in the context of application specific implementation. However, in the current trends for most system design, such as embedded multipliers in FPGAs or processors, the multipliers are predesigned and/or given as blocks to be integrated into the systems.

This paper presents a power reduction technique, which is applicable to many system-on-chip (SOC) designs regardless of whether the multipliers are custom designed or predesigned. The main idea is to characterize the multiplier's power consumption and optimize the coefficient that is suitable for the specific

implementation so that the overall power dissipation can be minimized. The coefficient optimization is based on a novel power dissipation characterization technique where actual power consumption can be accurately estimated. We can accurately estimate the actual power dissipation of the multipliers using the characterization technique. The coefficient optimization based on the model can save as much as 34.02%. The paper assumes two pieces of information: first, that the multiplier is designed and given to the system designers, and second, that the input bit pattern is random such that the data bit patterns are uncorrelated.

The remainder of this paper has 5 sections. Section 2 discusses the source of power consumption of multipliers. In Section 3 we describe the power consumption characterization and estimation of multipliers. Section 4 describes our methodology for designing low-power multipliers based on the power consumption characterization. In Section 5 we present results from the application of our methodology for multiplication power savings on 64-point FFT. Our contributions are summarized in Section 6.

2. Source of Power Consumption

The array multiplier computes the partial products in parallel and then shifting and accumulating them. The multiplier width corresponds to the word length of the data samples and the multiplier depth corresponds to the word length of the coefficients. In the multiplier, power consumption depends on the switching of individual transistors. It can be seen from the multiplier structure that there is power consumed by switching of transistors within the multiplier itself. In fact the power consumption of the multiplier is mainly due to internal switching activity caused by the propagated data switching.

As an example, if the kth bit of a coefficient is 0, the kth row of adders does not need to be activated and the partial product of the previous adder rows need to be shifted and bypassed to the next row of adders. In this case, the function of the kth row of adders is simply a one-bit shift of the partial products. However, the adders of the array multiplier corresponding to zero coefficients are still switching even though an adding operation is not required since the partial product value needs to be propagated. As shown in Fig. 1, the shaded part of the multiplier is the possible switching area caused by the activated coefficient bit.

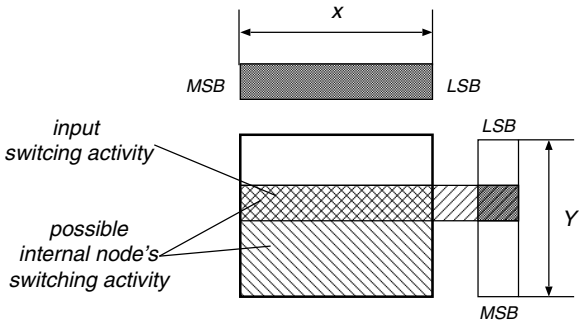


Figure 1. Active region of $X \times Y$ array multiplier that's generated by the input and internal switching activity.

3. Power Consumption Characterization

3.1. Power Weight Factor of Multipliers

We define power weight factor as a metric for estimating power consumption and we use it to evaluate power consumption of multipliers. The power weight factor considers all the switching power upon change in inputs. Thus, lower power factor in multiplier design is highly desirable. The power weight factors are plotted in Fig. 2 for the carry-save array multiplier, in Fig. 3 for the Booth-recoded multiplier, and in Fig. 4 for the Wallace-tree multiplier. The relative power weight factor PW_i is defined as the power consumed by the multiplier when coefficient bit i is set to "1" and all other bits are set to "0". The relative power weight factor incorporates power consumed by the multiplier due to the data. These plots are obtained for random input data patterns. The vertical axis is the relative average power weight factor and the horizontal axis is the position of the most significant active bit of the coefficient. In the case for the Booth-recoded multiplier, encoded bits are used on the horizontal axis. These power weight factors incorporate the internal switching of the multipliers.

As shown in the figure, the power weight factors of a carry-save multiplier strongly depend on the position of the bit while the power weight factor of the Wallace-tree multiplier is constant. This shows that induced switching is presented in array-type multiplier architectures. On the contrary, the power weight factor for a Wallace-tree multiplier is greater when the coefficient bit position is located in the middle. In the figure illustrating the Booth-recoded multiplier, we have four separate curves for different possible encoding bits.

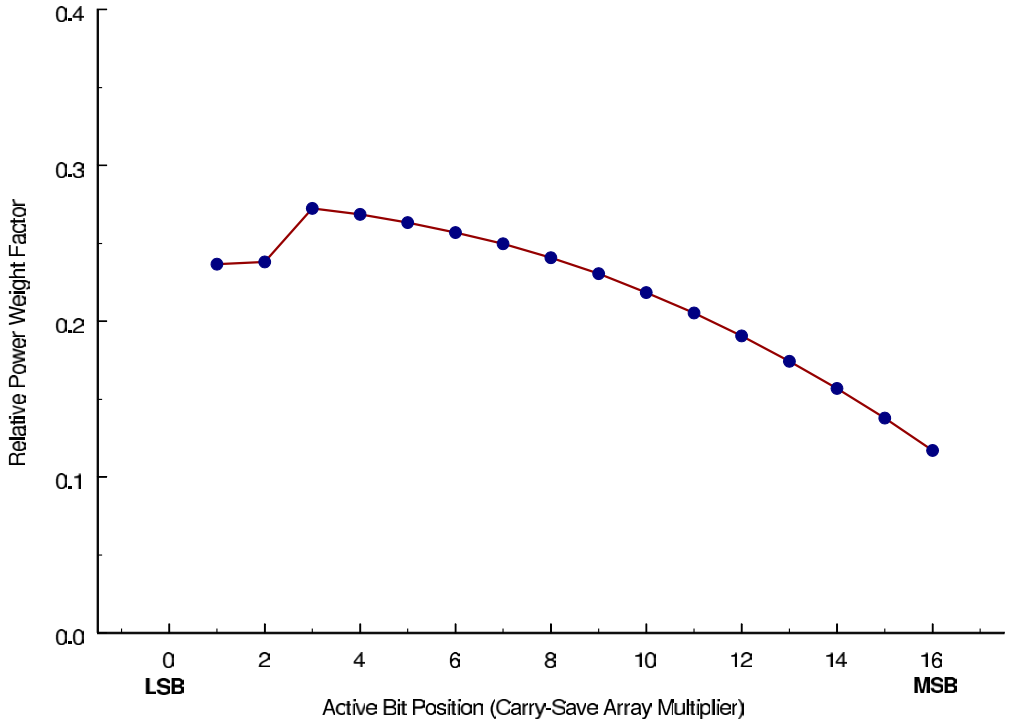


Figure 2. Relative power weight factor of a 16 × 16 carry-save array multiplier.

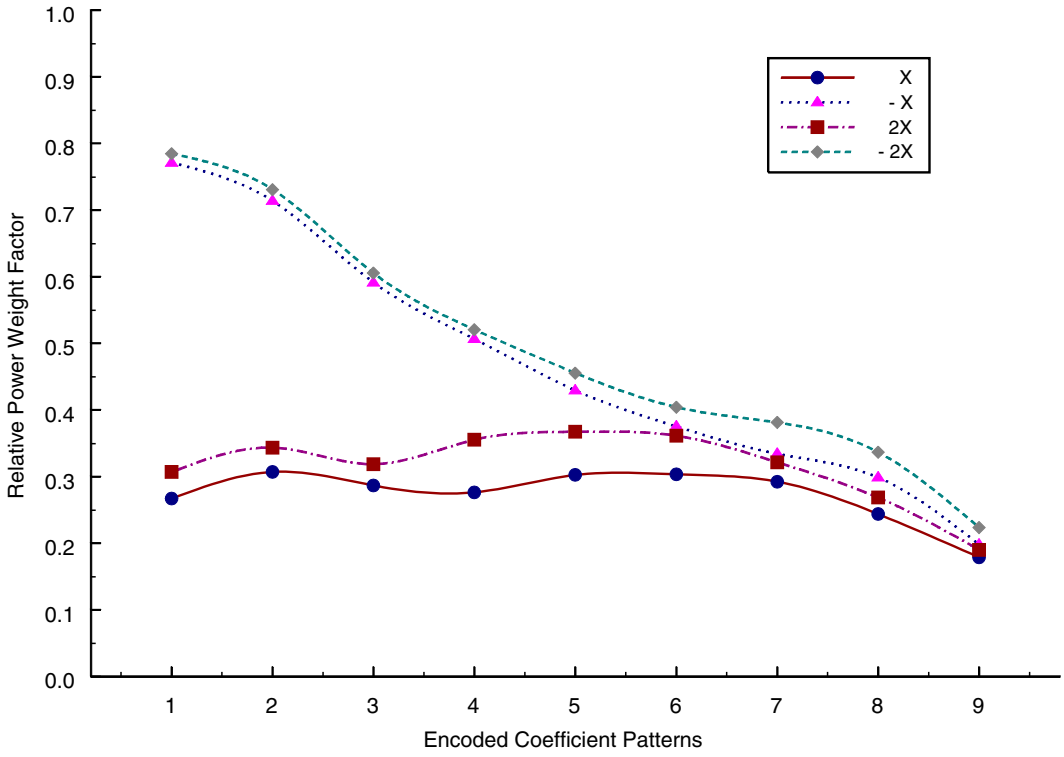


Figure 3. Relative power weight factor of a 16 × 16 Booth-recoded multiplier.

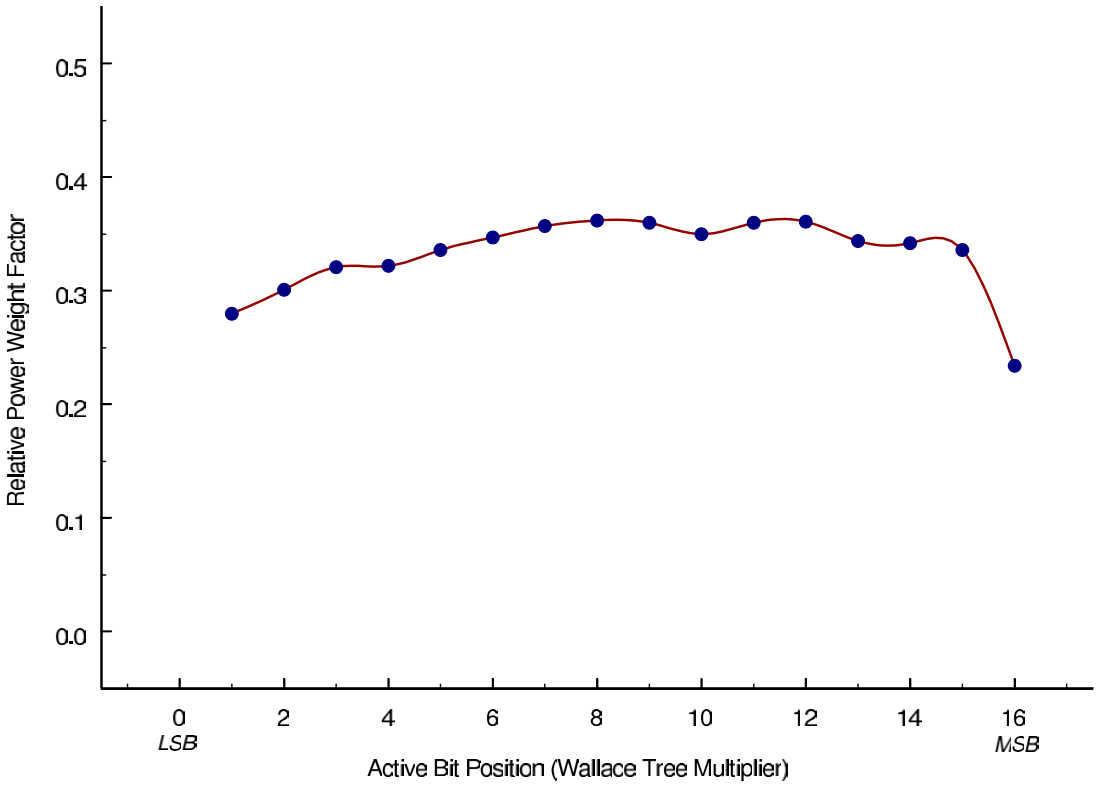


Figure 4. Relative power weight factor of a 16 × 16 Wallace-tree multiplier.

3.2. *Estimation vs. Actual Power*

In the previous section, we defined the power weight factor for a coefficient bit i as switching power dissipated by the multiplier caused by the i-th adder row and below. Since we assumed that input bit patterns are random such that the power weight factor represents the average power dissipation.

When more than one coefficient bit are active, for example bit i and bit j, the power dissipation due to these two coefficient bits can be viewed as a sum of individual power weight factor for i and j. For illustration, consider a array multiplier and let bit i is more significant than bit j (i.e., $2^2 > 2^j$). In this situation, power dissipation due to switching of i-th row and below corresponds to the power weight factor of i. At the same time, power dissipation due to switching of j-th row and below, including i-th row, corresponds to the power weight factor of j. Thus, the switching due to these two bits is independent.

However, this sum does not include possible switching due to glitching of signals. We have obtained empirically that the effect of glitching power is proportional to the sum of power weight factors and it depends on the structure.

Then, the power consumption of the carry-save array multiplier considered in this paper can be represented as

$$P_{\text{array}} = K \sum_{i=0}^{N-1} PW_i \tag{1}$$

where PW_i is the value of the relative power weight factor of the multiplier due to the active coefficient bit position i, and N is the width of the coefficient. Thus, the value of PW_i and K are multiplier architecture and implementation dependent and can be easily obtained from simulations. P_{array} is the sum of power weight factors PW_i multiplied by a constant scaling factor K obtained from a SPICE simulation using a 0.35 μm CMOS technology. Random data bit patterns are used for the simulation and this assumption is valid since the inputs to the multipliers are often random signals such as speech. From the empirical study that we have conducted, the K factor can be interpreted as glitching power dissipation that has not been included in the

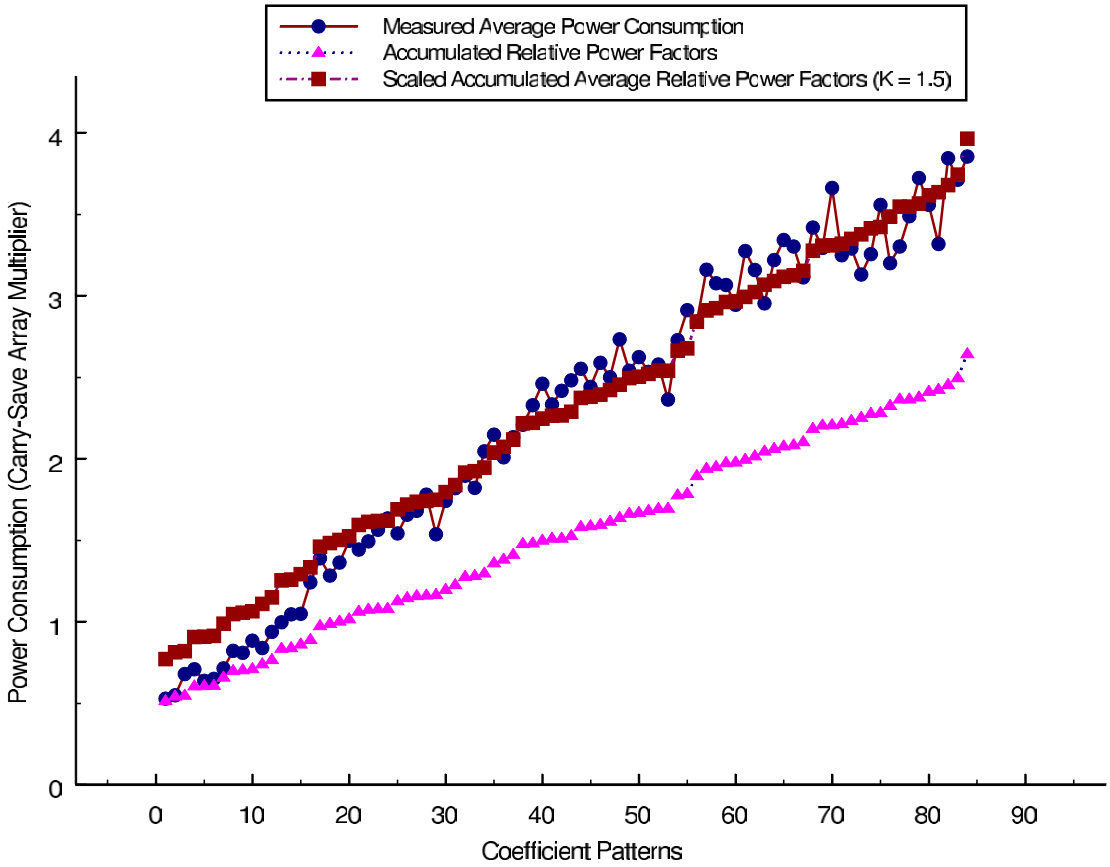


Figure 5. Ordered power consumption model derived from power weight factor of carry-save array multiplier. The model is compared with the actual power consumption.

power weight factor estimation. As we will show on Booth encoded and Wallace tree multipliers, K is influenced by the depth of multipliers.

Figure 5 illustrates the actual power consumption obtained from the simulation and from the sum of the relative power weight factors. 84 sets of randomly selected coefficient bit patterns were studied and the set was formed according to the number of "1" 's in the coefficient bit pattern. This graph plots all coefficients considered in the study on one line according to their power consumption. There are 3 lines: measured or actual power consumption, sum of the relative power factors, and the scaled sum of the power weight factors. As shown in this figure, the actual power consumption of a given coefficient very closely agrees with the scaled sum of the relative power weight factors. Hence, the model is valid. For the carry-save multiplier designed in this paper, a value for K of 1.5 was chosen which is an implementation dependent parameter.

Similarly, the power consumption of the booth-recoded multiplier considered in this paper can be represented as

$$P_{\text{coded}} = K \sum_{i=0}^{N'-1} PW_i \tag{2}$$

where PW_i is the value of relative power weight factor of the multiplier due to active encoded bit position i and N' is the number of encoded bit width of the coefficient. P_{coded} is the sum of power weight factors PW_i multiplied by a constant scaling factor K. Random data bit patterns are used in the simulation. Thus, the value of PW_i and K are multiplier implementation dependent and can be easily obtained from the simulation. Figure 6 illustrates the actual power consumption obtained from the simulation and from the sum of the relative power weight factors. As shown in the figure, the actual power consumption of a given coefficient

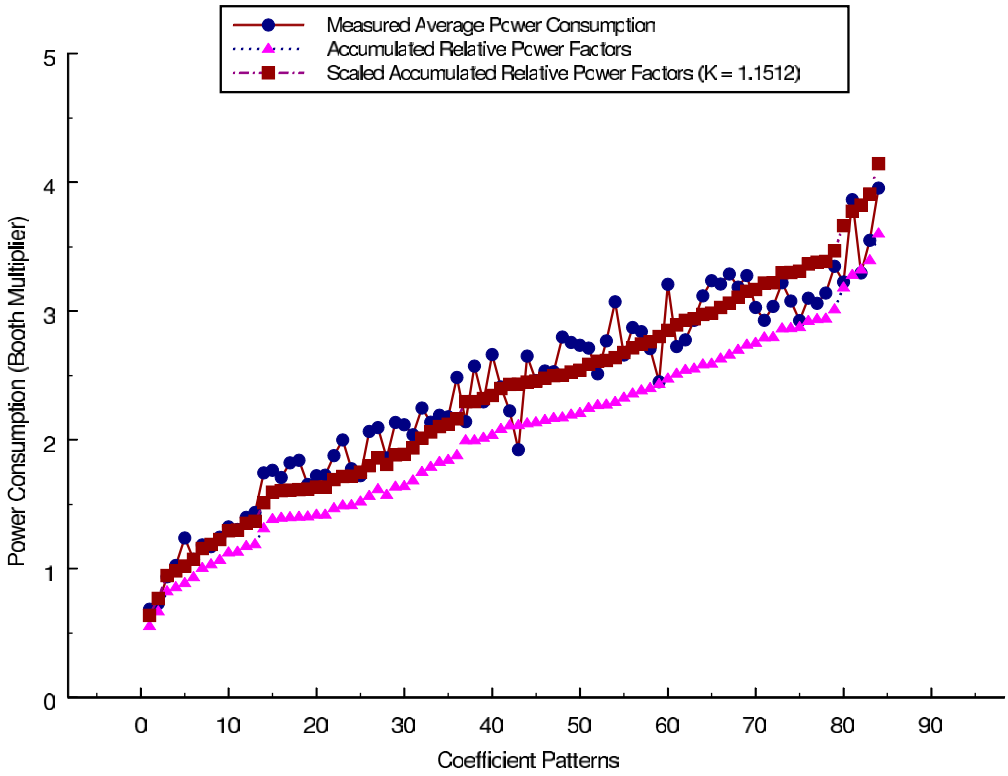


Figure 6. Ordered power consumption model derived from power weight factor of Booth recoded multiplier. The model is compared with the actual power consumption.

given by the sum of the relative power weight factors closely follows the measured power consumption. The fluctuation of the measured power is due to the power dissipated in the encode logic. The measured power consumption very closely agrees with the scaled sum of the relative power weight factors. For the Booth recoded multiplier designed in this paper, the value of K of 1.15 was chosen, which is an implementation dependent parameter. Because of its less number of multiplier depth, the Booth encoded multiplier has smaller K implying that it suffers less from the glitching power dissipation.

The power consumption model of the Wallace tree multiplier can also be represented as

$$P_{\text{tree}} = K \sum_{i=0}^{N-1} PW_i. \tag{3}$$

Similarly, 84 different randomly selected coefficient bit patterns were studied. As shown in the figure, the actual power consumption of a given coefficient can be closely predicted using the relationship defined above. Similar to the results for the carry-save multiplier, Fig. 7 uses these results and plots them in according to their power consumption. Again there are 3 lines: measured power consumption, sum of the relative power factors, and scaled sum of the power weight factors. The measured power consumption very closely agrees with the scaled sum of the relative power weight factors. For the Wallace-tree multiplier designed in this paper, the value of K of 0.95 was chosen to fit actual power consumption of the multiplier, which is an implementation dependent parameter. The values of K for all of carry-save, Booth recoded, and Wallace tree multipliers are needed to fit the power consumption model to the measured power consumption. However, the actual value of K is not important for the purpose of selecting optimum coefficients since we are more concerned with the relative power among the sets of coefficients.

3.3. *Effect of Temporal Behavior*

It is readily observable that, for a coefficient set with similar accumulated power weights, minimizing the switching pattern of the coefficients can result in a

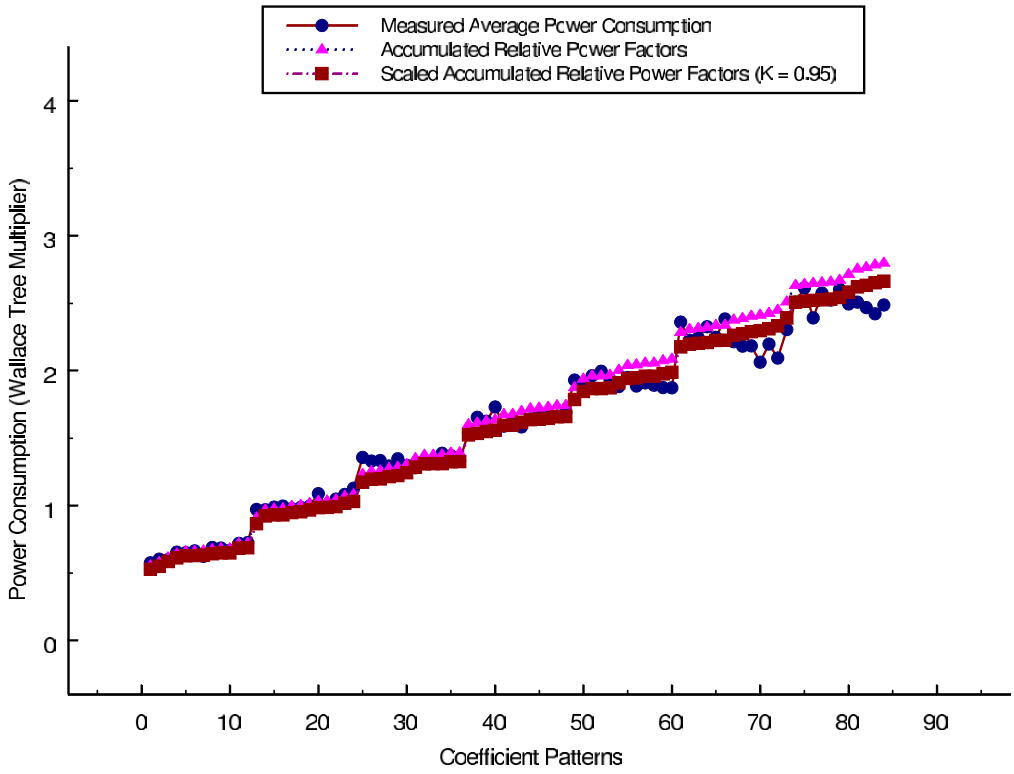


Figure 7. Ordered power consumption model derived from power weight factor of Wallace tree multiplier. The model is compared with the actual power consumption.

reduction of power dissipation. For example, if the accumulated power weights are kept unchanged, changing from 1000000010101 to 0100001000101 results in less power consumption than 1000000010101 to 0010010001010 because the first case 4 rows of adders in the multiplier are triggered by the partial product switching whereas the second case 8 rows of adders are triggered. Figure 8 shows the comparison of averaged power consumption for different numbers of switching partial products in a carry-save array multiplier. Twenty random data and five coefficient sets with the same accumulated power weights were selected. It can be seen that the multiplier with more rows of adders being triggered consumes more power in each case.

When temporal effects are included in the power estimation, power weights of a bit that changes from 0 to 1 are added (See Fig. 9). However, a bit change from 1 to 0 has very little effect on the overall power consumption. For illustration, consider two coefficient sequences that have identical overall power weight factors.

As shown in Fig. 10, the power weight factors of $0 \rightarrow 1$, $1 \rightarrow 1$, and $1 \rightarrow 0$ are very similar. This indicates that the power consumption of the multiplier is dominated by internal switching.

Although the aggregate power weight factors of two sequences are identical when temporal effects are not considered, the actual power consumption is different because of their ordering. After the incorporation of the temporal effects on power estimation, their estimated power is expressed as:

$$P_{\text{case1}} = P_{\text{spatial}} + PW_0 + PW_2 + PW_4 \quad (4)$$

$$P_{\text{case2}} = P_{\text{spatial}} + PW_0 + PW_1 + PW_2 + PW_4 \quad (5)$$

where P_{spatial} is the same for both cases and PW_i represents the power weight factors of the ith bit of the coefficient due to temporal switching of coefficient bits. These additions are illustrated in bold face.

4. Coefficient Optimization Method

We have described in the previous sections a model derived from the corresponding relative power weight

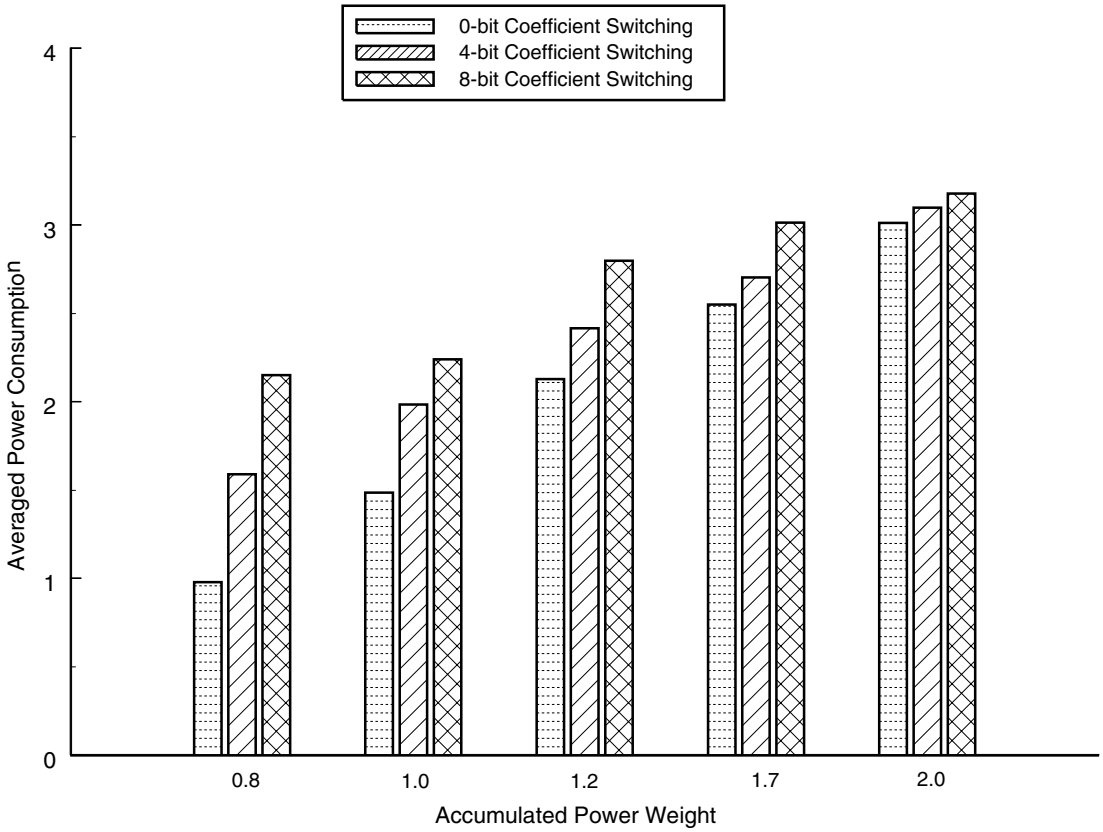


Figure 8. Temporal effect of different accumulated power weights on a carry-save array multiplier.

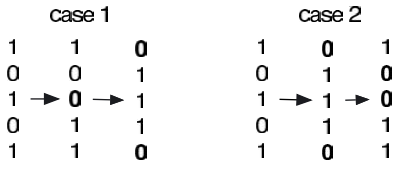


Figure 9. Illustration of temporal effects of coefficients.

factors that accurately represents the multiplier power consumption. In this section, we describe a method for coefficient optimization that minimizes power consumption considering the type of multiplier and coefficient bit patterns. This optimization modifies the pattern of the coefficients such that the switching activities of the adders are minimized. We assume that the input data has random bit patterns. This assumption is usually valid in many signal processing and communications applications and our focus is on optimizing the set of coefficients.

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

$$\hat{X}[k] = \frac{1}{\alpha} \sum_{p=0}^{P-1} \alpha(X[p] + e_d[p])\big(W_P^{kp} + e_w[p]\big), \quad (6)$$

where W_P^{kp} is the pth coefficient, $X[p]$ is the input data sample, $e_w[p]$ is the error due to the 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}, $p = 0, 1, \ldots, P-1$, and an error-ratio bound δ, 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 the coefficient quantization error ratio is less than δ, and the multiplier's total power weight factor is minimal.

The pseudo-code of our coefficient optimization algorithm is given in Fig. 11. This algorithm comprises two nested loops. The outer loop steps through the

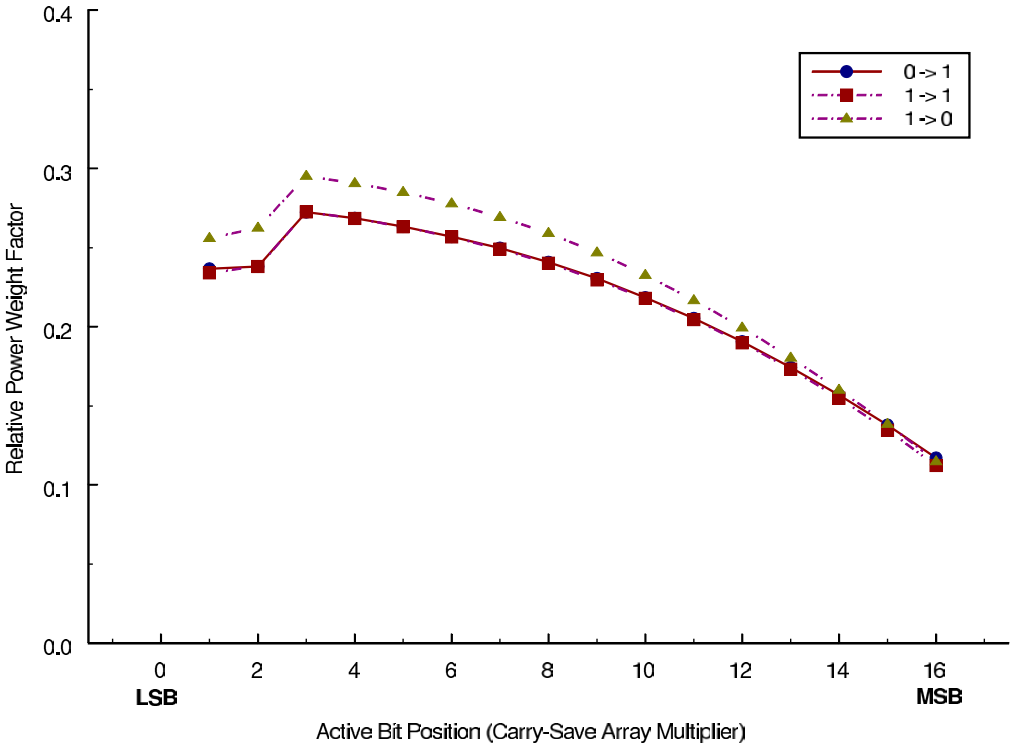


Figure 10. Effect of coefficient bit switching.

COPT(δ, P_{max})
1. Initialize $temp$ to an encoded set $\mathcal{Y}$
2. with $power(temp) \leq P_{max}$ and $error(temp) \leq \delta$
3. for ($\alpha = 1.0$; $\alpha \geq 0.5$; $\alpha = \alpha - \Delta\alpha$)
4. for each set of coefficients f
5. accumulate relative power weight factors PW
6. if $power(\mathcal{Y}) < power(temp)$
7. and $error(\mathcal{Y}) < error(temp)$
8. then $temp \leftarrow \mathcal{Y}$
9. return $\mathcal{Y}$

Figure 11. Algorithm COPT for optimal coefficient selection.

possible scaling factors α while the inner loop steps through the possible number of representations f. It subsequently compares its total power weight factor and error ratio with the best previous encoding of a scaled coefficient set. The error ratio constraint δ is checked using the expression

$$\frac{\sum_p (\tilde{Y}_p - Y_p)^2}{\sum_p \tilde{Y}_p^2} < \delta, \tag{7}$$

where $\tilde{Y}_p$ is the un-quantized coefficient value. The best solution encountered in any given iteration is stored in the variable *temp* which can be initialized to any possible encoding of the coefficients.

Our optimization algorithm performs coefficient perturbation and coefficient encoding. During coefficient perturbation, each coefficient Y_p is scaled by a factor α to change the bit patterns of the coefficient encoding. The twiddle-factor coefficients are initially scaled so that the largest coefficient has a value of unity. Throughout the perturbation process, the value of α is changed by the small amount $\Delta\alpha < 2^{-16}$. Since there is no systematic way to find the optimum α, the entire possible coefficient sets are searched. This process is usually done one at the design level. The shape of the FFT's frequency response remains unaffected when all twiddle-factor coefficients are multiplied by some constant α. This scaling simply contributes an additional gain or attenuation to the frequency response.

5. Case Study: FFT

Even though the technique can be applied to DSP algorithms such as FIR and DCT that include coefficient

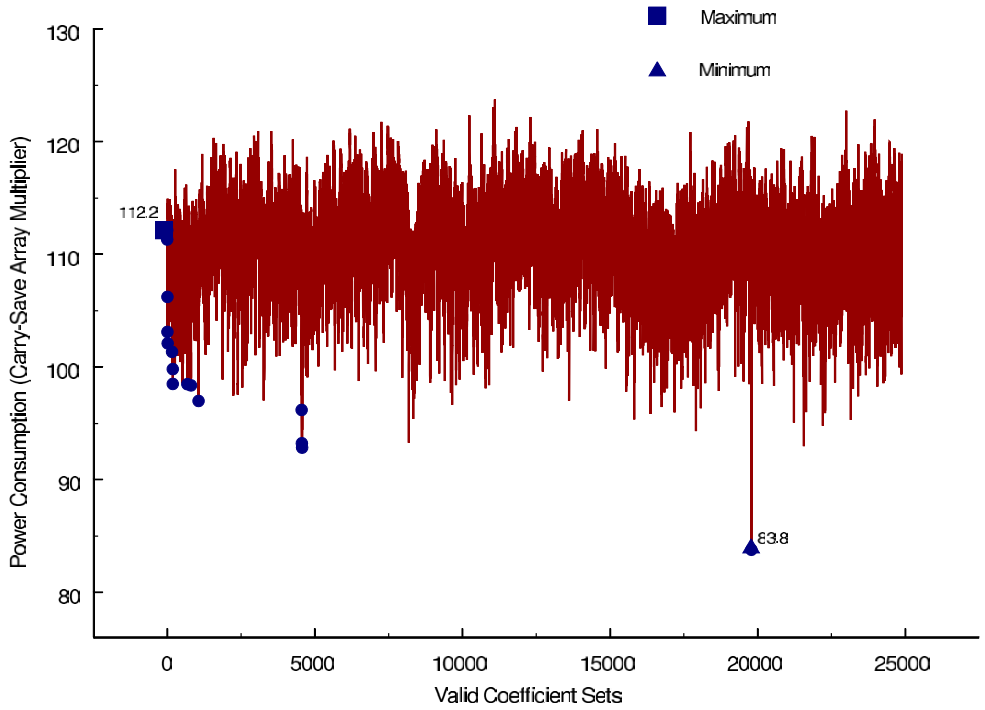


Figure 12. Power consumption profile of the Carry-Save multiplier. Maximum and minimum power dissipations are indicated.

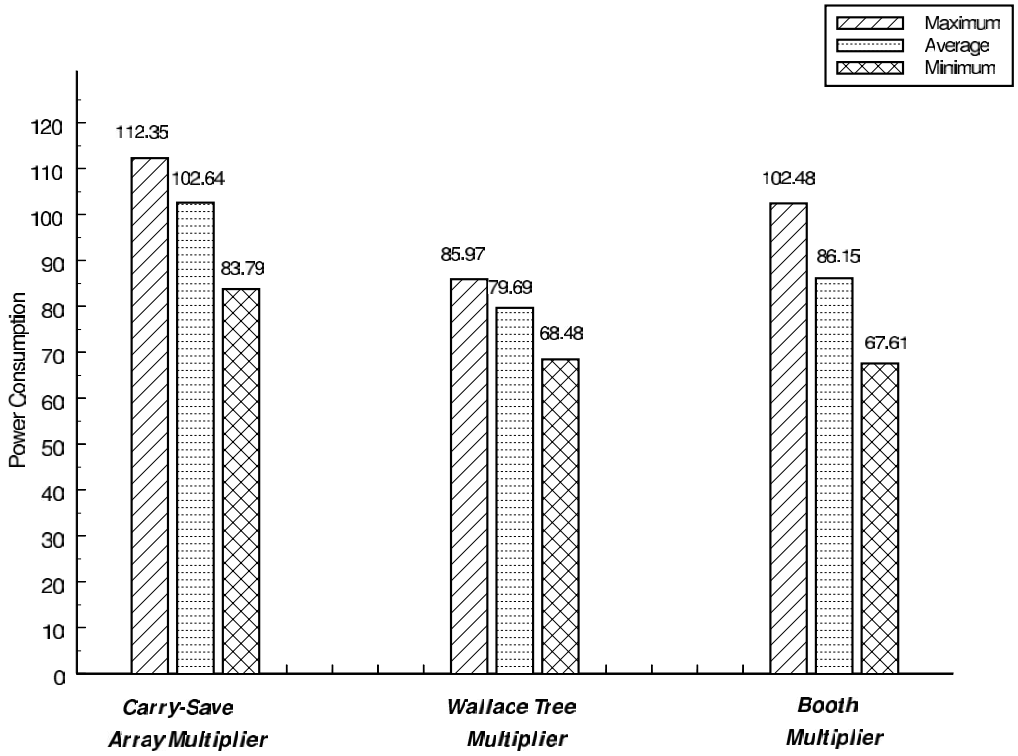


Figure 13. Power dissipation profile of the multipliers. The simulated power consumption for valid sets of coefficients that satisfy required error ratio constraint is plotted.

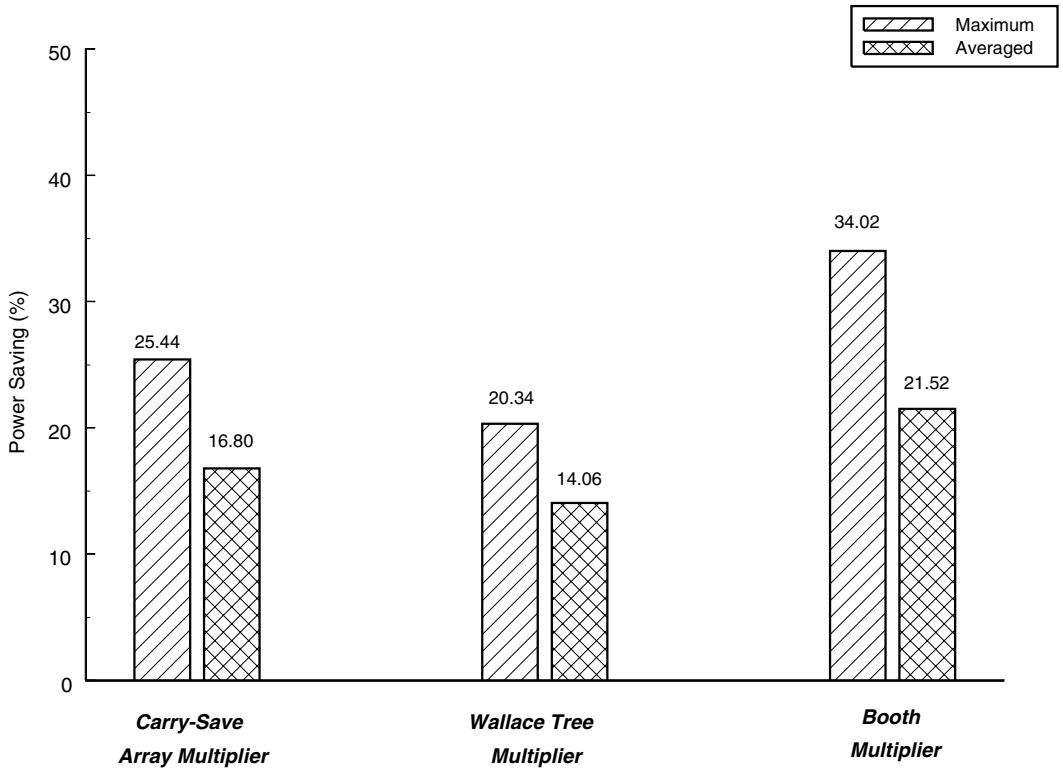


Figure 14. Power saving profile of the multipliers.

multiplications, we present analysis on the power dissipation of an FFT that has 64 distinct coefficients. We have applied our algorithm to implement coefficient multiplication for the FFT. Two's complement number representation is used for the data. The target error ratio is made to be smaller or equal to 16-bit precision. The error ratio is obtained for the minimum power dissipating coefficient set for the multiplier.

Figure 12 shows the power dissipation profiles for different sets of coefficients for the Carry-Save multiplier. We assumed constant supply voltage and throughput requirement. The power consumption ranges from 83.8 to 112.2 where, as much as 25.44% and 16.80% of reduction in power consumption can be demonstrated from the worst case and the average case, respectively.

Figure 13 shows the power dissipation profiles of different sets of coefficients for three different multipliers. Fig. 14 shows the power saving profiles which illustrated the amount of power saving from the average case and the worst case power dissipations. For the Wallace tree multiplier, power consumption ranges from 85.97 to 68.48 corresponding to 20.34% and 14.06% of reduction in power consumption from the worst case and the average case, respectively. For the Booth recoded multiplier, power consumption ranges from 102.48 to 67.61 corresponding to 34.02% and 21.52% of reduction in power consumption from the worst case and the average case, respectively. These plots illustrate power consumption variation among sets of coefficients that satisfy the minimum required error ratio bound of 16 bit precision. From these plots, we can say that the power consumption of the multiplier strongly depends on the coefficient bit patterns and the patterns that dissipate a least amount of power can be determined using the power weight factor and the power consumption models.

6. Conclusions

In this paper we have presented a novel power reduction methodology by characterizing power consumption of multiplier architectures. The paper first describes a model characterizing the power consumption of the multiplier and then the coefficient optimization is made based on this model. This methodology is applicable to multiplications requiring a large set of coefficients and random data sets. The method finds an optimum set of coefficients for given multiplier design, which is

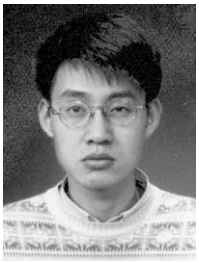
very critical in reducing power consumption hardware in computationally intensive applications. The power weight factor presented in this paper can also be used in general purpose processor programming for DSP applications.

References

1. A.P. Chandrakashan and R. Brodersen, *Low Power Digital CMOS Design*, Kluwer Academic Publisher, 1996.
2. N. Sankarayya, K. Roy, and D. Bhattacharya, "Algorithms for Low Power FIR Filter Realization Using Differential Coefficients," in *International Conference on VLSI Design*, 1997, pp. 174–178.
3. M. Mehendale, S.D. Sherlekar, and G. Venkatesh, "Coefficient Optimization for Low Power Realization of FIR Filters," *IEEE Workshop on VLSI Signal Processing*, Japan, 1995.
4. H. Samueli, "An Improved Search Algorithm for the Design of Multiplierless FIR Filters with Powers-of-Two Coefficients," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 7, 1989, pp. 1044–1047.

Sangjin Hong received the B.S and M.S degrees in EECS from the University of California, Berkeley in 1985 and 1992 respectively. He received his Ph.D in EECS from the University of Michigan, Ann Arbor. He is currently with the department of Electrical and Computer Engineering at State University of New York, Stony Brook. Before joining SUNY, he has worked at Ford Aerospace Corp. Computer Systems Division as a systems engineer. He also worked at Samsung Electronics in Korea as a technical consultant. His current research interests are in the areas of low power reconfigurable SoC design and optimization for DSP and wireless communication systems.

Shu-Shin Chin was born in Kaohsiung, Taiwan, ROC, in 1974. He received his M.S. degree in electrical and computer engineering from SUNY at Stony Brook in 1999. He joined the Mobile Systems Design Laboratory, SUNY at Stony Brook in 1999 as a Research Assistant where he is pursuing his Ph.D. degree. His research interests include low-power coarse-grained reconfigurable architectures for high-performance particle filter designs.



Suhwan Kim received the B.S. and M.S. degrees in Electrical Engineering and Computer Science from Korea University, Korea, in 1990 and 1992, respectively and the Ph.D. degree in Electrical Engineering and Computer Science from the University of Michigan, Ann Arbor, in 2001. From 1993 to 1997, Dr. Kim was with LG Electronics, Seoul Korea, where he designed several multimedia systems-on-a-chip (SOC), including an MPEG2 CODEC for audio, video, and system. Dr. Kim joined IBM Thomas J. Watson Research Center in 2001, where he currently is a Research Staff Member. His research interests encompass circuits and technology for low-power and high-performance SOC and low-power design methodologies for high-performance VLSI signal processing. Dr. Kim has received the 1991 Best Student Paper Award of the IEEE Korea Section and the First Prize in the VLSI Design Contest of the 2001 ACM/IEEE Design Automation Conference. He has participated several times in the Organizing Committee and Technical Program Committee of the IEEE International ASIC/SOC Conference and in the Technical Committee of the International Symposium on Low Power Electronics and Designs.

Wei Hwang is the Director of Microelectronic and Information Systems Research Center, Director of SoC Research Center, tsmc Chair Professor of Electronics Engineering Department of National Chiao-Tung University in Hsinchu, Taiwan since August 2002. Prior to that, he was a Research Staff Member at the IBM Thomas J. Watson Research Center, Yorktown Heights, NY from 1984 to 2002, where he has contributed to several areas of microelectronics, DRAM, microprocessor and merged logic memory design. He also served as an Adjunct Professor of Electrical Engineering at Columbia University in New York, NY from 1993 to 2001. He was Associate Professor of Electrical Engineering Department at Columbia University in

New York from 1979 to 1984. He was Assistant Professor of Electrical Engineering at Concordia University in Montreal from 1975 to 1978. He received his M.S. and Ph.D. degrees from the University of Manitoba in 1970 and 1974 respectively, his M.S. degree from National Chiao-Tung University in 1967, and his B.S. degree from National Cheng-Kung University in 1964. His interests are in the general area of VLSI circuits and technology, semiconductor memories, high-frequency server microprocessors, and embedded systems. Dr. Hwang is a member of the New York Academy of Science, Phi Tau Phi and Sigma Xi. He is also an active member of the Chinese American Academic and Professional Society (CAAPS) where he has served separately as President and Chairman of the Board. For his leadership, he received Courvoisier Leadership Award in 1992 and the CAAPS Special Service Award in 1995. Dr. Hwang is Co-Principal Investigator of National SoC Research Program and a Fellow of the Institute of Electrical and Electronics Engineers (IEEE).