Sign Bit Reduction Encoding for Low Power Applications

M. Saneei
Low-Power High-Performance Nanosystems Lab., Elec. and Comp. Eng. Dept., University of Tehran, Tehran, Iran
mohsen.saneei@ece.ut.ac.ir

A. Afzali-Kusha
Low-Power High-Performance Nanosystems Lab., Elec. and Comp. Eng. Dept., University of Tehran, Tehran, Iran
afzali@ut.ac.ir

Z. Navabi
Computer Aided Design Lab., Elec. and Comp. Eng. Dept. University of Tehran, Tehran, Iran
navabi@cad.ece.ut.ac.ir

ABSTRACT

This paper proposes a low power technique, called SBR (Sign Bit Reduction) which may reduce the switching activity in multipliers as well as data buses. Utilizing the multipliers based on this scheme, the dynamic power consumption of some digital systems such as digital filters based on CMOS logic system can be reduced considerably compared to those based on 2's complement implementation. To verify the efficacy of the SBR, a 16-bit multiplier was implemented by this scheme. The results for voice data show an average of 29% to 35% switching reduction compared to the 2's complement implementation. For 16-bit random data, this scheme decreases the switching of 16-bit multipliers by an average of 21%. Finally, the application of the technique to a 16-bit data bus leads up to 14.5% switching reduction on average.

Categories and Subject Descriptors

B. Hardware

B.2 ARITHMETIC AND LOGIC STRUCTURES

General Terms

Design

Keywords

Switching Activity, Low Power, Signed Multiplier, Bus Encoding, Sing Extension.

1. INTRODUCTION

In the recent years, the rapid increase in the complexity of chips and the popularity of portable and mobile devices have caused the power/energy consumption to become one of the main design criteria. A major part of this energy is consumed in the charging and discharging of the device and interconnect/bus capacitances. This component is called the dynamic power and is given by:

$$P_{dynamic} = \sum_{i=0}^{N-1} C_{load,i} . V_{dd} . V_{swing} . f . \alpha_i \qquad (1)$$

where the summation is over all the N nodes of the circuit, $C_{load,i}$ is the load capacitance at node i, V_{dd} is the power supply voltage, V_{swing} is the node high-to-low voltage difference, f is the clock frequency, and α_i is the probability of the switching activity (SA) at node i. Therefore, for reducing the power consumption in CMOS circuits, the number of node switching from high to low must be decreased. In this paper, our objective is to reduce the number of switching.

In DSP systems, a SA reduction in the arithmetic blocks and buses have an important effect on dynamic power consumption. The data encoding scheme may have a significant impact on the SA in the modules. Two's complement (2'sc) number representation is the most commonly used encoding in DSP applications [1]. This is due to ease of performing arithmetic operations such as additions and subtractions [1]. A major drawback of the representation, however, is the sign extension, which causes the MSB sign-bits to switch when signals transition from positive to negative or vice-versa, leading to an increase in the power overhead [1].

One of the ways of analyzing the power consumption in VLSI circuits is to make use of the switching activity (SA) of the circuit. This method has been used for the power estimation at the gate level [3][9][2]. In this paper, we introduce a new scheme, named SBR (*Sign Bit Reduction*), which reduces SA in the multipliers and buses, leading to the reduction of the energy consumption.

The rest of this paper is organized as follows. Section 2 presents a review of the previous works while Section 3 explains the main idea behind the SBR. In Section 4, we implement a typical multiplier by this method as an example. Experimental results are given and discussed in Section 5 with the summary and conclusion given in Section 6.

2. PREVIOUS WORKS

In this section, we will review some of the previously proposed schemes in reducing the power consumption in arithmetic blocks, FIR filters, and buses.

2.1 Power Reduction in Arithmetic Blocks

The choice of number representation has a significant impact on the reduction of the SA in arithmetic block. Several data representations have been suggested to reduce the power

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

consumption in multipliers and adders [11][10]. *Sign Magnitude (SM)* [11] and *Reduced 2'sc* [10] are some of these representations. The use of sign-magnitude avoids extra switching related to 2'sc due to the elimination of the sign extension. This technique, however, causes some difficulty in performing arithmetic operations such as additions and subtractions. In the *Reduced 2'sc* method, the maximum magnitude of a 2'sc number is detected and its reduced representation is dynamically generated to represent the signal. A constant error is introduced by this representation where the error is also dynamically compensated [10].

FIR filtering is one of the most commonly used functions in DSP systems. The output Y[n] of an N-tap FIR filter is given by the weighted sum of latest N inputs as

$$Y(n) = \sum_{i=0}^{N-1} A(i).X(n-i) \qquad (1)$$

where *A(i)* is a filter coefficient and *X(i)* is the input data. Several approaches for reducing the power consumption in FIR filters have been proposed [1],[4]. *Multi-rate Architecture* [4] and *Coefficient Segmentation* [1] are some of these approaches. *Multi-rate Architecture* involves the implementation of the FIR filter in terms of its decimated sub-filters. This architecture can be derived using Winograd's algorithm for reducing computational complexity of polynomial multiplications [4]. The *Coefficient Segmentation* algorithm reduces the SA by decomposing individual coefficients into two less complex subcomponents such that one part is implemented by a simple shift operation and the other by a hardware multiplier. This technique results in a significant reduction in the amount of SA and consequently power consumption [1]. In the approach proposed here we use a new scheme for reducing the SA in a multiplier which may be used in implementing FIR filters.

2.2 Power Reduction in Bus Encoding

Since the capacitances associated with buses in a system are much larger than interconnect/device capacitances, one should pay a special attention to their SA's. Note that in address buses, addresses are normally sequential except when the control flow instructions are encountered or exceptions occur [8]. In contrast, in a data bus, consecutive data in many cases are independent and, therefore, may need more attention in terms of reducing the SA. *Bus Invert* (BI) [6], *Bus Invert and Transaction Signaling* (BITS) [7], and *Partial Bus Invert* (PBI) [5] are some of the examples of the proposed encoding for data buses. In the BI method, if the Hamming distance between the two consecutive N-bit patterns is larger than $N/2$, then the second pattern can be inverted to reduce the inter-pattern hamming distance to below $N/2$ [6]. One redundant bit is needed to distinguish between the original and inverted patterns that are transmitted on the bus. In the BITS method, if the majority of bits in an N-bit pattern are 1, this pattern is inverted and then is XOR'ed with the previous pattern (transition-encoded) [7]. Otherwise, each bit of this pattern is transition-encoded without any change. In PBI, bus width is divided into several parts and every part encoded independent of others [5]. As observed here, in all mentioned approach, the data must be decoded to be ready to use at target module.

3. SBR REPRESENTATION

An N-bit 2'sc number is represented by N bits as

$$X = \{x_{N-1}, x_{N-2}, \ldots, x_1, x_0\}$$

The most significant bit, x_{N-1}, is the sign bit. If X has a magnitude less than 2^{M-1}, x_{M-1} is the sign bit and x_{N-1} to x_M only repeat x_{M-1} and form a string of 0s or 1s which called the sign-extension. We can rewrite X as

$$X = \{x_{M-1}, x_{M-1}, \ldots, x_{M-1}, x_{M-2}, \ldots, x_1, x_0\}$$

where all sign bits are represented by x_{M-1}. Since the sign extension is composed of repeated sign bits, the information contained in the sign extension is redundant and its switching can dissipate some power. For example, let us assume a small number, e.g. 00000010 (+2), on the bus in a clock period. Now, if in the next clock period, a negative small number, e.g. 11111101 (-3), is transmitted through this bus, there occur 8 transitions on the bus. Since these numbers can be shown by 3 bits and, hence, 5 of 8 transitions are not necessary. In our method, these extra transitions are compressed to one transition.

The same transition saving may be applied to a multiplier where the number of redundant transitions is higher. In a multiplier, regardless of the multiplicand, if the multiplier is 11111101 (-3), then there are 7 partial products which equals to 6 additions and 1 subtraction. Because, this number has 6 sign bits, 5 of 7 partial products are unnecessary. As is explained next, the scheme proposed here eliminates these extra partial products.

In this paper we propose a new representation for signed numbers where all sign bits are changed to zero except the first x_{M-1} from the LSB side. Here, an extra sign bit is added to the left side of the number to indicate the first sign bit. Figure 1 shows this idea which we call it Sign Bit Reduction (SBR).

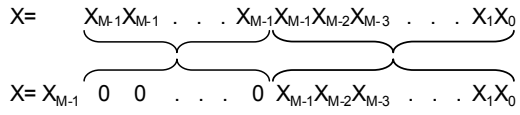

Figure 1 conversion from 2'sc to SBR

For example, two 8-bit 2'sc numbers 11110100 and 00010110 change to two 9-bit SBR numbers 100010100 and 000010110, respectively. So, if X change to -X, in 2'sc representation, N-M sign bit will change while in SBR representation, only 2 sign bits will change. If $N - M > 2$, then the SA in SBR is less than the SA in 2'sc representation. We, therefore, convert 2'sc numbers to SBR representation for one the inputs of the multiplier with an SBR encoder.

Note that most DSP systems have a regular structure which makes the use of this approach more suitable for them. For example, the block diagram of a generic FIR filter using this approach is shown in Figure 2. It consists of two memory blocks for storing the coefficients and the input data samples, two register for holding the coefficient and the input data, an output register, the controller along with the MAC (Multiply and Accumulate), an SBR encoder, and an SBR decoder. The multiplier in MAC is an SBR multiplier which will be explained in the next section. The filter coefficients in the coefficient memory are in SBR format. If the input data samples are in SBR format, we do not need an SBR encoder.

To see the effect of different encodings in the SA of the bus, consider an 8-bit bus whose data in ten successive clocks are -2, 1, 2, -3, 0, -1, 4, -3, 2, and -2. For this set of data, the binary representations of the data for 2'sc, SM and SBR encodings are shown in Figure 3. As is observed from this figure, there are 67, 26, and 24 transitions (SA) for 2'sc, SBR, and SM, respectively. This shows about a 62% reduction in SA for SBR and SM in comparison to the 2'sc. Of course, if the magnitudes of the data are large, then the SA for all three encodings converges.

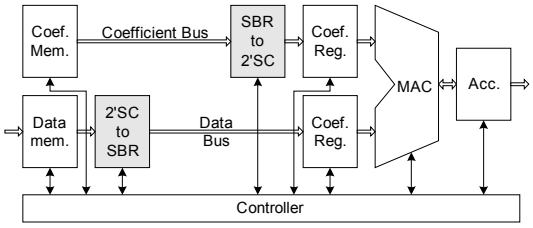

Figure 2 Block Diagram of an FIR Filter.

For an N bit data input, the SBR encoder circuit contains (3N-4) 2-input gates and corresponding decoder circuit contain (2N-3) 2-input gates and (N-2) inverters. Figure 4 shows these circuits where X is a 2'sc number and Y is its corresponding SBR representation.

(a)	(b)	(c)
0000 0000	0 0000 0000	0000 0000
1111 1110	1 0000 0010	1000 0010
0000 0001	0 0000 0001	0000 0001
0000 0010	0 0000 0010	0000 0010
1111 1101	1 0000 0101	1000 0011
0000 0000	0 0000 0000	0000 0000
1111 1111	1 0000 0001	1000 0001
0000 0100	0 0000 0100	0000 0100
1111 1101	1 0000 0101	1000 0011
0000 0010	0 0000 0010	0000 0010
1111 1110	1 0000 0010	1000 0010

Figure 3. 10 successive data on bus in a) 2'sc b) SBR c) SM

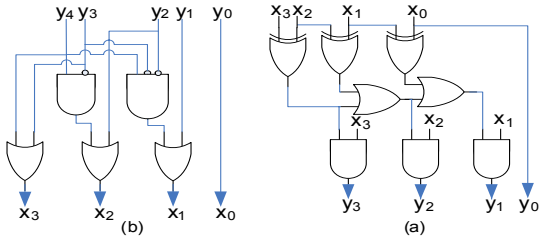

Figure 4. a) 2'sc to SBR encoder (4 bit) b) SBR to 2'sc decoder (4 bit)

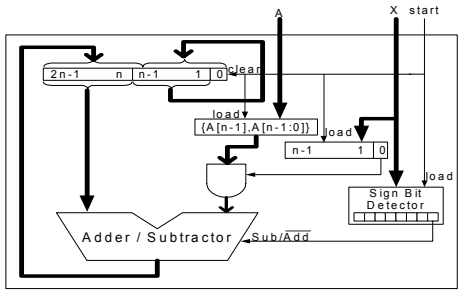

Figure 5. Multiplier block diagram with SBR input

4. MULTIPLICATION WITH SBR INPUT DATA

As mentioned above, usually one of two inputs of multipliers in FIR filters is input data that coming from a digital signal such as voice or video, and the second input is a coefficient vector. If the input data is coded by SBR representation, the power consumption in the buses and computational blocks will be reduced. In this section, we explain a shift/add multiplier which is suitable for SBR inputs. Figure 5 shows our multiplier called *shift/add SBR multiplier*. In this architecture, X is multiplier with SBR representation while A is multiplicand with 2'sc representation. In an FIR filter, X can be input data and A is a coefficient.

Equation (2) and (3) show the multiplication in 2'sc and SBR encoding, respectively. In the worst case, 2'sc multiplier requires N partial products while SBR multiplier requires only M partial products.

$$X.A = -2^{N-1}.x_{N-1}.A + \sum_{i=0}^{N-2} 2^i x_i A \qquad (2)$$

$$X.A = -2^{M-1} x_{M-1} A + \sum_{i=0}^{M-2} 2^i x_i A \qquad (3)$$

A 2'sc number in the range of [-2N-1, +2N-1) requires N-bit for the representation while the same number in SBR encoding should be shown with N+1 bits. If X∈[-2N-2,0), number of 1s in SBR code (except redundant bit) is less than the 1s in the 2'sc code. Thus, in a shift/add SBR multiplier, the number of partial products will be less than a shift/add 2'sc multiplier. Figure 6 shows an example of the multiplication of two numbers in 2'sc and SBR multipliers. In this figure, 00001101 (+13) is multiplied by 11111101 (-3). In SBR multiplier, the sign bit is the third bit and there are only 2 partial products. In Figure 5, at the first clock of every multiplication, the *sign-bit-detector* block detects the sign bit location of SBR input data and then set the corresponding bit to 1 and the rest of the bits to zero. For the above example, after detecting the sign bit, it contains the pattern "00000100". For the following clocks, this block only shifts right this pattern bit by bit where its LSB controls sub/add input of adder. This explanation indicates that this block does not affect the critical path.

```
    00001101 x
    11111101
    --------
     00001101
    00001101
   00001101
  00001101
 00001101
00001101
- 00001101
---------------
1111111111011001
      (a)
```

```
    00001101 x
    100000101
    ---------
     00001101
   - 00001101
---------------
1111111111011001
      (b)
```

Figure 6 2'sc and SBR multiplication a) 2'sc b) SBR

5. RESULTS AND DISCUSSION

To evaluate the efficiency of SBR encoding in the SA reduction of data bus and multiplier, we simulate our design with the following set of input patterns.

- Speech data: a man speech with 16-bit samples and sampling frequency of 41.667 KHz that is extracted from MATLAB.
- Water run voice: Voice of waterfall with 16-bit samples that is extracted from a wave file of the Windows XP operating system.
- Random data: 16-bit random data.

Table 1 shows the features of the patterns.

Table 1 Specifications of input patterns.

Input Data	Positive		Negative		Total
	# of Samples	Average	# of Samples	Average	Average
Speech	8315	1128	1684	-2471	522.4
Water-run	5084	4099	4915	-4231	4.43
Random	5000	16310	5000	-16175	67.6

We design an SBR multiplier and its corresponding 2'sc multiplier using VHDL. In the SBR multiplier, multiplicand (A) is a 2'sc random number and multiplier (X) is a SBR number from the patterns of Table 1. We synthesize two multipliers and extract the numbers of switching on all the circuit nodes for 10,000 data samples. The resulting numbers of the switching for the two multipliers are shown in Table 2.

Table 2 Number of the switching in two multipliers (×1000).

Input Data	2'sc Multiplier	SBR Multiplier	Switching Reduction	Percentage Switching Reduction
Speech	15047	9790	5257	34.94
Water	15706	11168	4538	28.89
Random	16905	13365	3540	20.94

The switching reductions in the SBR multiplier (including the encoder) for the random data is about 21% and for the two types of voice data are about 29% and 35%. The switching reduction for the speech data is more than other data due to the fact that the average absolute value for the speech samples is smaller than that for the water-run samples and the random data. Thus, the average number of sign bits in speech samples is more than that of the water-run samples and random data. This means partial product saving in speech samples is more than water-run samples.

We also report the results that were obtained from the data transfer on a 16-bit bus. In this experiment, we used a 16-bit bus and 10,000 samples of the three types of input patterns. Table 3 shows the results of switching in the bus for 2'sc and SBR samples.

Table 3 Number of the switching in a 16-bit bus.

Input Data	2'sc Data	SBR Data	SBR Reduction.
Speech	44072	42856	2.76%
Water run	70344	66045	6.11%
Speech with 1/3 sample rate	61537	56975	7.41%
Water run with 1/3 sample rate	74222	63474	14.48%
Random data	80046	78271	2.22%

As the table reveals, the application of our technique for the random data does not lead to a considerable reduction in the bus power consumption. For the water run data which have more negative values, it reduces the switching up to 14.5%. When two successive data have different signs and absolute values, the technique proposed here has the maximum effect in reducing switching in the bus. In table 3, reducing the sample rate by 1/3 leads to more sign changes in the successive input data and more switching reductions. The power reduction is not significantly high for pure bus encoding applications.

6. SUMMARY AND CONCLUSIONS

In this paper, we proposed a new data encoding techniques, named SBR (Sign Bit Reduction) to decrease the switching activity on a data bus and a signed multiplier. The effect of our techniques in reducing switching activity (SA) in multiplier is more than SA in buses. Experimental results of the simulated multiplier circuit (including encoder) show that our techniques can achieve up to 35% reduction in switching activity for the multiplication and up to 14.5% reduction in switching activity for the data buses.

7. REFERENCES

[1] Erdogan, T. and Arslan, T. A Coefficient Segmentation Algorithm for Low Power Implementation of FIR filters. Proceedings of the 1999 IEEE International Symposium on Circ. and Sys., Vol. 3 , 30 May-2 June 1999, PP 359 – 362

[2] Ghosh, A. Devadas, S. Keutzer, K. and White, J. Estimation of average switch activity in combinational and sequential circuits. in proc. 29th DAC Conf., June 1992, pp. 253-259

[3] Landman, P. E. and Rabaey, J. M. Architectural Power Analysis: The Dual Bit Type Method. IEEE Trans, on VLSI system, vol. 3, no. 2, June 1995, pp. 173-187

[4] Mou, Z. J. and Duhamel, P. Short-length FIR filters and their use in fast nonrecursive filtering. IEEE Trans. Signal Processing, June 1991, pp. 1322–1323.

[5] Shin, Y. Choi, K. and Chang, Y. H. Narrow Bus Encoding for Low-Power DSP Systems. IEEE Trans. VLSI Syst., vol. 9, no. 5, pp. 656-660, October 2001

[6] Stan, M. R. Burleson, W. P. Bus-Invert Coding for Low-Power I/O. IEEE Trans. VLSI Syst., vol. 3, no. 1, pp. 49-58, March 1995

[7] Stan, M. R. Burleson, W. P. Limited-Weight Codes for Low Power I/O. International Workshop on Low Power Design, Napa, CA, Apr. 1994

[8] Su, C. L. Tsui, C. Y. and Despain, A. M. Low power architecture design and compilation technique for high-performance processors. in Proc. IEEE COMPCON, San Francisco, CA, Feb. 1994, pp. 209–214

[9] Tjarnstrom, R. Power Dissipation Estimate by switch level simulation. Proc. IEEE ISCAS, May 1989, pp. 881-884

[10] Yu, Z. Yu, M. L. Azadet, K. and Willson, A. N. Jr. the use of reduced two's complement representation in low power DSP design. IEEE ISCAS, vol. 1 , May 2002 pp. I-77 - I-80

[11] Zheng, M. and Albicki, A. Low Power and High Speed Multiplication Design Through Mixed Number Representations. International Conference on Computer Design, October 02-04, 1995, Austin, Texas