Dynamic Operand Transformation for Low-Power Multiplier-Accumulator Design

Fujino, Masayoshi
Department of Electronics Engineering and Computer Science, Fukuoka University

Moshnyaga, Vasily G.
Department of Electronics Engineering and Computer Science, Fukuoka University

https://hdl.handle.net/2324/7658

出版情報：Proceedings of the Internatipnal Symposium on Circuits and Systems. 5, pp.345-348, 2003-05
バージョン：
権利関係：

DYNAMIC OPERAND TRANSFORMATION FOR LOW-POWER MULTIPLIER-ACCUMULATOR DESIGN

Masayoshi Fujino and Vasily G. Moshnyaga
Dept. Electronics Engineering and Computer Science, Fukuoka University
8-19-1 Nanakuma, Jonan-ku, Fukuoka 814-0180, JAPAN
Tel/Fax: (+81) 92-801-0833,
email: {fujino,vasily}@v.tl.fukuoka-u.ac.jp

Abstract: The design of portable battery-operated devices requires low-power computation circuits. This paper presents a new multiplier-accumulator (MAC) design approach, which in contrast to existing methods exploits dynamic operand transformation to reduce power consumption. The key idea is to compare current values of input operands with previous values and depending on computed Hamming distance to use either original or two's complement form of the operands in order to decrease the transition activity of multiplication. Experiments show that such a formulation outperforms the related approaches minimizing the power dissipation of traditional MAC design almost by half with 31% area and 12% delay overhead. The circuit implementation is outlined.

1. Introduction

1.1. Motivation

Multiplier-Accumulators (MAC) are essential arithmetic blocks for many applications. To achieve high execution speed, parallel (array) multipliers and fast carry propagate adders are widely used. Due to the high capacitive load and large bit-width, these MAC structures become the most energy-consuming units in modern Digital Signal Processors. In the NEC's 16-bit SPX processor, for example, two MAC units dissipate almost 2/3 of the total power [1]. As result optimizing the MAC for energy is important.

In static CMOS circuits, transition activity dominates the total energy dissipation due to charging and discharging of capacitors. Given the average load capacitance (*C*), the supply voltage (V_{dd}), and the number (*a*) of energy consuming signal transitions per operation, the average energy dissipation of a CMOS MAC can be expressed by $E_{avg} = a*C*V^2$. Although reducing energy dissipation amounts to all of these factors, the energy saving obtained by lowering the transition activity per operation (*a*), is fairly independent of integration technology and hence less expensive.

This paper focuses on the transition activity reduction in MAC structures and presents a new technique, which involves dynamic operand transformation.

1.2. Related Research

There have been reported a number of works on transition activity reduction in digital MAC structures. Chandrakasan, et al [2] indicated that sign-magnitude representation, operation ordering, and algebraic transformations are very advantageous for transition activity minimization. Callaway, et al. [3] investigated various multiplier structures and demonstrated that switching activity within just the partial product reduction hardware is substantially better for the tree structure over the array, if one ignores the wires. Nishimura, et al. [4] proposed to insert AND gates into the MAC structure to avoid unwanted spurious transitions through the carry save array. Song, et al. [5] suggested a data-detecting module, which is incorporated into multiplier to selectively activate its hardware and thus decrease unnecessary signal transitions. Lemonds, et al. [6] utilized synchronization latches to eliminate race glitches. Musoll, et al. [7] used transition-retaining barriers to stop the transitions until the logic block is enabled. Moshnyaga, et al. [8] reported on activity reduction due to the adding compressors, the modified sign-extension, and encoding. Sakuta, et al. [9] and Sobelman, et al.[10] advocated to balance delays within the multiplier in order to minimize spurious transitions. Schulte, et al. [11] exploited bit-truncation as a mean to reduce both the switching activity and the area of multipliers. Zheng, et al. [12] studied a mixed number representation with canonical sign digit numbers in order to stop glitches within parallel multipliers. Oban, et al. [13] reduced transitions by bypassing additions whenever the multiplier bit was zero.

Despite differences all these techniques have one feature in common. They optimize internal multiplier structure, which however may not always be possible, especially when the multiplier comes to the designer as a library unit or an IP property. Up to our knowledge, the only approach capable of lowering the signal transitions in multipliers without affecting its internal structure has been reported so far. The operand-interchange method proposed by Ahn, et al. [14] is based on the fact that positions of two inputs to the multiplier unit can considerably affect its power consumption. The main idea is to swap the input operands when both of them change their signs. Although the method is able to interchange the operands dynamically without a large area and timing overhead, it is limited to the only case, when both operands alter their signs. For the cases of single operand variation, the method however, is inefficient.

1.3 Contribution

In this paper we propose a new approach to switching activity reduction in MAC. In comparison to the existing research, the approach exploits another way to transition activity optimization, namely operand transformation, by modifying the multiplier inputs dynamically in order to achieve low power operation of

MAC. Experiments show that such an approach can reduce the MAC’s power considerably.

The rest of the paper is organized as follows. Section 2 discusses the proposed approach and outlines its implementation. Section 3 shows experimental results. Section 4 presents conclusions.

2. The Proposed Approach

2.1. Main idea

The approach we propose is based on the observation that the fewer input bits of multiplier transit the less switching activity within the MAC circuitry. Our main idea is to control the input operands of MAC and dynamically transform them to the 2’s complement form whenever more than half of bits transit. If both operands are transformed, the product is computed as usual. Otherwise, it is also transformed to its two’s complement form.

	traditional		operand interchange		Ours	
	A	B	A	B	A	E
t0	00011	00100	00011	00100	00011	00100
t1	11101	00101	00101	11101	10011	00101
input trans.	5		5		2	

(a)

```
           11101  (A=-3)
             ↓
           00011  (A=+3)
        *) 00101   (B=5)
        ---------------------
           00011
   +)     00000
         00011
        ---------------------
(b)      0001111 (temporal product)
         1110000 (inverse)
        ---------------------
         1110001 result (-15)
```

Figure 1: An illustration of the proposed approach: (a) input transitions; (b) the modified multiplication

Figure 1 illustrates the proposed approach on a simple example. Here, *A* and *B* denote the MAC inputs, *t0* and *t1* the clock cycles. In traditional design, the new values, which fed the multiplier in the clock *t1*, cause 5 input bits to transit, as shown in Fig.1(a). Our approach dynamically detects that the new value of the multiplicand *A* differs from its previous one by 4 bits and therefore transforms it to the 2’s complement form, reducing the total number of input transitions to 2. Notice, that interchanging the operands, as it proposed in [14], does not help: it produces 5 input transitions. To ensure correct computation, our approach also changes the product to its 2’s complement form, as shown in Fig.1(b). Due to small capacitive load, this extra signal switching however dissipates less power, than the input transitions; so the total MAC power reduction becomes possible.

2.1. Implementation scheme

Figure 2 shows an implementation scheme. Here, blocks labeled by *L* define latches, + is adder, *m1, m2, m3* are multiplexors. We assume that input operands *A, B* of the multiplier are stored in registers *rA, rB*, respectively. The Decision Logic placed on each input compares the incoming operand value, *A(t)* or *B(t),* with the value, *A(t-1)* or *B(t-1),* currently used in the multiplication and sets to 1 the corresponding control signal (*c1* or *c2)* to select the complemented outputs of the registers and define the sign of the product. These signals are delayed one clock cycle and applied to the multiplexors *m1*, *m2* when the operands *A(t), B(t)* are fed to the multiplier. When both signals *c1=c2*, the accumulator uses the computed product as it is. Otherwise, the product’s inverse taken from the multiplexor *m3* is added to the signal *cP* to produce the 2’s complement form of the product.

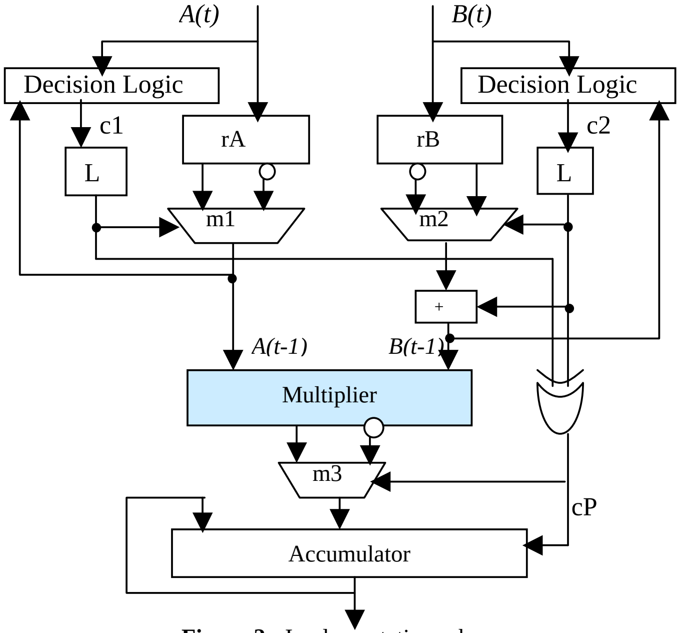

Figure 2: Implementation scheme

To reduce the cost of Hamming-Distance Detection, we implement the Decision Logic based on two data-dependent propagation paths (P1, P2), as shown in Figure 3. The key feature is that the delay of each path varies proportionally to the number of ones on its inputs. Data inputs to the path P0 are fixed (half of them are zeros, and half are ones) while the inputs (*X*) to the path P1 come from the comparison of *A(t)* and *A(t-1).* That is, the signal *X* has as many ones as there are different bits in pairs among *A(t)* and *A(t-1).* Because the path delay increases with the number of ones on inputs, the shortest path eventually shows whether the number of ones in *X* is larger than half of the bit-width or visa versa. The circuit operates as follows. When CLK is low, the transistors T1, T2 switch ON, connecting the nodes V and U to the ground and producing the high impedance on the output. When CLK is high, the transistors T1, T2 are OFF, the transistors T3, T4 are ON; so the pulse CLK will be propagated through that DE which has less number of ones on its input. The fastest circuit changes the voltage level on node V or U, disabling the other propagation path. Thus Out=0 if more than half of bits in the word *X* are ones; otherwise Out=1.

Figure 4 exemplifies a 4-bit path that adjusts its delay to the number of ones in the binary word *X*. The clock pulse CLK fetched from the left propagates to the right of the circuit, changing the polarity by each inverter. When the input signal X_{i-1} is low, the inverter is grounded to zero through an nFET, which is always ON. When X_{i-1} is high, the inverter is grounded to zero through two parallel FETs. In the latter case, the inverter delay is shorter than in the former case because of the lower source resistance in the pull-down operation. After the pulse propagates through the inverter, its polarity is changed, so the subsequent delay circuit is configured as a dual one using the pFETs for pull-up in place of the nFETs for pull-down. Thus the delay of the nFET based inverter is short, when A_{i-1}=1, and long, when when A_{i-1}=0. In opposite, the pFET controlled inverter has a short delay, when Ai =0, and a long delay, when A_i =1.

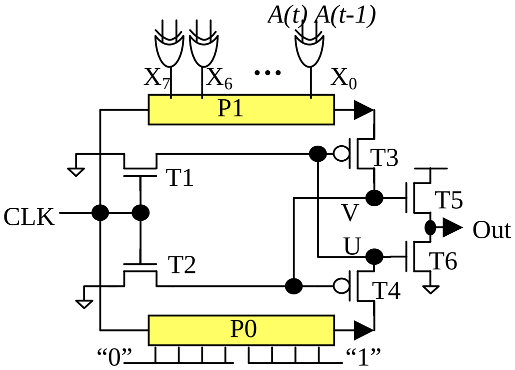

Figure 3. The delay-based implementation of the Decision Logic

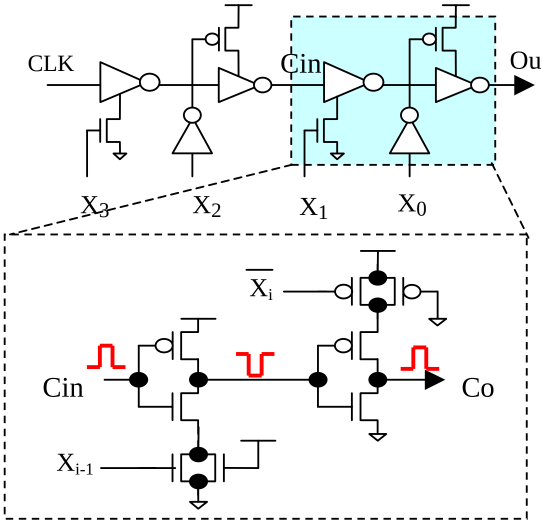

Figure 4. The variable Delay Circuit

3. Experimental Results

We evaluated the proposed approach on a standard MAC configuration used in DSP application (8x8-bit multiplier, 24-bit accumulator) and compared it to three MAC designs built on the traditional non-optimized Braun's array multiplier [15]; the Brown’s multiplier with operand interchange [12]; and the Brown multiplier with bypassing of partial products [14]. All the designs were implemented with 0.35µm CMOS standard cell technology using the 36-transistor 1-bit full adder [16] and the 4-transistor 2-to-1multiplexor. The power consumption was measured at 3.3V with POWERMILL [17] using data taken from two standard video streams (*Football* and *Tennis*), when one input of the MAC received 8-bit video stream data (240x320 pixels per frame) and the other one was fed by 64 Discrete Cosine Transform (DCT) coefficients of 8 bit each.

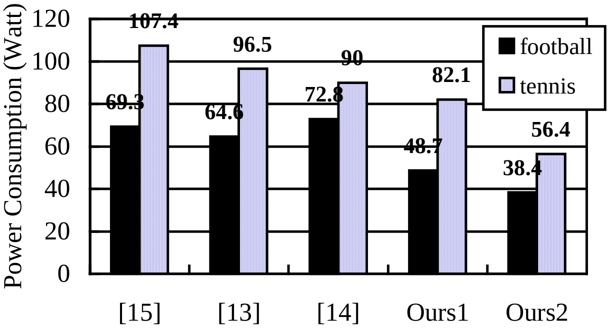

Figure 5: Power consumption

Figure 5 shows the simulation results in terms of average power dissipated per input pattern. In this figure, the related designs are denoted by corresponding references; O*urs1* depicts the proposed MAC design with Decision Logic circuits placed on each input of the multiplier; *Ours2* characterizes a reduced implementation of our MAC design with a single Decision Logic (placed at input B). We observe that the proposed approach outperforms the related optimization techniques saving as much as 29.7% of the total power on the *football* data when both MAC inputs are transformed (*Ours1*), and 44.6% when only one input, namely the multiplier (B), is transformed (*Ours2*). For the *tennis* video stream data, the savings are 23.3% and 47.3%, respectively. The difference in savings between our two designs can be explained by additional switching activity imposed by the overhead circuitry. We experimentally found, that the frequency of the operand transformation as well as the product change is almost by one third higher for the Ours1 design than for the Ours2. The decision logic itself, however, does not take a large amount of power.

Figure 6 shows the power/delay estimation of the Decision Logic. We see that the proposed delay-based implementation consumes only 0.44 mW of power for the worst case when all the bits are ones. Also, though the circuit delay depends on the input pattern, the maximal delay of the 8-bit chain is quite small (only 4 ns).

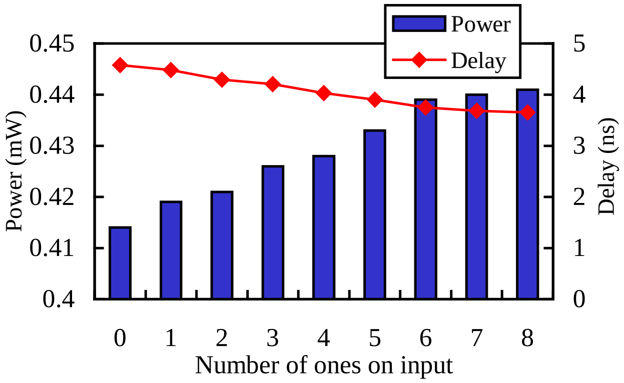

Figure 6: Decision Logic power dissipation and delay as a function of the number of ones on inputs

Table 1 outlines the Area/Delay overhead of the compared designs as a ratio to the traditional MAC implementation. In comparison to the tested structures the proposed approach has a larger delay mainly because of the adder, involved in producing the two's complement of the input operand. By restricting the operand transformation to the multiplier only, we reduce the area overhead to 31 %, while increasing the power savings significantly.

Table 1: Area/ Performance overhead

	Trad	[13]	[14]	Ours1	Ours2
Area (%)	-	49	4	55	31
Delay (%)	-	3.4	0.3	12	12

4. Conclusion

We presented a novel technique for reducing power consumption of digital multiplier-accumulator. The technique differs to existing research by exploiting a new freedom in the transition activity optimization, namely operand transformation. By dynamically transforming the operand to that representation which enables fewer transitions on input, the approach is able to diminish the total amount of signal transitions in the high-capacitive multiplier array and therefore save power. As experiments showed, the proposed technique can almost halve the total power of the traditional MAC design on the real video data processing application. Unfortunately, up to date we have been unable to experimentally compare our approach to the related research on large inputs. To provide such a comparison, we are currently working on a prototype MAC design, which includes a 16x1-bit multiplier and a 54-bit adder. We expect even larger power savings for the design due to the large multiplier array.

References

[1] T. Nishitani, "Micro-programmable DSP Chip'', 14th Workshop on Circuits and Systems in Karuizawa, Digest Tech. Papers, pp. 279-280, 2001 (in Japanese).

[2] A. Chandrakasan, et al., ''Design of Portable systems'', Proc. IEEE 1994 Custom Integrated Circuits Conference, pp.12.1.1-12.1.8, 1994.

[3] T. K. Callaway and E. E. Swartzlander Jr., "Low Power Arithmetic Circuits", in Low Power Design Methodologies, J. Rabaey and M. Pedram, eds. pp.161-198, Kluwer Ac. Publ., 1996.

[4] E. Nishimura, T. Nakamura and H. Ishida, ''Multiplying Unit Circuit '', US Patent No.5010510, 1991.

[5] M. Song and K. Asada, "Design Methodology for Low-Power Data Compressors Based on Window Detector in a 54x54 bit Multiplier", Proc. IEEE Int. Symposium on Circuits and Systems, pp. 1564-1567, 1995.

[6] C. Lemonds and S. Shetti,, ''A Low Power 16 by 16 bit Multiplier Using Transition Reduction Circuitry'', Proc. Int. Workshop on Low Power Design, pp. 139-142, 1994.

[7] E. Musoll and J. Cortadella, "High-Level Synthesis techniques for Reducing the Activity of Functional Units", Int. Symposium on Low Power Design", pp.99-104, 1995.

[8] V.G. Moshnyaga and K.Tamaru,, "A Comparative Study of Switching Activity Reduction Techniques for Design of Low-Power Multipliers", Proc. IEEE Int. Symposium on Circuits and Systems, pp. 1560-1563, 1995.

[9] T. Sakuta, W. Lee, and P.T. Balsara, "Delay Balanced Multipliers for Low Power/Low Voltage DSP Core", 1995 IEEE Int. Symposium on Low Power Electronics, pp.36-37, 1995.

[10] G. Sobelman, and D. Raatz, "Low Power Multiplier Design Using Delays Evaluation", Proc. IEEE Int. Symposium on Circuits and Systems, pp. 1564-1567, 1995.

[11] M.J. Schulte and J. E. Stine, "Reduced Power Dissipation Trough Truncated Multiplication", Proc. IEEE Alessandro VOLTA Memorial Workshop on Low-Power Design, 1998.

[12] M. Zheng and A. Albicki, "Low Power and High Speed Multiplication Design Through Mixed Number Representation", Proc. IEEE Int. Conf. on Computer Design", pp.566-570, 1995.

[13] J. Oban, et al., "Multiplier Energy Reduction through Bypassing of Partial Products", IEEE Asia South Pathific Circuits and Systems, Sept. 2002.

[14] T. Ahn and K.Choi, "Dynamic Operand Interchange for Low Power", IEE Electronic Letters, Sept.1997.

[15] K. Hwang, "Computer Arithmetic: Principles, Architecture and Design", John Willey and Sons, 1979.

[16] K. Yano, T. Yamanaka, T. Nishida, et al., "A 3.8ns CMOS 16x16-b Multiplier Using Complementary Pass Transistor Logic", IEEE Journal on Solid-State Circuits, Vol.25, No. 2, pp. 388-395, April 1990.

[17] C. Deng, ``Power Analysis of CMOS/BiCMOS circuits", Proc. 1994 Int. Workshop on Low Power Design, pp.3-8, 1994.