Optimized Multiplier Structure Using Radix -2 with Truncated Multiplier

P.Tharani[1] K.Baskaran[2]
[1] PG Scholar [2]Assistant Professor
[1, 2] Centre For Advanced Research, Muthayammal Engineering College,Rasipuram,

Abstract--- Truncated multiplication reduces part of the power required by multipliers by only computing the most-significant bits of the product. The truncation and radix-2 multiplication includes physical reduction of the partial product matrix and a compensation for the reduced bits via different hardware compensation sub circuits. However, these results in fixed systems optimized for a given application at design time. A novel approach to truncation is proposed, where a full precision multiplier is implemented, but the active section of the partial product matrix is selected dynamically at run-time. This allows a power reduction tradeoff against signal degradation which can be modified at run time. Such an architecture brings together the power reduction benefits from truncated multipliers and the flexibility of reconfigurable and general purpose devices. Efficient implementation of such a multiplier is presented in a custom digital signal processor where the concept of software compensation is introduced and analyzed for different applications. Experimental results and power measurements are studied, including power measurements from both post-synthesis simulations and a fabricated IC implementation. This is the first system-level DSP core using a fine-grain truncated multiplier. Results demonstrate the effectiveness of the programmable truncated MAC (PTMAC) in achieving power reduction, with minimum impact on functionality for a number of applications. Software compensation is also shown to be effective when deploying truncated multipliers in a system.

keywords: — Arithmetic, flexible dsp, low power, Radix -2 with truncated multiplier, truncated multiplication.

I. Introduction

The high increase of portable communication and computing devices and the advance in mobile multimedia systems has made power consumption critical to optimize in the design digital signal processing architectures. Advances in very large scale integrated-circuit technology (VLSI) has been one of the major driving forces for the growth of digital signal processors (DSP), enabling the implementation of complex algorithms in programmable DSP architectures and fixed application specific hardware. Within DSP systems, multipliers are among the most fundamental building blocks and parallel implementations, required. this paper represent the main component in terms of power consumption of any hardware dedicated to complex mathematic functions, such as filtering, compression, or classification. The relationship between the physical characteristics of the multiplier and its resolution determine the efficiency and accuracy of the systems. The optimization of multipliers in terms of area, power and timing has been extensively studied in the past Full or direct multiplier implementation

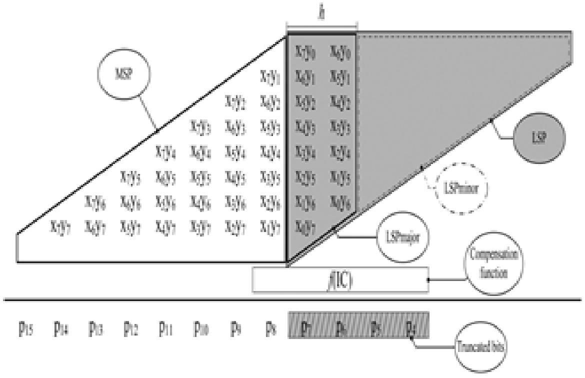


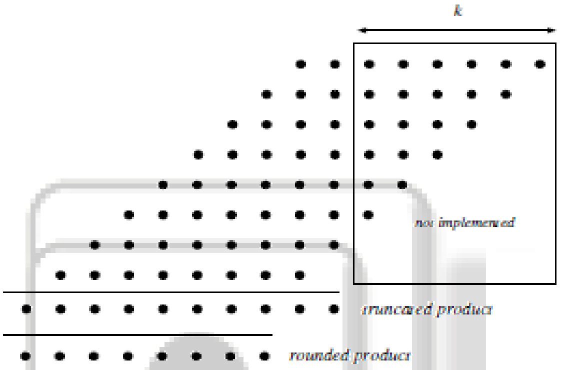
Fig. 1: Partial product matrix for a fixedwidth 16-bit truncated multiplier.

Such techniques skip the implementation and/or disable parts of the partial product matrix to trade energy spent by the computational process for a degradation on the output signal. These techniques fall into two categories, namely Word length Reduction and Truncated Multiplier. Word length reduction techniques minimize switching activity at the expense of data precision by input shifting or truncation, but since the reduction is done from the input operands, power reductions are obtained at the cost of high levels of output noise Truncated multiplication structures based on Baugh- Wooley or Booth algorithms, do not (by design) compute the lowest sections of the partial product matrix. By doing so, a certain error is introduced in the output while savings in power, area, complexity, and timing are achieved. Focused on fixed-width multipliers that produce a fixed -bit output, and their gain-vs-error ratio is set by hardware. Configurable techniques result in flexible architectures, which while losing their benefits resulting from area optimization and static power reduction, provide the multiplier with the advantage of adaptability. This is desirable in systems with a certain degree of programmability. The architecture presented in this paper, a programmable truncated multiplier (PTM), describes a full-precision multiplier, where the elements of the partial product matrix can be disabled through an external control word in a column-wise mode. Benefits of PTMs include:

- Real-time control of the power-SNR exchange in applications where power modes are switchable at run-time.
- Flexibility on accuracy selection for programmable devices such as DSP structures, that would benefit from different truncation levels for different applications.

II. Truncated Multipliers

Multiplication is a common requirement in DSP systems and plays an important role in terms of power consumption. In those systems where it is not necessary to compute the exact least significant part of the product, truncated multipliers achieve power, area and timing improvements by skipping the implementation of a part of the least significant part of the partial product matrix. Instead of computing the full-precision output, the output results from the sum of the most significant columns (where) plus an estimation of the unformed bits. Fig. 1 displays a generic partial product matrix, where the partial product matrix is split into two main regions, the least significant part (LSP), which contains the least significant columns of the partial product matrix, and the most significant part (MSP), that includes the most significant columns of partial product terms. LSP can be also be split in two regions, LSP major being the most significant column, and LSP minor being the least significant columns of the partial product matrix. The product resulting from a full-width multiplier, where the partial product is fully implemented with an exact -bit result can be described as , where is the sum of the partial product bits belonging to MSP and is the sum of those belonging to the LSP. In many applications, product values generated by fixed width bit multipliers are truncated or rounded back to the original bit width in latter stages of the algorithm flow.



.

Fig. 2: Bit-matrix of a truncated magnitude multiplier

III. FIR FILTER BASED ON TRUNCATED MULTIPLIER

Finite impulse response (FIR) digital filter is one of the fundamental components in many digital signal processing (DSP) and communication. It is also widely used in many portable applications which require fast operations. In general the transfer function for a linear, time-invariant, filter can be expressed in Z domain as,

$$H\ (z) = \frac{B(Z)}{A(Z)} = \frac{b_0+b_1z^{-1}+\ \ldots\ldots..+b_Nz^{-N}}{1+a_1z^{-1}+\ \ldots\ldots.+a_Mz^{-M}}$$

When the denominator part is made equal to unity then the equation becomes for an FIR filter. So the final equation for the FIR filter of order N can be expressed as,

$$Y(n) = b_0x(n) + b_1\,x(n-1) + \ \ldots + \ b_n\,x(n-N)$$
$$= \sum_{i=0}^{N} b_i \text{ x (n-i)}$$

Single-Multiplier MAC FIR Filter

The single-multiplier MAC FIR is one of the simplest DSP filter structures. The MAC structure uses a single multiplier with an accumulator to implement a FIR filter sequentially versus a full parallel FIR filter. This trade-off reduces hardware by a factor of N, but also reduces filter throughput by the same factor. The general FIR filter equation is a summation of products defined as,

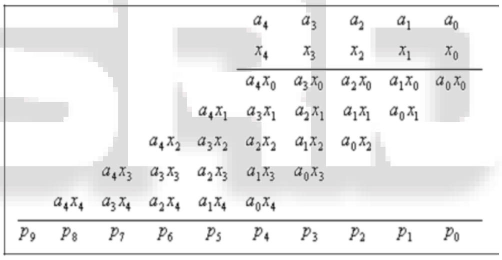
$$\text{Y (n)} = \sum_{i=0}^{N-1} X_{n-i}\ h_i$$

The single MAC FIR filter is well suited and dual-port block RAM is the optimal choice for the memory buffer. The filter coefficients are also stored in the same dual-port block RAM, and are output at port B. Hence, the RAM is used in a mixed-mode configuration. The data is written and read from port A (RAM mode), and the coefficients are read only from port B (ROM mode). The control logic provides the necessary address logic for the dual-port block RAM and creates a cyclic RAM buffer for port A (data buffer) to create the FIR filter delay line. An optional output capture register maybe required for streaming operation, if the accumulation result cannot be immediately used in downstream processing.

IV. FRAME WORK

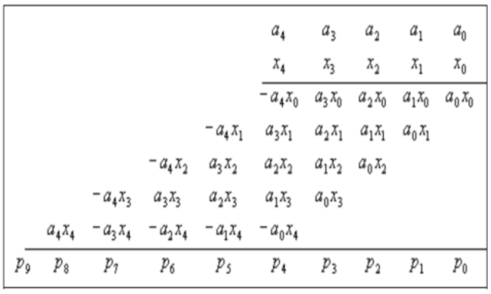
A. *Baugh-Wooley Two's Complement Multiplier:*

Baugh–Wooley technique was developed to design direct multipliers for Two’s compliment numbers. When multiplying two’s compliment numbers directly, each of the partial products to be added is a signed numbers. Thus each partial product has to be sign extended to the width of the final product in order to form a correct sum by the Carry Save Adder (CSA) tree. According to Baugh-Wooley approach, an efficient method of adding extra entries to the bit matrix suggested to avoid having deal with the negatively weighted bits in the partial product matrix. Baugh-Wooley schemes become an area consuming when operands are greater than or equal to 32 bits.



					a_4	a_3	a_2	a_1	a_0
					x_4	x_3	x_2	x_1	x_0
					a_4x_0	a_3x_0	a_2x_0	a_1x_0	a_0x_0
				a_4x_1	a_3x_1	a_2x_1	a_1x_1	a_0x_1	
			a_4x_2	a_3x_2	a_2x_2	a_1x_2	a_0x_2		
		a_4x_3	a_3x_3	a_2x_3	a_1x_3	a_0x_3			
	a_4x_4	a_3x_4	a_2x_4	a_1x_4	a_0x_4				
p_9	p_8	p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

Fig: 3: 5*5 unsigned multiplications



					a_4	a_3	a_2	a_1	a_0
					x_4	x_3	x_2	x_1	x_0
					$-a_4x_0$	a_3x_0	a_2x_0	a_1x_0	a_0x_0
				$-a_4x_1$	a_3x_1	a_2x_1	a_1x_1	a_0x_1	
			$-a_4x_2$	a_3x_2	a_2x_2	a_1x_2	a_0x_2		
		$-a_4x_3$	a_3x_3	a_2x_3	a_1x_3	a_0x_3			
	a_4x_4	$-a_3x_4$	$-a_2x_4$	$-a_1x_4$	$-a_0x_4$				
p_9	p_8	p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

Fig:4: 5*5 Signed Multiplication

Baugh-Wooley Multiplier is used for both unsigned and signed number multiplication. Signed Number operands which are represented in 2 have complemented form. Partial Products are adjusted such that negative sign move tolast step, which in turn maximize the regularity of the multiplication array. Baugh-Wooley Multiplier operates on signed operands with 2’s complement representation to make sure that the signs of all partial products are positive.

V. BOOTH ALGORITHM

It is a powerful algorithm for signed-number multiplication, which treats both positive and negative numbers uniformly. For the standard add-shift operation, each multiplier bit generates one multiple of the multiplicand to be added to the partial product. If the multiplier is very large, then a large number of multiplicands have to be added. In this case the delay of multiplier is determined mainly by the number of additions to be performed. If there is a way to reduce the number of the additions, the performance will get better. Booth algorithm is a method that will reduce the number of multiplicand multiples. For a given range of numbers to be represented, a higher representation radix leads to fewer digits. Since a k-bit binary number can be interpreted as K/2-digit radix-4 number, a K/3-digit radix-8 number, and so on, it can deal with more than one bit of the multiplier in each cycle by using high radix multiplication

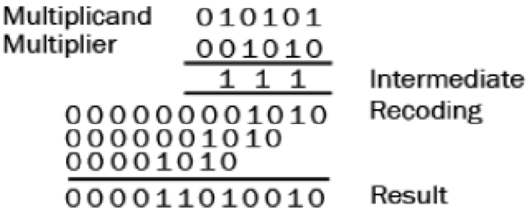


Fig. 5: Booth Multiplication

VI. BOOTH RECODING

The Booth technique has its major advantage if,

- the operands have a large number of bits.
- multiplier contains longsequences of 1's and it has its limitations if the multiplier nd of 1's or even alternating 0–1 pairs.

It can be enhanced by bit-pairing,

- 50% maximum number of summands
- handling 0–1 pairs efficiently

additional overhead for multiplicand

Consider the example of encoding 56:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	1	1	1	0	0	0
0	+1	0	0	−1	0	0	0

$$\begin{aligned} 2^5+2^4+2^3 &= 32+16+8 \\ &= 56 \end{aligned}$$

$$\begin{aligned} 2^6-2^3 &= 64-8 \\ &= 56 \end{aligned}$$

VII. PROPOSED WORK

This proposed work is based on Radix-2 multiplier. It is the simplest and most common form of the Cooley–Tukey algorithm, although highly optimized Cooley–Tukey implementations typically use other forms of the algorithm.Extend Range By Replicating The Sign Bit Of Multiples

- Product Has 2n Bits
- The Multiple $xy_{n-1}2^{n-1}$ Subtracted Instead Of Added

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

Radix-2 Algorithm

Let us consider the computation of the $N = 2^v$ point DFT by the divide-and conquer approach. We split the N-point data sequence into two $N/2$-point data sequences $f_1(n)$ and $f_2(n)$, corresponding to the even-numbered and odd-numbered samples of $x(n)$, respectively, that is,

$$f_1(n) = x(2n)$$
$$f_2(n) = x(2n+1), \qquad n = 0,1,\ldots,\tfrac{N}{2}-1$$

Thus $f_1(n)$ and $f_2(n)$ are obtained by decimating $x(n)$ by a factor of 2, and hence the resulting FFT algorithm is called a *decimation-in-time algorithm*.

Now the N-point DFT can be expressed in terms of the DFT's of the decimated sequences as follows:

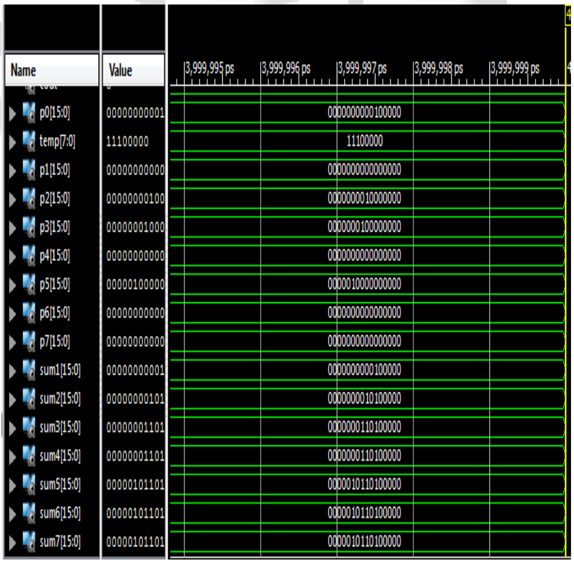
$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad k = 0,1,\ldots,N-1 \\ &= \sum_{n\ \text{even}} x(n) W_N^{kn} + \sum_{n\ \text{odd}} x(n) W_N^{kn} \\ &= \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{k(2m+1)} \end{aligned}$$

But $W_N^2 = W_{N/2}$. With this substitution, the equation can be expressed as

$$\begin{aligned} X(k) &= \sum_{m=0}^{(N/2)-1} f_1(m) W_{N/2}^{km} + W_N^k \sum_{m=0}^{(N/2)-1} f_2(m) W_{N/2}^{km} \\ &= F_1(k) + W_N^k F_2(k), \qquad k = 0,1,\ldots,N-1 \end{aligned}$$

where $F_1(k)$ and $F_2(k)$ are the $N/2$-point DFTs of the sequences $f_1(m)$ and $f_2(m)$, respectively.Since $F_1(k)$ and $F_2(k)$ are periodic, with period $N/2$, we have $F_1(k+N/2) = F_1(k)$ and $F_2(k+N/2) = F_2(k)$. In addition, the factor $W_N^{k+N/2} = -W_N^k$. Hence the computation of $X(k)$ requires $2(N/2)^2 + N/2 = N^2/2 + N/2$ complex multiplications. This first step results in a reduction of the number of multiplications from N^2 to $N^2/2 + N/2$, which is about a factor of 2 for N large.

VIII. SIMULATION RESULTS



Name	Value	
p0[15:0]	00000000001	0000000000100000
temp[7:0]	11100000	11100000
p1[15:0]	00000000000	0000000000000000
p2[15:0]	00000000100	0000000010000000
p3[15:0]	00000001000	0000000100000000
p4[15:0]	00000000000	0000000000000000
p5[15:0]	00000100000	0000100000000000
p6[15:0]	00000000000	0000000000000000
p7[15:0]	00000000000	0000000000000000
sum1[15:0]	00000000001	0000000000100000
sum2[15:0]	00000000101	0000000010100000
sum3[15:0]	00000001101	0000000110100000
sum4[15:0]	00000001101	0000000110100000
sum5[15:0]	00000101101	0000010110100000
sum6[15:0]	00000101101	0000010110100000
sum7[15:0]	00000101101	0000010110100000

Fig. 6: .simulation output

Power consumption of the chip at the target clock frequency(50 MHz) with maximum resolution is 2.4 mW. The power used by the multiply-and-accumulate unit is 1.45 mW when operating as a full precision multiplier. Power reductions in the arithmetic computation block improves at high truncation levels such as 16 bits (0.75 mW). Higher configurations achievable byprogrammable truncation include truncated values higher than16. As an example applying a truncation value of, the power consumption measured in the arithmetical unit of PTMAC chip was 0.55 mW, with power savings achieved reaching 61%.Normalization of silicon power measurements indicate a consumption range of [48 38] uW/MHz for the whole chip, and[24 10] uW/MHz for the arithmetic section, allowing comparison with recent reference publications measured power consumption levels of [36.6 12.2] uW/MHz for a 16-bit multiplierin 0.13 technology in [22], and [71.3 38.6] uW/MHz for a 16-bitmultiplier and a 20-bit accumulator in 0.18 technology in [23]. Both papers study variable compensation implementations on fixed-width multipliers. Results for different post-layout simulations where PTMAC is used to perform several DSP applications are presented in this section. Performing typical filtering tasks where the arithmeticunit is active during half the operations at a clock frequency of 50 MHz indicated a clear dominance of dynamic over static. Simulation results presented in this section are followed by details of measurements performed on the actual PTMAC chip.

IX. Conclusion

In this paper, a new architecture to perform truncated multiplication has been presented and compared against previous implementations. This represents the first DSP system with a truncated multiplier integrated into a programmable DSP block. Software based error compensation offers a practical and effective solution in practice. Programmable truncation has been thoroughly tested within a DSP block suitable for simple but arithmetically intensive algorithms, and its performance is evaluated at a system level. Results show that conservative power savings of up to 17% of the full DSP power can be saved while still maintaining reasonable SNR values. A number of real algorithms implemented on the device show the actual benefits of using truncated multiplication in a practical scenario. The architecture presented has been fabricated on the TSMC 90 nm process and samples have verified correct operation in terms of functionality and power reductions obtainable.

References

[1] M. Schulte And E. E. Swartzlander, Jr., "Truncated Multiplication With Correction Constant [For Dsp]," In Proc. Workshop Vlsi Signal Process. Vi, October. 1993, Pp. 388–396.

[2] S. Kidambi, F. El-Guibaly, And A. Antoniou, "Area-Efficient Multipliers For Digital Signal Processing Applications," Ieee Trans. Circuits Syst. Ii, Analog Digit. Signal Process., Vol. 43, No. 2, Pp. 90–95, Feb. 1996.

[3] E. King And E. E. Swartzlander, Jr., "Data-Dependent Truncation Scheme For Parallel Multipliers," In Conf. Rec. 31st Asilomar Conf. Signals, Syst. Comput.,, Nov. 1997, Vol. 2, Pp. 1178–1182.

[4] J. M. Jou And S. R. Kuang, "Design Of Low-Error Fixed-Width Multiplier For Dsp Applications," Electron. Lett., Vol. 33, No. 19, Pp. 1597–1598,September. 1997.

[5] E. E. Swartzlander, Jr., "Truncated Multiplication With Approximate Rounding," In Conf. Rec. 33rd Asilomar Conf. Signals, Syst., Comput.,1999, Vol. 2, Pp. 1480–1483.

[6] M. Schulte, J. Stine, And J. Jansen, "Reduced Power Dissipation Through Truncated Multiplication," In Proc. Ieee Alessandro Volta Memorial Workshop Low-Power Design, 1999, Pp. 61–69.

[7] L.-D. Van, S.-S. Wang, And W.-S. Feng, "Design Of The Lower Error Fixed-Width Multiplier And Its Application," Ieee Trans. Circuits Syst. Ii, Analog Digit. Signal Process., Vol. 47, No. 10, Pp. 1112–1118, Oct. 2000.

[8] F. Curticapean And J. Niittylahti, "A Hardware Efficient Direct Digital Frequency Synthesizer," In Proc. 8th Ieee Int. Conf. Electron., Circuits, Syst., 2001 (Icecs 2001), Vol. 1, Pp. 51–54.

[9] J. Stine And O. Duverne, "Variations On Truncated Multiplication," In Proc. Euromicro Symp. Digital Syst. Design, 2003, Pp. 112–119.

[10] A. Strollo, N. Petra, And D. Decaro, "Dual-Tree Error Compensation For High Performance Fixed-Width Multipliers," Ieee Trans. Circuits Syst. Ii, Exp. Briefs, Vol. 52, No. 8, Pp. 501–507, August. 2005.

[11] L.-D. Van And C.-C.Yang, "Generalized Low-Error Area-Efficient Fixedwidth Multipliers," Ieee Trans. Circuits Syst. I, Reg. Papers, Vol. 52, No. 8, Pp. 1608–1619, August. 2005.

[12] J.-P.Wang, S.-R. Kuang, And Y.-C. Chuang, "Design Of Reconfigurable Low-Power Pipelined Array Multiplier," In Proc. Int. Conf. Commun.,

[13] Circuits, Syst., 2006, Vol. 4, Pp. 2277–2281. Manuel De La Guia Solaz, Student Member, Ieee, Weihan, Member, Ieee, And Richard Conway, Member, Ieee"A Flexible Low Power Dsp With Truncated Multiplier" Vol. 59, No. 11, November2012.