ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 3, Issue 8, August 2016

Efficient Multiplication for the DSP Applications using Static Segment Method

[1] Swathi Samanthapudi, [2] Murali Krishna D
[1] M.Tech-VLSID [2] Associate Professor
[1][2] SVECW (Autonomous), Bhimavaram, Andhra Pradesh

Abstract: **-- The requirement to hold up different digital signal processing (DSP) and classification applications on energy-constrained devices has regularly developed. This applications usually execute matrix multiplications using fixed-point arithmetic , while indicating tolerance for some counting errors. Hence, improving the energy efficiency of multiplications is critical. In this brief, we introduce multiplier architectures that can tradeoff counting accuracy with energy consumption at design time. Compared with a actual multiplier, the suggested multiplier can consume 57% lower energy/op with average counting error of ~1%. Absolutely, we signify such a little counting error does not particularly failure the effect of DSP and the preciseness of classification applications.**

I. INTRODUCTION

Acquiring high valuable energy has become a major design objective for inserted and mobile counting devices due to their controlled battery sufficiency and power budget. To improve valuable energy of such counting devices, important efforts have already been dedicated at different levels, from software to architecture, and all the way down to circuit and technology levels. Embedded and mobile counting devices are regularly used to maintain some major digital signal processing (DSP) and classification applications. To further develop valuable energy of executing such applications, first, particular purpose processors are generally mingled in counting devices. It has been noted that the use of such particular processors can develop valuable energy by 10–100× compared with general-purpose processors at the same technology generation & voltage.

Second, various DSP and classification applications mostly depends on complicated probabilistic mathematical forms and are framed to process data that normally consists of noise. Thus, for some counting error, they display delicate degeneration in overall DSP quality and classification accuracy on behalf of destructive failure .Such counting error tolerance has been exploited by trading accuracy with energy consumption .Using floating points arithmetic ,initially these algorithms are designed and trained, and then it converts into fixed point arithmetic because the power and area costs for supporting floating points in embedded counting .In spite of this transformation process ,it leads to some loss of counting precision, it does not particularly affect the feature of DSP and the precision of classification applications due to counting error steadiness .Most of such algorithms thoroughly perform matrix multiplication as their fundamental operation, while a multiplier is typically an inherently energy-hungry component.

II. APPROXIMATE MULTIPLIER EXPLOITING SIGNIFICANT SEGMENTS OF OPERANDS

In order to provoke and characterize our proposed multiplier, we specify an *p*-bit segment as *p* continuous bits starting with the leading one in an *q*-bit positive operand. We entitle this dynamic segment method (DSM) in contrast to static segment method (SSM) that will be considered later in this section. With two *p*-bit segments from two *q*-bit operands, we can perform a multiplication using an $p \times p$ multiplier. Fig. 1 shows an example of a multiplication after taking 16-b segments from 32-b operands. In this example, we can achieve 99.3% accuracy for a 32 × 32 multiplication even with an 16 × 16 multiplier.

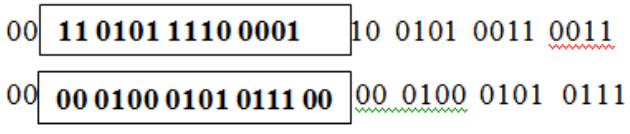

Fig. 1. Example of a multiplication with 16-b segments of two 32-b operands; bold-font bits comprise the segments

Such a multiplication process has little negative concussion on counting efficiency because it can eliminates

ISSN (Online) 2394-6849
International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 3, Issue 8, August 2016

unwanted bits (i.e., sign-extension bits) while satisfying the most useful p significant bits to the multiplier; we will provide detailed evaluations of counting accuracy for various p. Furthermore, an $p \times p$ multiplier consumes much less energy than an $q \times q$ multiplier, because the complexity (and thus energy consumption) of multipliers qradually increases with q. For example, the 4 × 4 and 8 × 8multipliers consume almost 20 × and 5 × less energy than a 16×16 multiplier per operation on average.

However, a DSM requires:

1) Two LODs;

2) two q-bit shifters to align the leading one position of each q-bit operand to the MSB position of each p-bit segment to apply their p-bit segments to the $p \times p$ multiplier; and 3) one $2q$-bit shifter to expand a $2p$-bit result to $2q$ bits.

1)–3) incur considerable area and energy penalties completely negating the energy benefit of using the $p \times p$multiplier; we provide detailed evaluations for two p values

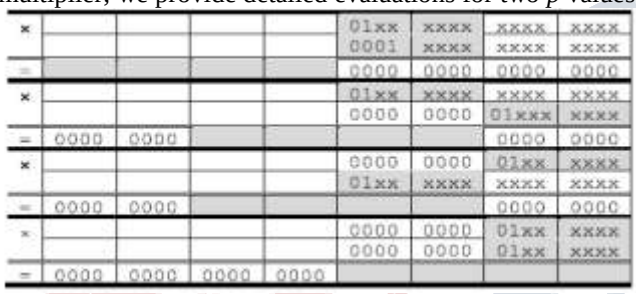

Fig.2. Examples of 16×16 multiplications based on 8-b segments with two possible starting bit positions for 8-b segments. The shaded cells represent 8-b segments and the aligned position of 8×8 multiplication results.

For a multiplication, we choose the p-bit segment that have leading one bit of each operand and apply the chosen segments from both operands to the $p \times p$ multiplier. The SSM greatly simplifies the circuit that chooses p-bit segments and steers them to the $p \times p$ multiplier by replacing two q-bit LODs and shifters for the DSM with two $(q–p)$-input OR gates and q-bit 2-to-1 multiplexers; if the first $(q–p)$ bits starting from the MSB are all zeros, the lower SSM also allows us to replace the $2q$-bit shifter used for the DSM with a $2q$-bit 3-to-1 multiplexer. Since the segment for each operand is taken from one of two possible segments in an q-bit operand, a $2p$-bit result can be expanded to a $2q$bit result by left-shifting the $2p$-bit result by one of three possible shift amounts:

1) no shift when both segments are from the lower p-bit segments;

2) $(q–p)$ shift when two segments are from the upper and lower ones, respectively;

3) $2\times$ $(q–p)$ shift when both segments are from the upper ones, as shown in fig2

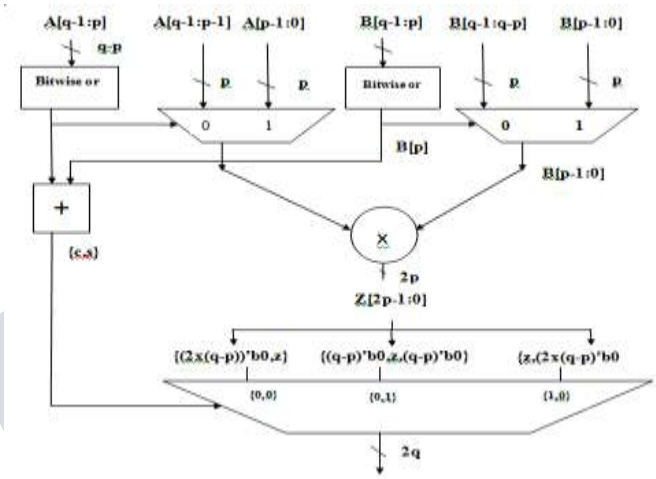

Fig 3 shows the Approximate Multiplier Architecture, the multiplier used in this architecture is Array Multiplier.

III. ARRAY MULTIPLIER

An array multiplier structure is shown in fig 3.for an n-bit array multiplier, to generate partial products it uses n x n array of AND gates, n half adders n x (n-2) full adders . In this multiplier ,number of rows denotes multiplier length and each row width denotes multiplicand width

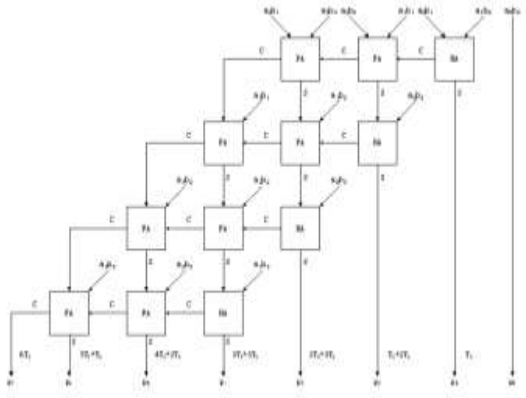

Fig 3 shows 3 bit array multiplier

ISSN (Online) 2394-6849
International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 3, Issue 8, August 2016

For the shifting of partial products ,it does not require any logic for propoer alignment...Array multiplier is well known due to its regular structure. Add and shift algorithm is used in multiplier circuit. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. According to their order the bits are shifted and then added to partial products. normal carry propagate adder can be used to perform addition and it requires N-1 adders & N is the multiplier length. . Each partial product bit is fed into a full adder which sums the partial product bit with the sum from the previous adder and a carry from the less significant previous adder. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. Algorithm:

1. Define variables for multiplier and multiplicand.
2. Use the shift and add method.
3. Cascade 3 blocks of 3x3 bit blocks together to form 9x9 bit multiplier
4. The result of 9 x 9 multiplier will be 18 bits

IV. SIMULATION RESULT

Fig 4 shows the output of approximate multiplier Architecture for different inputs

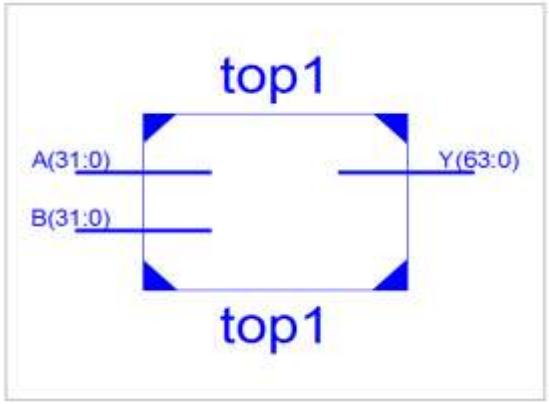

Fig 5 shows the RTL schematic of approximate multiplier Architecture

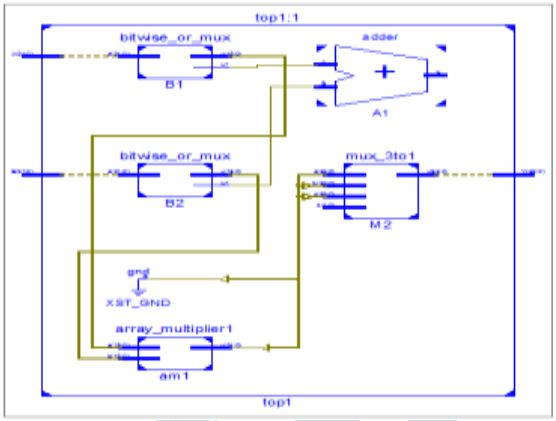

Fig 6 shows the detailed view of Approximate multiplier Architecture

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	339	15,360	2%
Number of occupied Slices	174	7,680	2%
Number of Slices containing only related logic	174	174	100%
Number of Slices containing unrelated logic	0	174	0%
Total Number of 4 input LUTs	339	15,360	2%
Number of bonded IOBs	128	333	38%
Average Fanout of Non-Clock Nets	3.12		

Table 1 shows the design summary of Approximate multiplier Architecture

V. CONCLUSION

In this brief, we propose an approximate multiplier that cans tradeoff accuracy and energy/op at design time for DSP and recognition applications. Our proposed approximate multiplier takes *p* consecutive bits (i.e., an *p*-bit segment) of an *q*-bit operand either starting from the MSB or ending at the LSB and apply two multiplier. Compared with an approach that identifies the exact leading one positions of two operands and applies two *p*-bit segments starting from the leading one positions (i.e., DSM), ours consumes much less energy and area than DSM. This improved energy and area efficiency comes at the cost of slightly lower compute accuracy However, we demonstrate that the loss of small compute accuracy using SSM does not notably impact QoC of applications we evaluated.

ISSN (Online) 2394-6849
International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 3, Issue 8, August 2016

REFERENCES

[1] Srinivasan Narayanamoorthy, Hadi Asghari Moghaddam, Zhenhong Liu, Taejoon Park, and Nam Sung Kim, July 2014, "Energy-Efficient Approximate Multiplication for Digital Signal Processing and Classification Applications", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Early Access Articles, pp. 1 – 5.

[2] R. K. Krishnamurthy and H. Kaul, "Ultra-low voltage technologies for energy-efficient special-purpose hardware accelerators," Intel Technol. *J.*, vol. 13, no. 4, pp. 102–117, 2009.

[3]R. Hegde and N. R. Shanbhag, "Energy-efficient signal processing viaalgorithmic noise-tolerance," in Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED*)*, Aug. 1999, pp. 30–35.

[4] D. Menard, D. Chillet, C. Charot, and O. Sentieys, "Automatic floatingpoint to fixed-point conversion for DSP code generation," in Proc. ACM Int. Conf. Compilers, Archit., Syn. Embedded Syst. *(CASES)*, 2002,pp. 270–276.

[5] V. K. Chippa, D. Mohapatra, A. Raghunathan, K. Roy, and S. T. Chakradhar, "Scalable effort hardware design: Exploiting algorithmic resilience for energy efficiency," in Proc. 47th IEEE/ACM Design Autom. Conf., Jun. 2010,pp. 555–560.

[6] D. Mohapatra, G. Karakonstantis, and K. Roy, "Significance driven computation: A voltage-scalable, variation-aware, quality-tuning motion estimator," in Proc. 14th IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED*)*, Aug. 2009, pp. 195–200.

[6] C. H. Chang and R. K. Satzoda, "A low error and high performance multiplexer-based truncated multiplier," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 12, pp. 1767–1771, Dec. 2010.