Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

New method for high performance multiply-accumulator design*

Bing-jie XIA†, Peng LIU†‡, Qing-dong YAO

(*Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China*)

†E-mail: icysummer@zju.edu.cn; liupeng@isee.zju.edu.cn

Received July 27, 2008; Revision accepted Oct. 28, 2008; Crosschecked Apr. 27, 2009

Abstract: This study presents a new method of 4-pipelined high-performance split multiply-accumulator (MAC) architecture, which is capable of supporting multiple precisions developed for media processors. To speed up the design further, a novel partial product compression circuit based on interleaved adders and a modified hybrid partial product reduction tree (PPRT) scheme are proposed. The MAC can perform 1-way 32-bit, 4-way 16-bit signed/unsigned multiply or multiply-accumulate operations and 2-way parallel multiply add (PMADD) operations at a high frequency of 1.25 GHz under worst-case conditions and 1.67 GHz under typical-case conditions, respectively. Compared with the MAC in 32-bit microprocessor without interlocked piped stages (MIPS), the proposed design shows a great advantage in speed. Moreover, an improvement of up to 32% in throughput is achieved. The MAC design has been fabricated with Taiwan Semiconductor Manufacturing Company (TSMC) 90-nm CMOS standard cell technology and has passed a functional test.

Key words: Multiply-accumulator (MAC), Pipeline, Compressor, Partial product reduction tree (PPRT), Split structure
doi:10.1631/jzus.A0820566 **Document code:** A **CLC number:** TP332

INTRODUCTION

Multiply-accumulate operation is one of the basic arithmetic operations extensively used in modern digital signal processing (DSP). Most arithmetic, such as digital filtering, convolution and fast Fourier transform (FFT), requires high-performance multiply-accumulate operations. The multiply-accumulator (MAC) unit always lies in the critical path that determines the speed of the overall hardware systems. Therefore, a high-speed MAC that is capable of supporting multiple precisions and parallel operations is highly desirable.

The existing MAC implementation methods in the literature can be generally classified into three categories. The first category is the recursive MAC method (Clark *et al*., 2001; Liao and Roberts, 2002), which builds wider vector elements out of several narrower ones and then adds the multiple results together. It is achieved iteratively by recalculating the data back through the unit over more than one cycle. This method saves hardware resource but requires several clock cycles per operation. The second category involves the parallel MAC method (Perri *et al*., 2005; Parandeh-Afshar *et al*., 2006; MIPS Technologies Inc., 2006; 2007) implemented by unrolling the iterative loop of recursive MAC method, which achieves high speed at the cost of hardware resources. This method has been applied in most field programmable gate array (FPGA) chips of the Xilinx Corporation. The last category is the shared-segmentation vector MAC method (Tan *et al*., 2003; Danysh and Tan, 2005; Wang *et al*., 2008), which shares partial product reduction tree (PPRT) and final carry-propagate adder (CPA) between different precision modes by inserting mode-dependent logics. This method is capable of supporting multiple precision operations. However, the single shared output register structure limits the throughput and the complex PPRT structure results in lower speed.

In this study, we propose a high-speed split MAC that takes advantage of parallel and shared-segmen-

‡ Corresponding author

* Project (No. 60873112) supported by the National Natural Science Foundation of China

tation vector MAC approaches for high-performance systems. The main features include:

(1) A new split 32-bit MAC architecture built of parallel sub MAC units is proposed. It inherits high parallelism from the parallel MAC approach to realize high speed and high throughput. In addition, the proposed MAC learns from the shared-segmentation vector method to provide multiple operation modes, which enhances MAC with the feature of powerful calculation capabilities. A high frequency of 1.25 GHz under worst-case conditions and 1.67 GHz under typical-case conditions is achieved. Compared to the MIPS 32-bit processors (MIPS Technologies Inc., 2006; 2007) and other published MAC designs (Liao and Roberts, 2002; Danysh and Tan, 2005), the proposed MAC design shows a great advantage in speed and the overall throughput increases by up to 32%.

(2) A novel compression circuit based on interleaved adders (IA), referred to as an IA-based compressor, is presented. Moreover, a modified hybrid PPRT scheme that makes full use of various compressors is proposed to speed up the design. Compared to the popular fast compression methods (Wallace, 1964; Kwon *et al*., 2000; Chang *et al*., 2005; Chen and Chu, 2007), the proposed hybrid PPRT provides an improvement of up to 14.0% in speed and a reduction of up to 34.8% in area.

ARCHITECTURE OF THE MAC

A high-performance MAC unit, especially with powerful calculation capabilities and high throughput features, is considered to be a great competitive candidate for multimedia applications. Such a MAC unit is implemented in this research by adopting a split structure based on four parallel 16-bit sub MAC units.

Algorithm

Due to the parallel structure, the MAC can be easily implemented to support multiple precisions by inserting very few mode control logic gates. The proposed MAC provides three operation modes in terms of functions and precisions: unified mode, split mode, and parallel multiply add (PMADD) mode.

(1) Unified mode

Under unified mode, four sub MAC units work together for one multiply or multiply-accumulate operation. Let A and B be n-bit multiplicand and multiplier, respectively:

$$A = A_{\mathrm{H}} \times 2^{n/2} + A_{\mathrm{L}}, \tag{1}$$

$$B = B_{\mathrm{H}} \times 2^{n/2} + B_{\mathrm{L}}, \tag{2}$$

where A_{H} is the upper $n/2$ bits of A, and A_{L} is the lower $n/2$ bits of A. It is similar for B_{H} and B_{L}.

The product:

$$\begin{aligned} UNIFIED\_OP &= A \times B \\ &= (A_{\mathrm{H}} \times B_{\mathrm{H}}) \times 2^{n} + (A_{\mathrm{H}} \times B_{\mathrm{L}}) \times 2^{n/2} \\ &\quad + (A_{\mathrm{L}} \times B_{\mathrm{H}}) \times 2^{n/2} + (A_{\mathrm{L}} \times B_{\mathrm{L}}). \end{aligned} \tag{3}$$

Eq.(3) illustrates that an n-bit multiply consists of four $n/2$-bit multiplies and a summation of these n-bit results after shifting. The result of $(A_{\mathrm{H}}\times B_{\mathrm{H}})\times 2^{n}+(A_{\mathrm{L}}\times B_{\mathrm{L}})$ can be realized by concatenating these two n-bit sub products simply since there is no overlap. There are actually three numbers to be summed.

(2) Split mode

Under split mode, four sub MAC units are split to work for four independent operations in parallel. Let A, B, C and D be four n-bit operands, each of them treated as a pair of individual $n/2$-bit values. Take A for example: A_{H} and A_{L} are considered as two independent $n/2$-bit values, and similarly for B, C and D.

The operations are:

$$SPLIT\_OP_0 = A_{\mathrm{H}} \times B_{\mathrm{H}}, \tag{4}$$

$$SPLIT\_OP_1 = A_{\mathrm{L}} \times B_{\mathrm{L}}, \tag{5}$$

$$SPLIT\_OP_2 = C_{\mathrm{H}} \times D_{\mathrm{H}}, \tag{6}$$

$$SPLIT\_OP_3 = C_{\mathrm{L}} \times D_{\mathrm{L}}. \tag{7}$$

Therefore, the MAC can perform 4-way independent split $n/2$-bit multiply or multiply-accumulate operations as given in Eqs.(4)~(7), which are introduced to increase the data processing capability.

(3) PMADD mode

Under PMADD mode, two results of adjacent split multiply operations are added together. This mode is designed to handle media streams more efficiently.

The operations are:

$$PMADD\_OP_0 = A_H \times B_H + A_L \times B_L, \quad (8)$$

$$PMADD\_OP_1 = C_H \times D_H + C_L \times D_L. \quad (9)$$

Two n-bit adders are applied to perform the PMADD operations in Eqs.(8) and (9).

Structure of the MAC

The structure of split MAC, which implements the above algorithms, is illustrated in Fig.1. For unified mode, two 32-bit operands A and B are stored in R_{00} and R_{01}, respectively, while for split and PMADD modes two more 32-bit operands are stored in R_{02} and R_{03}. In addition, three control signals A_mul, G_mul, and U_mul are designed for setting the operation mode. The accumulate control signal A_mul is used to select whether the design performs MAC or multiply operation. The granularity control signal G_mul selects whether unified mode or split/PMADD mode is to operate. The sign control signal U_mul selects whether unsigned or signed operation is valid. The whole design is arranged into a 4-pipelined structure.

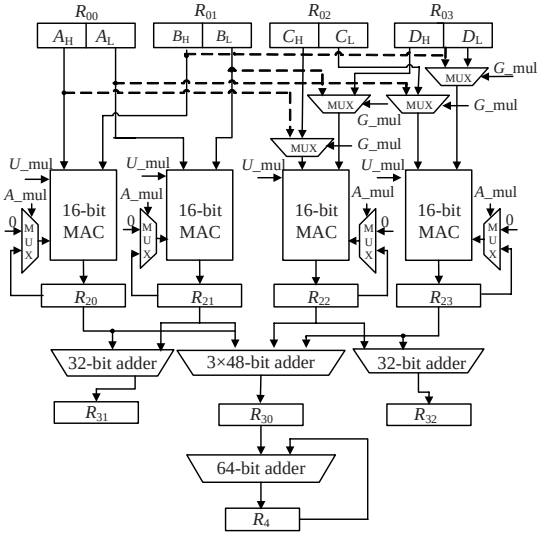


Fig.1 Block diagram of the proposed MAC

Pipestage 1 and Pipestage 2 Generate products of the four 16-bit multiply or MAC operations.

The first two pipestages work together as four 16-bit multipliers or MACs, which will be introduced in detail in Section 4. In front of these 16-bit MACs, four multiplexers (MUXs) are used to select proper operands according to the value of G_mul. When G_mul is asserted, the inputs of four multipliers are shown in Eq.(10) and the MAC works in the unified mode. When G_mul is deserted, the inputs are described in Eq.(11) and the MAC works in the split or PMADD mode. The selection between multiply or MAC operation is controlled by A_mul. When A_mul is asserted, the accumulated values are fed back to the 16-bit MACs, whereas zeros are selected when A_mul=0. The results of these four 16-bit MACs are stored in registers R_{20}, R_{21}, R_{22}, R_{23}, respectively.

Unified mode:

$$\begin{cases} R_{20} = A_H \times B_H, R_{21} = A_L \times B_L, \\ R_{22} = A_H \times B_L, R_{23} = A_L \times B_H; \end{cases} \quad (10)$$

Split mode/PMADD mode:

$$\begin{cases} R_{20} = A_H \times B_H, R_{21} = A_L \times B_L, \\ R_{22} = C_H \times D_H, R_{23} = C_L \times D_L. \end{cases} \quad (11)$$

Pipestage 3 Forms PMADD and 32-bit multiply results.

The third pipestage consists of two-part logics. One part involves two 32-bit adders with the aim of calculating two 16-bit PMADD operations. The two results are stored in R_{31} and R_{32} as shown in Eqs.(13) and (14). The other part involves a 3-operand adder to work out the result of 32-bit multiply operation as illustrated in Eq.(12). As the lower 16 bits of R_{30} are equal to the lower 16 bits of R_{21}, a 3-operand 48-bit adder is sufficient.

$$R_{30} = \{R_{20}, R_{21}\} + (sign\_extension\{R_{22}\} + sign\_extension\{R_{23}\}) \times 2^{16}, \quad (12)$$

$$R_{31} = R_{20} + R_{21}, \quad (13)$$

$$R_{32} = R_{22} + R_{23}, \quad (14)$$

$$R_4 = R_{30} + R_4\_\mathrm{acc}, \quad (15)$$

where R_4_acc denotes the value stored in R_4 to be accumulated.

Pipestage 4 Generates the result of 32-bit multiply-accumulate operation.

The last pipestage performs a 64-bit add operation as illustrated in Eq.(15). The 64-bit adder adopts a carry-select adder (Kim and Kim, 2001) structure built of 4-bit carry look-ahead adder (Pai and Chen, 2004) blocks.

In conclusion, 4-way parallel 16-bit signed/unsigned multiply or MAC results can be output

through R_{20}, R_{21}, R_{22} and R_{23} at the second pipestage for split mode. Two-way 16-bit PMADD results can be output through R_{31}, R_{32} and 1-way 32-bit multiply result can be output through R_{30} at the third pipestage. For unified mode, 1-way 32-bit MAC result can be output through R_4 at the fourth pipestage. Different operations can be processed simultaneously in the MAC, which gives it the advantage of powerful data processing capabilities.

HYBRID PARTIAL PRODUCT REDUCTION TREE

MAC unit generally consists of three parts: a partial product generator (PPG), a PPRT and a final CPA (Liao and Roberts, 2002; Danysh and Tan, 2005; Wang *et al*., 2008). The PPRT, which is always the bottleneck in speeding up multiply and MAC operations, has attracted much more attention from designers. Several popular compression schemes, including the Wallace tree (Wallace, 1964; Chang *et al*., 2005), the 4:2 compressor (Chang *et al*., 2004) and the 5:2 compressor (Chang *et al*., 2004; Kwon *et al*., 2000), have been exploited to solve this problem. This study proposes a novel compression algorithm based on interleaved adders, referred to as an IA-based compressor, which is more efficient than existing fast compressors in some situations. A hybrid PPRT scheme making full use of existing fast compressors and the proposed IA-based compressor, is developed to speed up the PPRT.

Novel IA-based compressor

Fig.2 shows a conventional Wallace PPRT (Wallace, 1964; Chang *et al*., 2005) based on full adders with five partial products A, B, C, D and E. It is found that the function of two full adders (such as FA00 and FA11, FA01 and FA12) is actually the same as that of a 2-bit adder. Hence, a novel compression algorithm that uses a stage of interleaved adders to replace the first two stages of full adders is proposed, as shown in Fig.3. This stage of adders is arranged into two interleaved groups that are aligned with one bit shifted. The partial products A and B are added as the two operands of the first group of adders, and D and E are added by the second group of adders. The last partial product C is added as the carry-in bits of these two groups of adders. In this way, five partial products are reduced to three: a, b and c, where a and c are composed of sum bits of these two groups of adders respectively, and b is composed of all the carry-out bits. Taking a 32-bit PPRT as an example, the performance comparisons are given in Table 1. The circuit is synthesized in TSMC 90-nm standard cell technology with Physical Compiler of Synopsys. Compared to the fast 5:2 compressor (Kwon *et al*., 2000; Chang *et al*., 2004), the proposed IA-based compressor shows a 10.7% improvement in speed and a 24.0% reduction in area. Thus, a more efficient IA-based 5:2 compressor of great importance in constructing PPRT is developed.

To generalize the IA-based compression algorithm to other cases, assume the number of partial products (PPs) is M.

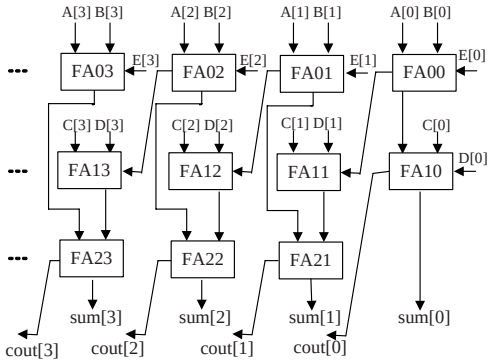


Fig.2 Conventional Wallace PPRT. FA: full adder

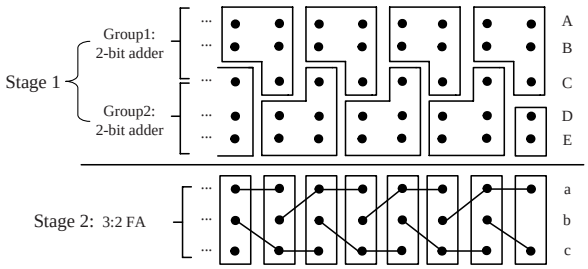


Fig.3 Proposed IA-based 5:2 compressor

Table 1 Comparisons of speed and area performance

	Delay (ns)	Gate count
Proposed	0.25	1470
Wallace tree	0.33	2035
5:2 compressor	0.28	1933

Note: synthesized in TSMC 90-nm CMOS standard cell technology under worst-case process

When M is an odd number, a stage of adders with the bit width of $(M-1)/2$ is arranged into $(M-1)/2$ interleaved groups. Two adjacent groups are aligned with one bit shifted. Thus, the number of PPs can be reduced from M to $(M+1)/2$.

When M is an even number, a stage of adders with the bit width of $M/2$ is arranged into $M/2$ groups. The number of PPs can be reduced from M to $(M+2)/2$.

Obviously, this algorithm works more efficiently when M is odd. It could use fast adders to boost speed. It realizes a good combination of carry-propagate circuit and carry-save circuit, which performs compression in both vertical and horizontal directions. Better area efficiency is achieved.

Hybrid PPRT scheme

Most MAC designs (Abdelgawad and Bayoumi, 2007; Parandeh-Afshar *et al*., 2006; Sundeepkumar *et al*., 2008) adopt a simplified PPRT structure that is composed of a single type of compressor to achieve better regularity. To overcome the limitations resulting from simplified PPRT and to achieve better compression efficiency, a hybrid PPRT (Oklobdzija and Villeger, 1995; Chong *et al*., 2007) is introduced which uses different types of compressors. This study presents a modified hybrid PPRT scheme that uses both existing efficient compressors and the proposed IA-based compressor to achieve a higher speed and a smaller area.

The proposed IA-based compressor shows its superiority in some situations. However, the delay and area of an adder grow rapidly with an increase in width. It works well when the width of interleaved adder is small, such as 2 bits, 3 bits and 4 bits. Therefore, we use only these three types of adders in our hybrid PPRT. When the PP count is larger than 10, parallel compressors composed of adders with small width, such as efficient IA-based 5:2 compressors, are employed to implement compression. The modified hybrid PPRT scheme is constructed as follows.

When M is odd and smaller than 10, a stage of IA-based compressors is applied to reduce the number of PPs from M to $(M+1)/2$.

When M is even and smaller than 10:

If M is a multiple of 4, a stage of 4:2 compressors is applied to reduce the PP count from M to $M/2$;

If M is a multiple of 3, a stage of full adders is applied to reduce the PP count from M to $2M/3$;

Otherwise, a stage of IA-based compressors is applied.

When M is equal or larger than 10, two or more parallel groups of IA-based 5:2 compressors are applied to reduce the PP count.

After the first stage compression is completed, the next stage proceeds in the same way until the PP count has been reduced to two. The modified hybrid PPRT scheme combines each compressor's advantages, resulting in a much shorter delay and smaller area. The critical path analysis of the proposed hybrid PPRT and existing fast PPRT scheme without IA-based compressors are illustrated in Table 2. The adders in the proposed IA-based compressors can be constructed according to the design objective. For an area-driven design, an area-efficient adder should be employed, such as ripple carry adder (Fang *et al*., 2002; Sundeepkumar *et al*., 2008), while for a timing-driven design, a fast adder should be used to boost speed. In this design, we suggest a fast carry look-ahead adder (Rabaey, 2002; Pai and Chen, 2004) as the basic component to construct the IA-based compressor. A comparison of the results of a 32-bit PPRT structure based on the proposed PPRT scheme and conventional fast PPRT after physical synthesis is given in Table 3. The tool used was Physical Compiler of Synopsys. Up to 14.0% improvement in speed and up to 34.8% reduction in area have been achieved. The proposed hybrid PPRT shows an obvious superiority for implementing a high-performance MAC.

Table 2 Critical path analysis of two PPRT schemes

PP count	PPRT without IA-based	Proposed hybrid PPRT
5	5:2	IA-based 5:2
6	3:2 and 4:2	3:2 and 4:2
7	4:2 and 4:2	3-bit adder and 4:2
8	4:2 and 4:2	4:2 and 4:2
9	5:2 and 4:2	4-bit adder and IA-based 5:2
10	5:2 and 4:2	IA-based 5:2 and 4:2
11	5:2 and 5:2	IA-based 5:2 and IA-based 5:2
12	4:2, 3:2 and 4:2	IA-based 5:2, 3:2 and 4:2
13	5:2, 3:2 and 4:2	IA-based 5:2, 3:2 and 4:2

Note: 5:2 is short for 5:2 compressor, and similarly for 4:2 and 3:2

Table 3 Comparisons of speed and area performance between the conventional and the proposed PPRT schemes

PP count	Timing (ns)		Improvement (%)
	Conventional PPRT	Proposed PPRT	
5	0.28	0.25	10.7
6	0.34	0.34	0.0
7	0.43	0.37	14.0
8	0.45	0.45	0.0
9	0.50	0.44	12.0
10	0.51	0.46	9.8
11	0.58	0.52	10.3
12	0.59	0.59	0.0
13	0.65	0.60	7.7
PP count	Gate count		Improvement (%)
	Conventional PPRT	Proposed PPRT	
5	1933	1470	24.0
6	2645	2645	0.0
7	3259	2363	27.5
8	3875	3875	0.0
9	4511	2939	34.8
10	5256	4229	19.6
11	5800	4410	24.0
12	6712	5585	16.8
13	7489	6263	16.4

Note: synthesized in TSMC 90-nm CMOS standard cell technology under worst-case process

16-BIT MAC STRUCTURE

The 16-bit MAC unit occupies the first two pipestages introduced in Section 2, and forms the basic building block of high-performance split MAC. The 16-bit MAC is composed of PPG, PPRT and a final CPA (Fig.4). The PPRT lies in the middle of the critical path and consumes most time and area, thus the 16-bit MAC is divided into two pipestages on the PPRT. Accumulate value is sent to PPRT at the second pipestage.

The radix-4 Booth encoding algorithm (Elguibaly, 2000; Chong *et al*., 2007) is applied in PPG, which reduces the number of PPs by half. Since the design can perform both signed and unsigned operations, there are nine PPs generated from the PPG. In PPRT, the suggested modified hybrid scheme is adopted. A stage of 4-bit interleaved adders is employed first, to reduce nine PPs to five. Then a second stage of 2-bit adders is employed to reduce five PPs to three. These logics, along with PPG, are arranged in the first pipestage.

In the second pipestage, there are four PPs to be reduced, including three from the first pipestage and an additional accumulate value. Therefore, a stage of 4:2 compressors is sufficient. These 4:2 compressors and a final 32-bit CPA constitute the second pipestage.

The delay of the two pipestages is equal to 0.60 and 0.69 ns, respectively (Fig.4). Since there is still a stage of MUXs in front of the first pipestage (Fig.1), these two pipestages are partitioned evenly which indicates an efficient pipeline scheme.

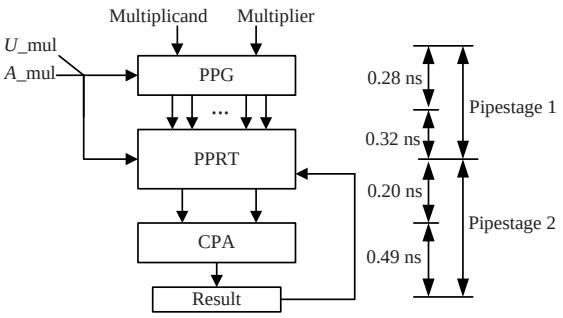


Fig.4 Block diagram of 16-bit sub MAC

PERFORMANCE OF THE PROPOSED MAC

Many aspects of this new high-performance MAC design have been evaluated in detail, including front-end functional tests, back-end synthesis and layout. Table 4 gives the delay information under worst-case and typical-case conditions respectively, after placement and routing in TSMC 90-nm CMOS standard cell technology. The proposed MAC design can achieve a high frequency of 1.25 GHz under worst-case conditions and 1.67 GHz under typical-case conditions, respectively. The speed results compared with those from the previous MAC designs (Liao and Roberts, 2002; Danysh and Tan, 2005; MIPS Technologies Inc., 2006; 2007) are shown in Table 5. The proposed MAC shows a great advantage in implementing high-speed systems.

Besides the high frequency property, the proposed MAC has the merit of high data throughput. It adds four parallel 16-bit multiply/MAC and two PMADD operation capabilities to MAC. The throughput comparison results for 32-bit operations are shown in Table 6. An improvement of up to 32% is achieved for 32-bit multiply operation because 32-bit

multiply operation in the proposed MAC unit is generated one cycle in advance of those in MIPS 32-bit cores.

Table 4 Delay and frequency after placement and routing

MAC	Delay (ns)					Frequency (GHz)
	PS 1	PS 2	PS 3	PS 4	Total	
Worst-case	0.78	0.79	0.79	0.78	0.79	1.25
Typical-case	0.59	0.60	0.60	0.60	0.60	1.67

PS: pipestage

Table 5 Speed comparisons of different MAC designs

MAC	Frequency (GHz)	Process
Liao and Roberts (2002)	0.60	SMIC 180 nm
Danysh and Tan (2005)	0.93	SOI 90 nm
MIPS32 34K (2006)	0.775	TSMC 65 nm
MIPS32 74K (2007)	1.10	TSMC 65 nm
Proposed MAC	1.25	TSMC 90 nm

SOI: silicon on insulator

Table 6 Throughput comparisons for 32 bit operations

MAC	32-bit (Multiply/cycle)	32-bit (MAC/cycle)
Liao and Roberts (2002)	0.25 to 1	0.25 to 1
Danysh and Tan (2005)	N/A	N/A
MIPS32 34K (2006)	0.25 to 1	0.25 to 1
MIPS32 74K (2007)	0.25 to 1	0.20 to 1
Proposed MAC	0.33 to 1	0.25 to 1

N/A: not applicable

To evaluate the proposed MAC further, we select three test cases of signal processing systems to verify the function and investigate the performance improvement. These three cases involve dot product of two 16D vectors, 1-point 50-order finite impulse response (FIR) filter and 1-point 2-order infinite impulse response (IIR) filter. The performance comparisons by Modelsim v6.0 are shown in Fig.5. The performance of the proposed MAC that supports multiple parallel operations and PMADD operation is almost 2~4 times better than the performance of traditional MAC without such features (Wang *et al.*, 2008) at the same frequency. The proposed MAC has been embedded in our media DSP and the floorplan of the DSP with MAC highlighted is shown in Fig.6a. Fig.6b provides the floorplan of the MAC design, which is implemented in TSMC 90-nm CMOS process by Astro of Synopsys. The corresponding characteristics of the MAC are listed in Table 7.

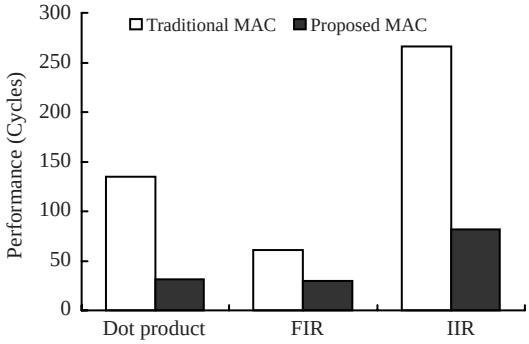
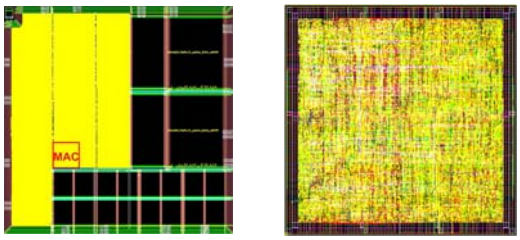


Fig.5 Performance comparisons for signal processing test cases



(a) (b)

Fig.6 Floorplan of media DSP with MAC highlighted (a) and floorplan of MAC (b)

Table 7 Characteristics of the proposed MAC

Parameter	Value
Gate count	32 963
Power consumption (μW/MHz)	99.63
Power supply (V)	1.20
IR drop (mV)	36.60
Size (μm×μm)	394×393

CONCLUSION

This study presents a high-performance MAC architecture. The merits of high throughput and powerful data processing capabilities are realized with a fully pipelined, split structure based on four parallel 16-bit sub MAC units. The high-speed feature is enhanced by proposing a novel IA-based compression algorithm and a modified hybrid PPRT scheme. The MAC can perform one 32-bit, four 16-bit signed/unsigned multiply or MAC and two PMADD operations at a high frequency of 1.25 GHz under

worst-case conditions and 1.67 GHz under typical-case conditions, respectively. The design has been implemented in TSMC 90-nm CMOS standard cell technology and has passed a functional test.

ACKNOWLEDGEMENT

The authors would like to acknowledge all of our colleagues in the ISEE R&D Group of Zhejiang University, China, and many others in Hangzhou Silan Micro-electronic Joint-stock Co. Ltd who have contributed to the design.

References

Abdelgawad, A., Bayoumi, M., 2007. High Speed and Area-efficient Multiply Accumulate (MAC) Unit for Digital Signal Processing Applications. Proc. IEEE Int. Symp. on Circuits and Systems. New Orleans, USA, p.3199-3202. [doi:10.1109/ISCAS.2007.378152]

Chang, C.H., Gu, J.M., Zhang, M.Y., 2004. Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits. *IEEE Trans. Circuits Syst. I: Fundam. Theory Appl.*, **51**(10):1985-1997. [doi:10.1109/TCSI.2004.835683]

Chang, C.H., Gu, J.M., Zhang, M.Y., 2005. A review of 0.18-μm full adder performances for tree structured arithmetic circuits. *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, **13**(6):686-695. [doi:10.1109/TVLSI.2005.848806]

Chen, K.H., Chu, Y.S., 2007. A low-power multiplier with the spurious power suppression technique. *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, **15**(7):846-850. [doi:10.1109/TVLSI.2007.899242]

Chong, K.S., Gwee, B.H., Chang, J.S., 2007. Low energy 16-bit Booth leapfrog array multiplier using dynamic adders. *IET Proc. Circ., Devices & Syst.*, **1**(2):170-174. [doi:10.1049/iet-cds:20060109]

Clark, L., Hoffman, E.J., Miller, J., Biyani, M., Liao, L.Y., Strazdus, S., Morrow, M., Velarde, K.E., Yarch, M.A., 2001. An embedded 32-b microprocessor core for low-power and high-performance applications. *IEEE J. Solid-State Circ.*, **36**(11):1599-1608. [doi:10.1109/4.962279]

Danysh, A., Tan, D., 2005. Architecture and implementation of a vector/SIMD multiply-accumulate unit. *IEEE Trans. Comput.*, **54**(3):284-293. [doi:10.1109/TC.2005.41]

Elguibaly, F., 2000. A fast parallel multiplier-accumulator using the modified Booth algorithm. *IEEE Trans. Circuits Syst. II: Analog Digital Sign. Process.*, **47**(9):902-908. [doi:10.1109/82.868458]

Fang, C.J., Huang, C.H., Wang, J.S., Yeh, C.W., 2002. Fast and Compact Dynamic Ripple Carry Adder Design. Proc. IEEE Asia-Pacific Conf. on ASIC. Taipei, Taiwan, p.25-28. [doi:10.1109/APASIC.2002.1031523]

Kim, Y., Kim, L., 2001. 64-bit carry-select adder with reduced area. *Electr. Lett.*, **37**(10):614-615. [doi:10.1049/el:20010430]

Kwon, O., Nowka, K., Swartzlander, E.E., 2000. A 16-bit×16-bit MAC Design Using Fast 5:2 Compressors. Proc. IEEE Int. Conf. on Application-specific Systems, Architectures, and Processors. Boston, USA, p.235-243. [doi:10.1109/ASAP.2000.862394]

Liao, Y., Roberts, D., 2002. A high-performance and low-power 32-bit multiply-accumulate unit with single-instruction-multiple-data (SIMD) feature. *IEEE J. Solid-State Circ.*, **37**(7):926-931. [doi:10.1109/JSSC.2002.1015692]

MIPS Technologies, Inc., 2006. MIPS32 34K^{TM} Processor Core Family Software User's Manual. p.29-52.

MIPS Technologies, Inc., 2007. MIPS32 74K^{TM} Processor Core Family Software User's Manual. p.29-40.

Oklobdzija, V., Villeger, D., 1995. Improving multiplier design by using improved column compression tree and optimized final adder in CMOS technology. *IEEE Trans. Very Large Scale Integration (VLSI) Syst.*, **3**(2):292-301. [doi:10.1109/92.386228]

Pai, Y.T., Chen, Y.K., 2004. The Fastest Carry Lookahead Adder. IEEE Int. Workshop on Electronic Design, Test and Applications, Perth, Australia, p.434-436. [doi:10.1109/DELTA.2004.10071]

Parandeh-Afshar, H., Ahmadvand, M., Safari, S., 2006. A Novel Merged Multiplier-accumulator Embedded in DSP Coprocessor. Proc. IEEE Int. Conf. on Electronics, Circuits and Systems, Nice, France, p.119-122. [doi:10.1109/ICECS.2006.379734]

Perri, S., Corsonello, P., Cocorullo, G., 2005. Efficient recursive multiply architecture for FPGAs. *Electr. Lett.*, **41**(24):1314. [doi:10.1049/el:20053313]

Rabaey, J.M., 2002. Digital Integrated Circuits—A Design Perspective. Prentice-Hall International Publisher, New Jersey, USA, p.564-586.

Sundeepkumar, A., Pavankumar, V., Yorkesh, R., 2008. Energy Efficient, High Performance Circuits for Arithmetic Units. Proc. Int. Conf. on VLSI Design, Bangalore, India, p.371-376. [doi:10.1109/VLSI.2008.49]

Tan, D., Danysh, A., Liebelt, M., 2003. Multiple-precision Fixed-point Vector Multiply-accumulator Using Shared Segmentation. Proc. IEEE Symp. on Computer Arithmetic, Santiago de. Compostela, Spain, p.12-19. [doi:10.1109/ARITH.2003.1207655]

Wallace, C.S., 1964. A suggestion for a fast multiplier. *IEEE Trans. Electr. Comput.*, **13**(1):14-17. [doi:10.1109/PGEC.1964.263829]

Wang, L.R., Jou, S.J., Lee, C.L., 2008. A Well-structured Modified Booth Multiplier Design. IEEE Int. Symp. on VLSI Design, Automation and Test, Hsinchu, Taiwan, p.85-88. [doi:10.1109/VDAT.2008.4542418]