ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.15, No.7, 1–12

A high-speed realization of the delayed dual sign LMS algorithm

Mingjiang Wang[a)], **Boya Zhao**[b)], **and Ming Liu**

Harbin Institute of Technology (Shenzhen),

Xili University Town, Shenzhen 518055, Guangdong, China

a) *mjwang@hit.edu.cn*

b) *zhaoboya@stu.hit.edu.cn, Corresponding Author*

Abstract: Motivated by reduction of computational complexity and improvement of convergence speed, this work develops a high-speed VLSI realization of the adaptive FIR filter based on the delayed dual sign LMS (DDSLMS) algorithm. Not only is the computation of the proposed algorithm the same as that of the sign LMS algorithm, but also the convergence characteristic is close to that of the LMS algorithm. The fine-grained dot-product unit and multiple-input-addition unit are adopted to reduce the latency of critical path. From the ASIC synthesis results we find that the proposed architecture of an 8-tap filter has nearly 38% less power and nearly 43% less area-delay-product (ADP) than the best existing structure.

Keywords: delayed dual sign LMS algorithm, fined-grained, dot-product unit, multiple-input-addition unit

Classification: Integrated circuits

References

[1] P. K. Meher and S. Y. Park: "Critical-path analysis and low-complexity implementation of the LMS adaptive algorithm," IEEE Trans. Circuits Syst. I, Reg. Papers **61** (2014) 778 (DOI: 10.1109/TCSI.2013.2284173).

[2] M. D. Meyer and D. P. Agrawal: "A high sampling rate delayed LMS filter architecture," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. **40** (1993) 727 (DOI: 10.1109/82.251841).

[3] S. C. Douglas, *et al.*: "A pipelined LMS adaptive FIR filter architecture without adaptation delay," IEEE Trans. Signal Process. **46** (1998) 775 (DOI: 10.1109/78.661345).

[4] G. Long, *et al.*: "The LMS algorithm with delayed coefficient adaptation," IEEE Trans. Accoust. Speech, and Signal Processing **37** (1989) 1397 (DOI: 10.1109/29.31293).

[5] P. K. Meher and M. Maheshwari: "A high-speed FIR adaptive filter architecture using a modified delayed LMS algorithm," ISCAS (2011) 121 (DOI: 10.1109/ISCAS.2011.5937516).

[6] S. Y. Park and P. K. Meher: "Low-power, high-throughput, and low-area adaptive FIR filter based on distributed arithmetic," IEEE Trans. Circuits Syst. II, Exp. Briefs **60** (2013) 346 (DOI: 10.1109/TCSII.2013.2251968).

[7] Y. Yi, *et al.*: "High speed FPGA-based implementations of delayed-LMS filters," J. VLSI Signal Process. **39** (2005) 113 (DOI: 10.1023/B:VLSI.

DOI: 10.1587/elex.15.20180116
Received February 2, 2018
Accepted March 13, 2018
Publicized March 27, 2018
Copyedited April 10, 2018

0000047275.54691.be).
[8] L.-K. Ting, *et al.*: "Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **13** (2005) 86 (DOI: 10.1109/TVLSI.2004.840403).
[9] B. K. Mohanty and P. K. Meher: "Delayed block LMS algorithm and concurrent architecture for high-speed implementation of adaptive FIR filters," Proc. IEEE Region 10 TENCON2008 Conference (2008) 19 (DOI: 10.1109/TENCON.2008.4766786).
[10] L.-D. Van and W.-S. Feng: "An efficient systolic architecture for the DLMS adaptive filter and its applications," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process. **48** (2001) 359 (DOI: 10.1109/82.933794).
[11] S. Srinivasan, *et al.*: "Split-path fused floating point multiply accumulate (FPMAC)," Proc. Symp. Comput. Arith. (2013) 17 (DOI: 10.1109/ARITH.2013.32).
[12] D. Liu, *et al.*: "Delay-optimized floating point fused add-subtract unit," IEICE Electron. Express **12** (2015) 20150642 (DOI: 10.1587/elex.12.20150642).
[13] M. Lotfizad and H. S. Yazdi: "Modified clipped LMS algorithm," EURASIP J. Appl. Signal Process. **8** (2005) 1229 (DOI: 10.1155/ASP.2005.1229).

1 Introduction

Adaptive filtering is the important topic in signal processing field with applications in areas such as system identification, adaptive noise cancelation (ANC), channel equalization, measure system, etc. [1, 2, 3, 4, 5]. The most widely used algorithm for adaptive filters is the least-mean-square (LMS) algorithm due to its merits of superior performance and simple calculation. However, in practical applications, the computational complexity of the adaptive filter is the key factor for VLSI implementation [6]. We presented the architecture with less amount of calculation and good convergence characteristics based on the delayed dual sign LMS algorithm.

Some researchers have done a great deal to do with improving the systolic architectures of the DLMS algorithm in [6, 7, 8]. B. K. Mohanty *et al.* [9] have proposed delayed block LMS algorithm and its efficient implementation. P. K. Meher [1] have proposed a direct form DLMS adaptive filter with only 1

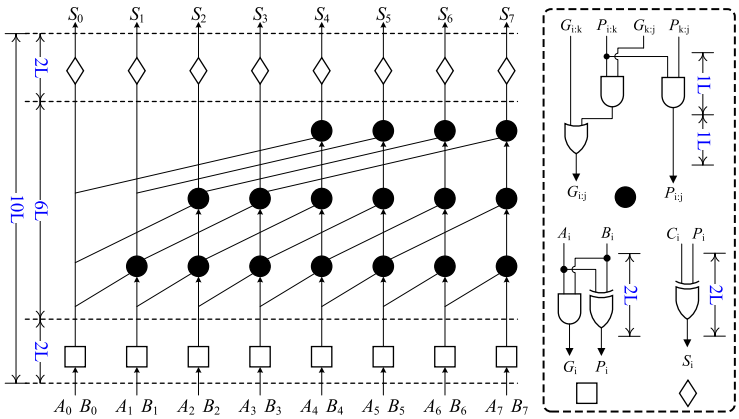


Fig. 1. The Kogge-Stone adder (8-bit).

DOI: 10.1587/elex.15.20180116
Received February 2, 2018
Accepted March 13, 2018
Publicized March 27, 2018
Copyedited April 10, 2018

adaptation-delay. Van *et al.* [10] have proposed a systolic structure using a tree rule for reducing the adaptation delay. Y. Yi *et al.* [7] have proposed a retimed 8-tap predictor system with the TDF-RDLMS architecture that has the delay of a multiplier on critical path.

In order to analyze the delay of the critical path of different architectures based on uniform standard, we use logic level [11, 12] to estimate the latency of different circuits which is independent of semiconductor manufacturing process. Often, we assumed the XOR gate involves 2 logic levels of delay which is two times that of the AND/OR gate. Fig. 1 shows an 8-bit *Kogge-Stone* adder structure for computing (A+B) which has a delay of 10 logic levels consisting of 1 level of ◇ (2LL), $\lceil log_2 8 \rceil = 3$ levels of ● ($3 \times 2\text{LL} = 6\text{LL}$) and 1 level of □ (2LL). The N-bit *Kogge-Stone* adder will involve roughly $(2 + 2 \times \lceil log_2 N \rceil + 2)$ logic levels of delay.

The proposed architecture using dual SLMS algorithm not only has the good convergence performance but also has the low computational complexity. In addition to the advantages mentioned above, the proposed fine-grained operation units are used to reduce the delay of critical path.

The rest of chapters are arranged as follows. In Sec. 2, we give a simple description of delayed dual sign LMS algorithm. The derivation process of the proposed architecture is given in Sec. 3. Sec. 4 presents the experimental simulation results and performance analyses. The conclusions are presented in Sec. 5.

2 The proposed delayed dual sign LMS algorithm

To decrease the computational complexity, we can get the sign algorithm as the most famous variant of the standard LMS algorithm [13]. The only difference between the sign algorithm and the standard LMS algorithm lies in the usage of sign function applied to the error signal in the weight coefficients updating process. The iterative formula of sign LMS algorithm is given by

$$y(n) = \boldsymbol{W}^T(n)\boldsymbol{X}(n) \tag{1}$$

$$e(n) = d(n) - \boldsymbol{W}^T(n)\boldsymbol{X}(n) \tag{2}$$

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + \mu\,\mathrm{sgn}\{e(n)\}\boldsymbol{X}(n), \tag{3}$$

where N is equal to the filter length, $e(n)$ is the error signal of the SLMS algorithm, and μ is a positive constant called step size parameter. $d(n)$ is the desired response that supervises the adjustment of the filter weight vector to make the filter output $y(n)$ resemble $d(n)$. The filter weight vector and the input sample vector indicated by $\boldsymbol{W}(n)$ and $\boldsymbol{X}(n)$ respectively are expressed by

$$\boldsymbol{X}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^T \tag{4}$$

$$\boldsymbol{W}(n) = [w_0(n), w_1(n), \ldots, w_{N-1}(n)]^T. \tag{5}$$

Although the SLMS algorithm has the very fast computation characteristic, it has two obvious drawbacks consisting of larger steady-state error and worse convergence rate compared to the standard LMS algorithm.

The dual-sign algorithm uses a new quantization scheme that involves the threshold clipping the error signal to avoid the crude quantization of gradient estimates inherent the sign LMS algorithm.

DOI: 10.1587/elex.15.20180116
Received February 2, 2018
Accepted March 13, 2018
Publicized March 27, 2018
Copyedited April 10, 2018

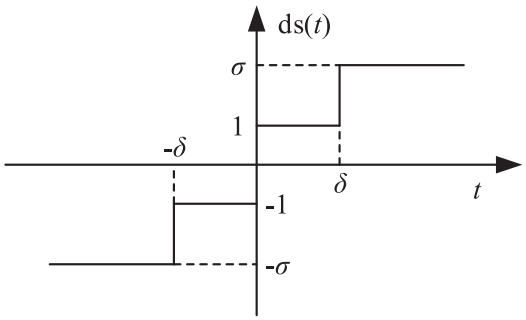


Fig. 2. The proposed 2-parallel adaptive filter.

For the dual-sign algorithm, the quantization function, as shown in Fig. 2, is given by

$$ds(t) = \begin{cases} \sigma\,\mathrm{sgn}\{t\}; & |t| > \delta \\ \mathrm{sgn}\{t\}; & |t| \le \delta \end{cases} \tag{6}$$

where $\sigma > 1$ is a power of two. The dual-sign algorithm utilizes the quantization function above described to generate the new gradient estimate, and the weight updating is performed as

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + \mu \times ds[e(n)]\boldsymbol{X}(n). \tag{7}$$

As we can see that the dual-sign LMS algorithm can not be computed in parallel caused by the strict order of execution. We can draw on the experience of the DLMS algorithm clearly illuminated by G. Long. The new weight updating of the delayed dual-sign algorithm is expressed as

$$\boldsymbol{W}(n+1) = \boldsymbol{W}(n) + \mu \times ds[e(n-m)]\boldsymbol{X}(n-m), \tag{8}$$

where m is the number of adaptation delay. Except for the m mentioned above, the choices of σ and δ can determine the convergence property of the delayed dual-sign algorithm, typically, a large σ tends to increase both convergence speed and excess mean squared error (MSE). A large δ tends to reduce both the convergence speed and the excess MSE.

3 The proposed architecture

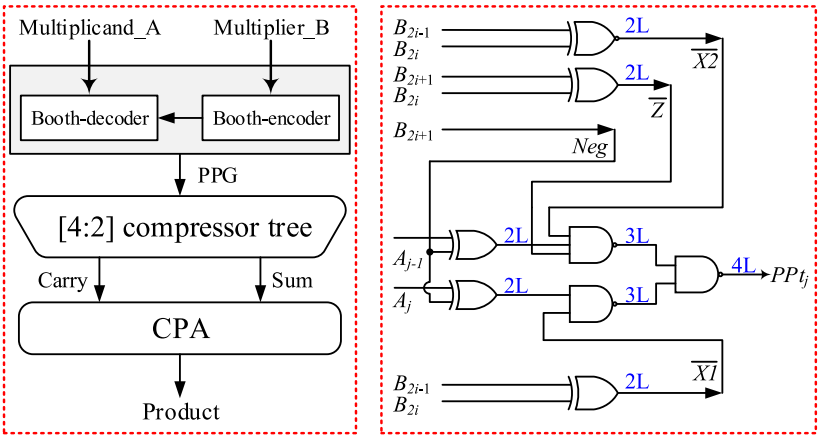
Before presenting the architecture of adaptive filter, we should set the data width used in this architecture. In order to ensure the accuracy of the output of adaptive filter compared to the software algorithm, the word-length of input signal sample is chosen to be 16.

3.1 Fast multiplier based on modified booth algorithm

As we all know that the delay of the multiplier is about twice that of the adder for same word-length. In order to get more precisely the delay relationship between adder and multiplier, we should study the hardware architecture of the multiplier by the logic level. The most common multiplier structure includes partial product generation module using booth algorithm, partial product compression module and the final adder as shown in Fig. 3(a).

As the traditional booth multiplier has some drawbacks in latency and area, too many researchers have done a lot of work on the booth encoding style. Combined

DOI: 10.1587/elex.15.20180116
Received February 2, 2018
Accepted March 13, 2018
Publicized March 27, 2018
Copyedited April 10, 2018



(a) Multiplier structure (b) Partial product generation

Fig. 3. The structure of modified booth multiplier.

with the existing design, this paper presents a modified booth encoding method. The modified partial product generation unit of the new booth algorithm is shown in Fig. 3(b). The critical path of the partial product generation (PPG) module has the delay consumed by the booth encoder and booth decoder circuit. Based on the previous assumptions, the PPG module has a delay of 4 logic levels. There are 9 partial products for the 16×16 bit signed multiplier as shown in Fig. 4, where e means the sign bit of the partial product that can be expressed as $e_i = PP_{iM}$. Partial products 0–7 are 17 bits. Each partial product i is sign extended with $s_i = Neg_i = B_{2i+1}$, which is 1 for negative multiples or 0 for positive multiples.

The partial product compression module implemented by the [4:2] compressor tree structure is shown in Fig. 5(b). This module has a three-stage logic circuit delay consisting of 2 levels of CSA and 1 level of [4:2] compressor. Because the CSA has two levels of XOR gate, it has a delay of 4 logic levels. Fig. 5(a) shows a new [4:2] compressor structure that has a delay of 6 logic levels. Based on the delay analysis above, the whole module has a delay of 14 logic levels for 16-bit signed multiplier.

Owing to the utilization of *Koggle-Stone* adder as the CPA, the delay of final sum operation of 32-bit words is roughly $\{2 + (log_2 32) \times 2 + 2\} = 14$ logic levels.

Through the previous analysis of three modules in terms of delay, we can reach the following conclusion that the 16×16 bit fast multiplier based on modified booth algorithm has a delay of $\{4 + 14 + 14\} = 32$ logic levels. The 16 bit CPA

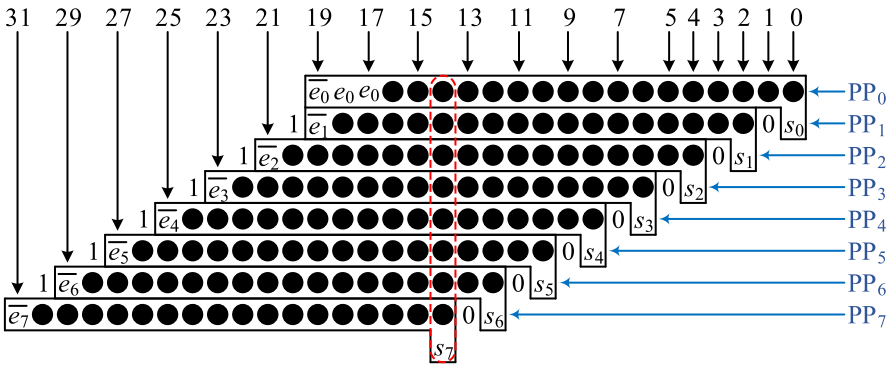
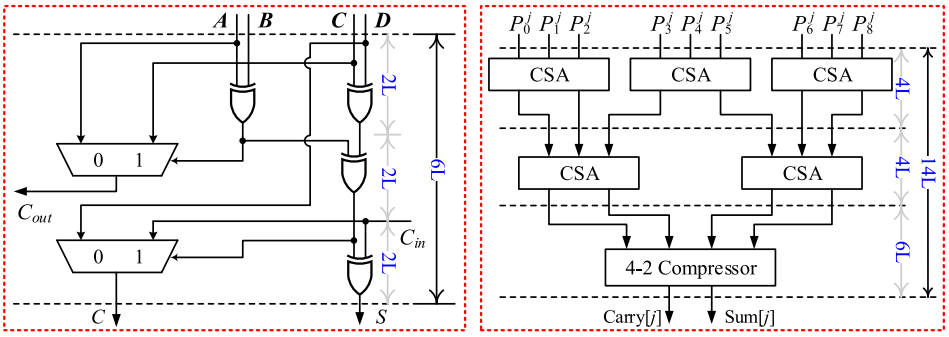


Fig. 4. Radix-4 booth-encoded partial products with sign extension.

DOI: 10.1587/elex.15.20180116
Received February 2, 2018
Accepted March 13, 2018
Publicized March 27, 2018
Copyedited April 10, 2018



(a) [4:2] compressor structure (b) Compressor tree of Multiplier

Fig. 5. The compressor tree structure of fast multiplier.

implemented by the Koggle-Stone adder, by contrast, has a delay of $\{2 + (log_2 16) \times 2 + 2\} = 12$ logic levels. The latency of booth multiplier is almost 2.7 times that of traditional adder.

3.2 Implementation of direct-form delayed DSLMS algorithm

Before giving the final architecture of proposed adaptive algorithm, we first introduce the 8-tap filter architecture of the DSLMS algorithm that has 8 processing modules (PM) as shown in Fig. 6. The step size μ is chosen to the power of two, so the multiplication operation can replaced by shift operation.

From Fig. 6, we can see that the critical path involves 1 level of multiplier, $\lceil log_2 8/2 \rceil = 2$ levels of [4:2] compressor, 2 levels of adder and 1 level of special adder. With the increase of filter tap, the delay of critical path will be very large. The delay of critical path except the special adder is roughly $\{32 + 6 \times 2 + 14 + 12\} = 70$ logic levels. The dual algorithm can be implemented by the special adder that can perform addition or subtraction depending on sign bit of the error signal as shown in Fig. 7. The OR-tree architecture generating control signal determines the size of the prescribed threshold.

The delay analysis for the *special adder* unit:

- *OR Tree Module.* The delay of this module involves one level of NOR gate and one level of standard three input NAND gate. Because the NOR gate and

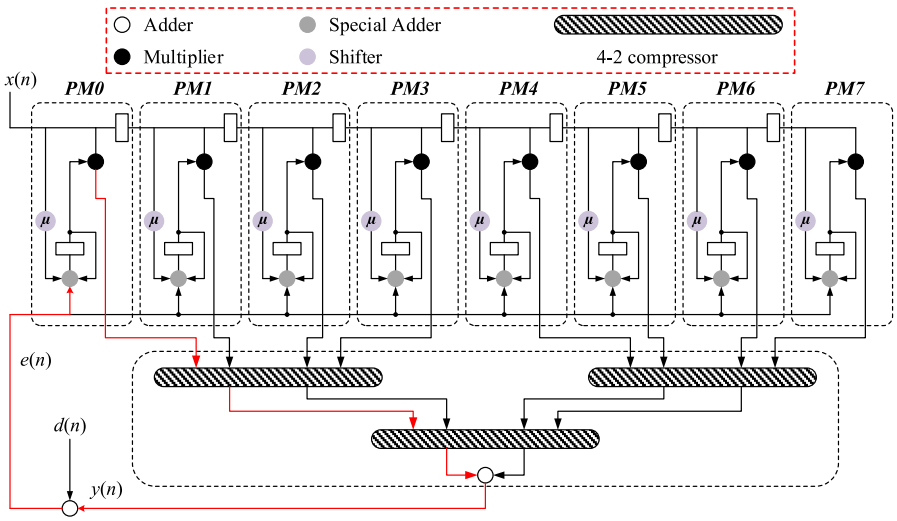


Fig. 6. The structure of dual sign LMS adaptive filter (8-tap).

DOI: 10.1587/elex.15.20180116
Received February 2, 2018
Accepted March 13, 2018
Publicized March 27, 2018
Copyedited April 10, 2018

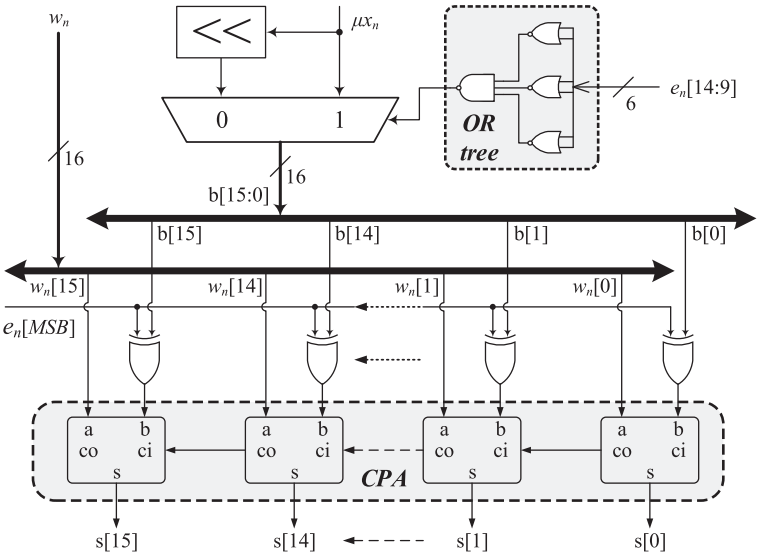


Fig. 7. The structure of special adder unit.

NAND gate have almost the same delay, that is roughly 2 logic levels, the delay of this module is about $\{1 + 1\} = 2$ logic levels—**2L**.

- *16-bit Multiplexer.* This module is used to select one of the two level parameters of the dual algorithm. Although XOR gate has a delay of 2 logic levels as mentioned above, the delay of this module involves 3 logic levels due to the large date width. We can get the delay for this module is roughly 3 logic levels—**3L**.
- *The XOR gate module.* This module is the 16-bit XOR gate implemented in parallel. According to the previous delay specification, this block has a delay of 2 logic levels—**2L**.
- *The CPA module.* This module is the 16-bit adder implemented by the *Koggle-Stone* architecture, so the delay of this module is roughly $\{2 + (log_2 16) \times 2 + 2\} = 12$ logic levels—**12L**.

According to the delay analysis of the four modules above, we can draw the conclusion that the delay of proposed *special adder* unit has a delay of 19 logic levels—**19L**.

On the basis of the above analysis, we know that the dual sign algorithm has 0 adaptation delay at the cost of large critical path with a delay of 89 logic levels. In some real time applications, the adaptive filter using dual sign algorithm fails in high throughput. The most straightforward way to reduce the critical path is to insert registers in its path, introducing the delayed algorithm. In this case, there is really a tradeoff between the critical path and the number of inserted registers. With the increase of the number of inserted registers, the overall performance of the algorithm will deteriorate. We introduced a new approach that fine-grained some of the modules on the critical path. This can be done with less registers to meet the high throughput requirements.

The proposed architecture has two adaptation delays that can ensure the convergence characteristics of the adaptive filter as shown in Fig. 8. Compared to the traditional method inserting the registers, the proposed design using the hardware reuse technique is effectively reduced by $N + 2$ registers. Only one

DOI: 10.1587/elex.15.20180116
Received February 2, 2018
Accepted March 13, 2018
Publicized March 27, 2018
Copyedited April 10, 2018

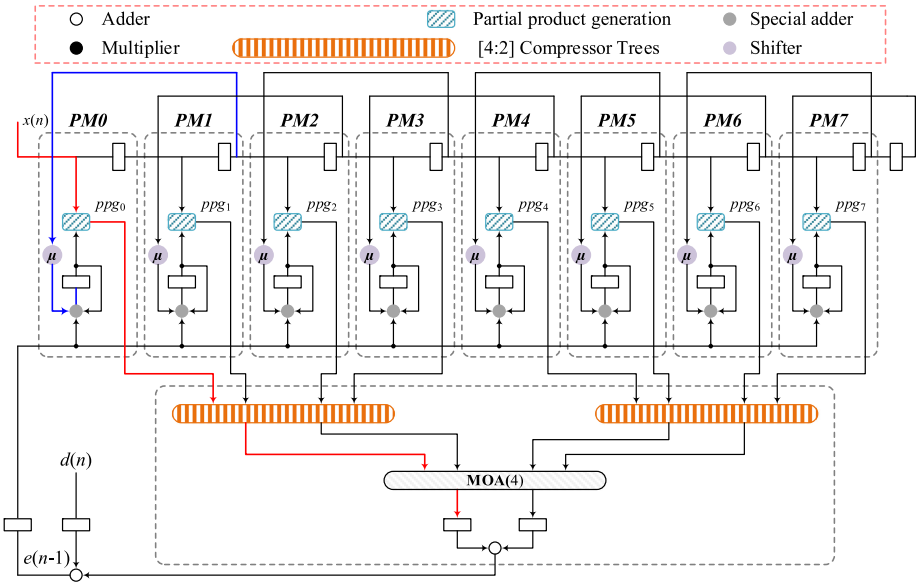


Fig. 8. The modification of the SFG (Filter length N = 8-tap).

register is added compared to the zero-adaptation architecture. Although the hardware reuse technique has no effect on the critical path, the input date should be delayed 2 cycles to compute the weight coefficients.

In addition to the optimization technique above, we have designed a fine-grained the *dot-product* unit that can perform the operation of $A1 \times B1 + A2 \times B2 + A3 \times B3 + A4 \times B4$. The proposed *dot-product* unit implemented by PPG module and partial product tree compression module is shown in Fig. 9.

From Fig. 9, we can see that the interconnection module has 0 logic level of delay due to the characteristic with no additional circuits. In addition, the partial

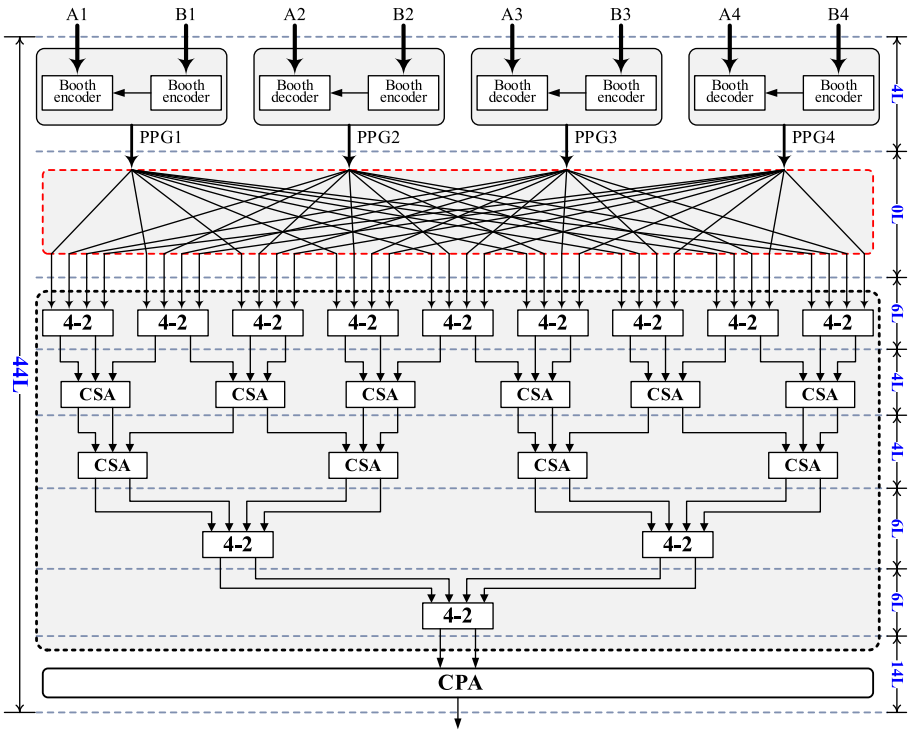


Fig. 9. The structure of dot-product unit.

DOI: 10.1587/elex.15.20180116
Received February 2, 2018
Accepted March 13, 2018
Publicized March 27, 2018
Copyedited April 10, 2018

product compression module is made up of 2 layers of CSA and 3 layers of [4:2] compressor. This module has a delay of $\{4 \times 2 + 6 \times 3\} = 26$ logic levels.

In order to reduce the computational complexity, the step size μ is set to be a power of two. Through doing this, the multiplication operation can be replaced with shift operation. Using the two main optimization strategies, we designed the new architecture shown in Fig. 8. In order to highlight the advantages of the proposed new architecture, we use an 8-tap adaptive filter for quantitative analysis.

The critical path in the second step is given by

$$T = t_{ppg} + \left\lceil log_2 \frac{N}{4} \right\rceil * t_{[4:2]} \tag{9}$$

where t_{ppg} represents the delay of PPG unit, $t_{[4:2]}$ represents the delay of [4:2] compressor. The Multi-Operand Adder (MOA) is usually implemented by the [4:2] compressor. The MOA(4) represents the 4-input addition in the 8-tap adaptive filter. With the increase of the filter length, it need many MOA module.

Red path analysis for Delayed DSLMS in Fig. 8

- *The Dot-Product Module.* This module is made up of PPG unit and partial products compression-tree. From the detail delay analysis above for the 16-bit data, this module has a delay of $\{4 + 26\} = 30$ logic levels—**30L**.
- *The MOA Module*. The architecture of MOA is the compression-tree consisting of [4:2] compressor and CSA. The MOA4 module only needs one layer of [4:2] compressor to generate the carry and sum. So the MOA(4) module has a delay of 6 logic levels—**6L**.

From the detailed analysis above, we can conclude that the red path has a delay of 36 logic levels—**36L**.

In order to prove that the red path is the critical path, we also need to analyze the delay of the blue path. The blue path is made up of two parts: the right shifter and the special adder. This right shifter is generally implemented by a barrel shifter as shown in Fig. 10. Using this architecture, the right shifter has 4 layers of multiplexer with a delay of 2 logic levels, so the delay of this module is $2 \times 4 = 8$ logic levels—**8L**. Through the previous analysis of delay, we know that the delay of special adder is 19 logic levels, so the blue path has a delay of $\{8 + 19\} = 27$ logic levels—**27L**. Now we can conclude that the critical path has a delay of 36 logic levels.

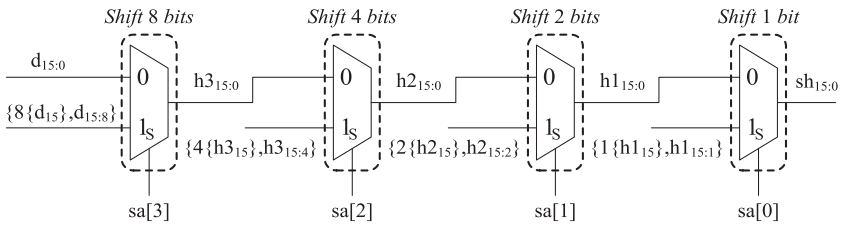


Fig. 10. The structure of the right shifter.

4 Results and comparisons

In order to illustrate the advantages of the proposed architecture, we can use Area-Delay-Power efficient structure in the recent works proposed by Y. Yi for comparison. Fig. 11 shows the overall 8-tap architecture with the same assumption above.

DOI: 10.1587/elex.15.20180116
Received February 2, 2018
Accepted March 13, 2018
Publicized March 27, 2018
Copyedited April 10, 2018

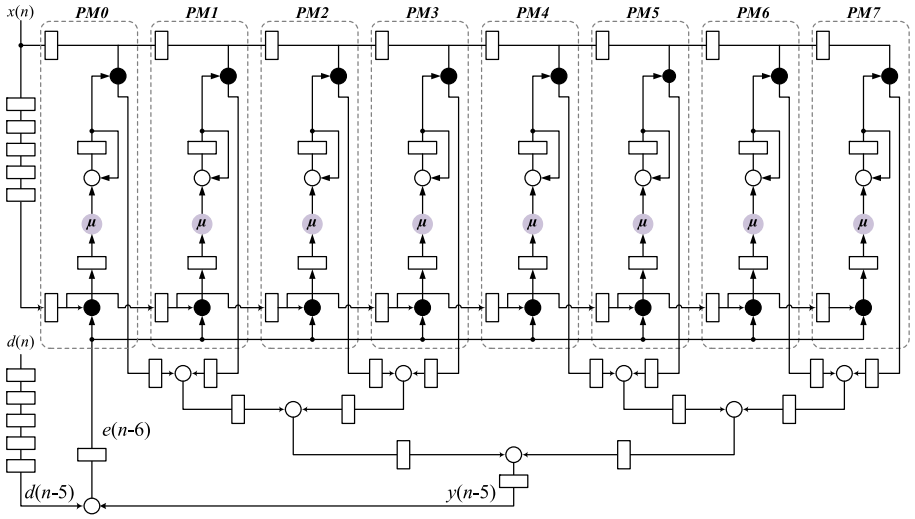


Fig. 11. The retimed 8-tap architecture for TDF-RDLMS adaptive filter with 6-adaptation-delay.

Y. Yi and L. K. Ting have proposed a retimed 8-tap TDF-RDLMS architecture, the critical path latency of this architecture is only that of the multiplier. Although this architecture has the short critical path, the power is very large due to the large number of registers inserted in the circuit. The step size μ is equal to the power of 2 in this design, so the multiplication operation can be replaced with the shift operation leading to a large reduction in area.

Each processing Module (PM) has the same structure in this architecture proposed by Y. Yi. The proposed PM has one more register than that of this architecture. From Fig. 11, we can see that the critical path is the delay of a multiplier. Using the previous knowledge of the fast multiplier, we can give an analysis for the critical path with 16-bit words.

Critical path analysis for TDF-RDLMS adaptive filter in Fig. 11

- *The Partial Product Generation.* This module is made up of booth decoder module and booth encoder module. From the detail delay analysis above, this module has a delay of 4 logic levels—**4L**.
- *The Compression-Tree Module*. This module is made up of 2 layers of CSA and 1 layer of [4:2] compressor. So the *Compression-Tree* module has a delay of $\{4 \times 2 + 6\} = 14$ logic levels—**14L**.

Table I. Comparison of hardware and time complexities

Design	Critical-Path	AD	Number of calculator		
			ADD	*MUL*	*REG*
Meher *et al.* [1]	$t_{mul} + (log_2N + 1)t_{add}$	1	$2N$	$2N$	$2N + 1$
Y. Yi *et al.* [7]	t_{mul}	$log_2N + 2$	$2N$	$2N$	$6N + log_2N + 2$
Long *et al.* [4]	$t_{add} + t_{mul}$	$log_2N + 1$	$2N$	$2N$	$3N + 2log_2N + 1$
M. D. Meyer [2]	$t_{add} + 2t_{mul}$	N	$2N + 1$	$2N$	$2N$
Van&Feng [10]	$t_{add} + t_{mul}$	$N/4 + 3$	$2N$	$2N + 1$	$5N + 3$
This work	$t_{dot} + log_2(N/4) \times t_{[4:2]}$	2	$N + 2$	0	$9N/4 + 3$

AD: adaptation-delay, ***ADD***: adders, ***MUL***: multipliers, ***REG***: registers.

DOI: 10.1587/elex.15.20180116
Received February 2, 2018
Accepted March 13, 2018
Publicized March 27, 2018
Copyedited April 10, 2018

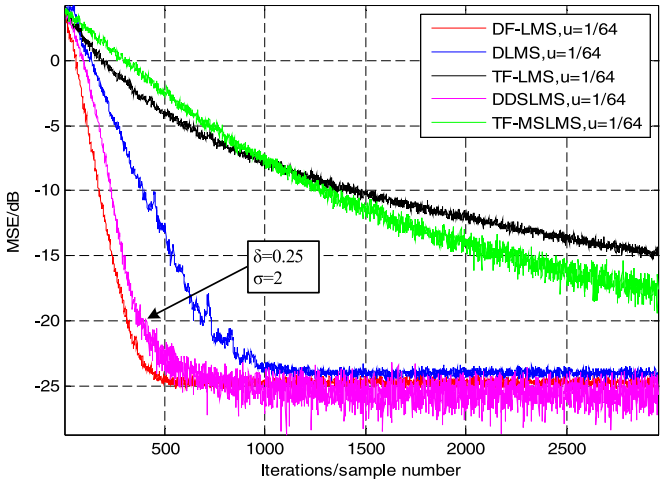


Fig. 12. The convergence performance with DDSLMS and other different adaptive algorithms.

- *The CPA Module.* This module is a 32-bit adder implemented by the *Koggle-Stone* adder, this module has a delay of 14 logic levels—**14L**.

Based on the delay analysis above, the critical path has a delay of 32 logic levels—**32L**.

Table I shows the comparison of hardware and the time complexities for the proposed design and other ones. The t_{add} and t_{mul} are the time consumed to perform the addition and multiplication operation. The $t_{[4:2]}$ is the delay of the [4:2] compressor implemented in Fig. 5(a). From this table, we can see that the architecture proposed by Y. Yi *et al.* [7] has the shortest critical path, but it has the largest registers generating more power. Due to the use of fine-grained dot product unit, there are no multipliers in proposed architecture. The t_{dot} of proposed design is the delay consisting of the *PPG* unit and [4:2] compressor-tree unit except the last CPA module, so it has less latency than a multiplier.

Simulation on computer as a key step is used to prove the correctness of the proposed algorithm as shown in Fig. 12. We use the same size convergence factor μ to distinguish the performance of different algorithms containing the direct-from (DF) and the transpose-form (TF). The input signal $x(n)$ is a mixed signal consisting of the trigonometric sine function signal and the white Gaussian noise with $SNR = 15$ dB. From the simulation results, it is easy to see that the convergence characteristics of proposed architecture is nearly the same as that of the LMS algorithm. In order to reduce the computational complexity, the σ and δ are

Table II. Synthesis results using TSMC 65 nm CMOS library

	DDSLMS	Meher [1]	Long [4]	Y. Yi [7]
DAT (ns)	1.54 (106%)	1.98 (137%)	1.76 (121%)	1.45 (100%)
logic level (L)	36 (112%)	62 (172%)	46 (128%)	32 (100%)
Throughput (1/ns)	0.65 (94%)	0.51 (74%)	0.57 (83%)	0.69 (100%)
Area (μm^2)	32346 (53%)	46631 (77%)	51649 (85%)	60559 (100%)
AD (cycles)	2	1	8	6
ADP ($\mu m^2 \times$ ns)	49813 (57%)	92329 (105%)	90902 (103%)	87811 (100%)
Power (mW)	7.66 (62%)	10.43 (84%)	10.72 (86%)	12.41 (100%)

DAT: data arrive time, ADP: area-delay product, AD: adaptation-delay.

DOI: 10.1587/elex.15.20180116
Received February 2, 2018
Accepted March 13, 2018
Publicized March 27, 2018
Copyedited April 10, 2018

chosen to be 2 and 0.25. The best existing architecture proposed by Y. Yi [7] has the same convergence characteristics as the DLMS algorithm so that the proposed design has better convergence characteristics than that of Y. Yi. These architectures proposed Meher [1] and Long [4] are all VLSI implemented based on the DLMS algorithm except for that proposed by Y. Yi.

In order to make a compelling comparison of circuit structure, the proposed design and other ones were coded in Verilog HDL, synthesized with DC Complier tools and the word-length of these adaptive filters are all chosen to be 16. Convergence characteristics and hardware cost, as we know, are the key factors in determining whether the adaptive filter structure is optimal. If a design has a good performance in both respects mentioned above, then the structure is optimal. Although the conventional structures are implemented with the DLMS algorithm, both the DDSLMS algorithm and the DLMS algorithm are derived from the LMS algorithm.

The synthesized results are as shown in Table II. To reflect the advantages of circuit delay optimization, we have compared to the design of Long [4] in terms of logic levels. The critical path of proposed architecture has a delay of 36 logic levels (36L) which is 22% lower than that of Long (36 VS 46), which is nearly matching DC synthesis comparison results with the DAT. Although the design presented by Y. Yi [7] has the shortest CP with only a multiplier, it involves nearly 62% more power (12.41 VS 7.66). Compared to the architecture proposed by P. K. Meher, the proposed design saves nearly 31% of area (32346 VS 46631) and 27% of power (7.66 VS 10.43). It is found that the proposed architecture involves nearly 43% less ADP than the best existing work designed by Y. Yi for the 8-tap filter.

5 Conclusion

This paper proposed a delay-optimized architecture using fine-grained dot product unit and a special adder for the delayed dual sign LMS algorithm. The proposed algorithm has a nearly equal convergence behavior with the LMS algorithm. Based on the detailed analysis above, the proposed design has almost the same critical path as that of the design of Y. Yi in terms of logic levels (36 VS 32). Owing to utilization of the fine-grained unit mentioned above, the throughput of DDSLMS architecture is increased by 27% compared to that of P. K. Meher. Furthermore, the proposed design involves nearly 43% less ADP and nearly 38% less power than that of the best existing structure proposed by Y. Yi. The proposed design can be achieved with 2-adaptation-delay, small hardware area and lower power consumption, simultaneously.

Acknowledgments

This work is supported by the project of Shenzhen Infrastructure layout (Grant No.: JCYJ20170412151226061).

DOI: 10.1587/elex.15.20180116
Received February 2, 2018
Accepted March 13, 2018
Publicized March 27, 2018
Copyedited April 10, 2018