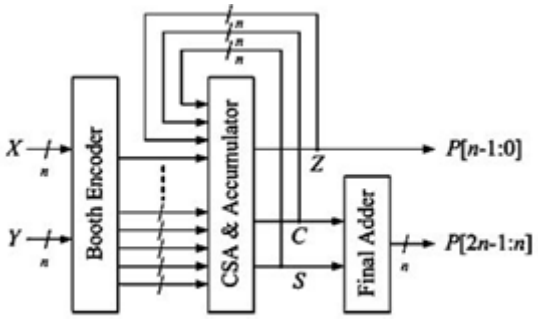


Fig. 4. Hardware architecture of the proposed MAC

The reason for separating the partial product addition as (7) is that three types of data are fed back for accumulation, which are the sum, the carry, and the preadded results of the sum and carry from lower bits. Now, the proposed concept is applied to Z in (5). If Z is first divided into upper and lower bits and rearranged, (8) will be derived. The first term of the right-hand side in (8) corresponds to the upper bits. It is the value that is fed back as the sum and the carry. The second term corresponds to the lower bits.

$$Z = \sum_{i=0}^{N-1} z_i 2^i + \sum_{i=N}^{2N-1} z_i 2^i. \quad (8)$$

The second term can be separated further into the carry term and sum term as

$$\sum_{i=N}^{2N-1} z_i 2^i = \sum_{i=0}^{N-1} z_{N+i} 2^i 2^N = \sum_{i=0}^{N-2} (c_i + s_i) 2^i 2^N. \quad (9)$$

Thus, (8) is finally separated into three terms as

$$Z = \sum_{i=0}^{N-1} z_i 2^i + \sum_{i=0}^{N-2} c_i 2^i 2^N + \sum_{i=0}^{N-2} s_i 2^i 2^N. \quad (10)$$

If (7) and (10) are used, the MAC arithmetic in (5) can be expressed as

$$P = \left(d_0 2Y + \sum_{i=1}^{N/2-2} d_i 2^{2i} Y + d_{N/2-1} 2^{N-2} Y \right) + \left(\sum_{i=0}^{N-1} z_i 2^i 2^N + \sum_{i=0}^{N-2} c_i 2^i 2^N + \sum_{i=0}^{N-2} s_i 2^i 2^N \right). \quad (11)$$

$$P = \left(d_0 2Y + \sum_{i=0}^{N-1} z_i 2^i \right) + \left(\sum_{i=1}^{N/2-1} d_i 2^{2i} Y + \sum_{i=0}^{N-2} c_i 2^i 2^N \right) + \left(d_{N/2-1} 2^{N-2} Y + \sum_{i=0}^{N-2} s_i 2^i 2^N \right). \quad (12)$$

B. Proposed MAC Architecture

If the MAC process proposed in the previous section is rearranged, it would be as Fig. 3, in which the MAC is organized into three steps. When compared with Fig. 1, it is easy to identify the difference that the accumulation has been merged into the process of adding the partial products. Another big difference from Fig. 1 is that the final addition process in step 3 is not always run even though it does not appear explicitly in Fig. 3. Since accumulation is carried out using the result from step 2 instead of that from step 3, step 3 does not have to be run until the point at which the result for the final accumulation is needed.

C. Proposed CSA Architecture

The architecture of the hybrid-type CSA that complies with the operation of the proposed MAC is shown in Fig. 5, which performs 8 x 8-bit operation. It was formed based on (12). In Fig. 5, Si is to simplify the sign expansion and Ni is to compensate 1's complement number into 2's complement number .$S[i]$ and $C[i]$ correspond to the ith bit of the feedback sum and carry. $Z[i]$ is the ith bit of the sum of the lower bits for each partial product that were added in advance and $Z'[i]$ is the previous result.

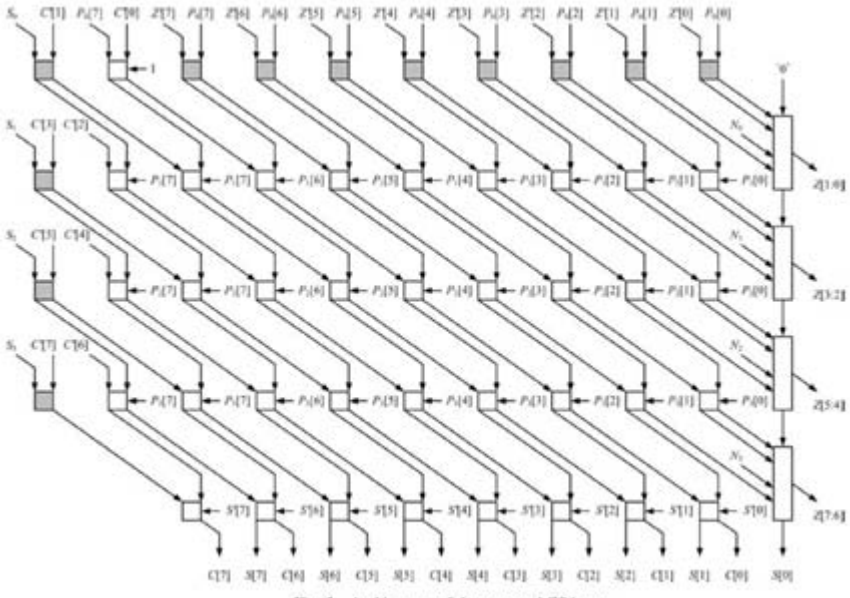


Fig. 5. Architecture of the proposed CSA tree.

4. Implementation and Experiment

In this section, the proposed MAC is implemented and analyzed. Then it would be compared with some previous researches.

A. Hardware Resource

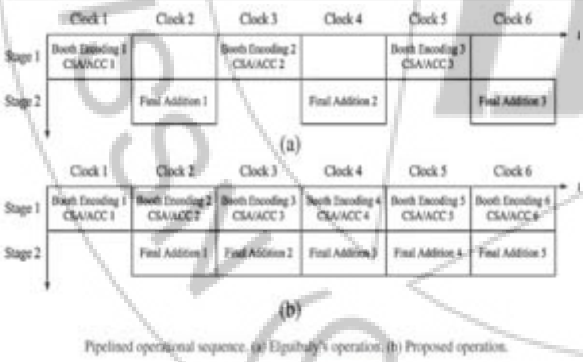
Analysis of Hardware Resource: The three layer architecture mentioned before is analyzed to compare the hardware resources and the results are given in Table II. The hardware resources in Table II are the results from counting all the logic elements for a general 16-bit architecture. The 90 nm CMOS HVT standard cell library from TSMC was used as the hardware library for the 16 bits. As Table II shows, the standard design uses the most hardware resources and the proposed architecture uses the least.

TABLE I
CHARACTERISTICS OF CSA

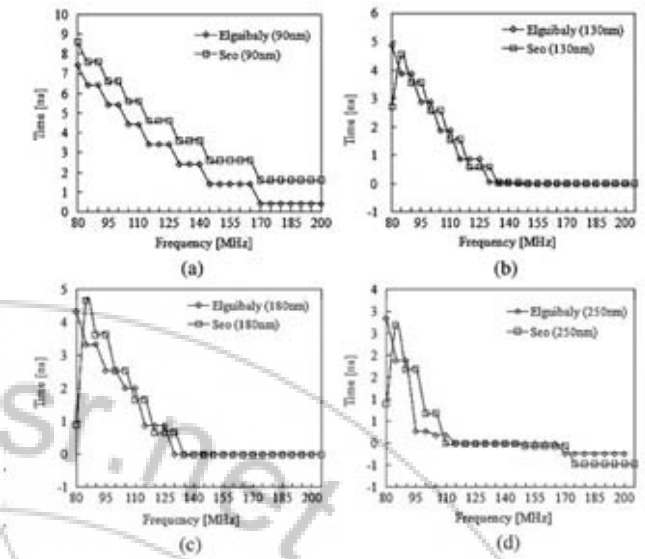
	[6]	[17]	The Proposed
Number System	2's Complement	1's Complement	1's Complement
Sign Extension	Used	Used	Not Used
Accumulation	Result Data of Final Addition	Result Data of Final Addition	Sum and Carry of CSA
CSA Tree	FA, HA	FA, 2 bits CLA	FA, HA, 2 bits CLA
Final Adder	$2n$ bits	$(n+2)$ bits	n bits

TABLE II
CALCULATION OF HADWARE RESOURCE

Component	[6]		[17]		The Proposed	
	General	16 bits	General	16 bits	General	16 bits
FA	$(\frac{n^2}{2}+n)$	964.8	$(\frac{n^2}{2}+2n+3)$	1092.1	$(\frac{n^2}{2}+\frac{n}{2})$	911.2
HA	0	0	0	0	$\frac{3n}{2}$	76.8
2 bit CLA	0	0	$(\frac{n}{2}-1)$	49	$\frac{n}{2}$	56
Accumulator	$(2n+1)$ bits CLA	214	-		-	
Final adder	$2n$ bits	197	$(n+2)$ bits	109.5	n bits	97
Total		1375.8		1250.6		1141



Pipelined operational sequence. (a) Elguibaly's operation. (b) Proposed operation.



5. Conclusion

In this paper, a new MAC architecture to execute the multiplication-accumulation operation, which is the key operation, for digital signal processing and multimedia information processing efficiently, was proposed. By removing the independent accumulation process that has the largest delay and merging it to the compression process of the partial products, the overall MAC performance has been improved almost twice as much as in the previous work.

References

[1] J. J. F. Cavanagh, Digital Computer Arithmetic. New York: McGraw-Hill, 1984.
[2] Information Technology-Coding of Moving Picture and Associated Autio, MPEG-2 Draft International Standard, ISO/IEC 13818-1, 2, 3, 1994.
[3] JPEG 2000 Part I Fina1119l Draft, ISO/IEC JTC1/SC29 WG1.
[4] O. L. MacSorley, "High speed arithmetic in binary computers," Proc. IRE, vol. 49, pp. 67-91, Jan. 1961.
[5] S. Waser and M. J. Flynn, Introduction to Arithmetic for Digital Systems Designers. New York: Holt, Rinehart and Winston, 1982.
[6] A. R. Omondi, Computer Arithmetic Systems. Englewood Cliffs, NJ:
[7] D. Booth, "A signed binary multiplication technique," Quart. J. Math., vol. IV, pp. 236-240, 1952.
[8] S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Electron Comput., vol. EC-13, no. 1, pp. 14-17, Feb. 1964.
[9] R. Cooper, "Parallel architecture modified Booth multiplier," Proc. Inst. Electr. Eng. G, vol. 135, pp. 125-128, 1988.
[10] N. R. Shanbag and P. Juneja, "Parallel implementation of a 4 X 4-bit multiplier using modified Booth's algorithm," IEEE J. Solid-State Circuits, vol. 23, no. 4, pp. 1010-1013, Aug. 1988.
[11] G. Goto, T. Sato, M. Nakajima, and T. Sukemura, "A 54 X 54 regular structured tree multiplier," IEEE J. Solid-State Circuits, vol. 27, no. 9, pp. 1229-1236, Sep. 1992.

[12] J. Fadavi-Ardekani, "M N Booth encoded multiplier generator using optimized Wallace trees," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 1, no. 2, pp. 120-125, Jun. 1993.
[13] N. Ohkubo, M. Suzuki, T. Shinbo, T. Yamanaka, A. Shimizu, K. Sasaki, and Y. Nakagome, "A 4.4 ns CMOS 54 54 multiplier using pass-transistor multiplexer," IEEE J. Solid-State Circuits, vol. 30, no. 3, pp. 251-257, Mar. 1995.
[14] A. Tawfik, F. Elguibaly, and P. Agathoklis, "New realization and implementation of fixed-point IIR digital filters," J. Circuits, Syst., Comput., vol. 7, no. 3, pp. 191-209, 1997.
[15] Tawfik, F. Elguibaly, M. N. Fahmi, E. Abdel-Raheem, and P. Agathoklis, "High-speed area-efficient inner-product processor," Can. J. Electr. Comput.Eng., vol. 19, pp. 187-191, 1994.
[16] F. Elguibaly and A. Rayhan, "Overflow handling in inner-product processors," in Proc. IEEE Pacific Rim Conf. Commun., Comput., Signal Process., Aug. 1997, pp. 117-120.
[17] F. Elguibaly, "A fast parallel multiplier-accumulator using the modified Booth algorithm," IEEE Trans. Circuits Syst., vol. 27, no. 9, pp.902-908, Sep. 2000.
[18] A. Fayed and M. Bayoumi, "A merged multiplier-accumulator for high speed signal processing applications," Proc. ICASSP, vol. 3, pp. 3212-3215, 2002.
[19] P. Zicari, S. Perri, P. Corsonello, and G. Cocorullo, "An optimized adder accumulator for high speed MACs," Proc. ASICON 2005, vol.2, pp. 757-760, 2005.
[20] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," IEEE J. Solid-State Circuits, vol. 25, no. 2, pp. 584-594, Feb. 1990
[21] Soojin Kim and Kyeongsoon Cho, Design of High-speed Modified Booth Multipliers Operating at GHz Ranges, World Academy of Science, Engineering and Technology; pp. 1-4, 6.1. 2010.
[22] Qingzheng LI, Guixuan LIANG, Amine BERMAK, A High-speed 32-bit Signed/Unsigned Pipelined Multiplier, IEEE 5th International Symposium on Electronic Design, Test & Applications; pp. 207-211, 2010.
[23] Rajput, R.P,Swamy,M.N.S, "High Speed Modified Booth Encoder Multiplier for signed and unsigned numbers", IEEE page 649-654, 28-30 March 2012.