

A Design of High-Performance Multiplier for Digital Video Transmission

Keisuke Okada^{† ‡} Shun Morikawa[†] Sumitaka Takeuchi[‡] Isao Shirakawa[†]

[†]Dept. Information Systems Engineering,
Faculty of Engineering, Osaka University
2-1 Yamada-Oka, Suita, Osaka 565 Japan
Tel: +81-6-879-7808
Fax: +81-6-875-5902
e-mail: {okada, morikawa, sirakawa}
@ise.eng.osaka-u.ac.jp

[‡] System LSI Lab.,
Mitsubishi Electric Corporation
4-1 Mizuhara, Itami, Hyogo, 664 Japan
Tel: +81-727-84-7343
Fax: +81-727-84-7439
e-mail {okada, takeuti1}@lsi.melco.co.jp

Abstract— A high performance design methodology is described for a multiplier to be used dedicatedly for digital video transmission. The key factor for such a multiplier is to operate at the speed of 30–100MHz but with the precision of 8–10 bits, since it is intended for FIR filtering of digital video data. In terms of implementing an FIR filter with more than ten taps, the same number of multipliers are required to be integrated. Moreover, for the preloadability of coefficients to the filter, each coefficient can be treated as a constant during the filtering operation. Motivated by these requirements and functionalities, a novel multiplier architecture is described, which is to be synthesized with the use of a high level synthesis tool PARTHENON in conjunction with manually designed macroblocks. Design results of the multiplier are also shown.

I. INTRODUCTION

One of the most basic operations in the digital signal processing is multiplication. High operation speed, low power consumption, and small die area are key factors in the design of such multipliers. Although various high-speed multipliers have been implemented[1, 2, 3], it is not easy to accomplish both the high speed operation and the low power consumption in a small die area. To overcome these difficulties, it is important to develop an application specific multiplier architecture.

Recently the digital video transmission is building up a promising market. In such digital transmission, multipliers are mainly used in a digital filter block, as shown in Fig. 1. The requirements for such a multiplier are the data precision of 8–10 bits and the operation speed of 30–100MHz. Moreover, the following two features have to be taken into account from the viewpoint of functional and circuit operations.

1. Programmability and preloadability of coefficients: This means that each coefficient can be treated as a constant during the filtering operation.
2. *Finite impulse response* (FIR) filtering with more than 10 taps: By the limitation of the operation speed of multipliers, the same number of multipliers should be integrated.

This paper proposes a novel multiplier architecture aiming at these two features. A number of design concepts are described in Section II, putting stress mainly on the new architecture associated with these two features. Design methodologies for functional blocks are described in Section III, together with the hardware evaluation for these blocks. Several implementation results are also shown to demonstrate the expected performances.

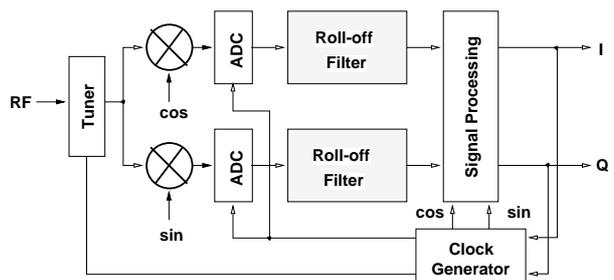


Fig. 1. Block diagram of demodulation part in video transmission system

II. ARCHITECTURE

The basic idea of high speed parallel multipliers proposed so far is based on the improvement of Booth's decoding method, the reduction of partial products, and

the development of high speed adders to sum up partial products[4, 5]. Instead, the proposed architecture is based on the idea that the signal level of each bit in the product is to be determined by simple combination of the multiplier and multiplicand numbers. To this end, our architecture takes a step of converting a binary coded multiplier number to a 1-out-of-N code, as adopted for the bit/word-line selection signal in memory devices. This idea seems to be realistic because the bit size of our targeting multiplier is up to 10 bits, instead of 32 and 64 bits.

When a signal line represents a given multiplier number, the multiplication can be executed by connecting that line to the final output which represents the required product, where the connection is determined in accordance with a given multiplicand number. In Fig. 2, a block diagram of the proposed multiplier is shown, which is constructed of the following four functional blocks; (1) Multiplier Decoder, (2) Control Signal Generator, (3) Control Signal Register, and (4) Product Generator.

Henceforth, for simplicity, functions of these blocks are exemplified, where both multipliers and multiplicands are represented in 2-bit positive binary numbers.

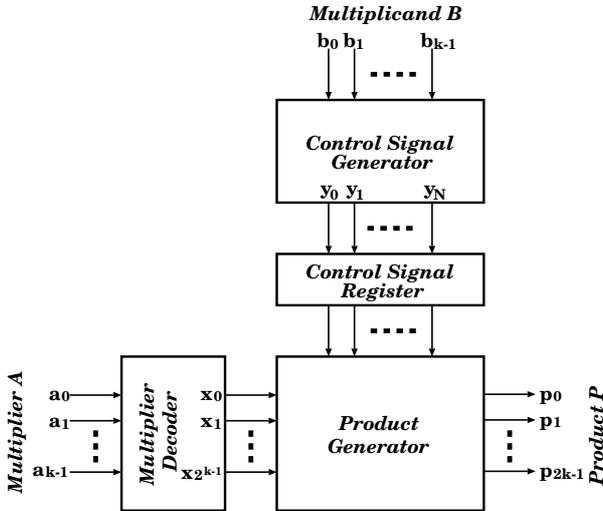


Fig. 2. Basic configuration of proposed multiplier

A. Multiplier Decoder

The input to this block is a binary coded multiplier $A (= a_0 * 2^0 + a_1 * 2^1)$. This block outputs a 1-out-of-N code $X (= x_1 \sim x_3)$, where x_i represents the value of multiplier A . The relation between A and X is summarized in Table I, and a logic diagram of the decoder is shown in Fig. 3.

Table I
Conversion from binary code of multiplier A
to 1-out-of-N code of X

	multiplier A		code X		
	a_1	a_0	x_1	x_2	x_3
0	0	0	0	0	0
1	0	1	1	0	0
2	1	0	0	1	0
3	1	1	0	0	1

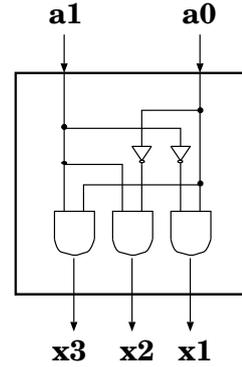


Fig. 3. Logic diagram of multiplier decoder

B. Control Signal Generator

Given a code $X (= x_1 \sim x_3)$ representing a multiplier A and a multiplicand $B (= 0 \sim 3)$, each bit of the product $P (= p_0 * 2^0 + p_1 * 2^1 + p_2 * 2^2 + p_3 * 2^3)$ is obtained as shown in Table II. It can be verified from Table II that $p_i (i = 0 \sim 3)$ can be set to logic "1" by the following different combinations of values of B ;

- (i) $B = "1" \text{ or } "3"$
- (ii) $B = "2"$
- (iii) $B = "2" \text{ or } "3"$
- (iv) $B = "1" \text{ or } "2"$
- (v) $B = "3"$

Define control signal $Y (y_0 \sim y_4)$ such that y_0 corresponds to combination (i), y_1 to (ii), y_2 to (iii), y_3 to (iv), and y_4 to (v). Table III shows the relation between

Table II
Relation between 1-out-of-N code X and multiplicand B
for each bit p_i of product $P = p_3 p_2 p_1 p_0$

code X (multiplier A)	product P			
	p_0	p_1	p_2	p_3
x_1 (1)	B=1,3	B=2,3		
x_2 (2)		B=1,3	B=2,3	
x_3 (3)	B=1,3	B=1,2	B=2	B=3

Table III

Relation between multiplicand B and control signal Y

	multiplicand B		control signal Y				
	b_1	b_0	y_0	y_1	y_2	y_3	y_4
1	0	1	1	0	0	1	0
2	1	0	0	1	1	1	0
3	1	1	1	0	1	0	1

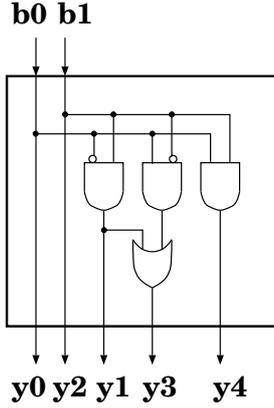


Fig. 4. Logic diagram of control signal generator

B and Y . We can see from this table that for example, when multiplicand B is equal to "3", y_0, y_2 , and y_4 of control signal Y are set to "1". Thus a logic diagram of the control signal generator is derived as shown in Fig. 4.

C. Product Generator

Table IV indicates that each bit p_i in product P can be derived from a simple combinational logic of x_i and y_j , that is, the logic equations can be written for p_i as

$$\begin{aligned}
 p_0 &= x_1 y_0 + x_3 y_0, \\
 p_1 &= x_1 y_2 + x_2 y_0 + x_3 y_3, \\
 p_2 &= x_2 y_2 + x_3 y_1, \\
 p_3 &= x_3 y_4.
 \end{aligned}$$

Table IV

Relation between 1-out-of- N code X and control signal Y for each bit p_i of product P

code X (multiplier A)	product P			
	p_0	p_1	p_2	p_3
x_1 (1)	y_0	y_2		
x_2 (2)		y_0	y_2	
x_3 (3)	y_0	y_3	y_1	y_4

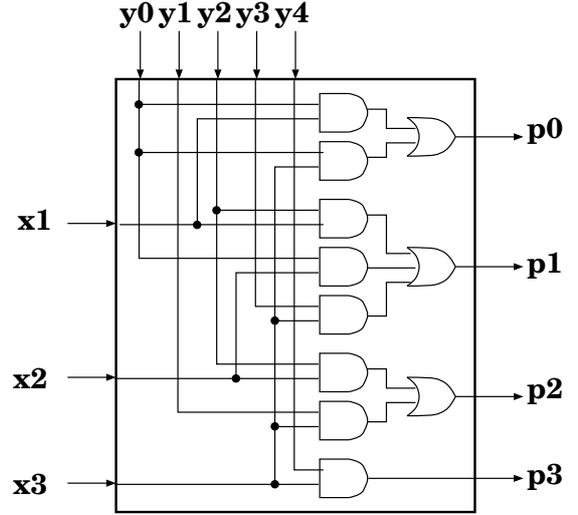


Fig. 5. Logic diagram of product generator

Thus a logic diagram of this block can be synthesized as shown in Fig. 5. The logic equation for each bit p_i can be given by the union of logical AND's of two variables x_i and y_j , without being affected by the bit sizes of multiplier A and multiplicand B . Each operation of this block can be speeded up, since the critical path in this block is composed only of a two-input AND gate and a multiple-input OR gate.

D. Consideration of Feature 1

As described in Section I, we can assume that coefficient data are given by multiplicand B , which can be fixed during the filtering operation, and hence we can see that control signal Y can be fixed as well. This means that, before the filtering operation, it is possible to store to the register the value of Y generated by the control signal generator. In this filtering operation, video data enter into the multiplier block as multiplier A , and then they are decoded through the multiplier decoder. Thus each component of the multiplier is divided into two parts; static and dynamic parts. Control Signal Generator and Control Signal Register are categorized in the static part, and Multiplier Decoder and Product Generator are in the dynamic one. This categorization is shown in Figs. 6 and 7.

E. Consideration of Feature 2

The basic FIR-filter block diagram is shown in Fig. 8, and the same transfer function can be derived by a modified FIR-filter block as shown in Fig. 9. As described in Section I, it is impossible to use only one multiplier repeatedly, since the operation speed required of a multiplier is 30-100MHz for the digital video transmission.

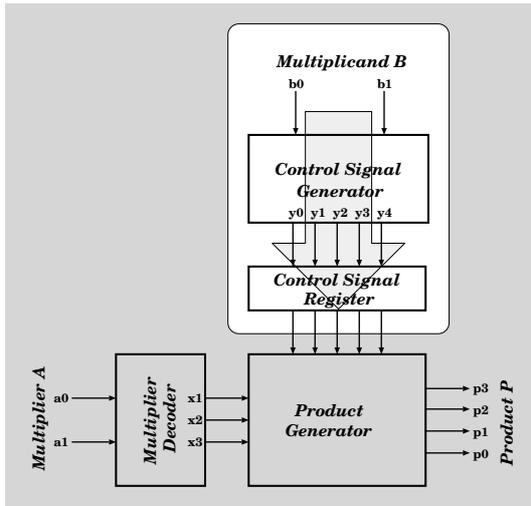


Fig. 6. Categorization of static operation

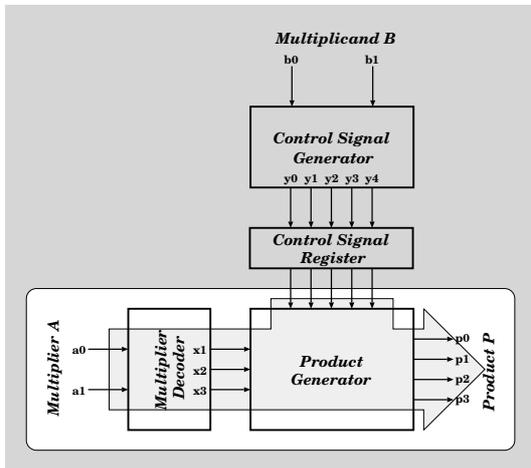


Fig. 7. Categorization of dynamic operation

This means that the same number of multipliers and taps are to be integrated in an FIR filter. However, the proposed multiplier configuration is constructed as shown in Fig. 10, where it should be noticed that there are only one multiplier-decoder and only one control-signal generator. Hence it can be seen that the design of Product Generator in the minimum possible area is the most important issue for the FIR filter implementation.

III. DESIGN METHODOLOGY

A design methodology is described for each block mainly in terms of the functionality stated in section II, together with the estimation of gates/transistors.

A. Multiplier Decoder

This block is categorized in the dynamic part, whose function is simple, with k inputs and $2^k - 1$ outputs. For

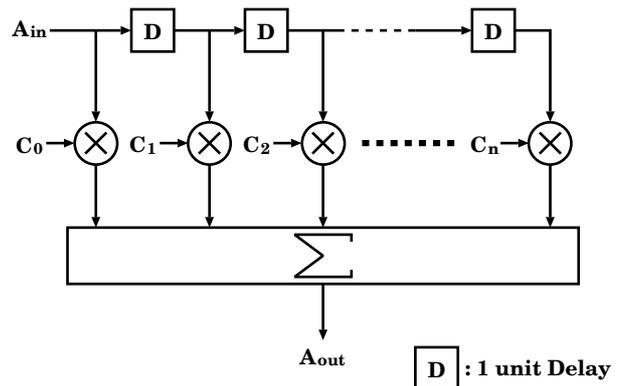


Fig. 8. Basic FIR-filter block diagram

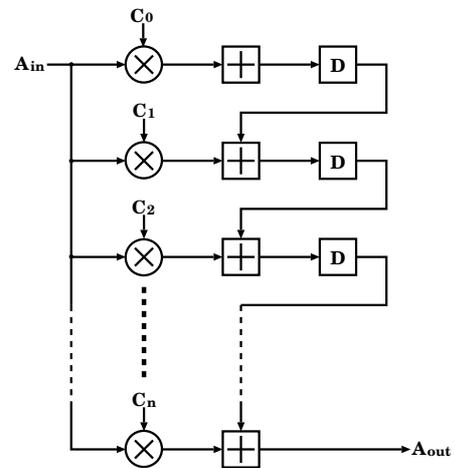


Fig. 9. Modified FIR-filter block diagram

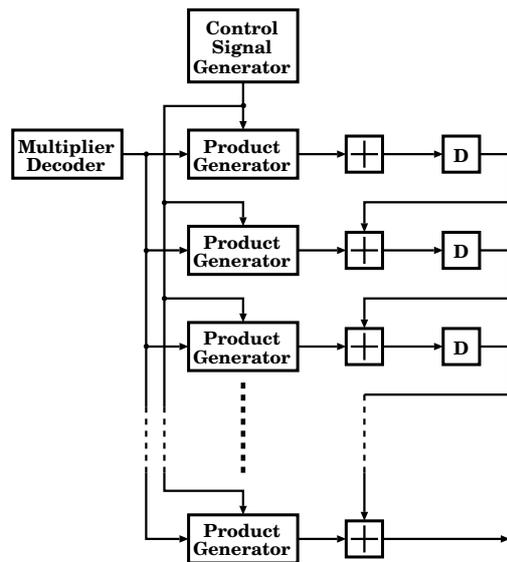


Fig. 10. FIR-filter block diagram using the proposed multiplier

example, in the 8-bit case, the circuit is synthesized by PARTHENON[6] using 619 gates.

B. Control Signal Generator

This block is categorized in the static part. A C-language program has been developed to generate the SFL (Structured Function description Language) for the synthesis of this control signal generator, with the use of a high-level synthesis tool PARTHENON. For example, in the cases of 8-bit by 8-bit and 10-bit by 10-bit, the numbers of control signals output from this block are 891 and 4563, respectively, but the synthesis was failed for both because of the hardware limitation.

However, the functionality of this block is combinational logic with 8- or 10-bit inputs and 891 or 4563-bit outputs. This means that the block can be expressed by ROM with a realizable hardware size, together with memory capacities of 222.7K-bit ($256W \times 891b$) and 4.5M-bit ($1024W \times 4563b$), respectively. The synthesized results are summarized in Table V in equivalent ROM size.

Table V
Summary of control signal generator block

	the number of control signal	gate count	ROM size
4 !_ 4	27	101.00	16 x 27
5 !_ 5	68	344.50	32 x 68
6 !_ 6	162	1770.00	64 x 162
8 !_ 8	891	*	256 x 891
10 !_ 10	4563	*	1024 x 4653
4 !_ 8	64	1348.25	256 x 64
5 !_ 10	160	6835.50	1024 x 160

C. Product Generator

This block is categorized in the dynamic part, and has enough margin for the operation speed. Here we attempted manual design utilizing TG's (Transmission Gates) to reduce the chip area. Fig. 11 shows the block diagram in the case of 2-bit by 2-bit. The logic equation for each bit of product P is generated by a C-language program. In this block, "0" input for multiplier A is executed simply by discharging n-channel transistor. In the cases of 8-bit by 8-bit and 10-bit by 10-bit, the numbers of transistors are 16,142 and 82,882, respectively. On the other hand, the critical path in this block is composed only of one TG and one OR-gate, even if the bit sizes of multiplier A and multiplicand B increase.

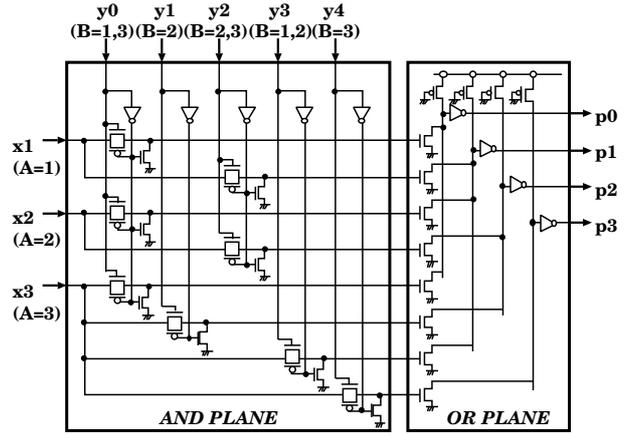


Fig. 11. Circuit configuration of product generator using TG

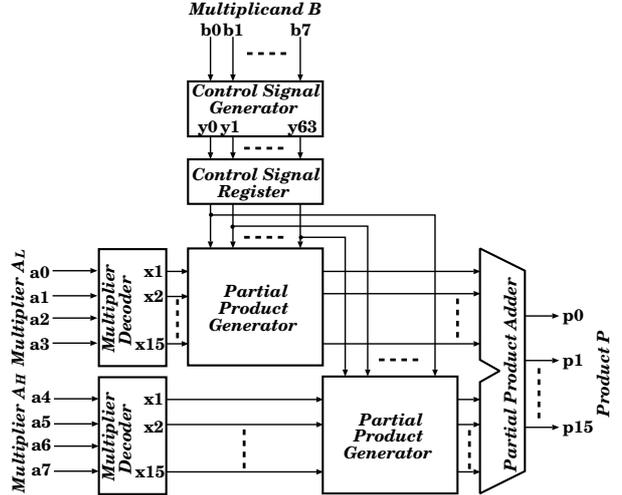


Fig. 12. Configuration of proposed multiplier in parallel use

IV. PROPOSED MULTIPLIER CONFIGURATION

Through the evaluation in Section III, the number of transistors especially in Product Generator is too large, although the expected operation speed is fast enough. To reduce the number of transistors, such an architecture as shown in Fig. 12 is devised, where two Multiplier Decoders and Product Generators are used in parallel.

Another configuration is also possible as shown in Fig. 13, where Multiplier Decoder and Product Generator are used twice. The statistics of transistors/gates and critical paths are summarized in Table VI.

Table VI
Summary of dynamic part

		transistor count			gate number in critical path
		multiplier decoder	product generator	total	
Fig. 12	8×8	256	1560	3300	$14 + 3(\text{TG})$
	10×10	640	3994	7796	$15 + 2(\text{TG})$
Fig. 13	8×8	128	780	2730	$18 + 6(\text{TG})$
	10×10	320	1997	5741	$19 + 5(\text{TG})$

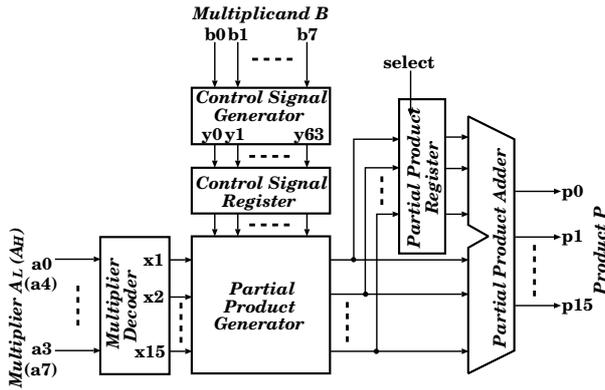


Fig. 13. Configuration of proposed multiplier
in repeating use

V. CONCLUSION

Two design approaches have been attempted for a multiplier dedicated to filtering, with the use of a high level synthesis tool PARTHENON, in conjunction with manually designed macroblocks. Expected performances are derived due to the features specified exclusively for the digital video transmission. All the evaluation data excludes the wiring area and delay. Thus development is continuing on the VLSI implementation of the two multiplier configurations.

REFERENCES

- [1] J. Mori, M. Nagamatsu, M. Hirano, S. Tanaka, M. Noda, Y. Toyoshima, K. Hashimoto, H. Hayashida, and K. Maeguchi, "A 10-NS 54x54-b parallel structure full array multiplier with $0.5\mu\text{m}$ CMOS technology," *IEEE J. Solid-State Circuits*, vol. 26, no. 4, pp. 600–605, 1991.
- [2] M. Hatamian, and G. Cash, "A 7-MHz 8-bitx8-bit parallel pipelined multiplier in $2.5\mu\text{m}$ CMOS," *IEEE J. Solid-State Circuits*, vol. SC-21, no. 4, pp. 505–513, 1986.
- [3] M. Ercegovac, and T. Lang, "Fast multiplication without carry-propagate addition," *IEEE Trans. on Computer*, vol. 39, no. 11, pp. 1385–1390, 1990.
- [4] A. D. Booth, "A signed binary multiplication technique," *Qt. J. Mech. Appl. Math.*, vol. 4, part 2, 1951.
- [5] C. S. Wallace, "A suggestion for fast multipliers," *IEEE Trans. Electronic Computers*, vol. EC-13, pp. 14–17, 1964.
- [6] Y. Nakamura, K. Oguri, A. Nagoya, M. Yukishita, and R. Nomura, "High-level synthesis design at NTT systems labs," *Trans. IEICE*, vol. E76-D, no. 9, pp. 1047–1054, 1993.