

A FAST MULTI-OPERAND MULTIPLICATION SCHEME

Hideaki Kobayashi

Electrical and Computer Engineering
Department
University of South Carolina
Columbia, SC 29208

ABSTRACT

Recent developments in integrated circuit technology have made efficient schemes for computer arithmetic possible. This paper discusses a generation-summation scheme for fast multi-operand multiplication. Synthesis of three-operand multipliers utilizing a single type of standard LSI device is also discussed.

INTRODUCTION

Due to the current advances in LSI circuits, large high-speed arithmetic functions such as multi-operand addition [1]-[4] and parallel multiplication [3]-[8] can be performed faster and at less cost than before. Recently, Stenzel et al. [7] described a compact high-speed multiplication scheme of generation-summation type using read-only memories (ROM's). Their scheme uses a carry lookahead adder to obtain the final product.

This paper describes a generation-summation scheme for fast multi-operand multiplication using a multiplier array for generation and a counter network for summation. An implementation of three-operand multiplication utilizing standard 256 x 8-bit ROM's (Texas Instruments 745471) will be used as an example scheme.

MULTI-OPERAND MULTIPLICATION

Fig. 1 shows an example of three-operand n -bit multiplication using two separate multipliers in cascade. The intermediate product is necessary to obtain the final product. This approach is disadvantageous for a larger number of operands, d , since the multiplication delay increases linearly with d . Fig. 2 shows another example of three-operand n -bit multiplication using a generation-summation-type multiplier. Here, a multiplier array is used to generate partial products. These are then summed by a counter network to the final

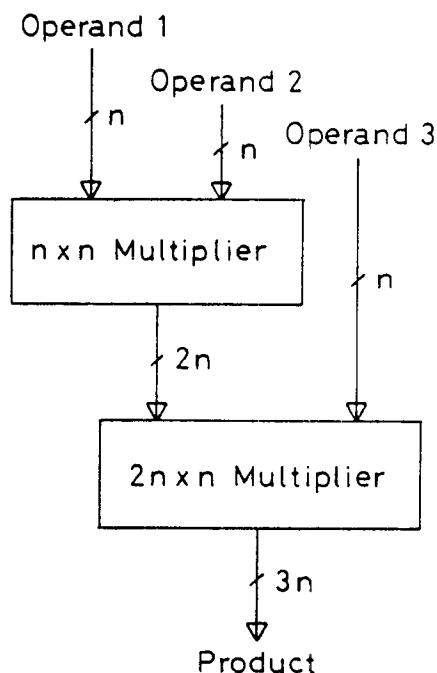


Fig. 1. Three-operand, n -bit multiplication example using two separate multipliers.

product. This approach is much faster for a larger number of operands since the total multiplication delay increases linearly with the logarithm of d . Pseudo-multipliers (or multiple multipliers) may be used to generate the partial products.

MULTIPLE MULTIPLIERS

A multiple multiplier is a device that performs simultaneous multiplication of a number of operands. Multipliers of this type can be denoted by

$$(p_1, p_2, \dots, p_d; q)$$

multipliers, where p_i is the i -th operand length and q is the product length. An example of a (2, 2, 4; 8) multiplier is shown in Fig. 3, where \circ represents a binary digit. The product length q is equal to the total operand length:

$$q = \sum_{i=1}^d p_i.$$

The ROM lookup technique may be used to implement multiple multipliers. A $2^p \times q$ -bit ROM can be programmed to treat the p address lines as d operands and perform a table lookup on the product. For example, a 256 x 8-bit ROM can be used to implement a (2, 2, 4; 8) multiplier. ROM's programmed in this fashion may be used to generate partial products in larger multipliers.

LARGE MULTIPLIER SYNTHESIS

An array of multi-input AND gates may be used to generate a partial-product matrix. The number N_{AND} of partial-product bits formed by d -input AND gates can be written as

$$N_{AND} = n^d.$$

An alternate approach using an array of multiple multipliers rather than multi-input AND gates could be employed to generate a smaller partial-product matrix. The number N_{MUL} of partial-product bits formed by $(p_1, p_2, \dots, p_d; q)$ multipliers can be written as

$$q \prod_{i=1}^d \left\lfloor \frac{n}{p_i} \right\rfloor \leq N_{MUL} \leq q \prod_{i=1}^d \left\lceil \frac{n}{p_i} \right\rceil$$

where $\lfloor X \rfloor$ represents the largest integer not greater than X and $\lceil X \rceil$ represents the smallest integer not less than X . For a larger number of operands, we have

$$N_{MUL} \ll N_{AND}.$$

Thus the use of multiple multipliers to generate partial products saves a significant number of counters for successive summation.

Fig. 4 shows an example of three-operand, 4-bit partial-product generation using four (2, 2, 4; 8) multipliers. Operand 1 and 2 are each divided into two 2-bit sub-operands. These sub-operands and Operand 3 are shown in boxes. Sets of the sub-operands and Operand 3 form the partial-product matrix (Matrix 1). A counter network can then sum the partial products to form the final product.

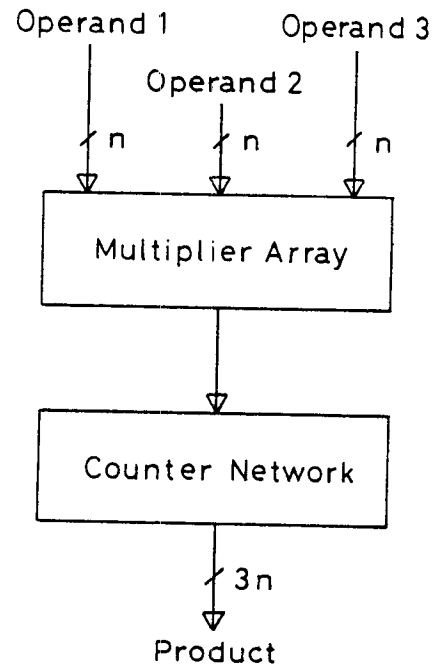


Fig. 2. Three-operand, n -bit multiplication example using a single multiplier.

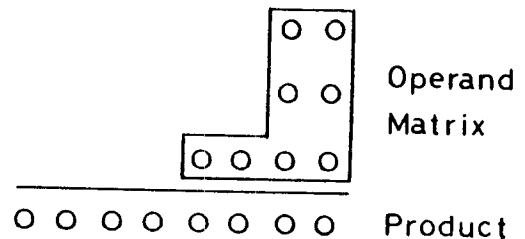


Fig. 3. (2, 2, 4; 8) multiplier example.

Fig. 5 shows an example of three-operand, 4-bit partial-product summation using generalized counters. These counters are shown in boxes. Each circle in Matrix 1 (Fig. 5) corresponds to a partial-product bit in Fig. 4. In the first stage, 32 bits of Matrix 1 are grouped into sets of 8 binary inputs, which the counters reduce into sets of v binary outputs. The number v is

$$v = \lfloor \log_2 S \rfloor + 1$$

where S is the sum of the counter inputs. The

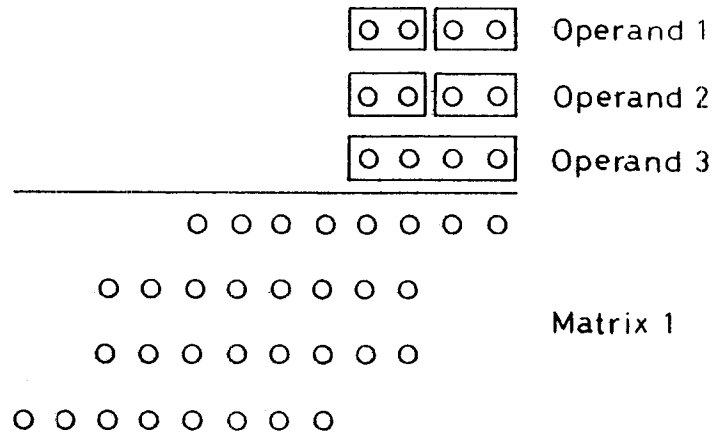


Fig 4. Three-operand, 4-bit partial-product generation.

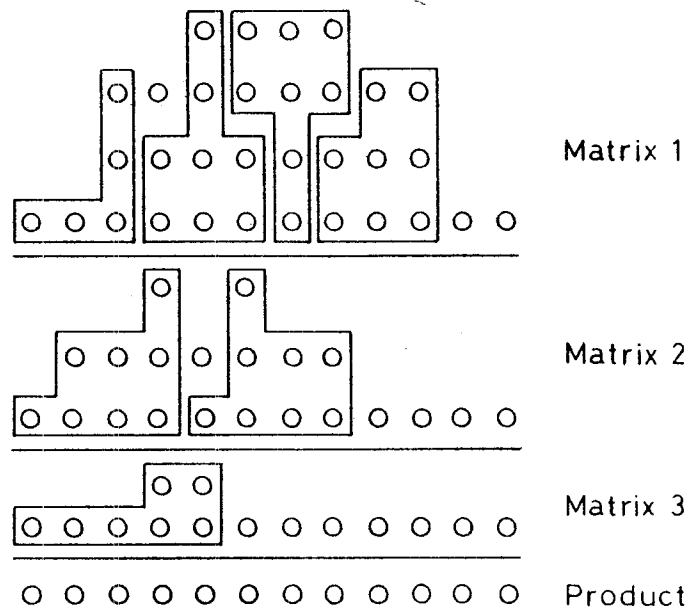


Fig. 5. Three-operand, 4-bit partial-product summation.

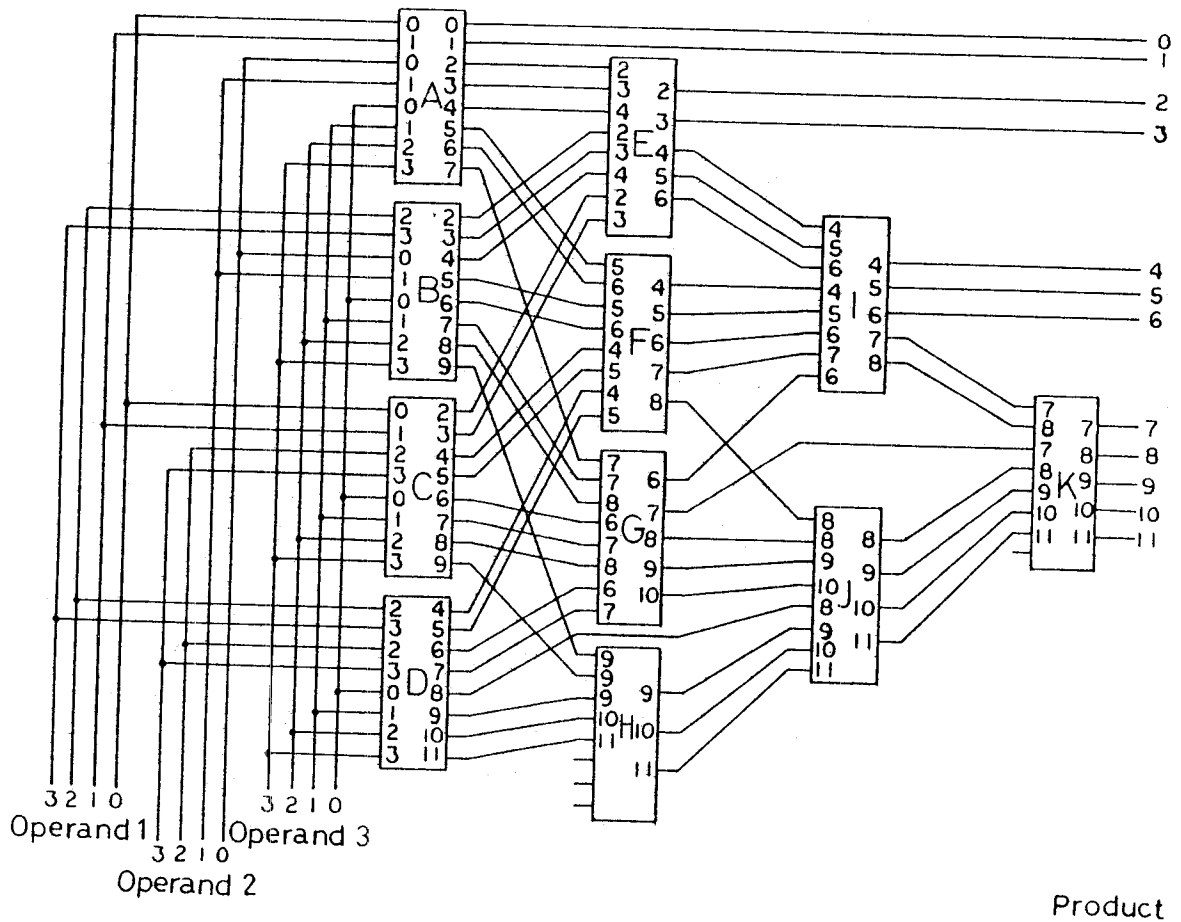


Fig. 6. Three-operand, 4-bit multiplier implementation with 256 x 8-bit ROM's.

final output(s) and the extra bit(s) outside of these groups of 8-bit sets are passed along to the second stage with the counter outputs from the first stage. The left-most counter may receive less than 8 bits. In the second stage, 21 bits of Matrix 2 are grouped and then reduced using a similar technique. The above process is continued through the final stage where the product is obtained in a 12-bit binary form. A carry lookahead

adder is not necessary since various counter types allow high-speed reduction of the individual portions of the whole matrices (see Fig. 5). Fig. 6 shows a three-operand 4-bit multiplier using 256 x 8-bit ROM's. Note that these multipliers (A-D) and counters (E-K) can be implemented with a single type of LSI device. Assuming that each ROM exhibits one unit delay, the total number of unit delays for the circuit is equal to the number of stages.

CONCLUSION

In conclusion, the generation-summation concept for two-operand multiplication has been extended to include multi-operand multiplication. Multiple multipliers allow efficient generation of a partial-product matrix. A high-speed summation of the partial-product matrix can be achieved without a carry lookahead adder. An important advantage is that only a single type of standard LSI device is required to synthesize larger multipliers. This proposed approach is also applicable for implementing other arithmetic processors with LSI/VLSI circuits.

REFERENCES

- [1] I.T. Ho and T. C. Chen, "Multiple Addition by Residue Threshold Functions and Their Representation by Array Logic," IEEE Trans. Comput., vol. C-22, pp. 762-767, Aug. 1973.
 - [2] H. Kobayashi and H. Ohara, "A Synthesis Method for Multiple Input Adders with a ROM Network," Trans. IECE Japan, E-62, pp. 9-15, Jan. 1980.
 - [3] S. Sigh and R. Waxman, "Multiple Operand Addition and Multiplication," IEEE Trans. Comput., C-22, pp. 113-120, Feb. 1973.
 - [4] E. E. Swartzlander, Jr., "Merged Arithmetic," IEEE Trans. Comput., C-29, pp. 964-950, Oct. 1980.
 - [5] C. S. Wallace, "A Suggestion for Fast Multiplier," IEEE Trans. Comput., EC-13, pp. 14-17, Feb. 1964.
 - [6] L. Dadda, "Some Schemes for Parallel Multipliers," Alta Freq., vol. 34, pp. 349-356, May 1965.
 - [7] W. J. Stenzel, W. J. Kubitz, and G. H. Garcia, "A Compact High-Speed Parallel Multiplication Scheme," IEEE Trans. Comput. C-26, pp. 948-957, Oct. 1977.
 - [8] H. Kobayashi, T. Yamada, and H. Ohara, "Two's Complement Parallel Implementation of Large Multipliers," in Proc. IEEE 1980 Int. Conf. on Circuits and Comput., Port Chester, NY, Oct. 1980, pp. 1085-1088.
-