Writing VHDL Models of Parallel $n \times n$ Bit Multiplication Blocks for DSP Applications

Siddika Berna Örs, Ahmet Dervisoglu

Istanbul Technical University, Electrical and Electronics Engineering Faculty,
Department of Electronics and Communication Engineering 80626 Maslak, Istanbul, TURKEY
Tel: 90 212 285 36 12, 90 212 285 36 02, Fax: 90 212 285 36 79,
e-mail: berna@ehb.itu.edu.tr, dervis@ehb.itu.edu.tr

Abstract

In this paper, we propose two models of multiplication blocks by using VHDL. The algorithms that are used for writing the models are suitable for high speed multiplication and have regular cellular array structures. We have simplified some equations given in the references and then have written the VHDL model accordingly. Thus, a circuit synthesized by using the models proposed in this paper will have less area and gate count on the longest path than those given in the literature. We propose formulas for calculating area values and the gate count on the longest path of the circuits for both of two multiplication blocks for any value of n for, $2 \leq n \leq 54$. Also we propose a method to determine the types of gates on the longest path of the circuit.

1 Introduction

Multiplication is one of the basic operations in digital signal processing systems providing filtering, correlation and range measurement. A high-speed multiplier is required for real-time signal processing systems. Designing fast multipliers has long been a great theoretical and practical interest for computer scientists and engineers. Various multiplication algorithms have been proposed and practically used [1, 2, 3, 4, 5, 6].

In this paper, we propose two models of multiplication blocks by using VHDL. The algorithms that are used for writing the models are suitable for high speed multiplication and have regular cellular array structure [1, 2, 3, 4, 5, 6] .

In literature all the equations for implementing some blocks of the multiplier circuits are given for general situation. In this paper, we have simplified some equations given in the references for some special cases and then have written the VHDL model accordingly. Therefore, a circuit synthesized by using the models proposed in this paper will have less area value and gate count on the longest path than those given in the literature.

The area values are given as proportional to n^2or n and the gate counts on the longest paths of the circuits are given as proportional to $log_2 n$ or n in references that the algorithms are proposed. In this paper, we propose formulas for calculating area values and the gate count on the longest path of the circuits depending on n for both of two multiplication blocks. Also we propose a method to determine the types of gates on the longest path of the circuits. By using the formulas about the area and the gate count on the longest path of the circuits, we have calculated these critical values of the circuits for different n values and have given the results in two tables. So a DSP chip designer can use these tables to decide about the multiplication block which meets the requirements. In literature, to our knowledge, VHDL models of the circuits are given for certain n values; we have written the VHDL models of the circuits as taking the n as a parameter: n can take any value for $2 \leq n \leq 54$.

2 VHDL Models of the Multiplier Blocks

2.1 Computation Model

We will be concerned with modeling of a multiplier as a combinational circuit. We assume that fan-in (in degree) of each computation element is restricted in a certain constant, but fan-out (out degree) is not restricted. Because all the multiplication blocks have regular cellular array suitable for VLSI implementation and all the connections between sub-blocks are short, we assume that wires have no delay to simplify the evaluation. According to these assumptions, the computation time on a combinational circuit is linearly proportional to the depth of it. The depth of a combinational circuit is the number of computation elements on the longest path, i.e., the path that has the most computation elements in it. We estimate the complexity of a combinational circuit by its depth and gate count.

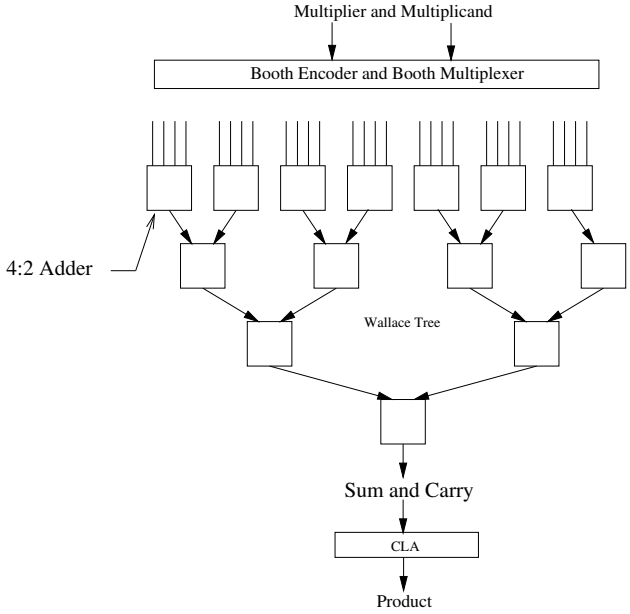

Figure 1: The block diagram of parallel multiplier

2.2 Parallel Structured Full Array Multiplier

The block diagram of parallel structured full array multiplier is shown in Fig. 1. This multiplier uses a parallel structure with Booth's algorithm and Wallace's tree [1].

The number of terms to be added is halved in the modified Booth recoding scheme by comparing every other bit of the multiplier operand with the next and previous bits and using this 3-bit code to determine which operation to perform [2].

The operations are add once (001, 010), add twice (011), subtract once (101, 110), subtract twice (100) or do nothing (000, 111). The subtract twice operation is performed by adding the two's complement of the multiplicand shifted left by a single bit. In the circuit for Booth multiplier the control vectors ADD, SUB, ZERO, ADD×2, SUB×2 and NEG are generated from the multiplier operand using Booth encoding logic [2, 3].

The inputs of the encoder circuit are groups of the multiplier's three bits sequences. Multiplier is shown as $b_{n-1}b_{n-2}\cdots b_1b_0$. If n is even the multiplier is grouped as shown in Fig. 2, if it is odd the multiplier is grouped as in Fig. 3. In the literature, only the case that n is even is considered, in this paper because of we have written the models for any values of n between 2 and 54, we have also considered the case that n is odd. ith value of ADD, SUB, ZERO, ADD×2, SUB×2 and NEG vectors are calculated by using ith group of the multiplier ($i = 0, 1, \cdots, \frac{n}{2}$ if n is even, $i = 0, 1, \cdots, \frac{n-1}{2}$ if n is odd).

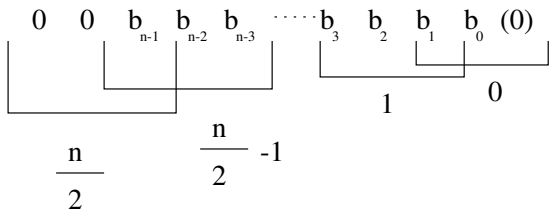

Figure 2: If n is even, the way that the multiplier is grouped

ith values of the output vectors of encoder circuit are calculated according to the following equations ($i = 1, 2, \cdots, \frac{n}{2} - 1$ if n is even, $i = 1, 2, \cdots, \frac{n-1}{2} - 1$ if n is odd).

$ZERO(i) = \overline{(b_{2i+1} + b_{2i} + b_{2i-1})},\ ADD(i) = \overline{b_{2i+1}}(b_{2i} \oplus b_{2i-1}),\ ADD \times 2(i) = \overline{b_{2i+1}}b_{2i}b_{2i-1}$

$SUB(i) = b_{2i+1}(b_{2i} \oplus b_{2i-1}),\ SUB \times 2(i) = b_{2i+1}\overline{(b_{2i} + b_{2i-1})},\ NEG(i) = SUB(i) + SUB \times 2(i)$ (1)

The name of the block we have designed to perform above equations is EN_1. The inputs of the block are b_{2i+1}, b_{2i} and b_{2i-1}, the outputs of the block are ith values of ADD, SUB, ZERO, ADD×2, SUB×2 and NEG vectors. There are three three-input AND (AND3) gates, four INVERTER (INV) gates, two two input OR (OR2) gates, one three input NAND (NAND3) gates and one XOR gate in this block. We consider that a two input NAND gate has unit areas, so the area is 17.5 units. There are one two input AND (AND2) gate and one XOR gate on the longest path of the block.

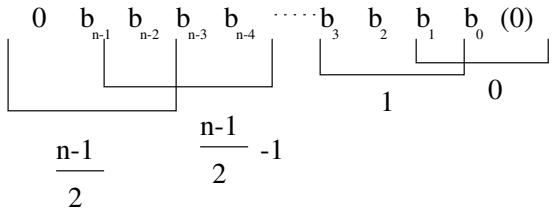

Figure 3: If n is odd, the way that the multiplier is grouped

Because b_{-1} is always "0", we have simplified (1) to calculate $0th$ value of the vectors as follows.
$ZERO(0) = \overline{(b_1 + b_0)},\ ADD(0) = \overline{b_1}b_0,\ ADD \times 2(0) = 0,\ SUB(0) = b_1b_0,\ SUB \times 2(0) = b_1\overline{b_0},$
$NEG(0) = b_1$ (2)

The name of the block we have designed to perform above equations is EN_0. The inputs of the block are b_1 and b_0, the outputs of the block are the $0th$ values of ADD, SUB, ZERO, ADD×2, SUB×2 and NEG vectors. There are three AND2 gates, two INV gates and one two input NOR (NOR2) gate in this block, so the area is 6.5 units instead of 17.5 units.

If n is even, because of b_n and b_{n+1} are always "0" we have simplified (1) to calculate $\frac{n}{2}th$ values of the output vectors of encoder circuit as follows.
$ZERO(\frac{n}{2}) = \overline{b_{n-1}},\ ADD(\frac{n}{2}) = b_{n-1},\ ADD \times 2(\frac{n}{2}) = SUB(\frac{n}{2}) = SUB \times 2(\frac{n}{2}) = NEG(\frac{n}{2}) = 0$ (3)

The name of the block we have designed to perform above equations is EN_2. The input of the block is b_{n-1}, the outputs are $\frac{n}{2}th$ values of ADD, SUB, ZERO, ADD×2, SUB×2 and NEG vectors. There is one INV gate in this block, so the area is 0.5 unit.

If n is odd, because of b_n is always "0" we have simplified (1) to calculate $\frac{n-1}{2}th$ values of the output vectors of encoder circuit as follows.
$ZERO(\frac{n-1}{2}) = \overline{(b_{n-1} + b_{n-2})},\ ADD(\frac{n-1}{2}) = (b_{n-1} \oplus b_{n-2}),\ ADD \times 2(\frac{n-1}{2}) = b_{n-1}b_{n-2},$
$SUB(\frac{n-1}{2}) = SUB \times 2(\frac{n-1}{2}) = NEG(\frac{n-1}{2}) = 0$ (4)

The name of the block we have designed to perform above equations is EN_3. The inputs of the block are b_{n-1} and b_{n-2}, the outputs of the block are $\frac{n-1}{2}th$ values of ADD, ZERO and ADDx2 ADD, SUB, ZERO, ADD×2, SUB×2 and NEG vectors. There are one AND2 gate, one NOR2 gate and one XOR gate in this block, so the area is 4.5 units.

If n is even one EN_0, $\frac{n}{2} - 1$ EN_1 and one EN_2 blocks are used while writing the VHDL model of the encoder circuit. So the area of the encoder circuit is calculated according to the following equation.
$6.5 + 17.5 \cdot \left(\frac{n}{2} - 1\right) + 0.5 = 8.75n - 10.5$ (5)

If n is odd, one EN_0, $\frac{n-1}{2} - 1$ EN_1 and one EN_3 blocks are used while writing the VHDL model of the encoder circuit. So the area of the encoder circuit is calculated according to the following equation.
$6.5 + 17.5 \cdot \left(\frac{n-1}{2} - 1\right) + 4.5 = 8.75n - 15.25$ (6)

As all the values of the output vectors are calculated in parallel, so the longest path is the same as the longest path of EN_1 for both of the cases.

Fig. 2 shows a multiplexer for a single bit position of the multiplicand. It consists of five three-state buffers (TSBs), which use the five control signals generated by a Booth encoder, to select the appropriate bit of the multiplicand or its inverse. Subtraction is performed using two's complement addition. This involves adding the inverse of the multiplicand and a further "1" at the least significant bit (LSB) position for SUB and at the next most significant bit for SUB×2. The extra "1", the subtract flag, is generated by the Booth encoding logic in NEG vector. Because all the bits of partial products are generated parallel the longest path of the multiplexer circuit is only one TSB long. For one partial product the area of the multiplexer is $5(n + 1)$ units. If n is even there are $\frac{n}{2} + 1$ partial products, so the total area of multiplexer circuit is $5(n + 1)(\frac{n}{2} + 1)$ units. If n is odd there are $\frac{n-1}{2} + 1$ partial products, so the total area of multiplexer circuit is $5(n + 1)(\frac{n-1}{2} + 1)$ units.

As shown in Fig. 1 addition of partial products are done by a Wallace tree formed by 4:2 compressors. The structure of 4:2 compressor is shown in Fig. 3 [4]. The area of a 4:2 compressor cell is 21 units and there are two XOR gates on the longest path of the cell.

If n is even the number of levels on the adder tree is $\lceil log_2\left(\frac{n+2}{4}\right)\rceil$. If n is odd the number of levels on the adder tree is $\lceil log_2\left(\frac{n+1}{4}\right)\rceil$. Because all the additions are done in parallel on one level of the tree, the longest path of one level is the same as a 4:2 compressor's longest path. So the latency of the adder tree is proportional to $\lceil log_2\left(\frac{n}{4}\right)\rceil$.

Total area of the adder tree according to the different values of n is shown in Table 1. If n is even $A = \frac{n+2}{2}$, If n is odd $A = \frac{n+1}{2}$.

The sums and carries from the final row of the adders in the carry save array are combined in a CLA to form the result. The latency of CLA is proportional to $log_2 n$ and the area of CLA is proportional to n. The area of multiplier circuit is

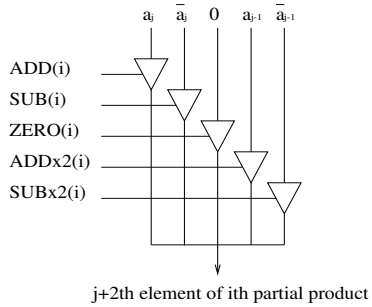

Figure 4: Multiplexer for $j + 2th$ element of ith partial product

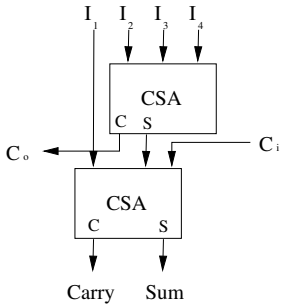

Figure 5: The structure of 4:2 compressor (CSA: Carry Save Adder)

proportional to n^2 and the latency of the circuit is proportional to $log_2 \frac{n}{4}$.

2.3 High Speed Multiplier Using Redundant Binary Representation

The structure of high speed multiplier using redundant binary (RB) representation for 24×24 bit is shown in Fig. 4 [5].

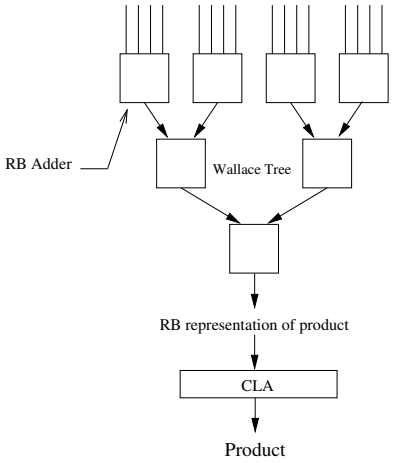

Figure 6: The structure of high speed multiplier using RB representation for 24×24 bit

The inputs of encoder circuit are five bit sequences of the multiplier and the outputs are the control signals which are used for selecting the appropriate value of multiplicand as partial product. $\frac{n}{4}$ partial products are produced by this way. Because the partial products are represented in RB representation, we have written the model as to store a partial product in two vectors, PP_+ and PP_-. ith bit of kth partial product is calculated according to the following equations. The truth table for PE_k and PO_k associated to kth group of the multiplier and multiplicand, A, is given in Table 2 and Table 3.

$$PP_{k+}(i) = PE_k(i) \cdot \overline{PO_k(i)},\ PP_{k-}(i) = \overline{PE_k(i)} \cdot PO_k(i) \quad (7)$$

If n is a multiple of four $\frac{n+4}{4}$, if n is one more than a multiple of four $\frac{n+3}{4}$, if n is two more than a multiple of four $\frac{n+2}{4}$ and if n is three more than a multiple of four $\frac{n+1}{4}$ partial products are generated. Booth encoders and Booth multiplexers

Table 1: If n is even, the area of the adder tree

n	Area of the adder tree
$2 \leq n \leq 7$	$21n + 168$
$8 \leq n \leq 15$	$(21n + 168) \left\lfloor \frac{A}{4} \right\rfloor + 21n + 336$
$16 \leq n \leq 31$	$(21n + 168) \left\lfloor \frac{A}{4} \right\rfloor + (21n + 336) \left\lfloor \frac{A}{8} \right\rfloor + 21n + 504$
$32 \leq n \leq 54$	$(21n + 168) \left\lfloor \frac{A}{4} \right\rfloor + (21n + 336) \left\lfloor \frac{A}{8} \right\rfloor + (21n + 504) \left\lfloor \frac{A}{16} \right\rfloor + 21n + 630$

Table 2: The truth table of PE_k

b_{4k+1}	b_{4k}	b_{4k-1}	PE_k
0	0	0	0
0	0	1	A
0	1	0	A
0	1	1	2A
1	0	0	-2A
1	0	1	-A
1	1	0	-A
1	1	1	0

mentioned in Section 2.2 are used for producing $n + 1$-bit PE_k and PO_k vectors. Number of the blocks used in encoder for different values of n are shown in Table 4. The total area value of the encoder and multiplexer circuits can be calculated according to the number of blocks used and the area values of them. These results are shown in Table 5.

PE and PO vectors are produced by encoder and multiplexer circuits. We have produced "+" and "$-$" vectors of partial products by using these vectors according to (7). There are two AND2 gates and two INV gates in the circuits designed to perform (7). So the area is $4(n + 4)$ units more for every partial product.

Additions are performed by a tree formed by redundant binary adders (RBAs) as shown in Fig. 4. Addition of two RB numbers is performed in a constant time. Therefore, the computation at each level in the tree is performed in a constant time independent of n. A RBA cell is designed to perform the following equations [6].

$$d_i^+ = \overline{\alpha_i} \cdot \beta_{i-1},\ d_i^- = \alpha_i \cdot \overline{\beta_{i-1}},\ \alpha_i = \ell_i \oplus h_{i-1},\ \beta_i = \overline{\ell_i} \cdot k_i + \ell_i \cdot \overline{h_{i-1}},\ \ell_i = (a_i^+ + a_i^-) \oplus (b_i^+ + b_i^-),$$
$$h_i = a_i^- + b_i^-,\ k_i = a_i^+ + b_i^+ \quad (8)$$

The cell designed to perform the above equations is used for the addition of two digits of two RB numbers that are not least significant bit (LSB). There are five two input OR (OR2) gates, four AND2 gates, four INV gates and two XOR gates in the cell. The area of the cell is 19.5 units. There are two AND2 gates, one XOR gate, two INV gates and two OR2 gates on the longest path. The β and h used in the addition of LSB of the partial products are "0". Then we have simplified (8) as follows.

$$d_0^+ = 0,\ d_0^- = \ell_0,\ \beta_0 = k_0 + \ell_0,\ \ell_0 = (a_0^+ + a_0^-) \oplus (b_0^+ + b_0^-),\ h_0 = a_0^- + b_0^-,\ k_0 = a_0^+ + b_0^+ \quad (9)$$

There are five OR2 gates and one XOR gate in the circuit designed for addition of LSB of two partial products. The area of the circuit is 9.5 units instead of 19.5 units. After adding the MSBs of two partial products there is one more addition operation that is performed according to the following equations. Because of $a_n^- = b_n^- = a_n^+ = b_n^+ = 0$ we have simplified (8) for this situation also.

$$d_n^+ = \overline{h_{n-1}} \cdot \beta_{n-1},\ d_n^- = h_{n-1} \cdot \overline{\beta_{n-1}} \quad (10)$$

There are two INV gates and two AND2 gates in the circuit designed to perform above equations, so the area of the circuit is 4 units instead of 19.5 units.

If n is a multiple of four there are $\lceil log_2 \frac{n+4}{4} \rceil$, if n is one more than a multiple of four $\lceil log_2 \frac{n+3}{4} \rceil$, if n is two more than a multiple of four $\lceil log_2 \frac{n+2}{4} \rceil$ and if n is three more than a multiple of four $\lceil log_2 \frac{n+1}{4} \rceil$ levels on the adder tree. Because additions in one level of the tree is performed in parallel, the longest path of one level is the same as one RBA.

Total area of the addition tree according to the value of n is given in Table 6.

In Table 6, if n is a multiple of four $A = \frac{n+4}{4}$, if n is one more than a multiple of four $A = \frac{n+3}{4}$, if n is two more than a multiple of four $A = \frac{n+2}{4}$ and if n is three more than a multiple of four $A = \frac{n+1}{4}$.

After completing the addition of partial products, RB representation of the product, S_+ and S_-, is produced. It must be converted from RB representation to binary representation to find the result of the multiplication. The convertion is performed by subtracting S_- from S_+ by CLA. The computation time is proportional to $log_2 n$ and the gate count of CLA is proportional to n. Total area of the circuit is proportional to n^2 and the latency of the circuit is proportional to $log_2 \frac{n}{4}$.

Area values of the multiplication blocks depending on different values of n are given in Table 7. The gate count on the

Table 3: The truth table of PO_k

b_{4k+3}	b_{4k+2}	b_{4k+1}	PO_k
0	0	0	0
0	0	1	-A·2^2
0	1	0	-A·2^2
0	1	1	-2A·2^2
1	0	0	2A·2^2
1	0	1	A·2^2
1	1	0	A·2^2
1	1	1	0

Table 4: Number of blocks used in encoder for different values of n

n	EN_0	EN_1	EN_2	EN_3
a multiple of four	1	$1 + 2 \cdot \frac{n-4}{4}$	1	
one more than a multiple of four	1	$1 + 2 \cdot \frac{n-5}{4}$		1
two more than a multiple of four	1	$2 + 2 \cdot \frac{n-6}{4}$	1	
three more than a multiple of four	1	$2 + 2 \cdot \frac{n-7}{4}$		1

longest path of the multiplication blocks are given in Table 8. A DSP chip designer can decide to use which of the five multiplication blocks according to Table 7 and Table 8.

3 Conclusions

In this paper, we propose two different multiplication blocks for DSP applications by using VHDL. We have written the models by taking the number of multiplier's and multiplicand's digits n as a parameter. The models are suitable for all of the multiplications from 2×2 bits to 54×54 bits. We have simplified some equations given in literature, which decreased the area of the corresponding circuit. We propose formulas to calculate the area values and the gate count of the longest path of both of the two blocks. Also we have indicated the types of gates on the longest paths of the circuit. By using the formulas we have calculated the critical values for both of the two circuits for different n values and we have given the results in Table 7 and Table 8. The DSP chip designers can choose the multiplication blocks which meet the requirements by these tables.

The VHDL models are available on the URL address: http://www.titan.ehb.itu.edu.tr/~berna.

References

[1] J., Mori, M., Nagamatsu, M., Hitano, S., Tanaka, M., Noda, Y., Toyoshima, K., Hashimoto, H., Hayashida, K., Maeguchi, "A 10-ns 54x54-b Parallel structured full array multiplier with 0.5-μ CMOS technology", IEEE J. Solid-State Circuits, vol. 26, no. 4, pp. 600-606, April 1991.

[2] A. R., Cooper, "Parallel architecture modified Booth multiplier", IEE Proceedings, vol. 135, Pt. G, no. 3, pp. 125-128, June 1988.

[3] N. R., Shanbhag, P., Juneja, "Parallel Implementation of a 4x4-bit Multiplier Using Modified Booth's Algorithm", IEEE J. Solid-State Circuits, vol. 23, no. 4, August 1988.

[4] M. R., Santoro, "SPIM: A pipelined 64×64-bit iterative multiplier", IEEE J. Solid-State Circuits, vol. 24, no. 2, pp. 487-493, April 1989.

[5] S., Kuninobu, T., Nishiyama, T., Taniguchi, "High speed MOS multiplier and divider using redundant binary representation and their implementation in a microprocessor", IEICE Trans. Electron., vol. E76-C, no. 3, pp. 436-445, Mar. 1993.

[6] H., Makino, Y., Nakase, H., Suzuki, H., Morinaka, H., Shinohara, K., Mashiko, "An 8.8-ns 54×54-Bit Multiplier with High Speed Redundant Binary Architecture", IEEE Jour. of Solid-State Cir., vol. 31, no. 6, pp. 773-783, June 1996.

Table 5: The total area value of the encoder and multiplexer circuits depending on the value of n

n	Area
a multiple of four	$7 + 17.5 \cdot \left(\frac{2n-4}{4}\right) + 5(n+1)\left(\frac{n+4}{4}\right)$
one more than a multiple of four	$11 + 17.5 \cdot \left(\frac{2n-6}{4}\right) + 5(n+1)\left(\frac{n+3}{4}\right)$
two more than a multiple of four	$7 + 17.5 \cdot \left(\frac{2n-4}{4}\right) + 5(n+1)\left(\frac{n+2}{4}\right)$
three more than a multiple of four	$7 + 17.5 \cdot \left(\frac{2n-6}{4}\right) + 5(n+1)\left(\frac{n+1}{4}\right)$

Table 6: Total area of the addition tree according to the value of n

n	Area
$4 \leq n \leq 7$	$19.5n + 208$
$8 \leq n \leq 15$	$(19.5n + 208) \cdot \left\lfloor \frac{A}{2} \right\rfloor + 19.5n + 364.5$
$16 \leq n \leq 31$	$(19.5n + 208) \cdot \left\lfloor \frac{A}{2} \right\rfloor + (19.5n + 364.5) \left\lfloor \frac{A}{4} \right\rfloor + 19.5n + 676.5$
$32 \leq n \leq 54$	$(19.5n + 208) \cdot \left\lfloor \frac{A}{2} \right\rfloor + (19.5n + 364.5) \left\lfloor \frac{A}{4} \right\rfloor + (19.5n + 676.5) \left\lfloor \frac{A}{8} \right\rfloor + 19.5n + 1066.5$

Table 7: Area values of multiplication blocks depending on different n values

n	1^{st} Circuit	2^{nd} Circuit
4	351	424
8	1124	1223
12	1557	2093
16	3414	3659
20	4175	4985
24	5688	5863
28	6693	8478
32	10928	11346
36	12345	13584
40	14850	15062
44	16511	18901
48	20772	20679
52	22761	23751

Table 8: Gate counts on the longest paths of multiplication blocks depending on the value of n

n	1^{st} Circuit	2^{nd} Circuit
4	AND2+TSB+3 XOR	3 AND2+TSB+2 XOR+3 INV+2 OR2
8	AND2+TSB+5 XOR	5 AND2+TSB+3 XOR+5 INV+4 OR2
12	AND2+TSB+5 XOR	6 AND2+TSB+3 XOR+5 INV+4 OR2
16	AND2+TSB+7 XOR	8 AND2+TSB+4 XOR+7 INV+6 OR2
20	AND2+TSB+7 XOR	8 AND2+TSB+4 XOR+7 INV+6 OR2
24	AND2+TSB+7 XOR	8 AND2+TSB+4 XOR+7 INV+6 OR2
28	AND2+TSB+7 XOR	8 AND2+TSB+4 XOR+7 INV+6 OR2
32	AND2+TSB+9 XOR	10 AND2+TSB+5 XOR+9 INV+8 OR2
36	AND2+TSB+9 XOR	10 AND2+TSB+5 XOR+9 INV+8 OR2
40	AND2+TSB+9 XOR	10 AND2+TSB+5 XOR+9 INV+8 OR2
44	AND2+TSB+9 XOR	10 AND2+TSB+5 XOR+9 INV+8 OR2
48	AND2+TSB+9 XOR	10 AND2+TSB+5 XOR+9 INV+8 OR2
52	AND2+TSB+9 XOR	10 AND2+TSB+5 XOR+9 INV+8 OR2