Self-timed multiplier for multiply-add unit *

B. Stepanov, Y. Diachenko, Y. Rogdestvenski, D. Diachenko
Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences,
Institute of Informatics Problems, IIP RAS,
Moscow, Russian Federation
gtx360@email.ru, diaura@mail.ru

***Abstract* — Paper discusses the peculiarities of self-timed multiplier implementation for unit multiplying two operands and then adding the product to third operand without an intermediate rounding according to the IEEE 754 Standard. The multiplier is a hardware implementation of modified Booth algorithm on a base of self-timed adder with redundant signal code. An optimal self-timed redundant coding of internal and output signals in the multiplier was proposed. The circuitry and layout problems were solved for self-timed multiplier implementation. Wallace tree structure, which is the main part of the multiplier, was optimized for the facilities of 65-nm CMOS process with six metal layers taking into account more number of signals in the multiplier circuit, than in the synchronous analog. A release of the self-timed multiplier implementation in CMOS process with 65-nm design rules is introduced.**

Keywords — Booth algorithm; self-timed multiplier; ternary coding; ternary Wallace tree; layout

I. INTRODUCTION

In modern supercomputers, less than 35% of total processing power is spent on the direct calculations. Remaining resources are spent on providing validity of the calculation results. At that, their mean time between failures equals about 55 hours (for supercomputer with 100 000 processor cores in 2013 year) [1]. Further increasing performance of the supercomputers will lead to possible degradation of the mean time between failures and computation reliability. A solution of this problem today is possible at the expense of utilizing hardware control methods of the reliability and self-repair of the computer aids in a real time on a base of a self-timed (ST) circuitry, which is sufficiently developed and corresponds to laid down requirements. Researches had proved [2] that что correctly designed ST circuits have, as a rule, better performance and power consumption in comparison with synchronous analogues.

In contrast with synchronous circuits, ST circuitry utilizes request-acknowledge discipline for unit interfacing, ST coding for data signals, and advanced system indicating computational process evolution. Therefore, an appearance of any constant failure in any wire of such circuit leads to a termination of the calculations at appropriate indicator and immediate location of the problem. This allows for providing hardware resource and continuing correct operation practically in real time for entire unit (self-repair).

Research was performed with financial support of the Russian Foundation for Basic Research (project № 13-07-12068 ofi_m).

One of the most meaningful functional units of the modern computers is multiply-add unit performing two sequential operations: multiplying two operands and adding product to third operand, - without rounding intermediate results (Fused Multiply-Add, FMA). This provides more performance and better computational accuracy.

Multiplier is the most complicate functional unit of the FMA. It determines FMA's consumer characteristics. Therefore, a development of its optimal ST implementation is an actual problem.

Paper is devoted to the problems of designing and realizing ST multiplier for FMA unit operating in accordance with IEEE 754 Standard. It processes either three double precision operands, or six single precision operands.

II. ST MULTIPLIER

Synchronous multiplier implementation is studied sufficiently well. With relation to complexity and performance, the best solution is to use modified Booth algorithm. Such multiplier consists of Booth coder, partial products generator, and partial products adder (Wallace tree) as Fig. 1 demonstrates. Sown width of the operands corresponds to double precision case of IEEE 754.

In many respects, circuitry realization of multiplier and its parameters depend on a type of coding input operands and intermediate results.

A. ST signal coding in the multiplier

We choose an algorithmic solution given in paper [3] for synchronous Wallace tree implementation as a prototype. An essence of that method lies in a redundant coding of the Wallace tree summands shown in Table I.

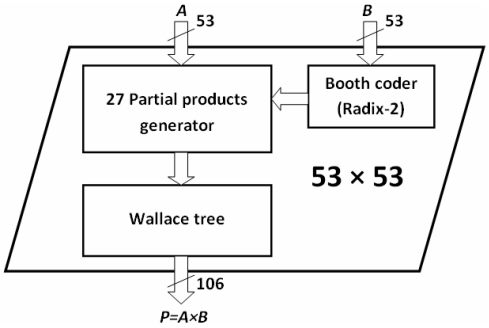


Fig. 1. Flow chart of the multiplier 53×53

TABLE I. SYNCHRONOUS BINARY CODE

Coded state	Binary code	
	A	***B***
+1	1	0
0	0	0
–1	0	1
unused	1	1

Such redundancy allows for representing a sum of two one-bit numbers in a direct and complement code by means of one code number, and for adding in carry save mode in each code bit. This adding manner provides compression ratio as 4:1 at first stage of the Wallace tree, and 2:1 at the following its stages, providing the fastest adding with minimal hardware redundancy.

An attempt to use a paraphase (dual-rail) code for each binary bit of this representation, as it is made at converting synchronous circuits to ST one, leads to sufficient degradation of the algorithm's efficiency and to large hardware redundancy. As a result of performed analysis of the possible ST codes, a ST code shown in Table II was offered in paper [4].

Fig. 2 demonstrates the Wallace tree implementation for 53-bit numbers with redundant ST coding of the intermediate and output results. Here PP1 – PP27 are the partial products in a dual-rail code, CS is a correcting summand in a dual-rail code, while internal and output signals are represented in the ST ternary code. Due to usage of the procedure of each second partial product conversion suggested in [3], first stage of the Wallace tree compresses 27 partial products and correcting summand into 7 ternary sums.

For reference, Fig.3 shows an appropriate ST realization of the Wallace tree with dual-rail coding. Here CI1– CI24 are the input carries from a previous Wallace tree bit, CO1– CO24 are the output carries into a next bit. All signals have dual-rail code.

Comparing Fig. 2 and Fig. 3 shows that a number of the stages of the ST ternary Wallace tree is less than stage number in the dual-rail Wallace tree by factor of 1.75. As a result, a performance of the ternary ST multiplier is higher than that of classical ST dual-rail multiplier by 20%.

TABLE II. ST TERNARY CODE

Coded state	Ternary code		
	Ap	***Am***	***An***
+1	1	0	0
0	0	0	1
–1	0	1	0
spacer	0	0	0

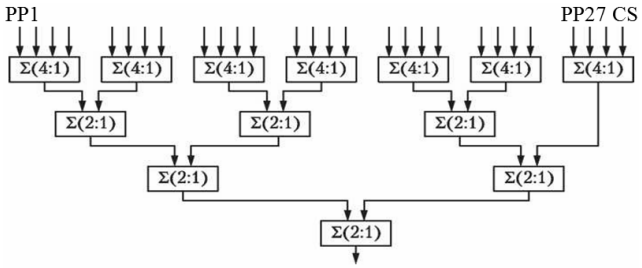


Fig. 2. Ternary ST implementation of the Wallace tree

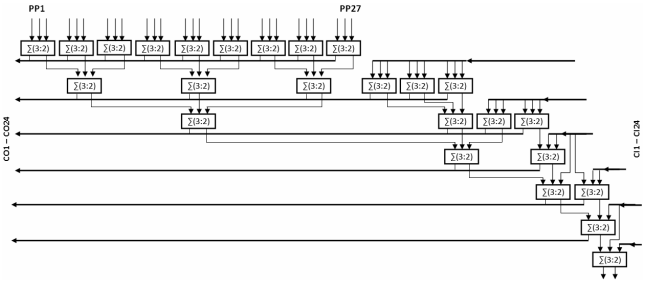


Fig. 3. Dual-rail ST implementation of the Wallace tree

B. Circuitry of the ST multiplier

One can separate three parts in the combinational multiplier implementing modified Booth algorithm shown in Fig. 1: Booth coder, partial products generator, and Wallace tree. Major hardware complexity of the ST multiplier comes from partial products adder implemented as Wallace tree, which is based on one-bit adders. Figures 4 and 5 demonstrate the circuits of one-bit adder with dual-rail and ternary pins correspondingly. A subcircuit of indication of the intermediate and output signals of the adder occupies a right side of both circuits. It is marked by a dash-dot oval.

Ternary ST adder for first stage of the Wallace tree adds four dual-rail summands. Before adding, it converts each pare of the summands to one ternary operand. Fig. 6 demonstrates a circuit converting two dual-rail signals to one ternary signal.

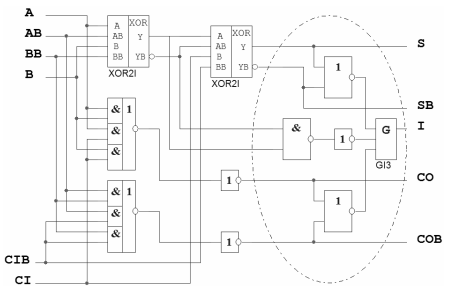


Fig. 4. Dual-rail ST adder

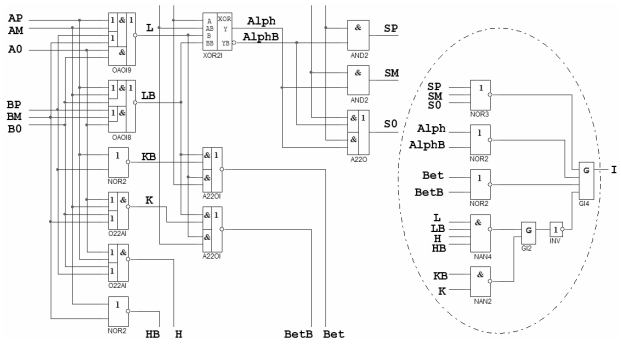


Fig. 5. Ternary ST adder

Comparative analysis shows that ST adder with dual-rail pins has less complexity (by factor of 2) than the ST ternary one (in CMOS basis they have 78 and 158 transistors

correspondingly). The circuit converting two dual-rail signals to one ternary signal complicates the ternary ST adder for first stage of the Wallace tree by 30 transistors additionally.

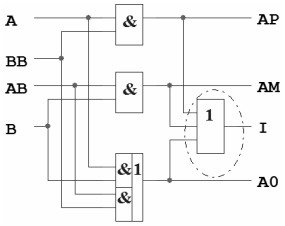


Fig. 6. Dual-rail pair to ternary signal converter

However, the ternary ST Wallace tree built is more complex than dual-rail ST Wallace tree only by factor of 1.1 (2470 transistors versus 2260 ones correspondingly). Higher by 20% performance of the ternary ST Wallace tree compensates this disadvantage.

Layout realization significantly affects ST multiplier's performance.

III. Layout of the Multiplier

ST circuits utilize redundant coding of the data. Executed logical functions are dualized during converting synchronous combinational prototype circuit to ST implementation with dual-rail signals. For each logical function, its inverse representation is introduced. Besides, indication subcircuit is added. As a result, complexity of a combinational circuit rises by factor of 2.3 – 2.5. Signal quantity in the circuit increases in the same proportion.

Therefore, at layout design of the combinational ST circuits including ST multiplier, one should beforehand choose data flow direction in the layout to prevent an excessive depression of the layout because of the necessity to free a space for routing signal wires. The following criteria are a base of this alternative:

- a complexity of the functional basis for implementing designed circuit,
- degree of connectivity of the circuit cells within one bit and in different bits,
- restrictions on size of the circuit layout,
- a number of routing layers.

Modern CADs for CMOS VLSI utilize standard cell libraries with ready layouts for all library cells. A height of the cell's layout equals to the height of one row, while cell's length depends on its complexity. Complex cells (one-bit adders, triggers an so on) are drawn across, and they are more transparent for vertical routing.

Besides, at even number of the available routing layers and their traditional usage (even layers for vertical routing, odd layers for horizontal one), a number of the vertical routes appears to be larger. Therefore, at layout design of the multi-bit ST units with strongly coupled bits, as a rule, it is expedient to use the structures with vertical signal propagation through layout. In this case, a multitude of the connections between bits can aggravate a problem of the layout's transparency for routing global wires.

Ternary coding of the operands in ST multiplier alleviates this problem. First, Wallace tree is a composition of the carry save adders. At ternary coding of the operands, each adder has 3 outputs, while dual-rail carry save adders have 4 outputs. Second, inter-bit carries in the ternary ST adder are transferred into adjacent bit in the same stage of the Wallace tree, so they do not occupy any vertical routes.

The following requirements were taken into account for designing layout of the ST multiplier:

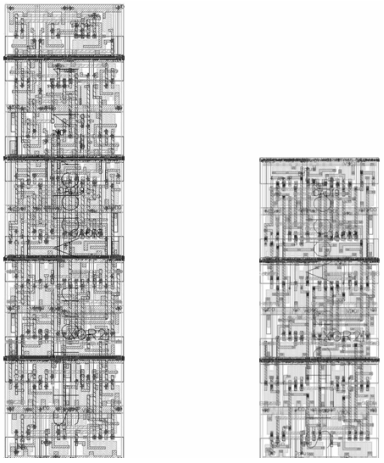
- multiplicands are either two 53-bit operands, or two pairs of 24-bit operands,
- horizontal size of the layout should not exceed 500 μm,
- some vertical routes are reserved for global wires,
- metal layers M1 through M6 are used for routing.

Hence, a size of one-bit ternary adder layout was chosen.

A. Layout of the 1-bit ternary adder

Fig. 7 demonstrates a layout of the 1-bit ternary adder for first stage of the ST Wallace tree with compression ratio 4:1 (Fig. 7(a)), and 1-bit adder layout for the following stages of the Wallace tree with compression ratio 2:1 (Fig. 7(b)). Here inputs occupy top side, outputs are situated at down side, and inter-bit carries are located at the lateral sides. This makes easier joining adjacent 1-bit adders and provides data propagation at vertical direction.

In standard CMOS 65-nm process the layout of 1-bit adder for first stage of the ST Wallace tree with compression ratio 4:1 (adds 4 dual-rail operands) has a size of 4.56 × 18.0 μm, while the layout of 1-bit adder with compression ratio 2:1 (adds 2 ternary operands) has a size of 4.56 × 12.0 μm. Their routing uses only M1 through M3 metal layers. This provides their transparency for routing global wires in M4 – M6 metal layers.



(a) (b)

Fig. 7. Layout of the 1-bit ternary adder 4:1 (a) and 2:1 (b)

B. *Layout of the multiplier*

Analysis of the circuits of the Booth coder and partial products generator, and their output signals entering the Wallace tree has shown that total number of the Wallace tree inputs (more than 3000) exceeds a number of free vertical routes at given restriction for horizontal size of the designed multiplier. Therefore, we have decided to integrate the Booth coder and partial product generator into Wallace tree structure, as Fig. 8 shows. Block "Generator & Σ" implements generator and adder of the partial products specified in the parenthesis. Block "Σ" adds the partial products specified in the parenthesis.

Fig. 9 shows a layout of the whole multiplier. Its size is 470 × 430 μm. Adders in up part of the multiplier have larger width, as they should be capable to make simultaneous multiplication of two pair of 24-bit single precision operands. In this case, it is sufficient to add the partial products PP1 – PP13 and correcting summand CS to obtain two single precision products. Adding partial products PP14 – PP27 is needed only for multiplying double precision 53-bit operands. Therefore, they are shorter as one can see in Fig. 9, which also demonstrates denser horizontal routing in the partial products generators. It actively utilizes fifth metal layer. Partial product adders do not use fifth metal layer for routing.

Indication signals at output of the multiplier provide it bitwise indication.

ST multiplier was designed as a part of the FMA unit. Now we are builting it into layout of entire FMA unit.

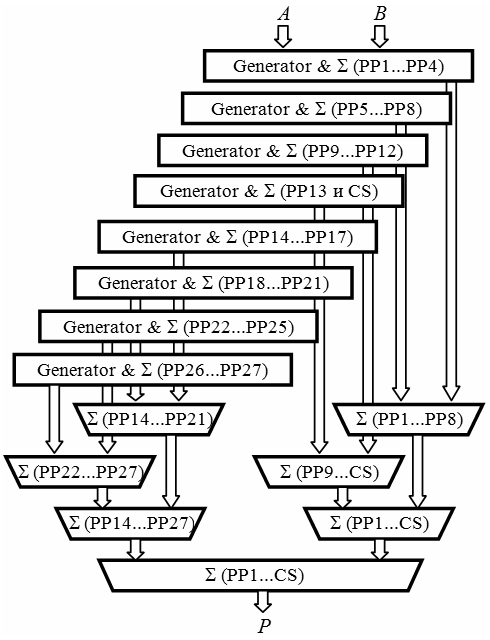


Fig. 8. Flow chart of calculations in the multiplier

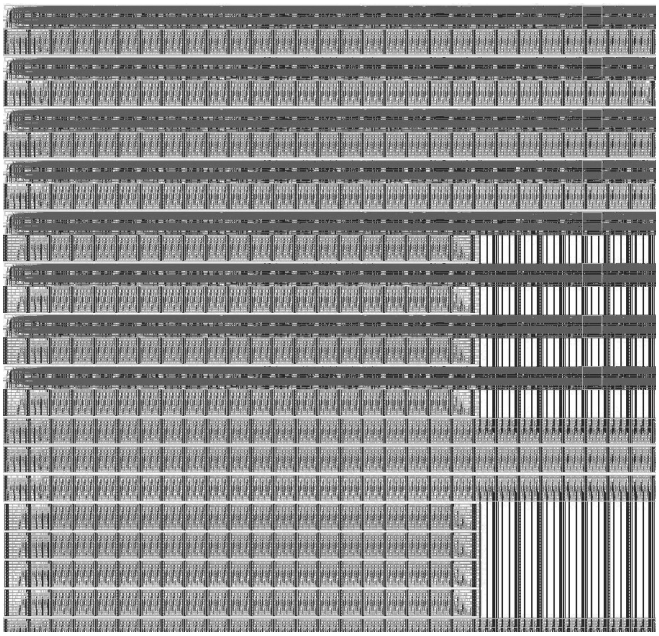


Fig. 9. Layout of the multiplier

IV. Conclusions

ST circuits have the evident advantages in comparison with synchronous analogs. They demonstrate much wider workability range on supply voltage and temperature, detect, and locate constant failures, allowing circuit self-repair organization in real time.

Developed ST multiplier utilizes ternary ST-coding. It has allowed for reducing number of stages of the Wallace tree from 7 down to 4 and for rising multiplier's performance by 20% due to penalty of multiplier's by 10%.

Layout of the main blocks of the multiplier was made manually to achieve maximal density and decrease its size. In CMOS 65-nm process with 6 metal layers, its size equals to 470 × 430 μm.

References

[1] Yu. Semenov. (2013) Supercomputers and Watson. [Online]. Available: http://book.itep.ru/10/supercomp.htm.

[2] Y. Stepchenkov, Y. Diachenko, and G. Gorelkin, "Self-timed circuits are a future of microelectronics", Radioelectronic questions, CSRI "Electronics", Moscow, 2011, no. 2, pp. 153-184 (in Russian).

[3] H. Makino, Y. Nakase, H. Suzuki, H. Morinaka, H. Shinohara, and K. Mashiko, "An 8.8 ns 54x54 bit Multiplier with High Speed Redundant Binary Architecture", IEEE Journal of Solid-State Circuits, 1996, vol. 31, no. 6, pp. 773-783.

[4] I. Sokolov, Y. Stepchenkov, S. Bobkov, Y. Rogdestvenski, Y. Diachenko, "Multiplier with accumulation: methdological aspects", Systems and means of informatics, Moscow, 2014, vol. 24, no. 3, pp. 44-62 (in Russian).