INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 3, Issue 4 OCT 2014

VLSI IMPLEMENTATION OF N×M-BIT RSFQ MULTIPLIER USING HDL

CH.VINAY KUMAR(1),N.KALYANI(2)
M.TECH SCHOLAR, VLSI AND EMBEDDED SYSTEM DESIGN, PRASAD ENGINEERING COLLEGE(1)
M.TECH, ASSOC.PROFESSOR, , PRASAD ENGINEERING COLLEGE (2)

ABSTRACT:

We have developed and experimentally evaluated at high-speed a complete set of arithmetic circuits (multiply, add, and accumulate) for high performance digital signal processing (DSP). These circuits take advantage of the unique features of the Rapid Single-Flux Quantum (RSFQ) logic/memory family, including fusion of logic and memory functions at the gate level, pulse representation of clock and data, and the ability to maintain inter cell propagation delays using Josephson transmission lines (JTLs). The circuits developed have been successfully used in the implementation of a serial radix 2 butterfly, a decimation digital filter, and of an arithmetic unit for digital beam forming. The 8×8-bit RSFQ multiplier uses a two-level parallel carry-save reduction tree that significantly reduces the multiplier latency. The 80-GHz carry-save reduction is implemented with asynchronous data-driven wave-pipelined [4:2] compressors built with toggle flip-flop cells. The design has mostly regular layout with both local and global connections between modules. The multiplier core (without SFQ-to-DC and DC-to-SFQ converters) has 5948 Josephson junctions occupying the area of 3.5 mm2. The multiplier is designed with the target operation frequency of 20 GHz and has the latency of 447 ps at the bias voltage of 2.5 mV.

Keywords—Carry select adder, Modified adder, High performance computing, multiplying circuits.

I. INTRODUCTION

Decimal arithmetic received an increased attention in the last few decades because of its growing need in many commercial applications and database systems where the binary arithmetic is not sufficient. The arithmetic operations in these applications need to be executed in decimal format. Decimal multiplication is one of the most frequently used by many financial, business applications but current implementations in FPGAs are very inefficient in terms of both area and latency when compared to binary multipliers. An important and frequent operation in many applications is decimal multiplication. There are several works which focus on fixed-point multiplication. Designed decimal multiplier with the various carry save adder for the addition of the partial product generation. The generated partial product is compressed using the carry save adder. In this paper, design of multipliers is presented, using Carry Select Adder with BEC logic for addition of partial product terms. The use of modified carry select adder for the partial product reduction has several advantages such as reduction in the area; multiplication operation can be performed faster using the MCSA compared with the existing system.

In this paper, we discuss the micro architecture, design, and testing of the first 8×8-bit (by modulo 256) parallel carry-save
RSFQ multiplier implemented using the ISTEC 10-kA/cm^2 1.0-μm fabrication technology.

A. Why Superconductive DSP Now

Digital superconductive electronics can be inserted into digital signal processing systems sooner than into any other digital application. Superconductivity can offer solutions for a number of DSP problems which cannot be solved by semiconductor electronics. One of the most important among these is the ever-increasing demand on circuit throughput. Another is low-power processing, which is essential for many critical DSP applications with cooled front-end devices. Furthermore, future applications of superconductor analog-to-digital converters (ADCs) with performance unattainable by semiconductor counterparts will inevitably make the use of superconductor DSPs even more advantageous.

B.RSFQ-Based DSP: Merits and Challenges

Superconductive RSFQ digital technology possesses a number of features making RSFQ-based DSP implementations extremely attractive

.

I) High Throughput

The high throughput capability of RSFQ circuits is especially valuable for DSP implementations. Since this capability is even greater than currently required for some applications, it allows us to use a single bit-wide serial processing architecture. This enables the required functional complexity with much less hardware, thus bringing it within reach of present-day superconductive technology. With an increasing maturity level, parallel DSP implementations will be used when higher throughput is necessary.

2) Low Power

Extremely low-power dissipation enables high-density, compact packaging at the chip level. Low-level SFQ data and clock signals allow us to avoid any detrimental digital noise effects in mixed-signal circuits, e.g. the integrated high-sensitivity ADC front-end with decimation digital filter.

3) Internal Gate Memory

The internal memory of RSFQ gates allows the implementation of pipelined arithmetic modules using fewer gates. The basic RSFQ gates are essentially combinations of sequential and' combinational gates which perform more complex logic functions than traditional semiconductor counterparts. This results in a significant reduction of the circuit complexity.

4) Flow Clocking

The physical identity of clock and data SFQ pulses along with the synchronous nature of the majority of RSFQ gates makes distributed flow Clocking advantageous, and probably the only possible method at multi-GHz speeds. The use of JTLs provides predictable and controlled delays. For clock and data transfer between gates. SFQ flow synchronization allows us to avoid the requirement of a high-power external clock since an SFQ clock is generated on-chip.

II. DESIGNGOALS

Our 8×8-bit unsigned integer multiplier performs multiplication by modulo 256, calculating the eight least significant bits of the product.

Fig. 1 shows three major steps of multiplication of two unsigned integer operands: partial product generation, partial
product compression (reduction), and final summation.

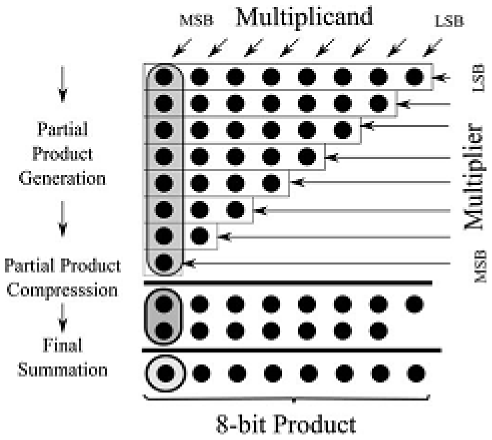

Fig. 1. Three steps of the 8×8 (by modulo 256) multiplication

When designing our 8×8-bit parallel integer multiplier, we had four major targets: high operation frequency of 20 GHz, multiplication time below 500ps, complexity around 6000 Josephson junctions (JJs), and mostly regular layout employing both local and global connections. To achieve these challenging goals, we used several advanced techniques, such as wave-pipelining, parallel partial product generation, and partial product compression with two level carry-save reduction trees built with [4:2] asynchronous wave-pipelined compressors operating at the internal "hardwired" rate of 80 GHz. Those techniques have been developed and verified by simulation during the recent work on 32×32 multipliers done at Stony Brook University (SBU) with a use of the SBU VHDL RSFQ cell library.

III. MICROARCHITECTURE AND DESIGN

Fig. 2 shows the block diagram of our 8×8-bit multiplier. The multiplier consists of three major blocks: a partial product generator, a parallel carry-save partial reduction (compression) tree, and a ripple-carry adder for final summation of carry-sum operands for the three most significant bits.

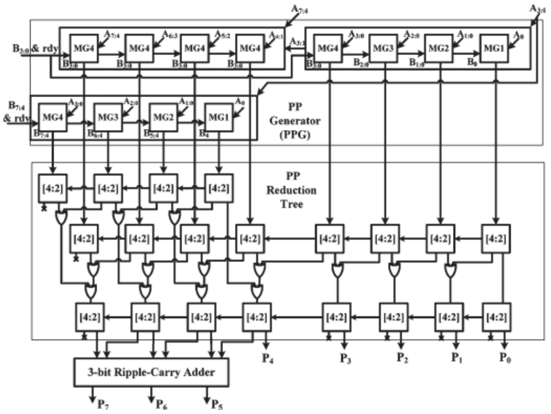

Fig. 2. 8×8-bit multiplier structural block diagram

A. Partial Product Generator with 80-GHz Output Streams

The multiplier partial product generator (PPG) consists of 36 partial product (PP) bit generators built with clocked AND gates operating on their multiplicand and multiplier bits. These circuits are organized into three PPG groups, one (top left) with 16 and two other with 10 PP generators each. PPs in each PPG group are calculated in parallel, significantly reducing the partial product generation time.

The PPG groups are implemented with four different types of modules MG1–MG4 with their indexes corresponding to the number of PPs generated by the modules.

When generated, PPs within each MG are merged together with confluence buffers (implementing asynchronous OR operations) and sent~12.5ps apart over a single passive transmission line (PTL) to their first-level [4:2] compressor. The minimum time gap of 11–12 ps between PP signal pulses is necessary to meet timing constraints and provide some DC bias margins of the confluence buffers and [4:2] compressors.

The required time separation between PPs is achieved with a use of carefully designed operand and control distribution networks utilizing JJ-based delay lines, parallel and serial signal splitting. Working in parallel, the 12 MG blocks asynchronously generate and send PPs (36 total) to the [4:2] compressors at the “hardwired” rate of 80 GHz.

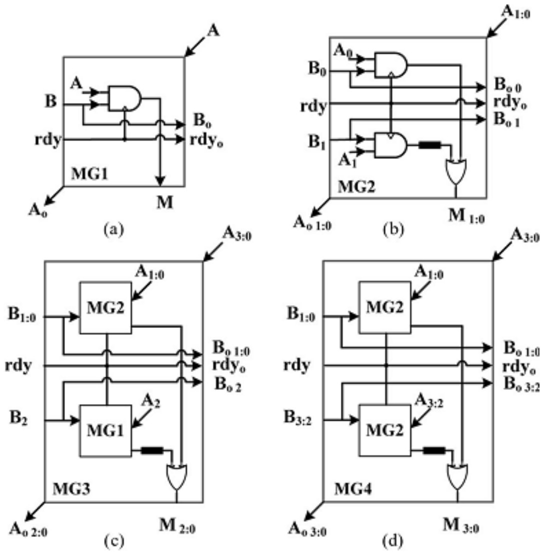

Fig. 3. PP generation modules: (a) MG1; (b) MG2; (c) MG3; (d) MG4. Dark rectangles represent JJ-based delay lines used to create 12.5ps time intervals between output PP signals called Mi

.

B. 80-GHz Partial Product Reduction and Final Summation

Adders are commonly found in the critical path of many building blocks of microprocessors and digital signal processing chips. Adders are essential not only for addition, but also for subtraction, multiplication, and division. Addition is one of the fundamental arithmetic operations. A fast and accurate operation of a digital system is greatly influenced by the performance of the resident adders. The most important for measuring the quality of adder designs in the past were propagation delay and area. For 2-bit addition the circuit consists of adder with the 3- bit Binary to excess -1 converter(BEC) the output of adder is then given as the input to the mux.

To reduce (compress) partial products in each column, we use a two-level binary carry-save reduction tree built with [4:2] compressors (see Fi are reduced to 4 by two [4:2] compressors working in parallel, each producing 2 PPs. The 4 PPs from the two first-level compressors are merged together with asynchronous confluence buffers and sent~12.5 ps apart over a single PTL to a second-level [4:2] compressor for that column. Then, the second-level [4:2] compressor will reduce those 4 PPs to 2.

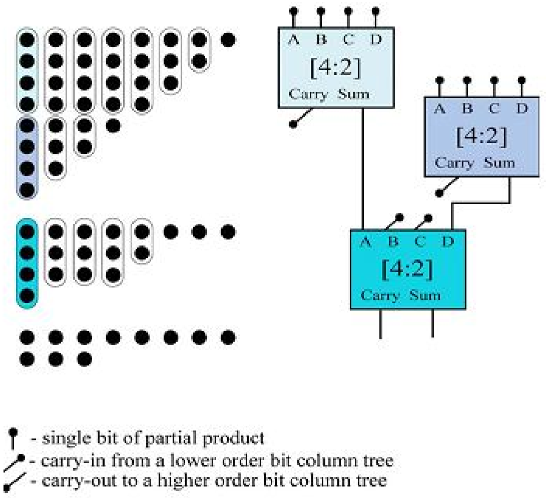
A B C D
[4:2]
Carry Sum
A B C D
[4:2]
Carry Sum
A B C D
[4:2]
Carry Sum
- single bit of partial product
- carry-in from a lower order bit column tree
- carry-out to a higher order bit column tree

Fig. 4. Two-level partial product reduction tree.

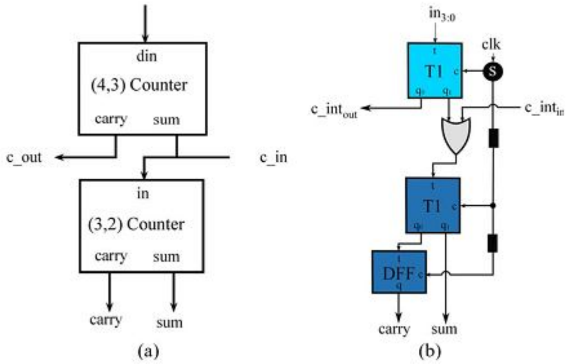

Fig. 5. [4:2] compressor: (a) conceptual diagram; (b) cell-level RSFQ implementation. Dark rectangles represent JJ-based delay lines

The [4:2] carry-save compressors shown in Fig. 5 are implemented with (4,3) and (3,2) counters playing a role of carry-save adders. Each (4,3) counter can count up to 4 PPs arriving at its input and produce two inter-column carries and one intermediate sum bit. These intermediate sum and the two carries coming out from the previous bit column are then added by a (3,2) counter producing one carry (to the next bit column) and one sum bits. The inter-column carries from the previous bit column are processed by the (3,2) counters, not g. 4). First, up to 8 PPs in each column affecting the inter-column carry signals from the (4,3) counters to the next bit column.

Both counters have very efficient hardware implementation and small PP reduction time. They are implemented with T1 (toggle flip-flop) cells that asynchronously generate upto two carry-out (one per every two input PPs received) and one clocked sum (the XOR sum of all PPs received before the clock signal arrival) output signals. The propagation of the carry signals and clocking of the T1 cells is properly tuned using JJ-based delay lines. Additional D flip-flops (DFFs), one per [4:2] compressor, are used to buffer carries from the (3, 2) counters.

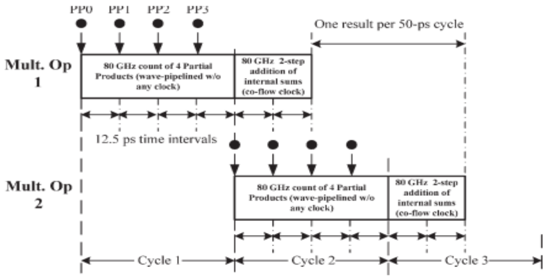
PP0
PP1
PP2
PP3
One result per 50-ps cycle
Mult. Op 1
80 GHz count of 4 Partial Products (wave-pipelined w/o any clock)
80 GHz 2-step addition of internal sums (co-flow clock)
12.5 ps time intervals
Mult. Op 2
80 GHz count of 4 Partial Products (wave-pipelined w/o any clock)
80 GHz 2-step addition of internal sums (co-flow clock)
Cycle 1
Cycle 2
Cycle 3

Fig. 6. [4:2] compressor pipeline diagram. The 50ps clock cycle time of the multiplier is determined by the time to complete four 12.5-ps micro-steps

The PP reduction pipeline diagram is shown in Fig. 6. The operation of each [4:2] compressor is asynchronously wave pipelined and data-driven by PPs coming~12.5ps apart at the internal rate of 80 GHz. It takes six 12.5-ps micro-steps (75ps) to complete the 4-to-2 reduction operation.

In each [4:2] compressor, the execution of the last two micro steps of one multiply operation is done in parallel with the execution of the first two micro-steps of the next multiply operation. As a result, 8×8-bit multiply operations can start and produce results every

50ps at the rate of 20 GHz.

The five least significant bits of the product are calculated during the PP reduction by the [4:2] compressors. The partial products in the three most significant bit columns are reduced to carry-sum pairs and then go through the final summation done by a wave-pipelined ripple-carry adder

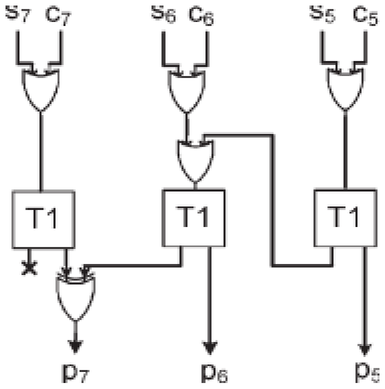

Fig. 7. Ripple-carry adder for final summation. Clock signal distribution lines for T1 and XOR gates are not shown.

IV. RESULTS AND VERI FICATION

The designed multiplier was simulated on the Xilinx ISE tool and implemented in FPGA. Results are obtained for the various N x M. The obtained results are compared with the other multipliers and this proposed multiplier has the high performance than the others.

The RTL view shows the proposed multiplier has been shown in the fig 8. The area has also been reduced by using the proposed multiplier. Thus the proposed multiplier has the greater advantage in terms of area and delay . The test vectors are used to verify the correct operation of the circuit.

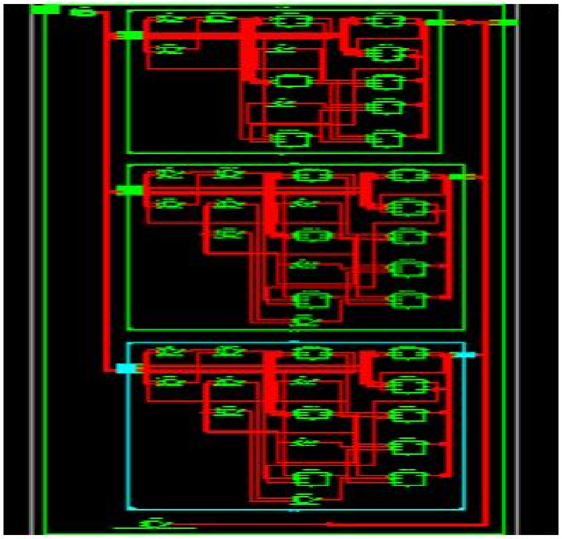

FIG 8. RTL View of 4 X 4 Multiplier

V. CONCLUSION

In this paper, the Multiplications are carried on the Xilinx ISE tool. Multipliers and their Implementations are carried on FPGA devices. Previous techniques and designs are analyzed to carry out performance comparison in terms of area and delay . The new approach for the Multiplier is to improve the efficiency and to reduce the area. The proposed circuit shows a high performance design of multiplier. The proposal shows encouraging results: its figures are comparable and outperformed than other multipliers

REFERENCES

l. K. Likharev, "The New Superconducting Electronics", H. Weinstock and R. Ralston (eds.), Kluwer Academic Publishers, Dordrecht, 1993, p.423.

2. O. A. Mukhanov, Extended Abstracts of 4th Int. Superconductive Electronics Conf., Boulder, 1993, p.l9.

3. O. A. Mukhanov, S. V. Rylov, V. K. Semenov, and SV. Vyshenskii, IEEE Trans. Magn., 25, 857 (1989).
4. S. B. Kaplan and O. A. Mukhanov, IEEE Trans. Magn., 5,2853 (1995).

5. A. F. Kirichenko and O. A. Mukhanov, IEEE Trans. Magn., 5, 30 I 0 (1995).

6. S. V. Polonsky, J. C. Lin, and A. V. Ry\yakov, IEEE Trans. Magn., 5, 2823 (1995).

7. S. S. Martinet, D. K. Brock, M. J. Feldman, and M. F. Bocko, IEEE Trans. Magn., 5, 3006 (1995).

8. O. A. Mukhanov and A. F. Kirichenko, IEEE Trans. Magn., 5, 2461 (1995).

9. K. Gaj, E. Friedman, M. J. Feldman, and A. Krasniewski, IEEE Trans. Magn., 5, 3320 (1995).

10. S. V. Rylov and R. P. Robertazzi, IEEE Trans. Magn., 5, 2260 (1995).

11. HYPRES Design Rules available from Hypres, Inc., 175 Clearbrook Rd., Elmsford, NY 10523, Attn. John Coughlin.

12. S. V. Polonsky, V. K. Semenov, and P. N. Shevchenko, Supercond. Sci. Technol., 4, 667 (1991).