

An FPGA Implementation of Modulo Multiplier Using Radix-8 Booth Algorithm

Y.Jeevan

ECE Department, Guru Nanak Institute of Technology, Hyderabad

CH.Charan

charankanna9849@gmail.com ECE Department, Guru Nanak Institute of Technology, Hyderabad

B.Pallavi

pallavibadhavath@gmail.com

ECE Department, Guru Nanak Institute of Technology, Hyderabad

A.Sai Sree

saisree2535@gmail.com ECE Department, Guru Nanak Institute of Technology, Hyderabad

ABSTRACT

Due to its high modularity and carry-free addition, a redundant binary (RB) representation can be used when designing high performance multipliers. The conventional RB multiplier requires an additional RB partial product (RBPP) row, because an error-correcting word (ECW) is generated by both the radix-8 Modified Booth encoding (MBE) and the RB encoding. This incurs in an additional RBPP accumulation stage for the MBE multiplier. In this paper, a new RB modified partial product generator (RBMPPG) is proposed; it removes the extra ECW and hence, it saves one RBPP accumulation stage. Therefore, the proposed RBMPPG generates fewer partial product rows than a conventional RB MBE multiplier. Simulation results show that the proposed RBMPPG based designs significantly improve the area and power consumption when the word length of each operand in the multiplier is at least 32 bits.

INTRODUCTION

Digital multipliers are widely used in arithmetic units of microprocessors, multimedia and digital signal processors. Many algorithms and architectures have been proposed to design high-speed and low power multipliers [1-13]. A normal binary (NB) multiplication by digital circuits includes three steps. In the first step, partial products are generated; in the second step, all partial products are added by a partial product reduction tree until two partial product rows remain. In the third step, the two partial product rows are added by a fast carry propagation adder. Two methods have been used to perform the second step for the partial product reduction. A first method uses 4-2 compressors, while a second method uses redundant binary (RB) numbers [5-6]. Both methods allow the partial product reduction tree to be reduced at a rate of 2:1. The redundant binary number representation has been introduced by Avizienis [1] to perform signed-digit arithmetic; the RB number has the capability to be represented in different ways. Fast multipliers can be designed using redundant binary addition trees [2-3]. The redundant binary representation has also been applied to a floating-point processor and implemented in VLSI [4]. High performance RB multipliers have become popular due to the advantageous features, such as high modularity and carry-free addition [5-9]. A RB multiplier consists of a RB partial product (RBPP) generator, a RBPP reduction tree and a RB-NB converter. A Radix-4 Booth encoding or a modified Booth encoding (MBE) is usually used in the partial product generator of parallel multipliers to reduce the number of partial product rows by half [5-6] [10-13].

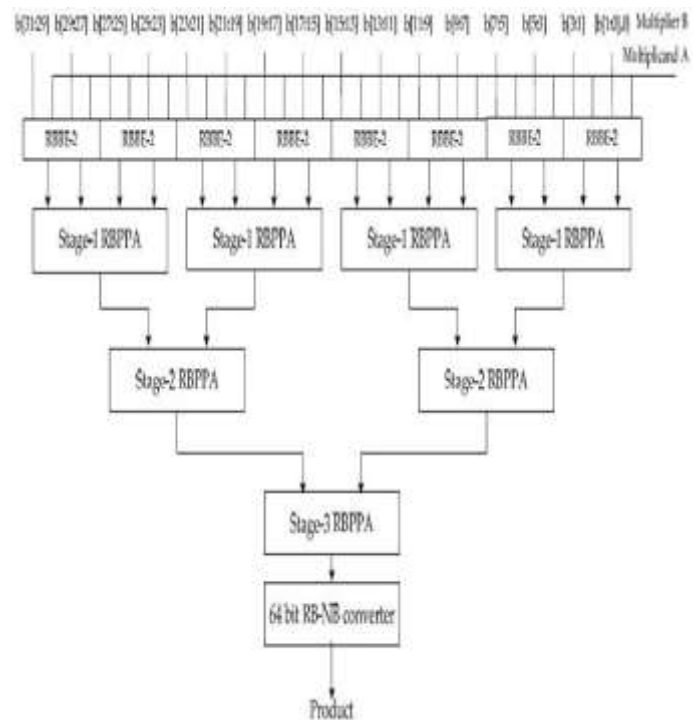
I. EXISTING SYSTEM

Booth encoding has been proposed to facilitate the multiplication of two's complement binary numbers. It was revised as modified Booth encoding (MBE) or radix-4 Booth encoding [18]. The MBE scheme is summarized for the multiplicand, and for the multiplier. The multiplier bits are grouped in sets of three adjacent bits. The two side bits are overlapped with neighbouring groups except the first multiplier. Methods have been proposed to solve the problem of correction bits for NB radix-4 Booth encoding (NBBE-2) multipliers. However, this problem has not been solved

The proposed RBMPPG-2 can be applied to any 2_n -bit RB multipliers with a reduction of a RBPP accumulation stage compared with conventional designs. Although the delay of RMPPG-2 increases by 1-stage of TG delay, the delay of one RBPP accumulation stage is significantly larger than a 1-stage TG delay. Therefore, the delay of the entire multiplier is reduced. The improved complexity, delay and power consumption are very attractive for the proposed design. A 32-bit RB MBE multiplier using the proposed RBPP generator. The multiplier consists of 4 stages in a conventional 32-bit RB MBE multiplier architecture; however, by using the proposed RBMPPG-2, the number of RBPP accumulation stages is reduced from 4 to 3. These are significant savings in delay, area as well as power consumption. The improvements in delay, area and power consumption are further demonstrated in the next section by simulation. Table V compares the number of RBPP accumulation stages in different 2_n -bit RB multipliers, i.e., 8×8 -bit, 16×16 -bit, 32×32 -bit, 64×64 -bit multipliers. For a 64-bit multiplier, the proposed design has 4 RBPP accumulation stages; it reduces the partial product accumulation delay time by 20% compared with CRBBE-2 multipliers. Although both the proposed design and RBBE-4 have the same number of RBPP accumulation stages, RBBE-4 is more complex, because it uses radix-16 Booth encoding.

for RB MBE multipliers..

PROPOSED SYSTEM



2. Hardware Security Modules

Because modular multiplication is a core operation in many encryption schemes, this architecture can be used to build secure hardware blocks for financial transactions and authentication devices.

3. Reconfigurable Communication Hardware

Modern communication equipment often requires fast arithmetic for coding, decoding and symbol-level operations. This multiplier can be embedded in FPGAs used in software-defined radios and baseband processors.

4. Energy-Efficient Arithmetic Cores

The reduced number of partial products in Radix-8 Booth multiplication helps design low-power computation blocks for battery-powered embedded devices.

5. High-Performance Control Systems

Industrial controllers and robotic platforms that rely on rapid feedback loops can incorporate this multiplier to speed up mathematical computations inside control

algorithms. Industrial Automation.

6. Modular Processing in RNS (Residue Number System) When implementing RNS-based computing, the proposed FPGA design can serve as a dedicated modulo multiplication unit, improving overall computational speed and area efficiency.

HARDWARE DETAILS

An integrated circuit or monolithic integrated circuit (also referred to as IC, chip, or microchip) is an electronic circuit manufactured by the patterned diffusion of trace elements into the surface of a thin substrate of semiconductor material. Additional materials are deposited and patterned to form interconnections between semiconductor devices. Integrated circuits are used in virtually all electronic equipment today and have revolutionized the world of electronics. Computers, mobile phones, and other digital appliances are now inextricable parts of the structure of modern societies, made possible by the low cost of production of integrated circuits. ICs were made possible by experimental discoveries showing that semiconductor devices the functions of vacuum tubes could perform and by mid-20th-century technology advancements in semiconductor device fabrication .

The integration of large numbers of tiny transistors into a small chip was an enormous improvement over the manual assembly of circuits using discrete electronic components . The integrated circuits mass production block approach to circuit design capability, reliability, and building ensured the rapid adoption of standardized ICs in place of designs using discrete transistors. There are two main advantages of ICs over discrete circuits : cost and performance. Cost is low because the chips, with all their components, are printed as a unit by photolithography rather than being constructed one transistor at a time. Furthermore, much less material is used to construct a packaged IC die than to construct a discrete circuit. Performance is high because the components switch quickly and consume little power (compared to their discrete counterparts) as a result of the small size and close proximity of the components. As of 2006, typical chip areas range from a few square millimeters to around 350 mm², with up to 1 million.

SOFTWARE DETAILS

→ Verification Tool

- Modelsim 6.4c

→ Synthesis Tool

- Xilinx ISE 13.2

MODELSIM

ModelSim is a useful tool that allows you to stimulate the inputs of your modules and view both outputs and internal signals. It allows you to do both behavioural and timing simulation; however, this document will focus on behavioural simulation. Keep in mind that these simulations are based on models and thus the results are only as accurate as the constituent models. ModelSim

/VHDL, ModelSim /VLOG, ModelSim /LNL, and ModelSim /PLUS are produced by Model Technology™ Incorporated. Unauthorized copying, duplication, or other reproduction is prohibited without the written consent of Model Technology. The information in this manual is subject to change without notice and does not represent a commitment on the part of Model Technology. The program described in this manual is furnished under a license agreement and may not be used or copied except in accordance with the terms of the agreement. The online documentation provided with this product may be printed by the end-user. The number of copies that may be printed is limited to the number of licenses purchased. ModelSim is a registered trademark of Model Technology Incorporated. Model Technology is a trademark of Mentor Graphics Corporation.

VERILOG

Verilog is one of the two major Hardware Description Languages (HDL) used by hardware designers in industry and academia Verilog is very C-like and liked by electrical and computer engineers as most learn the C language in college. Verilog was introduced in 1985 by Gateway Design System Corporation, now a part of Cadence Design Systems, Inc.'s Systems Division. Until May, 1990, with the formation of Open Verilog International (OVI), Verilog HDL was a proprietary language of Cadence. Cadence was motivated to open the language to the Public Domain with the expectation that the market for Verilog HDL-related software products would grow more rapidly with broader acceptance of the language. Cadence realized that Verilog HDL users wanted other software and service companies to embrace the language and develop Verilog-supported design tools.

RTL SCHEMATIC

The RTL schematic is abbreviated as the register transfer level it denotes the blue print of the architecture and is used to verify the designed architecture to the ideal architecture that we are in need of development .The hdl language is used to convert the description or summery of the architecture to the working summery by use of the coding language i.e verilog ,vhdl. The RTL schematic even specifies the internal connection blocks for better analyzing .The figure represented below shows the RTL schematic diagram of the designed architecture.

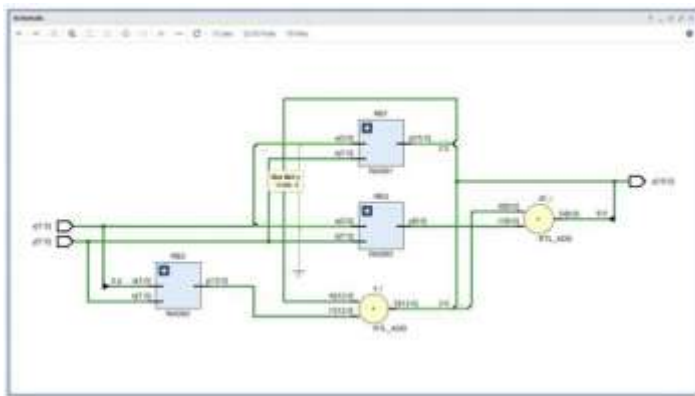


Fig 2: RTL Schematic

SIMULATION

Simulation in verilog is the process of verifying the functionality of a digital circuit described using the verilog hardware description language it allows designers to test how their design behaves under various input conditions before implementing it in hardware.

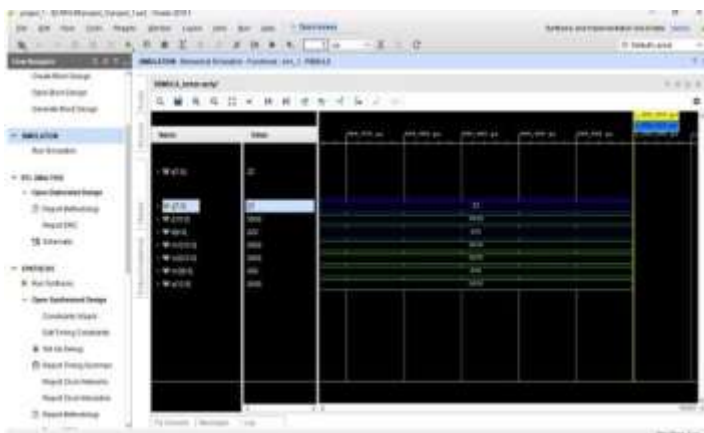


Fig 3: Stimulation

CONCLUSION

A new modified RBPP generator has been proposed in this paper; this design eliminates the additional ECW that is introduced by previous designs. Therefore, a RBPP accumulation stage is saved due to the elimination of ECW. The new RB partial product generation technique can be applied to any 2_-bit RB multipliers to reduce the number of RBPP rows from $/4 + 1_to_/_4_$. Simulation results have shown that the performance of RB MBE multipliers using the proposed RBMPPG-2 is improved significantly in terms of delay and area. The proposed designs achieve significant reductions in area and power consumption when the word length is at least 32 bits. The PDP can be reduced by up to 59% using the proposed RB multipliers when compared with existing RB multipliers. Hence, the proposed RBPP generation method is a very useful technique when designing area and PDP efficient power-of-two RBMBEmultipliers.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to our internal guide, **Mr.Y.Jeevan**, Assistant Professor, Electronics and Communication Engineering, for his Valuable guidance , encouragement, and continuous Support throughout the duration of this project.

I am also thankful to **Dr.S.P.Yadav**, HOD and Dean Academics, Electronics and Communication Engineering, For his expert supervision and helpful suggestions, Which contributed significantly to the successful completion of this project.

I would like to express my profound sense of gratitude to **Dr. K. Venkata Rao**, Principal, for his constant and valuable guidance.

I would like to thank **Dr. Sanjeev Shrivastava**, Director, for his valuable support

I would like to express my deep sense of gratitude to **Dr. H. S. Saini**, Managing Director, Guru Nanak Group Of Institutions for his tremendous support, encouragement, and inspiration.

I would also like to thank the faculty members of the **Electronics and Communication Engineering** and the Lab Technicians for their assistance and cooperation during the practical work of my project.

I am grateful to my friends and well-wishers for their Encouragement ,collaboration ,and feedback throughout the project journey. Lastly,sincerely thank my parents for their constant support patience , and motivation, which helped me complete this project successfully.

REFERENCES:

- [1] A. Avizienis, "Signed-digit number representations for fast parallel arithmetic," IRE Trans. Electron. Computers, vol. EC-10, pp. 389–400, 1961.
- [2] N. Takagi, H. Yasuura, and S. Yajima, "High-speed VLSI multiplication algorithm with a redundant binary addition tree," IEEE Trans. Computers, vol. C-34, pp. 789- 796, 1985.
- [3] Y. Harata, Y. Nakamura, H. Nagase, M. Takigawa, and N. Takagi, "A high speed multiplier using a redundant binary adder tree," IEEE J. Solid-State Circuits, vol. SC- 22, pp. 28-34, 1987.
- [4] H. Edamatsu, T. Taniguchi, T. Nishiyama, and S. Kuninobu, "A 33 MFLOPS floating point processor using redundant binary representation," in Proc. IEEE Int. Solid- State Circuits Conf. (ISSCC), pp. 152–153, 1988.
- [5] H. Makino, Y. Nakase, and H. Shinohara, "A 8.8-ns 54x54-bit multiplier using new redundant binary architecture," in Proc. Int. Conf. Comput. Design (ICCD), pp. 202-205, 1993.
- [6] H. Makino, Y. Nakase, H. Suzuki, H. Morinaka, H. Shinohara, and K. Makino, "An 8.8-ns 54×54-bit multiplier with high speed redundant binary architecture," IEEE J. Solid- State Circuits, vol. 31, pp. 773-783, 1996.
- [7] Y. Kim, B. Song, J. Grosspietsch, and S. Gillig, "A carry- free 54b×54b multiplier using equivalent bit conversion algorithm," IEEE J. Solid-State Circuits, vol. 36, pp. 1538– 1545, 2001.
- [8] Y. He and C. Chang, "A power-delay efficient hybrid carrylookahead carry-select based redundant binary to two's complement converter," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, pp. 336–346, 2008.
- [9] G. Wang and M. Tull, "A new redundant binary number to 2'scomplement number converter," in Proc. Region 5 Conference: Annual Technical and Leadership Workshop, pp. 141-143, 2004.
- [10] W. Yeh and C. Jen, "High-speed Booth encoded parallel multiplier design," IEEE Trans. Computers, vol. 49, pp. 692-701, 2000.
- [11] S. Kuang, J. Wang, and C. Guo, "Modified Booth multiplier with a regular partial product array," IEEE Trans. Circuits Syst. II, vol. 56, pp. 404 408, 2009.
- [12] J. Kang and J. Gaudiot, "A simple high-speed multiplier design," IEEE Trans. Computers, vol. 55, pp.1253-1258, 2006.