

# FPGA Implementation of a Resource Efficient Vedic Multiplier using SPST Adders

**K. V. Gowreesrinivas**

Department of ECE, Anil Neerukonda Institute of Technology & Sciences (ANITS), India  
srinivas.ece@anits.edu.in (corresponding author)

**Srinivas Sabbavarapu**

Department of ECE, Anil Neerukonda Institute of Technology & Sciences (ANITS), India  
ssrinivas.ece@anits.edu.in

**Punniakodi Samundiswary**

Department of Electronics Engineering, Pondicherry University, India  
samundiswary\_pdy@yahoo.com

Received: 19 February 2023 | Revised: 27 March 2023 | Accepted: 29 March 2023

Licensed under a CC-BY 4.0 license | Copyright (c) by the authors | DOI: <https://doi.org/10.48084/etasr.5797>

## ABSTRACT

Nowadays, the requirement for very high-speed operations in processors constantly increases. Multiplication is a crucial operation in high power-consuming processes such as image and signal processing. The main characteristics of a multiplier are good accuracy, speed, reduction in area, and little power consumption. Speed plays a major role in multiplication operations, and an increase in speed can be obtained by reducing the number of steps involved in the computation process. Since a multiplier has the largest delay among the basic blocks in a digital system, the critical path is determined by it. Furthermore, the multiplier consumes more area and dissipates more power. Hence, designing multipliers that offer high speed, lower power consumption, less area, or a combination of them is of prime concern. Hence, an attempt is made in this paper to achieve the above design metrics using a Spurious Power Suppression Technique (SPST) adder. A resource-efficient SPST-based Vedic multiplier is developed and implemented using Artix 7 FPGA and is finally compared with the ripple carry adder-based Vedic multiplier.

**Keywords-**vedic multiplier; Urdhva Triyakbhyam; Spurious Power Suppression Technique (SPST); Verilog; Artix7

## I. INTRODUCTION

The operation of multiplication plays a pivotal role in many operations and it is important to be computed fast with great efficiency in Digital Signal Processing (DSP) and Image Processing, where functions like Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT) are used. The research to obtain improved multiplier versions and establish new method types for efficacious multiplier architecture is still ongoing. Each multiplier is different in its own way. High performance with a less number of partial products is preferred in a multiplier. Much research has been conducted regarding the ancient Vedic mathematics, which brought many efficacious multipliers, dividers, squaring, and cube architectures. During the last few years, various decimal algorithms have been extended to the binary number system based on the Vedic mathematics [1-4]. Reducing the number of stages in a Vedic multiplication process leads to a great reduction in the propagation delay. The multiplication process is based on Urdhva Tiryagbhyam algorithm/sutra, which belongs to the 16 sutras of Vedic mathematics [5-7]. Generally, most multipliers

consume area, power, and work with little accuracy. Therefore, the design of multipliers which offer high speed, low power consumption, less area, or combinations of all these, is of immense research interest [8-15].

## II. VEDIC MULTIPLIER

Indian mathematics has a unique technique of arithmetic computation based on 16 sutras or formulas, which is applicable in many fields. One of the fastest developing fields is DSP. So, there is a need to implement as fast as possible multiplication operation for the digital processing of signals. The Vedic multiplier is mainly based on the vertical and crosswise algorithm of Urdhva Triyakbhyam generating all partial products and sums in a single step. The UT Sutra has been conventionally utilized for the multiplication of decimal numbers in comparatively less time. We can use this to multiply two numbers. The bits on the two extremes of the number are multiplied and the result is added to the previous carry. When there are more digits in each number, all the results are added to the previous carry. The process of this scripture is discussed below with the help of an example.

Let us consider two decimal numbers: 234 and 159.

- Initially, the least significant digits of both numbers are multiplied and the result is 36.
- Next, we have cross-wise multiplication of the first digit of the first number with the second digit of the second number. Another cross-wise multiplication of the first number's second digit with the second number's first digit. Finally, we add those two products.
- Cross-wise multiplication of the first number's least significant digit with the second number's most significant digit and vice-versa. Multiplication of the middle digits gives a total of three products which are finally added.
- Without considering the least significant digits of the two numbers, we perform cross-wise multiplication and addition.
- Vertical multiplication of most significant digits of the two numbers.

For all the steps, with the exception of the first step, each compartment needs to have only one digit. If not, then the carry is forwarded to the initial digits in the previous compartment.

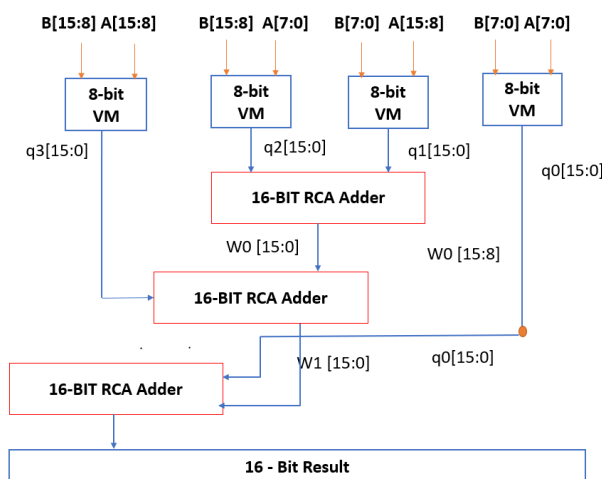


Fig. 1. The 16-bit Vedic multiplier.

The delay of the multiplication is reduced using UT sutra and initially this was used only for decimal number multiplication, although, it can be used for binary numbers. Binary multiplication can be done using this method for 2-bit, 3-bit, and 4-bit numbers, since the PPs and their arithmetic sum are calculated autonomously. Delay is produced due to the existence of the carry from the PPs which is to be added in each next step's partial product. Figure 1 shows the process of 16-bit Vedic multiplication. In the first stage, 8x8 Vedic multipliers are used. The obtained products are added according to the Vedic multiplication discussed above. Finally, the 32-bit result is obtained.

### III. SPST ADDER

An adder, either a Full Adder (FA) or a Half Adder (HA), is a combinational circuit used to calculate the sum of three or two inputs. In a parallel adder, FAs are connected in series to produce an n-bit adder. In each section, the sum and carry are not computer until the input carry arises, slowing down the summation process. To avoid delay issues, the Carry Look Ahead (CLA) adder has been suggested. To improve the speed of CLA, Spurious Power Suppression Technique (SPST) is applied. This can reduce the dissipated power of combinational VLSI for DSP purposes. The SPST consists of two parts, the most significant part and the least significant part. It disables the MSP when it doesn't modify the results saving power [16-20]. The SPST adder/subtractor is classified into two parts, MSP and LSP. The MSP of the first adder/ subtractor is modified to include:

- detection unit,
- data controlling unit,
- sign extension unit, and
- circuits for calculating Cin and Cout signals.

In SPST, the detection unit is used to detect if a transition in data bits of the result will occur in the circuits (adders or multipliers) or not. The detection logic unit works as follows:

- When the detection unit disables the MSP, then the output of the MSP are re-compensated so that the time saved for skipping computations will cancel out the delay of the detection logic circuit.
- When it enables the MSP, then the MSP will wait for acknowledgement of the detection unit to enable the data latches. So, the delay of the detection unit will contribute to the entire circuit's delay.
- When the detection unit stays in its decision irrespective enabling or disabling of the MSP, then the delay of detection unit is insignificant.

There is a data asserting control unit realized by using the registers to further remove any unwanted spurious signals of the arithmetic unit whenever the latched portion has been turned on. This asserting control obviously brings power reduction [16-20].

The 16-bit addition using the SPST process is interpreted in 5 cases. The first case is the transient state, in which spurious transitions of carry signals occur in MSP although the result of the MSP is not altered. The second and third cases illustrate the situation of a negative operand adding to a positive operand without and with carry from LSP, respectively. The fourth and fifth cases demonstrate the condition of two negative operand additions without and with carry-in from LSP, respectively. In all the above cases, the results of the MSP are predictable, so the computations in the MSP aren't required and can be neglected. Removing those spurious computations will not only save the power consumed by the SPST adder or subtractor but also reduce the noises which are going to affect the subsequent arithmetic circuits.

IV. THE PROPOSED MULTIPLIER

Our main goal is to differentiate the performances of the Vedic multiplier with and without using the SPST adder by comparing both models. To improve the Vedic multiplier's efficiency, the SPST adder is introduced at intermediate stages to add the sub module outputs. Finally, performance analysis is conducted with respect to area and power.

To develop 16-bit Vedic multiplication, four 8x8 multipliers are required. In the second stage, 16-bit SPST adders are used to add intermediate partial products.

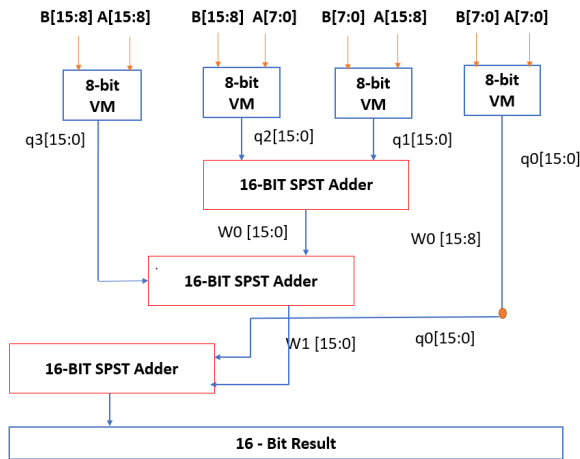


Fig. 2. 16-bit Vedic multiplier using SPST adders.

Both multipliers models were simulated and synthesized using Xilinx Vivado tool. Two 16-bit Vedic multipliers, one with RCA and one with SPST, were programmed using HDL and their performance was compared with respect to area and power.

V. SIMULATION RESULTS

All the blocks were programmed and simulated using Xilinx Vivado tool. After verification, the proposed multiplier was implemented on Nexsys 4 DDR Artix 7 FPGA. The synthesis and simulation results of VM using SPST with respect to main modules and sub modules are shown in Figures 3 to 10.

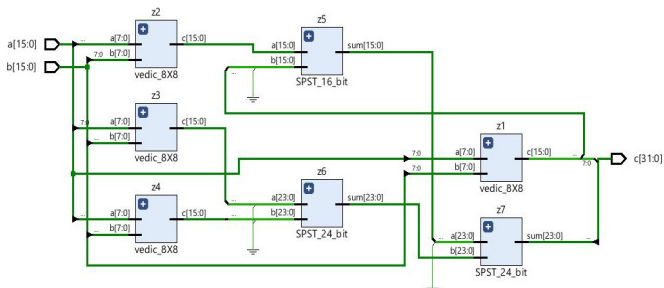


Fig. 3. RTL schematic of a Vedic multiplier using SPST blocks.

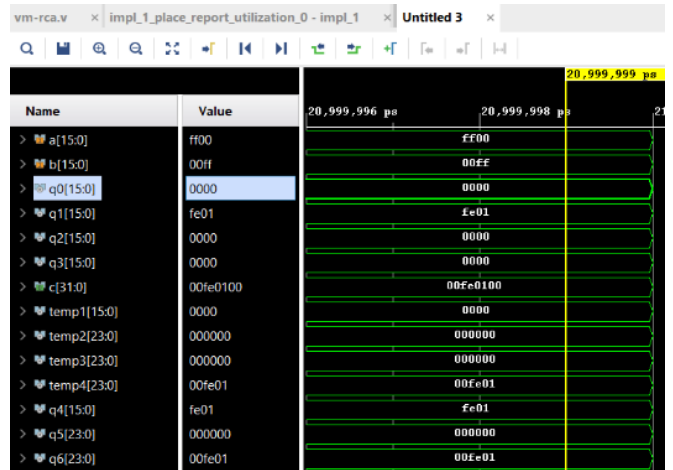


Fig. 4. Vedic multiplier using RCA simulation report.

Figure 2 represents the RTL diagram of the Vedic multiplier which utilizes four 8-bit Vedic multipliers and three SPST adders to obtain the results. Figure 3 gives the simulation output for two 16-bit numbers. From the synthesized results it is concluded that the Vedic multiplier using RCA logic requires 402 slice LUTs and 120 slice registers as shown in Figure 5. Further, it is noticed that it requires 33.77W on-chip power which includes 32.586W dynamic power as shown in Figure 6.

. Slice Logic

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	402	0	63400	0.63
LUT as Logic	402	0	63400	0.63
LUT as Memory	0	0	19000	0.00
Slice Registers	120	0	126800	0.09
Register as Flip Flop	120	0	126800	0.09
Register as Latch	0	0	126800	0.00
F7 Muxes	0	0	31700	0.00
F8 Muxes	0	0	15850	0.00

Fig. 5. Vedic multiplier using RCA implementation report.

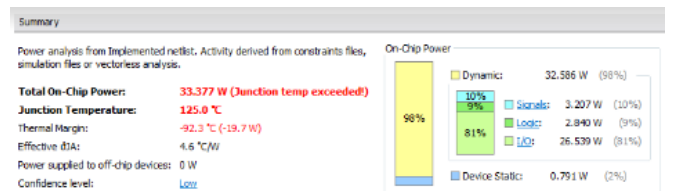


Fig. 6. Power report of the 16-bit Vedic multiplier using RCA.

TABLE I. PERFORMANCE COMPARISON

Vedic multiplier	Appearance		
	Slice LUTs	Slice registers	Power(W)
Using RCA	402	120	33.54
Using SPST	361	0	25.49

From the synthesized results it is concluded that the Vedic multiplier using SPST logic (Figure 7) requires 361 slice LUTs

(Figure 8) and 25.49W on-chip power which includes 24.708W dynamic power as shown in Figure 9. The result comparison can be seen in Table I. It can be seen that the SPST-based Vedic multiplier provides better performance in terms of LUTs and delay.

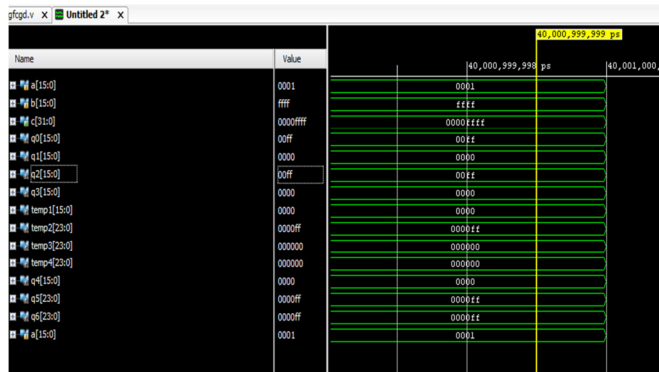


Fig. 7. Vedic multiplier using SPST simulation report.

. Slice Logic  
-----

Site Type	Used	Fixed	Available	Util1%
Slice LUTs*	361	0	63400	0.57
LUT as Logic	361	0	63400	0.57
LUT as Memory	0	0	19000	0.00
Slice Registers	0	0	126800	0.00
Register as Flip Flop	0	0	126800	0.00
Register as Latch	0	0	126800	0.00
F7 Muxes	0	0	31700	0.00
F8 Muxes	0	0	15850	0.00

Fig. 8. Vedic multiplier using SPST implementation report.

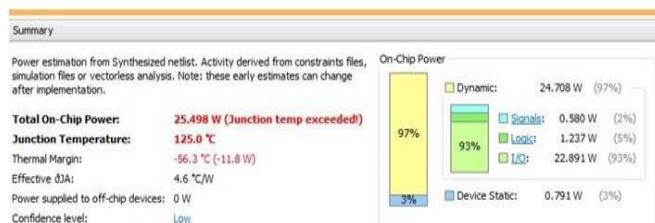


Fig. 9. Vedic multiplier using SPST power report.

### VI. HARDWARE IMPLEMENTATION

The external hardware used for implementation consists of:

- ADC kit
- Slide switches and LEDs
- Jumper wires, 3.3V fixed voltage source (from RPS)
- Nexys 4 DDR

After simulation, the proposed Vedic multiplier using the SPST model was dumped into the FPGA to verify the response. In this work, the proposed model was implemented on Nexys 4 DDR Artix7 FPGA with the help of Xilinx Vivado. For the implementation, the first task is to map the original input and

output ports with board I/Os by generating a constraint file. Next, the program can be dumped to the FPGA kit to verify the simulation results practically.

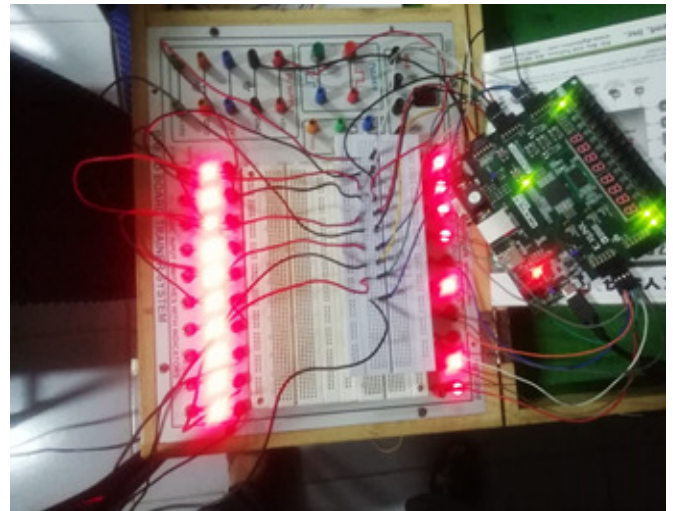


Fig. 10. Implementation of the Vedic multiplier with SPST in Artix7.

### VII. CONCLUSION

In this paper, a 16-bit Vedic multiplier was designed using SPST adders and was programmed in Verilog. The proposed Vedic multiplier was synthesized in the Artix7 FPGA family using the Xilinx tool and its response in terms of area and delay was observed. This work is mainly focused on developing efficient Vedic multipliers using SPST adders and its performance was compared with the one of the ripple carry adder-based Vedic multiplier. From the implementation results, it is observed that the performance of the Vedic multiplier using SPST is better by 10% in respect of area, while power dissipation is improved by 24% compared to that of Vedic using the RCA adder. After the validation, the SPST-based Vedic multiplier was successfully implemented on Nexys 4 DDR Artix FPGA.

### REFERENCES

- [1] M. Ramalatha, K. D. Dayalan, P. Dharani, and S. D. Priya, "High speed energy efficient ALU design using Vedic multiplication techniques," in *2009 International Conference on Advances in Computational Tools for Engineering Applications*, Beirut, Lebanon, Jul. 2009, pp. 600–603, <https://doi.org/10.1109/ACTEA.2009.5227842>.
- [2] Y. Bansal and C. Madhu, "A novel high-speed approach for 16 × 16 Vedic multiplication with compressor adders," *Computers & Electrical Engineering*, vol. 49, pp. 39–49, Jan. 2016, <https://doi.org/10.1016/j.compeleceng.2015.11.006>.
- [3] R. K. Barik, M. Pradhan, and R. Panda, "Time efficient signed Vedic multiplier using redundant binary representation," *The Journal of Engineering*, vol. 2017, no. 3, pp. 60–68, Mar. 2017, <https://doi.org/10.1049/joe.2016.0376>.
- [4] Y. Bansal, C. Madhu, and P. Kaur, "High speed vedic multiplier designs-A review," in *2014 Recent Advances in Engineering and Computational Sciences (RAECS)*, Chandigarh, India, Mar. 2014, <https://doi.org/10.1109/RAECS.2014.6799502>.
- [5] A. Kumar and A. Raman, "Low power ALU design by ancient mathematics," in *2010 The 2nd International Conference on Computer*

- and Automation Engineering (ICCAE), Singapore, Oct. 2010, vol. 5, pp. 862–865, <https://doi.org/10.1109/ICCAE.2010.5451892>.
- [6] D. Jaina, K. Sethi, and R. Panda, "Vedic Mathematics Based Multiply Accumulate Unit," in *2011 International Conference on Computational Intelligence and Communication Networks*, Gwalior, India, Jul. 2011, pp. 754–757, <https://doi.org/10.1109/CICN.2011.167>.
- [7] R. K. Bathija, R. S. Meena, S. Sarkar, and R. Sahu, "Low Power High Speed 16x16 bit Multiplier using Vedic Mathematics," *International Journal of Computer Applications*, vol. 59, no. 6, pp. 41–44, Dec. 2012.
- [8] S. Anjana, C. Pradeep, and P. Samuel, "Synthesize of High Speed Floating-point Multipliers Based on Vedic Mathematics," *Procedia Computer Science*, vol. 46, pp. 1294–1302, Jan. 2015, <https://doi.org/10.1016/j.procs.2015.01.054>.
- [9] N. H. Nguyen, S. A. Khan, C.-H. Kim, and J.-M. Kim, "A high-performance, resource-efficient, reconfigurable parallel-pipelined FFT processor for FPGA platforms," *Microprocessors and Microsystems*, vol. 60, pp. 96–106, Jul. 2018, <https://doi.org/10.1016/j.micpro.2018.04.003>.
- [10] Y.-H. Seo and D.-W. Kim, "A New VLSI Architecture of Parallel Multiplier–Accumulator Based on Radix-2 Modified Booth Algorithm," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 2, pp. 201–208, Oct. 2010, <https://doi.org/10.1109/TVLSI.2008.2009113>.
- [11] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi, "Design of Approximate Radix-4 Booth Multipliers for Error-Tolerant Computing," *IEEE Transactions on Computers*, vol. 66, no. 8, pp. 1435–1441, Dec. 2017, <https://doi.org/10.1109/TC.2017.2672976>.
- [12] A. Mittal, A. Nandi, and D. Yadav, "Comparative study of 16-order FIR filter design using different multiplication techniques," *IET Circuits, Devices & Systems*, vol. 11, no. 3, pp. 196–200, 2017, <https://doi.org/10.1049/iet-cds.2016.0146>.
- [13] D. M. Perisic and M. Bojovic, "Application of Time Recursive Processing for the Development of a Time/Phase Shifter," *Engineering, Technology & Applied Science Research*, vol. 7, no. 3, pp. 1582–1587, Jun. 2017, <https://doi.org/10.48084/etasr.1179>.
- [14] D. M. Perisic, A. C. Zoric, and Z. Gavric, "A Frequency Multiplier Based on Time Recursive Processing," *Engineering, Technology & Applied Science Research*, vol. 7, no. 6, pp. 2104–2108, Dec. 2017, <https://doi.org/10.48084/etasr.1499>.
- [15] D. M. Perisic, V. Petrovic, and B. Kovacevic, "Frequency Locked Loop Based on the Time Nonrecursive Processing," *Engineering, Technology & Applied Science Research*, vol. 8, no. 5, pp. 3450–3455, Oct. 2018, <https://doi.org/10.48084/etasr.2256>.
- [16] K.-H. Chen and Y.-S. Chu, "A Low-Power Multiplier With the Spurious Power Suppression Technique," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 7, pp. 846–850, Jul. 2007, <https://doi.org/10.1109/TVLSI.2007.899242>.
- [17] H. S. K. Puttam, P. S. Rao, and N. V. G. Prasad, "Implementation of Low Power and High Speed Multiplier- Accumulator Using SPST Adder and Verilog," *International Journal of Modern Engineering Research*, vol. 2, no. 5, pp. 3390–3397, 2012.
- [18] A. Prashanth, R. P. Waran, and S. K. and S. Pawar, "Low Power High Speed based Various Adder Architectures using SPST," *Indian Journal of Science and Technology*, vol. 9, no. 29, pp. 1–3, May 2016, <https://doi.org/10.17485/ijst/2016/v9i29/93197>.
- [19] V. S. Narayan, S. M. Pratima, V. S. Saroja, and R. M. Banakar, "High Speed Low Power VLSI Architecture for SPST Adder Using Modified Carry Look Ahead Adder," in *Proceedings of International Conference on Advances in Computing*, New Delhi, India, 2012, pp. 461–466, [https://doi.org/10.1007/978-81-322-0740-5\\_56](https://doi.org/10.1007/978-81-322-0740-5_56).
- [20] A. Purna Ramesh, A. V. N. Tilak, and A. M. Prasad, "Efficient Implementation of 16-Bit Multiplier-Accumulator Using Radix-2 Modified Booth Algorithm and SPST Adder Using Verilog," *International Journal of VLSI Design & Communication Systems*, vol. 3, no. 3, pp. 107–118, Jul. 2012.