Design of a 16-bit unsigned Multiplier

***Abstract*—This paper presents the design and verification of a 16-bit unsigned multiplier topology. The designed multiplier topology presented in this work comprises an array of carry-save adder (CSA), and carry-propagate adder (CPA) topologies, that enable the multiplication of two 8-bit operands. The schematic and layout have been designed using TSMC 65nm technology node. A testbench schematic has been designed using both the symbolic view created from the design schematic and the symbolic view generated from the Verilog code and implemented along with a Random Number Generator (RNG) to ensure a thorough testing of the multiplier topology for different inputs. The work presented combines the design and testing of the functionality of the multiplier comprehensively.**

Keywords— Carry-propagate adder, carry-save adder, multiplier, random number generator, Verilog code.

I. Introduction

The modern era of high-speed digital signal operation requires the use of high-speed design methodologies that help decrease the computation time. Multiplication is one of the fundamental arithmetic operations, that not only is a core step involved in signal processing tasks but is also a crucial step in various complex scientific calculations. As technology continues to scale the design of a suitable multiplier technology becomes a significant step in contributing towards the high-speed digital signal operations.

This paper highlights the design and development of a 16-bit unsigned multiplier topology which acts as one of the significant building blocks in digital signal processing. The designed multiplier can be used in various signal-processing applications as well as for commercial applications like mobiles, computers, high-speed calculators, and some general-purpose processors. The multiplier designed in this work is based on a combination of two adder topologies, explicitly carry select adder (CSA) and carry propagate adder (CPA). The multiplier is designed to be appropriately used in high-speed applications. The paper also extensively discusses the multiplier design validation techniques by testing the design through testbench schematic simulations and by implementing Verilog code for random number generation enabling the multiplier to be tested for random values, thus ensuring proper operation.

The paper is structured in the following manner: section II provides an in-depth analysis of the design methodologies for the multiplier, including the design of adders to be used in the multiplier, followed by the layout design of the multiplier. Section III shows the simulation results and description of the testbenches designed to test the multiplier functionality. Conclusions are drawn in Section IV. The design, simulation, and layout are done using TSMC 65nm process node.

II. Design Methodology

We need to select a full adder topology that we will implement in the 16-bit multiplier design. First, we chose 28 transistors full adder topology shown in Fig. 1. This design has identical NMOS and PMOS. This design has three inputs (A, B, and CIN) and two outputs (SOUT and COUT). This design is used as a carry-propagate adder (CPA) cell where each sum bit depends on all previous carries. We combined a CPA cell and a 2-input AND gate to create a carry-save adder (CSA) cell. The AND gate forms the partial product, and the full adder adds the partial product into the running sum. Fig. 2 presents the schematic of the AND gate. In Fig. 3, the structure of the CSA is shown. CSAs are used in the fast multiplier design to sum the partial products. Fig. 4 depicts the 8 X 8 rectangular array multiplier for unsigned numbers using an array of CSAs. The initial row transforms the first partial product into the carry-save redundant format. Subsequent rows employ the CSA to add each corresponding partial product to the carry-save redundant outcome from the prior row, producing a new carry-save redundant result. The least significant output 8-bits are readily accessible as sum outputs directly from CSAs. The most significant output bits are delivered in carry-save redundant form and necessitate an 8-bit CPA for conversion into standard binary form. The CPA is implemented as a carry-ripple adder in Fig. 4.

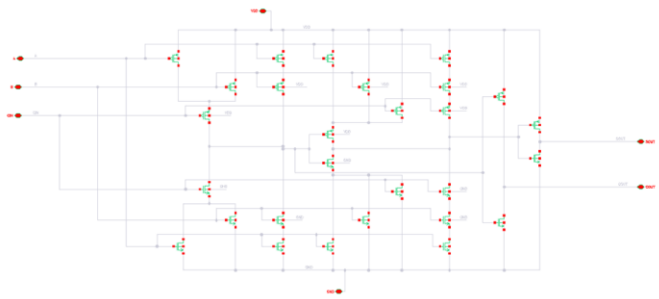

Fig. 1. Schematic of the full adder design used in the CPA cell.

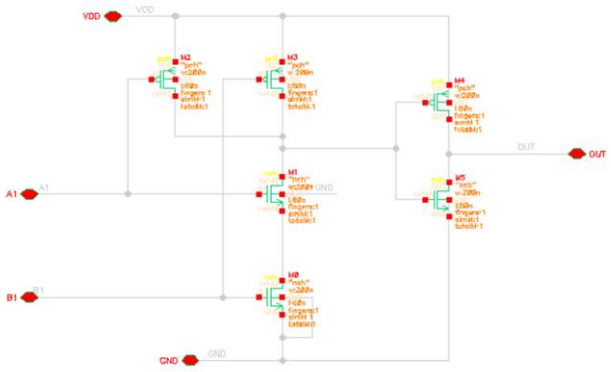

Fig. 2. Schematic of the 2-input AND gate.

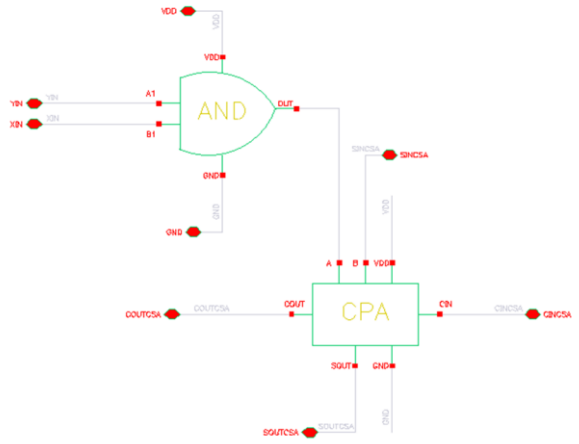

Fig. 3. Schematic of the CSA cell by combining 2-input AND gate and CPA.

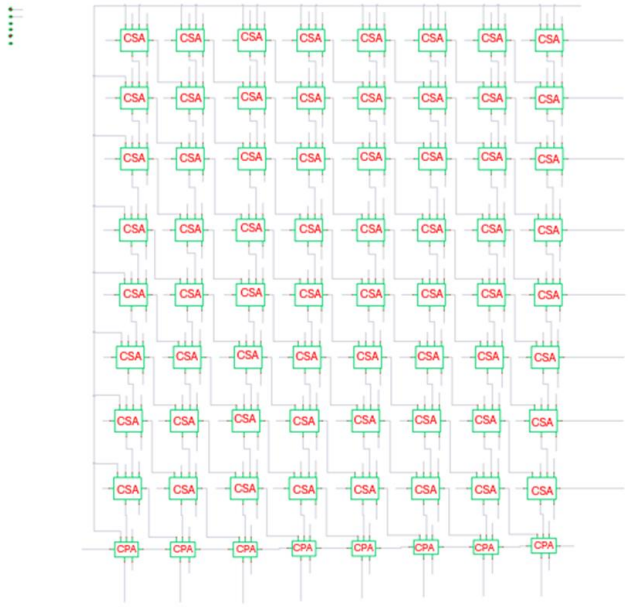

Fig. 4. Rectangular array multiplier.

The layout for the CPA cell and a 2-input AND gate is shown in Fig. 5 and Fig. 6, respectively. Guard-ring is used for both the NMOS and PMOS devices. Fig. 7 presents the layout for the CSA cell by combining the 2-input AND gate and the CPA cell. The complete 16-bit unsigned multiplier layout is shown in Fig. 8 where all the adder blocks are placed in an array pattern to create a rectangular floorplan. This layout technique reduces the area and improves the performance because of the

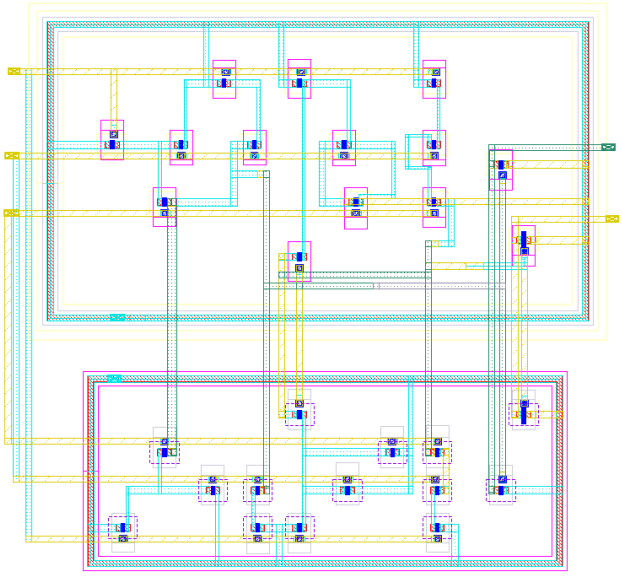

Fig. 5. Layout of a full adder (CPA).

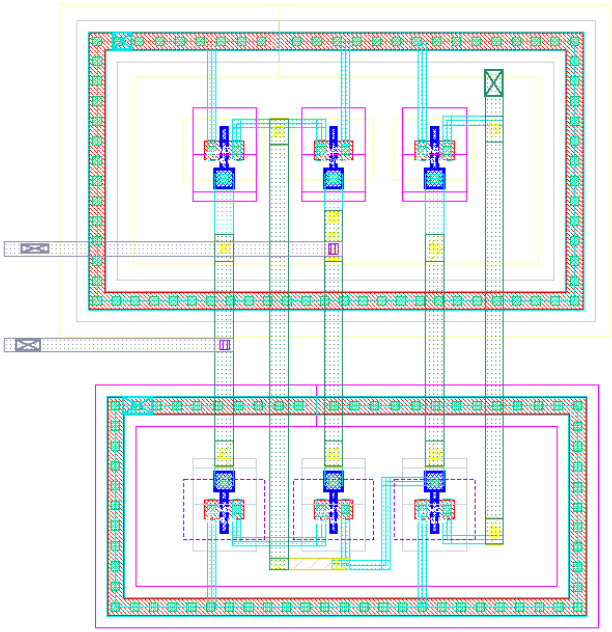

Fig. 6. Layout of a 2-input AND gate.

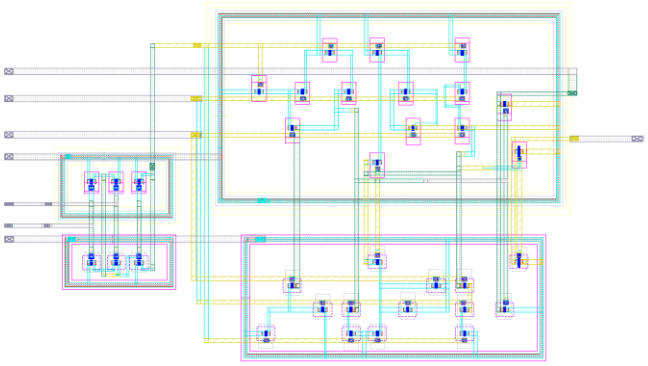

Fig. 7. Layout of a CSA cell using a 2-input AND gate and a CPA cell.

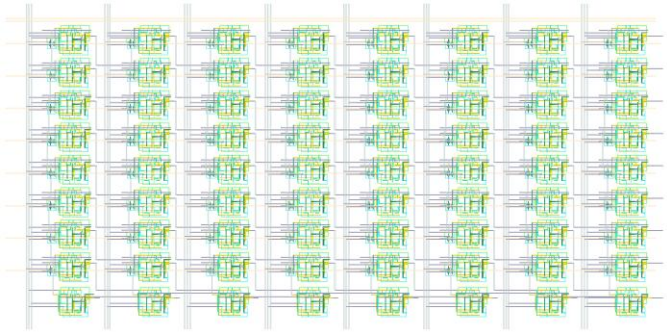

Fig. 8. Layout of the 16-bit rectangular array multiplier.

short wires with low wire capacitance. The design rule check (DRC) and layout versus schematic (LVS) are performed on the multiplier layout and shown in Fig. 9 and Fig. 10, respectively.

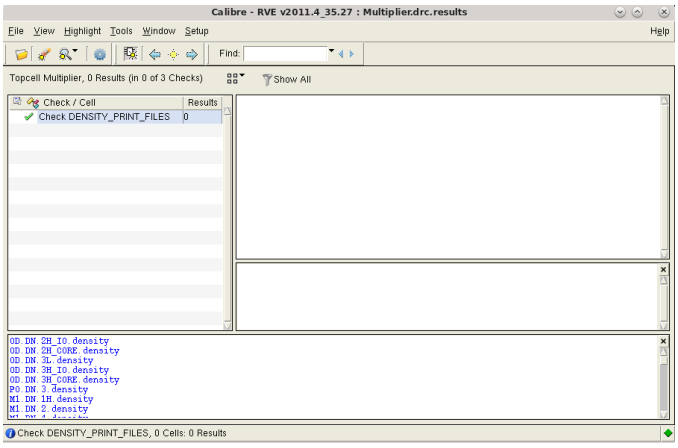

Fig. 9. DRC clean.

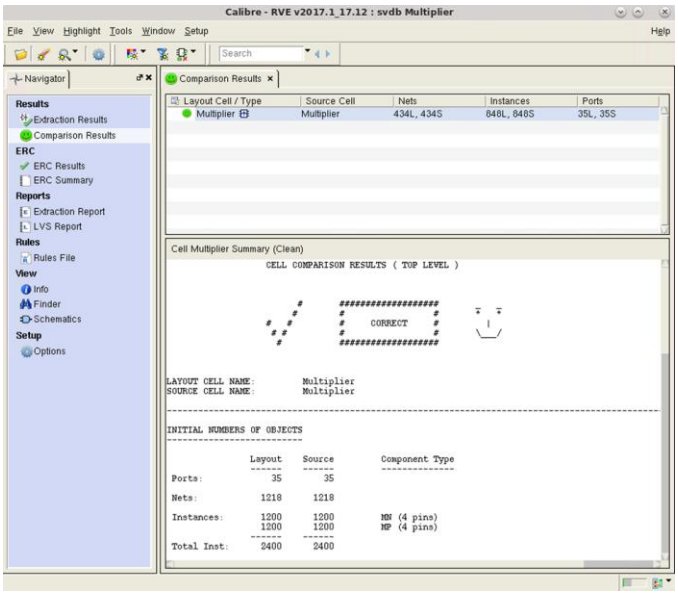

Fig. 10. LVS clean.

III. Simulation Results and Discussions

At first, the multiplier is simulated by creating a testbench schematic cell-view using pulse signals with different periods for both inputs X<7:0> and Y<7:0> to check the functionality. Then, we developed a Verilog code to implement within a testbench that fully tests the 16-bit multiplication via random number generation (RNG). The random number generator Verilog code is shown in Fig. 11. The code generates two random numbers at each positive edge of the clock cycle. From this Verilog code, we created a symbol for the RNG shown in Fig. 12 and connected it to the multiplier as input signals to test the functionality using the analog mixed signal (AMS) simulation. This test-bench structure is shown in Fig. 13.

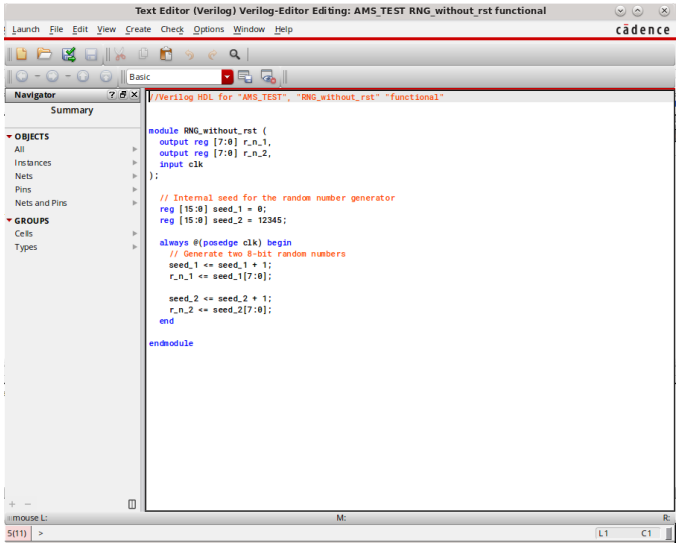

Fig. 11. Verilog code for RNG.

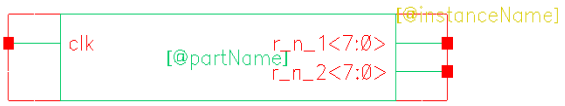

Fig. 12. RNG symbol.

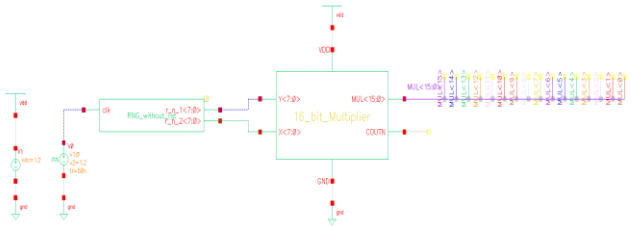

Fig. 13. Multiplier test-bench with random number generator (RNG) block.

Fig. 14 and Fig. 15, respectively, show the waveform of two 8-bit random numbers that were generated by each clock cycle once the testbench was simulated. We did the transient simulations for 500µs and the clock period was set to 60 µs. The multiplier successfully multiplies two random 8-bit inputs at each clock period and shows the 16-bit product at the output node MUL<15:0> plotted in Fig. 16. To understand the functionality of the system, we created input and output buses from these waveforms and plotted them in Fig. 17 up to four clock cycles.

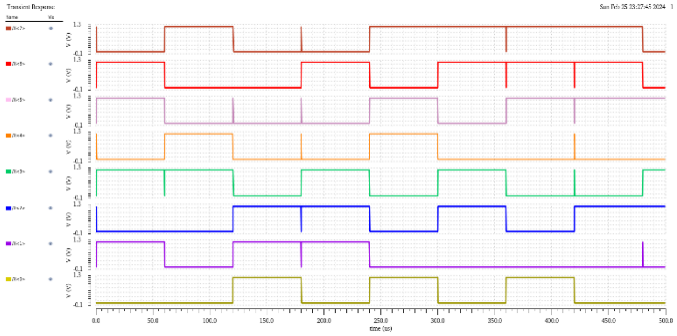

Fig. 14. Random number 1 generated using the RNG Verilog code (X<7:0>).

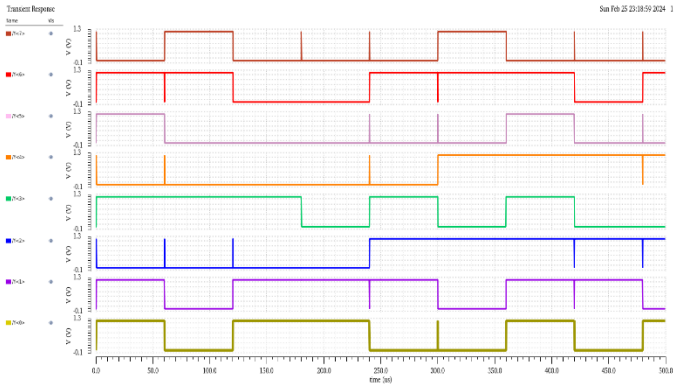

Fig. 15. Random number 2 generated using the RNG Verilog code (Y<7:0>).

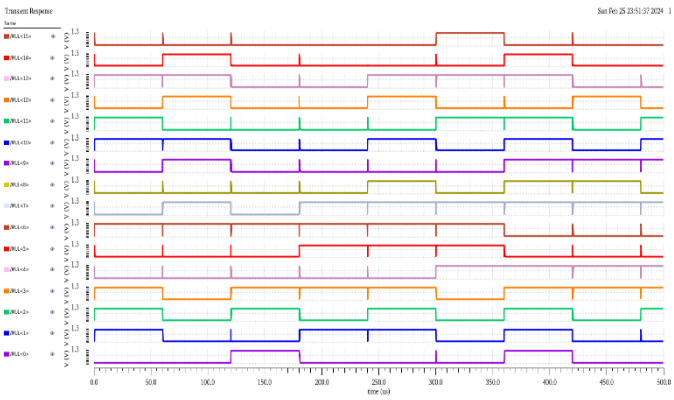

Fig. 16. Multiplier result generated from two 8-bit random numbers (MUL<15:0>).

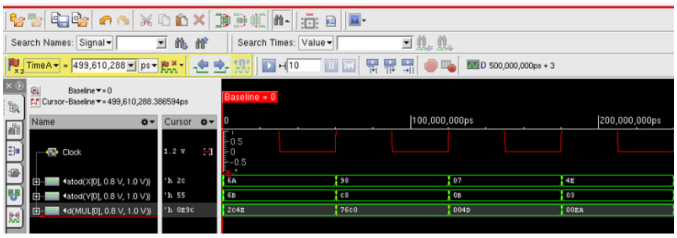

Fig. 17. Input and output wave bus (Hexadecimal format).

IV. Conclusions

This paper shows a successful design, layout, and testing of a 16-bit unsigned multiplier design using TSMC 65nm process node. Rectangular array multiplier topology is used using CSA and CPA adder cell to implement the multiplier. The designed multiplier has been tested and verified to multiply two 8-bit numbers through schematic testbench simulations using pulsed waveforms as well as an RNG functional block developed by Verilog code for the two 8-bit input generation at each clock cycle. Simulation results verified the functionality of the 16-bit unsigned multiplier design. Finally, the layout has been done for the multiplier which passed both DRC and LVS check.

References

[1] N. H. E. Weste and D. M. Harris, CMOS VLSI Design: A Circuits and Systems Perspective, 4th edition, 2011.

[2] J. Dix, Integrated Circuit Design Lab I: Class Lecture, Spring 2024.