Digital Sign Calculator Verification and Implementation of Verilog HDL via FPGA Simulation

Dr. Sarabjeet Kaur
Aditya Soraut
Amit Verma
Abhishek Kumar

Abstract:-
It is a possible to set up a digital sign calculator in the course of daily living. In this exploration composition, the pattern of a sign calculator is brought forward by using a format of Verilog HDL. Much synthetic Verilog code was created and obtained in Artix-7. The design of a 9-bit digital sign calculator can break fine mathematical operations similar to adder, subtract, multiplier, and divider. This digital sign calculator consists of nine-digit figures. The simulation procedure will be carried out in this article using devices from the Xilinx family. A shaft-style simulation of the calculator design is one of the outputs displayed in the waveform and RTL view. Calculator facts have been used favorably using Verilog HDL in relation to the behavior of the sign calculator functions.

Keywords-
Digital Sign Calculator, Circuit design, Verilog, Mathematical function

I. Introduction

The digital sign calculator consists of computer programs and hardware that can do basic arithmetic operations, such as adder, subtractor, multiplier, and dividor. In the scientific calculator there are numerous high-potential tasks, such as logarithmic, exponential, trigonometric, and hyperbolic functions, which might simplify challenging mathematical tasks.The development of calculators reduced the amount of time needed to explain complicated numbers and the mistakes that occur when calculating integers by hand. Scientific calculators included a number of extended operations, such as logarithmic, exponential, trigonometric, and hyperbolic Functions, that can break through complex mathematical functions. Binary format is also used for the calculators' input data. Additionally, the connected circuit will translate the decimal numbers into the base-two binary number system. Double data strings are used in integrated circuits to regulate transistors so that mathematical calculations may be performed. Mathematical tasks are completed, and the binary data is returned to the basic ten systems. The conclusion will show on the display screen. A complicated circuit is created by combining many logical gates to build adders. Numerous mathematical operations, including addition, subtraction, multiplication, division, and more, may be carried out by different combinations of logical gates inside the chips.

The creation of quicker, physically smaller designs with a greater number of gates and smaller sizes presents substantial problems for digital designers. To operate the device at the worst possible temperature, FPGA designers must produce designs that satisfy crucial requirements that other designers can comprehend. Furthermore, it is evident that the process variable circumstances may match the specifications and do not surpass the power consumption objectives. They are also reliable and verifiable. As a result, the designer writes and synthesizes hardware description language (HDL) code, which is then executed in the hardware chips. Verilog is crucial for business testing and replication of the outputs since it

DOI: https://doi.org/10.5281/zenodo.15273110

prevents mistakes. Free experiments are available in the Verilog simulation, and software tools for evaluating FPGA logic devices are also free. Hardware An overview Systems based on the chips with schematics have been developed using languages like Verilog HDL and VHDL. Signal logic is carried out by RTL, the design abstraction that controls the modeling of digital signals between hardware payrolls and synchronous digital circuits. Numerous studies on calculator design have been conducted, including RISC (64-bit) calculator design using Verilog HDL and FPGA Board; translation of splitting algorithms into Verilog programming language; FPGA prototype and a basic 9-bit calculator design bit slicing technique; updated non-restoring square root calculators with reduced VHDL code; and a novel approach for creating FPGA-integrated square root calculators.

II. The Flow Design and Data Type

The data types' block illustrations and the computational design processes are shown in Figure 1.Initially, the design was provided the input values; For data processing reasons, the structure is found in a decimal shape and transformed into a binary format. The digital sign calculator may be used for multiplication, division, addition, and subtraction, among other binary operations. The push-button portion of the synchronous unit has transformed the data from the asynchronous unit.A binary-to-decimal format is created for the unit operating at the 7-segment unit using the computed output data in binary format. The first result of the mathematical operations is created by indicating the number by illuminating different positions of the seven-segmented LEDs. These are nine sets of output LEDs with seven segments each. Eight of them display the number derived from the computation's outcome. The other, on the far left, displays the number indicator.

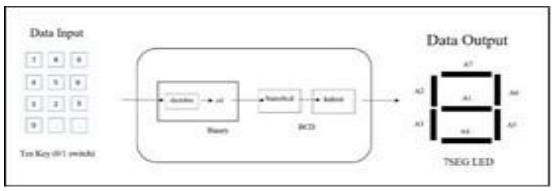


Figure 1. Flow and Data Type

The digital sign calculator's block depiction is seen in Figure 2. The investigation and advancement are performed utilizing Verilog HDL with the Xilinx ISE 14.0 computer program. In the Verilog HDL programming language, the calculator plan is divided into many significant corridors, each of which is programmed in a distinct module and activated using the RTL view and waveform lines. The "sign calculator" module is the best work module for the digital sign calculator in this business strategy.

All computing handling and capacities run in this module. As appeared in Figure 2,In the process of converting the input data from decimal to binary format, inputs from the "push" section were received in the blocks based on the block illustration. The "DECIMAL" function performs the information computation. This function stacks the twice supplied data to register A. Sign up The amount of memory consumed is returned by a function. For processing reasons, Register B stored the input number memory on the seven-segment LED screen. Implemented, reset (Rst), sign control(S), mode of operation (M), add, subtract, multiply, and divide are among the operators.

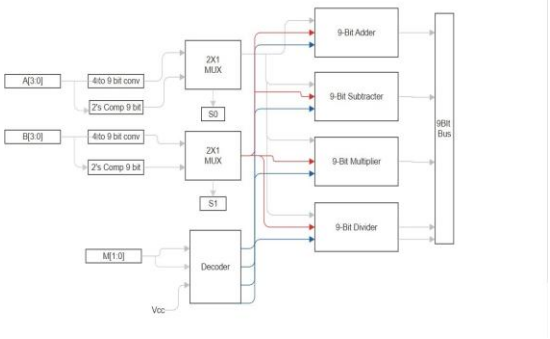


Figure-2. Sign Calculator block diagram

III. Bit Width Register

To ensure that the calculator's design can execute eight-digit computations, it is essential to evaluate the bit width range of the registers.The number of digits in a register is its bit width. Enroll A's inputs fall between 0 and 9999, meaning that in its two-fold version, they

DOI: https://doi.org/10.5281/zenodo.15273110

fall between 0 and 10111110101111000001111111. -199999998 (-99999999-99999999) to 199999998 (99999999+99999999) is the range of values in the decimal frame. When converted to its two-fold structure using the values of 199999998, it has 28 bits and is 10111110101111000001111111. The formula for the sum of all bit ranges is 2 . Then let n be the representative of the number of bits, and 2 = 2^{28} = 268435456.2 = 2^{28} = 268435456. The binary representation of the decimal 268435456 is 10000000000000000000000000. To arrange, the memory address has to be in the two-fold format.

IV. Overflow

When the number estimate exceeds the maximum bit area allowed in the registers, an overflow occurs. In order to prevent overflow, the articles require that entries be limited to a range between -999999999 and 999999999. The design overflow is calculated based on the decimal mode behavior. With the signed binary form of 0000_0101_1111_0101_1110_0000_1111_111 1, the input values must be greater than -999999999. Converting to its 2's complement yields the following value: 1111_1010_0000_1010_0001_1111_0000_000 1.

Therefore, when its 2's complement double value is translated back into decimal form, it becomes 4194967297. However, as this is the maximum value for a nine-digit calculator, the input values must not be greater than 999999999.

V. The Synchronous Circuit

Flip-Flop is made up of an edge detector circuit and a latch with an activation input. A signal clock serves as the edge locator's input. A square shaft with a fixed frequency is called a signal clock. The edge circuit for producing impulses during ascent is a positive edge-trigger flip-flop.

In the nine-digit calculator, flip-flops share a common reset and a signal clock, and an n-bit shift register is linked in series. This ensures accurate numerical operations by producing a counter where only one bit changes value between two consecutive counts. The offbeat flag "reset," which starts the count, is shared by all of the flip-flops.

D flip-flops make up the n-bit circuit; each flip-flop stores data on its own. These flip-flops can store and perform multi-digit computations efficiently since they are linked in parallel and share a reset and signal clock.

VI. Result And Discussion

The Modulus-designed RTL viewer is shown in Figure 4 at the higher value of the code in the appendix. "Sign calculator" is the name of the top-level module. There are fifteen outcome pins and fourteen input pins in the blueprint.

The syncro blocks operate based on the function selected by the user, directing the data into the ‘sign calculator’ module. In the meantime syncro10 will work based on the number entered by the user. Ultimately, all of the data enters the "sign calculator" module, which carries out the primary arithmetic operations, including addition, subtraction, multiplication, and division. This module also manages the bit width of registers and overflow detection.

The waveform summary in the Report Timing part of the circuit's timing performance is displayed in Figure 4. The Nexys A7 FPGA was used to implement the sign calculator blueprint. It enables effective data management and contains the most registers overall. The quantity of data the system can process at any given time depends on the size of these registers.

A time request for a cycle of 20.0 ns is provided by this circuit. With a positive setup slack value of 12.711 ns, the design satisfies the time requirements. The time required to extend the clock signal from the device pins to the target/source flip-flop's clock signal is known as the tactful delay, and it is 4.985 ns.

There is a 5.758 ns data delay. This illustrates how long it takes for a signal to get from its source to the intended flip-flop. The waveform illustration guarantees that the time restriction is fulfilled and that the data reaches the target before the required amount of time.

DOI: https://doi.org/10.5281/zenodo.15273110

Register A may receive the alternative input values. Register B is where the final values are entered. The calculated data is sent to register B. In order to process the result, Register B has stored the input number in memory. As with the expansion, let's say the customer selects the operation. In this instance, a high or active state is indicated by the multiplexer changing from 0 to 1. Division, multiplication, and subtraction are further operations.

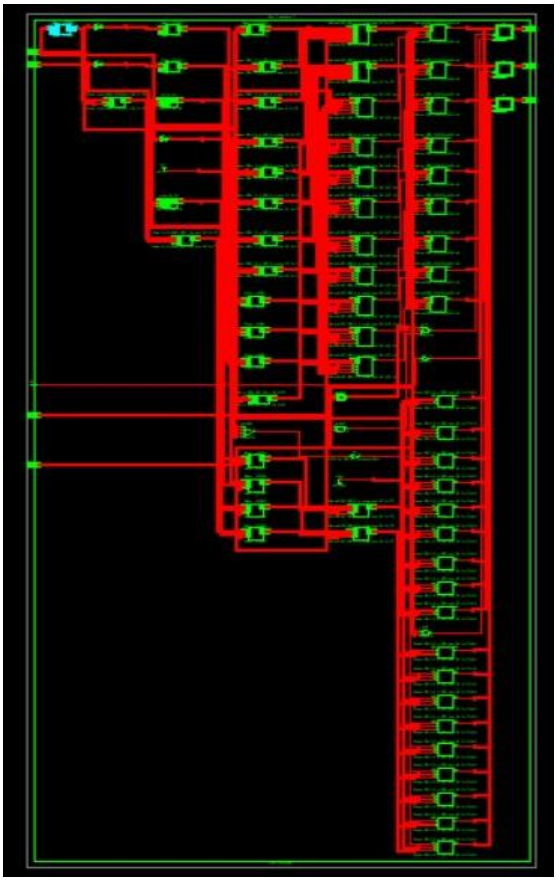


Figure-3. Top-level module RTL view ‘sign calculator’

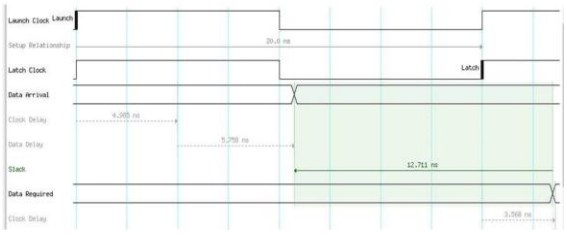


Figure-4. The Report Timing section of the circuit’s waveform

An example of a waveform simulation with increment is shown in Figure 6. 20 ns per period was the timer's initial setting. The "equal" waveform was consistently raised to the highest possible level. The "reset" waveform was inserted into a d flip-flop type to manage the calculator design flux, depending on the proper circumstances. Pressing the double format value into a decimal waveform fits the desired values within the calculator's architecture. By placing their waveform in an advanced position, the desired chosen processes may be given names.

To enlist A, double input data is stacked into this function. The number of customers in the memory is returned by the Enlist A function. The first input number is taken by Enrol A. Subsequently, the input number that is substituted begins at 0. once the customer choose what they want. Figure 5 shows that 14 and 7 are the two inputs from the "decimal" signal that are stacked to enroll A. "Add," which stands for the addition operation, is the administrator selection. The calculated result yields at the "out" waveform and is stored in registers B and 21. This suggests that 21 is the consequence of adding the numbers 14 and 7. The system used for the final interpretation is the same for all other steps.

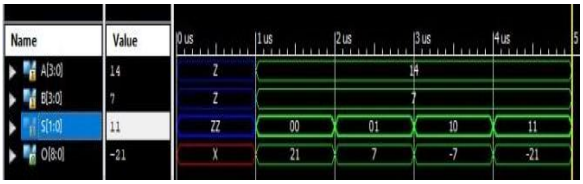
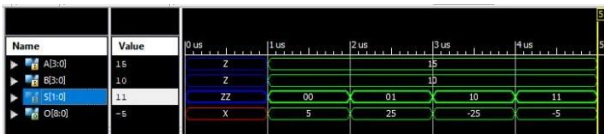


Figure-5. The Waveform of the ‘add’ function simulation.

The waveform reconstruction of the unique functions of the calculator design using Verilog HDL in this study, such as independent division, multiplication, and deduction, is shown in Figures 6, 7, and 8.



DOI: https://doi.org/10.5281/zenodo.15273110

Figure-6. The Waveform of the ‘sub’ function simulation

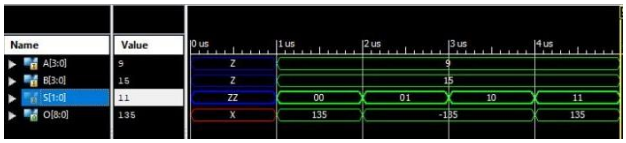


Figure-7. The Waveform of the ‘mult function simulation

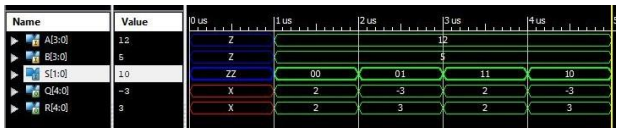


Figure-8 the Waveform of the ‘div’ function simulation

VII. Conclusion

This paper's goal is to plan the digital sign calculator's four integers and use the administrators in Verilog HDL. Verilog HDL programming was used to encode the calculator, and Xilinx ISE 14.0 was used to deconstruct it. In order for designers to create schematic chip frameworks that can be implemented into Field Programmable Gate Arrays (FPGAs), Verilog HDL is essential. By carrying out operations in a more advantageous manner than other languages, Verilog HDL appears to be able to represent and describe the self-assertive collection of computerized circuits. The device chosen from the Altera family of devices is the Nexys A7.

The calculator design has several limitations when it comes to constructing complicated digital circuits utilizing Verilog HDL because of comparable considerations as a lack of software and installations. Since the calculator's designs can only generate a 9-digit output and induce an integer answer, the outcome is not ideal if the answer contains decimal numbers. It is suggested that unborn work would produce a more accurate outcome; this is part of the floating-point process. Finally, it is advised to use more mathematical processes, such as logarithmic, to deconstruct intricate fine mathematical issues.

VIII. Confirmation

I would want to express my gratitude to Noida Institute of Engineering and Technology for giving me the space and chance to create this module.

IX. Source

[1] Vranesic, Z. & Brown, S. Digital Logic Fundamentals with Verilog Design. McGraw-Hill Learning.[2013]

[2] Verilog HDL: An Introduction to Digital Design and Synthesis by S. Palnitkar. Pearson Learning.[2003]

[3] Digital Design: Principles and Practices by J. F. Wakerly. Pearson Prentice Hall.[2005]

[4] Kinney, L. L., & Roth, C. H. Designing Digital Systems with Verilog. Cengage Education.[2015]

[5] Digital Design with an Introduction to the Verilog HDL by Mano, M. M., and Ciletti, M. D. Pearson.[2017]

[6] Casio.http://www.vintagecalculators.com/html/casio.html is the access point. [2019].

[7] Vivado Design Suite: Synthesis, Xilinx Inc. https://www.xilinx.com is accessible.[2023]

[8] IEEE Standards Association, IEEE VHDL standard.[2005].

[9] ModelSim-Intel FPGA Edition User Guide, Intel Corporation. https://www.intel.com is the link.

[10] Xilinx Inc., Logic Simulation (UG900), Vivado Design Suite User Guide, 2023. [Online]. https://www.xilinx.com is accessible.

[11] Basys 3 Reference Manual, Digilent Inc. https://reference.digilentinc.com is the URL. [2022]

[12] [12] All About Circuits, "Comprehending Binary Arithmetic and Two's Complement." https://www.allaboutcircuits.com is the link.

[13] "Verilog Code for Arithmetic Units and ALUs," FPGA4Student. https://www.fpga4student.com is the link.

DOI: https://doi.org/10.5281/zenodo.15273110

[14] "Calculator's History."

[15] This is the URL: https://www.thecalculatorsite.com/articles/units/history-of-the-calculator.php. [2019].

[16] "Basic Logic Gates: A Digital Logic Gate Tutorial." https://www.electronicstutorials.ws/logic/logic_1.html is the link.[2019].

[17] "What Is the Process of a Calculator?"[Online].

[18] https://www.wonderopolis.org/wonder/how-does-a-calculator-work is what is accessible. [retrieved: November 20, 2019].

[19] Real world design with verilog "Link: https://www.learnabout-electronics.org/Digital/dig21.php" [2019].

[20] K.Coffman, FPGA design in the real world using Verilog. Prentice Hall PTR, Upper Saddle River, NJ, 2000.

[21] The link to register for the transfer level is https://semiengineering.com/knowledge_centers/edadesign/definitions/register-transfer-level/. [2020].

[22] Huang, J. L. & F.Cauley, S. [2009]. Chapter 8. In C. K. Koh (Ed.), Logic and circuit simulation (pp. 449-512).

[23] Furman, M. D., and Hamblen, J. O. [2002]. Chapter 3 of Digital Systems Rapid Prototyping: A Guide (pp. 37-52). US: Springer, Boston, MA.

[24] Mathew, B., Stopyak, M., and Wagner, M. (2017) Final Project Paper for Simple Calculator ECE 2700-Digital Logic Design. [2017] ECE 2700: Digital Logic Design Final Project Paper: Simple Calculator.

[25] Colinares, D. S., Lousie, J. D., L., I. E., Q., E. R., & Caldo, R. B. (n.d.). Trigonometric operations are included in this 12-bit Verilog calculator. Trigonometric function 12-bit Verilog calculator, 3(No.), 43–48.

[26] Idris, N. R., Sutikno, T., Jidin, A. Z., and Jidin, A. (2012). Modified Non-Restoring Square Root Calculator with Simplified VHDL Coding. IJRES

[27] Shelke, M. G., Jaunjare, N. V., & Penshanwar, S. K. (2017). Calculator design with RISC (64 bit) architecture using VERILOG and FPGA. Calculator Design with RISC (64 Bit) Architecture Using VERILOG and FPGA.

[28] Yusmardiah, Y, Darmawaty, M., A, Abdul Karimi, H, Abdul Aziz, A., R, Ahmad, K., S. Translation of division algorithm into v h d l .translation of division algorithm into verilog HDL.[210

DOI: https://doi.org/10.5281/zenodo.15273110