VERILOG IMPLEMENTATION OF MOVING SUM CALCULATOR

R.Aditya
Dept. of EECE, GITAM Deemeed to be University
S.Sanath Kumar
Dept. of EECE, GITAM Deemeed to be University
S.V.S.Sumanth
Dept. of EECE, GITAM Deemeed to be University
M.M.Ali Baig
Dept. of EECE, GITAM Deemeed to be University
S.Maneesha
Dept. of EECE, GITAM Deemeed to be University

Received: Jan. 2020 Accepted: Feb. 2020 Published: Feb. 2020

Abstract: System-on-a-chip (SOC) has become an essential technique to lower product costs and maximize power efficiency, particularly as the mobility and size requirements of electronics continues to grow It has therefore become increasingly important for engineers to develop a strong understanding of the key stages of hardware description language (HDL)
Moving sum calculator is type of device which computes the sum of specified number of bytes when sequence of bytes are coming in for every clock and computation is performed on the most recently arrived data bytes eliminating the previous bytes on the basis of FIFO. Proposed moving sum calculator adds 16 bytes for the first 16 clock cycles and then for the next clock cycle when the 17^{th} byte comes in then the first byte is eliminated from the result and adds the 17^{th} byte to the result and the resultant sum will be computed and it does the same for the next consecutive bytes also for every clock cycle i.e. then onwards for every clock cycle it computes the sum of most recent 16 bytes. Performance of this proposed moving sum calculator for a sequence of data bytes is presented in the simulation results. It has applications of data extraction and analysis in industries, television signals, computer icons, stock markets etc.

I. Introduction: VLSI chiefly comprises of Front End Design and Back End design these days. While front end design includes digital design using HDL, design verification through simulation and other verification techniques, the design from gates and design for testability, backend design comprises of CMOS library design and its characterization. It also covers the physical design and fault simulation. While Simple logic gates might be considered as SSI devices and multiplexers and parity encoders as MSI, the world of VLSI is much more diverse. Generally, the entire design procedure follows a step by step approach in which each design step is followed by simulation before actually being put onto the hardware or moving on to the next step.

Verilog is a Hardware Description Language; a textual format for describing electronic Circuits and systems. Applied to electronic design, Verilog[1] is intended to be used for Verification through simulation, for timing analysis, for test analysis (testability analysis and Fault grading) and for logic synthesis. The Verilog HDL[2] is an IEEE standard - number 1364. The first version of the IEEE standard For Verilog was published in 1995. A revised version was published in 2001; this is the Version used by most Verilog users. The IEEE Verilog standard document is known as the Language Reference Manual or LRM. This is the complete authoritative definition of the Verilog HDL.A further revision of the Verilog standard was published in 2005, though it has little extra compared to the 2001 standard. System Verilog is a huge set of extensions to Verilog, and was first published as an IEEE standard in 2005

IEEE Std 1364 also defines the Programming Language Interface, or PLI. This is a collection of software routines which permit a bidirectional interface between Verilog and other languages (usually C). Note that VHDL is not an abbreviation for Verilog HDL [3] - Verilog and VHDL are two Different HDLs. They have more similarities than differences; however**.** Verilog was started initially as a proprietary hardware modeling language by Gateway Design Automation Inc. around 1984. It is rumored that the original language was designed by taking Features from the most popular HDL language of the time, called Hilo, as well as from Traditional computer languages such as C. At that time, Verilog was not standardized and the Language modified itself in almost all the revisions that came out within 1984 to 1990.Verilog Simulator were first used beginning in 1985 and was extended substantially through 1987. The implementation was the Verilog simulator sold by Gateway.

II. Problem Statement: The idea is to design a Moving sum Calculator which can compute the sum of a sequence of numbers that is updated each time a new number is added to the sequence, it subtracts the very first byte of the sequence from the running sum and adds the new byte to the previous running Sum. In this paper the objective is to design such a moving sum calculator. This circuit receives a byte at every positive transition of the master clock. It generates the output of sum of the last 16 bytes received as a 12-bit output. The design uses a 12-bit subtractor to subtract the byte which was received 16 clocks ago, and a 12-bit adder to add the current byte.

The data is first latched into an 8-bit latch. The latch output is fed to a 12bit adder, as also to a 16 stage 8bit Shift Register. After the n^{th} byte has been latched by the clock pulse, the last stage of Shift Register contains data [n-16]. This is subtracted from the latched output of the adder. The output of the subtractor forms one input to the adder. The output of the adder is presented as the moving window sum.

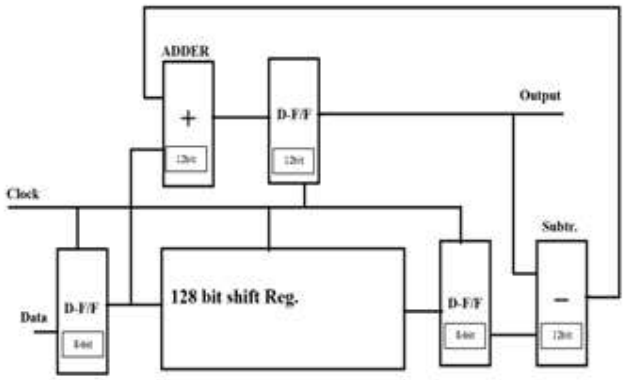

Fig. 1: Block Diagram

III. Shift Register:

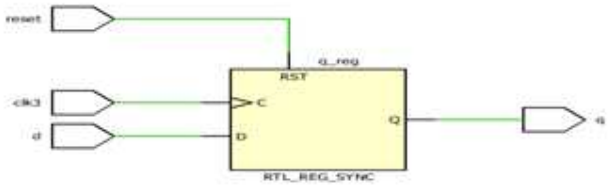

Fig. 2: RTL Schematic of D Flip-Flop

The shift register is essentially made with a group of Flip-flops. The main purpose of the shift register is to shift the data. It takes one clock cycle to shift one bit of data and for 128 bit it takes 128 clock cycles. It can shift up to 16th bytes and when the 17th byte is entered the 1st byte is eliminated.

Flip flops can be used to store a single bit of binary data (1or 0). However, in order to store multiple bits of data, we need multiple flip flops. N flip flops are to be connected in an order to store n bits of data. A Register is a device that is used to store such information. It is a group of flip flops connected in series used to store multiple bits of data. The information stored within these registers can be transferred with the help of shift registers. Shift Register is a group of flip flops used to store multiple bits of data. The bits stored in such registers can be made to move within the registers and in/out of the registers by applying clock pulses. An n-bit shift register can be formed by connecting n flip-flops where each flip flop stores a single bit of data.

IV. Adder & Subtractor:
A) Adder

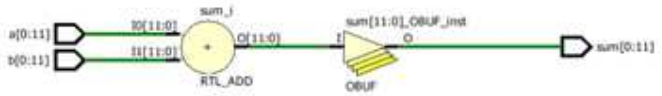

Fig. 3: RTL Schematic of 12-Bit Adder

Adders [4] are used for the addition of 2bytes of data. A 12bit adder is used to perform the addition of new data bytes and previous sum, resultant is stored in a flip-flop, An adder is a digital circuit that performs the addition of numbers. In many computers and other kinds of processors, adders are used in the arithmetic logic units or ALU. They are also used in other parts of the processor, where they are used to calculate addresses, table indices, increment and decrement operators and similar operations.

B) Subtractor

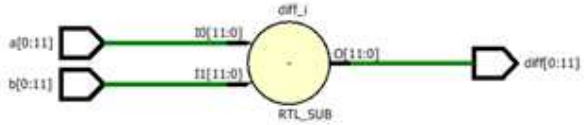

Fig. 4: RTL schematic of 12-Bit Subtractor

Subtractor is used for the subtraction of 2bits of data, we use 12bit subtractor after the 16 clock cycle it subtracts the sum and the first data byte. A full subtractor is a combinational circuit that performs subtraction of two bits, one is minuend and the other is subtrahend, taking into account borrow of the previous adjacent lower minuend bit.

V. Buffer: Buffer is the basic storage element in sequential logic circuits. This is used to delay the outputs or to store the data. In this, the buffer used is D Flip-flop,12-bit &8-bit D Flip-flop used for storage purpose.The D Flip-flop captures the value of D-input at a definite portion of the clock cycle(such as the rising edge of the clock & falling edge of the clock). In this reset, clock, D is input & Q is the output, this work on the basis of clock & reset.

When the reset is high it set the flip-flop to the initial state or zero states and we can set the clock (either positive edge or negative edge) as per our requirements. Simple flip-flops can be built around a single pair of cross-coupled inverting elements like bipolar transistors, field-effect transistors, inverters, and inverting logic gates which are been used in practical circuits.

VI. Simulation Results:

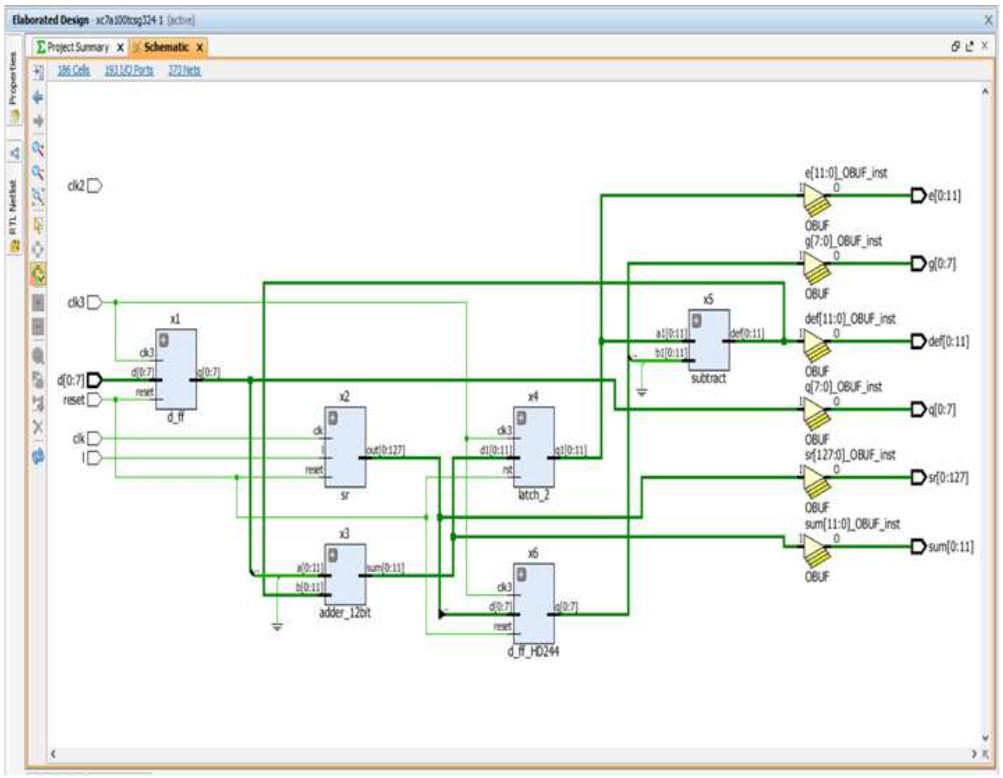

Fig. 5: RTL Schematic of Moving Sum Calculator

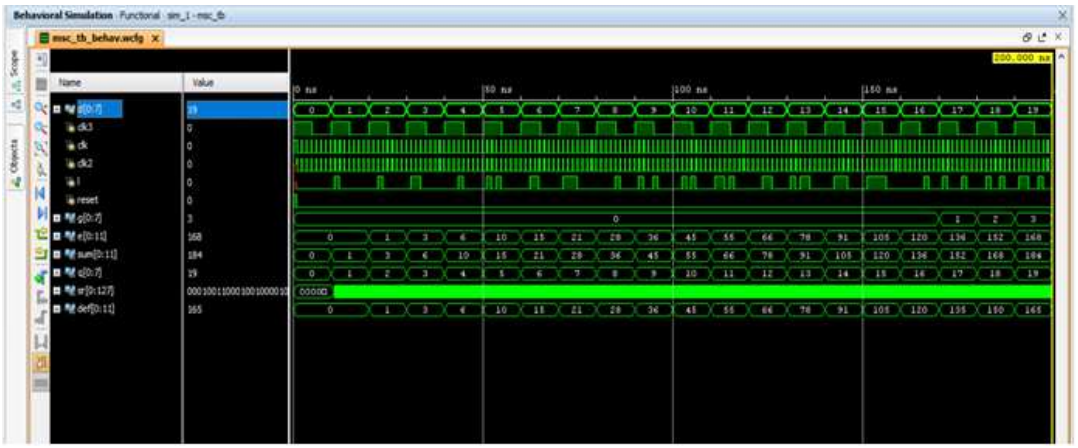

Fig. 6: Timing Diagram of Output of Moving Sum Calculator

The proposed design is modeled in Verilog and simulated on Xilinx Vivado. The RTL schematic of moving sum calculator and its output for the sequence of data byte are shown in fig.5 and fig.6 respectively. Here the sequence of data bytes 0,1,2.....15,16,7,18,19 are considered as test inputs. The sum of first 16 data bytes(0 to 15) is computed as 120 at the end of 16th clock cycle of clk3 and this sum is represented by e[0:11]. On the next clock cycle of clk3, next data byte of sequence (16) is added to the sum and the very first data byte (0) is subtracted which results in updated output sum as 136. On the next clock cycle of clk3,next data byte of sequence(17) is added to the sum and data byte(1) is subtracted

which results in updated output sum as 152 and the same operation continues on consecutive clock cycles of clk3 every time when new data byte comes in.

The signals d[0:7], q[0:11], g[0:7], sum[0:11], def[0:11] are representing data bytes, inputs for shift register, input to subtractor, output of the adder and output of the subtractor respectively. The output of moving sum calculator is represented by e [0:11].

References:

1. Gaurav Sharma, Lava bhargava, "MDAB: Module Design Automation Block for Verification using System Verilog Testbench",3rd International Conference and Workshops on Recent Advances and Innovations in Engineering, 22-25 November 2018.
2. LinYu, "Design of Bidirectional Bus Based on Verilog HDL", ICITBS 2019.
3. Phong X. Nguyen, "Verilog Library Development Using Cadence Central Delay Calculator",IEEE 1994.
4. Varun Partap Singh,Manish Rai, "Verilog Design of Full Adder Based on Reversible Gates",2016.
5. R Landauer,(1961) "Irreversibility and Heat Generation in the Computational Process,"IBM Journal of Research and Development, vol. 5, no. 3, pp. 183-191.
6. K. Prudhvi Raj and Y. Syamala, 2014, "Transistor Level Implementation Of Digital Reversible Circuits" International Journal of VLSI design & Communication Systems (VLSICS) Vol.5, No.6.
7. Dr. Esam Al-Qaralleh, "Introduction to Verilog Hardware Description Language", Princess Sumaya, University for Technology.
8. Thanasis Oikonomou. "Verilog HDL Basics Computer Science Department, University of Crete Greece. October 1998.
9. Samir Palnitkar. Verilog HDL, A guide to digital design and synthesis", Prentice HallProfessional, 2nd Edition, Sunsoft Press 1996.

***