International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 09 | Sep 2020 www.irjet.net p-ISSN: 2395-0072

DESIGN AND IMPLEMENTATION OF 32-BIT UNSIGNED DIVIDER ON FPGA

Priyanka S D[1], Dr. Ambresh P Ambalgi[2], Sujata A A[3]

[1]PG Student Dept. of ECE Faculty of Engineering and Technology (Exclusively for Women) Sharnbasva University Kalaburagi, Karnataka, India.
[2]Assistant Professor Dept. of Electronics Mangalore University, Mangalagangothri Konaje, Dakshinakannada, Karnataka, India.
[3]Assistant Professor Dept. of ECE Faculty of Engineering and Technology (Exclusively for Women) Sharnbasva University Kalaburagi, Karnataka, India.

Abstract - : *Abstract during this project, a 32-bit unsigned divider is intended and enforced victimization verilog code in activity model. The divider is synthesizable and might be enforced on FPGA victimization verilog code. A divider could be a purposeful a part of arithmetic and logic unit. It's enforced to perform whole number primarily based division operations. The division is one in all the foremost difficult operations of the fundamental arithmetic operations. During this project we've designed a divider that performs specific division operation on thirty two bit numbers. As thought of, division operation is a lot of incredible within the calculation of the advanced applications. Binary and decimal division calculations are performed during this work. The synthesized schematic results are enforced on RTL compiler. Finally, the projected absolutely pipelined implementation architectures is co-simulated by Modalism and Simulink, and it's synthesized on FPGA victimization Verilog and VHDL.*

Key Words: 32 bit divider, binary division, VLSI design, FPGA.

1. INTRODUCTION

Division is one in all the four essential operations of arithmetic arithmetic. The division of 2 common numbers is that the manner toward ascertaining the amount of times one number is contained within one another. Divider is prime instrumentation utilizes in speedy and progressed advanced signal process (DSP) units. Divider is Associate in Nursing inevitable and basic instrumentation module utilised in advanced and quick processed flag handling (DSP) units of high accuracy. In distinction with different scientific operations, division is that the ordered quite operation that outcomes in complicated instrumentation usage. The exceptionally correct division calculations are the elemental necessity of signal and image handling applications. The various division styles are created to reduce the machine difficulties.

In this paper, to appreciate the whole number division rule in one single FPGA chip, we tend to adopted model-based style technology by victimization the Xilinx System Generator (XSG) chest. Once utilizing FPGA technology, programming is inescapable for researchers and engineers. This rule describes a straightforward manner of effecting the division for any variety of digits within the divisor, dividend and therefore the quotient, victimization easy multiplications and easier subtractions. Additionally it doesn't want any difficult prediction algorithms or expansive hunt tables. This greatly reduces all parameters adore power consumed, and resulting propagation delay thereby fulfilling all the objectives of VLSI. The performance of the changed thirty two bit unsigned division rule is analyzed victimization XILINX eight.1 ISE machine tools.

Transistor level implementation of such division electronic equipment was meted out by the mix of Boolean logic with binary arithmetic, performance parameters like propagation delay, dynamic power, and static power consumption. Binary division operation is of huge importance within the field of field. Inherently, division operation could be a ordered variety of operation, thereby it's a lot of pricey in terms of machine complexness and latency (propagation delay) compared with different mathematical operations like multiplication and addition.

2. PROPOSED 32 BIT DIVISION ARCHITECTURE

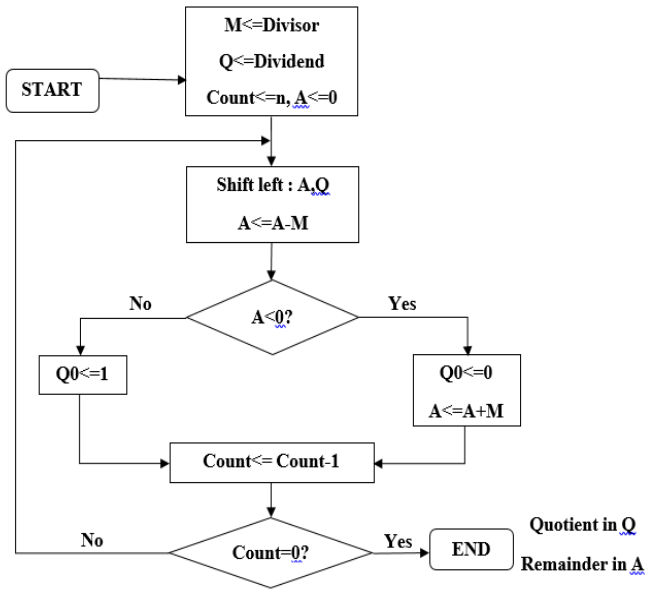

Figure -1: Flow diagram of divider.

A novel high performance pipelined implementation design for user-defined thirty two bit unsigned complicated division is bestowed. By the on top of diagram. Here it performs division operation between 32-bit divisor and 32-bit dividend. The operation is performed by shifting left dividend and subtracting from remainder in every iteration till remainder A become greater than or equal to zero

A	Q	M = 0011	
1111	1001	Initial values	
1111	0010	Shift	1
0010		Add	
1111	0010	Restore	
1110	0100	Shift	2
0001		Add	
1110	0100	Restore	
1100	1000	Shift	3
1111		Add	
1111	1001	$Q_0 = 1$	
1111	0010	Shift	4
0010		Add	
1111	0010	Restore	

(-7) / (3)

A	Q	M = 1101	
1111	1001	Initial values	
1111	0010	Shift	1
0010		Subtract	
1111	0010	Restore	
1110	0100	Shift	2
0001		Subtract	
1110	0100	Restore	
1100	1000	Shift	3
1111		Subtract	
1111	1001	$Q_0 = 1$	
1111	0010	Shift	4
0010		Subtract	
1111	0010	Restore	

(-7) / (-3)

Figure -2 : division operation example with 4-bits

2.1 How to Implement the Division Algorithm

As we are able to see from the on top of example, the division algorithmic rule repeatedly subtracts the divisor (multiplied by one or zero) from acceptable bits of the dividend. Therefore, subtraction and shift operations are the 2 basic operations to implement the division algorithmic rule.

After every subtraction, the divisor (multiplied by one or zero) is/are shifted to the correct by one bit relative to the dividend. For the circuit implementation, we are going to shift the dividend to the left instead of shifting the divisor to the correct.

2.2 How is division implemented in hardware

Integer division in hardware is completed through subtraction and shifting of digits specifically like typical longhand division within the crudest technique, whereas higher ways use algorithms that calculate the result a lot of faster and in fewer steps. The code written in the software is dumped on to FPGA kit to show the quotient part with LED glow.

2.3 What is division algorithm formula

The division algorithmic rule states that for any whole number, a, and any positive whole number b, there exists distinctive integers letter of the alphabet and r specified a = bq + r (where r is larger than or adequate zero and fewer than b). We tend to decision a dividend, b the divisor, letter of the alphabet the quotient, and r the rest.

3. BINARY DIVISION METHOD

To begin, contemplate associate example of dividing 11000101 by 1010.

Just as in decimal division, we are able to compare the four most vital bits of the dividend (i.e., 1100) with the divisor to seek out the primary digit of the quotient. We tend to are operating with binary numbers, that the digits of the quotient cn be either zero or one.

Since 1100 is larger than 1010, the primary digit of the quotient are going to be one. The obtained digit should be increased by the divisor and also the result must be subtracted from the dividend. Hence, we have

Now, we should write the next bit of the dividend (shown in red) to the right of the difference and continue the procedure just as we do in a decimal division. Hence, we obtain

The below example shows the decimal equivalent of the parameters as well as the letter used to represent them. We can verify the calculations by evaluating 'z = q × d + s'

and that s<d. First, the divisior is subtracted from the four most significant bits of the dividend the default of this subtraction i.e., 1010 is shown in blue

		10011	Quotient (q=19)
Divisor (d=10)	1010	11000101	Dividend (z=197)
		1010	
The 1st subtraction			
		0100	
The 2nd subtraction		0000	
		1001	
The 3rd subtraction		0000	
		10010	
The 4th subtraction		1010	
		10001	
The 5th subtraction		1010	
		0111	Remainder (s=7)

To get a better insight into the implementation of the division algorithm, we rewrite the above example as

We ca replace the four MSBs of the dividend with 0010 and obtain s(0)=00100101. The four LSBs of s(0), which are the same as the four LSBs of the dividend, are shown in red, note that we no longer longer need the original dividend and we can replace it with s(0). From an implementation point of view, this means that we can use the register which was originally storing the value of the dividend to store s(0).

For the second subtraction, the divisor is shifted to the right by one bit. After subtraction, we obtain (1)=00100101. Again, the bits obtained from subtraction are shown in blue and the unaltered bits of s(0) are shown in red. We can now update the dividend register with s(1).

		10011	Quotient (q=19)
Divisor (d=10)	1010	1100 0101	Dividend (z=197)
		1010	
The 1st subtraction			
$s^{(0)}$		0 0100 101	
The 2nd subtraction		0000	
$s^{(1)}$		0 01001 01	
The 3rd subtraction		0000	
$s^{(2)}$		00 10010 1	
The 4th subtraction		1010	
$s^{(3)}$		000 10001	
The 5th subtraction		1010	
$s^{(4)}$		0000 0111	Remainder (s=7)

This procedure goes on until the final subtraction in which the LSB of the shifted divisor is aligned with the LSB of the dividend. After this final subtraction, the remainder will be less than the divisor.

Note that, as we proceed with the algorithms, the high order bits of the s(.) terms become zero. This suggests that some bit positions of the dividend register will be no longer required. In the next section, we'll see which bit positions are redundant. In the above example, the bit positions that can be discarded are underscored.

4. RESULT

The algorithm discussed above is implemented in verilog HDL and it is simulated and synthesized using Xilinx ISE using FPGA family as Spartan 3E. The results are given as,

A. RTL schematic
B. Test bench waveform

A. RTL schematic

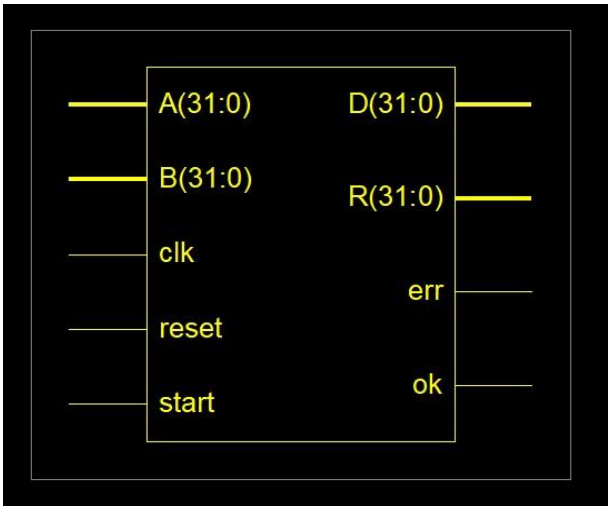

Figure -3 : RTL block schematic (D=Q above)

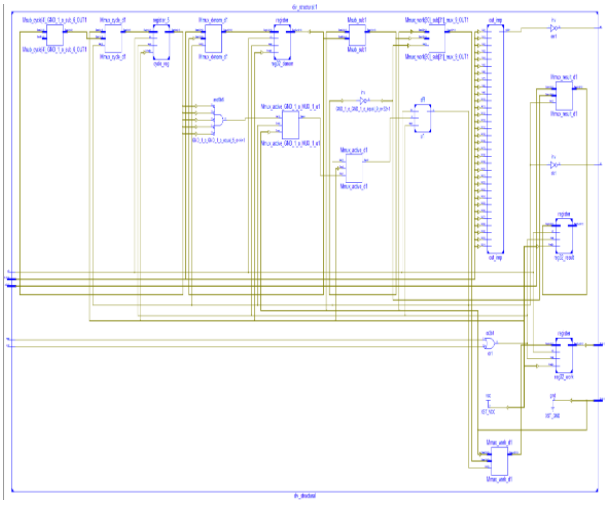

Figure -4 : RTL layout Schematic

B. Test bench waveform

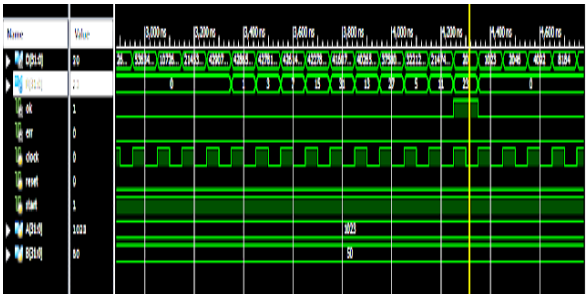

Figure -5 : Test Bench Waveform

5. FUTURE SCOPE

The performance of the system is generally measured by the performance of the divider because the divider is generally the slowest element in the arithmetic system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the divider is a major design issue. In future, Arithmetic based sub systems will consists of high speed counter using GDI technique and Vedic divider designs. They can be used in the implementation of DSP which is advantageous for low power applications.

6. CONCLUSION

In this paper we report on a novel 32 bit dividend by 32 bit divisor architecture based on the formulas of Mathematics. In this work, division operation was carried out through addition and small digit multiplication that eventually reduces the iteration, owing the achievement of high speed. The pipelined designs of user-defined binary division. The user-defined binary complex division using modified algorithms was done with a co-simulation techniques with Modalism and Matlab/Simulink. The whole pipelined architectures was synthesized using on FPGA. The fully pipelined implementation tremendously raises the system's throughput. Hence our designs could be applied to those field with high performance requirement. The design has been able to achieve best power inflation over the past work in the same field.s

ACKNOWLEDGEMENT

I would like to thank our dean Dr. Lakshmi Maka, Head of the Department, Prof. Nagaveni K and my coordinator(guide) Asst. Prof. Sujata kamlapurkar for their valuable advise, support and technical assistance

REFERENCES

[1] Implementation of 24-bit Vedic Multiplier in 32-bit Floating-Point Hardware Divider C R S Hanuman Department of ECE, CEG, Guindy Anna University Chennai, India [2018]

[2] Using One FPGA to Control Two High-SwitchingFrequency PMSM Drive Systems Through A Novel Time-division Multiplexing Method Fanning Jin, Wei Qian& Hua Bai DingguoLu and Bing Cheng Department of Electrical and Computer Engieering Software Control [2018]

[3] Design and implementation of 32-Bits MIPS processor to Perform QRD Based on FPGA 1 Safaa S. Omran , 2 Ahmed K. Abdul-abbas 1,2 Electrical Engineering Technical College Baghdad/Iraq [2018]
[4] A Review on Various Divider Circuit Designs in VLSI Ramadevi Vemula Dr.K.Manjunatha Chari Research Scholar,ECE Dept., Professor & HOD, Dept., of ECE [2018]
[5] Design and Analysis of FPGA Based 32 Bit ALU Using Reversible Gates S.M.Swamynathan[1], V.Banumathi[2], Dept. of ECE, Dept. of ECE [2017]
[6] A Pipelined Architecture for User-defined Floatingpoint Complex Division on FPGA Shaobing Huang, Li Yu, Fang-jian Han and Yiwen Luo School of Electronic Science and Engineering National University of Defense Technology [2017]
[7] Architecture for Quadruple Precision Floating Point Division with Multi-Precision Support Manish Kumar Jaiswal1 , and Hayden K.-H So2 Dept. of EEE, The University of Hong Kong, Hong Kong [2016]
[8] VLSI (FPGA) Design for Distinctive Division Architecture using the Vedic Sutra 'Dhwajam' Shantanu Oke[1],Suraj Lulla[2],Prathamesh Lad[3] Maharashtra Institute of Technology, Kothrud , Pune-38 [2014]
[9] Novel Binary divider architecture for high speed VLSI applications Ratiranjan Senapati, Bandan Kumar Bhoi, Manoranjan Pradhan VSS University of Technology, Burla, Odisha, IN [2013]
[10] 32 bit Multiplication and Division ALU Design Based on RISC Structure Kui YI Department of Computer Science and Information Engineer, WuHan Polytechnic University Wuhan, HuBei Province 430023, China [2009]
[11] Vedic Divider: Novel Architecture (ASIC) for High Speed VLSI Applications Prabir Saha* , Arindam Banerjee**, Partha Bhattacharyya* , Anup Dandapat# * Bengal Engineering and Science University, Shibpur, Howrah-711103, India[2009]
[12] Sujata A A, Lalitha Y S "Performance Analysis Of Analog Data Compression And Decompression Using ANN In 32nm Finfet Technology" [dec 2019]
[13] Sujata A A, Dr.Lalitha Y S " Survey on latest FinFET Technology and Its Challenges in Digital Circuits and IC's" Intternattiionall Journall of Ellecttroniics Engiineeriing (ISSN:: 0973-7383) Vollume 10 • Issue 2 pp.. 577--585 [June 2018-Dec 2018]
[14] Sujata A A, Dr.Lalitha Y S " Performance Analysis of FINFET Based 16:4 Encoder and 4:16 Decoder-A Design Perspective" IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 8, Issue 6.