International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 4, April - 2014

A VHDL Implementation of a Flexible 16-Bit Arithmetic and Logical Unit

Archana Singh Yadav[#1], Pratyush Tripathi[$2]
[#]Student, M.Tech
Kanpur Institute of Technology, Kanpur
[$]Asst. Professor
KIT, Kanpur

***Abstract*— In this paper a VHDL structural design for Arithmetic and Logical Unit is proposed. This design of 16 bit ALU consists of two input pins for 16-bit operands along with one input pin for 5 selection lines at the input and at the output a pin for the 32-bit output and a pin for one bit carry or borrow. It may perform all the logical (bitwise) and arithmetic operations such as addition, multiplication, subtraction, division, shift, increment, comparison etc. In this design for 16-bit addition and subtraction, the component used is 16-bit Parallel Adder. This 16-bit parallel design includes a 1-bit full adder. In this design multiplier is made of an algorithm called add and shift algorithm. This design may be made more compact using the statement "generic" in VHDL which also makes the design more flexible. These design units are yet independent but using structural modeling these may be simulated through a single design. In this unit the bitwise operations are also added for different selection lines for the logical operations. In this design the multiplexers may be used to select the appropriate inputs for the arithmetic and logic design units. These multiplexers may be used to perform some simple logical operations like programmable shifter or to invert the operands. In accordance with the selection line we may find out about the operations to be performed on the given input. This VHDL design proves to be very flexible and efficient when it is required to add some new complex operations in it.**

Keywords-Design units, adder, multiplier, substractor, VHDL Implementation

I. INTRODUCTION

This paper describes a suite of utilities for behavior of a 16 bit ALU design that has been implemented through mixed modeling in VHDL. In this design the arithmetic package facility given by VHDL is not used. Different elements are modeled separately. The main advantage to make this in different blocks is high flexibility of the design. This design is a very good utilization of structural design technique for VHDL. This ALU design implements the logical operations like AND, OR, NOT, XOR, Rotate Left, Rotate Left and arithmetic operations like addition, subtraction, division and multiplication.[1][2][3]

II. DESIGN DESCRIPTION

The ALU design may be broken into two parts arithmetic block and logical block. Arithmetic block consists of adder, substractor, multiplier etc and logical block does bitwise logical operations. The logical operations are done directly with the help of dataflow design facility in VHDL but the arithmetic block is implemented in structural design technique.[4] The designed ALU is capable of executing 17 instructions. Only two inputs are provided as the operands. The instruction set for different select lines are shown in following table.[5]

S	OPERATION	OPERAND
00000	NOT	inp1
00001	AND	inp1 & inp2
00010	OR	inp1 & inp2
00011	NAND	inp1 & inp2
00100	NOR	inp1 & inp2
00101	XOR	inp1 & inp2
00110	XNOR	inp1 & inp2
00111	LEFT SHIFT	inp1
01000	RIGHT SHIFT	inp1
01001	LEFT ROTATION	inp1
01010	RIGHT ROTATION	inp1
01011	ADDITION	inp1 & inp2
01100	SUBSTRACTION	inp1 & inp2
01101	INCREMENT	inp1
01110	DECREMENT	inp1
01111	MULTIPLICATION	inp1 & inp2
10000	COMPARISON	inp1 & inp2

Figure 1. Instruction set for ALU

The synthesized ALU have

- Two input pins for 16-bit operands inp1 and inp2.
- One input pin for 5 selection lines
- A pin for the 32-bit output
- A pin for one bit Carry or Borrow.

Sometimes multiplexers are used for the input to logical and arithmetic unit but in this design the use of multiplexers is ignored because the design structure is not partitioned. This makes the design more fast and flexible. It may directly add an instruction because 5 selection lines are provided which are able to handle 32 instructions but only 17 are used and the structure is not partitioned so it becomes very easy to add an instruction in it. It makes design more versatile and increases the flexibility in accordance with the application.[6][7]

III. DESIGN UNITS

A. *Adder*

A 16 bit parallel adder is used in the design. This parallel adder uses the facility of structural modeling, consisting of 1 bit full adder as the structural component. In this an initial carry may also provided or may be considered as '0'.The carry is transferred from one full adder to another full adder and the addition process goes on. An addition process till four bit is shown in Figure 1.

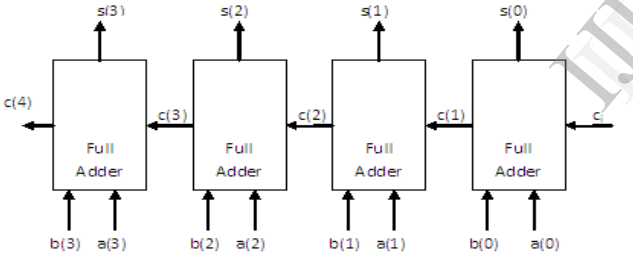

Figure 2 Addition Process for 4 bits

B. *Substractor*

In substractor the adder design unit is used as the structural component. The basic concept behind this design of subtraction is that binary subtraction of b from a is identical to the addition of a with 1's complement of b. This makes a very good utilization of adder unit in the design.[9][10]

C. *Multiplier*

In this design the Add and Shift Algorithm used in multiplier algorithm is used. In this algorithm if the multiplier's multiplying bit is '1' then add and shift both operations are held and if the multiplying bit is '0' then only shift operation is held. In the state diagram it consists of 10 states with 1st for loading the data and last for providing the status that the multiplication is complete.[7][9]

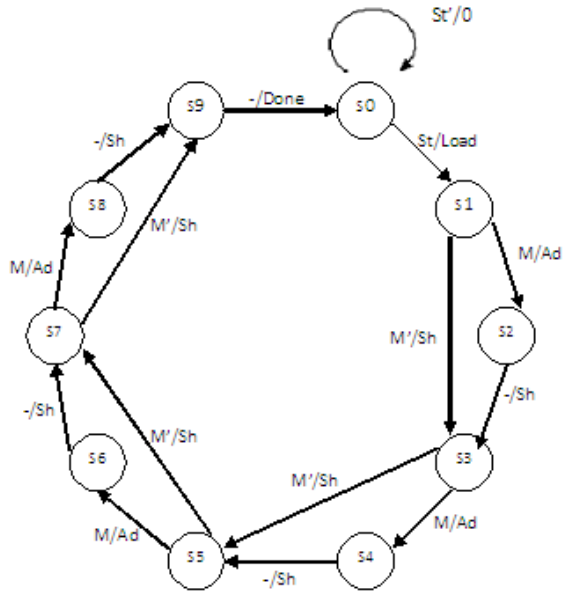

Figure 3.Staate Diagram of multiplier

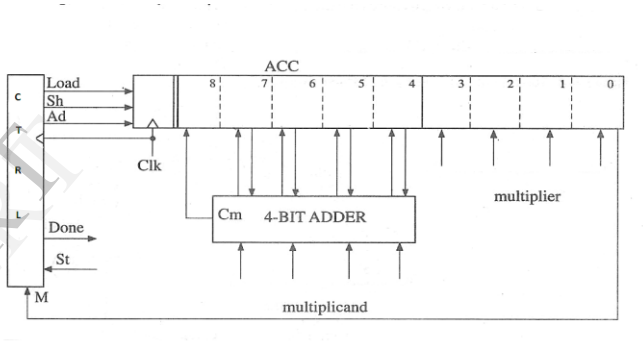

Figure 4. Process of multiplication algorithm

IV. VHDL IMPLEMENTATION

After the design flow completion the design is implemented through VHDL (VHSIC Hardware Descriptive Language).In VHDL this ALU design is implemented in mixed modeling technique. In it the arithmetic blocks are modeled with the structural modeling and then these along with logical block are implemented with the help of Dataflow modeling technique. [9][10][11] Some operations simulation results is shown in the figure:

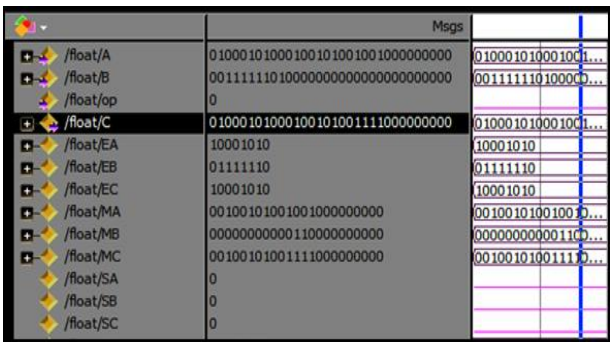

Figure 5: Addition operation simulation output

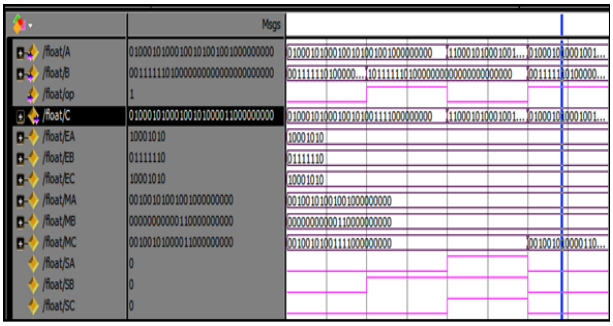

Figure 6. Subtraction Operation simulation output

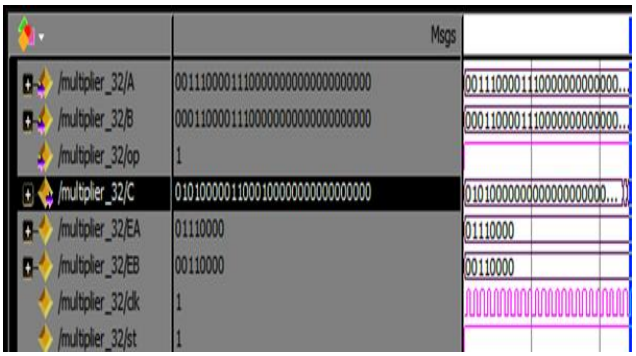

Figure 7. Multiplication Operation simulation output

V. CONCLUSION

The design was tested and simulated, no error was encountered. The project is ready to implement further. The presented 16 bit ALU Design is tested for 17 logical and arithmetic operations. It's a very flexible and fast design which may be integrated to more bit operations.

ACKNOWLEDGMENT

The authors thank the assistance provided by Mr. Sapan Kumar Gupta of DGI Greater Noida and Mr. Nikhil Srivastava of Prasad Institute of Technology, Jaunpur, Pushkar Mishra of TCS Ltd. and Aditya Pathak of YLM Tube Solutions &Service India Pvt. Ltd. Additional thanks go to Ragini Rai, Sangam Gupta, Vishwajeet Sahu and Pushkar Mishra for their great effort and inspiring me for this paper.

REFERENCES

[1] Gupta, Sapan Kr, Rahul Pandey, and Manuj Kumar. "Future Generation Processors." *International Journal of Engineering* 1, no. 6 (2012)

[2] *VHDL Techniques, Experiments, and Caveats by Joseph Pick* .

[3] *Zainalabedin Navabi "Analysis and Modelling of Digital Systems"*

[4] *The VHDL Cookbook by Peter J. Ashenden*

[5] Zimmermann, Reto. "VHDL library of arithmetic units." In *Proc. First Int. Forum on Design Languages (FDL'98), Lausanne, Switzerland*, pp. 267-272. 1998.

[6] Chang, Kou-Chuan. *Digital design and modeling with VHDL and synthesis*. IEEE Computer Society Press, 1996.

[7] Reaz, Mamu Bin Ibne, Md Shabieul Islam, and Md S. Sulaiman. "Pipeline floating point ALU design using VHDL." In *Semiconductor Electronics, 2002. Proceedings. ICSE 2002. IEEE International Conference on*, pp. 204-208. IEEE, 2002.

[8] Soni, Ashish, Nikhil Saxena, Narayan Sharma, and Sapan Kumar Gupta. "TDD for Embedded Systems: Feasibility and Issues." IJETT, Vol 4 Issue 6 (2012)

[9] *HDL Chip Design - A Practical Guide for Designing, Synthesizing and simulating ASICs and FPGAs using VHDL or Verilog by Douglas J. Smith*

[10] *VHDL: A Logic Synthesis Approach By David Naylor and Simon Jones*

[11] Stefan Sjoholm and Lennart Lindh "VHDL for designers "