IOSR Journal of Electronics and Communication Engineering (IOSRJECE)
ISSN : 2278-2834 Volume 1, Issue 1 (May-June 2012), PP 07-11
www.iosrjournals.org

VHDL Implementation of 8-Bit ALU

Suchita Kamble[1], Prof .N. N. Mhala[2]
[1]P.G.Student
B.D.C.E.Sewagram, Wardha (M.S.)
[2]Associate Professor
B.D.C.E.Sewagram, Wardha (M.S.)

Abstract- *In this paper VHDL implementation of 8-bit arithmetic logic unit (ALU) is presented. The design was implemented using VHDL Xilinx Synthesis tool ISE 13.1 and targeted for Spartan device. ALU was designed to perform arithmetic operations such as addition and subtraction using 8-bit fast adder, logical operations such as AND, OR, XOR and NOT operations, 1's and 2's complement operations and compare. ALU consist of two input registers to hold the data during operation, one output register to hold the result of operation, 8-bit fast adder with 2's complement circuit to perform subtraction and logic gates to perform logical operation. The maximum propagation delay is 13.588ns and power dissipation is 38mW. The ALU was designed for controller used in network interface card.*

***Keywords*:** ALU, Fast adder, Network interface card, VHDL implementation.

I. INTRODUCTION

As the performance of network servers increases, network interface cards (NIC) will have a significant impact on a system performance. Most modern network interface cards implement simple tasks to allow the host processor to transfer data between the main memory and the network, typically Ethernet. These tasks are fixed and well defined, so most NICs use an Application Specific Integrated Circuit (ASIC) controller to store and forward data between the system memory and the Ethernet. However, current research indicates that existing interfaces are optimized for sending and receiving large packets. Experimental results on modern NICs indicate that when frame size is smaller than 500-600 bytes in length, the throughput starts decreasing from the wire-speed throughput. As an example, the Intel PRO/1000 MT NIC can achieve up to about 160 Mbps for minimum sized 18-byte UDP packet (leading to minimum sized 64-byte Ethernet packet). This throughput is far from saturating a Gigabit Ethernet bidirectional link, which is 1420Mbps. Recent studies have shown that the performance bottleneck of small packets traffic is because that there is not enough memory bandwidth in current NICs. In a back-to-back stream of packets, as packet size decreases the frame rate increases. This implies that the controller in the NIC must be able to buffer larger number of incoming smaller packets. If the controller does not provide adequate resources, the result will be lost packets and reduced performance. The other reason for this problem is that current devices do not provide enough processing power to implement basic packet processing tasks efficiently as the frame rate increases for small packet traffic. Previous research has shown that both increased functionality in the network interface and increased bandwidth on small packets can significantly improve the performance of today's network servers. New network services like network interface data caching improve network server performance by offloading protocol processing and moving frequently requested content to the network interface. Such new services may be significantly more complex than existing services and it is costly to implement and maintain them in nonprogrammable ASIC-based NICs with a fixed architecture. Software-based programmable network interfaces excel in their ability to implement various services. These services can be added or removed in the network interface simply by upgrading the code in the system. However, programmable network interfaces suffer from instruction processing overhead. Programmable NICs must spend time executing instructions to run their software whereas ASIC based network interfaces implement their functions directly in hardware. To address these issues, an intelligent, configurable network interface is an effective solution. A reconfigurable NIC allows rapid prototyping of new system architectures for network interfaces. The architectures can be verified in real environment, and potential implementation bottlenecks can be identified. Thus, what is needed is a platform, which combines the performance and efficiency of special- purpose hardware with the versatility of a programmable device. Architecturally, the platform must be processor-based and must be largely implemented using a configurable hardware. An FPGA with an embedded processor is a natural fit with this requirement. Also, the reconfigurable NIC must have different memory interfaces providing including high capacity memory and high speed memory for adding new networking services [1] [2] [3].

The controller consist of central processing unit whose instruction set is customize to processes the network data [3] [4]. The features of controller used in the network interface card are as follows.

- 8 bit Processor (8 bit Data bus)
- 8 bit ALU for performing arithmetic and logical operations on signed and unsigned numbers such as Addition, Subtraction, AND, OR, NOT, 1's & 2's Complement and Universal shift register.
- 32 Registers 8 bit each for storing partial results during operation.
- Address and data register to buffered store current address and data.
- Program counter to hold the address of the current instruction.
- Instruction decoder
- Control unit
- Data memory
- Program Memory
- Interrupt Controller

In this paper VHDL implementation of 8-bit ALU is presented which performs operations for controller in network interface card. Paper is organized as follows section II discusses the design of ALU; results are presented section III and concluded.

II. ARITHMETIC LOGIC UNIT

ALU was designed to perform arithmetic and logical operations for controller. Arithmetic operations performed are 8-bit addition and subtraction. Logical operations performed are AND, OR, XOR and NOT. ALU also calculates 1's and 2's complement for the 8-bit input and compares the two inputs using 8-bit comparator. ALU also consist of two input 8-bit registers to hold that data during operation and output register to hold result of operation. Fig. 1 shows the entity for ALU.

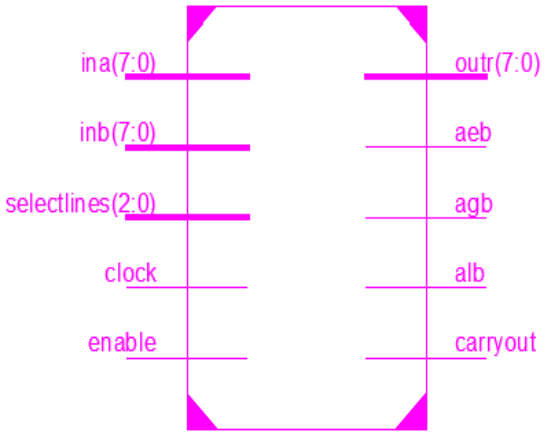

Fig. 1 Entity ALU

A. Design of 8-bit Adder and Subtractor

8-bit adder and subtractor was implemented using fast adder based on the principle of carry look ahead. The mode control signal was used to decide on the operation of addition and subtraction. The mode control signal performs 2's complement operation during subtraction only.

B. Design of 8-bit Logical block

8-bit logical operations were performed on the data bitwise. This block simply consists of parallel gates connected to perform desired operation.

C. Design of comparator

8-bitcomparator compares the two inputs and generates the high signal for A greater than B, A equal to B and A less than B.

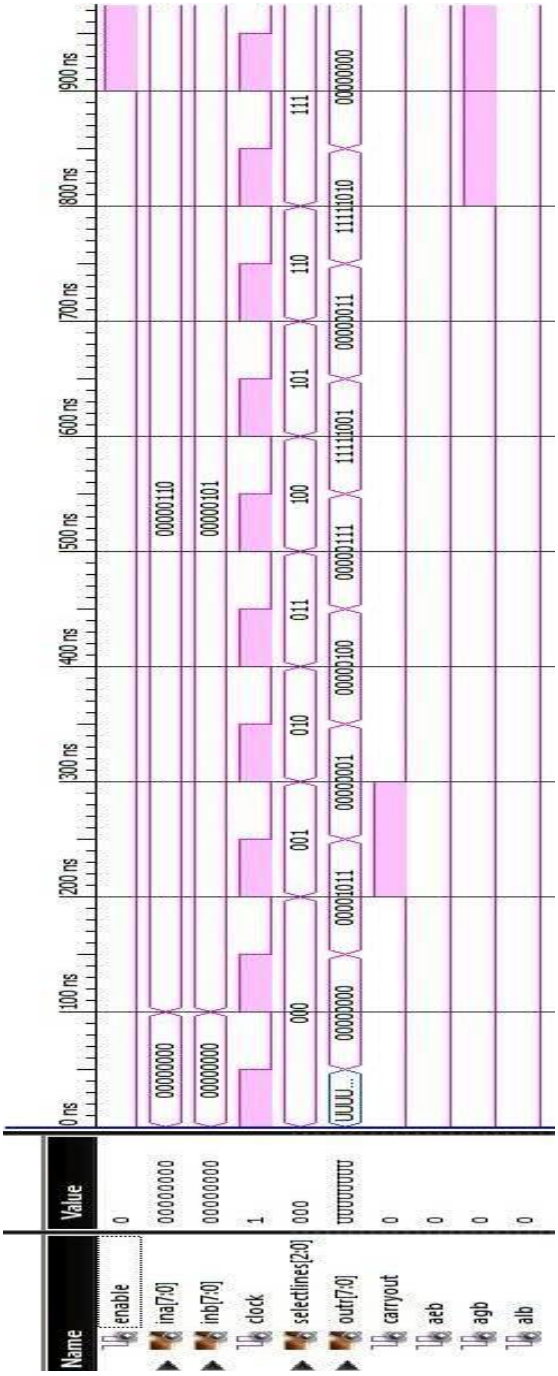

Fig. 2 Simulation Results

Finally the output of the desired operation is selected by select lines. Table I shows the status of select lines and operations performed by ALU.

TABLE I
SELECT LINES

Select Lines	Operation
000	Addition
001	Subtraction
010	AND

011	OR
100	NOT
101	XOR
110	2's Complement
111	Compare

III. RESULT

VHDL implementation of the ALU was done using Xilinx Synthesis tool 13.1 and targeted for Spartan device. Detail synthesis report is presented in table II. Fig. 2 shows the simulation results of ALU.

TABLE II
SYNTHESIS REPORT

Nos	Blocks	Utilized
1	Registers	4
	3-bit registers	1
	8-bit registers	3
2	Latches	3
3	Comparator	3
4	8-bit 8:1 Multiplexers	1
5	Flip Flops	27
6	IOs	33

IV. CONCLUSIONS

VHDL implementation of 8-bit arithmetic logic unit (ALU) is presented. The design was implemented using VHDL Xilinx Synthesis tool ISE 13.1 and targeted for Spartan device. ALU was designed to perform arithmetic operations such as addition and subtraction using 8-bit fast adder, logical operations such as AND, OR, XOR and NOT operations, 1's and 2's complement operations and compare. The maximum propagation delay is 13.588ns and power dissipation is 38mW. The ALU was designed for controller used in network interface card.

REFERENCES

[1] Toshio Fujisawa, et al, "A Single-Chip 802.11a MAC/PHY With a 32-b RISC Processor", in *IEEE Journal Of Solid-State Circuits*, Vol. 38, No. 11, November 2003.

[2] J. R. Allen, et al, "IBM PowerNP network processor: Hardware, software, and applications," in *IBM Journal of Research & Development*, Vol. 47, No. 2/3 March/May 2003.

[3] Xiaoning Nie, et al, "A New Network Processor Architecture for High-speed Communications," in *IEEE Workshop on Signal Processing Systems*, 1999.

[4] H. Peter Hofstee, "Power Efficient Processor Architecture and The Cell Processor," in *Proceedings of the 11th International Symposium on High-Performance Computer Architecture*, 2005.

[5] D. L. Perry, " VHDL", Tata Mcgraw Hill Edition, 4th Edition, 2002.

[6] C. Maxfiled, "The Design Warriors Guide to FPGAs", Elsevier, 2004.

[7] J. Bhaskar, " VHDL Primer", Pearson Education, 3rd Edition, 2000.

[8] J. Bhaskar, " VHDL Synthesis Primer", Pearson Education, 1st Edition, 2002.