DESIGN AND IMPLEMENTATION OF APPLICATION SPECIFIC 32-BITALU USING XILINX FPGA

T.MALLIKARJUNA[1] *,K.SREENIVASA RAO[2]

[1]PG Scholar, Annamacharya Institute of Technology & Sciences, Rajampet, A.P, India.

[2]Asso.prof, Annamacharya Institute of Technology & Sciences, Rajampet, A.P, India.

Abstract-- **ALU is a combinational device which performs arithmetic and logical operations. It plays an important role in most of the existing commercial processors. The power consumption and area of a processor settle on the structure of an ALU. There are two structures of an ALU one is tree and another one Chain. This paper aims to design and implementation of 32-bit ALU with chain structure. With this structure we are reducing the overall delay and area of an ALU by repositioning functional components based on the frequency of an operation within the specific application. The FPGA implementation and functionality test of the 32-bit ALU is done by using the Xilinx ISE Design suite 14.7 Tool.**

Keywords**: 32-bit ALU, Verilog, ALU Functional Blocks, Behavioral, chain structure of 32-Bit ALU.**

1. ALU

1.1 Introduction

The design of low power and high speed microprocessors requires that its components should consume less power. Arithmetic and Logic Unit (ALU) is one of the most power consuming components in a microprocessor. To reduce the power consumption of the entire ALU each of its components should consume less power. The ALU performs the arithmetic operations such as addition, subtraction multiplication etc. and logical operations. Such as Negation, AND, OR, XOR as well

As rotate and Shift operations. The power consumption and area of a processor resolve on the structure of an ALU. There are two structures of an ALU one is tree and another one Chain. The main aim of this paper is to design and implementation of a 32-bit ALU which performs addition, subtraction, xor, or, and negation operations. Allthe above module are connected in structure.

1.2 Tree Structure

The Tree Structure of an ALU all functional components are connected in parallel to the multiplexor. The multiplexer selects one of the functional block output. The Tree Structure of the ALU contains five functional components for addition, and, xor, or, not. The tree structure design as below figure 1(a) shown,

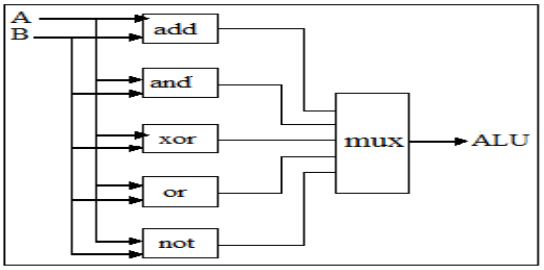

Figure 1(a): Tree Structure Design

The above tree structure the multiplexer selects one result as the ALU output among results from all functional components. The tree structure often demands more area. With a modern

processor design, where the processor is pipelined into a number of stages, the speed of the processor is determined by the longest delay that is that is associated with the critical path.

1.3 Chain Structure

The chain structure functional components are concatenated through a series of multiplexers, each multiplexer takes a subset of functional components and pass the result to the output. The Chain Structure of an ALU shown below figure 1(b).

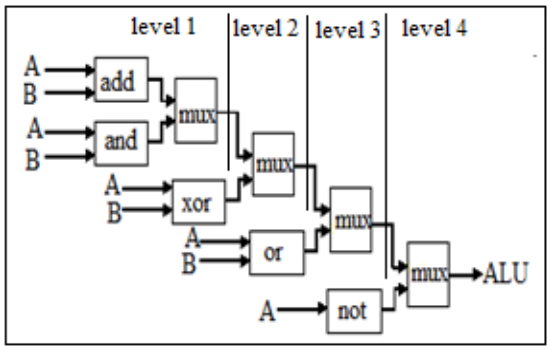

Figure 1(b): Chain Structure

The chain structure takes more levels of multiplexor transmission to the output than Tree Structure. Even though the chain structure for the ALU save area. In this paper, we investigate the effect on path delay of functional component placement in the chain structure we proposed a functional component placement approach to reduce path delay for a given application.

2. ALU Functional Blocks

The proposed 32-bit ALU consists of adder, and, or, xor & not. All functional components are below shown.

2.132-BIT ADDER

In digital adders, the speed of addition is limited by the time required to propagate a carry through the intermediate stages of an adder, the sum for each bit position in anelementary adder is generated sequentially only after the previous bit position has been summed and carry propagated into the next position. In this paper a 32-bit carry select adder(CSLA) was designed byusing Ripple Carry Adder (RCA), 5-bit and 7-bit BECmodules along with multiplexer. The CSLA is used in many computational systems to generating multiple carries and then select a carry to generate the sum. The CSLA is not area efficient because it uses multiple pairs of RCA, to generate partial sum and carry .In proposed CSLA architecture uses Binary to Excess-1 Converter (BEC) instead of RCA with C_in=1 in the regular CSLA to achieve lower area and power consumption.The main advantage of this Binary to Excess-1 Converter (BEC) logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure. The structure and the function table of a 5-bit BEC and 7-bit BEC are shown in Figure 2(a) and 2(b),

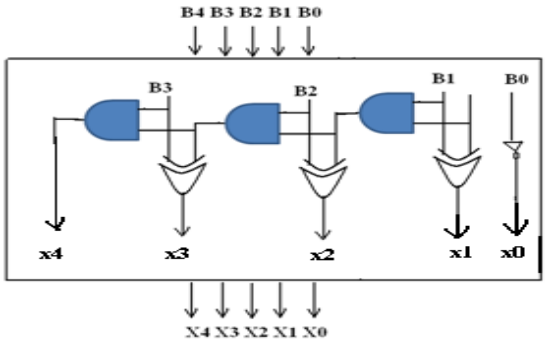

Figure 2(a): 5-Bit BEC.

The Boolean expressions of the 5-bit BEC is listed as

X0 = ~B0
X1 = B0^B1
X2 = B2^ (B0 & B1)
X3 = B3^ (B0 & B1 & B2)

X4=B0&B1&B2&B3

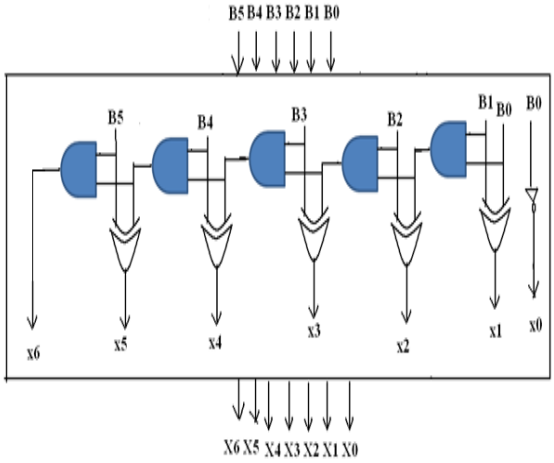

Figure 2(b): 7-Bit BEC

The Boolean expressions of the 7-bit BEC is listed as

X0 = ~B0
X1 = B0^B1
X2 = B2^ (B0 & B1)
X3 = B3^ (B0 & B1 & B2)
X4= B4^ (B0&B1&B2&B3)
X5 = B5^ (B0&B1&B2&B3&B4)
X6 = B0&B1&B2&B3&B4&B5

The 32-bit adder RTL Diagram as shown in figure 2(c).

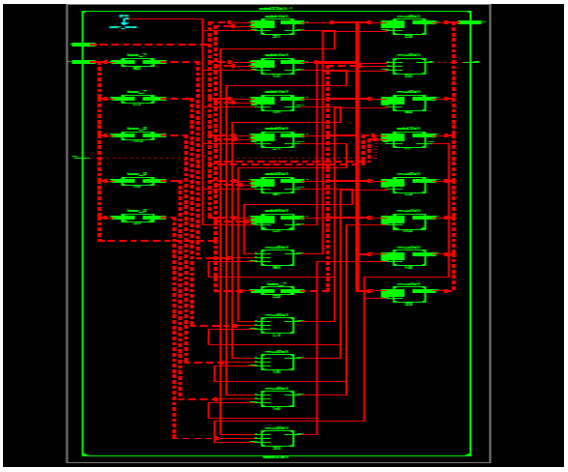

Figure 2 (c): 32-Bit adder

3. Verilog ALU Compilation

The customization technique can be integrated into a processor design environment. A general design platform is given in Figure 3. The design flow starts from a given application written in a high level programming language. TheTargetmachine architecture is selected and the program is compiled for the target machine. Based on the instruction set of the machine architecture model for the processor model is developed by either commercial or in-house development tool.

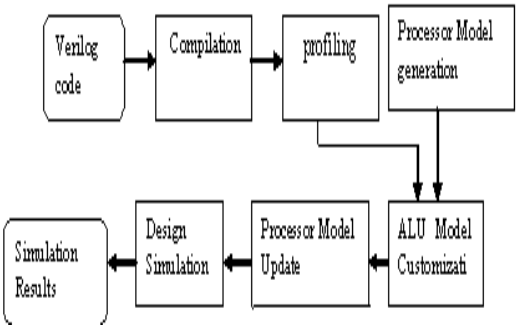

Figure 3: Design Flow with Verilog

The processor model is then updated with the customize ALU. Since the update does not functionally affect the processor and instruction set architecture, no modification is required to the instruction code. Next, the new processor model for the application code is simulated. The design is synthesized using a synthesis tool. Based on the synthesis, the design is evaluated.

4. Chain Structure of 32-bit ALU

In the 32-bit Chain Structure of an ALU hasadder, and, xor, or, andnot modules,which are concatenated by 2-input 32-bit multiplexer with single selection line shown in figure 4.1.

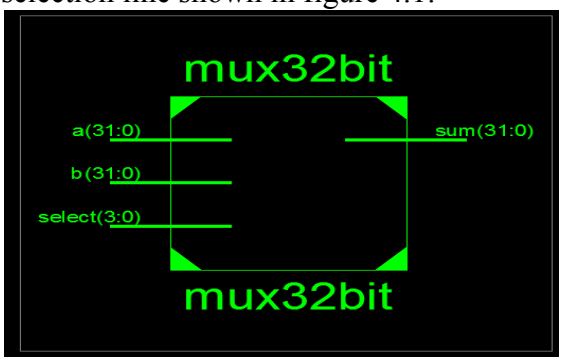

Figure 4.1: 2 input 32-bit multiplexer

The complete structure of32-bit Chain Structured ALU as shown in below figure 4.2.

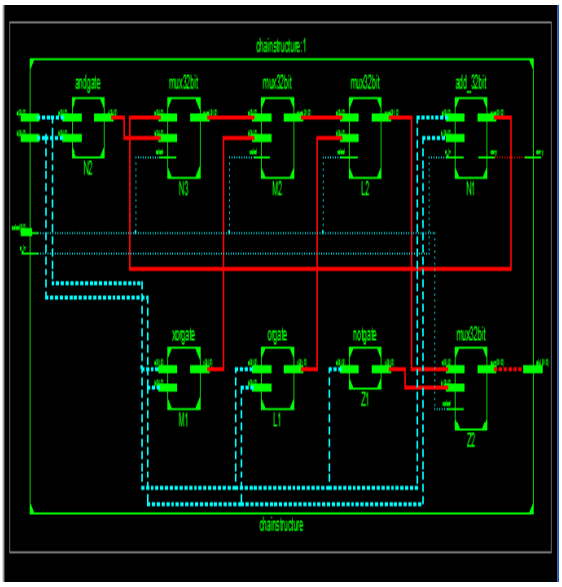

Figure 4.2: 32-bit Chain Structure

Design

For the chain structure, there are a variety of options to place functional components. So far, the functional component placement in the chain is arbitrarily chosen in the processor design. To our best knowledge, no work related to this design issue has been reported. In fact, placing a functional component differently in the chain structure may cause different power consumption. For example, swapping the add component and the not component may favor in some applications and can save a considerable amount of ALU power.

5. Simulation results

To verify our simulation results, we first developed a small stand-alone ALU for full design space exploration. The ALU contains five functional components each for *addition, xor, and, or, not.* The experiment is given below figure 5 shows.

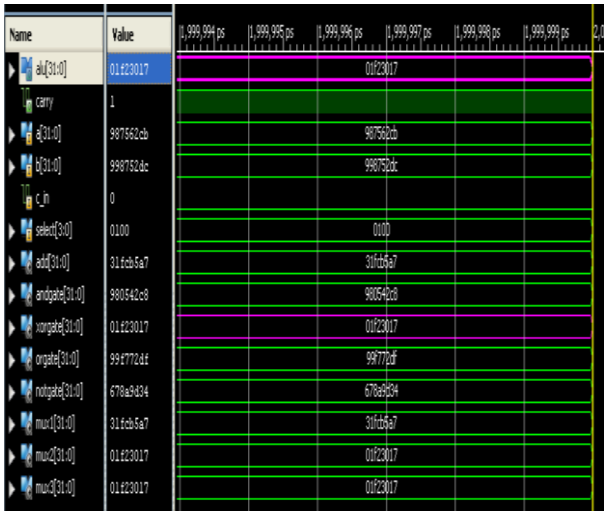

Figure 5: Simulation Results for 32-bit ALU Chain Structure

The above figure shows that all functional components are present in the simulation result. To enter an input values from A, B and c_in(carry in) and selection. The ALU of five functional components was used for exploring designs of all possible placements in order to verify the effectiveness of this approach. Based on the Chain Structure program, the adder is longest component, therefore when it is positioned next to the output in the chain, the overall delay is reduced. The power always reaches a minimum level when the related functional component is placed closet to the output.

We can see that the CPU clock time remains changed throughout all designs, which demonstrates that the critical path and differing functional components in ALU affect the processor clock speed. Finally, the processor with the customized ALU is evaluated using the Xilinx ISE Design Suite 14.7 Tool.

6.Conclusion

In this paper we discussed the effect of component placement on the chain structure design. We found that the order of functional components in the chain effects is shown in below table 1.

Table 1: **Different levels of 32-bit adder**

Adder	Logic Utilization	Used	Available	Utilization	Path Delay
Level 1	Number of Slices	138	4656	2%	21.335ns
	Number of 4 inputs LUTs	250	9312	2%	
	Number of bonded IOBs	102	232	43%	
Level 2	Number of Slices	122	4656	2%	20.289ns
	Number of 4 inputs LUTs	216	9312	2%	
	Number of bonded IOBs	102	232	43%	
Level 3	Number of Slices	113	4656	2%	20.335ns
	Number of 4 inputs LUTs	211	9312	2%	
	Number of bonded IOBs	102	232	43%	
Level 4	Number of Slices	99	4656	2%	19.961ns
	Number of 4 inputs LUTs	181	9312	1%	
	Number of bonded IOBs	102	232	43%	

To reduce latency and area, the frequently operating component should be positioned close to the output of the chain. If the adder is placed at the higher stage (level 1), the path delay is 21.335ns and it takes 250 LUTs. The adder is changed to below XOR placed (level 2). The path delay is 20.289ns and its takes 211 LUTs. Again the adder is changed to OR place (level 3),the path delay is 20.113ns and it takes 216 LUTs. The adder is changed to NOT (level 4), the path delay is 19.961ns and its takes 181 LUTs. So the adder is placed higher stage the path delay is 21.335ns (level 1) and the adder is placed closest to the ALU output, the path delay is 19.961ns (level 4). The overall path delay is 1.37ns will be saved. Therefore the adder is placed closest to the ALU output will take less amount. We developed a Verilog customization approach for the ALU design. The customization is extremely simple. In details neither additional control logic for modification to the inter face of the ALU model in the processor design.

7. References

[1] M. Borah, R. M. Owens, and M. j. Irwin. Transistor sizing for low power cmos circuits, IEEE Trans. On computer Aided Design of integrated circuits and systems, 15:665-671, 1996.

[2] Y.-T Ho and T.-T. Hwang. Low Power design using dual threshold voltage. In Proceedings of the Asia and South pacific, Design Automation Conference, page 205-208, 2005.

[3] C. Thimmannagari. CPU Design: Answers to Frequently Asked Questions, Springer, 2005.

[4] L. Shang, L. Peh, and N, Jha. Dynamic Voltage Scaling with links for power optimization of interconnection networks. In proceeding of International High Performance Computer Architecture, page 91-102, 2003.

8. BIODATA

T. Mallikarjuna, born in Kadapa, A.P., India in 1990. He received his B.Tech Degree in Electronics and Communication Engineering from J.N.T University Anantapuramu, India. Presently pursuing M.Tech (EMBEDDED SYSTEM DESIGN) from Annamacharya Institute of Technology and Sciences, Rajampet, A.P., India. His research interests include VLSI, Digital Signal Processing and micro processor.

Mr. K. Sreenivasa Raohas received his M. Tech degree in DSCE. Currently, he is working as Associate Professor in the Department of Electronics & Communication Engineering, Annamachrya Inst of Technology & Science, Rajampet, Kadapa, A.P, and India. He has published a number of research papers in various National and International Journals and Conferences. He is currently working towards Ph.D Degree in at RayalaseemaUniversity, Kurnool, A.P, and India. His areas of interests are VLSI, Micro processor, Embedded Systems and Signals and Systems.

