Techniques for ALU Advancements in Throughput, Space, and Efficiency in the New Millennium

Devin Hobby
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, FL 32816-2362

Abstract—Pipelining is a parallelization of instructions that would otherwise be executed in sequential fashion. With new sophisticated pipelining structures and smaller transistors reaching 10 nm, more instructions can be computed in less clock cycles and using less space. RISC processors, FIR Design, Feedback Switch Logic, and multi-dimensional pipelining are the advancements that make possible the speed and precision of these systems. Tested primarily on FPGA’s and in simulations, these methods mainly affect the ALU where mathematical computations and numerical manipulation take place. RISC or Reduced Instruction Set Computing for example, uses a crucial set of instructions rather than a vast library. This optimized instruction set combined with a 4-level pipelining structure can enhance system performance with little to no drawbacks, and can then be implemented in conjunction with other enhancements for amplified system speed, enhanced efficiency, minimized space, and maximized precision.

Keywords— RISC processors, FIR Design, Feedback Switch Logic, and 2-D pipelining, ALU Floating Point Analysis, Pipelining

I. INTRODUCTION

The ALU or Arithmetic Logic Unit is a crucial part of any CPU because that is where all arithmetic calculations are made. Numbers are the bridge between human language and computer language and the ALU is essentially the interpreter. The ALU performs operations like addition, subtraction, multiplication, division, AND, OR, NAND, NOR, XOR, and shifting logic. When the CPU needs to calculate any number the ALU is the unit that takes the data and returns the necessary output, working as a key player in the big picture. A typical loop, for example, has a variable that increments over each iteration and is then used to make comparisons, select elements of an array, or any number of functions. The incrementation of the variable is handled by the ALU preforming an addition of the original value plus one. Loops are a common practice in programming and each loop uses the ALU not only to increment, but to compare values, change bits, and any other mathematical operations that may be needed.

Since the ALU is such a crucial part of the processing power of a CPU, there are many minds at work to improve the performance and versatility of the ALU. The main concerns being, accuracy, speed, size, and power consumption. Generally accuracy is the simple part, but many steps have been made to increase speed such as pipelining. When an ALU first receives an input the data comes as the operands and the operation. The operation is the instruction to add, multiply, etc… and the operands are the numbers, represented in binary, to be used in the operation. The operands are temporarily stored in the register file, and the input code contains their respective addresses which are received through the data-path to the ALU. The operation or function to be executed comes across the control path which, like a multiplexer, are used in the select lines to select the set of gates to operands are to pass through.

Various changes are made to the ALU to increase its capabilities while decreasing its execution time and physical size. Increasing the data bus width for example would increase the amount of data that is coming into the ALU at one time, yet it also physically does increase the width of each stage. The International Technology Roadmap for Semiconductors technology node is the physical size of transistors and semiconductor components. This is decreasing in size as the process of making semiconductors becomes more refined. This decrease in size means there is more room on a chip for other things, like a larger data bus width. Of course, more components in a circuit results in more energy consumption and power dissipation, especially as heat. Then cooling and thermodynamics comes into play and the space and power cost associated with that. Many of these advancements are focused on a faster execution time so data can be processed faster and be ready to receive the next instruction that much faster. A CPU can only be as fast as its slowest member so every aspect is undergoing change to reduce execution time across the board.

The next section describe the efforts being made to make advance the technology, design, and layout of the ALU in an attempt to further advance the capabilities of what we do today. Various techniques of pipelining are used to reduce execution time, smaller components are used that use less power and demand less space. Different adder and multiplier designs that increase the productivity of the ALU with little to no side effects in other areas. The ALU is the arithmetic brain of the CPU and these advancement are making the ALU smarter than ever in this review from 1999 until the present day. The next section will discuss 5 different base computer systems, focusing on the ALU, on which can be drawn comparison data from another 5 systems.

II. LITERATURE REVIEW

The first system to consider is a high throughput power-aware FIR filter design based on fine-grain pipeline multipliers.
and adders [4]. The advantages of this system from 2003 is its 2-D pipelining which gates redundant stage of the system and therefore improves the throughput. In addition to the throughput the 2-D pipelining allows unused registers to be disabled which reduces the switches made and therefore saving power. This design is what helps increase the throughput limit on FIR filters in the design with a 1.25 GHz clock rate.

Then next system to consider is an also a FIR filter using 2 Dimensional pipelining but with a focus on power awareness, instead of throughput [5]. This is accomplished using signed and unsigned power aware pipelined multipliers. The 2 dimensional design does take more space but the increase in area is almost negligible. The sign extension is avoided by gating the clock to registers in the direction of dataflow and within each pipeline stage (hence 2 dimensional) thus reducing power and latency. A recorded savings of 65-66% in power and 44-47% in latency was captured by this design.

The energy-efficient multiplier-less discrete convolver which uses probabilistic domain transformations [3] and runs on a Virtex 6 FPGA device using only 166.63 nJ is a design from 2014. Probabilistic domain transforms are a light weight alternative to techniques like Fourier transforms which can be used for encoding and before an operation and decoding following the operation. Fourier transforms can be computationally intensive, especially with a large amount of data, like a video or series of images. This allows complex multipliers to be replaced by simple adders, which provides more space and less energy consumption.

Another system to consider is a 15 nm CMOS Floating-Point Unit energy and area focused design [1]. A 15 nm CMOS compared to a 45 nm node using IEEE 754 Single Precision Floating-Point Unit provides a footprint three times small with energy consumption that is 4 times less. The flaw in this design is in the FinFET semiconductor devices used which self-heat which is amplified in high densities. Yet the size of the node is what brings this design of 2015 to consideration

The ultra-area-efficient fault-tolerant QCA full adder [2] is considered as well. This system uses less area and less power than its typical CMOS counterpart, but the draw back are that it is more prone to errors. CMOS transistors are made in a very refined process that is highly accurate due to advances in photolithography. This design is proposed to be fault tolerant along with the added benefits that come with Quantum-Dot Cellular Automata’s boasting a small 18 nm² cell area and energy efficient design.

In 1999 the multi-level low power 16 bits ALU Design [8] was created using technology of the time. With an ITRS node width of 0.6 μm and running at 54 mW this design used an ELM adder for addition. Running at 200 MHz this ALU design saved energy by leaving the arithmetic side of the ALU inactive while logical operations were performed. This in turn reduced switching which reduces the energy consumed by the design. This is a simplified version of the 2 dimensional pipelining seen in more modern designs, because power is saved due to less switching.

The design of a low power, high speed ALU using Feedback Switching Logic [9] was tested in 2009. FSL was a new circuit design that reduced switching time, capacitance, and also was not always dependent on the clock. 90 nm ITRS node with was used and a Kogge-Stone adder for testing. It was found that using FSL the ALU returned results with a 14% decrease in delay, although there was an 8% increase in power consumption in comparison with a standard static CMOS ALU of the time.

A 32 bit multiplication and division ALU design based on RISC Structure [10] of 2009 was tested on an FPGA GW48 EDA. RISC is a refined instruction set, the lack of available instructions results in higher performance. Using Floating point IEE 754 single precision units and a QuartusII with VHDL multiplier, the design implemented a 4 level pipelining structure for ‘0’ operation number check, exponent addition/subtraction, fraction multiplication/division, and result normalization. The pipelining allowed a decrease in size and increase in performance of the system.

In 2013 an energy efficient design and implementation of ALU on 40nm FPGA [7] was carried out testing many different components. Using a Carry save multiplier and an ELM adder a low power usage was achieved, boasting a 68.34% dynamic power reduction using LVCMOS12 and LVCMOS15 in place of LVCMOS25. With this and many other switches to the system’s components the power consumption dropped drastically coming from the ITRS node of 90 nm to 40 nm while considering dynamic power due to the direct proportionality of power to frequency.

A system designed in 2015 with a 16 bit RISC Processor used low power pipelining [6]. The RISC Processor ALU carry select adder and efficient multiplier where used and due to the simple instruction set of the RISC processor in conjunction with the 2 stage pipelining with a positive and negative edge. The system was run on an XILINX KINTEX XC7K1607-3fg676 using 28 nm technology. In the end the power usage was 0.22 W and a latency of 1.5 cycles. This systems pipelining provided that each instruction could be completed with only 2 clock cycles.

### III. Data Analysis

![Data Bus Width vs. Year](image)

Fig. 1. Data Bus Width vs. Year
IV. CONCLUSION

Using techniques like multi-dimensional pipelining, RISC processors with reduced instruction sets, FIR Design, Feedback Switch Logic, and smaller transistors. By these means engineers have amplified system speed, enhanced efficiency, minimized space, and maximized precision. ITRS Node width has followed the predictions of Moore’s Law, leading to smaller component also contributing to the system. Generally the carry select adder was used, but effective pipelining both saved energy and space. These engineering feats along with the other advancements mentioned here the world of powerful computing will continue to grow rapidly.

REFERENCES

<table>
<thead>
<tr>
<th>ALU or Floating Point Architecture Name</th>
<th>Datapath width (bits)</th>
<th>Time for Operation or Design Type</th>
<th>ITRS Technology Node (nm) or Area or Model of Chip used</th>
<th>Energy/Power Consumption (W or J) else indicate “low” or “high”</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Energy and Area Analysis of a Floating-Point Unit [1]</td>
<td>32 bits (Operands)</td>
<td>N/A</td>
<td>N/A</td>
<td>IEEE-754 Single Precision</td>
<td>2015</td>
</tr>
<tr>
<td>Ultra-area-efficient fault-tolerant QCA full adder [2]</td>
<td>1 bit (Operands)</td>
<td>Ultra-area-efficient fault-tolerant QCA full adder</td>
<td>N/A</td>
<td>18nm² (Cell Area)</td>
<td>2015</td>
</tr>
<tr>
<td>HIGH THROUGHPUT POWER-AWARE FIR FILTER DESIGN BASED ON FINE-GRAIN PIPELINING MULTIPLIERS AND ADDERS [4]</td>
<td>16 bit</td>
<td>FIR filter adder with pipelining</td>
<td>N/A</td>
<td>N/A</td>
<td>Low</td>
</tr>
<tr>
<td>Improving Power-awareness of Pipelined Array Multipliers using 2-Dimensional Pipeline Gating and its Application to FIR Design [5]</td>
<td>16 bit</td>
<td>Fine-grain 2-Dimensional pipelined</td>
<td>N/A</td>
<td>Low Increase</td>
<td>Very Low (~50% Less)</td>
</tr>
<tr>
<td>Energy efficient design and implementation of ALU on 40nm FPGA [7]</td>
<td>8 bit</td>
<td>ELM Adder</td>
<td>Carry Save Multiplier</td>
<td>N/A</td>
<td>40 nm</td>
</tr>
<tr>
<td>32 Bit Multiplication and Division ALU Design Based on RISC Structure [10]</td>
<td>32 bit</td>
<td>N/A</td>
<td>ALU Multiplier using QuartusII with VHDL</td>
<td>IEEE-754</td>
<td>FPGA GW48 EDA</td>
</tr>
<tr>
<td>Multi-Level Approaches To Low Power 16 Bits ALU Design [8]</td>
<td>16 bit</td>
<td>ELM Adder</td>
<td>N/A</td>
<td>N/A Fixed Point</td>
<td>0.6 μm single-poly triple Metal CMOS</td>
</tr>
</tbody>
</table>