ALU and FPU Design Patterns Over The Last Decade

Julian Rojas
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, FL 32816-2362

Abstract—Over the last decade we have been focusing on performance and power consumptions, whether it’s architecture complexity or time and space tradeoffs. Throughout the years we have managed to implement ALU designs that enable us to achieve faster running times, or cheaper production costs. Building a Pipelined Array Multipliers using 2-Dimensional Pipeline Gating has allowed us Simulation results averaging a power saving of 65% and latency reduction of 45%, or developing a system that allows for different frequencies to have an important effect on the power consumption such as implementing a clock gating techniques in ALU. Clock power reduces to 17.85%, 23.39%, 26.49% and 27.19% of total dynamic power, when device operating frequency is 1GHz, 10GHz, 100GHz and 1THz.

Keywords—Execution time, Floating Point Unit, Energy Consumption, ITRS technology Node, Datapath Width, power consumption, high scalability, CMOS technology.

I. INTRODUCTION

Different types of ALU designs yield different types of computational devices; some can be multipliers, others can be as fundamental as adders. Depending on the level of precision, the execution time, or the complexity level of the system we can use varying types of FPUs (Floating Point Units) or ALUs (Arithmetic Logic Units).

A Floating Point Unit is a very important part of the computer architecture; it is designed in order to perform operations with floating point numbers given a certain level of precision using ALUs. Such operations can range from: multiplication to bit shifting. Floating Point hardware supports these operations by breaking them into fields and allowing the decimal point to float just like it would in the scientific notation. The inputs to the FPU/ALUs are the operands, which are the data that has been passed to the unit. The ALU is the actual hardware that performs the computations: addition, subtraction, OR, AND. The Datapath is the part of the processor that holds the data that will be processed; it also includes the instruction register and the program counter. While the control path is the one that commands these other components with the desired instructions.

Data bus width determines how much data can be transmitted simultaneously. It can be 16 bits or 32 bits for example. International Technology Roadmap for Semiconductors (ITRS) technology node is defined as the minimum metal half pitch (minimum width and minimum spacing). The execution time, is the time required for the system to complete a task, the power dissipation is equal to the current passing multiplied by the applied voltage, and energy consumption of processors is due to the rate at which computations are perform in relation to the amount of power that has been utilized, the higher the rate, the more energy efficient a unit will be.

In the following sections I will be reviewing ten ALU designing ranging from the year 2000 until today. There will be a data analysis with plots comparing and contrasting Data bus width vs. Year, Execution Time vs. year, and also Energy consumption vs. Year.

II. LITERATURE REVIEW

2015-2010

In 2015, a study showed that a standard IEEE- 754 Floating Point Unit with two different technologies was tested, one was a 15nm, the other one was a 45nm, and the results indicated that using the one with the smaller cell area saved 4 times more in energy consumption. In other words, the cell area reduction of about 30% can lead to around a 4-fold increase in energy efficiency with the help of the 15nm technology [1].

In 2015, another study showed that using Quantum- dot cellular automata (QCA) was a promising technology that could help aid the “Scalability limits, leakage power consumption, and short channel effects” that follows the CMOS technology [2], the result they came up with was that a more robust full adder could be developed where it would outperform previous full adder designs in terms of latency, complexity, and area. This is verified through implementation of a 4-bit carry save adder. [2]

In 2015, an analysis of a 16-bit RISC processor showed that using low pipelining reduced the power dissipation to 220mW, and the latency to about 1.5 cycles compared to the 2.0 needed before. The design included a Universal Shifter and a Barrel shifter, and using the Verilog language, “ The Carry select adder structures are employed verified through exhaustive simulation and lower power dissipation and it can increases the speed.”[6]

In 2014, with the challenges of energy efficiency and algorithm complexity generally go on opposite sides of the scale, the development of smaller technologies (10nm CMOS) create room for improvement in the development of low power
consumption and high scalability while maintaining high performance. Or as stated “While the conventional multiplier based architecture can only achieve $O(N^2)$, the proposed architecture, through the proposed probabilistic domain transformation, can achieve approximately $O(N)$ in algorithmic complexity, therefore highly scalable and energy efficient.”[3], the study concluded that Probability Density Function (PDF) enabled “more primitive operations to be performed to achieve higher energy-efficiency and low hardware cost.”[3]

In another study in 2013, a group of scholars proposed that a technique to gate the clock to registers in both vertical direction (data flow direction in pipeline) and horizontal direction (within each pipeline stage). [4] They came to the result that an average of 66% power saving and a latency reduction of up to 47% were accomplished. [4] In conclusion, to a very small increase on the area, they were able to either maintain or reduce latency while decreasing the power consumption.

With the study in 2013 based on Fault Demotion Using Reconfigurable Slack the team was able to show that through experimental testing of the FaDReS algorithm there were positive results for fault-handling events. “In particular, in absence of a uniplex health metric (such as PSNR), the designer can trade off periodic temporal CED or spatial CED redundant computations based on throughput, cost, and reliability constraints.” [5]. Using the DSP48 multiplier helped them reduce the power dissipation to about 950mW.

On the paper published on April 2013, a “latch free clock gating techniques is applied in ALU to reduce clock power and dynamic power consumption of ALU. Clock power is 50%, 41.46%, 51.30%, 55.15% and 55.78% of total dynamic power when device operating frequency is 100MHz, 1GHz, 10GHz, 100GHz and 1 THz.”[7], meaning that there was a major reduction in power consumption compared to systems where no clock gating was used. There was an obvious reduction of power dissipation at varying frequencies of at least 25% on the 90nm Spartan-3 FPGA.

2009-2005

In 2008, a study on the effect of components placement in a chain structure. The results showed that depending on the order, there was a change in power dissipation, and to reduce said attribute, the most frequently operated components had to be placed closer to the output of the given chain. “The experiments on a set of benchmarks have shown that on average, 46.9% ALU power can be achieved with our design approach and the power saving is at no cost of processor performance.”[9]. The key finding was that this achievement did not increase neither the cost nor the hardware complexity, and the processor performance was left intact.

In 2005, a 4GHz 300mW 64b Integer Execution ALU with Dual Supply Voltages in 90nm CMOS was studied, the conclusions were evidence to the fact that achieving single latency while maintaining throughput are key features to achieving high-performance microprocessors [10], with this design, the results were outstanding: “The sparse-tree ALU architecture coupled with a semi-dynamic implementation and single-rail dynamic circuits enables 70% fewer carry-merge gates and 56% reduction in switching and active leakage energy at equal performance compared to the reference Kogge-Stone implementation” [10] In other words, an increase in energy efficiency was achieved without sacrificing the performance of the processor.

2004-2000

In 2002, a study based on the CMOS technology found that “A speed advantage of a factor of about 2–2.5 is obtained with BGFSB over the conventional design.” Where BGFSB stands for: back-gate forward substrate bias. [8], with the increase in chip density many problems occur, for example power dissipation increases, circuit delays at lower voltages, and a decrease in performance, that is why on of the most important ALUs (the adder) needs to perform at its best. A ripple carry adder will give a “smaller layout area but the delay time for the worst case is a little longer.” In conclusion, the paper shows that the “BGFSB method has been highlighted for low-voltage and high-speed applications. A fast 4-bit ALU has been designed in 1.2mm, N-well CMOS technology for 1 V operation to demonstrate the usefulness of the BGFSB method” [8].

![Image 1](http://example.com/image1.png)

**Fig. 1.** Plot taken from “Clock Gating Based Energy Efficient ALU Design and Implementation on FPGA”[7]

![Image 2](http://example.com/image2.png)

**Fig. 2.** Taken from module 8 “Device Technology”
IV. CONCLUSION

In conclusion, many tradeoffs take place when designing ALUs and FPUs, from performance, to scalability, to execution time, or even chip density. The difficult task is to know what can be given up to achieve a better product. Interconnection is key with these kinds of designs. Also, the most fundamental ALU is definitely one of the most important, that is the adder since it’s key to building multipliers and counter. The common theme for all the Arithmetic Logic Units and all the Floating Point Units was energy efficiency and decrease in power consumption, and as the years go on, this is what has been achieved; higher performance at a lower operating cost.

REFERENCES


<table>
<thead>
<tr>
<th>ALU or Floating Point Architecture Name</th>
<th>Datapath width (bits) or #bits in operands</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>
</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>IEEE-754 Single Precision</td>
<td>45nm and 15nm (ITRS Node)</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>18nm^2 (Cell Area)</td>
</tr>
<tr>
<td>Improving Power-awareness of Pipelined Array Multipliers using 2-Dimensional Pipeline Gating and its Application to FIR Design [4]</td>
<td>4 bit, 8 bit, 16 bit</td>
<td>N/A</td>
<td>2D pipeline array multiplier</td>
<td>N/A</td>
</tr>
<tr>
<td>Fault Demotion Using Reconfigurable Slack (FaDReS), [5]</td>
<td>32 bits</td>
<td>N/A</td>
<td>DSP48 multipliers</td>
<td>N/A</td>
</tr>
<tr>
<td>Design &amp; analysis of 16 bit RISC processor using low power pipelining, [6]</td>
<td>16 bit</td>
<td>In carry select adder unit (CSA)</td>
<td>N/A</td>
<td>28 nm technology</td>
</tr>
<tr>
<td>Clock gating based energy efficient ALU design and implementation on FPGA[7]</td>
<td>8 bit</td>
<td>90-nm Spartan-3 Adder</td>
<td>N/A</td>
<td>90-nm cell Area</td>
</tr>
<tr>
<td>A Fast ALU Design in CMOS for Low Voltage Operation [8]</td>
<td>4-bit</td>
<td>Ripple carry adder</td>
<td>N/A</td>
<td>1.2 mm N-well CMOS technology</td>
</tr>
<tr>
<td>Application Specific Low Power ALU Design. [9]</td>
<td>8-bit</td>
<td>Low Power Adder Design</td>
<td>N/A</td>
<td>CMOS technology</td>
</tr>
<tr>
<td>A 4-GHz 300-mW 64-bit integer execution ALU with dual supply voltages in 90-nm CMOS. [10]</td>
<td>64 bit</td>
<td>sparse-tree semi-dynamic adder architecture</td>
<td>N/A</td>
<td>90nm dual-Vt CMOS technology</td>
</tr>
</tbody>
</table>