Analysis of Multiple Low-Power ALU Designs
Including Pipeline, FSL Logic, DFT scheme and Chain Structure Techniques

Lindsay Renee Davis
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, FL 32816-2362

Abstract—In this paper, I will present several summaries of ALU designs that focus on low-power operation by implementing a number of different designs. A Floating-Point Unit has been included for a comparison in design and metrics. The fundamental metrics that were used for comparison are design year, model type, execution time, design complexity, number of operands, expense, and power dissipation. Some types of ALU designs that will be discussed include the pipelining technique [4][5][6][9], others include FSL logic [10] and DFT scheme with logic design [8].

Keywords—ALU, ITRS technology node, power, bits, pipeline, low-power, FSL, DFT, 2-D gate technique, block scheme, execution time, multipliers, adders.

I. INTRODUCTION

The focus of this paper is on low-power ALU designs because these designs are becoming increasingly used and important to technology innovations as more and more devices are reliant on batteries. Since devices are operating from battery power a large focus of design is trying to find low-power solutions to all aspects of the hardware operation. There are many solutions or ways to decrease power dissipation in a hardware system of a device but the details of this paper will only explain low-power ALU solutions.

The calculation of operands is important because the larger the operand that is used in the system corresponds to the size of the operands required in the design of the ALU which if it is large then requires more space, time, power, components and therefore money.

Some ALU operations include addition, multiplication or Boolean instructions. Addition in one OR gate is when two bits are added together, or addition can be in the form of a Full or Half Adder. In that case there are numerous individual gates inside that add multiple bits together at a time. Multiplication in one AND gate is when two bits are multiplied together. Boolean instructions have a specified order of gate operations that control the operation control path.

An ALU or Arithmetic Logic Unit is a circuit that, based on specific opcodes, performs arithmetic and bitwise operations with integer binary numbers. Arithmetic operations include multiplication, addition, subtraction, or division. The bitwise operations is the binary equivalent operation used for the specific arithmetic operation. For example, if you want to multiply two binary numbers you use the AND gate, and if you want to add two binary numbers you use the OR gate. ALU inputs include two integer operands, status input signal, and an input opcode. Integer operands are the numbers desired for a specific arithmetic operation. Status input signal carries in potentially additional information to the ALU, this could be a carry-in that was stored as a register from the previous operation as the carry-out. An opcode tells the ALU which operation to use, for example 0x08 opcode tells the ALU to ADDi both integer operands. ALU outputs include the integer result and a status. Status output signals carry additional information from the ALU results, such as, overflow, zero, carry-out, and more. The integer input and output operands follow the data path and the status input and output and opcode input follows the control path. The status control path output can store values in registers for the input status control path to potentially use.

A Floating Point Unit performs arithmetic operation on floating point numbers.

Some terms to be familiar with include data bus width, ITRS technology node, execution time, power dissipation, and energy consumption of processors. Data bus width describes how big of a number or “word” can be transmitted, for example, a 4-bit bus can transmit 4 bits of data which as a decimal number can only hold numbers up to 15. ITRS technology node is the size which corresponds to a year and type of semiconductor manufacturing process set forth by a group of industry experts. Execution time is how long the ALU takes to compute an integer based off of a certain operation and integer inputs. Power dissipation is the manner the CPU uses energy for the action of switching devices or energy lost in the form of heat. Energy consumption is how much energy the ALU used in order to complete an operation.

To summarize the next sections of this paper, Section 2 will go into detail about the ten different ALU designs chosen starting with current designs in 2015 all the way back to a 1999 design. Section 3 will go into the specific data details and analysis and Section 4 will conclude this paper.
II. LITERATURE REVIEW

The fundamental metrics that were used for comparison are design year, model type or ITRS technology node, execution time, maximum operation frequency, design complexity, number of operands, expense, and power dissipation. Not every design has all of this information, but as much as possible will be analyzed.

This paragraph will analyze three ALU designs and one FPU design between 2015-2010. In 2015, an Ultra-area-efficient fault-tolerant QCA full adder was designed with an area of 18nm^2 [2]. The results of this design have shown that this full adder decreases the probability of error by 30%. Advantages of this design are small area and high fault-tolerance. Also in 2015, a 16-bit RISC processor using low power pipelining was designed [6]. The results of this design show that the maximum frequency is 100MHz, using the ITRS technology node of 28nm, which is very advanced, saving power. For comparison to the ALUs in 2015, a design was created from energy and area analysis of a FPU [1]. The results of this design show small energy consumption of 2.048mW with the 45nm ITRS technology node and 0.634mW with the 15nm ITRS technology node considering the 32 bit operand width. In 2014, an energy-efficient multiplier-less discrete convolver was designed [3]. The results of this design show that its execution time is 4.09us and its energy consumption is 166.63mJ with a 128-bit operand width. There was no one popular design during this time frame, although some of the same designs are talked about in past designs.

This paragraph will analyze four ALU designs between 2009-2004. In 2009, a low power high speed ALU Using Feedback Switch Logic was designed [10]. This design is a feedback switch with no clock connection. The results of this design is a 14% reduction in delay compared to static CMOS with no clock connection and a 32-bit operand width. In 2008, a chain structure low-power ALU was designed [7]. This design utilizes a chain structure. The results of this design include an execution time of 7.77ns and a 622mW power dissipation with a 4 bit operand width. In 2006, a power-aware pipelined ALU was designed [5]. This design uses the pipeline structure for the multipliers and adders and also the 2-D gating technique. The results of this design are a low power dissipation with a 1.25GHz frequency and a 16-bit operand width. In 2005, a 1.7 GHz low-power delay-fault-testable 32-bit ALU was designed [8]. This design uses a DFT scheme with logic design flow with the multipliers and adders. The result of this design is a reduction of total ALU energy by 18% in the 180nm design and 24% in the 54nm design. Each one of these designs are different, there is not one popular design from this time period.

This paragraph will analyze two ALU designs between 2003-1998. In 2003, a 1.25GHz high throughput power aware FIR filter was designed [4]. This design includes pipelining the multipliers and adders with the use of the 2-D gating technique. The results of this design has a power saving of 62.5% over original design. In 1999, a 200MHz multilevel approach to a low power 16 bit ALU was designed [9]. This design utilizes a two stage pipeline with addition of efficient propagation and generation block schemes of ELM adders. The results of this design are a .86 mm^2 area and power dissipation of 54mw.

During this time period and others the pipelining effect is shown to be more popular.

III. DATA ANALYSIS

In this section the data of the metrics collected will be discussed. In Figure 1, the bits vs. year are shown and the trend is increasing, which shows that wider data paths are being used to send more data as the year’s progress. In Figure 2, the ITRS technology node vs. year can be seen, the trend is following predictions by decreasing through the years. In Figure 3, execution time is increasing, this is probably due to the data path increasing in bit size which necessitates more components and more switching and more time because of the complex design. In Figure 4, the graph for power vs. year is showing a decreasing trend of power consumption, this is good, it shows that as new designs are created each new design, in theory, should save more power than the one before it.

![Figure 1 Bits vs. Year](image1)

![Figure 2 ITRS Technology Node vs. Year](image2)
IV. CONCLUSION

After analyzing these ten ALU designs clear trends can be seen. After seeing the data in the graphs above it is clear to see that several things are increasing and decreasing for very explainable reasons. First, the data and operand bit number is increasing, which is due to technology advances and the necessity of transferring large amounts of data at a time. Second, the ITRS technology node is following the trend almost just as expected, decreasing in size as the years move forward. Third, execution time is increasing which may seem unwanted, but it is actually necessary for the desired power reduction and data bit width increase. Lastly, the power consumption is decreasing as new designs are created. All in all the pipelining technique proved to be the most popular along with using the 2-D gate technique along with it seemed to save even more power.

REFERENCES

<table>
<thead>
<tr>
<th>ALU or Floating Point Architecture Name</th>
<th>Year</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>2015</td>
<td>32 bits (Operands)</td>
<td>N/A</td>
<td>N/A</td>
<td>IEEE-754 Single Precision</td>
</tr>
<tr>
<td>Ultra-area-efficient fault-tolerant QCA full adder [2]</td>
<td>2015</td>
<td>1 bit (Operands)</td>
<td>Ultra-area-efficient fault-tolerant QCA full adder</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>1.25GHz High Throughput Power-aware FIR Filter Design [4]</td>
<td>2003</td>
<td>16 bits (Operands)</td>
<td>Pipelining multipliers and adders with 2-D gating technique</td>
<td>N/A</td>
<td>0.24μm static CMOS logic (Model)</td>
</tr>
<tr>
<td>1.25GHz Improving Power-awareness of Pipelined Array Multipliers [5]</td>
<td>2006</td>
<td>16 bits (Operands)</td>
<td>Pipelining multipliers and adders with 2-D gating technique</td>
<td>N/A</td>
<td>0.24μm static CMOS logic (Model)</td>
</tr>
<tr>
<td>100MHz 16 bit RISC processor using low power pipelining [6]</td>
<td>2015</td>
<td>16 bit (Operands)</td>
<td>Two stage pipelining multipliers and adders with 2-D gating technique</td>
<td>N/A</td>
<td>28nm (ITRS node)</td>
</tr>
<tr>
<td>Application Specific Low Power ALU Design [7]</td>
<td>2008</td>
<td>4 bits (Operands)</td>
<td>Chain structure ALU design with multipliers and adders</td>
<td>N/A</td>
<td>Custom Design</td>
</tr>
<tr>
<td>1.7-GHz low-power delay-fault-testable 32-b ALU in 180-nm CMOS technology [8]</td>
<td>2005</td>
<td>32 bits (Operands)</td>
<td>200ps DFT scheme with logic design flow with multipliers and adders</td>
<td>N/A</td>
<td>180nm and 65nm (ITRS Node)</td>
</tr>
<tr>
<td>200MHz Multi-level approaches to low power 16-bit ALU design [9]</td>
<td>1999</td>
<td>16 bit (Operands)</td>
<td>5ns Two stage pipeline with addition of efficient propagation and generation block schemes of ELM adder</td>
<td>N/A</td>
<td>.86 mm^2</td>
</tr>
<tr>
<td>[Low Power High Speed ALU Using Feedback Switch Logic [10]</td>
<td>2009</td>
<td>32 Bit (Operands)</td>
<td>Feedback switch logic with no clock connection</td>
<td>N/A</td>
<td>90nm CMOS and FSL (Model)</td>
</tr>
</tbody>
</table>