
Jose Ramirez
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, FL 32816-2362

Abstract—Through the time computers and their components have advanced rapidly. The technology that makes the chips is getting better thus the channel size of the transistors is getting smaller. With this decrease in size, the supply voltage of the computer system can be reduced thus reducing the power consumption of the system as well. I will be focusing on the components involved with bit crunching. Over the years, there have been many different processor designs made for use in specific or very general purposes. I will be looking at 10 different designs. Five baseline designs and five non-baseline designs will be analyzed in this paper. I will talk about how many bits are put in at a time, the time to crunch the bits, size of technology and the energy consumption of the device. An interesting design was the ultra-area-efficient fault-tolerant QCA full adder. Another interesting design was the scalable FPGA-based architecture for DCT. Using these specifications I will look at how certain aspects have changed over the last fifteen years.

Keywords—ALU, Power, Datapath, ITRS Technology, Operands, CMOS, FPGA, Clock rate

I. INTRODUCTION

The ALU and Floating Point unit are central parts of the chip. The ALU is responsible for many functions such as addition, subtraction, multiplication, division, and Boolean operations are done. The ALU cannot handle decimal values, which is where the Floating Point unit comes in. The Floating Point unit converts a decimal number into three fields: sign, mantissa, and exponent. Once the decimal number is broken into these fields then the bits can be added, subtracted, multiplied, and divided. The calculation done by these units are crucial to the computer system because every code that is written today is turned into machine code which is then added, subtracted, etc. in order to execute the desired instructions. So in essence everything that is input into a computer is turned into bits. For example, every instruction is encoded as bits in the computer. Since the instructions are encoded as bits they are actually inputs to the ALU. The ALU has inputs called opcode, which tells the ALU which function to execute with the data bits input into the unit. Most analog inputs have to pass through the Floating Point unit in order to normalize them and turn them into bits the ALU can use.

ALU is short for arithmetic logic unit. This is one of the central parts for the computer system. This unit is responsible for most all function done. The ALU has inputs A and B as well as the output C each having the same number of datapata width. Along with the inputs A and B, the ALU has control inputs as well in order to tell the unit what function to perform. The Floating point unit is similar to the ALU as to that is does computations, but the floating point unit only deals with decimal values. This unit breaks down the decimal into an exponent field, mantissa, and sign. The exponent tells the system where the decimal point should be place. The mantissa is the value of the decimal number, and the sign tells the system if the number is positive or negative. Once the floating point unit has turned the number into these fields, the values are then calculated.

It is important to understand some terms before we proceed. One term that is important to know is data bus width. Data bus width is the number of bits that can be input or output from the ALU. This determines how large of a value you can input into the program. Another term that is good to know is ITRS technology node. This is related to the semiconductor devices. When people refer to nanometers with regards to CMOS they are talking about the distance between the gate and substrate of the semiconductor device. This is important because depending on the width of this device you must apply different voltages to switch bits quickly and reliably. Execution time is a very simple concept. Executing time is the time it take for a program to run and terminate successfully. Power dissipation is the amount of power the system will consume while it is not crunching bits. Another way to describe this is by relating it to gas in your car. Power dissipated is like saying that this is the amount of gas your car will use while in neutral. While energy consumption is how much energy it took for the program to run the desired functions successfully. This is like how much gas your car used to drive to a destination.

In the section to come I will be reviewing ten different ALU and Floating Point designs ranging from the year 2000 to 2015. I will be specifically looking at how the size of the technology has changed over the last fifteen years. As well as how the amount of power consumed or power dissipated compares to each ALU or Floating Point unit as time has progressed.
II. LITERATURE REVIEW

2000-2005

2002-NULL convention multiply and accumulate unit with conditional rounding, scaling, and saturation

The Multiply-accumulate unit, or MAC, is a self-timed unit. This unit has a 72 bit datapath and used 250nm CMOS technology. In this class of self-timed circuits, the functional correctness is independent of any delays in circuit elements, through circuit construction, and independent of any wire delays, through the isochronic fork assumption, where wire delays are assumed to be much less than gate delays. Therefore self-timed circuits provide distinct advantages for System-on-a-Chip applications [4]. At the time of operation the unit ran using 3.3V which for today seems high since 1.2V are used instead. This MAC after being pipelined with the Modified Baugh–Wooley algorithm could execute the desired benchmark in 12.7ns [4].

2002-Fast ALU Design in CMOS for Low Voltage Operation

This specific ALU design was used to illustrate how powerful the backgate-forward substrate bias method was for the CMOS technology. The technology used was 1200nm with a 1V supply voltage. The 4-bit ALU employs a ripple carry adder and is capable of performing eight operations - four arithmetic and four logical operations. The BGFSB voltage has been limited to [0.4]V [7]. Some other designs that this paper talks about use different voltages on the chip for different portions of the processor, but this posed a problem due to having to isolate and switch to different voltages quickly. They suggest that by reducing the threshold voltage of the digital circuit you can minimize the time it takes to switch voltages and reduce the space and thus the cost of the adder [7].

2005- Integer Execution ALU with Dual Supply Voltages in 90nm CMOS

This ALU design could run with 32 bits or 64 bits depending on which mode was selected. This processor consumed 300mW of power when crunching bits. This processor consisted of 90nm CMOS technology. This processor was described as a single-cycle integer ALU [8]. This chip was able to run using power performance due to the secondary off-chip voltage [8]. The speed of this processor was achieved due to the implementation of the sparse-tree architecture. This designed helped to speed up the critical path of the adder by moving portions to the non-critical portion if the values were not needed for other calculations [8]. This tree worked by containing conditional branches in order to determine the importance of the bits and compute the result as quickly as possible.

2006-2010

2009-Scalable FPGA-based Architecture for DCT Computation Using Dynamic Partial Reconfiguration

The DCT using the FPGA uses 8 bit operands. This design can compute discrete cosine transforms for eight different zones. The design can change the processing elements for precision or complexity. Finally unused PE can be used for motion estimation calculations [5]. This design uses 24.03-26.27mW for the discrete cosine transforms and 29.23-131.93mW for the motion estimation calculations. Due to the hardware’s ability to reconfigure itself depending on the calculation, this FPGA can have 80 different configurations using the same parts [5]. As the FPGA reconfigures it trades-off visual quality, power consumption, processing clock cycles, and reconfiguration overhead depending on which format it takes.

2009-32 bit Multiplication and Division ALU Design Based on RISC Structure

This RISC floating point multiplication and division unit was used with IEEE 754 standard. This device supported 32 bit operands and GW48 EDA system. This computed the results of multiplication and division in the homo-hardware with level 4 pipelining [10]. The supply voltage was low so it did not consume a lot of power. The design also included four sections that were each pipelined and used dependent functions to ensure the accuracy of the calculated results[10].

2011-2015

2013-Green ALU Design Using Clock Gating Technique

This 64 bit ALU design was implemented on the Virtex-6 FPGA. Using the XC6VLX75T technology, this design consumed 736mW dynamic power and 81mW of leakage power [9]. The main proponent of this design is the clock gating. Clock gating is an energy efficient technique used in order to reduce the power the ALU uses by switching off parts of the ALU that are not in use. This is determined by the instruction decoder units [9]. The clock gating technique was first used and tested on an 8 bit ALU to reduce the dynamic and clock power consumption. Then after the design was tested it was scaled to the 64 bit design with varying clock rates ranging from 1 GHz – 1THz in increments of ten [9].


With this 128 bit Energy-Efficient Multiplier the execution time was 4.09 us with the design implemented on the Virtex 6 FPGA. The energy consumed by this chip in order to run the desired operation at this speed was 166.63mJ which can translate to 40.7mW by dividing the energy consumed by the execution time of the program. This design was inspired by
Fourier transforms. In this paper it was proposed that a new domain transformation be used. This transformation was called Probabilistic Domain Transformation. In this PDT method, the input signal will be converted to a series of samples through the transform [3].

2015-Energy and Area Analysis of a Floating-Point Unit

In this paper the 45nm and 15nm CMOS technology is analyzed using the IEEE 754 single precision floating point unit. This 32 bit unit was found to consume 2.048 mW with the 45nm technology, and 0.634 mW with the 15nm technology [1]. With this reduction in channel size a higher density can be achieved. A smaller supply voltage can also be utilized due to the smaller channel. But the possibility of more leakage current relative to size could be present. The results have shown that using the 15nm technology we can have 4 times less energy and 3-fold smaller footprint [1].

2015-Ultra-area-efficient fault-tolerant QCA full adder

The fault tolerant full adder designed in this paper is very area efficient due to its high density. The design only occupies 18nm²/2 of cell area on the chip. Although the paper only shows the design for 1 bit, this is just to illustrate the high fault tolerance of the quantum-dot cellular automata full adder [2]. This low power design was constructed using one majority gate and one rotated majority gate [2]. With this design the adder was improved greatly in regards to the cell count complexity and area. The logic test is tested with the PTM evaluation model. Also in future designs the gates implemented here can be used to find the most common faults in the arithmetic circuits of the processor [2].

2015-16 bit RISC Processor Using low Power Pipelining

The 16 bit RISC Processor uses low power pipelining using 28nm technology while expending 71mW of dynamic power. The RISC processor consists of mainly the ALU, a universal shift registers and barrel shifter. A modified Harvard architecture was used to separate the memory for data and instructions. The architecture modification made utilized in carry select adder unit. Calculations in the RISC or Fetch, Decode, execute, write back is implemented in the 2 stage pipelining with both the positive edge & negative edge [6]. This design had a total latency of 1.5 clock cycles.

III. DATA ANALYSIS

![Fig. 1. As time has progressed technology size has decreased.](image1)

![Fig. 2. Although the power consumption has fluctuated, it has been slowly decreasing.](image2)

![Fig. 3. Datapath width does not really show a trend](image3)
IV. CONCLUSION

After a review of the literature and the figures, as time has progressed the chips have become more dense and more efficient. The size of the CMOS technology has gone from 1200 and 250nm in 2002 to 45 and 15nm in 2015 as seen in figure 1. Also the energy consumption has decreased as well due to the decrease in CMOS technology as seen in figure 2. The supply voltage has dropped from 3.3V to almost 1V in the last 13 years. One thing that was noticed was that the datapath or the bits or the operands did not have a specific trend with time as seen in figure 3. This may be due to the fact that the number of operands does not really have a direct effect on the size or energy of the components that make up the registers the bit hold, but the components that make up the registers to hold the bits affect the energy and size of the system. I also noticed that most of these worked with the adder portion as seen in figure 4. This is probably due to the reliability of the adder. The adder may be slower than the multiplier but the adder should be more reliable since the system is doing a simple instruction many times quickly, as opposed to a more complex instruction.

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] 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] 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>NULL convention multiply and accumulate unit with conditional rounding, scaling, and saturation [4] 2002</td>
<td>72 bits (Operands)</td>
<td>N/A</td>
<td>12.7ns MAC unit</td>
<td>N/A</td>
</tr>
<tr>
<td>16 bit RISC Processor Using low Power Pipelining [6] 2015</td>
<td>16 bit (Operands)</td>
<td>RISC Processor</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Fast ALU Design in CMOS for Low Voltage Operation [7] 2002</td>
<td>4 bit (Operands)</td>
<td>BGFSB 1.2 um N-well CMOS</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Integer Execution ALU with Dual Supply Voltages in 90nm CMOS [8] 2005</td>
<td>32 and 64 bit (Operands)</td>
<td>Adder core sparse-tree architecture</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Green ALU Design Using Clock Gating Technique [9] 2013</td>
<td>64 bit (Operands)</td>
<td>N/A</td>
<td>Virtex-6 FPGA</td>
<td>N/A</td>
</tr>
<tr>
<td>32 bit Multiplication and Division ALU Design Based on RISC Structure [10] 2009</td>
<td>32 bit (Operands)</td>
<td>N/A</td>
<td>N/A</td>
<td>ALU multiplication Floating-Point IEEE754 standard</td>
</tr>
</tbody>
</table>