The Evolution of Processor Architecture through Hardware & Software Advances for Energy Optimization and Power Reduction through the 21st Century

Hieu Pham
Department of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816-2362

Abstract—The evolution of processor architecture has benefited the engineering community through optimization and power reduction. With optimization comes analysis of error and speedup times for different systems. The classical components of a computer are analyzed throughout this paper in order to compare values and how they have developed over a certain period of time. The metrics that will be presented are memory capacity, core count, clock rate, and word width. Throughout this paper, multiple examples will be given showing that faster does not always mean better and bigger does not always mean more storage or optimal performance. Other factors such as Moore's Law is also included in this paper to show the reoccurring effects of previous theory. From the early 1900s to present day, the influence of pipelining and hardware advances in engineering have helped developers find ways to optimize certain architecture.

Keywords—Memory Capacity, Clock rate, Moore's Law, speedup, pipelining, power reduction, scaling, Amdahl's Law, SNAP, OTBSAF, transistor growth

I. OVERVIEW OF PROCESSOR ARCHITECTURE

The 5 classic components of a computer system are input, output, memory, datapath and control. The input is the data that the user puts into the machine. The memory is the part of the machine that stores information. The datapath is a conveyer belt where bits move a long in order to transfer information. The control is part of the processor which gives instructions on how to use the data on the datapath. The output is the result of the given commands by the control given from memory.

A bus in general is a bundle of wires. A processor can only operate on values that reside internally. Any value in memory must be transferred into the processor before the corresponding operation occurs. To obtain or provide a data value from or to memory, the processor must specify an address. Separate buses for address and data are used to connect the processor to memory. Operations the processor performs have different categories (i.e. arithmetic operations or data transfer between memory).

Separate buses for address and data are used to connect the memory to processor. The data bus is a highway for data bits as d wires between processor and memory. The address bus has a =log(2) (memory size in bytes) wires which is determined by memory capacity only.

The CPU clock rate determines how many instructions are executed per second. The number of different machine instructions is the instruction set size. Reducing the number of clock cycles is used to increase performance. Therefore it is necessary to reduce the instruction set size. The instruction set is the number of bits used for data representation. A bit is an acronym for Binary Digit. One bit indicates a single binary state (true/false). A byte is always 8 bits. A KB is 2^10 bytes. A MB is 2^20 (GB is 2^30 etc.). For memory, the prefixes always use base 2. For speed (frequency), the prefixes always use base 10. For example, 10KHz is 10^3 Hz.

The specific metrics in Figure 1 are significant to describe the architectures at a high level of design abstraction because all computer systems have become faster and more efficient throughout the years. According to Moore’s law, the number of transistors on a chip doubles (2 fold) every 2 years. The clock rates are also increasing making program going through the instruction set exponentially faster.

The processor performance equations for execution time in terms of instruction count, CPI, and cycle time is:

\[
\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{Cycle time}
\]

The instruction count is the dynamic count of instructions executed at runtime, not the number of instructions of assembly code. The CPI is the effective or average number of clock periods that each instruction in the dynamic instruction mix of the program. The cycle time is the duration of time for one clock cycle of the processor.

The use of parallelism and multiple cores to reduce execution time is an effective approach to improve performance by providing more processors or channels in
parallel rather than trying to increase one processor’s speed. Higher performance limits are possible with parallelism and systems may be more cost effective. Parallelism allows the user to achieve a transfer of more data bits quickly between processor and memory for higher performance. It provides independent channels to transmit multiple bits. Throughput can be increased even though execution time remains fixed. Based on Amdahl’s law for parallelization, the overall speedup is the old execution time over the new execution time.

\[
    \text{Speedup}_{\text{overall}} = \frac{\text{Execution time}_{\text{old}}}{\text{Execution time}_{\text{new}}}
\]

Providing multiple CPUs to process tasks in parallel may achieve higher performance. The maximum speed up equation is shown below:

\[
    \text{Maximum Speedup} = \frac{T(1-f) + T_f}{T(1-f) + \frac{T_f}{N}}
\]

In Section 2 there are ten processor architectures spanning from 1990s until today that are reviewed. There are five baseline computer systems which metrics are provided in Table 1. These values are compared to other architectures to compare how these systems have evolved over time.

II. LITERATURE REVIEW

Five Baseline Computer Systems:

I) SNAP-1 Parallel AI Prototype [5]

II) OTBSAF [1]

III) Two-dimensional pipeline gating [2]

IV) Fuzzy ARTMAP Networks [3]

V) Fine Grain Pipelining [4]

In 1992, the SNAP-1 Parallel AI Prototype was developed at the University of Southern California [5]. It contained 144 CPUs, each of which was a Texas Instruments TMS320C30 32-bit DSP chip clocked at 25 MHz and had 256KB of memory which were organized on to 8 large circuit boards. The SNAP-1 ISA had 20 instructions for the special purpose computation of marker-passing. It bridged the semantic gap by providing complex powerful instructions close to the need of Natural Language Understanding (NLU) applications.

In 1997, the Fuzzy ARTMAP Networks architecture was developed at the University of Central Florida. The purpose of the architecture was to use Parallel Implementation on a DECmp/Sx-1208. It had 8,192 Processors and a 4 bit word width. The parallel implementation of Fuzzy ARTMAP indicated a speedup of 1000-fold which resulted in a combined training and testing time of under 4 minutes. [3]

In 2003, the Fine-Grain Pipelining multipliers and adders was developed at the University of Central Florida in order to have a high throughput power aware FIR filter [4]. The clock rate it had was 1250 MHz and it also had a 16 bit word width. The results of this architecture yielded a 62.5% 2D technique power savings.

In 2005, the OTBSAF scalability technique was developed at University of Central Florida [1]. Its general purpose was entity scaling. The primary use was to show that processors in laptop computers have sufficient capability to simulate the quantity of entities required for Battalion sized force on force training exercises [1]. The use of 2 processors gained in increase of 144 vehicles from 1 processor. When using 3 processors, there was an increase of 92 vehicles.

In 2006, Two-dimensional pipeline gating was developed at the University of Central Florida. The purpose of this architecture was to improve power awareness on FIR design. It had a 1250 MHz clock rate and a 16 bit word width. Using the pipeline gating, 65-66% power savings & 44-47% latency reduction was found throughout testing. The basic premise behind the architecture is to gate the clock to registers in both vertical direction and horizontal direction [2].

In 2009, a new architecture using subthreshold processor designed was developed [8]. The purpose of this architecture was to utilize it in ultralow power research. This architecture is a highly efficient subthreshold microprocessor targeting sensor application. The Processor has 26 chips and operates at 833KHz. The memory size is 64KB and the word width is a 4 entry 16 bit width. The optimal testing results yielded a 2.6pJ/instruction at 360 mV [8].

In 2009, a five-stage parallel pipeline architecture was developed in Kollam India. The purpose of this architecture was power reduction. The operating clock rate was 205.7MHz. The memory for the data was 32 x 256 bytes and the instruction was 32 x 1024 bytes. The word width was 32 bits. At 200MHz, the testing results yielded a 19 percent decrease in power [9].

In 2009, Dynamic Partial Reconfiguration was developed at the University of Central Florida. The purpose of this architecture was to scale FPGA based architecture for DCT (Discrete Cosine Transform). It had a 100MHz clock rate and a 8 bit word width. The scalable architecture for video coding is applicable for systems that deliver the contents to a wide range of terminals [6].

In 2011, a 48-Core IA-32 Processor in 45 nm CMOS Using On-Die Message-Passing and DVFS was developed by members of IEEE [10]. The purpose of this architecture was performance and power scaling. The architecture used 6x4 2D-mesh network on chip with 1.3 billion transistors and operated at 1GHz. The memory was 384KB and featured a 36
bit word width in the core architecture. It had 48 cores and the cross benchmark error was 6.5-7.9% [10].

In 2014, Processor Design using Asymmetric detection was developed UMIC Research Centre in Germany [7]. The purpose of the architecture was to use asymmetric detection to detect error and correct it. The clock rate was 200MHz and the word width was 32 bits. The concept was based on the continuous shrinking of device size which introduced reliability as a new design challenge for embedded processors. Asymmetric reliability utilizes unequal protection levels for different system components [7].

III. DATA ANALYSIS

Fig 1: The graph shows the memory capacity of vs the year. As time goes by, the amount of memory required to store a data in a program decreases which saves space for storage.

Fig 2: The word width varies throughout the 20th and 21st centuries. During the early 1900s, the word width was high then decreased then increased to the present. However word width is dependent on the application for each architecture. Therefore not all architectures require a word width of large value.

Fig 3: The number of cores per year decrease as a result of optimization of hardware and software.

Fig 4: The clock rate varies from each architecture throughout the 20th and 21st centuries. During the early 2000s, the clock rate was high and dropped back down during present day. In the 20th century, the operating clock rate was relatively low to today’s values.

Metrics covered analyzed in this paper:
- CPU clock rate (MHz) vs. Year
- Memory Capacity (MB) vs. Year
- Number of Processors or Cores vs. Year
- Data bus Word Width (bits) vs. Year

IV. CONCLUSION

In conclusion, the different architectures analyzed in this report showed how optimization through different techniques has changed the need for certain types of design. For example, Moore’s law shows that the number of transistors double every 2 years and is proven by the architectures developed in 2014 compared to the 2011 values. Furthermore, I found that higher word width does not necessarily mean more efficiency. Also, I concluded that the memory capacity used to run through an instruction set of a program decreased because of optimization of hardware and software. Overall, the evolution of architecture and design using various methods have changed the world of computers and programming for the future.
REFERENCES

V. REFERENCES


<table>
<thead>
<tr>
<th>Name of Architecture [reference]</th>
<th>Year</th>
<th>Purpose: Application-Specific or General-purpose Computation</th>
<th>Die Area, Number of Transistors, or Number of Chips/Boards/etc.</th>
<th>CPU Clock Rate (MHz)</th>
<th>Memory Capacity (MB)</th>
<th>Data Bus Word Width (bits)</th>
<th>Number of Cores or CPUs</th>
<th>Ideal Speedup for 99% parallel code (ignoring overheads)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNAP-1 Parallel AI Prototype [5]</td>
<td>1993</td>
<td>NLU: Special Purpose</td>
<td>144 DSP Chips on 8 large circuit boards</td>
<td>25</td>
<td>256KB/144 CPU = 36.86MB</td>
<td>32</td>
<td>144 cores</td>
<td>144 cores so $\text{Told}/\text{Tnew}= 1/(0.01+ (0.99/144)) = 59.26$-fold</td>
</tr>
<tr>
<td>OTBSAF Scalability on Pentium III/4 and Athlon 64/XP3000 [1]</td>
<td>2005</td>
<td>Entity Scaling</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Incremental increase 144 vehicles for 2 processors and increase of 92 veh. for 3 processors</td>
</tr>
<tr>
<td>Two-dimensional pipeline gating [2]</td>
<td>2006</td>
<td>FIR: Improve power-awareness</td>
<td>N/A</td>
<td>1250</td>
<td>N/A</td>
<td>16</td>
<td>N/A</td>
<td>65-66% power savings &amp; 44-47% latency reduction</td>
</tr>
<tr>
<td>Fuzzy ARTMAP Networks [3]</td>
<td>1997</td>
<td>Parallel Implementation on DECmmp/Sx-1208</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>4</td>
<td>8,192 Processors 1000 fold</td>
</tr>
<tr>
<td>Fine Grain Pipelining [4]</td>
<td>2003</td>
<td>High Throughput Power awareness</td>
<td>N/A</td>
<td>1250</td>
<td>N/A</td>
<td>16</td>
<td>N/A</td>
<td>62.5% 2D technique power savings</td>
</tr>
<tr>
<td>Dynamic Partial Reconfiguration [6]</td>
<td>2009</td>
<td>Scaling FPGA based architecture for DCT</td>
<td>N/A</td>
<td>100</td>
<td>N/A</td>
<td>8</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Processor Design Asymmetric Detection [7]</td>
<td>2014</td>
<td>Versatile asymmetric error detection/correction</td>
<td>N/A</td>
<td>200</td>
<td>N/A</td>
<td>32</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Subliminal Processor [8]</td>
<td>2009</td>
<td>Subthreshold operation; Optimize Energy</td>
<td>26 Chips</td>
<td>833KHz</td>
<td>64KB</td>
<td>4 entry 16 bit</td>
<td>N/A</td>
<td>2.6 pJ/instruction at 360 mV</td>
</tr>
<tr>
<td>Five-Stage Parallel Pipeline [9]</td>
<td>2009</td>
<td>Power reduction</td>
<td>N/A</td>
<td>205.7</td>
<td>Data: 32 by 256 bytes Instruction: 32 by 1024 bytes</td>
<td>32</td>
<td>N/A</td>
<td>@200MHz 1.139 from 1.359, 19% decrease in power</td>
</tr>
<tr>
<td>A 48-Core IA-32 Processor in 45 nm CMOS Using On-Die Message-Passing and DVFS [10]</td>
<td>2011</td>
<td>Performance and Power Scaling</td>
<td>6x4 2D-mesh network-on-chip, 1.3 billion transistors</td>
<td>1GHz</td>
<td>384KB</td>
<td>36 bit core architecture</td>
<td>48</td>
<td>Cross-benchmark error 6.5-7.9%</td>
</tr>
</tbody>
</table>