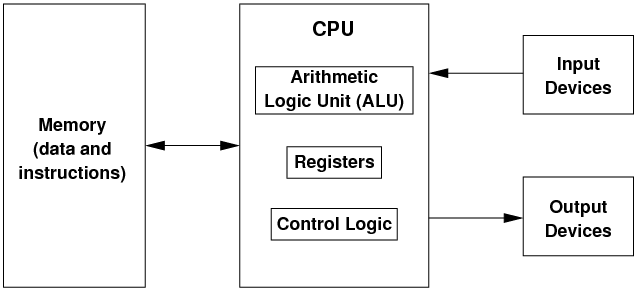
Computer Architecture

Derrick Murphy

Winston –Salem State University

Babbage and Ada Lovelace, describing the *Analytical Engine* was the first recognized computer architecture. The term “architecture” in computer literature can be traced to the work of Lyle R. Johnson, Mohammad Usman Khan and Frederick P. Brooks, Jr., 1959 members of the Machine Organization department in IBM.  In attempting to characterize his chosen level of detail, he noted that his description of formats, instruction types, hardware parameters, and speed enhancements was at the level of “system architecture”. “Computer architecture, like other architecture, is the art of determining the needs of the user of a structure and then designing to meet those needs as effectively as possible within economic and technological constraints.” (Jr, 1962)

Computer architecture is the art of assembling logical elements into a computing device; the specification of the relation between parts of a computer system. The purpose of computer architecture is to design a computer which gets the most out of performance while keeping power consumption in reasonable, inexpensive in relation to its performance, and is also dependable. Many aspects are to be measured which includes Instruction Set Design, Functional Organization, Logic Design, and Implementation. The implementation involves Integrated Circuit Design, Packaging, Power, and Cooling. Optimization of the design requires familiarity with Compilers, Operating Systems to Logic Design and Packaging.



An instruction set architecture is the interface between the computer's software and hardware. The instruction set delivers instructions to the processor; computers do not understand high level languages which have few language elements that translate directly into a machine's native operation code. The instruction set consists of addressing modes, instructions, native data types, registers, memory architecture, interrupt and exception handling, and external I/O. A processor only understands instructions encoded in some numerical fashion, usually as binary numbers. Compilers are needed to translate high level languages into instructions.

Computer organization helps increase performance-based merchandise. Software engineers need to know the processing ability of processors. They may need to enhance software in order to gain the most performance at the least expense. This requires detailed analysis of the computer organization. Computer organization and features also affect power consumption and processor cost.

Once an instruction set and micro-architecture are defined, a practical machine must be created. This stage is called the implementation, but known as engineering design process. The engineering design process embodies several steps, feasibility assessment, establishing design requirements, preliminary design, detailed design, production planning and tool design, and finally production.

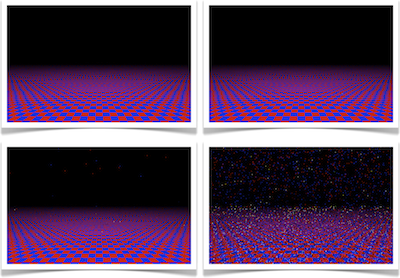
The purpose of a feasibility assessment is to determine whether the engineer's project can proceed into being developed. The feasibility assessment measures if the project is based on an attainable idea, and it needs to be within cost limitations. It is of highest significance to have an engineer with experience and good judgment to be involved in this portion of the feasibility study, for they know whether the engineer's project is possible or not.

One of the most essential elements of the design process is forming design requirements, normally performed alongside the feasibility analysis. Design requirements that are set control the design of the project throughout the remaining process. Some design requirements include hardware and software parameters, maintainability, availability, and testability.

The preliminary design ties the break between the design concept and the detailed design phase. In this task, the overall system configuration is defined, and plans, illustrations, and drafts of the project will provide early project structure. Parameters may change during detailed design and optimization, but the preliminary design focuses on creating the general framework to build the project on.

At the detailed design portion of the engineering design process engineer completely describe a product through solid modeling and sketches. In this phases all aspects of the project are composed, this includes operating parameters, external dimensions, material requirements, packaging requirements, external markings, etc. Once the detailed design is done, the production planning and tool design is begun. This step is nothing more than preparing how to mass-produce the project and which tools should be used in the manufacturing of the part. Once testing and prototype testing is complete the engineering design process is finalized and production begins.

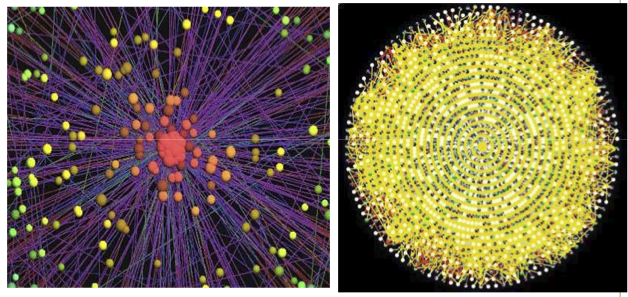
Even today there are still improvements to be made to technology. Research groups formulate various projects that could expand how effects computers may really become. A few projects in development right now are EnerJ, Grappa, and WaveScalar.



Energy is very significant in computer systems design. Battery life is a huge limitation in mobile systems. More fundamentally, “current trends point toward a ``utilization wall,'' in which the amount of active die area is limited by how much power can be fed to a chip.” The current method of reducing energy consumption has been using low-power architectures, devices are designed with performance and power trade-offs, and resource supervision. Even though those methods have been operational and can be applied without programmer involvement, “exposing energy considerations to higher-level software can enable a whole new set of energy optimizations.” The goal of the EnerJ project is to develop new programming models, system support, and hardware for energy-aware programming, with the goal of reducing energy consumption in modern computing system.

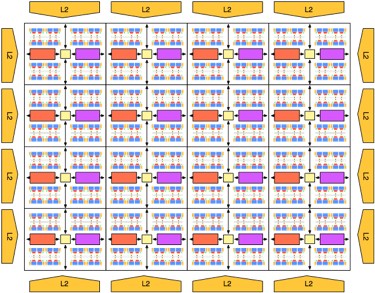
The goal of the project is to design language and runtime monitoring techniques to express QoS (quality of service) where errors can be endured to accomplish energy savings, applying tools for helping programmers use these techniques and for compilers to transfer the information to the hardware, and design micro-architectures and hardware accelerators that can better control this information.

*Grappa* is a latency-tolerant runtime for mass-market clusters that moderates junk, allowing graph processing to scale up even in the presence of diminishing locality and increasing latency. Node memories become likely to join vertices as system size grows due to separation. This reduces the rate of transversal. Though parallelism and hardware resources rise, performance reduces.



Grappa enables high-throughput graph processing on commodity clusters. Unfortunately, this solution is not available with commodity parts alone. Modern mass produced computer systems are designed to achievement “spatial locality via cache and local memory to achieve high efficiency. Unfortunately, when processing graphs spatial locality is often difficult, if not impossible, to express.” Our goal is to develop a latency-tolerant system built mostly in software and commodity parts.

The basis of up and coming applications is to crunch large graphs, such as social network analysis and bioinformatics. Graph analytics algorithms exhibit little locality, this present substantial performance issues. Hardware multithreading systems show that with enough concurrency, we can tolerate long latencies.



WaveScalar is a dataflow instruction set architecture and execution model designed for scalable, low-complexity high-performance processors. Due to silicon technology raw transistors are exponential increasing availability. Successfully rendering this source into application execution is an open challenge.

Ever increasing wire delay relative to switching speed and the exponential cost of circuit complexity make simply scaling up present processor designs useless. It is unique among dataflow architectures in that it efficiently provides traditional memory semantics in order to perform applications written in vital languages.

The WaveScalar ISA is designed to run on a smart memory system. Each instruction in a WaveScalar binary executes in place in the memory system and openly communicates with its dependents in dataflow manner. WaveScalar architectures cache instructions and the values they operate on in a WaveCache. By making computation local and data in physical space, the WaveCache reduces long wire, high-latency communication.

**Work Cited**

Jr, F. P. (1962). Planning a Computer System: Project Stretch.

computer architecture - definition of computer architecture by the Free Online Dictionary, Thesaurus and Encyclopedia. (n.d.). Retrieved from http://www.thefreedictionary.com/computer+architect computer architecture - definition of computer architecture by the Free Online Dictionary, Thesaurus and Encyclopedia. (n.d.). Retrieved from http://www.thefreedictionary.com/computer+architecture

EnerJ - Sampa. (n.d.). Retrieved from http://sampa.cs.washington.edu/sampa/EnerJ

Grappa: high-throughput graph processing on mass-market clusters. (n.d.). Retrieved from http://sampa.cs.washington.edu/grappa/

WaveScalar | Computer Science & Engineering. (n.d.). Retrieved from http://www.cs.washington.edu/node/2481/

What is Instruction Set? (n.d.). Retrieved from http://www.computerhope.com/jargon/i/instset.htm