ADVANCE COMPUTER ARCHITECTURE

Monika Jatiwal, Chetna Arora, Charu Arora

OUTLINE

ABSTRACT

This paper presents novel methods in teaching advanced computer architecture courses. These methods include presenting fundamental computer architecture issues employing visual aids to teach fundamentals concepts like pipelining and processing. This paper studies about the history for Advance computer architecture. This paper explains about the parallel processing, architecture of parallel processing, pipelining and its role.

1. INTRODUCTION

What is computer architecture?

Architecture: How are things organized and what you can do with their functionality? Many different "Architectures" exist in a system like application/System architecture; Structure of the application .It also involves Interface to outside world like API, libraries, GUIs, etc. The term architecture is used here to describe the attributes of a system as seen by the programmer, i.e., the conceptual structure and functional behavior as distinct from the organization of the dataflow and controls, the logic design, and the physical implementation. Computer architecture is the Interface between hardware and software

1.1 HISTORY

The first document computer architecture was in the correspondence between Charles Babbage and Ada Lovelace ,describing the analytical engine. The term architecture in the computer literature can be traced to the work of Lyle Johnson ,Mohammad Usman khan and Frederick P.Brooks, Jr.,members in 1959 of the machine Organization department in IBM's main research center .Johnson had the opportunity to write a proprietary research center. Johnson had the opportunity to write a proprietary research communication about the Stretch, an IBM developed supercomputer.

The earliest computer architectures were designed on paper and then directly built into the final hardware form. Later, computer architecture prototypes were physically built in the form of a TTL computer -- such as the prototypes of the 6800. As of the 1990s, new computer architectures are typically "built", tested, and tweaked inside some other computer architecture in a computer architecture simulator; or inside a FPGA as a soft microprocessor; or both -- before committing to the final hardware form.

1.2 SUB-CATEGORIES OF ADVANCE COMPUTER ARCHITECTURE

The discipline of advance computer architecture has three main subcategories.

- *Instruction Set Architecture*, or ISA. The ISA defines the codes that a central processor reads and acts upon. It is the machine language, including the instruction set, word size, memory address modes, processor registers, and address and data formats.
- *Microarchitecture*, also known as *Computer organization* describes the data paths, data processing elements and data storage elements, and describes how they should implement the ISA. The size of a computer's CPU cache for instance, is an organizational issue that generally has nothing to do with the ISA.
- *System Design* includes all of the other hardware components within a computing system.

1.3 PARALLEL PROCESSING

Parallel processing is the ability of the brain to simultaneously process incoming stimuli of differing quality. This becomes most important in vision, as the brain divides what it sees into four components: color, motion, shape, and depth.

In computers, parallel processing is the processing of program instructions by dividing them among multiple processors with the objective of running a program in less time. In the earliest computers, only one program ran at a time. A computationintensive program that took one hour to run and a tape copying program that took one hour to run would take a total of two hours to run. An early form of parallel processing allowed the interleaved execution of both programs together. The computer would start an I/O operation, and while it was waiting for the operation to complete, it would execute the processor-intensive program. The total execution time for the two jobs would be a little over one hour, The next improvement was multiprogramming. In a multiprogramming system, multiple programs submitted by users were each allowed to use the processor for a short time.

Vector processing was another attempt to increase performance by doing more than one thing at a time. In this case, capabilities were added to machines to allow a single instruction to add or subtract, or multiply, two arrays of numbers. This was valuable in certain engineering applications where data naturally occurred in the form of vectors or matrices. In applications with less well-formed data, vector processing was not so valuable. The next step in parallel processing was the introduction of multiprocessing. In these systems, two or more processors shared the work to be done. The earliest versions had a master/slave configuration.

1.4 THE ARCHITECTURE OF PARRALEL COMPUTERS

Hardware Issues

- Number and Type of Processors
- Processor Control
- Memory Hierarchy
- I/O devices and Peripherals
- Operating System Support
- Applications Software Compatibility

Operating System Issues

- Allocating and Managing Resources
- Access to Hardware Features

– Multi-Processing

– Multi-Threading

• I/O Management
• Access to Peripherals
• Efficiency

Applications Issues
• Compiler/Linker Support
• Programmability
• OS/Hardware Feature Availability
• Compatibility
• Parallel Compilers
– Preprocessor – Pre-compiler
– Parallelizing Compiler

Architecture Evolution
• Scalar Architecture
• Pre-fetch Fetch/Execute Overlap
• Multiple Functional Units
• Pipelining
• Vector Processors
• Lock-Step Processors
• Multi-Processor

1.5 PIPELINING

An instruction pipeline is a technique used in the design of computers to increase their instruction throughput (the number of instructions that can be executed in a unit of time). The basic instruction cycle is broken up into a series called a pipeline. Rather than processing each instruction sequentially (one at a time, finishing one instruction before starting the next), each instruction is split up into a sequence of steps so different steps can be executed concurrently and in parallel (at the same time).

Instr. No.	Pipeline Stage						
1	IF	ID	EX	MEM	WB		
2		IF	ID	EX	MEM	WB	
3			IF	ID	EX	MEM	WB
4				IF	ID	EX	MEM
5					IF	ID	EX
Clock Cycle	1	2	3	4	5	6	7

Pipelining is the property through which it start handling next instruction while the current instruction is in progress. Feasible when different devices at different stages

Time between instructions pipelined = Time between instructions nonpipelined/
Number of pipe stages

2. WHY STUDY COMPUTER ARCHITECTURE?

Computer architecture builds faster/better processors.
We study computer architecture because:
1: User requirements are constantly changing.

2: Technology changes rapidly making past choices often obsolete.

3: Also opens up new opportunities.

4: To design and construct application specific solutions in the field of computer architecture.
5: To appreciate that the solution to any problem in computer science is likely to be quickly invalidate by time to strive for solution that minimize the effects of reality.

6: To develop confidence in specifying computational requirements and formulating original solution.

3. ROLE OF THE COMPUTER ARCHITECTURE

Architect: Define hardware/software interface

It defines the hardware organization. **Goal:**

1. Determine important attributes e.g., performance
2. Design machine to maximize those attributes under constraints e.g., cost, complexity, power. It studies how to :

Study applications

Consider underlying technology

Cost

Performance

Complexity Power

Reliability

4.

5. SUMMARY

A number of concepts and system configurations related to obtaining highperformance computing via parallelism were introduced. Parallel processing , pipelining systems and sub-categories of Advanced computer architecture were Introduced. Why we should study and role of Advanced computer architecture is briefly explained.

6. REFERENCE

1. J.L. Hennessy & D.A. Patterson, "Computer Architecture A Quantitative Approach 3rd edition,"
2. www.csee.umbc.edu
3. www.slideshare.net
4. Advance computer architecture by kai Hawang

5. Heald et. all, “Implementation of the third generation SPARC V9 64-b microprocessor,” ISSCC Digest of Technical Papers, pp 412-413, 2000.