International Journal of Engineering Research & Management Technology ISSN: 2348-4039

March-2016 Volume 3, Issue-2

Email: editor@ijermt.org www.ijermt.org

Review On Computer Architecture: A Survey

Rishabh Raman, Vinay Shrivastava' Shivanand Mishra
Students
IIMT College of Engineering
Gr. Noida

ABSTRACT:

In this paper we suggest a different computing environments a worthy new direction for computer Architecture research: personal mobile computing, where portable devices are used for visual computing and personal communications tasks. Such a device supports in an integrated fashion all the functions provided today by a portable computer, a cellular phone, a digital camera and a video game. The requirements placed on the processor in this environment are energy efficiency, high performance for multimedia and DSP functions, and area efficient, scalable designs.

In recent years, advances in computer architecture have slowed dramatically with most simulation results demonstrating only incremental architectural innovation. This is further exacerbated by increased processor and system complexity spurred by a seemingly unlimited number of transistors at computer architect's disposal. Computer architects produce a myopic view of their systems through the lens of slow, highly-detailed software simulation or fast, coarse-grained software simulation, with fidelity always in question. By leveraging silicon technology scaling in Field Programmable Gate Arrays (FPGAs), hardware can be used to accelerate simulation, emulation, or prototyping of systems.

KEYWORDS: multimedia, software, communication.

INTRODUCTION:

Historically, software simulation has been the vehicle of choice for studying computer architecture because of its flexibility and low cost. Regrettably, users of software simulators must choose between high performance or high fidelity emulation. In contrast, building hardware in Application Specific Integrated Circuits (ASICs) provides high performance and accurate results, but lacks the flexibility to explore multiple designs. It is also very expensive. These tradeoffs have impeded our ability to thoroughly explore and evaluate new computer architectures. This lack of simulation fidelity and speed is further aggravated by the increase in multithreaded and/or multicourse microprocessor architectures.

Traditionally, computer architects have leveraged increasing transistor density to implement a single large processor that exploits instruction level parallelism (ILP). However, continued performance gains from ILP are becoming increasingly difficult to achieve due to limited parallelism among instructions in typical applications [1]. Likewise, the problems associated with designing ever-larger and more complex monolithic processor cores are becoming increasingly significant. These problems include higher bug rates, longer design and verification times caused by the design complexity, and the need to design for increasing wire delay [2]. This fact has spurred great interest in exploiting thread-level parallelism (TLP) among independent threads to continue historical microprocessor performance trends.

Advances in integrated circuits technology will soon provide the capability to integrate one billion transistors in a single chip [1]. This exciting opportunity presents computer architects and designers with the challenging problem of proposing microprocessor organization sable to utilize this huge transistor budget efficiently and meet the requirements of future applications. To address this challenge, IEEE Computer magazine hosted.

For too long, operating systems researchers and developers have pretty much taken whatever computer architects.

have dished out. With occasional exceptions (e.g., virtualization support), architecture researchers do not appear to have sought or encouraged innovations that would improve the execution environment for an OS.

Even worse, many do not bother to simulate and report on OS behaviour when evaluating their proposals

THE BIG QUESTION IS WHAT GOES INTO ARCHITECTURE

TOO MUCH:

– Too restrictive

• Additions take 1 cycle to complete

• TOO LITTLE:

– Lost opportunity

– Substandard performance

• Subtract and branch if negative is good enough

• Multimedia instruction set extensions.

This is the proceeding of the ten year anniversary Student Workshop in Computer Architecture. The first year it was web based only, but the following nine years we havebeen blessed (some might say cursed) with oral presentations as well. In the year 2000we gave up on the web based proceedings and printed a book instead. The combinationof both oral presentations and a printed proceeding has been such a success thatwe have continued in that manner ever since .The structure of the book has changed only slightly since last year, mainly because of fewer non-processor submissions. This proceeding is thinner than ever before —due to a declining student enrolment — but the material is as good as ever and well worth reading. With this note we leave you to it.

Architecture	**Source**	**Key Idea**	**Transistors used for Memory**
Advanced	[4]	wide-issue superscalar processor with speculative Superscalar execution and multilevel on-chip caches	910M
Super speculative Architecture	[5]	wide-issue superscalar processor with aggressive data and control speculation and multilevel on-chip caches	820M
Trace Processor	[6]	multiple distinct cores, that speculatively execute program traces, with multilevel on-chip caches	600M
Multithreaded (SMT)	[7]	wide superscalar with support for aggressive sharing among multiple threads and multilevel on-chip caches	810M
Chip Multiprocessor (cmp)	[8]	symmetric multiprocessor system with shared second level cache	450M
IA-64	[9]	VLIW architecture with support for predicated execution and long instruction bundling	600M
RAW	[10]	multiple processing tiles with reconfigurable logic and memory, interconnected through are configurable network	670M

Table 1: The billion transistor microprocessors and the number of transistors used for memory cells for each one1.We assume a billion transistor implementation for the Trace and IA-64 architecture.

2. WHAT IS COMPUTER ARCHITECTURE?

• **ARCHITECTURE:**
How are things organized and what you can do with them (functionality)
• Many different "Architectures" exist in a system
– Application/System architecture
• Structure of the application itself
– Interface to outside world (API, libraries, GUIs, etc.)
– Operating system calls
– Often appear as layers
• **For our purposes Computer architecture is** the Interface between hardware and software
• Goal:
– Build the **best** "processor"
• Today this means:
– Here's a piece of silicon
– Here are some of its properties
– Tell me what to build
• Two challenges:
1. Understand your building blocks:
• today its semiconductors
2. Understand what best means
• **Take into account design/production time**
• Takes 4-5 years to design a new high-performance processor

3. IMPLICATIONS OF IMPLEMENTATION TECHNOLOGY:

• **Caches ("bad" for IBM-XT, "a must" for Pentium 4):**
• 70's: thousands of xtors, DRAM faster than 8088 microprocessor
• nice way of slowing down your program
• 80's: depends on machine
• 90's: millions of xtors, what to do with them, DRAM much slower than processor
• a must, otherwise your ~3Ghz processor spends most of its time waiting for memory
• **#2: Technology changes rapidly making past choices often obsolete**
• **#3: Also opens up new opportunities (e.g., out-oforder**
• **Computer "Architecture": HW/SW interface**
– instruction set
– memory management and protection
– interrupts and traps
– floating-point standard (IEEE)
– Could include others: designer beware
• **µMarch (micro-Arch)**: also called organization
– number/location of functional units
– pipeline/cache configuration
– programmer transparent techniques: prefetching
• **Implementation (Hardware)**: low-level circuits

4. ROLE OF THE COMPUTER (M) ARCHITECT:

• **Architect:** Define hardware/software interface
• **µArchitect:** Define the hardware organization, usually same person as above
• **Goal:**
– 1. Determine important attributes (e.g., performance)
– 2. Design machine to maximize those attributes under constraints (e.g., cost, complexity, power).
• **How :** Study applications
Consider underlying technology
Cost
Performance
Complexity
Power
Reliability

5. OVERVIEW OF THE BILLION TRANSISTOR PROCESSORS:

Table 1 summarizes the basic features of the billion transistor implementations for the proposed architectures as presented in the corresponding references. Forth case of the Trace Processor and IA-64, descriptions of billion transistor implementations have not been presented, hence certain features are speculated.1These numbers include transistors for main memory, caches and tags. They are calculated based on information from the referenced papers. Note that CMP uses considerably less than one blithe first two architectures (Advanced Superscalar and Super speculative Architecture) have very similar characteristics. The basic idea is a wide superscalar organization with multiple execution units or functional cores that uses multi-level caching and aggressive prediction of data, control and even sequences of instructions (traces) to utilize all the available instruction level parallelism (ILP). Due their similarity, we group them together and call them "Wide Superscalar" processors in the rest of this paper.

6. THE DESKTOP/SERVER COMPUTING DOMAIN:

Current processors and computer systems are being optimized for the desktop and server domain, with SPEC'95 and TPC-C/D being the most popular bench marks This computing domain will likely be significant when the billion transistor chips will be available and similar benchmark suites will be in use. We playfully call them "SPEC'04" for technical/scientific applications and "TPC-F" for on-line transaction processing (OLTP) workloads. Table 2 presents our prediction of the performance of these processors for this domain using a grading system of"+" for strength,"_" for neutrality, and weakness .For the desktop environment, the Wide Superscalar ,Trace and Simultaneous Multithreading processors are expected to deliver the highest performance on integerSPEC'04, since out-of-order and advanced prediction techniques can utilize most of the available ILP of a single sequential program. IA-64 will perform slightly worse because VLIW compilers are not mature enough to outperform the most advanced hardware ILP techniques, which exploit run-time information. CMP and RAW will have inferior performance since desktop application shave not been shown to be highly parallelizable.CMP will still benefit from the out-of-order features of its cores. For floating point applications on the other hand, parallelism and high memory bandwidth are more important than out-of-order execution, hence SMT and CMP will have some additional advantage .For the server domain, CMP and SMT will provide the best performance, due to their ability to utilize coarse-grain parallelism even with a single chip. WideSuperscalar, Trace processor or IA-64 systems will perform worse, since current evidence is that out-of-orderexecution provides little benefit to database-like applications [11]. With the RAW architecture it is difficult .to predict any potential success of its software to map the parallelism of databases on reconfigurable logic and software controlled caches.

For any new architecture to be widely accepted, it has to be able to run a significant body of software [10].Thus, the effort needed to port existing software or develop new software is very important. The

Wide Supers caller and Trace processors have the edge, since they can run existing executables. The same holds for SMT and CMP but, in this case, high performance can be done.

Desktop

For the desktop environment, the wide superscalar, trace, and simultaneous multithreading processors should deliver the highest performance on SPECint04.These architectures use out-of-order and advanced prediction techniques to exploit most of the available instruction level parallelism (ILP) in a single sequential program. IA-64 will perform slightly worse because very long instruction word (VLIW) compilers are not mature enough to outperform the most advanced hard harelip techniques—those which exploit runtime information. The chip multiprocessor (CMP) and Raw will have inferior performance since research has shown that desktop applications are not highly parallelizable. CMP will still benefit from the out-of-order features of its cores.

Desktop

For the desktop environment, the wide superscalar, trace, and simultaneous multithreading processors should deliver the highest performance on SPECint04.These architectures use out-of-order and advanced prediction techniques to exploit most of the available instruction level parallelism (ILP) in a single sequential program. IA-64 will perform slightly worse because very long instruction word (VLIW) compilers are not mature enough to outperform the most advanced hard ware ILP techniques—those which exploit runtime information. The chip multiprocessor (CMP) and Raw will have inferior performance since research has shown that desktop applications are not highly parallelizable. CMP will still benefit from the out-of-order features of its cores.

Server

In the server domain, CMP and SMT will provide the best performance, due to their ability to use coarse-grained parallelism even with a single chip. Wide superscalar, trace, or IA-64 systems will perform worse, because current evidence indicates that out-of order for dyer execution provides only a small benefit to online transaction processing (OLTP) applications.3 For the Raw architecture, it is difficult to predict any potential success of its software to map the parallelism of databases on reconfigurable logic and software controlled caches.

Software effort

For any new architecture to gain wide acceptance ,it must run a significant body of software.4 Thus the effort needed to port existing software or develop new software is very important. In this regard, the wide superscalar, trace, and SMT processors have the edge, since they can run existing executables. The same holds for CMP, but this architecture can deliver the highest performance only if applications are rewritten in a multithreaded or parallel fashion. As the past decade has taught us, parallel programming for high-performance is neither easy nor automated

Complexity

One last issue is physical design complexity, which includes the effort devoted to the design, verification and testing of an integrated circuit.

7. A NEW TARGET FOR FUTURE COMPUTERS: PERSONAL MOBILE COMPUTING:

In the last few years, we have experienced a significant change in technology drivers. While high-end systems alone used to direct the evolution of computing, current technology is mostly driven by the low-end systems duet their large volume. Within this environment, two important trends have evolved that could change the shape of computing. The first new trend is that of multimedia applications. The recent improvements in circuits technology and innovations in software development have enabled the use of real-time media data-types like video, speech, animation and music. These dynamic data types greatly improve the usability, quality, productivity and enjoyment of personal computers [15]. Functions like 3D graphics, video and visual imagings are already included in the most popular applications and its common knowledge that their influence on computing will only increase:

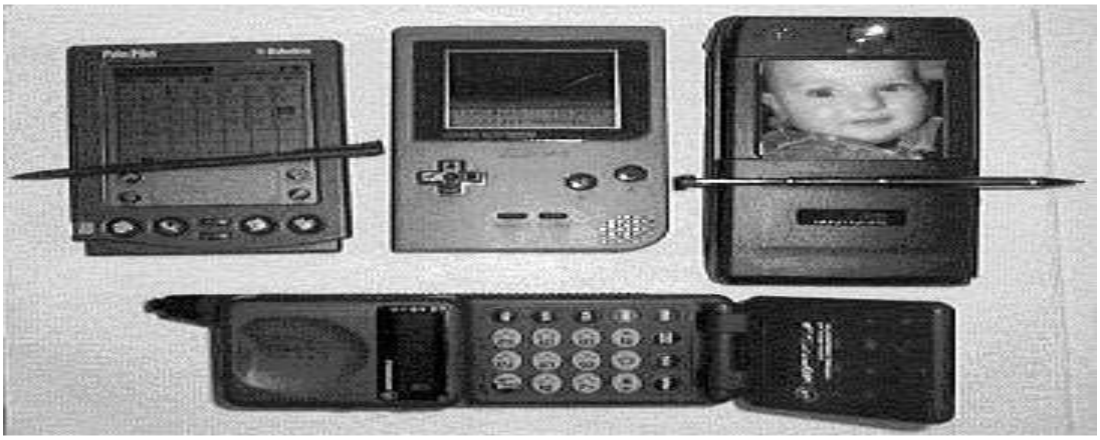

Figure 1: Personal mobile devices of the future will integrate the functions of current portable devices

like PDAs, video games, digital cameras and cellular phones "many users would like outstanding 3D graphics and multimedia" [12] "image, handwriting, and speech recognition will be other major challenges" [15]At the same time, portable computing and communication devices have gained large popularity. Inexpensive "gadgets", small enough to fit in a pocket

8. BEE3: REVITALIZING COMPUTER ARCHITECTURE:

The initial bring up and testing of the BEE3 system has been very successful. We have tested all of the subsystems at or above target operating frequency and have found no problems. The complete production BEE3 system in its 2U chassis is shown in Figure 5. The delivery of the first production run of licensed BEE3 system took place in August 2008. The BEE3 system was completed faster and better than previous academic designed multi-FPGA systems. The result is a system with better signal integrity and cheaper PCB manufacturing costs. Like all previous multi-FPGA PCBs, the BEE3 system is nowhere near the initial $5,000 target PCB price, even if all the PCB parts were free. It is also the case that the BEE3 will not replace software simulation in computer architecture, but augment the research cycle with hardware either as a simulation accelerator, software development platform, or a prototyping platform.

9. METRICS, EVALUATION ACCURACY AND VALIDATION:

Quantitative evaluations must give accurate insights about trends and behaviour. Model inaccuracies can lead to incorrect predictions and even spurious research threads that take years tore solve Yet many studies need not predict exact values for performance, power, and other metrics. Rather, they need only provide a reliable projection of how different parts of the design space perform relative to one another. Such results are especially important for exploring hypothetical architectures and targeted future technologies in which the lack of detailed design information makes absolute accuracy impossible.

10. PROBLEMS:

Some modelling assumptions are essential for achieving relative accuracy, while others add needless Complexity. The current understanding of correct abstraction levels and other important aspects of accurate models is poor. This leads to wasted effort on models and simulations that contain unnecessary detail while simultaneously lacking certain essential information. For hypothetical systems, high precision—no matter how detailed the Model can be wasted if the assumptions that underlie the detail are inappropriate or change overtime. Early-stage studies should focus on characterizations of broad parameter spaces.

11. RECOMMENDATIONS FOR VALIDATION TECHNIQUES:

The community must improve its understanding of an accurate model's essential components. This Understanding underlies the development of techniques for defining less-detailed simulations that still provide relative accuracy. It also supports the development of methods to verify that accuracy. Computer architects need better metrics as wells statistical techniques and tools that are accessible and easy to use.9 They also need metrics for new areas, including power, temperature, reliability, and quality of service. Even existing metrics, such as the energy-delay product now widely used for power aware computing, need expansion to encompass real-time computing and other design goals. The field is rife with different simulation techniques. There is little agreement on when to use certain bench marks or inputs or, despite recent work,10-12 on what configurations to model for various types of experiments and what areas require the greatest investment in modelling detail. Sound and verifiable modelling methodologies require further research.

12. PROCESSOR EVALUATION FOR MOBILE MULTIMEDIA APPLICATIONS:

Table 3 summarizes our evaluation of the billion transistor architectures with respect to personal mobile computing. The support for multimedia applications is limited in most architectures. Out-of-order techniques and caches make the delivered performance quite unpredictable for guaranteed real-time response, while hardware controlled caches also complicate support for continuous media data-types. Fine-grained parallelism is exploited by using MMX-like or reconfigurable execution units. Still, MMX-like extensions expose data alignment issues to the software and restrict the number of vector orchid elements operations per instruction, limiting this way their usability and scalability. Coarse-grained parallelism, on the other hand, is best on the Simultaneous.

13. CONCLUSIONS:

For almost two decades architecture research has been focussed on desktop or server machines. As a result of that attention, today’s microprocessors are 1000 times faster. Nevertheless, we are designing processors of the future with a heavy bias for the past. For example, the programs in the SPEC’95 suite were originally written many years ago, yet these were the main driver’s foremost papers in the special issue on billion transistor processors for 2010.

REFERENCES:

1. Semiconductor Industry Association. The National Technology Roadmap for Semiconductors.SEMATECH Inc., 1997.
2. D. Burger and D. Goodman. Billion-Transistor Architectures - Guest Editors’ Introduction. IEEEComputer, 30(9):46–48, September 1997.
3. J. Crawford and J. Huck. Motivations and Design Approach for the IA-64 64-Bit Instruction Set Architecture. In the Proceedings of the Microprocessor Forum, October 1997.
4. Y.N. Patti, S.J. Patel. Evers, D.H. Friendly, andJ. Stark. One Billion Transistors, One Uni processor ,One Chip. IEEE Computer, 30(9):51–57,September 1997.
5. M. Lipasti and L.P. Shen. Super speculative Micro architecturefor Beyond AD 2000. IEEE Computer,30(9):59–66, September 1997
6. J. Smith and S. Vajap eyam Trace Processors: Moving to Fourth Generation Micro architectures.IEEE Computer, 30(9):68–74, September 1997.
7. S.J. Eggers, J.S. Emmer, H.M. Leby, J.L. Lo, R.L.Stamm, and D.M. Tullsen. Simultaneous Multithreading: a Platform for Next-Generation Processors.IEEE MICRO, 17(5):12–19, October1997.
8. L. Hammond, B.A. Nayfeh, and K. Olukotun. ASingle-Chip Multiprocessor. IEEE Computer,30(9):79–85, September 1997.
9. E. Waingold, M. Taylor, D. Srikrishna, V. Sarkar,W. Lee, V. Lee, J. Kim, M. Frank, P. Finch,R. Barua, J. Babb, S. Amarasinghe, and A. Agarwal.Baring It All to Software: Raw Machines.IEEE Computer, 30(9):86–93, September 1997.
10. J Hennessy and D. Patterson. Computer Architecture:A QuantitativeApproach, second edition.Morgan Kaufmann, 1996.
11. K. Keeton, D.A. Patterson, Y.Q. He, and BakerW.E. Performance Characterization of the QuadPentiumPro SMP Using OLTPWorkloads. In theProceedings of the 1998 InternationalSymposium.on Computer Architecture (to appear), June 1998.
12. G. Grohoski. Challenges and Trends in ProcessorDesign: Reining in Complexity. IEEE Computer,31(1):41–42, January 1998.
13. P. Rubinfeld. Challenges and Trends in ProcessorDesign: Managing Problems in High Speed. IEEEComputer, 31(1):47–48, January 1998.
14. R. Colwell. Challenges and Trends in ProcessorDesign: Maintaining a Leading Position. IEEEComputer, 31(1):45–47, January 1998.
15. E. Killian. Challenges and Trends in ProcessorDesign: Challenges, NotRoadblocks.