From early RISC to CMPs
Perspectives on Computer Architecture

Valentina Salapura
salapura@us.ibm.com
IBM T.J. Watson Research Center
Hi, My name is Valentina Salapura

- And I am a Computer Architect
  - I work at IBM T.J. Watson research center
  - I used to be a professor of Computer Architecture at Technische Universität Wien in Vienna
  - I always wanted to be an Architect (but of a different kind of architect)
What Is Computer Architecture?

- The manner in which the components of a computer or computer system are organized and integrated.
  - Mirriam-Webster Dictionary

- 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.
  - Gene Amdahl, IBM Journal of R&D, Apr 1964

- In this book the word architecture is intended to cover all three aspects of computer design – instruction set architecture, organization, and hardware.
  - Hennessy and Patterson, *Computer Architecture: A Quantitative Approach*
Let us look at real Architecture

- **Building Architecture**
  - Sand, clay, wood, etc
  - Bricks, timber, …
  - Compose them to form buildings
  - Build cities

- **Computer Architecture**
  - Transistors, logic gates
  - ALUs, flip-flops, bit cells, crossbars, …
  - Compose them to form processors
  - Build machines
My Opinion: Architecture vs. Design

- Computer architecture applies at many levels
  - System architecture
  - Memory system architecture
  - Cache architecture
  - Network architecture
  - ...

- Architecture is the plan; design is the implementation

- Good architects understand design; good designers understand architecture

- You have to know both!
Architecture of the IBM System/360

Amdahl, Blaauw and Brooks, IBM Journal of R&D, Apr 1964

- First separation of architecture from design
  - IBM 360 instruction set architecture (ISA) completely hid the underlying technological differences between various models
  - The first true ISA designed as portable hardware-software interface

- “Before 1964, each new computer model was designed independently; the system/360 was the first computer system designed as a family of machines, all sharing the same instruction set.”
  - IBM Journal of R&D
My Opinion: Architecture vs. Software

- Architecture – software interaction at many levels
  - Applications and algorithms
  - Programming models
  - Compilers
  - Operating systems
  - …

- Architecture is the plan; software exploits it

- Good architects understand software; good software developers understand architecture

- You have to understand both!
CMOS Scaling

- Semiconductor scaling - driving force behind computer architecture research
  - Dennard’s scaling theory
  - Feature size decreases about 15% each year
  - Compute density increases about 35% each year
  - Enables higher compute speed
  - Die size increases about 10-20% each year
  - Therefore, transistor count per chip increases about 55% per year
History of Computer architecture

- Computer architecture research has changed over time
  - Build better adders
  - Build better units
  - Build better ISA
  - Build better microarchitecture

- Today
  - build better multiprocessor and better synchronization, coherence, communication, programming models and languages
Impact of IC Scaling

- Much more fits on a chip
  - Early 1980s – entire 32-bit microprocessor
  - Late 1980s – on-chip caches
  - Late 1990s – dynamic+static ILP
  - Early 2000s – on-chip router

- System balances always changing
  - Pipelining – cycle time vs. wire delay
  - Memory wall – cycle time vs. memory latency
  - I/O bottleneck – transistor count vs. pin count
  - Power wall – transistor count vs. power consumption
  - ILP wall – increasing Instruction Level Parallelism vs. performance increase
Predicting future trends for computer systems

“Computers in the future may weigh no more than 1.5 tons.”
Popular Mechanics, 1949

“There is no reason anyone would want a computer in their home.”
Ken Olsen, founder of DEC, 1977

“640K ought to be enough for anybody.”
Bill Gates, 1981

“Prediction is difficult, especially about the future”
Yogi Berra

Source: IBM GTO
pre-history  her-story
Early 80s

- LSI became VLSI

- Challenges:
  - Can we fit a whole processor on one chip?
    - Yes, if it is sufficiently simple
    - RISC processor revolution
    - Keep it simple
  - Can we design 115 thousand transistors
    - Yes, with discipline and tools
    - Mead-Conway VLSI design revolution
    - Keep it simple
Research agenda for 80s – 90s

- **Build better processor cores**
  - Riding up Moore’s Law
  - CMOS Scaling means we can use more transistors
  - Find ways to use the transistors profitably

- **Build better design methodologies and tools**
  - More complex systems require better ability to control design complexity
  - More complex systems require better performance prediction

Graph Source: www.physics.udel.edu/~watson/scen103/intel.html
Better modeling of microprocessors

- MIPS R2000 clone

- FPU & System Components model working with Siemens
  - Behavioral timing-exact modeling using a new language – VHDL
  - Exploration of other modeling approaches
    - StateCharts – David Harel

- MIPS clone
  - Modeling and design using a new approach
  - Logic synthesis and VHDL
MIPS-I processor core

- processor core as starting point
  - MIPS-I architecture
  - designed from public information
- compatible MIPS-I ISA implementation
- compatible hardware interface
- different internal operation
  - not a clone
- written in VHDL
  - described at behavioral/register transfer level
- hardware implementation using logic synthesis
MIPS I core on FPGA

- FPGAs for rapid prototyping
  - one VHDL model for FPGA and for the target technology
- Significant reduction in verification time

IEEE Transactions on VLSI, April 2001
Building better cores

- Application-specific processors
  - Can we optimize functions for specific applications?
  - Extend processor with application-specific units

- Prototyping with FPGAs
  - A nascent technology taking advantage of increased integration
  - Methodology for FPGA design with synthesis
    - Share the same design for prototype and actual fabrication
Hardware/Software Co-evaluation

- instruction set design based mostly on software benchmarks
  - minimize cycle count
  - minimize code size
- but introducing new instructions changes hardware implementation
  - cycle time
  - resource usage
- software analysis is not enough
- for a balanced view of costs and benefits ⇒ co-design
  - integrates hardware and software analysis
Joining IBM

- Projects I worked on at IBM
  - Cyclops
  - SANlite
  - Blue Gene/L
  - Blue Gene/P
Late-late nineties and on

- Diminishing returns on investment in transistors

- Massively parallel many-threaded CMPs
  - Cyclops
  - SANlite

- Application-specific accelerations
  - Networking & Protocol Processing
  - Networked Storage
SANlite: Multiprocessor subsystem for network processing

- **Self-contained, scalable multiprocessor system**
  - With its own general-purpose processors, interconnect, interfaces and memory

- **Connected to the SoC bus via a bridge**
  - Easy integration in the basic SoC structure

- **Inner structure and complexity is hidden from the SoC designer**
  - Significantly simplifies the SoC design

---

Patent US7353362: Multiprocessor subsystem in SoC with bridge between processor clusters interconnection and SoC system bus
Details of the multiprocessor subsystem

- **One or more multiple processor clusters**
  - Cyclops cluster has 8 processor cores, 32 KB shared instruction cache, local SRAM
  - Very simple processor cores
    - Single-issue, in-order execution;
    - four stages deep pipeline
- **One or more local memory banks**
  - For storing data and instructions
- **Bridge to the SoC bus**
- **One or more application-specific hardware interfaces**
  - Gigabit Ethernet, Fibre Channel, Infiniband, etc.
The 00’s

- The Power Challenge and how to build supercomputers in a power-constrained regime
  - Scale-out
  - SIMD
  - Cost-effective scaling of multicore
Key Supercomputer Challenges

- More Performance ➔ More Power
  - Systems limited by data center power supplies and cooling capacity
    - New buildings for new supercomputers
- Scaling single core performance degrades power-efficiency
  - FLOPS/W not improving from technology
- traditional supercomputer design hitting power & cost limits
Blue Gene/L: from Chip to System

The Blue Gene/L project has been supported and partially funded by the Lawrence Livermore National Laboratories on behalf of the United States Department of Energy under Lawrence Livermore National Laboratories Subcontract No. B517552.
Blue Gene/L compute ASIC

- High integration
- eDRAM
- Leverage components
- DFPU

Diagram showing the architecture of the Blue Gene/L compute ASIC, including components such as PPC440 processors, L2 caches, shared SRAM, and various communication interfaces like Ethernet and JTAG. The diagram also highlights features such as 4MB eDRAM, L3 Cache or On-Chip Memory, and components like Shared L3 Directory for eDRAM w/ECC, DDR Controller w/ECC, and various memory and communication interfaces with different bandwidths and data widths.
BlueGene/L ASIC

- IBM Cu-11 0.13µ CMOS ASIC process technology
- 11 x 11 mm die size
- 1.5/2.5V
- CBGA package, 474 pins
- Transistor count 95M
- Clock frequency 700 MHz
- Power dissipation 12.9 W
Blue Gene/L Exploring the Benefits of SIMD

- Power efficient
- Low overhead (doubles data computation without paying cost of instruction decode, issue etc.)

- UMT2K runs on 1024 nodes
- Code optimized to exploit SIMD floating point

IEEE Micro, Vol. 26 No. 5. 2006
Blue Gene/L packaging
Trends ’00: Processor performance slowing down

Microprocessors are dead!

- “Today's microprocessors will become extinct by the end of the decade, to be replaced by computers built onto a single chip.”
  - Greg Papadopoulos, Sun Microsystems, Fall processor forum 2003
Leakage Power

- **Yesterday:**
  - Power to clock latches dominant power dissipation component
  - Active power dominates

- **Today:**
  - Power consumed even if not clocking latches
  - Leakage power has become a significant component
  - Must develop means to “disconnect” unused circuits
Hardware Synthesis and Floorplanning

- **Synthesis**
  - Yesterday: reduce number of gates to make timing
  - Today: Placement Driven Synthesis (PDS)
    - Wires have delays we have to consider

- **Floorplan**
  - Yesterday: did not worry about a floorplan
  - Today: do not have architecture without a floorplan
    - Floor-plan the architecture from the beginning
    - Process variability makes timing closure hard
Improving Performance

- No longer possible by scaling alone
  - New Device Structures
  - New Device Design point
  - New Materials

Before 90’s

Since the 90’s

Beyond 2006

Source: IBM GTO
The Discontinuity

- **Then 2002**
  - Scaling drives performance
  - Performance constrained
  - Active power dominates
  - Performance tailoring in manufacturing
  - Focus on technology performance
  - Single core architectures

- **Now**
  - Architecture drives performance
    - Scaling drives down cost
  - Power constrained
  - Standby power dominates
  - Performance tailoring in design
  - Focus on system performance
  - Multi core architecture
Designing Blue Gene/P

- **Emphasis on modular design and component reuse**
- **Reuse of Blue Gene/L design components when feasible**
  - Optimized SIMD floating point unit
    - Protect investment in Blue Gene/L application tuning
  - Basic network architecture
- **Add new capabilities when profitable**
  - PPC 450 embedded core with hardware coherence support
  - New data moving engine to improve network operation
    - DMA transfers from network into memory
  - New performance monitor unit
    - Improved application analysis and tuning
Blue Gene/P - next generation node design
Multiprocessor cache management issues

- In multiprocessor systems, each processor has a separate private cache
- What happens when multiple processors cache and write the same data?
Multiprocessor cache management issues: Coherence protocol

- Every time any processor modifies data
  - Every other processor needs to check its cache
- High overhead cost
  - Cache busy snooping for significant fraction of cycles
  - Increasing penalty as more processors added
Snoop filtering of unnecessary coherence requests

Increases performance (remove unnecessary lookups, reduced cache interference)
Reduces power (and energy)
Snoop filtering improves power and performance

Actual hardware measurements
UMT2k application

V. Salapura, From early RISC to CMPs – Perspectives on Computer Architecture
CompArch Summer School on Parallel Programming and Architecture

© 2008 IBM Corporation
Blue Gene/P ASIC

- IBM Cu-08 90nm CMOS ASIC process technology
- Die size 173 mm²
- Clock frequency 850MHz
- Transistor count 208M
- Power dissipation 16W
Blue Gene/P compute card and node card

The Blue Gene/P project has been supported and partially funded by Argonne National Laboratory and the Lawrence Livermore National Laboratory on behalf of the United States Department of Energy under Subcontract No. B554331.
Blue Gene/P cabinet

- 16 node cards in a half-cabinet (midplane)
- 512 nodes (8 x 8 x 8)
- All wiring up to this level (>90%) card-level
- 1024 nodes in a cabinet
- Pictured 512 way prototype (upper cabinet)
Research and Engineering is a team sport

- Wherever you want to go, you will likely not get there alone
  - Find like-minded colleagues for technical and moral support
  - You will not get there alone, so share success with the team and treat everybody with respect

- Being part of a successful team is key
  - Enthusiasm and commitment to team success
  - Learn to understand how your team works and how to get results
Building the World’s Fastest Supercomputer: The Blue Gene/L team

BlueGene

Team

No. 1
The Blue Gene/P team – Yorktown
My support team
Career Opportunities and Challenges

- **Find an interesting subject that you are passionate about**
  - To make it work, you will spend many hours with it

- **Find interesting challenges of TOMORROW**
  - Everybody else is already working on TODAY’s problems
  - And solving today’s problems by tomorrow is usually too late
Challenges which influenced the course of Computer Architecture

- **What can we build?**
  - Technology opportunities and threats
    - VLSI & Dennard Scaling
    - Power Crisis
    - SER exposures

- **How can we build it**
  - Tools
  - Methodology
  - Controlling complexity
    - Humans
    - Tools
Pioneers of Computer Architecture (and Related Fields)
Future Pioneers: 2006 Summer School